Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Тамбовский государственный технический университет»

А.В. Майстренко, Н.В. Майстренко

ЧИСЛЕННЫЕ МЕТОДЫ РАСЧЁТА, МОДЕЛИРОВАНИЯ И ПРОЕКТИРОВАНИЯ ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ И ОБОРУДОВАНИЯ

Утверждено Учёным советом университета
в качестве учебного пособия для студентов
направлений 260100 «Продукты питания из растительного сырья»,
260200 «Продукты питания животного происхождения»,
240700 «Биотехнология» всех форм обучения

Тамбов
Издательство ФГБОУ ВПО «ТГТУ»
2011

УДК 520.88(076)
ББК В183я73
М149

Р е ц е н з е н т ы:

Доктор технических наук, профессор
заведующий кафедрой «Компьютерное и математическое моделирование» ФГБОУ ВПО «ТГУ им. Г.Р. Державина»
А.А. Арзамасцев

Доктор технических наук, профессор
заведующий кафедрой «Автоматизированное проектирование технологического оборудования» ФГБОУ ВПО «ТГТУ»
В.А. Немтинов

Майстренко, А.В.

М149 Численные методы расчёта, моделирования и проектирования технологических процессов и оборудования : учебное пособие / А.В. Майстренко, Н.В. Майстренко. – Тамбов : Изд-во ФГБОУ ВПО «ТГТУ», 2011. – 144 с. – 100 экз.
ISBN 978-5-8265-1069-8.

Содержится материал для изучения и приобретения навыков использования численных методов и методов оптимизации.

Предназначено для студентов направлений 260100 «Продукты питания из растительного сырья», 260200 «Продукты питания животного происхождения», 240700 «Биотехнология», 230100 «Информатика и вычислительная техника» всех форм обучения, также может быть полезным для студентов других направлений и специальностей, аспирантов и преподавателей, изучающих численные методы для их последующего применения при математическом моделировании и проектировании различных объектов, процессов и явлений.

УДК 520.88(076)
ББК В183я73

ISBN 978-5-8265-1069-8

ВВЕДЕНИЕ

В настоящее время методы вычислительной математики широко используются при решении различного рода задач в науке, технике, производстве. Однако инженеру, не имеющему специального математического образования, часто бывает трудно как сделать правильный выбор методов решения стоящих перед ним математических задач, так и в случае обоснованного выбора грамотно реализовать выбранный метод. Среди множества математических задач, с которыми приходится сталкиваться инженеру в своей практике, можно выделить:

– решение алгебраических и трансцендентных уравнений и их систем;
– решение определённых интегралов;
– решение обыкновенных дифференциальных уравнений, дифференциальных уравнений в частных производных и их систем;
– обработка массивов числовых данных;
– решение задач оптимизации.

В данном пособии приводятся численные методы решения наиболее часто встречающихся в инженерной практике математических задач, в ряде случаев приводятся примеры алгоритмов их реализации и использования.

1. ПРИБЛИЖЁННЫЕ ЧИСЛА И ОЦЕНКА ПОГРЕШНОСТЕЙ

Приближённым числом x называется число, незначительно отличающееся от точного x^* и заменяющее последнее в вычислениях. Если известно, что $x < x^*$, то x называется приближённым значением числа x^* по недостатку; если же $x > x^*$, то – по избытку.

Под *ошибкой*, или *погрешностью*, Δx приближённого числа x обычно понимается разность между соответствующим точным числом x^* и данным приближённым, т.е.

$$\Delta x = x^* - x\,.$$

Как видно из равенства, ошибка Δx может быть как положительной, если $\Delta x > 0$, так и отрицательной, если $\Delta x < 0$.

Во многих случаях знак ошибки неизвестен. Тогда целесообразно пользоваться *абсолютной погрешностью приближённого числа*

$$\Delta = \left|\Delta x\right|.$$

Определение: *Абсолютной погрешностью* Δ *приближённого числа* x *называется абсолютная величина разности между соответствующим точным числом* x^* *и числом* x*, т.е.*

$$\Delta = \left|x^* - x\right|.$$

В некоторых случаях точное число x^* бывает неизвестным, что сказывается на невозможности определения абсолютной погрешности. В таких ситуациях вместо неизвестной теоретической абсолютной погрешности вводится её оценка сверху так называемая *предельная абсолютная погрешность*.

Определение: Под *предельной абсолютной погрешностью* $\boldsymbol{\Delta_x}$ *приближённого числа принимается всякое число, не меньшее абсолютной погрешности данного числа*

$$\Delta = \left|x^* - x\right| \le \Delta_x\,.$$

Откуда следует, что точное число x^* заключено в границах:

$$x - \Delta_x \le x^* \le x + \Delta_x\,.$$

Следовательно, $\underline{x} = x - \Delta_x$ есть приближение числа x^* по недостатку, а $\overline{x} = x - \Delta_x$ − приближение числа x^* по избытку.

Однако абсолютная погрешность недостаточна для характеристики точности измерения или вычисления. В этих ситуациях пользуются *относительной погрешностью*.

Определение: *Относительной погрешностью* δ *приближённого числа x называется отношение абсолютной погрешности* Δ *этого числа к модулю соответствующего точного числа* $x^*(x \neq 0)^*$, *т.е.* $\delta = \dfrac{\Delta}{x^*}$.

Так же как и для абсолютной погрешности, для относительной погрешности существует понятие *предельной относительной погрешности.*

Определение: *Предельной относительной погрешностью* δ_x *данного приближённого числа x называется всякое число, не меньшее относительной погрешности этого числа*:

$$\delta = \frac{\Delta}{|x^*|} \leq \delta_x .$$

1.1. ОСНОВНЫЕ ИСТОЧНИКИ ПОГРЕШНОСТЕЙ

Погрешности, встречающиеся в математических и вычислительных задачах, могут быть разбиты на следующие группы:

1. Погрешности, связанные с самой постановкой математической задачи и возникающие за счёт неточности исходной информации. Такие погрешности называются *неустранимыми погрешностями или погрешностями задачи.*

2. Погрешности, возникающие при численном решении поставленной математической задачи, называются *погрешностями метода*. Они появляются при использовании того или иного численного метода, дающего однако лишь некоторое приближение к точному решению конкретной задачи. В то же время разумный выбор метода позволяет, как правило, получить решение с нужной для применения точностью.

3. Погрешности, возникающие за счёт неточностей самих вычислений, называют *вычислительными погрешностями* или *погрешностями округления.* Их появление связано с необходимостью округления бесконечных десятичных дробей, с действиями над приближёнными числами. Сюда же относятся и погрешности, возникающие при округлении результатов расчётов средствами вычислительной техники.

Таким образом, *полная погрешность решения*, т.е. разность истинного решения исходной задачи и практически полученного конечного результата, будет складываться из неустранимой погрешности, погрешности метода и вычислительной погрешности.

1.2. ЗНАЧАЩАЯ ЦИФРА. ЧИСЛО ВЕРНЫХ ЗНАКОВ

Значащей цифрой приближённого числа называется всякая цифра в его десятичном изображении, отличная от нуля, и нуль, если он содержится между значащими цифрами или является представителем сохранённого десятичного разряда. Все остальные нули, входящие в состав приближённого числа и служащие лишь для обозначения его десятичных разрядов, не причисляются к значащим цифрам.

В случае если последний ноль в данном числе не является значащей цифрой, то это число должно быть записано в виде 0,003087.

При написании больших чисел нули справа могут служить как для обозначения значащих цифр, так и для определения разрядов остальных цифр. Поэтому при обычной записи могут возникнуть неясности. Например, о числе 32800 можно сказать, что оно имеет не менее трёх значащих цифр. Этой неопределённости можно избежать, если записать число в виде $3{,}28\cdot10^4$ или $3{,}2800\cdot10^4$. Тогда мы можем сказать, что первое число имеет три значащих цифры, а второе − пять.

Помимо понятия значащих цифр, вычислительная математика оперирует с понятием верных знаков приближённого числа.

Определение: *Говорят, что n первых значащих цифр* (*десятичных знаков*) *приближённого числа являются* ***верными****, если абсолютная погрешность этого числа не превышает половины единицы разряда, выражаемого значащей цифрой, считая слева направо, т.е. если для приближённого числа*

$$x = x_m \cdot 10^m + x_{m-1} \cdot 10^{m-1} + \ldots + x_{m-n+1} \cdot 10^{m-n+1} + \ldots \, (x_m \neq 0),$$

делящего точное число x^* *известно, что* $\Delta = \left|x^* - x\right| \le \frac{1}{2} \cdot 10^{m-n+1}$, *где* m *− старший десятичный разряд, то по определению первые* n *цифр* $a_m, a_{m-1}, \ldots, a_{m-n+1}$ *этого числа являются верными.*

***Пример*:**

$$38{,}76 = 3\cdot10^1 + 8\cdot10^0 + 7\cdot10^{-1} + 6\cdot10^{-2},$$

$K = 4$ − количество разрядов (общее),

$m = 1$ − старший разряд.

Для точного числа $x^* = 38{,}76$ приближённое число $x = 38{,}80$ является приближённым с тремя верными знаками ($n = 3$), так как

$$\Delta = \left|x^* - x\right| = \left|38{,}76 - 38{,}80\right| = 0{,}04 < \frac{1}{2} \cdot 10^{1-3+1} = \frac{1}{2} \cdot 10^{-1} = 0{,}05.$$

В большинстве случаев можно утверждать, что верные знаки приближённого числа совпадают с соответствующими числами точного числа.

Между количеством верных знаков приближённого числа и его относительной погрешностью существует связь, определяемая следующей теоремой.

Теорема: *Если положительное приближённое число x имеет n верных десятичных знаков, то относительная погрешность* δ *этого числа не превосходит* 10^{1-n}, *делённую на первую значащую цифру данного числа, т.е.* $\delta \le \frac{10^{1-n}}{x_m}$, *где* x_m *– первая значащая цифра числа.*

***Следствие 1*:** За предельную относительную погрешность числа *x* можно принять:

$$\delta_x \le \frac{1}{x_m} \cdot 10^{1-n}.$$

***Следствие 2*:** Если число имеет больше двух верных знаков, т.е. $n \ge 2$, то справедлива формула: $\delta_x = \frac{1}{2x_m} \cdot 10^{1-n}$.

1.3. ОКРУГЛЕНИЕ ЧИСЕЛ

При решении многих задач, в частности задач с применением численных методов решения, часто возникает необходимость в округлении промежуточных или конечных результатов расчёта. В этом случае пользуются следующим правилом:

Чтобы округлить число до *n* значащих цифр, отбрасывают все его цифры, стоящие справа от *n*-й значащей цифры, или, если это нужно для сохранения разрядов, заменяют нулями. При этом:

1) если первая из отброшенных цифр меньше 5, то оставшиеся десятичные знаки сохраняются без изменения;

2) если первая их отброшенных цифр больше 5, то к последней оставшейся цифре прибавляется единица;

3) если первая из отброшенных цифр равна 5 и среди остальных отброшенных цифр имеются ненулевые, то последняя оставшаяся цифра увеличивается на единицу;

4) если же первая из отброшенных цифр равна 5 и все остальные отброшенные цифры являются нулями, то последняя оставшаяся цифра сохраняется неизменяемой, если она чётная, и увеличивается на единицу, если она нечётная:

$$38{,}76500 \approx 38{,}76;$$
$$43{,}23500 \approx 43{,}24.$$

Точность приближённого числа зависит не от количества значащих цифр, а от количества верных значащих цифр. В этих случаях, когда приближённое число содержит излишнее количество неверных

значащих цифр, прибегают к округлению. Обычно руководствуются следующим практическим правилом: при выполнении приближённых вычислений число значащих цифр промежуточных результатов не должно превышать числа верных цифр более чем на одну или две единицы. Окончательный результат может содержать не более чем одну излишнюю значащую цифру, по сравнению с верными. Если при этом абсолютная погрешность не превышает двух единиц последнего сохранённого разряда, то излишняя цифра называется *сомнительной.*

1.4. ПОГРЕШНОСТЬ АРИФМЕТИЧЕСКИХ ВЫРАЖЕНИЙ

Выполняя различные арифметические операции над приближёнными числами, как правило, возникает вопрос: насколько верным является полученный результат? Можем ли мы ему доверять? Ответить на него можно, лишь вычислив погрешность результата. Для этого руководствуются следующими правилами:

1.4.1. Погрешность суммы (разности)

Абсолютная погрешность алгебраической суммы нескольких приближённых чисел не превышает суммы абсолютных погрешностей этих чисел.

$$|u| \le |\Delta x_1| + |\Delta x_2| + ... + |\Delta x_n|, \text{ где } u = x_1 + x_2 + ... + x_n .$$

За предельную абсолютную погрешность алгебраической суммы можно принять сумму предельных абсолютных погрешностей слагаемых $\Delta_u = \Delta_{x_1} + \Delta_{x_2} + ... + \Delta_{x_n}$.

Это же правило верно и для вычисления погрешности разности (предельная абсолютная погрешность разности равна сумме предельных абсолютных погрешностей уменьшаемого и вычитаемого, т.е. $\Delta_{x_1 - x_2} = \Delta_{x_1} - \Delta_{x_2}$).

Предельная же относительная погрешность суммы (разности) не превышает наибольшей из предельных относительных погрешностей слагаемых (уменьшаемого и вычитаемого), т.е. $\delta \le \max(\delta_{x_1}, \delta_{x_2}, ..., \delta_{x_n})$.

1.4.2. Погрешность произведения

Относительная погрешность произведения нескольких приближённых чисел, отличных от нуля, не превышает суммы относительных погрешностей этих чисел.

$$\delta(x_1 \cdot x_2 \cdot ... \cdot x_n) \le \delta(x_1) + \delta(x_2) + ... + \delta(x_n) .$$

Очевидно, что предельная относительная погрешность произведения равна сумме предельных относительных погрешностей сомножителей, т.е.

$$\delta_{x_1 \cdot x_2 \cdot \ldots \cdot x_n} = \delta_{x_1} + \delta_{x_2} + \ldots + \delta_{x_n} .$$

В частном случае при умножении приближённого числа x на точный множитель k относительная предельная погрешность не изменяется, а абсолютная предельная погрешность увеличивается в $|k|$ раз.

1.4.3. Погрешность частного

Относительная погрешность частного не превышает суммы относительных погрешностей делимого и делителя, а предельная относительная погрешность равна сумме предельных относительных погрешностей делимого и делителя, т.е.

$$\delta(x_1 / x_2) \le \delta(x_1) + \delta(x_2)$$

$$\text{и } \delta_{x_1 / x_2} = \delta_{x_1} + \delta_{x_2} .$$

1.4.4. Относительная погрешность степени

Предельная относительная погрешность m-й степени числа x в m раз больше предельной относительной погрешности самого числа: $\delta_{x^m} = m\delta_x$.

1.4.5. Относительная погрешность корня

Предельная относительная погрешность корня m-й степени в m раз меньше предельной относительной погрешности подкоренного числа: $\delta_{\sqrt[m]{x}} = \frac{1}{m}\delta_x$.

2. ЧИСЛЕННОЕ РЕШЕНИЕ АЛГЕБРАИЧЕСКИХ И ТРАНСЦЕНДЕНТНЫХ УРАВНЕНИЙ

Обычно нелинейные уравнения делят на трансцендентные и алгебраические.

Трансцендентными называются нелинейные уравнения, содержащие тригонометрические или другие специальные функции, например $\lg x$ или e^x.

Для решения нелинейных уравнений широко используются методы, позволяющие получать приближённое решение с любой заданной степенью точности (итерационные методы).

Пусть дано уравнение:

$$f(x)=0, \tag{2.1}$$

где функция $f(x)$ определена и непрерывна в некотором конечном или бесконечном интервале $[a, b]$. В некоторых случаях от функции $f(x)$ требуется существование и непрерывность первой $f'(x)$ и второй $f''(x)$ производных.

Всякое значение x^*, обращающее функцию $f(x)$ в ноль, т.е. такое, что $f(x^*)=0$, называется корнем уравнения (2.1) или нулем функции $f(x)$.

Задача нахождения корней уравнения (2.1) обычно решается в два этапа:

I. отделение корней, т.е. установление отрезка $[a, b]$, принадлежащего области определения функции $f(x)$, на котором имеется один и только один корень уравнения $f(x)=0$;

II. уточнение приближённых корней, т.е. доведение их до заданной степени точности.

При отделении корней уравнения можно пользоваться следующей теоремой математического анализа:

Теорема 1: *Если непрерывная функция $f(x)$ принимает значения разных знаков на концах отрезка $[a,b]$, т.е. $f(a)f(b)<0$, то внутри этого отрезка содержится по меньшей мере один корень уравнения $f(x)=0$, т.е. найдётся хотя бы одно число $x^*\in[a, b]$, такое, что $f(x^*)=0$* (рис. 2.1).

Если же при этом функция $f(x)$ имеет первую производную $f'(x)$, которая не меняет своего знака внутри интервала $[a, b]$, т.е. $f'(x)>0$ или $f'(x)<0$ при $a\le x\le b$, то корень x^* будет единственным.

В общем случае, если $f(x)$ является аналитической функцией переменной x на отрезке $[a, b]$ и, если на концах отрезка $[a, b]$ функция $f(x)$

принимает значение разных знаков, то между a и b имеется нечётное число корней уравнения $f(x)=0$; если же на концах отрезка $[a, b]$ функция $f(x)$ принимает значение одинаковых знаков, то между a и b или нет корней этого уравнения, или имеется их чётное число (учитывая кратность корней).

Определение корня можно производить аналитически или графически.

Аналитический метод определения корней заключается в следующем. Вначале определяются знаки функции $f(x)$ в граничных точках $x = a$ и $x = b$ области её существования. Затем определяются знаки функции $f(x)$ в ряде промежуточных точек $x = c_1, c_2, ...,$ выбор которых учитывает особенности функции $f(x)$. Если окажется, что $f(c_k)f(c_{k1})<0$, то, согласно теореме 1, в интервале $[c_k, c_{k1}]$ имеется хотя бы один корень уравнения $f(x)=0$. Остаётся лишь убедиться в единственности корня на этом интервале.

Задача отделения корней может несколько упроститься, если функция $f(x)$ имеет непрерывную первую производную $f'(x)$ на интервале $[a, b]$. В этом случае вначале определяются интервалы монотонности функции $f(x)$, т.е. интервалы, на которых $f'(x)$ не меняет своего знака.

Для этого решается уравнение $f'(x)=0$, а затем вычисляются значения $f(x)$ в точках перемены знака производной $f'(x)$ и определяются интервалы, на которых функция $f(x)$ имеет разные знаки.

Графически корни уравнения (2.1) можно определить, построив график функции $y = f(x)$, определив абсциссы точек пересечения графика функции $y = f(x)$ с осью $0X$ (рис. 2.1). Если уравнение (2.1) не имеет близких между собой корней, то этим способом его действительные корни легко определяются (для определения комплексных корней нелинейного уравнения $f(x)=0$ и их численного решения существуют специальные методы, которые в данном пособии рассматриваться не будут).

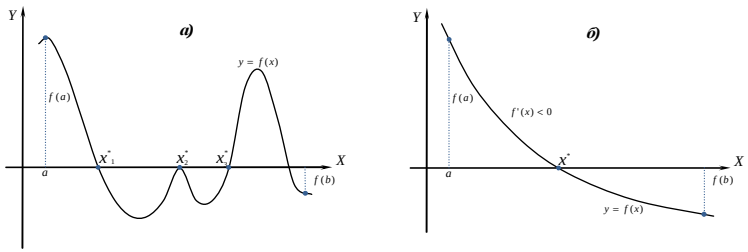
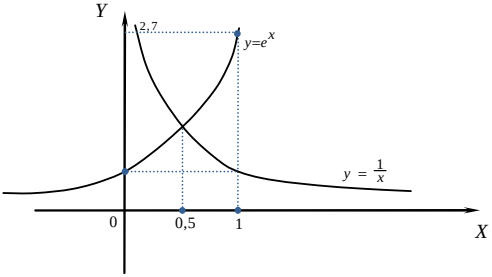


Рис. 2.1. Иллюстрация наличия корней уравнения по графику функции



$$xe^x = 1 \Rightarrow e^x = \frac{1}{x} \Rightarrow \begin{cases} y = \dfrac{1}{x}; \\ y = e^x. \end{cases}$$

Рис. 2.2. Пример графиков функций для уравнения (2.2)

На практике часто бывает выгодно уравнение (2.1) заменить равносильным[1] ему уравнением

$$\varphi(x) = \psi(x), \tag{2.2}$$

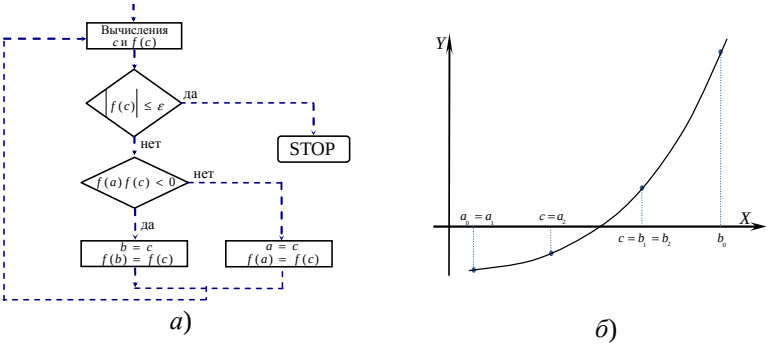
где функции $\varphi(x)$ и $\psi(x)$ – более простые, чем функция $f(x)$. Тогда построив графики функций $y = \varphi(x)$ и $y = \psi(x)$, искомые корни получим как абсциссы точек пересечения этих графиков (рис. 2.2.).

Для решения алгебраических и трансцендентных уравнений вида $f(x) = 0$ разработано много различных итерационных методов. Сущность этих методов заключается в следующем. Пусть известна достаточно малая область, в которой содержится единственный корень $x = x^*$ этого уравнения. В этой области выбирается точка x_0 – начальное приближение корня, – достаточно близкая к искомому корню $x = x^*$. Собственно говоря, любую точку c интервала $[a, b]$, отделяющего корень, можно считать приближённым значением корня, так как разность между истинным значением корня x^* и его приближённым значением c ограничена величиной отрезка $[a, b]$, т.е. $|x^* - c| < |b - a|$. Далее с помощью некоторого рекуррентного соотношения строится последовательность $x_1, x_2, \ldots, x_k, \ldots$, сходящаяся к $x = x^*$. Сходимость последовательности обеспечивается соответствующим выбором рекуррентного соотношения и начального приближения x_0.

2.1. МЕТОД ПОЛОВИННОГО ДЕЛЕНИЯ

Пусть дано уравнение $f(x) = 0$, где функция $f(x)$ непрерывна на $[a, b]$ и $f(a)f(b) < 0$. Для нахождения корня этого уравнения по формуле $c = \dfrac{a+b}{2}$ вычисляется среднее значение x в интервале $[a, b]$ и находится соответствующее ему значение функции $f(c)$.

[1] Два уравнения называются равносильными, если они имеют одинаковые корни.



а) *б*)

Рис. 2.3. Блок-схема (*а*) и графическая иллюстрация (*б*) метода половинного деления

Если $f(c)=0$, то $x^* = c = \dfrac{a+b}{2}$ является корнем уравнения. Если $f(c)\neq 0$, то выбирают ту из половин $[a, c]$ или $[c, b]$, на концах которой функция $f(x)$ имеет противоположные знаки (рис. 2.3). В результате интервал, в котором заключено значение корня, сужается. Новый суженный отрезок $[a_1, b_1]$ снова делим пополам и проводим то же рассмотрение, и т.д. В итоге получаем на каком-то этапе или точный корень уравнения $f(x)=0$, если $f(c_k)$ достаточно близко к нулю либо $|b_k - a_k|$ достаточно мало, или бесконечную последовательность вложенных друг в друга отрезков $[a_1, b_1]$, $[a_2, b_2]$, … $[a_k, b_k]$, … таких, что $f(a_k)f(b_k)<0$ (k = 1, 2, …) и $b_k - a_k = \dfrac{1}{2^k}(b-a) \Rightarrow \;\; k \geq \log_2 \dfrac{b-a}{\varepsilon}$ (число вычислений для достижения точности ε).

Хотя метод половинного деления не обладает высокой вычислительной эффективностью, с увеличением числа итераций он обеспечивает получение всё более точного приближённого значения корня и может использоваться для грубого нахождения корня данного уравнения.

2.2. МЕТОД ХОРД

При решении уравнения $f(x)=0$, где функция $f(x)$ непрерывна на $[a, b]$ и $f(a)f(b)<0$, более естественно делить отрезок $[a, b]$ в отношении $f(a)/f(b)$, а не пополам, как в предыдущем методе. Такое деление можно произвести, если провести через точки $A(a, f(a))$ и $B(b, f(b))$ хорду AB, стягивающую концы дуги графика функции $y = f(x)$, и в качестве приближённого значения выбрать число c, являющееся абсциссой точки пересечения хорды AB с осью OX (рис. 2.4). Тогда для определения значения c можно записать уравнение хорды, как уравнение прямой, проходящей через точки A и B.

$$\frac{x-a}{b-a}=\frac{y-f(a)}{f(b)-f(a)}.$$

Отсюда, полагая $x = c$ и $y = 0$, получим

$$c=a-\frac{f(a)(b-a)}{f(b)-f(a)} \tag{2.3}$$

Аналогичную формулу можно записать, если прикрепить хорду к точке b:

$$c=b-\frac{f(b)(b-a)}{f(b)-f(a)}. \tag{2.4}$$

Значение c, принимаемое за первое приближение к исходному корню, обозначим через x_1, т.е. $x_1 = c$. Эта точка разделит отрезок $[a, b]$ на два интервала $[a, x_1]$ и $[x_1, b]$, в одном из которых находится искомый корень.

Новый отрезок, отделяющий корень, можно определить, пользуясь следующими правилами:

1) точка x_1 находится со стороны вогнутости кривой $y = f(x)$;

2) приближённое значение x_1 лежит по ту сторону от истинного корня x^*, на которой функция $f(x)$ имеет знак, противоположный знаку её второй производной $f''(x)$, а знак $f'(x)$ совпадает со знаком $f''(x)$;

3) неподвижным остаётся тот конец интервала, а, следовательно, и хорды, для которого знак функции совпадает со знаком её второй производной $f''(x)$, а знак $f(x)$ – нет.

В общем случае возможны следующие варианты поведения функции $f(x)$ (рис. 2.5).

Повторяя многократно описанную процедуру построения хорды, получим последовательность значений x_2, x_3, … x_k, которая будет стремиться к истинному корню x^*. При этом для вычисления значений x_k можно пользоваться следующими итерационными формулами:

$$x_{k+1}=x_k-\frac{f(x_k)(b-x_k)}{f(b)-f(x_k)},\ \text{если}\ f'(x)f''(x)>0\ \text{или}\ f(b)f''(x)>0;$$

$$x_{k+1}=x_k-\frac{f(x_k)(x_k-a)}{f(x_k)-f(a)},\ \text{если}\ f'(x)f''(x)<0\ \text{или}\ f(a)f''(x)>0.$$

Процедура вычисления корня уравнения $f(x)=0$ прекращается, когда оценка полученного приближения x_k удовлетворяет заданной точности. Для упрощения вычислений обычно задают некоторое, достаточно малое число $\varepsilon > 0$ и прекращают вычисления, когда разность между двумя последующими приближениями становится меньше δ, т.е.

$$|x_{k+1}-x_k|\le\delta=\frac{\varepsilon m_1}{M_1-m_1};\quad \left(m_1=\min_{x\in[a,\,b]}|f'(x)|;\ M_1=\max_{x\in[a,\,b]}|f'(x)|\right).$$

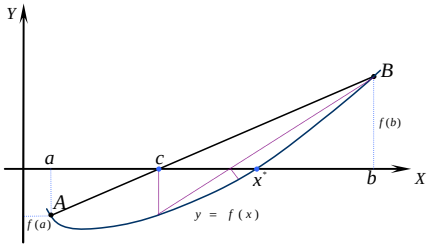


Рис. 2.4. Иллюстрация метода хорд

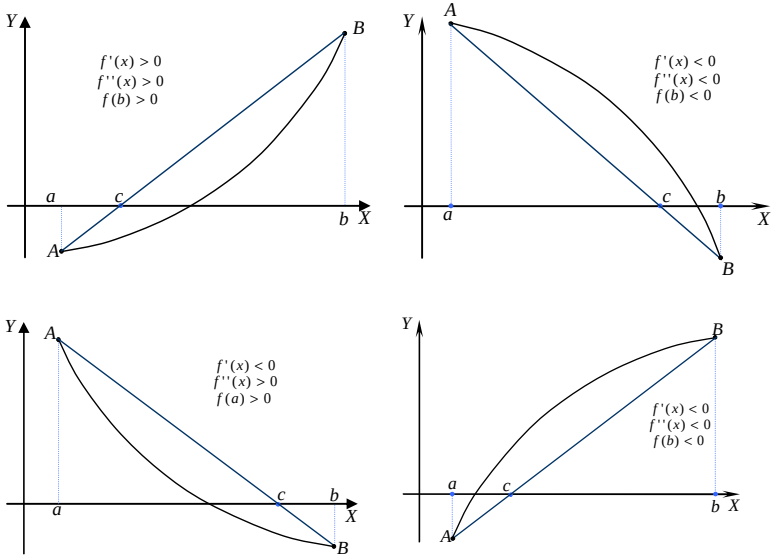


Рис. 2.5. Варианты поведения функции *f*(*x*) для метода хорд

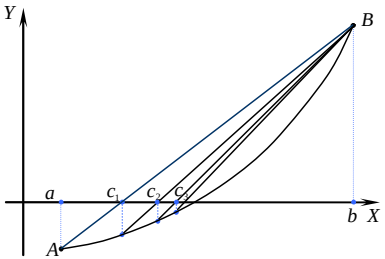


Рис. 2.6. Иллюстрация итерационного процесса нахождения корня уравнения методом хорд

Число x_{k+1} принимают за приближённое значение корня x^*.

В том случае, если вторая производная $f''(x)$ на отрезке $[a, b]$ непостоянна, для определения приближённого значения корня можно использовать формулы (2.3) или (2.4), а последующее отделение корня производить, сравнивая знак $f(c)$ со знаками $f(a)$ или $f(b)$. Блок-схема такого варианта метода хорд будет схожа с блок-схемой метода половинного деления, а критерием остановки поиска будет являться близость к нулю значения $f(c)$ (рис. 2.6).

2.3. МЕТОД КАСАТЕЛЬНЫХ (НЬЮТОНА)

Пусть корень x^* уравнения $f(x)=0$ отделён на отрезке $[a, b]$, причём $f'(x)$ и $f''(x)$ непрерывны и сохраняют определённые знаки на всём интервале $[a, b]$. В основе метода Ньютона лежит разложение функции $f(x)$ в ряд Тейлора:

$$f(x+h)= f(x)+hf'^{(x)}+\frac{h^2}{2}f''(x)+...,$$

где h − достаточно малая величина.

Отбросив члены, содержащие h во второй и более высоких степенях, и, предполагая, что переход от x к $\hat{x}=x+h$ приближает значение функции к нулю, так как $f(x+h)=0$, получим $\hat{x}=x-\frac{f(x)}{f'(x)}$.

Значение $\hat{x}$ соответствует точке, в которой касательная к кривой в точке x пересекает ось OX (рис. 2.7). Таким образом, геометрический метод Ньютона эквивалентен замене небольшой дуги кривой $y=f(x)$ касательной, проведённой в некоторой точке кривой. Действительно уравнение касательной в точке $(\tilde{x}, f(\tilde{x}))$ имеет вид $y-f(\tilde{x})=f'(\tilde{x})(x-\tilde{x})$.

И если положить $y = 0$, $x = \hat{x}$, то $\hat{x}=\tilde{x}-\frac{f(\tilde{x})}{f'(\tilde{x})}$.

Возвращаясь к поставленной задаче, построим касательную к функции $y=f(x)$ в точке, ограничивающей отрезок локализации корня x^*, например в точке $B(b, f(b))$. Тогда найденное значение $c=b-\frac{f(b)}{f'(b)}$ будет первым приближением к корню x^*. Обозначим его через x_1, т.е. $x_1 = c$. Очевидно, что точка x_1 будет находиться со стороны выпуклости кривой $y=f(x)$. Проведём в точке $(x_1, f(x_1))$ новую касательную к кривой $y=f(x)$, получив таким образом следующее приближение x_2 к истинному корню x^* (рис. 2.8). Этот процесс про-

должается пока значение $f(x_k)$ не станет меньше заданной точности, либо пока разность между двумя последними приближениями не станет меньше заданного числа

$$\left|x_{k+1} - x_k\right| \le \delta = \sqrt{\frac{2m_1\varepsilon}{M_2}},$$

где ε – точность, с которой требуется найти корень; $m_1 = \min\limits_{x\in[a,\,b]} \left|f'(x)\right|$;

$M_2 = \max\limits_{x\in[a,\,b]} \left|f''(x)\right|$.

Для вычисления x_k можно пользоваться следующей итерационной формулой:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)},\ k = 0, 1, 2, \ldots.$$

Совершенно очевидно, что быстрота сходимости метода Ньютона в большой мере зависит от удачного выбора исходной точки. Если в процессе итераций тангенс угла наклона $f'(x)$ обращается в ноль, то применение метода осложняется. Не будет достаточно эффективным использование метода и в случае слишком большого $f''(x)$.

Для удачного выбора начального приближения x_0 следует пользоваться теоремой 2.

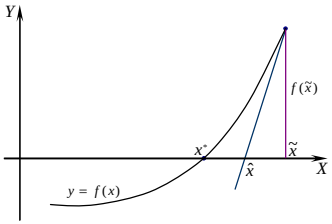


Рис. 2.7. Определение точки пересечения касательной к кривой с осью 0*X*

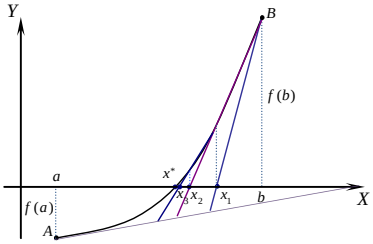


Рис. 2.8. Иллюстрация итерационного процесса нахождения корня уравнения методом Ньютона (касательных)

Теорема 2: *Если* $f(a)\,f(b) < 0$, *причём* $f'(x)$ и $f''(x)$ *отличны от нуля и сохраняют определённые знаки при* $a \le x \le b$, *то исходя из начального приближения* $x_0 \in [a, b]$, *удовлетворяющего неравенству* $f(x_0)\,f''(x_0) > 0$, *можно вычислить методом Ньютона*

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \quad k = 0, 1, 2, \ldots,$$

единственный корень x^* *уравнения* $y = f(x)$ *с любой степенью точности.*

Таким образом, в качестве исходной точки x_0 выбирается тот конец интервала $[a, b]$, которому отвечает ордината того же знака, что и знак $f''(x)$.

2.4. МОДИФИЦИРОВАННЫЙ МЕТОД НЬЮТОНА

Если производная $f'(x)$ мало изменяется на отрезке $[a, b]$, то в формуле метода Ньютона можно принять: $f'(x_{k+1}) \approx f'(x_0)$.

Тогда для получения последовательных приближений истинного корня x_0 уравнения $f(x) = 0$ можно использовать формулу:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_0)}, \quad k = 0, 1, 2, \ldots.$$

Геометрически это означает, что мы заменяем касательные в точках $(x_k,\ f(x_k))$ прямыми, параллельными касательной в точке $(x_0,\ f(x_0))$ (рис. 2.9).

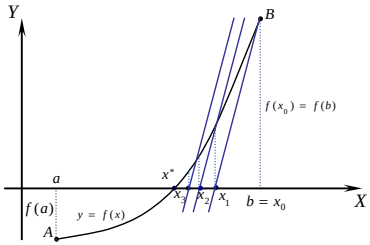


Рис. 2.9. Иллюстрация итерационного процесса нахождения корня уравнения модифицированным методом Ньютона

2.5. МЕТОД СЕКУЩИХ

Один из недостатков метода Ньютона состоит в том, что, пользуясь им, приходится дифференцировать функцию $f(x)$. Если нахождение производной затруднено, то можно воспользоваться некоторым приближением, которое и составляет основу метода секущих. Для этого производную $f'(x)$, используемую в методе Ньютона, заменяют разностью последовательных значений функции, отнесённой к разности значений аргумента

$$f'(x) = \frac{df(x_k)}{dx_k} \approx \frac{\Delta f(x_k)}{\Delta x_k} = \frac{f(x) - f(x_{k-1})}{x_k - x_{k-1}}.$$

В результате получается следующая итерационная формула:

$$x_{k+1} = x_k - \frac{f(x_k)(x_k - x_{k-1})}{f(x_k) - f(x_{k-1})},\ k = 1, 2, \ldots .$$

Для выбора начального приближения x_0 можно пользоваться тем же правилом, что и в методе Ньютона, а значение x_1 определить как $x_1 = x_0 + h$, где h – некоторая малая величина. В целом же схема метода секущих такая же, что и метода Ньютона.

Геометрически метод секущих означает, что через рассматриваемую точку будет проводиться не касательная, а секущая (рис. 2.10).

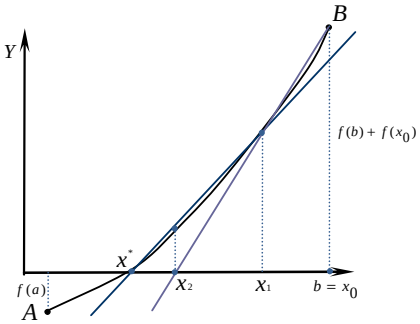


Рис. 2.10. Иллюстрация итерационного процесса нахождения корня уравнения методом секущих

2.6. КОМБИНИРОВАННЫЙ МЕТОД ХОРД И КАСАТЕЛЬНЫХ

Пусть $f(a)f(b) < 0$, а $f'(x)$ и $f''(x)$ сохраняют постоянные знаки на отрезке $[a, b]$. Соединяя метод хорд и метод Ньютона, получаем метод, на каждом этапе которого находим два значения точного корня x^* уравнения $f(x) = 0$: значение по недостатку $\underline{x}$ и значение по избытку $\overline{x}$.

При отыскании корней комбинированным методом за начальные приближения значений по недостатку и по избытку следует принять следующие значения: $\underline{x_0} = a$, $\overline{x_0} = b$. Здесь так же, как и в методе хорд, возможны четыре случая, различающиеся знаками функции $f(x)$, её первой и второй производных $f'(x)$ и $f''(x)$. Для расчётов комбинированным методом используются формулы:

1) если $f'(x)f''(x) > 0$ или $f(b)f''(x) > 0$, то

$$\underline{x}_{k+1} = \underline{x}_k - \frac{f(\underline{x}_k)(\overline{x}_k - \underline{x}_k)}{f(\overline{x}_k) - f(\underline{x}_k)},$$

$$\overline{x}_{k+1} = \overline{x}_k - \frac{f(\overline{x}_k)}{f'(\overline{x}_k)};$$

2) если $f'(x)f''(x) < 0$ или $f(a)f''(x) > 0$, то $\underline{x}_{k+1} = \underline{x}_k - \frac{f(\underline{x}_k)}{f'(\underline{x}_k)}$,

$$\overline{x}_{k+1} = \underline{x}_k - \frac{f(\underline{x}_k)(\overline{x}_k - \underline{x}_k)}{f(\overline{x}_k) - f(\underline{x}_k)}.$$

Легко видеть, что истинное значение корня x^* лежит между $\underline{x}_{k+1}$ и $\overline{x}_{k+1}$, т.е. $\underline{x}_{k+1} < x^* < \overline{x}_{k+1}$ и $0 < x^* - \underline{x}_{k+1} < \overline{x}_{k+1} - \underline{x}_{k+1}$.

Если допустимая абсолютная погрешность приближённого корня $\underline{x}_{k+1}$ задана заранее и равна ε, то процесс сближения прекращается в тот момент, когда будет обнаружено, что $\overline{x}_{k+1} - \underline{x}_{k+1} < \varepsilon$. По окончании процесса за значение корня x^* лучше всего взять среднее арифметическое полученных последних значений: $x^* \approx \frac{1}{2}(\underline{x}_{k+1} + \overline{x}_{k+1})$.

2.7. МЕТОД ПРОСТОЙ ИТЕРАЦИИ

Рассмотрим уравнение $x = g(x)$. Это уравнение может быть получено из уравнения $f(x) = 0$ путём прибавления к обоим частям x и заменой $g(x) = x + f(x)$ либо каким-то другим способом. Пусть $[a, b]$ – отрезок, определяющий корень x^* уравнения $f(x) = 0$, а следовательно, и равносильного ему уравнения $x = g(x)$.

Выберем произвольную точку x_0, которую примем за грубое приближение корня и подставим её в правую часть уравнения $x = g(x)$. Тогда получим некоторое число $x_1 = g(x_0)$.

По найденному значению x_1 определим вторую точку x_2 и т.д. Повторяя этот процесс, будет иметь последовательность чисел $x_{k+1} = g(x_k)$, $k = 0, 1, 2, \ldots$.

Если полученная таким образом последовательность x_k сходящаяся, т.е. существует предел $x^* = \lim\limits_{k \to \infty} x_k$, то она сходится к корню x^*, и за конечное число итераций можно получить приближённое значение корня x^* с любой степенью точности.

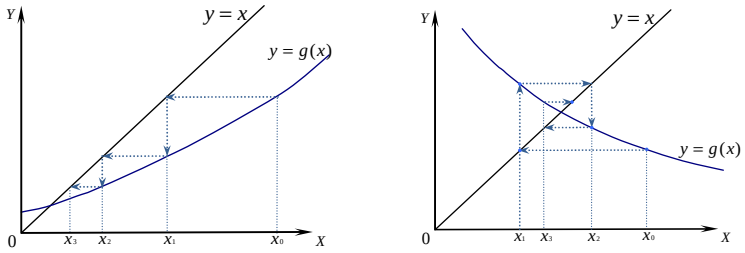


Рис. 2.11. Иллюстрация итерационного процесса нахождения корня уравнения методом простой итерации

Геометрически метод итерации может быть пояснён следующим образом. Построим на плоскости XOY графики функций $y = x$ и $x = g(x)$. Каждый действительный корень x^* уравнения $x = g(x)$ является абсциссой точки пересечения M кривой $x = g(x)$ с прямой $y = x$ (рис. 2.11).

Однако процесс итерации сходится не всегда (рис. 2.12).

Достаточным условием сходимости итерационного процесса является следующая теорема:

Теорема 3: *Пусть функция* $g(x)$ *определена и дифференцируема на отрезке* $[a, b]$, *причём все её значения* $g(x) \in [a, b]$. *Тогда если существует правильная дробь* q (*за* q *можно принять* $q = \min\limits_{x \in [a, b]} |g'(x)|$) *такая, что* $|g'(x)| \le q < 1$ *при* $a < x < b$, *то*:

1) *процесс итерации* $x_{k+1} = g(x_k)$, $k = 0, 1, 2, \ldots$ *сходится независимо от начального значения* $x_{k+1} \in [a, b]$;

2) *предельное значение* $x^* = \lim\limits_{k \to \infty} x_k$ *является единственным корнем уравнения* $x = g(x)$ *на отрезке* $[a, b]$.

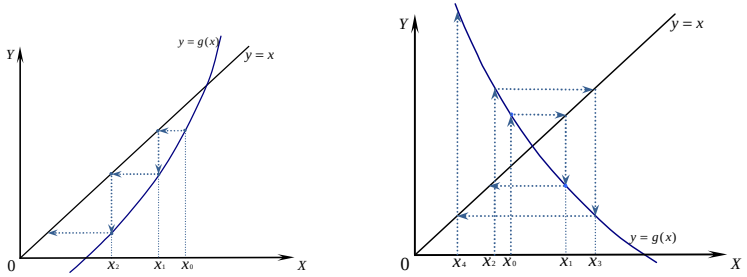


Рис. 2.12. Иллюстрация итерационного процесса нахождения корня уравнения методом простой итерации (расходящийся процесс)

Оценить погрешность приближённого значения x_{k+1} можно по расхождению двух последовательных приближений x_k и x_{k+1}. Однако использование неравенства $|x_{k+1} - x_k|$ в качестве критерия остановки процесса приближений не всегда правомерно (например, в случае если $\varphi'(x)$ близка к 1). Поэтому в общем случае процесс итерации следует продолжать до тех пор, пока для двух последовательных приближений x_k и x_{k+1} не будет обеспечено выполнение неравенства $|x_{k+1} - x_k| \le \frac{1-q}{q}\varepsilon$, где ε – заданная предельная абсолютная погрешность корня x^* и $|g'(x)| \le q$.

3. ЧИСЛЕННОЕ РЕШЕНИЕ СИСТЕМ ЛИНЕЙНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ

Пусть дана система n линейных алгебраических уравнений с n неизвестными

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1; \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2; \\ \ldots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n = b_n \end{cases} \tag{3.1}$$

или в матричной форме

$$\mathrm{A}\mathrm{x} = \mathrm{b}\,, \tag{3.2}$$

где

$$\mathrm{A} = (a_{ij}) = \begin{pmatrix} a_{11}a_{12}\ldots a_{1n} \\ a_{21}a_{22}\ldots a_{2n} \\ \ldots \\ a_{n1}a_{n2}\ldots a_{nn} \end{pmatrix} \tag{3.2.1}$$

– матрица коэффициентов,

$$\mathrm{b} = \begin{pmatrix} b_1 \\ b_2 \\ \ldots \\ b_n \end{pmatrix} \tag{3.2.2}$$

и

$$\mathrm{x} = \begin{pmatrix} x_1 \\ x_2 \\ \ldots \\ x_n \end{pmatrix} \tag{3.2.3}$$

– столбец свободных членов и столбец неизвестных соответственно.

Если матрица A неособенная, т.е.

$$\det \mathrm{A} = \Delta = \begin{vmatrix} a_{11}a_{12}\ldots a_{1n} \\ a_{21}a_{22}\ldots a_{2n} \\ \ldots \\ a_{n1}a_{n2}\ldots a_{nn} \end{vmatrix} \neq 0\,,$$

то система (3.1) имеет единственное решение. В этом случае решение системы (3.1) с теоретической точки зрения не представляет труда. Значения неизвестных $x_i (i = 1, 2, \ldots, n)$ могут быть получены по известным формулам Крамера:

$$x_i = \frac{\det A_i}{\det A},$$

где матрица A_i получается из матрицы A заменой i-го столбца столбцом свободных членов.

Действительно, если $\det A \neq 0$, то существует обратная матрица A^{-1}. Тогда, умножая обе части уравнения (3.2) слева на A^{-1}, получим:

$$A^{-1}Ax = A^{-1}b$$

или

$$x = A^{-1}b . \qquad (3.4)$$

Формула (3.4) является матричной записью формул Крамера (3.3).

Однако подобный способ решения линейной системы с n неизвестными приводит к вычислению $(n + 1)$ определителей порядка n, что представляет собой трудоёмкую операцию при сколько-нибудь большом числе n.

Применяемые в настоящее время методы решения систем линейных алгебраических уравнений (СЛАУ) можно разбить на две группы: *точные* и *приближённые*.

В *точных* (или *прямых*) методах решение системы (3.1) находится за конечное число арифметических действий. Примерами прямых методов могут служить метод Гаусса, метод Гаусса с выбором главных элементов, метод квадратных корней и др. Однако необходимо отметить, что вследствие погрешностей округления при решении задач прямые методы на самом деле не приводят к точному решению системы (3.1) и называть их точными можно лишь отвлекаясь от погрешностей округления.

Приближёнными (или *итерационными*) методами называются такие методы, которые даже в предположении, что вычисления ведутся без округлений, позволяют получить решение системы (3.1) лишь с заданной точностью. Точное решение системы в этих случаях может быть получено теоретически как результат бесконечного процесса. К приближённым методам относятся метод простой итерации, метод Зейделя и др.

Однако прежде чем перейти к изучению различных численных методов решения СЛАУ, необходимо вспомнить некоторые сведения о матрицах из курса высшей математики.

1. Пусть дана матрица

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}.$$

Тогда обратная матрица A^{-1} может быть вычислена следующим образом:

$$A^{-1} = \frac{1}{\Delta}\begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix},$$

где A_{ij} – алгебраические дополнения $i, j = 1, 2, 3$.

2. Алгебраическим дополнением элемента называется минор этого элемента, умноженный на $(-1)^P$, где P – сумма номеров строки и столбца, на пересечении которых расположен этот элемент.

3. Минором элемента a_{ij} называется определитель, получаемый вычёркиванием i-й строки и j-го столбца.

4. Определитель матрицы равен сумме произведений элементов какого-нибудь столбца или строки на их алгебраические дополнения.

3.1. МЕТОД ГАУССА

Наиболее распространённым методом решения СЛАУ является алгоритм последовательного исключения неизвестных. Этот метод носит название *метода Гаусса* или *метода исключения Гаусса*.

Алгоритм исключения рассмотрим для простоты на примере решения системы из четырёх линейных уравнений с четырьмя неизвестными.

$$\begin{cases} a_{11}^{(0)}x_1 + a_{12}^{(0)}x_2 + a_{13}^{(0)}x_3 + a_{14}^{(0)}x_4 = b_1^{(0)}; \\ a_{21}^{(0)}x_1 + a_{22}^{(0)}x_2 + a_{23}^{(0)}x_3 + a_{24}^{(0)}x_4 = b_2^{(0)}; \\ a_{31}^{(0)}x_1 + a_{32}^{(0)}x_2 + a_{33}^{(0)}x_3 + a_{34}^{(0)}x_4 = b_3^{(0)}; \\ a_{41}^{(0)}x_1 + a_{42}^{(0)}x_2 + a_{43}^{(0)}x_3 + a_{44}^{(0)}x_4 = b_4^{(0)}. \end{cases} \tag{3.5}$$

Пусть $a_{11}^{(0)} \neq 0$. Коэффициент $a_{11}^{(0)}$ в этом случае будет называться ведущим элементом. Поделим все коэффициенты первого уравнения системы (3.5) на $a_{11}^{(0)}$. В результате получим систему

$$\begin{cases} x_1 + a_{12}^{(1)}x_2 + a_{13}^{(1)}x_3 + a_{14}^{(1)}x_4 = b_1^{(1)}; \\ a_{21}^{(0)}x_1 + a_{22}^{(0)}x_2 + a_{23}^{(0)}x_3 + a_{24}^{(0)}x_4 = b_2^{(0)}; \\ a_{31}^{(0)}x_1 + a_{32}^{(0)}x_2 + a_{33}^{(0)}x_3 + a_{34}^{(0)}x_4 = b_3^{(0)}; \\ a_{41}^{(0)}x_1 + a_{42}^{(0)}x_2 + a_{43}^{(0)}x_3 + a_{44}^{(0)}x_4 = b_4^{(0)}, \end{cases} \tag{3.6}$$

где $a_{1j}^{(1)} = a_{1j}^{(0)} \big/ a_{11}^{(0)}$, $j = 2, 3, 4$; $b_1^{(1)} = b_1^{(0)} \big/ a_{11}^{(0)}$.

Пользуясь первым уравнением системы (3.6), можно легко исключить неизвестное x_1 из второго, третьего и четвёртого уравнений этой системы. Для этого следует умножить первое уравнение последней системы на $a_{21}^{(0)}$, $a_{31}^{(0)}$, $a_{41}^{(0)}$ и вычесть результат соответственно из второго, третьего и четвёртого уравнений системы (3.6). Тогда система (3.6) преобразуется к виду:

$$\begin{cases} x_1 + a_{12}^{(1)} x_2 + a_{13}^{(1)} x_3 + a_{14}^{(1)} x_4 = b_1^{(1)}; \\ a_{22}^{(1)} x_2 + a_{23}^{(1)} x_3 + a_{24}^{(1)} x_4 = b_2^{(1)}; \\ a_{32}^{(1)} x_2 + a_{33}^{(1)} x_3 + a_{34}^{(1)} x_4 = b_3^{(1)}; \\ a_{42}^{(1)} x_2 + a_{43}^{(1)} x_3 + a_{44}^{(1)} x_4 = b_4^{(1)}, \end{cases} \tag{3.7}$$

где $a_{ij}^{(1)} = a_{ij}^{(0)} - a_{1j}^{(1)} a_{i1}^{(1)}$, $b_i^{(1)} = b_i^{(0)} - b_i^{(1)} a_{i1}^{(0)}$, $i = 2, 3, 4$; $j = 1, 2, 3, 4$.

На следующем этапе алгоритма разделим все коэффициенты второго уравнения системы (3.7) на $a_{22}^{(1)}$ (при условии, что $a_{22}^{(1)} \neq 0$) и исключим неизвестные x_2 из уравнений системы (3.7), начиная с третьего, так же, как это делалось ранее при исключении неизвестной x_1. В результате получим систему вида:

$$\begin{cases} x_1 + a_{12}^{(1)} x_2 + a_{13}^{(1)} x_3 + a_{14}^{(1)} x_4 = b_1^{(1)}; \\ x_2 + a_{23}^{(2)} x_3 + a_{24}^{(2)} x_4 = b_2^{(2)}; \\ a_{33}^{(2)} x_3 + a_{34}^{(2)} x_4 = b_3^{(2)}; \\ a_{43}^{(2)} x_3 + a_{44}^{(2)} x_4 = b_4^{(2)}, \end{cases} \tag{3.8}$$

где $a_{2j}^{(2)} = a_{2j}^{(1)} \big/ a_{22}^{(1)}$, $j = 2, 3, 4$, $b_2^{(2)} = b_2^{(1)} \big/ a_{22}^{(1)}$ $j = 1, \ldots 4$ и $a_{ij}^{(2)} = a_{ij}^{(1)} - a_{2j}^{(2)} a_{i2}^{(1)}$, $b_i^{(2)} = b_i^{(1)} - b_2^{(2)} a_{i2}^{(1)}$, $j = 3, 4$.

Приведём коэффициент перед x_3 в третьем уравнении системы (3.8) к единице, поделив это уравнение на $a_{33}^{(2)} \left(a_{33}^{(2)} \neq 0 \right)$; исключим из четвёртого уравнения системы (3.8) неизвестную x_3 по вышеописанному алгоритму и разделим четвёртое уравнение системы на коэффициент перед x_4. В результате система (3.8) преобразуется к виду:

$$\begin{cases} x_1 + a_{12}^{(1)} x_2 + a_{13}^{(1)} x_3 + a_{14}^{(1)} x_4 = b_1^{(1)}; \\ x_2 + a_{23}^{(2)} x_3 + a_{24}^{(2)} x_4 = b_2^{(2)}; \\ x_3 + a_{34}^{(3)} x_4 = b_3^{(3)}; \\ x_4 = b_4^{(4)}. \end{cases} \tag{3.9}$$

Таким образом, исходная система линейных алгебраических уравнений (3.5) свелась к системе треугольного вида (3.9). Эта процедура называется *прямым ходом метода Гаусса*. Теперь из системы (3.9), проводя вычислительный процесс «снизу вверх», легко определяются неизвестные x_4, x_3, x_2, x_1:

$$x_i = b_i^{(i)} - \sum_{j=i+1}^{4} a_{ij}^{(i)} x_j, \; i = 4, 3, 2, 1. \tag{3.10}$$

Эта операция называется *обратным ходом метода Гаусса*.

Аналогично решаются и СЛАУ, состоящие из n уравнений. В общем случае количество уравнений, составляющих систему и решаемых методом Гаусса, не должно превышать 100. Число арифметических действий (операций умножения и деления), которое необходимо выполнить для реализации метода Гаусса, определяются следующей формулой:

$$S = \frac{2n(n+1)(n+2)}{3} + n(n-1) \sim n^3,$$

где n – число неизвестных.

***Пример*:** Решить методом Гаусса систему уравнений

$$\begin{cases} x + y + z = 4; \\ 2x + 3y + z = 9; \\ x - y - z = -2. \end{cases}$$

Решение:

$$1)\ \begin{cases} x + y + z = 4; \\ y - z = 1; \\ -2y - 2z = -6. \end{cases} \qquad 2)\ \begin{cases} x + y + z = 4; \\ y - z = 1; \\ -4z = -4. \end{cases} \qquad 3)\ \begin{cases} x + y + z = 4; \\ y - z = 1; \\ z = 1. \end{cases}$$

4) $z = 1;\ y = 1 + z = 1 + 1 = 2;\ x = 4 - y - z = 4 - 2 - 1 = 1.$

Рассмотренный метод Гаусса может с успехом быть применён лишь в том случае, если все ведущие элементы отличны от нуля, т.е. $a_{ii} \neq 0,\ i = \overline{1,n}$. В противном случае обычный метод Гаусса может оказаться непригодным. Избегать указанных трудностей позволяет *метод Гаусса с выбором главного элемента*. Основная идея метода состоит в том, чтобы на очередном шаге исключать не следующее по номеру неизвестное, а то неизвестное, коэффициент при котором является наибольшим по модулю. Таким образом, в качестве ведущего элемента здесь выбирается главный, т.е. наибольший по модулю элемент. Тем самым, если $\det \mathrm{A} \neq 0$, то в процессе вычислений не будет происходить деления на ноль.

Выбор главного элемента в методе Гаусса может осуществляться по строкам и по столбцам.

В первом случае для выбора главного элемента сравниваются модули коэффициентов перед неизвестными в i-й строке, и коэффициент с наибольшим модулем a_{ip} занимает место ведущего элемента. Для того чтобы подобная перестановка не вызывала ошибок при численном решении СЛАУ, необходимо поменять местами i-й и p-й столбцы во всех оставшихся уравнениях системы.

При выборе главного элемента по столбцу сравниваются абсолютные значения коэффициентов крайнего левого столбца матрицы А и строка, содержащая наибольший по модулю элемент a_{qj}, занимает место главной строки с ведущим элементом a_{qj}.

После выбора главного элемента СЛАУ решают также, как и в обычном методе Гаусса. Таким образом, метод Гаусса с выбором главного элемента эквивалентен применению обычного метода Гаусса к системе, в которой на каждом шаге исключения проводится соответствующая перенумерация переменных или уравнений.

Иногда применяется и метод Гаусса с выбором главного элемента по всей матрице, когда в качестве ведущего выбирается максимальный по модулю элемент среди всех элементов матрицы системы.

***Пример*:** Решить систему уравнения методом Гаусса с выбором главного элемента.

$$\begin{cases} x+y+z=4; \\ 2x+3y+z=9; \\ x-y-z=2. \end{cases}$$

Решение:

1) $\begin{cases} 3y+2x+z=9; \\ y+x+z=4; \\ -y+x-z=-2. \end{cases}$

2) $\begin{cases} y+\frac{2}{3}x+\frac{1}{3}z=3; \\ \frac{1}{3}x+\frac{2}{3}z=1; \\ \frac{5}{3}x-\frac{2}{3}z=1. \end{cases}$

3) $\begin{cases} y+\frac{2}{3}x+\frac{1}{3}z=3; \\ \frac{5}{3}x-\frac{2}{3}z=1; \\ \frac{1}{3}x-\frac{2}{3}z=1. \end{cases}$

4) $\begin{cases} y+\frac{2}{3}x+\frac{1}{3}z=3; \\ x-\frac{2}{5}z=\frac{3}{5}; \\ \frac{4}{5}z=\frac{4}{5}z. \end{cases}$

5) $\begin{cases} y+\frac{2}{3}x+\frac{1}{3}z=3; \\ x-\frac{2}{5}z=\frac{3}{5}; \\ z=1. \end{cases}$

6) $\begin{cases} y=2; \\ x=1; \\ z=1. \end{cases}$

3.2. СХЕМА ХАЛЕЦКОГО

Более эффективным методом решения СЛАУ по сравнению с методом Гаусса является *вторая модификация метода Гаусса*, более известная под названием *схема Халецкого*.

Рассмотрим СЛАУ, записанную для удобства в матричной форме (3.2):

$$Ax = b,$$

где матрица коэффициентов A, вектор-столбцы свободных членов и неизвестных определяются в виде (3.2.1), (3.2.2), (3.2.3) соответственно.

Обозначим через Δ_S угловой минор порядка S матрицы A, т.е.

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

Тогда, согласно нижеследующей теореме, матрица A может быть представлена в виде произведения двух матриц.

Теорема: *Пусть все угловые миноры матрицы* A *отличны от нуля*, $\Delta_S \neq 0$, $S = 1, 2, \ldots, n$. *Тогда матрицу* A *можно представить, причём единственным образом в виде произведения*

$$A = LU,$$

где L – *нижняя треугольная матрица с ненулевыми диагональными элементами*

$$L = (l_{ij}) = \begin{pmatrix} l_{11} & 0 & \ldots & 0 \\ l_{21} & l_{22} & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ l_{n1} & l_{n2} & \ldots & l_{nn} \end{pmatrix}$$

и U – *верхняя треугольная матрица с единичной диагональю*

$$U = (u_{ij}) = \begin{pmatrix} 1 & u_{12} & \ldots & u_{1n} \\ 0 & 1 & \ldots & u_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & 1 \end{pmatrix}.$$

Для вычисления элементов l_{ij} и u_{ij} нижней и верхней треугольной матриц можно воспользоваться следующими формулами:

$$\begin{cases} l_{i1} = a_{i1}; \\ l_{ij} = a_{ij} - \sum\limits_{k=1}^{j-1} l_{ik} u_{kj}; 1 < j \le i \end{cases} \tag{3.11}$$

и

$$\begin{cases} u_{1i} = a_{1i} / l_{11}; \\ u_{ji} = \left(a_{ji} - \sum\limits_{k=1}^{j-1} l_{jk} u_{ki} \right) \Big/ l_{jj}; \end{cases} \quad 1 < j \le i. \tag{3.12}$$

Используя матрицы L и U, искомый вектор неизвестных x может быть вычислен из цепи уравнений:

$$Ly = b, \quad Ux = y. \tag{3.13}$$

Однако, ввиду того, что матрицы L и U треугольные, то системы (3.13) легко решаются, а именно:

$$\begin{cases} y_1 = b_1 / l_{11}; \\ y_i = \left(b_i - \sum\limits_{k=1}^{j-1} l_{ik} y_k \right) \Big/ l_{ii}; \; i > 1 \end{cases} \tag{3.14}$$

и

$$\begin{cases} x_n = y_n; \\ x_i = y_i - \sum\limits_{k=i+1}^{n} u_{ik} x_k;\ i < n. \end{cases} \tag{3.15}$$

Как видно из формул (3.14), значения вспомогательных чисел y_i выгодно вычислять вместе с коэффициентами l_{ij} и u_{ij}, а затем так же, как и в методе Гаусса, вычислить значения неизвестных x_i «снизу вверх».

При решении СЛАУ по схеме Халецкого вычисление элементов l_{ij} и u_{ij} необходимо осуществить одновременно, последовательно используя формулы (3.11) и (3.12), пока полностью не будут определены элементы m-й строки нижней треугольной матрицы L и m-го столбца верхней треугольной матрицы. Только после этого следует определить вспомогательное значение y_m и перейти к вычислению $(m + 1)$-й строки и $(m + 1)$-го столбца матриц L и U соответственно.

***Пример*:** Решить СЛАУ по схеме Халецкого

$$\begin{cases} x + y + z = 4; \\ 2x + 3y + z = 9; \\ x - y - z = -2. \end{cases}$$

Решение:

$$\mathrm{A} = \begin{pmatrix} 1 & 1 & 1 \\ 2 & 3 & 1 \\ 1 & -1 & -1 \end{pmatrix} \quad \mathrm{b} = \begin{pmatrix} 4 \\ 9 \\ 2 \end{pmatrix}.$$

1) $l_{11} = a_{11} = 1$; $u_{11} = a_{11}/l_{11} = 1/1 = 1$; $y_1 = b_1/l_{11} = 4/1 = 4$;

2) $l_{21} = a_{21} = 2$; $u_{12} = a_{12}/l_{11} = 1/1 = 1$; $l_{22} = a_{22} - l_{21}u_{12} = 3 - 2 \cdot 1 = 1$;

$u_{22} = (a_{22} - l_{21}u_{12})/l_{22} = (3 - 2 \cdot 1)/1 = 1$;

$y_2 = (b_2 - l_{21}y_1)/l_{22} = (9 - 2 \cdot 4)/1 = 1$;

3) $l_{31} = a_{31} = 1$;

$u_{13} = a_{13}/l_{11} = 1/1 = 1$;

$l_{32} = a_{32} - l_{31}u_{12} = -1 - 1 \cdot 1 = -2$;

$u_{23} = \dfrac{a_{23} - l_{21}u_{13}}{l_{22}} = \dfrac{1 - 2 \cdot 1}{1} = -1$;

$l_{33} = a_{33} - l_{31}u_{13} - l_{32}u_{23} = -1 - 1 \cdot 1 - (-2)(-1) = -4$;

$u_{33} = (a_{33} - l_{31}u_{13} - l_{32}u_{23})/l_{33} = (-1 - 1 \cdot 1 - (-2)(-1))/(-4) = 1$;

$y_3 = (b_3 - l_{31}y_1 - l_{32}y_2)/l_{33} = (2 - 1 \cdot 4 - (-2) \cdot 1)/(-4) = 1$;

4) $x_3 = y_3 = 1$;

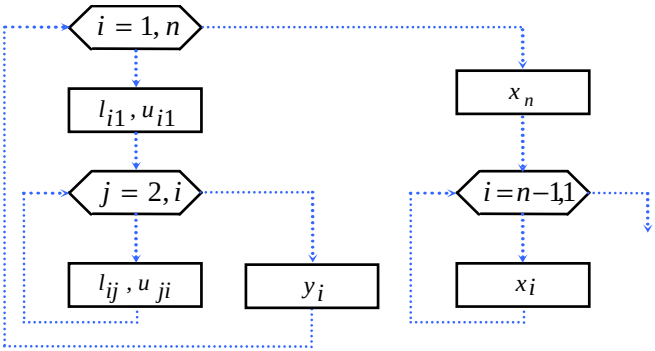


Рис. 3.1. Блок-схема алгоритма схемы Халецкого

$$x_2 = y_2 - u_{23}x_3 = 1 - (-1)\cdot 1 = 2;$$
$$x_1 = y_1 - u_{12}x_2 - u_{13}x_3 = 4 - 1\cdot 2 - 1\cdot 1 = 1.$$

Полученные при решении СЛАУ по схеме Халецкого матрицы L и U имеют следующий вид:

$$\mathrm{L} = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & -2 & -4 \end{pmatrix}; \quad \mathrm{U} = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{pmatrix}.$$

Таким образом, алгоритм схемы Халецкого может быть представлен в виде блок-схемы (рис. 3.1):

3.3. МЕТОД ОРТОГОНАЛИЗАЦИИ

Определение 1: *Скалярным произведением* двух векторов $\bar{p} = \{p_1, p_2, ..., p_n\}$ и $\bar{q} = \{q_1, q_2, ..., q_n\}$ в n-мерном векторном пространстве называется сумма произведений их координат, т.е.

$$(\bar{p}, \bar{q}) = \sum_{i=1}^{n} p_i q_i . \qquad (3.16)$$

Определение 2: Система векторов $\bar{p}_i$ в n-мерном векторном пространстве называется *линейно независимой*, если

$$\sum_{j=1}^{n} C_j \bar{p}_j = 0 , \qquad (3.17)$$

только тогда, когда все C_j одновременно равны нулю (C_j – некоторые константы).

Определение 3: Система векторов $\bar{p}_j \left(j = \overline{1,n}\right)$ называется *ортогональной*, если $\left(\bar{p}_i,\ \bar{p}_j\right) = 0$ при $i \neq j$.

***Лемма*:** Пусть задана линейно независимая система элементов $\bar{p}_1, ..., \bar{p}_n$. Можно построить ортогональную линейно независимую систему элементов

$$\bar{q}_j = \sum_{i=1}^{j} d_{ji}\,\bar{p}_i,\ \ j = \overline{1,n}, \tag{3.18}$$

где d_{ji} – некоторые коэффициенты, причём $\forall_j,\ d_{jj} = 1$,
или

$$\bar{q}_j = \bar{p}_j - \sum_{i=1}^{j-1} C_{ji}\,\bar{q}_i,\ \ j = \overline{1,n}\ . \tag{3.19}$$

Коэффициенты C_{ji} в выражении (3.19) выбираются при условии ортогональности $\left(\bar{q}_j,\ \bar{q}_i\right) = 0$ при $i < j$. Для этого обе части выражения (3.19) умножаются скалярным образом на $\bar{q}_i$. В итоге получаем:

$$\left(\bar{q}_j,\ \bar{q}_i\right) = \left(\bar{p}_j, \bar{q}_i\right) - C_{ji}\left(\bar{q}_i,\ \bar{q}_i\right) = 0,\ i < j\ . \tag{3.20}$$

Выражая из последнего равенства C_{ji} и подставляя результат в (3.19), получим:

$$\bar{q}_j = \bar{p}_j - \sum_{i=1}^{j-1} \frac{\left(\bar{p}_j,\ \bar{q}_i\right)}{\left(\bar{q}_i,\ \bar{q}_i\right)} \bar{q}_i\,. \tag{3.21}$$

Перейдём к рассмотрению алгоритма метода ортогонализации. Запишем систему уравнений (3.2) в виде

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n + b_1 = 0; \\ a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n + b_2 = 0; \\ \qquad\qquad ... \\ a_{n1}x_1 + a_{n2}x_2 + ... + a_{nn}x_n + b_n = 0. \end{cases} \tag{3.22}$$

Подобная запись СЛАУ эквивалентна системе, состоящей из n скалярных произведений векторов $\bar{a}_i = \{a_{i1}, a_{i2}, ..., a_{in}, b_i\},\ \ i = \overline{1,n}$ и $\bar{y} = \{x_1,\ x_2,\ ...,\ x_n,\ 1\}^{\mathrm{T}}$, т.е.

$$\left(\bar{a}_i,\ \bar{y}\right) = 0,\ \ i = \overline{1,n}\,. \tag{3.23}$$

Для того чтобы система (3.23) была полностью определённой (в настоящий момент имеется n уравнений с «n + 1»-й неизвестными) добавим к ней ещё одно скалярное произведение

$$\left(\bar{a}_{n+1},\ \bar{y}\right)=0\,, \tag{3.24}$$

где $\bar{a}_{n+1}=\{0,\ 0,\ ...,\ 0,\ 1\}$.

Векторы $\bar{a}_1,\ \bar{a}_2,\ ...,\ \bar{a}_{n+1}$ являются линейно независимыми, так как $\sum_{i=1}^{n+1} c_i \bar{a}_i = 0$, только когда некоторые числа $c_i\left(i=\overline{1,n+1}\right)$ одновременно равны нулю (в противном случае должны быть равны нулю компоненты векторов $\bar{a}_i\left(i=\overline{1,n+1}\right)$, что нарушает требование неособенности матрицы $\mathrm{A}\left(\det \mathrm{A}\neq 0\right)$ или её невырожденности (существование A^{-1}), а, следовательно, приводит к невозможности найти единственное решение системы (3.22)).

Применим к системе векторов $\bar{a}_i\left(i=\overline{1,n+1}\right)$ алгоритм последовательной ортогонализации с нормировкой. Будем строить две пересекающиеся последовательности ортогональных векторов $\bar{u}_1,\ \bar{\nu}_1,\ \bar{u}_2,\ \bar{\nu}_2,\ ...,\ \bar{u}_{n+1},\ \bar{\nu}_{n+1}$ по следующему правилу:

$$\bar{u}_1=\bar{a}_1,\ \bar{\nu}_1=\frac{\bar{u}_1}{\left\|\bar{u}_1\right\|},\ ...,\ \bar{u}_k=\bar{a}_k-\sum_{i=1}^{k-1} c_{ki}\bar{\nu}_i,\ \bar{\nu}_k=\frac{\bar{u}_k}{\left\|\bar{u}_k\right\|},\ ...\,.$$

В качестве нормы $\left\|\bar{u}_k\right\|$ будем использовать норму вида

$$\left\|\bar{u}_k\right\|=\sqrt{\left(\bar{u}_k,\ \bar{u}_k\right)}=\sqrt{\sum_{i=1}^{n+1} u_{ki}^2}\ . \tag{3.25}$$

Для вычисления констант c_{ki} воспользуемся выражением (3.20), где место скалярного произведения $\left(\bar{q}_i,\ \bar{q}_i\right)$ займёт произведение

$$\left(\bar{\nu}_i,\ \bar{\nu}_i\right)=\sum_{j=1}^{n+1}\nu_{ij}^2=\sum_{j=1}^{n+1}\frac{u_{ij}^2}{\left(\sqrt{\sum_{j=1}^{n+1}u_{ij}^2}\right)^2}=\frac{\sum_{j=1}^{n+1}u_{ij}^2}{\sum_{j=1}^{n+1}u_{ij}^2}=1,\ \ i=\overline{1,n+1}\,.$$

Таким образом, используя равенство (3.21), имеем:

$$\begin{cases} \bar{u}_k = \bar{a}_k - \sum_{i=1}^{k-1} (\bar{a}_k, \bar{\nu}_i)\bar{\nu}_i; \\ \bar{\nu}_k = \dfrac{\bar{u}_k}{\sqrt{\sum_{i=1}^{n+1} u_{ki}^2}}, \; k = \overline{1, n+1}. \end{cases} \tag{3.26}$$

У вектора $\bar{u}_{n+1} = \{u_{n+1,1}, u_{n+1,2}, ..., u_{n+1,n+1}\}$ последняя координата $u_{n+1,n+1}$ отлична от нуля, т.е. $u_{n+1,n+1} \neq 0$. Тогда, исходя из условия ортогональности, можно записать: $\forall_i \leq n$.

$$\begin{aligned} &(\bar{u}_{n+1}, \bar{\nu}_i) = \left(\bar{u}_{n+1}, \frac{\bar{u}_i}{\|\bar{u}_i\|}\right) = \frac{1}{\|\bar{u}_i\|}(\bar{u}_{n+1}, \bar{u}_i) = 0 = \frac{1}{\|\bar{u}_i\|}\left(\bar{u}_{n+1}, \bar{a}_i - \sum_{j=1}^{i-1} c_{ij}\bar{\nu}_j\right) = \\ &= \frac{1}{\|\bar{u}_i\|}\left[(\bar{u}_{n+1}, \bar{a}_i) - \left(\bar{u}_{n+1}, \sum_{j=1}^{i-1} c_{ij}\bar{\nu}_j\right)\right] = \\ &= \frac{1}{\|\bar{u}_i\|}\left[(\bar{u}_{n+1}, \bar{a}_i) - \left[c_{i1}(\bar{u}_{n+1}, \bar{\nu}_i)_{=0} + c_{i2}(\bar{u}_{n+1}, \bar{\nu}_2)_{=0} + ... + c_{i,i-1}(\bar{u}_{n+1}, \bar{\nu}_{i-1})_{=0}\right]\right] = \\ &= \frac{1}{\|\bar{u}_i\|}(\bar{u}_{n+1}, \bar{a}_i) = 0 \text{ или } (\bar{u}_{n+1}, \bar{a}_i) = 0. \end{aligned} \tag{3.27}$$

Скалярные произведения (3.27) можно переписать в виде

$$a_{i1}u_{n+1,1} + a_{i2}u_{n+1,2} + ... + a_{in}u_{n+1,n} + b_i u_{n+1,n+1} = 0; \tag{3.28}$$
$$i = \overline{1, n}.$$

Отсюда следует, что значения $\frac{u_{n+1,1}}{u_{n+1,n+1}}, \frac{u_{n+1,2}}{u_{n+1,n+1}}, ..., \frac{u_{n+1,n}}{u_{n+1,n+1}}$ являются решением исходной системы, т.е.

$$\bar{y} = \{x_1, x_2, ..., x_n, 1\} = \left\{\frac{u_{n+1,1}}{u_{n+1,n+1}}, \frac{u_{n+1,2}}{u_{n+1,n+1}}, ..., \frac{u_{n+1,n}}{u_{n+1,n+1}}, 1\right\}. \tag{3.29}$$

Значение координаты $u_{n+1,n+1}$ в (3.29) отлично от нуля, как было предположено выше. В противном случае система линейных уравнений

$$\sum_{j=1}^{n} a_{ij}u_{n+1,j} = 0, \; i = \overline{1, n}$$

образует систему либо с нулевым решением, т.е. $u_{n+1,\,j}=0,\ \left(j=\overline{1,n,}\right)$, что противоречит исходной постановке задачи, либо при ненулевых решениях, т.е. $u_{n+1,\,j}\neq 0,\ \left(j=\overline{1,n,}\right)$, определитель исходной матрицы А должен быть равен нулю $\det \mathrm{A}=0$, что также противоречит исходной постановке задачи.

Таким образом, алгоритм метода ортогонализации заключается в последовательном нахождении координат ортогональных векторов $\bar{u}_k$ и $\bar{\nu}_k\left(k=\overline{1,n+1}\right)$ по формулам (3.26), а затем, используя выражения (3.29), вычисляют значение корней СЛАУ.

***Пример*:** Решить СЛАУ методом ортогонализации

$$\begin{cases} x+y+z=4; \\ 2x+3y+z=9; \\ x-y-z=-2. \end{cases}$$

Решение:

$\bar{a}_1=\{1,\,1,\,1,\,-4\};\ \bar{a}_3=\{1,\,-1,\,-1,\,2\};$

$\bar{a}_2=\{2,\,3,\,1,\,-9\};\ \bar{a}_4=\{0,\,0,\,0,\,1\};$

$\bar{\mathrm{X}}=\{x,\,y,\,z,1\}.$

1) $\bar{u}_1=\bar{a}_1=\{1,\,1,\,1,\,-4\};$

$$\left\|\bar{u}_1\right\|=\sqrt{\sum_{i=1}^{4}u_{1i}^2}=\sqrt{1^2+1^2+1^2+(-4)^2}=\sqrt{19};$$

$$\bar{\nu}_1=\bar{u}_1\Big/\left\|\bar{u}_1\right\|=\left\{\frac{1}{\sqrt{19}},\,\frac{1}{\sqrt{19}},\,\frac{1}{\sqrt{19}},\,\frac{-4}{\sqrt{19}},\right\}.$$

2) $\bar{u}_2=\bar{a}_2-\left(\bar{a}_2,\,\bar{\nu}_1\right)\bar{\nu}_1;$

$$\left(\bar{a}_2\bar{\nu}_1\right)=\sum_{i=1}^{4}a_{2i}\nu_{1i}=2\frac{1}{\sqrt{19}}+\frac{31}{\sqrt{19}}+\frac{11}{\sqrt{19}}+(-9)\frac{-4}{\sqrt{19}}=\frac{42}{\sqrt{19}};$$

$$\bar{u}_2=\begin{pmatrix} 2-42\Big/\sqrt{19}\cdot 1\Big/\sqrt{19}=2-42/19=-4/19; \\ 2-42\Big/\sqrt{19}\cdot 1\Big/\sqrt{19}=3-42/19=15/19; \\ 1-42\Big/\sqrt{19}\cdot 1\Big/\sqrt{19}=1-42/19=-23/19; \\ -9-42\Big/\sqrt{19}\cdot 1\Big/\sqrt{19}=-9+168/19=-3/19. \end{pmatrix}$$

$$\left\|\bar{u}_2\right\|=\sqrt{\sum_{i=1}^{4}u_{2i}^2}=\sqrt{(4/19)^2+(15/19)^2+(-23/19)^2+(-3/19)^2}=\sqrt{779}/19;$$

$$\bar{\nu}_2 = \bar{u}_2 \Big/ \left\|\bar{u}_2\right\| = \left\{-\frac{4}{\sqrt{779}}, \frac{15}{\sqrt{779}}, \frac{-23}{\sqrt{779}}, \frac{-3}{\sqrt{779}}\right\}.$$

3) $\bar{u}_3 = \bar{a}_3 - (\bar{a}_3, \bar{\nu}_1)\bar{\nu}_1 - (\bar{a}_3, \bar{\nu}_2)\bar{\nu}_2$;

$$(\bar{a}_3\bar{\nu}_1) = \sum_{i=1}^{4} a_{3i}\nu_{1i} = 1\frac{1}{\sqrt{19}} + (-1)\frac{1}{\sqrt{19}} + (-1)\frac{1}{\sqrt{19}} + (2)\frac{-4}{\sqrt{19}} = \frac{-9}{\sqrt{19}};$$

$$(\bar{a}_3\bar{\nu}_2) = \sum_{i=1}^{4} a_{3i}\nu_{2i} = 1\frac{-4}{\sqrt{779}} + (-1)\frac{15}{\sqrt{779}} + (-1)\frac{-23}{\sqrt{779}} + 2\frac{-3}{\sqrt{779}} = \frac{-2}{\sqrt{779}};$$

$$\bar{u}_3 = \begin{pmatrix} 1 - \frac{-9}{\sqrt{19}}\cdot\frac{1}{\sqrt{19}} - \frac{-2}{\sqrt{779}}\cdot\frac{-4}{\sqrt{779}} = 1 + \frac{9}{19} - \frac{8}{779} = \frac{1140}{779}; \\ -1 - \frac{-9}{\sqrt{19}}\cdot\frac{1}{\sqrt{19}} - \frac{-2}{\sqrt{779}}\cdot\frac{15}{\sqrt{779}} = -1 + \frac{9}{19} + \frac{30}{779} = \frac{-380}{779}; \\ -1 - \frac{-9}{\sqrt{19}}\cdot\frac{1}{\sqrt{19}} - \frac{-2}{\sqrt{779}}\cdot\frac{-23}{\sqrt{779}} = -1 + \frac{9}{19} - \frac{46}{779} = \frac{-456}{779}; \\ 2 - \frac{-9}{\sqrt{19}}\cdot\frac{-4}{\sqrt{19}} - \frac{-2}{\sqrt{779}}\cdot\frac{-3}{\sqrt{779}} = 2 - \frac{36}{19} - \frac{6}{779} = \frac{76}{779}. \end{pmatrix}$$

$$\left\|\bar{u}_3\right\| = \sqrt{\sum_{i=1}^{4} u_{3i}^2} =$$

$$= \sqrt{(1140/779)^2 + (-380/779)^2 + (-456/779)^2 + (76/779)^2} = \sqrt{1\,657\,712}/779;$$

$$\bar{\nu}_3 = \bar{u}_3 \Big/ \left\|\bar{u}_3\right\| = \left\{\frac{1140}{\sqrt{1\,657\,712}}, \frac{-380}{\sqrt{1\,657\,712}}, \frac{-456}{\sqrt{1\,657\,712}}, \frac{76}{\sqrt{1\,657\,712}}\right\}.$$

4) $\bar{u}_4 = \bar{a}_4 - (\bar{a}_4, \bar{\nu}_1)\bar{\nu}_1 - (\bar{a}_4, \bar{\nu}_2)\bar{\nu}_2 - (\bar{a}_4, \bar{\nu}_3)\bar{\nu}_3$;

$$(\bar{a}_4\bar{\nu}_1) = \sum_{i=1}^{4} a_{4i}\nu_{1i} = 0\frac{1}{\sqrt{19}} + 0\frac{1}{\sqrt{19}} + 0\frac{1}{\sqrt{19}} + 1\frac{-4}{\sqrt{19}} = \frac{-4}{\sqrt{19}};$$

$$(\bar{a}_4\bar{\nu}_2) = \sum_{i=1}^{4} a_{4i}\nu_{2i} = 0\frac{-4}{\sqrt{779}} + 0\frac{15}{\sqrt{779}} + 0\frac{-23}{\sqrt{779}} + 1\frac{-3}{\sqrt{779}} = \frac{-3}{\sqrt{779}};$$

$$\left(\bar{a}_4\bar{v}_3\right)=\sum_{i=1}^{4}a_{4i}v_{3i}=0\frac{1140}{\sqrt{1\,657\,712}}+0\frac{-380}{\sqrt{1\,657\,712}}+0\frac{-456}{\sqrt{1\,657\,712}}+$$

$$+1\frac{76}{\sqrt{1\,657\,712}}=\frac{76}{\sqrt{1\,657\,712}};$$

$$\bar{u}_4=\begin{pmatrix}0-\frac{-4}{\sqrt{19}}\cdot\frac{1}{\sqrt{19}}-\frac{-3}{\sqrt{779}}\cdot\frac{-4}{\sqrt{779}}-\frac{76}{\sqrt{1657712}}\cdot\frac{1140}{\sqrt{1657712}}=\frac{4}{19}-\frac{12}{779}-\frac{86640}{1657712};\\ 0-\frac{-4}{\sqrt{19}}\cdot\frac{1}{\sqrt{19}}-\frac{-3}{\sqrt{779}}\cdot\frac{15}{\sqrt{779}}-\frac{76}{\sqrt{1657712}}\cdot\frac{-380}{\sqrt{1657712}}=\frac{4}{19}+\frac{45}{779}+\frac{28880}{1657712};\\ 0-\frac{-4}{\sqrt{19}}\cdot\frac{1}{\sqrt{19}}-\frac{-3}{\sqrt{779}}\cdot\frac{-23}{\sqrt{779}}-\frac{76}{\sqrt{1657712}}\cdot\frac{-456}{\sqrt{1657712}}=\frac{4}{19}-\frac{69}{779}+\frac{34656}{1657712};\\ 0-\frac{-4}{\sqrt{19}}\cdot\frac{-4}{\sqrt{19}}-\frac{-3}{\sqrt{779}}\cdot\frac{-3}{\sqrt{779}}-\frac{76}{\sqrt{1657712}}\cdot\frac{76}{\sqrt{1657712}}=1-\frac{16}{19}-\frac{9}{779}=\frac{5776}{1657712}.\end{pmatrix}$$

$$\bar{u}_4=\left\{\frac{236\,816}{1\,657\,712},\frac{473\,632}{1\,657\,712},\frac{236\,816}{1\,657\,712},\frac{236\,816}{1\,657\,712}\right\}.$$

5) $\mathrm{X}=\dfrac{u_{4i}}{u_{44}}$, $i=1,\,2,\,3$;

$$x=\frac{u_{41}}{u_{44}}=1;\ y=\frac{u_{42}}{u_{44}}=2;\ z=\frac{u_{43}}{u_{44}}=1.$$

3.4. МЕТОД ПРОСТОЙ ИТЕРАЦИИ

При численном решении СЛАУ большой размерности схемы точных методов становятся очень сложными. В этих условиях для нахождения корней системы пользуются приближёнными численными методами. Один из них – ***метод итерации*** *(метод простой итерации, метод последовательных приближений).*

Рассмотрим систему линейных уравнений (3.1), которая может быть записана в виде матричного уравнения (3.2). Преобразуем эту систему к виду:

$$\begin{cases}x_1=\alpha_{11}x_1+\alpha_{12}x_2+\ldots+\alpha_{1n}x_n+\beta_1;\\ x_2=\alpha_{21}x_1+\alpha_{22}x_2+\ldots+\alpha_{2n}x_n+\beta_2;\\ \ldots\\ x_n=\alpha_{n1}x_1+\alpha_{n2}x_2+\ldots+\alpha_{nn}x_n+\beta_n,\end{cases}\qquad(3.30)$$

где $\alpha = \alpha_{ij} = \begin{pmatrix} \alpha_{11} & \alpha_{12} & ... & \alpha_{1n} \\ \alpha_{21} & \alpha_{22} & ... & \alpha_{2n} \\ ... & ... & ... & ... \\ \alpha_{n1} & \alpha_{n2} & ... & \alpha_{nn} \end{pmatrix}$ и $\beta = \begin{pmatrix} \beta_1 \\ \beta_2 \\ ... \\ \beta_n \end{pmatrix}$ матрица коэффициентов и вектор-столбец свободных членов. Уравнение (3.30) может также быть записано в матричной форме:

$$x = \alpha x + \beta. \qquad (3.31)$$

Преобразование системы (3.1) к виду (3.30) может быть выполнено следующим образом. Исходя из предположения о неравенстве нулю диагональных коэффициентов матрицы А (3.2.1), т.е. $a_{ii} \neq 0$, $\left(i = \overline{1, n}\right)$, разрешают каждое *i*-е уравнение системы (3.1) относительно неизвестных x_i. В результате получают систему вида (3.30), у которой:

$$\beta_i = \frac{b_i}{a_{ii}};\ \alpha_{ij} = \begin{cases} -\dfrac{\alpha_{ij}}{a_{ii}},\ i \neq j; \\ 0,\ i = j, \end{cases}\ i,\ j = 1, 2, ..., n\,.$$

Систему (3.30) будем решать методом последовательных приближений. Выберем некоторое начальное (нулевое) приближение $x^{(0)}$. Далее будем последовательно находить первое приближение к решению системы $x^{(1)} = \alpha x^{(0)} + \beta$, второе $x^{(2)} = \alpha x^{(1)} + \beta$ и т.д.

В общем случае любое (*k* + 1)-е приближение вычисляют по формуле

$$x^{(k+1)} = \alpha x^{(k)} + \beta\,, \qquad (3.32)$$

или в развёрнутом виде

$$x_i^{(k+1)} = \sum_{j=1}^{n} \alpha_{ij} x_j^{(k)} + \beta_i,\ i = \overline{1, n};\ k = 0, 1, 2.... \qquad (3.33)$$

Если последовательность приближений $x^{(0)}$, $x^{(1)}$, … имеет предел $x = \lim\limits_{k \to \infty} x^{(k)}$, то этот предел является решением системы (3.30).

Однако итерационный процесс решения СЛАУ сходится не всегда. Условия сходимости итерационного процесса определяет следующая теорема.

Теорема: *Процесс итерации* (3.32) *для приведённой линейной системы* (3.31) *сходится к единственному решению из любого начального приближения, если какая-нибудь каноническая норма матрицы* α *меньше единицы, т.е.*

$$\|\alpha\| < 1\,. \qquad (3.34).$$

Эта теорема определяет достаточные условия сходимости. В качестве нормы матрицы могут быть приняты следующие значения:

1) $\|\alpha\| = \max\limits_{i} \sum\limits_{j=1}^{n} \alpha_{ij}$, или

2) $\|\alpha\| = \max\limits_{j} \sum\limits_{i=1}^{n} \alpha_{ij}$, или (3.35)

3) $\|\alpha\| = \sqrt{\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{n} |\alpha_{ij}|^2}$.

Следствие: Для системы $\sum\limits_{j=1}^{n} a_{ij} x_j = b_i \quad (i = \overline{1, n})$ процесс итерации сходится, если выполнены неравенства:

$$|a_{ii}| > \sum_{\substack{j=1 \\ i \neq j}}^{n} |a_{ij}|, \; i = \overline{1, n}, \tag{3.36}$$

или

$$|a_{jj}| > \sum_{\substack{i=1 \\ i \neq j}}^{n} |a_{ij}|, \; j = \overline{1, n}\,.$$

Таким образом, если для системы (3.31) выполняются условия (3.34), (3.35) или для системы (3.2) выполняется условие (3.36), то эта система может быть решена методом простой итерации. Причём итерационный процесс сойдётся при любом выборе начального приближения. На практике в качестве начального приближения $x^{(0)}$ часто используют вектор – столбец свободных членов β, т.е $x^{(0)} = \beta$.

Оценить погрешность вычисления приближённого решения СЛАУ методом простой итерации можно с помощью неравенства

$$\left\| x - x^{(k)} \right\| \leq \varepsilon\,. \tag{3.37}$$

Это условие будет выполнено, если в процессе вычислений будет обнаружено, что

$$\left\| x^{(k)} - x^{(k-1)} \right\| \leq \frac{1-q}{q} \varepsilon\,, \tag{3.38}$$

где $q = \|\alpha\| < 1$, а ε – требуемая точность решения.

В качестве нормы в неравенстве (3.38) могут использоваться следующие выражения:

$$\left\| x^{(k)} - x^{(k-1)} \right\| = \max_{1 \leq i \leq n} \left| x_i^{(k)} - x_i^{(k-1)} \right|,$$

или

$$\left\| x^{(k)} - x^{(k-1)} \right\| = \sum_{i=1}^{n} \left| x_i^{(k)} - x_i^{(k-1)} \right|, \qquad (3.39)$$

или

$$\left\| x^{(k)} - x^{(k-1)} \right\| = \sqrt{\sum_{i=1}^{n} \left| x_i^{(k)} - x_i^{(k-1)} \right|^2} .$$

3.5. МЕТОД ЗЕЙДЕЛЯ

Метод Зейделя (Гаусса–Зейделя) представляет собой некоторую модификацию метода простой итерации. Основная его идея заключается в том, что при вычислении (k + 1)-го приближения неизвестной x_i учитываются уже вычисленные ранее (k + 1)-е приближения неизвестных $x_1, x_2, \ldots, x_{i-1}$.

Пусть дана приведённая линейная система (3.31). Выберем произвольно начальное приближение $x^{(0)}$. Тогда, предполагая, что k-е приближение $x_i^{(k)}$ корней известны, (k + 1)-е приближение корней будем находить по формулам:

$$x_i^{(k+1)} = \beta_i + \sum_{j=1}^{i-1} \alpha_{ij} x_j^{(k+1)} + \sum_{j=i}^{n} \alpha_{ij} x_j^{(k)}, \; i = \overline{1, n}, \; k = 0, 1, 2.... \qquad (3.40)$$

Хотя области сходимости метода итерации и метода Зейделя не совпадают, а лишь пересекаются, тем не менее условия сходимости для метода Зейделя определяются теми же выражениями ((3.34), (3.35), (3.36)), что и для метода простой итерации. Обычно метод Зейделя даёт лучшую сходимость. Однако бывают случаи, когда процесс Зейделя сходится медленнее процесса простой итерации. Более того, бывают случаи, когда процесс итерации сходится, а метод Зейделя расходится.

При оценке погрешности вычисления методом Зейделя необходимо выполнение неравенства (3.37). Это произойдёт в том случае, когда выполнится следующее неравенство:

$$\left\| x^{(k)} - x^{(k-1)} \right\| \le \frac{1-\mu}{\mu} \varepsilon, \qquad (3.41)$$

где

$$\mu = \max_i \frac{\sum_{j=i}^{n} \left| \alpha_{ij} \right|}{1 - \sum_{j=1}^{i-1} \left| \alpha_{ij} \right|} .$$

4. ПРИБЛИЖЁННОЕ РЕШЕНИЕ СИСТЕМ НЕЛИНЕЙНЫХ УРАВНЕНИЙ

В отличие от систем линейных уравнений для систем нелинейных уравнений неизвестны прямые методы решения, и поэтому всегда применяются итерационные методы.

Рассмотрим нелинейную систему уравнений

$$\begin{cases} f_1(x_1,\ x_2,\ ...,\ x_n) = 0; \\ f_2(x_1,\ x_2,\ ...,\ x_n) = 0; \\ \qquad ... \\ f_n(x_1,\ x_2,\ ...,\ x_n) = 0. \end{cases} \tag{4.1}$$

Обозначим через x n-мерный вектор аргументов $x_1, x_2, \ldots, x_n$

$$x = (x_1,\ x_2,\ ...,\ x_n)^T, \tag{4.2}$$

а через f n-мерный вектор функций $f_1, f_2, \ldots, f_n$ (вектор-функцию):

$$f = (f_1,\ f_2,\ ...,\ f_n)^T. \tag{4.3}$$

Тогда система (4.1) может быть представлена в более компактном виде:

$$f(x) = 0. \tag{4.4}$$

Наиболее известными и часто используемыми методами решения нелинейных систем (4.4) являются метод простой итерации, метод Ньютона, метод скорейшего спуска.

4.1. МЕТОД ИТЕРАЦИИ

Преобразуем систему (4.1) к специальному виду:

$$\begin{cases} x_1 = \varphi_1(x_1,\ x_2,\ ...,\ x_n); \\ x_2 = \varphi_2(x_1,\ x_2,\ ...,\ x_n); \\ \qquad ... \\ x_n = \varphi_n(x_1,\ x_2,\ ...,\ x_n), \end{cases} \tag{4.5}$$

где функции $\varphi_1, \varphi_2, \ldots, \varphi_n$ действительны, определены и непрерывны в некоторой окрестности единственного решения этой системы $x^* = (x_1^*,\ x_2^*,\ ...,\ x_n^*)$.

Введя в рассмотрение вектор неизвестных (4.2) и вектор функций $\varphi = (\varphi_1, \varphi_2, ..., \varphi_n)^T$, запишем систему (4.5) более кратко:

$$x = \varphi(x). \tag{4.6}$$

Тогда для итерационного поиска приближённого решения системы (4.6) можно воспользоваться формулами:

$$x^{(k+1)} = \varphi\left(x^{(k)}\right),\ k = 0, 1, 2, \ldots. \tag{4.7}$$

Если процесс итерации (4.7) сходится, то предельное значение $x^* = \lim\limits_{k\to\infty} x^{(k)}$ обязательно является корнем уравнения (4.6).

Определим условия, при которых метод итерации сходится к единственному решению. Рассмотрим функции $y_i = \varphi_i(x),\ i = \overline{1,n}$, где x есть вектор-столбец (4.2), которые отображают n-мерное действительное пространство само в себя, так как область определения и область значения функции $y_i = \varphi_i(x)$ полностью совпадают. Запишем эту систему функций в матричной форме:

$$y = \varphi(x), \tag{4.8}$$

где $y = (y_1, y_2, ..., y_n)^T$.

Отображения (4.8) называются *сжимающими*, если для любых двух векторов $\tilde{x}$ и $\tilde{\tilde{x}}$ выполняются неравенства:

$$\left\|\varphi(\tilde{x}) - \varphi(\tilde{\tilde{x}})\right\| \le q\left\|\tilde{x} - \tilde{\tilde{x}}\right\| \tag{4.9}$$

для $0 \le q < 1$. В качестве норм неравенства (4.9) можно использовать следующие канонические нормы:

$$\|x\| = \max_i |x_i|, \text{ или } \|x\| = \sum_i |x_i|, \text{ или } \|x\| = \sqrt{\sum_i x_i^2}\ . \tag{4.10}$$

Тогда справедливой является следующая теорема.

Теорема: *Пусть отображение* (4.8) *является сжимающим в некоторой замкнутой области, т.е. выполняются условия* (4.9). *Тогда, если для итерационного процесса* (4.7) *все последовательные приближения* $x^{(k)}$ (k = 0, 1, 2, …) *принадлежат этой области, то*

1) *независимо от выбора начального приближения* $x^{(0)}$ *процесс* (*4.7*) *сходится, т.е. существует*

$$x^* = \lim_{k\to\infty} x^{(k)}; \tag{4.11}$$

2) *предельный вектор* x^* *является единственным решением уравнения* (4.6) *в искомой области*;

3) *справедлива оценка*

$$\left\|x^* - x^{(k)}\right\| \le \frac{q^k}{1-q}\left\|x^{(1)} - x^{(0)}\right\|, \tag{4.12}$$

или

$$\left\|x^* - x^{(k)}\right\| \le \frac{q}{1-q}\left\|x^{(k)} - x^{(k-1)}\right\|. \tag{4.13}$$

Если ужесточить требования, накладываемые на правильную дробь q, в частности, принять $0 \le q \le \frac{1}{2}$, то из формулы (4.13) следует, что при

$$\left\|x^{(k)} - x^{(k-1)}\right\| \le \varepsilon \tag{4.14}$$

справедливо неравенство

$$\left\|x^* - x^{(k)}\right\| \le \varepsilon .$$

Таким образом, неравенство (4.14), где норма вычисляется по одной из формул (4.10), может использоваться в качестве критерия остановки поиска методом итерации.

Хотя выше приведённая теорема и доказывает существование единственного решения нелинейной системы (4.6), найденного с заданной точностью, но не позволяет достаточно легко определить условия, при которых отображение (4.8) является сжимающим, а следовательно, система (4.6) может быть решена методом итераций.

Для этого существует следующая теорема.

Теорема: *Пусть функции* $\varphi(x)$ *определены и непрерывны вместе со своими производными* $\varphi'^{(x)} = \left(\frac{\partial \varphi_i(x)}{\partial x_j}\right)$, $i,\ j = \overline{1, n}$ *в некоторой ограниченной замкнутой области, для которой выполняются неравенства*:

$$\left\|\varphi'(x)\right\| = \max_x \max_i \sum_{j=1}^{n} \left|\frac{\partial \varphi_i(x)}{\partial x_j}\right| \le q < 1, \tag{4.15}$$

или

$$\left\|\varphi'(x)\right\| = \max_x \max_j \sum_{i=1}^{n} \left|\frac{\partial \varphi_i(x)}{\partial x_j}\right| \le q < 1,$$

где q – некоторая постоянная. Тогда, если последовательные приближения (4.7) *не выходят из этой замкнутой области, то процесс итерации* (4.7) *сходится и предельный вектор* (4.11) *является единственным решением системы* (4.6) *в рассматриваемой области.*

Следствие из этой теоремы позволяет записать условие (4.15) в более простом виде.

***Следствие 1*:** Процесс итерации (4.7) сходится к единственному решению, если выполняются неравенства:

$$\sum_{i=1}^{n}\left|\frac{\partial\varphi_i(x)}{\partial x_j}\right| \le q_j < 1,\ \ j=\overline{1,n}\,, \tag{4.16}$$

или

$$\sum_{j=1}^{n}\left|\frac{\partial\varphi_i(x)}{\partial x_j}\right| \le q_j < 1,\ \ i=\overline{1,n}$$

для всех значений *x*, принадлежащих некоторой замкнутой области.

Таким образом, неравенства (4.16) определяют условия сходимости метода простой итерации для системы нелинейных уравнений.

4.2. МЕТОД НЬЮТОНА

Рассмотрим нелинейную систему уравнений (4.4). Точный корень x^* этого уравнения может быть представлен в виде суммы некоторого *k*-го приближения $x^{(k)}$ и поправки $\varepsilon^{(k)}$ (погрешности корня), т.е.

$$x^* = x^{(k)} + \varepsilon^{(k)}, \tag{4.17}$$

где

$$x^{(k)} = \left(x_1^{(k)},\ x_2^{(k)},\ ...,\ x_n^{(k)}\right)^T$$

и

$$\varepsilon^{(k)} = \left(\varepsilon_1^{(k)},\ \varepsilon_2^{(k)},\ ...,\ \varepsilon_n^{(k)}\right)^T.$$

Тогда, подставив выражение (4.17) в уравнение (4.4), будем иметь:

$$f\left(x^{(k)} + \varepsilon^{(k)}\right) = 0\,. \tag{4.18}$$

Пусть функция $f(x)$ непрерывно дифференцируема в некоторой выпуклой области, содержащей x^* и $x^{(k)}$. Разложим левую часть уравнения (4.18) по степеням малого вектора $\varepsilon^{(k)}$ в ряд Тейлора, ограничившись при этом лишь линейными членами:

$$f\left(x^{(k)}+\varepsilon^{(k)}\right)= f\left(x^{(k)}\right)+ f'\left(x^{(k)}\right)\varepsilon^{(k)}=0\,. \qquad (4.19)$$

Под производной $f'(x)$ в выражении (4.19) следует понимать *матрицу Якоби* (*якобиан*) системы функций f_1, f_2, …, f_n относительно переменных x_1, x_2, …, x_n, т.е.

$$f'(x)=w(x)=\begin{pmatrix} \dfrac{\partial f_1}{\partial x_1}, & \dfrac{\partial f_1}{\partial x_2}, & \ldots & \dfrac{\partial f_1}{\partial x_n}; \\ \dfrac{\partial f_2}{\partial x_1}, & \dfrac{\partial f_2}{\partial x_2}, & \ldots & \dfrac{\partial f_2}{\partial x_n}; \\ \ldots & \ldots & \ldots & \ldots \\ \dfrac{\partial f_n}{\partial x_1}, & \dfrac{\partial f_n}{\partial x_2}, & \ldots & \dfrac{\partial f_n}{\partial x_n}. \end{pmatrix}=\left(\frac{\partial f_i}{\partial x_j}\right),\ i,\ j=\overline{1,\,n}\,. \qquad (4.20)$$

Выразим из (4.19) $\varepsilon^{(k)}$, предположив при этом, что матрица $W(x^{(k)})$ – неособенная:

$$\varepsilon^{(k)}=-w^{-1}\left(x^{(k)}\right)f\left(x^{(k)}\right).$$

Тогда с учётом (4.17) получим:

$$x^{(k+1)}=x^{(k)}-w^{-1}\left(x^{(k)}\right)f\left(x^{(k)}\right),\ k=0,\,1,\,2,\,.... \qquad (4.21)$$

Полученное рекуррентное соотношение определяет сущность метода Ньютона. За нулевое приближение $x^{(0)}$ можно взять грубое значение искомого корня. Условия сходимости итерационного процесса Ньютона определяется следующей теоремой.

Теорема: *Пусть дана нелинейная система уравнений* (*4.4*), *где вектор-функция* (*4.3*) *определена и непрерывна вместе со своими частными производными первого и второго порядков в некоторой области, и пусть* $x^{(0)}$ *– есть произвольная точка, лежащая в этой области вместе со своей замкнутой r-окрестностью*:

$$\left\|x-x^{(0)}\right\|\le r\,,$$

где норма понимается в виде $\|x\|=\max_i|x_i|$, *причём выполнены следующие условия:*

1) *матрица Якоби* $w(x)=\dfrac{\partial f_i}{\partial x_j}$ *при* $x=x^{(0)}$ *имеет обратную матрицу* $w^{-1}\left(x^{(0)}\right)$ *такую, что*

$$\left\| w^{-1}\left(x^{(0)}\right) \right\| \le A_0 , \tag{4.22}$$

где норма матрицы понимается как

$$\|\mathrm{A}\| = \max_i \sum_{j=1}^{n} \left| a_{ij} \right|;$$

2) $\left\| w^{-1}\left(x^{(0)}\right) f\left(x^{(0)}\right) \right\| \le B_0 \le \frac{r}{2}$; (4.23)

3) $\sum_{m=1}^{n} \left| \frac{\partial^2 f_i(x)}{\partial x_j \partial x_m} \right| \le C$ *при* $i,\ j = \overline{1,\ n}$; (4.24)

4) *константы* A_0, B_0, *и* C *удовлетворяют неравенству*

$$\delta = 2nA_0B_0C \le 1. \tag{4.25}$$

Тогда процесс Ньютона (4.21) *при любом начальном приближении* $x^{(0)}$ *из* r*-окрестности сходится и предельный вектор*

$$x^* = \lim_{k\to\infty} x^{(k)}$$

есть единственное решение системы (4.4) *такое, что*

$$\left\| x^* - x^{(0)} \right\| \le 2B_0 , \tag{4.26}$$

или

$$\left\| x^* - x^{(k)} \right\| \le \frac{B_0}{2^{k-1}} \delta^{2^k - 1}.$$

Из этой теоремы следует, что прежде чем браться за решение системы нелинейных уравнений методом Ньютона, необходимо проверить выполнение для этой системы условий (4.22) – (4.25). И только в случае положительного результата проверки система нелинейных уравнений может быть решена методом Ньютона. В качестве критерия окончания приближённого поиска корней нелинейных уравнений может служить неравенство (4.26).

4.3. МОДИФИЦИРОВАННЫЙ МЕТОД НЬЮТОНА

При использовании метода Ньютона существенным неудобством является необходимость для каждого шага вычислить обратную матрицу $w^{-1}\left(x^{(k)}\right)$. Если матрица $w^{-1}(x)$ непрерывна в окрестности искомого решения x^* и начальное приближение $x^{(0)}$ достаточно близко x^*, то приближённо можно положить:

$$w^{-1}\left(x^{(k)}\right) \approx w^{-1}\left(x^{(0)}\right).$$

Таким образом, получается рекуррентное соотношение модифицированного метода Ньютона

$$x^{(k+1)} = x^{(k)} - w^{-1}\left(x^{(0)}\right) f\left(x^{(k)}\right), \quad k = 0, 1, 2, \ldots . \tag{4.27}$$

Условия сходимости и окончания поиска для модифицированного метода Ньютона определяется по тем же формулам, что и для простого метода Ньютона.

4.4. МЕТОД ЗЕЙДЕЛЯ

Для решения систем нелинейных уравнений специального вида (4.5) может применяться *метод покоординатного спуска* или *метод Зейделя*, аналогичный одноименному методу, применяемого для решения СЛАУ.

В методе покоординатного спуска при вычислении $(k + 1)$-го приближения неизвестной x_i учитываются уже вычисленные ранее $(k + 1)$-е приближения неизвестных $x_j\left(j = \overline{1, i-1}\right)$. Тогда рекуррентные соотношения метода Зейделя могут быть записаны в виде:

$$\begin{gathered}
x_1^{(k+1)} = \varphi_1\left(x_1^{(k)},\ x_2^{(k)},\ \ldots,\ x_n^{(k)}\right); \\
x_2^{(k+1)} = \varphi_2\left(x_1^{(k+1)},\ x_2^{(k)},\ \ldots,\ x_n^{(k)}\right); \\
\ldots \\
x_i^{(k+1)} = \varphi_i\left(x_1^{(k+1)},\ x_2^{(k+1)},\ \ldots,\ x_{i-1}^{(k+1)},\ x_i^{(k)},\ \ldots,\ x_n^{(k)}\right); \\
\ldots \\
x_n^{(k+1)} = \varphi_n\left(x_1^{(k+1)},\ x_2^{(k+1)},\ \ldots,\ x_{n-1}^{(k+1)},\ x_n^{(k)}\right).
\end{gathered} \tag{4.28}$$

Условия сходимости метода Зейделя определяются так же, как и в методе простой итерации – по формулам (4.16).

5. ИНТЕРПОЛИРОВАНИЕ ФУНКЦИЙ

Простейшая задача интерполирования заключается в следующем. На отрезке $[a, b]$ заданы $n + 1$ точки $x_0, x_1, \ldots, x_n$, которые называются *узлами интерполяции*, и значение некоторой функции $f(x)$ в этих точках: $f(x_0) = y_0$, $f(x_1) = y_1, \ldots, f(x_n) = y_n$.

Требуется построить функцию $F(x)$ (*интерполирующую функцию*), принадлежащую известному классу и принимающую в узлах интерполяции те же значения, что и $f(x)$, т.е. такую, что $F(x_0) = y_0$, $F(x_1) = y_1, \ldots, F(x_n) = y_n$.

Геометрически это означает, что нужно найти кривую $y = F(x)$ некоторого определённого типа, проходящую через заданную систему точек $M_i(x_i, y_i)$, $i = 0, 1, 2, \ldots, n$ (рис. 5.1).

В такой общей постановке задача может иметь бесчисленное множество решений или совсем не иметь решений. Однако эта задача становится однозначной, если вместо произвольной функции $F(x)$ искать полином $P_n(x)$ степени не выше n, удовлетворяющий следующим условиям: $P_n(x_0) = y_0$, $P_n(x_1) = y_1, \ldots, P_n(x_n) = y_n$.

Полученную интерполяционную формулу $y = F(x) = P_n(x)$ обычно используют для приближённого вычисления значений данной функции $f(x)$ для значений аргумента x, отличных от узлов интерполирования. Такая операция называется *интерполированием функции* $f(x)$.

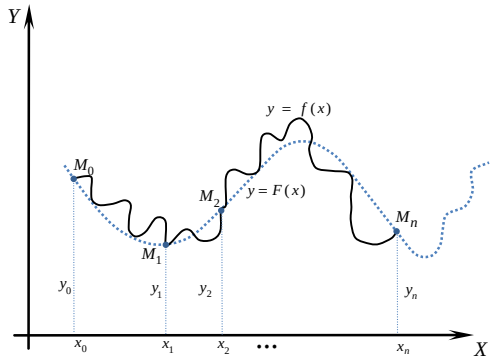


Рис. 5.1. Графическая иллюстрация интерполяции

При этом различают *интерполирование в узком смысле*, когда $x \in [x_0, x_n]$, т.е. значение x является промежуточным между x_0 и x_n, и *экстраполирование*, когда $x \notin [x_0, x_n]$. В дальнейшем под термином интерполирование будем понимать как первую, так и вторую операции.

5.1. ИНТЕРПОЛЯЦИОННЫЕ ФОРМУЛЫ НЬЮТОНА

Для определения интерполяционных формул Ньютона введём понятие конечных разностей. Пусть $y = f(x)$ – функция, заданная табличными значениями $y_i = f(x_i)$ для системы равноотстоящих точек $x_i (i = 0, 1, 2, ..., n)$, где $\Delta x_i = x_{i+1} - x_i = h = \text{const}$.

Тогда конечными разностями табличной функции $y = f(x)$ называются следующие соотношения:

$\Delta y_i = y_{i+1} - y_i$ – конечная разность 1-го порядка;

$\Delta^2 y_i = \Delta(\Delta y_i) = \Delta y_{i+1} - \Delta y_i$ – конечная разность 2-го порядка;

…

$\Delta^k y_i = \Delta(\Delta^{k-1} y_i) = \Delta^{k-1} y_{i+1} - \Delta^{k-1} y_i$ – конечная разность *k*-го порядка;

…

$\Delta^n y_i = \Delta(\Delta^{n-1} y_i) = \Delta^{n-1} y_{i+1} - \Delta^{n-1} y_i$ – конечная разность *n*-го порядка.

Конечные разности различных порядков удобно располагать в форме таблиц двух видов: горизонтальной таблицы конечных разностей или диагональной таблицы конечных разностей (табл. 5.1, 5.2).

Перейдём к рассмотрению интерполяционных формул Ньютона.

Пусть для функции $y = f(x)$ заданы значения $y_i = f(x_i)$ для равностоящих значений независимой переменной $x_i = x_0 + ih$, $i = 0, 1, 2, ..., n$, где h – шаг интерполяции. Требуется подобрать полином $P_n(x)$ степени не выше *n*, принимающий в точках x_i значения $P_n(x_i) = y_i$, $i = 0,1, 2, ..., n$.

5.1. Горизонтальная таблица конечных разностей

x_i	y_i	Δy_i	$\Delta^2 y_i$	$\Delta^3 y_i$	$\Delta^4 y_i$	$\Delta^5 y_i$
x_0	y_0	Δy_0	$\Delta^2 y_0$	$\Delta^3 y_0$	$\Delta^4 y_0$	$\Delta^5 y_0$
x_1	y_1	Δy_1	$\Delta^2 y_1$	$\Delta^3 y_1$	$\Delta^4 y_1$	
x_2	y_2	Δy_2	$\Delta^2 y_2$	$\Delta^3 y_2$		
x_3	y_3	Δy_3	$\Delta^2 y_3$			
x_4	y_4	Δy_4				
x_5	y_5					

5.2. Диагональная таблица конечных разностей

x_i	y_i	Δy_i	$\Delta^2 y_i$	$\Delta^3 y_i$
x_0	y_0	Δy_0	$\Delta^2 y_0$	$\Delta^3 y_0$
x_1	y_1	Δy_1	$\Delta^2 y_1$	
x_2	y_2	Δy_2		
x_3	y_3			

Введём некоторый параметр $q = \dfrac{x - x_0}{h}$, который определяет число шагов, требуемое для достижения числа x из точки x_0. Тогда искомый полином может быть определён следующим выражением:

$$P_n(x) = y_0 + q\Delta y_0 + \frac{q(q-1)}{2!}\Delta^2 y_0 + \frac{q(q-1)(q-2)}{3!}\Delta^3 y_0 + \ldots + \frac{q(q-1)\ldots(q-n+1)}{n!}\Delta^n y_0. \quad (5.1)$$

Формула (5.1) называется *первой интерполяционной формулой Ньютона*. Её удобно использовать для интерполирования функции $y = f(x)$ в окрестности начального значения x_0, где q – мало по абсолютной величине. В этой формуле используется верхняя горизонтальная строка горизонтальной таблицы конечных разностей.

Если в формуле (5.1) положить $n = 1$, то получим формулу линейного интерполирования $P_1(x) = y_0 + q\Delta y_0$.

При $n = 2$ будем иметь формулу параболического или квадратичного интерполирования: $P_n(x) = y_0 + q\Delta y_0 + \dfrac{q(q-1)}{2!}\Delta^2 y_0$.

Первая интерполяционная формула Ньютона практически неудобна для интерполирования функции вблизи конца таблицы. В этом случае обычно применяется *вторая интерполяционная формула Ньютона*, имеющая следующий вид:

$$P_n(x) = y_n + q\Delta y_{n-1} + \frac{q(q+1)}{2!}\Delta^2 y_{n-2} + \frac{q(q+1)(q+2)}{3!}\Delta^3 y_{n-3} + \ldots + \frac{q(q+1)\ldots(q+n-1)}{n!}\Delta^n y_0, \quad (5.2)$$

где $q = \dfrac{x - x_n}{n}$.

В этой формуле используется нижняя диагональная строка таблицы конечных разностей.

Как первую, так и вторую интерполяционные формулы Ньютона можно использовать для экстраполирования функции, т.е. для нахождения значений функции y для значения аргументов x, лежащих вне пределов таблицы. Если $x < x_0$ и x близко к x_0, то выгодно применять первую интерполяционную формулу Ньютона, причём тогда $q = \dfrac{x - x_n}{n} < 0$.

Если же $x > x_n$ и x близко к x_n, то удобней пользоваться второй интерполяционной формулой Ньютона, причём $q = \dfrac{x - x_n}{n} > 0$.

Таким образом, первая интерполяционная формула Ньютона обычно используется для интерполирования вперёд и экстраполирования назад, а вторая интерполяционная формула Ньютона, наоборот – для интерполирования назад и экстраполирования вперёд. Однако операция экстраполирования является менее точной по сравнению с операцией интерполирования в узком смысле слова.

В общем случае погрешность интерполирования табличной функции формулами Ньютона можно оценить с помощью остаточных членов этих формул.

Остаточный член первой интерполяционной формулы Ньютона имеет вид

$$R_n(x) = h^{n+1} \frac{q(q-1)\ldots(q-n)}{(n+1)!} f^{(n+1)}(\xi), \qquad (5.3)$$

где ξ – некоторое промежуточное значение между узлами интерполирования $x_0, x_1, \ldots, x_n$ и рассматриваемой точкой x. При этом для случая интерполирования в узком смысле $\xi \in [x_0, x_n]$, а для случая экстраполирования возможно, что $\xi \notin [x_0, x_n]$.

Для второй интерполяционной формулы остаточный член может быть рассчитан по формуле

$$R_n(x) = h^{n+1} \frac{q(q+1)\ldots(q+n)}{(n+1)!} f^{(n+1)}(\xi), \qquad (5.4)$$

где ξ – некоторое промежуточное значение между узлами интерполирования $x_0, x_1, \ldots x_n$ и точкой x.

Обычно при практических вычислениях интерполяционная формула Ньютона обрывается на членах, содержащих такие разности, которые в пределах заданной точности можно считать постоянными.

Предполагая, что $\Delta^{n+1} y$ почти постоянная для функции $y = f(x)$ и h достаточно мало, и учитывая, что $f^{(n+1)}(x) = \lim\limits_{h \to 0} \dfrac{\Delta^{n+1} y}{h^{n+1}}$, приближённо можно положить:

$$f^{(n+1)}(\xi) \approx \frac{\Delta^{n+1} y_0}{h^{n+1}}.$$

Тогда формулы (5.3) и (5.4) соответственно можно записать в виде:

$$R_n(x) \approx \frac{q(q-1)\ldots(q-n)}{(n+1)!}\Delta^{n+1}y_0,\ R_n(x) \approx \frac{q(q+1)\ldots(q+n)}{(n+1)!}\Delta^{n+1}y_n.$$

При построении таблицы разностей, использующейся в формулах Ньютона, исходили из предположения, что значения аргумента функции – равноотстоящие, т.е. имеют постоянный шаг. Однако на практике часто приходится иметь дело с неравноотстоящими значениями аргумента, которые изменяются с переменным шагом. Для таких неравноотстоящих узлов интерполирования вводится понятие *разделённых разностей*, являющееся более общим по сравнению с понятием конечных разностей.

Пусть функция $y = f(x)$ задана таблично и x_0, x_1, x_2, … – значения аргумента, а y_0, y_1, y_2, … – соответствующие значения функции, где $\Delta x_i = x_{i+1} - x_i \neq \text{const} \neq 0,\ i = 0, 1, 2, \ldots$. Тогда отношения $\delta(x_i,\ x_{i+1}) = \dfrac{y_{i+1} - y_i}{x_{i+1} - x_i}$, i = 0, 1, 2, … называются разделёнными разностями первого порядка. Аналогично определяются разделённые разности второго порядка

$$\delta(x_i,\ x_{i+1}, x_{i+2}) = \frac{\delta(x_{i+1},\ x_{i+2}) - \delta(x_i,\ x_{i+1})}{x_{i+2} - x_i},\ i = 0, 1, 2, \ldots.$$

В общем случае разделённые разности k-го порядка получаются из разделённых разностей $(k-1)$-го с помощью рекуррентного соотношения

$$\delta(x_i, x_{i+1}, \ldots, x_{i+k}) = \frac{\delta(x_{i+1}, \ldots, x_{i+k}) - \delta(x_i, \ldots, x_{i+k-1})}{x_{i+k} - x_i},\ k = 1, 2, \ldots,\ i = 0, 1, 2, \ldots.$$

Разделённые разности также обычно располагаются в таблицах (табл. 5.3).

Используя понятия конечных разностей, формула Ньютона для случая неравноотстоящих узлов интерполирования может быть записана в виде

$$\begin{aligned} P_n(x) = y_0 + \delta(x_0,\ x_1)(x - x_0) + \delta(x_0,\ x_1,\ x_2)(x - x_0)(x - x_1) + \ldots \\ \ldots + \delta(x_0,\ x_1, \ldots, x_n)(x - x_0)(x - x_1)\ldots(x - x_{n-1}). \end{aligned} \quad (5.5)$$

Погрешность формулы (5.5) определяется выражением для остаточного члена:

$$R_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}(x - x_0)(x - x_1)\ldots(x - x_n), \quad (5.6)$$

где ξ – некоторое промежуточное значение между точками x_0, x_1, …, x_n и x.

5.3. Таблица разделённых разностей

x_i	y_i	1-й порядок	2-й порядок	3-й порядок	4-й порядок
x_0	y_0	$\delta(x_0, x_1)$	$\delta(x_0, x_1, x_2)$	$\delta(x_0, x_1, x_2, x_3)$	$\delta(x_0, x_1, x_2, x_3, x_4)$
x_1	y_1	$\delta(x_1, x_2)$	$\delta(x_1, x_2, x_3)$	$\delta(x_1, x_2, x_3, x_4)$	
x_2	y_2	$\delta(x_2, x_3)$	$\delta(x_2, x_3, x_4)$		
x_3	y_3	$\delta(x_3, x_4)$			
x_4	y_4				

5.2. ИНТЕРПОЛЯЦИОННЫЕ ФОРМУЛЫ ГАУССА

При построении интерполяционных формул Ньютона используются лишь те значения функции, которые лежат по одну сторону от выбранного начального значения (больше y_0 в первой формуле и меньше y_n – во второй). Во многих случаях это оказывается неудобным и возникает необходимость использования при интерполяции как предшествующих, так и последующих значений функции по отношению к её начальному значению. В этом смысле наиболее удобной для использования в качестве начального значения является горизонтальная и непосредственно примыкающие к ней строки диагональной таблицы конечных разностей. Конечные разности, заполняющие данную таблицу в описанном случае, называются *центральными разностями* (табл. 5.4.).

Таблицы центральных разностей берутся за основу при построении интерполяционных формул Гаусса, Стирлинга, Бесселя.

Рассмотрим интерполяционную формулу Гаусса.

5.4. Таблица центральных разностей

x_i	y_i	Δy_i	$\Delta^2 y_i$	$\Delta^3 y_i$	$\Delta^4 y_i$	$\Delta^5 y_i$	$\Delta^6 y_i$
x_{-3}	y_{-3}	Δy_{-3}	$\Delta^2 y_{-3}$	$\Delta^3 y_{-3}$	$\Delta^4 y_{-3}$	$\Delta^5 y_{-3}$	$\Delta^6 y_{-3}$
x_{-2}	y_{-2}	Δy_{-2}	$\Delta^2 y_{-2}$	$\Delta^3 y_{-2}$	$\Delta^4 y_{-2}$	$\Delta^5 y_{-2}$	
x_{-1}	y_{-1}	Δy_{-1}	$\Delta^2 y_{-1}$	$\Delta^3 y_{-1}$	$\Delta^4 y_{-1}$		
x_0	y_0	Δy_{-0}	$\Delta^2 y_0$	$\Delta^3 y_0$			
x_1	y_1	Δy_1	$\Delta^2 y_1$				
x_2	y_2	Δy_2					
x_3	y_3						

Пусть имеется $2n + 1$ равноотстоящих узлов интерполирования

$$x_{-n},\ x_{-(n-1)},\ ...,\ x_{-1},\ x_0,\ x_1,\ ...,\ x_{n-1},\ x_n,$$

где $\Delta x_i = x_{i+1} - x_i = h = \text{const},\ \ i = -n,\ -n+1,\ ...,\ n-1,\ n$, и для функции $y = f(x)$ известны её значения в этих узлах $y_i = f(x_i)$, $i = 0, \pm1, \ldots, \pm n$. Требуется построить полином $P(x)$ степени не выше $2n$ такой, что $P(x_i) = y_i$ при $i = 0, \pm1, \ldots, \pm n$.

Введя переменную $q = \dfrac{x - x_0}{h}$, *первая интерполяционная формула Гаусса* может быть записана в виде

$$\begin{aligned} P(x) &= y_0 + q\Delta y_0 + \frac{q(q-1)}{2!}\Delta^2 y_{-1} + \frac{(q+1)q(q-1)}{3!}\Delta^3 y_{-1} + \\ &+ \frac{(q+1)q(q-1)(q-2)}{4!}\Delta^4 y_{-2} + \frac{(q+2)(q+1)q(q-1)(q-2)}{5!}\Delta^5 y_{-2} + ... \\ &... + \frac{(q+n-1)...(q-n+1)}{(2n-1)!}\Delta^{2n-1} y_{-(n-1)} + \frac{(q+n-1)...(q-n)}{(2n)!}\Delta^{2n} y_{-n}. \end{aligned} \qquad (5.7)$$

Как видно из (5.7) центральные разности, содержащиеся в первой интерполяционной формуле Гаусса, образуют нижнюю ломаную строку таблицы центральных разностей.

При построении *второй интерполяционной формулы Гаусса* используется верхняя ломаная строка таблицы центральных разностей:

$$\begin{aligned} P(x) &= y_0 + q\Delta y_{-1} + \frac{q(q+1)}{2!}\Delta^2 y_{-1} + \frac{(q+1)q(q-1)}{3!}\Delta^3 y_{-2} + \\ &+ \frac{(q+2)(q+1)q(q-1)}{4!}\Delta^4 y_{-2} + ... + \frac{(q+n-1)...(q-n+1)}{(2n-1)!}\Delta^{2n-1} y_{-n} + \\ &+ \frac{(q+n)...(q-n+1)}{(2n)!}\Delta^{2n} y_{-n}. \end{aligned} \qquad (5.8)$$

Остаточный член для обеих формул Гаусса будет равен

$$R(x) = \frac{h^{2n+1} f^{(2n+1)}(\xi)}{(2n+1)!} q(q^2 - 1^2)(q^2 - 2^2)..(q^2 - n^2), \qquad (5.9)$$

где ξ – некоторое промежуточное значение, находящееся между точками $x_{-n},\ x_{-(n-1)},\ ...,\ x_0,\ x_1,\ ...,\ x_n$ и точкой x.

5.3. ИНТЕРПОЛЯЦИОННАЯ ФОРМУЛА СТИРЛИНГА

Интерполяционная формула Стирлинга получается из первой и второй интерполяционных формул Гаусса как их среднее арифметическое. Она имеет следующий вид:

$$
\begin{aligned}
P(x) = y_0 + q\frac{\Delta y_{-1} + \Delta y_0}{2} + \frac{q^2}{2}\Delta^2 y_{-1} + \frac{q\left(q^2-1^2\right)}{3!}\frac{\Delta^3 y_{-2} + \Delta^3 y_{-1}}{2} + \\
+ \frac{q^2\left(q^2-1^2\right)}{4!}\Delta^4 y_{-2} + \frac{q\left(q^2-1^2\right)\left(q^2-2^2\right)}{5!}\frac{\Delta^5 y_{-3} + \Delta^5 y_{-2}}{2} + \\
+ \frac{q^2\left(q^2-1^2\right)\left(q^2-2^2\right)}{6!}\Delta^6 y_{-3} + \ldots + \frac{q\left(q^2-1^2\right)\left(q^2-2^2\right)\left(q^2-3^2\right)\ldots\left(q^2-(n-1)^2\right)}{(2n-1)!}\times \\
\times\frac{\Delta^{2n-1} y_{-n} + \Delta^{2n-1} y_{-(n-1)}}{2} + \frac{q^2\left(q^2-1^2\right)\left(q^2-2^2\right)\ldots\left(q^2-(n-1)^2\right)}{(2n)!}\Delta^{2n} y_{-n},
\end{aligned}
\tag{5.10}
$$

где $q = \dfrac{x - x_0}{h}$.

Погрешность интерполяционной формулы Стирлинга может быть определена в соответствии с выражением для остаточного члена формулы Гаусса (5.9).

5.4. ИНТЕРПОЛЯЦИОННАЯ ФОРМУЛА БЕССЕЛЯ

Интерполяционная формула Бесселя так же, как и формула Стирлинга, получается путём некоторых преобразований из формул Гаусса. Она имеет следующий вид:

$$
\begin{aligned}
P(x) = \frac{y_0 + y_1}{2} + \left(q - \frac{1}{2}\right)\Delta y_0 + \frac{q(q-1)}{2!}\cdot\frac{\Delta^2 y_{-1} + \Delta^2 y_0}{2} + \frac{\left(q - \frac{1}{2}\right)q(q-1)}{3!}\Delta^3 y_{-1} + \\
+ \frac{q(q-1)(q+1)(q-2)}{4!}\frac{\Delta^4 y_{-2} + \Delta^4 y_{-1}}{2} + \frac{\left(q - \frac{1}{2}\right)q(q-1)(q+1)(q-2)}{5!}\Delta^5 y_{-2} + \\
+ \frac{q(q-1)(q+1)(q-2)(q+2)(q-3)}{6!}\frac{\Delta^6 y_{-3} + \Delta^6 y_{-2}}{2} + \ldots \\
\ldots + \frac{q(q-1)(q+1)(q-2)(q+2)\ldots(q-n)(q+n-1)}{(2n)!}\frac{\Delta^{2n} y_{-n} + \Delta^{2n} y_{-n+1}}{2} +
\end{aligned}
$$

$$+\frac{\left(q-\frac{1}{2}\right)q(q-1)(q+1)(q-2)(q+2)..(q-n)(q+n-1)}{(2n+1)!}\Delta^{2n+1}y_{-n}, \quad (5.11)$$

где $q=\dfrac{x-x_0}{h}$.

Остаточный член интерполяционной формулы Бесселя, определяющий погрешность интерполяции, вычисляется по формуле

$$R(x)=\frac{h^{2n+2}}{(2n+2)!}f^{(2n+2)}(\xi)q\left(q^2-1^2\right)\left(q^2-2^2\right)..\left(q^2-n^2\right)\left(q-(n+1)\right), \quad (5.12)$$

где $\xi\in[x_0-nh,\ x_0+(n+1)h]$.

Давая общую характеристику рассмотренным интерполяционным формулам, можно отметить следующее: при построении интерполяционных формул Ньютона в качестве начального значения x_0 выбирается первый или последний узел интерполирования, для центральных же формул интерполирования (Гаусса, Стирлинга, Бесселя) начальный узел является средним.

Более детальное исследование интерполяционных формул показывает, что при $|q|\le 0{,}25$ целесообразно применять формулу Стирлинга, а при $0{,}25\le q\le 0{,}75$ – формулу Бесселя. Первую и вторую интерполяционные формулы Ньютона выгодно применять тогда, когда интерполирование производится в начале или соответственно в конце таблицы и нужных центральных разностей не хватает.

5.5. ИНТЕРПОЛЯЦИОННАЯ ФОРМУЛА ЛАГРАНЖА

Рассмотренные ранее интерполяционные формулы могут быть использованы лишь в случае равноотстоящих узлов интерполяции (за исключением формулы Ньютона для неравноотстоящих узлов). Для произвольно заданных узлов интерполяции пользуются более общей интерполяционной формулой Лагранжа.

Пусть на отрезке $[a,\ b]$ даны $n+1$ различных значений аргумента: $x_0, x_1, x_2, \ldots, x_n$ и для функции $y=f(x)$ известны соответствующие значения: $y_0=f(x_0),\ y_1=f(x_1),\ \ldots,\ y_n=f(x_n)$. Требуется построить полином $L_n(x)$ степени не выше n, имеющий в заданных узлах $x_0, x_1, \ldots, x_n$ те же значения, что и функция $f(x)$, т.е. такой, что

$$L_n(x_i)=y_i,\ i=0,\ 1,\ 2,\ \ldots,\ n.$$

Этим условиям соответствует полином вида:

$$L_n(x)=\sum_{i=0}^{n} y_i \frac{(x-x_0)(x-x_1)...(x-x_{i-1})(x-x_{i+1})...(x-x_n)}{(x_i-x_0)(x_i-x_1)...(x_i-x_{i-1})(x_i-x_{i+1})...(x_i-x_n)}. \quad (5.13)$$

Выражение (5.13) является *интерполяционной формулой Лагранжа*.

Все построенные выше формулы можно получить из формулы (5.13) при соответствующем выборе узлов интерполяции. В частности, если узлы интерполяции – равноотстоящие, то интерполяционный полином Лагранжа совпадает с соответствующим интерполяционным полиномом Ньютона.

Входящие в формулу (5.13) коэффициенты

$$L_i^{(n)}(x)=\frac{(x-x_0)(x-x_1)...(x-x_{i-1})(x-x_{i+1})...(x-x_n)}{(x_i-x_0)(x_i-x_1)...(x_i-x_{i-1})(x_i-x_{i+1})...(x_i-x_n)}$$

называются *коэффициентами Лагранжа*.

Оценить погрешность интерполирования табличной функции $y=f(x)$ формулой Лагранжа можно с помощью остаточного члена, вычисляемого по формуле

$$R(x)=\frac{f^{(n+1)}(\xi)}{(n+1)!}(x-x_0)(x-x_1)...(x-x_n), \quad (5.14)$$

где ξ лежит внутри отрезка $[a, b]$.

5.6. ИНТЕРПОЛИРОВАНИЕ СПЛАЙНАМИ

Интерполирование одним из приведённых выше многочленов на всём отрезке $[a, b]$ с использованием большого числа узлов интерполяции часто приводит к плохому приближению, что объясняется сильным накоплением погрешностей в процессе вычислений. Во избежание этого поступают следующим образом: весь отрезок $[a, b]$ разбивают на частичные отрезки $[x_{i-1}, x_i]$, $i=\overline{1, n}$ и на каждом из частичных отрезков приближённо заменяют функцию $y=f(x)$ многочленом невысокой степени. Одним из способов такого интерполирования функции $y=f(x)$ на всём отрезке $[a, b]$ является интерполирование с помощью сплайн-функций (сплайнов).

Сплайном называется функция, которая вместе с несколькими производными непрерывна на всём заданном отрезке $[a, b]$, а на каждом частичном отрезке $[x_{i-1}, x_i]$ в отдельности является некоторым алгебраическим многочленом.

Максимальная по всем частичным отрезкам степень многочленов называется степенью сплайна, а разность между степенью сплайна и порядком наивысшей непрерывной на $[a, b]$ производной – дефектом сплайна.

Рассмотрим простейшую задачу интерполирования непрерывно кусочно-линейной функцией (ломанной) – сплайном первой степени (линейным сплайном). Дефект такого сплайна равен единице, так как непрерывна только сама функция (нулевая производная), а первая производная уже разрывна.

Поставим вопрос построения линейного сплайна $S_1(x)$, совпадающего в точках x_0, x_1, …, x_n с функцией $f(x)$. Получится система уравнений:

$$S_{i,1}(x_{i-1}) = f(x_{i-1}),\ i = \overline{1,\ n}; \qquad (5.15)$$

$$S_{i,1}(x_i) = f(x_i),\ i = \overline{1,\ n}.$$

Так как линейный сплайн на отрезке $[x_{i-1},\ x_i]$ ищется в виде $S_{i,1}(x) = a_{i,0} + a_{i,1}x$, то система (5.15) относительно коэффициентов отдельных многочленов перепишется в виде:

$$S_{i,1}(x_{i-1}) = a_{i,0} + a_{i,1}x_{i-1} = f(x_{i-1}),\ i = \overline{1,\ n};$$

$$S_{i,1}(x_i) = a_{i,0} + a_{i,1}x_i = f(x_i),\ i = \overline{1,\ n},$$

отсюда находим

$$a_{i,1} = (f(x_i) - f(x_{i-1}))/(x_i - x_{i-1}),\ i = \overline{1,n}; \qquad (5.16)$$

$$a_{i,0} = f(x_{i-1}) - a_{i,1}x_{i-1}.$$

Таким образом, линейный сплайн на каждом частичном отрезке $[x_{i-1},\ x_i]$ может быть найден в виде выражения:

$$S_{i,1}(x) = \frac{x_i - x}{h_i} f(x_{i-1}) + \frac{x - x_{i-1}}{h_i} f(x_i),\ i = \overline{1,\ n}, \qquad (5.17)$$

где $h_i = x_i - x_{i-1}$.

На практике наиболее широкое распространение получили сплайны третьей степени $S_3(x)$ – функции, удовлетворяющие следующим условиям:

1) на каждом частичном отрезке $[x_{i-1},\ x_i]$, $i = \overline{1,\ n}$ функция $S_{i,3}(x)$ является многочленом третьей степени;

2) функция $S_3(x)$, а также её первая и вторая производные непрерывны на отрезке $[a,\ b]$, т.е. дефект сплайна равен единице;

3) $S_3(x_i) = f(x_i),\ i = \overline{0,\ n}$.

Будем искать функцию $S_{i,3}(x)$ на каждом из отрезков $[x_{i-1},\ x_i]$, $i=\overline{1,\ n}$ в виде многочлена третьей степени следующего вида:

$$S_{i,3}(x)=a_i+b_i(x-x_i)+\frac{c_i}{2}(x-x_i)^2+\frac{d_i}{6}(x-x_i)^3, \qquad (5.18)$$

где a_i, b_i, c_i, d_i – коэффициенты, подлежащие определению.

Приведём без доказательств расчётные формулы для вычисления этих коэффициентов.

$$a_i=f(x_i),\ i=\overline{1,\ n}; \qquad (5.19)$$

$$\frac{h_i}{6}c_{i-1}+\frac{h_i+h_{i+1}}{3}c_i+\frac{h_{i+1}}{6}c_{i+1}=\frac{f(x_{i+1})-f(x_i)}{h_{i+1}}-\frac{f(x_i)-f(x_{i-1})}{h_i}; \qquad (5.20)$$

$$i=\overline{1,\ n-1},\ c_0=c_n=0;$$

$$d_i=\frac{c_i-c_{i-1}}{h_i},\ i=\overline{1,\ n}; \qquad (5.21)$$

$$b_i=\frac{h_i}{2}c_i-\frac{h_i^2}{6}d_i+\frac{f(x_i)-f(x_{i-1})}{h_i},\ i=\overline{1,\ n}, \qquad (5.22)$$

где $h_i=x_i-x_{i-1}$.

Таким образом, для определения кубического сплайна вида (5.18) на отрезке $[x_{i-1},\ x_i]$ необходимо решить систему линейных уравнений (5.20). Эта система имеет единственное решение, которое может быть найдено методом исключения, либо в силу особенности этой системы (матрица системы трёхдиагональная) – методом прогонки (частный случай метода исключения). Оставшиеся коэффициенты кубического сплайна a_i, b_i, d_i находятся по формулам (5.19), (5.20), (5.21) соответственно.

Если в качестве многочлена третьей степени, определяющего кубический сплайн, используется многочлен вида $P_3(x)=S_3(x)=a_0+a_1x+a_2x^2+a_3x^3$, то кубический сплайн может быть построен в соответствии с формулой:

$$S_{i,3}(x)=c_{i-1}\frac{(x_i-x)^3}{6h_i}+c_i\frac{(x-x_{i-1})^3}{6h_i}+\left(f(x_{i-1})-\frac{c_{i-1}h_i^2}{6}\right)\frac{x_i-x}{h_i}+$$
$$+\left(f(x_i)-\frac{c_ih_i^2}{6}\right)\frac{x-x_{i-1}}{h_i},\ i=\overline{1,\ n}, \qquad (5.23)$$

где коэффициенты c_i могут быть определены также из решения системы (5.20).

Нетрудно убедиться, что значения коэффициентов c_i, находимые в результате решения системы (5.20), совпадают со значениями второй производной функции $S_3(x)$ в точках x_i, т.е. $c_i = S''_{i,3}(x_i),\ i=\overline{1,\ n}$.

Запишем ещё одну формулу для построения кубического сплайна, введя предварительно понятие наклона сплайна. *Наклоном сплайна* в узле x_i называется величина $m_i = S'_{i,3}(x_i)$. Тогда кубический сплайн на частном отрезке $[x_{i-1},\ x_i]$ может быть определён следующим образом:

$$S_{i,3}(x)=\frac{(x_i-x)^2(2(x-x_{i-1})+h_i)}{h_i^3}f(x_{i-1})+\frac{(x-x_{i-1})^2(2(x_i-x)+h_i)}{h_i^3}f(x_i)+$$
$$+\frac{(x_i-x)^2(x-x_{i-1})}{h_i^2}m_{i-1}+\frac{(x-x_{i-1})^2(x_i-x)}{h_i^2}m_i. \qquad (5.24)$$

Таким образом, чтобы задать кубический сплайн $S_{i,3}(x)$ на всём отрезке $[a,\ b]$, необходимо задать в $n + 1$ узлах x_i его значения f_i и наклоны $m_i,\ i=\overline{1,\ n}$.

Для определения наклонов интерполяционного кубического сплайна существует несколько способов.

1. Упрощённый способ. Получается сплайн с дефектом, равным 2.

$$m_i=\frac{f(x_{i+1})-f(x_{i-1})}{x_{i+1}-x_{i-1}},\ i=\overline{1,\ n-1}; \qquad (5.25)$$

$$m_0=\frac{4f(x_1)-f(x_2)-3f(x_0)}{x_2-x_0};\ m_n=\frac{3f(x_n)-f(x_{n-2})-4f(x_{n-1})}{x_n-x_{n-2}}.$$

2. Если известны значения $f'(x_i)$, то можно положить $m_i = f'(x_i),\ i=\overline{0,\ n}$.

Способы 1 и 2 называются локальными, поскольку с их помощью на каждом отрезке $[x_{i-1},\ x_i]$ сплайн строится отдельно. Однако при этом соблюдается непрерывность первой производной $S'_{i,3}(x)$, а непрерывность второй производной $S''_{i,3}(x)$ – не гарантируется.

3. Глобальный способ. Он позволяет построить сплайн с дефектом, не большим 1, т.е. обеспечивает непрерывность второй производной $S''_{i,3}(x)$.

Для определения наклонов этим способом необходимо решить систему линейных уравнений:

$$m_{i-1}+4m_i+m_{i+1}=\frac{3(f(x_{i+1})-f(x_{i-1}))}{h_i};\ i=\overline{1,\ n-1}. \qquad (5.26)$$

Однако система (5.26), состоящая из $(n-1)$ уравнения, содержит $n+1$ неизвестную. Поэтому её необходимо дополнить краевыми условиями. Как правило, используют следующие краевые условия:

а) если известны $f'(x_0)$ и $f'(x_n)$, то можно положить

$$m_0 = f'(x_0) \text{ и } m_n = f'(x_n); \tag{5.27}$$

б) $$m_0 = \frac{1}{2(x_3 - x_0)}\left(-11 f(x_0) + 18 f(x_1) - 9 f(x_2) + 2 f(x_3)\right); \tag{5.28}$$

$$m_n = \frac{1}{2(x_n - x_{n-3})}\left(11 f(x_n) - 18 f(x_{n-1}) + 9 f(x_{n-2}) - 2 f(x_{n-3})\right);$$

в) если известны $f''(x_0)$ и $f''(x_n)$, то

$$m_0 = -\frac{m_1}{2} + \frac{3}{2}\frac{f(x_1) - f(x_0)}{x_1 - x_0} - \frac{x_1 - x_0}{4} f''(x_0); \tag{5.29}$$

$$m_n = -\frac{m_{n-1}}{2} + \frac{3}{2}\frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}} + \frac{x_n - x_{n-1}}{4} f''(x_n).$$

Краевые условия (5.27) – (5.29) можно комбинировать, т.е. выбирать в левом и правом крайних узлах независимо.

Система (5.26) при всех рассмотренных краевых условиях имеет единственное решение, для нахождения которого могут быть применены методы итераций и прогонки.

6. АППРОКСИМАЦИЯ ФУНКЦИЙ

В ходе автоматизированной обработки результатов испытаний технических систем возникает необходимость аппроксимации функций, заданных экспериментальными таблицами данных. Одной из простейших задач аппроксимации, когда от аппроксимирующей функции требуется прохождение через все точки, заданные таблицей, является задача интерполяции.

На практике интерполяционные формулы применяются лишь в тех случаях, когда ошибки в табличных данных можно не учитывать, и число n точек x_i является небольшим. Это объясняется тем, что в реальных задачах ошибки в экспериментальных данных необходимо учитывать. Кроме того, при больших n интерполяционные формулы становятся громоздкими, что влечёт за собой определённые трудности при их решении.

В таких условиях задача приближения функции может быть сформулирована следующим образом. Требуется построить функцию $F(x)$, принадлежащую известному классу, такую, что значение функции $F(x)$ в точках x_i не слишком сильно отличается от заданных значений табличной функции $y_i = f(x_i)$, т.е. разности $y_i - F(x_i)$ – достаточно малы. В такой постановке задача аппроксимации может быть решена с помощью *метода наименьших квадратов*.

Предположим, что функция $y = f(x)$ задана на отрезке $[a, b]$ экспериментальными значениями $y_i = f(x_i)$, $i = \overline{0, n}$. Аппроксимирующую функцию будем искать в виде линейной модели

$$F(x) = a_0\varphi_0(x) + a_1\varphi_1(x) + \ldots + a_k\varphi_k(x). \qquad (6.1)$$

Тогда согласно методу наименьших квадратов (МНК) наилучшее приближение функции (6.1) к табличной функции $y = f(x)$ будет достигаться при минимальном значении следующей функции невязки:

$$\Phi(a_0, a_1, \ldots, a_k) = \sum_{i=0}^{n} (y_i - F(x_i))^2 \to \min. \qquad (6.2)$$

Задача (6.2), т.е. нахождение таких значений коэффициентов (a_0, a_1, …, a_k), при которых функция Ф достигает минимального значения, может быть решена с использованием классических методов математического анализа. Тогда условие минимума Ф определяется системой уравнений

$$\frac{\partial \Phi}{\partial a_j} = 0, \quad j = \overline{0, k}, \qquad (6.3)$$

так как функция Ф может быть записана в виде

$$\Phi = \sum_{i=0}^{n} \left(a_0\varphi_0(x_i) + a_1\varphi_1(x_i) + \ldots + a_k\varphi_k(x_i) - y_i\right)^2 ,$$

то условия минимума функции нескольких переменных (6.3) эквивалентны следующей системе уравнений:

$$\frac{\partial \Phi}{\partial a_j} = 2\sum_{i=0}^{n} \left(a_0\varphi_0(x_i) + a_1\varphi_1(x_i) + \ldots + a_k\varphi_k(x_i) - y_i\right)\varphi_j(x_i) = 0, \;\; j = \overline{0, k} . \quad (6.4)$$

Эти (k + 1) уравнений представляют собой систему линейных алгебраических уравнений, в которой в качестве неизвестных выступают коэффициенты линейной модели (a_0, a_1, …, a_k). В матричном виде эта система может быть представлена так:

$$\begin{pmatrix} \sum_{i=0}^{n}\varphi_0^2(x_i) & \sum_{i=0}^{n}\varphi_0(x_i)\varphi_1(x_i) & \ldots & \sum_{i=0}^{n}\varphi_0(x_i)\varphi_k(x_i) \\ \sum_{i=0}^{n}\varphi_1(x_i)\varphi_0(x_i) & \sum_{i=0}^{n}\varphi_1^2(x_i) & \ldots & \sum_{i=0}^{n}\varphi_1(x_i)\varphi_k(x_i) \\ \ldots & \ldots & \ldots & \ldots \\ \sum_{i=0}^{n}\varphi_k(x_i)\varphi_0(x_i) & \sum_{i=0}^{n}\varphi_k(x_i)\varphi_2(x_i) & \ldots & \sum_{i=0}^{n}\varphi_k^2(x_i) \end{pmatrix} \times \begin{pmatrix} a_0 \\ a_1 \\ \ldots \\ a_k \end{pmatrix} = \quad (6.5)$$

$$= \begin{pmatrix} \sum_{i=0}^{n}\varphi_0(x_i)y_i \\ \sum_{i=0}^{n}\varphi_1(x_i)y_i \\ \ldots \\ \sum_{i=0}^{n}\varphi_k(x_i)y_i \end{pmatrix}.$$

Так как элементы матрицы в левой части и вектора-столбца в правой определяются табличными данными, то система (6.5) может быть решена. В качестве функций $\varphi_j(x)$ можно выбирать любые функции, лишь бы они отвечали требованию линейности относительно своих коэффициентов. Фактически выбор функции должен осуществляться с учётом специфики табличных данных, т.е. их периодичности, экспоненциального или логарифмического характера, наличия асимптотики.

На практике очень часто в качестве функций $\varphi_j(x)$ принимаются следующие функции: $\varphi_j(x) = x^j = \prod_{m=1}^{j} x$. Тогда линейная модель (6.1) представляется в виде полинома:

$$F(x) = a_0 + a_1 x + a_2 x^2 + ... + a_k x^k, \qquad (6.6)$$

где $\varphi_0(x) = 1,\ \varphi_1(x) = x, \varphi_2(x) = x^2, ...,\ \varphi_k(x) = x^k$.

Коэффициенты аппроксимирующего полинома (6.6) (a_0, a_1, …, a_k) находятся из системы линейных уравнений (6.5) после предварительной замены функций $\varphi_j(x)$. Например, если в качестве аппроксимирующей модели взять прямую, которая выражается многочленом первой степени $F(x) = a_1 x + a_0$, то значения a_0 и a_1 могут быть найдены из системы уравнений:

$$\begin{cases} a_1 \sum_{i=0}^{n} x_i^2 + a_0 \sum_{i=0}^{n} x_i = \sum_{i=0}^{n} x_i y_i; \\ a_1 \sum_{i=0}^{n} x_i + a_0 \sum_{i=0}^{n} 1 = \sum_{i=0}^{n} y_i. \end{cases}$$

Для аппроксимирующей линии параболического типа $F(x) = a_0 + a_1 x + a_2 x^2$ система (6.5) представляет собой систему трёх линейных уравнений:

$$\begin{cases} a_2 \sum_{i=0}^{n} x_i^4 + a_1 \sum_{i=0}^{n} x_i^3 + a_0 \sum_{i=0}^{n} x_i^2 = \sum_{i=0}^{n} x_i^2 y_i; \\ a_2 \sum_{i=0}^{n} x_i^3 + a_1 \sum_{i=0}^{n} x_i^2 + a_0 \sum_{i=0}^{n} x_i = \sum_{i=0}^{n} x_i y_i; \\ a_2 \sum_{i=0}^{n} x_i^2 + a_1 \sum_{i=0}^{n} x_i + a_0 (n+1) = \sum_{i=0}^{n} y_i. \end{cases}$$

Изложенный способ аппроксимации табличных функций полиномами имеет два существенных недостатка:

1) для отыскания коэффициентов многочлена приходится решать систему из $(k+1)$ уравнений, что при больших k затруднительно;

2) если, выбрав k и построив многочлен наилучшего приближения, оказалось, что точность приближения недостаточна, то увеличив k, придётся заново повторить все вычисления.

От указанных недостатков можно избавиться, если вместо произвольной линейной модели (6.1) использовать линейную систему ортогональных полиномов.

Система полиномов $\{\varphi_j(x)\}\Big|_{j=0}^{\infty}$ будет являться ортогональной на отрезке $[a, b]$, если $(\varphi_j(x), \varphi_m(x)) = \sum_{i=0}^{n} \varphi_j(x_i)\varphi_m(x_i) = 0, \quad j \neq m$.

Для полиномов такого вида матрица коэффициентов из уравнения (6.5) примет диагональный вид. Тогда коэффициенты a_0, a_1, …, a_k могут быть найдены из простых выражений:

$$a_j = \sum_{i=0}^{n} \varphi_j(x_i) y_i \Big/ \sum_{i=0}^{n} \varphi_j^2(x_i), \quad j = \overline{0, k}. \tag{6.7}$$

В настоящее время разработано несколько подходов к построению систем ортогональных полиномов. Одной из наиболее простых является система полиномов Чебышева.

Пусть нам дано $n + 1$ узлов x_0, x_1, x_2, …, x_n, где $x_i - x_{i-1} = h$. Вводя замену $q = \dfrac{x - x_0}{h}$, точки x_0, x_1, x_2, …, x_n перейдут соответственно в 0, 1, …, n. Тогда систему ортогональных полиномов Чебышева можно получить с помощью следующих рекуррентных формул:

$$\frac{(k+1)(n-k)}{2(2k+1)} P_{k+1}(q) = \left(\frac{h}{2} - q\right) P_k(q) - \frac{k(n+k+1)}{2(2k+1)} P_{k-1}(q), \tag{6.8}$$

где $P_0(q) = 1$, $P_1(q) = 1 - \dfrac{2q}{n}$.

Из формул (6.8) можно получить, что

$$P_2(q) = 1 - \frac{6q}{n-1} - \frac{6q^2}{n(n-1)};$$

$$P_3(q) = 1 - \frac{12n^2 - 6n + 4}{n(n-1)(n-2)} q + \frac{30q^2}{(n-1)(n-2)} - \frac{20q^3}{n(n-1)(n-2)}.$$

На отрезке $[-1, 1]$ может быть построена система ортогональных полиномов, называемых многочленами Лежандра, для которых справедлива рекуррентная формула:

$$(k+1)P_{k+1}(q) - (2k+1)qP_k(q) + kP_{k-1}(q) = 0, \tag{6.9}$$

где $P_0(q) = 1$, $P_1(q) = q$, $q = \dfrac{2(x - x_0)}{x_n - x_0} - 1$.

При решении практических задач степень аппроксимирующего многочлена обычно неизвестна. Если функция $y = f(x)$ аппроксимируется с помощью полинома (6.6), то выбор его степени часто осуществляется следующим образом. Начиная с некоторого малого числа k_0 (например, $k_0 = 1$) выбирается возрастающая последовательность целых чисел k_1, k_2, k_3, … и для этих степеней путём решения системы (6.5) вычисляются коэффициенты полинома. Для каждого значения k_j ($j = 1, 2, \ldots$) вычисляются остаточные дисперсии:

$$\sigma^2 = \frac{1}{n-k-1}\sum_{i=0}^{n}\left(y_i - F(x_i)\right)^2 .$$

При увеличении k остаточная дисперсия обычно убывает, а позже наступает момент, когда она начинает возрастать. Поэтому степень аппроксимирующего полинома k выбирается равной значению k_m, при котором остаточная дисперсия является минимальной.

При аппроксимации обычными полиномами на каждом шаге все коэффициенты аппроксимирующего многочлена приходится вычислять заново. Если для аппроксимации функции $y = f(x)$ используются ортогональные полиномы, то при переходе от полинома степени k к полиному степени $k + 1$ приходится вычислять только коэффициент a_{k+1} при полиноме $\varphi_{k+1}(x)$, а все остальные коэффициенты (a_0, …, a_k) остаются без изменений.

7. ЧИСЛЕННОЕ ИНТЕГРИРОВАНИЕ

Если функция $f(x)$ непрерывна на отрезке $[a, b]$ и известна её производная $F(x)$, то определённый интеграл от этой функции в пределах от *a* до *b* может быть вычислен по формуле Ньютона–Лейбница

$$\int_a^b f(x)dx = F(b) - F(a), \qquad (7.1)$$

где $F'(x) = f(x)$.

Однако во многих случаях функция $F(x)$ не может быть найдена с помощью элементарных средств или является слишком сложной. Кроме того, на практике подынтегральная функции $f(x)$ часто задаётся таблично и тогда само понятие первообразной теряет смысл. Таким образом, вычисление определённого интеграла по формуле (7.1) зачастую бывает затруднительным или даже невозможным.

В этой ситуации важное значение приобретают приближённые и в первую очередь численные методы вычисления определённых интегралов. Задача численного интегрирования функции заключается в вычислении значения определённого интеграла на основании ряда значений подынтегральной функции. Численное вычисление однократного интеграла называется механической квадратурой, а соответствующие формулы – квадратурными.

Обычный приём механической квадратуры состоит в том, что данную функцию $f(x)$ на рассматриваемом отрезке $[a, b]$ заменяют интерполирующей или аппроксимирующей функцией простого вида (например, полиномом). Однако может случиться, что подынтегральная функция исходного интеграла $f(x)$ плохо приближается многочленами. В этом случае её заменяют следующим произведением: $f(x) = P(x)\varphi(x)$, где $P(x) > 0$ – некоторая функция простого вида, хорошо приближаемая многочленами (весовая функция), и $\varphi(x)$ – достаточно гладкая функция. Тогда задача численного интегрирования заключается в вычислении интеграла

$$\int_a^b f(x)dx = \int_a^b P(x)\varphi(x)dx. \qquad (7.2)$$

Квадратурные формулы для вычисления интеграла (7.2) получим путём замены $\varphi(x)$ интерполяционным многочленом *n*-й степени на всём отрезке $[a, b]$ (такие формулы называются квадратурными фор-

мулами интерполяционного типа, их точность возрастает с увеличением узлов интерполирования).

Воспользуемся в качестве интерполяционного многочлена многочленом Лагранжа:

$$L_n(x)=\sum_{i=0}^{n}\frac{\omega(x)}{(x-x_i)\omega'(x_i)}\varphi(x_i),$$

где $\omega(x)=\prod_{j=0}^{n}(x-x_j)$; $\omega'(x_i)=\prod_{\substack{j=0\\ i\neq j}}^{n}(x_i-x_j)$.

Тогда приближённая квадратурная формула для вычисления интеграла (7.2) будет иметь вид

$$\int_a^b P(x)\varphi(x)dx=\sum_{i=0}^{n}A_i\varphi(x_i)+R, \qquad (7.3)$$

где

$$A_i=\int_a^b\frac{P(x)\omega(x)}{(x-x_i)\omega'(x_i)}dx,\ \ i=0,\,1,\,...,\,n, \qquad (7.4)$$

где R – остаточный член квадратурной формулы.

Получим явные выражения для коэффициентов A_i. Разобьём для этого отрезок $[a,\,b]$ с помощью равноотстоящих точек $x_0=a$, $x_i=x_0+ih$ $\left(i=\overline{1,\,n-1}\right)$, $x_n=b$ на n равных частей, причём $h=(b-a)/n$.

Введём обозначение $q=\dfrac{x-x_0}{h}$. Тогда формула (7.4) может быть переписана в виде

$$A_i=(b-a)\frac{(-1)^{n-i}}{i!(n-i)!n}\int_0^n p(a+qh)\frac{q(q-1)...(q-n)}{q-i}dq,\ \ i=\overline{0,\,n}, \qquad (7.5)$$

где выражение

$$\frac{(-1)^{n-i}}{i!(n-i)!n}\int_0^n p(a+qh)\frac{q(q-1)...(q-n)}{q-i}dq=H_i,\ \ i=\overline{0,\,n}, \qquad (7.6)$$

записанное для интерполяционного многочлена Лагранжа степени n, называется коэффициентами Котеса, а квадратурные формулы

$$\int_a^b P(x)\varphi(x)dx=(b-a)\sum_{i=0}^{n}H_i\varphi(x_i)+R \qquad (7.7)$$

квадратурными формулами Ньютона–Котеса.

Для коэффициентов Котеса справедливы следующие соотношения:

$$\sum_{i=0}^{n} H_i = 1; \quad H_i = H_{n-i}.$$

Рассмотрим некоторые из формул Ньютона–Котеса, использующие интерполяционные многочлены невысоких степеней. При этом будем полагать, что функция $f(x)$ в формуле (7.2) является достаточно гладкой функцией, т.е. можно положить, что $f(x) \equiv \varphi(x)$. Тогда весовую функцию можно принять: $p(x) \equiv 1$.

7.1. ФОРМУЛЫ ПРЯМОУГОЛЬНИКОВ

Вычислим интеграл $\int_a^b f(x)dx$, используя для этого квадратурную формулу Ньютона–Котеса степени $n = 0$. Тогда согласно (7.6) $H_0 = 1$ и, используя (7.7), получим

$$\int_a^b f(x)dx = (b-a)f(a) + R. \tag{7.8}$$

При построении формулы (7.8) исходим из предположения, что функция $f(x) = \text{const}$ на отрезке $[a, b]$. Однако на практике, когда отрезок $[a, b]$ достаточно велик, а подынтегральная функция $f(x)$ не является постоянной на $[a, b]$, интерполирование её полиномом нулевой степени сразу на всём отрезке может привести к большим вычислительным погрешностям. В такой ситуации поступают следующим образом. Интервал интегрирования $[a, b]$ разбивают на m элементарных участков равноотстоящими точками: $x_i = a + ih$; $i = \overline{0, m}$; $h = \dfrac{b-a}{m}$ и применяют формулу (7.8) к каждому из элементарных отрезков $[x_i, x_{i+1}]$, $i = \overline{0, m-1}$. В итоге получается следующая составная формула:

$$\int_a^b f(x)dx = h\sum_{i=0}^{m-1} f(x_i) + R. \tag{7.9}$$

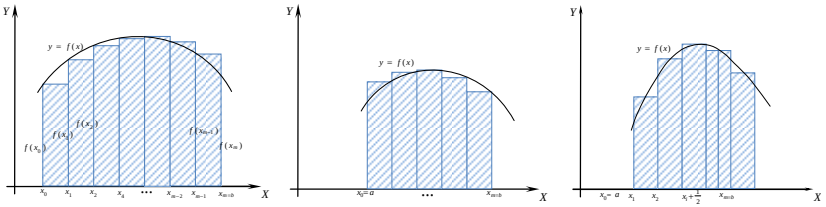
Формула (7.9) носит название формулы левых прямоугольников. Аналогично можно получить формулу правых прямоугольников

$$\int_a^b f(x)dx = h\sum_{i=1}^{m} f(x_i) + R \tag{7.10}$$

и формулу центральных прямоугольников

$$\int_a^b f(x)dx = h\sum_{i=0}^{m-1} f(x_i + h/2) + R. \qquad (7.11)$$

Геометрическая интерпретация методов прямоугольников (рис. 7.1) заключается в замене площади под подынтегральной функцией суммой площадей прямоугольников с основаниями h и высотой $f(x_i)$.



а) *б*) *в*)

Рис. 7.1. Геометрическая интерпретация методов прямоугольников: *а* – левых; *б* – правых; *в* – центральных

Остаточный член, определяющий погрешность вычисления интеграла методом прямоугольников, рассчитывается по формуле

$$R = -\frac{h^2(b-a)}{24} f''(\xi), \text{ где } \xi \in [a, b].$$

7.2. ФОРМУЛА ТРАПЕЦИЙ

Применим формулу (7.6) для $n = 1$:

$$H_0 = -\int_0^1 \frac{q(q-1)}{q} dq = \frac{1}{2}; \quad H_1 = \int_0^1 q dq = \frac{1}{2}.$$

Тогда формула Ньютона–Котеса (7.7) может быть записана в виде:

$$\int_a^b f(x)dx = (b-a)\left(\frac{f(a)}{2} + \frac{f(b)}{2}\right) + R. \qquad (7.12)$$

Разделим интервал интегрирования на m равных частей с помощью равноотстоящих точек $x_i = a + ih$; $i = \overline{0, m}$; $h = \frac{b-a}{m}$ и к каждому из них применим формулу (7.12). В результате получается общая формула трапеций:

$$\int_a^b f(x)dx = \frac{h}{2}(f(x_0) + f(x_m)) + h(f(x_1) + f(x_2) + \ldots + f(x_{m-1})) + R. \quad (7.13)$$

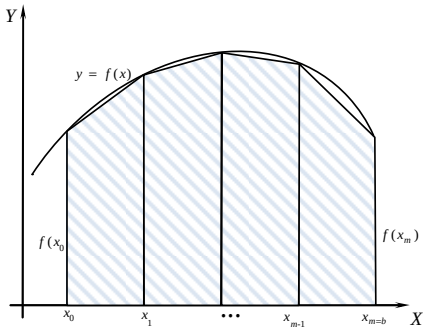


Рис. 7.2. Геометрическая иллюстрация метода трапеций

Геометрически формула (7.13) получается при замене площади под подынтегральной функцией суммой площадей прямоугольных трапеций, высоты которых равны h, а основания совпадают со значениями функции $f(x)$ в точках x_i, $i = \overline{0,\ m}$ (рис. 7.2).

Остаточный член в квадратурной формуле (7.13) равен

$$R = -\frac{h^2(b-a)}{12} f''(\xi), \text{ где } \xi \in [a, b].$$

7.3. ФОРМУЛА СИМПСОНА

Воспользуемся формулой (7.6) для вычисления коэффициентов квадратурной формулы Ньютона–Котеса, использующей интерполяционный полином степени $n = 2$:

$$H_0 = \frac{(-1)^{2-0}}{0!(2-0)!2}\int_0^2 \frac{q(q-1)(q-2)}{q}dq = \frac{1}{4}\left(\frac{8}{3} - 6 + 4\right) = \frac{1}{6};$$

$$H_1 = \frac{(-1)^{2-1}}{1!(2-1)!2}\int_0^2 \frac{q(q-1)(q-2)}{q-1}dq = -\frac{1}{2}\left(\frac{8}{3} - 4\right) = \frac{2}{3};$$

$$H_2 = \frac{(-1)^{2-2}}{2!(2-2)!2}\int_0^2 \frac{q(q-1)(q-2)}{q-2}dq = -\frac{1}{4}\left(\frac{8}{3} - 2\right) = \frac{1}{6}.$$

С учётом найденных коэффициентов формула (7.7) примет вид

$$\int_a^b f(x)dx = (b-a)\left(\frac{1}{6}f(x_0) + \frac{2}{3}f(x_1) + \frac{1}{6}f(x_2)\right) + R. \qquad (7.14)$$

Интерполируя функцию $f(x)$ на интервале $[a, b]$ полиномом второй степени необходимы три точки, делящие отрезок $[a, b]$ на два участка, т.е. $b - a = 2h$. Тогда формула (7.14) перепишется в виде

$$\int_a^b f(x)dx = \frac{h}{3}\int_a^b (f(x_0) + 4f(x_1) + f(x_2)) + R. \qquad (7.15)$$

Для вывода общей формулы Симпсона прибегнем к тому же способу, что и в двух предыдущих случаях.

Пусть $m = 2k$ есть чётное число, и $f(x_i)$ есть значения подынтегральной функции $f(x)$ для равноотстоящих точек $a = x_0, x_1, x_2, ..., x_m = b$ с шагом $h = \frac{b-a}{m} = \frac{b-a}{2k}$.

Применяя формулу (7.15) к каждому удвоенному промежутку $[x_0, x_2]$, $[x_2, x_4]$, …, $[x_{2k-2}, x_{2k}]$ длиной $2h$, будем иметь:

$$\int_a^b f(x)dx = \frac{h}{3}(f(x_0) + f(x_{2k}) + 4(f(x_1) + f(x_3) + ... + f(x_{2k-1})) + 2(f(x_2) + f(x_4) + ... + f(x_{2k-2}))) + R. \qquad (7.16)$$

Формула (7.16) для вычисления определённого интеграла называется формулой Симпсона или формулой парабол.

Второе название этой формулы связано с тем, что на каждом из элементарных отрезков длиной $2h$ осуществляется замена подынтегральной функции параболой (полиномом второй степени).

Остаточный член в формуле Симпсона вычисляется по формуле

$$R = \frac{h^4(b-a)}{180} f^{(IV)}(\xi),$$

где $\xi \in [a, b]$.

7.4. ПРАВИЛО ТРЁХ ВОСЬМЫХ

Интерполируя подынтегральную функцию полиномом третьей степени ($n = 3$), можно получить соответствующую квадратурную формулу:

$$H_0 = \frac{(-1)^{3-0}}{0!(3-0)!3}\int_0^3 (q-1)(q-2)(q-3)dq = -\frac{1}{18}\left(\frac{81}{4} - 54 + \frac{99}{2} - 18\right) = \frac{1}{8};$$

$$H_1 = \frac{(-1)^{3-1}}{1!(3-1)!3}\int_0^3 q(q-2)(q-3)dq = \frac{1}{6}\left(\frac{81}{4} - \frac{135}{3} + 27\right) = \frac{3}{8};$$

$$H_2 = \frac{(-1)^{3-2}}{2!(3-2)!3}\int_0^3 q(q-1)(q-3)dq = -\frac{1}{6}\left(\frac{81}{4} - \frac{108}{3} + \frac{27}{2}\right) = \frac{3}{8};$$

$$H_3 = \frac{(-1)^{3-3}}{3!(3-3)!3}\int_0^3 q(q-1)(q-2)dq = \frac{1}{18}\left(\frac{81}{4} - 27 + 9\right) = \frac{1}{8}.$$

Тогда интеграл можно определить как

$$\int_a^b f(x)dx = \frac{(b-a)}{8}\left(f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)\right) + R.$$

С учётом того, что $(b-a) = 3h$ для полиномов третьей степени, то

$$\int_a^b f(x)dx = \frac{3h}{8}\left(f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)\right) + R. \qquad (7.17)$$

Для вывода общей формулы разобьём интервал интегрирования на элементарные отрезки системой из $m = 3k$ равноотстоящих точек:

$$x_i = a + ih;\ i = \overline{0, m};\ h = \frac{b-a}{m} = \frac{b-a}{3k}.$$

Применяя формулу (7.17) к каждому интервалу длиной $3h$: $[x_0,\ x_3]$, $[x_3,\ x_6]$, ..., $[x_{3k-3},\ x_{3k}]$, получим:

$$\int_a^b f(x)dx = \frac{3h}{8}\left(f(x_0) + f(x_{3k}) + 2\left(f(x_3) + f(x_6) + \ldots + f(x_{3k-3})\right) + \right. \qquad (7.18)$$
$$\left. + 3\left(f(x_1) + f(x_2) + f(x_4) + f(x_5) + \ldots + f(x_{3k-2}) + f(x_{3k-1})\right)\right) + R.$$

Формула (7.18) называется формулой Ньютона или правилом трёх восьмых.

Остаточный член в этой формуле имеет вид

$$R = \frac{(b-a)h^4}{80} f^{(\mathrm{IV})}(\xi),\ \text{где}\ \xi \in [a, b].$$

Приведённые выше квадратурные формулы Ньютона–Котеса являются наиболее часто используемыми. Дальнейшее увеличение порядка квадратурной формулы повышает сложность вычислений, хотя и даёт более точный результат.

Формулы Ньютона–Котеса высоких порядков $n \geq 10$ на практике используются крайне редко. Это связано с их численной неустойчивостью, возникающей из-за того, что коэффициенты Котеса при больших n имеют различные знаки, и приводящей к резкому возрастанию вычислительной погрешности.

7.5. ВЫБОР ШАГА ИНТЕГРИРОВАНИЯ

Эта задача заключается в выборе такого значения шага h, разбивающего интервал интегрирования $[a, b]$, который обеспечивал бы заданную точность ε вычисления определённого интеграла по выбранной формуле численного интегрирования.

Один способ решения этой задачи заключается в выборе шага по оценке остаточного члена.

Пусть требуется вычислить интеграл с точностью ε. Используя формулу соответствующего остаточного члена R, выбирают h таким, чтобы выполнялось неравенство $|R| < \varepsilon/2$. Затем вычисляют интеграл по приближённой формуле с полученным шагом. При этом вычисления следует производить с таким числом знаков, чтобы погрешность округления не превышала $\varepsilon/2$.

Однако отыскание производной подынтегральной функции нередко приводит к слишком громоздким вычислениям. Поэтому на практике часто используют другой способ – двойной пересчёт. Для этого вычисляют интеграл по выбранной квадратурной формуле дважды: сначала с некоторым шагом $h = h_0$, затем с шагом $h = h_0/2$, т.е. удваивая число m.

Обозначив результаты вычислений через I_m и I_{2m} соответственно, сравнивают их. Если $|I_m - I_{2m}| < \varepsilon$, где ε – допустимая погрешность, то полагают $I^* \approx I_{2m}$. Если же окажется, что $|I_m - I_{2m}| \geq \varepsilon$, то расчёт повторяют с шагом $h = h_0/4$.

7.6. КВАДРАТУРНЫЕ ФОРМУЛЫ ГАУССА

При выводе рассмотренных выше формул узлы интерполяции в квадратурных формулах задавались заранее. При этом, используя в квадратурной формуле $n + 1$ узел интерполяции, получали формулу, точную для алгебраических многочленов степени n. Однако оказывается, что за счёт выбора узлов можно получить квадратурные формулы, которые будут точными и для многочленов степени выше n. Такие квадратурные формулы называются квадратурными формулами наивысшей алгебраической степени точности или формулами Гаусса.

Рассмотрим функцию $y = f(t)$, заданную на стандартном промежутке $[-1; 1]$. Поставим задачу: подобрать такие точки t_1, t_2, …, t_n и коэффициенты A_1, A_2, …, A_n, чтобы квадратурная формула

$$\int_{-1}^{1} f(t)dt = \sum_{i=1}^{n} A_i f(t_i) + R_n \qquad (7.19)$$

была точной для всех полиномов $f(t)$ наивысшей возможной степени N.

Так как в нашем распоряжении имеется $2n$ постоянных t_i и $A_i\left(i=\overline{1,n}\right)$, то эта наивысшая степень в общем случае равна $N=2n-1$ (так как полином степени $2n-1$ определяется $2n$ коэффициентами).

Для обеспечения равенства (7.19) необходимо и достаточно, чтобы оно было верным при $f(t)=1,\ t,\ t^2,\ ...,\ t^{2n-1}$. В результате подстановки этих функций в (7.19) получается система из $2n$ нелинейных уравнений (7.20), содержащая $2n$ неизвестных t_i и A_i. Решение такой системы обычным путём представляет большие математические трудности.

$$\begin{cases} \sum\limits_{i=1}^{n} A_i = 2; \\ \sum\limits_{i=1}^{n} A_i t_i = 0; \\ \quad ... \\ \sum\limits_{i=1}^{n} A_i t_i^{2n-2} = \dfrac{2}{2n-1}; \\ \sum\limits_{i=1}^{n} A_i t_i^{2n-1} = 0. \end{cases} \tag{7.20}$$

Этих трудностей можно избежать, если в качестве точек t_i взять нули соответствующего полинома Лежандра:

$$P_n(x)=\frac{1}{2^n n!}\frac{d}{dx^n}\left[\left(x^2-1\right)^n\right],\ n=0,\ 1,\ 2,\ ...,$$

обладающего следующими свойствами:

1. $P_n(1)=1,\ P_n(-1)=(-1)^n$.

2. $\int\limits_{-1}^{1} P_n(x)Q_k(x)dx=0$, где $Q_k(x)$ – любой полином степени $k<n$.

3. Полином Лежандра $P_n(x)$ имеет n различных и действительных корней, расположенных на интервале $[-1;\ 1]$.

Действительно, положив $f(t)=t^k P_n(t)$, $k=\overline{0,n-1}$, где $P_n(t)$ – полином Лежандра, в силу свойства 2 имеем:

$$\int\limits_{-1}^{1} f(t)dt=\int\limits_{-1}^{1} t^k P_n(t)dt=\sum_{i=1}^{n} A_i t_i^k P_n(t_i)=0,$$

откуда $P_n(t_i)=0$.

7.1. Элементы формул Гаусса

n	i	t_i	A_i
1	1	0	2
2	1; 2	∓ 0,57735027	1
3	1; 3	∓ 0,77459667	5/9 = 0,55555556
	2	0	8/9 = 0,55555556
4	1; 4	∓ 0,86113631	0,34785484
	2; 3	∓ 0,33998104	0,65214516
5	1; 5	∓ 0,90617985	0,23692688
	2; 4	∓ 0,53846931	0,47862868
	3	0	0,56888889
6	1; 6	∓ 0,93246951	0,17132450
	2; 5	∓ 0,66120939	0,36076158
	3; 4	∓ 0,23861919	0,46791394
7	1; 7	∓ 0,94910791	0,12948496
	2; 6	∓ 0,74153119	0,27970540
	3; 5	∓ 0,40584515	0,38183006
	4	0	0,41795918
8	1; 8	∓ 0,96028986	0,10122854
	2; 7	∓ 0,79666648	0,22238104
	3; 6	∓ 0,52553242	0,31370664
	4; 5	∓ 0,18343464	0,36268378

Зная абсциссы t_i, коэффициенты A_i могут быть легко определены из линейной системы первых n уравнений системы (7.20). Таким образом, формула (7.19), где t_i – нули полинома Лежандра $P_n(t)$ и $A_i\left(i=\overline{1,n}\right)$ определяются из системы (7.20), называется квадратурной формулой Гаусса. Для этих формул существуют специальные таблицы, содержащие значения t_i и A_i при различных степенях n (табл. 7.1).

Неудобство применения квадратурной формулы Гаусса состоит в том, что абсциссы точек t_i и коэффициенты A_i – иррациональные числа. Этот недостаток отчасти искупается её высокой точностью при сравнительно малом числе ординат.

В общем случае для вычисления интеграла $\int_a^b f(x)dx$ с использованием квадратурной формулы Гаусса необходимо сделать замену переменной $x = \frac{b+a}{2} + \frac{b-a}{2}t$.

Тогда получим: $\int_a^b f(x)dx = \frac{b-a}{2}\int_{-1}^{1} f\left(\frac{b+a}{2} + \frac{b-a}{2}t\right)dt$.

Применяя к последнему интегралу квадратурную формулу Гаусса (7.19), будем иметь:

$$\int_a^b f(x)dx = \frac{b-a}{2}\sum_{i=1}^{n} A_i f\left(\frac{b+a}{2} + \frac{b-a}{2}t_i\right) + R_n, \qquad (7.21)$$

где остаточный член определяется в соответствии с выражением:

$$R_n = \frac{(b-a)^{2n+1}(n!)^4 f^{2n}(\xi)}{((2n)!)^3(2n+1)}, \ \xi \in [a, b].$$

7.7. МЕТОД МОНТЕ–КАРЛО

На практике очень часто приходится иметь дело с задачами, для которых построение детерминированного алгоритма решения оказывается невозможным либо сам алгоритм является чрезмерно сложным. В этих случаях часто прибегают к статистическим методам решения задачи, в основе которых лежит теория вероятностей с механизмом случайных чисел.

Способы решения задач, использующие случайные числа, получили общее название метода Монте–Карло. В частности, метод Монте–Карло широко используется для вычисления кратных интегралов.

Пусть функция $y = f(x_1, x_2, ..., x_m)$ непрерывна в ограниченной замкнутой области S и требуется вычислить m-кратный интеграл:

$$I = \int\int ... \int_{(S)} f(x_1, x_2, ..., x_m)dx_1, dx_2, ..., dx_m. \qquad (7.22)$$

Геометрически число I представляет собой $(m + 1)$-мерный объём прямого цилиндроида в пространстве $0x_1\ x_2\ \dots\ x_m y$, построенного на основании S и ограниченного сверху данной поверхностью $y = f(x_1, x_2, ..., x_m)$. Рассмотрим простейший случай, когда область S представляет собой m-мерный параллелепипед $a_i \le x_i \le b_i$, $i = \overline{1, m}$.

Тогда интеграл (7.22) можно приближённо вычислить по следующей формуле:

$$I \approx \frac{1}{n}\left[\sum_{i=1}^{n} f\left(x_{1,i},\, x_{2,i},\, ...,\, x_{m,i}\right)\right] \cdot \prod_{i=1}^{m}\left(b_i - a_i\right), \qquad (7.23)$$

где $x_{ji} = a_j + \left(b_j - a_j\right)R_{i,j}$, $j = \overline{1,m};\ i = \overline{1,n}$, $R_{i,j}$ – случайное число, равномерно распределённое на отрезке $[0, 1]$, n – количество вычислений значений функции $y = f\left(x_1,\, x_2,\, ...,\, x_m\right)$, достаточно большое число.

В частности для вычисления определённого интеграла $I = \int_a^b f(x)dx$ формула (7.23) упростится

$$I \approx \frac{1}{n}(b-a)\sum_{i=1}^{n} f\left(a + (b-a)R_i\right). \qquad (7.24)$$

8. ЧИСЛЕННОЕ РЕШЕНИЕ ОБЫКНОВЕННЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ

Дифференциальными уравнениями называются уравнения, содержащие одну или несколько производных. В зависимости от числа независимых переменных и, следовательно, типа входящих в них производных дифференциальные уравнения делятся на две существенно различные категории: *обыкновенные*, содержащие одну независимую переменную и производные по ней, и уравнения *в частных производных*, содержащие несколько независимых переменных и производные по ним, которые называются *частными*.

Чтобы решить обыкновенное дифференциальное уравнение необходимо знать значения независимой переменной и(или) её производных при некоторых значениях независимой переменной. Если эти дополнительные условия задаются при одном значении независимой переменной, то такая задача называется *задачей Коши*, а дополнительные условия – *начальными условиями*. Если же условия задаются при двух или более значениях независимой переменой, то задача называется *краевой*, а дополнительные условия – *граничными*.

Рассмотрим обыкновенное дифференциальное уравнение первого порядка

$$y' = f(x, y), \tag{8.1}$$

заданное при начальном условии

$$y(x_0) = y_0. \tag{8.2}$$

Задача Коши для этого уравнения заключается в нахождении функции $y(x)$, удовлетворяющей уравнению (8.1) и начальному условию (8.2). Обычно численное решение этой задачи получают, вычисляя сначала значение производной, а затем задавая малое приращение x, переходят к новой точке $x_1 = x_0 + h$. Положение новой точки определяется по наклону кривой, вычисленному с помощью дифференциального уравнения. Таким образом, график численного решения представляет собой последовательность коротких прямолинейных отрезков, которыми аппроксимируется истинная кривая $y = y(x)$. Сам численный метод определяет порядок действий при переходе от одной точки кривой к следующей.

Среди множества методов численного решения задачи Коши можно выделить следующие две большие группы:

1. *Одношаговые методы*, в которых для нахождения следующей точки на кривой $y = y(x)$ требуется информация лишь об одном предыдущем шаге. К этим методам относятся методы Эйлера, Рунге–Кутта.

2. *Методы прогноза и коррекции* (многошаговые), в которых для отыскания следующей точки кривой $y = y(x)$ требуется информация более чем об одной из предыдущих точек. К числу таких методов относятся методы Адамса, Милна, Хемминга.

8.1. МЕТОД ЭЙЛЕРА

Пусть дано дифференциальное уравнение (8.1), где $y' = dy/dx$ при начальном условии (8.2). Разложим $y(x)$ в ряд Тейлора в окрестности точки x_0:

$$y(x_0 + h) = y(x_0) + hy'(x_0) + \frac{1}{2!}h^2 y''(x_0) + \ldots .$$

Если h мало, то члены, содержащие h во второй и более высоких степенях, являются малыми более высоких порядков и ими можно пренебречь. Тогда

$$y(x_0 + h) = y(x_0) + hy'(x_0),$$

где $y'(x_0) = f(x_0, y(x_0))$ находится из дифференциального уравнения (8.1) при подстановке в него начального условия. Таким образом, можно получить приближённое значение зависимой переменной при малом смещении h от начальной точки. Этот процесс можно продолжить, используя соотношение:

$$y_{k+1} = y_k + hf(x_k, y_k), \; k = 0, 1, 2, \ldots \qquad (8.3)$$

и делая сколько угодно много шагов.

Ошибка метода Эйлера (рис. 8.1) имеет порядок h^2, так как члены, содержащие h во второй и более высоких степенях, отбрасываются, а сам метод является методом первого порядка и иногда называется методом Рунге–Кутта первого порядка.

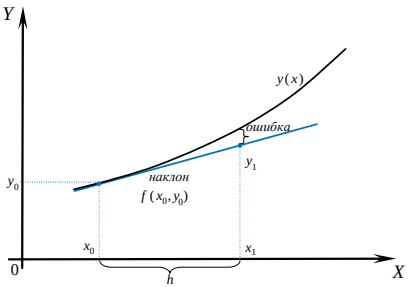


Рис. 8.1. Графическая иллюстрация метода Эйлера

8.2. МОДИФИКАЦИИ МЕТОДА ЭЙЛЕРА

Хотя тангенс угла наклона касательной к истинной кривой в исходной точке известен и равен $y'(x_0)$, он изменяется одновременно с изменениями независимой переменной. Поэтому в точке $x_0 + h$ наклон касательной уже не таков, каким он был в точке x_0. Следовательно, при сохранении начального наклона касательной на всём интервале h в результаты вычислений вносится погрешность. Точность метода Эйлера можно существенно повысить, улучшив аппроксимацию производной. Это улучшение достигается в модифицированных методах Эйлера, являющихся по сути методами Рунге–Кутта 2-го порядка точности.

Первая модификация метода Эйлера для решения задачи (8.1), (8.2) состоит в том, что сначала вычисляют

$$x_{k+1/2} = x_k + \frac{h}{2},$$

$$y_{k+1/2} = y_k + \frac{h}{2} f(x_k, y_k), \tag{8.4}$$

а затем полагают:

$$y_{k+1} = y_k + hf(x_{k+1/2}, y_{k+1/2}). \tag{8.5}$$

По *второму модифицированному методу Эйлера* (*методу Эйлера–Коши*) сначала определяется «грубое» приближение значения функции в следующей точке по методу Эйлера:

$$\tilde{y}_{k+1} = y_k + hf(x_k, y_k), \tag{8.6}$$

а затем находят более точное значение y_{k+1} по формуле

$$y_{k+1} = y_k + \frac{h}{2}[f(x_k, y_k) + f(x_{k+1}, \tilde{y}_{k+1})]. \tag{8.7}$$

Ошибка при использовании этих методов на каждом шаге имеет порядок h^3.

8.3. МЕТОДЫ РУНГЕ–КУТТА

Вторую модификацию метода Эйлера можно получить, если сохранить в ряде Тейлора член с h^2, аппроксимировав при этом вторую производную при этом члене конечной разностью

$$y''(x_0) = \frac{\Delta y'}{\Delta x} = \frac{y'(x_0 + h) - y'(x_0)}{h}.$$

В общем случае, чтобы удержать в ряде Тейлора член n-го порядка, необходимо вычислить n-ю производную зависимой переменной.

Для этого необходимо определить наклоны ещё в $(n-2)$ точках интервала h, т.е. между точками x_i и x_{i+1}. Таким образом, возрастает объём вычислений. Очевидно, что чем выше порядок вычисляемой производной, тем больше дополнительных вычислений потребуется внутри интервала. Семейство методов Рунге–Кутта даёт набор формул для расчёта координат внутренних точек, требуемых для реализации этой идеи. Существует несколько способов расположения внутренних точек и выбора относительных весов для найденных производных.

Запишем некоторые формулы из семейства методов Рунге–Кутта:

1) метод Рунге–Кутта 2-го порядка.

$$y_{k+1} = y_k + \frac{1}{4}(k_1 + k_2);\ k_1 = hf(x_k, y_k);$$
$$k_2 = hf\left(x_k + \frac{2}{3}h,\ y_k + \frac{2}{3}k_1\right); \qquad (8.8)$$

2) метод Рунге–Кутта 3-го порядка.

$$y_{k+1} = y_k + \frac{1}{6}(k_1 + 4k_2 + k_3);\ \ k_1 = hf(x_k, y_k);\ \ k_2 = hf\left(x_k + \frac{h}{2},\ y_k + \frac{k_1}{2}\right);$$
$$k_3 = hf(x_{k+h}, y_k - k_1 + 2k_2); \qquad (8.9)$$

3) метод Рунге–Кутта 3-го порядка.

$$y_{k+1} = y_k + \frac{1}{4}(k_1 + 3k_3);\ k_1 = hf(x_k, y_k);\ k_2 = hf\left(x_k + \frac{h}{3},\ y_k + \frac{k_1}{3}\right);$$
$$k_3 = hf\left(x_k + \frac{2h}{3}, y_k + \frac{2k_2}{3}\right); \qquad (8.10)$$

4) метод Рунге–Кутта 4-го порядка.

$$y_{k+1} = y_k + \frac{1}{8}(k_1 + 3k_2 + 3k_3 + k_4);$$
$$k_1 = hf(x_k, y_k);\ \ k_2 = hf\left(x_k + \frac{h}{2},\ y_k + \frac{k_1}{2}\right);$$
$$k_3 = hf\left(x_k + \frac{2}{3}h, y_k - \frac{k_1}{3} + k_2\right);\ \ k_4 = hf(x_k + h, y_k + k_1 - k_2 + k_3); \qquad (8.11)$$

5) метод Рунге–Кутта 4-го порядка.

$$y_{k+1} = y_k + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4);\ \ k_1 = hf(x_k, y_k);$$

$$k_2 = hf\left(x_k + \frac{h}{2},\ y_k + \frac{k_1}{2}\right);\ k_3 = hf\left(x_k + \frac{h}{2}, y_k + \frac{k_2}{2}\right);$$
$$k_4 = hf\left(x_k + h, y_k + k_3\right); \qquad (8.12)$$

6) метод Рунге–Кутта 4-го порядка.

$$y_{k+1} = y_k + \frac{1}{6}\left(k_1 + 4k_3 + k_4\right);\ \ k_1 = hf\left(x_k, y_k\right);$$
$$k_2 = hf\left(x_k + \frac{h}{4},\ y_k + \frac{k_1}{4}\right);\ \ k_3 = hf\left(x_k + \frac{h}{2}, y_k + \frac{k_2}{2}\right);$$
$$k_4 = hf\left(x_k + h, y_k + k_1 - 2k_2 + 2k_3\right). \qquad (8.13)$$

Среди приведённых наиболее распространённым является метод, при котором удерживаются все члены, включая h^4. Это метод четвёртого порядка точности (8.12), для которого ошибка на шаге имеет порядок h^5.

Всем рассмотренным одношаговым методам присущи определённые общие черты:

1. Чтобы получить информацию в новой точке, надо иметь данные лишь в одной предыдущей точке. Это свойство можно назвать «*самостартованием*».

2. В основе всех одношаговых методов лежит разложение функции в ряд Тейлора, в которых сохраняются члены, содержащие h в степени до n включительно. Целое число n называется *порядком метода*. Погрешность на шаге имеет порядок h^{n+1}.

3. Все одношаговые методы не требуют действительного вычисления производных – вычисляется лишь сама функция, однако могут потребоваться её значения в нескольких промежуточных точках. Это влечёт за собой дополнительные затраты времени и усилий.

4. Свойство «самостартования» позволяет легко менять величину шага h.

8.4. МЕТОДЫ ПРОГНОЗА И КОРРЕКЦИИ

В этих методах для вычисления положения новой точки используется информация о нескольких ранее полученных точках. Для этого применяются две формулы, называемые соответственно *формулами прогноза и коррекции*. Схемы алгоритмов для всех таких методов примерно одинаковы, а сами методы отличаются лишь формулами.

Рассмотрим дифференциальное уравнение

$$y' = f\left(x, y\right).$$

Так как в методах прогноза и коррекции используется информация о нескольких ранее полученных точках, то в отличие от одношаго-

вых методов они не обладают свойством «самостартования». Поэтому прежде необходимо вычислить исходные данные с помощью какого-либо одношагового метода. Затем вычисления проводятся следующим образом. Сначала по формуле прогноза и исходным значениям переменных определяют значение $y_{k+1}^{(0)}$ (верхний индекс (0) означает, что прогнозируемое значение является одним из последовательности значений y_{k+1}, располагающихся в порядке возрастания точности). По прогнозируемому значению $y_{k+1}^{(0)}$ с помощью дифференциального уравнения находят производную

$$y_{k+1}^{(0)'} = f\left(x_{k+1}, y_{k+1}^{(0)}\right),$$

которая затем подставляется в формулу коррекции для вычисления уточнённого значения $y_{k+1}^{(i+1)}$ (i = 0, 1, 2, …). В свою очередь $y_{k+1}^{(i+1)}$ используется для получения более точного значения производной с помощью дифференциального уравнения

$$y_{k+1}^{(i+1)'} = f\left(x_{k+1}, y_{k+1}^{(i+1)}\right).$$

Если это значение производной недостаточно близко к предыдущему, то оно вводится в формулу коррекции и итерационный процесс продолжается. Если же производная изменяется в допустимых пределах, то значение $y_{k+1}^{(i+1)'}$ используется для вычисления окончательного значения y_{k+1}. После этого процесс повторяется – делается следующий шаг, на котором вычисляется y_{k+2}.

Обычно при выводе формул прогноза и коррекции решение уравнений рассматривают как процесс приближенного интегрирования, а сами формулы получают с помощью конечно-разностных методов.

8.4.1. Метод Эйлера–Коши с итерациями

Формула прогноза:

$$y_{k+1}^{(0)} = y_k + hf\left(x_k, y_k\right). \tag{8.14}$$

Формула коррекции:

$$y_{k+1}^{(i+1)} = y_k + \frac{h}{2}\left[f\left(x_k, y_k\right) + f\left(x_{k+1}, y_{k+1}^{(i)}\right)\right], \quad i = 0, 1, 2, \ldots.$$

Метод 2-го порядка точности, погрешность метода $R \sim \left(h^3\right)$.

8.4.2. Метод Милна

В этом методе на этапе прогноза используется формула Милна

$$y_{k+1}^{(0)} = y_{k-3} + \frac{4}{3}h\left[2f(x_k, y_k) - f(x_{k-1}, y_{k-1}) + 2f(x_{k-2}, y_{k-2})\right], \quad (8.15)$$

а на этапе коррекции – формула Симпсона

$$y_{k+1}^{(i+1)} = y_{k-1} + \frac{1}{3}h\left[f(x_{k+1}, y_{k+1}^{(i)}) + 4f(x_k, y_k) + f(x_{k-1}, y_{k-1})\right].$$

Метод является методом 4-го порядка точности, погрешность метода $R \sim (h^5)$.

8.4.3. Метод Хемминга

Это устойчивый метод 4-го порядка точности, в основе которого лежат следующие формулы:
прогноза

$$y_{k+1}^{(0)} = y_{k-3} + \frac{4}{3}h\left[2f(x_k, y_k) - f(x_{k-1}, y_{k-1}) + 2f(x_{k-2}, y_{k-2})\right] \quad (8.16)$$

и коррекции

$$y_{k+1}^{(i+1)} = \frac{1}{8}\left(9y_k - y_{k-2} + 3h\left[f(x_{k+1}, y_{k+1}^{(i)}) + 2f(x_k, y_k) - f(x_{k-1}, y_{k-1})\right]\right).$$

По сравнению с методом Милна этот метод обладает большей устойчивостью, что делает его одним из наиболее распространённых.

8.4.4. Методы Адамса

Представляют собой целое семейство методов, записанных для разных порядков точности:

а) метод Адамса 2-го порядка точности:

$$y_{k+1}^{(0)} = y_k + \frac{h}{2}\left[3f(x_k, y_k) - f(x_{k-1}, y_{k-1})\right]; \quad (8.17)$$

$$y_{k+1}^{(i+1)} = y_k + \frac{h}{2}\left[f(x_{k+1}, y_{k+1}^{(i)}) + f(x_k, y_k)\right] \text{ (метод трапеций);}$$

б) метод Адамса 3-го порядка точности:

$$y_{k+1}^{(0)} = y_k + \frac{h}{12}\left[23f(x_k, y_k) - 16f(x_{k-1}, y_{k-1}) + 5f(x_{k-2}, y_{k-2})\right]; \quad (8.18)$$

$$y_{k+1}^{(i+1)} = y_k + \frac{h}{12}\left[5f(x_{k+1}, y_{k+1}^{(i)}) + 8f(x_k, y_k) - f(x_{k-1}, y_{k-1})\right];$$

в) метод Адамса 4-го порядка точности:

$$y_{k+1}^{(0)} = y_k + \frac{h}{24}\left[55f(x_k, y_k) - 59f(x_{k-1}, y_{k-1}) + 37f(x_{k-2}, y_{k-2}) - 9f(x_{k-3}, y_{k-3})\right];$$

$$y_{k+1}^{(i+1)} = y_k + \frac{h}{24}\left[9f\left(x_{k+1}, y_{k+1}^{(i)}\right) + 19f(x_k, y_k) - 5f(x_{k-1}, y_{k-1}) + f(x_{k-2}, y_{k-2})\right]; \quad (8.19)$$

г) метод Адамса 5-го порядка точности:

$$y_{k+1}^{(0)} = y_k + \frac{h}{720}\left[1901f(x_k, y_k) - 2774f(x_{k-1}, y_{k-1}) + 2616f(x_{k-2}, y_{k-2}) - 1274f(x_{k-3}, y_{k-3}) + 251f(x_{k-4}, y_{k-4})\right]; \quad (8.20)$$

$$y_{k+1}^{(i+1)} = y_k + \frac{h}{720}\left[251f\left(x_{k+1}, y_{k+1}^{(i)}\right) + 646f(x_k, y_k) - 264f(x_{k-1}, y_{k-1}) + 106f(x_{k-2}, y_{k-2}) - 19f(x_{k-3}, y_{k-3})\right].$$

По сравнению с одношаговыми методами рассмотренные методы прогноза и коррекции имеют следующие особенности:

1. Для старта метода прогноза и коррекции при получении исходной информации о нескольких предыдущих точках приходится прибегать к услугам какого-либо одношагового метода. Переходить временно на одношаговый метод приходится и в том случае, если в процессе решения дифференциальных уравнений методом прогноза и коррекции изменяется шаг.

2. Методы прогноза и коррекции обладают более высокой скоростью сходимости, но предъявляют в то же время повышенные требования к вычислительным ресурсам при их численной реализации.

3. Методы прогноза и коррекции являются более эффективными по сравнению с одношаговыми методами, так как при прочих условиях предъявляют менее «жёсткие» требования к величине шага h.

8.5. ВЫБОР ШАГА

Одним из важных практических вопросов, возникающих в процессе решения дифференциального уравнения, является выбор подходящей величины шага.

Если шаг слишком мал, то расчёт потребует неоправданно большого машинного времени, а число ошибок на отдельных шагах, складывающихся в суммарную ошибку, будет весьма велико. Если же шаг выбран слишком большим, то значительной будет и локальная погрешность на каждом шаге и, вследствие этого, накопившаяся суммарная погрешность будет также недопустимо большой.

При выборе величины шага стремятся, чтобы локальная ошибка на шаге, определяемая в общем случае выражением Ch^{n+1} (где C – некоторая постоянная, h – шаг, n – порядок точности метода), была меньше некоторой заданной допустимой величины.

При использовании метода прогноза и коррекции для оценки локальной погрешности (значение постоянной C) существуют явные выражения. Кроме того, погрешность аппроксимации истинной кривой $y(x)$ может быть значительно уменьшена за счёт использования итерационной процедуры расчёта на стадии коррекции.

При использовании одношаговых методов оценить локальную погрешность в явной форме, а, следовательно, определить погрешность аппроксимации (точность метода) в явной форме не удаётся. В этом случае для повышения точности одношагового метода применяют следующий подход, основанный на экстраполяции Ричардсона.

Пусть $y_{k+1}^{(h)}$ – значение искомой функции в точке x_{k+1}, найденное при величине шага $h = h_0$. Уменьшим шаг h вдвое, т.е. $h = \frac{h_0}{2}$ и вычислим значение функции в точке $x_{k+1} = x_k + \frac{h_0}{2} + \frac{h_0}{2}$, сделав для этого два шага $\frac{h_0}{2}$. Полученное значение искомой функции обозначим через $y_{k+1}^{(h/2)}$. Тогда если выполняется неравенство

$$\left| \frac{y_{k+1}^{(h)} - y_{k+1}^{(h/2)}}{2^n + 1} \right| \le \varepsilon , \qquad (8.21)$$

где ε – требуемая точность на шаге, n – порядок одношагового метода, то шаг $h = h_0$ можно считать приемлемым для решения дифференциального уравнения с заданной точностью ε, а $y_{k+1}^{(h)}$ можно считать решением в точке x_{k+1}. В противном случае шаг $h = \frac{h_0}{2}$ необходимо уменьшить в два раза и все вычисления повторить, исходя из узла x_k.

Для получения решения в следующей точке x_{k+2} необходимо проделать аналогичные вычисления, исходя из узла x_{k+1}. При этом начальный шаг рекомендуется выбирать по шагу h, с которым было получено решение в узле x_{k+1}.

Формула (8.21) называется формулой Рунге, а процедура её использования часто включается в вычислительный алгоритм для автоматического изменения шага в процессе вычислений, хотя объём вычислений при этом увеличивается более чем в два раза.

8.6. РЕШЕНИЕ СИСТЕМ ОБЫКНОВЕННЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ

Рассмотрим систему обыкновенных дифференциальных уравнений 1-го порядка:

$$\begin{aligned}
\frac{dy_1}{dx} &= f_1(x, y_1, y_2, ..., y_N);\\
\frac{dy_2}{dx} &= f_2(x, y_1, y_2, ..., y_N);\\
&\dots\\
\frac{dy_N}{dx} &= f_N(x, y_1, y_2, ..., y_N),
\end{aligned} \tag{8.22}$$

заданную при начальных условиях

$$\begin{aligned}
y_1(x_0) &= y_{10};\\
y_2(x_0) &= y_{20};\\
&\dots\\
y_N(x_0) &= y_{N0}.
\end{aligned} \tag{8.23}$$

Задача Коши для данной системы заключается в нахождении таких функций $y_1(x)$, $y_2(x)$, …, $y_N(x)$, которые удовлетворяли бы начальным условиям (8.23) и системе (8.22). Для её решения может быть использован любой из рассмотренных выше методов. При этом вычисление значений зависимых переменных $y_{i(k+1)}$ в точке x_{k+1} осуществляется одновременно по однотипным формулам.

Например, для решения системы

$$\begin{aligned}
y' &= f(x, y, z);\\
z' &= g(x, y, z),
\end{aligned}$$

заданной с начальными условиями $y(x_0) = y_0$, $z(x_0) = z_0$, формулы Эйлера имеют вид:

$$\begin{aligned}
y_{k+1} &= y_k + hf(x_k, y_k, z_k);\\
z_{k+1} &= z_k + hg(x_k, y_k, z_k).
\end{aligned}$$

При решении систем обыкновенных дифференциальных уравнений необходимо помнить, что величина шага h при вычислении всех значений зависимых переменных должна оставаться постоянной на интервале $[x_k, x_{k+1}]$.

8.7. РЕШЕНИЕ СИСТЕМ ОБЫКНОВЕННЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ ВЫСШЕГО ПОРЯДКА

Любое дифференциальное уравнение n-го порядка

$$y^{(n)} = f\left(x, y, y', y'', ..., y^{(n-1)}\right),$$

заданное с начальными условиями

$$y(x_0) = y_0,\ y'(x_0) = y_0',\ ...,\ y^{(n-1)}(x_0) = y_0^{(n-1)},$$

может быть сведено к системе n дифференциальных уравнений 1-го порядка с помощью следующих преобразований:

$$y' = z_1;$$
$$y'' = z_1' = z_2;$$
$$\dots$$
$$y^{(n-1)} = z_{n-2}' = z_{n-1};$$
$$y^{(n)} = z_{n-1}'.$$

Получаемая в результате система:

$$y' = z_1;$$
$$z_1' = z_2;$$
$$\dots$$
$$z_{n-1}' = f\left(x, y, z_1, z_2, ..., z_{n-1}\right)$$

с начальными условиями:

$$y(x_0) = y_0,\ z_1(x_0) = z_{10},\ ...,\ z_{n-1}(x_0) = z_{n-1,0},$$

имеет 1-й порядок и может быть решена любым из рассмотренных выше методов.

9. РЕШЕНИЕ КРАЕВЫХ ЗАДАЧ ДЛЯ ОБЫКНОВЕННЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ ВТОРОГО ПОРЯДКА

Пусть дано дифференциальное уравнение 2-го порядка

$$F(x,\ y,\ y',\ y'') = 0\ . \tag{9.1}$$

Двухточечная краевая задача для уравнения (9.1) ставится следующим образом: необходимо найти функцию $y = y(x)$, которая внутри отрезка $[a,\ b]$ удовлетворяет уравнению (9.1), а на концах отрезка – краевым условиям

$$\begin{cases} \varphi_1[y(a),\ y'(a)] = 0\ ; \\ \varphi_2[y(b),\ y'(b)] = 0. \end{cases} \tag{9.2}$$

Рассмотрим частый случай, когда уравнение (9.1) и граничные условия (9.2) линейны. Такая краевая задача называется *линейной краевой задачей*. В этом случае дифференциальное уравнение и краевые условия записываются так:

$$y'' + p(x)y' + q(x)y = f(x); \tag{9.3}$$

$$\begin{cases} \alpha_0 y(a) + \alpha_1 y'(a) = A\ ; \\ \beta_0 y(b) + \beta_1 y'(b) = B, \end{cases} \tag{9.4}$$

где $p(x),\ q(x),\ f(x)$ – известные непрерывные на отрезке $[a,\ b]$ функции; $\alpha_0,\ \alpha_1,\ \beta_0, \beta_1,\ A,\ B$ – заданные постоянные, причём $|\alpha_0| + |\alpha_1| \neq 0$ и $|\beta_0| + |\beta_1| \neq 0$.

Краевые условия (9.4) в общем случае задают линейную связь между значением искомого решения и его производной на концах отрезка $[a,\ b]$ в отдельности. В частном случае, если $\alpha_0 = 1$ и $\alpha_1 = 0$ или $\beta_0 = 1$ и $\beta_1 = 0$, то на соответствующем конце отрезка задано значение искомого решения. Такое краевое условие называется *условием первого рода*. Если $\alpha_0 = 0$ и $\alpha_1 = 1$ или $\beta_0 = 0$ и $\beta_1 = 1$, то на конце отрезка задано значение производной решения. Это краевое условие называется *условием второго рода*. В общем случае, когда $\alpha_j \neq 0$ и $\beta_j \neq 0$ краевые условия называются *условиями третьего рода*. Если $A = B = 0$, то краевые условия называются *однородными*.

В отличие от имеющей всегда единственное решение задачи Коши, краевая задача (9.3), (9.4) может иметь или одно решение, или бесконечное множество решений, или, наконец, может совсем не иметь решений.

Для того чтобы существовало единственное решение неоднородной краевой задачи (9.3), (9.4), необходимо и достаточно, чтобы однородная краевая задача

$$y'' + p(x)y' + q(x)y = 0; \tag{9.5}$$

$$\begin{cases} \alpha_0 y(a) + \alpha_1 y'(a) = 0; \\ \beta_0 y(b) + \beta_1 y'(b) = 0 \end{cases} \tag{9.6}$$

имела только тривиальное решение $y(x) \equiv 0$.

Одним из наиболее часто используемых методов приближённого решения линейных краевых задач являются разностные методы.

9.1. МЕТОД КОНЕЧНЫХ РАЗНОСТЕЙ

Разобьём отрезок $[a, b]$, на котором ищется решение дифференциального уравнения (9.3) системой равноотстоящих точек с некоторым шагом $h = \dfrac{b-a}{n}$: $x_0 = a$, $x_n = b$, $x_i = x_0 + ih$ ($i = 1, 2, \ldots, n-1$). Обозначим значения функций $p(x)$, $q(x)$, $f(x)$ в узлах x_i через $p_i = p(x_i)$, $q_i = q(x_i)$, $f_i = f(x_i)$, а получаемые в результате расчёта приближённые значения искомой функции $y(x)$ и её производных $y'(x)$, $y''(x)$ в точках x_i через y_i, y_i', y_i'' соответственно.

Заменим приближённо в каждом внутреннем узле производные $y'(x_i)$, $y''(x_i)$ конечно-разностными отношениями:

$$y_i' = \frac{y_{i+1} - y_i}{h}, \quad y_i'' = \frac{y_{i+2} - 2y_{i+1} + y_i}{h^2}, \tag{9.7}$$

а на концах отрезка $[a, b]$ положим

$$y_0' = \frac{y_1 - y_0}{h}, \quad y_n' = \frac{y_n - y_{n-1}}{h}. \tag{9.8}$$

Используя формулы (9.7), (9.8), приближённо заменим уравнение (9.3) и краевые условия (9.4) системой уравнений

$$\begin{cases} \dfrac{y_{i+2} - 2y_{i+1} + y_i}{h^2} + p_i \dfrac{y_{i+1} - y_i}{h} + q_i y_i = f_i, \\ i = 0, 1, 2, \ldots, n-2; \\ \alpha_0 y_0 + \alpha_1 \dfrac{y_1 - y_0}{h} = A, \ \beta_0 y_n + \beta_1 \dfrac{y_n - y_{n-1}}{h} = B. \end{cases} \tag{9.9}$$

Система (9.9) представляет собой систему $(n + 1)$ линейных алгебраических уравнений с $(n + 1)$ неизвестными $\left(y_i,\ i=\overline{0,\ n}\right)$. Для её решения может быть использован любой из методов решения подобных систем. Однако при большом n непосредственное решение системы (9.9) становится громоздким. В этом случае достаточно эффективным является *метод прогонки* решения систем линейных алгебраических уравнений трёхдиагонального вида. Суть этого метода заключается в следующем. Первые $(n - 1)$ уравнений системы (9.9) приводятся к виду

$$y_{i+1} = c_i\left(d_i - y_{i+2}\right),\ i = 0,\ 1,\ 2,\ ...,\ n-2, \tag{9.10}$$

где числа c_i и d_i последовательно вычисляются по формулам:

$$\text{при } i = 0:\ c_0 = \frac{\alpha_1 - \alpha_0 h}{m_0\left(\alpha_1 - \alpha_0 h\right) + k_0\alpha_1},\ d_0 = \frac{k_0 Ah}{\alpha_1 - \alpha_0 h} + f_0 h^2; \tag{9.11}$$

$$\text{при } i=\overline{1,\ n-2}:\ c_i = \frac{1}{m_i - k_i c_{i-1}},\ d_i = f_i h^2 - k_i c_{i-1} d_{i-1}\ . \tag{9.12}$$

$$\text{Здесь } m_i = hp_i - 2,\ k_i = 1 - hp_i + h^2 q_i,\ i=\overline{0,\ n-2}. \tag{9.13}$$

Вычисления производятся в следующем порядке:

Прямой ход. Сначала по формулам (9.13) вычисляются значения m_i, k_i, а затем по формулам (9.11) c_0, d_0 и, применяя последовательно рекуррентные формулы (9.12), получают значения c_i, d_i при $i=\overline{1,\ n-2}$.

Обратный ход. Из уравнения (9.10) при $i = n - 2$ и последнего уравнения системы (9.9) можно получить выражение для вычисления y_n:

$$y_n = \frac{\beta_1 c_{n-2} d_{n-2} + Bh}{\beta_1\left(1 + c_{n-2}\right) + \beta_0 h}\ . \tag{9.14}$$

Используя уже известные числа c_{n-2}, d_{n-2}, находим y_n. Затем вычисляются значения y_i $(i = n - 1,\ \ldots,\ 1)$, последовательно применяя рекуррентные формулы (9.10). Значение y_0 находится из предпоследнего уравнения системы (9.9):

$$y_0 = \frac{\alpha_1 y_1 - Ah}{\alpha_1 - \alpha_0 h}\ . \tag{9.15}$$

Таким образом, все вычисления как бы «прогоняются» два раза. Вычисления прямого хода определяют вспомогательные числа c_i, d_i в порядке возрастания индекса i. При этом для вычисления значений c_0, d_0 используется краевое условие на левом конце отрезка интегрирования. Затем на первом шаге обратного хода происходит согласование полученных чисел c_{n-2}, d_{n-2} с краевым условием на правом конце отрезка интегрирования, после чего последовательно получаются значения искомой функции y_i в порядке убывания индекса i.

Погрешность метода конечных разностей при использовании конечно-разностных отношений (9.7) в каждом узле x_i не превышает

$$\frac{h^2 M_4}{96}(b-a)^2, \text{ где } M_4 = \max_{[a,\,b]}\left|y^{(\mathrm{IV})}(x)\right|.$$

Более точные, однако, не превышающие указанную выше величину погрешности, формулы можно получить, если использовать вместо конечно-разностных отношений (9.7) центральные конечные разности:

$$y_i' = \frac{y_{i+1} - y_{i-1}}{2h}; \quad y_i'' = \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2}. \tag{9.16}$$

Тогда краевая задача (9.3), (9.4) запишется в виде системы линейных уравнений:

$$\begin{cases} \dfrac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + p_i \dfrac{y_{i+1} - y_{i-1}}{2h} + q_i y_i = f_i,\ i = \overline{1,\, n-1}; \\ \alpha_0 y_0 + \alpha_1 \dfrac{y_1 - y_0}{h} = A,\ \beta_0 y_n + \beta_1 \dfrac{y_n - y_{n-1}}{h} = B. \end{cases} \tag{9.17}$$

При использовании метода прогонки для решения этой системы её первое уравнение необходимо привести к виду:

$$y_i = c_i(d_i - y_{i+1}),\ i = 1, 2, ..., n-1, \tag{9.18}$$

где коэффициенты c_i, d_i вычисляются по формулам:

$$c_1 = \frac{\alpha_1 - \alpha_0 h}{m_1(\alpha_1 - \alpha_0 h) + k_1 \alpha_1}; \ d_1 = \varphi_1 + k_1 \frac{Ah}{\alpha_1 - \alpha_0 h}; \tag{9.19}$$

$$c_i = \frac{1}{m_i - k_i c_{i-1}}; \ d_i = \varphi_i - k_i c_{i-1} d_{i-1},\ i = 2, 3, ..., n-1; \tag{9.20}$$

$$m_i = \frac{2q_i h^2 - 4}{2 + hp_i}; \ k_i = \frac{2 - hp_i}{2 + hp_i}; \ \varphi_i = \frac{2h^2 f_i}{2 + hp_i},\ i = 1, ..., n-1. \tag{9.21}$$

Последующее вычисление значений y_i ($i = n, \ldots, 1, 0$) осуществляется аналогично описанному выше алгоритму.

Точность разностного метода можно значительно повысить, если при замене производных использовать многоточечные разностные схемы.

Конечно-разностные отношения с успехом могут быть применены и для решения нелинейных дифференциальных уравнений.

Рассмотрим такое уравнение

$$y'' = f(x, y, y') \tag{9.22}$$

при линейных краевых условиях

$$\begin{cases} \alpha_0 y(a) - \alpha_1 y'(a) = A; \\ \beta_0 y(b) + \beta_1 y'(b) = B. \end{cases} \tag{9.23}$$

Разобьём отрезок $[a, b]$ системой равноотстоящих узлов $x_0 = a$, $x_i = x_0 + ih$, $\left(i = \overline{1, n-1}\right)$, $x_n = b$ с некоторым шагом $h = \dfrac{b-a}{n}$ и заменим приближённо уравнение (9.22) и краевые условия (9.23) системой

$$\begin{cases} \dfrac{y_{i+1} - 2y_i + y_{i-1}}{h^2} = f\left(x_i, y_i, \dfrac{y_{i+1} - y_{i-1}}{2h}\right), \; i = 1, 2, ..., n-1; \\ \alpha_0 y_0 - \alpha_1 \dfrac{y_1 - y_0}{h} = A; \; \beta_0 y_n + \beta_1 \dfrac{y_n - y_{n-1}}{h} = B. \end{cases} \tag{9.24}$$

Система (9.24) представляет собой систему из (n + 1) нелинейных уравнений с (n + 1)-й неизвестными y_i (i = 0, 1, …, n). Она может быть решена методом простой итерации. Специальный (трёхдиагональный) вид этой системы позволит записать её решения в явном виде:

$$\begin{gathered} y_i^{(k+1)} = \frac{h}{\Delta}\left(A\beta_0(b-a) + A\beta_1 + \alpha_1 B\right) + \frac{i}{\Delta}\left(\alpha_0 B - A\beta_0\right) + \\ + h^2 \sum_{j=1}^{n-1} g_{ji} f\left(x_j, y_j^{(k)}, \frac{y_{j+1}^{(k)} - y_{j-1}^{(k)}}{2h}\right) \\ i = \overline{0, n}; \; k = 0, 1, 2, ..., \end{gathered} \tag{9.25}$$

где числа a, b, A, B, α_0, α_1, β_0, β_1 известны, а Δ и g_{ji} вычисляются по формулам:

$$\Delta = \frac{1}{h}\left(\alpha_0\beta_0(b-a) + \alpha_0\beta_1 + \alpha_1\beta_0\right); \tag{9.26}$$

$$g_{ji} = \begin{cases} \dfrac{1}{\Delta}\left(j\alpha_0 + \dfrac{\alpha_1}{h}\right)\left(i\beta_0 - \beta_0 n - \dfrac{\beta_1}{h}\right), \; j \le i; \\ \dfrac{1}{\Delta}\left(i\alpha_0 + \dfrac{\alpha_1}{h}\right)\left(j\beta_0 - \beta_0 n - \dfrac{\beta_1}{h}\right), \; j > i. \end{cases} \tag{9.27}$$

Таким образом, решение краевой задачи (9.22), (9.23) сводится к достаточно простой итерационной схеме (9.25) – (9.27).

10. РЕШЕНИЕ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ В ЧАСТНЫХ ПРОИЗВОДНЫХ

Дифференциальные уравнения в частных производных классифицируются либо в зависимости от из математической природы – эллиптические, параболические, гиперболические, либо в зависимости от физического смысла решаемых задач – уравнение диффузии, волновое уравнение и т.п.

С математической точки зрения дифференциальные уравнения 2-го порядка в частных производных с двумя независимыми переменными

$$A(x,\,y)\frac{\partial^2 f(x,\,y)}{\partial x^2}+B(x,\,y)\frac{\partial^2 f(x,\,y)}{\partial x\partial y}+C(x,\,y)\frac{\partial^2 f(x,\,y)}{\partial y^2}+ \\ +E\left(x,\,y,\,f(x,\,y),\frac{\partial f}{\partial x},\frac{\partial f}{\partial y}\right)=0 \qquad (10.1)$$

классифицируются в зависимости от характера функций *A*, *B* и *C*, зависящих от переменных *x* и *y*. Если $B^2-4AC<0$, уравнение называется эллиптическим, если $B^2-4AC=0$ – параболическим, а если $B^2-4AC>0$ – гиперболическим. Зависимость функций *A*, *B* и *C* от *x* и *y* усложняет ситуацию, так как делает возможным изменение типа уравнения при переходе из одной части рассматриваемой области в другую.

Дополнительными условиями для дифференциальных уравнений 2-го порядка в частных производных могут служить граничные или начальные условия, а также их комбинации. Эллиптические уравнения описывают установившиеся (стационарные) процессы; задача ставится в замкнутой области, и в каждой точке границы этой области задаются граничные условия. Параболическими и гиперболическими уравнениями описываются эволюционные процессы (процессы «распространения»). В таких задачах на одной части границы ставятся граничные условия, на другой – начальные; возможны также открытые области, в которые «распространяется решение».

В инженерной практике чаще всего приходится иметь дело со следующими уравнениями в частных производных:

1. Уравнение Лапласа (установившееся течение жидкости, стационарные тепловые поля)

$$\Delta f=0\,.$$

2. Уравнение Пуассона (теплопередача с внутренними источниками тепла)

$$\Delta f=\varphi(x,\,y)\,.$$

3. Уравнение диффузии (нестационарная теплопроводность)

$$\frac{\partial f}{\partial t} = a^2 \Delta f \ .$$

4. Волновое уравнение (распространение звуковых волн)

$$\frac{\partial^2 f}{\partial t^2} = a^2 \Delta f \ .$$

5. Бигармоническое уравнение (деформация пластин)

$$\Delta^2 f = F(x, y) .$$

В этих уравнениях Δ – оператор Лапласа, который в случае двух независимых переменных x и y записывается так:

$$\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} ,$$

а в случае одной независимой переменной x:

$$\Delta f = \frac{\partial^2 f}{\partial x^2} .$$

Оператор Δ^2 называется *бигармоническим оператором* и в случае двух независимых переменных имеет вид

$$\Delta^2 f = \frac{\partial^4 f}{\partial x^4} + 2\frac{\partial^4 f}{\partial x^2 \partial y^2} + \frac{\partial^4 f}{\partial y^4} .$$

Для решения дифференциальных уравнений в частных производных обычно используется метод конечных разностей. В его основе лежит конечно-разностная аппроксимация производных, которая осуществляется в три этапа (рис. 10.1).

На первом этапе решения дифференциального уравнения с частными производными выбор сетки осуществляется в соответствии с характером задачи и граничных условий. Обычно используются сетки вида, показанного на рис. 10.2.

Однако на практике нередко приходится иметь дело с областями неправильной формы (рис. 10.3).

Границы такой области нельзя точно задать с помощью какой-либо из приведённых выше сеток. Однако существуют специальные методы, которые позволяют так модифицировать стандартные сетки, что с их помощью становится возможным описание границ сложной конфигурации.

На практике наибольшее распространение получили сетки прямоугольного вида, которые получаются при построении на плоскости двух семейств параллельных прямых: $x = x_0 + ih_x \ \ (i = 0 \pm 1, \pm 2, ...)$ и $y = y_0 + jh_y \ \ (j = 0 \pm 1, \pm 2, ...)$.

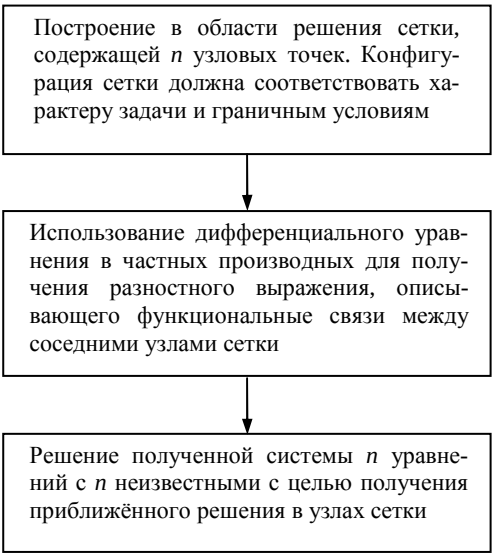


Рис. 10.1. Этапы решения дифференциальных уравнений в частных производных

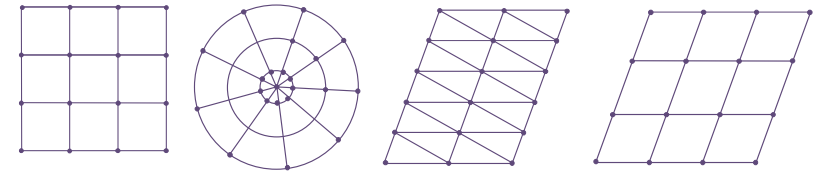


Рис. 10.2. Виды сеток для решения дифференциальных уравнений в частных производных

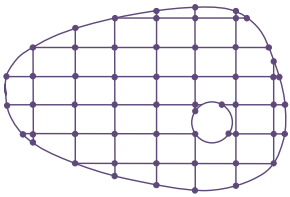


Рис. 10.3. Область решения дифференциальных уравнений в частных производных в общем случае

Точки пересечения этих сеток называются *узлами*. Два узла называются *соседними*, если они удалены друг от друга в направлении оси $0x$ или $0y$ на расстояние, равное шагу сетки h_x или h_y соответственно. Все узлы, принадлежащие заданной области G, можно разделить на *внутренние, и внешние и граничные*.

Значение искомой функции $f = f(x, y)$ в узлах сетки будем обозначать $f_{ij} = f(x_0 + ih_x, y_0 + jh_y)$. Используя разложения в ряд Тейлора, в каждом внутреннем узле $(x_0 + ih_x, y_0 + jh_y)$ частные производные могут быть замены конечно-разностными отношениями:

$$\left(\frac{\partial f}{\partial x}\right)_{ij} \approx \frac{f_{i+1,\,j} - f_{i-1,\,j}}{2h_x}, \quad \left(\frac{\partial f}{\partial y}\right)_{ij} \approx \frac{f_{i,\,j+1} - f_{i,\,j-1}}{2h_y}. \qquad (10.2)$$

В граничном узле приходится пользоваться менее точными формулами вида:

$$\left(\frac{\partial f}{\partial x}\right)_{ij} \approx \frac{\tilde{f}_{i+1,\,j} - f_{ij}}{\Delta_x}; \left(\frac{\partial f}{\partial x}\right)_{ij} \approx \frac{\tilde{f}_{i,\,j+1} - f_{ij}}{\Delta_y}, \qquad (10.3)$$

где $\tilde{f}$ – значение функции на границе области G, Δ_x, Δ_y – расстояния от узла i, j до границы.

Аналогично могут быть заменены частные производные 2-го порядка:

$$\left(\frac{\partial^2 f}{\partial x^2}\right)_{ij} \approx \frac{f_{i+1,\,j} - 2f_{ij} + f_{i-1,\,j}}{h_x^2}; \left(\frac{\partial^2 f}{\partial y^2}\right)_{ij} \approx \frac{f_{i,\,j+1} - 2f_{ij} + f_{i,\,j-1}}{h_y^2} \qquad (10.4)$$

$$\left(\frac{\partial^2 f}{dxdy}\right)_{ij} \approx \frac{f_{i+1,\,j+1} - f_{i-1,\,j+1} - f_{i+1,\,j-1} + f_{i-1,\,j-1}}{4h_x h_y}. \qquad (10.5)$$

Вычислительные шаблоны (10.2) – (10.5) имеют погрешность порядка h^2. Для построения более точных вычислительных шаблонов необходимо в формулах вводить в рассмотрение новые узлы.

Вторые частные производные для узлов, лежащих на границе области, можно записать в виде (рис. 10.4):

$$\left(\frac{\partial^2 f}{\partial x^2}\right)_{ij} \approx \left(\frac{\tilde{f}_{i+1,\,j} - f_{ij}}{\Delta_x} - \frac{f_{ij} + f_{i-1,\,j}}{h_x}\right) \Big/ (0{,}5(\Delta_x + h_x)); \qquad (10.6)$$

$$\left(\frac{\partial^2 f}{\partial y^2}\right)_{ij} \approx \left(\frac{\tilde{f}_{i,\,j+1} - f_{ij}}{\Delta_y} - \frac{f_{ij} + f_{i,\,j-1}}{h_y}\right) \Big/ (0{,}5(\Delta_y + h_y)).$$

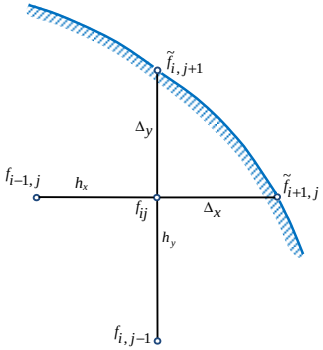


Рис. 10.4. Графическая иллюстрация получения формул (10.6).

Применив вычислительные шаблоны (10.2) – (10.6) к каждому из N узлов сетки, получается система из N уравнений, которая может быть линейной, если исходное дифференциальное уравнение имеет соответствующую структуру. Для решения системы уравнений может быть использован любой из рассмотренных ранее методов решения систем линейных или нелинейных уравнений.

Рассмотрим примеры решения наиболее часто встречающихся уравнений эллиптического, параболического и гиперболического типов.

10.1. ПЕРВАЯ КРАЕВАЯ ЗАДАЧА ДЛЯ УРАВНЕНИЯ ПУАССОНА

Данная задача для уравнения Пуассона

$$\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = \varphi(x, y) \qquad (10.7)$$

ставится следующим образом: найти функцию $f = f(x, y)$, удовлетворяющую внутри некоторой области G уравнению (10.7), а на границе этой области $G_\text{г}$ – условию:

$$f\big|G_\text{г} = \psi(x, y), \qquad (10.8)$$

где $\psi(x, y)$ – заданная непрерывная функция.

Выберем шаги h_x и h_y по x и y соответственно и построим сетку

$$x_i = x_0 + ih_x \quad (i = 0, \pm 1, \pm 2, ...),$$

$$y_j = y_0 + jh_y \quad (i = 0, \pm 1, \pm 2, ...),$$

заменив при этом в каждом внутреннем узле (x_i, y_j) производные $\frac{\partial^2 f}{\partial x^2}$ и $\frac{\partial^2 f}{\partial y^2}$ конечно-разностными отношениями (10.4), а уравнение (10.7) – системой конечно-разностных уравнений

$$\frac{f_{i+1,\,j} - 2f_{ij} + f_{i-1,\,j}}{h_x^2} + \frac{f_{i,\,j+1} - 2f_{ij} + f_{i,\,j-1}}{h_y^2} = \varphi_{ij}, \qquad (10.9)$$

где $\varphi_{ij} = \varphi(x_i, y_j)$.

Уравнения (10.9) вместе со значениями f_{ij} в граничных узлах, вычисленных по уравнению (10.8), образуют систему линейных алгебраических уравнений относительно значений функции $f(x, y)$ в узлах (x_i, y_j).

Наиболее простой вид система (10.9) имеет для прямоугольной области и для $h_x = h_y = h$. В этом случае уравнения (10.9) записываются следующим образом:

$$f_{i+1,\,j} + f_{i-1,\,j} + f_{i,\,j+1} + f_{i,\,j-1} - 4f_{ij} = h^2\varphi_{ij}. \qquad (10.10)$$

При $\varphi(x, y) \equiv 0$ уравнение (10.7) называется уравнением Лапласа и при аналогичных выкладках для него система конечно-разностных уравнений запишется в виде

$$f_{ij} = \frac{1}{4}\left(f_{i+1,\,j} + f_{i-1,\,j} + f_{i,\,j+1} + f_{i,\,j-1}\right). \qquad (10.11)$$

Для решения систем (10.10) и (10.11) может быть использован любой из рассмотренных ранее итерационных методов решения систем линейных алгебраических уравнений (простой итерации, Зейделя и др.).

10.2. РЕШЕНИЕ УРАВНЕНИЙ ПАРАБОЛИЧЕСКОГО ТИПА

Рассмотрим смешанную задачу для уравнения теплопроводности, а именно: найти функцию $f(x, t)$, удовлетворяющую уравнению:

$$\frac{\partial f}{\partial t} = a^2 \frac{\partial^2 f}{\partial x^2}, \qquad (10.12)$$

начальному условию

$$f(x, 0) = u(x),\ 0 < x < l \qquad (10.13)$$

и краевым условиям

$$f(0, t) = \varphi(t), \quad f(l, t) = \psi(t). \tag{10.14}$$

Подобная постановка задачи описывает, в частности, процесс распространения тепла в однородном стержне длиной l.

Примем $a = 1$ (в противном случае в задаче (10.12) – (10.14) необходимо сделать замену переменной $\tau = a^2 t$).

Построим в полуполосе $t \geq 0$, $0 \leq x \leq l$ (рис. 10.5) сетку: $x = ih \ (i = 0, 1, 2, ..., n)$, $t = j\Delta \ (j = 0, 1, 2, ...)$. Обозначим $x_i = ih$, $t_j = j\Delta$, $f(x_i, t_j) = f_{ij}$ и приближённо заменим в каждом внутреннем узле (x_i, t_j) производную $\dfrac{\partial^2 f}{\partial x^2}$ разностным отношением

$$\left(\frac{\partial^2 f}{\partial x^2}\right)_{ij} \approx \frac{f_{i+1,\,j} - 2 f_{ij} + f_{i-1,\,j}}{h^2}, \tag{10.15}$$

а производную $\dfrac{\partial f}{\partial t}$ одним из двух разностных отношений:

$$\left(\frac{\partial f}{\partial t}\right)_{ij} \approx \frac{f_{i,\,j+1} - f_{ij}}{\Delta}, \tag{10.16}$$

$$\left(\frac{\partial f}{\partial t}\right)_{ij} \approx \frac{f_{ij} - f_{i,\,j-1}}{\Delta}. \tag{10.17}$$

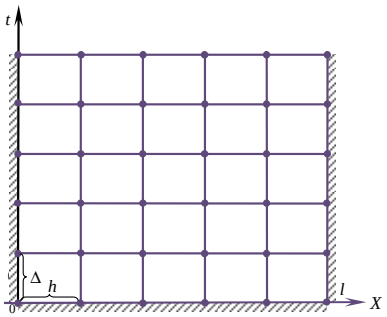


Рис. 10.5. Построение сетки для задачи параболического типа

Тогда уравнение (10.12) при $a = 1$ может быть записано в виде следующих конечно-разностных уравнений:

$$\frac{f_{i,\,j+1} - f_{ij}}{\Delta} = \frac{f_{i+1,\,j} - 2f_{ij} + f_{i-1,\,j}}{h^2}; \qquad (10.18)$$

$$\frac{f_{ij} - f_{i,\,j-1}}{\Delta} = \frac{f_{i+1,\,j} - 2f_{ij} + f_{i-1,\,j}}{h^2}. \qquad (10.19)$$

Обозначив $\sigma = \Delta / h^2$, приведём эти уравнения к виду

$$f_{i,\,j+1} = (1 - 2\sigma) f_{ij} + \sigma\left(f_{i+1,\,j} + f_{i-1,\,j}\right); \qquad (10.20)$$

$$(1 + 2\sigma) f_{ij} - \sigma\left(f_{i+1,\,j} + f_{i-1,\,j}\right) - f_{i,\,j-1} = 0. \qquad (10.21)$$

При записи уравнений (10.20) использовалась явная схема расположения узлов (рис. 10.6, *а*), а при записи уравнений (10.21) – неявная схема (рис. 10.6, *б*).

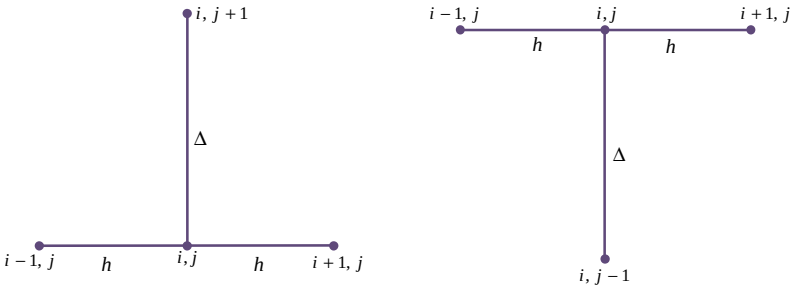
При выборе шагов сетки для решения задачи (10.12) – (10.14) необходимо учитывать, что уравнения (10.20) будут устойчивыми при $0 < \sigma \le 1/2$, а уравнения (10.21) – при любом σ.

Наиболее удобный вид система (10.20) имеет при $\sigma = 1/2$:

$$f_{i,\,j+1} = \frac{f_{i-1,\,j} + f_{i+1,\,j}}{2} \qquad (10.22)$$

и при $\sigma = 1/6$

$$f_{i,\,j+1} = \frac{1}{6}\left(f_{i-1,\,j} + 4f_{ij} + f_{i+1,\,j}\right). \qquad (10.23)$$



а) *б*)

Рис. 10.6. Варианты сеток для решения уравнения параболического типа

Оценки погрешностей приближённых решений, полученных из уравнений (10.21), (10.22), (10.23) в полосе $0 \le x \le l,\ 0 \le t \le T$, соответственно имеет вид:

$$\left| f - \tilde{\tilde{f}} \right| \le T\left(\frac{\Delta}{2} + \frac{h^2}{12} \right) M_1 ; \qquad (10.24)$$

$$\left| f - \tilde{\tilde{f}} \right| \le \frac{T}{3} M_1 h^2 ; \qquad (10.25)$$

$$\left| f - \tilde{\tilde{f}} \right| \le \frac{T}{135} M_2 h^4 , \qquad (10.26)$$

где $\tilde{\tilde{f}}$ – точное решение задачи (10.12) – (10.14),

$$M_1 = \max\left\{ \left| u^{(4)}(x) \right|, \left| \varphi''(t) \right|, \left| \psi''(t) \right| \right\};$$

$$M_2 = \max\left\{ \left| u^{(6)}(x) \right|, \left| \varphi^{(4)}(t) \right|, \left| \psi^{(4)}(t) \right| \right\}.$$

Приведённые оценки (уравнения (10.24) – (10.25)) позволяют сделать следующие выводы. Уравнение (10.23) даёт более высокую точность решения по сравнению с уравнением (10.22). Но уравнение (10.22) имеет более простой вид, а кроме того, шаг Δ по аргументу t для уравнения (10.23) должен быть значительно меньше, что приводит к большему объёму вычислений. Уравнение (10.21) даёт меньшую точность, но при этом шаги Δ и h выбираются независимо друг от друга. Уравнения (10.22) и (10.23) позволяют вычислить значения функции $f(x, t)$ на каждом слое по явным формулам через значения на предыдущем слое; уравнение (10.21) (неявная схема) этим свойством не обладает.

В случае решения смешанной краевой задачи для неоднородного параболического уравнения $\dfrac{\partial f}{\partial t} = \dfrac{\partial^2 f}{\partial x^2} + F(x, t)$ при записи разностных уравнений, использующих явные схемы узлов, в правых частях уравнений (10.20), (10.22), (10.23) добавится произведение $\Delta \cdot F_{ij}$, где $F_{ij} = F(x_i, t_j)$.

Для решения систем линейных уравнений (10.20), (10.22), (10.23) могут быть применены итерационные методы, а разностного уравнения (10.21) – метод прогонки решения систем линейных уравнений специального вида [19].

10.3. МЕТОД СЕТОК ДЛЯ УРАВНЕНИЯ ГИПЕРБОЛИЧЕСКОГО ТИПА

Рассмотрим смешанную задачу для уравнения колебания струны, заключающуюся в отыскании функции, удовлетворяющей уравнению

$$\frac{\partial^2 f}{\partial t^2} = a^2 \frac{\partial^2 f}{\partial x^2}, \tag{10.27}$$

начальным условиям

$$f(x, 0) = u(x), \quad f_t(x, 0) = g(x), \quad 0 \le x \le l \tag{10.28}$$

и краевым условиям

$$f(0, t) = \varphi(t), \quad f(l, t) = \psi(t). \tag{10.29}$$

Пусть a = 1. В противном случае необходимо сделать замену переменной $\tau = at$.

Введём в полуполосе $t \ge 0, \ 0 \le x \le l$ прямоугольную сетку (рис. 10.7) $x_i = ih \ (i = 0, 1, 2, ..., n)$, $t_j = j\Delta \ (j = 0, 1, 2, ...)$ и заменим в уравнении (10.27) при a = 1 производные разностными отношениями вида (10.15). В результате будем иметь:

$$\frac{f_{i,\,j+1} - 2f_{ij} + f_{i,\,j-1}}{\Delta^2} = \frac{f_{i+1,\,j} - 2f_{ij} + f_{i-1,\,j}}{h^2}. \tag{10.30}$$

Обозначив $\alpha = \Delta/h$, получим разностное уравнение

$$f_{i,\,j+1} = 2f_{ij} + f_{i,\,j-1} + \alpha^2\left(f_{i+1,\,j} - 2f_{ij} + f_{i-1,\,j}\right). \tag{10.31}$$

При $\alpha \le 1$ система (10.31) будет устойчивой. В частности, при $\alpha = 1$ уравнение (10.31) имеет наиболее простой вид:

$$f_{i,\,j+1} = f_{i+1,\,j} + f_{i-1,\,j} - f_{i,\,j-1}. \tag{10.32}$$

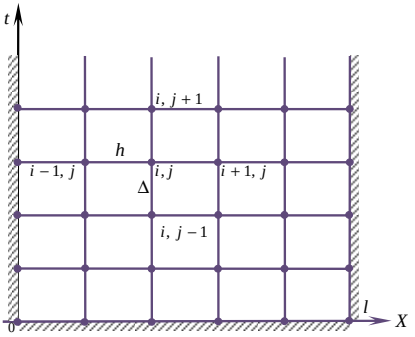


Рис. 10.7. Прямоугольная сетка для решения уравнения гиперболического типа

Используемая при получении уравнения (10.31) схема узлов является явной, так как уравнение (10.31) позволяет найти значения функции $f(x, t)$ на слое t_{j+1}, если известны значения на двух предыдущих слоях. Для того чтобы найти приближённое решение задачи (10.27) – (10.29) необходимо знать значение решения на двух начальных слоях. Их можно найти из начальных условий одним из следующих способов:

1. Заменяем в начальном условии (10.28) производную $f_t(x, 0)$ разностным отношением

$$\frac{f_{i1} - f_{i0}}{\Delta} = g(x_i) = g_i .$$

Тогда для определения значений $f(x, t)$ на слоях $j = 0,\ j = 1$ получаем:

$$f_{i0} = u_i ,\quad f_{i1} = u_i + \Delta g_i . \tag{10.33}$$

2. Заменяем производную $f_t(x, 0)$ разностным отношением $\frac{f_{i1} - f_{i,-1}}{2\Delta}$, где $f_{i,-1}$ – значение функции $f(x, t)$ на слое $j = -1$. Тогда из начальных условий (10.28) будем иметь:

$$f_{i0} = u_i ,\ \frac{f_{i1} - f_{i,-1}}{2\Delta} = g_i . \tag{10.34}$$

Напишем разностное уравнение (10.32) для слоя $j = 0$.

$$f_{i,1} = f_{i+1,0} + f_{i-1,0} - f_{i,-1} . \tag{10.35}$$

Исключив из уравнений (10.34), (10.35) $f_{i,-1}$, получим:

$$f_{i0} = u_i ,\ f_{i1} = \frac{1}{2}(u_{i+1} + u_{i-1}) + \Delta g_i . \tag{10.36}$$

3. Если функция $u(x)$ имеет конечную вторую производную, то значение f_{i1} можно определить с помощью формулы Тейлора.

$$f_{i1} \approx f_{i0} + \Delta \frac{\partial f_{i0}}{\partial t} + \frac{\Delta^2}{2} \frac{\partial^2 f_{i0}}{\partial t^2} .$$

Используя уравнения (10.27) при a = 1 и начальные условия (10.28), можно записать: $f_{i0} = u_i ,\ \frac{\partial f_{i0}}{\partial t} = g_i ,\ \frac{\partial^2 f_{i0}}{\partial t^2} = \frac{\partial^2 f_{i0}}{\partial x^2} = u_i''$.

В результате получаем:

$$f_{t0} = u_i ,\ f_{i1} \approx u_i + \Delta g_i + \frac{\Delta^2}{2} u_i'' . \tag{10.37}$$

Решение системы (10.37) осуществляется итерационными методами.

11. ОСНОВЫ ТЕОРИИ ОПТИМИЗАЦИИ

Оптимизация – это целенаправленная деятельность, заключающаяся в получении наилучших результатов при соответствующих условиях. Постановка задачи оптимизации предполагает наличие объекта оптимизации (независимо от того человеческая это деятельность в течение определённого периода времени или производственный процесс).

Решение любой задачи оптимизации начинают с выявления цели оптимизации, т.е. формулировки требований, предъявляемых к объекту оптимизации. От того, насколько правильно выражены эти требования, может зависеть возможность решения задачи.

Математическая формулировка задачи оптимизации выглядит следующим образом: заданы множество X и функция $f(x)$, определённая на X; требуется найти точки минимума или максимума функции f на X, т.е.

$$f(x) \to \min, \quad x \in X \quad \text{или} \tag{11.1}$$

$$f(x) \to \max, \quad x \in X\ . \tag{11.2}$$

Решение задач (11.1) и (11.2), т.е. точки минимума и максимума функции f на X, называются точками экстремума, а сами задачи (11.1) и (11.2) – экстремальными задачами. Как видно, от задачи (11.2) можно легко перейти к задаче (11.1), заменив знак перед функцией $f(x)$ на противоположный, т.е.

$$f(x) \to \max, \quad x \in X \Leftrightarrow -f(x) \to \min, \ x \in X\ .$$

Это позволяет без труда переносить результаты, полученные для задачи минимизации, на задачи максимизации и наоборот.

Функция $f(x)$ в задачах (11.1) и (11.2) называется *целевой функцией*, множество X – *допустимым множеством* или *пространством проектирования*, а любой элемент $x \in X$ – *допустимой точкой задачи или проектными параметрами*.

Среди множества решений задачи (11.1) или (11.2) различают глобальные и локальные экстремумы. Точка $x^* \in X$ называется:

1) точкой глобального минимума функции f на множестве X, или глобальным решением задачи (11.1), если

$$f(x^*) \le f(x)\ \forall x \in X\ ; \tag{11.3}$$

2) точкой локального минимума f на X, или локальным решением задачи (11.1), если существует число $r > 0$ такое, что

$$f(x^*) \le f(x)\ \forall x \in X \cap U_r(x^*), \tag{11.4}$$

где $U_r(x^*) = \{x \,\|\, \|x - x^*\| \le r\}$ – шар радиусом $r > 0$ с центром x^*.

Если неравенство (11.3) или (11.4) выполняется как строгое при $x \neq x^*$, то говорят, что x^* – точка строгого минимума (строгое решение) в глобальном или локальном смысле. При этом глобальное решение всегда является и локальным, а вот обратное неверно.

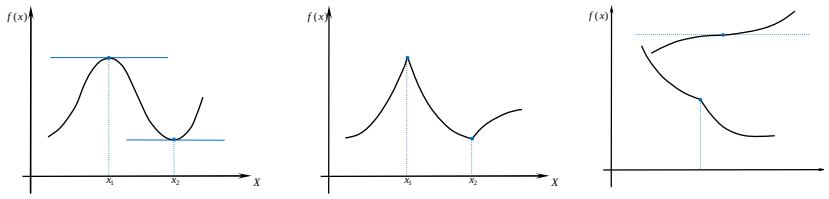
В том случае, если точка $x^* \in X$ является точкой глобального минимума функции f на X, то это может быть записано следующим образом:

$$f(x^*) = \min_{x \in X} f(x) \quad \text{или} \quad x^* = \arg\min_{x \in X} f(x).$$

В некоторых случаях, когда известен аналитический вид зависимости оптимизируемой функции f от независимых переменных x и легко вычисляются в аналитическом виде производные оптимизируемой функции, для решения экстремальных задач (11.1) или (11.2) могут быть применены методы исследования функций классического анализа, в основе которых лежат формулировки необходимых и достаточных условий существования экстремума.

Необходимые условия существования экстремума у непрерывной функции $f(x)$ получаются из анализа первой производной $f'(x)$. При этом функция $f(x)$ может иметь экстремальные значения при таких значениях независимой переменной x, при которых производная $f'(x)$ равна нулю либо вообще не существует. Графически равенство нулю производной означает, что касательная к кривой $f(x)$ в этой точке параллельна оси абсцисс (рис. 11.1, *а*). В точках излома функции $f(x)$ производная не существует (рис 11.1, *б*, *в*).

Однако выполнение необходимого условия не означает, что в данной точке функция имеет экстремум (рис. 11.1, *в*). Для определения действительных экстремумов необходимо провести дополнительные исследования. Для этого в классическом анализе используется один из следующих способов.



а) *б*) *в*)

Рис. 11.1. Иллюстрация необходимых условий существования экстремума

1. *Сравнение значений функции*. Этот способ сводится к тому, что со значением функции в точке x^*, «подозреваемой» на экстремум, сравнивают два её значения, рассчитанные в точках, достаточно близких к исследуемой и расположенных слева и справа от неё, т.е. при значениях переменной $x^* - \varepsilon$ и $x^* + \varepsilon$, где ε – малая положительная величина. Если при этом окажется, что оба рассчитанных значения $f(x^* + \varepsilon)$ и $f(x^* - \varepsilon)$ меньше или больше $f(x^*)$, то в точке x^* существует максимум или минимум соответственно. Если же $f(x^*)$ имеет промежуточное значение между $f(x^* + \varepsilon)$ и $f(x^* - \varepsilon)$, то в точке x^* функция *f* экстремума не имеет.

2. *Сравнение знаков первой производной*. В основе этого способа лежит первый достаточный признак существования экстремума, согласно которому, если при переходе через точку, «подозреваемую» на экстремум, первая производная меняет знак, то в этой точке x^* функция имеет экстремум. Причём если знак меняется с «+» на «–», то в точке x^* функция имеет максимум, а если с «–» на «+», – то минимум.

3. *Исследование знаков второй производной*. Этот способ использует второй достаточный признак существования экстремума. Если функция $f(x)$ – непрерывна и имеет непрерывные первую и вторую производные, а точка x^* является «подозрительной» на экстремум, тогда в точке x^* будет минимум функции *f*, если $f''(x^*) > 0$ и максимум – если $f''(x^*) < 0$.

4. *Исследование знаков высших производных*. Если вторая производная в точке x^* равна нулю, то для дальнейшего исследования необходимо вычислить следующую производную и исследовать её знак. При этом в общем случае руководствуются следующим правилом: когда порядок первой, не обращающейся в ноль производной в точке x^*, «подозреваемой» на экстремум, нечётный, то в этой точке функция $f(x)$ не имеет ни максимума, ни минимума, т.е. точка x^* не является точкой экстремума функции $f(x)$. Если же порядок первой, не обращающейся в ноль производной в точке x^*, чётный, то в данной точке есть экстремум функции $f(x)$, который будет максимумом или минимумом в зависимости от того, отрицательна или положительна эта производная.

Рассмотренные способы классического анализа исследования функций верны в том случае, если функция $f(x)$ является функцией

одной переменной. Решение задачи оптимизации значительно усложняется, когда функция f является функцией нескольких независимых переменных. Однако, если функция $f = f(x_1, x_2, ..., x_n)$, а также её первые и вторые производные непрерывны, то для неё можно записать также необходимые и достаточные условия существования экстремума.

Необходимым условием экстремума функции f в точке $x_i^* \left(i = \overline{1, n}\right)$ служит равенство нулю в этой точке первых производных по всем переменным, т.е. точки, в которых возможен экстремум функции, могут быть определены решением системы уравнений:

$$\frac{\partial f(x_1, ..., x_n)}{\partial x_i} = 0, \quad i = \overline{1, n}.$$

Дальнейшее исследование функции $f = f(x_1, ..., x_n)$ сводится к проверке достаточного условия существования экстремума для точки $x_i^* \left(i = \overline{1, n}\right)$, «подозрительной» на экстремум. Для этого составляется матрица вторых частных производных – матрица Гессе (гессиан):

$$G = \left(G_{ij}\right) = \frac{\partial^2 f}{\partial x_i \partial x_j}.$$

Тогда если матрица $G\left(x^*\right)$ положительно определена, то точка x^* является точкой минимума, а если $G\left(x^*\right)$ отрицательно определена, то точка x^* – точка максимума. Для положительной определённости матрицы G необходимо и достаточно, чтобы все главные миноры её были строго положительны, т.е.:

$$\Delta_1 = |G_{11}| > 0, \ \Delta_2 = \begin{vmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{vmatrix} > 0, ..., \quad \Delta_n = \begin{vmatrix} G_{11} & G_{12} & ... & G_{1n} \\ G_{21} & G_{22} & ... & G_{2n} \\ ... & ... & ... & ... \\ G_{n1} & G_{n2} & ... & G_{nn} \end{vmatrix} > 0$$

и наоборот.

Рассмотренные методы классического анализа могут быть использованы при решении оптимизационных задач, если целевая функция имеет простой аналитический вид. В противном случае необходимо пользоваться численными методами. При этом по различным признакам все методы могут быть классифицированы следующим образом:

1. В зависимости от размерности вектора независимых переменных x численные методы делятся на методы одномерной и методы многомерной оптимизации.

2. Градиентные и безградиентные методы в зависимости от того, используются ли при вычислении оптимального значения функции производные целевой функции.

3. В зависимости от того, накладываются на независимые переменные дополнительные ограничения или нет, все задачи делятся на задачи условной и безусловной оптимизации. Задача (11.1) или (11.2) называется задачей безусловной оптимизации, если $X = R^n$ – n-мерное евклидово пространство, т.е., например,

$$f(x) \to \min,\ x \in R^n\ .$$

Если X в постановке (11.1) или (11.2) есть собственное подмножество пространства R^n, то эти задачи представляют собой задачи условной оптимизации. Кроме того, в задачах на условный экстремум на проектные параметры могу быть наложены дополнительные ограничения. Эти ограничения бывают двух типов: ограничения-равенства и ограничения-неравенства. С учётом этого задача на условный экстремум может быть записана в виде:

$$f(x) \to \min,\ x \in X\ ;$$

$$h_i(x) = 0,\ i = \overline{1,\ m}\ ;$$

$$g_i(x) \le 0,\ i = \overline{m+1,\ k}\ .$$

4. В зависимости от вида целевой функции и ограничений: задача линейного программирования, нелинейного или математического программирования (выпуклое, квадратичное), дискретного программирования, геометрического программирования и др.

12. МЕТОДЫ ОДНОМЕРНОЙ ОПТИМИЗАЦИИ

Задача отыскания экстремумов дифференцируемой функции f сводится к решению уравнения $f'(x)=0$. Однако лишь в отдельных случаях решение этого уравнения удаётся найти в явном виде. Поэтому в этом случае прибегают к численным методам экстремального поиска.

Необходимость отдельного рассмотрения численных методов поиска экстремума функции одной переменной диктуется следующими обстоятельствами. Во-первых, эти методы используются во многих алгоритмах поиска экстремума функций, зависящих от нескольких переменных. Во-вторых, иногда удаётся, используя те или иные приёмы, непосредственно с помощью алгоритмов одномерной оптимизации получить решение многомерных задач. В-третьих, классы функций одной переменной служат удобной моделью для теоретического исследования эффективности методов оптимизации.

Задачу одномерной оптимизации можно сформулировать следующим образом. Требуется найти экстремум унимодальной функции одной переменной $f(x)$, непрерывной вместе со своей первой производной на интервале $[a, b]$: $f(x) \to \text{extr}$, $x \in [a, b]$.

Рассмотрим некоторые из методов одномерного поиска на примерах поиска минимумов функции $f(x)$.

12.1. МЕТОД ЛОКАЛИЗАЦИИ

Разобьём весь интервал $[a, b]$ на N равных частей. На границах всех подынтервалов, включая конечные точки интервала $[a, b]$, вычисляются значения функции $f(x_i)$, $i = \overline{0, N}$. Среди полученных значений $f(x_i)$ выбирается наилучшее, т.е. то, которое соответствует типу отыскиваемого экстремума (например, при отыскании минимума наилучшей будет точка x_2 (рис. 12.1)).

После выбирается новый интервал локализации экстремума, состоящий из двух подынтервалов с наилучшей точкой посередине (в нашем случае новый интервал будет равен $[x_1, x_3]$).

Применяя к новому интервалу тот же приём разбиения, и вычисляя значение $f(x)$ на границах полученных подынтервалов, можно ещё больше сузить интервал локализации экстремума. Описанная процедура повторяется до тех пор, пока неравенство $\left|b^{(k)} - a^{(k)}\right| \le \varepsilon$ не станет истинным, где ε – требуемая точность вычислений, k – номер итерации (k = 0, 1, 2, …). Тогда за точку экстремума можно принять среднюю точку интервала $\left[a^{(k)}, b^{(k)}\right]$: $x^* \approx \dfrac{b^{(k)} + a^{(k)}}{2}$.

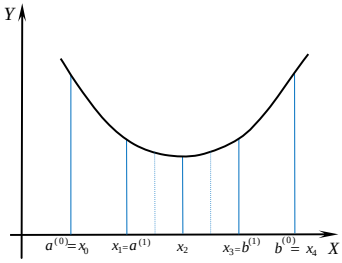


Рис. 12.1. Иллюстрация метода локализации

Очевидно, что наилучшие результаты поиска будут достигнуты в том случае, если интервал локализации разбивается на четыре подынтервала, $N = 4$. В этом случае для каждого разбиения нужно вычислять значения целевой функции только дважды.

Абсолютная и относительная ошибки в нахождения экстремума определяются выражениями: $\Delta_x = (b-a)\cdot 2^{-\frac{s-1}{2}}$ и $\delta_x = \frac{\Delta}{b-a} = 2^{-\frac{s-1}{2}}$, где s – число расчётов значений целевой функции, которое при $N = 4$ может быть только нечётным.

12.2. МЕТОД ДИХОТОМИИ

На интервале локализации экстремума $[a, b]$ определяются две точки $x_1 = \frac{a+b}{2} - \frac{\delta}{2}$ и $x_2 = \frac{a+b}{2} + \frac{\delta}{2}$, где $\delta > 0$ – некоторая константа, называемая параметром метода. Обычно δ определяется количеством верных десятичных знаков при задании аргумента. Как правило, в практических расчётах величину δ принимают равной величине точности ε, с которой ищется решение экстремальной задачи, т.е. $\delta \approx \varepsilon$.

В точках x_1 и x_2 вычисляются значения функции $f(x_1)$ и $f(x_2)$, которые сравниваются между собой. Если окажется, что $f(x_1) \geq f(x_2)$, то новый интервал локализации экстремума будет равен $[x_1, b]$. В противном случае, т.е. если $f(x_1) \leq f(x_2)$, интервал локализации сузится до $[a, x_2]$ (рис. 12.2).

На новом интервале снова определяются две точки x_1 и x_2 и процедура повторяется до тех пор, пока на некотором k-м шаге не выполнится неравенство: $\left|b^{(k)} - a^{(k)}\right| \leq \varepsilon$.

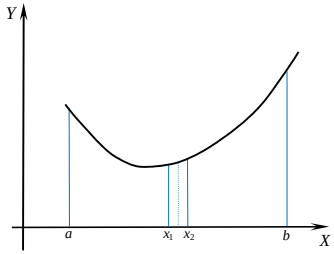


Рис. 12.2. Иллюстрация метода дихотомии

Абсолютная и относительная погрешность метода определяется в соответствии с выражениями: $\Delta_x = (b-a)\cdot 2^{-\frac{s+1}{2}}$ и $\delta_x = 2^{-\frac{s+1}{2}}$. Здесь s показывает сколько раз уменьшался интервал локализации экстремума.

12.3. МЕТОД «ЗОЛОТОГО СЕЧЕНИЯ»

В предыдущих методах среди всех значений функции, вычисленных в интервале неопределённости, в дальнейшем используются только некоторые, а остальные не дают дополнительной информации и в дальнейшем не используются. В методе «золотого сечения» целевая функция вычисляется в точках интервала неопределённости, расположенных таким образом, чтобы каждое вычисленное значение целевой функции давало новую полезную информацию.

«Золотым сечением» отрезка называется такое его деление на две неравные части, что отношение длины большей части отрезка l_1 к длине всего отрезка (l_1 + l_2) равно отношению длины меньшей части l_2 к длине большей части, т.е. $\frac{l_1}{l_1+l_2} = \frac{l_2}{l_1}$.

Суть метода «золотого сечения» заключается в следующем. На интервале $[a, b]$ определяются две точки:

$$x_1 = a + \left(3-\sqrt{5}\right)\frac{b-a}{2} = a + 0{,}381966...(b-a) \text{ и}$$

$$x_2 = a + \left(\sqrt{5}-1\right)\frac{b-a}{2} = a + 0{,}618034...(b-a),$$

находящиеся на одинаковом расстоянии от концов интервала $[a, b]$ соответственно. Вычисляются значения целевой функции в этих точках $f(x_1)$ и $f(x_2)$, которые сравниваются между собой. Если $f(x_1) \geq f(x_2)$, то новый интервал неопределённости будет равен $[x_1, b]$, в противном случае, т.е. если $f(x_1) \leq f(x_2)$, интервал неопределённости равен $[a, x_2]$.

На новом интервале неопределённости необходимо заново определить две промежуточные точки и сравнить значение функции в них. Однако для интервала $[a, x_2]$ точка x_1 уже является его «золотым сечением» $\left(\frac{ax_1}{ax_2} = \frac{x_1x_2}{ax_1}\right)$, также как и точка x_2 является «золотым сечением» интервала $[x_1, b]$ $\left(\frac{x_1x_2}{x_2b} = \frac{x_2b}{x_1b}\right)$. Поэтому в методе «золотого сечения» на каждом шаге требуется всего лишь один раз вычислить новое значение функции.

Процесс уменьшения интервала неопределённости продолжается до тех пор, пока неравенство $\left|b^{(k)} - a^{(k)}\right| \le \varepsilon$ не станет верным.

Точность метода определяется выражением:

$$\Delta_x = \frac{b-a}{2}\left(\frac{\sqrt{5}-1}{2}\right)^{s-3},$$

где Δ_x – абсолютная ошибка в определении экстремума после s вычислений значений $f(x)$.

12.4. МЕТОД ФИБОНАЧЧИ

Последовательность чисел, описываемая рекуррентным соотношением

$$F_k = F_{k-1} + F_{k-2},\ F_0 = F_1 = 1,$$

называется числами Фибоначчи. Эти числа можно использовать для организации поиска экстремума функции одной переменной. Алгоритм оптимального поиска состоит из следующих шагов:

1. По заданной точности ε, с которой необходимо найти положение экстремума функции $f(x)$ в интервале $[a, b]$, рассчитывается вспомогательное число N: $N = \frac{b-a}{s}$.

2. Для полученного значения N выбирается такое число Фибоначчи F_s, чтобы выполнялось неравенство: $F_{s-1} < N < F_s$.

3. Определяется минимальный шаг поиска $h = \frac{b-a}{F_s}$.

4. Рассчитывается значение функции в точке a, т.е. $f(a)$.

5. Определяется следующая точка, в которой вычисляется значение функции: $x_1 = a + hF_{s-2}$.

6. Если полученная точка x_1 оказалась удачной, т.е. $f(x_1) < f(a)$, то следующая точка определяется как $x_2 = x_1 + hF_{s-3}$.

В противном случае, если шаг неудачный, т.е. $f(x_1) > f(a)$, точка x_2 определяется в соответствии с выражением $x_2 = x_1 - hF_{s-3}$.

7. Последующие шаги выполняются с уменьшающейся величиной шага, которая для i-го шага будет равна: $\Delta x_i = hF_{s-i-2}$, в соответствии со следующим правилом.

Если при выполнении шага Δx_i значение функции улучшилось, т.е. шаг оказался удачным и $f(x_{i+1}) < f(x_i)$, то следующий шаг будет выполняться из точки x_{i+1} в том же направлении, что и шаг Δx_i, т.е. $x_{i+2} = x_{i+1} + \Delta x_{i+1}$.

Если же шаг Δx_i оказался неудачным, т.е. $f(x_{i+1}) > f(x_i)$, то следующий шаг выполняется из точки x_i в противоположном направлении: $x_{i+2} = x_i - \Delta x_{i+1}$ (рис. 12.3).

Поиск заканчивается, когда будет сделан последний шаг, использующий число F_0.

Отличительной особенностью метода Фибоначчи является то, что используя этот метод, практически в самом начале экстремального поиска определяется количество шагов, которое необходимо сделать для нахождения экстремума целевой функции с заданной точностью. Абсолютная погрешность описанного алгоритма определяется максимальным шагом поиска h.

Алгоритм экстремального поиска функции одной переменной, использующий последовательность чисел Фибоначчи, может быть организован и по аналогии с рассмотренными выше методами локализации экстремума.

В этом случае на начальном этапе по заданной точности ε определяется вспомогательное число N, выбирается число F_s и вычисляется минимальный шаг поиска h (см. выше и п. 1 – 3). Затем на интервале неопределённости $[a, b]$ определяются две точки x_1 и x_2: $x_1 = a + hF_{s-2}$ и $x_2 = a + hF_{s-1}$, равноотстоящие от концов отрезка $[a, b]$ (рис. 12.4).

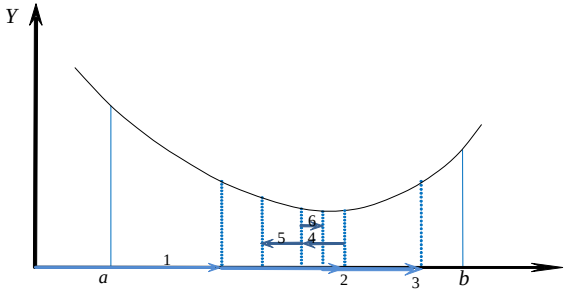


Рис. 12.3. Иллюстрация метода Фибоначчи

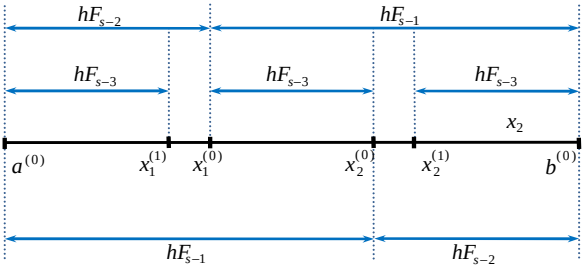


Рис. 12.4. Иллюстрация метода Фибоначчи
(2 вариант)

В точках x_1 и x_2 вычисляются значения функции и сравниваются между собой. Если $f(x_1) < f(x_2)$, то новый интервал неопределённости будет $[a, x_2]$, т.е. $a^{(1)} = a^{(0)}$, $b^{(1)} = x_2$. Для этого интервала снова определяются две точки, при этом $x_2^{(1)} = x_1^{(0)}$, $x_1^{(1)} = a^{(1)} + hF_{s-3}$, вычисляются значения функции в них и сравниваются между собой. Если $f(x_1) > f(x_2)$, то новый интервал неопределённости будет $[x_1, b]$, т.е. $a^{(1)} = x_1^{(0)}$, $b^{(1)} = b^{(0)}$. Для этого интервала вычисляются $x_1^{(1)} = x_2^{(0)}$, $x_2^{(1)} = b^{(1)} - hF_{s-3}$, определяются $f\left(x_1^{(1)}\right)$ и $f\left(x_2^{(1)}\right)$, которые затем сравниваются между собой.

Процедура сужения интервала неопределённости продолжается до тех пор, пока в расчётах не будет использовано число F_0.

12.5. МЕТОД ДСК (ДЭВИСА, СВЕННА, КЕМПИ)

Из начальной точки интервала локализации экстремума a с некоторым начальным шагом Δx_0 (или из точки b с шагом $-\Delta x_0$) делают первый шаг и вычисляют значение функции $f(a + \Delta x_0)$. Если функция улучшилась, т.е. $f(a + \Delta x_0) < f(a)$, то начальный шаг удваивают и проверяют значение функции в следующей точке $f(a + 2\Delta x_0)$. Таким образом, шагают, удваивая каждый раз величину шага до тех пор, пока функция улучшается (рис. 12.5).

Как только некоторое значение функции $f\left(x^{(k+1)}\right)$ становится хуже предыдущего, т.е. $f\left(x^{(k+1)}\right) > f\left(x^{(k)}\right)$, то текущий шаг $\left(\Delta x = x^{(k+1)} - x^{(k)}\right)$ уменьшают в два раза $\left(\Delta x = \Delta x/2\right)$ и делают один шаг в обратном направлении, т.е. $x^{(k+2)} = x^{(k+1)} - \Delta x$. В итоге получаем четыре равноотстоящие точки $x^{(k-1)}$, $x^{(k)}$, $x^{(k+2)}$, $x^{(k+1)}$.

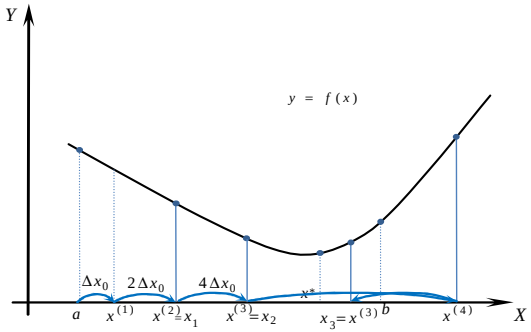


Рис. 12.5. Иллюстрация метода ДСК

Если $f\left(x^{(k)}\right) < f\left(x^{(k+2)}\right)$, то точку $x^{(k+1)}$ можно отбросить. В противном случае, т.е. если $f\left(x^{(k)}\right) > f\left(x^{(k+2)}\right)$ отбрасываем из рассмотрения точку $x^{(k-1)}$. Оставшиеся три точки ($x^{(k-1)}$, $x^{(k)}$, $x^{(k+2)}$ или $x^{(k)}$, $x^{(k+2)}$, $x^{(k+1)}$) обозначим как x_1, x_2, x_3 соответственно. Тогда приближённое значение минимума может быть вычислено по формуле

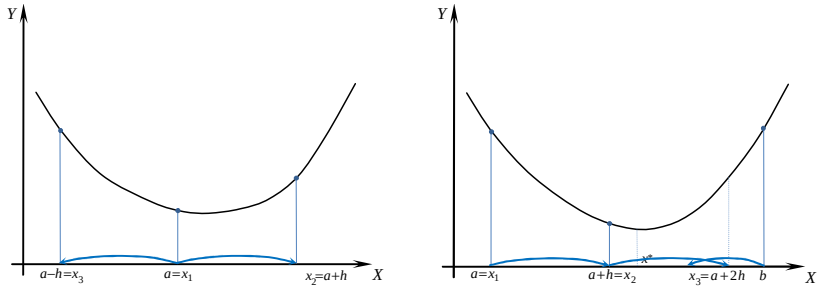
$$x_k^* = x_2 + \frac{\Delta x\left(f(x_1) - f(x_3)\right)}{2\left(f(x_1) - 2f(x_2) + f(x_3)\right)},\ k = 1, 2, 3, \ldots.$$

Если требуемая точность не достигнута, т.е. $\left|f\left(x_k^*\right) - f\left(x_{k-1}^*\right)\right| > \varepsilon$ или $\left|x_k^* - x_{k-1}^*\right| > \varepsilon$, то вся процедура вычислений повторяется сначала, но уже из точки x_2 или x_k^* с начальным шагом $\Delta x_0 = \Delta x/2$.

12.6. МЕТОД ПАУЭЛЛА

Этот метод также встречается под названием метода квадратичной аппроксимации и основан на последовательном применении процедуры оценивания с использованием квадратичной аппроксимации.

Зададим некоторый шаг h, являющийся величиной того же порядка, что и расстояние от некоторой точки $a = x_1$ до точки истинного минимума. Определим некоторую точку $x_2 = a + h$ и вычислим значения функции $f(a)$ и $f(a+h)$. Если $f(a) < f(a+h)$, то определяем третью точку $x_3 = a - h$ (рис. 12.6, *а*). В противном случае, т.е. если $f(a) > f(a+h)$, в качестве третьей точки выберем точку $x_3 = a + 2h$ (рис. 12.6, *б*).



а) *б*)

Рис. 12.6. Иллюстрация метода Пауэлла

Вычислим приближённое значение минимума целевой функции по формулам:

$$x_1^* = \frac{1}{2}\frac{\left(x_2^2 - x_3^2\right)f\left(x_1\right) + \left(x_3^2 - x_1^2\right)f\left(x_2\right) + \left(x_1^2 - x_2^2\right)f\left(x_3\right)}{\left(x_2 - x_3\right)f\left(x_1\right) + \left(x_3 - x_1\right)f\left(x_2\right) + \left(x_1 - x_2\right)f\left(x_3\right)} \qquad (12.1)$$

или

$$x_k^* = \frac{\left(x_1 + x_2\right)}{2} + \frac{1}{2}\frac{\left(f\left(x_1\right) - f\left(x_2\right)\right)\left(x_2 - x_3\right)\left(x_3 - x_1\right)}{\left(x_2 - x_3\right)f\left(x_1\right) + \left(x_3 - x_1\right)f\left(x_2\right) + \left(x_1 - x_2\right)f\left(x_3\right)},$$

$$k = 2, 3, 4, \ldots. \qquad (12.2)$$

Для вычисления минимума в первом приближении используется формула (12.1), а на последующих итерациях – формула (12.2).

Проверяется неравенство, выполнение которого заканчивает процедуру поиска экстремума:

$$\left| f\left(x_k^*\right) - f_{\min} \right| \le \varepsilon \text{ и(или) } \left| x_k^* - x_{\min} \right| \le \varepsilon, \qquad (12.3)$$

где

$$f_{\min} = \min\left\{f\left(x_1\right),\ f\left(x_2\right),\ f\left(x_3\right)\right\},\ x_{\min} = \arg\min_{x_i = \{x_1,\, x_2,\, x_3\}} f\left(x_i\right).$$

Если неравенства (12.3) не выполняются, то из четырёх точек x_1, x_2, x_3, x_k^* выбирается наилучшая, и две точки по обе стороны её, которые переобозначаются как x_1, x_2, x_3 и поиск повторяется по формуле (12.2).

13. МЕТОДЫ НЕЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ БЕЗ ОГРАНИЧЕНИЙ

Математическая формулировка задачи нелинейного программирования без ограничений может быть представлена как задача отыскания наибольшего или наименьшего значения функции нескольких переменных:

$$f = f(x_1, x_2, ..., x_n) \to \min_{x_i \in X} . \qquad (13.1)$$

В большинстве случаев сложный вид целевой функции f не позволяет применить для решения задачи нелинейного программирования аналитические методы. Поэтому возникает необходимость применения вычислительной техники.

В настоящее время для решения задач нелинейного программирования разработано и применяется довольно значительное количество численных методов. Однако отдать предпочтение какому-либо одному из них пока не представляется возможным.

Рассмотрим вначале некоторые особенности функций нескольких переменных. Геометрическая иллюстрация таких функций (за исключением функций двух переменных) отсутствует. Поэтому прибегают к следующему приёму представления функции $f(x_1, x_2, ..., x_n) = f(\boldsymbol{x})$ на плоском чертеже. Пусть $\boldsymbol{x}^*$ – экстремальное значение функции (13.1). Тогда вокруг точки $\boldsymbol{x}^*$ можно провести множество линий, вдоль которых значение функции $f(\boldsymbol{x})$ меняться не будут. Эти линии называются *линиями уровня* (рис. 13.1).

Часто среди функций нескольких переменных встречаются «седловые» точки и «овраги». Поиск экстремальных точек в этом случае сильно затрудняется.

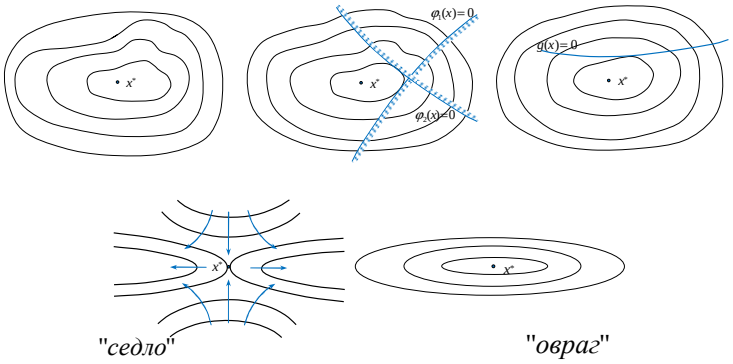


Рис. 13.1. Примеры линий уровня функций нескольких переменных

При решении задач нелинейного программирования также необходимо помнить и следующее. В конкретных задачах независимые переменные могут иметь самый различный физический смысл и соответственно разные единицы и диапазоны измерения. Поэтому при решении оптимальных задач численными методами целесообразно оперировать с безразмерными нормированными значениями независимых переменных. Обычно для нормирования независимой переменной её делят на возможный диапазон изменения, который всегда может быть установлен, исходя из физической сущности решаемой задачи: $x = \dfrac{\tilde{x} - x_{\min}}{x_{\max} - x_{\min}}$.

В этом случае значения x будут лежать в интервале от 0 до 1.

13.1. МЕТОД ПОКООРДИНАТНОГО СПУСКА

Метод покоординатного спуска (подъёма, поочерёдного изменения переменных, Гаусса–Зейделя) является одним из наиболее простых методов прямого поиска. Его суть заключается в поочерёдном изменении независимых переменных таким образом, чтобы по каждой из них достигалось экстремальное значение. Очерёдность варьирования независимых переменных при этом устанавливается произвольно и обычно не меняется в процессе поиска. Рассмотрим алгоритм метода покоординатного спуска.

Выбирают некоторую начальную точку для поиска $\boldsymbol{x}^{(0)} = \left(x_1^{(0)}, x_2^{(0)}, ..., x_n^{(0)}\right)$. Фиксируют все неизвестные, кроме первой ($j = 1$), т.е. $u_i = x_i^{(0)} \forall_i,\ i \neq j$.

Одним из методов одномерного поиска ищут экстремум функции $f\left(x_j, \boldsymbol{u}_i\right)$, т.е. $f\left(x_j^{(1)}, \boldsymbol{u}_i\right) = \text{extr} f\left(x_j, \boldsymbol{u}_i\right)$. После этого за варьируемую неизвестную принимают x_2, а все остальные фиксируют $\left(\boldsymbol{u}_i = \left\{x_1^{(1)}, x_3^{(0)}, x_4^{(0)}, ..., x_n^{(0)}\right\}\right)$ и повторяют одномерный поиск, т.е. $f\left(x_2, \boldsymbol{u}_i\right)$. Описанную процедуру продолжают до тех пор, пока не переберут все неизвестные до x_n включительно (рис. 13.2). Затем проверяют условия окончания поиска: $\left\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}\right\| \leq \varepsilon,\ k = 1, 2, 3, ...$ или $\left| f\left(\boldsymbol{x}^{(k)}\right) - f\left(\boldsymbol{x}^{(k-1)}\right)\right| \leq \varepsilon$.

Если одно из этих условий или их комбинация выполняется, то поиск оканчивается. В противном случае серия одномерных поисков повторяется уже для новой точки $\boldsymbol{x}^{(k)}$.

Для метода покоординатного спуска характерны простота и сравнительно небольшой объём вычислений. В то же время при наличии ограничений и особенностей целевой функции, например «оврагов», поиск этим методом затрудняется.

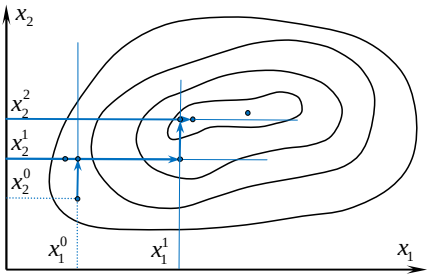


Рис. 13.2. Иллюстрация метода покоординатного спуска

Для тестирования метода покоординатного спуска и всех рассматриваемых далее методов могут использоваться функции Розенброка и Пауэлла.

Функция Розенброка

$$f(x_1, x_2) = 100\left(x_2 - x_1^2\right)^2 + (1 - x_1)^2, \quad \boldsymbol{x}^* = (1; 1).$$

Функция Пауэлла

$$f(x_1, x_2, x_3, x_4) = \left(x_1 + 10x_2\right)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4, \quad \boldsymbol{x}^* = (0; 0; 0; 0).$$

13.2. МЕТОД ХУКА–ДЖИВСА

Этот алгоритм прямого поиска состоит из следующих операций. Задаётся начальная точка $\boldsymbol{x}^{(0)} = \left\{x_1^{(0)}, x_2^{(0)}, ..., x_n^{(0)}\right\}$ и начальные приращения $\Delta\boldsymbol{x}^{(0)} = \left\{\Delta x_1^{(0)}, \Delta x_2^{(0)}, ...\right\}$ каждой координаты точки $\boldsymbol{x}^{(0)}$, которые могут быть различными. Процедура поиска состоит из «исследующих поисков» типа I и типа II и «поиска по образцу». Чтобы начать «исследующий поиск типа I», вычисляют значение функции в базисной точке (за неё принимается вначале точка $\boldsymbol{x}^{(0)}$). Затем в циклическом порядке изменяется каждая переменная точки $\boldsymbol{x}$ (каждый раз только одна) на выбранные величины приращений. Пока все координаты точки $\boldsymbol{x}^{(0)}$ не будут таким образом перебраны. При этом если приращение $\Delta x_i^{(0)}$ для переменной $x_i^{(0)}$ не улучшает целевую функцию $f(\boldsymbol{x})$, то проверяется приращение $-\Delta x_i^{(0)}$. А если и это не даёт улучшения, то координата $x_i^{(0)}$ точки $\boldsymbol{x}^{(0)}$ остаётся без изменения.

После того как все координаты точки $\boldsymbol{x}^{(0)}$ были подобным образом изменены, получается новая точка $\boldsymbol{x}^{(1)}$, принимаемая за новую базисную точку. На этом «последующий поиск типа I» заканчивается и начинается «поиск по образцу», который заключается в реализации единственного шага из полученной базисной точки вдоль прямой, соединяющей эту точку с предыдущей базисной точкой. Для полученной в ходе «поиска по образцу» (ПО) точки выполняется «исследующий поиск типа II» (ИП2), аналогичный «исследующему поиску типа I» (ИП1). Если точка, получаемая после ИП2, оказывается лучше предыдущей базисной точки, то вновь выполняется ПО. В противном случае изменяются величины приращений $\Delta\boldsymbol{x}$ и поиск повторяется, начиная с ИП1.

Пошаговый алгоритм метода поиска Хука–Дживса может быть записан следующим образом:

Шаг 1: Определить начальную точку $\boldsymbol{x}^{(0)}$; приращения Δx_i, $i=\overline{1,n}$; коэффициент уменьшения шага $\alpha>1$ и точность поиска $\varepsilon>0$.

Шаг 2: Провести ИП1.

Шаг 3: ИП1 – удачен? Да – шаг 5. Нет – шаг 4.

Шаг 4: Проверяем условия окончания поиска. Выполняется ли неравенство $\|\Delta\boldsymbol{x}\|<\varepsilon$? Если да – поиск окончен и текущая точка является точкой экстремума. В противном случае – уменьшить шаг $\Delta x_i=\Delta x_i/\alpha$, $i=\overline{1,n}$ и перейти к шагу 2.

Шаг 5: Провести ПО: $\boldsymbol{x}_p^{(k+1)}=\boldsymbol{x}^{(k)}+\left(\boldsymbol{x}^{(k)}-\boldsymbol{x}^{(k-1)}\right)$, где $\boldsymbol{x}^{(k)}$ – текущая базисная точка; $\boldsymbol{x}^{(k-1)}$ – предыдущая базисная точка; $\boldsymbol{x}_p^{(k+1)}$ – точка, построенная при движении по образцу.

Шаг 6: Провести ИП2, используя $\boldsymbol{x}_p^{(k+1)}$ в качестве базисной точки. Получаем точку $\boldsymbol{x}^{(k+1)}$ – новая базисная точка.

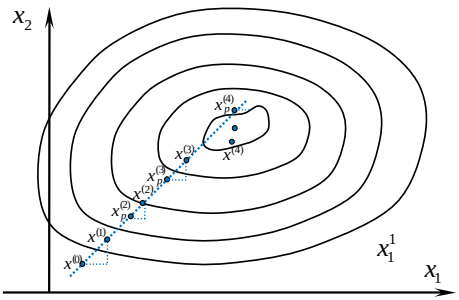


Рис. 13.3. Иллюстрация метода Хука–Дживса

Шаг 7: Проверяем неравенство $f\left(\boldsymbol{x}^{(k+1)}\right) < f\left(\boldsymbol{x}^{(k)}\right)$. Если оно выполняется, то принимают $\boldsymbol{x}^{(k-1)} = \boldsymbol{x}^{(k)}$, $\boldsymbol{x}^{(k)} = \boldsymbol{x}^{(k+1)}$ и переходят к шагу 5 (ПО). В противном случае делают переход к шагу 4 (рис. 13.3).

Метод Хука–Дживса, предложенный в 1961 г. является весьма эффективным и оригинальным. Его достоинство заключается в том, что в процессе поиска существует возможность возврата и поиска в новом направлении.

Рассмотрим работу алгоритма Хука и Дживса на следующем примере:

***Пример*:** найти максимум функции $f(x) = \dfrac{1}{(x_1+1)^2 + x_2^2}$.

1) Примем

$\boldsymbol{x}^{(0)} = (1;\ 2{,}8),\ \Delta\boldsymbol{x}^{(0)} = (0{,}6;\ 0{,}84),\ \varepsilon = 0{,}001,\ \alpha = 2$.

2) $f\left(x^{(0)}\right) = 0{,}059$.

3) ИП1: $x_1^{(1)} = 2 + 0{,}6 = 2{,}6 \to f(2{,}6;\ 2{,}8) = 0{,}048\ \ (-)$;

$x_1^{(1)} = 2 - 0{,}6 = 1{,}4 \to f(1{,}4;\ 2{,}8) = 0{,}073\ \ (+)$;

$x_2^{(1)} = 2{,}8 + 0{,}84 = 3{,}64 \to f(1{,}4;\ 3{,}64) = 0{,}052\ \ (-)$.

$x_2^{(1)} = 2{,}8 - 0{,}84 = 1{,}96 \to f(1{,}4;\ 1{,}96) = 0{,}104\ \ (+)$.

ИП1 – удачен. Базисная точка $x^{(1)} = (1{,}4;\ 1{,}96)$.

4) ПО: $x_{1p}^{(2)} = x_1^{(1)} + \left(x_1^{(1)} - x_1^{(0)}\right) = 0{,}8$;

$x_{2p}^{(2)} = x_2^{(1)} + \left(x_2^{(1)} - x_2^{(0)}\right) = 1{,}12$;

$f(0{,}8;\ 1{,}12) = 0{,}22$.

5) ИП2: $x_1^{(2)} = 0{,}8 + 0{,}6 = 1{,}4 \to f(1{,}4;\ 1{,}12) = 0{,}14\ \ (-)$;

$x_1^{(2)} = 0{,}8 - 0{,}6 = 0{,}2 \to f(0{,}2;\ 1{,}12) = 0{,}38\ \ (+)$;

$x_2^{(2)} = 1{,}12 + 0{,}84 = 1{,}96 \to f(0{,}2;\ 1{,}96) = 0{,}19\ \ (-)$;

$x_2^{(2)} = 1{,}12 - 0{,}84 = 0{,}28 \to f(0{,}2;\ 0{,}28) = 0{,}67\ \ (+)$;

Так как $f(0{,}2;\ 0{,}28) = 0{,}67 > f(1{,}4;\ 1{,}96) = 0{,}104$, то ПО – удачен.

6) Затем вновь проводится ПО и т.д.

13.3. МЕТОД ПОИСКА ПО ДЕФОРМИРУЕМОМУ МНОГОГРАННИКУ

Одна из наиболее интересных стратегий поиска экстремума положена в основу *метода поиска по симплексу*, предложенного Спендли, Хекстом, Химсвортом. В этом методе используется понятие *регу-*

лярного симплекса, который в N-мерном пространстве представляет собой многогранник, образованный $N + 1$ равноотстоящими друг от друга точками-вершинами. Например, в случае двух переменных симплексом является равносторонний треугольник; в трёхмерном пространстве симплекс представляется собой тетраэдр.

Работа алгоритма симплексного поиска начинается с построения регулярного симплекса в пространстве независимых переменных и оценивания значений целевой функции в каждой из вершин симплекса. При этом определяется вершина, которой соответствует наихудшее значение целевой функции. Затем найденная вершина проецируется через центр тяжести остальных вершин симплекса в новую точку, которая используется в качестве вершины нового симплекса. Итерации продолжаются до тех пор, пока не начнётся циклическое движение по двум или более симплексам. В этом случае размеры симплекса уменьшаются, и весь поиск повторяется, пока размеры симплекса не станут меньше некоторой заданной точности.

Однако в таком варианте алгоритм симплексного метода работает слишком медленно, так как полученная на предыдущих итерациях информация не используется для ускорения поиска. Этот недостаток частично устранён в модифицированной процедуре поиска по симплексу, разработанной Нелдером и Мидом, называемой *методом Нелдера-Мида* или *методом деформируемого многогранника*, являющегося одним из самых эффективных методов при $n \le 6$.

Метод Нелдера–Мида допускает использование неправильного симплекса, к которому могут быть применены операции *отражения*, *растяжения*, *сжатия* и *редукции*.

Рассмотрим алгоритм метода деформируемого многогранника на примере поиска min.

1. Строится начальный симплекс. Обычно (но не обязательно) начальный многогранник выбирается в виде регулярного симплекса. В вершинах симплекса вычисляются значения функции $f_1 = f(x_1),\ f_2 = f(x_2),\ ...,\ f_{n+1} = f(x_{n+1})$.

2. Среди точек симплекса отыскивается наибольшее значение f_h, следующее за наибольшим значением функции f_g, наименьшее значение функции f_l и соответствующие им точки x_h, x_g, x_l.

3. Ищется центр тяжести всех точек, за исключением x_h:

$$x_0 = \frac{1}{n}\sum_{i \ne h} x_i \quad \text{и} \quad f_0 = f(x_0).$$

4. Выполняют отражение точки x_h относительно точки x_0 с коэффициентом отражения $\alpha > 0$:

$$x_r = x_0 + \alpha(x_0 - x_h),\ \ f_r = f(x_r).$$

5. Если $f_r < f_l$, то направление из точки x_0 в точку x_r наиболее удобно для перемещения. Поэтому выполняют операцию растяжения с коэффициентом растяжения $\gamma > 0$:

$$x_p = x_0 + \gamma(x_r - x_0),\ f_p = f(x_p).$$

6. Если растяжение прошло удачно, т.е. $f_p < f_l$, то новый симплекс будет построен из точки x_p и оставшихся точек x_i предыдущего симплекса (кроме x_h). В противном случае, когда $f_p > f_l$, вместо точки x_h в новом симплексе берём точку x_r. Если новый симплекс не отвечает условиям окончания поиска

$$\delta = \sqrt{\frac{1}{n+1}\sum_{i=1}^{n+1}\left(f_i - \bar{f}\right)^2} < \varepsilon, \text{ где } \bar{f} = \frac{\sum_{i=1}^{n+1} f_i}{n+1}, \qquad (13.2)$$

то с новым симплексом возвращаются на шаг 2. Если же условие (13.2) выполняется, то любая из вершин симплекса может быть принята за решение экстремальной задачи.

7. Если отражение было неудачным $(f_r > f_l)$, то сравнивают f_r и f_g. Если $f_r < f_g$, то новый симплекс строят, используя вместо точки x_h точку x_r, и после проверки условия сходимости возвращаются на шаг 2. Если же $f_r > f_g$, то переходят к операции сжатия симплекса (шаг 8).

8. Сравнивают f_r и f_h. По результатам сравнения оставляют начальный симплекс без изменения (если $f_r > f_h$) либо строят новый, используя вместо x_h точку x_r (при $f_r < f_h$). Для выбранного симплекса выполняют операцию сжатия с параметром $\beta > 0$: $x_c = x_0 + \beta(x_h - x_0),\ f_c = f(x_c)$.

9. Выполнив сжатие, проверяют успешно ли прошла эта операция. Если $f_c < f_h$, то сжатие прошло удачно и для нового симплекса (заменяя x_h на x_c) проверяют условия сходимости. Если же сжатие неудачно, то осуществляют редукцию симплекса $(f_c > f_h)$.

10. Операция редукции заключается в уменьшении размеров симплекса по формулам $x_i = \frac{x_i + x_l}{2}$, $i = \overline{1, n+1}$.

После вычисления значений функций $f_i = f(x_i)$, $i = \overline{1, n+1}$ снова проверяют условия сходимости (13.2).

Коэффициенты отражения, растяжения и сжатия Нелдер и Мид рекомендуют выбирать следующими: $\alpha = 1$; $\beta = 0{,}5$; $\gamma = 2$. Другие рекомендации (Паркинсон, Хатчинсон, 1972 г.) предлагают следующие значения для коэффициентов: $\alpha = 2$; $\beta = 0{,}25$; $\gamma = 2{,}5$.

13.4. МЕТОД СОПРЯЖЁННЫХ НАПРАВЛЕНИЙ ПАУЭЛЛА

Наиболее эффективным из алгоритмов прямого поиска является метод, разработанный Пауэллом. При работе этого метода информация, полученная на предыдущих операциях, используется для построения векторов направлений поиска, а также для устранения зацикливания последовательности координатных поисков. Метод ориентирован на решение задач с квадратичными целевыми функциями и основывается на фундаментальных теоретических результатах.

Задачи с квадратичными целевыми функциями занимают важное место в теории оптимизации по двум причинам. Во-первых, квадратичная функция представляет собой простейший тип нелинейных функций, для которых может быть сформулирована задача безусловной оптимизации. А во-вторых, в окрестности точки оптимума любую нелинейную функцию можно аппроксимировать квадратичной функцией.

Основная идея алгоритма заключается в том, что если квадратичная функция *N* переменных приведена к виду суммы полных квадратов ($y = ax_1^2 + bx_2^2 + cx_1 + dx_2 + e$ – для функции двух переменных), то её оптимум может быть найден в результате реализации N^2 одномерных поисков по преобразованным координатным направлениям. В общем же случае, когда целевая функция не является квадратичной, необходимо провести более чем N^2 одномерных поисков.

Для проведения одномерных поисков строится система сопряжённых направлений.

Определение: Пусть задана квадратичная функция $f(x)$, две произвольные несовпадающие точки x_1 и x_2, а также направление *S*. Тогда, если $y_1 \sim \min f\left(x_1 + \lambda^* S\right)$, то направление $\left(y_2 - y_1\right)$ будет сопряжено с *S* (рис. 13.4).

Рассмотрим алгоритм метода сопряжённых направлений Пауэлла.

Шаг 1: Задаётся начальная точка $x^{(0)}(k = 0)$ и система *N* линейно независимых направлений. Обычно на этом шаге в качестве начальных направлений $S_i^{(0)}$ выбираются единичные вектора, т.е. $S_i^{(0)} = e_i$, $i = 1, 2, ..., N$, совпадающие с направлениями координатных осей.

Шаг 2: Проводится последовательно $N + 1$ одномерный поиск целевой функции $f(x)$ по направлениям S_N, S_{N-1}, S_{N-2}, ..., S_2, S_1, S_N; $fx_i^{(k)} + \lambda S_i^{(k)} \to \min \lambda$, $i = N, N-1, ..., 1, N$. При этом полученная ранее точка оптимума берётся в качестве исходной для следующего одномерного поиска, а направление S_N используется как при первом, так и последнем поиске.

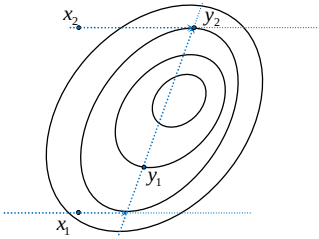


Рис. 13.4. Сопряжённые направления

Шаг 3: Определяется новое сопряжённое направление $r = \dfrac{\bar{\bar{x}}_N^{(k)} - \bar{x}_N^{(k)}}{\left\| \bar{\bar{x}}_N^{(k)} - \bar{x}_N^{(k)} \right\|}$, где $\bar{x}_N^{(k)}$ и $\bar{\bar{x}}_N^{(k)}$ – точки оптимума, полученные на шаге 2 при первом поиске в направлении S_N и $N + 1$ поиске в направлении S_N соответственно.

Шаг 4: Выбираются новые направления поиска: $S_1^{(k+1)} = S_2^{(k)}$; $S_2^{(k+1)} = S_3^{(k)}$; ...; $S_N^{(k+1)} = r$. Таким образом, направление $S_N^{(k)}$ заменяется сопряжённым направлением *r*. Полагая $k = k + 1$, переходят к шагу 2.

Описанный алгоритм позволяет отыскать оптимальное значение квадратичной функции в результате реализации циклов, включающих шаги 2, 3, 4 (*N* – количество независимых переменных). Если же целевая функция не является квадратичной, то итерационное выполнение шагов 2, 3, 4 продолжается до тех пор, пока не выполнится одно из условий: $\left\| x^{(k+1)} - x^{(k)} \right\| < \varepsilon$ или $\left| f\left(x^{(k+1)}\right) - f\left(x^{(k)}\right) \right| < \varepsilon$.

***Пример*:** Найти минимум функции

$$f = (x_1 - 3)^2 + (x_2 - 5)^2 \left(x^* = (3{,}5)\right).$$

Решение:

1. Выберем в качестве начальной точки $x^{(0)} = (0, 0)$, $f\left(x^{(0)}\right) = 34$ и начальные направления, совпадающие с направлениями координатных осей: $S_1^{(0)} = (1,\ 0)$ и $S_2^{(0)} = (0,\ 1)$.

2. Выполняем одномерные поиски:

а) $f\left(x^{(0)} + \lambda S_2^{(0)}\right) \to \min$

$$\left(\left(x_1^{(0)} + \lambda S_{2,1}^{(0)}\right) - 3\right)^2 + \left(\left(x_2^{(0)} + \lambda S_{2,2}^{(0)}\right) - 2\right)^2 =$$

$$= (0 + \lambda \cdot 0 - 3)^2 + (0 + \lambda \cdot 1 - 5)^2 = 9 + (\lambda - 5)^2 \to \min_\lambda;$$

$$\lambda^* = 5;$$

$$x^{(0)'} = x^{(0)} + \lambda^* S_2^{(0)} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} + 5\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 5 \end{pmatrix};\ f\left(x^{(0)}\right) = 0.$$

б) $f\left(x^{(0)} + \lambda S_1^{(0)}\right) \to \min$

$$\left(\left(x_1^{(0)'} + \lambda S_{1,1}^{(0)}\right) - 3\right)^2 + \left(\left(x_2^{(0)'} + \lambda S_{1,2}^{(0)}\right) - 5\right)^2 =$$

$$= (0 + \lambda \cdot 1 - 3)^2 + (5 + \lambda \cdot 0 - 5)^2 = +(\lambda - 3)^2 \to \min_\lambda;\ \lambda^* = 3;$$

$$x^{(0)''} = x^{(0)'} + \lambda^* S_1^{(0)} = \begin{pmatrix} 0 \\ 5 \end{pmatrix} + 3\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 3 \\ 5 \end{pmatrix};\ f\left(x^{(0)}\right) = 0.$$

в) $f\left(x^{(0)} + \lambda S_2^{(0)}\right) \to \min$

$$\left(\left(x_1^{(0)''} + \lambda S_{2,1}^{(0)}\right) - 3\right)^2 + \left(\left(x_2^{(0)''} + \lambda S_{2,2}^{(0)}\right) - 5\right)^2 =$$

$$= (3 + \lambda \cdot 0 - 3)^2 + (5 + \lambda \cdot 1 - 5)^2 = \lambda^2 \to \min_\lambda;\ \lambda^* = 0;$$

$$x^{(0)'''} = x^{(0)''} + \lambda^* S_2^{(0)} = \begin{pmatrix} 3 \\ 5 \end{pmatrix} + 0\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 3 \\ 5 \end{pmatrix};\ f\left(x^{(0)}\right) = 0.$$

3. $$r = \frac{x^{(0)'''} - x^{(0)'}}{\left\| x^{(0)'''} - x^{(0)'} \right\|} = \frac{\begin{pmatrix} 3 \\ 5 \end{pmatrix} - \begin{pmatrix} 0 \\ 5 \end{pmatrix}}{\sqrt{(3-0)^2 + (5-5)^2}} = \frac{\begin{pmatrix} 3 \\ 0 \end{pmatrix}}{\sqrt{9}} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

4. $S_1^{(1)} = S_2^{(0)} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, $S_2^{(1)} = r = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$.

5. $f\left(x^{(1)} + \lambda S_2^{(1)}\right) \to \min$, где $x^{(1)} = x^{(0)'''}$;

$$\left(\left(x_1^{(1)} + \lambda S_{2,1}^{(0)}\right) - 3\right)^2 + \left(\left(x_2^{(1)} + \lambda S_{2,2}^{(1)}\right) - 5\right)^2 =$$

$$= (3 + \lambda \cdot 1 - 3)^2 + (5 + \lambda \cdot 0 - 5)^2 = \lambda^2 \to \min_\lambda;\ \lambda^* = 0;$$

$$x_1^{(1)'} = x^{(1)} + \lambda S_2^{(1)} = \begin{pmatrix} 3 \\ 5 \end{pmatrix} + 0\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 3 \\ 5 \end{pmatrix};\ f\left(x^{(1)'}\right) = 0.$$

Таким образом, для квадратичной функции $f = (x_1 - 3)^2 + (x_2 - 5)^2$ точка $x = (3,5)$ является оптимальной.

13.5. МЕТОДЫ СЛУЧАЙНОГО ПОИСКА

Основная идея методов случайного поиска заключается в том, чтобы перебором случайных совокупностей значений независимых переменных найти оптимум целевой функции или направление движения к нему.

Из всего множества методов случайного поиска рассмотрим далее только наиболее простые и распространённые.

13.5.1. Слепой поиск

При использовании этого метода в допустимой области изменения независимых переменных случайным образом выбирается точка, в которой вычисляется значение целевой функции. Далее аналогично выбирается другая точка и рассчитывается в ней значение функции, которое сравнивается с полученным ранее. Если значение целевой функции во второй точке оказывается лучше, то это значение запоминается. Затем выборка случайных точек продолжается, причём каждый раз значение целевой функции в новой точке сравнивается с последним наилучшим значением.

Подобная процедура продолжается достаточно большое число раз, что является отрицательной чертой данного метода (например, для вычисления экстремума с точностью 0,001 требуется ~1,5 млн. вычислений функции).

Этот метод может иметь некоторую модификацию, когда после некоторого числа вычислений область поиска сужается до некоторой окрестности вокруг наилучшей точки.

13.5.2. Метод случайных направлений

Алгоритм этого метода заключается в том, что из произвольной точки *N*-мерного пространства $x^{(k)}$, в которой целевая функция принимает значение $f\left(x^{(k)}\right)$, производится шаг в случайном направлении, определяемом случайным вектором $S^{(k)}$. Величина шага задаётся параметром λ. В результате находится новая точка $x^{(k+1)} = x^{(k)} + \lambda S^{(k)}$, в которой вычисляется значение целевой функции.

Определить компоненты случайного вектора $S^{(k)}$ с помощью последовательности равномерно распределённых случайных чисел R_j можно следующим образом: $S_j^{(k)} = \dfrac{R_j}{\sqrt{\sum_{i=1}^{N} R_i^2}}$, $j = \overline{1,\ n}$.

Если шаг $\lambda S^{(k)}$ оказался удачным и значение функции в точке $x^{(k+1)}$ лучше, чем в точке $x^{(k)}$, то эта точка $x^{(k+1)}$ принимается за новое приближение к экстремуму. В противном случае определяется новое случайное направление $S^{(k)}$ до тех пор, пока не будет найдена лучшая, чем точка $x^{(k+1)}$.

Поиск заканчивается, если после выполнения некоторой серии из *m* шагов (обычно *m* принимается равным *N*-размерности решаемой задачи) не удалось найти лучшего значения функции.

Эффективность метода может быть повышена, если после серии из m неудачных шагов уменьшить величину шага λ. После этого поиск продолжается до тех пор, пока шаг поиска λ не станет меньше заданной величины λ_{min}, принимаемой за точность определения оптимума.

13.5.3. Метод случайных направлений с обратным шагом

Этот метод является своеобразной модификацией метода случайных направлений, улучшающий его эффективность. Отличительной особенностью этого метода является то, что в случае неудачного шага $\lambda S^{(k)}$ из точки $x^{(k)}$ сразу же делается шаг в обратном направлении $-\lambda S^{(k)}$. Если же это не приносит положительного результата, то лишь тогда либо выбирают новое направление, либо уменьшают шаг λ.

13.5.4. Метод спуска с «наказанием случайностью»

Этот метод является своего рода аналогом метода покоординатного спуска с той лишь разницей, что направление спуска выбирается случайным образом. Таким образом, если выбранное случайное направление оказывается удачным, то в этом направлении шагают до тех пор, пока целевая функция улучшается. Как только улучшение функции прекращается, выбирается новое случайное направление и поиск продолжается.

13.6. ГРАДИЕНТНЫЕ МЕТОДЫ ПОИСКА

При решении задачи нелинейного программирования рассмотренными выше методами прямого поиска использовались только значения целевой функции. С одной стороны, это является преимуществом прямых методов, поскольку во многих практических инженерных задачах информация о значениях целевой функции является единственной надёжной информацией, которой располагает исследователь. С другой стороны, при использовании даже самых эффективных прямых методов для получения решения иногда требуется чрезвычайно большое количество вычислений значений функции. Это обстоятельство вынуждает использовать градиентные методы для решения задач нелинейного программирования. Все эти методы носят итерационный характер, так как компоненты градиента оказываются нелинейными функциями независимых переменных. Итерационная процедура градиентных методов реализуется в соответствии с формулой:

$$x^{(k+1)} = x^{(k)} + \lambda^{(k)} S^{(k)}, \qquad (13.3)$$

где $x^{(k)}$ – текущее приближение к решению x^*; $\lambda^{(k)}$ – параметр, характеризующий длину шага; $S^{(k)}$ – направление поиска в N-мерном пространстве независимых переменных x_i, $i = \overline{1, N}$. Способ определения $S^{(k)}$ и $\lambda^{(k)}$ на каждой итерации связан с особенностями применяемого метода.

При исследовании градиентных методов полагают, что целевая функция $f(x)$, её первые $f'(x)$ и вторые $f''(x)$ производные существуют и непрерывны.

13.6.1. Метод наискорейшего спуска

Предположим, что в некоторой точке x пространства независимых переменных требуется определить направление наискорейшего локального спуска, т.е. направление наибольшего локального уменьшения целевой функции. Эта задача была решена ещё известным французским математиком Коши, поэтому метод наискорейшего спуска иногда называют методом Коши.

Из курса высшей математики известно, что градиент скалярной функции в точке $x^{(k)}$ направлен в сторону наискорейшего увеличения функции, и что он ортогонален касательной к линии равного уровня функции $f(x)$, проходящей через точку $x^{(k)}$. Следовательно, при поиске минимума целевой функции $f(x)$ нужно двигаться в направлении, противоположном градиенту $f(x)$, т.е. в направлении *наискорейшего спуска*, поскольку отрицательный градиент функции $f(x)$ в точке $x^{(k)}$ направлен в сторону наибольшего уменьшения $f(x)$ по всем компонентам x и ортогонален линии уровня $f(x)$ в точке $x^{(k)}$. Таким образом, требуемое направление может быть определено по формуле $S^{(k)} = -\nabla f\left(x^{(k)}\right)$ или, если рассматривать нормированный (единичный) градиент, $S^{(k)} = -\dfrac{\nabla f\left(x^{(k)}\right)}{\left\|\nabla f\left(x^{(k)}\right)\right\|}$.

С учётом последнего выражения итерационная формула метода наискорейшего спуска может быть записана в виде

$$x^{(k+1)} = x^{(k)} - \lambda^{(k)} \frac{\nabla f\left(x^{(k)}\right)}{\left\|\nabla f\left(x^{(k)}\right)\right\|}, \qquad (13.4)$$

где $\lambda^{(k)}$ – заданный положительный параметр (шаг поиска). При выборе размера шага применяют два общих подхода.

В первом при переходе из точки $x^{(k)}$ в точку $x^{(k+1)}$ целевая функция минимизируется по λ с помощью любого метода одномерного поиска. В этом случае ищется минимум функции

$$f^*\left(x^{(k)} - \lambda^{(k)} \frac{\nabla f\left(x^{(k)}\right)}{\left\|\nabla f\left(x^{(k)}\right)\right\|}\right) = \min_{\lambda^{(k)} \geq 0} f\left(x^{(k)} - \lambda^{(k)} \frac{\nabla f\left(x^{(k)}\right)}{\left\|\nabla f\left(x^{(k)}\right)\right\|}\right).$$

Другой подход предполагает использование фиксированной величины $\lambda^{(k)}$, которую выбирают из условия монотонного убывания функции $f\left(x^{(k+1)}\right) < f\left(x^{(k)}\right)$.

Формулу (13.4) применяют до тех пор, пока не выполнится одно из условий:

$$\left\|x^{(k+1)} - x^{(k)}\right\| < \varepsilon_1 \text{ или } \left|f\left(x^{(k+1)}\right) - f\left(x^{(k)}\right)\right| < \varepsilon_2, \text{ или } \left|f'\left(x^{(k+1)}\right)\right| < \varepsilon_2, \quad (13.5)$$

или их комбинация.

13.6.2. Метод Ньютона

В методе наискорейшего спуска направление поиска можно проинтерпретировать как следствие линейной аппроксимации целевой функции в окрестности точки $x^{(k)}$ (при выводе формулы используется только линейный член ряда Тейлора). Однако движение в этом направлении быстро приведёт к решению лишь в том случае, если целевая функция является «правильной» (линии уровня – концентрические окружности). Увеличить скорость поиска экстремума можно, если использовать при определении направления поиска информацию, содержащуюся во вторых частных производных целевой функции $f(x)$ по независимым переменным. Выбор направления поиска, таким образом, соответствует квадратичной аппроксимации целевой функции в окрестности точки $x^{(k)}$. Одним из наиболее известных методов, использующих вторые производные целевой функции, является метод Ньютона.

$$x^{(k+1)} = x^{(k)} - \lambda^{(k)} \frac{\left[\nabla^2 f\left(x^{(k)}\right)\right]^{-1} \nabla f\left(x^{(k)}\right)}{\left\|\left[\nabla^2 f\left(x^{(k)}\right)\right]^{-1} \nabla f\left(x^{(k)}\right)\right\|}. \quad (13.6)$$

Алгоритм определения шага $\lambda^{(k)}$ и критерии окончания поиска в методе Ньютона те же, что и в методе наискорейшего спуска.

13.6.3. Метод сопряжённых градиентов

Рассмотренный выше метод наискорейшего спуска эффективен при поиске на значимых расстояниях от точки минимума x^* и плохо «работает» в окрестности этой точки. В то же время метод Ньютона не отличается высокой надёжностью при поиске x^* из удалённой точки, однако оказывается весьма эффективным в тех случаях, когда $x^{(k)}$ находится вблизи точки экстремума.

Положительные свойства методов Коши и Ньютона сочетают в себе методы сопряжённых градиентов, которые, с одной стороны, отличаются высокой надёжностью при поиске x^* из удалённой точки $x^{(k)}$ и, с другой стороны, быстро сходятся в окрестности точки мини-

мума. Одним из таких методов является *метод Флетчера-Ривса*, в котором направление поиска на *k*-м шаге является сопряжённым с направлением на $(k-1)$-м шаге и вычисляется через первые производные целевой функции.

Для метода сопряжённых градиентов Флетчера–Ривса итерационный процесс осуществляется по формуле (13.3), где

$$S^{(k)} = -\frac{\nabla f\left(x^{(k)}\right)}{\left\|\nabla f\left(x^{(k)}\right)\right\|} + \frac{\left\|\nabla f\left(x^{(k)}\right)\right\|^2}{\left\|\nabla f\left(x^{(k-1)}\right)\right\|^2} S^{(k-1)},\ k = 1, 2, ...,$$
$$S^{(0)} = -\frac{\nabla f\left(x^{(0)}\right)}{\left\|\nabla f\left(x^{(0)}\right)\right\|}. \qquad (13.7)$$

13.7. МЕТОД «ТЯЖЁЛОГО ШАРИКА»

Этот метод используется в задачах с целевыми функциями, имеющими несколько локальных экстремумов, т.е. для поиска глобального экстремума.

Уравнение, описывающее движение тела с конечной массой («тяжёлого шарика») в вязкой среде под действием силы $f(x)$, величина и направление которой зависят от местоположения тела в рассматриваемый момент времени, представляется в виде

$$m\frac{d^2x}{dt^2} + v\frac{dx}{dt} + \nabla f(x) = 0,$$

где m – масса тяжёлого шарика; v – вязкость среды. Заменив в этом уравнении производные конечными разностями

$$\left(\frac{d^2x}{dt^2} = \frac{x^{(k+1)} - 2x^{(k)} + x^{(k-1)}}{\Delta t^2},\ \frac{dx}{dt} = \frac{x^{(k+1)} - x^{(k)}}{\Delta t}\right),$$

получим, перейдя к нормированному значению градиента:

$$x^{(k+1)} = x^{(k)} - \lambda^{(k)}\frac{\nabla f\left(x^{(k)}\right)}{\left\|\nabla f\left(x^{(k)}\right)\right\|} + \alpha^{(k)}\left(x^{(k)} - x^{(k-1)}\right), \qquad (13.8)$$

где $\alpha^{(k)}$ – дополнительный рабочий шаг. На начальном этапе поиска полагается $\alpha^{(k)} = 0$. Когда же скорость поиска уменьшается, т.е. $\nabla f(x) \approx 0$, «включается» $0 \le \alpha^{(k)} \le 1$, которое учитывает «инерцию движения» и помогает проскочить не очень глубокие локальные экстремумы и плоские участки целевой функции $f(x)$. Обычно полагают $0{,}8 \le \alpha^{(k)} \le 1$.

13.8. МЕТОД «ШАГОВ ПО ОВРАГУ»

Очень часто целевые функции имеют «овраги». В этом случае большинство методов прекращают свою работу вовсе или оказываются чрезвычайно неэффективными. Дно «оврага» или вершина «гребня» характеризуется незначительным изменением функции при значительном изменении одной или нескольких независимых переменных.

Алгоритм метода шагов по «оврагу» заключается в следующем. Из некоторой начальной точки $x^{(01)}$ проводится поиск экстремума любым из ранее рассмотренных методов, который заканчивается в точке $x^{(1)}$.

На следующем этапе из начальной точки $x^{(01)}$ делается шаг в направлении наибольшего изменения переменных, несущественно влияющих на значение целевой функции. Получается точка $x^{(02)}$, из которой вновь выполняется экстремальный поиск, заканчивающийся в точке $x^{(2)}$. Направление $\left(x^{(1)}x^{(2)}\right)$ характеризует направление изменения функции по «оврагу». В этом направлении делается шаг, в результате которого получается точка $x^{(03)}$. Из неё снова производится спуск на дно «оврага» и находится критическая точка $x^{(3)}$ и т.д. (рис. 13.5).

Этот процесс продолжается до тех пор, пока значение функции в точке $x^{(k+1)}$ не оказывается хуже $f\left(x^{(k)}\right)$. Это позволяет сделать вывод, что экстремум находится между $x^{(k)}$ и $x^{(k+1)}$. Следовательно, поиск можно повторить на этом участке с меньшим значением шагов по «оврагу».

Разбиение независимых переменных по характеру их влияния на величину целевой функции производится либо перед началом поиска, либо во время его выполнения. Так, если в процессе поиска некоторые переменные изменились незначительно, то новое состояние $x^{(0i)}$ можно найти изменением именно этой группы переменных.

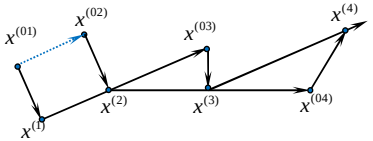


Рис. 13.5. Иллюстрация метода шагов по «оврагу»

14. ЧИСЛЕННЫЕ МЕТОДЫ УСЛОВНОЙ ОПТИМИЗАЦИИ

Решение большинства инженерных задач связано с оптимизацией при наличии некоторого количества ограничений на управляемые переменные. Такие ограничения существенно уменьшают размеры области, в которой проводится поиск оптимума. Но в то же время делают процесс оптимизации более сложным, так как рассмотренные ранее методы оптимизации нельзя использовать при наличии ограничений. При этом может нарушаться даже основное условие существования экстремума, в соответствие с которым оптимум должен достигаться в стационарной точке, характеризующейся нулевым градиентом.

В общем случае задача оптимизации может быть записана в виде

$$f = f(x_1, x_2, ..., x_N) \to \min_{x \in X} \tag{14.1}$$

при ограничениях на независимые переменные x_i $\left(i = \overline{1, N}\right)$ в форме равенств

$$h_j(x_1, x_2, ..., x_N) = 0, \quad j = \overline{1, J} \tag{14.2}$$

и(или) неравенств

$$g_k(x_1, x_2, ..., x_N) \geq 0, \quad k = \overline{1, K}. \tag{14.3}$$

14.1. МЕТОД МНОЖИТЕЛЕЙ ЛАГРАНЖА

Рассмотрим задачу оптимизации, записанную в виде (14.1), (14.2), т.е. содержащую несколько ограничений в виде равенств. Эта задача может быть в принципе решена как задача безусловной оптимизации, полученная путём исключения из целевой функции J независимых переменных с помощью заданных ограничений – равенств. Таким образом, уменьшается размерность исходной задачи с N до $N - J$. Например, из ограничений (14.2) можно выразить: $x_j = \tilde{h}_j(x_1, x_2, ..., x_{j-1}, x_{j+1}, ..., x_N)$, $j = \overline{1, J}$ и задача (14.1), (14.2) запишется в виде $f(x_1, x_2, ..., x_{N-J}) \to \min_{x \in X}$.

Однако метод исключения переменных применим лишь в тех случаях, когда уравнения, представляющие ограничения, можно разрешить относительного некоторого конкретного набора независимых переменных. Но при наличии большого числа ограничений в виде равенств или когда уравнение не удаётся разрешить относительно переменной, этот метод не может быть применим.

В таких ситуациях целесообразно использовать *метод множителей Лагранжа*. При этом задача с ограничениями (14.1), (14.2) преобразуется в эквивалентную задачу безусловной оптимизации некоторой функции Лагранжа, в которой фигурируют неизвестные параметры, называемые *множителями Лагранжа*. Функция Лагранжа имеет вид

$$L(x, \nu) = f(x) - \sum_{j=1}^{J} \nu_j h_j . \qquad (14.4)$$

Здесь ν_j – множители Лагранжа, значения которых требуется определить. На знаки ν_j никаких требований не накладывается. Тогда, если при некоторых ν_j^* достигается минимум функции $L(x, \nu)$ в точке x^*, в которой выполняются ограничения $h_j(x^*) = 0$, то точка x^* будет точкой минимума функции (14.1), так как $L(x^*, \nu^*) = f(x^*) = \min L(x, \nu)$.

Для нахождения решения задачи (14.4) приравняем частные производные функции $L(x, \nu)$ по x к нулю, получаем следующую систему N уравнений:

$$\begin{gathered} \frac{\partial L(x, \nu)}{\partial x_1} = 0 ; \\ \dots \\ \frac{\partial L(x, \nu)}{\partial x_N} = 0 . \end{gathered} \qquad (14.5)$$

Если удаётся найти решение системы (14.5) как функции от ν, т.е. $x^* = x(\nu)$, то затем, выбрав значения ν, при которых выполняются условия (14.2), можно найти окончательное решение задачи (14.1), (14.2).

Если подобный путь решения системы (14.5) оказывается затруднительным, то можно расширить систему путём включения в неё ограничений – равенств (14.2). Тогда решение расширенной системы, состоящей из $N + J$ уравнений с $N + J$ неизвестными x и ν, определяет стационарную точку функции L. Затем реализуется процедура проверки на минимум или максимум, которая проводится на основе вычисления элементов матрицы Гессе функции L, рассматриваемой как функции от x (если матрица положительно определена – минимум, в противном случае – максимум).

14.2. УСЛОВИЯ КУНА–ТАККЕРА

Метод множителей Лагранжа позволяет решить задачу оптимизации с ограничениями в виде равенств. Кун и Таккер обобщили этот подход на случай общей задачи нелинейного программирования с ограничениями как в виде равенств, так и в виде неравенств.

Рассмотрим задачу (14.1) – (14.3). Ограничения (14.3) в виде неравенства $g_k(x) \ge 0$ называются *активными* или *связывающими* в точке $\bar{x}$, если $g_k(\bar{x}) = 0$, и *неактивными* или *несвязывающими*, если $g_k(\bar{x}) > 0$.

Если существует возможность обнаружить ограничения, которые неактивны в точке оптимума, до непосредственного решения задачи, то эти ограничения можно исключить из модели и тем самым уменьшить её размеры.

Кун и Таккер построили необходимые и достаточные условия оптимальности для задач нелинейного программирования, исходя из предположения о дифференцируемости функций f, g_k и h_j. Эти условия оптимальности, называемые условиями Куна–Таккера, можно сформулировать в виде задачи нахождения решения некоторой системы нелинейных уравнений и неравенств: найти векторы x, u, v, удовлетворяющие условиям:

$$\begin{gathered}\nabla f(x)-\sum_{k=1}^{K} u_k \nabla g_k(x)-\sum_{j=1}^{J} \nu_j \nabla h_j(x)=0;\\ u_k g_k(x)=0,\ k=\overline{1,K};\\ h_j(x)=0,\ j=\overline{1,J};\\ \left.\begin{gathered} g_k(x)\geq 0;\\ u_k \geq 0. \end{gathered}\right\}\end{gathered} \tag{14.6}$$

Тогда необходимые и достаточные условия оптимальности решения задачи нелинейного программирования могут быть сформулированы следующим образом.

Теорема 1: Необходимость условия Куна–Таккера.

Пусть f, g и h – дифференцируемые функции в задаче (14.1) – (14.3), *а* x^* *– допустимое решение данной задачи. Положим* $\overline{K}=\left\{k \middle| g_k(x^*)=0\right\}$. *Пусть* $\nabla g_k(x^*)$ *при* $k\in\overline{K}$ *и* $\nabla h_j(x^*)$ *при* $j=\overline{1,J}$ *линейно независимы. Тогда если* x^* *– оптимальное решение задачи нелинейного программирования, то существует такая пара векторов* (u^*, ν^*), *что* (x^*, u^*, ν^*) *является решением задачи Куна–Таккера* (14.6).

Теорема 2: Достаточность условий Куна–Таккера.

Пусть в задаче (14.1) – (14.3) *целевая функция* $f(x)$ *выпуклая, все ограничения в виде неравенств содержат вогнутые функции* $g_k(x)$, $k=\overline{1,K}$, *а ограничения в виде равенств содержат линейные функции* $h_j(x)$, $j=\overline{1,J}$. *Тогда если существует решение* (x^*, u^*, ν^*), *удовлетворяющее условиям Куна–Таккера* (14.6), *то* x^* *– оптимальное решение задачи нелинейного программирования.*

В тех случаях, когда необходимые условия Куна–Таккера выполняются в нескольких точках, следует воспользоваться условиями Куна–Таккера 2-го порядка, которым должна удовлетворять точка локального оптимума.

14.3. МЕТОДЫ ШТРАФНЫХ ФУНКЦИЙ

В основе этих методов решения задачи (14.1) – (14.3) лежит построение конечной последовательности точек $x^{(t)}$, $t = 0, 1, ..., T$, которая начинается с заданной точки $x^{(0)}$ и заканчивается точкой $x^{(T)}$, дающей наилучшее приближение к x^* среди всех точек построенной последовательности. В качестве точек $x^{(t)}$, $t = 1, 2, ..., T$ берутся стационарные точки так называемой штрафной функции. С помощью штрафной функции исходная задача условной минимизации преобразуется в последовательность задач безусловной минимизации. Конкретные методы, основанные на указанной общей схеме, определяются видом штрафной функции, а также правилами, по которым производится пересчёт штрафных параметров по окончании очередного цикла безусловной минимизации.

Штрафная функция определяется выражением

$$P(x,\ R) = f(x) + \Omega(R,\ g(x),\ h(x)), \qquad (14.7)$$

где R – набор штрафных параметров, а так называемый штраф Ω является функцией R и функций, задающих ограничения.

Рассмотрим наиболее широко используемые типы штрафов.

1. Квадратичный штраф, используемый для ограничений – равенств $\Omega = R[h(x)]^2$.

При минимизации этот штраф препятствует отклонению величины $h(x)$ от нуля, а при увеличении R стационарная точка соответствующей штрафной функции $P(x, R)$ приближается к x^*.

2. Бесконечный барьер – штраф, используемый для ограничений – неравенств $\Omega = 10^{20} \sum_{k \in \overline{K}} |g_k(x)|$, где $\overline{K}$ – множество индексов нарушенных ограничений, т.е. $g_k < 0$ при $k \in \overline{K}$. В этом случае штраф приобретает бесконечно большие значения. Если же неравенство выполняется, то штраф равен нулю (рис. 14.1).

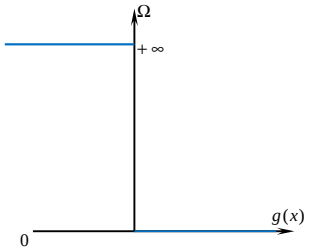


Рис. 14.1. Штраф «бесконечный барьер»

3. Логарифмический штраф $\Omega = -R\ln[g(x)]$.

Этот штраф положителен при всех *x*, таких, что $0 < g(x) < 1$ и отрицателен при $g(x) > 1$. Логарифмический штраф – это барьерная функция, не определенная в недопустимых точках $(g(x) < 0)$ (рис. 14.2).

Поэтому для недопустимых точек требуется специальная процедура, обеспечивающая возврат в допустимую область. Итерационный процесс начинается из допустимой начальной точки при положительном начальном значении $R(R = 10$ или $100)$, которое в процессе итераций уменьшается и в пределе стремится к нулю.

4. Обратная функция $\Omega = R\dfrac{1}{g(x)}$, которая не имеет отрицательных значений в допустимой области. Как и предыдущий штраф является барьером. В недопустимых точках штраф принимает отрицательные значения и поэтому требуется специальная процедура для возврата в допустимую область. В процессе вычислений значения *R* уменьшаются до нуля.

5. Квадрат срезки $\Omega = R\langle g(x)\rangle^2$, где

$$\langle g(x)\rangle = \begin{cases} g(x), & \text{если } g(x) \le 0; \\ 0, & \text{если } g(x) > 0. \end{cases}$$

Вычисления проводятся с положительным *R*, которое увеличивается от итерации к итерации.

В общем случае простейший алгоритм, в котором используется штрафная функция, может быть представлен в виде

Шаг 1: Задаются значения ε_1, ε_2, ε_3, $x^{(0)}$, $R^{(0)}$, где

ε_1 – параметр окончания одномерного поиска;

ε_2 – параметр окончания безусловной минимизации;

ε_3 – параметр окончания работы алгоритма;

$R^{(0)}$ – начальный вектор штрафных параметров.

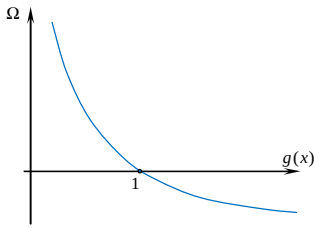


Рис. 14.2. Логарифмический штраф

Шаг 2: Построить $P(x,R)=f(x)+\Omega(R,g(x),h(x))$.

Шаг 3: Найти $x^{(t+1)}$, доставляющее минимум функции $P\left(x^{(t+1)},R^{(t)}\right)$ при фиксированном $R^{(t)}$. В качестве начальной точки используется $x^{(t)}$, а в качестве параметра окончания поиска ε_2 .

Шаг 4: Проверить условие $\left|P\left(x^{(t+1)},R^{(t)}\right)-P\left(x^{(t)},R^{(t)}\right)\right|\le\varepsilon_3$. Если оно выполняется, положить $x^{(t+1)}=x^{(T)}$ и закончить процесс решения; в противном случае перейти к следующему шагу.

Шаг 5: Положить $R^{(t+1)}=R^{(t)}+\Delta R^{(t)}$ в соответствии с используемым правилом пересчёта, после чего перейти к шагу 2.

В качестве примеров штрафных функций можно привести следующие:

$$\left.\begin{aligned}P(x,R)&=f(x)+R\sum_{k=1}^{K}\frac{1}{g_k(x)}+\frac{1}{R}\sum_{j=1}^{J}\left[h_j(x)\right]^2;\\ P(x,R)&=f(x)-R\sum_{k=1}^{K}\ln\left[g_k(x)\right]+\frac{1}{R}\sum_{j=1}^{J}\left[h_j(x)\right]^2.\end{aligned}\right\}$$

$$P(x,R)=f(x)+10^{20}\sum_{k=1}^{K}\left|\left\langle g_k(x)\right\rangle\right|+10^{20}\sum_{j=1}^{J}\left|h_j(x)\right|,$$

где R монотонно убывает и стремится к 0.

ЗАКЛЮЧЕНИЕ

В деятельности инженера ничто, пожалуй, не вызывает такого удовлетворения, как успешное решение задач, возникающих в процессе проектирования. Действительно, между итерационными методами решения задач проектирования и природой творчества инженера много общего. Инженер всегда стремился ускорить и автоматизировать механические операции, чтобы высвободить время для творчества. Поэтому естественно применение при решении сложных проектных задач средств вычислительной техники. Однако применение вычислительной техники в инженерной деятельности связано с дополнительной ответственностью – при неумелом обращении с новой техникой могут быть допущены ошибки, снижающие эффективность полученных решений. Только грамотное использование вычислительной техники, правильный выбор методов и алгоритмов решения инженерных задач позволит специалисту быстро найти верное решение с наименьшими временными и трудовыми затратами. Именно на это и нацелено данное учебное пособие. Изучение рассмотренных в нём методов позволит будущим инженерам быстро находить эффективное проектное решение в большинстве производственных ситуаций.

СПИСОК ЛИТЕРАТУРЫ

1. Бахвалов, Н.С. Численные методы / Н.С. Бахвалов. – М. : Наука, 1975. – 632 с.
2. Копченова, Н.В. Вычислительная математика в примерах и задачах / Н.В. Копченова, И.А. Марон.– М. : Наука, 1972. – 386 с.
3. Демидович, Б.П. Основы вычислительной математики / Б.П. Демидович, И.А. Марон. – М. : Наука, 1966. – 664 с.
4. Самарский, А.А. Численные методы / А.А. Самарский, А.В. Гулин. – М. : Наука, 1989. – 432 с.
5. Бояринов, А.И. Методы оптимизации в химической технологии / А.И. Бояринов, В.В. Кафаров. – М. : Химия, 1975. – 576 с.
6. Крылов, В.И. Вычислительные методы / В.И. Крылов, В.В. Бобков, П.И. Монастырский. – М. : Наука, 1976. – Т. 1. – 304 с.
7. Крылов, В.И. Вычислительные методы / В.И. Крылов, В.В. Бобков, П.И. Монастырский. – М. : Наука, 1977. – Т. 2. – 400 с.
8. Березин, И.С. Методы вычислений. / И.С. Березин, Н.П. Жидков. – М. : Наука, 1966. – Т. 1, 2.
9. Волков, Е.А. Численные методы / Е.А. Волков. – М. : Наука, 1987. – 248 с.
10. Бахвалов, Н.С. Численные методы / Н.С. Бахвалов, Н.П. Жидков, Г.М. Кобельков. – М. : Наука, 1987. – 600 с.
11. Самарский, А.А. Введение в численные методы / А.А. Самарский. – М. : Наука, 1987.
12. Шуп, Т. Решение инженерных задач на ЭВМ / Т. Шуп. – М. : Мир, 1982. – 238 с.
13. Банди, Б. Методы оптимизации. Вводный курс / Б. Банди. – М. : Радио и связь, 1988. – 128 с.
14. Химмельблау, Д. Прикладное нелинейное программирование / Д. Химмельблау. – М. : Мир, 1975. – 536 с.
15. Реклейтис, Г. Оптимизация в технике : в 2 кн. / Г. Реклейтис, А. Рейвиндран, К. Рэгсдел. – М. : Мир, 1986.
16. Сухарев, А.Г. Курс методов оптимизации / А.Г. Сухарев, А.В. Тимохов, В.В. Федоров. – М. : Наука, 1986. – 328 с.
17. Поляк, Б.Т. Введение в оптимизацию / Б.Т. Поляк. – М. : Наука, 1983. – 384 с.
18. Численные методы условной оптимизации / под ред. Ф. Гилла, У. Мюррея. – М. : Мир, 1977. – 290 с.
19. Демидович, Б.П. Численные методы анализа / Б.П. Демидович, И.А. Марон, Э.З. Шувалова. – М. : Наука, 1967. – 368 с.

СОДЕРЖАНИЕ

Учебное издание

МАЙСТРЕНКО Александр Владимирович,
МАЙСТРЕНКО Наталья Владимировна

ЧИСЛЕННЫЕ МЕТОДЫ РАСЧЁТА, МОДЕЛИРОВАНИЯ И ПРОЕКТИРОВАНИЯ ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ И ОБОРУДОВАНИЯ

Учебное пособие

Редактор Л.В. К о м б а р о в а
Инженер по компьютерному макетированию М.С. А н у р ь е в а

Подписано в печать 07.12.2011.
Формат 60×84 / 16. 8,37 усл. печ. л. Тираж 100 экз. Заказ № 558

Издательско-полиграфический центр ФГБОУ ВПО «ТГТУ»
392000, г. Тамбов, ул. Советская, д. 106, к. 14