Никифоров А. А.
Байбурин В. Б.

Математическое обеспечение информационно-измерительных и управляющих систем

Учебное пособие

Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Саратовский государственный технический университет
имени Ю. А. Гагарина»

А. А. Никифоров, В. Б. Байбурин

МАТЕМАТИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННО-ИЗМЕРИТЕЛЬНЫХ И УПРАВЛЯЮЩИХ СИСТЕМ

Учебное пособие

Электронное издание
локального распространения

Санкт-Петербург
Наукоемкие технологии
2025

ISBN 978-5-907946-46-0

УДК 519.7:681.51(075.8)
ББК 22.18я73
Н62

Рецензент:
Дмитрий Александрович Зимняков, доктор физико-математических наук, профессор кафедры медицинской физики Саратовского национального исследовательского государственного университета имени Н. Г. Чернышевского

Н62 Никифоров А. А., Байбурин В. Б. Математическое обеспечение информационно-измерительных и управляющих систем [Электронный ресурс]: учебное пособие / А. А. Никифоров, В. Б. Байбурин. – Электрон, текстовые дан. (2,2 Мб). – СПб.: Наукоемкие технологии, 2025. – 106 с. – 1 электрон., опт. диск (CD-ROM).

ISBN 978-5-907946-46-0

Учебное пособие представляет собой комплексное руководство по созданию и использованию математических методов и алгоритмов, необходимых для эффективного функционирования современных автоматизированных систем управления и измерения. В книге детально освещены ключевые аспекты математического обеспечения, начиная от базовых понятий и заканчивая передовыми методами обработки данных, моделирования и оптимизации.

Учебное пособие предназначено для бакалавров и магистрантов обучающихся по направлениям подготовки 11.03.01 Радиотехника (бакалавриат), 11.03.02 Инфокоммуникационные технологии и системы связи (бакалавриат), 11.04.02 Инфокоммуникационные технологии и системы связи (магистратура) по дисциплинам «Системы искусственного интеллекта», «Основы научных исследований», «Синтез технических систем», «Вычислительная техника и информационные технологии», «Теория систем и системный анализ», «Теоретико-множественный и теоретико-информационный анализ сложных систем», «Программное и математическое обеспечение систем анализа, управления и обработки информации».

Текстовое электронное издание
Минимальные системные требования:

- процессор: Intel x86, x64, AMD x86, x64 не менее 1 ГГц;
- оперативная память RAM ОЗУ: не менее 512 МБайт;
- свободное место на жестком диске (HDD): не менее 120 МБайт;
- операционная система: Windows XP и выше;
- Adobe Acrobat Reader;
- дисковод CD-ROM;
- мышь.

УДК 519.7:681.51(075.8)
ББК 22.18я73

ISBN 978-5-907946-46-0

Учебное издание

Никифоров Александр Анатольевич
Байбурин Вил Бариевич

Математическое обеспечение информационно-измерительных и управляющих систем

Учебное пособие

Электронное издание
локального распространения

Издательство «Наукоемкие технологии»
ООО «Корпорация «Интел Групп»
https://publishing.intelgr.com
E-mail: publishing@intelgr.com
Тел.: +7 (812) 945-50-63
Интернет-магазин издательства
https://shop.intelgr.com/

Подписано к использованию 26.03.2025 г.
Объем издания – 2,2 Мб.
Комплектация издания – 1 CD.
Тираж 500 CD.

ISBN 978-5-907946-46-0

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Современный мир невозможно представить без сложных информационных и управляющих систем, обеспечивающих функционирование различных технологических процессов, от производства до научных исследований. Эффективность работы таких систем во многом зависит от качества их математического обеспечения, которое играет ключевую роль в обработке данных, моделировании процессов и принятии решений. Настоящее учебное пособие направлено на изучение методов и алгоритмов вычислительной математики, применяемых для разработки математического обеспечения информационно-измерительных и управляющих систем.

Вычислительная математика – основная компьютерная дисциплина при решении научных и инженерных задач (фундаментальных и прикладных) в самых различных областях науки и техники: физике, электронике, механике, экономике, биологии, медицине, социологии и др.

Ряд методов вычислительной математики были предложены столетия назад Ньютоном, Эйлером Гауссом и др., однако в то время они не получили развития, т. к. не было технических вычислительных средств.

Интенсивное развитие и применение вычислительной математики связано с появлением электронно-вычислительной техники.

Основные особенности вычислительной математики

1. Вычислительная математика в отличие от традиционных математических методов имеет дело не с непрерывными величинами независимых переменных и соответствующих функций, а с их дискретными значениями.

2. Использование одного и того же метода вычислительной математики может дать различные итоговые результаты. Обычно это связано с параметрами дискретизации (их конкретными значениями).

3. Применение методов вычислительной математики, требующих как правило, в большинстве случаев большого объёма вычислений, невозможно без применения средств вычислительной техники.

4. С развитием методов вычислительной математики сформировалось современное понятие математической модели как абстрактного объекта, описывающего различные реальные процессы, механизмы, явления, устройства.

В данном учебном пособии будут рассмотрены наиболее распространенные методы вычислительной математики, хорошо зарекомендовавшие себя и при решении фундаментальных и прикладных задач.

Первая глава посвящена численным методам решения систем линейных уравнений: прямым (Гаусса, Крамера, прогонки) и итерационным (метод Гаусса – Зейделя).

Вторая глава посвящена методам решения нелинейных уравнений и систем методами Хорд, Ньютона, методом деления отрезка пополам.

В третьей главе рассмотрены методы получения аппроксимирующих функций: методы Лагранжа, Ньютона, Метод наименьших квадратов, Чебышева.

В четвертой главе рассмотрено численное интегрирование методом трапеций, прямоугольников, Симпсона и методы интегрирования несобственных интегралов, в частности метод Монте-Карло.

В пятой главе рассмотрены методы численного дифференцирования одномерных и многомерных функций.

Шестая глава посвящена методам оптимизации, численным и аналитическим методам одномерных и многомерных функций: методу полного перебора, методу Лагранжа, Симплекс-методу и другим.

Изложенные численные методы, помимо самостоятельного интереса, являются, по существу, основой для численного решения основных дифференциальных уравнений математической физики: обыкновенных и в частных производных.

Пособие предназначено для студентов технических специальностей, изучающих курсы, связанные с автоматизацией производственных процессов, информационными технологиями и системами управления. Оно также будет полезно специалистам, занимающимся разработкой и внедрением автоматизированных систем в различных отраслях промышленности и науки.

Материал изложен таким образом, чтобы обеспечить последовательное освоение теоретических знаний и практических навыков, необходимых для успешного выполнения инженерных задач в области математического обеспечения информационно-измерительных и управляющих систем.

1. СИСТЕМЫ ЛИНЕЙНЫХ УРАВНЕНИЙ

1.1. Общие сведения

К решению системы линейных уравнений сводятся многочисленные практические задачи, поэтому решение линейных систем – одна из самых распространенных и важных задач вычислительной математики.

Система n уравнений (линейных алгебраических) с n неизвестными имеет вид

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1N}x_N &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2N}x_N &= b_2 \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{NN}x_N &= b_N \end{aligned} \qquad (1.1)$$

Совокупность коэффициентов этой системы можно записать в виде квадратной матрицы порядка n

$$\mathrm{A} = \begin{bmatrix} a_{11}a_{12} \ldots . & a_{1N} \\ a_{21}a_{12} \ldots . & a_{2N} \\ \ldots \ldots \ldots . . & \ldots \ldots . \\ a_{N1}a_{N2} \ldots . & a_{NN} \end{bmatrix} \qquad (1.2)$$

Если матрица содержит m-строк и n-столбцов, то она называется прямоугольной матрицей.

Систему (1.1) можно записать в матричной форме:

$$AX = B \qquad (1.3)$$

где $X = \begin{bmatrix} x_1 \\ x_2 \\ .. \\ x_N \end{bmatrix}$ – вектор-столбец неизвестных; $B = \begin{bmatrix} b_1 \\ b_2 \\ .. \\ b_N \end{bmatrix}$ – вектор-столбец правых частей.

В ряде случаев получаются системы уравнений с некоторыми специальными видами матриц. Некоторые примеры:

$$A = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 3 & 2 \\ 1 & 2 & 4 \end{bmatrix} \quad (1.4)$$

симметрическая матрица (элементы симметричны относительно главной диагонали);

$$B = \begin{bmatrix} 1 & 2 & 3 \\ 0 & -1 & 1 \\ 0 & 0 & 2 \end{bmatrix} \quad (1.5)$$

верхняя треугольная матрица (с нулевыми элементами ниже диагонали);

$$\mathrm{C} = \begin{bmatrix} 1 & 2 & 1 & 0 & 0 & 0 \\ 2 & 1 & -2 & 0 & 0 & 0 \\ 3 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 1 & 1 \\ 0 & 0 & 0 & -1 & 4 & -1 \\ 0 & 0 & 0 & 2 & 1 & 1 \end{bmatrix} \quad (1.6)$$

клеточная матрица (ненулевые элементы составляют отдельные группы (клетки);

$$D = \begin{bmatrix} 1 & 2 & 0 & 0 & 0 & 0 \\ 1 & 2 & 1 & 0 & 0 & 0 \\ 0 & 1 & 3 & 2 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 3 & 2 \end{bmatrix} \quad (1.7)$$

Ленточная матрица (ненулевые элементы составляют «ленту» параллельно диагонали). В данном случае трехдиагональная матрица;

$$E = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.8)$$

единичная матрица;

$$\mathrm{F} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (1.9)$$

нулевая матрица (все элементы – нули).

Определителем (*детерминантом*) матрицы *A* *n*-го порядка называется число *D* (*det A*), равное

$$\mathrm{D} = \sum^{n!} (-1)^k \, a_{1\alpha} a_{2\beta} \ldots a_{N\omega}, \quad (1.10)$$

где индексы *α, β, …, ω* – приобретают все возможные *n*! перестановок номеров *1, 2, …, N*; *k – число инверсий в данной перестановке*, то есть число случаев, когда меньший номер идет после большего.

Необходимым и достаточным условием существования единственного решения систем линейных уравнений является условие

$D \neq 0$. Если $D = 0$, то матрица называется вырожденной. В этом случае система либо не имеет решения, либо имеет бесконечное множество решений. Эти случаи легко показать геометрически для системы

$$\begin{aligned} a_1 x_1 + b_1 x_2 &= c_1 \\ a_2 x_1 + b_2 x_2 &= c_2 \end{aligned} \quad (1.11)$$

Каждому уравнению соответствует прямая. Координаты точек пересечения есть решение системы.

Рассмотрим три возможных случая взаимного расположения прямых на плоскости:

1. Прямые пересекаются – это значит коэффициенты системы непропорциональны $\frac{a_1}{a_2} \neq \frac{b_1}{b_2}$ и определитель $D = \begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \end{bmatrix} \neq 0$ – система имеет единственное решение;

2. Прямые параллельны – коэффициенты системы подчиняются условиям

$$\frac{a_1}{a_2} = \frac{b_1}{b_2} \neq \frac{c_1}{c_2} \tag{1.12}$$

$D = 0$ – решение отсутствует;

3. Прямые совпадают (все коэффициенты пропорциональны):

$$\frac{a_1}{a_2} = \frac{b_1}{b_2} = \frac{c_1}{c_2} \tag{1.13}$$

$D = 0$ – бесконечное множество решений.

На практике, особенно при вычислении на ЭВМ (когда происходит округление, или отбрасывание младших результатов), определитель может быть не равен нулю: $D \neq 0$.

При $D \approx 0$ прямые могут оказаться почти параллельными. Координаты точки пересечения этих прямых очень чувствительны к изменению коэффициентов системы.

Поэтому малые погрешности вычислений или исходных данных могут привести к существенным погрешностям в решении. Такие системы уравнений называются плохо обусловленными.

Условие $D \approx 0$ – это необходимое условие плохой обусловленности, но не достаточное.

Пример: система уравнений n-го порядка с диагональной матрицей (так как только по диагонали ненулевые элементы) с элементами $a_{ii} = 0{,}1$ – не является плохо обусловленной, хотя ее определитель ($D = 10^{-n}$) и близок к нулю.

1.2. Методы решения линейных систем

Методы решения линейных систем можно разделить на две группы: *прямые* и *итерационные*.

Прямые методы используют конкретные соотношения (формулы) для вычисления неизвестных. Они просты и универсальны, пригодны для решения широкого класса линейных систем.

Однако имеют недостатки:

1. Требуют хранения в оперативной памяти сразу всей матрицы, что при больших n требует много памяти.
2. Они не учитывают, что могут быть разреженные матрицы с большим числом нулевых элементов, которые тоже занимают место в памяти.
3. Накапливание погрешностей в процессе решения, поскольку на любом этапе вычисления используется результат предыдущих операций.

Это очень опасно при большом N, то есть возрастет число операций, а также для плохо обусловленных систем, весьма чувствительных к погрешностям. Поэтому прямые методы

используются при $N < 200$ для систем с плотно заполненной матрицей и не близким к нулю определителем.

Иногда прямые методы называют точными, поскольку решения выражаются в виде точных формул. Однако точное решение может быть получено лишь при вычислениях с большим, а вернее с бесконечным числом разрядов. Однако разрядность всегда ограничена, поэтому неизбежны погрешности.

Итерационные методы – методы последовательных приближений. Вначале задается некоторое приближенное решение – так называемое *начальное* приближение. После этого с помощью некоторого алгоритма находят *новое* приближение.

Проводится один цикл вычислений, называемый *итерацией.* И так неоднократно до получения решения с заданной точностью. Алгоритм итераций обычно более сложен, чем в обычных методах. Объем вычислений заранее предвидеть трудно.

Итерационные модели в ряде случаев предпочтительнее. Они требуют хранения в памяти не всей матрицы, а лишь нескольких векторов с n компонентами, найденные элементы матрицы можно совсем не хранить, а вычислять их по мере необходимости. *Погрешность здесь не накапливается*, поскольку определяется лишь предыдущей итерацией и *практически* не зависит от ранее выполненных вычислений. Сходимость может быть медленной – поэтому ищутся эффективные пути ускорения.

Итерационные методы – могут использоваться для уточнения решений, полученных прямыми методами, то есть получаются смешанные методы, которые довольно эффективны, особенно для плохо обусловленных систем.

1.3. Прямые методы

1.3.1. Правило Крамера

Правило Крамера – неизвестные представляются в виде отношения определителей.

Пример:

$$\begin{aligned} a_1 x_1 + b_1 x_2 &= c_1 \\ a_2 x_1 + b_2 x_2 &= c_2 \end{aligned} \qquad (1.14)$$

Тогда $x_1 = {}^{D_1}/_{D}$ и $x_2 = {}^{D_2}/_{D}$,

где $D = \begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \end{bmatrix}$; $D_1 = \begin{bmatrix} c_1 & b_1 \\ c_2 & b_2 \end{bmatrix}$; $D_2 = \begin{bmatrix} a_1 & c_1 \\ a_2 & c_2 \end{bmatrix}$.

При большом числе уравнений нужно выполнить огромное количество операций. Чтобы вычислить один определитель, необходимо выполнить число операций $K \approx n*n!$, где n – число неизвестных переменных $x_1, x_2, \dots, x_N$

В качестве примера оценим значение K в зависимости от n.

n	3	10	20
K	17	$3{,}6*10^7$	$5*10^{19}$

Пусть компьютер имеет скорость вычислений $1*10^6$/с, тогда при $n = 20$ время вычислений составит $t_c = \frac{5*10^{19}}{10^6} = 5*10^{13}$с,

в 1 часе – 3600 сек., то есть $t_{\text{час}} = \frac{5*10^{13}}{3{,}6*10^{2}}$ ч$\sim 1{,}5 * 10^{10}$ч; $t_{\text{сут}} = \frac{1{,}5* 10^{10}\text{час}}{24} \sim 1{,}5*10^{8}$ сут; $t_{\text{год}} = \frac{5*10^{19}}{10^{6}*3600*24*360} \sim 1{,}5$ млн лет.

Поэтому правило Крамера можно использовать для решения систем, состоящих всего из нескольких уравнений.

Метод обратной матрицы – система записывается в виде $AX=B$. Умножаем обе части на обратную матрицу A^{-1}; $X=A^{-1}B$. Однако если не использовать специальные методы для вычисления обратной матрицы, то этот метод при больших n также практически непригоден.

1.3.2. Метод Гаусса (метод исключения)

Метод Гаусса (метод исключения) – наиболее распространенный метод. Мы рассмотрим применение этого метода для решения системы линейных уравнений, вычисления определителя, вычисления обратной матрицы.

Метод основан на приведении матрицы *системы* к треугольному виду. *Основная идея алгоритма* заключается в том, что на первом этапе с помощью первого уравнения исключается x_1 из всех последующих уравнений, на втором этапе с помощью второго уравнения исключается переменная x_2, из всех последующих и так далее – до тех пор, пока в левой части последнего уравнения не останется лишь один член с последним неизвестным x_n. – эти этапы реализуют *прямой ход* метода Гаусса.

Обратный ход – заключается в последовательном определении $x_N \dots x_1$, начиная с x_N. В последнем случае могут использоваться методы регуляризации.

Рассмотрим для случая трех уравнений:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 &= b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 &= b_3 \end{aligned} \qquad (1.15)$$

Умножим первое уравнение на $(-\frac{a_{21}}{a_{11}})$:

$$-\frac{a_{11}a_{21}x_1}{a_{11}} - \frac{a_{21}}{a_{11}}a_{12}x_2 - \frac{a_{21}}{a_{11}}a_{13}x_3 = -\frac{a_{21}}{a_{11}}b_1 \qquad (1.16)$$

и прибавим его ко 2-му уравнению, получим

$$\left(a_{22} - \frac{a_{21}}{a_{11}}a_{12}\right)x_2 + \left(a_{23} - \frac{a_{21}}{a_{11}}a_{13}\right)x_3 = b_2 - \frac{a_{21}}{a_{11}}b_1 \qquad (1.17)$$

Пере обозначим: $a'_{22} = a_{22} - \frac{a_{21}}{a_{11}}a_{12}$; $a'_{23} = a_{23} - \frac{a_{21}}{a_{11}}a_{13}$; $b'_2 = b_2 - \frac{a_{21}}{a_{11}}b_1$.

Получим

$$\begin{aligned} a'_{22}x_2 + a'_{23}x_3 &= b'_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 &= b_3 \end{aligned} \qquad (1.18)$$

Теперь умножим первое уравнение на $(-\frac{a_{31}}{a_{11}})$, получим

$$a_{31}x_1 - \frac{a_{31}}{a_{11}}a_{12}x_2 - \frac{a_{31}}{a_{11}}a_{13}x_3 = -\frac{a_{31}}{a_{11}}b_1 \qquad (1.19)$$

и сложим с третьим уравнением. Получим:

$$\left(a_{32} - \frac{a_{31}}{a_{11}}a_{12}\right)x_2 + \left(a_{33} - \frac{a_{31}}{a_{11}}a_{13}\right)x_3 = b_3 - \frac{a_{31}}{a_{11}}b_1 \qquad (1.20)$$

и после переобозначений

$$a'_{32} = a_{32} - \frac{a_{31}}{a_{11}} a_{12}; a'_{33} = a_{33} - \frac{a_{31}}{a_{11}} a_{13};\ b'_3 = b_3 - \frac{a_{31}}{a_{11}} b_1 \quad (1.21)$$

получим: $a'_{32}x_2 + a'_{33}x_3 = b'_3$.

$$\text{Т. е.} \begin{cases} a_{ij} = a_{ij} - \frac{a_{i1}}{a_{11}} a_{1j}\ , i,j = 2,3 \\ b_i = b_i - \frac{a_{i1}}{a_{11}} b_1\ , i = 2,3 \end{cases} \quad (1.22)$$

Таким образом, система уравнений примет вид:

$$\begin{gathered} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 \\ a'_{22}x_2 + a'_{23}x_3 = b'_2 \\ a'_{32}x_2 + a'_{33}x_3 = b'_3 \end{gathered} \quad (1.23)$$

Теперь реализуем II-й этап и из 3-го уравнения – исключим x_2 по той же методике. Для этого умножим 2-е уравнение на $(-\frac{a'_{32}}{a'_{22}})$ и прибавим к третьему, получим

$$\begin{gathered} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 \\ a'_{22}x_2 + a'_{23}x_3 = b'_2 \\ a''_{33}x_3 = b''_3 \end{gathered} \quad (1.24)$$

где $a''_{33} = a'_{23} - \frac{a'_{32}}{a'_{22}} a'_{23}$; $b''_3 = b'_3 - \frac{a'_{32}}{a'_{22}} b'_2$.

На этом заканчивается прямой ход. Матрица этой системы имеет треугольный вид. Мы видим, что в процессе исключения переменных приходится делить на коэффициенты a_{11}, a_{22} и т.д. Поэтому они должны быть отличны от нуля. Для этого необходимо предусматривать в вычислительном алгоритме перестановку уравнений системы, если будут нулевые координаты.

Обратный ход начинается с решения третьего уравнения, далее находим x_2 из второго уравнения и x_1:

$$x_3 = \frac{b_3''}{a_{33}''}$$

$$x_2 = \frac{1}{a_{22}'}(b_2' - a_{23}' x_3); \qquad (1.25)$$

$$x_1 = \frac{1}{a_{11}}(b_1 - a_{12}x_2 - a_{13}x_3).$$

Аналогично строится алгоритм для линейной системы с производным числом уравнений.

Метод Гаусса целесообразно использовать для решения систем с плотно заполненной матрицей. Все элементы матрицы и правые части уравнений находятся в оперативной памяти машины. Число арифметических операций примерно равно $(\frac{2}{3})n^3$.

1.4. Обсуждение погрешностей

Вычисленное по методу Гаусса решение X^*, отличается от точного решения матричного уравнения $X=A^{-1}B$ из-за погрешностей округления. Существуют две величины, характеризующие степень отклонения точного решения от приближенного: $\varepsilon = X - X^*$, r – равна разности между правой и левой частями уравнений при подстановке в них решения:

$r = B - AX^*$ – называется *невязкой*.

При $\varepsilon \sim 0$ обычно $r \sim 0$, но обратное справедливо не всегда, в частности для плохо обусловленных систем.

Если система не является плохо обусловленной, то в практических расчетах контроль точности решения осуществляется с помощью *невязки.*

1.5. Метод прогонки (модификация прямого метода Гаусса)

Метод прогонки применяется в случае трехдиагональной системы уравнений, в которой каждое уравнение содержит не более трех неизвестных. В общем виде трехдиагональную систему уравнений можно записать в виде:

$$\begin{gathered}
b_1x_1 + c_1x_2 + 0x_3 + 0x_4 + \ldots + 0x_N = d_1 \\
a_2x_1 + b_2x_2 + c_2x_3 + 0x_4 + \ldots + 0x_N = d_2 \\
0x_1 + a_3x_2 + b_3x_3 + c_3x_4 + 0x_5 + \ldots + 0x_N = d_3 \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
0x_1 + 0x_2 + 0x_3 + \ldots + a_{N-i}x_{N-2} + b_{N-1}x_{N-1} \\
+ c_{N-1}x_N = d_{N-1} \\
0x_1 + 0x_2 + 0x_3 + \cdots + a_Nx_{N-1} + x_Nb_N = d_N
\end{gathered} \qquad (1.26)$$

Введем соотношение $x_i = A_ix_{i+1} + B_i$, где $i = 1, 2, \ldots, N\text{-}1$, которое позволяет при известных значениях A_i и B_i определить меньшее по индексу i переменное x_i, по известному x_{i+1}.

Реализуем прямой ход, т.е. определим все A_i и B_i. Для этого выразим из первого уравнения x_1:

$$x_1 = -\frac{c_1}{b_1}x_2 + \frac{d_1}{b_1}. \qquad (1.27)$$

Обозначим $A_1 = -\frac{c_1}{b_1}$; $B_1 = \frac{d_1}{b_1}$; т.е. получим $x_1 = A_1x_2 + B_1$;

Подставим это выражения для x_1во второе уравнение:

$$a_2(A_1x_2 + B_1) + \ b_2x_2 + \ c_2x_3 = d_2 \tag{1.28}$$

Отсюда

$$x_2(A_1a_2 + b_2) = d_2 - a_2B_1 - c_2x_3 \tag{1.29}$$

или

$$\begin{gathered} x_2 = \frac{-\mathrm{c}_2}{(a_2A_1 + b_2)}x_3 + \frac{d_2 - a_2B_1}{(a_2A_1 + b_2)} \\ A_2 = \frac{-\mathrm{c}_2}{(a_2A_1 + b_2)};\ B_2 = \frac{d_2 - a_2B_1}{(a_2A_1 + b_2)} \end{gathered} \tag{1.30}$$

По аналогии можем записать

$$\begin{gathered} A_{N-1} = \frac{-\mathrm{c}_{N-1}}{(a_{N-1}A_{N-1} + b_{N-1})} \\ B_{N-1} = \frac{d_{N-1} - a_{N-1}B_{N-2}}{(a_{N-1}A_{N-2} + b_{N-1})} \end{gathered} \tag{1.31}$$

и далее

$$x_{N-1} = A_{N-1}x_N + B_{N-1} \tag{1.32}$$

К этому соотношению добавим еще одно вытекающее из последнего уравнения нашей системы уравнений:

$$x_{N-1} = \frac{d_N}{a_N} - b_Nx_N \tag{1.33}$$

Таким образом, мы получаем систему из двух уравнений:

$$x_{N-1} = A_{N-1}x_N + B_{N-1} \tag{1.34}$$

$$x_{N-1} = \frac{d_N}{a_N} - b_Nx_N \tag{1.35}$$

решая которую, определим x_N и x_{N-1}:

$$x_N = \frac{d_N - a_N B_{N-1}}{b_N + a_N A_{N-1}} \tag{1.36}$$

Теперь реализуем обратный ход

$$x_{N-2} = A_{N-2} x_{N-1} + B_{N-2} \tag{1.37}$$

И так далее, последовательно определяя все неизвестные вплоть до x_2 и x_1. Это очень быстрый метод с числом операций $N \approx 5n$. При этом должны выполняться условие:

$$|b_i| \geq |a_i| + |c_i| \tag{1.38}$$

Тогда система имеет единственное решение, и оно является устойчивым.

1.6. Итерационные методы

Часто применяются, чтобы уточнить решение, полученное с помощью прямого метода (погрешности возникают из-за округления и иногда могут быть значительными).

Рассмотрим один из методов, позволяющий уточнять решение, полученное прямым методом.

Пусть имеем систему линейных уравнений:

$$\begin{gathered} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1N}x_N = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2N}x_N = b_2 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{NN}x_N = b_N \end{gathered} \tag{1.39}$$

Пусть с помощью некоторого прямого метода вычислили приближенные значения неизвестных: $x_1^{(0)}$, $x_2^{(0)}$, …. $x_N^{(0)}$. Если мы

их подставим в левые части (1.39), получим справа некоторые значения $b_i^{(0)}$, отличные от b_i(*i* = *1, 2, ..., N*):

$$
\begin{aligned}
&a_{11}x_1^{(0)} + a_{12}x_2^{(0)} + \cdots + a_{1N}x_N^{(0)} = b_1^{(0)} \\
&a_{21}x_1^{(0)} + a_{22}x_2^{(0)} + \cdots + a_{2N}x_N^{(0)} = b_2^{(0)} \\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
&a_{N1}x_1^{(0)} + a_{N2}x_2^{(0)} + \cdots + a_{NN}x_N^{(0)} = b_N^{(0)}
\end{aligned}
\qquad (1.40)
$$

Введем обозначения: $\xi_i^{(0)} = x_i - x_i^{(0)}$ – погрешности значений неизвестных, $r_i^{(0)} = b_i - b_i^{(0)}$ – невязки *i* = *1, 2, ..., N*

Вычтем каждое уравнение системы (1.40) из соответствующего уравнения системы (1.39), с учетом введенных обозначений получим

$$
\begin{aligned}
&a_{11}\xi_1^{(0)} + a_{12}\xi_2^{(0)} + \cdots + a_{1N}\xi_n^{(0)} = r_1^{(0)} \\
&a_{21}\xi_1^{(0)} + a_{22}\xi_2^{(0)} + \cdots + a_{2n}\xi_N^{(0)} = r_2^{(0)} \\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
&a_{N1}\xi_1^{(0)} + a_{N2}\xi_2^{(0)} + \cdots + a_{NN}\xi_N^{(0)} = r_N^{(0)}
\end{aligned}
\qquad (1.41)
$$

Решаем эту систему, находим погрешности $\xi_i^{(0)}$, которые используются в качестве поправок к предыдущему решению, и тогда в первом приближении будем иметь:

$$
\begin{aligned}
&x_1^{(1)} = x_1^{(0)} + \xi_1^{(0)}, \\
&x_2^{(1)} = x_2^{(0)} + \xi_2^{(0)}, \\
&\ldots\ldots\ldots\ldots\ldots \\
&x_N^{(1)} = x_N^{(0)} + \xi_N^{(0)}
\end{aligned}
\qquad (1.42)
$$

Теперь эти решения можно опять подставить в исходную систему уравнений и найти новые поправки $\xi_1^{(1)}$, $\xi_2^{(1)}$, …, $\xi_N^{(1)}$ и решения второго приближения: $x_1^{(2)}=x_1^{(1)}+\xi_1^{(1)}$, $x_2^{(2)}=x_2^{(1)}+\xi_2^{(1)}$, …, $x_N^{(2)}=x_N^{(1)}+\xi_N^{(1)}$. Процесс продолжается до тех пор, пока погрешности (поправки) ξ_i не станут достаточно малыми. Это и есть итерационный метод.

Поскольку при итерациях используется *одна и та же матрица коэффициентов* с элементами a_{ij}, а меняются только правые части исходной системы, то можно строить экономичные алгоритмы.

Например, при использовании метода Гаусса сокращается объем вычислений на этапе прямого хода.

Иногда в программе вместо задания разных значений двух последовательных приближений *задается допустимое число итераций*, при достижении которого счет просто прекращается.

1.7. Метод Гаусса – Зейделя

Один из самых распространенных итерационных методов отличается простотой и легкостью программирования.

Проиллюстрируем на примере:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 &= b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 &= b_3 \end{aligned} \qquad (1.43)$$

Пусть диагональные элементы a_{ii} отличны от нуля, в *противном случае можно поменять уравнения местами*. Выразим неизвестные x_1, x_2, x_3 соответственно из уравнений

$$x_1 = \frac{1}{a_{11}}(b_1 - a_{12}x_2 - a_{13}x_3) \quad (1.44)$$

$$x_2 = \frac{1}{a_{22}}(b_2 - a_{21}x_1 - a_{23}x_3) \quad (1.45)$$

$$x_3 = \frac{1}{a_{33}}(b_3 - a_{31}x_1 - a_{32}x_2) \quad (1.46)$$

Зададим некоторые начальные (нулевые) приближения значений неизвестных: $x_1 = x_1^{(0)}$, $x_2 = x_2^{(0)}$, $x_3 = x_3^{(0)}$. Подставляя эти значения в правую часть (1.46), получаем первое приближение для x_1:

$$x_1^{(1)} = \frac{1}{a_{11}}\left(b_1 - a_{12}x_2^{(0)} - a_{13}x_3^{(0)}\right) \quad (1.47)$$

Используем это выражение для $x_1^{(1)}$ и приближение $x_3^{(0)}$ для x_3. Находим из (1.46) приближение для x_2:

$$x_2^{(1)} = \frac{1}{a_{22}}\left(b_2 - a_{21}x_1^{(1)} - a_{23}x_3^{(0)}\right) \quad (1.48)$$

Наконец, используя вычисленные значения $x_1^{(1)}$, $x_2^{(1)}$, находим с помощью (1.47) первое приближение уравнения $x_3^{(1)}$:

$$x_3^{(1)} = \frac{1}{a_{33}}\left(b_3 - a_{31}x_1^{(1)} - a_{32}x_2^{(1)}\right) \quad (1.49)$$

На этом заканчивается первая итерация решения системы.

Используя теперь значения $x_1^{(1)}$, $x_2^{(1)}$, $x_3^{(1)}$, можно аналогично провести вторую итерацию, в результате которой будут найдены вторые приближения: $x_1^{(2)}$, $x_2^{(2)}$, $x_3^{(2)}$.

Приближение с номером k можно представить в виде

$$x_1^{(k)} = \frac{1}{a_{11}}\left(b_1 - a_{12}x_2^{(k-1)} - a_{13}x_3^{(k-1)}\right), \quad (1.50)$$

$$x_2^{(k)} = \frac{1}{a_{22}}\left(b_2 - a_{21}x_1^{(k)} - a_{23}x_3^{(k-1)}\right)$$

$$x_3^{(k)} = \frac{1}{a_{33}}\left(b_3 - a_{31}x_1^{(k)} - a_{32}x_2^{(k)}\right).$$

Итерационный процесс продолжается до тех пор, пока значения $x_1^{(k)}$, $x_2^{(k)}, x_3^{(k)}$ не станут близкими с заданной погрешностью к значениям $x_1^{(k-1)}, x_2^{(k-1)}, x_3^{(k-1)}$.

Рассмотрим теперь систему *n* уравнений с *N* неизвестными

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{i(i-1)}x_{i-1} + a_{ii}x_i + a_{i(i+1)}x_{i+1} + \cdots + a_{iN}x_N = b_i,$$

где *i = 1, 2, ..., N*.

Будем опять полагать, что все диагональные элементы отличны от 0. Тогда в соответствии с методом *k*-е приближение можно представить в виде

$$x_i^{(k)} = \frac{1}{a_{ii}}(b_i - a_{i1}x_1^{(k)} - \cdots - a_{i(i-1)}x_{i-1}^{(k)} - a_{i(i+1)}x_{i+1}^{(k-1)} - \cdots - a_{iN}x_N^{(k-1)}) \quad (1.51)$$

где *i = 1, 2, ..., N*.

Критерий окончания итерационного процесса можно записать в виде $\delta = max\left|x_i^{(k)} - x_i^{(k-1)}\right| < \varepsilon$ – по *max* абсолютного отклонения $1 \le i \le N$ или в виде (при $|x_i|$>>1), $max\left|\frac{x_i^k - x_i^{(k-1)}}{x_i^k}\right| < \varepsilon$ – критерий по относительным разностям.

Для сходимости итерационного процесса достаточным условием является

$$|a_{ii}| \geq \sum_{i \neq j} |a_{ij}| \qquad (1.52)$$

где *i и j = 1, 2, …, N*.

То есть для каждого уравнения системы модуль диагонального элемента был не меньше суммы модулей всех остальных коэффициентов. При этом хотя бы для одного уравнения это условие должно выполняться строго.

Это условие не является необходимым, так как для некоторых систем уравнений итерации сходятся и при нарушении этого условия.

2. НЕЛИНЕЙНЫЕ УРАВНЕНИЯ

Пусть имеем уравнение (нелинейное) вида $F(x) = 0$, $F(x)$ – некоторая *непрерывная функция*. Необходимо найти его корни. Нелинейные уравнения можно разделить на два класса: алгебраические и трансцендентные.

Алгебраические – содержат только алгебраические функции (целые, рациональные, иррациональные), в частности многочлен степени n: $P(x)$ является целой рациональной алгебраической функцией, дробно-рациональной $\frac{P(x)}{q(x)}$, иррациональной $\frac{\sqrt{P(x)}}{q(x)}$.

Трансцендентные – содержат тригонометрические, показательные, логарифмические функции.

Методы решения делятся на прямые и итерационные. Прямые методы позволяют записать корни в виде некоторого конечного соотношения (формулы) (до 4-й степени включительно, если 5-я степень, взять в радикалах невозможно).

Итерационные методы (то есть методы последовательных приближений) состоят из двух этапов:

а) отыскание приближенного значения корня или содержащего его отрезка;

б) уточнение приближенного значения до заданного уровня точности.

Мы рассмотрели случаи только вещественного решения нелинейных уравнений (функция меняет знак).

2.1. Метод деления отрезка пополам

Допустим, что удалось найти отрезок $[a, b]$, в котором расположено значение корня $x = c$, то есть $a < c < b$. Берем в качестве начального приближения корня середину отрезка, то есть $c_0 = \frac{(a+b)}{2}$. Далее смотрим на значения $F(x)$ в точках a, c_0, b. Тот отрезок, на котором $F(x)$ принимает значения разных знаков, содержит искомый корень, поэтому рассмотрим новый отрезок (отрезок $[c_0, b]$) и делим его пополам, получаем точку c_1 и соответственно $F(c_1)$, и далее рассмотрим отрезок $[c_0, c_1]$, делим его пополам и получаем значение c_2 и соответственно $F(c_2)$. Из рис. 1 видно, что точка c_2 уже близка к значению корня c. Таким образом, после каждой итерации отрезок, на котором содержится корень c, уменьшается вдвое.

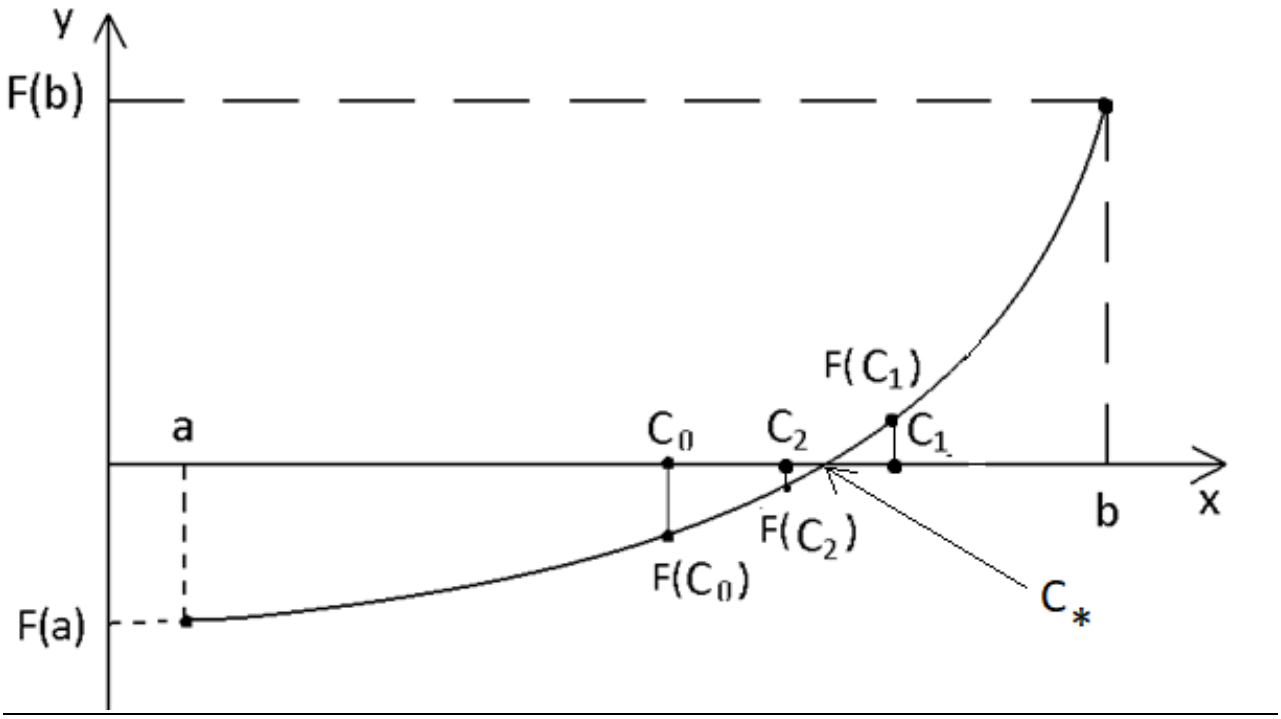

Рисунок 1. Метод деления отрезка пополам

Итерационный процесс продолжается до тех пор, пока значение $F(c_N)$ после N-й итерации не станет меньше по модулю некоторого заданного малого числа ε, то есть $|F(c_N)|<\varepsilon$.

Этот метод медленный, *но всегда сходится*, то есть решение получается всегда, причем с заданной точностью.

2.2. Метод хорд

Пусть мы нашли отрезок $[a, b]$, на котором $F(x)$ меняет знак. Пусть для определенности $F(a)>0$, $F(b)<0$. Процесс итераций заключается в том, что в качестве приближений к корню принимаются значения c_0, c_1, …, c_N точек пересечения хорды с осью абсцисс. Сначала находим уравнение хорды, проходящей через точки $F(a)$ и $F(b)$: $\frac{F(x)-F(a)}{F(b)-F(a)} = \frac{x-a}{b-a}$, чтобы найти точку пересечения, положим $F(x) = 0$. Следовательно, $x = c_0 = \frac{-(b-a)}{F(b)-F(a)}F(a)+a;$ Далее сравниваем $F(a)$ и $F(c_0)$ – для рассматриваемого случая корень будет находиться в интервале (a, c_0), так как $F(c_0)<0$. Отрезок $[c_0, b]$ – отбрасываем и так далее (см. рис. 2).

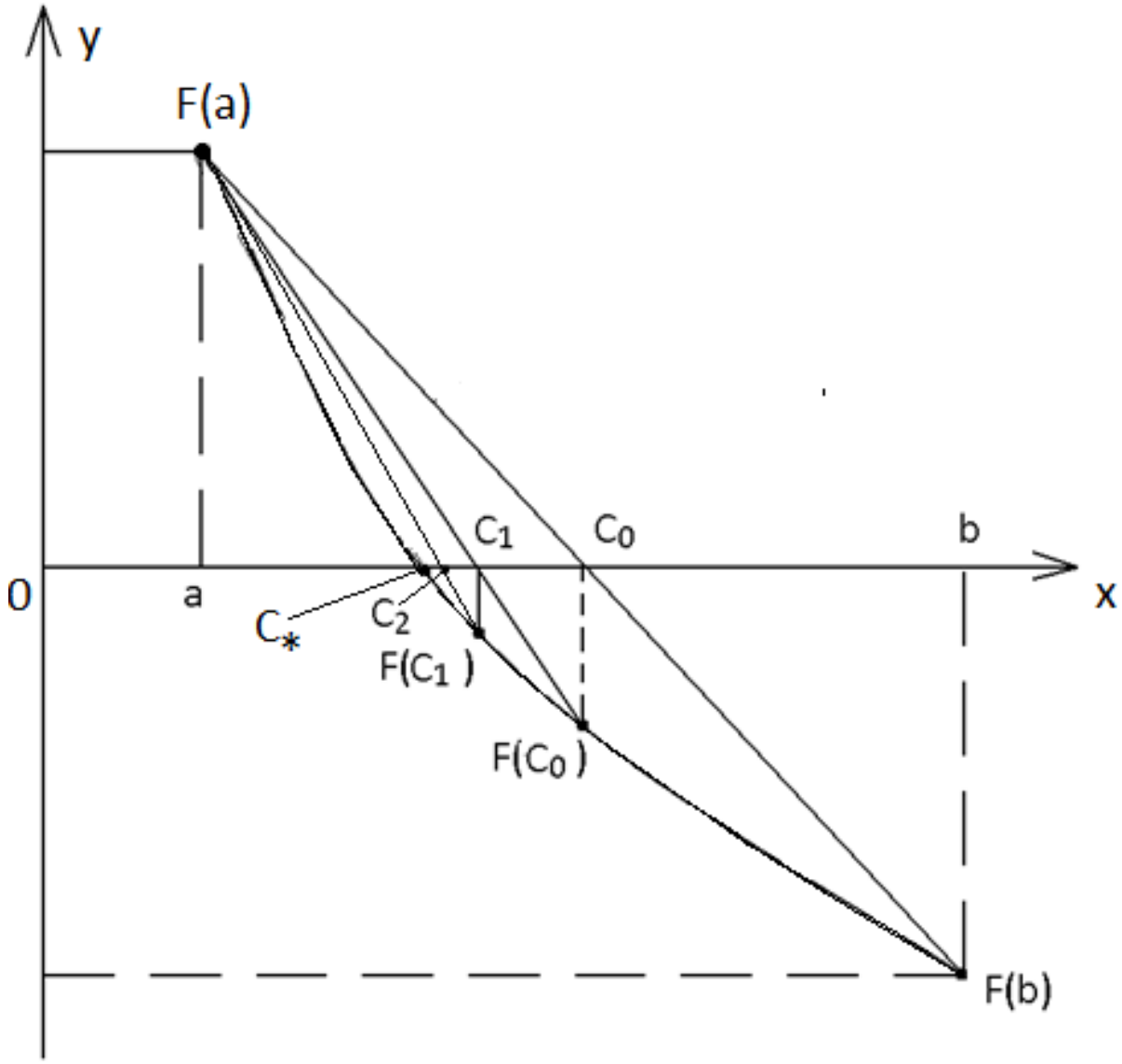

Рисунок 2. Метод хорд

Соответственно для точки c_1, являющейся точкой пересечения хорды, соединяющей $F(a)$ и $F(c_0)$, получим

$$x = C_1 = \frac{-F(a)(c_0 - a)}{F(c_0) - F(a)} + a \qquad (2.1)$$

Соответственно для точки c_2, являющейся точкой пересечения хорды, соединяющей $F(a)$ и $F(c_1)$, получим

$$x = C_2 = \frac{-F(a)(c_1 - a)}{F(c_1) - F(a)} + a \qquad (2.2)$$

Как видно из рис. 2, точка c_2 уже близка к значению корня c.

Соответственно для случая N итераций точки пересечения хорды $F(a)$, $F(c_{N-1})$ получим

$$x = c_N = \frac{-F(a)(c_{N-1} - a)}{F(c_{N-1}) - F(a)} + a \qquad (2.3)$$

Итерационный процесс продолжается до тех пор, пока $|F(c_N)|<\varepsilon$, где ε – заданная погрешность.

2.3. Метод Ньютона

По сути, аналогичен методу секущих. Только здесь вместо хорды на k-й итерации проводится касательная к кривой $y = F(x)$ при $x = c_k$ и ищется точка пересечения касательной с осью абсцисс. Уравнение касательной в точке M_0 с координатами $F(c_0)$, c_0 имеет вид

$$F(x) - F(c_0) = F'(c_0)(x - c_0).$$

Значение x, равное c_1, при котором касательная пересекает ось x, в которой $F(x) = 0$, определится по соотношению

$$x = c_1 = \frac{-F(c_0)}{F'(c_0)} + c_0 \qquad (2.4)$$

Далее все повторяется: находим уравнение касательной в точке c_1, находим точку пересечения этой касательной с осью $x = c_2$. По аналогии формулы c_1 для $x = c_2$можно записать

$$x = c_2 = \frac{-F(c_1)}{F'(c_1)} + c_1 \qquad (2.5)$$

Соответственно для N+1 приближения можно записать

$$C_{N+1} = C_N - \frac{F(c_N)}{F'(c_0)} \qquad (2.6)$$

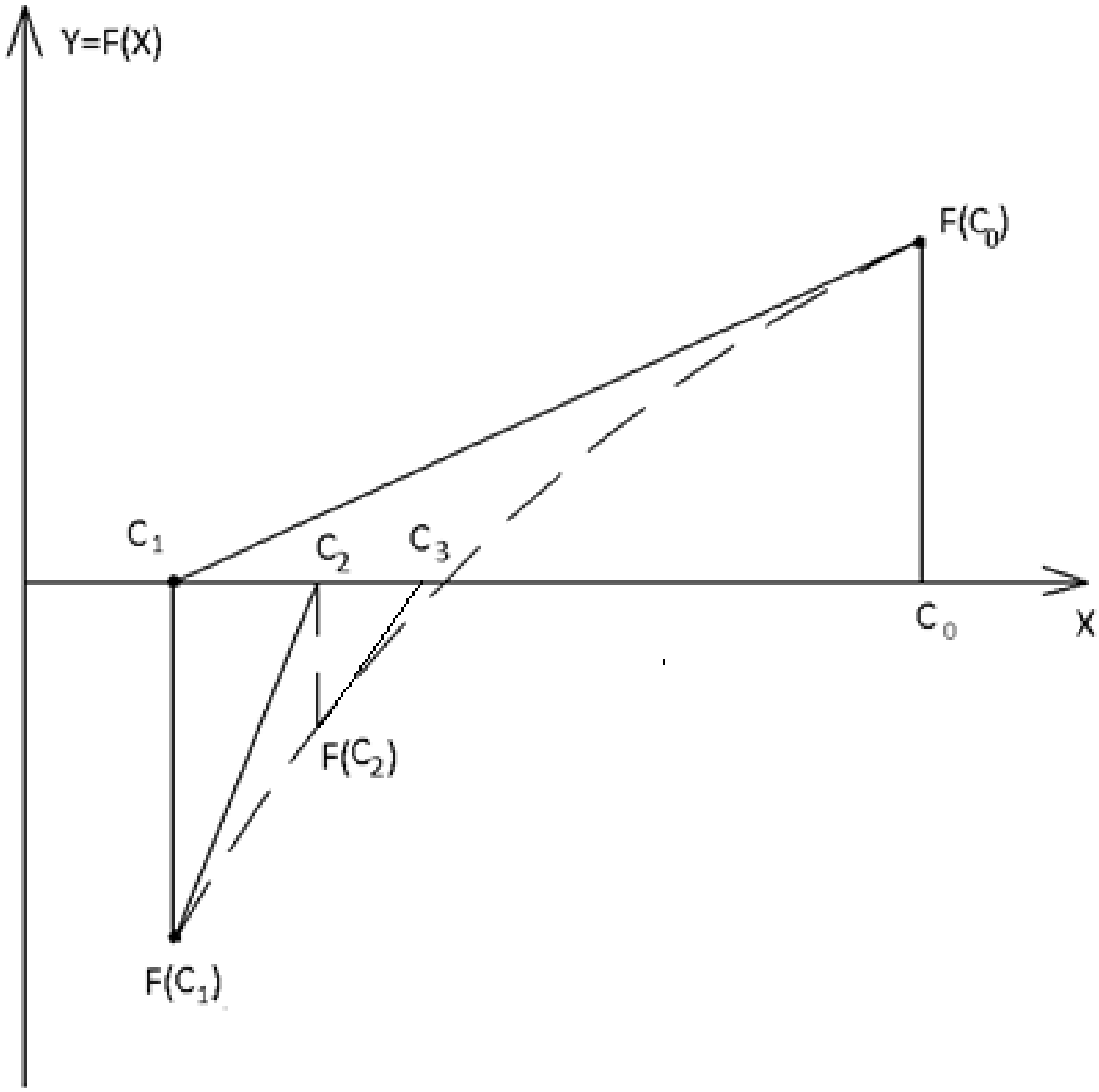

Рисунок 3. Метод Ньютона

При этом, естественно, *необходимо*, чтобы $F'(c_N)\neq 0$. Для окончания итерационного процесса может быть использовано условие $|F'(c_N)|<\varepsilon$ или условие близости двух последовательных приближений

$$|c_{N+1} - c_N| < \varepsilon \qquad (2.7)$$

Мы видим, что здесь *не нужно искать* интервал, где есть корень уравнения. На каждой итерации объем вычислений больше, чем в предыдущих методах, но сходится он быстрее. Например, после 5 - 6 итераций погрешность метода может быть такой же, как у предыдущих после 50 итераций.

Трудность в применении метода Ньютона состоит в выборе начального приближения, для поиска которого иногда целесообразно применить всегда сходящийся метод (например, деления отрезка пополам).

Рассмотренные выше методы применимы для решения как трансцендентных, так и алгебраических уравнений, однако алгебраические уравнения имеют некоторые особенности, которые могут быть использованы при их решении.

2.4. Метод простой итерации

Пусть имеем нелинейное уравнение

$$F(x) = 0. \qquad (2.8)$$

Часто его можно представить в виде $x = f(x)$.

Например: $x\sin x + e^x = 0; x = -\frac{e^x}{\sin x}$.

Исходя из условий задачи, можно выбрать некое начальное приближение $x = c_0$. Подставим c_0 в правую часть и получим $x = c_1$:

$$c_1 = f(c_0) \qquad (2.9)$$

Теперь c_1 подставляем в правую часть и получаем $x = c_2$, $c_2 = f(c_1)$ и так далее, до того пока $|c_{N+1} - c_N| < \varepsilon$, где ε – заданная малая величина.

2.5. Системы нелинейных уравнений

Пусть имеем систему

$$\begin{aligned} F_1(x_1, x_2, \ldots . x_N) &= 0 \\ F_2(x_1, x_2 \ldots . x_N) &= 0 \\ F_N(x_1, x_2 \ldots \ldots x_N) &= 0 \end{aligned} \qquad (2.10)$$

Не существует прямых (в отличие от линейных уравнений) методов решения этих систем. Обычно используют итерационные методы:

а) *метод простой итерации* (напоминает метод Гаусса – Зейделя для линейных уравнений).

Представим (2.10) в виде

$$\begin{aligned} x_1 &= f_1(x_1, x_2, \ldots, x_N) \\ x_2 &= f_2(x_1, x_2, \ldots, x_N) \\ &\ldots\ldots\ldots\ldots\ldots\ldots \\ x_n &= f_n(x_1, x_2, \ldots, x_N) \end{aligned} \qquad (2.11)$$

Пусть в результате предыдущей итерации получены значения неизвестных $x_1 = x_1^{(0)}$; $x_2 = x_2^{(0)}$; …; $x_N = x_N^{(0)}$. Тогда для неизвестных на следующей итерации можно записать

$$x_1^{(1)} = f_1(x_1^{(0)}, x_2^{(0)}, \ldots x_N^{(0)}) \qquad (2.12)$$

$$x_2^{(1)} = f_2\left(x_1^{(1)}, x_2^{(0)}, \dots x_N^{(0)}\right)$$

$$x_N^{(1)} = f_N(x_1^{(1)}, x_2^{(1)}, \dots x_{N-1}^{(1)}, x_N^0)$$

Для k-й итерации можно записать

$$\begin{gathered} x_1^{(k)} = f_1(x_1^{(k-1)}, x_2^{(k-1)}, \dots x_N^{(k-1)}) \\ x_2^{(k)} = f_2(x_1^k, x_2^{(k-1)}, x_3^{(k-1)}, \dots x_N^{(k-1)}) \\ \dots\dots\dots\dots\dots\dots\dots\dots \\ x_N^{(k)} = f_N(x_1^{(k)}, x_2^{(k)}, \dots\ x_{N-1}^{(k)}, x_N^{(k-1)}) \end{gathered} \qquad (2.13)$$

Итерационный процесс продолжается до тех пор, пока значения неизвестных в двух последовательных итерациях не станут достаточно малыми $\left[x_i^{(k)} - x_i^{(k-1)}\right] < \varepsilon_i$.

Здесь успех определяется удачным выбором начальных приближений. В противном случае итерационный процесс может не сойтись.

б) *метод Ньютона.* Обладает более быстрой сходимостью.

Зададим значения неизвестных в нулевом приближении $x_1^{(0)}, x_2^{(0)}, \dots, x_N^{(0)}$ Задача состоит в нахождении приращений (поправок) к этим значениям: $\Delta x_1^{(0)}, \Delta x_2^{(0)}, \dots, \Delta x_N^{(0)}$, благодаря которым решение системы запишется в виде:

$$\begin{gathered} x_1 = x_1^{(0)} + \Delta x_1^{(0)} \\ x_2 = x_2^{(0)} + \Delta x_2^{(0)} \\ \dots\dots\dots\dots\dots\dots \\ x_N = x_N^{(0)} + \Delta x_N^{(0)} \end{gathered} \qquad (2.14)$$

С этой целью проведем разложение левых частей уравнения (2.11) в ряд Тейлора, ограничиваясь лишь линейными членами, относительно этих приращений.

$$
\begin{aligned}
&F_1(x_1, x_2, \dots, x_N) \\
&\quad = F_1\left(x_1^{(0)}, x_2^{(0)}, \dots, x_N^{(0)}\right) + \frac{\partial F_1}{\partial x_1}\Delta x_1^{(0)} + \cdots \\
&\quad + \frac{\partial F_1}{\partial x_2}\Delta x_N^{(0)} \\
&F_2(x_1, x_2, \dots, x_N) \\
&\quad = F_2\left(x_1^{(0)}, x_2^{(0)}, \dots, x_N^{(0)}\right) + \frac{\partial F_2}{\partial x_1}\Delta x_1^{(0)} + \cdots \\
&\quad + \frac{\partial F_2}{\partial x_N}\Delta x_N^{(0)} \\
&\quad \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\
&F_n(x_1, x_2, \dots, x_N) \\
&\quad = F_n\left(x_1^{(0)}, x_2^{(0)}, \dots, x_N^{(0)}\right) + \frac{\partial F_n}{\partial x_1}\Delta x_1^{(0)} + \cdots \\
&\quad + \frac{\partial F_n}{\partial x_N}\Delta x_N^{(0)}
\end{aligned}
\tag{2.15}
$$

Приравнивая нулю правые части, получим систему линейных уравнений относительно приращений $\Delta x_1^{(0)}, \Delta x_2^{(0)}, \dots \dots \Delta x_N^{(0)}$.

$$
\begin{aligned}
&\frac{\partial F_1}{\partial x_1}\Delta x_1^{(0)} + \frac{\partial F_1}{\partial x_2}\Delta x_2^{(0)} + \cdots + \frac{\partial F_1}{\partial x_N}\Delta x_N^{(0)} \\
&\qquad = -F_1\left(x_1^{(0)}, x_2^{(0)}, \dots, x_N^{(0)}\right)
\end{aligned}
\tag{2.16}
$$

$$\frac{\partial F_2}{\partial x_1}\Delta x_1^{(0)} + \frac{\partial F_2}{\partial x_2}\Delta x_2^{(0)} + \cdots + \frac{\partial F_2}{\partial x_N}\Delta x_N^{(0)} = -F_2\left(x_1^{(0)}, x_2^{(0)}, \ldots, x_N^{(0)}\right)$$

……

$$\frac{\partial F_n}{\partial x_1}\Delta x_1^{(0)} + \frac{\partial F_n}{\partial x_2}\Delta x_2^{(0)} + \cdots + \frac{\partial F_n}{\partial x_N}\Delta x_N^{(0)} = -F_N\left(x_1^{(0)}, x_2^{(0)}, \ldots, x_N^{(0)}\right)$$

Значения F_i и $\frac{\partial F_i}{\partial x_i}$ – вычисляются при $x_1 = x_1^{(0)}, x_2 = x_2^{(0)}, \ldots, x_N = x_N^{(0)}$.

Определив $\Delta x_1^{(0)}, \Delta x_2^{(0)}, \ldots \Delta x_N^{(0)}$ получим значения неизвестных в первом приближении: $x_1^{(1)} = x_1^{(0)} + \Delta x_1^{(0)}, x_2^{(1)} = x_2^{(0)} + \Delta x_2^{(0)}, \ldots, x_N^{(1)} = x_N^{(0)} + \Delta x_N^{(0)}$.

Аналогично для второго приближения можно записать

$$x_1^{(2)} = x_1^{(1)} + \Delta x_1^{(1)}, \ldots, x_N^{(2)} = x_N^{(1)} + \Delta x_N^{(1)} \qquad (2.17)$$

Для определения приращений первого приближения можно записать систему уравнений, аналогичную (2.16), и т.д.

Для k-го приближения можно записать

$$\begin{aligned} x_1^{(k)} &= x_1^{(k-1)} + \Delta x_1^{(k-1)} \\ x_2^{(k)} &= x_2^{(k-1)} + \Delta x_2^{(k-1)} \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ x_N^{(k)} &= x_N^{(k-1)} + \Delta x_N^{(k-1)} \end{aligned} \qquad (2.18)$$

Счет прекращается, когда на *k*-м приближении $max\left|\Delta x_i^k\right| < \varepsilon$. В методе Ньютона важен *удачный выбор начального*

приближения. Сходимость *ухудшается* с ростом числа уравнений системы.

Раздел имеет не только самостоятельный интерес, но и имеет существенное значение для решения главных задач вычислительной математики, а именно краевых задач.

3. АППРОКСИМАЦИЯ ФУНКЦИЙ

Аппроксимирующая (приближенная) функция – это функция, которая с минимальным отклонением описывает другую функцию, полученную следующими способами:

- в виде результатов расчета по некоторой сложной функции;
- в виде результатов расчета по некоторой программе;
- в виде таблицы экспериментальных данных.

Что дает получение аппроксимирующей функции?

– использование ее при интегрировании и дифференцировании;

– дает возможность определить ее значение как в узлах, так и между узлами;

Решение многих краевых задач различными численными методами представляется в виде набора чисел (таблиц), и по ним нужно восстановить искомую функцию.

В любом из этих случаев для построения аппроксимации функции мы будем иметь таблицу, которая в одномерном случае будет выглядеть так. Пусть

$$y_i = f(x_i) \qquad (3.1)$$

где x_i находиться в пределах $a \le x_i \le b$.

y	y_0	y_1	y_2	$\cdots$	y_i	$\cdots$	y_n
x	x_0	x_1	x_2	$\cdots$	x_i	$\cdots$	x_n

$$\phi(x) = a_0\varphi_0(x) + a_1\varphi_1(x) + \cdots + a_n\varphi_n(x). \quad (3.2)$$

Здесь $\phi(x)$ - аппроксимирующая функция; $\varphi_0(x), \varphi_1(x), \ldots \varphi_N(x)$ – заданные функции; $a_0, a_1, \ldots, a_N$ – коэффициенты неизвестны, $x_0, x_1, \ldots, x_N$ – узлы таблицы.

При построении аппроксимирующей функции $\phi(x)$ возникают следующие вопросы:

Каков критерий минимальности отклонения $\phi(x)$ от функции, заданной таблично $y = f(x)$?

Каков класс функций $\varphi(x)$?

Как определять неизвестные коэффициенты?

Как располагать узлы – равномерно или неравномерно?

Какова погрешность?

Эти вопросы чаще всего связаны друг с другом.

Используемые критерии:

Критерий интерполяции. Значения аппроксимируемой и заданной функции должны совпадать в узлах: $f(x_i) = \phi(x_i)$, $i = 0,1,2, \ldots, N$. Казалось, естественно, однако не в случае экспериментальных данных. Если здесь требовать совпадения, то мы будем повторять ошибки эксперимента.

Критерий наименьших квадратов:

$$\min \sum_{i=1}^{N} (f(x_i) - \phi(x_i))^2 \quad (3.3)$$

Этот критерий дает возможность использовать многочлен степени $m << N$;

Критерий наилучшего равномерного приближения

На заданном отрезке $[a, b]$ максимальная величина абсолютного отклонения должна быть минимальной

Критерий минимакса

$$\underline{min.max}|f(x_i) - \phi(x_i)| \qquad (3.4)$$

3.1. Используемые классы функций

Классы функций

Алгебраические функции – в виде полиномов. Тригонометрические функции – sin, cos.

Определение коэффициентов

Часто связано с критерием, например МНК, критерий интерполяции.

Оцениваемая погрешность

Оценивается в узлах, между узлами.

Часто при изложении материала различные методы аппроксимации классифицируют исходя из критерия малости отклонения. Мы поступим также, но будем иметь в виду, что в зависимости от некоторых условий один и тот же вид аппроксимирующей функции может удовлетворять разным критериям.

Рассмотрим наиболее распространенные, так называемые, интерполяционные многочлены, удовлетворяющие критерию интерполяции.

Пусть функция $y = f(x)$ задана таблицей:

y	y_0	y_1	y_2	$\cdots$	y_i	$\cdots$	y_n
x	x_0	x_1	x_2	$\cdots$	x_i	$\cdots$	x_n

Аппроксимирующую функцию ищем в виде

$$\phi(x) = a_0\varphi_0(x) + a_1\varphi_1(x) + \cdots + a_N\varphi_N(x) \qquad (3.5)$$

Рассмотрим некоторые характерные (или часто применяемые) аппроксимирующие функции и соответствующие методы.

3.2. Интерполяционные многочлены

В данном случае в качестве заданных функций выбирается следующая линейно независимая система функций, например:

$$\varphi_0(x) = 1, \varphi_1(x) = x, \varphi_2(x) = x^2, \ldots \varphi_N(x) = x^n \qquad (3.6)$$

В этом случае аппроксимирующая функция примет вид

$$\varphi(x) = a_{0+}a_1x + a_2x^2 + \cdots a_Nx^n \qquad (3.7)$$

Используя данные таблицы и критерии интерполяции можно записать следующую систему уравнений для определения неизвестных коэффициентов:

$$y_{i=}a_0 + a_1x_i + a_2x_i^2 + \cdots a_Nx_i^n$$

где *i = 0, 1, 2, ..., N.*

Определитель этой системы имеет вид

$$\begin{vmatrix} 1 & x_0 & x_0^2 & \cdot & x_0^n \\ 1 & x_1 & x_1^2 & \vdots & x_1^n \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & x_N & x_N^2 & \cdot & x_N^n \end{vmatrix} \neq 0 \qquad (3.8)$$

Это *определитель Вандермонда.* Он не равен нулю в случае несовпадающих узлов. Это означает, что решение указанной системы уравнений существует, и оно единственно.

Важное замечание, касающееся всех аппроксимирующих функций в виде интерполяционных многочленов: существует единственный интерполяционный многочлен, но различные формы его записи, например в виде полиномов Лагранжа, в виде полиномов Ньютона. Они имеют различные формы записи, но, по существу, это один и тот же единственный интерполяционный многочлен.

3.3. Аппроксимация с помощью алгебраических полиномов Лагранжа

Аппроксимация с помощью алгебраических полиномов Лагранжа. Применяется на основе критерия интерполяции, то есть совпадения в узлах $x_0, \dots, x_i, \dots, x_N$.

Роль коэффициентов играют сами значения функции в узлах, деленные на произведение скобок. Формула справедлива для *произвольного расположения* узлов. Когда подставляем $x = x_i$, получаем y_i, то есть имеет место совпадение в узлах.

Преимущества аппроксимирующей функции в форме полинома Лагранжа. Эта форма записи справедлива как для равноотстоящих узлов, так и для неравномерно отстоящих узлов;

Число арифметических операций для построения многочлена Лагранжа $\approx n^2$ и является наименьшим для всех форм записи:

$$L(x) = y_0 \frac{(x-x_1)(x-x_2)\dots(x-x_N)}{(x_0-x_1)(x_0-x_2)\dots(x_0-x_N)} + y_1 \frac{(x-x_1)(x-x_2)\dots(x-x_N)}{(x_1-x_0)(x_1-x_2)\dots(x_1-x_N)} + \\ + \dots + y_N \frac{(x-x_1)(x-x_2)\dots(x-x_{N-1})}{(x_N-x_0)(x_N-x_1)\dots(x_N-x_{N-1})} \quad (3.9)$$

$$L(x) = \sum_{i=0}^{n} \frac{(x-x_0)(x-x_1)\dots(x-x_{i-1})(x-x_{i+1})\dots(x-x_N)}{(x_i-x_0)(x_i-x_1)\dots(x_i-x_{i-1})(x_i-x_{i+1})\dots(x_i-x_N)}$$

Формула содержит в явном виде значение функции y_i, что бывает удобным, в частности, при проведении численного интегрирования.

Полином Лагранжа удобен, когда значения функции меняются, а узлы неизменны.

Недостатки:

При изменении числа узлов все расчеты нужно делать заново.

Полином Лагранжа чаще всего используется до второй степени

$n = 2$, то есть по данным трех узлов. Например, в узлах x_{i-1}, x_i, x_{i+1}. Иными словами, если таблица содержит большое количество узлов, то описывать с помощью полинома Лагранжа данные всей таблицы нельзя, так как возникают большие погрешности. Полиномы Лагранжа можно использовать лишь для локальной части таблицы, а не глобально, то есть для всей таблицы.

3.4. Аппроксимация с помощью интерполяционных полиномов Ньютона

Аппроксимация с помощью интерполяционных полиномов Ньютона применяется с использованием критерия интерполяции, то есть требования совпадения в узлах

$$N(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \cdots + \\ + a_M(x - x_0)(x - x_1) \dots (x - x_{M-1}) + \cdots + \\ + a_N(x - x_0)(x - x_1) \dots (x - x_{N-1}) \quad (3.10)$$

Неизвестные коэффициенты определяются из критерия интерполяции

$$a_m = \sum_{i=0}^{M} \frac{y_i}{(x_i - x_0)(x_i - x_1) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_{M-1})} \quad (3.11)$$

$m = 0, 1, \dots, N.$

Формула справедлива для *произвольного расположения* узлов. Если будут добавляться новые узлы, ранее рассчитанные коэффициенты не изменяются.

Рассмотрим определение коэффициентов полинома Ньютона исходя из принципа интерполяции. Для случая трех узлов:

y_0	y_1	y_2	$\cdots$
x_0	x_1	x_2	$\cdots$

Исходя из принципа интерполяции, определим коэффициенты a_0, a_1, a_2 для случая трех узлов.

При i = 0, во все скобки полинома Ньютона вместо x подставим x_0, получим $a_0, = y_0$.

При $i = 1$, $N(x_1) = y_1 = a_0 + a_1(x_1 - x_0)$

все остальные члены полинома равны нулю, так как содержат $(x_1 - x_1)$

Отсюда $a_1 = \frac{y_1 - y_0}{(x_1 - x_0)} = -\frac{y_0}{(x_1 - x_0)} + \frac{y_1}{(x_1 - x_0)}$

При *i = 2*, $N(x_2) = y_2 = a_0 + a_1(x_2 - x_0) + a_2(x_2 - x_0)(x_2 - x_1)$

Тогда $a_2 = \frac{y_2 - a_0 - a_1(x_2 - x_0)}{(x_1 - x_0)(x_2 - x_1)}$.

Остальные коэффициенты равны нулю.

Можно показать, что эти коэффициенты $a_0, a_1 \ldots a_m \ldots a_n$ определяются по формуле

$$a_M = \sum_{k=0}^{M} \frac{y_k}{L_M^{(k)}(x_k)} \qquad (3.12)$$

где *M = 0, 1, 2, …, N*

Замечания: Отметим, что, как видно из формулы, для a_M нужно знать только первые *M* узлов. То есть (*0, 1, 2, …, M*) первые табличные данные, а остальные не нужно. Поэтому добавление новых узлов не изменяет уже найденные коэффициенты.

Очень важное замечание: В силу единственности решения *при заданном наборе узлов существует только один интерполяционный многочлен,* разница лишь в алгоритмах его построения (в форме *L*, *H*, канонической форме либо другой) и в его форме.

Если надо вычислить приближенное значение *y* при $x \neq x_i$ (между узлами), то не надо привлекать интерполяционный полином, построенный *по всем узлам*. Достаточно построить

полином невысокой степени по узлам, ближайшим к *x*. При этом x_i можно назвать x_1, а ближайшие точки слева или справа узлами x_0, x_2 и так далее.

При заданном наборе узлов существует один и только один многочлен степени не выше *n*, принимающий в узлах заданные значения. Форма разная, но многочлен один.

Если заданная в узлах функция – многочлен, то между узлами также будет совпадение. Если функция – не многочлен, то погрешность между узлами определяется соотношением

$$R_{L/N}(\mathrm{x}) = \frac{(x-x_0)(x-x_1)\ldots(x-x_N)}{(N+1)!} f^{n+1}(x) \qquad (3.13)$$

где $f^{n+1}(x)$ – производная *n*+1 порядка.

К сожалению, для большинства функций с ростом *N* производная порядка *n* растет как *n*!. То есть погрешность зависит от расположения узлов и величины производной. Например, для функции *ln x* производная *n*-го порядка приблизительно равна *n*!, как будет вести себя производная функции, неизвестно.

Формулой для $R_{L/N}$ можно узнать величину погрешности. Обычно для увеличения точности увеличивают число узлов. Рунге показал, что при увеличении *N*; для функции $\frac{1}{1+25x^2}$ в интервале $[-1, +1]$ аппроксимирующая функция в интервале $0{,}75 \leq x \leq 1$ расходится с заданной таблично (см. рис. 4).

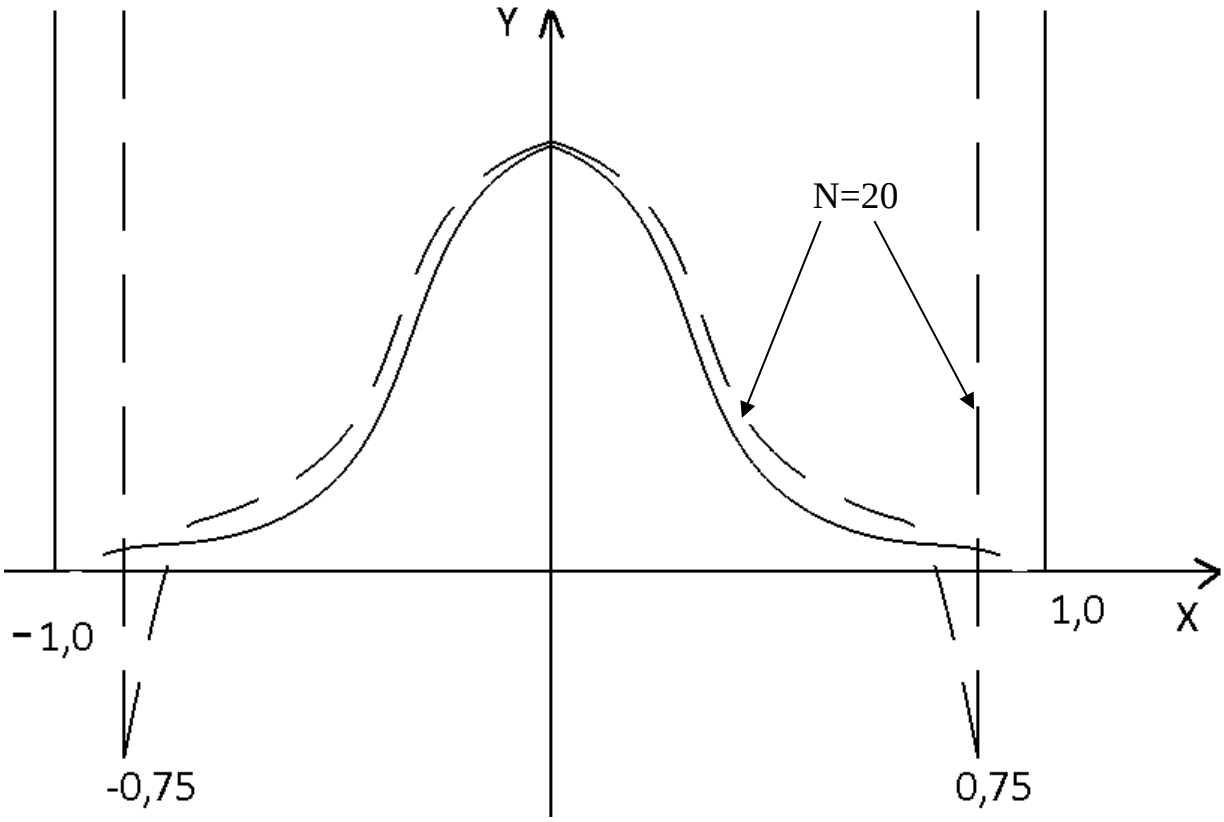

Рисунок 4. Аппроксимируемая функция

Пунктир на рис. 4 соответствует аппроксимирующей функции $\varphi(x)$ при числе узлов n, равнov 20. Сплошная кривая соответствует истинной кривой $f(x)=\frac{1}{1+25x^2}$.

Видно, что при $x > 0{,}75$ и меньших -0,75 аппроксимирующая кривая уходит в бесконечность.

Однако если узлы располагать в нулях полиномов Чебышева, то аппроксимирующая функция будет сходиться к заданной. К сожалению, это правило действует не для всех непрерывных функций. Безусловно, недостатком L и N является рост $R_{L/N}$ с числом узлов. И для глобальной интерполяции они мало удобны.

3.5. Метод наименьших квадратов (МНК)

Пусть $y = f(x_i)$ задана в узлах $i = 0, 1, 2, \ldots, N$; $y_0, y_1, \ldots, y_i, \ldots, N; x_0, x_1, \ldots, x_i, \ldots, x_N$.

Представим аппроксимирующую функцию в виде $\phi_m(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_M x^m = \sum_{i=0}^{m} a_i x^i$. Где $m << N$.

Необходимо найти такие $a_0 \ldots a_M$, чтобы сумма квадратов отклонений в узлах была минимальна, то есть *min*: $\Phi = \sum_{i=0}^{N} [\phi(x_i, a_0, \ldots a_M) - y_i]^2$.

Необходимые условия минимума:

$$\frac{\partial \Phi}{\partial a_i} = 0;$$

$$\frac{\partial \Phi}{\partial a_0} = \sum_{i=0}^{N} 2\left[a_0 + a_1 x_i + a_2 x_i^2 + \cdots + a_m x_i^m - y_i\right] = 0$$

$$\frac{\partial \Phi}{\partial a_1} = \sum_{i=0}^{N} 2\left[a_0 + a_1 x_i + a_2 x_i^2 + \cdots + a_m x_i^m - y_i\right] x_i = 0 \quad (3.14)$$

……

$$\frac{\partial \Phi}{\partial a_m} = \sum_{i=0}^{N} 2\left[a_0 + a_1 x_i + a_2 x_i^2 + \cdots + a_m x_i^m - y_i\right] x_i^m = 0$$

В итоге

$$\begin{cases} (N+1)a_0 + a_1\sum_{i=0}^{N} x_i + \cdots + a_m\sum_{i=0}^{N} x_i^m = \sum_{i=0}^{N} y_i \\ a_0\sum_{i=0}^{N} x_i + a_1\sum_{i=0}^{N} x_i^2 + \cdots + a_m\sum_{i=0}^{N} x_i^{m+1} = \sum_{i=0}^{N} y_i x_i \\ a_0\sum_{i=0}^{N} x_i^2 + a_1\sum_{i=0}^{N} x_i^3 + \cdots + a_m\sum_{i=0}^{N} x_i^{m+2} = \sum_{i=0}^{N} y_i x_i^2 \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ a_0\sum_{i=0}^{N} x_i^m + a_1\sum_{i=0}^{N} x_i^{m+1} + \cdots + a_m\sum_{i=0}^{N} x_i^{2m} = \sum_{i=0}^{N} y_i x_i^m \end{cases} \quad (3.15)$$

Рассмотрим одно важное свойство этой системы. Пусть точки неравномерно расположены на интервале [0,1]. В этом случае определитель матрицы

$$det = \frac{[1!2!\ldots(p-1)!]^3}{p!(p+1)!\ldots(2p-1)!}; \quad (3.16)$$

где $p = M - 1$

С ростом M величина определителя быстро падает, так для $M = 1$ $det = 10^{-2}$, для $M = 2$; $det = 10^{-5}$. Система становится плохо обусловленной и теряется точность. Поэтому при МНК берут обычно $M \leq 3$.

Чтобы повысить M, можно разбить систему уравнений на две системы меньшего порядка.

При $M = N$ – совпадение в узлах, но между узлами могут быть существенные отклонения.

3.6. Аппроксимация ортогональными функциями, полиномами Чебышева, тригонометрическими функциями

Общие понятия. Система функций $\varphi_0(x), \varphi_1(x) \dots \varphi_i(x) \dots \varphi_N(x)$, заданная на дискретном множестве точек $\{x_0, x_1, \dots x_i \dots x_N\}$ называется ортогональной, если

$$\sum_{i=0}^{n} \varphi_k(x_i)\varphi_e(x_i) = 0 \qquad k \neq l \tag{3.17}$$

$$\sum_{i=0}^{n} \varphi_k^2(x_i) = c$$

Ряд $a_0\varphi_0(x) + a_1\varphi_1(x) + \cdots + a_N\varphi_N(x)$, в котором $a_k = \frac{\sum_{i=0}^{N} f(x_i)\varphi_k(x_i)}{c}$, называется рядом Фурье: обычно этот ряд сходится к $f(x)$. Здесь для определения коэффициентов не требуется применения критерия близости кривых, а значения коэффициента a_k находятся из приведенных выше условий ортогональности.

3.7. Полином Чебышева

Полином Чебышева имеет вид

$$\phi(x) = \sum_{i=0}^{N} a_i T_i(x),\ x_0, x_1, \dots, x_i;\ -1 \le x \le 1,\ -1 \le T_i(x) \le 1$$

$$T_i(x) = \frac{1}{2}\left[(x + \sqrt{x^2-1})^n + (x - \sqrt{x^2-1})^n\right];\ T_{i+1}(x) = 2xT_i(x) - T_{i-1}(x) \tag{3.18}$$

При $i = 0,\ T_0 = 1;\ i = 1,\ T_1(x) = x;\ T_2(x) = 2x^2 - 1$ и т.д. в соответствии с формулой (3.18)

Графики многочленов $T_0(x), T_1(x), T_2(x)$ представлены ниже на рис. 5.

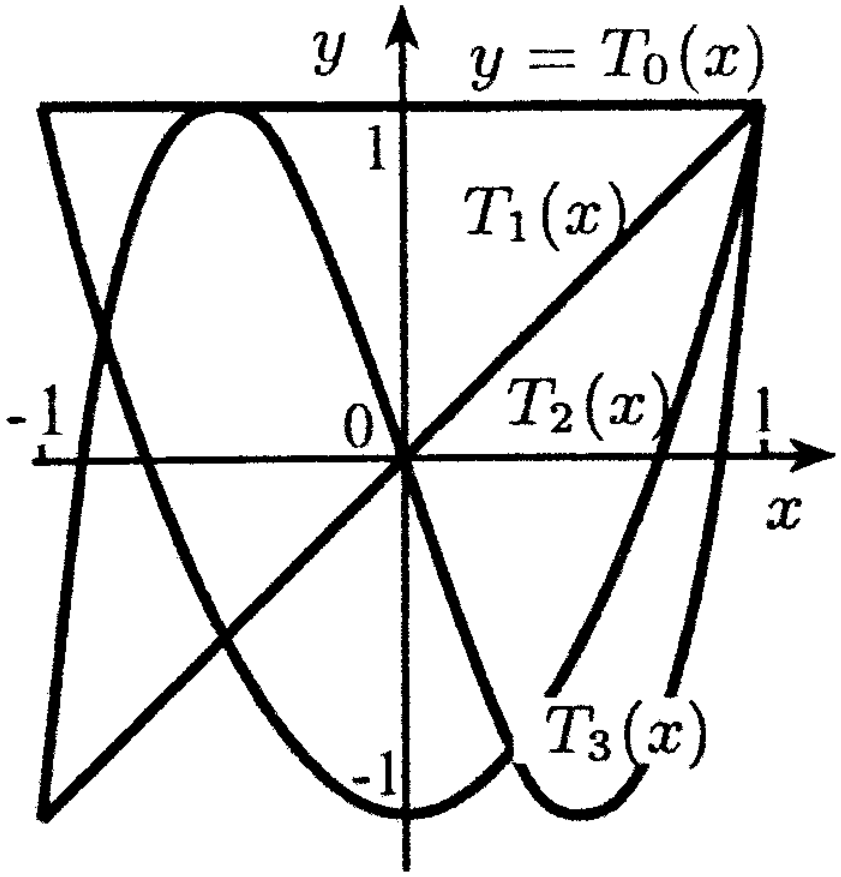

Рисунок 5. Графики многочленов Чебышева

Следует отметить, что при применении полиномов Чебышева переменные *x* лежат в пределах от -1 до 1, и сами полиномы также лежат в пределах от -1 до 1. Нули (корни) многочленов Чебышева на отрезке [-1, +1] определяются формулами:

$$x_k = cos(\frac{2k-1}{2n})\pi, k = 1,2,\dots,n-1. \qquad (3.19)$$

Если табличные значения $y = f(x)$ заданы в нулях полиномов Чебышева, то есть при x_k, то неизвестные коэффициенты определяются по формулам:

$$\begin{cases} a_0 = \dfrac{1}{n+1}\displaystyle\sum_{k=1}^{N} f(x_k) \\ a_n = \dfrac{2}{n+1}\displaystyle\sum_{k=1}^{N} f(x_k)T_n(x_k) \end{cases} \qquad (3.20)$$

Замечания: если степень многочлена (или число неизвестных коэффициентов) равна *N* – числу узлов, то в узлах – совпадения с табличными значениями, а между узлами – *равномерное приближение* (при узлах в нулях).

Если узлы равномерны, то между узлами – выполнение МНК.

Если ряд усечь, то есть $M < N$, то *при равномерном* расположении узлов в узлах – МНК, можно ожидать что и между узлами, *при расположении узлов* в нулях Чебышева в узлах – равномерное приближение, можно ожидать, что и между узлами.

Если функция задана в нулях Чебышева, то:

а) при $N > \infty$, погрешность $R > 0$;

б) Аппроксимационная функция малочувствительна к ошибкам заданной таблично функции.

Аппроксимация с помощью ортогональных тригонометрических функций.

Ряды Фурье

Пусть *f*(*x*) задана на множестве узлов: *x = 0, 1, 2, …, N-1*. Тогда аппроксимирующую функцию можно представить в виде ряда Фурье для случая *N* – четное число

$$\varphi(x) = \frac{a_0}{2} + \sum_{k=1}^{\frac{N}{2}-1}\left(a_k \cos\frac{2\pi}{N}kx + b_k \sin\frac{2\pi}{N}kx\right) + \frac{a_{N/2}}{2}\cos \pi x$$

В соответствии с критерием ортогональности для коэффициентов ряда Фурье можно записать:

$$a_0 = \frac{2}{N}\sum_{x=0}^{N-1} f(x);\ a_k = \frac{2}{N}\sum_{x=0}^{N-1} f(x)\cos\frac{2\pi}{N}kx;$$

$$b_k = \frac{2}{N}\sum_{x=0}^{N-1} f(x)\sin\frac{2\pi}{N}kx;$$

$$a_{N/2} = \frac{2}{N}\sum_{x=0}^{N-1} f(x)\cos \pi x;$$

При $k = \frac{N}{2}$ – совпадение аппроксимирующей функции и заданной функции в узлах, между узлами удовлетворяется критерий МНК.

$k < \frac{N}{2}$ – в узлах минимум среднеквадратичного отклонения МНК), между узлами можно также ожидать МНК.

При увеличении числа N (узлов) погрешность аппроксимации не увеличивается.

Аппроксимация с помощью рядов Фурье широко применяется при обработке как теоретических, так и экспериментальных результатов.

4. МЕТОДЫ ЧИСЛЕННОГО ИНТЕГРИРОВАНИЯ

Не всегда удается вычислить определенный интеграл по формуле Ньютона-Лейбница:

$$\int_a^b f\,(x)dx = F(x)\Big|_b^a = F(b) - F(a) \qquad (4.1)$$

так как вид функции $f(x)$ не допускает непосредственного интегрирования, т. е. вид первообразной функции $F(x)$ неизвестен и значения $f(x)$ заданы только на фиксированном множестве точек x_i, то есть функция задана в виде таблицы.

В этих случаях используются методы численного интегрирования. Они основаны на аппроксимации подынтегральной функции некоторыми более простыми выражениями, например, многочленами. В частности, можно представить подынтегральную функцию в виде степенного ряда (ряда Тейлора). Таким образом, вычисление интеграла от сложной функции сводится к интегрированию нескольких первых членов ряда.

Пример: вычислить $I = \int_0^1 e^{-x^2}\,dx$ с погрешностью 10^{-4}. Разложим экспоненту в ряд: $e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots + 0(x^4)$. Заменим в этом выражении x на $(-x^2)$ и записываем интеграл в виде

$$I = \int_0^1 \left(1 - x^2 + \frac{x^4}{2!} + \frac{x^6}{3!} + \cdots\right) dx = x - \frac{x^3}{3} + \frac{x^5}{5\cdot 2!} - \frac{x^7}{7\cdot 3!} + \cdots \Big|_0^1 =$$

$$= 1 - \frac{1}{3} + \frac{1}{10} - \frac{1}{42} + \ldots \approx 0{,}7408$$

Более универсальным методом, пригодным для обоих случаев 1) и 2), является метод численного интегрирования, основанный на

аппроксимации с помощью интерполяционных многочленов. Будем использовать кусочную аппроксимацию. Это позволит приближенно заменить определенный интеграл интегральной суммой следующим образом.

Пусть на отрезке $[a, b]$ задана функция $y=f(x)$. Разобьем отрезок $[a, b]$ на n элементарных отрезков $[x_{i-1}, x_i]$ $(i = 1, 2, 3)$, причем $x_0=a$, $x_N=b$. На каждом отрезке выберем произвольную точку $\xi_i(x_{i-1} \le \xi_i \le x_i)$ и найдем произведение $S_i = f(\xi_i)\Delta x_i$.

Составим сумму таких произведений $S_N = \sum_{i=1}^{N} f(\xi_i)\Delta x_i$ – это и есть интегральная сумма. При неограниченном увеличении числа точек разбиения и стремлении длины наибольшего отрезка к нулю будем иметь

$$\int_a^b f(x)dx = \lim_{N \to \infty} \sum_{i=1}^{N} f(\xi_i)\Delta x_i, \qquad max\Delta x_i \to 0, \tag{4.2}$$

Это формула для интегральной суммы.

В зависимости от способа вычисления интегральной суммы получаются различные способы численного интегрирования (метод прямоугольника, метод трапеций, параболы, сплайнов). Вычисление кратных интегралов, в конечном счете, можно свести к вычислению интегральной суммы.

4.1. Метод прямоугольников

Здесь в формуле для интегральной суммы в качестве ξ_i можно взять левые или правые границы элементарных отрезков: $\xi_i=\xi_{i-1}$,

ξ_i=ξ_{i+1}; а можно – центр отрезка, т.е. $\xi_i = \frac{\xi_{i-1}+\xi_{i+1}}{2}$. Эти случаи представлены на рисунке.

Положим для простоты, что интервалы отрезков h везде постоянны, и перейдем к обозначениям с помощью переменной x:

$$x_0 = \xi_0, \ldots, x_{i-1} = \xi_{i-1}, x_i = \xi_i;$$
$$x_{i+1} = \xi_{i+1}, \xi_n = x_n = b;\ \xi_0 = x_0 = a \qquad (4.3)$$

Тогда если в качестве ξ_i выбираются левые точки отрезков, для искомого интеграла можно записать

$$I = \int_a^b f(x)dx \approx f(x_0)h + \ldots + f(x_{i-1})h + \ldots + f(x_{N-1})h = \sum_{i=0}^{N-1} f(x_i)h \qquad (4.4)$$

Если в качестве ξ_i выбираются правые точки отрезков, то по аналогии можно записать

$$I = \int_a^b f(x)dx \approx \sum_{i=1}^{N} f(x_i)h \qquad (4.5)$$

Изложенный метод называется методом прямоугольников. Видно, что искомая площадь под кривой, величина которой есть интеграл, заменяется площадями прямоугольников с погрешностью $R \approx O(h^3)$.

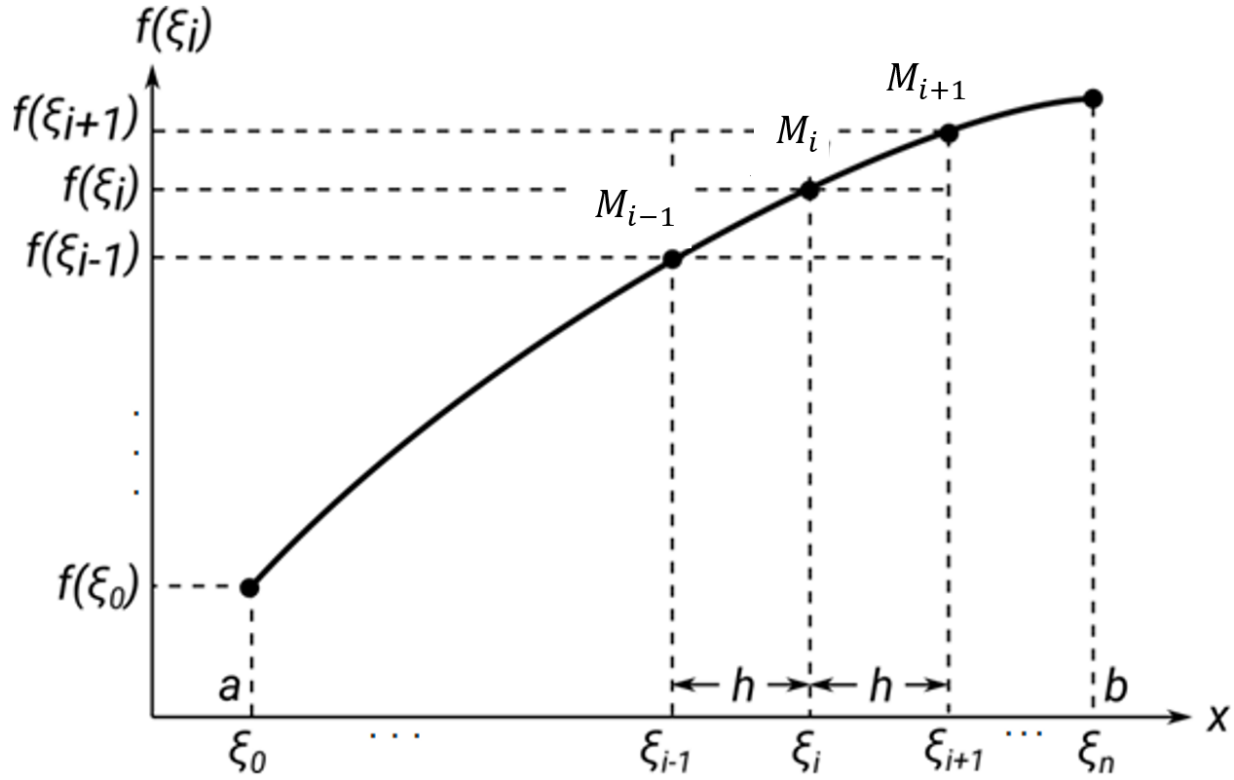

Рисунок 6. Метод прямоугольников

4.2. Метод Симпсона

Здесь используется квадратичная интерполяция на каждом отрезке, то есть $f(x) \approx \phi_i(x) \approx a_i x^2 + b_i x + c_i$; $x_{i-1} \le x \le x_{i+1}$. Коэффициенты находятся из условий равенства многочлена в точках x_i, соответствующих табличным данным y_i.

В качестве $\varphi_i(x)$ можно применить интерполяционный многочлен Лагранжа второй степени, проходящий через три точки M: $M_{i-1}(x_{i-1}, f(x_{i-1}))$, $M_i(x_i, f(x_i))$, $M_{i+1}(x_{i+1}, f(x_{i+1}))$ (см. рис. 6).

$$\phi_i(x) = \frac{(x - x_i)(x - x_{i+1})}{(x_{i+1} - x_i)(x_{i-1} - x_{i+1})} f(x_{i-1}) + \qquad (4.6)$$

$$+\frac{(x-x_i)(x-x_{i+1})}{(x_i-x_{i-1})(x_i-x_{i+1})}f(x_i)$$
$$+\frac{(x-x_{i-1})(x-x_i)}{(x_{i+1}-x_{i-1})(x_{i+1}-x_i)}f(x_{i+1})$$

элементарная площадь S_i может быть вычислена с помощью определенного интеграла, учитывая $x_{i+1} - x_i = x_i - x_{i-1} = h$.

$$S_i = \int_{x_i-1}^{x_i+1} \phi_i(x)dx =$$
$$= \frac{1}{2h^2}\int_{x_i-1}^{x_i+1}[(x-x_i)(x-x_{i+1})f(x_{i-1}) - \quad (4.7)$$
$$-2(x-x_i)(x-x_{i+1})f(x_i)$$
$$+ (x-x_{i-1})(x-x_i)f(x_{i+1})]dx =$$
$$= \frac{h}{3}\left(f(x_{i-1}) + 4f(x_i) + f(x_{i+1})\right)$$

Просуммируем по всем отрезкам и получим

$$\int_a^b f(x)dx \approx \frac{h}{3}\{f(x_0) + 4[f(x_1) + f(x_3) + \ldots + f(x_{N-1})] + \quad (4.8)$$
$$+ 2[f(x_2) + f(x_4) + \ldots + f(x_{N-2})] + f(x_N)\}.$$

Следует отметить, что во многих случаях наибольшую точность расчета интеграла можно получить, если подсчитывать $f(x_i)$, т.е. в середине интервала:

$x_i = \frac{x_{i-1}+x_{i+1}}{2}$; тогда $I = \int_a^b f(x_i) = \sum_{i=1}^{n-1} f(x_i)\, 2h.$ (4.9)

Погрешность метода Симпсона: $R \approx -\frac{h^4}{180}f^{(IV)}(x)$.

В общем случае погрешность численного интегрирования зависит от шага h. Ее можно представить в виде $R_n = 0(h^k)$. При $h \to 0$, численные значения сходятся к точному значению.

Но не всегда можно уменьшить шаг, например, когда подынтегральная функция задана таблично. Тогда имеет смысл увеличивать степень многочлена, используя интерполяционные многочлены (метод Симпсона, сплайны).

О других методах (особые случаи)

Подынтегральная функция или ее производные имеют разрывы на отрезке интегрирования. В этом случае интеграл вычисляют численно на каждом участке непрерывности и результаты складывают.

Например, в случае одной точки разрыва $x = C(a \leq c \leq b)$ имеем

$$\int_a^b f(x)dx = \int_a^c f(x)dx + \int_c^b f(x)dx$$

Несобственные интегралы. Это интегралы, которые имеют хотя бы одну бесконечную границу интегрирования или хотя бы одну точку отрезка, в которой подынтегральная функция образуется в ∞.

Рассмотрим случай бесконечной границы $\int_a^\infty f(x)dx, 0 < x < \infty$. Существует несколько приемов введения замены переменных, например так: $x = \frac{a}{1-t}$, которая превращает интервал интегрирования для $x\ [a, \infty]$ в отрезок $[0, 1]$ для t. $x(1 - t) = a$; $xt = a$; $xt = x - a$; $t = \frac{x-a}{x}$. При этом подынтегральная функция и ее производная для некоторого порядка должна оставаться ограниченной.

Еще один прием: бесконечная граница заменяется некоторым достаточно большим числом, так чтобы принятое значение интеграла отличалось от исходного на малый остаток

$$\int_a^{\infty} f\,(x)dx \;=\; \int_a^{b} f\,(x)dx + R, \qquad \text{где } R \;=\; \int_b^{\infty} f\,(x)dx. \quad (4.10)$$

Если функция $f(x)$ обращается в ∞ в точке $a \le c \le b$, $x = c$, то иногда удается представить: $f(x) = f_1(x) + f_2(x)$, где $f_1(x)$ – ограниченная функция, а $f_2(x)$ обращается в ∞ при x = c, но при этом $\int_a^b f_1\,(x)dx$ – берется. Тогда численный метод используется только для $\int_a^b f_1\,(x)dx$.

Во многих прикладных задачах часто встречается такой вид несобственного интеграла:

$$\int_a^b \frac{\nu(x)}{x-c}\,dx, \text{где } a \le c \le b \qquad (4.11)$$

Видно, что при $x = c$ подынтегральная функция становится бесконечной. Тогда можно окружить точку c симметричной малой окрестностью c + ε, c – ε.

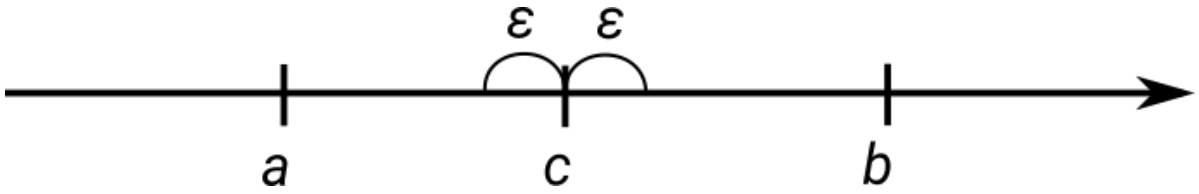

Рисунок 7. Определение интеграла по Коши

Если вырезать эту окрестность из отрезка ab, тогда интеграл в таком смысле называется интегралом в смысле главного значения по Коши, или сингулярным интегралом. Можно построить простую формулу типа прямоугольников для вычисления сингулярных интегралов.

Пусть надо вычислить записанный выше интеграл на отрезке [-1, 1]. Возьмем равноотстоящие точки x_1, x_2, x_3, …, x_n, такие, что точка c делит пополам отрезок между ближайшими точками к ней из этого семейства точек. При этом крайние точки x_1 и x_n лежат на расстоянии не менее полушага от a и b. Тогда

$$\int_{-1}^{+1} \frac{\nu(x)dx}{x-c} \approx \sum_{k=1}^{n} \frac{\nu(x_k)h}{x_k - c} \qquad (4.12)$$

4.3. Кратные интегралы

Ограничимся рассмотрением двойных интегралов:

$$\iint_G f\,(x,y)dxdy \qquad (4.13)$$

Рассмотрим случай, когда G – прямоугольник. Если G – имеет более сложный вид, то с помощью замены переменных ее можно свести к прямоугольному виду либо разбить на простые элементы.

4.3.1. Метод ячеек

Один из простейших методов – метод ячеек (см. рис. 8). Область G разбивается на прямоугольные ячейки ΔG_{ij}; $i = 1, 2, \ldots, M$; $j = 1, 2, \ldots, N$.

По теореме о среднем среднее значение функции по некоторой области ΔG_{ij} равно

$$\overline{f}(x,y) = \frac{1}{\Delta G_{ij}} \iint_{\Delta G} f\,(x,y)dxdy, \qquad (4.14)$$

где $\Delta G_{ij} = \Delta x_i \Delta y_j$

Отсюда

$$\iint_{\Delta G_{ij}} f(x,y)dxdy = \overline{f}(x_i, y_j)\Delta x_i \Delta y_j, \qquad (4.15)$$

но точное среднее значение нам неизвестно, поэтому за среднее значение примем приближенное значение $f(x, y)$ точно в центре прямоугольника-ячейки, то есть $\overline{f}(x_i, y_j) = f(\overline{x_i}, \overline{y_j})$. Тогда получим $\iint_{\Delta G_{ij}} f(x,y)dxdy = \overline{f}(x_i, y_j)\Delta x_i \Delta y_j \approx f(\overline{x_i}, \overline{y_j})\Delta x_i \Delta y_j$. Интегрируя по всем ячейкам, получим

$$\iint_G f(x,y)dxdy \approx \sum_{i=1}^{m}\sum_{j=1}^{n} f(\overline{x_i}, \overline{y_j})\Delta x_i \Delta y_j \qquad (4.16)$$

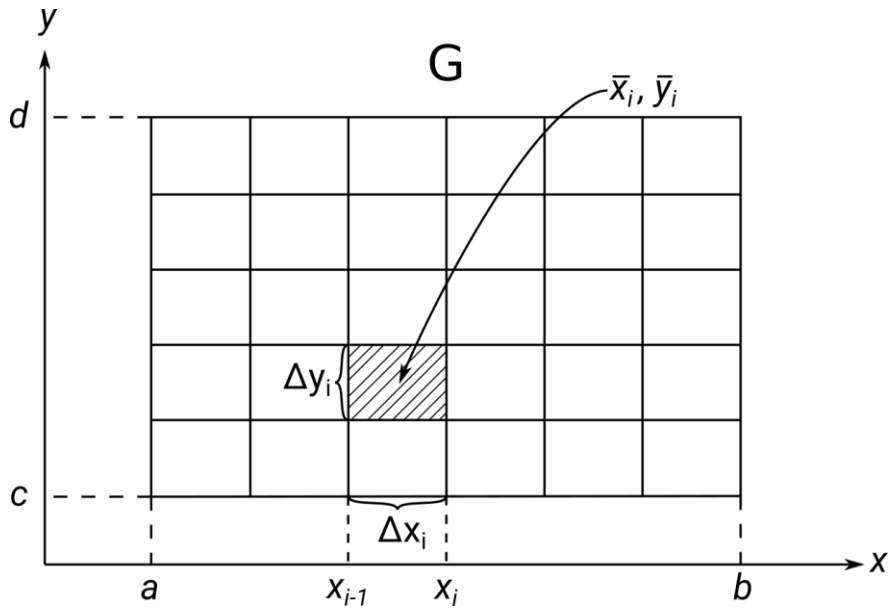

Рисунок 8. Схема метода ячеек

При увеличении числа ячеек эта сумма будет стремиться к значению интеграла для любой непрерывной функции $f(x, y)$.

Можно показать, что погрешность метода для одной ячейки

$$R_{ij} \sim \frac{\Delta x_i \Delta y_j}{24}\left[\left(\frac{b-a}{m}\right)^2 f''_{xx} + \left(\frac{d-c}{n}\right)^2 f''_{yy}\right] \qquad (4.17)$$

Суммируя по всем элементам и считая все их площади одинаковыми, получим оценку метода ячеек (сеток)

$$R \approx 0\left(\frac{1}{m^2} + \frac{1}{n^2}\right) \approx 0(\Delta x^2 + \Delta y^2), \qquad (4.18)$$

то есть формула имеет второй порядок точности.

4.3.2. Метод Монте-Карло

Пусть требуется вычислить определенный интеграл $I = \int_0^1 f(x)dx$. Пусть η – равномерно распределенная на отрезке [0, 1] случайная величина, то есть ее плотность распределения задается соотношением

$$P_\eta(x) = \begin{cases} 0, x < 0 \\ 1, 0 \le x \le 1 \\ 0, x > 0 \end{cases} \qquad (4.19)$$

Тогда любая функция ξ = *f*(η) также будет случайной величиной, а ее математическое ожидание определяется соотношением

$$M[\xi] = \int_{-\infty}^{\infty} f(x)P_\eta(x)dx = \int_0^1 f(x)dx. \qquad (4.20)$$

Читая это равенство в обратном порядке, получим $\int_0^1 f(x)dx = M[\xi]$, то есть интеграл может быть вычислен как математическое ожидание некоторой случайной величины ξ, которая определяется $\xi_i = f(\eta_i)$ независимыми реализациями случайной величины η, распределенной по равномерному закону. Иными словами:

$$\int_0^1 f\,(x)dx\; = \widehat{M}[\xi]\; = \frac{1}{n}\sum_{i=1}^{n}\xi_i\; = \frac{1}{\mathrm{n}}\sum_{i=1}^{n} f\,(\eta_i). \qquad (4.21)$$

Аналогично, для двойного интеграла

$$\iint_G f\,(x,y)dxdy\; \approx\; \frac{1}{n}\sum_{i=1}^{n} f\,(\eta_i,\xi_i), \qquad (4.22)$$

где G: $0 \le x \le 1$, $0 \le y \le 1$, η_i, ξ_i – независимые реализации случайных величин ξ и η, равномерно распределенных на отрезке [0, 1]. Обычно в ЭВМ хранится некоторый алгоритм выработки случайных величин с заданным законом распределения. Поскольку алгоритм задан наперед, то эти числа не совсем случайны, а псевдослучайны, хотя и обладают статическими характеристиками случайных величин.

5. ЧИСЛЕННОЕ ДИФФЕРЕНЦИРОВАНИЕ

Известно, что производная функции $f(x)$ – есть предел

$$y' = f'(x) = \lim_{\Delta x \to 0} \frac{\Delta y}{\Delta x}, \qquad \text{где } \Delta y = f(x + \Delta x) - f(x) \qquad (5.1)$$

Обычно для вычисления производных используются соответствующие формулы, однако это при применении ЭВМ не всегда удобно и возможно. Поэтому используют так называемую разностную аппроксимацию $y \approx \frac{\Delta y}{\Delta x}$ где Δy, Δx имеют конечные значения.

Пусть функция $y = f(x)$ задана в табличном виде.

y	y_0	y_1	y_2	…	y_i
x	x_0	x_1	x_2	…	x_i

Пусть шаг по x равен h.

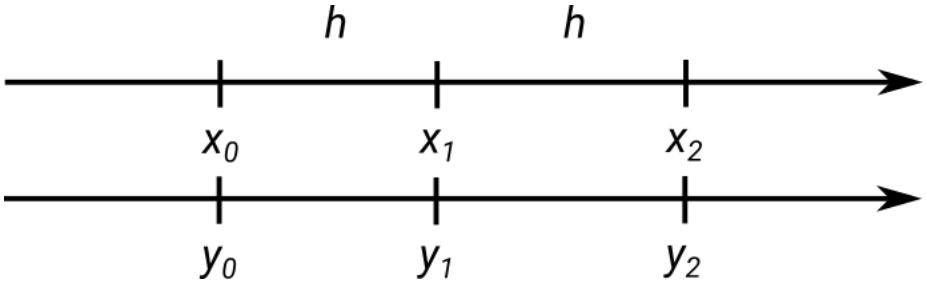

Рисунок 9. Дифференцируемая функция

Запишем выражение для производной y_1 при $x = x_1$. Здесь могут быть различные способы:

левая разность (аппроксимация) $y_1' \approx \frac{y_1 - y_0}{h}$;

правая разность (аппроксимация) $y_1' \approx \frac{y_2 - y_1}{h}$;

центральная разность (аппроксимация) $y_1' \approx \frac{y_2 - y_0}{2h}$.

Найдем вторую производную при $x = x_1$

$$y_1'' \approx (y_1')' \approx \frac{y'_{1\text{прав}} - y'_{1\text{лев}}}{h} \approx \frac{\frac{(y_2 - y_1)}{h} - \frac{(y_1 - y_0)}{h}}{h} = \frac{y_2 - 2y_1 + y_0}{h^2} \quad (5.2)$$

Встает вопрос о точности аппроксимаций.

5.1. Погрешность численного дифференцирования

Пусть $f^k(x)$ – производная k-го порядка, $\varphi^k(x)$ – приближенная, т. е. аппроксимирующая производную функция. Тогда функция, характеризующая отклонение приближенного значения производной от ее истинного значения, называется *погрешностью аппроксимации производной:*

$$R^k(x) = f^{(k)}(x) - \varphi^k(x);\ R^{(k)}_{(x)} = O(h^p) \quad (5.3)$$

При численном дифференцировании функции эта погрешность зависит от шага h и ее записывают в виде $O(h^p)$, где «p» называется порядком погрешности аппроксимации (иногда порядком точности). Предполагается, что $|h| < 1$.

Для оценки погрешности используем ряд Тейлора:

$$f(x + \Delta x) = f(x) + f'(x)\Delta x + f''(x)\left(\frac{\Delta x^2}{2!}\right) + f'''(x)\left(\frac{\Delta x^3}{3!}\right) \quad (5.4)$$

Пусть функция $f(x)$ задана в виде таблицы $f(x_i) = y_i$ (i = 0, 1, …, N). Запишем ряд Тейлора при $x = x_1$, $\Delta x = -h$ с точностью до членов порядка h:

$$f(x_1 - h) = f(x_0) = y_0 = y_1 - y_1' h + O(h^2) \quad (5.5)$$

Отсюда найдем значения производной в точке $x = x_1$

$$y_1' \approx \frac{y_1 - y_0}{h} + O(h) \quad (5.6)$$

Это выражение совпадает с формулой для левой аппроксимации, которая, как видно, является аппроксимацией первого порядка.

$$y_1' \approx \frac{y_1 - y_0}{h} \quad (5.7)$$

Аналогично

$$f(x_1 + h) = f(x_2) = y_2 = y_1 + y_1' h + O(h^2)$$
$$y_1' = \frac{y_2 - y_1}{h} + O(h), \quad (5.8)$$

что совпадает с формулой для правой аппроксимации

$$y_1' \approx \frac{y_2 - y_1}{h} + O(h), \quad (5.9)$$

которая тоже имеет первый порядок погрешности.

Для оценки погрешности центральной аппроксимации первой производной и оценки погрешности второй производной положим $\Delta x = h$, $\Delta x = -h$. Соответственно:

$$\begin{cases} f(x_1 + h) = y_2 = y_1 + y_1' h + \dfrac{y_1'' h^2}{2!} + \dfrac{y_1''' h^3}{3!} + O(h^4) \\ f(x_1 - h) = y_0 = y_1 - y_1' h + \dfrac{y_1'' h^2}{2!} - \dfrac{y_1''' h^3}{3!} + O(h^4) \end{cases} \quad (5.10)$$

Вычитая, получим

$$y_2 - y_0 = 2y_1' h + 2\frac{y''' h^3}{3!};$$

$$y_1' = \frac{y_2 - y_0}{2h} - \frac{y''' h^3}{2h \cdot 3!}; \quad (5.11)$$

$$y_1' = \frac{y_2 - y_0}{2h} + O(h^2)$$

Таким образом, центральная аппроксимация первой производной имеет второй порядок точности.

Складывая, получим

$$y_2 + y_0 = 2y_1 + 2\frac{y_1'' h^2}{2!} + 2O(h^4), \quad (5.12)$$

преобразуем и поделим на h^2

$$\frac{y_2 - 2y_1 + y_0}{h^2} = y_1'' + \frac{2 \cdot O(h^4)}{h^2};$$

$$y_1'' = \frac{y_2 - 2y_1 + y_0}{h^2} - 2 \cdot O(h^2) \quad (5.13)$$

Таким образом, вторая производная имеет второй порядок точности. $O(h^p)$ – остаточный член. Его анализ нетривиален. Однако при уменьшении h погрешность аппроксимации, как правило, уменьшается. Вместе с тем с уменьшением h погрешность округления возрастает, поэтому суммарная погрешность при уменьшении h может убывать лишь до некоторого предельного значения, после чего дальнейшее уменьшение шага не повысит точность результатов.

Для получения оптимальной точности используют так называемую *регуляризацию* процедуры численного дифференцирования.

Простейший способ: выбирается такое h, что $|f(x+h) - f(x)| > \varepsilon$, где ε – некоторое малое число. При вычислении производной это исключает вычитание близких по величине чисел, что приводит обычно к увеличению погрешности.

Другой способ: сглаживание табличных данных подбором гладкой аппроксимирующей функции, например многочлена.

5.2. Метод неопределенных коэффициентов

Аналогичные формулы можно получить и для *случая произвольного расположения узлов*. Использование многочлена Лагранжа в этом случае приводит к вычислению громоздких выражений, поэтому удобнее применять *метод неопределенных коэффициентов.* Он заключается в следующем.

Искомое выражение для производной k-го порядка в некоторой точке $x = x_i$ представляется в виде линейной комбинации заданных значений функции в узлах $x_0, x_1, \ldots, x_n$:

$$y_i^{(k)} \approx c_0 y_0 + c_1 y_1 + c_2 y_2 + \ldots + c_n y_n. \qquad (5.14)$$

Предполагается, что это соотношение выполняется точно, если функция y является многочленом степени не выше n, т. е. может быть представлена в виде

$$y = b_0 + b_1(x - x_0) + \ldots + b_n(x - x_0)^n. \qquad (5.15)$$

Отсюда следует, что соотношение (1), в частности, должно выполняться точно для многочленов $y = 1$, $y = x - x_0$, …, $y = (x - x_0)^n$. Подставляя последовательно эти выражения в (1) и требуя выполнения точного равенства, получаем систему $n + 1$ линейных алгебраических уравнений для определения неизвестных коэффициентов $c_0, c_1, \ldots, c_n$.

Пример. Найти выражение для производной y_1' в случае четырех равноотстоящих узлов ($n = 3$). Приближение (1) запишется в виде

$$y_1' \approx c_0 y_0 + c_1 y_1 + c_2 y_2 + c_3 y_3. \qquad (5.16)$$

Используем следующие многочлены:

$$y = 1,\ y = x - x_0,\ y = (x - x_0)^2,\ y = (x - x_0)^3. \qquad (5.17)$$

Вычислим их производные:

$$y' = 0,\ y' = 1,\ y' = 2(x - x_0),\ y' = 3(x - x_0)^2 \qquad (5.18)$$

Подставляем последовательно соотношения (3) и (4) соответственно в правую и левую части (2) при $x = x_1$, требуя выполнения точного равенства:

$$0 = c_0 \cdot 1 + c_1 \cdot 1 + c_2 \cdot 1 + c_3 \cdot 1;$$
$$1 = c_0(x_0 - x_0) + c_1(x_1 - x_0) + c_2(x_2 - x_0) + c_3(x_3 - x_0)$$
$$2(x_1 - x_0) = c_0(x_0 - x_0)^2 + c_1(x_1 - x_0)^2 + c_2(x_2 - x_0)^2 + c_3(x_3 - x_0)^2;$$
$$3(x_1 - x_0)^2 = c_0(x_0 - x_0)^3 + c_1(x_1 - x_0)^3 + c_2(x_2 - x_0)^3 + c_3(x_3 - x_0)^3.$$

Получаем окончательно систему уравнений в виде

$$\begin{aligned} c_0 + c_1 + c_2 + c_3 &= 0, \\ hc_1 + 2hc_2 + 3hc_3 &= 1, \\ hc_1 + 4hc_2 + 9hc_3 &= 2, \\ hc_1 + 8hc_2 + 27hc_3 &= 3. \end{aligned} \qquad (5.19)$$

Решая эту систему, получаем

$$c_0 = -\frac{1}{3h},\quad c_1 = -\frac{1}{2h},\quad c_2 = \frac{1}{h},\quad c_3 = -\frac{1}{6h}. \qquad (5.20)$$

Подставляя эти значения в (5.19), находим выражение для производной:

$$y_1' \approx \frac{1}{6h}(-2y_0 - 3y_1 + 6y_2 - y_3). \qquad (5.21)$$

Численное дифференцирование функции многих переменных (частные производные)

Для простоты рассмотрим функцию двух переменных: $u = u(x, y)$. Пусть по данным нижеприведенной таблицы нужно найти частные производные:

$$\frac{\partial u}{\partial x};\ \frac{\partial u}{\partial y};\ \frac{\partial^2 u}{\partial x^2};\ \frac{\partial^2 u}{\partial y^2};\ \frac{\partial^2 u}{\partial x \partial y}. \qquad (5.22)$$

yx	x_0	x_1	⋯	x_{i-1}	x_i	x_{i+1}	⋯	x_n
y_0	u_{00}	u_{10}	⋯	$u_{i-1,0}$	$u_{i,0}$	$u_{i+1,0}$	⋯	$u_{n,0}$
y_1	u_{01}	u_{11}	⋯	$u_{i-1,1}$	$u_{i,1}$	$u_{i+1,1}$	⋯	$u_{n,1}$
⋮	⋮	⋮	⋯	⋮	⋮	⋮	⋯	⋮
y_{j-1}	$u_{0,j-1}$	$u_{1,j-1}$	⋯	$u_{i-1,j-1}$	$u_{i,j-1}$	$u_{i+1,j-1}$	⋯	$u_{n,j-1}$
y_j	$u_{0,j}$	$u_{1,j}$	⋯	$u_{i-1,j}$	$u_{i,j}$	$u_{i+1,j}$	⋯	$u_{n,j}$
y_{j+1}	$u_{0,j+1}$	$u_{1,j+1}$	⋯	$u_{i-1,j+1}$	$u_{i,j+1}$	$u_{i+1,j+1}$	⋯	$u_{n,j+1}$
⋮	⋮	⋮	⋯	⋮	⋮	⋮	⋯	⋮
y_n	$u_{0,n}$	$u_{1,n}$	⋯	$u_{i-1,n}$	$u_{i,n}$	$u_{i+1,n}$	⋯	$u_{n,n}$

В качестве примера рассмотрим частные производные в точке (x_i, y_i). Для простоты положим $x_{i+1} - x_i = y_{j+1} - y_j = h$; тогда по

аналогии с численным дифференцированием в случае одной переменной можно записать

$\left(\frac{\partial u}{\partial x}\right)_{ij} = \frac{u_{i+1,j} - u_{i-1,j}}{2h}$ – центральная аппроксимация;

$\left(\frac{\partial u}{\partial x}\right)_{ij} = \frac{u_{i+1,j} - u_{i,j}}{h}$ – правая аппроксимация;

$\left(\frac{\partial u}{\partial x}\right)_{ij} = \frac{u_{i,j} - u_{i-1,j}}{h}$ – левая аппроксимация;

$\left(\frac{\partial^2 u}{\partial x^2}\right)_{ij} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2}$ – аппроксимация второй производной;

$\left(\frac{\partial u}{\partial y}\right)_{ij} = \frac{u_{i,j+1} - u_{i,j-1}}{2h}$ – центральная аппроксимация;

$\left(\frac{\partial u}{\partial y}\right)_{ij} = \frac{u_{i,j+1} - u_{i,j}}{h}$ – правая аппроксимация;

$\left(\frac{\partial u}{\partial y}\right)_{ij} = \frac{u_{i,j} - u_{i,j-1}}{h}$ – левая аппроксимация;

$\left(\frac{\partial^2 u}{\partial y^2}\right)_{ij} = \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h^2}$ – аппроксимация второй производной;

$\left(\frac{\partial^2 u}{\partial x \partial y}\right)_{ij} = \frac{u_{i+1,j+1} - u_{i+1,j-1} - u_{i-1,j+1} + u_{i-1,j-1}}{4h^2}$ – аппроксимация второй производственной.

5.3. Метод Рунге – Ромберга

Пусть $F(x)$ – истинная производная, которую мы должны аппроксимировать; пусть $f(x, h)$ – конечно-разностная аппроксимация этой производной по данным таблицы, в которой $x_1, x_2, \ldots, x_n$ расположены с одинаковым шагом h, т. е. $x_2 - x_1 = x_{i+1} - x_i = h$.

Формула Рунге – Ромберга дана

$$F(x) = f(x,h) + \frac{f(x,h) - f(x,kh)}{k^p - 1} + O(h^{p+1}). \qquad (5.23)$$

Эта формула позволяет по данным одной и той же таблицы по результатам двух расчетов значений производной $f(x, h)$ и $f(x, kh)$ с порядком точности «p» найти ее уточненное значение с порядком точности $(p + 1)$.

Пример. Вычислить производную функции $y = x^3$ в точке $x = 1$. Очевидно, что $y' = 3x^2$; поэтому $y'(1) = 3$. Найдем теперь эту производную численно. Составим таблицу значений функции:

x	0,8	0,9	1,0
y	0,512	0,729	1,0

Воспользуемся аппроксимацией производной с помощью левых разностей, имеющей первый порядок ($p = 1$). Примем шаг равным 0,1 и 0,2, т. е. $k = 2$. Получим

$$f(x,h) = y'(1, 0.1) = \frac{f(1) - f(0,9)}{0,1} = \frac{1 - 0,729}{0,1} = 2,71,$$

$$f(x,kh) = y'(1, 0.2) = \frac{f(1) - f(0,8)}{0,2} = \frac{1 - 0,512}{0,2} = 2,44.$$

По формуле Рунге найдем уточненное значение производной:

$$F(x) = y'(1) \approx 2,71 + \frac{2,71-2,44}{2^1-1} = 2,98.$$

Таким образом, формула Рунге дает более точное значение производной. В общем случае порядок точности аппроксимации увеличивается на единицу.

6. ОПТИМИЗАЦИЯ (В НАУКЕ И ТЕХНИКЕ)

Теория оптимизации представляет собой совокупность фундаментальных математических результатов и численных методов, ориентированных на нахождение и идентификацию наилучших вариантов из множества возможных и позволяющих избежать полного перебора этих вариантов.

Постановка задачи оптимизации – ключ к успеху оптимизации; во многом определяется искусством исследователя. Здесь несколько этапов:

– определение границ изучаемой среды (объекта), необходимое для выделения объекта из окружающей среды;

– выбор характеристического критерия (критерия оптимизации), на основе которого и выбираются наилучшие (в некотором смысле) варианты.

Независимо от того, какой критерий выбирается, *наилучшему* варианту обычно соответствует *минимальное* или *максимальное* значение критерия.

Важно отметить, что независимо от содержания и сути методов оптимизации, мы будем рассматривать случаи *только одного критерия.* Дело в том, что, как правило, невозможно получить решение, которое бы одновременно удовлетворяло всем возможным критериям. Например, нельзя получить решение которое бы одновременно обеспечивало максимум надежности и в то же время минимум затрат, минимум потребляемой энергии.

Конечно, на практике часто существует сразу несколько критериев – тогда какой-либо из критериев считается главным, а остальные – второстепенными. Главный критерий используется как характеристическая мера, а для остальных устанавливаются границы, из которых они не должны выходить при оптимизации главного критерия.

Вместе с тем существуют подходы, позволяющие сводить многокритериальную задачу к однокритериальной путем использования некоторого обобщенного критерия.

Выбор независимых переменных. Нужно выбирать те переменные, которые оказывают существенное влияние на характеристический критерий и не перегружать задачу большим количеством несущественных деталей.

Модель системы. Модель представляет некоторый набор управлений, который определяет взаимосвязь между переменными системы и ограничивают область допустимых изменений переменных.

Типичная постановка задачи. Минимизировать вещественнозначную функцию $f(\mathrm{x})$, где x – N-мерный векторный аргумент $x = (x_1, x_2, \ldots, x_N)$, при ограничениях:

$h_k(x) = 0$, $k = 1, 2, \ldots, K$ (ограничение в виде уравнений);

$g_j(x) \geq 0$, $j = 1, 2, \ldots, J$ (ограничение в виде неравенств);

$x_i^{(\ell)} \geq x_i \geq x_i^{(L)}$, $i = 1, 2, \ldots, N$ (ограничение значений переменных сверху и снизу).

Такая постановка называется задачей оптимизации с ограничениями или задачей условной оптимизации.

Задача, в которой нет ограничений, называется оптимизационной задачей без ограничений или задачей безусловной оптимизации. Для таких задач $J = K = 0$ и $-\infty \leq x_i \leq \infty$.

Если задача условной оптимизации содержит линейные функции h_k и g_j, то это задача с линейными ограничениями. В таких задачах целевая функция $f(x)$ может быть линейной – тогда это задача линейного программирования (частный случай – целочисленное программирование – целые значения x).

Если $f(x)$ нелинейная, то это задача нелинейного программирования с линейными ограничениями. При этом если f(x) – квадратичная функция, то задача квадратичного программирования, если f(x) – отношение линейных функций, то дробно-линейного программирования.

Такая классификация определяет соответствующие методы.

6.1. Одномерная оптимизация

Формулируется в общем случае так: найти наименьшее (или наибольшее) значение целевой функции y = f(x), заданной на множестве δ и определить значение x ∈ δ, при котором целевая функция принимает экстремальное значение. Существование решения этой задачи вытекает из *теоремы Вейерштрасса.*

Всякая функция f(x), непрерывная на отрезке [a, b], принимает на этом отрезке наименьшее и наибольшее значения, то есть на отрезке [a, b] существуют такие точки x_1 и x_2, что для любого x ∈ [a, b] имеют место неравенства $f(x_1) \le f(x) \le f(x_2)$.

Эта теорема не доказывает единственности решения. Вполне возможно, когда равные экстремальные значения могут быть достигнуты сразу в нескольких точках данного отрезка (например, если функция периодическая на отрезке).

Рассмотрим некоторые методы оптимизации.

6.2. Аналитические методы оптимизации

Класс дифференцируемой на отрезке [a, b] целевой функции f(x), *заданной аналитически:* y = $f(x)$, для которой может быть найдено явное выражение для производной y' = $f'(x)$.

Кратко опишем суть метода: вычисляются значения f(x) в граничных точках отрезка: f(a), $f(b)$. Находятся так называемые критические точки, то есть значения x, в которых f '(x) обращается в ноль. Как известно, это экстремальные точки, соответствующие *max* или *min*. *Вычисляются значения f(x) в критических точках.* Из значений $f(x)$ в критических и граничных точках выбирают наибольшее (или наименьшее) искомое значение.

Пример: $f(x) = \frac{x^3}{3} - x^2$ на отрезке [1, 3]. Находим $f(x)$ на границах.

$$f(1) = \frac{1}{3} - 1 = -\frac{2}{3};\ f(3) = \frac{3^3}{3} - 9 = 0.$$

Находим критические точки: $f'(x) = \frac{3x^2}{3} - 2x = 0;\ x^2 - 2x = 0;\ x_1 = 0;\ x_2 = 2.$

Вычисляем $f(0) = 0;\ f(2) = -\frac{4}{3}.$

Из сравнения видим, что $f_{max}(x) = f(3) = 0;\ f_{min}(x) = f(2) = -\frac{4}{3}.$

Здесь f '(x) – простая функция. Для более сложных видов f '(x) необходимо использовать численные методы решения нелинейных уравнений.

6.3. Численные методы поиска

Численные методы поиска могут использоваться, если f(x) задана таблично или имеет очень сложный вид, затрудняющий нахождение критических точек.

Наиболее простой метод – это метод прямого перебора (или сканирования). Заключается в том, что отрезок [a, b] сразу разбивается на заведомо большое число отрезков и в каждом таком отрезке вычисляется значение функции на концах отрезка.

Метод хорош тем, что он обязательно выявит все максимальные и минимальные значения функции и, естественно, значения x, которым они соответствуют. То есть это *глобальный метод* поиска, который может определить и

глобальные экстремальные значения. Не требуется предположение об унимодальности.

Недостаток очевиден: могут быть неприемлемо большие затраты машинного времени в случае сложной функции.

Метод перебора с допущением об унимодальности функции, то есть о существовании только одного минимума (или максимума). Кстати на практике такой случай довольно часто встречается. Суть метода заключается в последовательном сужении так называемого *интервала неопределенности,* то есть интервала, в котором находится искомое экстремальное значение.

Пусть, $f(x)$ задана на [a, b]. В начале процесса оптимизации длина интервала неопределенности равна (b-a), а к концу процесса оптимизации должна стать менее заданного допустимого значения ε. То есть оптимальное значение x должно находиться в интервале неопределенности – отрезке длиной $x_{n+1} - x_n < \varepsilon$. Конечно, можно было бы сразу разделить интервал [a, b] на число отрезков, равное $\frac{b-a}{\varepsilon/2}$, однако это будет, по сути дела, полный перебор. Можно действовать экономичнее, а именно разделить отрезок (b-a) много раз, например, в 10 меньшее число отрезков. После выяснения, в какой паре отрезков находится *min* (*max*), можно провести новое разбиение.

Поясним на примере. Пусть начальная длина интервала равна a – b = 1. Нужно добиться его уменьшения в 100 раз, то

есть a – b = 0,01. Этого можно достичь разбиением интервала на 200 частей:

$$x_i - x_{x-1} = 0{,}005; \quad x_{i+1} - x_i = 0{,}005; \text{ то есть } x_{i+1} - x_{i-1} = 0{,}01.$$

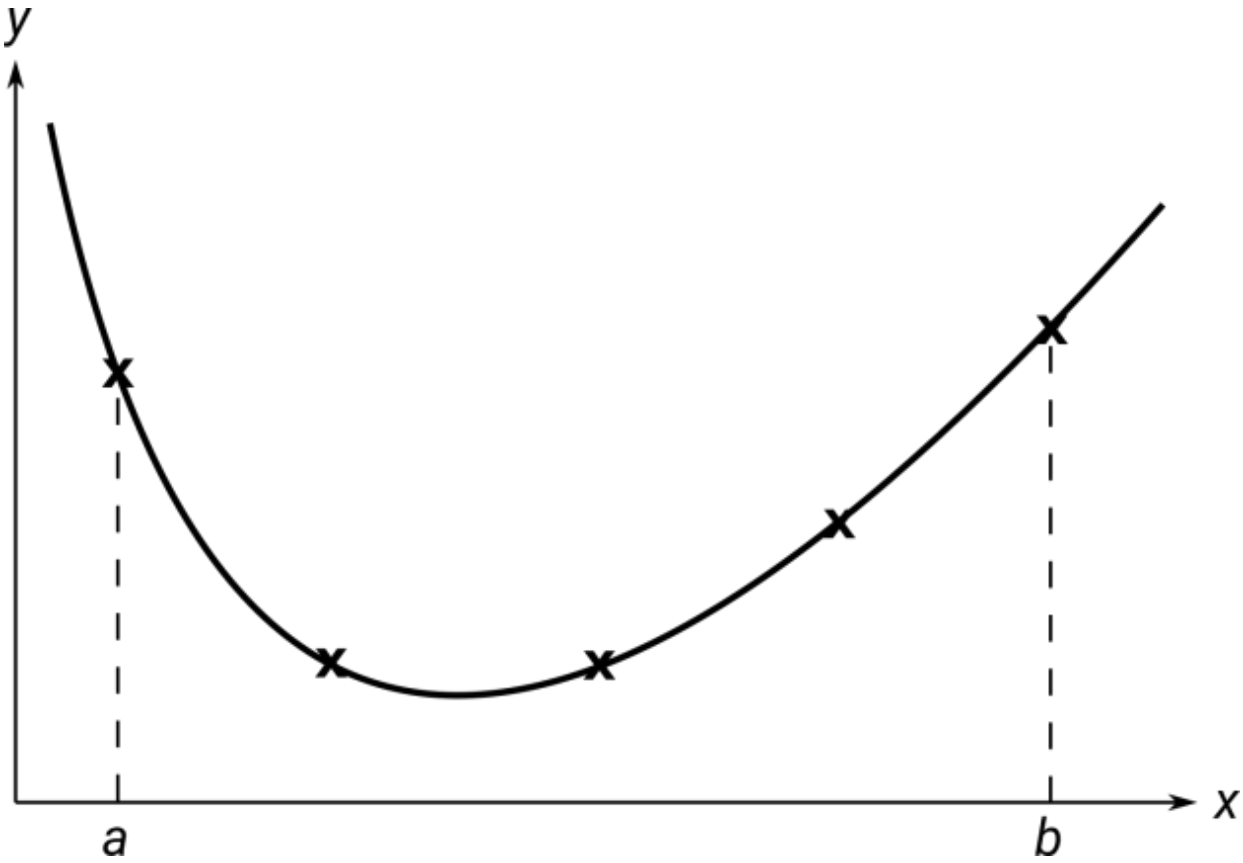

Рисунок 10. Пример метода перебора

Однако можно поступить иначе. Сначала разобьем [a, b] на 20 частей и найдем интервал неопределенности длиной 0,1. При этом мы вычислим значения целевой функции в точках $x_k = a + 0{,}05k \ \ (k = 0{,}1, \ldots, 20)$. Теперь отрезок [$x_{i-1}$, x_{i+1}] снова разобьем на 20 частей, получим искомый интервал неопределенности длиной

0,01, причем значения целевой функции вычисляем в точках $x_k = x_{i-1} + 0{,}005k$ $(k = 0{,}1, \ldots, 19)$. В точках x_{i-1} и x_{i+1} значения $f(x)$ уже найдены. В этом случае мы производим 40 вычислений против 201 в первом случае. Таким образом, путем определенной изобретательности можно сэкономить на вычислениях. Можно использовать и другие идеи: метод деления отрезка пополам, метод золотого сечения и так далее.

6.4. Многомерная оптимизация

6.4.1. Аналитический метод поиска

Минимум дифференцируемой функции многих переменных $f(x_1, x_2, \ldots, x_n)$ можно найти, исследуя ее значение в критических точках, которые определяются из решения системы дифференциальных уравнений:

$$\frac{\partial f}{\partial x_1} = 0;\ \frac{\partial f}{\partial x_2} = 0;\ \ldots;\ \frac{\partial f}{\partial x_n} = 0. \qquad (6.1)$$

Пример. Необходимо спроектировать железный контейнер в форме параллелепипеда объемом V = 1 м^3, причем чтобы *материала израсходовать как можно меньше.* Если толщина стенок задана, то последнее требование означает, что целевой функцией здесь будет площадь полной поверхности контейнера: $f(x) = S$. Пусть x_1, x_2, x_3 – длины ребер контейнера, тогда задача сведется к минимизации функции

$$S = 2(x_1x_2 + x_1x_3 + x_2x_3). \qquad (6.2)$$

При этом условие $V = 1$, будет условием-ограничением (равенством), которое позволяет исключить один параметр

$$V = x_1 x_2 x_3 = 1; \quad x_3 = \frac{1}{x_1 x_2}; \quad \text{тогда } S = 2\left(x_1 x_2 + \frac{1}{x_2} + \frac{1}{x_1}\right). \tag{6.3}$$

Задачу можно усложнить, например потребовать, чтобы контейнер имел длину не менее 2 метра, тогда появится дополнительное ограничение – неравенство $x_1 \geq 2$.

В соответствии с $\frac{\partial f}{\partial x_1} = 0$ получим

$$\frac{\partial S}{\partial x_1} = 2\left(x_2 - \frac{1}{x_1^2}\right) = 0;$$

$$\frac{\partial S}{\partial x_2} = 2\left(x_1 - \frac{1}{x_2^2}\right) = 0, \quad x_2 x_1^2 = 1; \quad x_1 x_2^2 = 1; \tag{6.4}$$

$$\frac{x_2 x_1^2}{x_1 x_2^2} = 1; \; \frac{x_1}{x_2} = 1; \; x_1 = x_2; x_1^3 = 1; x_1 = x_2 = x_3 = 1 \text{ м},$$

то есть куб со стороной 1 метр.

Этот метод можно использовать лишь для дифференцируемой целевой функции, кроме того, можно прийти к системе сложных нелинейных уравнений. Поэтому целесообразно рассматривать численные методы поиска.

6.4.2. Метод полного перебора

Каждый из интервалов x_1, x_2, …, x_N разбивается на отрезки с шагом h_1, h_2, …, h_N, так что мы получаем многомерную сетку с узлами. Вычисляем значения функции $f(x)$ в каждом узле и выбираем наименьшее (или наибольшее) значение.

Пусть n = 5 и число узлов по каждой переменной – 100. Тогда общее количество узлов $100^5 = 10^{10}$. Пусть быстродействие равно 10^6/с, тогда время счета ~10^4 с = 2.5 часа, где N – число операций для вычисления одного значения. 10 операций = 25 часов, т. е. этот метод малоэкономичен.

6.4.3. Метод покоординатного спуска

Пусть требуется найти наименьшее значение целевой функции $u = f(x_1, x_2, \ldots, x_N)$. В качестве начального приближения выберем в n-мерном пространстве некоторую точку M_0 с координатами $x_1^0, x_2^0, \ldots, x_N^0$. Зафиксируйте все координаты кроме первой, тогда $u = f(x_1, x_2^0, \ldots, x_N^0)$ – одномерная функция. Решая одномерную задачу оптимизации найдем $x_1^{(1)}$, в которой f принимает минимальное значение, то есть переходим к точке $M_1(x_1^1, x_2^0, \ldots, x_N^0)$ – это первый шаг, который состоял в спуске по координате x_1.

Зафиксируем теперь все координаты, кроме x_2, и рассмотрим функцию этой переменной $u = f\left(x_1^{(1)}, x_2, x_3^0, \ldots, x_N^0\right)$. Снова решая одномерную задачу оптимизации, находим ее наименьшее значение при $x_2 = x_2^1$ и переходим в точку $M_2\left(x_1^{(1)}, x_2^{(1)}, x_3^{(0)} \ldots, x_N^{(0)}\right)$ – это *второй шаг*, то есть осуществим спуск по координате x_2 и т. д. Осуществляется спуск по координатам x_3, x_4, …, x_N.

В результате процесса получается последовательность точек $M_0, M_1, M_2, \ldots, M_N$ в которых целевая функция монотонно убывает $f(M_0) \geq f(M_1) \geq f(M_2) \geq \ldots \geq f(M_N)$. На любом шаге k процесс можно прервать, если полученное значение f удовлетворяет. Метод легко пояснить графически для случая двух переменных x_1, x_2.

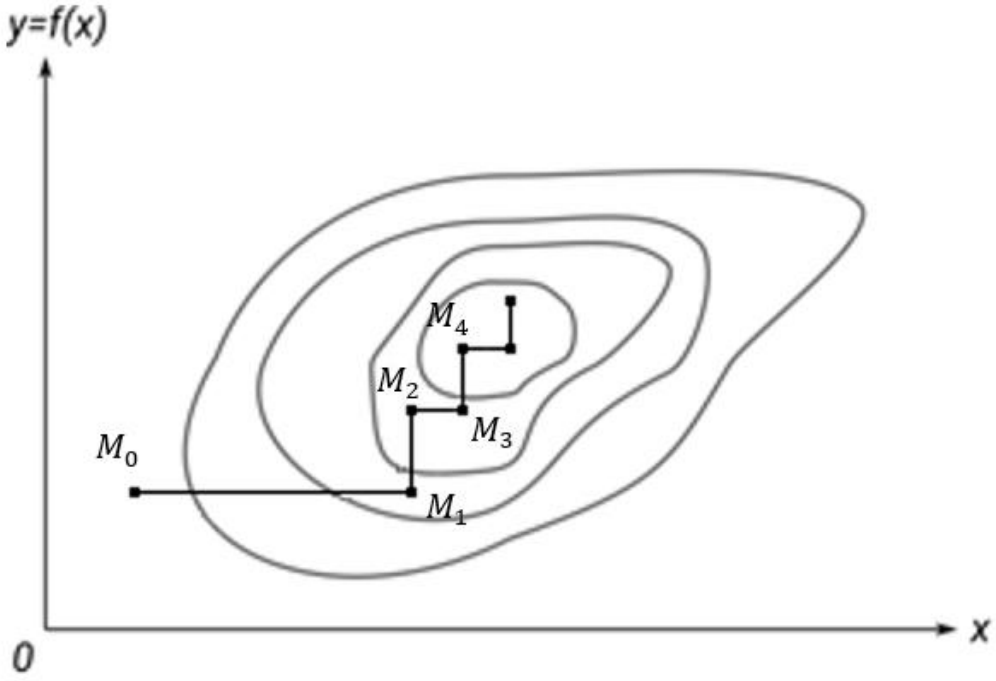

Рисунок 11. Метод покоординатного спуска

Важен вопрос сходимости к наименьшему значению f в данной области. Это зависит от вида самой функции и выбора начального приближения. Например, метод не сработает, если есть овраги на поверхности целевой функции. Допустим, спуск по одной из переменных привел на дно оврага, тогда любое движение вдоль другой координаты приведет к возрастанию функции, соответствующему подъему на склон оврага. Овраги встречаются нередко. Выход: можно сделать большой шаг, чтобы выйти из оврага.

После того, как нашли наименьшее значение, перебрав все координаты, можно начать новый цикл по координатам спуска. То есть опять менять x_1, затем x_2.

6.4.4. Аналитический метод многомерной оптимизации по Лагранжу

Пусть задан критерий оптимизации (целевая функция) $f(x_1, x_2, \ldots, x_N)$, максимум (или минимум) которой нужно найти, т. е. найти соответствующие значения переменных $x_1, x_2, \ldots, x_N$. При этом задано ограничение в виде равенства: $F(x_1, x_2, \ldots, x_N) = 0$.

Строится функция Лагранжа в виде

$$\Phi(x_1, x_2, \ldots, x_N, \lambda) = f(x_1, x_2, \ldots, x_N) + \lambda \cdot F(x_1, x_2, \ldots, x_N), \quad (6.5)$$

где λ – константа.

Т. к. $F(x_1, x_2, \ldots, x_N) = 0$, то ясно, что экстремум функции Ф $(x_1, x_2, \ldots, x_N, \lambda)$ будет совпадать с экстремумом (максимум или минимум функции $f(x_1, x_2, \ldots, x_N)$). Следовательно, поиск экстремума функции f можем заменить поиском экстремума функции Ф.

Для этого нужно приравнять нулю (как мы это делали ранее) первые производные, т. е.

$$\frac{\partial \Phi}{\partial \lambda} = 0; \quad \frac{\partial \Phi}{\partial x_1} = 0; \quad \frac{\partial \Phi}{\partial x_2} = 0; \quad \ldots; \quad \frac{\partial \Phi}{\partial x_n} = 0. \quad (6.6)$$

Решение этой системы уравнений и даст значения $x_1, x_2, \ldots, x_N$, обеспечивающие экстремум функции $f(x_1, x_2, \ldots, x_N)$.

Рассмотрим пример, когда функция (критерий оптимизации) зависит от двух переменных (x, y). Необходимо найти

прямоугольник наибольшей площади, вписанный в круг радиуса R (см. рис. 12).

Из рисунка ясно, что площадь вписанного прямоугольника будет состоять из площадей четырех одинаковых прямоугольников, т. е. критерий оптимизации: $f = S = 4xy$, т. е. функция F запишется так:$F(x, y) = x^2 + y^2 – R^2 = 0$ – уравнение окружности. Тогда функция Лагранжа запишется так: $\Phi(x, y, \lambda) = 4xy + \lambda(x^2 + y^2 – R^2)$.

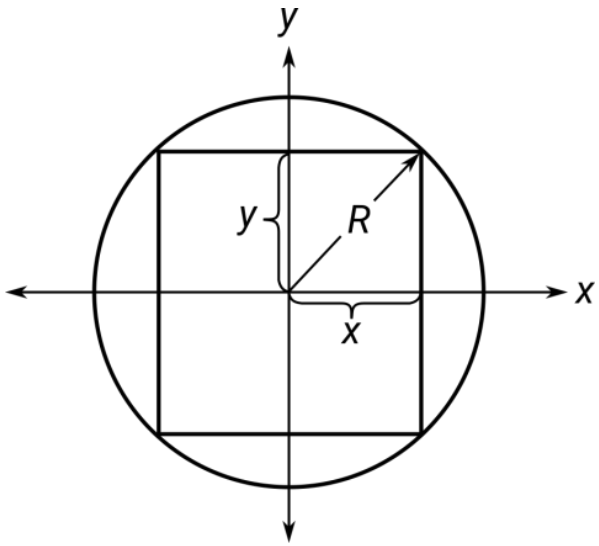

Рисунок 12. Аналитический метод многомерной оптимизации по Лагранжу

Соответственно:

$$\begin{aligned} \frac{\partial \Phi}{\partial \lambda} &= x^2 + y^2 - R^2 = 0, \\ \frac{\partial \Phi}{\partial x} &= 4y + 2\lambda x = 0, \\ \frac{\partial \Phi}{\partial y} &= 4x + 2\lambda y = 0, \end{aligned} \qquad (6.7)$$

отсюда $4y = -2\lambda x$; $4x = -2\lambda y$, $\frac{y}{x} = \frac{x}{y}$, $y^2 = x^2$, $2x^2 = R^2$; $x = \sqrt{\frac{R^2}{2}} = \frac{R}{\sqrt{2}}$;

$y = \frac{R}{\sqrt{2}}$; $S_{max} = 4\frac{R^2}{2} = 2 \cdot R^2$, т. е. это квадрат.

6.4.5. Метод градиентного спуска

Из курса математики известно, что направление наибольшего изменения значения функции характеризуется вектором – градиентом:

$$\overrightarrow{grad U} = \frac{\partial u}{\partial x_1}\overrightarrow{e_1} + \frac{\partial u}{\partial x_2}\overrightarrow{e_2} + \ldots + \frac{\partial u}{\partial x_n}\overrightarrow{e_n}, \qquad (6.8)$$

где $u = u(x_1, x_2, \ldots, x_n)$, $\overrightarrow{e_1}, \overrightarrow{e_2}, \ldots, \overrightarrow{e_n}$ – единичные векторы.

Если мы ищем *max*, то должны двигаться в направлении градиента, если *min*, то в направлении антиградиента.

Суть метода в следующем. Допустим, ищем экстремум в виде *min*. Выбираем начальную току $M^0(x_1^0, x_2^0, \ldots, x_N^0)$ и рассчитываем в ней вектор-градиент, делаем вдоль него шаг – приходим в точку, где значение u меньше; если это так, то можно сделать еще шаг и так далее, пока u уменьшается.

Если оказалось так, что u не изменилась или даже увеличилась, можно уменьшить шаг, в новой точке определить градиент и опять двинуться вдоль него. Процесс продолжается до достижения минимального значения. *Сигналом окончания процесса может быть равенство нулю градиента.*

Реализуем изложенный алгоритм. Имеем функцию u (x_1, x_2, …, x_N). Берем частные производные от u в точке M^0:

$$\overrightarrow{grad u} = \frac{\partial u}{\partial x_1}\Big|_{M^0}\overrightarrow{e_1} + \frac{\partial u}{\partial x_2}\Big|_{M^0}\overrightarrow{e_2} + \ldots + \frac{\partial u}{\partial x_N}\Big|_{M^0}\overrightarrow{e_n}, \qquad (6.9)$$

где $\overrightarrow{e_1}, \overrightarrow{e_2}, \ldots, \overrightarrow{e_n}$ – единичные векторы. Обозначим

$\frac{\partial u}{\partial x_1}\Big|_{M^0} = \alpha$; $\frac{\partial u}{\partial x_2}\Big|_{M^0} = \beta$; … ; $\frac{\partial u}{\partial x_n}\Big|_{M^0} = \gamma$; тогда

$$\overrightarrow{gradu} = \alpha \cdot \overrightarrow{e_1} + \beta \cdot \overrightarrow{e_2} + \ldots + \gamma \cdot \overrightarrow{e_n}. \quad (6.10)$$

Выполним шаги вдоль направления вектора градиента. Пусть вдоль координаты x_1 выбран шаг Δx_1. Шаги по остальным координатам должны быть рассчитаны в соответствии с величинами коэффициентов α, β, …, γ, т. е. $\Delta x_2 = \Delta x_1 \frac{\beta}{\alpha}; \ldots; \Delta x_n = \Delta x_1 \frac{\gamma}{\alpha}$. После этого подсчитываем координаты точки M', которая таким образом лежит в направлении градиента:

$$x_1' = x_1^0 + \Delta x_1;\ x_2' = x_2^0 + \Delta x_2;\ \ldots; x_N' = x_N^0 + \Delta x_N;$$
$$M^1(x_1', x_2', \ldots, x_N'). \quad (6.11)$$

Если в точке M' функция становится меньше (если мы ищем минимум), то делаем следующий шаг в этом направлении. Для значения координат некоторой точки на m-м шаге можно записать

$$\begin{gathered} x_1^m = x_1^0 + m \cdot \Delta x_1, \\ x_2^m = x_2^0 + m \cdot \Delta x_2, \\ \ldots\ldots\ldots\ldots\ldots, \\ x_n^m = x_n^0 + m \cdot \Delta x_n \end{gathered} \quad (6.12)$$

На каждом шаге m подсчитывается значение функции u. Если, например, на этом шаге u увеличивается, то (если мы ищем минимум) нужно вернуться на шаг (m-1) и в этой точке опять подсчитать уточненное значение вектора градиента и двигаться шагами в новом направлении.

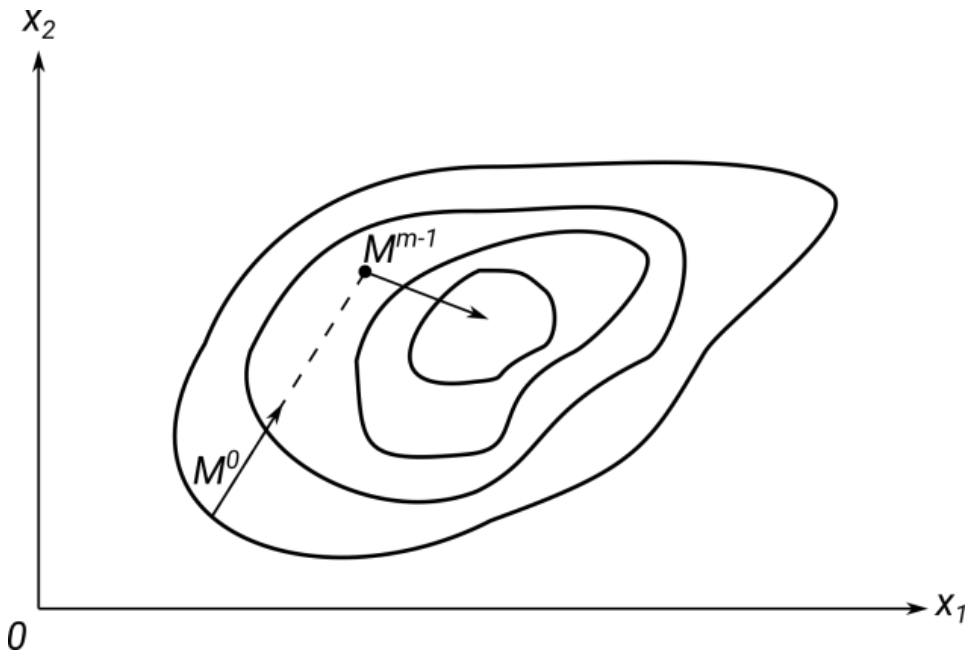

Рисунок 13. Метод градиентного спуска

Если функция не задана аналитически, то производные можно вычислять с помощью численного дифференцирования.

$$\frac{\partial f}{\partial x_i} \approx \frac{1}{\Delta x_i}[f(x_1, \ldots, x_i + \Delta x_i, \ldots, x_N) - f(x_1, \ldots, x_i, \ldots, x_N)], \quad i = 1, 2, \ldots, N \tag{6.13}$$

6.4.6. Симплекс-метод

Не следует путать с симплекс-методом в линейном программировании. В последнем ставится задача поиска минимума значения *линейной функции,* а у нас нелинейной, да к тому же и идея, и сам процесс оптимизации другой. Симплекс – означает «простой». В геометрии нужен для обозначения

простейшей пространственной фигуры в евклидовом пространстве.

Симплекс – это множество $k + 1$ независимых точек, образующих выпуклую фигуру в k-мерном пространстве. Если все ребра симплекса имеют одинаковую длину, то он называется правильным (регулярным) симплексом. Если имеем одну переменную, т. е. критерий оптимизации $f(x_1)$, то симплекс – отрезок. Если имеем две переменные f (x_1, x_2), то симплекс – треугольник. Если имеются три переменных, то симплекс – пирамида, тетраэдр.

Основные принципы метода

Покажем на примере функции двух переменных: $u = f$ (x, y). В этом случае симплекс – равносторонний треугольник. Пусть область определения – прямоугольник (см. рисунок 14). Здесь изображены линии равного уровня.

Главный принцип поиска экстремальных значений – это принцип отражения. В область D произвольным образом забрасывается симплекс. Координаты вершин его известны. И в этих вершинах подсчитываются значения:

$$u_1 = f(x_1, x_1);\ u_2 = f(x_2, x_2);\ u_3 = f(x_3, x_3). \qquad (6.14)$$

Вершина, в которой u оказалась минимальной, считается наихудшей (если мы ищем максимум). В нашем случае это u_1, так как $u_1<u_2<u_3$. Наихудшая вершина зеркально отражается через противоположную грань (в двухмерном случае – через противоположную сторону). В новом треугольнике (2-3-4) опять

ищется наихудшая вершина (в нашем случае – 2), которая вновь отражается и так далее. Мы видим, что в пределах некоторой полосы, определяемой размерами симплекса, происходит движение в правильном направлении. Математически показывается, что движение симплекса близко по направлению к вектору-градиенту. На рис. 14 показаны наихудшие вершины и в какие по номеру вершины они отражаются: 1 → 4; 2 → 5; 3 → 6; 6 → 8. Вершина 8 отражается в максимум.

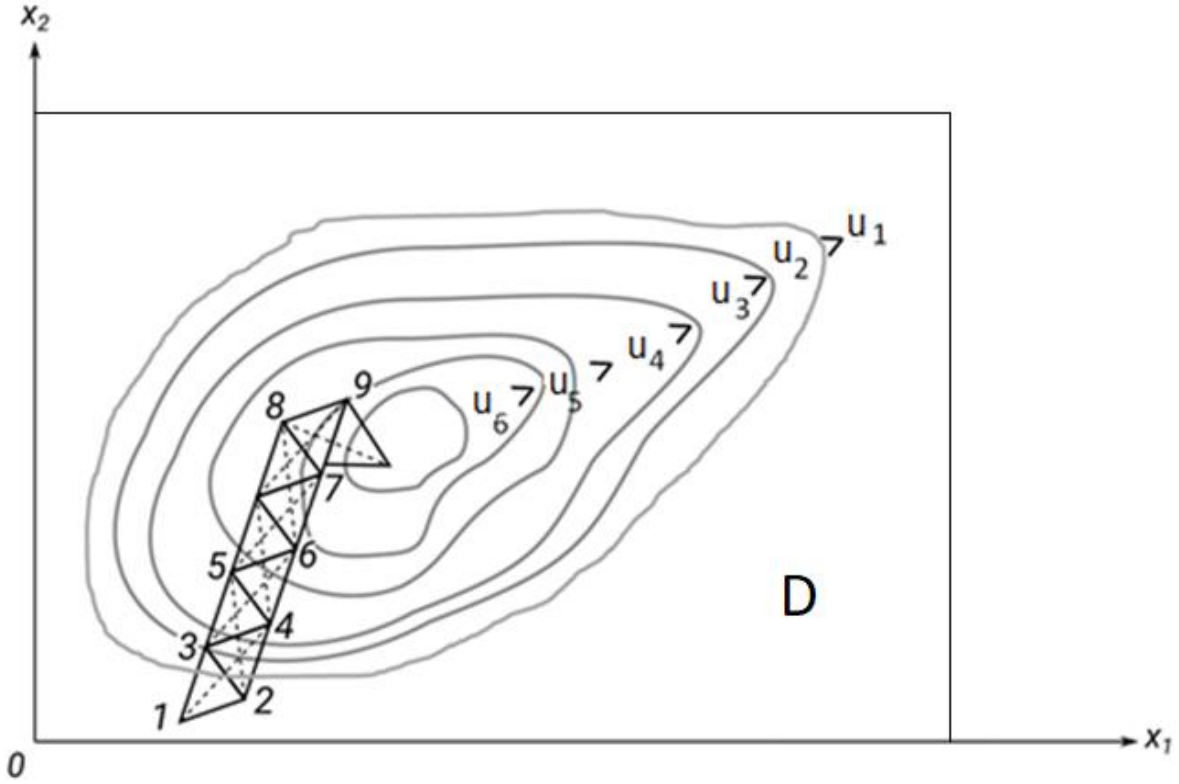

Рисунок 14. Движение симплекса

Расчет *отраженной вершины* достаточно прост. Пусть имеем k переменных: $u = f(x_1, x_2, \ldots, x_k)$. В этом случае в k-мерном пространстве симплекс имеет $(k+1)$ вершину: пусть $\overrightarrow{v_1}, \overrightarrow{v_2}, \ldots, \overrightarrow{v_{i-1}}, \overrightarrow{v_i}, \overrightarrow{v_{i+1}}, \ldots, \overrightarrow{v_{k+1}}$ – векторы, проведенные из начала

координат в эти вершины. И пусть вершина $\overrightarrow{v_i^{\text{н}}}$ – наихудшая. Тогда вектор отраженной вершины $\overrightarrow{v_i^*}$ определится так:

$$\overrightarrow{v_1^*} = \frac{2}{k}(\overrightarrow{v_1} + \overrightarrow{v_2} + \ldots + \overrightarrow{v_{i-1}} + \overrightarrow{v_{i+1}} + \ldots + \overrightarrow{v_{k+1}}) - \overrightarrow{v_i^{\text{н}}} \qquad (6.15)$$

	$x_1\ x_2\ \ldots\ x_i\ \ldots\ x_k$
вершина № 1	$x_{11}\ x_{21}\ \ldots\ x_{i1}\ \ldots\ x_{k1}$
вершина № 2	$x_{12}\ x_{22}\ \ldots\ x_{i2}\ \ldots\ x_{k2}$
……..	……..
вершина № ℓ (наихудшая)	$x_{1\ell}\ x_{2\ell}\ \ldots\ x_{i\ell}\ \ldots\ x_{k\ell}$
…….	……..
вершина № k+1	$x_{1\ k+1}\ x_{2\ k+1}\ \ldots\ x_{i\ k+1}\ \ldots\ x_{k\ k+1}$
отраженная вершина	$x^*_{1\ k+2}\ x^*_{2\ k+2}\ \ldots x^*_i\ \ldots x^*_{k\ k+2}$

$$x^*_{i\ k+2} = \frac{2}{k}\left(\sum_{n=1}^{k+1} x_{in} - x_{i\ell}\right) - x_{i\ell}, \qquad i = 1, 2, \ldots, k \qquad (6.16)$$

В этой формуле фиксированное ℓ. Кроме того, в процессе счета нужно учитывать следующие правила:

Если в новом симплексе отраженная вершина окажется вновь наихудшей и возникнут «колебания» симплекса, то нужно отразить вторую по наихудшести вершину. Это бывает, если симплекс попадает на «гребень» (см. рисунок 15).

Вершины «4» и «3» повторяются. Чтобы сойти с гребня, нужно отразить вторую по «наихудшести» вершину. Если мы ищем минимум, то вторая по наихудшести – вершина номер 2. Ее

отражением получаем вершину «5», а затем вершину «6», таким образом, уходим от колебаний симплекса.

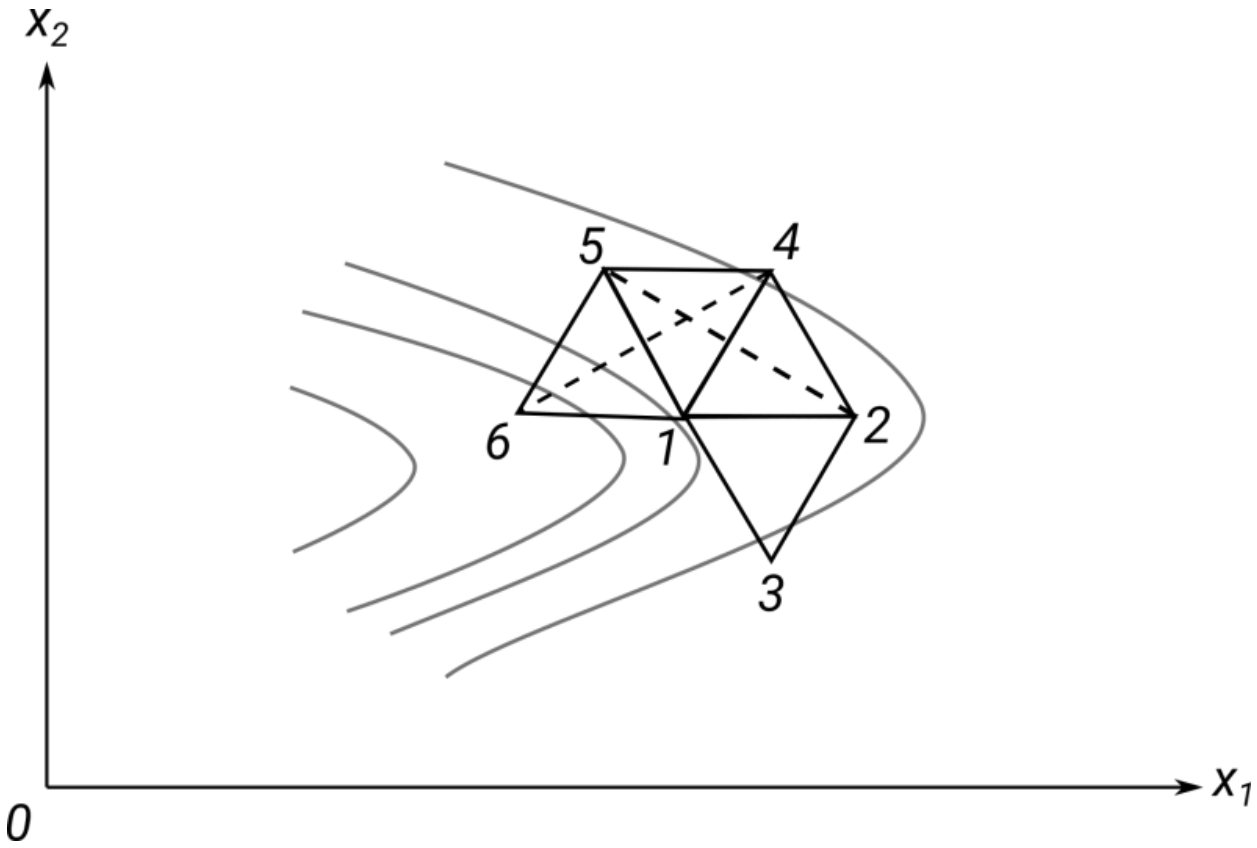

Рисунок 15. «Гребень»

Если симплекс зацикливается, значит вершина, около которой идет зацикливание – экстремальная. Это локальный или глобальный максимум, или минимум.

Симплекс вращается около вершины в точке М. Значит, точка М – экстремум.

Как конкретно рассчитывать координаты вершин исходного симплекса, учитывая к тому же, что каждая переменная x_1, x_2, …, x_k может иметь свои интервалы изменения? Один из подходов может быть следующим.

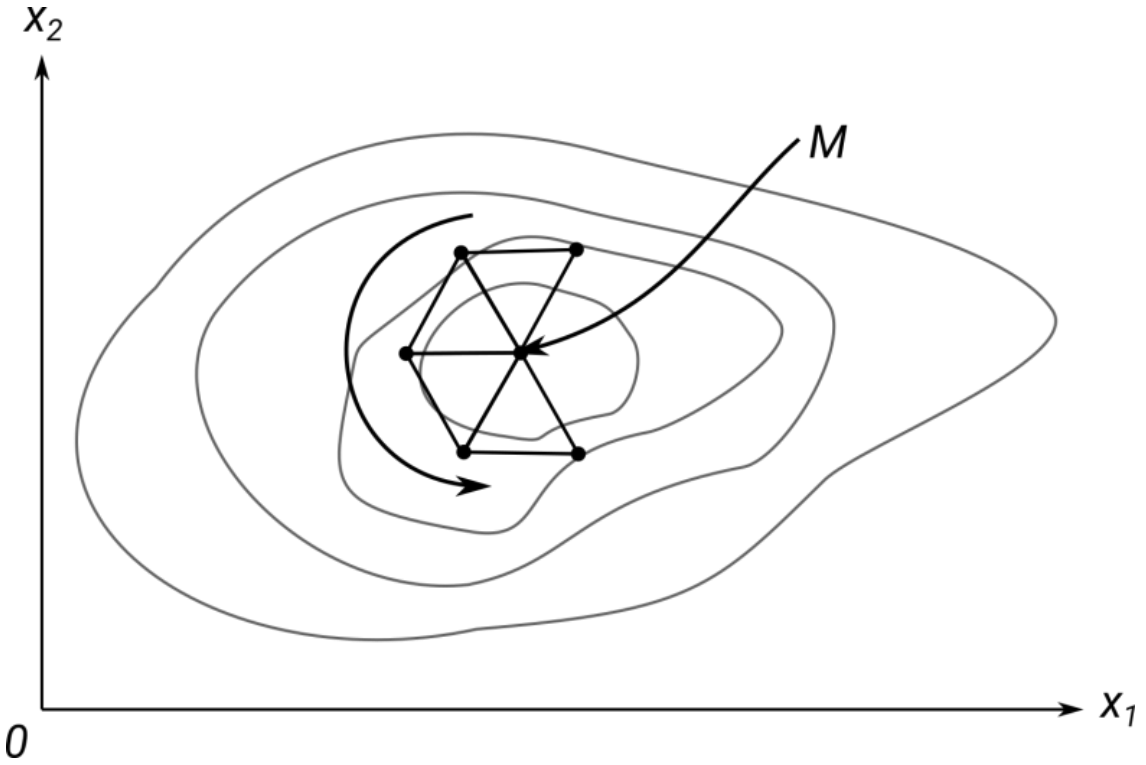

Рисунок 16. Экстремум

Пусть произвольная переменная x_i имеет интервал изменения $[a_i, b_i]$. Имеет смысл ввести единый для всех переменных интервал изменения в безразмерных единицах [-N, N], симметричный относительно нуля. Здесь N задается исходя из условия задачи и должен быть $N >> 1$. Дело в том, что в литературе имеются координаты вершин симплекса, симметричного относительно начала координат, ребра которого равны 1. Целесообразно сделать привязку к такому симплексу.

Для этого имеет смысл перейти для каждой переменной к кодированному виду:

$$X_i = \frac{x_i - \bar{x}_i}{b_i - a_i} N \cdot 2, \qquad (6.17)$$

$$\text{где } \bar{x}_i = \frac{b_i + a_i}{2}$$

Введение $\bar{x}_i$ дает возможность *центрировать* диапазон $(b_i - a_i)$ относительно нуля. Пусть

$$x_i = b_i;\ X_i = \frac{b_i - \frac{b_i + a_i}{2}}{b_i - a_i} 2N = N \tag{6.18}$$

$$x_i = \bar{x}_i;\ X_i = \frac{x_i - \bar{x}_i}{(b_i - a_i)} 2N = 0 \tag{6.19}$$

$$x_i = a_i;\ X_i = \frac{a_i - \frac{b_i + a_i}{2}}{b_i - a_i} 2N = -N \tag{6.20}$$

Переход от кодированных переменных, следующий:

$$x_i = \frac{X_i}{2N}(b_i - a_i) + \bar{x}_i. \tag{6.21}$$

Теперь мы можем воспользоваться координатами симметричного симплекса

$$X = \begin{vmatrix} X_1 & X_2 & \dots & X_i & \dots & X_{k-1} & X_k \\ -X_1 & X_2 & \dots & X_i & \dots & X_{k-1} & X_k \\ 0 & -2X_2 & \dots & X_i & \dots & X_{k-1} & X_k \\ \dots & 0 & \dots & X_i & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & -iX_i & \dots & X_{k-1} & X_k \\ \dots & \dots & \dots & 0 & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 & \dots & -(k-1)X_k & X_k \\ 0 & 0 & \dots & 0 & \dots & 0 & -kX_k \end{vmatrix},$$

$$\text{где } X_i = \sqrt{\frac{1}{2i(i+1)}}.$$

Приведем пример для случая $k = 4$, $k + 1 = 5$

		X_1	X_2	X_3	X_4
	1	0,5	0,289	0,204	0,158
	2	-0,5	0,289	0,204	0,158
	3	0	-0,578	0,204	0,158
	4	0	0	-0,612	0,158
наихудшая вершина→	5	0	0	0	-0,632
отраженная вершина→	6	0	0	0	$\frac{2 \cdot 0{,}632}{4} + 0{,}632 = 0{,}948$

Может оказаться необходимостью забросить симплекс в другие участки области и начать из них движение. Это можно сделать, например, сдвигая его вдоль какой-либо координатной оси. Для этого достаточно будет прибавить к каждому элементу соответствующего столбца величину сдвига $\Delta x_i = n$, где $n = 1, 2, \ldots, N$.

Метод весьма привлекателен, имеет несомненные преимущества: простота расчетов, на каждом шаге нужно определять координаты лишь одной (отраженной) вершины и значение *f* в этой вершине.

В сравнении с градиентом или спуском метод универсален и может быть использован даже при качественной оценке *f*.

Возможны модификации: отражение двух вершин (если *f* в них примерно одинаково).

6.4.7. Метод вращаемого поиска

Несколько похож на симплекс. В пространстве координат переменных строится квадрат, куб, гиперкуб. Допустим, имеем две переменные x_1, x_2 и $f(x_1, x_2)$. Забрасываем квадрат в случайную подобласть. Определяем f в вершинах. Выбираем наилучшую x_1 в смысле значения f вершину и, рассматривая ее как центр, строим новый квадрат, затем опять находим лучшую вершину и строим квадрат, повернутый на угол и так далее. При этом можно менять сторону квадрата.

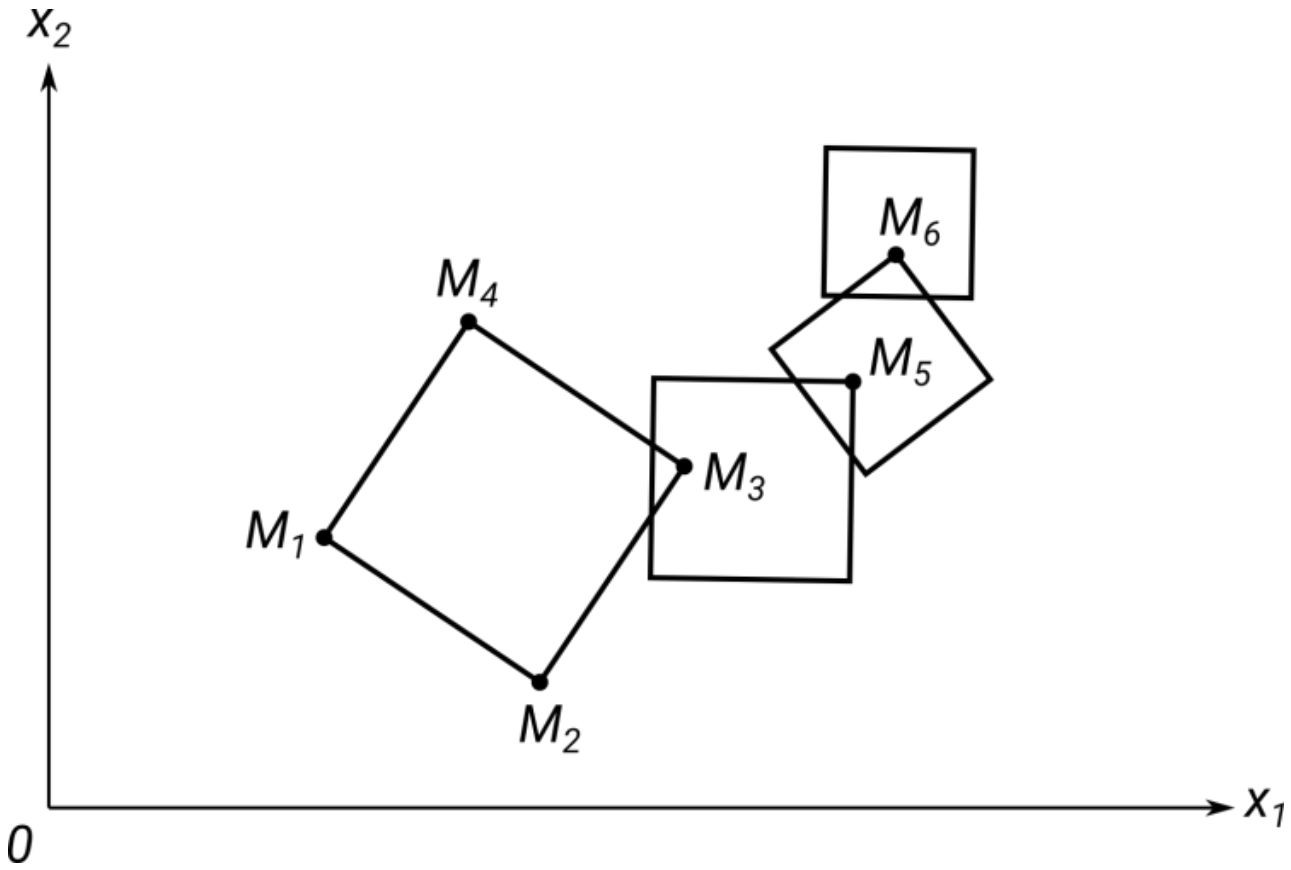

Рисунок 17. Метод вращаемого поиска

На этом рисунке наилучшими вершинами квадратов были M_3, M_5, M_6 соответственно.

6.5. Последовательный случайный поиск

Методы случайного поиска (весьма разнообразны)

(Методы Монте-Карло)

Пусть дана f (x_1, x_2, …, x_N). Вначале приведем интервалы изменения переменных к единому интервалу, как это делали ранее.

Введем для каждой переменной шкалу значений, например, от 1 до 100. Выбираем стартовую точку, например, с координатами $x_1 = 50$; $x_2 = 50$, …, $x_n = 50$. Следующая точка должна быть случайной. Используя генератор случайных чисел, выбираем случайное значение для x_1: пусть оно будет d_1.

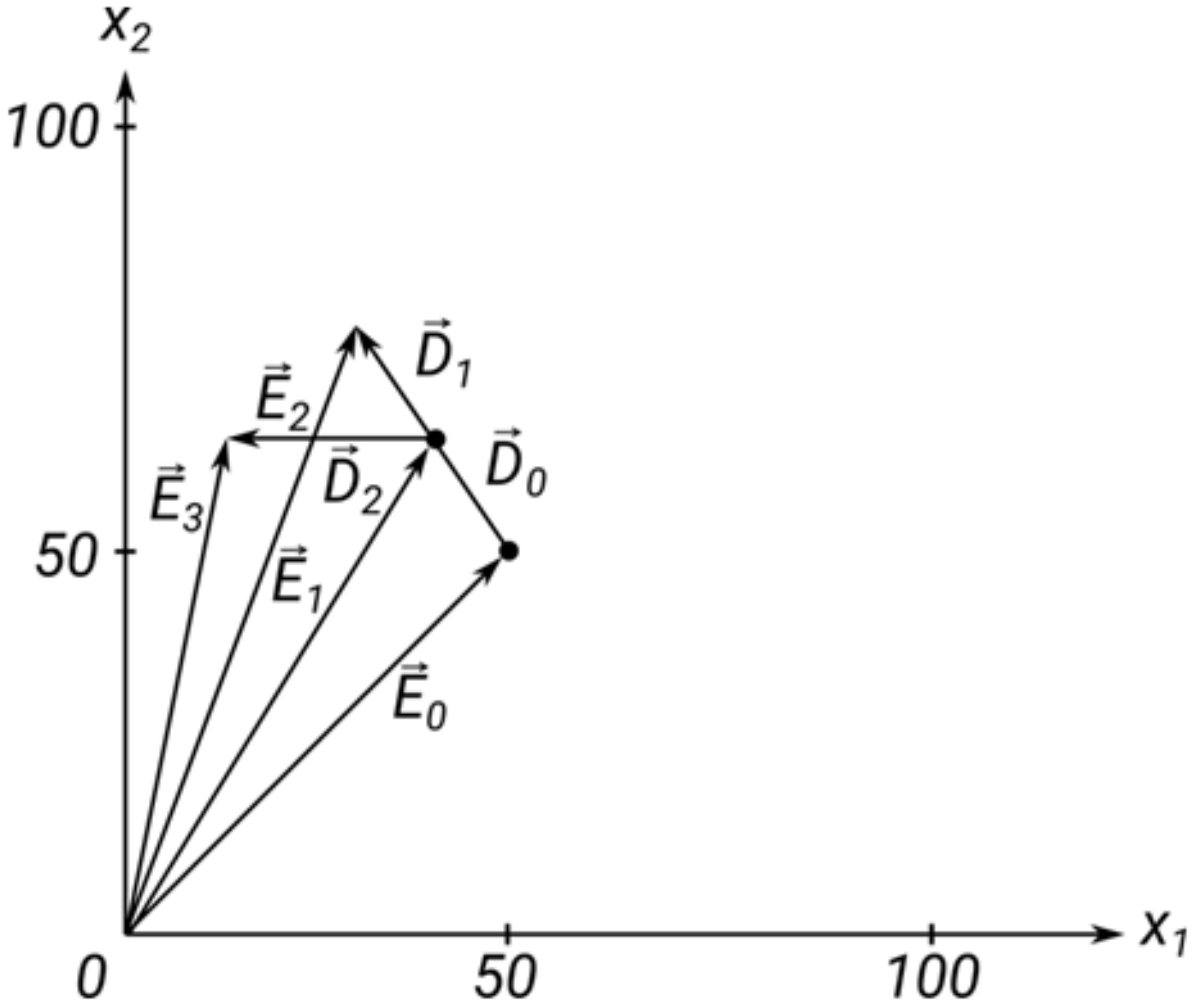

Рисунок 18. Последовательный случайный поиск

Затем подсчитываем $\Delta x_1 = \frac{d_i^2}{c}$, где c = 5000. Значение c выбрано так, чтобы Δx_1 не превышало, например, 2. Знак Δx_1 также выбирается случайным между «+» и «–». Так же вычисляются добавки по всем другим переменным. Таким образом, получаем вектор добавки $\overrightarrow{D_0}$. Если исходная точка соответствовала вектору $\overrightarrow{E_0}$, то новая будет соответствовать вектору $\overrightarrow{E_1} = \overrightarrow{E_0} + \overrightarrow{D_0}$.

Подсчитываем значение функции $f(x_1, x_2, \ldots, x_n)$ в точке, соответствующей вектору $\overrightarrow{E_1}$, и сравниваем с исходным значением в точке, соответствующей вектору $\overrightarrow{E_0}$. Если в $\overrightarrow{E_1}$ значение f лучше чем в $\overrightarrow{E_0}$, то формируем точку $\overrightarrow{E_2} = \overrightarrow{E_1} + \overrightarrow{D_1}$. Если нет, то возвращаемся в точку $\overrightarrow{E_1}$ и опять описанным образом находим в точке $\overrightarrow{E_1}$ новую добавку – вектор $\overrightarrow{D_2}$. $\overrightarrow{E_3} = \overrightarrow{E_3} + \overrightarrow{D_2}$.

6.6. Глобальный случайный поиск

Опять введем для каждой переменной шкалу значений, например, опять от 1 до 100. Используя генератор случайных чисел с равномерным распределением, для каждой переменной найдем некоторое число в интервале от 1 до 100.

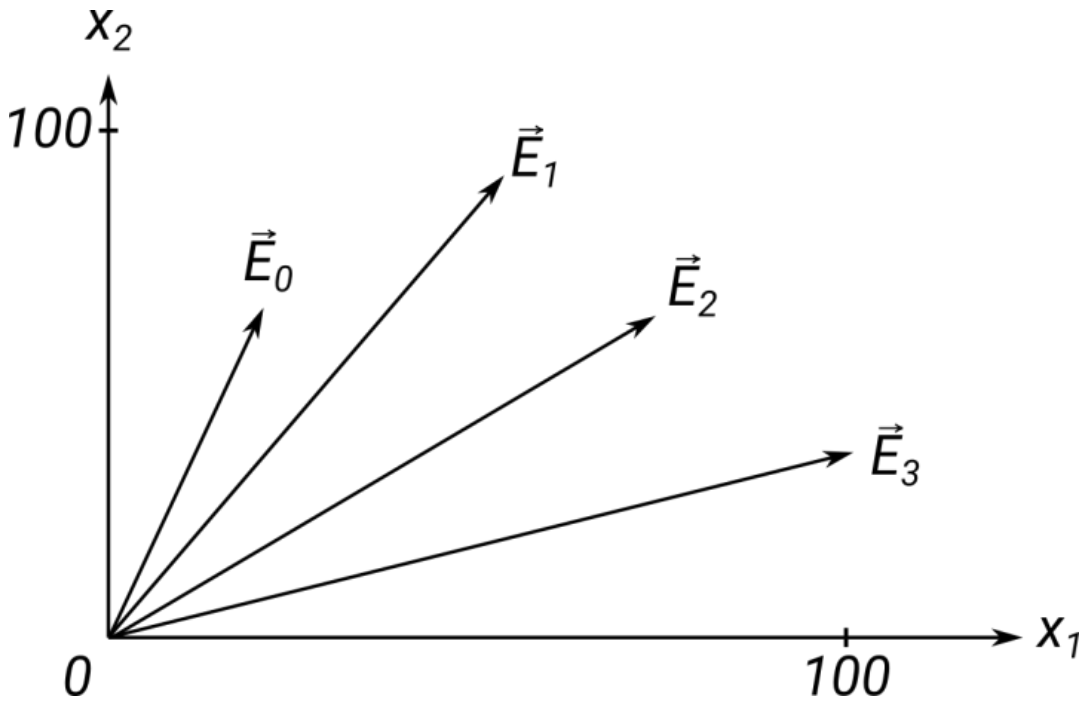

Рисунок 19. Глобальный случайный поиск

Получим набор значений переменных и находим соответствующее *f*. Выбор таких наборов проведем много раз и получим множество значений f – из них найдутся лишние. То есть это аналогично методу перебора. Кстати, здесь может оказаться полезным применение симплексного метода, то есть вокруг лучшей точки можно построить симплекс (в двумерном случае треугольник) и далее использовать принцип отражения.

СПИСОК ЛИТЕРАТУРЫ

1. Турчак Л.А. Численные методы / Л.А. Турчак. – М.: Физматлит, 2003. – 226 с.

2. Вержбицкий В.М. Вычислительная линейная алгебра: учеб. пособие для вузов / В.М. Вержбицкий. – 3-е изд.– Москва; Берлин: Директ-Медиа, 2021. – 354 с.

3. Слабнов В.Д. Численные методы: учеб. / В.Д. Слабнов. – СПб.: Лань, 2020. – 392 с.

4. Волков Е.А. Численные методы: учеб. пособие для вузов / Е.А. Волков. – 6-е изд., стер. – СПб.: Лань, 2021. – 252 с.

5. Байбурин В.Б., Розов А.С., Губенков А.А., Кожанова Е.Р., Никифоров А.А. Численные методы решения основных дифференциальных уравнений математической физики. – Саратов.: Сарат. гос. техн. ун-т.2022. – 84с.