Министерство образования и науки Российской Федерации

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

А. А. Мицель

ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ

Учебное пособие

Томск
«Эль Контент»
2013

УДК 519.6(075.8)
ББК 22.19я73
М 701

Рецензенты:
Старченко А. В., докт. физ.-мат. наук, профессор, зав. кафедрой вычислительной математики и компьютерного моделирования Томского государственного университета;
Кочегуров А. И., доцент кафедры прикладной математики Томского политехнического университета.

Мицель А. А.
М 701 Вычислительные методы : учебное пособие / А. А. Мицель. — Томск : Эль Контент, 2013. — 198 с.

ISBN 978-5-4332-0121-7

В учебном пособии изложены основные разделы вычислительной математики (решение нелинейных уравнений с одной переменной, решение задач линейной алгебры, решение систем нелинейных уравнений, приближение функций, численное дифференцирование и интегрирование, решение обыкновенных дифференциальных уравнений). Рассмотрены вопросы устойчивости численных алгоритмов. Каждый раздел снабжен примерами и вопросами для самопроверки.

Пособие представляет интерес для студентов, инженеров, аспирантов, преподавателей, ученых, занимающихся вопросами численного моделирования и решения прикладных задач.

УДК 519.6(075.8)
ББК 22.19я73

ISBN 978-5-4332-0121-7

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Математическое моделирование *представляет собой метод исследования объектов и процессов реального мира с помощью их приближенных описаний на языке математики —* ***математических моделей****.*

Этот метод чрезвычайно плодотворен и известен уже несколько тысячелетий. Насущные задачи земледелия и строительства еще в древние времена приводили к необходимости определения площадей и объемов, а следовательно, и к рассмотрению элементарных геометрических фигур, дающих пример простейших математических моделей. Возможности математического моделирования и его влияния на научно-технический прогресс неизмеримо возросли в последние десятилетия в связи с созданием и широким внедрением компьютеров.

Следует отметить, что современные успехи в решении таких важных проблем, как атомные, космические, экономические не были бы возможны без применения ЭВМ и численных методов. По оценкам ученых эффект, достигаемый за счет совершенствования численных методов, составляет 40% общего эффекта, достигаемого за счет повышения производительности ЭВМ.

Конечность скорости распространения сигнала — 300000 км/с является существенным ограничением роста быстродействия однопроцессорных ЭВМ. Поэтому, наряду с созданием многопроцессорных ЭВМ, все большую роль в повышении производительности ЭВМ приобретают численные методы.

Основные этапы решения инженерной задачи

Решение серьезной инженерной задачи с использованием компьютера — довольно длительный и сложный процесс. Условно его можно разбить на ряд последовательных этапов. Выделим следующие этапы [1, 2]:

1) постановка проблемы и построение математической модели;
2) постановка вычислительной задачи и выбор численного метода;
3) алгоритмизация и программирование;
4) счет по программе и интерпретация результатов;
5) использование результатов и коррекция математической модели.

1. *Постановка проблемы и построение математической модели*. Первоначально прикладная задача бывает сформулирована в самом общем виде: исследовать некоторое явление, спроектировать устройство, обладающее заданными свойствами, дать прогноз поведения некоторого объекта в определенных условиях и т. д. На данной стадии происходит конкретизация постановки задачи, и первостепенное внимание при этом уделяется выяснению цели исследования. Неудачная постановка проблемы может привести к тому, что длительный и дорогостоящий процесс решения задачи завершится получением бесполезных или тривиальных результатов.

Для последующего анализа исследуемого явления или объекта необходимо дать его формализованное описание на языке математики, т. е. построить математическую модель. Часто имеется возможность выбора модели среди известных и принятых для описания соответствующих процессов, но нередко требуется и существенная модификация известной модели, а иногда возникает необходимость в построении принципиально новой модели.

Удачный выбор математической модели является решающим шагом к достижению цели. Одна из существенных трудностей такого выбора состоит в объективном противоречии между желанием сделать описание явления как можно более полным (что приводит к усложнению модели) и необходимостью иметь достаточно простую модель (чтобы была возможность реализовать ее на компьютере). Важно, чтобы сложность математической модели соответствовала сложности поставленной проблемы. Если поставленных целей можно достигнуть с помощью более простой математической модели, то следует ей воспользоваться.

2. *Постановка вычислительной задачи и выбор численного метода*. На основе принятой математической модели формулируют вычислительную задачу и проводят предварительное исследование свойств вычислительной задачи. Большое внимание уделяют анализу корректности ее постановки, т. е. выяснению вопросов существования и единственности решения, а также исследованию устойчивости решения задачи к погрешностям входных данных.

На этом этапе полезным оказывается изучение упрощенных постановок задачи. Иногда для них удается провести исследование, позволяющее понять основные особенности исходной вычислительной задачи. Особую ценность имеют различные аналитические решения; они оказываются полезными не только для анализа явления, но и как основа для тестовых испытаний на этапе отладки программы.

Далее, для решения вычислительной задачи на компьютере требуется использование численных методов.

Часто решение инженерной задачи сводится к последовательному решению стандартных вычислительных задач, для которых разработаны эффективные численные методы. В этой ситуации происходит либо выбор среди известных методов, либо их адаптация к особенностям решаемой задачи. Однако если возникающая вычислительная задача является новой, то не исключено, что для ее решения не существует готовых методов.

Для решения одной и той же вычислительной задачи обычно может быть использовано несколько методов. Необходимо знать особенности этих методов, критерии, по которым оценивается их качество, чтобы выбрать метод, позволяющий решить проблему наиболее эффективным образом.

3. *Алгоритмизация и программирование*. Этап поиска и разработки алгоритма решения называют алгоритмизацией. Здесь могут использоваться как математические формулы и блок-схемы, так и словесные описания алгоритмов. Во многих случаях вслед за построением алгоритма выполняют так называемый контрольный просчет — грубую прикидку ожидаемых результатов, которые используются затем для анализа полученного решения.

Затем алгоритм решения задачи записывается на языке, понятном ЭВМ. Это — этап программирования. В настоящее время для вычислительных задач наиболее широко используются алгоритмические языки СИ++ и ФОРТРАН. В простейших случаях может оказаться, что на этом этапе вовсе и не составляется новая программа для ЭВМ, а дело сводится, например, к использованию имеющегося математического обеспечения ЭВМ.

Вопросы разработки программного продукта выходят за рамки данного пособия. Подчеркнем лишь, что большинство пользователей справедливо предпочитает строить свои программы из готовых модулей и использовать стандартные программы, реализующие те или иные алгоритмы.

После написания программы с помощью компьютера выявляют и исправляют ошибки в программе.

Как правило, начинающий пользователь компьютера убежден, что ошибок в составленной им программе нет или же они могут быть легко обнаружены и исправлены. Однако совершенно неожиданно для него отладка программы и доведение ее до рабочего состояния нередко оказывается длительным и весьма трудоемким процессом. Приобретая определенный опыт в составлении и отладке сравнительно сложных программ, пользователь убеждается в справедливости популярного афоризма: «В любой программе есть по крайней мере одна ошибка».

После устранения ошибок программирования необходимо провести тщательное тестирование программы — проверку правильности ее работы на специально отобранных тестовых задачах, имеющих известные решения.

4. *Счет по программе и интерпретация результатов*. На этом этапе происходит решение задачи на компьютере по составленной программе в автоматическом режиме. Этот процесс, в ходе которого входные данные с помощью компьютера преобразуются в результат, называют *вычислительным процессом*. Как правило, счет повторяется многократно с различными входными данными для получения достаточно полной картины зависимости от них решения задачи.

Первые полученные результаты тщательно анализируются, для того чтобы убедиться в правильности работы программы и пригодности выбранного метода ре-

шения. Счет по программе продолжается несколько секунд, минут или часов. Именно быстротечность этого этапа порождает распространенную иллюзию о возможности решать важные прикладные задачи на компьютере в очень короткое время. В действительности же, конечно, необходимо принимать во внимание весь цикл от постановки проблемы до использования результатов. Для серьезных задач часто полезные результаты получаются только в результате многолетней работы.

Для того чтобы исследователь мог воспользоваться результатами расчетов, их необходимо представить в виде компактных таблиц, графиков или в иной удобной для восприятия форме. При этом следует максимально использовать возможности компьютера для подготовки такой информации и ее представления с помощью печатающих и графических выходных устройств.

Для правильной интерпретации результатов расчетов и оценки их достоверности от исследователя требуется глубокое знание существа решаемой инженерной задачи, ясное представление об используемой математической модели и понимание (хотя бы в общих чертах) особенностей применяемого вычислительного метода.

5. *Использование результатов и коррекция математической модели*. Завершающий этап состоит в использовании результатов расчетов в практической деятельности, иначе говоря, во внедрении результатов.

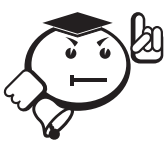
Очень часто анализ результатов, проведенный на этапе их обработки и интерпретации, указывает на несовершенство используемой математический модели и необходимость ее коррекции. В таком случае математическую модель модифицируют (при этом она, как правило, усложняется) и начинают новый цикл решения задачи.

Соглашения, принятые в книге

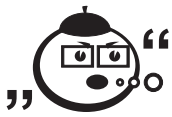
Для улучшения восприятия материала в данной книге используются пиктограммы и специальное выделение важной информации.

Эта пиктограмма означает определение или новое понятие.

Эта пиктограмма означает внимание. Здесь выделена важная информация, требующая акцента на ней. Автор здесь может поделиться с читателем опытом, чтобы помочь избежать некоторых ошибок.



В блоке «Замечание» автор может указать дополнительные сведения или другой взгляд на изучаемый предмет, чтобы помочь читателю лучше понять основные идеи.



Эта пиктограмма означает цитату.

Эта пиктограмма означает теорему.

Эта пиктограмма означает лемму.

Пример

Эта пиктограмма означает пример. В данном блоке автор может привести практический пример для пояснения и разбора основных моментов, отраженных в теоретическом материале.

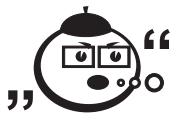
Контрольные вопросы по главе

Глава 1

ПОГРЕШНОСТИ ВЫЧИСЛЕНИЙ

1.1 Источники погрешностей

При использовании ЭВМ численные методы выступают как мощное математическое средство решения практических задач. При этом важно иметь в виду, что фактор использования ЭВМ не упрощает, а в некотором смысле даже усложняет решение вопросов оценки точности получаемых результатов (ввиду резкого возрастания количества выполняемых операций). Суть возникающих здесь проблем подмечена в известном принципе Питера:



«ЭВМ многократно увеличивает некомпетентность пользователя.»

На общую погрешность решения задачи влияет целый ряд факторов [1–6]. Отметим основные из них. Пусть R — точное значение результата решения некоторой задачи. Из-за несоответствия построенной математической модели реальной ситуации, а так же по причине неточности исходных данных вместо R будет получено R_1. Образовавшаяся погрешность $\varepsilon_1 = |R - R_1|$ называется погрешностью модели. Эта погрешность не может быть устранена (неустранимая погрешность).

Приступив к решению задачи в рамках математической модели, мы избираем приближенный (например численный) метод и еще, не приступив к вычислениям, допускаем новую погрешность, приводящую к получению результата R_2 (вместо R_1). Погрешность $\varepsilon_2 = |R_2 - R_1|$ называется погрешностью метода.

И наконец неизбежность округления приводит к получению результата R_3, отличающегося от R_2 на величину вычислительной погрешности $\varepsilon_3 = |R_3 - R_2|$.

Полная погрешность ε равна $\varepsilon = \varepsilon_1 + \varepsilon_2 + \varepsilon_3$.

Пример 1.1

Пусть маятник начал движение в момент времени t_0. Требуется предсказать угол φ отклонения от вертикали в момент $t = t_1$.

Модель. Колебания маятника можно описать уравнением вида

$$ml\varphi'' + mg\sin\varphi + \mu\varphi' = 0$$

Рис. 1.1

с начальными условиями $\varphi(t_0) = \varphi_0$; $\varphi'(t_0) = \varphi_0'$, где l — длина маятника; m — масса; g — ускорение свободного падения; μ — коэффициент трения.

Уже в этом модельном описании присутствует неустранимая погрешность модели: реальное трение зависит от скорости нелинейно; величины g, l, $\varphi(t_0)$, $\dot{\varphi}(t_0)$ могут быть определены и записаны с определенной точностью.

Погрешность метода: исходное уравнение аналитического решения не имеет, следовательно, для его решения необходимо применять некоторые приближенные численные методы.

Вычислительная погрешность может возникнуть, например, из-за конечной разрядности компьютера.

Будем далее исходить из предположения, что математическая модель фиксирована и входные данные задаются извне, так что повлиять на значение величины ε_1 в процессе решения задачи действительно нельзя. Однако это совсем не означает, что предварительные оценки значения неустранимой погрешности не нужны. Достоверная информация о порядке величины ε_1 позволяет осознанно выбрать метод решения задачи и разумно задать его точность. Желательно, чтобы погрешность метода ε_2 была в 2–10 раз меньше неустранимой погрешности. Большее значение ε_2 ощутимо снижает точность результата, меньшее — обычно требует увеличения затрат на вычисления, практически уже не влияя на значение полной погрешности. Иногда характер использования результата таков, что вполне допустимо, чтобы погрешность ε_2 была сравнима с ε_1 или даже несколько превышала ее.

Значение вычислительной погрешности в основном определяется характеристиками используемого компьютера. Желательно, чтобы погрешность ε_3 была хотя бы на порядок меньше погрешности метода ε_2, и совсем не желательна ситуация, когда она существенно ее превышает.

Умение анализировать погрешности при решении прикладной задачи и соблюдать между ними разумный компромисс позволяет существенно экономить используемые ресурсы и является признаком высокой квалификации.

Вопросы, связанные с приближенными числами и погрешностями вычислений, излагаются в целом ряде литературных источников (см. например [1–6]).

1.2 Приближенные числа

Абсолютная и относительная погрешности числа

Вопросы, связанные с приближенными числами, излагаются в целом ряде литературных источников (см. например [1–6]).

Пусть x — точное (в общем случае неизвестное) значение некоторой величины, x^* — известное приближенное значение той же величины (приближенное число).

Математическая запись: $x^* \approx x$.

*Под **абсолютной ошибкой** или **абсолютной погрешностью** (АП) $\Delta(x^*)$ приближенного числа x^* обычно понимается разность*

$$\Delta(x^*) = |x - x^*| \tag{1.1}$$

Отсюда следует, что x заключено в пределах

$$x^* - \Delta(x^*) \leqslant x \leqslant x^* + \Delta(x^*) \tag{1.2}$$

или $x = x^* \pm \Delta(x^*)$.

Определим абсолютную погрешность числа $x^* = 3.14$, заменяющего число π. ($\pi = 3.14159265$).

Так как $3.14 < \pi < 3.15$, то $|x^* - \pi| < 0.01$, то есть $\Delta = 0.01$.

Пример 1.2

Пусть длина отрезка измеряется линейкой с точностью до 0.5 см. Тогда если получилась $l = 154$ см, то пишут $l = 154 \pm 0.5$ см. Здесь $\Delta = 0.5$ см, а точная величина $153.5 \leqslant l \leqslant 154.5$.

***Относительной погрешностью** (ОП) $\delta(x^*)$ (приближенного числа x^*) называется отношение абсолютной погрешности $\Delta(x^*)$ этого числа к модулю соответствующего точного числа.*

$$\delta = \frac{\Delta(x^*)}{|x|}. \tag{1.3}$$

Так как x обычно неизвестно, то на практике применяют оценку

$$\delta = \frac{\Delta(x^*)}{|x^*|}. \tag{1.4}$$

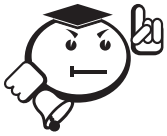
Отсюда $\Delta(x^*) = |x^*| \cdot \delta(x^*)$.

Для точного значения имеем

$$x = x^*\left(1 \pm \delta(x^*)\right)$$

или

$$x^* - x^*\delta(x^*) \leqslant x \leqslant x^* + x^*\delta(x^*).$$



Так как x неизвестно, то вычислить величины $\Delta(x^*)$ и $\delta(x^*)$ невозможно. Поэтому вместо $\Delta(x^*)$ и $\delta(x^*)$ используют верхние границы этих величин.

Пример 1.3

Скорость света в вакууме $c = (2.998 \pm 0.001) \cdot 10^5$ *км/сек. Определить относительную погрешность измерения.*

Решение:

Очевидно $\Delta(c) = 0.001 \cdot 10^5$ км/сек, тогда

$$\delta(c) = \frac{0.001}{2.998} \approx 3 \cdot 10^{-2}\%.$$

Пример 1.4

При определении газовой постоянной получили $R = 8.314$ *Дж/(К·моль). Зная, что относительная погрешность этой величины равна* 0.1%*, найти пределы, в которых заключено* R.

Решение:

Имеем $\delta(R) = 0.001$, тогда $\Delta(R) = R \cdot \delta(R) \approx 0.008$.

Следовательно, $8.306 \leqslant R \leqslant 8.322$.

Правила записи приближенных чисел

Пусть приближенное число x задано в виде конечной десятичной дроби [3]:

$$x^* = \alpha_m 10^m + \alpha_{m-1} 10^{m-1} + \ldots + \alpha_{m-n+1} 10^{m-n+1} + \ldots, \qquad (1.5)$$

где α_i — цифры числа x^* в i-ом разряде ($\alpha_i = 0, 1, \ldots, 9$), причем $\alpha_m \neq 0$; m — старший десятичный разряд числа x^* (некоторое целое число).

Например:

$$x^* = 3141.59\ldots = 3 \cdot 10^3 + 1 \cdot 10^2 + 4 \cdot 10^1 + 1 \cdot 10^0 + 5 \cdot 10^{-1} + 9 \cdot 10^{-2} + \ldots$$

Значащей цифрой приближенного числа *называется всякая цифра в его десятичном изображении, отличная от нуля, и нуль, если он содержится между значащими цифрами или является представителем сохраненного десятичного разряда.*

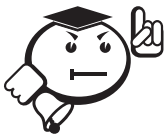
Пример 1.5

1. $x^* = 0.002080$ — здесь первые три нуля не являются значащими цифрами, так как они служат для установки десятичных разрядов других цифр. Остальные два нуля являются значащими.
2. $x^* = 6.89 \cdot 10^5$ — здесь три значащих цифры.
3. $x^* = 6.8900 \cdot 10^5$ — здесь пять значащих цифр.

Говорят, что n первых значащих цифр (десятичных знаков) приближенного числа являются ***верными****, если абсолютная погрешность этого числа не превышает единицы разряда, выражаемого n-ой значащей цифрой, считая слева направо* [3].

Пример 1.6

$$x^* = 0.03450, \quad \Delta_x^4 = 0.00001 = 10^{-5};$$
$$x^* = 0.034500, \quad \Delta_x^5 = 0.000001 = 10^{-6}.$$



Таким образом, если для приближенного числа (1.5), заменяющего точное x, известно, что

$$\Delta(x^*) = |x^* - x| \leqslant 10^{m-n+1}, \tag{1.6}$$

то по определению первые n цифр $\alpha_m, \alpha_{m-1}, \ldots, \alpha_{m-n+1}$ этого числа являются верными.

Например, для точного числа $x = 35.98$ число $x^* = 36.00$ является приближенным с тремя верными цифрами, так как $|x^* - x| = 0.02 < 10^{-1}$ (здесь $m = 1$, отсюда $n = 3$).

Связь относительной погрешности приближенного числа с количеством верных знаков этого числа

Имеет место следующая теорема.

Теорема 1.1. Если положительное приближенное число x^* имеет n верных десятичных знаков (верных значащих цифр) и $\alpha_m \neq 0$, $\alpha_{m-1} \neq 0$, то относительная погрешность δ этого числа при $n > 1$ удовлетворяет следующему неравенству:

$$\delta(x^*) \leqslant \frac{10^{-n+1}}{\alpha_m}, \tag{1.7}$$

где α_m — цифры числа x^* в m-ом (старшем) разряде.

Доказательство. Пусть число

$$x^* = \alpha_m 10^m + \alpha_{m-1} 10^{m-1} + \ldots + \alpha_{m-n+1} 10^{m-n+1} + \ldots \quad (\alpha_m \geqslant 1)$$

является приближенным значением x и имеет n верных знаков. Тогда из (1.3) имеем:

$$\begin{aligned} \delta(x^*) = \frac{\Delta(x^*)}{x^*} \leqslant \frac{\Delta(x^*)}{x^* - \Delta(x^*)} \leqslant \frac{\Delta(x^*)}{\alpha_m \cdot 10^m + \alpha_{m-1} \cdot 10^{m-1} - \Delta(x^*)} \leqslant \\ \leqslant \frac{10^{m-n+1}}{\alpha_m \cdot 10^m + \alpha_{m-1} \cdot 10^{m-1} - 10^{m-n+1}} = \frac{10^{-n+1}}{\alpha_m + \alpha_{m-1} \cdot 10^{-1} - 10^{-n+1}}. \end{aligned} \tag{1.8}$$

Поскольку $\alpha_m \geqslant 1$, $\alpha_{m-1} \geqslant 1$, а $n > 1$, то $\alpha_m + \alpha_{m-1} \cdot 10^{-1} - 10^{-n+1} \geqslant \alpha_m$, поэтому из (1.8) следует неравенство

$$\delta \leqslant \frac{10^{-n+1}}{\alpha_m + \alpha_{m-1} \cdot 10^{-1} - 10^{-n+1}} \leqslant \frac{10^{-n+1}}{\alpha_m}.$$

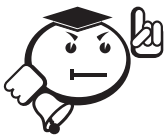
Теорема доказана.

Пример 1.7

Какова относительная погрешность числа π*, если вместо* π *взять* $x^* = 3.14$.

Решение:

Имеем $n = 3$. Следовательно, $\delta(x^*) = \dfrac{1}{\alpha_m} \cdot \left(\dfrac{1}{10}\right)^{3-1} = \dfrac{1}{3 \cdot 100} = 0.33\%$.



Для решения обратной задачи — определения количества верных знаков n числа x, если известна его относительная погрешность δ, можно воспользоваться формулой (1.7) в форме [9]:

$$\tilde{n} = 1 - \lg\bigl(\alpha_m \cdot \delta(x^*)\bigr) \tag{1.9}$$

и в качестве n взять ближайшее к $\tilde{n}$ целое число.

Пример 1.8

Со сколькими десятичными верными знаками надо взять $\sqrt{30}$*, чтобы* $\delta = 0.1\%$.

Решение:

Так как $\sqrt{30} \approx 5.4$, то $\alpha_m = 5$. Из (1.9) имеем $\tilde{n} = 1 - \lg(0.001 \cdot 5) = 3.3$; $n = 3$. Таким образом, получим $x^* = \sqrt{30} = 5.4772\ldots \approx 5.48$.

Округление

Часто возникает необходимость в ***округлении*** *числа* x*, т. е. в замене его другим числом* x^* *с меньшим числом значащих цифр. Возникающая при такой замене погрешность называется* ***погрешностью округления***.

Одним из способов округления числа до n значащих цифр является усечение, т. е. отбрасывание всех цифр, расположенных справа от n-й значащей цифры. Абсолютное значение погрешности округления усечением не превышает единицы разряда, соответствующего последней оставляемой цифре.

Более предпочтительным является ***округление по дополнению****. Это правило округления состоит в следующем. Если первая слева из отбрасываемых цифр меньше 5, то сохраняемые цифры остаются без изменения. Если же она больше либо равна 5, то в младший сохраняемый разряд добавляется единица.*

Абсолютное значение погрешности округления при округлении по дополнению не превышает половины единицы разряда, соответствующего последней оставляемой цифре.

Пример 1.9

При округлении числа $x = 1.72631$ усечением до трех значащих цифр получится число $x^* = 1.72$, а при округлении по дополнению — число $x^* = 1.73$.

Пример 1.10

Округление значений $\Delta = 0.003721$ и $\delta = 0.0005427$ до двух значащих цифр дает значения $\Delta = 0.0037$ и $\delta = 0.00054$.

1.3 Погрешности арифметических действий

Погрешность суммы

Предложение 1. Абсолютная погрешность алгебраической суммы нескольких приближенных чисел не превышает суммы абсолютных погрешностей этих чисел.

Доказательство. Пусть даны приближенные числа $x_1^*, x_2^*, \ldots, x_n^*$ и их сумма

$$u^* = \pm x_1^* \pm x_2^* \pm \ldots \pm x_n^*.$$

Очевидно, что $\Delta u = \pm\Delta x_1 \pm \Delta x_2 \pm \ldots \pm \Delta x_n$, и, следовательно,

$$\Delta(u^*) = |\Delta u| \leqslant \left|\Delta(x_1^*)\right| + \ldots + \left|\Delta(x_n^*)\right| \text{ или}$$

$$\Delta(u^*) \leqslant \Delta_1 + \Delta_2 + \ldots + \Delta_n, \tag{1.10}$$

где $\Delta_i = \Delta(x_i^*)$, $i = 1, \ldots, n$.

Предложение 2. Если слагаемые одного и того же знака, то относительная погрешность их суммы не превышает наибольшей из относительных погрешностей слагаемых.

Доказательство. Пусть $u^* = x_1^* + x_2^* + \ldots + x_n^*$ и пусть $x_i^* > 0$. Обозначим через x_i точные величины ($x_i > 0$, $i = 1 \ldots n$) слагаемых x_i^*, а через $u = \sum x_i$ точное значение суммы. Тогда ОП суммы равна

$$\delta(u^*) = \frac{\Delta(u^*)}{u} = \frac{\Delta(x_1^*) + \Delta(x_2^*) + \ldots + \Delta(x_n^*)}{x_1 + x_2 + \ldots + x_n}. \tag{1.11}$$

Так как $\delta_i = \dfrac{\Delta(x_i^*)}{x_i}$, $(i = 1 \ldots n)$, то $\Delta(x_i^*) = x_i \delta(x_i^*)$. Подставляя в (1.11), получим

$$\delta(u) = \frac{\sum x_i \delta_i}{\sum x_i} \leqslant \frac{\delta_m \sum x_i}{\sum x_i} = \delta_m. \tag{1.12}$$

где δ_m — максимальная погрешность, то есть $\delta_m = \max\limits_i (\delta_i)$.

Погрешность разности

Рассмотрим разность двух приближенных чисел $u^* = x_1^* - x_2^*$. По формуле (1.10) имеем $\Delta(u) = \Delta(x_1^*) + \Delta(x_2^*)$.

Предложение 3. Для верхней границы относительной погрешности разности справедлива оценка

$$\delta(u^*) = \frac{\Delta(x_1^*) + \Delta(x_2^*)}{|u^*|} = \frac{\Delta(x_1^*) + \Delta(x_2^*)}{|x_1^* - x_2^*|} \leqslant \frac{\delta_m \cdot |x_1^* + x_2^*|}{|x_1^* - x_2^*|} = v \cdot \delta_m, \tag{1.13}$$

где $v = |x_1^* + x_2^*| / |(x_1^* - x_2^*)|$.

Если x_1^* и x_2^* близки, то $\delta(u^*)$ может быть очень велика, даже если $\Delta(x_1^*)$, $\Delta(x_2^*)$ малы, то есть происходит потеря точности.

Таким образом, неравенство (1.12) означает, что при суммировании чисел одного знака не происходит потери точности, если оценивать точность в относительных единицах. Совсем иначе обстоит дело при вычитании чисел одного знака. Здесь граница относительной погрешности возрастает в $v > 1$ раз, и возможна существенная потеря точности. Если числа x_1^* и x_2^* близки настолько, что $|x_1^* + x_2^*| \gg |x_1^* - x_2^*|$, то $v \gg 1$ и не исключена полная или почти полная потеря точности. Когда это происходит, говорят о том, что произошла *катастрофическая потеря точности*.

Пример 1.11

$x_1^* = 47.132$; $x_2^* = 47.111$. Все знаки верные. Тогда $u^* = x_1^* - x_2^* = 0.021$.

$$v = \left|\frac{x_1^* + x_2^*}{x_1^* - x_2^*}\right| = \frac{94.243}{0.021} = 4487.762 \approx 5000,$$

$$\delta(x_1^*) = \frac{10^{-3}}{47.132} \approx 2 \cdot 10^{-5}, \quad \delta(x_2^*) = \frac{10^{-3}}{47.111} \approx 2 \cdot 10^{-5}, \quad \delta(u^*) \approx 5 \cdot 10^3 \cdot 2 \cdot 10^{-5} = 10^{-1}.$$

Таким образом, погрешность разности в 5000 раз больше, чем относительные погрешности исходных данных.

При вычитании чисел необходимо применять следующее правило:

1. Следует избегать вычитания двух почти равных приближенных чисел.
2. Если все же приходится вычитать такие числа то следует брать их с $(m+n)$ верными знаками, где m — количество пропадающих старших разрядов; n — количество верных знаков, которые мы хотим получить в разности.
3. Либо делать эквивалентные преобразования.

Пример 1.12

Найти разность

$$u = \sqrt{2.01} - \sqrt{2} \tag{1.14}$$

с тремя верными знаками.

Решение:

Так как $\sqrt{2.01} = 1.41774$, $\sqrt{2} = 1.41421$, то $u = 0.00353 = 3.53 \cdot 10^{-3}$.

Этот же результат можно получить, если записать (1.14) в виде:

$$u = \frac{2.01 - 2}{\sqrt{2.01} + \sqrt{2}} = \frac{0.01}{\sqrt{2.01} + \sqrt{2}}$$

и взять корни лишь с тремя верными знаками

$$u = \frac{0.01}{1.42 + 1.41} = 3.53 \cdot 10^{-3}.$$

Погрешность произведения

Пусть $u^* = x_1^* \cdot x_2^*$, $x_i^* > 0$, $(i = 1,\ 2)$ — произведение приближенных чисел x_1^* и x_2^*, заданных с относительной погрешностью $\delta(x_1^*)$, $\delta(x_2^*)$; $u = x_1 \cdot x_2$ — произведение точных чисел x_1, x_2.

Предложение 4. Для относительной погрешности произведения справедлива оценка [1]

$$\delta(u^*) \leqslant \delta(x_1^*) + \delta(x_2^*) + \delta(x_1^*) \cdot \delta(x_2^*). \tag{1.15}$$

Доказательство. Имеем

$$\Delta(u^*) = |x_1 \cdot x_2 - x_1^* \cdot x_2^*| = \left|(x_1 - x_1^*)x_2 + (x_2 - x_2^*)x_1 - (x_1 - x_1^*)(x_2 - x_2^*)\right| \leqslant$$
$$\leqslant |x_2|\Delta(x_1^*) + |x_1|\Delta(x_2^*) + \Delta(x_1^*)\Delta(x_2^*) = |x_1 x_2| \cdot \left[\delta(x_1^*) + \delta(x_2^*) + \delta(x_1^*)\delta(x_2^*)\right].$$

Отсюда получим искомый результат.

На практике обычно в качестве оценки относительной погрешности произведения используют сумму относительных погрешностей сомножителей.

Погрешность частного

Пусть $u^* = x_1^*/x_2^*$, $x_i^* > 0$ — частное приближенных чисел.

Предложение 5. Для относительной погрешности частного справедлива оценка [1]

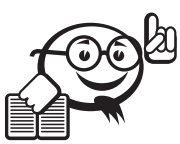
$$\delta(u^*) \leqslant \frac{\delta(x_1^*) + \delta(x_2^*)}{1 - \delta(x_2^*)}. \tag{1.16}$$

Доказательство. Имеем $|x_2^*| = \left|x_2 - (x_2 - x_2^*)\right| \geqslant |x_2| - \Delta_2 = |x_2|\left(1 - \delta\left(x_2^*\right)\right)$. Тогда

$$\delta(u^*) = \frac{|x_1/x_2 - x_1^*/x_2^*|}{|x_1/x_2|} = \frac{|x_1 x_2^* - x_2 x_1^*|}{|x_1 x_2^*|} = \frac{\left|x_2(x_1 - x_1^*) - x_1(x_2 - x_2^*)\right|}{|x_1 x_2^*|} \leqslant$$
$$\leqslant \frac{|x_2|\Delta(x_1^*) + |x_1|\Delta(x_2^*)}{|x_1 x_2^*|} \leqslant \frac{|x_2|\Delta(x_1^*) + |x_1|\Delta(x_2^*)}{|x_1 x_2|\left(1 - \delta(x_2^*)\right)} = \frac{\delta(x_1^*) + \delta(x_2^*)}{1 - \delta(x_2^*)}.$$

Предложение доказано.

Погрешность корня



Погрешность корня. Пусть $u^* = (x^*)^{1/m}$, тогда для относительной погрешности справедлива оценка

$$\delta(u^*) = \frac{1}{m}\delta(x^*). \tag{1.17}$$

То есть относительная погрешность (ОП) корня m-ой степени в m раз меньше относительной погрешности подкоренного числа.

Пример 1.13

Определить, с какой ОП и со сколькими верными цифрами можно найти сторону a квадрата, если его площадь $S = 12.34$ *известна с АП* $\Delta(S) = 0.01$.

Решение:

$$\delta(S) = \frac{0.01}{12.34} \approx 0.0008; \quad \delta(a) = \frac{1}{2}\delta(S) = 0.0004;$$
$$n = 1 - \lg\bigl(\alpha_m \cdot \delta(a)\bigr) = 1 - \lg(3 \cdot 0.0004) \approx 4; \quad a = \sqrt{S} = 3.513.$$

Погрешность функции

Пусть задана дифференцируемая функция $u^* = f\,(x_1^*, x_2^* \ldots x_n^*)$ и $\Delta_i = \Delta(x_i^*)$ — абсолютные погрешности аргументов функции.

Тогда ***абсолютная погрешность функции*** *равна:*

$$\Delta(u^*) = |\Delta u| \approx \left|df(x_1^*, \ldots, x_n^*)\right| = \left|\sum_{i=1}^{n} \frac{\partial f}{\partial x_i^*}\Delta x_i\right| \leqslant \sum_{i=1}^{n} \left|\frac{\partial f}{\partial x_i^*}\right| \cdot \Delta(x_i^*)$$

или

$$\Delta(u^*) \leqslant \sum_{i=1}^{n} \left|\frac{\partial f}{\partial x_i^*}\right| \Delta_i. \tag{1.18}$$

Поделив на u^*, *получим ОП*

$$\delta(u^*) = \sum \left|\frac{\partial}{\partial x_i^*} \ln u^*\right| \cdot \Delta_i. \tag{1.19}$$

Пример 1.14

Найти АП и ОП объема шара $V = \frac{1}{6}\pi d^3$, *если* $d = 3.7 \pm 0.05$ *см,* $\pi = 3.14$.

Решение:

$$\Delta(V) = \frac{\partial V}{\partial \pi}|\Delta\pi| + \frac{\partial V}{\partial d}|\Delta d| = 8.44 \cdot 0.01 + 21.5 \cdot 0.05 = 1.1 \text{ см}^2,$$
$$V = \frac{1}{6}\pi d^3 = (27.4 \pm 1.1) \text{ см}^3; \quad \delta(V) = \frac{\Delta(V)}{V} = \frac{1.1}{27.4} = 4\%.$$

1.4 Обратная задача теории погрешностей

На практике важна также обратная задача: каковы должны быть абсолютные погрешности аргументов функции, чтобы АП функции не превышала заданной величины.

Эта задача математически не определена, так как заданную АП $\Delta(u^*)$ функции $u^* = f(x_1^*, \ldots, x_n^*)$ можно обеспечить устанавливая по-разному предел абсолютной погрешности $\Delta(x_i^*)$ ее аргументов.

Простейшее решение обратной задачи дается ***принципом равных влияний*** [3]. *Согласно этому принципу предполагается, что все слагаемые в сумме погрешностей одинаково влияют на образование общей абсолютной погрешности* $\Delta(u^*)$.

Пусть $\Delta(u^*)$ задана. Тогда имеем

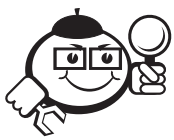
$$\Delta(u^*) = \sum_{i=1}^{n} \left| \frac{\partial f}{x_i^*} \right| \Delta(x_i^*).$$

Полагая, что все слагаемые равны между собой, будем иметь

$$\left| \frac{\partial f}{\partial x_1^*} \right| \Delta(x_1^*) = \left| \frac{\partial f}{\partial x_2^*} \right| \Delta(x_2^*) = \ldots = \left| \frac{\partial f}{\partial x_n^*} \right| \Delta(x_n^*) = \frac{\Delta(u^*)}{n}.$$

Отсюда

$$\Delta(x_i^*) = \frac{1}{n} \cdot \frac{\Delta(u^*)}{\left| \frac{\partial f}{\partial x_i^*} \right|} \quad (i = 1, \ldots, n). \tag{1.20}$$



Пример 1.15

Радиус основания цилиндра $R \approx 2$ *м; высота* $H \approx 3$ *м. С какими абсолютными погрешностями нужно определить* R *и* H, *чтобы его объем можно было вычислить с точностью до* 0.1 $м^3$.

Решение:

$V = \pi R^2 H$, $\dfrac{\partial V}{\partial \pi} = R^2 H = 12$; $\dfrac{\partial V}{\partial R} = 2\pi RH = 37.7$; $\dfrac{\partial V}{\partial H} = \pi R^2 = 12.6$; $n = 3$ (количество слагаемых).

$$\Delta\pi = \frac{1}{3}\cdot\frac{0.1}{\dfrac{\partial V}{\partial\pi}} = \frac{0.1}{3\cdot 12} \approx 0.003; \quad \Delta R = \frac{1}{3}\cdot\frac{0.1}{\dfrac{\partial V}{\partial R}} = \frac{0.1}{3\cdot 37.7} \approx 0.001;$$

$$\Delta(H) = \frac{1}{3}\cdot\frac{0.1}{\dfrac{\partial V}{\partial H}} = \frac{0.1}{3\cdot 12.6} \approx 0.003.$$

Контрольные вопросы по главе 1

1. Как связаны между собой относительная и абсолютная погрешности приближенного числа?
2. Что такое значащие цифры числа?
3. Дайте определение верных значащих цифр приближенного числа.
4. Как связана абсолютная погрешность числа с количеством верных значащих цифр этого числа?
5. Как связана относительная погрешность числа с количеством верных значащих цифр этого числа?
6. Как произвести округление числа до n значащих цифр?
7. Найти число верных знаков частного $U = 230/23$, если все цифры делимого и делителя верны.
8. Вычислить два числа $U_1 = (x^*)^{0.5}$ и $U_2 = (x^*)^2$ при $x^* = 9$. Какой из результатов будет точней и во сколько раз?
9. В чем состоит принцип равных влияний?
10. Со сколькими знаками нужно взять $x = 21^{1/2}$, чтобы относительная погрешность была не более 1%?

Глава 2

КОРРЕКТНОСТЬ И ОБУСЛОВЛЕННОСТЬ ВЫЧИСЛИТЕЛЬНЫХ ЗАДАЧ И АЛГОРИТМОВ

2.1 Постановка вычислительной задачи

*Под **вычислительной задачей** будем понимать одну из трех задач, которые возникают при анализе математических моделей: прямую задачу, обратную задачу и задачу идентификации. Слово «вычислительная» подчеркивает, что основные усилия будут направлены на то, чтобы найти (вычислить) ее решение.*

Будем считать, что постановка задачи включает в себя задание *множества допустимых входных данных* X и *множества возможных решений* Y. Цель вычислительной задачи состоит в нахождении решения $y \in Y$ по заданному входному данному $x \in X$. Входные данные и решение могут быть числами, наборами чисел (векторами, матрицами, последовательностями) и функциями. Предположим, что для оценки величин погрешностей приближенных входных данных x^* и приближенного решения y^* введены абсолютные и относительные погрешности Δ_x, Δ_y, δ_x, δ_y. Будем также использовать введенные ранее обозначения для погрешностей x^* и y^*: $\Delta(x^*)$, $\delta(x^*)$; $\Delta(y^*)$, $\delta(y^*)$.

Определение корректности задачи

Анализ важнейших требований, предъявляемых к различным прикладным задачам, приводит к понятию корректности математической задачи, которое было впервые сформулировано Жаком Адамаром и развито затем И. Г. Петровским.

Вычислительная задача называется ***корректной*** *(по Адамару — Петровскому), если выполнены следующие три условия* [1]:

1) *ее решение* $y \in Y$ *существует при любых входных данных* $x \in X$;
2) *это решение единственно;*
3) *решение устойчиво по отношению к малым возмущениям входных данных.*

В том случае, когда хотя бы одно из этих требований не выполнено, задача называется *некорректной*.

Существование решения вычислительной задачи — естественное требование к ней. Отсутствие решения может свидетельствовать, например, о непригодности принятой математической модели либо о неправильной постановке задачи. Иногда отсутствие решения является следствием неправильного выбора множества допустимых входных данных X или множества возможных решений Y. Например, если искать корни квадратного уравнения $x^2 + bx + c = 0$ на множестве вещественных чисел, то решение задачи существует только в том случае, если множество значений коэффициентов уравнения удовлетворяет условию $b^2 - 4c \geqslant 0$.

Единственность решения для многих задач является естественным свойством; для других же задач решение может быть и не единственным. Например, квадратное уравнение имеет два корня. Если задача имеет реальное содержание, то неединственность может быть устранена путем введения дополнительных ограничений (т. е. сужением множества Y).

Устойчивость решения. Решение y вычислительной задачи называется *устойчивым по входным данным* x, если оно зависит от входных данных непрерывным образом. Это означает, что для любого $\varepsilon > 0$ существует $\delta(\varepsilon)$, такое, что всякому исходному данному x^*, удовлетворяющему условию $\Delta(x^*) < \varepsilon$, отвечает приближенное решение y^*, для которого $\Delta(y^*) < \delta$. Таким образом, для устойчивой вычислительной задачи ее решение теоретически можно найти со сколь угодно высокой точностью δ, если обеспечена достаточно высокая точность входных данных ε.

Например, задача вычисления производной $u(x) = f'(x)$ приближенно заданной функции является неустойчивой. Действительно, пусть $f^*(x) = f(x) + \alpha \sin(x/\alpha^2)$ — приближенно заданная на отрезке $[a, b]$ непрерывно дифференцируемая функция ($\alpha \ll 1$), где $f(x)$ — точно заданная функция на отрезке $[a, b]$. Тогда $u^*(x) = u(x) + \frac{1}{\alpha}\cos(x/\alpha^2)$. Для абсолютной погрешности функции и ее производной получим: $\Delta(f^*) = \alpha$, $\Delta(u^*) = 1/\alpha$. Таким образом, сколь угодно малой погрешности задания функции f может отвечать сколь угодно большая погрешность производной $f'(x)$.

2.2 Обусловленность вычислительной задачи

Пусть вычислительная задача корректна (ее решение существует, единственно и устойчиво по входным данным). Теоретически наличие у задачи устойчивости означает, что ее решение может быть найдено со сколь угодно малой погрешностью, если только гарантировать, что погрешности входных данных достаточно малы. Однако на практике погрешности входных данных не могут быть сделаны сколь угодно малыми, точность их ограничена. Даже то, что исходные данные нужно будет ввести в компьютер, означает, что их относительная точность будет заведомо ограничена величиной порядка $\varepsilon_м$ (относительной точностью компьютера или машинной точностью). В реальности, конечно, уровень ошибок в исходной информации будет существенно выше. Как же повлияют малые, но конечные погрешности входных данных на решение, как сильно способны они исказить желаемый результат? Для ответа на этот вопрос введем новые понятия.

*Под **обусловленностью вычислительной задачи** понимают чувствительность ее решения к малым погрешностям входных данных. Задачу называют **хорошо обусловленной**, если малым погрешностям входных данных отвечают малые погрешности решения, и **плохо обусловленной**, если возможны сильные изменения решения* [1].

Введем количественную меру степени обусловленности вычислительной задачи — *число обусловленности*. Эту величину можно интерпретировать как коэффициент возможного возрастания погрешностей в решении вследствие наличия погрешностей входных данных.

Пусть между абсолютными и относительными погрешностями входных данных x и решения y установлено неравенство:

$$\Delta(y^*) \leqslant \nu_\Delta \Delta(x^*), \quad \delta(y^*) \leqslant \nu_\delta \delta(x^*). \tag{2.1}$$

Тогда величина ν_Δ называется *абсолютным числом обусловленности*, а ν_δ называют *относительным числом обусловленности*. Чаще все же под числом обусловленности понимают относительное число обусловленности. Для *плохо обусловленной задачи* $\nu \gg 1$. В некотором смысле неустойчивость задачи — это крайнее проявление плохой обусловленности, отвечающее значению $\nu = \infty$. Если $\nu \sim 10^N$, где ν — относительное число обусловленности, то порядок N показывает число верных цифр, которое может быть утеряно в результате по сравнению с числом верных цифр входных данных.

Каково то значение ν, при котором следует признать задачу плохо обусловленной? Ответ на этот вопрос существенно зависит, с одной стороны, от предъявляемых требований к точности решения и, с другой — от уровня обеспечиваемой точности исходных данных. Например, если требуется найти решение с точностью 0.1%, а входная информация задается с точностью 0.02%, то уже значение $\nu = 10$ сигнализирует о плохой обусловленности. Если же исходные данные задаются с точностью не ниже 0.0001%, то даже при $\nu = 10^3$ мы получим результат с точностью 0.1% и задача будет хорошо обусловлена.

Примером плохо обусловленной задачи является задача вычитания приближенных чисел одного знака. Для этой задачи относительное число обусловленности $v = |a + b| / |a - b|$, так как относительная погрешность разности $y = a - b$ равна $\delta(y) = v \cdot \delta_m$. Здесь в качестве δ_m взята максимальная погрешность, т. е. $\delta_m = \max\big(\delta(a), \delta(b)\big)$. Если вычитаются близкие числа, то $v \gg 1$ и происходит катастрофическая потеря точности.

Рассмотрим обусловленность задачи вычисления значения функции одной переменной $y = f(x)$. Абсолютная погрешность равна $\Delta(y) = \big|f'(x)\big|\Delta(x)$, относительная погрешность $\delta(y) = v \cdot \delta(x)$, где v — относительное число обусловленности.

$$v = |x| \big|f'(x)\big| / \big|f(x)\big| \tag{2.2}$$

В качестве примера рассмотрим функцию $y = \sin x$. Согласно формуле (2.2) имеем $v = |x \cdot \operatorname{ctg} x|$. На рис. 2.1 приведен график этой функции для $x \geqslant 0$.

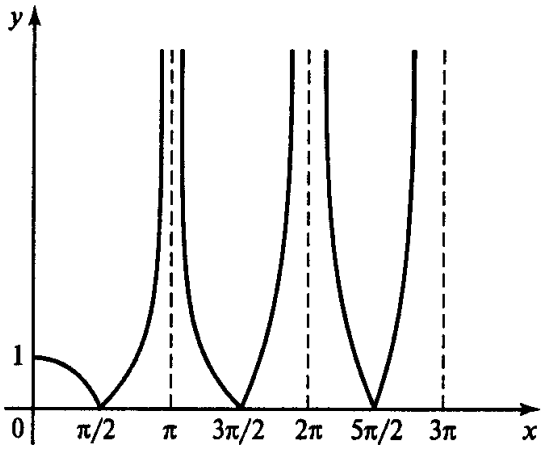


Рис. 2.1

Так как $v \to \infty$ при $x \to \pi \cdot k$ (для $k = 1, 2, \ldots$), то при $x \approx \pi k$ задача обладает плохой обусловленностью. Поэтому при вычислении функции $y = \sin x$ необходимо проводить вычисления так, чтобы аргумент находился в диапазоне $|x| < 2$, поскольку здесь $v \leqslant 1$.

2.3 Корректность вычислительных алгоритмов

Вычислительный алгоритм

Вычислительный метод, доведенный до степени детализации, позволяющий реализовать его на компьютере, принимает форму вычислительного алгоритма.

*Определим **вычислительный алгоритм** как точное предписание действий над входными данными, задающее вычислительный процесс, направленный на преобразование произвольных входных данных x (из множества допустимых для данного алгоритма входных данных X) в полностью определяемый этими входными данными результат.*

Реальный вычислительный алгоритм складывается из двух частей: *абстрактного вычислительного алгоритма*, формулируемого в общепринятых математических терминах, и *программы*, записанной на одном из алгоритмических языков и предназначенной для реализации алгоритма на компьютере. Как правило, в руководствах по методам вычислений излагаются именно абстрактные алгоритмы, но их обсуждение проводится так, чтобы выявить особенности алгоритмов, которые оказывают существенное влияние на качество программной реализации.

Определение корректности алгоритма

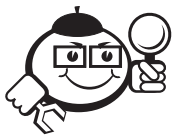
К вычислительным алгоритмам, предназначенным для широкого использования, предъявляется ряд весьма жестких требований. Первое из них — корректность алгоритма.

Будем называть вычислительный алгоритм ***корректным****, если выполнены три условия* [1]:

1) *он позволяет после выполнения конечного числа элементарных для вычислительной машины операций преобразовать любое входное данное* $x \in X$ *в результат* y;
2) *результат* y *устойчив по отношению к малым возмущениям входных данных;*
3) *результат* y *обладает вычислительной устойчивостью.*

Если хотя бы одно из перечисленных условий не выполнено, то будем называть алгоритм *некорректным*. Уточним и более подробно обсудим эти условия.

Необходимость выполнения первого условия понятна. Если для получения результата нужно выполнить бесконечное число операций либо требуются операции, не реализованные на компьютерах, то алгоритм следует признать некорректным.



Пример 2.1

Известный алгоритм деления чисел «углом» некорректен, так как он может продолжаться бесконечно, если не определен критерий окончания вычислений [1].

Пример 2.2

Алгоритм вычисления корней квадратного уравнения $ax^2 + bx + c = 0$ по формулам $x_{1,2} = \left(-b \pm \sqrt{b^2 - 4ac}\right)/2a$ некорректен, если он предназначен для использования на вычислительной машине, на которой не реализована операция извлечения квадратного корня [1].

Устойчивость по входным данным

Устойчивость результата y к малым возмущениям входных данных *(устойчивость по входным данным)* означает, что результат непрерывным образом зависит от входных данных при условии, что отсутствует вычислительная погрешность. Это требование устойчивости аналогично требованию устойчивости вычислительной задачи. Отсутствие такой устойчивости делает алгоритм непригодным для использования на практике.

Пример 2.3

Пусть алгоритм предназначен для вычисления корней квадратного уравнения $ax^2 + bx + c = 0$ с коэффициентами, удовлетворяющими условию $d = b^2 - 4ac \geqslant 0$. Если в нем используются формулы $x_{1,2} = \left(-b \pm \sqrt{b^2 - 4ac}\right)/2a$, то этот алгоритм некорректен. В самом деле, значение d^*, отвечающее приближенно заданным коэффициентам b^* и c^*, может оказаться отрицательным, если $d \approx 0$. Тогда вычисления завершатся аварийным остановом при попытке извлечь квадратный корень из отрицательного числа. Если же в формуле $x_{1,2}$ заменить d на $\max(d, 0)$, то алгоритм становится корректным [1].

Вычислительная устойчивость

Из-за наличия погрешностей округления при вводе входных данных в компьютер и при выполнении арифметических операций неизбежно появление вычислительной погрешности. На разных компьютерах она различна из-за различий в разрядности и способах округления, но для фиксированного алгоритма в основном значение погрешности определяется машинной точностью $\varepsilon_м$.

*Назовем алгоритм **вычислительно устойчивым**, если вычислительная погрешность результата стремится к нулю при $\varepsilon_м \to 0$. Обычно вычислительный алгоритм называют **устойчивым**, если он устойчив по входным данным и вычислительно устойчив, и — **неустойчивым**, если хотя бы одно из этих условий не выполнено.*

Пример 2.4

Пусть требуется составить таблицу значений интегралов $I_n = \int_0^1 x^n e^{1-x}\,dx$ для $n = 1, 2, \ldots$ на 6-разрядном десятичном компьютере. Интегрируя по частям, имеем $I_n = -1 + \int_0^1 nx^{n-1}e^{1-x}\,dx$. Поэтому мы можем записать

$$I_n = nI_{n-1} - 1,\ n \geqslant 1. \tag{2.3}$$

При этом для I_0 имеем

$$I_0 = \int_0^1 e^{1-x}dx = e - 1 \approx 1.71828.$$

$$I_1 = 1 \cdot I_0 - 1 = 0.71828; \quad I_2 = 2I_1 - 1 = 0.43656; \quad I_3 = 3I_2 - 1 = 0.30968; \ \ldots;$$
$$I_9 = 9I_8 - 1 = -0.55360; \quad I_{10} = 10I_9 - 1 = -6.5360; \ \ldots$$

Здесь вычисления следует закончить, так как мы получаем отрицательные значения интегралов, которые должны быть положительны. В чем же причина появления такой большой погрешности? В данном примере все вычисления проводились точно, а единственная и, на первый взгляд, незначительная ошибка была сделана при округлении значения I_0 до шести значащих цифр (заметим, что $\Delta_0 = |I_0 - I_0^*| \simeq 10^{-6}$). Однако при вычислении I_1 эта погрешность сохранилась, при вычислении I_2 умножилась на 2!, при вычислении I_3 — на 3!, ..., при вычислении I_9 — на 9! и т. д.

Таким образом, $\Delta_n = |I_n - I_n^*| \simeq n!10^{-6}$. Уже при $n = 9$ имеем $9! = 362\,880$, и поэтому $\Delta_9 = 9!\Delta_0 \approx 0.36$.

Если вычисления производятся без ограничений на число n, то рассматриваемый алгоритм следует признать вычислительно неустойчивым. Погрешности растут пропорционально $n!$ настолько быстро, что уже при довольно скромных значениях n попытки добиться приемлемого результата даже за счет увеличения разрядности мантиссы заранее обречены на неудачу [1].

Как изменить алгоритм, чтобы сделать его устойчивым? Перепишем формулу (2.3) в виде

$$I_{n-1} = \frac{I_n + 1}{n} \tag{2.4}$$

и будем вести вычисления значений I_n в обратном порядке, начиная, например, с $n = 54$. Положим $I_{54} \approx 0$. Так как точное значение интеграла $I_{54} \leqslant e\int_0^1 x^{54}dx = e/55$, то $\Delta_{54} \leqslant e/55 \approx 5 \cdot 10^{-2}$.

Однако при вычислении I_{53} эта погрешность уменьшится в 54 раза, при вычислении I_{52} — еще в 53 раза и т. д. В результате значения I_n при $n = 50, \ldots, 1$ будут вычислены с 6-ю верными значащими цифрами. Здесь погрешности не растут, а затухают. Поэтому этот алгоритм вычислительно устойчив.

. . . . . .

2.4 Требования к вычислительным алгоритмам

Выше были сформулированы два важнейших требования — корректность и хорошая обусловленность. Кроме них, к алгоритмам предъявляется еще целый ряд существенных требований.

2.4.1 Требования к абстрактным алгоритмам

К числу этих требований относятся: экономичность, надлежащая точность, экономия памяти, простота.

Экономичность алгоритма измеряется числом элементарных операций, необходимых для его реализации, и в конечном итоге сводится к затратам машинного времени. Это требование формулируют иногда как требование максимальной быстроты исполнения алгоритма. Экономичность особенно важна при массовых расчетах. Естественно, что при создании алгоритмов большое внимание уделяют минимизации числа операций. Для некоторых задач разработаны алгоритмы, требующие минимально возможного числа операций. Пример такого алгоритма — схема Горнера.

Пример 2.5

Пусть задача состоит в вычислении значения многочлена

$$P_n(x) = a_0 + a_1x + a_2x^2 + \ldots + a_nx^n \tag{2.5}$$

по заданным коэффициентам a_0, a_1, …, a_n и значению аргумента x. Если вычислять многочлен непосредственно по формуле (2.5), причем x^2, x^3, …, x^n находить последовательным умножением на x, то потребуется выполнить $2n - 1$ операций умножения и n операций сложения.

Значительно более *экономичным* является метод вычисления, называемый схемой Горнера [1, 3]. Он основан на записи многочлена в следующем эквивалентном виде:

$$P_n(x) = \Big(\big(\ldots\big((a_nx + a_{n-1})x + a_{n-2}\big)x + \ldots\big)x + a_1\Big)x + a_0.$$

Вычисления полинома теперь можно вести в следующем порядке: $S_0 = a_n$, $S_1 = S_0x + a_{n-1}$, $S_2 = S_1x + a_{n-2}$, …, $S_n = S_{n-1}x + a_0$. Здесь для вычисления значения полинома $P_n(x) = S_n$ требуется n операций умножения и n операций сложения.

Даже для самых простых задач выбор экономичного алгоритма может дать существенное уменьшение числа операций.

Пример 2.6

Пусть требуется вычислить x^n, где n — натуральное число. Вычисление этой величины последовательным умножением на x предполагает выполнение $(n - 1)$ операций умножения. Нетрудно убедиться в том, что этот способ не самый экономичный. Например, x^{64} можно найти, выполнив не 63, а всего шесть операций умножения, если последовательным возведением в квадрат вычислить x^2, x^4, x^8, x^{16}, x^{32}, x^{64}.

В общем случае представим n в виде разложения по степеням числа 2 (именно так число n хранится в памяти компьютера). Тогда

$$x^n = \left(x^{2^L}\right)^{\alpha_L} \cdot \left(x^{2^{L-1}}\right)^{\alpha_{L-1}} \ldots \left(x^2\right)^{\alpha_1} \cdot x^{\alpha_0}, \qquad (2.6)$$

где α_i — принимает значения 0 или 1. В формуле (2.6) следует учитывать только те сомножители, для которых $\alpha_i = 1$. Алгоритм, основанный на разложении (2.6), называется бинарным алгоритмом. Он позволяет найти x^n не более чем за $2 \cdot \log_2 n$ операций умножения [1, 3].

Требование *точности* означает, что вычислительный алгоритм должен давать решение задачи с заданной или приемлемой для задачи точностью ε.

Важным является требование *экономии памяти*. Хотя в последнее время доступная память компьютеров существенно расширилась, для «больших» задач требование экономии памяти может в ряде случаев стать основным. Интерес к экономному размещению информации в памяти возрастает в связи со все более широким использованием персональных компьютеров для решения научно-технических и инженерных задач.

Учитывая необходимость дальнейшей программной реализации алгоритма, подчеркнем, что *простота алгоритма* также является весьма желательным его свойством.

2.4.2 Требования к программным реализациям алгоритмов

К настоящему времени выработан ряд требований к программам, реализующим вычислительные алгоритмы и предназначенным для длительного и широкого использования. Перечислим некоторые из них: надежность, работоспособность (робастность), переносимость (портабельность), поддерживаемость, простота в использовании и др. Рассмотрим эти требования более подробно.

Надежность программы означает, что она не содержит ошибок и вычисляет именно тот результат, для которого она предназначена.

Работоспособность (робастность) включает в себя надежность и предполагает, что программа способна выявлять недопустимые исходные данные, обнаруживать различные критические для задачи или алгоритма ситуации. Робастная программа реагирует на такие ситуации приемлемым для пользователя образом. Она составлена так, чтобы исключить какие-либо аварийные остановы, в том числе по переполнению, из-за деления на нуль, неудачной попытки применить операцию извлечения квадратного корня или при взятии логарифма от отрицательного числа.

Алгоритм может «потерпеть неудачу» при решении задачи, если заданное входное данное не является для него допустимым. Конечно, в простых ситуациях пользователь должен сам различать допустимые для алгоритма входные данные от недопустимых. Однако чаще всего сделать это до вычислений очень трудно или невозможно, и в программе должны быть предусмотрен анализ данных и сообщение пользователю о недопустимых или сомнительных данных. Необходимо исключить ситуацию, характерную для некачественных программ, когда реакцией на задание данных, при которых алгоритм не может по объективным причинам найти решение задачи, является аварийный останов или же выдача внешне вполне правдоподобного, но совершенно бессмысленного результата.

Переносимость (портабельность) означает, что программа может работать на различных компьютерах без изменения или с незначительными изменениями. Всякая характеристика компьютера, используемая в программе (например, значение машинного эпсилон $\varepsilon_м$), должна или вычисляться самой программой, или задаваться пользователем.

Поддерживаемость означает прежде всего требование легкости модификации. Для того чтобы была возможность внесения в программу изменений с минимальной вероятностью появления ошибок, она должна быть составлена максимально ясно и логично. Полезно вносить в текст программы содержательные комментарии. Разобраться в плохо составленной программе может оказаться труднее, чем создать новую. Поддерживаемая программа должна быть хорошо документирована. Плохое описание программы в лучшем случае способно вызвать к ней недоверие, а в худшем — может не позволить пользователю правильно ее эксплуатировать. К сожалению, нередка ситуация, когда предназначенная для широкого использования программа настолько плохо документирована, что пользователь предпочитает потратить время на написание аналогичной программы (возможно, гораздо худшего качества) либо вообще отказаться от решения задачи.

Простота в использовании программы — весьма желательное, но трудно достижимое свойство. Часто добиться простоты в использовании можно только жертвуя надежностью или экономичностью. Существует ряд широко используемых программ, которые, в первую очередь, популярны благодаря простоте в использовании.

2.4.3 Противоречивость требований

Можно продолжить перечисление требований к вычислительным алгоритмам, добавив, например, требования универсальности и гибкости. Однако нетрудно понять, что сформулированные требования противоречивы. Большинство из них вступает в противоречие с экономичностью, выраженной через затраты машинного времени. В разных ситуациях на первый план может выступать то или иное требование, и, удовлетворяя в большей степени одним требованиям, программа с неизбежностью в меньшей степени удовлетворяет другим. Это частично объясняет наличие большого числа программ, предназначенных для решения одной и той же задачи.

Естественно, что хорошая программа, которую можно предъявить для широкого использования, не может быть простой. Следует признать, что составление таких программ — это работа, требующая высокой квалификации и специальных знаний. Ее выполняют специалисты по созданию математического обеспечения компьютеров. Рядовой пользователь должен по возможности стремиться максимально использовать стандартные программы, а не создавать новые.

Контрольные вопросы по главе 2

1. Дайте определение корректности вычислительной задачи.
2. Какая задача называется устойчивой?
3. Что такое обусловленность вычислительной задачи?
4. Что понимают под вычислительным алгоритмом?
5. Что включает в себя реальный вычислительный алгоритм?
6. Какой алгоритм называется корректным?
7. Что понимают под устойчивостью алгоритма?
8. Перечислите требования к абстрактным алгоритмам.
9. Перечислите требования к программным реализациям алгоритмов.

Глава 3

ПРИБЛИЖЕННОЕ РЕШЕНИЕ НЕЛИНЕЙНЫХ УРАВНЕНИЙ С ОДНОЙ ПЕРЕМЕННОЙ

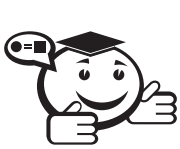
3.1 Локализация корней

Пусть дано уравнение

$$f(x) = 0, \tag{3.1}$$

где $f(x)$ определена и непрерывна в некотором конечном или бесконечном интервале $a \leqslant x \leqslant b$.

Всякое значение ξ, обращающее функцию $f(x)$ в нуль, то есть такое, что $f(\xi) = 0$, называется ***корнем уравнения*** (3.1)*, или нулем функции $f(x)$.*



Число ξ называется ***корнем k-ой кратности****, если при $x = \xi$ вместе с функцией $f(x)$ обращаются в нуль ее производные до $(k-1)$ порядка включительно:*

$$f(\xi) = f'(\xi) = \ldots = f^{k-1}(\xi) = 0.$$

Однократный корень называется простым.

Приближенное нахождение корней уравнения (3.1) обычно складывается из двух этапов:

1. Локализация (отделение) корней, то есть установление интервалов $[\alpha_i, \beta_i]$, в которых содержится один корень уравнения (3.1).
2. Итерационное уточнение корней, то есть доведение их до заданной точности.

Для локализации корней полезна следующая теорема [3].

Теорема 3.1. Если непрерывная функция $f(x)$ принимает значения разных знаков на концах отрезка $[a, b]$, то есть $f(a) \cdot f(b) < 0$, то внутри этого отрезка содержится по меньшей мере один корень уравнения $f(x) = 0$, то есть найдется хотя бы одно число $\xi \in (a, b)$, такое, что $f(\xi) = 0$.

Корень заведомо единственный, если $f'(x)$ существует и сохраняет постоянный знак внутри интервала $[a, b]$.

Рассмотрим способ локализации корней. В заданном интервале $[a, b]$ задается сетка $x_i = a + i \cdot h$, $i = 1, \ldots, n$; $h = (b-a)/n$ и вычисляются значения функции $f(x_i)$ (достаточно определить лишь знаки в узлах x_i). Если окажется, что $f(x_i) f(x_{i+1}) < 0$, то в силу теоремы 3.1 в интервале (x_i, x_{i+1}) имеется корень уравнения. Если за корень взять $\xi_i = 1/2\ (x_i + x_{i+1})$, то точность нахождения корня будет равна половине интервала $h/2$. Нужно еще убедиться, является ли найденный корень единственным. Для этого достаточно провести процесс половинного деления, деля интервал $[x_i; x_{i+1}]$ на две, четыре и т. д. равных частей и определить знаки функции $f(x)$ в точках деления. При делении мы повышаем точность определения корня.

Пример 3.1

Определить корни уравнения

$$f(x) = x^3 - 6x + 2 = 0. \tag{3.2}$$

Решение:

Составляем приблизительную схему, т. е. для заданного набора входных данных вычисляем знак функции.

Таблица 3.1

x	−4	−3	−2	−1	0	1	2	3	4
$f(x)$	−	−	+	+	+	−	−	+	+

Из таблицы 3.1 видно, что уравнение (3.2) имеет три действительных корня, лежащих в интервалах $(-3, -2)$, $(0, 1)$ и $(2, 3)$.

Итак, мы выделили интервалы, в которых содержится единственный корень. Рассмотрим теперь методы уточнения корней. Поскольку все методы итерационные, то необходимо дать определение сходимости последовательности чисел (или сходимости итерационного процесса) [7].

Говорят, что ***итерационный метод сходится*** *с линейной скоростью, если в области сходимости справедлива оценка*

$$|\xi - x_{k+1}| \leqslant \alpha \cdot |\xi - x_k|, \ 0 < \alpha < 1. \tag{3.3}$$

В неравенстве (3.3) x_k *— k-ое приближение корня,* ξ *— корень (нуль функции),* α *— коэффициент сходимости.*

Число r $(r \geqslant 1)$ *называют* ***порядком сходимости метода****, если в области сходимости имеет место оценка*

$$|\xi - x_{k+1}| \leqslant \alpha \cdot |\xi - x_k|^r, \ \alpha > 1. \tag{3.4}$$

Если $r = 1$*, то метод обладает линейной сходимостью, при* $r = 2$ *— квадратичной,* $r = 3$ *— кубической и т. д.*

Справедливы следующие утверждения.

Лемма 3.1. Если итерационный процесс обладает линейной скоростью сходимости (порядок сходимости $r = 1$), то имеет место оценка погрешности корня $|\xi - x_k| \leqslant \alpha^k |\xi - x_0|$.

Лемма 3.2. Если итерационный процесс обладает сверхлинейной скоростью сходимости $(r > 1)$, то справедлива оценка погрешности корня $|\xi - x_k| \leqslant \alpha \cdot c^{r^k}$, $c = \alpha^{1/(r-1)} \cdot |\xi - x_0|$.

3.2 Обусловленность задачи вычисления корня

Пусть ξ — корень уравнения $f(x) = 0$, который мы ищем. Будем считать, что входными данными для задачи вычисления корня ξ являются значения $f(x)$ в малой окрестности корня. Так как значения $f(x)$ будут вычисляться на компьютере, то в действительности мы имеем приближенные значения $f^*(x)$. Погрешности в значениях $f^*(x)$ могут быть связаны не только с неизбежными ошибками округления, но и с использованием для вычисления значений функции f приближенных

методов. При этом относительная погрешность $\delta(f)$ может оказаться достаточно большой. Реально рассчитывать можно лишь на то, что малой окажется абсолютная погрешность вычисления значений функции.

Будем предполагать, что в достаточно малой окрестности корня выполняется неравенство $|f(x) - f^*(x)| < \Delta(f)$, где $\Delta(f)$ — граница абсолютной погрешности. Сама погрешность функции ведет себя нерегулярно (осциллирует) и в первом приближении может восприниматься пользователем как некоторая случайная величина.

Если функция f непрерывна, то найдется такая малая окрестность $(\xi - \varepsilon, \xi + \varepsilon)$ корня ξ, имеющая радиус $\varepsilon > 0$, в которой выполняется неравенство

$$|f(x)| \leqslant \Delta(f). \tag{3.5}$$

Для $x \in (\xi - \varepsilon, \xi + \varepsilon)$ знак вычисленного значения $f^*(x)$ не обязан совпадать со знаком $f(x)$ и, следовательно, невозможно определить, какое именно значение x из интервала $(\xi - \varepsilon, \xi + \varepsilon)$ обращает функцию f в нуль.

Будем называть этот интервал *интервалом неопределенности корня* ξ. Найдем оценку величины ε. Пусть корень ξ — простой. Для близких к ξ значений x справедливо приближенное равенство

$$f(x) \approx f(\xi) + f'(\xi)(x - \xi). \tag{3.6}$$

Подставим в неравенство (3.5) и с учетом того, что $f(\xi) = 0$, получим:

$$|(x - \xi)| \leqslant \frac{\Delta(f)}{|f'(\xi)|}.$$

Отсюда

$$\varepsilon \approx v_\Delta \cdot \Delta(f), \tag{3.7}$$

где $v_\Delta = \dfrac{1}{|f'(x)|}$ — число, которое в рассматриваемой задаче играет роль абсолютного числа обусловленности.

Заметим, что радиус интервала неопределенности прямо пропорционален погрешности $\Delta(f)$ вычисления значения f. Кроме того, с уменьшением $|f'(\xi)|$ интервал неопределенности ε возрастает (обусловленность задачи ухудшается). Величина $f'(\xi)$ представляет собой тангенс угла наклона касательной в т. ξ. Таким образом, с уменьшением модуля тангенса угла наклона, под которым график функции f пересекает ось Ox, обусловленность задачи ухудшается.

Если же $f'(\xi) = 0$ (т. е. корень ξ — кратный), то формула (3.7) уже не верна. Пусть кратность корня равна m. Тогда вместо (3.6) мы можем записать приближенное равенство:

$$f(x) \approx f(\xi) + f'(\xi)(x - \xi) + \frac{f''(\xi)}{2!}(x - \xi)^2 + \ldots + \frac{f^{(m-1)}(\xi)}{(m-1)!}(x - \xi)^{m-1} + \frac{f^{(m)}(\xi)}{(m)!}(x - \xi)^m,$$

в правой части которого все слагаемые, кроме последнего, равны нулю. Поэтому

$$\left|\frac{f^{(m)}(\xi)}{(m)!}(x - \xi)^m\right| \leqslant \Delta(f),$$

откуда получим:

$$\varepsilon = \left|\frac{m!}{f^{(m)}(\xi)}\right|^{1/m} \cdot \left(\Delta(f)\right)^{1/m}.$$

Таким образом, для корня кратности m радиус интервала неопределенности пропорционален $\left(\Delta(f)\right)^{1/m}$, т. е. увеличивается, что свидетельствует о плохой обусловленности задачи вычисления кратных корней.

Отметим, что ε не может быть меньше $|\xi| \cdot \varepsilon_м$ — погрешности представления корня ξ в компьютере. На практике оценить величину радиуса неопределенности ε не представляется возможным.

3.3 Метод дихотомии

Считаем, что локализация корней произведена и на интервале $[a, b]$ расположен один корень, который необходимо уточнить с точностью ε.

Итак, имеем $f(a)f(b) < 0$.

. . . . . .

Метод дихотомии *заключается в следующем. Определяем половину отрезка* $c = \frac{1}{2}(a+b)$ *и вычисляем* $f(c)$. *Проверяем следующие условия:*

1. *Если* $|f(c)| < \varepsilon$, *то* c — *корень.*
2. *Если* $f(c)f(a) < 0$, *то корень лежит в интервале* $[a, c]$.
3. *Если условие (2) не выполняется, то корень лежит на отрезке* $[c, b]$.

Продолжая процесс половинного деления в выбранных подынтервалах, можно дойти до сколь угодно малого отрезка, содержащего корень ξ.

. . . . . .

Так как за каждую итерацию интервал, где расположен корень, уменьшается в два раза, то через n итераций интервал будет равен $b_n - a_n = \frac{1}{2^n}(b-a)$, при этом $a_n \leqslant \xi \leqslant b_n$,

$$|\xi - a_n| \leqslant \frac{1}{2^n}(b-a), \quad |\xi - b_n| \leqslant \frac{1}{2^n}(b-a). \tag{3.8}$$

В качестве корня ξ возьмем $\frac{1}{2}(b_n + a_n)$. Тогда погрешность определения корня будет равна $(b_n - a_n)/2$. Если выполняется условие

$$\frac{(b_n - a_n)}{2} < \varepsilon,$$

то процесс поиска заканчивается и $\xi = \frac{1}{2}(b_n + a_n)$.

Справедлива следующая теорема [3].

Теорема 3.2. Итерационный процесс половинного деления сходится к искомому корню ξ с любой наперед заданной точностью ε.

Пусть x_{n-1} и x_n — два последовательных приближения. Тогда из теоремы следует, что $|x_n - x_{n-1}| = \dfrac{1}{2^{n+1}}(b-a)$.

Графически метод дихотомии выглядит следующим образом:

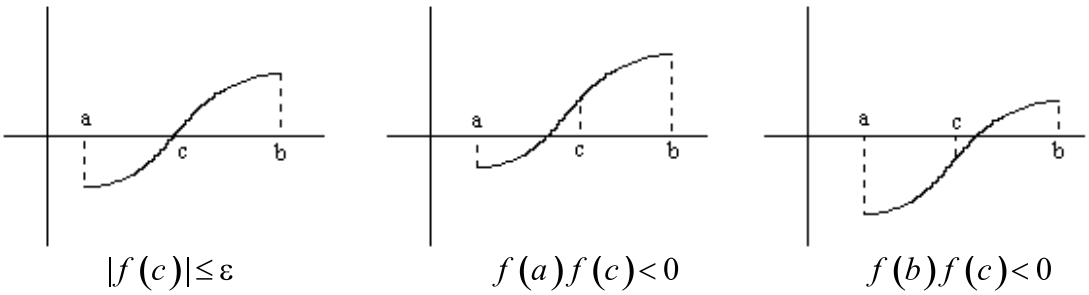


Рис. 3.1

Скорость сходимости метода дихотомии линейная с коэффициентом $\alpha = 0.5$. Покажем это.

Если в качестве x_n брать a_n, то из формулы (3.8) мы можем записать $|\xi - x_n| \leqslant \leqslant \dfrac{1}{2^n}(b-a)$, $|\xi - x_{n-1}| \leqslant \dfrac{1}{2^{n-1}}(b-a)$. Отсюда следует, что $|\xi - x_n| = \dfrac{1}{2}|\xi - x_{n-1}|$.

Погрешность метода оценивается формулой $|\xi - x_n| \leqslant \dfrac{1}{2^n}(b-a)$.

Отметим, что за 10 итераций ($n = 10$) интервал уменьшается в $2^{10} = 1024 \approx 10^3$ раз. За 20 итераций ($n = 20$) уменьшается в $2^{20} \approx 10^6$ раз.

3.4 Метод Ньютона

Пусть корень ξ уравнения $f(x) = 0$ отделен на отрезке $[a, b]$. Предположим, мы нашли $(n-1)$-ое приближение корня x_{n-1}. Тогда n-ое приближение x_n мы можем получить следующим образом. Положим

$$x_n = x_{n-1} + h_{n-1}. \tag{3.9}$$

Раскладывая в ряд $f(x_n)$ в точке x_{n-1} и ограничиваясь линейным приближением, получим

$$f(x_n) = f(x_{n-1} + h_{n-1}) = f(x_{n-1}) + f'(x_{n-1}) \cdot h_{n-1} = 0.$$

Отсюда следует

$$h_{n-1} = -\frac{f(x_{n-1})}{f'_{n-1}}. \tag{3.10}$$

Подставим (3.10) в формулу (3.9), получим

$$x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})};\ n = 1,\ 2,\ \ldots \tag{3.11}$$

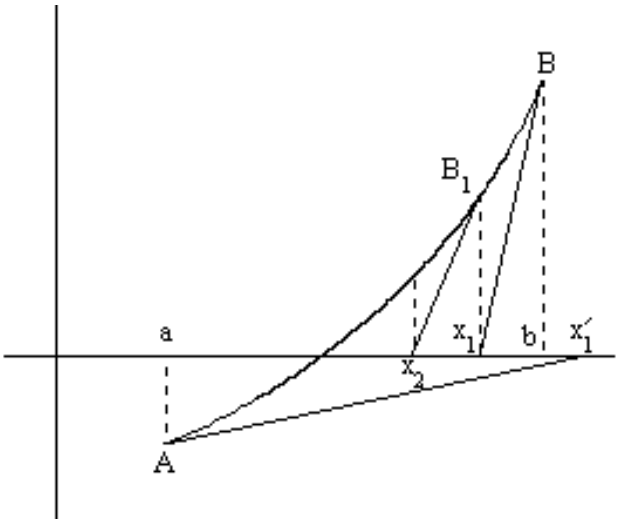


Рис. 3.2 – Геометрическая интерпретация метода Ньютона

Геометрически метод Ньютона эквивалентен замене дуги кривой $y = f(x)$ касательной, проведенной в некоторой точке кривой (см. рис. 3.2).

В точке B имеем $f(x_0)f''(x_0) > 0$. Здесь $x_0 = b$. Проведем касательную в точке B, получим на пересечении касательной с осью OX точку x_1. Далее проводим касательную в точке B_1, получим точку x_2 и т. д.

Если положить $x_0 = a$, то в точке x_0 будем иметь $f(x_0)f''(x_0) < 0$. Тогда касательная в точке A пересекла бы ось OX в точке x_1', лежащей вне отрезка $[a, b]$, то есть при таком выборе начальной точки метод Ньютона оказывается расходящимся.

Теорема 3.3. [3]. Если $f(a)f(b) < 0$, причем $f'(x)$ и $f''(x)$ отличны от нуля и сохраняют определенные знаки при $a \leqslant x \leqslant b$, то исходя из начального приближения $x_0 \in [a, b]$, удовлетворяющего неравенству

$$f(x_0)f''(x_0) > 0, \tag{3.12}$$

можно вычислить методом Ньютона (3.11) единственный корень ξ уравнения $f(x) = 0$ с любой степенью точности.

Оценим скорость сходимости метода Ньютона. Из (3.11) следует

$$\xi - x_n = \xi - x_{n-1} + \frac{f(x_{n-1})}{f'(x_{n-1})}. \tag{3.13}$$

Представим $f(\xi)$ в виде:

$$f(\xi) = f(x_{n-1}) + f'(x_{n-1})(\xi - x_{n-1}) + \frac{1}{2}f''(c_{n-1})(\xi - x_{n-1})^2 = 0,$$

откуда

$$f(x_{n-1}) = -f'(x_{n-1})(\xi - x_{n-1}) - \frac{1}{2}f''(c_{n-1})(\xi - x_{n-1})^2. \qquad (3.14)$$

Подставим (3.14) в (3.13), получим:

$$\xi - x_n = \xi - x_{n-1} - \frac{1}{f'(x_{n-1})}\left[f'_{n-1}(x_{n-1})\cdot(\xi - x_{n-1}) + \frac{1}{2}f''(c_{n-1})\cdot(\xi - x_{n-1})^2\right] =$$

$$= -\frac{1}{2}\cdot\frac{f''(c_{n-1})}{f'_{n-1}(x_{n-1})}(\xi - x_{n-1})^2.$$

Отсюда

$$|\xi - x_n| \leqslant \frac{M_2}{2m_1}(\xi - x_{n-1})^2. \qquad (3.15)$$

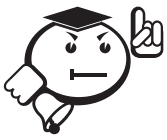
Здесь $M_2 = \max\limits_{x\in[a,b]}|f''(x)|$, $m_1 = \min\limits_{x\in[a,b]}|f'(x)|$.

Таким образом, скорость сходимости метода Ньютона квадратичная с коэффициентом сходимости $\alpha = \dfrac{M_2}{2m_1}$.

Погрешность метода может быть оценена по формуле [7] (см. лемму 3.2):

$$|\xi - x_n| < \left(\frac{M_2}{2m_1}\right)^{2^n-1}|\xi - x_0|^{2^n}.$$

Критерий останова $|x_n - x_{n-1}| < \varepsilon$.



В общем случае совпадение с точностью до ε двух последовательных приближений x_{n-1} и x_n не гарантирует, что с той же точностью совпадет x_n и ξ (см. рис. 3.3). Поэтому целесообразно проверять кроме разности $|x_n - x_{n-1}| < \varepsilon$ также значение функции $f(x_n)$: $|f(x_n)| < \varepsilon_1$.

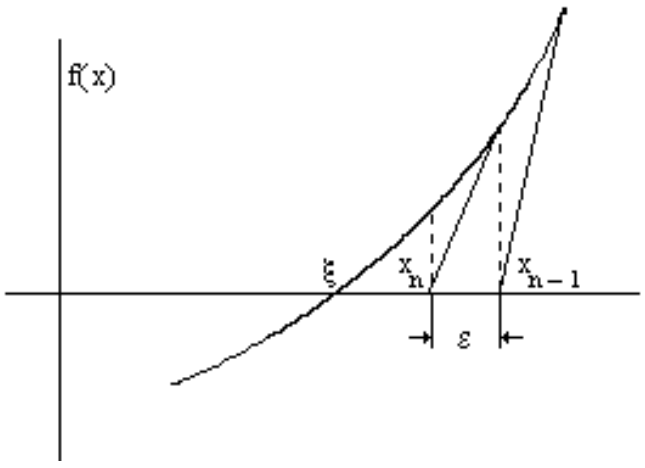


Рис. 3.3

3.4.1 Модификации метода Ньютона

Недостатком метода Ньютона является необходимость вычисления значения производной $f'(x_n)$ на каждой итерации. Рассмотрим некоторые модификации метода Ньютона, свободные от этого недостатка.

Упрощенный метод Ньютона

Модификация метода Ньютона заключается в замене производной $f'(x_{n-1})$ в точке x_{n-1} в формуле:

$$x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})}$$

на производную $f'(x_0)$ в точке x_0, т. е. полагаем $f'(x_{n-1}) \approx f'(x_0)$. В результате получим

$$x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_0)} \quad (n = 0, 1, \ldots). \tag{3.16}$$

Геометрически этот способ означает, что мы заменяем касательные в точках B_n прямыми, параллельными касательной к кривой $y = f(x)$ в точке B_0 (см. рис. 3.4).

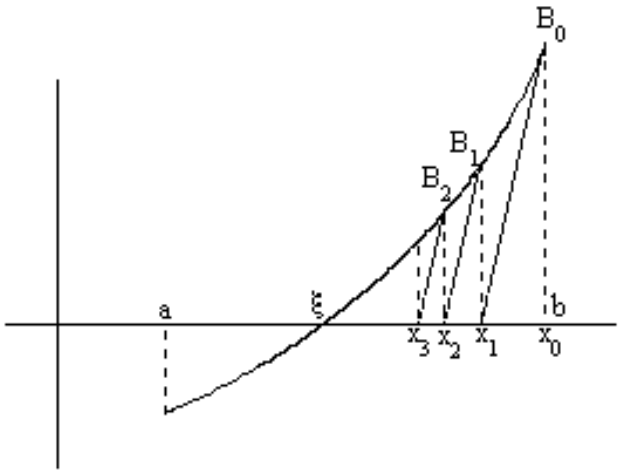


Рис. 3.4 – Модифицированный метод Ньютона

Здесь не надо вычислять каждый раз производную $f'(x_{n-1})$. Сходимость процесса (3.17) обеспечивается следующей теоремой [9].

Теорема 3.4. Пусть на $[a,b]$ задана дважды дифференцируемая функция $f(x)$, причем выполнены следующие условия:

а) $f(a)f(b) < 0$;

б) $f'(x)$ и $f''(x) \neq 0$ и сохраняют знаки на $[a,b]$.

Тогда исходя из начального приближения $x_0 \in [a,b]$, удовлетворяющего неравенству $f(x_0)f''(x_0) > 0$, можно вычислить модифицированным методом Ньютона единственный корень ξ с любой степенью точности.

Сходимость метода. В отличие от метода Ньютона здесь скорость сходимости уже не будет квадратичной. Действительно, из (3.16) имеем

$$\xi - x_n = \xi - x_{n-1} + \frac{f_{n-1}}{f_0'}. \tag{3.17}$$

Подставляя (3.15) в (3.22), получим

$$\begin{aligned} \xi - x_n &= \xi - x_{n-1} - \frac{1}{f_0'}\left[f_{n-1}' \cdot (\xi - x_{n-1}) + \frac{1}{2} \cdot f''(c) \cdot (\xi - x_{n-1})^2\right] = \\ &= (\xi - x_{n-1})\left[1 - \frac{f_{n-1}'}{f_0'}\right] - \frac{1}{2}\frac{f''(c)}{f_0'} \cdot (\xi - x_{n-1})^2. \end{aligned} \tag{3.18}$$

Здесь появился линейный член относительно $(\xi - x_{n-1})$. При $|\xi - x_{n-1}| \ll 1$ вторым слагаемым в правой части (3.18) можно пренебречь, в результате получим:

$$|\xi - x_n| \approx |\xi - x_{n-1}| \cdot \left|1 - \frac{f_{n-1}'}{f_0'}\right| \leqslant |\xi - x_{n-1}| \cdot \left|1 - \frac{m_1}{M_1}\right|,$$

где $M_1 = \max\limits_{x \in [a,b]} |f'(x)|$, $m_1 = \min\limits_{x \in [a,b]} |f'(x)|$.

Таким образом, скорость сходимости модифицированного метода Ньютона будет линейной с параметром сходимости $\alpha = 1 - \dfrac{m_1}{M_1}$.

3.4.2 Уточнение метода Ньютона для случая кратного корня

В принципе для вычисления корня уравнения $f(x) = 0$ кратности $m > 1$ можно использовать и стандартный метод Ньютона. Однако в этом случае скорость его сходимости является только линейной с коэффициентом сходимости $\alpha = 1 - 1/m$ [1].

Для того чтобы сохранить квадратичную сходимость, необходимо метод Ньютона модифицировать следующим образом [1]:

$$x_n = x_{n-1} - m\frac{f(x_{n-1})}{f'(x_{n-1})},\ n = 1, 2, \ldots$$

Можно показать [1], (это достигается раскрытием неопределенностей с помощью правила Лопиталя), что при таком выборе итерационной функции $\varphi(x) = x - m\dfrac{f(x)}{f'(x)}$ получим $\varphi'(\xi) = 0$, и сходимость метода снова окажется квадратичной.

3.5 Метод хорд

Этот метод можно рассматривать как одну из модификаций метода Ньютона.

Пусть т. c — фиксированная точка, расположенная в окрестности простого корня ξ. Заменим в расчетной формуле Ньютона (3.11) производную $f'(x_{n-1})$ приближенным равенством $f'(x_{n-1}) \approx \dfrac{f(c) - f(x_{n-1})}{c - x_{n-1}}$. В результате придем к новой расчетной формуле — формуле метода ложного положения (метод хорд)

$$x_n = x_{n-1} - \frac{c - x_{n-1}}{f(c) - f(x_{n-1})} f(x_{n-1}),\ n = 1, 2, \ldots \tag{3.19}$$

Геометрическая иллюстрация метода приведена на рис. 3.5.

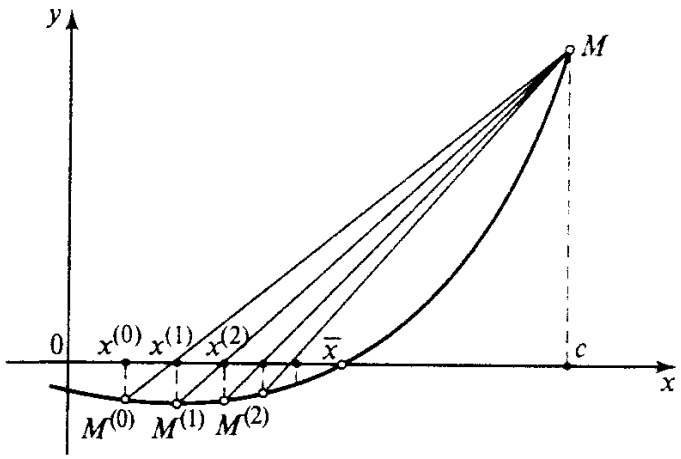
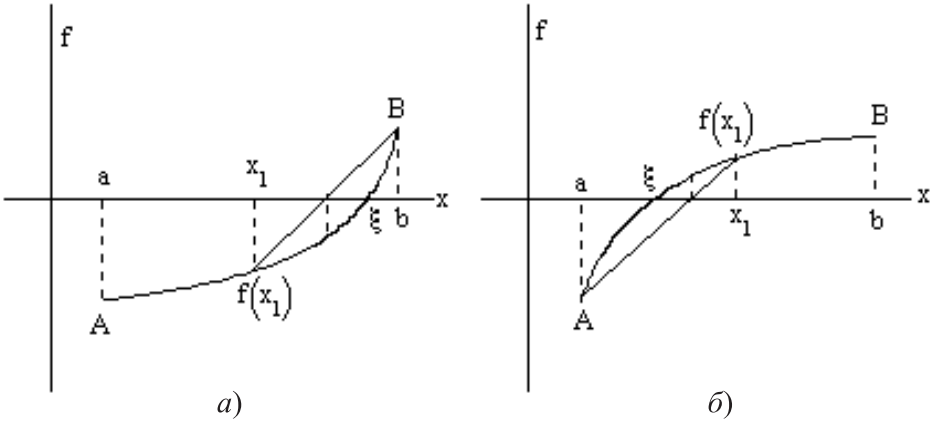


Рис. 3.5 – Метод ложного положения (общий случай)

Рассмотрим более детально метод хорд.

Пусть на интервале $[a,b]$ существует корень, т. е. выполняется $f(a)f(b) < 0$. На рис. 3.6, *а*, 3.6, *б* представлены графики выпуклой (рис. 3.6, *а*, $f''(x) > 0$) и вогнутой (рис. 3.6, *б*, $f''(x) < 0$) функций. На рис. 3.8, *а* подвижен конец *a*, поэтому для всех приближений корня x_i, $i = 0, 1, 2, \ldots n$ выполняются условия $f(x_i)f(b) < 0$.

На рис. 3.6, *б* подвижен конец *b*, поэтому для всех приближений корня x_i, $i = 0, 1, 2, \ldots, n$ выполняются условия $f(x_i)f(a) < 0$.



а) *б*)

Рис. 3.6

На рис. 3.7, *а* и рис. 3.7, *б* приведены вогнутая (рис. 3.7, *а*, $f''(x) < 0$) и выпуклая (рис. 3.7, *б*, $f''(x) > 0$) функции, для которых подвижен конец *a* и *b* соответственно.

Рассмотрим рис. 3.6, *а*. Проведем через точки *A* и *B* хорду. Уравнение хорды имеет вид:

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

В точке $x = x_1$ имеем $y = 0$. В результате получим первое приближение корня:

$$x_1 = a - \frac{f(a)}{f(b) - f(a)}(b - a). \tag{3.20}$$

Проверяем условия:

(а) $f(x_1)f(b) < 0$, (б) $f(x_1)f(a) < 0$.

Если выполняется условие (а), то в формуле (3.20) точку a заменяем на x_1, получим:

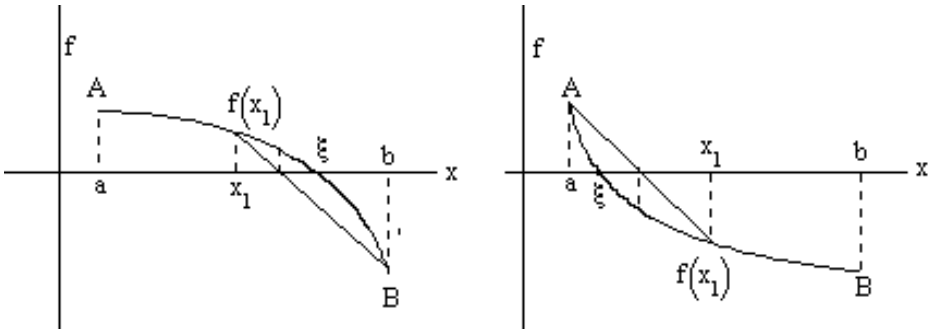
$$x_2 = x_1 - \frac{f(x_1)}{f(b) - f(x_1)}(b - x_1).$$

Продолжая этот процесс, получим для n-го приближения:

$$x_n = x_{n-1} - \frac{f(x_{n-1})}{f(b) - f(x_{n-1})}(b - x_{n-1}). \tag{3.21}$$

Здесь подвижен конец a, то есть $f(x_i)f(b) < 0$. Аналогичная ситуация на рис. 3.7, *а*.

Рассмотрим случай, когда неподвижен конец a (выполняется условие (б)).



а) *б*)

Рис. 3.7

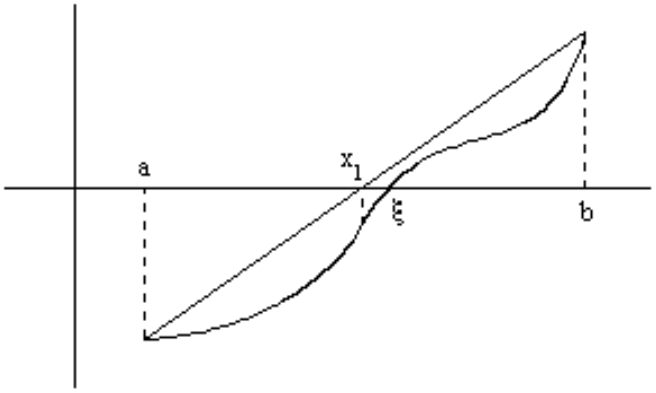


Рис. 3.8

На рис 3.6, *б*, 3.7, *б* выполняется $f(x_i)f(a) < 0$. Записав уравнение хорды, мы на первом шаге итерационного процесса получим x_1 (см (3.20)). Здесь выполняется $f(x_1)f(a) < 0$. Затем в формуле (3.20) точку b заменяем на x_1, получим 2-ое приближение корня:

$$x_2 = a - \frac{f(a)}{f(x_1) - f(a)}(x_1 - a).$$

Продолжая процесс, придем к формуле:

$$x_n = a - \frac{f(a)}{f(x_{n-1}) - f(a)}(x_{n-1} - a). \tag{3.22}$$

Завершение итерационного процесса происходит по условию $|x_n - x_{n-1}| < \varepsilon$. В качестве корня берется n-ое приближение $\xi \approx x_n$.

На рис. 3.8 $f''(x)$ меняет знак, поэтому подвижными будут оба конца.

Для выпуклой (вогнутой) функции имеет место следующая теорема о сходимости итерационного процесса [9].

Теорема 3.5. Пусть задана непрерывная дважды дифференцируемая функция $f(x)$ на $[a,b]$ и пусть $f(a)f(b) < 0$, а $f'(x)$ и $f''(x)$ сохраняют свои знаки на $[a,b]$ (см рис 3.6, *а*, 3.6, *б* и рис 3.7, *а*, 3.7, *б*). Тогда итерационный процесс метода хорд сходится к корню ξ с любой наперед заданной точностью ε.

Сходимость метода хорд линейная с коэффициентом $\alpha = \dfrac{M_1 - m_1}{M_1}$.

$$\frac{|\xi - x_n|}{|\xi - x_{n-1}|} \leqslant 1 - \frac{m_1}{M_1} = \frac{M_1 - m_1}{M_1}, \tag{3.23}$$

где $m_1 = \min|f'(x)|$, $M_1 = \max|f'(x)|$.

Это вытекает из следующих формул. Рассмотрим случай неподвижного конца b и $f(b) > 0$.

Имеем из (3.21) $x_n = x_{n-1} - \dfrac{f_{n-1}}{f_b - f_{n-1}}(b - x_{n-1})$. Отсюда

$$\xi - x_n = \xi - x_{n-1} + \frac{f_{n-1}}{f_b - f_{n-1}}(b - x_{n-1}).$$

Поделим обе части на $(\xi - x_{n-1})$, получим:

$$\frac{\xi - x_n}{\xi - x_{n-1}} = 1 + \frac{f_{n-1}}{f_b - f_{n-1}}\frac{(b - x_{n-1})}{(\xi - x_{n-1})} = 1 - \frac{f_\xi - f_{n-1}}{f_b - f_{n-1}} \cdot \frac{(b - x_{n-1})}{(\xi - x_{n-1})}.$$

В последнем выражении мы можем приближенно положить

$$\frac{f_\xi - f_{n-1}}{(\xi - x_{n-1})} \approx f'_\xi; \quad \frac{(b - x_{n-1})}{f_b - f_{n-1}} \approx \frac{1}{f'_b}.$$

В результате получим:

$$\frac{\xi - x_n}{\xi - x_{n-1}} \approx 1 - \frac{f'_\xi}{f'_b} \leqslant 1 - \frac{m_1}{M_1},$$

ч. т. д. (см. неравенство (3.23)).

3.6 Метод итераций

Одним из наиболее эффективных способов численного решения уравнений является метод итерации. Сущность этого метода заключается в следующем. Пусть дано уравнение $f(x) = 0$. Заменим его равносильным уравнением

$$x = \varphi(x). \tag{3.24}$$

Выберем начальное приближение корня x_0 и подставим его в правую часть уравнения (3.24). Тогда получим некоторое число

$$x_1 = \varphi(x_0). \tag{3.25}$$

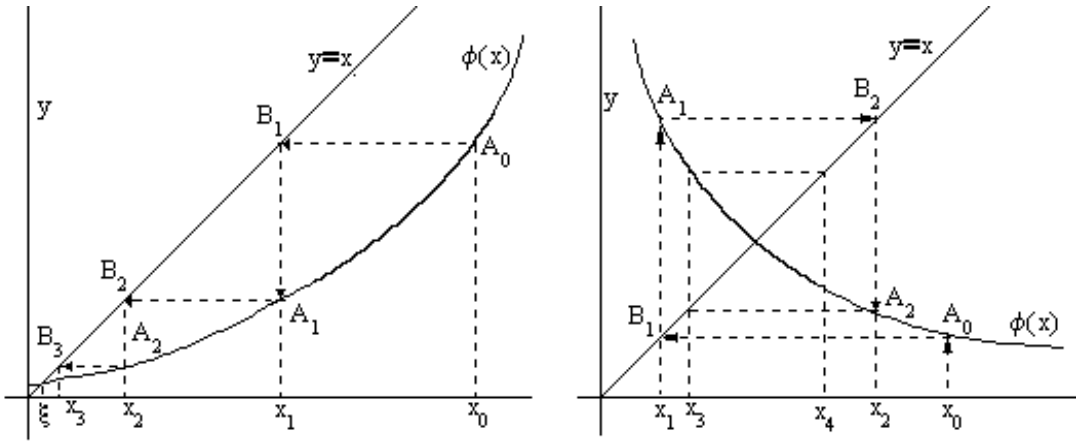
Подставляя теперь в правую часть (3.25) вместо x_0 число x_1, получим число $x_2 = \varphi(x_1)$. Повторяя этот процесс, будем иметь последовательность чисел

$$x_n = \varphi(x_{n-1}) \quad (n = 1, 2, \ldots). \tag{3.26}$$

Если эта последовательность сходящаяся, то есть существует предел $\xi = \lim\limits_{k\to\infty} x_n$, то, переходя к пределу в равенстве (3.26) и предполагая функцию $\varphi(x)$ непрерывной, найдем:

$$\lim_{n\to\infty} x_n = \varphi\left(\lim_{n\to\infty} x_{n-1}\right) \text{ или } \xi = \varphi(\xi).$$

Таким образом, предел ξ является корнем уравнения (3.24) и может быть вычислен по формуле (3.26) с любой степенью точности.



а) *б*)

Рис. 3.9

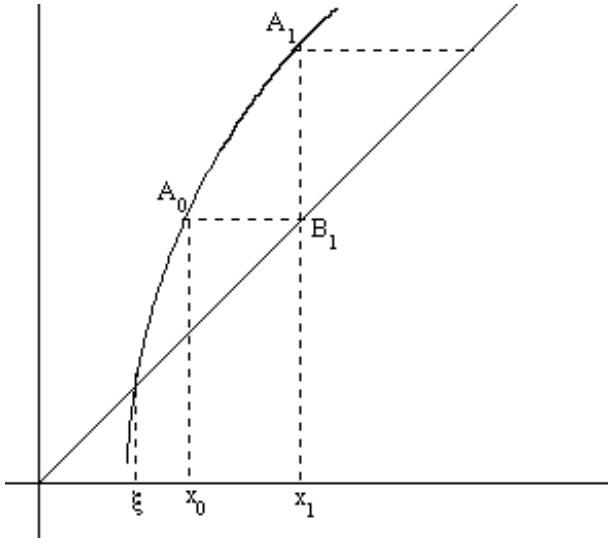


Рис. 3.10

На рис. 3.9, *а*, 3.9, *б* в окрестности корня $|\varphi'(x)| < 1$ и процесс итерации сходится. Однако если рассмотреть случай $|\varphi'(x)| > 1$, то процесс итерации может быть расходящимся (см. рис. 3.10).

Сформулируем достаточные условия сходимости метода итерации [3, 7].

Теорема 3.6. Пусть функция $\varphi(x)$ определена и дифференцируема на отрезке $[a,b]$, причем все ее значения $\varphi(x) \in [a,b]$ и пусть $|\varphi'(x)| \leqslant q < 1$ при $x \in [a,b]$. Тогда процесс итерации $x_n = \varphi(x_{n-1})$ сходится независимо от начального значения $x_0 \in [a,b]$ и предельное значение $\xi = \lim\limits_{k\to\infty} x_n$ является единственным корнем уравнения $x = \varphi(x)$ на отрезке $[a,b]$.

Доказательство. Рассмотрим два последовательных приближения $x_n = \varphi(x_{n-1})$ и $x_{n+1} = \varphi(x_n)$ и возьмем их разность $x_{n+1} - x_n = \varphi(x_n) - \varphi(x_{n-1})$. По теореме Лагранжа правая часть может быть представлена как

$$\varphi'(c)(x_n - x_{n-1}), \text{ где } c \in (a,b).$$

Тогда получим:

$$|x_{n+1} - x_n| \leqslant |\varphi'(c)|\,|x_n - x_{n-1}| \leqslant q\,|x_n - x_{n-1}|.$$

Полагая $n = 1, 2, \ldots$, будем иметь:

$$|x_2 - x_1| \leqslant q\,|x_1 - x_0|, \quad |x_3 - x_2| \leqslant q\,|x_2 - x_1| \leqslant q^2\,|x_1 - x_0|, \quad |x_{n+1} - x_n| \leqslant q^n\,|x_1 - x_0|. \quad (3.27)$$

Из (3.27), в силу условия $q < 1$, видно, что последовательность $\{x_n\}$ сходится к некоторому числу ξ, то есть $\lim\limits_{n\to\infty} x_n = \xi \in [a,b]$, и следовательно, $\lim\limits_{n\to\infty} x_n = \varphi\left(\lim\limits_{n\to\infty} x_{n-1}\right)$ (в силу непрерывности функции $\varphi(x)$) или $\xi = \varphi(\xi)$, ч. т. д.

Для погрешности корня ξ можно получить следующую формулу.

Имеем $x_n = \varphi(x_{n-1})$. Далее $\xi - x_n = \xi - \varphi(x_{n-1}) = \varphi(\xi) - \varphi(x_{n-1})$. Используя формулу $\varphi(x_{n-1}) = \varphi(x_n) - \varphi'(c)(x_n - x_{n-1})$, правую часть равенства представим в виде $\varphi(\xi) - \varphi(x_n) + \varphi'(c)(x_n - x_{n-1})$.

В результате получим:

$$\xi - x_n = \varphi'(c_1)(\xi - x_n) + \varphi'(c)(x_n - x_{n-1})$$

или

$$|\xi - x_n| \leqslant q|\xi - x_n| + q|x_n - x_{n-1}|.$$

Отсюда получим:

$$|\xi - x_n| \leqslant \frac{q}{1-q}|x_n - x_{n-1}|. \tag{3.28}$$

Из (3.28) видно, что при q, близком к 1, разность $|\xi - x_n|$ может быть очень большой, несмотря на то, что $|x_n - x_{n-1}| < \varepsilon$, где ε — заданная величина.

Для того чтобы вычислить ξ с точностью ε, необходимо обеспечить

$$|x_n - x_{n-1}| \leqslant \frac{1-q}{q}\varepsilon. \tag{3.29}$$

Тогда, подставляя (3.29) в (3.28), получим $|\xi - x_n| < \varepsilon$.

Если q очень мало, то вместо (3.29) можно использовать формулу:

$$|x_n - x_{n-1}| < \varepsilon.$$

Сходимость метода итерации линейная с коэффициентом сходимости $\alpha = q$. Действительно, имеем:

$$\xi - x_n = \varphi(\xi) - \varphi_{n-1} = \varphi'(c) \cdot (\xi - x_{n-1}),$$

отсюда

$$|\xi - x_n| \leqslant q \cdot |\xi - x_{n-1}|.$$

Рассмотрим способ представления уравнения $f(x) = 0$ в форме $x = \varphi(x)$.

Функцию $\varphi(x)$ необходимо задать такую, чтобы $|\varphi'(x)|$ была малой величиной в окрестности корня.

Пусть известно m_1 и M_1 — наименьшее и наибольшее по модулю значения производной $f'(x)$:

$$0 < m_1 \leqslant |f'(x)| \leqslant M_1, \tag{3.30}$$

где $M_1 = \max|f'(x)|$, $m_1 = \min|f'(x)|$.

Заменим уравнение $f(x) = 0$ эквивалентным ему уравнением $x = x - \lambda \cdot f(x)$.

Положим $\varphi(x) = x - \lambda \cdot f(x)$. Подберем параметр λ таким образом, чтобы в окрестности корня ξ выполнялось неравенство

$$0 \leqslant |\varphi'(x)| = |1 - \lambda \cdot f'(x)| \leqslant q < 1.$$

Отсюда на основании (3.30) получаем:

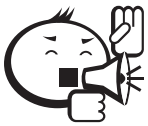
$$0 \leqslant |1 - \lambda M_1| \leqslant |1 - \lambda m_1| \leqslant q.$$

Тогда, выбирая $\lambda = 1/M_1$, получим:

$$q = 1 - \frac{m_1}{M_1} < 1.$$

Если $\lambda = 1/f'(x)$, то итерационная формула $x_n = \varphi(x_{n-1})$ переходит в формулу Ньютона:

$$x_n = x_{n-1} - \frac{f(x_n)}{f'(x_n)}.$$



Если $f'(x) < 0$, то в качестве λ следует взять $\lambda = -1/M_1$.

3.7 Обусловленность методов вычисления корня

Метод простой итерации

Выше мы рассматривали метод итерации при идеальном предположении о возможности точного вычисления значений функции $\varphi(x)$. В действительности же вычисления на компьютере дают приближенные значения $\varphi^*(x)$. Поэтому вместо последовательности x_n, удовлетворяющей равенству $x_n = \varphi(x_{n-1})$, получается последовательность $\tilde{x}_n$, для которой

$$\tilde{x}_n = \varphi^*(\tilde{x}_{n-1}). \tag{3.31}$$

Известно, что метод простой итерации и многие другие итерационные методы устойчивы к ошибке, допущенной на одной из итерации. Такая ошибка эквивалентна некоторому ухудшению очередного приближения; если она не вывела приближение за пределы области сходимости, то итерационная последовательность по-прежнему будет сходиться к решению ξ, а внесенная погрешность — затухать. Поэтому о таких итерационных методах говорят, что они обладают *свойством самоисправляемости* [1].

Однако погрешности допускаются не на одной, а на всех итерациях и совокупное их влияние будет иным.

Обусловленность задачи. Прежде чем сформулировать результат о поведении метода простой итерации при наличии погрешности в вычислении функции φ, отметим, что преобразование уравнения $f(x) = 0$ к виду $x = \varphi(x)$ изменяет обусловленность задачи. Покажем это [1].

Разложим функцию $f(x)$ в ряд в окрестности корня ξ и ограничимся линейным приближением

$$f(x) = f(\xi) + f'(\xi)(x - \xi) = f'(\xi)(x - \xi). \tag{3.32}$$

Так как значения функции $f(x)$ вычисляются с погрешностью, то в окрестности корня мы имеем

$$|f(x)| \leqslant \Delta(f), \tag{3.33}$$

где $\Delta(f)$ — погрешность вычисления функции. Тогда с учетом (3.32) и (3.33) получим неравенство $\left|f'(\xi)(x-\xi)\right| \leqslant \Delta(f)$. Отсюда

$$\Delta(\xi) \leqslant v \cdot \Delta(f), \tag{3.34}$$

где

$$v = \frac{1}{\left|f'(\xi)\right|} \tag{3.35}$$

— абсолютное число обусловленности корня. Чем меньше $f'(\xi)$ в окрестности корня, тем больше число обусловленности.

При переходе к уравнению $x = \varphi(x)$ вместо $f(x) = 0$ мы можем записать $\tilde{f}(x) = 0$, где $\tilde{f}(x) = x - \varphi(x)$. При этом $\Delta(\tilde{f}) = \Delta(\varphi)$, так как в действительности приближенно вычисляется только функция φ.

Тогда для числа обусловленности (3.35) в окрестности корня получим:

$$v = \frac{1}{\left|1 - \varphi'(\xi)\right|}. \tag{3.36}$$

При выполнении условия $\left|\varphi'(x)\right| \leqslant q < 1$ мы можем заменить величину $1/\left|1 - \varphi'(\xi)\right|$ числом $v = 1/(1-q)$. В этом случае (3.34) примет вид:

$$\Delta(\xi) \leqslant v \cdot \Delta(\varphi). \tag{3.37}$$

Если $\varphi'(x) \leqslant 0$, то $v \leqslant 1$. Если $\varphi'(x) > 0$, то $v = \dfrac{1}{1-q}$ и, следовательно, $v > 1$. Более того, если $\varphi'(\xi) \approx 1$, то задача поиска корня методом итерации становится плохо обусловленной. В этом случае следует ожидать потерю верных цифр $N = \lg(v)$ последовательности x_n в окрестности корня ξ.

Поделим обе части (3.37) на x, получим:

$$\delta(\xi) \leqslant v \cdot \delta(\varphi), \tag{3.38}$$

и, следовательно, абсолютное и относительное числа обусловленности здесь совпадают.

Пример 3.2

Для уравнения $x = \varphi(x)$ при $\varphi(x) = 0.9999x + 10^{-4}\sqrt{2}$ имеем $\varphi'(x) = 0.9999$ и, следовательно, $v = 10^4$. Поэтому при решении этого уравнения методом простой итерации будет потеряно примерно 4 значащих цифры.

Метод Ньютона

Метод Ньютона можно рассматривать как вариант метода итераций, связанный с преобразованием уравнения $f(x) = 0$ к виду $x = \varphi(x)$, где $\varphi(x) = x - f(x)/f'(x)$. Поэтому, вышеприведенные результаты остаются в силе и для метода Ньютона. В этом случае мы получим для числа обусловленности выражение:

$$\nu = \max_{x\in[a,b]} \frac{1}{\left|1 - \varphi'\right|} = \max_{x\in[a,b]} \frac{1}{\left|[f(x)/f'(x)]'\right|}. \tag{3.39}$$

В окрестности корня число обусловленности равно

$$\nu = \frac{1}{\left|1 - \varphi'(\xi)\right|} = \frac{1}{\left|[f(\xi)/f'(\xi)]'\right|}.$$

Метод хорд

Для метода хорд мы имеем две ситуации:

1) левый конец неподвижен. В этом случае $\varphi(x) = a - \dfrac{f(a)(x-a)}{f(x)-f(a)}$, $\varphi'(x) = f(a)\dfrac{f'(x)(x-a) - f(x) + f(a)}{\left[f(x)-f(a)\right]^2}$. Подставим $\varphi'(\xi)$ в (3.36), получим значение числа обусловленности в окрестности корня ξ;

2) правый конец неподвижен. В этом случае $\varphi(x) = x - \dfrac{f(x)(b-x)}{f(b)-f(x)}$, $\varphi'(x) = 1 - \dfrac{f'(x)(b-x)\left[1+f(x)\right] - f(x)}{\left[f(x)-f(a)\right]^2}$.

Чувствительность к погрешностям [1]. Рассмотренные одношаговые методы (упрощенный метод Ньютона, метод ложного положения) можно интерпретировать как различные варианты метода простой итерации. Поэтому исследование их чувствительности к погрешностям сводится (аналогично тому, как это было сделано ранее для метода Ньютона) к использованию соответствующих результатов. Например, нетрудно убедиться в хорошей обусловленности модифицированного метода Ньютона и метода ложного положения.

Метод итерации позволяет получить приближенное значение корня ξ с точностью, примерно совпадающей с радиусом ε интервала неопределенности корня.

Контрольные вопросы по главе 3

1. Что такое локализация корней и для чего необходимо выполнять этот этап поиска корней?
2. Что называют корнем k-ой кратности?
3. Запишите условие существования корня на заданном отрезке $[a, b]$.
4. Вычислите количество итераций (шагов) N поиска корня с заданной точностью ε на отрезке $[a, b]$ в методе перебора.
5. Вычислите количество итераций (шагов) N поиска корня с заданной точностью ε на отрезке $[a, b]$ в методе деления отрезка пополам.
6. Запишите число обусловленности задачи вычисления корня.
7. Сформулируйте достаточное условие сходимости итерационного метода.
8. Сформулируйте достаточное условие сходимости метода Ньютона.
9. Как привести исходное уравнение к виду, необходимому для метода итераций?
10. Запишите число обусловленности для метода итераций.
11. Запишите число обусловленности для метода Ньютона.
12. Запишите число обусловленности для метода хорд.

Глава 4

ЧИСЛЕННЫЕ МЕТОДЫ РЕШЕНИЯ СИСТЕМ ЛИНЕЙНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ

4.1 Постановка задачи

Выделяют четыре основных раздела линейной алгебры:

1. Решение систем линейных алгебраических уравнений (СЛАУ).
2. Обращение матриц.
3. Вычисление определителей.
4. Вычисление собственных значений и собственных векторов.

При формальном подходе решение этих задач не встречает затруднений: решение системы можно найти, раскрыв определители в формуле Крамера; для нахождения собственных значений матрицы достаточно выписать характеристическое уравнение и найти его корни. Однако в численной реализации все гораздо сложнее.

Так, при непосредственном раскрытии определителя решение системы с m неизвестными требует порядка $m!m$ арифметических операций; уже при $m = 30$ такое число операций недоступно для современных ЭВМ.

Другой причиной, по которой классические способы неприменимы, является сильное влияние на окончательный результат ошибок округлений при вычислении.

Методы решения алгебраических задач разделяются на точные, итерационные и вероятностные. Прямые (точные) методы характеризуются тем, что дают решение системы за конечное число арифметических операций. Если все операции выполняются точно, то решение задачи получается точным. К прямым методам решения СЛАУ относятся (см. например [2–4]):

1. Метод Крамера.
2. Методы последовательного исключения неизвестных (метод Гаусса и его модификации).

3. Метод ортогонализации.
4. Метод декомпозиции.

Итерационные методы являются приближенными. Они дают решение СЛАУ как предел последовательных приближений, выполненных по единообразной схеме. К итерационным методам решения СЛАУ относятся [2–4]:

1. Метод простой итерации.
2. Метод Зейделя.
3. Метод релаксаций.
4. Градиентные методы и их модификации.

Эти методы применяют на практике для систем порядка 10^4–10^7.

При изучении итерационных процессов нам понадобятся понятия норм вектора и матрицы.

В дальнейшем мы будем рассматривать методы решения систем линейных алгебраических уравнений с вещественными коэффициентами:

$$\left.\begin{array}{l} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1m}x_m = b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2m}x_m = b_2 \\ a_{m1}x_1 + x_{m2}x_2 + \ldots + a_{mm}x_m = b_m \end{array}\right\} \quad \Delta \neq 0. \tag{4.1}$$

Здесь Δ — определитель матрицы.

В матричной форме эта система принимает вид:

$$Ax = b, \tag{4.2}$$

где A — матрица системы размерности $m \times m$; x — вектор-столбец решения размерности m, b — вектор-столбец правой части размерности m.

В качестве погрешности решения может использоваться величина $e = x - \tilde{x}$ или $r = b - A\tilde{x}$, где $\tilde{x}$ — приближенное решение. Вектор r называют невязкой решения. Он связан с погрешностью решения соотношением $r = A(x - \tilde{x}) = Ae$, отсюда

$$e = A^{-1}r. \tag{4.3}$$

4.2 Нормы векторов и матриц

*Под **нормой** матрицы A понимается действительное число $\|A\|$, вычисляемое с помощью элементов матрицы и обладающее следующими свойствами* [3, 7]:

1) $\|A\| \geqslant 0$, *причем* $\|A\| = 0$ *только тогда, когда* $A = 0$;
2) $\|cA\| = |c|\|A\|$ *(c — скалярная величина), в частности, имеет место равенство* $\|-A\| = \|A\|$;
3) $\|A + B\| \leqslant \|A\| + \|B\|$;
4) $\|A - B\| \geqslant \|A\| - \|B\|$;
5) $\|AB\| \leqslant \|A\|\|B\|$ *(B — матрица соответствующей размерности, для которой операции сложения, вычитания и умножения с матрицей A имеют смысл).*

*Под **нормой** вектора x понимается действительное число $\|x\|$, вычисляемое с помощью элементов вектора и обладающее следующими свойствами* [3, 7]:

1) $\|x + y\| \leqslant \|x\| + \|y\|$;
2) $\|x - y\| \geqslant \|x\| - \|y\|$; *(вектор y имеет соответствующую размерность, для которой операции сложения и вычитания с вектором x имеют смысл);*
3) $\|Ax\| \leqslant \|A\|\|x\|$;
4) $\|cx\| \leqslant |c|\|x\|$.

Если в пространстве векторов $x = (x_1, \ldots, x_m)^T$ введена норма $\|x\|$, то согласованной с ней нормой в пространстве матриц A называют норму

$$\|A\| = \sup_{x \neq 0} \frac{\|Ax\|}{\|x\|}. \tag{4.4}$$

Наиболее употребительны в пространстве векторов следующие нормы:

$$\text{а) } \|x\|_1 = \max_{1 \leqslant i \leqslant m} |x_i|; \quad \text{б) } \|x\|_2 = \sum_{j=1}^{m} |x_j|; \quad \text{в) } \|x\|_3 = \sqrt{\sum_{j=1}^{m} x_j^2} = \sqrt{(x,x)}. \tag{4.5}$$

Согласованные с (4.5, *а*) – (4.5, *в*) нормы матрицы A (размера $m \times m$) равны

$$\text{а) } \|A\|_1 = \max_{1 \leqslant i \leqslant m} \left(\sum_{j=1}^{m} |a_{ij}| \right); \quad \text{б) } \|A\|_2 = \max_{1 \leqslant j \leqslant m} \left(\sum_{i=1}^{m} |a_{ij}| \right); \quad \text{в) } \|A\|_3 = \sqrt{\sum_i \sum_j a_{ij}^2}. \tag{4.6}$$

Покажем вывод (4.6, *а*) — (4.6, *в*). Вывод формулы (4.6, *а*):

$$\|Ax\|_1 = \max_i \left| \sum_{j=1}^{m} a_{ij} x_j \right| \leqslant \max_i \left(\sum_j |a_{ij}| \cdot |x_j| \right) \leqslant \max_j |x_j| \max_i \sum_{j=1}^{m} |a_{ij}| = \|x\|_1 \max_i \sum_{j=1}^{m} |a_{ij}|.$$

Отсюда, с учетом (4.4) получим (4.6, *а*).

Вывод формулы (4.6, *б*):

$$\|Ax\|_2 = \sum_i \left| \sum_j a_{ij} x_j \right| \leqslant \sum_i \sum_j |a_{ij}|\,|x_j| \leqslant \max_j \sum_{i=1}^{m} |a_{ij}| \sum_j |x_j| = \|x\|_2 \max_j \sum_i |a_{ij}|.$$

С учетом (4.4) получим (4.6, *б*).

Вывод формулы (4.6, *в*):

$$\|Ax\|_3 = \sqrt{\sum_i \left(\sum_j a_{ij} x_j \right)^2} \leqslant \sqrt{\sum_i \left(\sum_{j=1}^{m} a_{ij}^2 \right) \left(\sum_{k=1}^{m} x_k^2 \right)} \leqslant$$

$$\leqslant \sqrt{\sum_{k=1}^{m} x_k^2} \sqrt{\sum_i \sum_j a_{ij}^2} = \|x\|_3 \sqrt{\sum_i \sum_j a_{ij}^2}.$$

С учетом (4.4) получим $\|A\|_3 = \sqrt{\sum_i \sum_j a_{ij}^2}$. Здесь мы использовали неравенство Коши—Буняковского

$$\left(\sum_{k=1}^{n} g_k d_k \right)^2 \leqslant \sum_{k=1}^{n} g_k^2 \cdot \sum_{k=1}^{n} d_k^2.$$

Покажем, что

$$\|A\|_3 = \sqrt{\max_i \lambda^i_{A^T A}}, \tag{4.7}$$

где $\lambda^i_{A^T A}$ — собственное число матрицы $A^T A$.

Имеем по определению:

$$\|A\|_3 = \sup_x \frac{\|Ax\|_3}{\|x\|_3} = \sqrt{\sup_x \frac{(Ax, Ax)}{(x, x)}} = \sqrt{\sup_x \frac{(A^T Ax, x)}{(x, x)}},$$

где $A^T A$ — симметричная матрица.

Пусть B — симметричная матрица, $e_1, e_2, \ldots, e_m$ — ортонормированная система ее собственных векторов, $\lambda_1, \lambda_2, \ldots, \lambda_m$ — соответствующие собственные значения. Всякий вектор x представим в виде $\sum_i^m c_i e_i$.

Имеем

$$(Bx, x) = \left(B \sum c_i e_i, \sum c_j e_j \right) = \left(\sum \lambda_i c_i e_i, \sum c_j e_j \right) = \sum \lambda_i c_i^2,$$

поэтому

$$(Bx, x) \leqslant \max_i \lambda_i \sum_i c_i^2 = \max_i \lambda_i \,(x, x)\,.$$

Из этих соотношений следует, что

$$\sup_x \frac{|(Bx,x)|}{(x,x)} = \max_i |\lambda_i|. \qquad (4.8)$$

Так как A^TA — положительно определенная матрица, то все $\lambda^i_{A^TA} \geqslant 0$.
Полагая в (4.8) $B = A^TA$, получаем:

$$\sup_x \frac{|(A^TAx,x)|}{(x,x)} = \max_i |\lambda^i_{A^TA}| = \max_i \lambda^i_{A^TA}.$$

Из полученных соотношений следует (4.7). Отметим важный частный случай. Если A — симметричная матрица, то $\lambda^i_{A^TA} = (\lambda^i_{A^T})^2 = (\lambda^i_A)^2$, поэтому для нее $\|A\|_3 = \max_i |\lambda^i_A|$.

Норму $\|x\|_3$ называют евклидовой нормой. Для норм $\|x\|_1$, $\|x\|_2$, $\|x\|_3$ справедливо соотношение $\|x\|_1 \leqslant \|x\|_3 \leqslant \|x\|_2$ [1].

4.3 Абсолютная и относительная погрешности векторов

Будем считать, что в пространстве m-мерных векторов R^m введена некоторая норма $\|x\|$. Тогда в качестве меры близости векторов x и y естественно использовать величину $\|x - y\|$, которая является аналогом расстояния между точками x и y.

*Введем **абсолютную** и **относительную погрешности вектора** x^* с помощью соотношений*

$$\Delta(x) = \|x - x^*\|, \quad \delta(x) = \frac{\|x - x^*\|}{\|x^*\|}. \qquad (4.9)$$

4.4 Обусловленность задачи решения систем линейных алгебраических уравнений

Решения различных систем линейных алгебраических уравнений обладают разной чувствительностью к погрешностям входных данных. Задача вычисления решения x системы уравнений $Ax = b$ может быть как хорошо, так и плохо обусловленной.

Предложение. Для погрешности приближенного решения x^* системы (4.2) справедлива оценка

$$\Delta(x) \leqslant \left\|A^{-1}\right\| \cdot \|r\|, \tag{4.10}$$

где $r = b - Ax^*$ — невязка решения.

Это следует из (4.3), (4.9) и определения нормы векторов и матриц.

Пусть элементы матрицы A точно известны, а элементы вектора правой части — приближенно. Справедлива следующая теорема [1].

Теорема 4.1. Пусть x^* — решение системы (4.2) с приближенно заданной правой частью b^*. Тогда для абсолютной и относительной погрешностей решения справедливы оценки:

$$\Delta(x) \leqslant v_\Delta \Delta(b), \tag{4.11}$$

$$\delta(x) \leqslant v_\delta \delta(b), \tag{4.12}$$

где $v_\Delta = \left\|A^{-1}\right\|$, $v_\delta = \left\|A^{-1}\right\| \cdot \|b^*\| / \|x^*\| = \left\|A^{-1}\right\| \cdot \|Ax^*\| / \|x^*\|$, $\delta(b) = \Delta(b)/\|b^*\|$ — относительная погрешность вектора правой части b.

Доказательство теоремы следует из определения абсолютной и относительной погрешности вектора (4.9).

Величина $v_\Delta = \left\|A^{-1}\right\|$ называется ***абсолютным числом обусловленности матрицы****, а величина $v_\delta = \left\|A^{-1}\right\| \cdot \|b\| / \|x\|$ —* ***относительным числом обусловленности****, которое характеризует коэффициент возможного возрастания относительной погрешности решения.*

Оценим максимальное значение относительного числа обусловленности:

$$\max_{x \neq 0} v_\delta = \max_{x \neq 0} \frac{\left\|A^{-1}\right\| \cdot \|Ax\|}{\|x\|} = \left\|A^{-1}\right\| \cdot \|A\|. \tag{4.13}$$

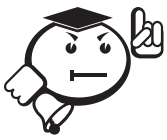
Полученную оценку принято называть стандартным числом обусловленности матрицы A и обозначать через $v(A)$ или $\mathrm{cond}(A)$.

Таким образом, мы можем записать

$$v(A) = \mathrm{cond}(A) = \left\|A^{-1}\right\| \cdot \|A\|. \tag{4.14}$$

С учетом (4.14) оценка (4.12) примет вид:

$$\delta(x) \leqslant \mathrm{cond}(A) \cdot \delta(b). \tag{4.15}$$



Свойства числа обусловленности

1. Для единичной матрицы $\mathrm{cond}(E) = 1$. Это следует из того, что $E^{-1} = E$ и $\|E\| = 1$. Действительно, $\|E\| = \max\limits_{x \neq 0} \dfrac{\|Ex\|}{\|x\|} = \max\limits_{x \neq 0} \dfrac{\|x\|}{\|x\|} = 1$. Поэтому $\mathrm{cond}(E) = \|E^{-1}\| \cdot \|E\| = 1$.
2. Справедливо неравенство $\mathrm{cond}(A) \geqslant 1$. Покажем это. Из равенства $E = A \cdot A^{-1}$ и свойства нормы $\|E\| \leqslant \|A^{-1}\| \cdot \|A\|$ имеем $1 \leqslant \|A^{-1}\| \cdot \|A\|$, что и требовалось доказать.
3. Число обусловленности матрицы A не меняется при умножении матрицы на произвольное число $c \neq 0$.

Имеет место теорема [9].

Теорема 4.2. Пусть x^* — решение системы (4.2) с приближенно заданной правой частью b^* и приближенной матрицей A^*. Тогда для относительной погрешности решения справедлива оценка

$$\delta(x) \leqslant v_\delta\big[\delta(b) + \delta(A)\big], \tag{4.16}$$

где $\delta(A) = \Delta(A)/\|A\|$ — относительная погрешность матрицы системы A.

Доказательство. Пусть дана система $Ax = b$.

Решение системы имеет вид:

$$x = A^{-1}b. \tag{4.17}$$

Предположим, что вместо точной матрицы A и правой части b нам даны приближённые т. е.:

$$A^* = A + \Delta A, \quad b^* = b + \Delta b,$$

где ΔA — матрица погрешностей элементов A; Δb — вектор погрешностей правой части.

Тогда вместо (4.17) мы получим приближённое решение:

$$x^* = (A + \Delta A)^{-1}(b + \Delta b) = x + \Delta x.$$

Вычислим Δx — погрешность решения.

$$\Delta x = x^* - x = A^{*-1}(b + \Delta b) - A^{-1}b = A^{*-1}b + A^{*-1}\Delta b - A^{-1}b =$$
$$= \left[A^{*-1}b - A^{-1}b\right] + A^{*-1}\Delta b = A^{*-1}\left[E - (A + \Delta A)A^{-1}\right]b + A^{*-1}\Delta b =$$
$$= A^{*-1}\left[E - E - \Delta A A^{-1}\right]b + A^{*-1}\Delta b = -A^{*-1}\Delta A A^{-1}b + A^{*-1}\Delta b = A^{*-1}\left[\Delta b - \Delta A x\right].$$

Таким образом, получим

$$\Delta x = A^{*-1}\left[\Delta b - \Delta A x\right]. \tag{4.18}$$

Поскольку точное решение x нам неизвестно, в (4.18) заменим x на $\tilde{x}$. В результате имеем:

$$\Delta x = A^{*-1}\left[\Delta b - \Delta A x^*\right],$$

где $x^* = A^{*-1}\tilde{b}$.

Переходя к нормам, получим:

$$\|\Delta x\| \leqslant \left\|A^{*-1}\right\| \left(\|\Delta b\| + \|\Delta A\|\,\|x^*\|\right). \tag{4.19}$$

Помножим и разделим правую часть (4.19) на $\|A^*\|$. В результате получим:

$$\|\Delta x\| \leqslant \left\|A^{*-1}\right\| \|A^*\| \left[\frac{\|\Delta b\|}{\|A^*\|} + \frac{\|\Delta A\|}{\|A^*\|}\|x^*\|\right].$$

Используя соотношение — неравенство $\|b^*\| \leqslant \|A^*\|\,\|x^*\|$, получим:

$$\|\Delta x\| \leqslant \left\|A^{*-1}\right\| \|A^*\| \left[\frac{\|\Delta b\|}{\|b^*\|}\|x^*\| + \frac{\|\Delta A\|}{\|A^*\|}\|x^*\|\right]. \tag{4.20}$$

Из (4.20) следует:

$$\frac{\|\Delta x\|}{\|x^*\|} \leqslant \operatorname{cond}(A)\left(\frac{\|\Delta b\|}{\|b^*\|} + \frac{\|\Delta A\|}{\|A^*\|}\right), \text{ ч. т. д.}$$

4.5 Прямые методы решения систем линейных алгебраических уравнений

4.5.1 Метод Гаусса

Рассмотрим один из самых распространенных методов решения СЛАУ — метод Гаусса.

Схема единственного деления

Пусть дана система

$$\left.\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &= b_2 \\ a_{n1}x_1 + x_{n2}x_2 + \ldots + a_{nn}x_n &= b_n \end{aligned}\right\}, \quad \Delta \neq 0. \tag{4.21}$$

Метод Гаусса *— это метод последовательного исключения неизвестных. Суть его состоит в преобразовании* (4.21) *к системе с треугольной матрицей (прямой ход), из которой затем последовательно (обратным ходом) получаются значения всех неизвестных.*

Рассмотрим одну из вычислительных схем. Эта схема называется схемой единственного деления.

Итак, рассмотрим эту схему. Пусть $a_{11} \neq 0$ (ведущий элемент первого шага).

Найдем величины [1]

$$m_{i1} = \frac{a_{i1}}{a_{11}} \ (i = 2, 3, \ldots, n), \tag{4.22}$$

которые называются множителями 1-го шага. Вычтем последовательно из второго, третьего, …, n-го уравнений системы (4.21) первое уравнение, умноженное соответственно на $m_{21}, m_{31}, \ldots, m_{n1}$. Это позволит обратить в нуль коэффициенты при x_1 во всех уравнениях, кроме первого. В результате получим эквивалентную систему:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1n}x_n &= b_1, \\ a_{22}^{(1)}x_2 + a_{23}^{(1)}x_3 + \ldots + a_{2n}^{(1)}x_n &= b_2^{(1)}, \\ a_{32}^{(1)}x_2 + a_{33}^{(1)}x_3 + \ldots + a_{3n}^{(1)}x_n &= b_3^{(1)}, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots & \\ a_{n2}^{(1)}x_2 + a_{n3}^{(1)}x_2 + \ldots + a_{nn}^{(1)}x_n &= b_n^{(1)}. \end{aligned} \tag{4.23}$$

Коэффициенты $a_{ij}^{(1)}$ на первом шаге вычисляются по формулам:

$$\begin{aligned} a_{ij}^{(1)} &= a_{ij} - m_{i1}a_{1j}, \\ b_i^{(1)} &= b_i - m_{i1}b_1, \\ i,\ j &= 2, 3, \ldots, n. \end{aligned} \tag{4.24}$$

На втором шаге мы исключаем из системы (4.24) x_2, начиная с третьего уравнения. Пусть $a_{22}^{(1)} \neq 0$ (ведущий элемент второго шага). Вычислим множители второго шага:

$$m_{i2} = \frac{a_{i2}^{(1)}}{a_{22}^{(1)}} (i = 3, 4, \ldots, n). \tag{4.25}$$

Из каждого уравнения системы (4.23), начиная с третьего, вычтем второе уравнение предварительно умноженное на соответствующий коэффициент $m_{32}, m_{42}, \ldots, m_{n2}$. В результате получим новую систему:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1n}x_n &= b_1, \\ a_{22}^{(1)}x_2 + a_{23}^{(1)}x_3 + \ldots + a_{2n}^{(1)}x_n &= b_2^{(1)}, \\ a_{33}^{(2)}x_3 + \ldots + a_{3n}^{(2)}x_n &= b_3^{(2)}, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots & \\ a_{n3}^{(2)}x_3 + \ldots + a_{nn}^{(2)}x_n &= b_n^{(2)}. \end{aligned} \tag{4.26}$$

Коэффициенты системы (4.26) рассчитываются по формулам:

$$\begin{aligned} a_{ij}^{(2)} &= a_{ij}^{(1)} - m_{i2}a_{2j}^{(1)}, \\ b_i^{(2)} &= b_i^{(1)} - m_{i2}b_2^{(1)}, \\ i,\ j &= 3,\ 4,\ \ldots,\ n. \end{aligned} \tag{4.27}$$

После $n-1$ шагов вместо (4.21) получим равносильную систему:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1n}x_n &= b_1, \\ a_{22}^{(1)}x_2 + a_{23}^{(1)}x_3 + \ldots + a_{2n}^{(1)}x_n &= b_2^{(1)}, \\ a_{33}^{(1)}x_3 + \ldots + a_{3n}^{(2)}x_n &= b_3^{(2)}, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots & \\ a_{nn}^{(n-1)}x_n &= b_n^{(n-1)}. \end{aligned} \tag{4.28}$$

Таким образом, на первом этапе мы получим треугольную систему с треугольной матрицей (4.28). Этот этап называется прямым ходом.

На втором этапе (обратный ход) мы находим последовательно из (4.28) значения $x_n, x_{n-1}, \ldots, x_1$ по формулам:

$$x_n = \frac{b_n^{(n-1)}}{a_{nn}^{(n-1)}}, \quad x_{n-i} = \frac{\left(b_{n-i}^{(n-i-1)} - \sum\limits_{j=0}^{i} a_{n-i,\,n-j}^{(n-i-1)} \cdot x_{n-j}\right)}{a_{n-i,\,n-i}^{(n-i-1)}},\ i = 1,\ 2,\ \ldots,\ n-1, \tag{4.29}$$

где $a_{1j}^{(0)} = a_{1j}$, $b_1^{(0)} = b_1$.

Трудоемкость метода

Оценим количество арифметических операций, необходимых для реализации схемы единственного деления.

На первом шаге прямого хода требуется $n-1$ операций деления, $(n+1)\cdot(n-1)$ операций умножения и $(n+1)\cdot(n-1)$ операций вычитания. Таким образом, общее число операций на первом шаге составляет $Q_1 = n-1+(n+1)(n-1)+(n+1)(n-1) = 2(n-1)^2+5(n-1)$. На втором шаге требуется $(n-2)$ операций деления, $n\cdot(n-2)$ операций умножения и $n\cdot(n-2)$ операций вычитания, т. е. $Q_2 = 2(n-2)^2+5(n-2)$. На k-ом шаге получим $Q_k = 2(n-k)^2+5(n-k)$, $k = 1,\ldots,n-1$. Найдем общее число арифметических операций Q прямого хода.

$$Q = \sum_{k=1}^{n-1} Q_k = \sum_{k=1}^{n-1}\left[2(n-k)^2+5(n-k)\right] = 2\sum_{j=1}^{n-1} j^2 + 5\sum_{j=1}^{n-1} j =$$
$$= \frac{2(n-1)n(2n-1)}{6} + \frac{5(n-1)n}{2} = \frac{4n^3+9n^2-13n}{6}.$$

При достаточно большом n можно пренебречь линейными членами, получим $Q \approx \frac{2}{3}n^3$.

Для реализации обратного хода по формулам (4.29) требуется операций: при $i = 1$ — одна операция умножения, одна операция вычитания и одна операция деления; при $i = 2$ — две операции умножения, две операции вычитания и одна операция

деления; при $i = (n-1)$ требуется $(n-1)$ операций умножения, $(n-1)$ операций вычитания и одна операция деления; при $i = n$ потребуется одна операция деления. Всего потребуется операций на обратном ходе $Q_1 = \sum_{k=1}^{n-1}(2k+1) + 1 = n^2$. При достаточно большом n величина Q_1 пренебрежимо мала по сравнению с числом операций прямого хода Q.

Таким образом, для реализации метода Гаусса требуется примерно $(2/3)n^3$ арифметических операций, причем подавляющая часть операций приходится на прямой ход.

4.5.2 QR-алгоритм решения СЛАУ

Рассмотрим другой метод приведения матриц к треугольному виду — метод вращения. Этот метод позволяет получить представление исходной матрицы A в виде произведения ортогональной матрицы Q на верхнюю треугольную матрицу R [1]:

$$A = QR. \tag{4.30}$$

На первом шаге прямого хода, состоящем из $(n-1)$ «малых» шагов, так же как и в методе Гаусса, исключаем неизвестное x_1 из всех уравнений, начиная со второго. Для исключения x_1 из второго уравнения вычисляют числа:

$$c_{12} = \frac{a_{11}}{\sqrt{a_{11}^2 + a_{21}^2}}, \quad s_{12} = \frac{a_{21}}{\sqrt{a_{11}^2 + a_{21}^2}}, \tag{4.31}$$

для которых выполняются условия:

$$c_{12}^2 + s_{12}^2 = 1, \quad -s_{12}a_{11} + c_{12}a_{21} = 0. \tag{4.32}$$

Затем первое уравнение системы заменяют линейной комбинацией первого и второго уравнений с коэффициентами c_{12} и s_{12}, а второе уравнение — линейной комбинацией первого и второго уравнений с коэффициентами $-s_{12}$ и c_{12}. В результате получим систему вида:

$$\begin{aligned}
a_{11}^{(1)}x_1 + a_{12}^{(1)}x_2 + a_{13}^{(1)}x_3 + \ldots + a_{1n}^{(1)}x_n &= b_1^{(1)},\\
a_{22}^{(1)}x_2 + a_{23}^{(1)}x_3 + \ldots + a_{2n}^{(1)}x_n &= b_2^{(1)},\\
a_{31}^{(1)}x_1 + a_{32}^{(1)}x_2 + a_{33}^{(1)}x_3 + \ldots + a_{3n}^{(1)}x_n &= b_3^{(1)},\\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots&\\
a_{n1}^{(1)}x_1 + a_{n2}^{(1)}x_2 + a_{n3}^{(1)}x_3 + \ldots + a_{nn}^{(1)}x_n &= b_n^{(1)},
\end{aligned} \tag{4.33}$$

в которой

$$a_{1j}^{(1)} = c_{12}a_{1j} + s_{12}a_{2j}, \quad a_{2j}^{(1)} = -s_{12}a_{1j} + c_{12}a_{2j}, \ (1 \leqslant j \leqslant n), \tag{4.34}$$

$$b_1^{(1)} = c_{12}b_1 + s_{12}b_2, \quad b_2^{(1)} = -s_{12}b_1 + c_{12}b_2. \tag{4.35}$$

Так как $-s_{12}a_{11} + c_{12}a_{21} = 0$, то $a_{21}^{(1)} = -s_{12}a_{11} + c_{12}a_{21} = 0$.

Преобразование исходной системы (4.1) к виду (4.33) эквивалентно умножению слева матрицы T_{12} на матрицу A и правую часть b, где матрица T_{12} имеет вид:

$$T_{12} = \begin{pmatrix} c_{12} & s_{12} & 0 & 0 & \dots & 0 \\ -s_{12} & c_{12} & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & 1 \end{pmatrix}.$$

Для исключения неизвестного x_1 из третьего уравнения вычислим числа:

$$c_{13} = \frac{a_{11}^{(1)}}{\sqrt{\left(a_{11}^{(1)}\right)^2 + a_{31}^2}}, \quad s_{13} = \frac{a_{31}}{\sqrt{\left(a_{11}^{(1)}\right)^2 + a_{31}^2}}, \tag{4.36}$$

удовлетворяющие условиям $c_{13}^2 + s_{13}^2 = 1$, $-s_{13}a_{11}^{(1)} + c_{13}a_{31} = 0$. Далее первое уравнение системы (4.33) заменим линейной комбинацией первого и третьего уравнений системы с коэффициентами c_{13} и s_{13}, а третье уравнение заменяем комбинацией с коэффициентами $-s_{13}$ и c_{13}. Это преобразование системы эквивалентно умножению слева на матрицу

$$T_{13} = \begin{pmatrix} c_{13} & 0 & s_{13} & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 \\ -s_{13} & 0 & c_{13} & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & 1 \end{pmatrix},$$

в результате чего, коэффициент при x_1 в преобразованном третьем уравнении обращается в нуль.

Таким же образом x_1 исключаем из уравнений с номерами $i = 4, \dots, n$. После завершения 1-го шага система приводится к виду:

$$\begin{aligned} a_{11}^{(n-1)}x_1 + a_{12}^{(n-1)}x_2 + a_{13}^{(n-1)}x_3 + \dots + a_{1n}^{(n-1)}x_n &= b_1^{(n-1)}, \\ a_{22}^{(1)}x_2 + a_{23}^{(1)}x_3 + \dots + a_{2n}^{(1)}x_n &= b_2^{(1)}, \\ a_{32}^{(1)}x_2 + a_{33}^{(1)}x_3 + \dots + a_{3n}^{(1)}x_n &= b_3^{(1)}, \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots & \\ a_{n2}^{(1)}x_2 + a_{n3}^{(1)}x_3 + \dots + a_{nn}^{(1)}x_n &= b_n^{(1)}. \end{aligned} \tag{4.37}$$

В матричной форме (4.37) имеет вид:

$$A^{(1)}x = b^{(1)},$$

где $A^{(1)} = T_{1n}\dots T_{13}T_{12}A$, $b^{(1)} = T_{1n}\dots T_{13}T_{12}b$.

Здесь через T_{kl} обозначена матрица элементарного преобразования, отличающаяся от единичной матрицы только четырьмя элементами, а именно: элементы

с индексами (k,k) и (l,l) равны c_{kl}, элемент с индексами (k,l) равен s_{kl}, а элемент с индексами (l,k) равен $-s_{kl}$. Элементы c_{kl} и s_{kl} удовлетворяют условию

$$c_{kl}^2 + s_{kl}^2 = 1. \qquad (4.38)$$

Действие матрицы T_{kl} на вектор x эквивалентно его повороту вокруг оси, перпендикулярной плоскости Ox_kx_l, на угол φ_{kl}, такой, что $c_{kl} = \cos\varphi_{kl}$, $s_{kl} = \sin\varphi_{kl}$. По этой причине метод назвали методом вращений. Матрица T_{kl} ортогональна, т. е. $T_{kl}^T = T_{kl}^{-1}$.

На втором шаге метода вращений, состоящем из $(n-2)$ «малых» шагов, из уравнений системы (4.37) с номерами $i = 3, 4, \ldots, n$ исключают неизвестное x_2. Для этого каждое i-е уравнение комбинируют со вторым уравнением. В результате получим систему:

$$\begin{aligned}
a_{11}^{(n-1)}x_1 + a_{12}^{(n-1)}x_2 + a_{13}^{(n-1)}x_3 + \ldots + a_{1n}^{(n-1)}x_n &= b_1^{(n-1)},\\
a_{22}^{(n-1)}x_2 + a_{23}^{(n-1)}x_3 + \ldots + a_{2n}^{(n-1)}x_n &= b_2^{(n-1)},\\
a_{33}^{(2)}x_3 + \ldots + a_{3n}^{(2)}x_n &= b_3^{(2)},\\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots&\\
a_{n3}^{(2)}x_3 + \ldots + a_{nn}^{(2)}x_n &= b_n^{(2)}.
\end{aligned}$$

В матричной форме эта запись имеет вид:

$$A^{(2)}x = b^{(2)},$$

где $A^{(2)} = T_{2n}\ldots T_{24}T_{23}A^{(1)}$, $b^{(1)} = T_{2n}\ldots T_{24}T_{23}b^{(1)}$.

После завершения $(n-1)$-го шага система примет вид:

$$\begin{aligned}
a_{11}^{(n-1)}x_1 + a_{12}^{(n-1)}x_2 + a_{13}^{(n-1)}x_3 + \ldots + a_{1n}^{(n-1)}x_n &= b_1^{(n-1)},\\
a_{22}^{(n-1)}x_2 + a_{23}^{(n-1)}x_3 + \ldots + a_{2n}^{(n-1)}x_n &= b_2^{(n-1)},\\
a_{33}^{(n-1)}x_3 + \ldots + a_{3n}^{(n-1)}x_n &= b_3^{(n-1)},\\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots&\\
a_{nn}^{(n-1)}x_n &= b_n^{(n-1)}.
\end{aligned}$$

или в матричной форме

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

где $A^{(n-1)} = T_{n-1,n}A^{(n-2)}$, $b^{(n-1)} = T_{n-1,n}b^{(n-2)}$.

Обозначим за R матрицу $A^{(n-1)}$. Тогда матрица R и A связаны соотношением

$$R = TA,$$

где $T = T_{n-1,n}\ldots T_{2n}\ldots T_{23}T_{1n}\ldots T_{13}T_{12}$ — матрица результирующего вращения. Отметим, что матрица T ортогональна как произведение ортогональных матриц. Обозначая $Q = T^{-1} = T^T$, получаем QR-разложение матрицы A.

Переходим к решению системы $Ax = b$. Представляя матрицу A в виде $A = QR$, получим $QRx = b$. Отсюда, $Rx = Q^Tb$. Обратный ход метода вращений совпадает с обратным ходом метода Гаусса.

Метод вращений обладает хорошей вычислительной устойчивостью. Погрешность решения может быть оценена по формуле [1]:

$$\delta(x) \leqslant 6n\|A\|_3\left\|A^{-1}\right\|_3\varepsilon_m,$$

где ε_m — машинное эпсилон (машинный нуль). Однако этот метод значительно более трудоемок по сравнению с методом Гаусса. Для квадратной матрицы требуется примерно $2n^3$ арифметических операций.

4.5.3 Метод ортогонализации

Справедлива теорема [3].

Теорема 4.3. Всякую действительную неособенную матрицу A можно представить в виде произведения матрицы с ортогональными столбцами R и верхней треугольной матрицы T с единичной диагональю $A = RT$.

Доказательство. Пусть имеем матрицу с действительными элементами

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}.$$

Столбцы матрицы A обозначим как векторы $a_j = \begin{pmatrix} a_{1j} \\ a_{2j} \\ \dots \\ a_{nj} \end{pmatrix}$, тогда $A = (a_1 \dots, a_n)$.

Так как A — неособенная, то a_j линейно-независимы. Будем искать матрицу R в виде:

$$R = (r_1, \dots, r_n),$$

где r_j — искомые ортогональные столбцы.

Имеем:

$$RT = \begin{pmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ r_{21} & r_{22} & \dots & r_{2n} \\ \dots & \dots & \dots & \dots \\ r_{n1} & r_{n2} & \dots & r_{nn} \end{pmatrix} \cdot \begin{pmatrix} 1 & t_{12} & \dots & t_{1n} \\ 0 & 1 & \dots & t_{2n} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}$$

Отсюда получим:

$$\begin{gathered} r_1 = a_1,\ t_{12}r_1 + r_2 = a_2,\ t_{13}r_1 + t_{23}r_2 + r_3 = a_3,\ \dots, \\ t_{1n}r_1 + t_{2n}r_2 + t_{3n}r_3 + \dots + t_{n-1n}r_{n-1} + r_n = a_n. \end{gathered} \tag{4.39}$$

Умножая второе равенство (4.39) на r_1, получим с учетом ортогональности:

$$t_{12} = \frac{(a_2 r_1)}{(r_1 r_1)}; \quad r_2 = a_2 - t_{12}r_1.$$

Умножим третье равенство (4.39) на r_1 и r_2, получим:

$$t_{13} = \frac{(r_1 a_3)}{(r_1 r_1)}, \quad t_{23} = \frac{(r_2 a_3)}{(r_2 r_2)}, \quad r_3 = a_3 - t_{13} r_1 - t_{23} r_2.$$

В общем случае имеем:

$$r_1 = a_1, \quad r_2 = a_2 - t_{12} r_1, \quad r_i = a_i - \sum_{k=1}^{i-1} t_{ki} r_k, \quad t_{ij} = \frac{(a_j r_i)}{(r_i r_i)}, \quad i < j.$$

Теорема доказана.

Применим теперь этот результат к решению системы

$$Ax = b.$$

Представим матрицу A в виде $A = RT$, получим

$$RTx = b. \tag{4.40}$$

Умножим (4.40) слева на R^T

$$R^T RTx = R^T b. \tag{4.41}$$

Так как R имеет ортогональные столбцы, то $R^T R = D$, где $D = \operatorname{diag}(d_1, \ldots, d_n)$, $d_i = \sum_{k=1}^{n} r_{ik}^2$. Поэтому из (4.41) имеем

$$Tx = D^{-1} R^T b, \tag{4.42}$$

где $D^{-1} = \operatorname{diag}(d_1^{-1}, \ldots, d_n^{-1})$.

Система (4.42) имеет верхнюю треугольную матрицу с единичной диагональю, поэтому она легко разрешается.

Введем обозначение $y = D^{-1} R^T b$, тогда (4.42) примет вид:

$$Tx = y. \tag{4.43}$$

Решение (4.43) имеет вид $x = T^{-1} y$ или (см. обратный ход метода Гаусса):

$$x_n = y_n,$$
$$x_{n-1} = y_{n-1} - t_{n-1,n} x_n,$$
$$x_{n-i} = y_{n-i} - \sum_{k=1}^{i} t_{n-k,n-k+1} x_{n-k+1}, \; i = 1, \ldots, n-1.$$

Трудоемкость метода

Оценим количество арифметических операций метода ортогонализации.

Для вычисления каждого элемента матрицы T требуется $2n$ операций сложения, $2n$ операций умножения и одна операция деления. Всего элементов требуется вычислить $(n^2 - n)/2$. Таким образом, для вычисления всех элементов матрицы T требуется $Q_T = (4n + 1) \cdot (n^2 - n)/2 \approx 2n^2(n - 1)$ арифметических операций.

Для вычисления вектора r_2 матрицы R требуется n операций умножения и n операций вычитания, всего $2n$ операций. Для вычисления вектора r_3 требуется выполнить $4n$ операций умножения и вычитания; для вектора r_n количество операций составляет $(n-1)\cdot 2n$. Таким образом, для вычисления матрицы R требуется $Q_R = 2n\sum_{k=2}^{n}(k-1) = 2n\cdot(n-1)n/2 = n^2(n-1)$ операций умножения и вычитания. Общее количество операций на ортогональное разложение матрицы A составляет $Q_A = 3n^2(n-1) \approx 3n^3$.

Для вычисления вектора y требуется $2n^2 + n$ операций умножения и сложения. И наконец, для вычисления вектора решения x необходимо выполнить n^2 операций (см. трудоемкость обратного хода схемы Гаусса). Общее количество операций метода ортогонализации составляет $Q = 3n^3 + 3n^2 + n$.

4.5.4 Метод Халецкого

Пусть дана система

$$Ax = d, \tag{4.44}$$

где A — квадратная матрица размерности $(n \times n)$. Представим матрицу A в виде произведения нижней треугольной матрицы B и верхней треугольной матрицы C с единичной диагональю [1, 3]:

$$A = BC. \tag{4.45}$$

Тогда система (4.44) может быть представлена в виде двух систем с треугольными матрицами:

$$By = d, \quad Cx = y. \tag{4.46}$$

Системы (4.46) легко решаются:

$$y_1 = \frac{d_1}{b_{11}}, \quad y_i = \frac{1}{b_{ii}}\left(d_i - \sum_{k=1}^{i-1} b_{ik}y_k\right),\ i = 2, 3, \ldots, n;$$

$$x_n = y_n, \quad x_i = y_i - \sum_{k=i+1}^{n} c_{ik}x_k,\ i = n-1, n-2, n-3, \ldots, 1. \tag{4.47}$$

Как вычислить элементы матриц B и C? Перемножая матрицы B и C и приравнивая элементы матрицы-произведения соответствующим элементам матрицы A, получим следующие вычислительные формулы:

$$\begin{cases} b_{i1} = a_{i1},\ b_{ij} = a_{ij} - \sum_{k=1}^{j-1} b_{ik}c_{kj},\ (i \geqslant j > 1), \\ c_{1j} = \frac{a_{1j}}{b_{11}},\ c_{ij} = \frac{1}{b_{ii}}\left(a_{ij} - \sum_{k=1}^{i-1} b_{ik}c_{kj}\right),\ (1 < i < j). \end{cases} \tag{4.48}$$

Формулы (4.48) представляют собой алгоритм разложения квадратной матрицы на две треугольные матрицы.

4.6 Итерационные методы решения СЛАУ

4.6.1 Метод простой итерации решения СЛАУ

При большом числе неизвестных схема метода Гаусса, дающая точное решение, становится весьма сложной. В этом случае для решения СЛАУ иногда удобнее пользоваться приближенными методами. Рассмотрим один из приближенных методов — метод итерации.

Итак, имеем СЛАУ

$$\left.\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &= b_2 \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n &= b_n \end{aligned}\right\} \ \Delta \neq 0. \tag{4.49}$$

Предполагая, что $a_{ii} \neq 0$ $(i = 1, \ldots, n)$, разрешим первое уравнение системы (4.49) относительно x_1, второе — относительно x_2, …, n-ое уравнение — относительно x_n. В результате получим:

$$\left.\begin{aligned} x_1 &= \beta_1 + \alpha_{12}x_2 + \alpha_{13}x_3 + \ldots + \alpha_{1n}x_n, \\ x_2 &= \beta_2 + \alpha_{21}x_1 + \alpha_{23}x_3 + \ldots + \alpha_{2n}x_n, \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ x_n &= \beta_n + \alpha_{n1}x_1 + \alpha_{n2}x_2 + \ldots + \alpha_{nn-1}x_{n-1}, \end{aligned}\right\} \tag{4.50}$$

где $\beta_i = b_i/a_{ii}$; $\alpha_{ij} = -a_{ij}/a_{ii}$ при $i \neq j$; $\alpha_{ii} = 0$. Система (4.50) в матричной форме имеет вид:

$$x = \beta + \alpha x. \tag{4.51}$$

Здесь β — вектор-столбец размерности n, состоящий из элементов β_i $(i = 1, 2, \ldots, n)$, α — матрица размерности $(n \times n)$ с элементами α_{ij}.

Систему (4.51) будем решать методом последовательных приближений. Пусть $x^{(0)} = \beta$, тогда

$$\left.\begin{aligned} x^{(1)} &= \beta + \alpha x^{(0)}, \\ x^{(2)} &= \beta + \alpha x^{(1)}, \\ x^{(k+1)} &= \beta + \alpha x^{(k)}. \end{aligned}\right\} \tag{4.52}$$

Если последовательность $\{x^{(k)}\}$ имеет предел $\xi = \lim\limits_{k\to\infty} x^{(k)}$, то этот предел является решением системы (4.51). В самом деле, переходя к пределу в равенстве (4.52), будем иметь $\xi = \beta + \alpha\xi$, то есть предельный вектор ξ является решением системы (4.51). Справедлива следующая теорема [3, 7, 9].

Теорема 4.4. Если $\|\alpha\| < 1$, то система уравнений (4.51) имеет единственное решение и итерационный процесс (4.52) сходится к решению независимо от начального приближения.

Доказательство. Имеем:

$$x^{(k+1)} = \alpha x^{(k)} + \beta.$$

Тогда $x^{(k+1)} - x^{(k)} = \alpha\left(x^{(k)} - x^{(k-1)}\right)$.

Отсюда

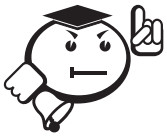
$$\left\|x^{(k+1)} - x^{(k)}\right\| \leqslant \|\alpha\| \left\|x^{(k)} - x^{(k-1)}\right\| \leqslant \|\alpha\|^k \left\|x^{(1)} - x^{(0)}\right\|.$$

Поэтому если $\|\alpha\| < 1$, то последовательность $\{x^{(k)}\}$ является сходящейся, то есть $x^{(k)} \underset{k\to\infty}{\longrightarrow} \xi$.

Тогда, переходя к пределу в итерационном процессе

$$\lim_{k\to\infty} x^{(k+1)} = \alpha \lim_{k\to\infty} x^{(k)} + \beta,$$

получим $\xi = \alpha\xi + \beta$, ч. т. д.



Следствие. Для системы

$$\sum_{j=1}^{n} a_{ij}x_j = b_i \ (i = 1, 2, \ldots, n)$$

метод итераций сходится, если выполнены неравенства

$$|a_{ii}| > \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}|, \ i = 1, 2, \ldots, n. \tag{4.53}$$

Действительно, для системы (4.50)

$$x = \alpha x + \beta$$

сходимость процесса выполняется, если $\|\alpha\|_1 < 1$. Так как $\alpha_{ij} = -a_{ij}/a_{ii}$, $i \neq j$, $\alpha_{ii} = 0$, то из определения нормы матрицы $\|\alpha\|_1$ получим (4.53).

4.6.2 Подготовка системы для итерационного процесса

Прежде чем применять метод простой итерации, необходимо переставить строки исходной системы таким образом, чтобы на диагонали стояли наибольшие по модулю коэффициенты матрицы. Если при этом условие (4.53) все таки не выполняется, то иногда удается обеспечить сходимость метода с помощью следующего приема [9].

Пусть дана система $Ax = b$. Преобразуем ее к виду $x = Qx + c$, где

$$Q = E - DA, \quad c = Db. \tag{4.54}$$

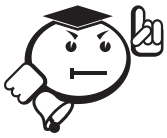
Здесь D — некоторая матрица. Нам необходимо подобрать такую матрицу D, чтобы выполнялось условие $\|Q\| < 1$. Докажем следующую теорему [9].

Теорема 4.5. Если диагональные элементы матрицы $R = A^TA$ удовлетворяют условию

$$r_{ii} > \sum_{j \neq i}^{n} |r_{ij}|, \ i = \overline{1,n}, \tag{4.55}$$

то метод итерации $x^{(k+1)} = Qx^{(k)} + c$ сходится независимо от начального приближения.

Матрица Q имеет вид $Q = E - DA = E - PA^TA$, где $P = \text{diag}(p_1, p_2, \ldots p_n) = \text{diag}\left\{(a^1a^1)^{-1}, (a^2a^2)^{-1}, \ldots, (a^na^n)^{-1}\right\}$ — диагональная матрица.
Здесь $a^j = (a_{1j}, a_{2j}, \ldots, a_{nj})^T$ — j-ый столбец матрицы A.



Если в качестве матрицы D взять A^{-1}, то в (4.54) $Q = 0$ и для вектора x получим $x = c = A^{-1}b$.

4.6.3 Метод Зейделя

Метод Зейделя представляет собой модификацию метода простой итерации.
Пусть дана приведённая система:

$$x_i = \beta_i + \sum_{\substack{j=1 \\ j \neq i}}^{n} \alpha_{ij}x_j, \ i = \overline{1,n}$$

и известно начальное приближение $x^{(0)} = \left(x_1^{(0)}, x_2^{(0)}, \ldots, x_n^{(0)}\right)$. Основная идея заключается в том, что при вычислении $(k+1)$-го приближения неизвестной x_i учитываются уже вычисленные ранее $(k+1)$ — приближение неизвестных $x_1, x_2, \ldots, x_{i-1}$.

Итерационная схема имеет вид:

$$\begin{aligned}
x_1^{(k+1)} &= \beta_1 + \sum_{j=2}^{n} \alpha_{ij}x_j^{(k)}, \\
x_2^{(k+1)} &= \beta_2 + \alpha_{21}x_1^{(k+1)} + \sum_{j=3}^{n} \alpha_{2j}x_j^{(k)}, \\
x_i^{(k+1)} &= \beta_i + \sum_{j=1}^{i-1} \alpha_{ij}\,x_j^{(k+1)} + \sum_{j=i+1}^{n} \alpha_{ij}\,x_j^{(k)}, \\
x_{n-1}^{(k+1)} &= \beta_{n-1} + \sum_{j=1}^{n-2} \alpha_{n-1,j}\,x_j^{(k+1)} + \alpha_{n-1,n}x_n^{(k)}, \\
x_n^{(k+1)} &= \beta_n + \sum_{j=1}^{n-1} \alpha_{nj}\,x_j^{(k+1)}, \ k = 0, 1, 2, \ldots
\end{aligned}$$

Положим $\alpha = B + C$, где

$$B = \left(\begin{array}{ccc:c} 0 & 0 & 0 & 0 \\ \alpha_{21} & 0 & 0 & 0 \\ \alpha_{31} & \alpha_{32} & 0 & 0 \\ \hdashline \alpha_{n1} & \alpha_{n2} & \alpha_{n3} & 0 \end{array}\right); \quad C = \left(\begin{array}{ccc:c} 0 & \alpha_{12} & \alpha_{13} & \alpha_{1n} \\ 0 & 0 & \alpha_{23} & \alpha_{2n} \\ 0 & 0 & 0 & \alpha_{3n} \\ \hdashline 0 & 0 & 0 & 0 \end{array}\right).$$

Тогда процесс Зейделя в матричном виде можно записать как

$$x^{(k+1)} = Bx^{(k+1)} + Cx^{(k)} + \beta.$$

Достаточное условие сходимости процесса Зейделя такое же, как и для метода итераций [3]. Имеет место теорема.

Теорема 4.6. Если для системы

$$x = \alpha x + \beta \tag{4.56}$$

выполняется условие $\|\alpha\| < 1$, то процесс Зейделя для системы (4.56) сходится к единственному её решению при любом выборе начального вектора $x^{(0)}$.

4.7 Оценка погрешности метода простой итерации и процесса Зейделя

Запишем разность между точным решением системы $x = \alpha \cdot x + \beta$ и n-ым приближением $x^{(n)} = \alpha \cdot x^{(n-1)} + \beta$

$$x - x^{(n)} = \alpha \cdot x - \alpha \cdot x^{(n-1)}. \tag{4.57}$$

Прибавим и отнимем в правой части (4.57) слагаемое $\alpha \cdot x^{(n)}$, в результате получим

$$x - x^{n} = \alpha \cdot x - \alpha \cdot x^{(n)} + \alpha \cdot x^{(n)} - \alpha \cdot x^{(n-1)} = \alpha \cdot \left(x - x^{(n)}\right) + \alpha \cdot \left(x^{(n)} - x^{(n-1)}\right). \tag{4.58}$$

Перейдем к нормам в (4.58)

$$\left\|x - x^{(n)}\right\| \leqslant \|\alpha\| \cdot \left\|x - x^{(n)}\right\| + \|\alpha\| \cdot \left\|x^{(n)} - x^{(n-1)}\right\|$$

или

$$\left\|x - x^{(n)}\right\| \leqslant \frac{\|\alpha\|}{1 - \|\alpha\|} \left\|x^{(n)} - x^{(n-1)}\right\|. \tag{4.59}$$

Из (4.59) следует, что в качестве критерия окончания итерационного процесса следует брать

$$\left\|x^{(n)} - x^{(n-1)}\right\| \leqslant \frac{1 - \|\alpha\|}{\|\alpha\|}\varepsilon. \tag{4.60}$$

Если $\|\alpha\|$ мало ($\|\alpha\| \ll 1$), то вместо (4.60) можно использовать

$$\left\|x^{(n)} - x^{(n-1)}\right\| \leqslant \varepsilon. \tag{4.61}$$

Можно показать, что для процесса Зейделя погрешность корня определяется формулой (4.59), и поэтому критерием окончания итерационного процесса будет (4.60).

4.8 Процесс Зейделя для нормальной системы

Рассмотрим один из способов преобразования системы

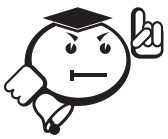
$$Ax = b, \tag{4.62}$$

позволяющий всегда получать сходящийся процесс Зейделя. Помножим (4.62) слева на A^T

$$A^T Ax = A^T b \text{ или } Cx = d, \tag{4.63}$$

где $C = A^T A$; $d = A^T b$.

Систему (4.63) принято называть *нормальной*. (Такая система получается естественным образом при использовании метода наименьших квадратов для решения переопределенной системы.)



Нормальная система обладает рядом замечательных свойств:

1) матрица C — симметрическая;
2) все элементы главной диагонали $c_{ij} > 0$;
3) матрица C — положительно определена.

Первое свойство следует из определения

$$c_{ij} = \sum_{k=1}^{n} a_{ik}^T a_{kj} = \sum a_{ki} a_{kj} = \sum a_{kj} a_{ki} = c_{ji}.$$

Второе свойство вытекает также из определения

$$c_{ii} = \sum_{k=1}^{n} a_{ki} a_{ki} = \sum_{k=1}^{n} a_{ki}^2 > 0.$$

Рассмотрим третье свойство: $x^T Cx > 0$, $x \neq 0$.
Действительно:

$$\sum_{i=1}^{n}\sum_{j=1}^{n} x_i c_{ij} x_j = \sum_{i=1}^{n} x_i \sum_{j=1}^{n} x_j \sum_{k=1}^{n} a_{ki} a_{kj} = \sum_{k=1}^{n} \left(\sum_{i=1}^{n} x_i a_{ki}\right) \sum_{j=1}^{n} x_j a_{kj} = \sum_{k=1}^{n} \left(\sum_{i=1}^{n} x_i a_{ki}\right)^2 > 0.$$

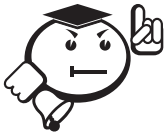
Приведем без доказательства следующую теорему [3].

Теорема 4.7. Пусть система

$$Ax = b$$

нормальная, т. е. A — симметричная, положительно определённая матрица и имеющая положительные диагональные элементы. Тогда процесс Зейделя сходится при любом выборе начального приближения.

4.9 Метод прогонки



Метод прогонки применяют для решения ленточных систем.

В качестве примера рассмотрим трехдиагональную систему ленточного вида. Такая система получается, например, в задаче построения кубических интерполяционных сплайнов.

Итак, пусть имеем трёхдиагональную систему $Ax = g$ размерности $n \times n$ или в скалярном виде:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 = g_1, \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = g_2, \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ a_{i\,i-1}x_{i-1} + a_{i\,i}x_i + a_{i\,i+1}x_{i+1} = g_i, \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ a_{n\,n-1}x_{n-1} + a_{n\,n}x_n = g_n. \end{cases}$$

Из последнего уравнения найдём: $x_n = (g_n - a_{n\,n-1}x_{n-1})\dfrac{1}{a_{n\,n}}$, подставим в предпоследнее уравнение и из него получим $x_{n-1} = (r_{n-1} - a_{n-1\,n-2}x_{n-2})\dfrac{1}{v_{n-1}}$, где $v_{n-1} = a_{n-1\,n-1} - a_{n-1\,n}a_{n\,n-1}\dfrac{1}{a_{n\,n}}$, $r_{n-1} = g_{n-1} - a_{n-1\,n}g_n\dfrac{1}{a_{n\,n}}$.

Продолжая этот процесс, придем к следующим рекуррентным формулам [9]:

$$\begin{cases} a_{10} = 0;\ x_0 = 0, \\ x_{n-j} = \dfrac{1}{v_{n-j}}(r_{n-j} - a_{n-j,\,n-j-1}x_{n-j-1}),\ r_{n-j} = g_{n-j} - \dfrac{a_{n-j,\,n-j+1}}{v_{n-j+1}}r_{n-j+1}, \\ v_{n-j} = a_{n-j,\,n-j} - a_{n-j,\,n-j+1}\dfrac{a_{n-j+1,\,n-j}}{v_{n-j+1}},\ j = 1, \ldots, n-1, \\ v_n = a_{n\,n};\ r_n = g_n. \end{cases}$$

Обозначим $n - j = i$. В результате будем иметь:

$$\begin{cases} r_i = g_i - \dfrac{a_{i,\,i+1}}{v_{i+1}} r_{i+1}, \\ v_i = a_{ii} - a_{i,\,i+1} \dfrac{a_{i+1,\,i}}{v_{i+1}}, \ i = n-1, n-2, \ldots, 1, \\ v_n = a_{nn}; \ r_n = g_n, \ a_{10} = 0, \ x_0 = 0, \\ x_i = \dfrac{1}{v_i}(r_i - a_{i,\,i-1} x_{i-1}), \ i = 1, \ldots, n. \end{cases}$$

4.10 Решение переопределенной системы линейных уравнений

Пусть дана система

$$Ax = b, \tag{4.64}$$

где A — $(n \times m)$-матрица, причем $n \geqslant m$; x — m-мерный вектор неизвестных; b — n-мерный вектор правой части.

Если $n > m$, то есть число уравнений больше числа неизвестных, то говорят, что система (4.64) переопределена. Переопределенные системы линейных уравнений возникают при обработке экспериментальных данных, в регрессионном анализе и при решении различных задач физики, геологии и т. д.

Для решения переопределенной СЛАУ используют метод наименьших квадратов (МНК) [3, 7, 9, 10]. Идея его состоит в минимизации суммы квадратов невязок.

Введем понятие *невязки*. Невязкой i-го наблюдения называется число r_i:

$$r_i = \sum_{j=1}^{m} a_{ij} x_j - b_i; \ i = 1, \ldots, n.$$

Суть МНК заключается в том, чтобы неизвестные x_j находить из условия минимума функции $R(x_1, x_2, \ldots, x_n)$, т. е.

$$R(x_1, x_2, \ldots, x_m) = \sum_{i=1}^{n} r_i^2 = \sum_{i=1}^{n} \left[\sum_{j=1}^{m} a_{ij} x_j - b_i \right]^2 \to \min_{x_j}$$

Необходимые условия минимума

$$\frac{\partial R(x_1, \ldots, x_m)}{\partial x_k} = 0, \ k = 1, \ldots, m$$

приводят к системе линейных уравнений, в которой количество уравнений будет равно m, то есть совпадает с числом неизвестных. Взяв производные по x_k от функции $R(x_1, \ldots, x_m)$, получим следующую систему:

$$\sum_{j=1}^{m} c_{kj} x_j = g_k, \ k = 1, \ldots, m, \tag{4.65}$$

где

$$g_k = \sum_{i=1}^{n} a_{ik} b_i, \quad c_{kj} = c_{jk} = \sum_{i=1}^{n} a_{ij} a_{ik}.$$

В матричном виде система (4.65) запишется как

$$Cx = g, \tag{4.66}$$

где $C = A^T A$; $g = A^T b$.

В системе (4.66) матрица C — квадратная, симметричная размерности $(m \times m)$. Для решения (4.66) можно использовать любой метод (например, Гаусса).

4.11 Вычисление определителей

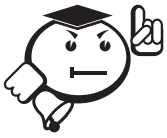
Пусть имеется квадратная матрица A размером $(n \times n)$.

Определителем *называется алгебраическая сумма всевозможных произведений элементов, взятых по одному из каждого столбца и каждой строки матрицы A. Если в каждом таком произведении (члене определителя) множители расположены в порядке следования столбцов (т. е. вторые индексы элементов a_{ij} в произведении расположены в порядке возрастания), то со знаком $(+)$ берутся те произведения, у которых перестановка первых индексов чётная, а со знаком $(-)$ берутся те, у которых она нечетная.*

$$\det A = |A| = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{vmatrix} = \sum (-1)^{[i_1,\dots,i_n]} \cdot a_{i_1 1} \cdot a_{i_2 2} \dots a_{i_n n}.$$

Здесь $[i_1, i_2, \dots, i_n]$ — число инверсий в перестановке индексов i_1, i_2, …, i_n.

4.11.1 Свойства определителей



Приведем известные из курса высшей математики свойства определителей:

1. При транспонировании матрицы её определитель не меняется.
2. Если поменять местами две строки или два столбца определителя, то определитель изменит знак, а по абсолютной величине не изменится.
3. Пусть $C = AB$, где A и B квадратные матрицы. Тогда $\det C = \det A \cdot \det B$.

4. Определитель с двумя одинаковыми строками или с двумя одинаковыми столбцами равен 0.
5. Определитель с двумя пропорциональными строками или столбцами равен 0.
6. Определитель треугольной матрицы равен произведению диагональных элементов.
7. Если все элементы строки (столбца) умножить на одно и то же число, то определитель умножится на это число.
8. Если каждый элемент некоторой строки (столбца) определителя представлен в виде суммы двух слагаемых, то определитель равен сумме двух определителей, у которых все строки (столбцы), кроме данной, прежние, а в данной строке (столбце) в первом определителе стоят первые, а во втором — вторые слагаемые.

. . . . . .

4.11.2 Вычисление определителей методом Гаусса

Итак, применим метод Гаусса для вычисления Δ. При решении системы уравнений:

$$Ax = b$$

методом Гаусса мы путём преобразования по схеме единственного деления привели её к треугольному виду:

$$Bx = \beta,$$

где $B = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & \dots & a_{2n}^{(1)} \\ & & a_{33}^{(2)} & \dots & a_{3n}^{(2)} \\ \dots & \dots & \dots & \dots & \dots \\ & & & & a_{nn}^{(n-1)} \end{pmatrix}$, $\beta = \begin{pmatrix} b_1^{(1)} \\ b_2^{(2)} \\ \dots \\ b_n^{(n-1)} \end{pmatrix}$.

Определитель $\det B = a_{11} \cdot a_{22}^{(1)} \cdot a_{33}^{(2)} \cdots a_{nn}^{(n-1)}$ (см. свойство 6).

Элементы матрицы B получили из матрицы A с помощью элементарных преобразований, а именно путем вычитания из строк матрицы A и промежуточных матриц A_i, $\left(i = \overline{1, n-1}\right)$ чисел, пропорциональных элементам соответствующих ведущих строк. Эти элементарные преобразования не изменяют величины определителя, поэтому $\det A = \det B$.

Если используется схема Гаусса с выбором главного элемента, то необходимо запоминать, сколько раз мы меняем местами строки, т. е. запоминать количество перестановок строк, поскольку по свойству 2 при перестановке строк определитель меняет знак, а по абсолютной величине не изменяется. Если количество перестановок четное, то $\det A = \det B$, если — нечетное, то $\det A = -\det B$.

4.11.3 Вычисление определителей методом Халецкого

Рассмотрим ещё один алгоритм вычисления определителя квадратной матрицы. Этот алгоритм основан на идее представления исходной матрицы в виде произведения двух треугольных матриц. Ранее (см. п. 4.6.6) мы получили формулы разложения квадратной матрицы A в виде произведения нижней треугольной матрицы B и верхней треугольной матрицы C с единичной диагональю (см. формулы (4.58)), т. е. $A = BC$. Тогда, используя 3 и 6 свойства определителей, мы можем записать

$$\det A = \det B \cdot \det C = \prod_{i=1}^{n} b_{ii} = b_{11} \cdot b_{22} \cdots b_{nn},$$

так как $\det C = 1$.

4.12 Вычисление обратной матрицы

Пусть дана неособенная матрица $A = (a_{ij})_{i,j=1}^{n}$. Для нахождения её обратной матрицы

$$X = (x_{ij})_{i,j=1}^{n} = A^{-1}$$

используем основное соотношение

$$AX = E, \tag{4.67}$$

где E — единичная матрица.

Выражение (4.67) представляет собой n систем уравнений относительно n^2 неизвестных x_{ij}

$$\sum_{k=1}^{n} a_{ik} x_{kj} = \delta_{ij};\ \left(i,\ j = \overline{1,n}\right), \tag{4.68}$$

где $\delta_{ij} = \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases}$.

Полученные n систем решаем одним из известных методов, при этом матрица коэффициентов одна и та же для всех n систем, а меняются лишь правые части (свободные члены).

Контрольные вопросы по главе 4

1. Дайте классификацию методов решения алгебраических задач.
2. Какие нормы векторов и согласованные с ними нормы матриц вам известны?
3. Что понимают под обусловленностью задачи решения системы линейных алгебраических уравнений? Как определяется число обусловленности?
4. Дайте определение абсолютной и относительной погрешности векторов.

5. Сформулируйте основную идею метода Гаусса. Трудоемкость метода (схема единственного деления).
6. Рассмотрите QR-алгоритм решения СЛАУ.
7. Сформулируйте основную идею метода ортогонализации решения СЛАУ.
8. Сформулируйте основную идею метода декомпозиции (схемы Халецкого) решения СЛАУ.
9. Чем отличается метод Зейделя от метода простой итерации?
10. Запишите критерий завершения итерационных методов.
11. Как решить переопределенную систему линейных алгебраических уравнений?
12. В чем состоит идея метода Гаусса вычисления определителя?
13. В чем состоит идея метода Халецкого вычисления определителя?
14. Какое свойство матриц используется в численных методах вычисления обратных матриц?

Глава 5

ВЫЧИСЛЕНИЕ СОБСТВЕННЫХ ЗНАЧЕНИЙ И СОБСТВЕННЫХ ВЕКТОРОВ МАТРИЦ

5.1 Постановка задачи

Пусть задана квадратная матрица A порядка n с вещественными элементами $(a_{ij})_{i,j=1}^{n}$.

Собственными значениями матрицы *A называются числа* λ*, удовлетворяющие системе уравнений* [1, 7]

$$Ax = \lambda x, \tag{5.1}$$

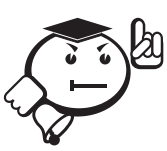
где x *— собственный вектор* $(x \neq 0)$*, отвечающий собственному значению* λ.

Собственное значение λ и собственный вектор x в общем случае комплексные. Перепишем (5.1) в виде:

$$(A - \lambda E)x = 0. \tag{5.2}$$

Эта однородная система имеет ненулевое решение x тогда и только тогда, когда определитель матрицы равен нулю, т. е.

$$\det(A - \lambda E) = 0. \tag{5.3}$$



Таким образом, собственные значения являются корнями характеристического уравнения.

Известно, что характеристическое уравнение имеет в области комплексных чисел ровно n корней $\lambda_1, \lambda_2, \ldots, \lambda_n$ (с учетом их кратности).

Если матрица A симметричная, то все ее собственные значения являются вещественными числами. Для несимметричных матриц возможно наличие комплексных собственных значений вида $\lambda = \alpha + i\beta$ с ненулевой мнимой частью. В этом случае собственным значением обязательно является и комплексно-сопряженное число $\lambda = \alpha - i\beta$.

В ряде задач механики, физики, химии, техники, биологии требуется получение всех собственных значений некоторых матриц, а иногда и всех собственных векторов. В такой постановке задачу называют *полной проблемой собственных значений*.

На практике часто определению подлежат не все собственные значения и собственные векторы, а лишь небольшая их часть. Например, существенный интерес во многих приложениях представляют максимальное или минимальное по модулю собственное значение или же собственное значение, наиболее близко расположенное к заданному значению. Такие задачи являются примерами *частичных проблем собственных значений*.

Численные методы решения проблемы собственных значений сводятся в конечном счете к решению характеристического уравнения (5.3):

$$f_n(\lambda) = \lambda^n + p_1\lambda^{n-1} + p_2\lambda^{n-2} + \ldots + p_{n-1}\lambda + p_n = 0. \qquad (5.4)$$

При реализации такого подхода основные усилия направлены на разработку эффективных методов быстрого вычисления коэффициентов характеристического уравнения p_i ($i = 1, 2, \ldots, n$). Методы такого класса получили названия *прямых* [1]; к ним относятся методы Крылова, Данилевского, Леверье и др.

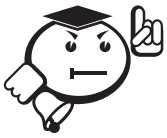
Однако указанный подход становится неудовлетворительным, если речь идет о вычислении собственных значений матриц, имеющих порядок n в несколько десятков (и тем более сотен), т. е. матриц по современным понятиям небольших размеров.

Одна из причин состоит в том, что хотя задачи (5.1) и (5.4) формально эквивалентны, они имеют разную обусловленность. Так как корни многочлена $f_n(\lambda)$ высокой степени чрезвычайно чувствительны к погрешностям в коэффициентах, то на этапе вычисления коэффициентов характеристического уравнения может быть в значительной степени потеряна информация о собственных значениях матрицы.

С появлением компьютеров широкое распространение получили итерационные методы решения проблемы собственных значений, не использующие вычисление характеристического многочлена.

5.2 Преобразование подобия

Говорят, что матрицы A и B ***подобны****, если существует невырожденная матрица P (матрица подобия), такая, что $B = P^{-1}AP$* [1]*. Само преобразование матрицы A к виду $B = P^{-1}AP$ называют* ***преобразованием подобия****.*



Из теории матриц известно, что преобразование подобия не изменяет характеристического многочлена, т. е. полученная в результате преобразования подобия матрица имеет тот же набор собственных чисел.

В методе Данилевского матрица A приводится к матрице Фробениуса

$$B = \begin{pmatrix} b_1 & b_2 & b_3 & \dots & b_n \\ 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}. \tag{5.5}$$

Тогда собственные значения являются корнями следующего уравнения:

$$f_n(\lambda) = (-1)^n \left[\lambda^n - b_1\lambda^{n-1} - b_2\lambda^{n-2} - \dots - b_{n-1}\lambda - b_n\right]. \tag{5.6}$$

Оказывается, преобразованием подобия матрицу A можно привести к более простому виду, чем (5.5). Справедливы следующие теоремы [1].

Теорема 5.1. Любую квадратную матрицу A с помощью преобразования подобия можно привести к следующему виду:

$$P^{-1}AP = \Lambda = \begin{pmatrix} \lambda_1 & \sigma_1 & 0 & \dots & 0 & 0 \\ 0 & \lambda_2 & \sigma_2 & \dots & 0 & 0 \\ 0 & 0 & \lambda_3 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & \lambda_{n-1} & \sigma_{n-1} \\ 0 & 0 & 0 & \dots & 0 & \lambda_n \end{pmatrix}. \tag{5.7}$$

Здесь $\lambda_1, \lambda_2, \dots, \lambda_n$ — собственные числа матрицы A. Числа σ_i принимают одно из двух значений 0 или 1, причем если $\sigma_i = 1$, то обязательно $\lambda_i = \lambda_{i+1}$, т. е. имеем кратные корни.

Матрица (5.7) называется жордановой формой матрицы A.

Если матрица A с помощью преобразования подобия приводится к диагональному виду:

$$P^{-1}AP = D = \begin{pmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \lambda_1 \end{pmatrix}, \tag{5.8}$$

то матрицу A называют матрицей простой структуры. В этом случае столбцы матрицы P являются собственными векторами матрицы A. Действительно, (5.8) мы можем переписать в виде $AP = PD$. Это равенство верно только в том случае, если матрица P является матрицей, состоящей из собственных векторов матрицы A.

Теорема 5.2. Если все собственные значения матрицы A различны, то она является матрицей простой структуры.

Теорема 5.3. Если A — вещественная симметричная матрица, то она подобна диагональной матрице, причем матрица подобия P может быть выбрана ортогональной (т. е. удовлетворяющей условию $P^{-1} = P^T$).

5.3 Локализация собственных значений

Пусть $r_i = \sum\limits_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}|$ — сумма модулей недиагональных элементов i-ой строки матрицы A. Обозначим через S_i замкнутый круг радиуса r_i на комплексной плоскости с центром в точке a_{ii}, т. е. $S_i = \{z \in C : |z - a_{ii}| \leqslant r_i\}$. Будем называть круги S_i *кругами Гершгорина*. Имеет место следующая теорема [1].

Теорема 5.4 (теорема Гершгорина). Все собственные значения матрицы A лежат в объединении кругов $S_1, S_2, S_1, \dots, S_n$.

Доказательство. Возьмем произвольное собственное значение λ матрицы A и соответствующий собственный вектор x. Пусть x_i — максимальная по модулю координата вектора x. Запишем i-е уравнение системы (5.1) в следующем виде:

$$(a_{ii} - \lambda)x_i = -\sum_{\substack{j=1 \\ j \neq i}}^{n} a_{ij}x_j.$$

Из этого равенства с учетом оценки $|x_j/x_i| \leqslant 1$ следует

$$|a_{ii} - \lambda| \leqslant \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}| \left|\frac{x_j}{x_i}\right| \leqslant r_i.$$

Таким образом, $\lambda \in S_i$.

Пример 5.1

Дана матрица $A = \begin{pmatrix} -2 & 0.5 & 0.5 \\ -0.5 & -3.5 & 1.5 \\ 0.8 & -0.5 & 0.5 \end{pmatrix}$. Круги Гершгорина изображены на рис. 5.1. Здесь $r_1 = 1$, $r_2 = 2$, $r_3 = 1.3$.

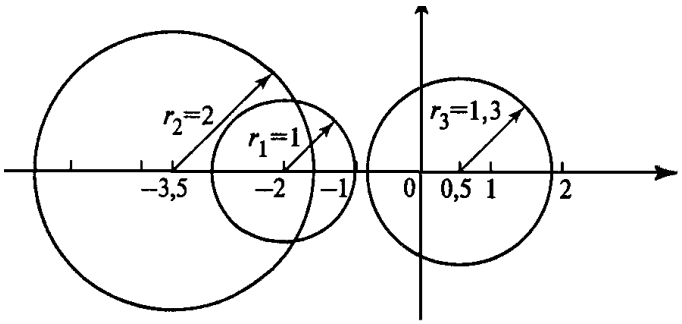
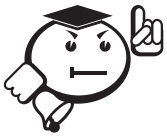


Рис. 5.1

Теорема 5.5 [1]. Если k кругов Гершгорина образуют замкнутую область G, изолированную от других кругов, то в области G находится ровно k собственных значений матрицы A (с учетом их кратности).



Следствие. Если какой-либо круг Гершгорина изолирован, то он содержит ровно одно собственное значение матрицы A.

Пример 5.2

Для матрицы A из примера 1 в объединении кругов S_1 и S_2 находится ровно два собственных значения λ_1 и λ_2, а круг S_3 содержит ровно одно собственное значение λ_3.

5.4 Обусловленность задачи вычисления собственных значений и собственных векторов

Пусть A^* — матрица с приближенно заданными элементами $a_{ij}^* \approx a_{ij}$. Обозначим через λ_j^* $(j = 1, 2, \ldots, n)$ собственные числа матрицы A^*.

Теорема 5.6 [1]. Пусть A и A^* — симметричные матрицы, а λ_j и λ_j^* — их собственные числа, упорядоченные по возрастанию. Тогда справедлива оценка погрешности

$$\left(\sum_{j=1}^{n} (\lambda_j - \lambda_j^*)^2 \right)^{1/2} \leqslant \|A - A^*\|_3.$$

Из теоремы 5.6 следует, что задача вычисления собственных значений симметричных матриц хорошо обусловлена. Следовательно, в этом случае собственные числа надежно определяются заданием элементов матрицы. Однако для несимметричных матриц дело обстоит совсем иначе. Хотя задача вычисления собственных значений и в этом случае является устойчивой, для многих несимметричных матриц собственные значения чрезвычайно чувствительны к погрешностям задания коэффициентов.

Отметим, что число обусловленности $\mathrm{cond}(A)$ не характеризует обусловленность матрицы A по отношению к проблеме собственных значений. Оказывается, что такой характеристикой чувствительности собственных значений относительно погрешности задания матрицы для матрицы простой структуры служит число обусловленности матрицы P, столбцы которой являются собственными векторами матрицы A.

Теорема 5.7. Пусть $P^{-1}AP = D$, где D — диагональная матрица из собственных значений матрицы A. Тогда каждое собственное значение матрицы A^* удалено от некоторого собственного значения матрицы A не более чем на $d = \mathrm{cond}_3(P)\|A - A^*\|_3$.

Пусть x — собственный вектор матрицы A, отвечающий собственному значению λ, а x^* — собственный вектор приближенно заданной матрицы A^*, отвечающий собственному значению λ^*. В качестве меры близости x^* к вектору x принимают величину $|\sin\varphi|$, где φ — угол между векторами x^* и x, вычисляемый по формуле [1]:

$$\varphi = \arccos\left(\frac{(x^*, x)}{\|x^*\|\|x\|}\right).$$

Задача вычисления собственных векторов симметричной матрицы хорошо обусловлена, если собственные значения хорошо отделены друг от друга. Имеет место следующая теорема [1].

Теорема 5.8. Пусть A и A^* — симметричные матрицы. Тогда справедлива оценка

$$|\sin\varphi| \leqslant \frac{\|A - A^*\|_3}{\gamma}.$$

Здесь φ — угол между векторами x^* и x, а γ — расстояние от λ^* до ближайшего из несовпадающих с λ собственных значений матрицы A.

В случае когда матрица A несимметрична, задача вычисления собственных векторов может оказаться плохо обусловленной.

5.5 Степенной метод вычисления максимального собственного числа

Пусть требуется вычислить максимальное по модулю вещественное собственное число λ_1 матрицы A, при этом известно, что

$$|\lambda_1| > |\lambda_2| \geqslant |\lambda_3| \geqslant \ldots \geqslant |\lambda_n|. \tag{5.9}$$

Возьмем произвольный начальный вектор $x^{(0)} \neq 0$ и построим последовательность векторов $\{u^{(k)}\}$, $\{y^{(k)}\}$ и приближений $\left\{\lambda_1^{(k)}\right\}$ к λ_1 по формулам [1, 3, 7]:

$$u^{(k)} = Ay^{(k-1)}, \quad \lambda_1^{(k)} = \left(u^{(k)}, y^{(k-1)}\right), \quad y^{(k)} = \frac{u^{(k)}}{\|u^{(k)}\|}, \; k = 1, 2, \ldots \tag{5.10}$$

В качестве $y^{(0)}$ выбирается вектор с единичной нормой.

Справедлива следующая теорема [1].

Теорема 5.9. Пусть A — матрица простой структуры, для которой выполнено условие (5.9). Предположим, что в разложении

$$y^{(0)} = c_1x_1 + c_2x_2 + \ldots + c_nx_n \tag{5.11}$$

по базису из собственных векторов коэффициент c_1 не равен нулю. Тогда $\lambda_1^{(k)} \underset{k\to\infty}{\longrightarrow} \lambda_1$ и верна оценка относительной погрешности

$$\delta\left(\lambda_1^{(k)}\right) = \frac{\left|\lambda_1^{(k)} - \lambda_1\right|}{|\lambda_1|} \leqslant C_0\left|\frac{\lambda_2}{\lambda_1}\right|^k, \tag{5.12}$$

где $C_0 = C_0\left(x^{(0)}\right)$.

5.6 QR-алгоритм вычисления собственных чисел

В настоящее время лучшим методом вычисления всех собственных значений квадратных заполненных матриц общего вида (умеренного порядка) является QR-алгоритм.

Алгоритм основан на разложении произвольной матрицы в произведение ортогональной и верхней треугольной матриц, т. е. на так называемом ***QR-разложении****.*

На 1-ой итерации с помощью метода вращений вычисляют QR-разложение матрицы $A^{(0)} = A$, которое имеет вид:

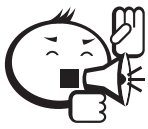
$$A^{(0)} = Q_1R_1.$$

Затем строят матрицу $A^{(1)} = R_1Q_1$. Из равенства $A^{(0)} = Q_1R_1$ следует, что $R_1 = Q_1^{-1}A^{(0)}$, а значит, $A^{(1)} = Q_1^{-1}A^{(0)}Q_1$. Таким образом, матрицы $A^{(1)}$ и $A^{(0)}$ подобны и поэтому имеют общий набор собственных чисел $\lambda_1, \lambda_2, \ldots, \lambda_n$.

На 2-ой итерации находят QR-разложение матрицы $A^{(1)} = Q_2R_2$ и вычисляют матрицу $A^{(2)} = R_2Q_2$, подобную матрице $A^{(1)}$, т. е. $A^{(2)} = Q_2^{-1}A^{(1)}Q_2$.

На $(k+1)$-й итерации вычисляют разложение $A^{(k)} = Q_{k+1}R_{k+1}$ и строят матрицу $A^{(k+1)} = R_{k+1}Q_{k+1}$, подобную матрице $A^{(k)}$, т. е. $A^{(k+1)} = Q_{k+1}^{-1}A^{(k)}Q_{k+1}$. Таким образом, мы получаем последовательность матриц $A^{(1)}, A^{(2)}, \ldots, A^{(n)}, \ldots$, подобных матрице A.

С увеличением k матрицы $A^{(k)}$ сходятся по форме к некоторой верхней треугольной или к верхней блочно-треугольной матрице $\tilde{A}$, имеющей те же собственные числа, что и матрица A.



. . . . . .

Приведем условия сходимости QR-алгоритма:

1) матрица A имеет простую структуру, причем модули всех собственных значений различны: $|\lambda_1| > |\lambda_2| > \ldots > |\lambda_n|$;
2) приведение матрицы A к диагональному виду (5.8) осуществляется с помощью матрицы подобия P, у которой все ведущие главные миноры отличны от нуля.

При выполнении этих двух условий последовательность $A^{(k)}$ сходится по форме к верхней треугольной матрице $\widetilde{A}$ вида:

$$\tilde{A} = \begin{pmatrix} \lambda_1 & \square & \square & \ldots & \square \\ 0 & \lambda_2 & \square & \ldots & \square \\ 0 & 0 & \lambda_3 & \ldots & \square \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & \lambda_n \end{pmatrix}.$$

Здесь квадратиками помечены элементы, в общем случае не равные нулю.

. . . . . .

Известно, что в рассматриваемом случае элементы $a_{ij}^{(k)}$ матрицы $A^{(k)}$, стоящие ниже главной диагонали, сходятся к нулю со скоростью геометрической прогрессии, при этом выполняется неравенство

$$\left|a_{ij}^{(k)}\right| \leqslant C\left|\frac{\lambda_i}{\lambda_j}\right|^k, \ i > j,$$

т. е. скорость сходимости $a_{ij}^{(k)}$ к нулю определяется значением отношения $|\lambda_i/\lambda_j| < 1$.

Если условие 2) не выполнено, то сходимость по-прежнему имеет место, но собственные значения в матрице $\tilde{A}$ уже не будут расположены в порядке убывания модулей.

В общем случае предельная матрица $\tilde{A}$ — блочно-треугольная (или получающаяся из блочно-треугольной симметричной перестановкой строк и столбцов). Наличие комплексно-сопряженных пар λ_k, $\overline{\lambda}_k$ собственных значений у вещественной матрицы A не является препятствием для применения QR-алгоритма. Каждой такой паре в предельной матрице будет отвечать некоторый диагональный блок — квадратная подматрица порядка 2, собственные числа которой совпадают с λ_k, $\overline{\lambda}_k$.

5.7 Метод обратных итераций вычисления собственных векторов

Метод обратных итераций применяют для нахождения собственных векторов матрицы A [1].

Рассмотрим задачу вычисления собственного вектора x_j при условии, что уже найдено достаточно точное приближение λ_j^* к собственному значению λ_j.

Если исходить непосредственно из определения собственного вектора, то x_j следует искать как нетривиальное решение однородной системы уравнений

$$(A - \lambda_j E)x = 0 \tag{5.13}$$

с вырожденной матрицей $(A - \lambda_j E)$. Однако λ_j известно лишь приближенно, и в действительности при таком подходе вместо системы (5.13) придется решать систему

$$(A - \lambda_j^* E)x = 0. \tag{5.14}$$

Так как матрица $(A - \lambda_j^* E)$ заведомо не вырождена, то решением системы (5.14) является только $x = 0$. Следовательно, непосредственное численное решение системы (5.14) не дает возможности вычислить собственный вектор.

Одним из эффективных методов вычисления собственных векторов является *метод обратных итераций* [1]. В этом методе приближения $y^{(k)}$ к собственному вектору x определяют последовательным решением систем уравнений

$$(A - \lambda_j^* E)y^{(k+1)} = x^{(k)}, \quad x^{(0)} = (1, 1, \ldots, 1)^T \tag{5.15}$$

с последующей нормировкой решения:

$$x^{(k+1)} = \frac{y^{(k+1)}}{\|y^{(k+1)}\|_3}. \tag{5.16}$$

Если абсолютная погрешность значения λ_j^* много меньше расстояния от λ_j до ближайшего из остальных чисел (т. е. выполняется условие $|\lambda_j - \lambda_j^*| \ll |\lambda_i - \lambda_j^*|$), то метод итерации обычно сходится за 1–3 итерации.

5.8 Метод Данилевского

5.8.1 Вычисление собственных чисел

Сущность ***метода Данилевского*** *заключается в приведении определителя матрицы* $A - \lambda E$ [3]:

$$D(\lambda) = \det(A - \lambda E)$$

к нормальному виду Фробениуса:

$$D(\lambda) = \begin{vmatrix} b_1 - \lambda & b_2 & b_3 & \ldots & b_n \\ 1 & -\lambda & 0 & \ldots & 0 \\ 0 & 1 & -\lambda & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & 1 & -\lambda \end{vmatrix}. \tag{5.17}$$

Если нам удалось записать определитель в форме (5.17)*, то, разлагая определитель* (5.17) *по элементам 1-ой строки, получим полином* (5.6)*:*

$$D(\lambda) = (b_1 - \lambda)(-\lambda)^{n-1} - b_2(-\lambda)^{n-2} + b_3(-\lambda)^{n-3} - \ldots + (-1)^{n-1}b_n$$

или

$$D(\lambda) = (-1)^n\left[\lambda^n - b_1\lambda^{n-1} - b_2\lambda^{n-2} - b_3\lambda^{n-3} - \ldots - b_n\right].$$

. . . . . .

Определитель (5.17) есть определитель матрицы $B - \lambda E$, где B — матрица Фробениуса (см. (5.5))

$$B = \begin{pmatrix} b_1 & b_2 & b_3 & \ldots & b_n \\ 1 & 0 & 0 & \ldots & 0 \\ 0 & 1 & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix},$$

определяемая преобразованием подобия $B = P^{-1}AP$ матрицы A, где P — неособенная матрица.

Как указывалось выше, подобные матрицы обладают одинаковыми характеристическими полиномами и, следовательно, одинаковыми собственными числами:

$$\det(A - \lambda E) = \det(B - \lambda E).$$

Итак, получим матрицу B. Начнём с последней строки. Нам нужно строку

$$a_{n,1}a_{n,2}\ldots a_{n,n-1}a_{n,n}$$

перевести в строку

$$00\ldots 10.$$

Предположим, что $a_{n,n-1} \neq 0$ и разделим элементы $(n-1)$-го столбца матрицы A на $a_{n,n-1}$. Тогда n-ая строка примет вид:

$$a_{n,1}a_{n,2}\ldots 1a_{n,n}.$$

Затем вычтем $(n-1)$-ый столбец преобразованной матрицы, умноженный соответственно на числа $a_{n,1}, a_{n,2}, \ldots, a_{n,n}$, из всех остальных её столбцов. В результате получим матрицу $\tilde{A}_1$:

$$\tilde{A}_1 = \begin{pmatrix} \tilde{a}^{(1)}_{1,1} & \tilde{a}^{(1)}_{1,2} & \ldots & \tilde{a}^{(1)}_{1,n-1} & \tilde{a}^{(1)}_{1,n} \\ \tilde{a}^{(1)}_{2,1} & \tilde{a}^{(1)}_{2,2} & \ldots & \tilde{a}^{(1)}_{2,n-1} & \tilde{a}^{(1)}_{2,n} \\ \tilde{a}^{(1)}_{3,1} & \tilde{a}^{(1)}_{3,2} & \ldots & \tilde{a}^{(1)}_{3,n-1} & \tilde{a}^{(1)}_{3,n} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & 1 & 0 \end{pmatrix}. \tag{5.18}$$

Указанные операции являются элементарными преобразованиями матрицы A.

Элементы матрицы $\tilde{A}_1$ вычисляются по формулам:

$$\left.\begin{aligned} \tilde{a}^{(1)}_{i,n-1} &= \frac{a_{i,n-1}}{a_{n,n-1}},\ i = \overline{1,n}; \\ \tilde{a}^{(1)}_{i,j} &= a_{i,j} - a_{n,j}\frac{a_{i,n-1}}{a_{n,n-1}},\ j = 1, 2, \ldots, n;\ j \neq n-1. \end{aligned}\right\} \qquad (5.19)$$

Произведённые операции над матрицей A равносильны умножению исходной матрицы A на M_{n-1}

$$\tilde{A}_1 = AM_{n-1},$$

где M_{n-1} — матрица, имеющая вид:

$$M_{n-1} = \begin{pmatrix} 1 & 0 & \ldots & 0 & 0 \\ 0 & 1 & \ldots & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ m_{n-1,1} & m_{n-1,2} & \ldots & m_{n-1,n-1} & m_{n-1,n} \\ 0 & 0 & \ldots & 0 & 1 \end{pmatrix}$$

с элементами

$$m_{n-1,i} = -\frac{a_{n,i}}{a_{n,n-1}} \text{ при } i = 1, 2, \ldots, n;\ i \neq n-1,$$

$$m_{n-1,n-1} = \frac{1}{a_{n,n-1}}. \qquad (5.20)$$

Построенная матрица $\tilde{A}_1 = AM_{n-1}$ (5.18) не будет подобна матрице A. Чтобы иметь преобразование подобия, нужно обратную матрицу M^{-1}_{n-1} умножить слева на $\tilde{A}_1$:

$$A_1 = M^{-1}_{n-1}\tilde{A}_1 = M^{-1}_{n-1}AM_{n-1}.$$

Матрица M^{-1}_{n-1} имеет вид:

$$M^{-1}_{n-1} = \begin{pmatrix} 1 & 0 & \ldots & 0 & 0 \\ 0 & 1 & \ldots & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ a_{n,1} & a_{n,2} & \ldots & a_{n,n-1} & a_{n,n} \\ 0 & 0 & \ldots & 0 & 1 \end{pmatrix}.$$

Матрица A_1 будет иметь следующие элементы:

$$A_1 = \begin{pmatrix} a^{(1)}_{1,1} & a^{(1)}_{1,2} & \ldots & a^{(1)}_{1,n-1} & a^{(1)}_{1,n} \\ a^{(1)}_{2,1} & a^{(1)}_{2,2} & \ldots & a^{(1)}_{2,n-1} & a^{(1)}_{2,n-1} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ a^{(1)}_{n-2,1} & a^{(1)}_{n-2,2} & \ldots & a^{(1)}_{n-2,n-1} & a^{(1)}_{n-2,n} \\ a^{(1)}_{n-1,1} & a^{(1)}_{n-1,2} & \ldots & a^{(1)}_{n-1,n-1} & a^{(1)}_{n-1,n} \\ 0 & 0 & \ldots & 1 & 0 \end{pmatrix},$$

т. е.

$$a_{i,j}^{(1)} = \tilde{a}_{i,j}^{(1)} \text{ при } i,j = 1, 2, \ldots, n;\ i \neq n-1;$$
$$a_{n-1,j}^{(1)} = \sum_{k=1}^{n} a_{n,k}\tilde{a}_{k,j}^{(1)};\ 1 \leqslant j \leqslant n. \tag{5.21}$$

Таким образом, умножение M_{n-1}^{-1} слева на $\tilde{A}_1$ меняет лишь $(n-1)$-ю строку матрицы $\tilde{A}_1$. Полученная матрица A_1 подобна матрице A. Этим заканчивается 1-ый этап процесса.

Далее, если $a_{n-1,n-2}^{(1)} \neq 0$, то над матрицей A_1 проделаем те же операции, взяв за основу $(n-2)$-ой столбец. В результате получим матрицу A_2:

$$A_2 = \begin{pmatrix} a_{1,1}^{(2)} & a_{1,1}^{(2)} & \ldots & a_{1,n-2}^{(2)} & a_{1,n-1}^{(2)} & a_{1,n}^{(2)} \\ a_{2,1}^{(2)} & a_{2,1}^{(2)} & \ldots & a_{2,n-2}^{(2)} & a_{2,n-1}^{(2)} & a_{2,n}^{(2)} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ a_{n-2,1}^{(2)} & a_{n-2,2}^{(2)} & \ldots & a_{n-2,n-2}^{(2)} & a_{n-2,n-1}^{(2)} & a_{n-2,n}^{(2)} \\ 0 & 0 & \ldots & 1 & 0 & 0 \\ 0 & 0 & \ldots & 0 & 1 & 0 \end{pmatrix}$$

с элементами

$$a_{i,j}^{(2)} = \tilde{a}_{i,j}^{(2)},\ \text{при } i,j = 1, 2, \ldots, n;\ i \neq n-2;$$
$$a_{n-2,j}^{(2)} = \sum_{k=1}^{n} a_{n-1,k}^{(1)}\tilde{a}_{k,j}^{(2)};\ 1 \leqslant j \leqslant n, \tag{5.22}$$

где

$$\begin{cases} \tilde{a}_{i,n-2}^{(2)} = \dfrac{a_{i,n-2}^{(1)}}{a_{n-1,n-2}^{(1)}};\ i = 1, 2, \ldots, n, \\ \tilde{a}_{i,j}^{(2)} = a_{i,j}^{(1)} - a_{n-1,j}^{(1)}\dfrac{a_{i,n-2}^{(1)}}{a_{n-1,n-2}^{(1)}};\ i, j = 1, 2, \ldots, n;\ j \neq n-2. \end{cases} \tag{5.23}$$

Матрица A_2 получена преобразованием подобия

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

где матрицы M_{n-2} и M_{n-2}^{-1} имеют вид:

$$M_{n-2} = \begin{pmatrix} 1 & 0 & \ldots & 0 & 0 & 0 \\ 0 & 1 & \ldots & 0 & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ m_{n-2,1} & m_{n-2,2} & \ldots & m_{n-2,n-2} & m_{n-2,n-1} & m_{n-2,n} \\ 0 & 0 & \ldots & 1 & 0 & 0 \\ 0 & 0 & \ldots & 0 & 1 & 0 \end{pmatrix},$$

$$M_{n-2}^{-1} = \begin{pmatrix} 1 & 0 & \ldots & 0 & 0 & 0 \\ 0 & 1 & \ldots & 0 & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ a_{n-1,1}^{(1)} & a_{n-1,2}^{(1)} & \ldots & a_{n-1,n-2}^{(1)} & a_{n-1,n-1}^{(1)} & a_{n-1,n}^{(1)} \\ 0 & 0 & \ldots & 1 & 0 & 0 \\ 0 & 0 & \ldots & 0 & 1 & 0 \end{pmatrix}.$$

Элементы матрицы M_{n-2} вычисляются по формулам:

$$\begin{cases} m_{n-2,j} = -\dfrac{a^{(1)}_{n-1,j}}{a^{(1)}_{n-1,n-2}},\ j = 1, 2, \ldots, n;\ j \neq n-2, \\ m_{n-2,n-2} = \dfrac{1}{a^{(1)}_{n-1,n-2}}. \end{cases} \tag{5.24}$$

Продолжая процесс, мы получим матрицу Фробениуса:

$$B = M_1^{-1} \ldots M_{n-2}^{-1} M_{n-1}^{-1} A M_{n-1} M_{n-2} \ldots M_1.$$

Таким образом, формулы (5.21)–(5.24) представляют собой алгоритм вычисления матрицы Фробениуса. Полученные элементы матрицы Фробениуса b_1, b_2, …, b_n подставляем в уравнение (5.6) и, приравнивая характеристический полином нулю, вычисляем собственные числа матрицы B, которые равны собственным числам матрицы A.

5.8.2 Вычисление собственных векторов

Найдем собственный вектор:

$$y = (y_1, y_2, \ldots, y_n)$$

матрицы Фробениуса, соответствующий данному λ:

$$By = \lambda y.$$

Отсюда $(B - \lambda E)\, y = 0$ или

$$\begin{pmatrix} b_1 - \lambda & b_2 & b_3 & \ldots & b_{n-1} & b_n \\ 1 & -\lambda & 0 & \ldots & 0 & 0 \\ 0 & 1 & -\lambda & \ldots & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & 1 & -\lambda \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \ldots \\ y_{n-1} \\ y_n \end{pmatrix} = 0.$$

Перемножая матрицы, получим систему для определения координат y_1, y_2, …, y_n собственного вектора y:

$$\left.\begin{aligned} (b_1 - \lambda)\, y_1 + b_2 y_2 + \ldots + b_n y_n &= 0, \\ y_1 - \lambda y_2 + \ldots \quad \ldots &= 0, \\ y_2 - \lambda y_3 + \quad \ldots &= 0, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots & \\ y_{n-1} - \lambda y_n &= 0. \end{aligned}\right\} \tag{5.25}$$

Система (5.25) однородная. С точностью до коэффициента пропорциональности решение её может быть найдено следующим образом. Положим $y_n = 1$, тогда последовательно получим:

$$y_n = 1,\ y_{n-1} = \lambda,\ y_{n-2} = \lambda^2,\ \ldots,\ y_1 = \lambda^{n-1}.$$

Теперь вспомним определение матрицы B:

$$B = M_1^{-1} \dots M_{n-1}^{-1} A M_{n-1} \dots M_1.$$

Отсюда

$$By = M_1^{-1} \dots M_{n-1}^{-1} A M_{n-1} \dots M_1 y$$

или

$$M_{n-1} \dots M_1 By = A M_{n-1} \dots M_1 y. \tag{5.26}$$

Заменим в левой части (5.26) By на λy, получим:

$$\lambda M_{n-1} \dots M_1 y = A M_{n-1} \dots M_1 y$$

или

$$Ax = \lambda x,$$

где $x = M_{n-1} \dots M_1 y$ — собственный вектор матрицы A.

Матрица M_1 имеет вид:

$$M_1 = \begin{pmatrix} m_{11} & m_{12} & \dots & m_{1n} \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{pmatrix}, \qquad \begin{array}{l} m_{1,k} = -\dfrac{a_{2,k}^{(n-2)}}{a_{2,1}^{(n-2)}};\ k = 2, 3, \dots, n \\ m_{1,1} = \dfrac{1}{a_{2,1}^{(n-2)}}, \end{array} \tag{5.27}$$

следовательно, преобразование $M_1 y$ изменит лишь первую координату вектора y:

$$x_1 = \sum_{k=1}^{n} m_{1,k} y_k = y_1^{(1)},$$

а остальные останутся без изменения: $y_k^{(1)} = y_k$, $k = \overline{2, n}$.

Преобразование $M_2 M_1 y$ изменит лишь вторую координату вектора $M_1 y$:

$$x_2 = \sum_{k=1}^{n} m_{2,k} y_k^{(1)} = y_2^{(2)}; \quad y_k^{(2)} = y_k^{(1)},\ k = \overline{1, n},\ k \neq 2.$$

Повторяя этот процесс, получим:

$$x_i = \sum_{k=1}^{n} m_{i,k} y_k^{(i-1)} = y_i^{(i)}; \quad y_k^{(i)} = y_k^{(i-1)};\ k = \overline{1, n},\ k \neq i,\ i = \overline{1, n-1};$$

$$\begin{cases} m_{i,k} = -\dfrac{a_{i+1,k}^{(j)}}{a_{i+1,i}^{(j)}};\ k = \overline{1, n},\ k \neq i,\ j = n - i - 1, \\ m_{i,i} = \dfrac{1}{a_{i+1,i}^{(j)}};\ i = \overline{1, n-1}. \end{cases}$$

$$x_n = y_n = 1.$$

Контрольные вопросы по главе 5

1. Дайте определение собственных чисел и собственных векторов матрицы.
2. Что такое преобразование подобия? Какая матрица называется подобной данной матрице A?
3. Приведите формулировку теоремы Гершгорина.
4. Запишите формулу погрешности собственных чисел симметричной матрицы.
5. Каким числом характеризуется обусловленность матрицы A по отношению к проблеме собственных значений?
6. Каким числом характеризуют близость собственных векторов точно заданной матрицы и ее приближения?
7. Опишите алгоритм степенного метода вычисления максимального собственного числа.
8. Опишите QR-алгоритм вычисления собственных чисел.
9. В чем суть метода обратных итераций нахождения собственных векторов?
10. В чем состоит идея метода Данилевского вычисления собственных чисел?
11. Как вычисляют собственные вектора матрицы Фробениуса?

Глава 6

ПРИБЛИЖЁННОЕ РЕШЕНИЕ СИСТЕМ НЕЛИНЕЙНЫХ УРАВНЕНИЙ

6.1 Постановка задачи

Рассмотрим нелинейную систему уравнений с действительными левыми частями:

$$\left.\begin{aligned} f_1(x_1, x_2, \ldots, x_n) &= 0, \\ f_2(x_1, x_2, \ldots, x_n) &= 0, \\ &\ldots\ldots\ldots\ldots\ldots \\ f_n(x_1, x_2, \ldots, x_n) &= 0, \end{aligned}\right\} \tag{6.1}$$

или в матричном виде:

$$f(x) = 0, \tag{6.2}$$

где

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \ldots \\ x_n \end{pmatrix}; \quad f = \begin{pmatrix} f_1 \\ f_2 \\ \ldots \\ f_n \end{pmatrix}.$$

Задача отыскания решения системы нелинейных уравнений с n неизвестными является существенно более сложной, чем задача отыскания решения уравнения с одним неизвестным. Однако на практике она встречается значительно чаще, так как в реальных исследованиях интерес представляет, как правило, определение не одного, а нескольких параметров.

Найти точное решение системы $\xi = (\xi_1, \xi_2, \ldots, \xi_n)^T$, удовлетворяющее уравнениям (6.1), практически невозможно [1, 3]. В отличие от случая решения СЛАУ использование прямых методов здесь исключается. Единственный путь решения системы (6.1) состоит в использовании итерационных методов для получения приближенного решения $x = (x_1, \ldots, x_n)^T$, удовлетворяющего при заданном $\varepsilon > 0$ неравенству $\|x - \xi\| < \varepsilon$.

Прежде чем перейти к изучению методов решения системы (6.1), подчеркнем, что задача может вообще не иметь решения, а если решения существуют, их число может быть произвольным.

Пример 6.1

Рассмотрим систему уравнений [1]:

$$4x_1^2 + x_2^2 = 4,$$
$$x_1 - x_2^2 + t = 0.$$

Здесь x_1, x_2 — неизвестные; t — параметр.

Первое уравнение задает на плоскости x_1Ox_2 эллипс, второе уравнение — параболу. Координаты точек пересечения этих кривых дают решения системы. Для диапазона изменения параметра $t \in [-2, 2]$ возможны следующие ситуации (см. рис. 6.1):

а) $t = -2$, решений нет;

б) $t = -1$, одно решение;

в) $t = 0$, два решения;

г) $t = 1$, три решения;

д) $t = 2$, четыре решения.

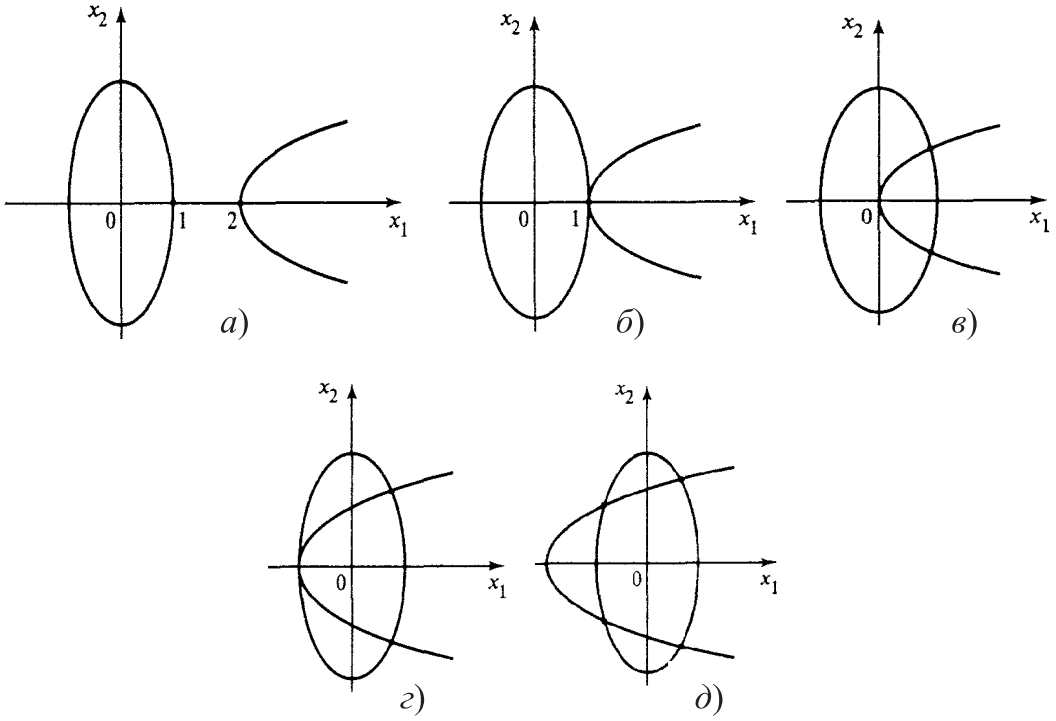


Рис. 6.1

Предположим, что функции $f_i(x)$ непрерывно дифференцируемы в некоторой окрестности решения ξ.

Введем для системы функций $f_1(x), f_2(x), \ldots, f_n(x)$ ***матрицу Якоби*** $f'(x) = W(x)$ [3]:

$$W(x) = \begin{pmatrix} \dfrac{\partial f_1}{\partial x_1} & \dfrac{\partial f_1}{\partial x_2} & \cdots & \dfrac{\partial f_1}{\partial x_n} \\ \dfrac{\partial f_2}{\partial x_1} & \dfrac{\partial f_2}{\partial x_2} & \cdots & \dfrac{\partial f_2}{\partial x_n} \\ \cdots & \cdots & \cdots & \cdots \\ \dfrac{\partial f_n}{\partial x_1} & \dfrac{\partial f_n}{\partial x_2} & \cdots & \dfrac{\partial f_n}{\partial x_n} \end{pmatrix}. \qquad (6.3)$$

6.2 Локализация корней

Как и для уравнения с одним неизвестным, отыскание решений начинают с этапа локализации [1]. Для каждого из искомых решений ξ указывают множество, содержащее только одно это решение и расположенное в достаточно малой его окрестности. Часто в качестве такого множества выступают параллелепипед или шар в n-мерном пространстве.

Иногда этап локализации не вызывает затруднений; соответствующие множества могут быть заданными, определяться из физических соображений, из смысла параметров x_i либо быть известными из опыта решения подобных задач. Однако чаще всего задача локализации (в особенности при больших n) представляет собой сложную проблему, от успешного решения которой в основном и зависит возможность вычисления решений системы (6.1). На этапе локализации особое значение приобретают квалификация исследователя, понимание им существа решаемой научной или инженерной проблемы, опыт решения этой или близких задач на компьютере. Во многих случаях полное решение задачи локализации невозможно и ее можно считать решенной удовлетворительно, если для ξ удается найти хорошее начальное приближение $x^{(0)}$.

Корректность и обусловленность задачи

Будем считать, что система (6.1) имеет решение ξ, причем в некоторой окрестности этого решения матрица Якоби $W(x)$ не вырождена [1]. Выполнение этого условия гарантирует, что в указанной окрестности нет других решений системы (6.1). Случай, когда в точке ξ матрица $W(x)$ вырождена, является существенно более трудным и в дальнейшем рассматриваться не будет. В одномерном случае первая ситуация отвечает наличию простого корня уравнения $f(x) = 0$, а вторая — кратного корня.

Ранее было показано, что погрешность вычисления функции f приводит к образованию вокруг корня уравнения $f(x) = 0$ интервала неопределенности, внутри которого невозможно определить, какая из точек является решением уравнения.

Аналогично, погрешности вычисления вектор-функции f приводят к появлению *области неопределенности D*, содержащей решение ξ системы (6.1), такой, что для всех $x \in D$ векторное уравнение $f(x) = 0$ удовлетворяется с точностью до погрешности. Область D может иметь сложную геометрическую форму (см. рис. 6.2). Мы будем рассматривать лишь радиус $\overline{\varepsilon}$ области D.

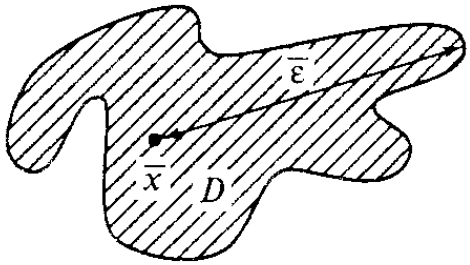


Рис. 6.2

Предположим, что для близких к корню ξ вычисляемые значения $f^*(x)$ удовлетворяют неравенству $\|f(x) - f^*(x)\| \leqslant \Delta(f)$, где $\Delta(f)$ — граница абсолютной погрешности. Тогда $\overline{\varepsilon}$ можно оценить с помощью неравенства $\overline{\varepsilon} \leqslant \left\| \left(W(\xi)\right)^{-1} \right\| \cdot \Delta(f)$.

Таким образом, в рассматриваемой задаче роль ***абсолютного числа обусловленности*** *играет норма обратной матрицы Якоби* $\left(W(\xi)\right)^{-1}$.

6.3 Метод Ньютона

Рассмотрим нелинейную систему уравнений (6.1). Пусть найдено p-е приближение:

$$x^{(p)} = \left(x_1^{(p)}, x_2^{(p)}, \ldots, x_n^{(p)}\right)$$

одного из изолированных корней $x = (x_1, \ldots, x_n)$ векторного уравнения (6.1). Тогда следующее приближение x^{p+1} можно представить в виде [1, 3, 5–9]:

$$x^{(p+1)} = x^{(p)} + e^{(p)}, \tag{6.4}$$

где $e^{(p)} = \left(e_1^{(p)}, \ldots, e_n^{(p)}\right)^T$ — поправка.

Подставляя в (6.1), будем иметь:

$$f\left(x^p + e^p\right) = 0. \tag{6.5}$$

Предположим, что $f(x)$ непрерывно дифференцируема. Разложим (6.5) в ряд по степеням e^p, ограничиваясь линейным приближением:

$$f\left(x^{(p)} + e^{(p)}\right) = f\left(x^{(p)}\right) + f'\left(x^{(p)}\right) e^{(p)} = 0. \tag{6.6}$$

Здесь $f'(x) = W(x)$ — матрица Якоби.

Система (6.6) является линейной относительно $e^{(p)}$. Из (6.6) получим:

$$e^p = -W^{-1}(x^p) f(x^p)$$

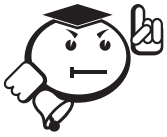
или с учётом (6.4):

$$x^{(p+1)} = x^{(p)} - W^{-1}\left(x^{(p)}\right) f\left(x^{(p)}\right). \tag{6.7}$$

Критерий завершения итерационного процесса имеет вид:

$$\left\| x^{(p+1)} - x^{(p)} \right\| < \varepsilon,$$

где ε — заданная точность.



Чтобы не вычислять обратную матрицу, обычно вместо (6.7) решают эквивалентную систему линейных алгебраических уравнений (6.6):

$$f'\left(x^{(p)}\right) e^{(p)} = -f\left(x^{(p)}\right) \tag{6.8}$$

относительно поправки $e^{(p)}$. Затем полагают

$$x^{(p+1)} = x^{(p)} + e^{(p)}. \tag{6.9}$$

Сходимость итерационного процесса обеспечивается следующей теоремой [1].

Теорема 6.1. Пусть в некоторой окрестности решения ξ системы (6.1) функции f_i $(i = 1, 2, \ldots, n)$ дважды непрерывно дифференцируемы и матрица Якоби $f'(\xi)$ невырождена. Тогда найдется такая малая δ-окрестность решения ξ, что при произвольном выборе начального приближения $x^{(0)}$ из этой окрестности итерационная последовательность метода Ньютона не выходит за пределы окрестности и справедлива оценка:

$$\left\| x^{(n+1)} - \xi \right\| \leqslant \delta^{-1} \left\| x^{(n)} - \xi \right\|^2, \; n \geqslant 0.$$

Эта оценка означает, что метод сходится с квадратичной сходимостью.

Квадратичная скорость сходимости метода Ньютона позволяет использовать простой критерий окончания итераций

$$\left\| x^{(n)} - x^{(n-1)} \right\| < \varepsilon. \tag{6.10}$$

Пример 6.2

Приближённо найти положительное решение системы уравнений:

$$\left.\begin{aligned} f_1(x_1, x_2) &= x_1 + 3\lg x_1 - x_2^2 = 0, \\ f_2(x_1, x_2) &= 2x_1^2 - x_1 x_2 - 5x_1 + 1 = 0. \end{aligned}\right\}$$

Решение:

Кривые f_1, f_2 пересекаются приблизительно в точках $M_1\,(1.4; -1.5)$, $M_2\,(3.4; 2.2)$. За начальное приближение берём:

$$x^{(0)} = \begin{pmatrix} 3.4 \\ 2.2 \end{pmatrix}.$$

Вычислим первое приближение корней с точностью до 4-х знаков.
Имеем:

$$f\left(x^{(0)}\right) = \begin{bmatrix} 3.4 + 3\lg 3.4 - 2.2^2 \\ 2 \cdot 3.4^2 - 3.4 \cdot 2.2 - 5 \cdot 3.4 + 1 \end{bmatrix} = \begin{bmatrix} 0.01544 \\ -0.3600 \end{bmatrix}.$$

Вычислим матрицу Якоби:

$$W(x) = \begin{pmatrix} 1 + \dfrac{3M}{x_1} & -2x_2 \\ 4x_1 - x_2 - 5 & -x_1 \end{pmatrix}, \text{ где } M = 0.43429 = \frac{1}{\ln 10}.$$

$$W\left(x^{(0)}\right) = \begin{pmatrix} 1 + \dfrac{3 \cdot .43429}{3.4} & -2 \cdot 2.2 \\ 4 \cdot 3.4 - 2.2 - 5 & -3.4 \end{pmatrix} = \begin{pmatrix} 1.3832 & -4.4 \\ 6.4 & -3.4 \end{pmatrix},$$

$$\Delta = \det W\left(x^{(0)}\right) = 23.4571 \neq 0,$$

$$W_0^{-1} = \frac{1}{\Delta}\begin{pmatrix} -3.4 & 4.4 \\ -6.4 & 1.3832 \end{pmatrix},$$

$$x^{(1)} = x^{(0)} - W_0^{-1} f\left(x^{(0)}\right) = \begin{pmatrix} 3.4 \\ 2.2 \end{pmatrix} - \frac{1}{23.4571}\begin{vmatrix} -3.4 & 4.4 \\ -6.4 & 1.3832 \end{vmatrix}\begin{pmatrix} .1544 \\ -.360 \end{pmatrix} = \begin{pmatrix} 3.4899 \\ 2.2633 \end{pmatrix}.$$

Аналогично находим $x^{(2)}$, $x^{(3)}$. Результаты занесем в таблицу 6.1:

Таблица 6.1

i	x_1	$E_1 = \Delta x_1$	x_2	$E_2 = \Delta x_2$
0	3.4	.0899	2.2	.0633
1	3.4899	−.0008	2.2633	−.0012
2	3.4891	−.0016	2.2621	−.0005
3	3.4875		2.2616	

Остановимся на $x^{(3)}$, будем иметь:

$$x_1^* = 3.4875; \quad x_2^* = 2.2616; \quad f(x^*) = \begin{pmatrix} 0.0002 \\ 0.0000 \end{pmatrix}.$$

. . . . . .

6.3.1 Модифицированный метод Ньютона

Модифицированный метод Ньютона используют для ослабления требований к начальному приближению. Суть его состоит в следующем [1].

На p-ом шаге решают систему:

$$f'\left(x^{(p)}\right) e^{(p)} = -f\left(x^{(p)}\right) \tag{6.11}$$

относительно поправки $e^{(p)}$. Затем находят число α_p из решения задачи одномерной оптимизации:

$$F_p(\alpha_p) = \min_{\alpha} F_p(\alpha); \quad F_p(\alpha) = \left\| f\left(x^{(p)} + \alpha e^{(p)}\right) \right\|.$$

Следующее приближение вычисляют по формуле:

$$x^{(p+1)} = x^{(p)} + \alpha_p e^{(p)}. \tag{6.12}$$

6.4 Метод итераций

Пусть дана система нелинейных уравнений специального вида:

$$\left.\begin{aligned} x_1 &= \varphi_1(x_1, x_2, \ldots, x_n), \\ x_2 &= \varphi_2(x_1, x_2, \ldots, x_n), \\ &\ldots\ldots\ldots\ldots\ldots \\ x_n &= \varphi_n(x_1, x_2, \ldots, x_n) \end{aligned}\right\} \tag{6.13}$$

или в векторном виде

$$x = \varphi(x), \tag{6.14}$$

где φ_i — функции, которые действительны, определены и непрерывны в некоторой окрестности ω изолированного решения $x^* = \{x_1^*, \ldots, x_n^*\}$.

Для нахождения корня ξ уравнения используют метод итераций [1, 3, 5–9]:

$$x^{(p+1)} = \varphi\left(x^{(p)}\right), \; p = 0, 1, \ldots \tag{6.15}$$

При этом если процесс (6.15) сходится, то есть $\lim\limits_{p\to\infty} x^{(p)} = \xi$, то ξ обязательно является корнем (6.14). Действительно, взяв предел от левой и правой частей (6.14), получим:

$$\lim_{p\to\infty} x^{(p+1)} = \varphi\left(\lim_{p\to\infty} x^{(p)}\right), \text{ т. е. } \xi = \varphi(\xi).$$

Таким образом, ξ есть корень уравнения (6.16).

Если, сверх того, все приближения $x^{(p)}$ $(p = 0,\ 1,\ 2,\ \ldots)$ принадлежат области ω и x^* — единственный корень в ω, то очевидно, что найденное решение и будет являться корнем уравнения

$$\xi = x^*.$$

Пусть дана система

$$f(x) = 0. \tag{6.16}$$

Приведём её к виду (6.13) или (6.15). Для этого перепишем (6.16) в виде

$$x = x + \Lambda f(x),$$

где Λ — неособенная матрица (т. е. $\det \Lambda \neq 0$). Введя обозначение $x + \Lambda f(x) = \varphi(x)$, будем иметь:

$$x = \varphi(x). \tag{6.17}$$

Ниже будет показано, что процесс итераций для (6.17) быстро сходится, если норма $\|\varphi'(x)\| = \|E + \Lambda f'(x)\|$ меньше единицы. Здесь полагается $\Lambda = \text{const}$, т. е. не зависит от x.

Поэтому выбираем матрицу Λ таким образом, чтобы

$$\varphi'\left(x^{(0)}\right) = E + \Lambda f'\left(x^{(0)}\right) = 0.$$

Отсюда

$$\Lambda = -\left[f'\left(x^{(0)}\right)\right]^{-1} = -W^{-1}\left(x^{(0)}\right).$$

Это есть, в сущности, упрощенный метод Ньютона, применённый к системе (6.16).

Если $\det\left(f'\left(x^{(0)}\right)\right) = 0$, то следует выбрать другое начальное приближение $x^{(0)}$.

6.4.1 Достаточные условия сходимости процесса итераций

Пусть дана приведённая нелинейная система:

$$x = \varphi(x). \tag{6.18}$$

Предполагается, что вектор-функция $\varphi(x)$ определена и непрерывна вместе со своей производной $\varphi'(x) = \left(\dfrac{\partial \varphi_i}{\partial x_j}\right)_{i,j=1}^{n}$ в выпуклой ограниченной замкнутой области $\sigma \subset E_n$.

Определим нормы матрицы $\varphi'(x)$:

$$\|\varphi'(x)\|_i = \max_{x \in \sigma} \|\varphi'(x)\|_i,\ i = 1,\ 2,\ 3,$$

где $\|\varphi'(x)\|_i$ — норма матрицы $\varphi'(x)$, определённая при фиксированном векторе x.

Имеет место теорема [3].

Теорема 6.2. Пусть функция $\varphi(x)$ и $\varphi'(x)$ непрерывны в области σ, причём в σ выполнено условие:

$$\|\varphi'(x)\|_i = q < 1,\ i = 1, 2, 3,$$

где q — произвольная постоянная.

Если последовательные приближения

$$x^{(p+1)} = \varphi\left(x^{(p)}\right),\ p = 0, 1, \ldots \tag{6.19}$$

не выходят из области σ, то процесс итераций (6.19) сходится и предельный вектор

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

является в области σ единственным решением (6.14).

Доказательство. Рассмотрим разность $\left(x^* - x^{(p)}\right)$

$$x^* - x^{(p)} = \varphi(x^*) - \varphi\left(x^{(p-1)}\right) = \varphi'\left(x^{(p-1)}\right)\left(x^* - x^{(p-1)}\right).$$

Возьмём норму от обеих частей:

$$\|x^* - x^{(p)}\| \leqslant \left\|\varphi'\left(x^{(p-1)}\right)\right\| \cdot \|x^* - x^{(p-1)}\| \leqslant q \cdot \|x^* - x^{(p-1)}\|.$$

Продолжая цепочку неравенств, получим:

$$\|x^* - x^{(p)}\| \leqslant q^p \|x^* - x^{(0)}\|. \tag{6.20}$$

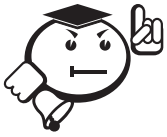
Так как, $q < 1$, то $q^p \underset{p\to\infty}{\longrightarrow} 0$, следовательно, последовательность $\{x^{(p)}\} \underset{p\to\infty}{\longrightarrow} x^*$. Переходя к пределу в (6.18), получим:

$$\lim_{p\to\infty} x^{(p+1)} = \lim_{p\to\infty} \varphi\left(x^{(p)}\right) = \varphi\left(\lim_{p\to\infty} x^{(p)}\right)$$

или

$$x^* = \varphi(x^*).$$

Теорема доказана.



Для погрешности корня ξ можно получить следующую формулу.

Имеем $x^{(n)} = \varphi\left(x^{(n-1)}\right)$. Далее $\xi - x^{(n)} = \xi - \varphi\left(x^{(n-1)}\right) = \varphi(\xi) - \varphi\left(x^{(n-1)}\right)$.

Теперь представим $\varphi\left(x^{(n-1)}\right) = \varphi\left(x^{(n)}\right) - \varphi'(c)\left(x^{(n)} - x^{(n-1)}\right)$, тогда получим:

$$\xi - x^{(n)} = \varphi(\xi) - \varphi\left(x^{(n)}\right) + \varphi'(c)\left(x^{(n)} - x^{(n-1)}\right).$$

Аналогичное представление функции $\varphi\left(x^{(n)}\right)$ имеет вид:

$$\varphi\left(x^{(n)}\right) = \varphi(\xi) - \varphi'(c_1)\left(\xi - x^{(n)}\right).$$

В результате получим:

$$\xi - x^{(n)} = \varphi'(c_1)\left(\xi - x^{(n)}\right) + \varphi'(c)\left(x^{(n)} - x^{(n-1)}\right).$$

Переходя к нормам, получим:

$$\|\xi - x^{(n)}\| \leqslant q\|\xi - x^{(n)}\| + q\|x^{(n)} - x^{(n-1)}\|.$$

Отсюда следует:

$$\|\xi - x^{(n)}\| \leqslant \frac{q}{1-q}\|x^{(n)} - x^{(n-1)}\|. \qquad (6.21)$$

. . . . . .

Из (6.21) следует критерий останова итерационного процесса:

$$\|x^{(p)} - x^{(p-1)}\| \leqslant \frac{1-q}{q}\varepsilon,$$

где ε — заданная точность.

Пример 6.3

Методом итерации приближённо решить систему:

$$\left.\begin{aligned} x_1^2 + x_2^2 &= 1, \\ x_1^3 - x_2 &= 0. \end{aligned}\right\}$$

Решение:

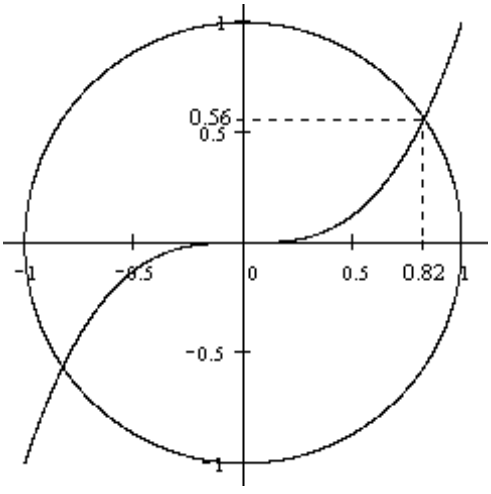


Рис. 6.3

Из графического построения (см. 6.3) видно, что имеется два решения, отличающиеся только знаком. Будем искать положительное решение.

Итак, $x^{(0)} = (0.9; 0.5)^T$

$$f(x) = \begin{pmatrix} x_1^2 + x_2^2 - 1 \\ x_1^3 - x_2 \end{pmatrix}; \quad f\left(x^{(0)}\right) = \begin{pmatrix} 0.060 \\ 0.229 \end{pmatrix}; \quad f'(x) = \begin{pmatrix} 2x_1 & 2x_2 \\ 3x_1^2 & -1 \end{pmatrix}.$$

Отсюда имеем $f'\left(x^{(0)}\right) = \begin{pmatrix} 1.8 & 1 \\ 2.43 & -1 \end{pmatrix} = W\left(x^{(0)}\right)$, $\det W\left(x^{(0)}\right) = -4.23 \neq 0$.

$$W^{-1}\left(x^{(0)}\right) = \left[f'\left(x^0\right)\right]^{-1} = -\frac{1}{4.23}\begin{pmatrix} -1 & -1 \\ -2.43 & 1.8 \end{pmatrix} = \Lambda.$$

Положим $\varphi(x) = x + \Lambda f(x)$, тогда исходная система будет эквивалентна стандартному матричному уравнению:

$$x = \varphi(x).$$

В результате получим:

$$x^{(1)} = \begin{pmatrix} x_1^{(0)} \\ x_2^{(0)} \end{pmatrix} - \frac{1}{4.23}\begin{pmatrix} 1 & 1 \\ 2.43 & -1.8 \end{pmatrix}\begin{pmatrix} 0.060 \\ 0.229 \end{pmatrix} = \begin{pmatrix} 0.9317 \\ 0.5630 \end{pmatrix},$$

$$x^{(2)} = \begin{pmatrix} x_1^{(1)} \\ x_2^{(1)} \end{pmatrix} - \frac{1}{4.23}\begin{pmatrix} 1 & 1 \\ 2.43 & -1.8 \end{pmatrix}\begin{pmatrix} \left(x_1^{(1)}\right)^2 + \left(x_2^{(1)}\right)^2 - 1 \\ \left(x_1^{(1)}\right)^3 - x_2^{(1)} \end{pmatrix} = \begin{pmatrix} 0.8268 \\ 0.5633 \end{pmatrix},$$

$$x^{(3)} = \begin{pmatrix} 0.8268 \\ 0.5633 \end{pmatrix} - \begin{pmatrix} 0.0007 \\ 0.0002 \end{pmatrix} = \begin{pmatrix} 0.8261 \\ 0.5631 \end{pmatrix},$$

$$x^{(4)} = \begin{pmatrix} 0.8261 \\ 0.5631 \end{pmatrix} - \begin{pmatrix} 0.0000 \\ -0.0005 \end{pmatrix} = \begin{pmatrix} 0.8261 \\ 0.5636 \end{pmatrix}.$$

Ограничиваясь 4-ым приближением, имеем корни

$$x_1 = 0.8261; \quad x_2 = 0.5636,$$

причём значения функции составляют

$$f(x) = \begin{pmatrix} 0.0000 \\ 0.0002 \end{pmatrix}$$

с точностью до 10^{-4}.

. . . . . .

Контрольные вопросы по главе 6

1. Для чего нужен этап локализации корней?
2. Что является числом обусловленности задачи решения системы нелинейных уравнений?
3. Запишите формулу итерационного процесса Ньютона решения системы нелинейных уравнений.
4. Сформулируйте условия сходимости метода Ньютона.
5. В чем состоит модификация метода Ньютона?
6. Сформулируйте достаточные условия сходимости метода итераций.
7. Как привести исходную систему к виду, необходимому для применения метода итераций?

Глава 7

ПРИБЛИЖЕНИЕ ФУНКЦИЙ

7.1 Постановка задачи

Вычисление значения функции $y = f(x)$ является одной из важных практических задач. Поэтому при решении на компьютере серьезных задач необходимо иметь быстрые и надежные алгоритмы вычисления значений используемых функций. Для элементарных и основных специальных функций такие алгоритмы разработаны и реализованы в виде стандартного математического обеспечения. Однако в расчетах нередко используются и другие функции, непосредственное вычисление которых затруднено либо приводит к слишком большим затратам машинного времени [1–3].

Приведем некоторые типичные ситуации:

- Пусть на сетке $\{x_i\}$ задана табличная функция $y_i = f(x_i)$, $i = 0, 1, \ldots, n$. Требуется найти значения функции $f(x)$ в точках x_j, не совпадающих с узлами исходной сетки x_i.
- Пусть на ЭВМ приходится многократно вычислять одну и ту же сложную функцию $f(x)$ в различных точках. Вместо ее непосредственного вычисления целесообразно вычислить ее значение в отдельных точках x_i, выбранных по какому-либо правилу, а в других точках x_j вычислять значение функции по каким-либо простым формулам, используя информацию об этих известных значениях. Эти простые формулы могут быть получены с помощью замены $f(x)$ другой функцией $g(x)$, имеющей более простой вид, чем функция $f(x)$.

- При заданном значении x значение $f(x)$ может быть найдено из эксперимента. В этом случае эти данные представляют собой таблицу $y_i = f(x_i)$, $i = 0, 1, \ldots, n$, при этом табличные значения y_i^* отличаются от «истинных» значений y_i, так как содержат ошибки эксперимента.

. . . . . .

. . . . . .

*Под **приближением функции** $f(x)$, заданной на интервале $[a, b]$, будем понимать замену $f(x)$ некоторой другой функцией $g(x)$, близкой к исходной функции $f(x)$* [1].

. . . . . .

В качестве класса G аппроксимирующих функций используют параметрическое семейство функций вида $y = g(x, a_0, a_1, \ldots, a_m)$. Выбор конкретной функции g осуществляется путем выбора параметров a_0, a_1, …, a_m.

Часто приближающую функцию $g(x)$ задают в форме обобщенного многочлена $\Phi_m(x)$ степени m:

$$\Phi_m(x) = \sum_{i=0}^{m} a_i \varphi_i(x), \tag{7.1}$$

где $\{\varphi_i(x)\}$ — система базисных функций, заданных на $[a, b]$ и являющихся гладкими (непрерывно дифференцируемыми); a_i — коэффициенты, которые выбирают таким образом, чтобы *отклонение* $f(x)$ от $\Phi_m(x)$ было минимальным на заданном множестве $X = \{x\}$.

Многочлен $\Phi_m(x)$ при этом называют *аппроксимирующим* или приближающим (от слова *approxio* — приближать).

Если в качестве базисных функций берутся степенные функции $\varphi_k(x) = x^k$, $k = 0, 1, 2, \ldots, m$, то имеем задачу приближения алгебраическими полиномами вида

$$P_m(x) = \sum_{k=0}^{m} a_k x^k. \tag{7.2}$$

Для аппроксимации периодических на отрезке $[0, 1]$ функций используют тригонометрические базисные функции вида $\varphi_0(x) = 1$, $\varphi_1(x) = \cos 2\pi x$, $\varphi_2(x) = \sin 2\pi x$, $\varphi_3(x) = \cos 4\pi x$, $\varphi_4(x) = \sin 4\pi x$, …. В этом случае мы имеем тригонометрический полином

$$S_m(x) = \frac{1}{2}a_0 + \sum_{k=1}^{m}(a_k \cos 2\pi kx + b_k \sin 2\pi kx). \tag{7.3}$$

Применим формулу Эйлера $\exp(iy) = \cos y + i \cdot \sin y$. Тогда полином (7.3) примет вид:

$$S_m(x) = \sum_{k=-m}^{m} c_k \exp(2\pi ikx), \tag{7.4}$$

где $c_k = c_{-k}^* = \frac{1}{2}(a_k - ib_k)$. Здесь a_k, b_k — действительные числа, а c_k — вообще говоря, комплексные числа; c_{-k}^* — комплексно сопряженное число.

Отсюда имеем:

$$c_k + c_{-k} = \frac{1}{2}\left[(a_k - ib_k) + (a_k + ib_k)\right] = a_k,\ k = 1, 2, \ldots, m;$$

$$i \cdot (c_k - c_{-k}) = \frac{1}{2}i\left[(a_k - ib_k) - (a_k + ib_k)\right] = b_k,\ k = 1, 2, \ldots, m.$$

Выбор класса G аппроксимирующих функций осуществляется с учетом того, насколько хорошо может быть приближена функция $f(x)$ функциями из этого класса.

. . . . . .

Если параметры a_i, $i = 0, 1, \ldots, n$ определяются из условия совпадения значений $f(x)$ и $g(x) \in G$ в узлах сетки x_i

$$g(x_i) = y_i, \tag{7.5}$$

то такой способ приближения называют ***интерполяцией*** *или интерполированием, а сетку $\{x_i\}$ называют интерполяционной сеткой* [1, 3]. *При этом полагается, что значение сетки $\{x_j\}$, в которой мы хотим вычислять $g(x_j)$, не выходят за пределы границ сетки $\{x_i\}$ $(a \leqslant x_j \leqslant b;\ j = 1, 2, \ldots, m)$. Если мы хотим вычислить значение $g(x_j)$ в точках $x_j \notin [a, b]$, то приближение называют* ***экстраполяцией****.*

. . . . . .

. . . . . .

Если множество X состоит из отдельных точек $x_0, x_1, \ldots, x_n$, то приближение называют ***точечным****. Если же X есть отрезок $a \leqslant x \leqslant b$, то приближение называют* ***интегральным*** [3].

. . . . . .

7.2 Интерполяция обобщенными многочленами

Назовем обобщенный многочлен $\Phi_m(x)$ интерполяционным, если он удовлетворяет системе алгебраических уравнений [1]:

$$\begin{aligned}
&\varphi_0(x_0)a_0 + \varphi_1(x_0)a_1 + \ldots + \varphi_m(x_0)a_m = y_0,\\
&\varphi_0(x_1)a_0 + \varphi_1(x_1)a_1 + \ldots + \varphi_m(x_1)a_m = y_1,\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&\varphi_0(x_n)a_0 + \varphi_1(x_n)a_1 + \ldots + \varphi_m(x_n)a_m = y_n
\end{aligned} \tag{7.6}$$

относительно коэффициентов $a_0, a_1, \ldots, a_m$.

В матричном виде систему (7.6) можно записать в виде:

$$Pa = y, \tag{7.7}$$

где

$$P = \begin{pmatrix} \varphi_0(x_0) & \varphi_1(x_0) & \dots & \varphi_m(x_0) \\ \varphi_0(x_1) & \varphi_1(x_1) & \dots & \varphi_m(x_1) \\ \dots & \dots & \dots & \dots \\ \varphi_0(x_n) & \varphi_1(x_n) & \dots & \varphi_m(x_n) \end{pmatrix}, \quad a = \begin{pmatrix} a_0 \\ a_1 \\ \dots \\ a_m \end{pmatrix}, \quad y = \begin{pmatrix} y_0 \\ y_1 \\ \dots \\ y_n \end{pmatrix}. \tag{7.8}$$

Введем векторы $\varphi_j = \left(\varphi_j(x_0), \varphi_j(x_1), \dots \varphi_j(x_n)\right)^T$, $j = 0, 1, \dots, m$. Система функций $\varphi_0, \varphi_1, \dots, \varphi_m$ называется *линейно зависимой в точках* $x_0, x_1, \dots, x_n$, если один из векторов φ_j системы $\varphi_0, \varphi_1, \dots, \varphi_m$ может быть представлен в виде линейной комбинации остальных векторов:

$$\varphi_j = \sum_{\substack{k=0 \\ k \neq j}}^{m} \alpha_k \varphi_k. \tag{7.9}$$

В противном случае систему функций $\varphi_0, \varphi_1, \dots, \varphi_m$ называют *линейно независимой в точках* $x_0, x_1, \dots, x_n$.

Пример 7.1

Покажем, что при $m \leqslant n$ система функций $1, x, x^2, \dots, x^m$ линейно независима в точках $x_0, x_1, \dots, x_n$, если они попарно различны [1].

Допустим обратное. Тогда справедливо равенство (7.9), которое в данном случае при $\varphi_k = \left(x_0^k, x_1^k, \dots, x_n^k\right)^T$ принимает вид:

$$x_i^j = \sum_{\substack{k=0 \\ k \neq j}}^{m} \alpha_k x_i^k, \ (i = 0, 1, \dots, n). \tag{7.10}$$

Положим $\alpha_j = -1$. Тогда получим: $P_m(x) = \sum_{k=0}^{m} \alpha_k x^k$ степени m обращается в ноль в точках $x_0, x_1, \dots, x_n$, число которых равно $n + 1$, т. е. больше m. Из курса алгебры известно, что многочлен степени m, тождественно не равный нулю, не может иметь более m корней. Полученное противоречие доказывает линейную независимость рассматриваемой системы функций.

Введем *матрицу Грама системы функций* $\varphi_0, \varphi_1, \dots, \varphi_m$

$$\Gamma = P^T P = \begin{pmatrix} (\varphi_0, \varphi_0) & (\varphi_1, \varphi_0) & \dots & (\varphi_m, \varphi_0) \\ (\varphi_0, \varphi_1) & (\varphi_1, \varphi_1) & \dots & (\varphi_m, \varphi_1) \\ \dots & \dots & \dots & \dots \\ (\varphi_0, \varphi_m) & (\varphi_1, \varphi_m) & \dots & (\varphi_m, \varphi_m) \end{pmatrix}.$$

Справедлива следующая теорема [1].

Теорема 7.1. Если $m = n$, то решение задачи интерполяции обобщенным многочленом (7.1) существует и единственно при любом наборе данных $y_0, y_1, \ldots, y_n$ тогда и только тогда, когда система функций $\varphi_0, \varphi_1, \ldots, \varphi_m$ линейно независима в точках $x_0, x_1, \ldots, x_n$.

При $m = n$ система (7.7) имеет квадратную матрицу. Из теоремы 7.1 следует, что в этом случае определитель матрицы P отличен от нуля: $\det P \neq 0$.

При $m < n$ решение системы (7.7) найдем методом наименьших квадратов, т. е. параметры a ищем из условия минимума функции $F(a_0, a_1, \ldots, a_m) = (r, r)$, где $r = Pa - y$ — вектор невязки. В результате получим систему уравнений

$$\Gamma a = d, \tag{7.11}$$

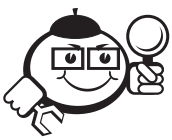
где

$$\Gamma = P^T P = \begin{pmatrix} (\varphi_0, \varphi_0) & (\varphi_1, \varphi_0) & \ldots & (\varphi_m, \varphi_0) \\ (\varphi_0, \varphi_1) & (\varphi_1, \varphi_1) & \ldots & (\varphi_m, \varphi_1) \\ \ldots & \ldots & \ldots & \ldots \\ (\varphi_0, \varphi_m) & (\varphi_1, \varphi_m) & \ldots & (\varphi_m, \varphi_m) \end{pmatrix} \tag{7.12}$$

— *матрица Грама системы функций* $\varphi_0, \varphi_1, \ldots, \varphi_m$; $d = P^T y$. Существование и единственность решения системы (7.11) следует из следующей теоремы [1].

Теорема 7.2. Если система функций $\varphi_0, \varphi_1, \ldots, \varphi_m$ линейно независима в точках $x_0, x_1, \ldots, x_n$, то определитель Грама $\det\Gamma$ отличен от нуля.

Система функций $\varphi_0, \varphi_1, \ldots, \varphi_m$ называется ортогональной на множестве точек $x_0, x_1, \ldots, x_n$, если $(\varphi_k, \varphi_j) = 0$ при $k \neq j$ и $(\varphi_k, \varphi_j) \neq 0$ при $k = j$ для всех k, $j = 0, 1, \ldots, m$. Для ортогональной системы функций матрица Грама диагональная, а определитель $\det\Gamma \neq 0$. Поэтому всякая ортогональная на множестве точек x_0, $x_1, \ldots, x_n$ система функций является линейно независимой.



Пример 7.2

Покажем, что система функций $\varphi_k(x) = \exp(2\pi i k x)$, $k = 0, 1, \ldots, N-1$ ортогональна на множестве точек $x_l = l/N$, $l = 0, 1, \ldots, N-1$, т. е.

$$(\varphi_k, \varphi_j) = N\delta_{kj}, \tag{7.13}$$

где $\delta_{kj} = \begin{cases} 1, & k = j, \\ 0, & k \neq j. \end{cases}$ Здесь i — мнимая единица.

В точке x_l мы имеем $\varphi_k(x_l) = \exp(2\pi ikl/N)$. Тогда для скалярного произведения комплексно сопряженных функций получим:

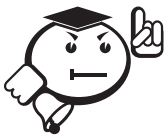
$$(\varphi_k, \varphi_j) = \sum_{l=0}^{N-1} \exp\bigl(2\pi il(k-j)/N\bigr). \tag{7.14}$$

При $k = j$ правая часть равенства (7.14) равна N. При $k \neq j$ просуммируем ряд (7.14), используя известную из курса элементарной математики формулу суммы членов геометрической прогрессии ($S_n = \alpha_1 \dfrac{1-q^n}{1-q}$, где α_1 — первый член геометрической прогрессии, q — знаменатель геометрической прогрессии). В результате получим:

$$(\varphi_k, \varphi_j) = \frac{1 - \exp\bigl(2\pi i(k-j)\bigr)}{1 - \exp\bigl(2\pi i(k-j)/N\bigr)} = 0,$$

что и требовалось доказать.

Здесь использовано равенство $\exp\bigl(2\pi i(k-j)\bigr) = 1$.



Для ортогональной системы функций φ_0, φ_1, …, φ_m на множестве точек x_0, x_1, …, x_n решение задачи интерполяции существенно упрощается. В этом случае, как указывалось выше, матрица системы (7.11) становится диагональной, а коэффициенты a_j вычисляются по формулам:

$$a_j = \frac{(y, \varphi_j)}{(\varphi_j, \varphi_j)},\ j = 0,\ 1,\ \ldots,\ m, \tag{7.15}$$

где $y = (y_0, y_1, \ldots, y_n)^T$, $y_i = f(x_i)$, $i = 0,\ 1,\ \ldots,\ n$.

7.3 Полиномиальная интерполяция. Многочлен Лагранжа

Интерполяционный многочлен

Пусть функция f задана таблицей своих значений $y_i = f(x_i)$, $i = 0,\ 1,\ \ldots,\ n$.

Многочлен $P_n(x) = \sum_{k=0}^{n} a_k x^k$ степени n называется ***интерполяционным многочленом****, если он удовлетворяет условиям:*

$$P(x_i) = y_i,\ i = 0,\ 1,\ \ldots,\ n. \tag{7.16}$$

Многочлен Лагранжа

Полином Лагранжа имеет вид [1, 3, 9]:

$$L_n(x) = \sum_{j=0}^{n} y_j p_{nj}(x), \tag{7.17}$$

где

$$p_{nj}(x) = \prod_{\substack{k=0 \\ k \neq j}}^{n} \frac{x - x_k}{x_j - x_k} = \frac{(x - x_0)(x - x_1)\ldots(x - x_{j-1})(x - x_{j+1})\ldots(x - x_n)}{(x_j - x_0)(x_j - x_1)\ldots(x_j - x_{j-1})(x_j - x_{j+1})\ldots(x_j - x_n)}. \tag{7.18}$$

Можно увидеть, что $p_{nj}(x)$ представляет собой многочлен степени n, удовлетворяющий условию

$$p_{nj}(x_i) = \begin{cases} 1 & \text{при } i = j \\ 0 & \text{при } i \neq j \end{cases}$$

На рис. 7.1 полином $L_n(x)$ совпадает в узлах сетки со значениями функции $f(x)$ ($L_n(x_i) = y_i,\ i = 0,\ 1,\ \ldots$). В качестве примера на рис. 7.2 показан полином $p_{n2}(x)$.

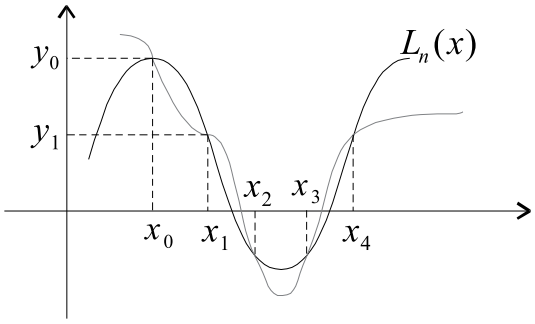


Рис. 7.1

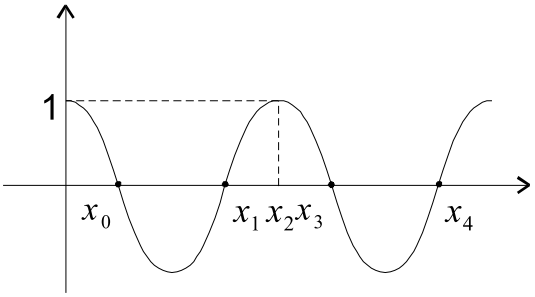
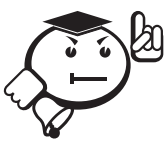


Рис. 7.2



Замечание 1. Формулу (7.18) можно получить следующим образом. Представим полином $p_{ni}(x)$ в форме

$$p_{ni}(x) = C_i (x - x_0)(x - x_1)\ldots(x - x_{i-1})(x - x_{i+1})\ldots(x - x_n). \tag{7.19}$$

Полагая $x = x_i$, получим:

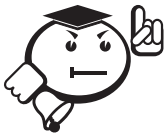
$$C_i = \frac{1}{(x_i - x_0)(x_i - x_1)\ldots(x_i - x_{i-1})(x_i - x_{i+1})\ldots(x_i - x_n)}. \qquad (7.20)$$

Подставив C_i в (7.18), приходим к выражению

$$p_{ni}(x) = \frac{(x - x_0)\ldots(x - x_{i-1})(x - x_{i+1})\ldots(x - x_n)}{(x_i - x_0)\ldots(x_i - x_{i-1})(x_i - x_{i+1})\ldots(x_i - x_n)}.$$

. . . . . .

. . . . . .



Замечание 2. Запись интерполяционного полинома в форме Лагранжа можно рассматривать как его запись в виде обобщенного многочлена (7.1) по системе функций $\varphi_k(x) = p_{nk}(x)$, $k = 0, 1, \ldots, n$.

. . . . . .

На практике наиболее часто используется интерполяция первой, второй и третьей степени (линейная, квадратичная и кубическая интерполяции). Полиномы первой и второй степени имеют вид:

$$L_1(x) = y_0\frac{x - x_1}{x_0 - x_1} + y_1\frac{x - x_0}{x_1 - x_0},$$

$$L_2(x) = y_0\frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} + y_1\frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} + y_2\frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}.$$

7.4 Погрешность интерполяции

Введем остаточный член $R_n(x) = f(x) - P_n(x)$, где $P_n(x)$ — интерполяционный полином степени n. Справедлива следующая теорема [3].

. . . . . .

Теорема 7.3. Пусть функция f дифференцируема $n + 1$ раз на отрезке $[a, b]$, содержащем узлы интерполяции x_i, $i = 0, 1, \ldots, n$. Тогда для погрешности интерполяции в точке $x \in [a, b]$ справедливо неравенство

$$|R_n(x)| \leqslant \frac{M_{n+1}}{(n+1)!}|\omega_{n+1}(x)|, \qquad (7.21)$$

в котором $\omega_{n+1}(x) = \prod_{i=0}^{n}(x - x_i) = (x - x_0)(x - x_1)\ldots(x - x_n)$, $\xi \in (a, b)$, $M_{n+1} = \max\limits_{a \leqslant x \leqslant b}|f^{(n+1)}(x)|$.

Максимальная погрешность интерполяции на отрезке $[a, b]$ равна

$$\max_{a \leqslant x \leqslant b} \left| R_n(x) \right| \leqslant \frac{M_{n+1}}{(n+1)!} \max_{a \leqslant x \leqslant b} \left| \omega_{n+1}(x) \right|. \qquad (7.22)$$

. . . . . .

Пример 7.3

С какой точностью можно вычислить $\sqrt{115}$ *с помощью интерполяционной формулы Лагранжа для функции* $y = \sqrt{x}$*, выбрав узлы интерполирования* $x_0 = 100$*,* $x_1 = 121$*,* $x_2 = 144$?

Решение:

Имеем $y' = \frac{1}{2}x^{-1/2}$; $y'' = -\frac{1}{4}x^{-3/2}$; $y''' = \frac{3}{8}x^{-5/2}$.

Отсюда $M_3 = \max \left| y''' \right| = y'''(x_0) = \frac{3}{8}(100)^{-5/2} = \frac{3}{8}10^{-5}$.

На основании формулы (7.22) получим:

$$\left| R_2(x) \right| \leqslant \frac{3}{8}10^{-5}\frac{1}{3!}\left| (115-100)(115-121)(115-144) \right| \approx \frac{1}{16}10^{-5} \cdot 15 \cdot 6 \cdot 29 \approx 1.6 \cdot 10^{-3}.$$

. . . . . .

7.5 Минимизация оценки погрешности

Предположим, что значение заданной на отрезке $[a, b]$ функции f можно вычислить в произвольных точках x_k. Пусть необходимо построить интерполяционный полином $P_n(x)$. Для этого необходимо получить таблицу значений функции f в выбранных на отрезке $[a, b]$ точках x_0, x_1, …, x_n. Естественное желание выбрать такие узлы интерполяции, чтобы максимальная погрешность интерполяции, вычисляемая по формуле (7.22) $\max\limits_{a \leqslant x \leqslant b} \left| R_n(x) \right| = \max\limits_{a \leqslant x \leqslant b} \left| f(x) - P_n(x) \right|$, была бы минимальной. Уменьшить ее можно лишь за счет уменьшения величины $\max\limits_{a \leqslant x \leqslant b} \left| \omega_{n+1}(x) \right|$. Показано (см. [1, 3]), что минимальная погрешность интерполяции достигается на сетке

$$x_k = \frac{a+b}{2} + \frac{b-a}{2}t_k = \frac{a+b}{2} + \frac{b-a}{2}\cos\left(\frac{2k+1}{2n}\pi\right), \quad k = 0,\ 1, \ldots,\ n, \qquad (7.23)$$

где $t_k = \cos\dfrac{(2k+1)\pi}{2n}$ — узлы полиномов Чебышева $(k = 0,\ 1,\ \ldots,\ n)$.

При таком выборе узлов для максимальной погрешности интерполяции мы получим соотношение

$$\max_{a \leqslant x \leqslant b} \left| R_n(x) \right| \leqslant \frac{M_{n+1}}{(n+1)!2^n}\left[\frac{b-a}{2}\right]^{n+1}. \qquad (7.24)$$

Узлы, определяемые формулой (7.23), не являются равноотстоящими, а сгущаются около концов отрезка.

7.6 Интерполяционная формула Ньютона для равномерной сетки

Конечные разности [1, 3]

Пусть функция f задана таблицей своих значений $y_i = f(x_i)$, $i = 0, 1, \ldots, n$ на равномерной сетке $x_i = a + i \cdot h$, $i = 0, 1, \ldots, n$, где $h = x_i - x_{i-1}$ — *шаг сетки*. Величину h называют также *шагом таблицы*, а узлы *равноотстоящими*. Величину $\Delta y_i = y_{i+1} - y_i$ называют *конечной разностью первого порядка* функции $y = f(x)$ в точке x_i. *Конечная разность второго порядка* вычисляется по формуле $\Delta^2 y_i = \Delta y_{i+1} - \Delta y_i$. Конечная разность k-го порядка определяется общей формулой:

$$\Delta^k y_i = \Delta^{k-1} y_{i+1} - \Delta^{k-1} y_i, \; k \geqslant 1, \; \Delta^0 y_i = y_i.$$

Пример 7.4

В табл. 7.1 даны значения функции для равномерной сетки.

Таблица 7.1

x	0	1	2	3	4
y	1	3	6	8	11

Составим таблицу конечных разностей (табл. 7.2):

Таблица 7.2

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$
0	1	2	1	−2	4
1	3	3	−1	2	
2	6	2	1		
3	8	3			
4	11				

Конечная разность k-го порядка может быть выражена через значения функции в $k + 1$ точках [1, 3]

$$\Delta^k y_i = \sum_{l=0}^{k} (-1)^{k-l} C_k^l y_{i+l}, \tag{7.25}$$

где $C_k^l = \dfrac{k!}{l!(k-l)!}$ — биномиальные коэффициенты. В частности,

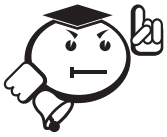
$$\Delta^2 y_i = y_{i+2} - 2y_{i+1} + y_i, \quad \Delta^3 y_i = y_{i+3} - 3y_{i+2} + 3y_{i+1} - y_i,$$
$$\Delta^4 y_i = y_{i+4} - 4y_{i+3} + 6y_{i+2} - 4y_{i+1} + y_i.$$

. . . . . .

Теорема 7.4 [1]. Пусть функция f дифференцируема k раз на отрезке $[x_i, x_{i+k}]$. Тогда справедливо равенство

$$\Delta^k y_i = h^k f^{(k)}(\xi), \ \xi \in (x_i, x_{i+k}). \tag{7.26}$$

. . . . . .



. . . . . .

На основе этой теоремы на практике часто используют следующую формулу численного дифференцирования функции $f(x)$ на интервале $x \in [x_i, x_{i+k}]$:

$$f^{(k)}(x) = \frac{\Delta^k y_i}{h^k}. \tag{7.27}$$

. . . . . .

Формула (7.27) хорошо воспроизводит производную k-го порядка, если она слабо меняется на интервале $[x_i, x_{i+k}]$.

В реальных вычислениях таблица конечных разностей $\Delta^k y_i$ строится по экспериментальным значениям $\tilde{y}_j$, каждое из которых содержит погрешность $\varepsilon_j = \tilde{y}_j - y_j$. Тогда найденные значения $\Delta^k \tilde{y}_j$ содержат неустранимые ошибки [1]:

$$\varepsilon_i^{(k)} = \Delta^k \tilde{y}_i - \Delta^k y_i = \sum_{l=0}^{k} (-1)^{k-l} C_k^l \varepsilon_{i+l}. \tag{7.28}$$

Если ошибки ограничены, т. е. $|\varepsilon_i| \leqslant \varepsilon$ для всех i, то получим:

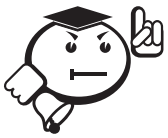
$$\left|\varepsilon_i^{(k)}\right| \leqslant \sum_{l=0}^{k} C_k^l \varepsilon = 2^k \varepsilon. \tag{7.29}$$

Таким образом, с ростом порядка k конечной разности ошибка возрастает с коэффициентом 2^k.

Использование статистического анализа позволяет получить более оптимистическую оценку погрешности конечной разности k-го порядка [1]

$$\sigma^{(k)} = \sqrt{C_{2k}^k}\,\sigma, \tag{7.30}$$

где σ^2 — дисперсия ошибки исходных данных y_j^*; C_{2k}^k — биномиальный коэффициент. Так как $\sqrt{C_{2k}^k} < 2^k$, то мы имеем более реалистическую оценку погрешности.



Иногда вместо конечных разностей вперед $\Delta^k y_i$ используют *конечные разности назад*, определяемые рекуррентной формулой:

$$\nabla^k y_i = \nabla^{k-1} y_i - \nabla^{k-1} y_{i-1},\ k \geqslant 1,\ \nabla^0 y_i = y_i.$$

В частности, для производной в т. $x_n = b$ на правой границе интервала имеем

$$\nabla^k y_n = \nabla^{k-1} y_n - \nabla^{k-1} y_{n-1}.$$

Разности вперед и назад связаны равенством $\Delta^k y_i = \nabla^k y_{i+k}$.

Полином Ньютона для равномерной сетки

Пусть для функции $f(x) = y$ заданы значения $y_i = f(x_i)$ в равноотстоящих узлах $x_i = x_0 + ih$, $i = 0, 1, \ldots, n$, h — шаг интерполяции. Требуется подобрать полином $P_n(x)$ степени не выше n, принимающий в узлах x_i значения

$$P_n(x_i) = y_i,\ (i = 0, 1, \ldots, n). \tag{7.31}$$

Будем искать полином в виде:

$$\begin{aligned} P_n(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + a_3(x - x_0)(x - x_1)(x - x_2) + \ldots \\ + a_n(x - x_0)(x - x_1)\ldots(x - x_{n-1}). \end{aligned} \tag{7.32}$$

Наша задача — найти коэффициенты a_i $(i = 0,\ 1,\ \ldots,\ n)$ полинома $P_n(x)$. Полагая в (7.32) $x = x_0$, получим:

$$a_0 = P_n(x_0) = y_0 = f(x_0),\ \text{т. е. } a_0 = y_0.$$

Чтобы найти a_1, составим первую конечную разность:

$$\begin{aligned} \Delta P_n(x) = P_n(x + h) - P_n(x) = a_1 h + 2a_2(x - x_0)h + 3a_3(x - x_0)(x - x_1)h + \ldots \\ + na_n(x - x_0)(x - x_1)\cdots(x - x_{n-2})h. \end{aligned} \tag{7.33}$$

Полагая в (7.33) $x = x_0$, получим:

$$\Delta P_n(x_0) = \Delta y_0 = a_1 h,$$

откуда

$$a_1 = \frac{\Delta y_0}{1!h}.$$

Для определения a_2 составим конечную разность 2-го порядка:

$$\begin{aligned} \Delta^2 P_n(x) = \Delta\big(P_n(x + h) - P_n(x)\big) = 1 \cdot 2 \cdot h^2 a_2 + 2 \cdot 3h^2 a_3(x - x_0) + \ldots \\ + (n - 1)nh^2 a_n(x - x_0)(x - x_1)\cdots(x - x_{n-3}). \end{aligned}$$

Полагая $x = x_0$, получим:

$$\Delta^2 P_n(x_0) = \Delta^2 y_0 = 2!h^2 a_2,$$

откуда

$$a_2 = \frac{\Delta^2 y_0}{2!h^2}.$$

Продолжая этот процесс, получим:

$$a_i = \frac{\Delta^i y_0}{i!h^i},\ (i = 0, 1, \ldots, n), \tag{7.34}$$

где $\Delta^0 y_0 = y_0$; $0! = 1$.

. . . . . .

Подставляя найденные значения a_i в (7.32)*, получим* ***интерполяционный полином Ньютона****:*

$$P_n(x) = y_0 + \frac{\Delta y_0}{1!h}(x - x_0) + \frac{\Delta^2 y_0}{2!h^2}(x - x_0)(x - x_1) + \ldots + \frac{\Delta^n y_0}{n!h^n}(x - x_0)(x - x_1)\ldots(x - x_{n-1}). \tag{7.35}$$

. . . . . .

. . . . . .

Полином $P_n(x)$, определяемый (7.35), является полиномом n-ой степени. В узлах x_k значения $P_n(x_k) = y_k$ (это нетрудно проверить).

При $h \to 0$ полином (7.35) превращается в полином Тейлора (ряд Тейлора).

. . . . . .

Действительно,

$$\lim_{h\to 0} \frac{\Delta^k y_0}{h^k} = \left(\frac{d^k y}{dx^k}\right)_{x=x_0} = y^{(k)}(x_0).$$

Кроме того,

$$\lim_{h\to 0} (x - x_0)(x - x_1)\ldots(x - x_{n-1}) = (x - x_0)^n.$$

Поэтому при $h \to 0$ получим из (7.35):

$$P_n(x) = y(x_0) + y'(x_0)(x - x_0) + \ldots + \frac{1}{n!} y^{(n)}(x_0)(x - x_0)^n.$$

Для практического использования формулу (7.35) перепишем в следующем виде. Введем $q = \dfrac{x - x_0}{h}$ — число шагов, необходимых для достижения точки x, исходя из точки x_0. Тогда

$$\frac{1}{h^i}(x - x_0)(x - x_1)\ldots(x - x_{i-1}) =$$

$$= \frac{x - x_0}{h} \cdot \frac{x - x_0 - h}{h} \cdot \frac{x - x_0 - 2h}{h} \cdots \frac{\left[x - x_0 - (i-1)h\right]}{h} = q(q-1)(q-2)\ldots(q-i+1).$$

Подставляя это в (7.35), получим:

$$P_n(x) = P_n(x_0 + h \cdot q) = y_0 + \frac{\Delta y_0}{1!}q + \frac{\Delta^2 y_0}{2!}q(q-1) + \ldots + \frac{\Delta^n y_0}{n!}q(q-1)\ldots(q-n+1). \tag{7.36}$$

Полином (7.36) называют *первой интерполяционной формулой Ньютона или интерполяционным полиномом с конечными разностями для интерполяции вперед*.

Если $n = 1$, то $P_1(x) = y_0 + q\Delta y_0$ — формула линейного интерполирования.

При $n = 2$ $P_2(x) = y_0 + q\Delta y_0 + \frac{q(q-1)}{2}\Delta^2 y_0$ — формула квадратичного (параболического) интерполирования.

Используя ***конечные разности назад****, запишем полином для* ***интерполяции назад*** [1, 3]:

$$\overline{P}_n(x) = P_n(x_n + h \cdot t) = y_n + \frac{\nabla y_n}{1!}t + \frac{\nabla^2 y_n}{2!}t(t+1) + \ldots + \frac{\nabla^n y_0}{n!}t(t+1)\ldots(t+n-1). \tag{7.37}$$

Здесь $t = (x - x_n)/h$ *— число шагов, необходимых для достижения точки* x*, исходя из точки* x_n*;* $\nabla^k y_n = \Delta^k y_{n-k}$*. Формулу* (7.37) *называют* ***второй интерполяционной формулой Ньютона****.*

Пример 7.5

В табл. 7.3 даны значения функции.

Таблица 7.3

x	0	1	2	3	4
y	1	3	6	8	11

Составим таблицу разностей (табл. 7.4)

Таблица 7.4

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$
0	1	2	1	−2	4
1	3	3	−1	2	
2	6	2	1		
3	8	3			
4	11				

$$q = x - x_0 = x_0,$$

$$P_3(x) = 1 + 2x + \frac{1}{2}1 \cdot x \cdot (x-1) + \frac{1}{6} \cdot (-2) \cdot x(x-1)(x-2) + 4 \cdot \frac{1}{4!}x(x-1)(x-2)(x-3).$$

. . . . . .

7.7 Интерполяционная формула Ньютона для неравномерной сетки

Разделенные разности

Пусть функция $f(x)$ задана на таблице $x_0, x_1, \ldots, x_n$ значений аргумента с произвольным шагом.

Введем следующие понятия:

1. Отношения $f(x_i; x_{i+1}) = \dfrac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}$, $(i = 0,\ 1, \ldots)$ называются *разделенными разностями первого порядка* для табличной функции $f(x) = y$.
2. Отношения $f(x_i; x_{i+1}; x_{i+2}) = \dfrac{f(x_{i+1}; x_{i+2}) - f(x_i; x_{i+1})}{x_{i+2} - x_i}$ называются *разделенными разностями второго порядка*.
3. Для разделенных разностей n-го порядка имеем

$$f(x_i; x_{i+1}; \ldots; x_{i+n}) = \frac{f(x_{i+1}; \ldots; x_{i+n}) - f(x_i; x_{i+1}; \ldots; x_{i+n-1})}{x_{i+n} - x_i}.$$

. . . . .  **Пример 7.6** . . . . . .

Пусть задана табличная функция (первые два столбца табл. 7.5). Найдем разделенные разности.

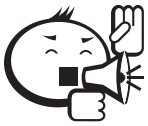
Таблица 7.5

x	y	1 порядка	2 порядка	3 порядка	4 порядка
0	0	5	10	$\dfrac{7.143 - 10}{0.9} = -3.174$	$\dfrac{1.323 + 3.174}{1.4} = 1.322$
0.2	1	10	$\dfrac{5}{0.7} = 7.143$	$\dfrac{5.555 - 7.143}{1.2} = -1.323$	$\dfrac{2.357 + 1.323}{1.8} = 2.044$
0.5	4	15	$\dfrac{5}{0.9} = 5.555$	$\dfrac{9.091 - 5.555}{1.5} = 2.357$	

продолжение на следующей странице

Таблица 7.5 – Продолжение

x	y	1 порядка	2 порядка	3 порядка	4 порядка
0.9	10	20	$\frac{10}{1.1} = 9.091$		
1.4	20	30			
2.0	38				



Разделенные разности обладают следующими свойствами:

1. Разделенная разность $f(x_i; x_{i+1}; \ldots; x_{i+k})$ является симметричной функцией своих аргументов x_i, x_{i+1}, …, x_{i+k} (т. е. ее значение не меняется при любой их перестановке).
2. Пусть функция f имеет на отрезке $[a, b]$, содержащем точки x_i, x_{i+1}, …, x_{i+k}, производную порядка k. Тогда справедливо равенство

$$f(x_i; x_{i+1}; \ldots; x_{i+k}) = \frac{f^{(k)}(\xi)}{k!}, \ \xi \in (a, b).$$

3. В случае когда таблица значений функции задана на равномерной сетке с шагом h, разделенная и конечная разности связаны равенством

$$f(x_i; x_{i+1}; \ldots; x_{i+k}) = \frac{\Delta^k y_i}{h^k k!}.$$

Интерполяционный многочлен Ньютона с разделенными разностями

Интерполяционная формула Ньютона для неравномерной сетки *имеет вид:*

$$P_n(x) = f(x_0) + f(x_0; x_1)(x - x_0) + f(x_0; x_1; x_2)(x - x_0)(x - x_1) + \ldots$$
$$+ f(x_0; x_1; \ldots; x_n)(x - x_0)(x - x_1)\ldots(x - x_{n-1}) =$$
$$= \sum_{k=0}^{n} f(x_0; x_1; \ldots; x_k)\omega_k(x). \qquad (7.38)$$

Пример 7.7

Составить интерполяционный полином для функции $y = f(x)$, заданной таблицей (табл. 7.6).

Таблица 7.6

x	0	1	4	6
y	1	3	12	20

Решение:

Разделенные разности вычисляем по общим формулам:

$$f(x_0) = 1; \quad f(x_0; x_1) = 2; \quad f(x_1; x_2) = 3; \quad f(x_2, x_3) = 4;$$

$$f(x_0; x_1; x_2) = \frac{3-2}{4} = \frac{1}{4}; \quad f(x_1; x_2; x_3) = \frac{4-3}{5} = \frac{1}{5};$$

$$f(x_0; x_1; x_2; x_3) = \frac{\frac{1}{5} - \frac{1}{4}}{6} = -\frac{1}{20 \cdot 6} = -\frac{1}{120}.$$

В результате получим: $P_3(x) = 1 + 2 \cdot x + \frac{1}{4}x(x-1) - \frac{1}{120}x(x-1)(x-4)$.

7.8 Чувствительность интерполяционного полинома к погрешностям входных данных

Кроме погрешности, которая возникает от приближенной замены функции f интерполяционным многочленом, возникает еще дополнительная погрешность, связанная с тем, что значения интерполируемой функции могут быть заданы с погрешностью.

Пусть вместо точных значений y_i заданы приближенные значения $\tilde{y}_i$ с погрешностью ε_i. Тогда вычисляемый по этим значениям многочлен $\tilde{P}_n = \sum_{j=0}^{n} \tilde{y}_j p_{nj}(x)$ содержит погрешность

$$P_n(x) - \tilde{P}_n(x) = \sum_{j=0}^{n} \varepsilon_j p_{nj}(x).$$

Обозначим максимальную погрешность $\varepsilon_m = \max_j \left(|\varepsilon_j|\right)$.

Тогда для верхней границы погрешности интерполяционного полинома справедлива оценка [1]

$$\max_{[a,b]} \left| P_n(x) - \tilde{P}_n(x) \right| = \Lambda_n \varepsilon_m,$$

где $\Lambda_n = \max\limits_{[a,b]} \left| p_{nj}(x) \right|$ *— величина, которую называют* ***константой Лебега****. Эта величина играет роль* ***абсолютного числа обусловленности****.*

Константа Лебега не зависит от длины отрезка $[a, b]$, а определяется только относительным расположением узлов на нем [1]. Минимальное значение $\Lambda_n \approx$ $\approx \dfrac{2}{\pi} \ln(n + 1) + 1$ достигается для чебышевской сетки, т. е. для сетки, состоящей из нулей полиномов Чебышева. Для равномерной сетки константа Лебега $\Lambda_n >$ $> \dfrac{2^{n-1}}{(2n - 1)\sqrt{n}}$ при $n \geqslant 4$ и погрешность интерполяции резко увеличивается с ростом n. Поэтому в вычислениях *не следует* использовать интерполяционные полиномы высокой степени с равномерными сетками.

7.9 Интерполяция с помощью «скользящего» полинома

Пусть функция задана следующей табл. 7.7.

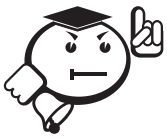
Таблица 7.7

i	0	1	2	3	4
x_i	0	1	2	3	4
y_i	1.0	1.8	2.2	1.4	1.0

Воспользуемся «скользящим» полиномом второй степени. Так, при $x \in [0.0, 1.5]$ для приближения используется полином $P_{(0,1,2)}(x)$, построенный на узлах x_0, x_1, x_2; при $x \in [1.5, 2.5]$ строится полином $P_{(1,2,3)}(x)$, построенный на узлах x_1, x_2, x_3; при $x \in [2.5, 4.0]$ — полином $P_{(2,3,4)}(x)$ на узлах x_2, x_3, x_4. Полученная таким образом аппроксимация имеет разрывы в точках $x = 1.5$ и $x = 2.5$.

7.10 Кусочно-полиномиальная аппроксимация

Исходный отрезок $[a, b]$ разбивают на несколько отрезков меньшей длины, на каждом из которых функция интерполируется своим многочленом. Так, для данных в табл. 7.7 разобьем интервал на два отрезка $[0, 2]$ и $[2, 4]$. На первом отрезке строим полином $P_{(0,1,2)}(x)$ на узлах x_0, x_1, x_2, а на втором отрезке — полином $P_{(2,3,4)}(x)$ на сетке x_2, x_3, x_4.



Приведем некоторые рекомендации выполнения интерполирования [1].

1. Если табличная функция $f(x)$ задана на равномерной сетке, то для интерполирования лучше всего использовать формулу Ньютона для равномерной сетки (7.36). Для неравномерной сетки необходимо использовать формулу Ньютона (7.38) или формулу Лагранжа (7.18).
2. На практике интерполировать многочленом высокой степени нежелательно. Необходимо использовать локальную интерполяцию с 3–5 узлами. Если 3–5 узлов (т. е. 3–5 параметров полинома) не обеспечивают требуемой точности, то надо не увеличивать число узлов, а уменьшать шаг таблицы. (Эта рекомендация применима для аналитической функции.)
3. Если функция $f(x)$ задана таблично и аналитическое выражение неизвестно, то оценка погрешности интерполяционного полинома, строго говоря, невозможна, т. к. данному интерполяционному полиному можно сопоставить бесчисленное множество функций, совпадающих в узлах с интерполяционным полином. Однако если известно, что $f(x)$ гладкая, то погрешность интерполирования можно вычислить по приведенным формулам.
4. Если разности максимального порядка практически постоянны, то результат интерполирования имеет столько верных десятичных знаков, сколько их есть в табличных данных, и поэтому оценка погрешности необязательна.

7.11 Тригонометрическая интерполяция

Рассмотрим кратко задачу интерполяции функции f, заданной в точках $0 \leqslant x_0 < x_1 < \ldots < x_{N-1} \leqslant 1$ тригонометрическим полиномом

$$S_N(x) = \sum_{k=-N/2}^{N/2} a_k \exp(2\pi i k x).$$

Не вдаваясь в сложную проблему оценки погрешности тригонометрической интерполяции, отметим тем не менее, что для гладкой периодической с периодом 1 функции f есть основание рассчитывать на выполнение приближенного равенства $f(x) \approx S_N(x)$ для всех $x \in [0, 1]$.

Рассмотрим важный вопрос о чувствительности многочлена S_N к погрешностям в исходных данных. Пусть значения $\tilde{y}_i \approx f(x_i)$ интерполируемой функции задаются с погрешностями ε_i и известно, что $|\varepsilon_i| \leqslant \varepsilon_m$ для $i = 0, 1, \ldots, N - 1$. Тогда

вычисляемый по значениям $\tilde{y}_i$ тригонометрический интерполяционный полином $\tilde{S}_N$ содержит погрешность. Для нее справедлива оценка

$$\max_{[0,1]} \left| \tilde{S}_N(x) - S_N(x) \right| \leqslant \Lambda_N \varepsilon_m.$$

Здесь Λ_N — постоянная, являющаяся аналогом константы Лебега Λ_n.

Примечательно то, что, в отличие от задачи интерполяции алгебраическими многочленами, оптимальным (т. е. дающим минимальное значение Λ_N) является равномерное распределение узлов, которому отвечает значение $\Lambda_N \approx \frac{2}{\pi} \ln\left[(N+1)/2\right]$.

Таким образом, при тригонометрической интерполяции выбор узлов $x_j = j/N$, $j = 0, 1, \ldots, N-1$ является наиболее естественным с точки зрения как простоты вычисления коэффициентов многочлена (быстрое дискретное преобразование Фурье), так и минимизации влияния ошибок исходных данных.

7.12 Приближение сплайнами

Сплайном называют кусочно-полиномиальную функцию, склеенную из различных многочленов, непрерывную на всем отрезке $[a, b]$ вместе со своими несколькими производными [15]. Если используются многочлены первой степени, то имеем линейный сплайн, если используются многочлены второй степени — параболический сплайн, третьей степени — кубический. На практике используют линейные, параболические и кубические сплайны [16].

7.12.1 Линейные сплайны

Линейный сплайн *представляет собой ломаную линию.*

Пусть интерполируемая функция задана своими значениями y_i в узлах x_i, $(i = 0, \ldots, n)$. $a = x_0 < x_1 < x_2 < \ldots < x_n = b$. Для каждого интервала $[x_i, x_{i+1}]$ функция $f(x)$ заменяется линейной функцией

$$S_i(x) = a_i + b_i(x - x_i), \; x_i \leqslant x \leqslant x_{i+1}, \; i = 0, \ldots, n-1. \tag{7.39}$$

Коэффициенты сплайна a_i, b_i находят из следующих условий:

1) в узлах сетки значения сплайна совпадают со значениями функции $f(x_i) = y_i$, т. е.

$$S_i(x_i) = y_i; \tag{7.40}$$

2) в узлах сетки сплайн должен быть непрерывным, т. е.

$$S_i(x_{i+1}) = S_{i+1}(x_{i+1}). \tag{7.41}$$

. . . . . .

Из первого условия находим: $a_i = y_i$, $i = 0, \ldots, n$; из второго условия получим:

$$b_i = \frac{y_{i+1} - y_i}{x_{i+1} - x_i} = \frac{y_{i+1} - y_i}{h_i},\ i = 0,\ \ldots,\ n-1;\ h_i = x_{i+1} - x_i. \tag{7.42}$$

7.12.2 Параболические сплайны

. . . . . .

На каждом интервале $[x_i, x_i + 1]$ *функция* $f(x)$ *интерполируется* ***квадратичной функцией*** *вида*

$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2. \tag{7.43}$$

. . . . . .

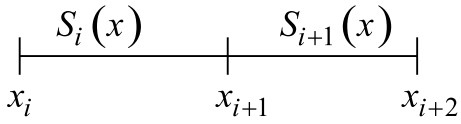
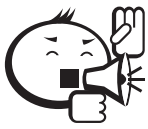


Рис. 7.3



. . . . . .

Здесь мы имеем уже $3n$ коэффициентов (a_i, b_i, c_i), $i = 0, \ldots, n-1$, определяемых из условий:

1) в узлах сетки значения сплайна совпадают со значениями интерполируемой функции $y_i = f(x_i)$ (см. (7.40));
2) в узлах сетки сплайн должен быть непрерывен (см. (7.41));
3) в узлах сетки первая производная должна быть непрерывна, т. е.

$$S_i'(x_{i+1}) = S_{i+1}'(x_{i+1}). \tag{7.44}$$

. . . . . .

Из этих условий получим следующие соотношения для коэффициентов сплайна (a_i, c_i):

$$a_i = y_i,\ i = 0,\ \ldots,\ n, \tag{7.45}$$

$$c_i = \frac{b_{i+1} - b_i}{2h_i},\ i = 0,\ \ldots,\ n-1. \tag{7.46}$$

Коэффициенты b_i находим из системы уравнений:

$$\frac{2}{h_i}(y_{i+1} - y_i) = b_i + b_{i+1},\ i = 0,\ \ldots,\ n-1. \tag{7.47}$$

Имеем n уравнений для определения $(n+1)$ неизвестных $b_0, b_1, \ldots, b_n$. Поэтому надо задать граничное условие — либо $b_0 = A_0$, либо $b_n = A_n$, где A_0, A_n — значения первой производной функции $f'(x)$ в точках $x = x_0$ и $x = x_n$.

Если известно значение $b_0 = A_0$, то из (7.47) имеем

$$b_{i+1} = z_i - b_i,\ i = 0, \ldots, n-1. \tag{7.48}$$

Если известно $b_n = A_n$, то из (7.47) следует алгоритм

$$b_{n-i} = z_{n-i} - b_{n-i+1},\ i = 1, \ldots, n. \tag{7.49}$$

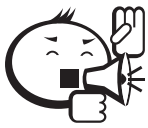
В формулах (7.48), (7.49) $z_i = \dfrac{2\,(y_{i+1} - y_i)}{h_i}$.

7.12.3 Кубические сплайны

Будем строить ***интерполяционный кубический сплайн*** *в следующем виде: на каждом интервале* $[x_i, x_i + 1]$ *функция* $f(x)$ *аппроксимируется полиномом*

$$S_i'(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3,\ i = 0, \ldots, n-1, \tag{7.50}$$

где a_i, b_i, c_i, d_i *— коэффициенты сплайна, которые необходимо определить (всего имеем 4n коэффициентов).*



Коэффициенты сплайна определяем из следующих условий:

1) в узлах сетки значения сплайна совпадают со значениями интерполируемой функции $y_i = f(x_i)$ (см. (7.40));
2) значения $S_i(x)$ в узлах x_{i+1} совпадают с $S_{i+1}(x)$ (условие непрерывности сплайна, см. (7.41));
3) значения производных $S_i'(x)$ в узлах x_{i+1} совпадают с $S_{i+1}'(x)$ (условие непрерывности первой производной, см. (7.44));
4) значения $S_i''(x)$ в узлах x_{i+1} совпадают с $S_{i+1}''(x)$ (условие непрерывности второй производной)

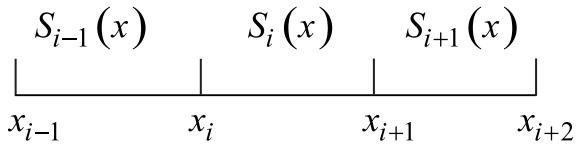


Рис. 7.4

В результате получим следующие соотношения для коэффициентов сплайна a_i, b_i, c_i, d_i:

$$a_i = y_i,\ i = 0,\ \ldots,\ n. \tag{7.51}$$

$$b_i = \frac{y_{i+1} - y_i}{h_i} - \frac{h_i}{6}(2M_i + M_{i+1}),\ i = 0, \ldots, n-1. \tag{7.52}$$

$$d_i = \frac{M_{i+1} - M_i}{6h_i}, \quad c_i = \frac{1}{2}M_i,\ i = 0, \ldots, n-1. \tag{7.53}$$

Здесь за M_i обозначена вторая производная сплайна.

Величины M_i определяем из системы линейных уравнений:

$$\frac{h_{i-1}}{6}M_{i-1} + \frac{h_{i-1} + h_i}{3}M_i + \frac{h_i}{6}M_{i+1} = \frac{y_{i+1} - y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}},\ i = 1, 2, \ldots, n-1. \tag{7.54}$$

Здесь $h_i = x_{i+1} - x_i$.

Эта система состоит из $(n-1)$ уравнения с $(n+1)$ неизвестным. Поэтому два неизвестных мы должны определить *apriory*, т. е. необходимо задать граничные условия:

$$M_0 = B_0; \quad M_n = B_n. \tag{7.55}$$

Введем следующие обозначения:

$$\left.\begin{aligned} a_{jj} &= \frac{1}{3}(h_{j-1} + h_j),\ j = \overline{1, n-1}; \\ a_{j,j+1} &= a_{j+1,j} = \frac{1}{6}h_j,\ j = \overline{1, n-3}. \end{aligned}\right\} \tag{7.56}$$

$$\begin{cases} g_1 = \dfrac{1}{h_1}(y_2 - y_1) - \dfrac{1}{h_0}(y_1 - y_0) - \dfrac{h_0}{6}B_0, \\ g_i = \dfrac{1}{h_i}y_{i+1} - \left(\dfrac{1}{h_i} + \dfrac{1}{h_{i-1}}\right)y_i + \dfrac{1}{h_{i-1}}y_{i-1},\ i = 2, \ldots, n-2, \\ g_{n-1} = \dfrac{1}{h_{n-1}}(y_n - y_{n-1}) - \dfrac{1}{h_{n-2}}(y_{n-1} - y_{n-2}) - \dfrac{h_{n-1}}{6}B_n. \end{cases} \tag{7.57}$$

Система (7.54) в матричном виде запишется как

$$AM = g, \tag{7.58}$$

где матрица A — трехдиагональная симметричная матрица размерности $(n-1) \times (n-1)$ с элементами, определяемыми формулами (7.56); g — $(n-1)$-мерный вектор, с элементами (7.57).

Вычислив значения M_i из (7.58), затем по формулам (7.51), (7.52), (7.53) рассчитываем коэффициенты сплайна a_i, b_i, c_i, d_i.

Если известны граничные условия на первую производную, т. е.

$$b_0 = A_0, \quad b_n = A_n, \tag{7.59}$$

то к системе (7.54) мы можем добавить еще два уравнения. В результате вместо (7.54) мы получим систему:

$$\begin{cases} \dfrac{h_0}{3}M_0 + \dfrac{h_0}{6}M_1 = \dfrac{y_1 - y_0}{h_0} - A_0, \\ \dfrac{h_{i-1}}{6}M_{i-1} + \dfrac{h_{i-1} + h_i}{3}M_i + \dfrac{h_i}{6}M_{i+1} = \dfrac{y_{i+1} - y_i}{h_i} - \dfrac{y_i - y_{i-1}}{h_{i-1}}, \ i = \overline{1, n-1}, \\ \dfrac{h_{n-1}}{6}M_{n-1} + \dfrac{h_{n-1}}{3}M_n = -\dfrac{y_n - y_{n-1}}{h_{n-1}} + A_n. \end{cases} \tag{7.60}$$

Эта система имеет $(n + 1)$ уравнение и $(n + 1)$ неизвестное M_0, M_1, …, M_n. Матрица системы (7.60) трехдиагональная, симметричная, с элементами:

$$\begin{aligned} &\tilde{a}_{0,0} = \frac{h_0}{3}, \quad \tilde{a}_{n,n} = \frac{h_{n-1}}{3}, \\ &\tilde{a}_{j,j} = \frac{1}{3}(h_{j-1} + h_j), \ j = 1, \ldots, n-1, \\ &\tilde{a}_{j,j+1} = \tilde{a}_{j+1,j} = \frac{h_j}{6}, \ j = 0, 1, \ldots, n-1. \end{aligned} \tag{7.61}$$

Введем вектор $\tilde{g}$ с компонентами:

$$\tilde{g}_0 = \frac{y_1 - y_0}{h_0} - A_0, \quad \tilde{g}_n = -\frac{y_n - y_{n-1}}{h_{n-1}} + A_n, \quad \tilde{g}_i = \frac{y_{i+1} - y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}}, \ i = \overline{1, n-1}. \tag{7.62}$$

С учетом (7.61) и (7.62) систему (7.60) можно записать в матричном виде:

$$\tilde{A}M = \tilde{g}. \tag{7.63}$$

Трехдиагональные системы линейных алгебраических уравнений (7.58) и (7.63) решают специальным методом *прогонки* (см. п. 4.9).

7.13 Интегральное квадратичное аппроксимирование функций на отрезке

Предположим, что данную непрерывную функцию $f(x)$ нужно аппроксимировать на отрезке $[a, b]$ с помощью обобщенного полинома [17]

$$\Phi_m(x) = \sum_{i=0}^{m} a_i \varphi_i(x), \tag{7.64}$$

где $\{\varphi_i(x)\}$ — заданная система непрерывных функций, a_i — постоянные.

Коэффициенты a_i будем подбирать из условия минимального квадратичного отклонения $\Phi_m(x)$ от $f(x)$ на отрезке $[a,b]$ (минимума невязки S):

$$S = \int_a^b \left[f(x) - \sum_{i=0}^{m} a_i \varphi_i(x) \right]^2 dx.$$

Запишем необходимое условие минимума невязки S:

$$\frac{\partial S}{\partial a_k} = -2 \int_a^b \left[f(x) - \sum_{i=0}^{m} a_i \varphi_i(x) \right] \varphi_k(x)\, dx = 0. \qquad (7.65)$$

Вводя сокращенные обозначения:

$$(\varphi_i, \varphi_k) = \int_a^b \varphi_i(x)\, \varphi_k(x)\, dx,$$

$$(f, \varphi_k) = \int_a^b f(x) \varphi_k(x)\, dx,$$

из (7.65) получим систему $(m+1)$ уравнений для нахождения коэффициентов a_i, $i = 0, \ldots, m$:

$$\sum_{i=0}^{m} a_i (\varphi_i, \varphi_k) = (f, \varphi_k), \; k = 0, \ldots, m. \qquad (7.66)$$

Если функции $\varphi_i(x)$ линейно независимы на $[a,b]$, то система (7.66) имеет единственное решение, которое соответствует наименьшему квадратичному отклонению.

Пример 7.8

Найти наилучшую квадратичную аппроксимацию функции $f(x) = \sqrt{x}$ на отрезке $[0,1]$ с помощью полинома $\Phi(x) = c_0 + c_1 x$.

Решение:

Здесь $\varphi_0(x) = 1$; $\varphi_1(x) = x$.

Имеем:

$$(\varphi_0, \varphi_0) = \int_0^1 1^2 dx = 1, \quad (f, \varphi_0) = \int_0^1 \sqrt{x} dx = \frac{2}{3},$$

$$(\varphi_1, \varphi_0) = \int_0^1 x dx = \frac{1}{2}, \quad (f, \varphi_1) = \int_0^1 x\sqrt{x} dx = \frac{2}{5},$$

$$(\varphi_1, \varphi_1) = \int_0^1 x^2 dx = \frac{1}{3}.$$

Система (7.66) имеет вид:

$$\begin{cases} c_0 + \frac{1}{2}c_1 = \frac{2}{3}, \\ \frac{1}{2}c_0 + \frac{1}{3}c_1 = \frac{2}{5}. \end{cases}$$

Решаем систему, получим $c_0 = \frac{4}{15}$, $c_1 = \frac{4}{5}$.

Таким образом, аппроксимирующий полином имеет вид:

$$\Phi(x) = \frac{4}{15} + \frac{4}{5}x, \quad S_{\min} = \int_0^1 \left(\sqrt{x} - \frac{4}{15} - \frac{4}{5}x\right)^2 dx = 0.396.$$

Неудобством интегральной квадратичной аппроксимации является необходимость вычисления определенных интегралов, которые не всегда выражаются через элементарные функции. Если $f(x)$ задана таблично, то интегралы (f, φ_i) приходится вычислять численным образом. В этом смысле способ точечной квадратичной аппроксимации предпочтительней.

7.14 Ортогональные системы функций

Выше мы рассматривали (см. п. 7.2) ортогональную систему функций, заданную на дискретном множестве точек. Введем теперь ортогональную систему функций на отрезке $[a, b]$.

Система интегрируемых функций $\{\varphi_i(x)\}$ *называется* ***ортогональной*** *на* $[a, b]$*, если*

$$(\varphi_m, \varphi_n) = \int_a^b \varphi_m(x)\varphi_n(x)\,dx = 0 \text{ при } m \neq n. \qquad (7.67)$$

Число $\|\varphi_m\| = \sqrt{(\varphi_m, \varphi_m)} = \sqrt{\int_a^b \varphi_m^2(x)\,dx}$ *называется* ***нормой*** *функции* $\varphi_m(x)$ *на* $[a, b]$*. Для ортонормированной системы выполняется условие:*

$$\int_a^b \varphi_m(x)\varphi_n(x)\,dx = \delta_{mn} = \begin{cases} 1, & m = n, \\ 0, & m \neq n. \end{cases}$$

Очевидно, всякую систему ортогональных функций можно нормировать. Системы функций

$$\psi_i(x) = \frac{\varphi_i(x)}{\|\varphi_i\|},\ i - 0,\ \ldots,\ n$$

нормированы, так как

$$\int_a^b \psi_i^2(x)\,dx = \int_a^b \frac{\varphi_i^2(x)}{\|\varphi_i\|^2}dx = \frac{1}{\|\varphi_i\|^2}\int_a^b \varphi_i^2(x)\,dx = 1,$$

$$\int_a^b \psi_m(x)\,\psi_n(x)\,dx = \frac{1}{\|\varphi_m\| \cdot \|\varphi_n\|}\int_a^b \varphi_m(x)\,\varphi_n(x)\,dx = \delta_{mn}.$$

Пример 7.9

Пронормировать систему функций

$$1,\ x,\ x^2,\ \ldots,\ x^m, \tag{7.68}$$

заданную на отрезке $[0,1]$.

Решение:

Имеем:

$$\psi_0(x) = \frac{1}{\sqrt{\int_0^1 1^2\,dx}} = 1,$$

$$\psi_i(x) = \frac{x^i}{\sqrt{\int_0^1 x^{2i}\,dx}} = \frac{x^i}{\sqrt{\frac{1}{2i+1}}}\sqrt{2i+1}\,x^i,\ i = 1,\ \ldots,\ m.$$

После нормировки система (7.68) имеет вид:

$$1,\ \sqrt{3}x,\ \sqrt{5}x^2,\ \ldots,\ \sqrt{2m+1}x^m.$$

Если система функций $\{\varphi_i(x)\}$ ортогональна на $[a,b]$, то задача о квадратичной аппроксимации $f(x)$ на $[a,b]$ с помощью обобщенного полинома

$$\Phi_m(x) = \sum_{i=0}^{m} a_i\varphi_i(x)$$

получает простое решение (см. (7.15)). В самом деле, из необходимого условия минимума интеграла

$$S = \int_a^b \left[f(x) - \Phi_n(x)\right]^2 dx = \int_a^b \left[f(x) - \sum_{i=0}^{m} a_i\varphi_i(x)\right]^2 dx \tag{7.69}$$

для определения коэффициентов a_j имеем систему:

$$\frac{\partial S}{\partial a_j} = -2 \int_a^b \left[f(x) - \sum_{i=0}^{m} a_i \varphi_i(x) \right] \varphi_j(x)\, dx = 0,\ j = 0, \ldots, m. \tag{7.70}$$

С учетом условия (7.67) из (7.70) получим:

$$a_j \int_a^b \varphi_j^2(x)\, dx = \int_a^b f(x) \varphi_j(x) dx$$

или

$$a_j = \frac{(f, \varphi_j)}{\|\varphi_j\|^2},\ j = 0, 1, \ldots, m. \tag{7.71}$$

Если $\{\varphi_i\}$ ортонормированна, то

$$a_j = \int_a^b f(x)\, \varphi_j(x)\, dx. \tag{7.72}$$

Коэффициенты a_j, определяемые (7.71) (*или* (7.72))*, называются* ***коэффициентами Фурье*** *функции $f(x)$ относительно заданной системы $\{\varphi_i(x)\}$* [3].

Вычислим отклонение (7.69):

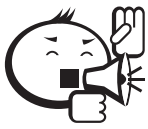
$$\begin{aligned} S &= \int_a^b \left[f^2 - 2 \sum_{i=0}^{m} a_i \varphi_i(x) f(x) + \sum_i \sum_k a_i a_k \varphi_i(x)\, \varphi_k(x) \right] dx = \\ &= \int_a^b f^2 dx - 2 \sum_{i=0}^{m} a_i^2 \|\varphi_i\|^2 + \sum_{i=0}^{m} a_i^2 \|\varphi_i\|^2 = \|f\|^2 - \sum_{i=0}^{m} a_i^2 \|\varphi_i\|^2. \end{aligned} \tag{7.73}$$

Так как $S > 0$, то из (7.73) следует

$$\|f\|^2 \geqslant \sum_{i=0}^{m} a_i^2 \|\varphi_i\|^2 \text{ — неравенство Бесселя.}$$

Для ортонормированной системы $\{\varphi_i(x)\}$

$$\|f\|^2 \geqslant \sum_{i=0}^{m} a_i^2.$$



. . . . . .

Отметим следующие свойства обобщенного полинома $\Phi(x)$ с коэффициентами Фурье (7.71) [3]:

1) обобщенный полином $\Phi_m(x)$ с коэффициентами (7.71) для данной функции $f(x)$ обладает наименьшим квадратичным отклонением от этой функции по сравнению со всеми другими обобщенными полиномами того же порядка m.

   Докажем это свойство. Для этого вычислим квадратичную форму d^2S:

$$d^2S = \sum_{ij} \frac{\partial^2 S}{\partial a_i \partial a_j} da_i\, da_j = 2 \sum_{ij} da_i\, (\varphi_i \varphi_j)\, da_j = \\ = 2 \sum_i (da_i)^2 \|\varphi_i\|^2 > 0,$$

   т. е. матрица вторых производных положительно определена. Следовательно, при значениях a_j, определяемых формулой (7.71), квадратичное отклонение S имеет минимум. Так как S имеет единственный минимум, то легко убедиться, что значение S, соответствующее коэффициентам Фурье a_j, является наименьшим в пространстве коэффициентов $a_0, a_1, \ldots, a_m$;

2) при увеличении числа слагаемых m в разложении (7.64) младшие коэффициенты a_j (см. 7.71) остаются неизменными, т. е. при добавлении новых членов проделанная вычислительная работа сохраняется;

3) при увеличении m квадратичная погрешность

$$S_m = \int_0^\infty \left[f(x) - \Phi_n(x) \right]^2 dx,$$

   в силу формулы (7.73), монотонно убывает, т. е.

$$S_1 \geqslant S_2 \geqslant S_3 \geqslant \ldots \geqslant S_m \geqslant S_{m+1} \geqslant \ldots$$

   Таким образом, присоединение новых слагаемых увеличивает точность аппроксимации.

. . . . . .

Если $\{\varphi_i\}$ такова, что для любой непрерывной функции $f(x)$ справедливо соотношение

$$\lim_{m \to \infty} S_m = 0,$$

то эта система называется *полной*. В противном случае эта система называется *неполной*.

Для полной системы справедливо равенство Парсеваля

$$\sum_{i=0}^{\infty} a_i^2 \|\varphi_i\|^2 = \|f(x)\|^2.$$

Рассмотрим примеры систем ортогональных функций.

7.14.1 Ортогональная система тригонометрических функций

Рассмотрим систему функций [17]:

$$1,\ \sin x,\ \cos x,\ \sin 2x,\ \cos 2x,\ \ldots,\ \sin nx,\ \cos nx. \tag{7.74}$$

Покажем, что эта система ортогональна на $[-\pi, \pi]$. Вычислим следующие интегралы для целых m и n:

а) $\int\limits_{-\pi}^{\pi} \sin mx \cos nx\, dx = 0$ при $m \neq n$ (так как подынтегральная функция нечетная);

б) $\int\limits_{-\pi}^{\pi} \sin mx \sin nx\, dx = \frac{1}{2}\left[\frac{\sin(m-n)x}{m-n} - \frac{\sin(m+n)x}{m+n}\right]_{-\pi}^{\pi} = 0;$

в) $\int\limits_{-\pi}^{\pi} \cos mx \cos nx\, dx = \frac{1}{2}\left[\frac{\sin(m-n)x}{m-n} + \frac{\sin(m+n)x}{m+n}\right]_{-\pi}^{\pi} = 0.$

Полагая в 1-м и 3-м интеграле $n = 0$, получим:

$$\int\limits_{-\pi}^{\pi} 1 \cdot \sin mx\, dx = 0, \quad \int\limits_{-\pi}^{\pi} 1 \cdot \cos mx\, dx = 0.$$

Пусть $m = n$. Тогда

а) $\int\limits_{-\pi}^{\pi} \sin mx \cos mx\, dx = 0$, так как подынтегральная функция нечетная;

б) $\int\limits_{-\pi}^{\pi} \sin mx \sin mx\, dx = \left[\frac{1}{2}x - \frac{\sin 2mx}{4m}\right]_{-\pi}^{\pi} = \pi;$

в) $\int\limits_{-\pi}^{\pi} \cos mx \cos mx\, dx = \left[\frac{1}{2}x + \frac{\sin 2mx}{4m}\right]_{-\pi}^{\pi} = \pi.$

Таким образом, система (7.74) является ортогональной на отрезке $[-\pi, \pi]$, а значит, и на любом отрезке $[a, a + 2\pi]$. Нормы функций равны $\|\sin mx\| = \sqrt{\pi}$, $\|\cos mx\| = \sqrt{\pi}$, т. е. не зависят от номера m.

$$\|1\| = \left\{\int\limits_{-\pi}^{\pi} 1\, dx\right\}^{1/2} = \sqrt{2\pi}.$$

Рассмотрим полином

$$Q(x) = \frac{a_0}{2} + \sum_{k=1}^{n} (a_k \cos kx + b_k \sin kx). \tag{7.75}$$

В формуле (7.75) «нулевое» слагаемое взято с коэффициентом $1/2$. Это сделано для того, чтобы унифицировать формулу получения коэффициентов a_k:

$$a_k = \frac{\int_{-\pi}^{\pi} f(x)\,\varphi_k(x)\,dx}{\|\varphi_k\|^2}.$$

Отсюда следует, что $a_0 = \dfrac{1}{\|\varphi_0\|^2}\int_{-\pi}^{\pi} f(x)\,\varphi_0(x)\,dx.$

Здесь $\varphi_0 = 1$, $\|\varphi_0\|^2 = 2\pi$ поэтому

$$a_0 = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(x)\,dx; \tag{7.76}$$

$$a_k = \frac{1}{\|\cos kx\|^2}\int_{-\pi}^{\pi} f(x)\cos kx\,dx = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\cos kx\,dx,\ k = 1, 2, \ldots \tag{7.77}$$

Формула (7.77) является общей для получения a_k. При $k = 0$ из (7.77) следует:

$$a_0 = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\,dx. \tag{7.78}$$

Сравнивая с (7.76), видим, что необходимо уменьшить a_0, вычисляемый по формуле (7.78), в два раза. Этот факт и отражён в формуле (7.75).

Пусть дана функция $f(x)$, которую мы хотим аппроксимировать полиномом (7.75). Для того, чтобы квадратичное отклонение

$$S = \int_{-\pi}^{\pi} \left[f(x) - Q(x)\right]^2 dx$$

было минимальным, коэффициенты a_0, a_k, b_k должны быть коэффициентами Фурье, т. е.

$$\left.\begin{aligned} a_k &= \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\cos kx\,dx, \\ b_k &= \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\sin kx\,dx. \end{aligned}\right\}\ k = 0, 1, \ldots, n. \tag{7.79}$$

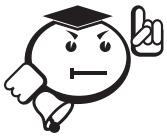
Из (7.79) следует, что если $f(x)$ четная, то $b_k = 0$ и

$$Q(x) = \frac{a_0}{2} + \sum_{k=1}^{n} a_k \cos kx; \quad a_k = \frac{2}{\pi}\int_{0}^{\pi} f(x)\cos kx\,dx.$$

Если $f(x)$ нечетная, то $a_k = 0$.

$$Q(x) = \sum_{k=1}^{n} b_k \sin kx; \quad b_k = \frac{2}{\pi}\int_{0}^{\pi} f(x)\sin kx\,dx.$$

Если $f(x)$ задана несложным аналитическим выражением, то коэффициенты полинома Фурье $Q(x)$ вычисляются по формулам (7.79). Если же $f(x)$ задана таблично или имеет сложный аналитический вид, то коэффициенты a_k и b_k вычисляются численным образом.



Если функция $f(x)$ задана на интервале $[a,b]$, то необходимо предварительно выполнить преобразование аргумента $x = a + \dfrac{y+\pi}{2\pi}(b - a) = \dfrac{b+a}{2} + \dfrac{b-a}{2\pi}y$, где $y \in [-\pi, \pi]$; $x = [a,b]$. Тогда формула (7.79) примет вид:

$$\left.\begin{aligned} a_k &= \frac{1}{\pi}\int_{-\pi}^{\pi} f\left(\frac{b+a}{2} + \frac{b-a}{2\pi}y\right)\cos ky\,dy \\ b_k &= \frac{1}{\pi}\int_{-\pi}^{\pi} f\left(\frac{b+a}{2} + \frac{b-a}{2\pi}y\right)\sin ky\,dy \end{aligned}\right\} \quad k = 0,\ 1,\ \ldots,\ n.$$

7.14.2 Полиномы Лежандра

Полиномы Лежандра определяются следующей формулой Родрига [17]:

$$P_n(x) = \frac{1}{2^n n!}\cdot\frac{d^n}{dx^n}\left(x^2 - 1\right)^n, \quad n = 0,\ 1,\ 2,\ \ldots$$

В частности, имеем:

$$P_0(x) = 1; \quad P_1(x) = x; \quad P_2(x) = \frac{1}{2}\left(3x^2 - 1\right);$$

$$P_3(x) = \frac{1}{2}\left(5x^3 - 3x\right); \quad P_4(x) = \frac{1}{8}\left(35x^4 - 30x^2 + 3\right);$$

$$P_5(x) = \frac{1}{8}\left(63x^5 - 70x^3 + 15\right);$$

$$P_6(x) = \frac{1}{16}\left(231x^6 - 315x^4 + 105x^2 - 5\right).$$

Теорема 7.5 [17]. Полиномы Лежандра образуют ортогональную систему на отрезке $[-1,1]$, т. е.

$$\int_{-1}^{1} P_n(x)\,P_m(x)\,dx = 0 \text{ при } m \neq n,\ (m,\ n = 0,\ 1,\ 2,\ \ldots).$$

Норма полинома Лежандра

$$\|P_n(x)\|^2 = \int_{-1}^{1} P_n^2(x)\,dx = \frac{2}{2n+1}. \tag{7.80}$$

. . . . . .

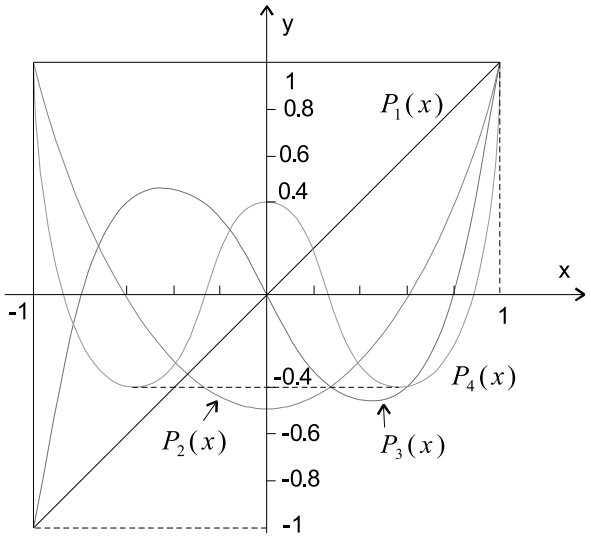
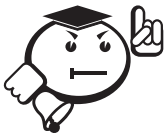


Рис. 7.5

. . . . . .



Замечание 1. С помощью линейного преобразования

$$z = \frac{b-a}{2}x + \frac{b+a}{2}, \quad \text{где } -1 \leqslant x \leqslant 1 \tag{7.81}$$

можно получить полиномы

$$\tilde{P}_m(z) = P_m\left(\frac{z-(b+a)/2}{(b-a)/2}\right), \tag{7.82}$$

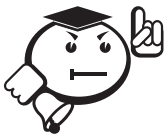
ортогональные на отрезке $[a,b]$, т. е.

$$\int_a^b \tilde{P}_m(z)\,\tilde{P}_n(z)\,dz = 0 \text{ при } m \neq n.$$

Используя (7.80), легко получить

$$\int_a^b \tilde{P}_n^2(z)\,dz = \frac{b-a}{2n+1},\ n = 0,\ 1,\ 2,\ \ldots$$

. . . . . .



Замечание 2. Для полиномов Лежандра справедливы следующие рекуррентные соотношения:

$$(n+1)\,P_{n+1}(x) - x\,(2n+1)\,P_n(x) + nP_{n-1}(x) = 0,\ n = 1, 2, 3, \ldots$$

Для функции $f(x)$, заданной на интервале $[-1, 1]$, коэффициенты разложения c_i определяются по формулам:

$$c_i = \frac{2i+1}{2}\int_{-1}^{1} f(x)P_i(x)\,dx,\ i = 0, 1, \ldots, n. \qquad (7.83)$$

Это следует из формул (7.71) и (7.80).

Погрешность аппроксимации равна

$$S = \int_{-1}^{1}\left[f(x) - \sum_{i=0}^{n} c_i P_i(x)\right]^2 dx = \int_{-1}^{1} f^2(x) - \sum_{i=0}^{n} c_i^2 \int_{-1}^{1} P_i^2(x)\,dx. \qquad (7.84)$$

Пример 7.10

Необходимо функцию $f(x) = |x|$ аппроксимировать полиномом Лежандра 5-ой степени на отрезке $[-1, 1]$.

Решение:

Полином $Q_5(x)$ ищем в виде

$$Q_5(x) = c_0P_0(x) + c_1P_1(x) + c_2P_2(x) + c_3P_3(x) + c_4P_4(x) + c_5P_5(x).$$

Так как функция $f(x) = |x|$ — четная и $P_k(x)$ — четны при четном k и нечетны при нечетном k, то из формулы (7.83) получим:

$$c_0 = \frac{1}{2}\int_{-1}^{1} |x|\,dx = \int_{0}^{1} x\,dx = \frac{1}{2};$$

$$c_1 = c_3 = c_5 = 0;$$

$$c_2 = \frac{5}{2}\int_{-1}^{1} |x|\,P_2(x)\,dx = \frac{5}{2}\int_{0}^{1} x(3x^2 - 1)\,dx = \frac{5}{8};$$

$$c_4 = \frac{9}{8}\int_{0}^{1} x(35x^4 - 30x^2 + 3)\,dx = -\frac{3}{16}.$$

В результате выражение для $Q_5(x)$ имеет вид:

$$Q_5(x) = \frac{1}{2} + \frac{5}{16}(3x^2 - 1) - \frac{3}{128}(35x^4 - 30x^2 + 3) = \frac{15}{128}(-7x^4 + 14x^2 + 1).$$

Следовательно,

$$|x| \approx \frac{15}{128}(-7x^4 + 14x^2 + 1).$$

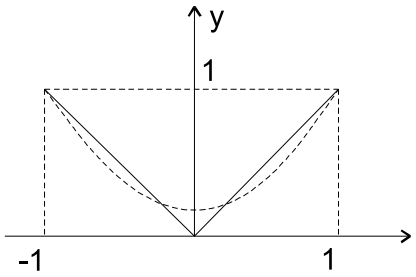


Рис. 7.6

Контрольные вопросы по главе 7

1. Что понимается под приближением функции? Приведите примеры.
2. Запишите обобщенный полином и условие, которому он должен удовлетворять.
3. Как вычисляются коэффициенты полинома, построенного на ортогональной системе функций?
4. Запишите интерполяционный полином Лагранжа для табличной функции. Определите степень этого полинома.
5. Запишите погрешность интерполяционного полинома Лагранжа.
6. Как можно минимизировать погрешность интерполяции?
7. Как строится интерполяционный полином Ньютона для равномерной сетки?
8. Чем отличается интерполяционный полином Ньютона для неравномерной сетки от полинома для равномерной сетки?
9. Как определяется чувствительность интерполяционного полинома к погрешностям входных данных?
10. Дайте определение сплайна. Какие сплайны вам известны?
11. Что такое интегральное квадратичное аппроксимирование функций на отрезке?

Глава 8

ЧИСЛЕННОЕ ДИФФЕРЕНЦИРОВАНИЕ ФУНКЦИЙ

К численному дифференцированию прибегают во всех случаях, когда функцию $f(x)$ невозможно или трудно продифференцировать аналитически, — например, если она задана в виде таблицы или имеет очень сложный аналитический вид. Численное дифференцирование необходимо также: а) при решении дифференциальных уравнений при помощи разностных методов; б) при решении нелинейных уравнений; в) при поиске точек экстремума функций.

8.1 Простейшие формулы численного дифференцирования

Вычисление первой производной

Предположим, что в окрестности точки x функция f дважды дифференцируема. Исходя из определения первой производной

$$f'(x) = \lim_{\Delta x \to 0} \frac{f(x+\Delta x) - f(x)}{\Delta x},$$

в качестве приближенных формул первой производной можно использовать [1]:

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}, \tag{8.1}$$

$$f'(x) \approx \frac{f(x) - f(x-h)}{h}. \tag{8.2}$$

Здесь $h > 0$ — шаг. Формулы (8.1) и (8.2) называют *правой* и *левой разностными производными* [1, 3].

Для оценки погрешностей формул (8.1) и (8.2):

$$r_+(x,h) = f'(x) - \frac{f(x+h) - f(x)}{h},$$

$$r_-(x,h) = f'(x) - \frac{f(x) - f(x-h)}{h}$$

воспользуемся формулами Тейлора:

$$f(x \pm h) = f(x) \pm f'(x)h + \frac{f''(\xi_\pm)}{2}h^2,\ \xi_+ \in (x, x+h),\ \xi_- \in (x-h, x). \tag{8.3}$$

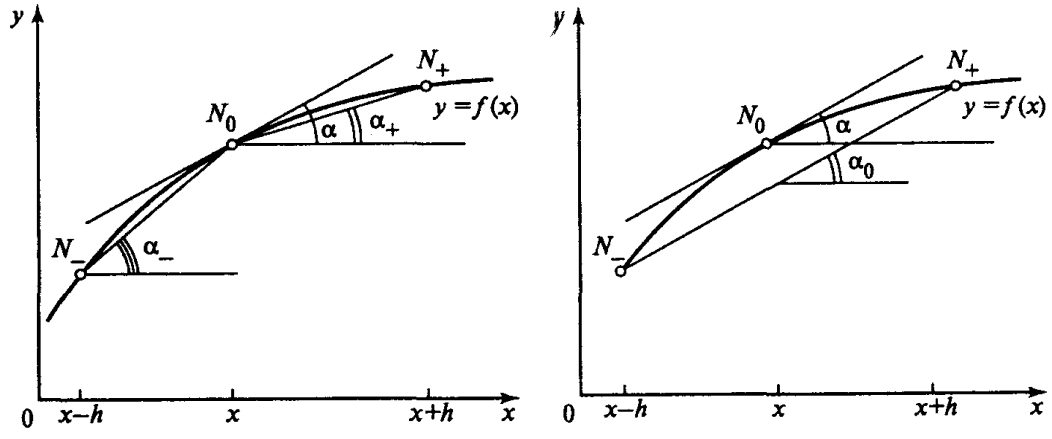
Подставим разложения (8.3) в выражения для $r_\pm$, получим $r_+ = -\frac{f''(\xi_+)}{2}h$, $r_- = \frac{f''(\xi_-)}{2}h$. Отсюда

$$\left|r_+(x,h)\right| \leqslant \frac{1}{2}M_2 h, \quad M_2 = \max_{|x,x+h|}\left|f''(\xi)\right|, \tag{8.4}$$

$$\left|r_-(x,h)\right| \leqslant \frac{1}{2}M_2 h, \quad M_2 = \max_{|x-h,x|}\left|f''(\xi)\right|. \tag{8.5}$$

Таким образом, формулы (8.1), (8.2) имеют первый порядок точности по h.

Приведенные формулы численного дифференцирования имеют простую геометрическую интерпретацию (см. рис 8.1, *а*). Пусть N_0, N_-, N_+ — точки с координатами $\big(x, f(x)\big)$, $\big(x-h, f(x-h)\big)$ и $\big(x+h, f(x+h)\big)$, расположенные на графике. Производная $f'(x)$ равна тангенсу угла α наклона к оси Ox касательной, проведенной к графику функции в точке N_0. Формула (8.1) соответствует замене $f'(x) = \operatorname{tg}\alpha$ на тангенс угла наклона α_+ секущей, проходящей через точки N_0 и N_+. Формула (8.2) соответствует замене $f'(x) = \operatorname{tg}\alpha$ на тангенс угла наклона α_- секущей, проходящей через точки N_0 и N_-.



а) *б*)

Рис. 8.1

Точность формул численного дифференцирования (8.1), (8.2) можно улучшить, если производную $f'(x)$ аппроксимировать *центральной разностной производной*

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}, \tag{8.6}$$

что соответствует углу наклона α_0 секущей, проведенной через точки N_- и N_+ (см. рис 8.1, *б*).

Погрешность формулы (8.6)

$$r_0(x,h) = f'(x) - \frac{f(x+h) - f(x-h)}{2h}$$

оценим также с помощью формулы Тейлора, ограничившись членами третьего порядка:

$$f(x \pm h) = f(x) \pm f'(x)h + \frac{f''(x)}{2}h^2 \pm \frac{f^{(3)}(\xi_\pm)}{6}h^3.$$

Получим:

$$r_0(x,h) = -\frac{f^{(3)}(\xi_+) + f^{(3)}(\xi_-)}{12}h^2.$$

Поэтому имеем оценку погрешности:

$$\left|r_0(x,h)\right| \leqslant \frac{M_3}{6}h^2, \quad M_3 = \max_{|x-h,\,x+h|}\left|f^{(3)}(\xi)\right|. \tag{8.7}$$

Таким образом, центральная разностная производная аппроксимирует производную $f'(x)$ со вторым порядком точности относительно h.

Вычисление второй производной

Для приближенного вычисления второй производной используют следующую формулу:

$$f''(x) \approx \frac{f(x-h) - 2f(x) + f(x+h)}{h^2}, \tag{8.8}$$

которую называют *второй разностной производной*.

Погрешность формулы (8.8) равна

$$r(x,h) = f''(x) - \frac{f(x-h) - 2f(x) + f(x+h)}{h^2}. \tag{8.9}$$

Запишем формулу Тейлора для функции $f(x \pm h)$ и ограничимся членами 4-го порядка:

$$f(x \pm h) = f(x) \pm f'(x)h + \frac{f''(x)h^2}{2} \pm \frac{f^{(3)}(x)h^3}{6} + \frac{f^{(4)}(\xi_\pm)h^4}{24}.$$

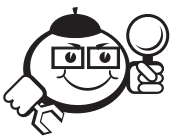
Подставляя это разложение в (8.9), получим:

$$r(x,h) = -\frac{f^{(4)}(\xi_+) + f^{(4)}(\xi_-)}{24}h^2.$$

Отсюда для оценки сверху имеем:

$$|r(x,h)| \leqslant \frac{M_4}{12}h^2, \quad M_4 = \max_{[x-h,\,x+h]}\left|f^{(4)}(\xi)\right|. \tag{8.10}$$

Таким образом, формула (8.8) имеет второй порядок точности.



Пример 8.1

Используя данные табл. 8.1, в которой приведены значения функции $f(x) = e^x$, найдем значение второй производной по формуле (8.8) во внутренних узлах таблицы и погрешность по формуле (8.10) (см. табл. 8.2).

Таблица 8.1

x	0.0	0.2	0.4	0.6	0.8	1.0
$f(x)$	1.00000	1.22140	1.49182	1.82212	2.22554	2.71828

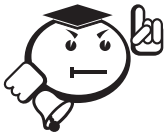
Таблица 8.2

x	0.2	0.4	0.6	0.8
$f''(x)$	1.22550	1.49700	1.82800	2.23300
$r(x)$	−0.00410	−0.00518	−0.00588	−0.00746

8.2 Общий способ получения формул численного дифференцирования

Основная идея *состоит в том, что данную функцию $f(x)$ заменяют аппроксимирующим интерполяционным полиномом $P_n(x)$ степени n с узлами интерполяции $a = x_0 < x_1 < x_2 < \ldots < x_n = b$, а затем полагают* [1, 3]:

$$f^{(k)}(x) \approx P_n^{(k)}(x),\ a \leqslant x \leqslant b,\ 0 \leqslant k \leqslant n \qquad (8.11)$$



Заметим, что приближенное дифференцирование представляет собой операцию менее точную, чем интерполирование (см. рис. 8.2). Действительно, близость друг к другу ординат двух кривых $f(x)$ и $P_n(x)$ и даже совпадение значений $f(x_i)$ и $P_n(x_i)$ в узлах x_i еще не гарантирует близости на этом отрезке их производных $f'(x)$ и $P_n'(x)$. Например, в точке $x = x_i$ производные $f'(x_i)$ и $P_n'(x_i)$ имеют разные знаки, в то время как сами значения $f(x_i)$ и $P_n(x_i)$ совпадают.

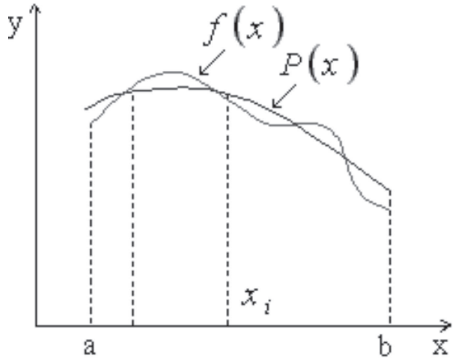


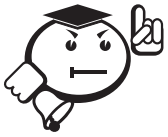
Рис. 8.2

Погрешность дифференцирования равна [3, 8]:

$$\left|f^{(k)}(x) - P_n^{(k)}(x)\right| = \left|\frac{d^k}{dx^k}R_n(x)\right| \leqslant \frac{M_{n+1}}{(n+1)!}\left|\frac{d^k}{dx^k}\omega_{n+1}(x)\right|, \tag{8.12}$$

где $\omega_{n+1}(x) = \prod_{i=0}^{n}(x - x_i) = (x - x_0)(x - x_1)\cdots(x - x_n)$, $M_{n+1} = \max\limits_{a\leqslant x\leqslant b}\left|f^{(n+1)}(x)\right|$.

Если производная $f^{(n+1)}(x)$ неизвестна, то можно использовать оценку $f^{(n+1)}(x) \approx \dfrac{\Delta^{(n+1)}y_0}{h^{n+1}}$ (в случае, если задана равномерная сетка). Здесь $\Delta^{(n+1)}y_0$ — конечная разность $(n+1)$-го порядка в точке y_0.



Заметим, что из формулы Ньютона с разделенными разностями следует, что $P_n^{(n)}(x) = n!f(x_0, x_1, \ldots, x_n)$. Поэтому справедлива приближенная формула:

$$f_n^{(n)}(x) \approx n!f(x_0; x_1; \ldots; x_n), \tag{8.13}$$

имеющая первый порядок точности.

Отсюда следует, что в формуле (8.12) вместо $f^{(n+1)}(x)$ мы можем использовать $f^{(n+1)}(x) \approx (n+1)! \cdot f(x_0; x_1; \ldots; x_{n+1})$, где $f(x_0; x_1; \ldots; x_{n+1})$ — разделенная разность $(n+1)$-го порядка в точке y_0.

Частными случаями (8.13) являются следующие формулы:

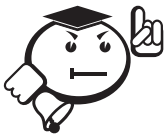
$$f'(x) \approx f(x_0; x_1) = \frac{f(x_1) - f(x_0)}{x_1 - x_0}, \tag{8.14}$$

$$f''(x) \approx 2f(x_0; x_1; x_2) = \frac{2}{x_2 - x_0}\left[\frac{f(x_2) - f(x_1)}{x_2 - x_1} - \frac{f(x_1) - f(x_0)}{x_1 - x_0}\right]. \tag{8.15}$$

При выборе в качестве узлов интерполяции значений $x_0 = x$, $x_1 = x + h$ формула (8.14) переходит в формулу (8.1). Если взять $x_0 = x - h$, $x_1 = x$, то из (8.14) получим (8.2), а если положить $x_0 = x - h$, $x_1 = x + h$, то получим (8.6). Аналогично, из (8.15) можно получить формулу (8.8).

Использование сетки с постоянным шагом

При использовании равномерной сетки формулы численного дифференцирования принимают наиболее простой вид.



Так, формула (8.13) будет иметь вид:

$$f^{(n)}(x) \approx \frac{\Delta^n y_0}{h^n}.$$

В тех случаях, когда значение производной нужно вычислять в крайних точках x_0 и x_n, используются *односторонние формулы численного дифференцирования*: $f^{(k)}(x_0) \approx P_n^{(k)}(x_0)$ и $f^{(k)}(x_n) \approx P_n^{(k)}(x_n)$. Получим односторонние формулы для первой производной путем дифференцирования полинома Ньютона с конечными разностями [1]:

$$f'(x_0) \approx \frac{1}{h}\sum_{j=1}^{n}\frac{(-1)^{j-1}}{j}\Delta^j y_0, \quad f'(x_n) \approx \frac{1}{h}\sum_{j=1}^{n}\frac{1}{j}\nabla^j y_n \tag{8.16}$$

имеющие n-й порядок точности.

При $n = 2$ (8.16) получаем формулы:

$$f'(x_0) \approx \frac{1}{2h}\bigl(-3f(x_0) + 4f(x_1) - f(x_2)\bigr), \tag{8.17}$$

$$f'(x_n) \approx \frac{1}{2h}\bigl(f(x_{n-2}) - 4f(x_{n-1}) + 3f(x_n)\bigr), \tag{8.18}$$

имеющие второй порядок точности по h.

Используя первую и вторую формулы Ньютона, можно получить формулы численного дифференцирования для внутренних точек сетки более высокого порядка точности. Пусть $n = 2$. Полиномы Ньютона для интерполирования вперед и назад имеют вид:

$$P_2(x) = y_0 + \frac{\Delta y_0}{1!}q + \frac{\Delta^2 y_0}{2!}q(q-1), \quad q = (x - x_0)/h;$$

$$\overline{P}_2(x) = y_2 + \frac{\nabla y_2}{1!}t + \frac{\nabla^2 y_2}{2!}t(t+1), \quad t = (x - x_2)/h, \quad \nabla y_2 = \Delta y_1, \quad \nabla^2 y_2 = \Delta^2 y_0.$$

Продифференцируем эти полиномы и вычислим в точке $x = x_1$ ($q = 1$, $t = -1$). В качестве приближенного значения первой производной возьмем среднеарифметическое

$$f'(x_1) \approx \frac{1}{2}\Bigl(P'_2(x_1) + \overline{P}'_2(x_1)\Bigr) = \frac{1}{2h}\left\{(\Delta y_0 + \Delta y_1) + \frac{\Delta^2 y_0 - \Delta^2 y_0}{2}\right\} = \frac{y_2 - y_0}{2h}.$$

Положим $x = x_1$, $x_0 = x - h$, $x_2 = x + h$. В результате получим:

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}.$$

Эта формула совпадает с формулой (8.6), полученной ранее, и имеет второй порядок точности по параметру h (см. формулу (8.7)).

Положим $n = 4$. Тогда полиномы Ньютона для интерполирования вперед и назад имеют вид:

$$P_4(x) = y_0 + \frac{\Delta y_0}{1!}q + \frac{\Delta^2 y_0}{2!}q(q-1) + \frac{\Delta^3 y_0}{3!}q(q-1)(q-2) + \frac{\Delta^4 y_0}{4!}q(q-1)(q-2)(q-3),$$

$$\overline{P}_4(x) = y_4 + \frac{\nabla y_4}{1!}t + \frac{\nabla^2 y_4}{2!}t(t+1) + \frac{\nabla^3 y_4}{3!}t(t+1)(t+2) + \frac{\nabla^4 y_4}{4!}t(t+1)(t+2)(t+3),$$

$$\nabla y_4 = \Delta y_3, \quad \nabla^2 y_4 = \Delta^2 y_2, \quad \nabla^3 y_4 = \Delta^3 y_1, \quad \nabla^4 y_4 = \Delta^4 y_0.$$

Вычислим производные полиномов Ньютона в точке $x = x_2$. В качестве приближенного значения первой производной в точке $x = x_2$ возьмем также среднеарифметическое, получим:

$$f'(x_2) \approx \frac{1}{2}\left(P_4'(x_2) + \overline{P}_4'(x_2)\right) = \frac{1}{2h}\left\{-\frac{1}{6}\Delta y_3 + \frac{7}{6}\Delta y_2 + \frac{7}{6}\Delta y_1 - \frac{1}{6}\Delta y_0\right\} = \frac{y_0 - 8y_1 + 8y_3 - y_4}{12h}.$$

Положим $x = x_2$, $x_0 = x - 2h$, $x_1 = x - h$, $x_3 = x + h$, $x_4 = x + 2h$. В результате получим:

$$f'(x) \approx \frac{f(x-2h) - 8f(x-h) + 8f(x+h) - f(x+2h)}{12h}. \tag{8.19}$$

Формула (8.19) имеет четвертый порядок точности по параметру h. Покажем это. Используя формулы Тейлора

$$f(x \pm h) = f(x) \pm f'(x)h + \frac{f''(x)}{2}h^2 \pm \frac{f^{(3)}(x)}{3!}h^3 + \frac{f^{(4)}(x)}{4!}h^4 \pm \frac{f^{(5)}(\xi_\pm)}{5!}h^5,$$

$$f(x \pm 2h) = f(x) \pm f'(x)2h + \frac{f''(x)}{2}4h^2 \pm \frac{f^{(3)}(x)}{3!}8h^3 + \frac{f^{(4)}(x)}{4!}16h^4 \pm \frac{f^{(5)}(\tilde{\xi}_\pm)}{5!}32h^5,$$

получим:

$$r(x,h) = f'(x) - \frac{f(x-2h) - 8f(x-h) + 8f(x+h) - f(x+2h)}{12h} =$$

$$= \frac{1}{12h} \cdot \frac{h^5}{5!}\left\{-32f^{(5)}(\tilde{\xi}_-) + 8f^{(5)}(\xi_-) + 8f^{(5)}(\xi_+) - 32f^{(5)}(\tilde{\xi}_+)\right\}.$$

Оценка сверху равна

$$\left|r(x,h)\right| \leqslant \frac{h^4}{18}M_5, \quad M_5 = \max_{[x-2h,\,x+2h]}\left|f^{(5)}(\xi)\right|. \tag{8.20}$$

Аналогично можно получить приближенную формулу для второй производной [1]. Она имеет вид:

$$f''(x) \approx \frac{-f(x-2h) + 16f(x-h) - 30f(x) + 16f(x+h) - f(x+2h)}{12h^2}. \tag{8.21}$$

Оценка верхней границы погрешности второй производной

$$r(x,h) = \frac{64}{6! \cdot 12h^2}h^6\left[f^{(6)}(\tilde{\xi}_+) + f^{(6)}(\tilde{\xi}_-)\right] - \frac{16}{6! \cdot 12h^2}h^6\left[f^{(6)}(\xi_+) + f^{(6)}(\xi_-)\right]$$

равна

$$\left|r(x,h)\right| \leqslant \frac{1}{54}h^4 M_6, \quad M_6 = \max_{[x-2h,\,x+2h]}\left|f^{(6)}(\xi)\right|. \tag{8.22}$$

Таким образом, формула (8.21) имеет четвертый порядок точности по параметру h.

Численное дифференцирование на основе сплайнов

Применение формулы (8.10) для вычисления производной $f^{(k)}(x)$ основано на кусочно-полиномиальной интерполяции. Полученная таким образом производная в точке «стыка» будет иметь разрыв. Поэтому если требуется получить глобальную аппроксимацию производной на промежутке $[a,b]$, то лучше использовать сплайны.

8.3 Численное дифференцирование на основе кубических сплайнов

Предположим, мы построили кубический сплайн для функции $f(x)$, заданной таблично в точках x_i $(i = 1,\ 2,\ \ldots,\ n)$ (см. п. 7.12.3). Пусть $h_i = x_{i+1} - x_i$ — шаг сетки $a = x_1 < x_2 < \ldots < x_n = b$ на интервале $[a,b]$. Тогда

$$f(x) \approx S(x),$$

где $S(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3$ для $x_i \leqslant x \leqslant x_{i+1}$.

Здесь a_i, b_i, c_i, d_i — коэффициенты сплайна:

$$a_i = y_i; \quad c_i = \frac{M_i}{2}; \quad d_i = \frac{(M_{i+1} - M_i)}{6h_i}.$$

В качестве первой производной функции берем

$$y'(x) \approx S'(x),$$

и в частности, в узлах сетки имеем

$$y'(x_i) = b_i = \frac{y_{i+1} - y_i}{h_i} - \frac{h_i}{6}(M_{i+1} + 2M_i),$$

где M_i — значения вторых производных сплайна, которые находятся из решения следующей системы [16]:

$$Am = Hy,$$

где $\{m\}_i = M_{i+1}$, $i = 1,\ \ldots,\ n-2$, m — $(n-2)$-мерный вектор; H — верхняя трехдиагональная положительно определенная матрица размера $(n-2) \times n$ с элементами

$$H_{j,\,j+1} = -\frac{1}{h_j} - \frac{1}{h_{j+1}}; \quad H_{j,\,j+2} = \frac{1}{h_{j+1}}; \quad j = 1,\ n-2; \quad H_{j,\,j} = \frac{1}{h_j};$$

A — трехдиагональная положительно определенная матрица $(n-2)\times(n-2)$.

$$A_{j,\,j} = \frac{(h_j + h_{j+1})}{3};\quad j = 1,\, n-2;$$

$$A_{j,\,j+1} = A_{j+1,\,j} = \frac{h_{j+1}}{6},\quad j = 1,\, n-3.$$

Здесь приняты граничные условия $M_1 = M_n = 0$.

Если значения функции сильно «зашумлены» случайными ошибками, то необходимо использовать сглаживающие сплайны.

8.4 Обусловленность формул численного дифференцирования

Несмотря на внешнюю простоту формул численного дифференцирования, их применение требует особой осторожности. Ранее было отмечено, что приближенное дифференцирование представляет собой операцию менее точную, чем интерполирование. При вычислении производных по формулам Ньютона необходимо предварительно рассчитать коэффициенты интерполяционных полиномов, которые представляют собой конечные либо разделенные разности значений функций. В обоих случаях приходится вычитать числа y_i, и если они близки, то вычитание приводит к уничтожению первых значащих цифр, то есть к потере части верных знаков числа. Кроме того, значения функции $f(x)$ могут быть отягощены погрешностями δy_i, если они получены в эксперименте. Возникает вопрос — останется ли в ответе хоть один достоверный знак? Поэтому к погрешности аппроксимации формул численного дифференцирования добавляется неустранимая погрешность, вызванная погрешностями вычисления функции. Для того чтобы погрешность аппроксимации была достаточно малой, требуется использование таблиц с малыми шагами h. Однако, к сожалению, при малых шагах формулы численного дифференцирования становятся плохо обусловленными и результат их применения может быть полностью искажен неустранимой погрешностью. Важно понимать, что действительная причина этого явления лежит не в несовершенстве предложенных методов вычисления производных, а в некорректности самой операции дифференцирования приближенно заданной функции.

Для понимания сути проблемы, рассмотрим линейный полином

$$P(x) = y_0 + \frac{y_1 - y_0}{x_1 - x_0}(x - x_0) = y_0 + \frac{y_1 - y_0}{h}(x - x_0),$$

где $h = x_1 - x_0$ — шаг сетки. Производная определяется как

$$P'(x) = \frac{1}{h}(y_1 - y_0) \tag{8.23}$$

с погрешностью $r_{\text{н}}(x,h) \leqslant \frac{1}{h}\left[\Delta(y_0) - \Delta(y_1)\right]$. Здесь $\Delta(y_i)$ — абсолютная погрешность измерения y_i, $i = 0,\, 1$.

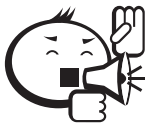
Таким образом, неустранимая погрешность оценивается следующим образом:

$$\delta P'(h) = \left|r_{\text{н}}(h)\right| \leqslant \frac{2}{h}\left|\delta y_m\right|, \tag{8.24}$$

где $\left|\Delta(y_m)\right| = \max\limits_i \left|\Delta(y_i)\right|$.

Из погрешности аппроксимации функции $R(x) = \frac{1}{2!}(x - x_0)(x - x_1) f''(\xi)$ определим погрешность аппроксимации производной $R'(x) = \frac{1}{2}(2x - x_1 - x_0) f''(\xi)$. Выражение в круглых скобках изменяется от $-h$ до $+h$ при изменении x от x_0 до x_1. Поэтому для верхней оценки погрешности получим:

$$\left|R'(h)\right| \leqslant \frac{1}{2} h \cdot M_2, \quad M_2 = \max_{[x_0, x_1]} \left|f''(\xi)\right| \tag{8.25}$$



Оценка (8.24) означает, что чувствительность формулы (8.23) к погрешностям входных данных характеризуется абсолютным числом обусловленности $\nu = 2/h$ и при малых h формула (8.24) становится плохо обусловленной. Поэтому, несмотря на то, что погрешность аппроксимации (8.25) при $h \to 0$ стремится к нулю, полная погрешность

$$r = \frac{2}{h}\left|\Delta(y_m)\right| + \frac{1}{2} h \cdot M_2 \tag{8.26}$$

будет неограниченно возрастать.

На рисунке 8.3 приведена зависимость погрешности производной интерполяционной формулы $R'(h)$ и погрешности, обусловленной ошибками определения значений функции $\Delta\big(P'(h)\big) = \left|r_{\text{н}}(h)\right|$.

Если бы значения функции y_i были известны точно, то есть $\left|\Delta(y_m)\right| = 0$, то мы бы имели $r \sim h$, и чем меньше шаг h, тем меньше погрешность дифференцирования.

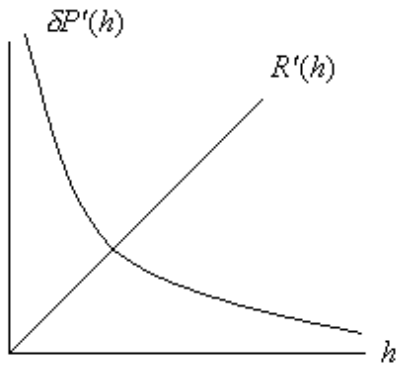


Рис. 8.3

С появлением составляющей $\frac{2}{h}|\Delta(y_m)|$ с уменьшением шага начинает доминировать именно эта составляющая и при $h \to 0$, $r \to \infty$.

Это свойство известно в математике как *некорректность* задачи дифференцирования.

Вычислим шаг, при котором $r \to \min$.

Пусть $M_2 \neq 0$. Продифференцируем (8.26) по h и приравняем нулю

$$r' = -\frac{2}{h^2}|\Delta(y_m)| + \frac{1}{2}M_2 = 0,$$

получим:

$$h_m = 2\sqrt{\frac{|\Delta(y_m)|}{M_2}}, \tag{8.27}$$

$$r_m = 2\sqrt{|\Delta(y_m)| \cdot M_2}. \tag{8.28}$$

Формула (8.27) определяет оптимальный шаг h_m сетки $\{x_i\}$, на которой должны быть заданы значения исследуемой функции $y = f(x)$. Чем больше погрешность измерения $|\Delta(y_m)|$, тем больше следует брать шаг.

Процедура выбора (8.27) *называется* ***регуляризацией дифференцирования по шагу***.

Существуют и другие способы регуляризации задачи дифференцирования, например дифференцирование с помощью сглаживающих сплайнов [16].

Контрольные вопросы по главе 8

1. Какая операция точнее — интерполирование или численное дифференцирование? Дайте пояснение.
2. Запишите правую и левую приближенные формулы для первой производной. Каков порядок точности этих формул?
3. Запишите формулу для центральной разностной производной. Каков порядок точности этой формулы?
4. Чему равна погрешность приближенных формул для первой производной?
5. Запишите приближенную формулу для второй производной и оцените ее погрешность и порядок точности.
6. В чем состоит идея общего способа получения формул численного дифференцирования?
7. Запишите формулу для первой производной четвертого порядка точности.

8. Запишите формулу для второй производной четвертого порядка точности.
9. Какие коэффициенты кубического сплайна можно использовать в качестве первой производной приближаемой функции?
10. Почему задача численного дифференцирования приближенно заданной функции является некорректной?
11. Что такое регуляризация дифференцирования по шагу?

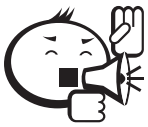
Глава 9

ЧИСЛЕННОЕ ИНТЕГРИРОВАНИЕ ФУНКЦИЙ

Если функция $f(x)$ непрерывна на $[a,b]$ и известна ее первообразная $F(x)$, то определенный интеграл от этой функции может быть вычислен по формуле Ньютона—Лейбница:

$$\int_a^b f(x)\,dx = F(b) - F(a), \tag{9.1}$$

где $F'(x) = f(x)$.



Однако во многих случаях $F(x)$ не может быть найдена с помощью элементарных функций или является слишком сложной; поэтому вычисление интеграла (9.1) может быть практически невыполнимым. Кроме того, на практике подынтегральная функция $f(x)$ часто задается таблично и тогда понятие первообразной теряет смысл. Поэтому важное значение имеют численные методы вычисления определенных интегралов.

Суть численного интегрирования заключается в вычислении значения (9.1) *на основании ряда значений* $f(x_i)$, $i = 0, \ldots, n$. *Для этого используют* ***квадратурные формулы*** *— приближенные равенства вида*

$$\int_a^b f(x)dx \approx \sum_{i=0}^{n} A_i f(x_i). \tag{9.2}$$

Здесь x_i *— узлы квадратурной формулы* ($x_i \in [a,b]$); A_i *— числовые коэффициенты (веса квадратурной формулы).*

Сумма $\sum_{i=0}^{n} A_i f(x_i)$*, которая принимается за приближенное значение интеграла, называется* ***квадратурной суммой****.*

Величина $R = \int_a^b f(x)dx - \sum_{i=0}^{n} A_i f(x_i)$ *называется* ***погрешностью*** *(или* ***остаточным членом****) квадратурной формулы.*

9.1 Квадратурные формулы Ньютона—Котеса

Пусть для данной функции $y = f(x)$ требуется вычислить интеграл $\int_a^b f(x)\,dx$.

Зададим шаг $h = \dfrac{b-a}{n}$ и разобьем $[a, b]$ на n отрезков с помощью равноотстоящих точек $x_i = a + ih$, $i = 0, \ldots, n$.

Пусть $y_i = f(x_i)$. Заменяя функцию $f(x)$ интерполяционным полиномом Лагранжа $L_n(x)$, придем к квадратурной формуле вида [3]:

$$\int_{x_0}^{x_n} f(x)dx \approx \int_{x_0}^{x_n} L_n(x)dx = \sum_{i=0}^{n} A_i y_i. \tag{9.3}$$

Получим явные выражения для коэффициентов A_i формулы (9.3).

С помощью подстановки

$$q = \frac{x - x_0}{h} \tag{9.4}$$

полином Лагранжа примет вид:

$$L_n(x) = \sum_{i=0}^{n} \frac{(-1)^{n-i}}{i!\,(n-i)!} \cdot \frac{\omega_{n+1}(q)}{q-i} y_i, \tag{9.5}$$

где $\omega_{n+1}(q) = q(q-1)(q-2)\ldots(q-n)$.

Подставим $L_n(x)$ в (9.3) и приравняем сомножители при y_i. В результате получим для A_i:

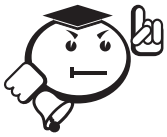
$$A_i = \int_{x_0}^{x_n} \frac{(-1)^{n-i}}{i!\,(n-i)!} \cdot \frac{\omega_{n+1}\big(q(x)\big)}{q-i} dx = h\frac{(-1)^{n-i}}{i!\,(n-i)!} \int_0^n \frac{\omega_{n+1}(q)}{q-i} dq. \tag{9.6}$$

Подставим в (9.6) $h = \dfrac{b-a}{n}$ и введем величины H_i (коэффициенты *Котеса*)

$$H_i = \frac{1}{b-a} A_i,$$

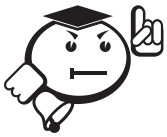
где

$$H_i = \frac{1}{n} \cdot \frac{(-1)^{n-i}}{i!\,(n-i)!} \int_0^n \frac{\omega_{n+1}(q)}{q-i} dq,\ i = 0, \ldots, n \qquad (9.7)$$



Тогда (9.3) примет вид

$$\int_a^b f(x)dx = (b-a) \sum_{i=0}^{n} H_i y_i, \quad y_i = f(x_0 + ih). \qquad (9.8)$$



Справедливы соотношения:

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

Первое свойство $\sum_{i=0}^{n} H_i = 1$ получается следующим образом. Если в формуле (9.6) положить $y \equiv 1$, то получим:

$$\int_a^b dx = (b-a) \sum_{i=0}^{n} H_i \to \sum_{i=0}^{n} H_i = 1.$$

Второе свойство доказывается непосредственными вычислениями.

9.2 Формула трапеций

Пусть $n = 1$ (функция задана на элементарном интервале $[x_0, x_1]$ в двух точках). Тогда из формулы (9.7) получаем:

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

Отсюда

$$\int_{x_0}^{x_1} f(x)\,dx = \frac{h}{2}(y_0 + y_1). \qquad (9.9)$$

Это известная ***формула трапеций****.*

Остаточный член (погрешность) $R = \int\limits_{x_0}^{x_1} f(x)\,dx - \frac{h}{2}(y_0 + y_1)$, *где* $y = f(x)$ *(см. рис. 9.1).*

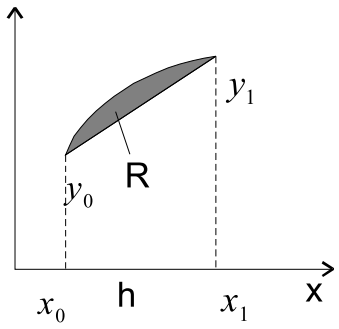
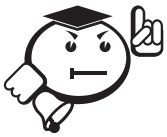


Рис. 9.1



Можно показать, что

$$R(h) = -\frac{h^3}{12} f''(\xi). \qquad (9.10)$$

Получим теперь формулу трапеций для $\int\limits_a^b f(x)\,dx$, т. е. для функции $f(x)$, заданной на произвольном интервале $[a, b]$. Пусть задана сетка $\{x_i\}$, где $x_i = a + ih$, $i = 0, \ldots, m$.

Тогда интеграл $\int\limits_a^b f(x)\,dx$ *можно записать в виде* ***составной формулы трапеций****:*

$$\int_a^b f(x)\,dx = \sum_{i=0}^{m-1} \int_{x_i}^{x_{i+1}} f(x)\,dx = \frac{h}{2}\left[y_0 + 2(y_1 + y_2 + \ldots + y_{m-1}) + y_m\right]. \qquad (9.11)$$

Погрешность *примет вид:*

$$R = -\frac{mh^3}{12} f''(\xi) = -\frac{(b-a)}{12} h^2 f''(\xi),\ \xi \in (a, b). \qquad (9.12)$$

Геометрически формула (9.11) получается, если график функции $y = f(x)$ заменить ломаной (см. рис. 9.2).

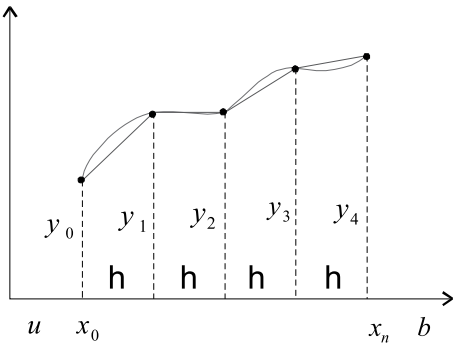
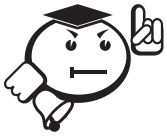


Рис. 9.2

Из формул (9.10) и (9.12) видно, что если $f'' > 0$, то формула трапеции (9.9), (9.11) даст значение интеграла с *избытком*, если $f'' < 0$, то — с *недостатком*.



Если сетка неравномерная, то вместо формулы (9.11) будем иметь:

$$\int_a^b f(x)\,dx = \sum_{i=1}^{m} \frac{h_i}{2}(y_{i-1} + y_i), \tag{9.13}$$

$$R(h) = -\frac{1}{12}\sum_{i=1}^{m} h_i^3 f''(\xi_i), \quad |R| \leqslant \frac{1}{12} M_2 \sum_{i=1}^{m} h_i^3. \tag{9.14}$$

9.3 Формула Симпсона

Из формулы (9.7) $H_i = \dfrac{1}{n} \cdot \dfrac{(-1)^{n-i}}{i!\,(n-i)!} \displaystyle\int_0^n \frac{\omega_{n+1}(q)}{q-i} dq$, $i = 0, \ldots, n$ при $n = 2$ получаем:

$$H_0 = \frac{1}{2} \cdot \frac{1}{2} \int_0^2 (q-1)(q-2)\,dq = \frac{1}{6},$$

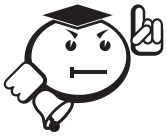
$$H_1 = -\frac{1}{2} \cdot \frac{1}{1} \int_0^2 q(q-2)\,dq = \frac{2}{3} = \frac{4}{6},$$

$$H_2 = \frac{1}{2} \cdot \frac{1}{2} \int_0^2 q(q-1)\,dq = \frac{1}{6}.$$

Так как $x_2 - x_0 = 2h$, *то имеем:*

$$\int_{x_0}^{x_2} f(x)\,dx = \frac{2h}{6}(y_0 + 4y_1 + y_2). \qquad (9.15)$$

Это формула ***Симпсона****.*



Геометрическая интерпретация дана на рис. 9.3. Кривую $y = f(x)$ мы заменяем параболой $y = L_2(x)$, проходящей через три точки: M_0, M_1, M_2.

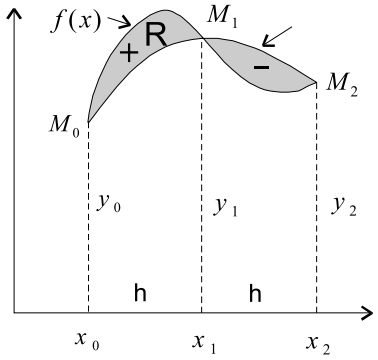
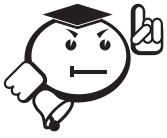


Рис. 9.3

Погрешность формулы Симпсона *равна*

$$R = \int_{x_0}^{x_1} f(x)\,dx - \frac{2h}{6}(y_0 + 4y_1 + y_2).$$



Запишем формулу для остаточного члена квадратурной формулы Симпсона:

$$R(h) = -\frac{h^5}{90} f^{IV}(\xi), \text{ где } \xi \in (x_0, x_2). \qquad (9.16)$$

Из (9.16) видно, что формула Симпсона является точной для полиномов не только второй, но и третьей степени.

Получим теперь формулу Симпсона для произвольного интервала $[a, b]$. Пусть $m = 2p$ есть четное число узлов сетки $\{x_i\}$, $x_i = a + ih$, $i = 0, \ldots, n$,

$$h = \frac{b-a}{m} = \frac{b-a}{2p}$$

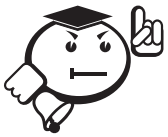
и $y_i = f(x_i)$.

Применяя формулу Симпсона (9.15) *к каждому удвоенному промежутку* $[x_0, x_2]$, $[x_2, x_4]$, …, $[x_{2p-2}, x_{2p}]$ *длины* $2h$, *будем иметь общую (или* ***составную****) формулу Симпсона:*

$$\int_a^b f(x)\,dx = \int_{x_0}^{x_m} f(x)\,dx = \frac{2h}{6}\sum_{k=1}^{p}[y_{2k-2} + 4y_{2k-1} + y_{2k}]. \qquad (9.17)$$

Ошибка для каждого удвоенного промежутка $[x_{2k-2}, x_{2k}]$ $(k = 1, \ldots, p)$ дается формулой (9.16):

$$r_k = -\frac{h^5}{90} f^{IV}(\xi_k),\ \xi_k \in (x_{2k-2}, x_{2k}).$$



Тогда для интервала $[a, b]$ можно получить:

$$R(h) = -\frac{mh^5}{90} f^{IV}(\xi) = -\frac{(b-a)}{180} h^4 f^{IV}(\xi). \qquad (9.18)$$

Если задана предельно допустимая погрешность ε, то, обозначив

$$M_4 = \max_{\xi \in [a,b]} f^{IV}(\xi),$$

получим для определения шага h

$$h < \left[\frac{180 \cdot \varepsilon}{(b-a) M_4}\right]^{1/4}.$$

9.4 Квадратурная формула Гаусса

Приведем без вывода квадратурную формулу Гаусса.

Для произвольного интервала $[a,b]$ ***формула Гаусса*** *имеет вид:*

$$\int_a^b f(x)\,dx = \frac{(b-a)}{2}\sum_{i=1}^{n} A_i f(x_i), \tag{9.19}$$

где

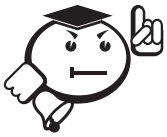
$$x_i = \frac{1}{2}(b+a) + \frac{1}{2}(b-a)\cdot t_i, \tag{9.20}$$

t_i *— нули полинома Лежандра.*

Коэффициенты квадратурной формулы A_i находим из решения системы линейных алгебраических уравнений:

$$\sum_{i=1}^{n} A_i t_i^k = \int_{-1}^{1} t^k dt = \frac{1-(-1)^{k+1}}{k+1} = \begin{cases} \dfrac{2}{k+1} & \text{при } k \text{ — четном,} \\ 0 & \text{при } k \text{ — нечетном,} \end{cases}, \; k = 1, \ldots, n. \tag{9.21}$$

Квадратурная формула Гаусса точна для всех полиномов наивысшей степени $N = 2n - 1$.



Остаточный член (погрешность) формулы Гаусса (9.29) с n узлами определяется следующим образом [3]:

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

В частности,

$$R_1 = \frac{(b-a)^3}{24} f''(\xi),$$

$$R_2 = \frac{1}{135}\left(\frac{b-a}{2}\right)^5 f^{(4)}(\xi),$$

$$R_3 = \frac{1}{15750}\left(\frac{b-a}{2}\right)^7 f^{(6)}(\xi) \text{ и т. д.}$$

В табл. 9.1 приведены значения t_i, A_i для $n = 1, 2, \ldots, 8$ [3].

Таблица 9.1

n	i	t_i	A_i
1	1	0	2
2	1; 2	±0.57735027	1
3	1; 3	±0.77459667	0.55555556
	2	0	0.88888889
4	1; 4	±0.86113631	0.34785484
	2; 3	±0.33998104	0.65214516
5	1; 5	±0.90617985	0.23692688
	2; 4	±0.53846931	0.47862868
	3	0	0.56888889
6	1; 6	±0.93246951	0.17132450
	2; 5	±0.66120939	0.36076158
	3; 4	±0.23861919	0.46791394
7	1; 7	±0.94910791	0.12948496
	2; 6	±0.74153119	0.27970540
	3; 5	±0.40584515	0.38183006
	4	0	0.41795918
8	1; 8	±0.96028986	0.10122854
	2; 7	±0.79666648	0.22238104
	3; 6	±0.52553242	0.31370664
	4; 5	±0.18343464	0.36268378

9.5 Квадратурная формула Чебышева

Формула Чебышева *имеет вид:*

$$\int_a^b f(x)\,dx = \frac{b-a}{n}\cdot\sum_{i=1}^{n} f(x_i),$$
$$x_i = \frac{1}{2}(b+a) + \frac{b-a}{2}t_i,\ i = 1, 2, \ldots, n, \tag{9.22}$$

где t_i — корни системы нелинейных уравнений:

$$\left.\begin{aligned} &t_1 + t_2 + \ldots + t_n = 0,\\ &t_1^2 + t_2^2 + \ldots + t_n^2 = \frac{n}{3},\\ &t_1^3 + t_2^3 + \ldots + t_n^3 = 0,\\ &\ldots\ldots\ldots\ldots\ldots\ldots\\ &t_1^n + t_2^n + \ldots + t_n^n = \frac{n\cdot\left[1-(-1)^{n+1}\right]}{2\cdot(n+1)} \end{aligned}\right\}. \tag{9.23}$$

Квадратурная формула (9.22) является точной для всех полиномов до степени n включительно.

В табл. 9.2 приведены значения t_i для $n = 2, 3, \ldots, 7$, найденные из решения системы (9.23) [3].

Таблица 9.2

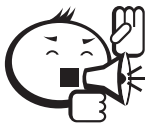
n	i	t_i	N	i	t_i
2	1; 2	0.577350	6	1; 6	0.866247
3	1; 3	0.707107		2; 5	0.422519
	2	0		3; 4	0.266635
4	1; 4	0.794654	7	1; 7	0.883862
	2; 3	0.187592		2; 6	0.529657
5	1; 5	0.832498		3; 5	0.323912
	2; 4	0.374541		4	0
	3	0			

9.6 Формула прямоугольников

Это самая простая квадратурная формула вычисления интеграла, в которой используется одно значение функции на $[x_0, x_1]$:

$$\int_{x_0}^{x_1} y\,dx = h \cdot y(\xi_0), \tag{9.24}$$

где $\xi_0 = \dfrac{x_0 + x_1}{2}$; $h = x_1 - x_0$.



Формула (9.24) представляет собой центральную формулу прямоугольников [1].

Вычислим остаточный член. Запишем формулу Тейлора для функции $y = f(x)$ в точке ξ_0:

$$f(x) = f(\xi_0) + f'(\xi_0)(x - \xi_0) + \frac{1}{2!} f''(\xi_1)(x - \xi_0)^2, \tag{9.25}$$

где $\xi_1 \in [x, \xi_0]$; $x \in [x_0, x_1]$.

Проинтегрируем (9.25):

$$\int_{x_0}^{x_1} f(x)\,dx = h \cdot f(\xi_0) + f'(\xi_0) \int_{x_0}^{x_1} (x - \xi_0)\,dx + \frac{1}{2!} \int_{x_0}^{x_1} f''\big(\xi_1(x)\big)(x - \xi_0)^2\,dx. \tag{9.26}$$

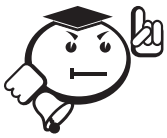
Второе слагаемое в (9.26) равно нулю. Таким образом, из (9.26) следует:

$$R = \frac{1}{2!} \int_{x_0}^{x_1} f''\big(\xi_1(x)\big)(x - \xi_0)^2\,dx.$$

Так как второй множитель подынтегрального выражения не меняет знак, то по теореме о среднем получим:

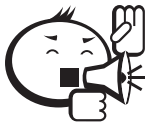
$$R = \frac{1}{2!} f''(\xi) \int_{x_0}^{x_1} (x - \xi_0)^2 dx,$$

где $\xi \in [x_0, x_1]$.



После интегрирования получим $R = \dfrac{(x_1 - x_0)^3}{24} f''(\xi)$ или, представляя R как функцию h, получим для погрешности:

$$R(h) = \frac{h^3}{24} f''(\xi). \tag{9.27}$$



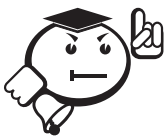
Сравнивая с остаточным членом формулы трапеций, мы видим, что погрешность формулы прямоугольников в два раза меньше, чем погрешность формулы трапеций. Этот результат верен, если в формуле прямоугольников мы берём значение функции в средней точке.

Получим остаточный член для интервала $[a, b]$. Пусть задана сетка $x_i = a + i \cdot h$, $i = 0, 1, \ldots, n$, $h = x_{i+1} - x_i = (b - a)\dfrac{1}{n}$.

Рассмотрим сетку $\xi_i = \xi_0 + i \cdot h$; $i = 1, 2, \ldots, n$; $\xi_0 = a - \dfrac{h}{2}$.

Тогда

$$\int_a^b f(x) dx \approx h \sum_{i=1}^{n} f(\xi_i). \tag{9.28}$$



Остаточный член для интервала $[a, b]$ равен:

$$R(h) = \frac{b - a}{24} h^2 f''(\xi). \tag{9.29}$$

Геометрически формула прямоугольников представлена на рисунке 9.4.

Если функция $f(x)$ задана таблично, то используют либо формулу *левых прямоугольников*

$$\int_a^b f(x)\, dx = h \sum_{i=0}^{n-1} y_i, \tag{9.30}$$

либо *правых прямоугольников*

$$\int_a^b f(x)\,dx = h\sum_{i=1}^{n} y_i. \tag{9.31}$$

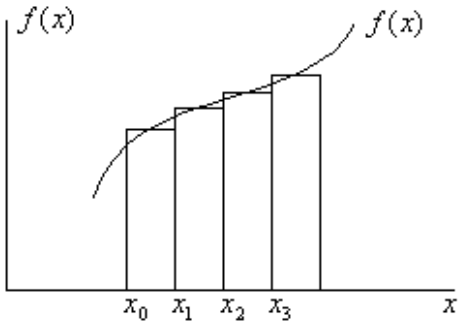


Рис. 9.4

Погрешность этих формул оценивается через первую производную. Для интервала $[x_0, x_1]$ имеем:

$$R(h) = y'(\xi)\frac{h^2}{2}, \quad x_0 < \xi < x_1. \tag{9.32}$$

Для произвольного интервала $[a, b]$ получим:

$$R(h) = \frac{(b-a)}{2}h \cdot y'(\xi), \quad a < \xi < b. \tag{9.33}$$

9.7 Обусловленность квадратурных формул

При вычислении интегралов часто приходится использовать не точные значения $f(x)$ подынтегральной функции, а приближенные значения $f^*(x)$. Пусть задана погрешность Δf подынтегральной функции $|f(x) - f^*(x)| \leqslant \Delta f$ для $x \in [a, b]$. Тогда для погрешности интеграла ΔI справедлива оценка:

$$\Delta I = |I - I^*| = \left| \int_a^b f(x)dx - \int_a^b f^*(x)dx \right| \leqslant (b-a)\Delta f.$$

Отсюда мы видим, что *абсолютное число обусловленности* $v_I = (b-a)$. Таким образом, задача вычисления определенного интеграла от приближенно заданной функции является устойчивой.

Найдем погрешность квадратурной формулы [1]:

$$\left| \sum_{i=0}^{n} A_i f(x_i) - \sum_{i=0}^{n} A_i f^*(x_i) \right| \leqslant \Delta f \sum_{i=0}^{n} |A_i|.$$

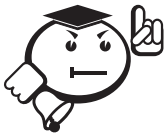
Таким образом, квадратурная формула устойчива к погрешностям задания функции и ее число обусловленности $v = \sum_{i=0}^{n} |A_i|$.

9.8 Правило Рунге оценки погрешности квадратурных формул

Пусть подынтегральная функция $f(x)$ задана на сетке $x_i = a + i \cdot h$, $i = 0, 1, 2, \ldots, n$, $h = (b - a)/n$ — шаг сетки. Пусть n будет четным числом. Обозначим за I^h значение интеграла, вычисленного по квадратурной формуле с использованием значений функции на сетке $\{x_i\}$, а за I^{2h} — значение интеграла, вычисленного на сетке $x_j = a + j \cdot h_1$, $j = 0, 1, \ldots, m$, $m = n/2$, $h_1 = 2h$. Тогда имеет место следующая приближенная оценка погрешности вычисления интеграла [1]:

$$\varepsilon = I - I^h = \frac{I^h - I^{2h}}{2^k - 1}. \tag{9.34}$$

Здесь $k = 2$ для формул центральных прямоугольников и трапеций и $k = 4$ для формулы Симпсона.



Используя формулу (9.34), можно строить процедуру вычисления интеграла с заданной точностью ε. Для этого последовательно дробят шаг сетки, вычисляют значения интеграла I^{h_r} и оценивают погрешность ε_r по формуле (9.34) для шага $h_r = h_0/2^r$, где h_0 — начальное значение шага, $r = 1, 2, \ldots$ Вычисления прекращают тогда, когда при некотором r выполняется неравенство $|\varepsilon_r| < \varepsilon$ (требуемая точность достигнута), либо тогда, когда величина $|\varepsilon_r|$ начинает возрастать (точность не может быть достигнута из-за влияния вычислительной погрешности).

Контрольные вопросы по главе 9

1. В каких случаях необходимо использовать численные методы интегрирования функций?
2. Запишите квадратурную формулу Ньютона—Котеса.
3. Запишите формулу трапеций для равномерной сетки.
4. Запишите остаточный член формулы трапеций.
5. Запишите квадратурную формулу Симпсона для равномерной сетки.
6. Запишите остаточный член формулы Симпсона.
7. Запишите квадратурную формулу Чебышева.
8. Запишите квадратурную формулу Гаусса.
9. Можно ли использовать формулы Чебышева или Гаусса для численного интегрирования таблично заданной функции?
10. Чему равно число обусловленности квадратурных формул прямоугольников, трапеций и Симпсона?
11. Чему равно число обусловленности квадратурных формул Чебышева и Гаусса?

Глава 10

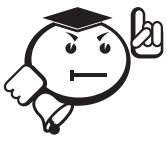
ЧИСЛЕННЫЕ МЕТОДЫ РЕШЕНИЯ ОБЫКНОВЕННЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ

10.1 Постановка задачи

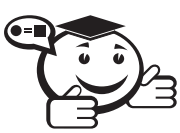
Обыкновенными дифференциальными уравнениями можно описать задачи движения системы взаимодействующих материальных точек, химической кинетики, электрических цепей, сопротивления материалов (например, статический прогиб упругого стержня) и многие другие.

Простейшим обыкновенным дифференциальным уравнением (ОДУ) *является уравнение первого порядка*

$$y'(x) = f\big(x, y(x)\big),\ x \in [a, b]. \tag{10.1}$$



Различают три основных типа задач для обыкновенных ДУ: задача Коши, краевые задачи и задачи на собственные значения.



Рассмотрим ***задачу Коши****. Необходимо найти решение* $y = y(x)$ *уравнения* (10.1)*, удовлетворяющее начальному условию:*

$$y(a) = y_0, \tag{10.2}$$

т. е. найти кривую $y(x)$, проходящую через заданную точку $A_0(a, y_0)$ (см. рис. 10.1).

График решения дифференциального уравнения называют ***интегральной кривой****. Процесс нахождения решений дифференциального уравнения называют интегрированием этого уравнения.*

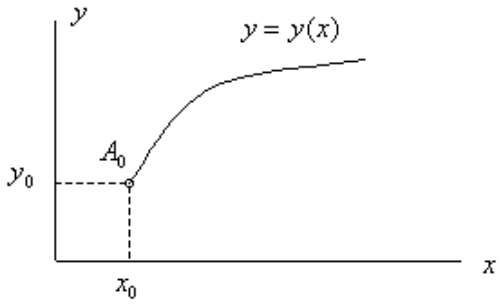


Рис. 10.1

Разрешимость задачи Коши

Пусть R — множество точек (x, y), удовлетворяющих условию $a \leqslant x \leqslant b$, $c \leqslant y \leqslant d$. Имеет место теорема [1, 2].

Теорема 10.1. Пусть правая часть $f(x, y)$ непрерывна в области R. Предположим также, что она удовлетворяет условию Липшица

$$\left|f(x, y_1) - f(x, y_2)\right| \leqslant L\,|y_1 - y_2| \tag{10.3}$$

для всех $a \leqslant x \leqslant b$ и произвольных y_1, y_2 из R — области, где L — постоянная Липшица $L = \max\limits_{y}\left|f'_y(x, y)\right|$, $(x, y) \in R$. Тогда существует единственное решение $y(x)$ задачи Коши (10.1), (10.2), определенное на отрезке $[a, b]$.

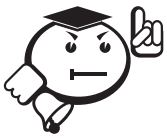
Для дифференциального уравнения n-го порядка

$$y^{(n)} = f\left(x, y, y', y'', \ldots, y^{(n-1)}\right) \tag{10.4}$$

задача Коши состоит в нахождении решения $y = y(x)$, удовлетворяющего условиям:

$$y(a) = y_0;\ y'(a) = y'_0;\ \ldots;\ y^{(n-1)}(a) = y_0^{(n-1)},$$

где y_0, y'_0, …, $y_0^{(n-1)}$ — заданные числа.



Известно, что дифференциальное уравнение n-го порядка (10.4) может быть сведено к системе ДУ первого порядка при помощи замены $y^{(k)}(x) = y_{k+1}(x)$, $k = 0, 1, \ldots, n-1$. Тогда получим:

$$\begin{cases} y_1'(x) = y_2(x), \\ y_2'(x) = y_3(x), \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ y_n'(x) = f(x, y_1, y_2, \ldots, y_n), \end{cases} \tag{10.5}$$

где $y_1(x) = y(x)$.

В векторном виде система (10.5) будет иметь вид:

$$y'(x) = F\big(x, y(x)\big),$$

где

$$y'(x) = \begin{pmatrix} y_1' \\ y_2' \\ \ldots \\ y_{n-1}' \\ y_n' \end{pmatrix}; \quad y(x) = \begin{pmatrix} y_1 \\ y_2 \\ \ldots \\ y_{n-1} \\ y_n \end{pmatrix}; \quad F = \begin{pmatrix} y_2 \\ y_3 \\ \ldots \\ y_n \\ f \end{pmatrix}$$

Пример 10.1

Колебания маятника в среде с сопротивлением описываются следующим нелинейным ДУ второго порядка:

$$\frac{d^2\theta}{dt^2} + \alpha\frac{d\theta}{dt} + \beta\sin\theta = 0, \tag{10.6}$$

где θ — угол отклонения, $\alpha\dfrac{d\theta}{dt}$ — сопротивление среды (α, β — const).

Начальные условия (НУ): $\theta(t_0) = \theta_0$; $\left.\dfrac{d\theta}{dt}\right|_{t=t_0} = \theta'_0$. Вводя обозначения $\theta = y_1$; $y_1' = y_2$, получим вместо (10.6) следующую систему:

$$\begin{cases} y_1' = y_2 \\ y_2' = -\alpha \cdot y_2 - \beta \cdot \sin y_1 \end{cases} \tag{10.7}$$

с НУ $y_1(t_0) = \theta_0$; $y_2(t_0) = \theta'_0$.

В общем случае система ОДУ может быть представлена в форме:

$$\begin{cases} y_1' = f_1(x, y_1, \ldots, y_n), \\ y_2' = f_2(x, y_1, \ldots, y_n), \\ \ldots\ldots\ldots\ldots\ldots \\ y_n' = f_n(x, y_1, \ldots, y_n), \end{cases} \tag{10.8}$$

где x — независимая переменная; y_i — искомые функции.

Справедлива теорема [1].

Теорема 10.2 (существования и единственности). Пусть в некоторой окрестности граничных значений:

$$R: \left\{ |x-a| < A; \left| y_1 - y_1^{(0)} \right| < B_1; \ldots; \left| y_n - y_n^{(0)} \right| < B_n \right\}$$

система (10.8) обладает следующими свойствами:

1) правые части f_i, $i = 1, 2, \ldots, n$ определены и непрерывны в R;
2) функции f_i, $i = 1, 2, \ldots, n$ в R — окрестности удовлетворяют условиям Липшица по зависимым переменным y_i, т. е.

$$\left| f_i(x, \tilde{y}_1, \ldots, \tilde{y}_n) - f_i(x, y_1, \ldots, y_n) \right| \leqslant L \sum_{j=1}^{n} |\tilde{y}_j - y_j|, \qquad (10.9)$$

где $(x, y_1, \ldots, y_n) \in R$; $(x, \tilde{y}_1, \ldots, \tilde{y}_n) \in R$, а L — const. В этом случае существует единственное решение системы (10.8):

$$y_1 = y_1(x), \ \ldots, \ y_n = y_n(x),$$

определённое на отрезке $|x - a| < A$ и удовлетворяющее заданным начальным условиям:

$$y_1(a) = y_1^{(0)}; \ \ldots; \ y_n(a) = y_n^{(0)}.$$

10.2 Метод Эйлера

Это простейший численный метод. Рассмотрим задачу Коши:

$$y' = f(x, y), \quad a \leqslant x \leqslant b, \quad y(a) = y_0.$$

Зададим равномерную сетку $x_i = a + i \cdot h$, $i = 0, 1, \ldots, n$. Введём обозначения $y(x_i) = y_i$.

Для интервала $[x_i, x_{i+1}]$ *мы можем приближённо записать*

$$y_{i+1} = y_i + h \cdot f(x_i, y_i); \quad y(a) = y_0. \qquad (10.10)$$

Алгоритм (10.10) *называют* ***методом Эйлера***.

Метод Эйлера является *явным одношаговым* методом. Геометрическая интерпретация следующая: интегральная кривая $y(x)$, проходящая через точку $M_0(a, y_0)$, заменяется ломаной линией (см. рис. 10.2).

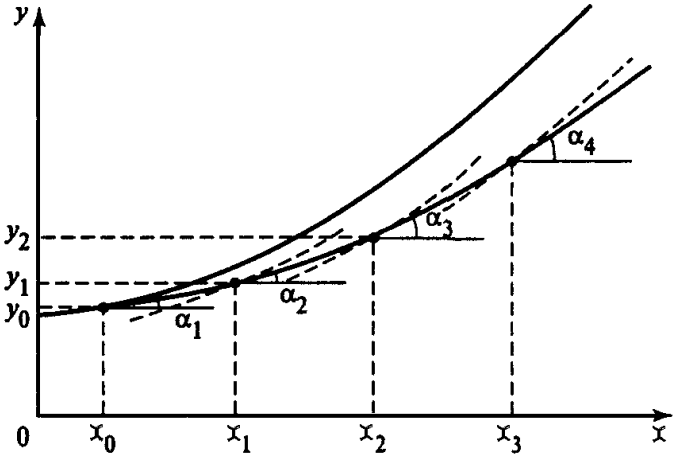


Рис. 10.2

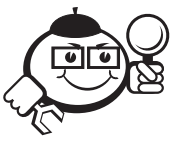
Погрешность аппроксимации имеет вид:

$$\psi_i = \frac{h}{2} y''(\xi_i).$$

Теорема 10.3 [1] (теорема устойчивости). Пусть функция f удовлетворяет условию $|f'_y| \leqslant L$. Тогда справедливо неравенство

$$\max_{0 \leqslant i \leqslant n} |y_i^* - y_i| \leqslant e^{L(b-a)} \left(|y_0^* - y_0| + h \sum_{k=0}^{n-1} |\psi_k| \right),$$

означающее, что метод Эйлера устойчив.



Пример 10.2

Найдите решение ОДУ

$$y' = x^2 + y^2, \quad 0 \leqslant x \leqslant 1, \quad y(0) = 0$$

методом Эйлера. Сравним решения, полученные при разном h:

$$y_{i+1} = y_i + h(x_i^2 + y_i^2)$$

Таблица 10.1

x_n	y_n				$y(x)$
	$h = 1$	$h = 0.5$	$h = 0.25$	$h = 0.1$	Точное решение
0.00	0.000	0.000	0.000	0.000	0.000
0.10				0.000	0.0003
0.20				0.001	0.003
0.25			0.000		0.005
0.30				0.005	0.009
0.40				0.014	0.021
0.50		0.000	0.016	0.030	0.042
0.60				0.055	0.072
0.70				0.091	0.116
0.75			0.078		0.143
0.80				0.141	0.174
0.90				0.207	0.250
1.00	0.000	0.125	0.220	0.292	0.350

. . . . . .

Видим, что при уменьшении шага решение улучшается.

Здесь $y(x)$ — точное решение, найденное методом Пикара.

10.3 Методы Рунге—Кутты

Семейство методов Рунге—Кутты определяются из следующих соотношений:

$$y(x+h) = y(x) + \sum_{i=1}^{q} p_i k_i(h), \tag{10.11}$$

где

$$\left.\begin{gathered} k_1(h) = h \cdot f(x,y), \\ k_2(h) = h \cdot f\big(x + \alpha_2 h, y + \beta_{21} k_1(h)\big), \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ k_q(h) = h \cdot f\big(x + \alpha_q h, y + \beta_{q1} k_1(h) + \beta_{q2} k_2(h) + \ldots + \beta_{qq-1} k_{q-1}(h)\big). \end{gathered}\right\} \tag{10.12}$$

Здесь $\alpha_2, \ldots, \alpha_q$; $p_1, p_2, \ldots, p_q$; β_{ij} — параметры; q — точность метода.

Рассмотрим выбор параметров α_i, p_i, β_{ij}. Введём функцию:

$$\varphi(h) = y(x+h) - y(x) - \sum_{i=1}^{q} p_i k_i, \tag{10.13}$$

которая представляет собой погрешность решения.

Если $f(x,y)$ — достаточно гладкая функция, то $k_i(h)$ и $\varphi(h)$ — гладкие функции параметра h. Предположим, что существуют производные $\varphi'(h)$, $\varphi''(h)$, …, $\varphi^{(q+1)}(h)$. Параметры α_i, p_i выбираем таким образом, чтобы выполнялись равенства:

$$\varphi(0) = \varphi'(0) = \ldots = \varphi^{(q)}(0) = 0.$$

Тогда, по формуле Тейлора справедливо равенство:

$$\varphi(h) = \sum_{i=0}^{q} \frac{\varphi^{(i)}(0)}{i!} h^i + \frac{1}{(q+1)!} \varphi^{(q+1)}(\theta \cdot h) \cdot h^{q+1} = \frac{\varphi^{(q+1)}(\theta \cdot h)}{(q+1)!} h^{q+1}, \tag{10.14}$$

где $\theta \in (0,1)$. Величину $\varphi(h)$ называют погрешностью метода на шаге, а q — порядком погрешности метода. При $q = 1$ из (10.13) можно получить $p_1 = 1$. Таким образом, при $q = 1$ из (10.11) следует метод Эйлера. Погрешность метода в соответствии с формулой (10.14) равна

$$\varphi(h) = \frac{y''(x + \theta \cdot h)}{2} h^2. \tag{10.15}$$

Следовательно, метод Эйлера имеет первый порядок точности.

При $q = 2$ можно получить две схемы. Зададим $p_1 = 1/2$. Тогда получим $p_2 = 1/2$; $\alpha_2 = 1$; $\beta_{21} = 1$ и приходим к схеме:

$$\left.\begin{aligned} \tilde{y}_{i+1} &= y_i + h \cdot f(x_i, y_i), \\ y_{i+1} &= y_i + \frac{h}{2}\left[f(x_i, y_i) + f(x_{i+1}, \tilde{y}_{i+1})\right]. \end{aligned}\right\} \tag{10.16}$$

Если задать $p_1 = 0$, то получим $p_2 = 1$; $\alpha_2 = 1/2$; $\beta_{21} = 1/2$. В результате получаем схему:

$$\left.\begin{aligned} \tilde{y}_{i+1/2} &= y_i + \frac{h}{2} f(x_i, y_i), \\ y_{i+1} &= y_i + h \cdot f\left(x_i + \frac{h}{2}, \tilde{y}_{i+1/2}\right). \end{aligned}\right\} \tag{10.17}$$

Схемы (10.16) и (10.17) имеют второй порядок точности.

Погрешность схем второго порядка точности равна (см. (10.14))

$$\varphi(h) = \frac{\varphi'''(\theta \cdot h)}{3!} h^3. \tag{10.18}$$

Пусть $q = 3$. Тогда можно получить следующие значения параметров:

$$p_1 = p_3 = \frac{1}{6};\ p_2 = \frac{4}{6};\ \alpha_2 = \frac{1}{2};\ \alpha_3 = 1;\ \beta_{21} = \frac{1}{2};\ \beta_{31} = -1;\ \beta_{32} = 2.$$

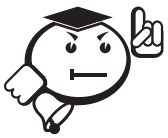
Расчётная схема будет иметь вид (см. (10.11), (10.12)):

$$\begin{cases} k_1 = h \cdot f(x_i, y_i), \\ k_2 = h \cdot f\left(x_i + \frac{h}{2}, y_i + \frac{1}{2} k_1\right), \\ k_3 = h \cdot f(x_i + h, y_i - k_1 + 2k_2), \\ y_{i+1} = y_i + \frac{1}{6}\left[k_1(x_i, y_i) + 4k_2(x_i, y_i) + k_3(x_i, y_i)\right]. \end{cases} \tag{10.19}$$

Эта схема имеет третий порядок точности.

При $q = 4$ получим схему четвёртого порядка точности:

$$\begin{cases} k_1 = h \cdot f(x_i, y_i), \\ k_2 = h \cdot f\left(x_i + \dfrac{h}{2}, y_i + \dfrac{1}{2}k_1(x_i, y_i)\right), \\ k_3 = h \cdot f\left(x_i + \dfrac{h}{2}, y_i + \dfrac{1}{2}k_2(x_i, y_i)\right), \\ k_4 = h \cdot f\big(x_i + h, y_i + k_3(x_i, y_i)\big), \\ y_{i+1} = y_i + \dfrac{1}{6}\big[k_1(x_i, y_i) + 2k_2(x_i, y_i) + 2k_3(x_i, y_i) + k_4(x_i, y_i)\big]. \end{cases} \tag{10.20}$$



. . . . . .

Схемы Рунге—Кутты имеют ряд достоинств:

1) все они устойчивы;
2) они являются явными, т. е. значения y_{i+1} вычисляются по ранее найденным значениям $y_1, y_2, \ldots, y_i$;
3) схемы допускают введение переменного шага h.

. . . . . .

10.4 Решение систем дифференциальных уравнений

Схемы Рунге—Кутты легко переносятся на системы ДУ. Рассмотрим систему уравнений n-го порядка:

$$\left.\begin{aligned} y_1'(x) &= f_1(x, y_1, y_2, \ldots, y_n), \\ y_2'(x) &= f_2(x, y_1, y_2, \ldots, y_n), \\ &\ldots\ldots\ldots\ldots \\ y_n'(x) &= f_n(x, y_1, y_2, \ldots, y_n). \end{aligned}\right\} \tag{10.21}$$

С начальными условиями

$$\left.\begin{aligned} y_1(x_0) &= y_1^0, \\ y_2(x_0) &= y_2^0, \\ &\ldots\ldots \\ y_n(x_0) &= y_n^0. \end{aligned}\right\} \tag{10.22}$$

Применим схему Рунге—Кутты первого порядка точности (метод Эйлера). В результате получим:

$$\left.\begin{aligned} y_1^{i+1} &= y_1^i + h \cdot f_1(x_i, y_1^i, \ldots, y_n^i), \\ y_2^{i+1} &= y_2^i + h \cdot f_2(x_i, y_1^i, \ldots, y_n^i), \\ &\ldots\ldots\ldots\ldots \\ y_n^{i+1} &= y_n^i + h \cdot f_n(x_i, y_1^i, \ldots, y_n^i). \end{aligned}\right\} \tag{10.23}$$

Здесь $y_j^i = y_j(x_i)$, $i = 0, 1, 2, \ldots$ — номер узла сетки.

Применяя схему Рунге—Кутты второго порядка точности, получим:

$$\left.\begin{aligned} &y_j^{i+1} = y_j^i + \frac{1}{2}\left[k_{j1}^i + k_{j2}^i\right],\ j = 1,\ 2,\ \ldots, n,\\ &\text{где } k_{j1}^i = h \cdot f_j\left(x_i, y_1^i, \ldots, y_n^i\right),\ j = 1,\ 2,\ \ldots,\ n,\\ &\quad k_{j2}^i = h \cdot f_j\left(x_i + h, y_1^i + k_{11}^i, \ldots, y_n^i + k_{n1}^i\right),\ j = 1,\ 2,\ \ldots,\ n. \end{aligned}\right\} \qquad (10.24)$$

Таким же образом можно записать схемы Рунге—Кутты третьего и четвёртого порядков точности. Например, для схемы четвёртого порядка точности получим:

$$\left.\begin{aligned} &y_j^{i+1} = y_j^i + \frac{1}{6}(k_{j1}^i + 2k_{j2}^i + 2k_{j3}^i + k_{j4}^i),\ j = 1,\ 2,\ \ldots,\ n,\\ &k_{j1}^i = h \cdot f_j(x_i, y_1^i, \ldots, y_n^i),\ j = 1,\ 2,\ \ldots,\ n,\\ &k_{j2}^i = h \cdot f_j\left(x_i + \frac{1}{2}h, y_1^i + \frac{1}{2}k_{11}^i, \ldots, y_n^i + \frac{1}{2}k_{n1}^i\right),\ j = 1,\ 2,\ \ldots,\ n,\\ &k_{j3}^i = h \cdot f_j\left(x_i + \frac{1}{2}h, y_1^i + \frac{1}{2}k_{12}^i, \ldots, y_n^i + \frac{1}{2}k_{n2}^i\right),\ j = 1,\ 2,\ \ldots,\ n,\\ &k_{j4}^i = h \cdot f_j\left(x_i + h, y_1^i + k_{13}^i, \ldots, y_n^i + k_{n3}^i\right),\ j = 1,\ 2,\ \ldots,\ n. \end{aligned}\right\}$$

10.5 Решение дифференциального уравнения *n*-го порядка

Рассмотрим уравнение *n*-го порядка, разрешённое относительно старшей производной:

$$y^{(n)} = f\left(x, y, y', y'', \ldots, y^{(n-1)}\right) \qquad (10.25)$$

с начальными условиями

$$\left.\begin{aligned} &y(x_0) = y_0,\\ &y'(x_0) = y_0',\\ &y''(x_0) = y_0'',\\ &\ldots\ldots\ldots\ldots\ldots\\ &y^{(n-1)}(x_0) = y_0^{(n-1)}. \end{aligned}\right\}$$

Введём обозначения:

$$\left.\begin{aligned} &y(x) = y_1(x),\\ &y'(x) = y_2(x),\\ &y''(x) = y_3(x),\\ &\ldots\ldots\ldots\ldots\ldots\\ &y^{(n-1)}(x) = y_n(x). \end{aligned}\right\}$$

При использовании этих обозначений, уравнение (10.25) может быть заменено на эквивалентную систему уравнений:

$$\left.\begin{aligned} &y_1'(x) = y_2(x),\\ &y_2'(x) = y_3(x),\\ &\ldots\ldots\ldots\ldots\ldots\ldots\\ &y_{n-1}'(x) = y_n(x),\\ &y_n'(x) = f(x, y_1, y_2, \ldots, y_n). \end{aligned}\right\} \tag{10.26}$$

Введём теперь привычные обозначения:

$$\left.\begin{aligned} &y_2(x) = f_1(x),\\ &y_3(x) = f_2(x),\\ &\ldots\ldots\ldots\ldots\ldots\ldots\\ &y_n(x) = f_{n-1}(x),\\ &f(x, y_1, y_2, \ldots, y_n) = f_n(x, y_1, y_2, \ldots, y_n). \end{aligned}\right\}$$

В результате получим систему уравнений, совпадающую с рассмотренной выше системой:

$$\left.\begin{aligned} &y_1'(x) = f_1(x),\\ &y_2'(x) = f_2(x),\\ &\ldots\ldots\ldots\ldots\ldots\ldots\\ &y_{n-1}'(x) = f_{n-1}(x),\\ &y_n'(x) = f_n(x, y_1, y_2, \ldots, y_n). \end{aligned}\right\} \tag{10.27}$$

Для системы (10.27) мы можем применить схемы Рунге–Кутты. В качестве примера рассмотрим уравнение второго порядка:

$$\begin{aligned} &y''(x) = f\big(x, y(x), y'(x)\big),\\ &y(x_0) = y_1^0; \quad y'(x_0) = y_2^0. \end{aligned} \tag{10.28}$$

Уравнение (10.28) заменяем эквивалентной системой:

$$\left.\begin{aligned} &y_1'(x) = f_1(x),\\ &y_2'(x) = f_2\big(x, y_1(x), y_2(x)\big). \end{aligned}\right\}, \tag{10.29}$$

где

$$f_1(x) = y_2(x) = y'(x),$$
$$f_2\big(x, y_1(x), y_2(x)\big) = f\big(x, y(x), y'(x)\big).$$

Начальные условия будут иметь вид: $y_1(x_0) = y_1^0$; $y_2(x_0) = y_2^0$.

Применим схему Рунге–Кутты четвёртого порядка точности для системы (10.29). В результате имеем:

$$\left.\begin{aligned} y_1^{i+1} &= y_1^i + \frac{1}{6}[k_1 + 2k_2 + 2k_3 + k_4],\\ y_2^{i+1} &= y_2^i + \frac{1}{6}[q_1 + 2q_2 + 2q_3 + q_4], \end{aligned}\right\},$$

где

$$\left.\begin{aligned}
&k_1 = h \cdot f_1(x_i) = h \cdot y_2(x_i),\\
&k_2 = k_3 = h \cdot y_2\left(x_i + \frac{1}{2}h\right),\\
&k_4 = h \cdot y_2(x_i + h),\\
&q_1 = h \cdot f(x_i, y_1^i, y_2^i),\\
&q_2 = h \cdot f\left(x_i + \frac{1}{2}h, y_1^i + \frac{1}{2}k_1, y_2^i + \frac{1}{2}q_1\right),\\
&q_3 = h \cdot f\left(x_i + \frac{1}{2}h, y_1^i + \frac{1}{2}k_2, y_2^i + \frac{1}{2}q_2\right),\\
&q_4 = h \cdot f(x_i + h, y_1^i + k_3, y_2^i + q_3).
\end{aligned}\right\}$$

10.6 Контроль погрешности

Численная реализация решения ДУ выполняется следующим образом. Задаётся сетка $x_i = a + ih_0$, $i = 1, \ldots, m$; $h_0 = \dfrac{b-a}{m}$. Затем выполняются следующие действия:

а) На каждом i-ом шаге в точке $x = x_i$ вычисляют два значения функции $y_i^{(1)}$ и $y_i^{(2)}$ по формулам:

$$\left\{\begin{aligned}
y_i^{(1)} &= y_{i-1} + \sum_{j=1}^{q} p_j(h_0) k_j(h_0, x_{i-1}, y_{i-1}),\\
\tilde{y}_i &= y_{i-1} + \sum_{j=1}^{q} p_j(h_1) k_j(h_1, x_{i-1}, y_{i-1}),\\
y_i^{(2)} &= \tilde{y}_i + \sum_{j=1}^{q} p_j(h_1) k_j(h_1, x_{i-1} + h_1, \tilde{y}_i),
\end{aligned}\right. \qquad (10.30)$$

где $h_1 = \dfrac{1}{2}h_0$.

б) Проверяется условие

$$\left| y_i^{(1)} - y_i^{(2)} \right| < \varepsilon, \qquad (10.31)$$

где ε — заданная точность. Если оно выполнено, то переходят на следующий шаг $i+1$ с тем же h_0. Если (10.31) не выполнено, то полагают $h_0 = h_1$, $h_1 = \dfrac{1}{2}h_0$ и переходят на шаг (а). Такая схема гарантирует получение решения с заданной точностью. При этом время счёта существенно увеличивается.

Контрольные вопросы по главе 10

1. В чем состоит решение задачи Коши для уравнения 1-го порядка?
2. В чем состоит решение задачи Коши для уравнения n-го порядка?
3. Как заменить дифференциальное уравнение n-го порядка на эквивалентную систему дифференциальных уравнений n-го порядка?
4. Сформулируйте теорему существования решения обыкновенного дифференциального уравнения.
5. Запишите формулу метода Эйлера решения дифференциального уравнения.
6. Запишите схему Рунге—Кутты первого порядка.
7. Запишите схему Рунге—Кутты второго порядка.
8. Запишите схему Рунге—Кутты третьего порядка.
9. Запишите схему Рунге—Кутты четвертого порядка.
10. Запишите схему Рунге—Кутты четвертого порядка для системы дифференциальных уравнений.
11. Запишите схему Рунге—Кутты четвертого порядка для дифференциального уравнения n-го порядка.

ЛИТЕРАТУРА

[1] Амосов А. А. Вычислительные методы для инженеров : учеб. пособие / А. А. Амосов, Ю. А. Дубинский, Н. В. Копченова. — 2-е изд., доп. — М. : Изд-во МЭИ, 2003. — 596 с.

[2] Зазарыкин В. М. Численные методы : учеб. пособие / В. М. Зазарыкин, В. Г. Житомирский, М. П. Лапчик. — М. : Просвещение, 1990. — 176 с.

[3] Демидович Б. П. Основы вычислительной математики / Б. П. Демидович, И. А. Марон. — М. : Наука, 1966. — 664 с.

[4] Меркулова Н. Н. Методы приближенных вычислений : в 2 ч. : учеб. пособие / Н. Н. Меркулова, М. Д. Михайлов. — Томск : Томск. гос. ун-т, 2005. — Ч. 1. — 257 с.

[5] Меркулова Н. Н. Методы приближенных вычислений : в 2 ч. : учеб. пособие / Н. Н. Меркулова, М. Д. Михайлов. — Томск : ТМЛ-Пресс, 2007. — Ч. 2. — 240 с.

[6] Бахвалов Н. С. Численные методы / Н. С. Бахвалов, Н. П. Жидков, Г. М. Кобельков. — М. : Наука, 2001. — 630 с.

[7] Формалеев В. Ф. Численные методы / В. Ф. Формалеев, Д. Л. Ревизников. — М. : Физмалит, 2004. — 398 с.

[8] Мицель А. А. Практикум по численным методам : учеб. пособие / А. А. Мицель. — Томск : Томск. гос. ун-т систем управления и радиоэлектроники, 2004. — 196 с.

[9] Мицель А. А. Вычислительные методы : учеб. пособие / А. А. Мицель. — Томск : В-Спектр, 2010. — 264 с.

[10] Мицель А. А. Вычислительная математика. Лабораторный практикум / А. А. Мицель. — Томск : ТУСУР, 1999. — 106 с.

[11] Сборник задач по методам вычислений : учеб. пособие для вузов / под ред. И. П. Монастырного. — М. : ФМЛ, 1994. — 319 с.

[12] Сборник задач по математике для ВТУЗов / под ред. А. В. Ефимова, Б. П. Демидовича. — М. : Наука, 1993. — Ч. 1 : Линейная алгебра.

[13] Мицель А. А., Методы оптимизации : учеб. пособие / А. А. Мицель, А. А. Шелестов. — Томск : Изд-во ТУСУР, 2004. — 255 с.

[14] Большой энциклопедический словарь. Математика. — М. : Изд-во «Большая Российская энциклопедия», 1998.

[15] Стечкин С. Б. Сплайны в вычислительной математике / С. Б. Стечкин, Ю. Н. Субботин. — М. : Наука, 1976. — 248 с.

[16] Мицель А. А. Приближение сплайнами : учеб. пособие / А. А. Мицель, М. Ю. Катаев. — Томск : ТУСУР, 2001. — 40 с.

[17] Демидович Б. П. Численные методы анализа / Б. П. Демидович, И. А. Марон, Э. З. Шувалова. — М. : ФИЗМАТЛИТ, 1963. — 400 с.

[18] Миньков С. Л. Основы численных методов : учеб. пособие / С. Л. Миньков, Л. Л. Миньков. — Томск : Изд-во НТЛ, 2006. — 260 с.

[19] Камке Э. Справочник по обыкновенным дифференциальным уравнениям / Э. Камке. — М. : Наука, 1976. — 575 с.

ГЛОССАРИЙ

QR-алгоритм вычисления собственных чисел квадратной матрицы A является итерационным и основан на разложении произвольной матрицы A в произведение ортогональной и верхней треугольной матриц. В результате получается последовательность матриц, подобных исходной матрице A. Последовательность матриц сходится по форме к некоторой верхней треугольной или к верхней блочно-треугольной матрице $\tilde{A}$, имеющей те же собственные числа, что и матрица A.

Абсолютная погрешность $\Delta(x^*)$ приближенного числа x^* есть разность между точным x и приближенным значением x^*, взятая по модулю $\Delta(x^*) = |x - x^*|$.

Абсолютная погрешность алгебраической суммы $u^* = \pm x_1^* \pm x_2^* \pm \ldots \pm x_n^*$ нескольких приближенных чисел приближенно равна сумме абсолютных погрешностей этих чисел, т. е. $\Delta(u^*) \approx \Delta(x_1^*) + \Delta(x_2^*) + \ldots + \Delta(x_n^*)$.

Абсолютная погрешность вектора x^* равна $\Delta(x) = \|x - x^*\|$.

Абсолютная погрешность матрицы A^* равна $\Delta(A) = \|A - A^*\|$.

Абсолютная погрешность функции. Пусть задана дифференцируемая функция $u^* = f(x_1^*, x_2^*, \ldots, x_n^*)$ и $\Delta(x_i^*)$ — абсолютные погрешности аргументов функции. Абсолютная погрешность функции приближенно равна $\Delta(u^*) \approx \sum_{i=1}^{n} \left| \frac{\partial f}{\partial x_i^*} \right| \Delta(x_i^*)$.

Абсолютное число обусловленности матрицы квадратной A — это коэффициент пропорциональности $v_\Delta = \|A^{-1}\|$ между абсолютной погрешностью $\Delta(x)$ решения системы $Ax = b$ и абсолютной погрешностью $\Delta(b)$ правой части b.

Верные значащие цифры. Говорят, что n первых значащих цифр (десятичных знаков) приближенного числа являются верными, если абсолютная погрешность этого числа не превышает единицы разряда, выражаемого n-ой значащей цифрой, считая слева направо.

Вычисление обратной матрицы сводится к решению матричного уравнения $AX = E$, где E — единичная матрица.

Вычисление определителя квадратной матрицы A методом Гаусса сводится к приведению исходной матрицы A к треугольному виду и последующему вычислению произведения диагональных элементов верхней треугольной матрицы. Вычисление определителя методом Халецкого сводится к разложению матрицы

на произведение двух матриц $A = BC$, где B — нижняя треугольная, а C — верхняя треугольная матрица с единичной диагональю. Определитель равен произведению диагональных элементов матрицы B.

Вычислительная задача. Под вычислительной задачей понимают одну из трех задач, которые возникают при анализе математических моделей: прямую задачу, обратную задачу и задачу идентификации.

Значащая цифра приближенного числа есть всякая цифра в его десятичном изображении, отличная от нуля, и нуль, если он содержится между значащими цифрами или является представителем сохраненного десятичного разряда.

Интегральное аппроксимирование функций непрерывной функции $f(x)$ на отрезке $[a, b]$ производится с помощью обобщенного полинома $\Phi_m(x) = \sum_{i=0}^{m} a_i \varphi_i(x)$, где $\{\varphi_i(x)\}$ — заданная система непрерывных функций; a_i — коэффициенты, определяемые из решения системы уравнений:

$$\sum_{i=0}^{m} a_i (\varphi_i, \varphi_k) = (f, \varphi_k), \ k = 0, \ \ldots, m,$$

где $(\varphi_i, \varphi_k) = \int_a^b \varphi_i(x)\, \varphi_k(x)\, dx$, $(f, \varphi_k) = \int_a^b f(x) \varphi_k(x)\, dx$.

Интегральное приближение — приближение на отрезке $a \leqslant x \leqslant b$.

Интерполяция — способ приближения функции $f(x)$ функцией $g(x)$, при котором значения приближающей функции $g(x_i) = y_i$, $i = 0, 1, \ldots, n$, При этом полагается, что значение сетки $\{x_j\}$, в которой вычисляют $g(x_j)$, не выходят за пределы интервала $[a, b]$: $(a \leqslant x_j \leqslant b;\ j = 1, 2, \ldots, m)$.

Интерполяционная формула Ньютона для равномерной сетки имеет вид:

$$P_n(x) = y_0 + \sum_{k=1}^{n} \frac{\Delta^k y_0}{k! h^k} \prod_{j=1}^{k} (x - x_{j-1}),$$

где $\Delta^k y_0$ — конечная разность k-го порядка в точке x_0.

Интерполяционная формула Ньютона для неравномерной сетки имеет вид:

$$P_n(x) = y_0 + \sum_{k=1}^{n} f(x_0; x_1; \ldots; x_k) \prod_{j=1}^{k} (x - x_{j-1}),$$

где $f(x_0; x_1; \ldots; x_k)$ — разделенная разность k-го порядка.

Интерполяционный многочлен $P_n(x) = \sum_{k=0}^{n} a_k x^k$ определяется из условий:

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

Итерационные методы решения СЛАУ являются приближенными. Они дают решение СЛАУ как предел последовательных приближений, выполненных по единообразной схеме. К итерационным методам решения СЛАУ относятся: метод простой итерации, метод Зейделя, релаксаций, градиентные методы и их модификации.

Квадратурные формулы — приближенные равенства вида:

$$\int_a^b f(x)dx \approx \sum_{i=0}^{n} A_i f(x_i).$$

Здесь x_i — узлы квадратурной формулы ($x_i \in [a,b]$); A_i — числовые коэффициенты (веса квадратурной формулы). Сумма $\sum_{i=0}^{n} A_i f(x_i)$, которая принимается за приближенное значение интеграла, называется *квадратурной суммой*.

Круги Гершгорина S_i квадратной матрицы A с элементами a_{ij} представляют собой замкнутые круги радиуса $r_i = \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}|$ на комплексной плоскости с центрами в точках a_{ii}, т. е. $S_i = \{z \in C : |z - a_{ii}| \leqslant r_i\}$. Все собственные значения матрицы A лежат в объединении кругов $S_1, S_2, S_1, \ldots, S_n$.

Конечная разность порядка k в точке x_i табличной функции $y_i = f(x_i)$, заданной на равномерной сетке $x_i = a + i \cdot h$, $i = 0, 1, \ldots, n$ определяется как $\Delta^k y_i = \Delta^{k-1} y_{i+1} - \Delta^{k-1} y_i$, $k \geqslant 1$, $\Delta^0 y_i = y_i$.

Корень уравнения. Всякое значение ξ, обращающее функцию $f(x)$ в нуль, то есть такое, что $f(\xi) = 0$, называется корнем уравнения $f(x) = 0$ или нулем функции $f(x)$. Если непрерывная функция $f(x)$ принимает значения разных знаков на концах отрезка $[a,b]$, то есть $f(a) \cdot f(b) < 0$, то внутри этого отрезка содержится по меньшей мере один корень уравнения $f(x) = 0$.

Корректность вычислительных алгоритмов. Вычислительный алгоритм называется *корректным*, если выполнены три условия: 1) он позволяет после выполнения конечного числа элементарных для вычислительной машины операций преобразовать любое входное данное $x \in X$ в результат y; 2) результат y устойчив по отношению к малым возмущениям входных данных; 3) результат y обладает вычислительной устойчивостью.

Корректность задачи. Вычислительная задача называется *корректной*, если выполнены следующие три условия: 1) ее решение $y \in Y$ существует при любых входных данных $x \in X$; 2) это решение единственно; 3) решение устойчиво по отношению к малым возмущениям входных данных.

Критерий поиска корня уравнения — это условие завершения итерационного процесса: $|x_n - x_{n-1}| \leqslant \varepsilon$, $|f(x_n)| \leqslant \varepsilon$, где ε — заданная точность.

Ленточная система — это система $Ax = b$, матрица A которой заполнена частично. Например, трехдиагональная система имеет матрицу, у которой заполнена главная диагональ и две боковые (нижняя и верхняя); остальные элементы матрицы A равны нулю. В пятидиагональной системе заполнена главная диагональ, две нижние диагонали и две верхние диагонали; остальные элементы нулевые.

Локализация корней на интервале $[a,b]$ означает установление подынтервалов $[\alpha_i, \beta_i] \in [a,b]$, в которых содержится один корень уравнения.

Локализация корней нелинейной системы. Для каждого из искомых решений ξ указывают множество, содержащее только одно это решение и расположенное в достаточно малой его окрестности. Часто в качестве такого множества выступает параллелепипед или шар в n-мерном пространстве.

Метод вращения решения системы $Ax = b$ с квадратной матрицей A — это метод представления матрицы A в виде произведения ортогональной матрицы Q на верхнюю треугольную матрицу R: $A = QR$ и переход к системе $Rx = Q^T b$. Обратный ход метода вращений совпадает с обратным ходом метода Гаусса.

Метод Гаусса решения системы $Ax = b$ — это метод последовательного исключения неизвестных. Суть его состоит в преобразовании исходной системы к системе с верхней треугольной матрицей (прямой ход), из которой затем последовательно (обратным ходом) получаются значения всех неизвестных.

Метод Данилевского вычисления собственных чисел квадратной матрицы A заключается в приведении матрицы A к матрице Фробениуса, которая является подобной матрице A. Разлагая определитель матрицы $B - \lambda E$ по элементам 1-ой строки, получим полином $D(\lambda) = (-1)^n [\lambda^n - b_1\lambda^{n-1} - b_2\lambda^{n-2} - b_3\lambda^{n-3} - \ldots - b_n]$, корни которого являются собственными числами матрицы A.

Метод Зейделя решения системы $Ax = b$ с квадратной матрицей A размерности $(n \times n)$ представляет собой модификацию метода итераций, суть которой состоит в том, что при вычислении $(k+1)$-го приближения неизвестной x_i учитываются уже вычисленные ранее $(k+1)$-приближение неизвестных $x_1, x_2, \ldots, x_{i-1}$.

Метод итерации решения системы $Ax = b$ с квадратной матрицей A размерности $(n \times n)$ — это метод замены исходной системы на эквивалентную ей систему $x = \beta + \alpha x$ и решение последней методом последовательных приближений: $x^{(0)} = \beta$, $x^{(k)} = \beta + \alpha x^{(k-1)}$, $k = 1, 2, \ldots$. Здесь компоненты вектора β и матрицы α вычисляются по формулам: $\beta_i = b_i / a_{ii}$; $\alpha_{ij} = -a_{ij}/a_{ii}$ при $i \neq j$; $\alpha_{ii} = 0$.

Метод итераций поиска корня векторного уравнения $f(x) = 0$ — это итерационный процесс вычисления последовательности векторов по формуле: $x^{(k)} = \varphi(x^{(k-1)})$, $k = 1, 2, \ldots$, где $\varphi(x) = x + \Lambda f(x)$, $\Lambda = -W^{-1}(x^{(0)})$.

Метод Ньютона поиска корня векторного уравнения $f(x) = 0$ — это итерационный процесс вычисления последовательности векторов по формуле: $x^{(k+1)} = x^{(k)} - W^{-1}(x^{(k)}) f(x^{(k)})$, $k = 0, 1, \ldots$, где $W(x^{(k)})$ — матрица Якоби в точке $x^{(k)}$.

Метод прогонки — это специальный метод решения ленточных систем.

Метод обратных итераций вычисления собственных векторов матрицы A, соответствующих собственным числам λ_j, представляет собой итерационную процедуру решения систем уравнений: $(A - \lambda_j E) y^{(k+1)} = x^{(k)}$, $x^{(0)} = (1, 1, \ldots, 1)^T$ с последующей нормировкой решения: $x^{(k+1)} = y^{(k+1)} / \| y^{(k+1)} \|_3$.

Метод ортогонализации решения системы $Ax = b$ с квадратной матрицей A размерности $(n \times n)$ — это метод представления матрицы A в виде произведения матрицы с ортогональными столбцами R и верхней треугольной матрицы T с единичной диагональю: $A = RT$ и переход к системе $Tx = DR^T b$, где

$$D = \operatorname{diag}\left\{1/\sum_{k=1}^{n} r_{k1}^2,\ 1/\sum_{k=1}^{n} r_{k2}^2,\ \ldots,\ 1/\sum_{k=1}^{n} r_{kn}^2\right\}.$$

Обратный ход метода ортогонализации совпадает с обратным ходом метода Гаусса.

Методы Рунге—Кутты — численные методы решения задачи Коши, представляют собой схемы последовательного вычисления значений функции в узлах сетки $x_i = a + i \cdot h$, $i = 0, 1, \ldots, n$.

Метод Халецкого решения системы $Ax = d$ с квадратной матрицей A размерности $(n \times n)$ — это метод представления матрицы A в виде произведения нижней треугольной матрицы B и верхней треугольной матрицы C с единичной диагональю: $A = BC$ и переход к двух системам с треугольными матрицами $By = d$, $Cx = y$, каждая из которых легко решается.

Метод Эйлера решения задачи Коши $y' = f(x,y)$, $a \leqslant x \leqslant b$, $y(a) = y_0$ представляет собой схему последовательного вычисления значений функции: $y_{i+1} = y_i + h \cdot f(x_i, y_i)$; $y(a) = y_0$ в узлах сетки $x_i = a + i \cdot h$, $i = 0, 1, \ldots, n$.

Минимальная погрешность интерполяции равна

$$\left|R_n(x)\right| \leqslant \frac{M_{n+1}}{(n+1)!2^n}\left[\frac{b-a}{2}\right]^{n+1}.$$

Достигается на сетке Чебышева $x_k = \dfrac{a+b}{2} + \dfrac{b-a}{2}\cos\left(\dfrac{2k+1}{2n}\pi\right)$.

Многочлен Лагранжа имеет вид: $L_n(x) = \sum_{j=0}^{n} y_j p_{nj}(x)$, где полиномы

$$p_{nj}(x) = \prod_{\substack{k=0 \\ k \neq j}}^{n} \frac{x - x_k}{x_j - x_k} = \frac{(x-x_0)(x-x_1)\ldots(x-x_{j-1})(x-x_{j+1})\ldots(x-x_n)}{(x_j-x_0)(x_j-x_1)\ldots(x_j-x_{j-1})(x_j-x_{j+1})\ldots(x_j-x_n)}$$

удовлетворяют условиям $p_{nj}(x_i) = \begin{cases} 1 & \text{при } i = j \\ 0 & \text{при } i \neq j \end{cases}$.

Норма вектора x — это действительное положительное число, вычисляемое с помощью элементов вектора x. Обозначается норма как $\|x\|$. Наиболее употребительны в пространстве векторов следующие нормы:

а) $\|x\|_1 = \max\limits_{1 \leqslant i \leqslant m} |x_i|$; б) $\|x\|_2 = \sum\limits_{j=1}^{m} |x_j|$; в) $\|x\|_3 = \sqrt{\sum\limits_{j=1}^{m} x_j^2} = \sqrt{(x,x)}$.

Норма матрицы A — это действительное положительное число, вычисляемое с помощью элементов матрицы A. Обозначается норма как $\|A\|$. Согласованные с нормой вектора x ($\|x\|_1$, $\|x\|_2$ и $\|x\|_3$) нормы матрицы A (размера $m \times m$) равны:

а) $\|A\|_1 = \max\limits_{1 \leqslant i \leqslant m}\left(\sum\limits_{j=1}^{m} |a_{ij}|\right)$; б) $\|A\|_2 = \max\limits_{1 \leqslant j \leqslant m}\left(\sum\limits_{i=1}^{m} |a_{ij}|\right)$; в) $\|A\|_3 = \sqrt{\sum\limits_i \sum\limits_j a_{ij}^2}$.

Обобщенный многочлен степени m определяется как $\Phi_m(x) = \sum_{i=0}^{m} a_i \varphi_i(x)$, где $\{\varphi_i(x)\}$ — система базисных функций, заданных на $[a,b]$ и являющихся гладкими (непрерывно дифференцируемыми); a_i — коэффициенты, которые выбирают таким образом, чтобы отклонение $f(x)$ от $\Phi_m(x)$ было минимальным на заданном множестве $X = \{x\}$.

Обусловленность задачи. Под *обусловленностью вычислительной задачи* понимают чувствительность ее решения к малым погрешностям входных данных. Задачу называют *хорошо обусловленной*, если малым погрешностям входных данных отвечают малые погрешности решения, и *плохо обусловленной*, если возможны сильные изменения решения.

Обусловленность задачи вычисления корня. Под обусловленностью задачи вычисления корня уравнения $f(x) = 0$ понимают чувствительность погрешности корня $\Delta(\xi)$ к погрешности функции $\Delta(f)$, т. е. $\Delta(\xi) = v_\Delta \cdot \Delta(f)$, где $v_\Delta = \dfrac{1}{|f'(x)|}$ — абсолютное число обусловленности.

Обусловленность задачи вычисления корня нелинейной системы. Под обусловленностью задачи вычисления корня векторного уравнения $f(x) = 0$ понимают чувствительность погрешности векторного корня $\Delta(\xi)$ к погрешности векторной функции $\Delta(f)$, т. е. $\Delta(\xi) = v_\Delta \cdot \Delta(f)$, где $v_\Delta = \left\| (W(\xi))^{-1} \right\|$ — абсолютное число обусловленности. Здесь $W(\xi)$ — матрица Якоби.

Обусловленность задачи вычисления собственных значений λ матрицы A выражается чувствительностью погрешности вычисляемых собственных значений $\Delta\lambda$ к погрешностям матрицы ΔA: $\Delta(\lambda) = \mathrm{cond}_3(P)\|\Delta A\|_3$, где P — матрица собственных векторов матрицы A.

Обусловленность задачи вычисления собственных векторов симметричной матрицы A определяется выражением $|\sin\varphi| \leqslant \dfrac{\|\Delta A\|_3}{\Delta(\lambda)}$, где $\varphi = \arccos\left((x^*, x)/\|x^*\|\|x\|\right)$ — угол между собственными векторами x^* и x, соответственно приближенной и точной матриц A^* и A, а $\Delta(\lambda)$ — погрешность собственного значения матрицы A.

Обусловленность квадратурных формул определяется зависимостью погрешности квадратуры от погрешности подынтегральной функции Δf:

$$\left| \sum_{i=0}^{n} A_i f(x_i) - \sum_{i=0}^{n} A_i f^*(x_i) \right| \leqslant \Delta f \sum_{i=0}^{n} |A_i|,$$

где $f(x_i)$ и $f^*(x_i)$ — точное и приближенное значения интегрируемой функции. Величина $\sum_{i=0}^{n} |A_i|$ играет роль числа обусловленности, которое для большинства квадратурных формул равно длине интервала $(b-a)$.

Обусловленность формул численного дифференцирования определяется чувствительностью погрешности дифференцирования r к погрешности δy_m табличной функции $y = f(x)$ и в случае использования линейного полинома для вычисления

первой производной определяется выражением $r = \frac{2}{h}\left|\Delta(y_m)\right| + \frac{1}{2}h \cdot M_2$, где h — шаг дискретизации функции, M_2 — максимальное значение второй производной.

Округление чисел. Если первая слева из отбрасываемых цифр меньше 5, то сохраняемые цифры остаются без изменения. Если же она больше либо равна 5, то в младший сохраняемый разряд добавляется единица.

Ортогональные системы функций $\{\varphi_i(x)\}$ определяются из условия:

$$(\varphi_m, \varphi_n) = \begin{cases} \int\limits_a^b \varphi_m(x)\,\varphi_n(x)\,dx = 0, \ m \neq n; \\ \int\limits_a^b \varphi_m^2(x)\,dx, \ m = n. \end{cases}$$

Ортонормированные системы функций удовлетворяют условию:

$$\int\limits_a^b \varphi_m(x)\,\varphi_n(x)\,dx = \delta_{mn} = \begin{cases} 1, & m = n, \\ 0, & m \neq n. \end{cases}$$

Относительная погрешность $\delta(x^*)$ приближенного числа x^* — это отношение абсолютной погрешности $\Delta(x^*)$ этого числа к модулю соответствующего приближенного числа $\delta(x^*) = \Delta(x^*)/|x^*|$.

Относительная погрешность вектора x^* равна $\delta(x) = \|x - x^*\|/\|x^*\|$.

Относительная погрешность корня $u^* = (x^*)^{1/m}$ приближенно равна

$$\delta(u^*) \approx \frac{1}{m}\delta(x^*).$$

Относительная погрешность матрицы A^* равна $\delta(A) = \|A - A^*\|/\|A^*\|$.

Относительная погрешность произведения $u^* = x_1^* \cdot x_2^*$ двух приближенных чисел приближенно равна $\delta(u^*) \approx \delta(x_1^*) + \delta_2(x_2^*) + \delta(x_1^*) \cdot \delta(x_2^*)$.

Относительная погрешность разности $u^* = x_1^* - x_2^*$ двух приближенных чисел приближенно равна

$$\delta(u^*) = \frac{\Delta(x_1^*) + \Delta(x_2^*)}{|x_1^* - x_2^*|} \leqslant \frac{\delta_m \cdot |x_1^* + x_2^*|}{|x_1^* - x_2^*|},$$

где $\delta_m = \max\limits_i\left(\delta(x_i^*)\right)$.

Относительная погрешность частного $u^* = x_1^*/x_2^*$ двух приближенных чисел приближенно равна

$$\delta(u^*) \approx \frac{\delta(x_1^*) + \delta(x_2^*)}{1 - \delta(x_2^*)}.$$

Относительное число обусловленности матрицы квадратной A — это коэффициент пропорциональности $v_\delta = \|A^{-1}\| \cdot \|A\|$ между относительной погрешностью $\delta(x)$ решения системы $Ax = b$ и относительной погрешностью $\delta(b)$ правой части b.

Переопределенная система линейных уравнений $Ax = b$ — это система уравнений, в которой число уравнений больше числа неизвестных. Матрица A такой системы прямоугольная размерности $(n \times m)$, $n > m$; размерность векторов x и b равны m и n соответственно.

Погрешность интерполяции $R_n(x) = f(x) - P_n(x)$ определяется формулой

$$R_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}\omega_{n+1}(x),$$

где $\omega_{n+1}(x) = \prod_{i=0}^{n}(x - x_i)$.

Погрешность квадратурной формулы — это величина $R = \int_a^b f(x)dx - \sum_{i=0}^{n} A_i f(x_i)$.

Правила записи приближенных чисел. Приближенное число x может быть представлено в виде конечной десятичной дроби:

$$x^* = \alpha_m 10^m + \alpha_{m-1} 10^{m-1} + \ldots + \alpha_{m-n+1} 10^{m-n+1} + \ldots,$$

где α_i — цифры числа x^* в i-ом разряде ($\alpha_i = 0, 1, \ldots, 9$), причем $\alpha_m \neq 0$; m — старший десятичный разряд числа x^* (некоторое целое число).

Преобразование подобия. Говорят, что матрицы A и B подобны, если существует невырожденная матрица P (матрица подобия), такая, что $B = P^{-1}AP$. Само преобразование матрицы A к виду $B = P^{-1}AP$ называют *преобразованием подобия*.

Приближение функции $f(x)$, заданной на интервале $[a, b]$, это замена $f(x)$ некоторой другой функцией $g(x)$, близкой к исходной функции $f(x)$.

Принцип равных влияний. Согласно этому принципу предполагается, что все слагаемые в сумме погрешностей одинаково влияют на образование общей абсолютной погрешности $\Delta(f)$ функции $f(x_1^*, \ldots, x_n^*)$.

Прямые (точные) методы решения систем линейных алгебраических уравнений дают решение системы за конечное число арифметических операций. Если все операции выполняются точно, то решение задачи получается точным. К прямым методам решения СЛАУ относятся: метод Крамера, методы последовательного исключения неизвестных (метод Гаусса и его модификации), метод ортогонализации, метод декомпозиции, метод вращений.

Разделенная разность k-го порядка в точке x_i табличной функции $y_i = f(x_i)$, заданной на неравномерной сетке $x_0, x_1, \ldots, x_n$, вычисляется по формуле:

$$f(x_i; x_{i+1}; \ldots; x_{i+k}) = \frac{f(x_{i+1}; \ldots; x_{i+k}) - f(x_i; x_{i+1}; \ldots; x_{i+k-1})}{x_{i+k} - x_i}, \; i = 0, \; 1, \; \ldots$$

Собственные значения λ квадратной матрицы A представляют собой корни характеристического уравнения $\det(A - \lambda E) = 0$, которое может быть представлено в виде: $f_n(\lambda) = \lambda^n + p_1\lambda^{n-1} + p_2\lambda^{n-2} + \ldots + p_{n-1}\lambda + p_n = 0$. Здесь E — единичная матрица, $p_1, p_2, \ldots, p_n$ — коэффициенты многочлена.

Сплайном называют кусочно-полиномиальную функцию, склеенную из различных многочленов, непрерывную на всем отрезке $[a, b]$ вместе со своими несколькими производными. Если используются многочлены первой степени, то имеем линейный сплайн, если используются многочлены второй степени — параболический сплайн, третьей степени — кубический. На практике используют линейные, параболические и кубические сплайны.

Степенной метод вычисления максимального собственного числа квадратной матрицы A представляет собой итерационный процесс: $u^{(k)} = Ay^{(k-1)}$, $\lambda_1^{(k)} = (u^{(k)}, y^{(k-1)})$, $y^{(k)} = u^{(k)} / \|u^{(k)}\|$. В качестве $y^{(0)}$ выбирается вектор с единичной нормой.

Сходимость итерационных методов. Говорят, что итерационный метод имеет сходимость порядка r, если в области сходимости справедлива оценка $|\xi - x_{k+1}| \leqslant \leqslant \alpha \cdot |\xi - x_k|^r$, $0 < \alpha < 1$, где x_k — k-ое приближение корня, ξ — корень (нуль функции), α — коэффициент сходимости. Если $r = 1$, то метод обладает линейной сходимостью, при $r = 2$ — квадратичной, $r = 3$ — кубической.

Точечное приближение — приближение на множестве дискретных точек x_0, $x_1, \ldots, x_n$.

Требования к вычислительным алгоритмам. 1) Требования к абстрактным алгоритмам: экономичность, точность, экономия памяти, простота. 2) Требования к программным реализациям алгоритмов: надежность, работоспособность, переносимость, поддерживаемость, простота в использовании.

Трудоемкость метода — это количество арифметических операций, необходимых для реализации какой-либо вычислительной схемы.

Упрощенный метод Ньютона поиска корня векторного уравнения $f(x) = 0$ — это итерационный процесс вычисления последовательности векторов по формуле: $x^{(k+1)} = x^{(k)} - W^{-1}(x^{(0)}) f(x^{(k)})$, $k = 0, 1, \ldots$, где $W(x^{(0)})$ — матрица Якоби в точке начального приближения $x^{(0)}$.

Численное дифференцирование — приближенное вычисление производных с помощью значений дифференцируемой функции, заданной в дискретных точках.

Численное интегрирование — приближенное вычисление интеграла с помощью значений подынтегральной функции, заданной в дискретных точках.

Число обусловленности — это коэффициент возможного возрастания погрешностей в решении вследствие наличия погрешностей входных данных. Пусть между абсолютными и относительными погрешностями входных данных x и решения y установлено неравенство $\Delta(y) \leqslant \nu_\Delta \Delta(x)$, $\delta(y) \leqslant \nu_\delta \delta(x)$. Величина ν_Δ называется *абсолютным числом обусловленности*, а ν_δ называют *относительным числом обусловленности*.

ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ

Учебное издание

Мицель Артур Александрович

ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ

Учебное пособие

Корректор Осипова Е. А.
Компьютерная верстка Хомич С. Л.

Подписано в печать 08.11.13. Формат 60x84/8.
Усл. печ. л. 23,25. Тираж 200 экз. Заказ

Издано в ООО «Эль Контент»
634029, г. Томск, ул. Кузнецова д. 11 оф. 17
Отпечатано в Томском государственном университете
систем управления и радиоэлектроники.
634050, г. Томск, пр. Ленина, 40
Тел. (3822) 533018.