САМАРСКИЙ ГОСУДАРСТВЕННЫЙ АЭРОКОСМИЧЕСКИЙ УНИВЕРСИТЕТ
имени академика С.П. Королева

А.Н. Коварцев

ЧИСЛЕННЫЕ МЕТОДЫ

Самара 2000

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

САМАРСКИЙ ГОСУДАРСТВЕННЫЙ АЭРОКОСМИЧЕСКИЙ УНИВЕРСИТЕТ
имени академика С.П. Королева

А.Н. Коварцев

ЧИСЛЕННЫЕ МЕТОДЫ

Курс лекций для студентов заочной формы обучения

Самара 2000

УДК 518

Численные методы: Курс лекций / А.Н. Коварцев. Самар. гос. аэрокосм. ун-т. Самара, 2000. 177 с.

ISBN 5-7883-0116-5

Книга посвящена изложению важнейших методов и приемов решения задач численного анализа на современных ЭВМ. Основная часть книги является учебным пособием по курсу приближенных вычислений. Предназначена для вузов, может быть полезна также для лиц, работающих в области прикладной математики и компьютерных технологий
Табл.2. Ил.33. Библиогр.:10 назв.

Печатается по решению редакционно-издательского совета Самарского государственного аэрокосмического университета имени академика С.П. Королева

Рецензент д-р.техн. наук., проф. С.А. Прохоров

ISBN 5-7883-0116-5

ПРЕДИСЛОВИЕ

Данное учебное пособие основано на курсе лекций "Численные методы", которые автор читал восемь лет на факультете информатики Самарского государственного аэрокосмического университета. Предполагается, что эта книга должна дать читателю основные понятия о численном решении различных задач вычислительной математики на современных ЭВМ. Будучи учебным пособием она, естественно, не претендует на полноту изложения всего арсенала существующих численных методов. Да и вероятно, в силу обширности наработанного материала, это уже и невозможно.

Главное внимание в книге уделяется связанному изложению основных тем численного анализа. По существу, была предпринята попытка наглядного представления смыслового содержания наиболее типичных представителей методов численного анализа. Там, где это необходимо, приводимые результаты подтверждаются аналитически.

Естественно, что читатель, изучивший настоящую книгу, не может считаться специалистом по численному анализу в полном смысле этого слова. Однако она позволяет дать ему необходимые знания для практического решения наиболее типичных задач и закладывает основы для дальнейшего более глубокого изучения предмета.

Автор считает своим долгом выразить глубокую признательность заведующему кафедрой информационных систем и технологий профессору Прохорову С.А., оказавшему неоценимую помощь при подготовке рукописи к печати.

ВВЕДЕНИЕ

Одной из характерных особенностей нашего времени является широкое применение ЭВМ для решения научных, технических и экономических задач. ЭВМ способны производить быстрые вычисления, выдавать точные результаты, запоминать большие массивы информации и производить сложные последовательности вычислений без вмешательства человека.

Однако сама ЭВМ не «решает задачу». Она лишь помогает нам с помощью программ, реализующих вычислительные алгоритмы, исследовать поставленную задачу. Основу многих таких алгоритмов часто составляют численные методы.

Численные методы - *это методы приближенного или точного решения задач чистой или прикладной математики, основанные на построении конечной последовательности действий над конечным множеством чисел.*

Численные методы являются предметом изучения вычислительной математики, которая свое начало ведет из глубокой древности, истоком которой можно считать правила вычисления иррациональных чисел. Современная вычислительная математика состоит из многих разделов. Важнейшими из них являются: вычисление значений функций, вычислительные методы линейной алгебры, численное решение алгебраических и трансцендентных уравнений, численное дифференцирование и интегрирование, численное решение дифференциальных уравнений, численные методы поиска экстремумов функций и т.д.

Математические методы давно и весьма успешно применяются в механике, физике, астрономии, экономике, медицине, технике и т.д. До появления ЭВМ основные усилия ученых были направлены на поиск решений, реализующихся в явном виде. Однако в математике часто встречаются задачи, решение которых не удается получить в виде формулы, связывающей искомые величины с заданными переменными. Для их решения стремятся найти какой-нибудь бесконечный процесс, сходящийся к искомому ответу. Если такой процесс указан, то, выполняя некоторое число шагов и затем обрывая вычисления, мы получим приближенное решение задачи. Эта процедура связана с проведением вычислений по строго определенной системе правил, которая называется алгоритмом.

Такой подход к решению задач был известен еще до появления ЭВМ, но применялся весьма редко из-за исключительной

трудоемкости. Применение численных методов на базе ЭВМ существенно расширило класс задач, допускающих исчерпывающий анализ. Теперь исследователю при построении математической модели не нужно стремиться к сильным упрощениям, которые были необходимы раньше для получения ответа в явном виде. Его внимание, прежде всего, должно быть направлено на то, чтобы правильно учесть все наиболее существенные особенности изучаемого объекта. Далее он решает вопрос о разработке алгоритма решения соответствующей задачи и о его реализации на ЭВМ.

Для иллюстрации алгоритмического подхода к решению математических задач рассмотрим простой пример, связанный с задачей вычисления числа π.

Как известно, вычисление числа π сводится к расчету периметров правильных многоугольников, вписанных в окружность с диаметром D=1 и описанных вокруг нее. Пусть p_n - периметр вписанного правильного многоугольника, а q_n - описанного вокруг окружности. Так, например, для правильных шестиугольников $p_6 = 3, q_6 = 2\sqrt{3}$, следовательно,

$$3 < \pi < 2\sqrt{3}$$

С ростом n периметры p_n растут, а периметры q_n убывают, стремясь в пределе к длине окружности:

$$\lim_{n\to\infty} p_n = \lim_{n\to\infty} q_n = \pi$$

Таким образом, периметры p_n определяют число π с недостатком, а q_n - с избытком:

$$p_n < \pi < q_n \qquad (1)$$

Двухсторонняя оценка (1)* позволяет легко контролировать точность на каждом шаге вычисления. Погрешность вычисления e_n числа π можно оценить: $e_n < q_n - p_n$

Приведем некоторые факты из истории вычисления числа π. Великий древнегреческий ученый Архимед, используя формулу удвоения, дошел до вычисления правильного 96-угольника и получил следующую двухстороннюю оценку π :

$$3{,}14084 < \pi < 3{,}14285.$$

К первой половине XV века в знаменитой обсерватории под Самаркандом придворный астроном Аль-Каши вычислил π с 17 знаками после запятой и дошел до $n=6\cdot 2^{27}$.К концу XIX века английский математик В. Шенкс вычислил 707 знаков π, затратив

на это более 20 лет. В настоящее время с помощью ЭВМ число π вычислено с фантастической точностью - более 500 тысяч знаков.

Понятие решение задачи с помощью ЭВМ включает в себя гораздо больше, нежели просто вычисления на ЭВМ. Для решения и исследования задач в настоящее время принята и представляется наиболее эффективной следующая методология (см. рис.1):

1. **Постановка задачи и определение конечных целей**- наиболее важный и ответственный этап решения задачи, на котором реализуется выбор общего подхода к ее решению, определяются основные критерии, которым должна удовлетворять разрабатываемая система, и дается формальное математическое описание задачи.

На данной стадии требуется глубокое понимание существа задачи. В этой работе вычислительная машина не может оказать практически никакой помощи.

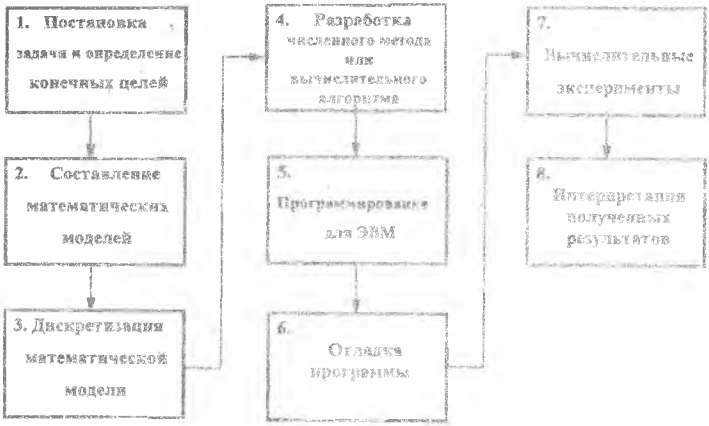


Рис.1. Технологическая цепочка решения прикладных математических задач

2. **Составление математической модели объекта исследования.** Обычно строится континуальная математическая модель, сформулированная в терминах дифференциальных, трансцендентных или алгебраических уравнениях для функций непрерывного аргумента. Она является наиболее экономичным способом описания исследуемого объекта.

3. **Дискретизация математической модели** часто вызвана конструктивными особенностями цифровых вычислительных

машин, имеющих ограниченную разрядную сетку и ограниченную память. Переход от континуальной к дискретной математической модели заключается в замене функций непрерывного аргумента функциями дискретного аргумента. При этом интеграл заменяется конечной суммой, производные разностным отношением и т.д. В результате, как правило, приходят к системе большого количества линейных уравнений с большим количеством неизвестных.

4. **Разработка численного метода**. Математическая формулировка задачи может оказаться непереводимой непосредственно на язык ЭВМ, так как ЭВМ способна выполнять только элементарные арифметические операции. Такие общеизвестные математические понятия, как тригонометрические функции, квадратные корни, логарифмы, корни уравнений и т.д. - все должны быть выражены через элементарные арифметические операции. Разработка вычислительного алгоритма, реализующего выбранный математический метод решения задачи, составляет суть этого этапа.

5. **Программирование для ЭВМ**. Численный алгоритм решения задачи теперь необходимо «переложить» на язык, понятный ЭВМ, т.е. закодировать его (численный метод) на любом из алгоритмических языков программирования.

6. **Отладка программы**. На этом этапе производится тестирование программы с целью обнаружения и устранения допущенных ошибок. Это наиболее трудоемкий и ответственный этап технологической цепочки.

7. **Вычислительный эксперимент**. По разработанной программе, используя исходные данные, производится необходимое количество расчетов на ЭВМ.

8. **Интерпретация результатов**. Полученные результаты расчетов подвергаются всестороннему анализу, на основании которого либо получается полный «ответ» для решаемой задачи, либо выявляются некоторые особенности поведения исследуемого объекта, требующие повторных экспериментов на ЭВМ. Достаточно часто возникает необходимость в изменении постановки задачи, что приводит к полному повторению всех перечисленных выше этапов решения задачи.

Из краткого рассмотрения технологической цепочки «решения задачи» на ЭВМ можно сделать некоторые выводы.

Во-первых, ЭВМ сама задач не решает, она только производит заранее определенную последовательность вычислений.

Во-вторых, использование ЭВМ не освобождает разработчика от тщательного осмысления своей работы, глубокого изучения

исследуемого объекта и применяемых для решения задачи разделов математики. Машина может производить вычисления быстрее и точнее человека, но она не способна решать, какой должна быть программа вычислений или что делать с полученными результатами.

В заключении вводной части данного пособия отметим наиболее важные проблемы вычислительной математики:

1. **Погрешности вычислений**. Любой численный результат на ЭВМ можно получить только с помощью конечной последовательности арифметических или логических операций. Ограниченность разрядной сетки ЭВМ и замена бесконечного сходящегося процесса конечным неизбежно приводит к возникновению погрешностей вычислений. Это вызывает необходимость проводить анализ возникающих погрешностей и производить гарантированные оценки точности вычислений.

2. **Анализ устойчивости** вычислительного алгоритма в численных методах имеет особое значение. Эта проблема связана с анализом критериев и условий роста погрешностей при реализации вычислительного алгоритма.

3. **Эффективность** (экономичность) вычислительного алгоритма. Для практического применения алгоритма весьма важна его эффективность. Ее иногда оценивают по количеству арифметических операций, необходимых для получения решения. Однако следует помнить, что часто уменьшение количества арифметических операций достигается в результате логического усложнения алгоритма, и поэтому программы для ЭВМ для такого логически усложненного алгоритма получаются столь неэкономичными, что весь выигрыш, полученный за счет снижения количества арифметических операций, может потеряться.

Основным вопросом численных методов является разработка вычислительного алгоритма, удовлетворяющего требованиям *высокой точности, устойчивости и экономичности*

ГЛАВА 1

Ошибки вычислений

1.1. Введение

Вычислить погрешность (ошибку) решения задачи было бы весьма просто, если бы мы знали точное решение. Она бы была получена путем сравнения вычисленного решения и точного. К сожалению, точного решения мы, как правило, не знаем. Выходом из создавшегося положения является изучение причин возникновения погрешностей и выработки способов их оценок.

Причины возникновения всевозможных погрешностей на пути решения задачи можно проследить, используя технологическую цепочку, представленную во введении на рис. 1.

1. При составлении математической модели объекта (процесса) неизбежно возникают погрешности в силу идеализации (упрощения) его действительных свойств, а также невозможности точного вычисления, измерения или наблюдения наиболее важных параметров объекта (процесса).

В качестве примера рассмотрим задачу об определении дальности полета (l) пушечного ядра, выпущенного под углом α к горизонту со скоростью v_0 при следующих предположениях:

а) Земля - инерциальная система отсчета;

б) ускорение свободного падения g постоянно;

в) кривизной Земли можно пренебречь и считать ее плоской;

г) действием воздуха на движущееся ядро можно пренебречь.

Как известно из школьного курса физики, при таких предположениях (выбранной схеме идеализации процесса) задача имеет достаточно простое решение:

$$l = \frac{v_0^2}{g} \cdot \sin 2\alpha ,$$

причем ядро движется по параболе.

Однако дальнейшие исследования этой задачи показывают, что только пренебрежение сопротивлением воздуха приводит к ошибкам в определении дальности до 15% (при $l = 1$ км, $\alpha = 45°$). Более строгие математические модели учитывают вращение Земли вокруг своей оси, зависимость плотности воздуха и ускорения свободного падения от высоты, кривизну земной поверхности и метеорологические данные, данные о скорости и направлении ветра,

давления воздуха и т.д.

2. При дискретизации математической модели возникают погрешности, связанные с ее упрощением и переходом от анализа функций непрерывной переменной к функциям дискретной переменной. Погрешность этого этапа решения задачи обычно называют погрешностями аппроксимации (приближения).

3. При реализации численных методов возникает большое количество погрешностей, связанных с конструктивными особенностями ЭВМ.

В данной главе будут рассмотрены причины возникновения и основные закономерности ошибок вычисления при реализации численных методов на ЭВМ.

1.2. Основные источники ошибок численных методов

Ограниченность разрядной сетки ЭВМ и ограниченность ее памяти являются источником возникновения различного рода ошибок при реализации вычислительных алгоритмов. Программируя математические формулы, использующие элементарные встроенные функции $\sin x, \cos x, \ln x$ и т.д., мы обычно забываем, что имеем дело не с функциями непрерывного аргумента, а с их дискретными аналогами. Например, простая функция $y = x$ на гипотетической ЭВМ с фиксированной одноразрядной арифметикой выглядит так, как это представлено на рис.1.1.

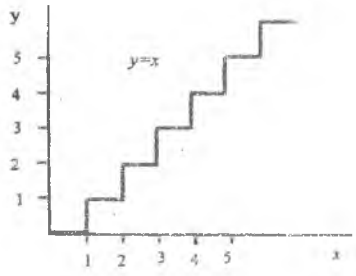


Рис 1.1

Попытка отыскания одного из корней уравнения

$x^2 + 0{,}4002x + 0{,}00008 = 0$, используя вычисления с точностью до 4-й значащей цифры, используя обычную формулу

$$x = \frac{-b - \sqrt{b^2 - 4ac}}{2a},$$

приводит к результату $x = -0{,}00015$ вместо точного решения $x = -0{,}0002$. Результат имеет ошибку в 25 %. Можно привести множество подобных примеров.

Однако ограниченность разрядной сетки ЭВМ не является единственным источником ошибок.

Рассмотрим ряд Тейлора для синуса:

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \ldots \qquad (1.1)$$

Ряд (1.1) сходится для любого x, но для вычисления $\sin x$ необходимо найти сумму бесконечного ряда чисел, что практически невозможно реализовать на ЭВМ. На практике вычисляют сумму какого-то числа первых членов ряда, а «хвост» ряда, содержащий бесконечное число членов, отбрасывают. В результате возникает неизбежная ошибка вычисления.

Исходя из вышесказанного, выделим следующие основные источники ошибок:

1. Ошибки, содержащиеся в исходной информации

Обычно возникают в результате неточности измерения параметров объекта, грубых просмотров, а также в связи с невозможностью представить в ЭВМ действительные числа конечной десятичной дробью (например число π, число e и т.д.).

Ошибки в исходной информации особенно сильно сказываются на результатах решения плохо обусловленных задач (некорректных задач), где даже относительно маленькая ошибка в 0,1% представления исходного данного может привести к погрешности в результатах вычислений в 70%, 100% и более процентов.

2. Ошибки ограничения

Данного рода погрешности возникают при ограничении бесконечного сходящегося процесса конечным числом операций. Например, вычисления функции $\sin x$ по формуле (1.1) для первых пяти членов ряда (четные члены ряда равны нулю) на ЭВМ приводят к модели

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + e_{огр},$$

где $e_{огр}$ *ошибка ограничения* равна сумме бесконечного числа отброшенных членов ряда. Точно вычислить $e_{огр}$ невозможно, однако ее можно оценить сверху. Для нашего случая величиной

$$|e_{огр}| < \frac{|x|^7}{7!}$$

3.Ошибки округления

Даже если исходная информация не содержит ошибок и вычислительный процесс некоторого алгоритма конечен (отсутствует ошибка ограничения), на ЭВМ невозможно производить точные вычисления без округления промежуточных результатов вычислений в силу ограниченности разрядной сетки ЭВМ.

Проблема округления чисел относится только к действительным числам. При выполнении арифметических операций с целыми числами потребность в их округлении не возникает.

Напомним, что любое действительное число можно представить в виде дроби (которую часто называют мантиссой), умноженной на целую степень 10. Пусть f мантисса действительного числа x, а e- его порядок. Тогда в общей форме число x можно записать следующим образом:

$$x = f \cdot 10^e$$

Если первая значащая цифра мантиссы не равна нулю, то число называется нормализованным. В общей форме мантиссу можно представить в виде

$$f = 0.a_1a_2a_3\ldots a_t,$$

где $a_i \in \{0,1,2,\ldots 9\}, a_1 \neq 0,$ t- размерность разрядной сетки ЭВМ. Очевидно, что f не может быть меньше 1/10 и больше 1, так как мантисса должна быть правильной дробью

Выполнение арифметических операций на ЭВМ рассмотрим на примере операций сложения и вычитания, которые в общем случае состоят из 3 этапов:

- на первом этапе выравниваются порядки действительных чисел, причем мантисса меньшего по абсолютной величине числа сдвигается вправо на столько разрядов, сколько необходимо для того, чтобы порядки стали одинаковыми;
- на втором этапе производится арифметическая операция

(сложение или вычитание);

- на последнем этапе реализуется операция округления результата, либо по общепринятым правилам округления до t значащих цифр в мантиссе, либо путем отбрасывания младших значащих цифр, для которых не нашлось места в разрядной сетке ЭВМ.

Например, рассмотрим два числа $x = f_x \cdot 10^e$ и $y = f_y \cdot 10^p$, причем $p<e$, тогда операцию сложения x и y в ЭВМ схематично можно представить следующим образом:

$$x + y = f_x \cdot 10^e + f_y \cdot 10^p = |\text{выравнивание порядка}| =$$

$$= f_x \cdot 10^e + \overline{f}_y \cdot 10^e = |\text{сложение}| =$$

$$= f_{x+y} \cdot 10^e + g_{x+y} \cdot 10^{e-t} = |\text{округление}| = f_{x+y} \cdot 10^e .$$

Любая из четырех арифметических операций дает в результате число, которое можно представить в виде

$$y = f_y \cdot 10^e + g_y \cdot 10^{e-t}, \qquad (1.2)$$

где $0{,}1 \le f_y < 1{,}0$, а $0 \le g_y < 1{,}0$.

Таким образом, при выполнении арифметических операций на ЭВМ возникает ошибка округления, которая пропорциональна величине $g_y \cdot 10^{e-t}$, и зависит от количества значащих цифр в ячейке памяти ЭВМ, поскольку g_y и e определяются только исходными числами, а t- зависит от разрядной сетки ЭВМ.

1.3. Распространение ошибок

Определение 1. *Абсолютная ошибка есть разность между истинным значением величины и ее приближенным значением*:

$$\Delta x = e_x = x - \bar{x}, \qquad (1.3)$$

где x - истинное значение величины, $\bar{x}$ - его приближенное значение.

Например, абсолютную погрешность арифметической операции из формулы (1.2) можно оценить величиной:

$$|e_y| = |g_y| \cdot 10^{e-t}$$

- для случая выполнения операции округления путем отбрасывания младших разрядов результатов вычисления и

$$|e_y| \le \frac{1}{2} \cdot |g_y| 10^{e-t}$$

- для симметричного способа округления, выполняемого по

общепринятым правилам.

Определение 2. *Относительная ошибка есть отношение абсолютной ошибки к приближенному значению*:

$$\delta_x = \frac{e_x}{\bar{x}}. \qquad (1.4)$$

Казалось бы, что более правильно определить ее как отношение абсолютной ошибки к точному значению, но обычно точное значение нам неизвестно.

Для величин, близких по значению к единице, абсолютная и относительная ошибки почти одинаковы.

Для рассмотренного выше примера оценим величину относительной ошибки округления результата арифметической операции:

$$\delta_y = \left|\frac{e_y}{\bar{y}}\right| = \left|\frac{g_y \cdot 10^{e-t}}{f_y \cdot 10^{e}}\right| \le \frac{1 \cdot 10^{e-1}}{0{,}1 \cdot 10^{e}} = 10^{-t+1}, \qquad (1.5)$$

так как $\max g_y = 1$, а $\min f_y = 0{,}1$.

Полученный результат показывает, что *относительные ошибки арифметических операций вообще не зависят от значений чисел, участвующих в ней, а определяются только разрядностью ЭВМ*.

Итак, в вычислительной машине в силу ряда причин невозможно точно представить как исходные числа, так и реализовать арифметические вычисления. Возникает естественный вопрос о том, как ошибка в исходных числах или в результатах арифметических операций распространяется далее в ходе вычислений. Становится ее влияние больше или меньше по мере того, как производятся последующие операции? Крайним случаем является вычитание двух почти равных чисел, где даже при очень маленьких ошибках в представлении обоих чисел относительная ошибка разности может оказаться очень большой. Причем эта ошибка может распространяться дальше при выполнении всех последующих арифметических операций.

Найдем выражение для оценок абсолютных и относительных ошибок результата основных арифметических операций (сложение, вычитание, умножение и деление).

Абсолютная ошибка сложения

Пусть даны числа $\bar{x}$ и $\bar{y}$, приближенно представляющие числа x и y и пусть e_x, e_y - соответствующие абсолютные ошибки представления чисел. Тогда в результате сложения имеем

$$x + y = \bar{x} + e_x + \bar{y} + e_y = \left(\bar{x} + \bar{y}\right) + \left(e_x + e_y\right),$$

Если обозначить e_{x+y} абсолютную ошибку суммы, тогда

$$e_{x+y} = e_x + e_y . \qquad (1.6)$$

Абсолютная ошибка разности

При выполнении операции вычитания аналогично получаем

$$e_{x+y} = e_x - e_y . \qquad (1.7)$$

При сравнении формул (1.6) и (1.7) может сложиться неправильное представление, что операция сложения увеличивает ошибку, а операция вычитания уменьшает ее. На самом деле все зависит от знаков ошибок в представлении чисел, участвующих в арифметических операциях. Если в формуле (1.6) e_x и e_y имеют разные знаки, то результирующая абсолютная ошибка уменьшается. На практике нас редко интересует знак абсолютной погрешности (более важным представляется величина отклонения результата от истинного значения), поэтому часто анализу подвергается не сама абсолютная ошибка, а ее модуль. В этом смысле формулы (1.6) и (1.7) можно объединить следующим образом:

$$|e_{x\pm y}| = |e_x| + |e_y| .$$

Абсолютная ошибка умножения

При выполнении операции умножения имеем

$$x \cdot y = (\bar{x} + e_x)\cdot(\bar{y} + e_y) = \bar{x}\cdot\bar{y} + \bar{x}\cdot e_y + \bar{y}\cdot e_x + e_x \cdot e_y .$$

Поскольку ошибки обычно гораздо меньше самих величин, то величиной $e_x \cdot e_y$ можно пренебречь, тогда ошибку произведения приближенно можно вычислить по формуле

$$e_{xy} \approx \bar{x}\cdot e_y + \bar{y}\cdot e_x \qquad (1.8)$$

Абсолютная ошибка деления

Для операции деления имеем

$$\frac{x}{y} = \frac{\bar{x}+e_x}{\bar{y}+e_y} = \frac{(\bar{x}+e_x)(\bar{y}-e_y)}{(\bar{y}+e_y)(\bar{y}-e_y)} = \frac{\bar{x}\cdot\bar{y} + \bar{y}\cdot e_x - \bar{x}\cdot e_y - e_x\cdot e_y}{y^2 - e_y^2} =$$

=|пренебрегая величинами e_y^2 и $e_x e_y$ большого порядка малости|≈

$$\approx \frac{\bar{x}\cdot\bar{y} + \bar{y}\cdot e_x - \bar{x}\cdot e_y}{y^2} \approx \frac{\bar{x}}{\bar{y}} + \frac{e_x}{\bar{y}} - \frac{\bar{x}}{\bar{y}^2}\cdot e_y .$$

Следовательно:

$$e_{x/y} \approx \frac{1}{y}\cdot e_x - \frac{\bar{x}}{\bar{y}^2}\cdot e_y \qquad (1.9)$$

После того как мы вывели формулы для распространения

абсолютных ошибок четырех основных арифметических операций, несложно вывести соответствующие формулы и для относительных ошибок.

Сложение:

$$\frac{e_{x+y}}{\bar{x}+\bar{y}} = \frac{\bar{x}}{\bar{x}+\bar{y}}\left(\frac{e_x}{\bar{x}}\right) + \frac{\bar{y}}{\bar{x}+\bar{y}}\left(\frac{e_y}{\bar{y}}\right),$$

откуда

$$\delta_{x+y} = \frac{\bar{x}}{\bar{x}+\bar{y}}\delta_x + \frac{\bar{y}}{\bar{x}+\bar{y}}\delta_y . \qquad (1.10)$$

Вычитание:

$$\delta_{x-y} = \frac{\bar{x}}{\bar{x}-\bar{y}}\delta_x - \frac{\bar{y}}{\bar{x}-\bar{y}}\delta_y . \qquad (1.11)$$

Умножение:

$$\frac{e_{x\cdot y}}{x\cdot y} = \frac{e_x}{\bar{x}} + \frac{e_y}{\bar{y}} ,$$

откуда

$$\delta_{x\cdot y} = \delta_x + \delta_y . \qquad (1.12)$$

Деление:

$$\frac{e_{x/y}}{\bar{x}/\bar{y}} = \frac{\bar{y}}{\bar{y}\cdot\bar{x}}\cdot e_x - \frac{\bar{x}\cdot\bar{y}}{\bar{y}^2\cdot\bar{x}}\cdot e_y = \frac{e_x}{\bar{x}} - \frac{e_y}{\bar{y}} ,$$

или

$$\delta_{x/y} = \delta_x - \delta_y . \qquad (1.13)$$

При использовании формул (1.6)- (1.13) важно четко понимать их смысл. Мы начинаем арифметическую операцию, имея в своем распоряжении два приближенных числа $\bar{x}$ и $\bar{y}$ с соответствующими ошибками e_x и e_y. Источник ошибок может быть любого происхождения: ошибки измерения, ограничения, округления и т.д. Формулы (1.6)- (1.13) дают способ вычисления ошибки результата операции как функции от $\bar{x}$, $\bar{y}$, e_x и e_y. При этом ошибка округления самой операции не учитывается. Если же в дальнейшем необходимо подсчитать, как распределяется в последующих арифметических операциях ошибка этого результата, то необходимо к вычисленной ошибке прибавить отдельно ошибку округления используемой операции.

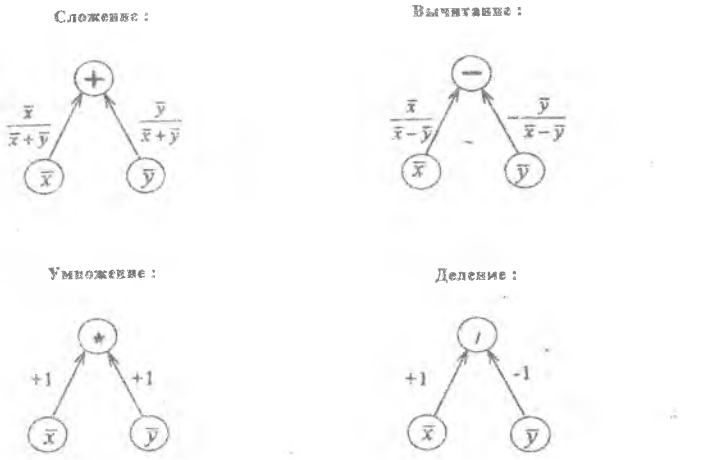


Рис.1.2. Графы вычислительных процессов основных арифметических операций

1.4. Графы вычислительных процессов

Для оценки общей ошибки вычислений некоторой последовательности арифметических операций существует достаточно удобный способ вычисления ошибок, основанный на графе вычислительного процесса.

Граф вычислительного процесса позволяет наглядно изобразить последовательность арифметических операций в некотором вычислении. С его помощью легко определить вклад любой ошибки, возникшей в процессе вычислений, в общую ошибку.

Пусть вершины графа суть переменные или арифметические операции. Дуги графа указывают направление вычислений и они помечены коэффициентами, оценивающими распространение ошибок.

Граф вычислительного процесса строится для анализа процесса распространения относительных ошибок в арифметических выражениях. Его следует читать снизу вверх в направлении дуг графа. Сначала выполняются операции, расположенные на каком-либо горизонтальном уровне, затем операции, расположенные на

более высоком уровне, и т.д. На рис.1.2. представлены графы вычислительных процессов основных арифметических операций.

Правило подсчета общей ошибки с использованием графа вычислительного процесса можно сформулировать следующим образом: относительная ошибка результата любой операции (кружка) входит в результат следующей операции, умножаясь на коэффициент у стрелки, соединяющей эти две операции.

В качестве примера рассмотрим выражение $u = (x+y)\cdot z$ и задачу оценки общей погрешности результата с учетом ошибок округления арифметических операций. Предположим, что $\delta_x, \delta_y, \delta_z$, - относительные ошибки округления чисел x, y, z при представлении их в ЭВМ, а r_+ и r^* относительные ошибки округления соответственно операций сложения и умножения.

Для нашего выражения, учитывая последовательность операций, граф вычислительного процесса будет выглядеть так, как это представлено на рис.1.3

Сначала рассмотрим операцию сложения (III уровень). Операция сложения использует числа x и y , заданные с относительными погрешностями δ_x и δ_y. Каждая из погрешностей входит в результат умножаясь на соответствующие коэффициенты $\frac{x}{x+y}$ и $\frac{y}{x+y}$, тогда ошибку операции сложения можно оценить величиной $\frac{x}{x+y}\cdot\delta_x+\frac{y}{x+y}\cdot\delta_y$, к которой следует прибавить ошибку округления, т.е. относительная ошибка операции сложения будет выглядеть $\frac{x}{x+y}\cdot\delta_x+\frac{y}{x+y}\cdot\delta_y+r_+$.

Исходный материал (числа и их ошибки) участвует еще в одной операции- умножения (I уровень графа), куда он передается, умножаясь на коэффициенты +1 и +1, тогда с учетом погрешности округления имеем

$$\delta_u = +1\left(\frac{x}{x+y}\cdot\delta_x+\frac{y}{x+y}\cdot\delta_y+r_+\right)+1\cdot\delta_z+r^*.$$

Более простым способом составления формулы оценки результирующей ошибки является движение от корня графа к его ветвям (метод сверху вниз). Следующая последовательность формул иллюстрирует идею этого метода: $\delta_u = +1(\quad)+1\ \delta_z+r^*$,

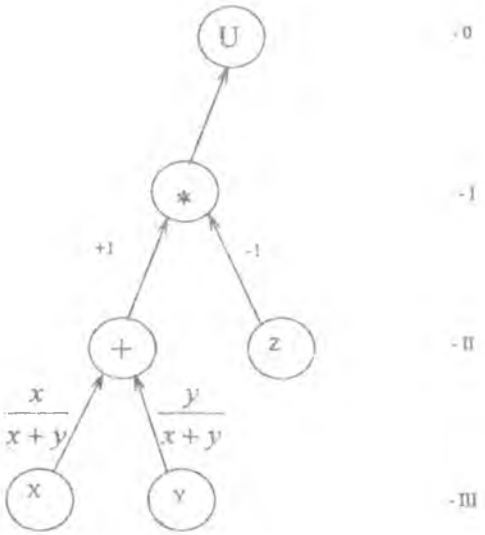


Рис. 1.3. Граф вычислительного процесса для выражения $u=(x+y)z$

$$\delta_u = \left(\frac{x}{x+y}\cdot\delta_x + \frac{y}{x+y}\cdot\delta_y + r_+\right) + \delta_z + r^*. \qquad (1.14)$$

Поскольку природа ошибок округления и представления чисел в ЭВМ одна - ограниченность ее разрядной сетки, то для относительных ошибок можно положить $\delta_x = \delta_y = \delta_z = r^* = r_+ = \delta$. Тогда формулу (1.14) можно упростить:

$$|\delta_u| = \delta\left[\left(\left|\frac{x}{x+y}\right| + \left|\frac{y}{x+y}\right| + 1\right) + 1 + 1\right] = \qquad (1.15)$$

$$\left|\text{при } x>0 \text{ и } y>0 \text{ и } \left|\frac{x}{x+y}\right| + \left|\frac{y}{x+y}\right| \le 1\right| \le 4\delta.$$

Учитывая формулу (1.5), величину относительной погрешности округления методом «отбрасывания» младших разрядов, можно оценить $\delta \le 10^{-t+1}$, тогда $|\delta_u| \le 4\cdot 10^{-t+1}$.

1.5. Общая формула для оценки погрешности вычисления функций

Основная задача теории погрешностей заключается в определении погрешности вычисления функции в зависимости от известных ошибок в представлении некоторой системы величин.

Пусть задана дифференцируемая функция

$$u = f(x_1, x_2, \ldots, x_n)$$

и пусть $|e_i| = |\Delta x_i|$ $(i=1,2,\ldots,n)$ - абсолютные погрешности аргументов функции. Тогда абсолютная погрешность функции приближенно равна дифференциалу функции (главной части приращения функции)

$$|\Delta u| \approx |du| = \left|\sum_i^n \frac{\partial f}{\partial x_i} e_i\right| \le \sum_i^n \left|\frac{\partial f}{\partial x_i}\right| \cdot |e_i|,$$

откуда

$$|e_u| = |\Delta u| \approx \le \sum_i^n \left|\frac{\partial f}{\partial x_i}\right| \cdot |e_i|. \qquad (1.16)$$

Разделив обе части выражения (1.16) на u, будем иметь оценку для относительной погрешности функции u:

$$|\delta_u| \le \sum_i^n \left|\frac{\frac{\partial f}{\partial x_i}}{u}\right| \cdot |e_i| = \sum_i^n \left|\frac{\partial}{\partial x_i} \ln f(x_1, \ldots, x_n)\right| \cdot |e_i| \approx \sum_i^n \left|\frac{\partial}{\partial x_i} \ln u\right| \cdot |e_i|.$$

Следовательно, за предельную относительную погрешность функции u можно принять

$$|\delta_u| \approx \sum_i^n \left|\frac{\partial}{\partial x_i} \ln u\right| \cdot |e_i| \qquad (1.17)$$

1.6. Практическая оценка погрешности вычислительных модулей

Вывод формулы оценки итоговой ошибки арифметических операций с помощью графа вычислительного процесса для произвольного алгоритма не всегда простая задача. Кроме того, в процессе вычислений в алгоритмах могут использоваться стандартные или специальные подпрограммы, относительно погрешностей вычислений которых ничего не известно. В то же

время оценить погрешность арифметических операций можно в результате проведения вычислительного эксперимента.

В работе [12] показано, что с увеличением числа арифметических операций количество решений уравнения $f(x,y)=0$ существенно возрастает, даже если функция имеет единственный корень. На рисунке 1.4 показано изменение диаметра области решений уравнения $f(x,y)=(x-1)^m+(y-1)^m=0$ в зависимости от числа арифметических операций. Здесь $\Omega_E=\{(x,y)\in| \ f(x,y)=0\}$ и

$$f(x,y)=1/((x^m-C_m^1x^{m-1}+..+(-1)^mC_m^mx^0)+(y^m-C_m^1y^{m-1}+...+(-1)^mC_m^my^0))$$

С точки зрения вычисляемой функции она эквивалентна предыдущей, однако количество арифметических операций в ней значительно больше. При m=4 она реализует 54 арифметические операции, а при m=40 около 440 и т.д. Как видно из рисунка, область решения исходного уравнения существенно увеличивается с ростом количества арифметических операций.

Полученный феномен можно объяснить следующими соображениями.

Пусть $x=x_t10^t+x_{t-1}10^{t-1}+\cdots+x_0$ и $a=a_t10^t+a_{t-1}10^{t-1}+\cdots+a_0$ - представление десятичных чисел в t-разрядной арифметике. Рассмотрим задачу нахождения корней уравнения $(x-a)^2=0$ на ЭВМ с t-значной арифметикой. Выражение $(x-a)^2$ можно записать в виде

$$(x-a)^2=(x_t-a_t)^210^{2t}+2(x_t-a_t)(x_{t-1}-a_{t-1})10^{2t-1}+$$
$$+[(x_{t-1}-a_{t-1})^2+(x_t-a_t)(x_{t-2}-a_{t-2})]10^{2t-2}+$$
$$+2[(x_t-a_t)(x_{t-3}-a_{t-3})+(x_{t-1}-a_{t-1})(x_{t-2}-a_{t-2})]10^{2t-3}+$$
$$+\cdots+2(x_1-a_1)(x_0-a_0)10+(x_0-a_0)^2.$$

Тогда исходное уравнение, с учетом ограниченности разрядной сетки (при условии, что округление результата производится путем отбрасывания младших разрядов), можно заменить системой уравнений:

$$\begin{cases}(x_t-a_t)^2=0,\\(x_t-a_t)(x_{t-1}-a_{t-1})=0,\\(x_{t-1}-a_{t-1})^2+2(x_t-a_t)(x_{t-2}-a_{t-2})]=0,\\\cdots\\(x_t-a_t)(x_0-a_0)+(x_{t-1}-a_{t-1})(x_1-a_1)=0.\end{cases}$$

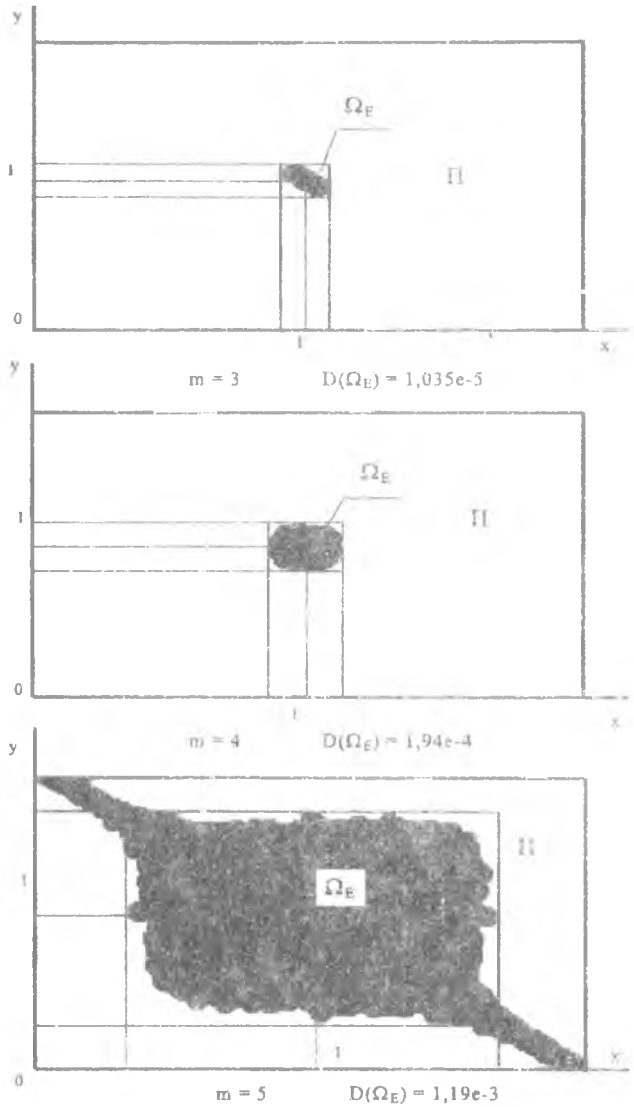


Рис. 1.4. Зависимость диаметра $D(\Omega_E)$ области решения функции $f(x,y)$ от числа арифметических операций

Например при $t=4$ система примет вид

$$\begin{cases} (x_4 - a_4)^2 = 0, \\ (x_4 - a_4)(x_3 - a_3) = 0, \\ (x_3 - a_3)^2 + 2(x_4 - a_4)(x_2 - a_2)] = 0, \\ (x_4 - a_4)(x_1 - a_1) + (x_3 - a_3)(x_2 - a_2) = 0, \\ (x_2 - a_2)^2 + 2(x_4 - a_4)(x_0 - a_0) = 0. \end{cases} \quad (1.18)$$

Корнями системы уравнений (1.18) являются

$$\begin{cases} x_4 = a_4, \\ x_3 = a_3, \\ x_2 = a_2, \\ \forall x_1, \\ \forall x_0. \end{cases} \quad (1.19)$$

Однако в реальных вычислениях на ЭВМ, учитывающих последовательность выполнения арифметических операций и операций округления чисел, реализуются не все корни (1.19). В то же время, как показывают вычислительные эксперименты, их достаточно большое количество.

Полученные результаты можно существенно усилить, если в процессе вычислений искусственно ограничить размеры разрядной сетки ЭВМ.

Пусть t - размер разрядной сетки гипотетической ЭВМ. Переопределим арифметические операции следующим образом:

$$a \pm b = \begin{cases} 0, & \left| [a]_t \pm [b]_t \right| < \dfrac{\max\left\{ \left| [a]_t \right|, \left| [b]_t \right| \right\}}{10^t} \\ \left[[a]_t \pm [b]_t \right]_t, & \text{иначе} \end{cases} \quad (1.20)$$

$$a * b = \left[[a]_t * [b]_t \right]_t, \quad (1.21)$$

$$a / b = \left[[a]_t / [b]_t \right]_t. \quad (1.22)$$

Вычисляя функцию, при различных значениях параметра t, можно определить величины погрешности арифметических операций экспериментально. В качестве примера рассмотрим задачу оценки погрешности арифметических операций вычисления определенного интеграла:

$$I = \int_0^{2.5} \frac{x^2}{10\pi \sin x} dx.$$

Пусть $I(t) = I_{ист} + e_{мет} + e_{окр}(t)$, t - размер разрядной сетки

ЭВМ. Тогда $\Delta I(t) = I(t) - I(t-1) = e_{окр}(t) - e_{окр}(t-1) = \Delta e(t)$.

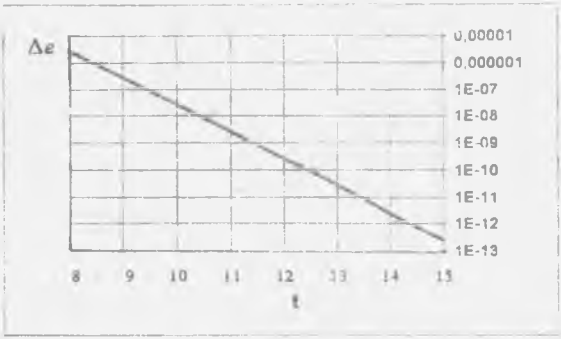


Рис. 1.5. Зависимость приращения погрешности округления от разрядности ЭВМ

На рисунке 1.5 показана зависимость приращения абсолютной погрешности округления при вычислении определенного интеграла от разрядности ЭВМ. Как видно из рисунка, $\lg \Delta e(t)$ линейно зависит от t, т.е. $\lg \Delta e(t) = \lg(I(t) - I(t-1)) = \gamma t + b$ или $\Delta e(t) = 10^{\gamma t + b}$. Вычислив $I(t)$ для трех значений t, можно определить параметры γ и b.

С другой стороны, $I(t) = I_{ист} + KI_{ист}\delta \approx I_{ист} + KI_{ист}10^{-t+1}$, т.е. $e(t) = KI_{ист}10^{-t+1}$, тогда $\Delta e(t) = KI_{ист}\frac{9}{10}10^{-t+1}$ и, сравнивая формулы, получим $K = -\frac{1}{9I_{ист}}10^{(1-\gamma)t+b-1}$. В нашем случае при n=200 K=135,83 и, следовательно, $e_{окр} = e(15) \approx 2{,}75 \cdot 10^{-13}$. На последнем рисунке показано изменение $e_{окр}$ в зависимости от n, построенная по результатам вычислительного эксперимента.

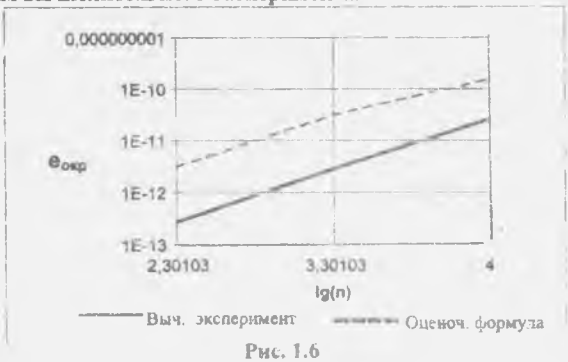


Рис. 1.6

ГЛАВА 2

Практическое вычисление функций на ЭВМ

2.1. Введение

Программируя математические выражения на ЭВМ, сегодня мы уже редко задумываемся, каким же образом вычислительная машина, способная выполнять лишь простейшие арифметические операции, вычисляет значения таких функций, как $\sin x$, $\mathrm{tg}\, x$, $\ln x$, e^x и т.д. В свое время усилия многих талантливых математиков и программистов были затрачены на разработку численных методов и программ, которые теперь позволяют вычислять значения многих элементарных функций эффективно и точно. Данная глава как раз и посвящена проблемам практического вычисления функций на ЭВМ.

Все многообразие существующих методов вычисления функций на ЭВМ можно условно разбить на три группы.

1. Группа методов, для которых доступна полная информация о рассматриваемой функции (**методы приближения функции**). Для этих методов считается, что известен аналитический вид функции (задано ее математическое выражение). В этом случае при необходимости можно реализовать полное исследование математических свойств функции. Основной задачей этой группы методов является *замена (приближение) исходной функции некоторой другой функцией, которую можно вычислить на ЭВМ.*

2. Группа методов, для которых доступна лишь информация о значениях функции на конечном множестве значений ее аргументов. При этом предполагается, что значения аргументов функции и самой функции заданы точно. Задачей этой группы методов является *восполнение значений функций с дискретного множества точек на непрерывную область.* Другими словами, как, имея дискретное, конечное множество значений функции, вычислить ее значение для произвольного значения аргумента. Эти методы получили названия **методы интерполирования функции**.

3. Группа методов, для которых доступна недостоверная информация о значениях функции на конечном множестве ее аргументов, т.е. значения функции или ее аргументов могут быть искажены разного рода неточностями (погрешностями измерений, округления, промахами и т.д.). Задачей является не только восстановление функции с дискретного множества точек на непрерывную область, но и устранение (фильтрация) погрешностей в исходной информации. Эта группа методов получила название **методы аппроксимация функции.**

2.2. Приближение функций

Пусть нам известна некоторая функция $y = f(x)$, для которой мы можем провести полное исследование ее свойств (вычислить любые производные, найти экстремум функции и т.д.). Требуется разработать численный метод вычисления ее значений для произвольных значений аргумента x.

Как известно, в ЭВМ непосредственно вычисляются функции, использующие элементарные арифметические операции сложения, вычитания, умножения и деления. Среди известных элементарных функций особенностям организации вычислительного процесса в ЭВМ удовлетворяют степенные функции, а наиболее сложными их представителями являются полиномиальные функции (многочлены). Таким образом, можно считать, что функция, с помощью которой мы будем приближать (заменять) исходную функцию $f(x)$, является полиномом $P_n(x)$:

$$P_n(x) = a_0 + a_1 \cdot x + \ldots + a_n \cdot x^n ,$$

где $a_0, a_1, \ldots a_n$ - коэффициенты многочлена.

Возникает вопрос: до какой степени правомочна такая постановка задачи, когда мы произвольную функцию $f(x)$ можем заменить полиномиальной функцией $P_n(x)$?

Для класса непрерывных на $[a,b]$ функций имеется положительный ответ на поставленный вопрос. На самом деле, из линейной алгебры известно, что последовательность многочленов $1, x, x^2, x^3, \ldots, x^n, \ldots$ образует бесконечномерный базис в пространстве функций непрерывных на $[a,b]$, а тогда любую функцию этого класса можно представить как линейную комбинацию базисных функций, т.е.

$$f(x)=\alpha_0+\alpha_1\cdot x+\alpha_2\cdot x^2+\ldots+\alpha_n\cdot x^n+\ldots.$$

Ограничивая бесконечный степенной ряд,мы имеем возможность приближать функцию $f(x)$ полиномом с любой наперед заданной степенью точности.

2.3. Формула Тейлора. Ряд Тейлора

Наиболее простым и достаточно эффективным способом приближения функций является использование формулы Тейлора для разложения функций в степенной ряд.

Пусть задана непрерывная функция $f(x)$, имеющая непрерывные производные до порядка $(n+1)$ включительно. Из математического анализа известно, что такую функцию можно разложить в окрестностях точки x_0 по степеням $(x-x_0)$ в ряд Тейлора:

$$f(x)=f(x_0)+\frac{f'(x_0)}{1!}(x-x_0)+\ldots+\frac{f^{(n)}(x_0)}{n!}(x-x_0)^n+R_n(\xi), \quad (2.1)$$

где $R_n(x)$- ошибка ограничения, связанная с заменой при вычислении $f(x)$ бесконечного степенного ряда первыми его n членами. Ошибку ограничения можно оценить по формуле

$$R_n(x)=\frac{f^{(n+1)}(\xi)}{(n+1)!}(x-x_0)^{n+1}, \quad (2.2)$$

где ξ находится между x и x_0.

Формула Тейлора не только дает возможность организовать численный метод вычисления значений функции $f(x)$, но и оценить величину ошибки ограничения по формуле (2.2). При ее использовании от вычислителя требуется определить точку x_0, в окрестностях которой будет производиться разложение функции. При выборе x_0 следует руководствоваться соображениями точности представления коэффициентов ряда (2.1) и величиной рабочего диапазона, внутри которого будут производиться вычисления.

Рассмотрим простой пример разложения в ряд Тейлора функции $\sin x$.Найдем соответствующие производные для функции $\sin x$, в результате получим последовательность функций:

$\cos x,-\sin x,-\cos x,\sin x,\ldots.$

Если x_0=0, то последовательность функций превратится в последовательность чисел 1,0,-1,0,1,0..., тогда по формуле (2.1) мы имеем

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \ldots$$

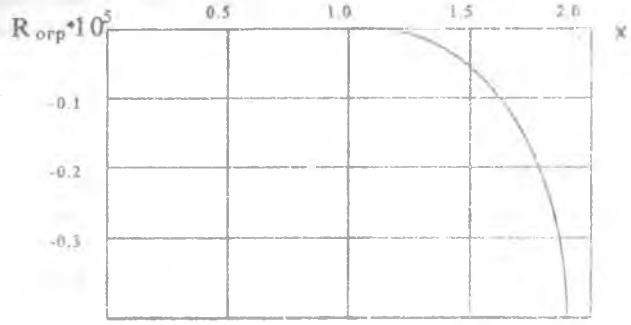


Рис. 2.1. Зависимость $R_{огр}$ от x при представлении функции sin x в ряд Тейлора

Попытаемся оценить величину ошибки ограничения при $n = 7$ и $x > 0$:

$$\left|R_{ogp}\right| = \left|R_7\right| = \left|\frac{\sin\xi \cdot x^8}{8!}\right| = \frac{\left|\sin\xi\right| \cdot \left|x^8\right|}{8!} < \frac{x^8}{8!}, \qquad (2.3)$$

поскольку $|\sin\xi| < 1$ для любых ξ. Из полученной оценки видно, что ошибка ограничения зависит от x и если не изменить число членов ряда в представлении функции $\sin x$, то для достаточно больших x $|R_{огр}|$ может превысить 1. На рис.2.1 представлен график зависимости ошибки ограничения для функции $\sin x$ от x.

Из рисунка видно, что погрешность быстро нарастает по мере удаления x от x_0 и в рабочем диапазоне вычисления функции $\sin x$ (на отрезке $[0,90^\circ]$) неравномерна для различных x.

2.4. Полиномы Чебышева

Ряды Тейлора дают неравномерную сходимость при разложении функции в степенной ряд. Возникает естественная идея поиска такого многочлена $T_n(x)$, чтобы максимальная ошибка приближения функции $f(x)$ была бы наименьшей.

Данная задача была решена великим русским математиком П.Л.Чебышевым и получила название задачи о наилучшем

приближения. Дадим ей следующую формальную постановку.

Пусть задана некоторая функция $f(x)$, которую на $[a,b]$ мы собираемся приблизить многочленом $T_n(x) = a_0 + a_1 \cdot x + ... + a_n \cdot x^n$ таким образом, чтобы:

$$\min_{a_0, a_1, ... a_n} \max_{x \in [a,b]} |f(x) - T_n(x)|,$$

т.е. подобрать такие коэффициенты $a_0, a_1, ... a_n$, чтобы максимальная величина модуля разности между $f(x)$ и $T_n(x)$ была наименьшей для любых x.

Определение. *Полиномом Чебышева называется многочлен вида* $T_n(x) = \cos(n \cdot \Theta)$, где $\Theta = \arccos x$.

Из определения может сложиться впечатление, что полиномы Чебышева - это тригонометрические функции. Однако это не так. Используя известные формулы тригонометрии, найдем первые три многочлена Чебышева:

$$T_0(x) = \cos(0) = 1,$$

$$T_1(x) = \cos(\arccos(x)) = x, \qquad (2.4)$$

$$T_2(x) = \cos(2\arccos(x)) = \cos^2(\arccos(x)) - \sin^2(\arccos(x)) = x^2 - (1 - x^2) = 2x^2 - 1.$$

Продолжая далее, можно было бы найти и все остальные многочлены Чебышева $T_3(x), ..., T_n(x)$. Однако для вычисления многочленов Чебышева практичнее использовать следующее рекуррентное соотношение:

$$T_{n+1}(x) = 2x \cdot T_n(x) - T_{n-1}(x). \qquad (2.5)$$

Свойства многочленов Чебышева

1. Учитывая формулу (2.5), можно установить, что

$$T_n(x) = 2^{n-1} \cdot x^n + ..., \qquad n \geq 1,$$

т.е. *коэффициент при старшем члене многочлена Чебышева равен* 2^{n-1}.

2. Полиномы Чебышева $T_0(x), T_1(x), ..., T_n(x)$ *образуют ортогональный базис с весом* $\frac{1}{\sqrt{1-x^2}}$ на множестве функций непрерывных на [-1,1], что означает:

$$\int_{-1}^{1} T_m(x)T_n(x)\frac{dx}{\sqrt{1-x^2}} = \begin{cases} 0, & m \neq n \\ \frac{\pi}{2}, & m = n \neq 0 \\ \pi, & m = n = 0 \end{cases} \qquad (2.6)$$

3. Многочлены Чебышева *сводят к минимуму максимальную ошибку приближения*, т.е. являются многочленами наилучшего приближения для класса функции непрерывных на отрезке [1,1].

Приведем схему доказательства этого важного факта. Чебышев показал, что точная верхняя грань многочлена $T_n(x)/2^{n-1}$ среди всех многочленов $P_n(x)$ со старшими коэффициентами 1 на отрезке [-1,1] наименьшая.

Действительно, $|T_n(x)| = |\cos(n\cdot\Theta)| \le 1$, откуда $\max|T_n(x)| = 1$, тогда $\max\left|T_n(x)/2^{n-1}\right| = \frac{1}{2^{n-1}}$, причем экстремумы принимают попеременно положительные и отрицательные значения на отрезке [-1,1], т.к. $T_n(x) = \cos(n\theta)$ - гармоническая функция. Количество экстремумов равно $n+1$. Рассмотрим разность: $\varphi_{n-1}(x) = \frac{T_n(x)}{2^{n-1}} - P_n(x)$, которая является многочленом степени $n-1$ (поскольку члены x^n уничтожаются). Если экстремальное значение у $P_n(x)$ меньше, чем у $T_n(x)$, то в $n+1$ экстремальных точках полинома $T_n(x)$ функция $\varphi_{n-1}(x)$ принимает по очереди положительные и отрицательные значения. Следовательно, $\varphi_{n-1}(x)$ имеет n действительных корней, что противоречит степени многочлена ($n-1$). Тогда $\varphi_{n-1}(x) \equiv 0$ или $P_n(x) = \frac{T_n(x)}{2^{n-1}}$.

Последнее свойство полиномов Чебышева представляет большой интерес для . численного анализа. Если какая-либо ошибка приближения может быть выражена многочленом Чебышева степени n, то любое другое выражение для ошибки в виде многочлена степени n, имеющего тот же самый старший коэффициент, будет иметь на [-1, 1] большую максимальную ошибку, чем чебышевское.

Практика использования полиномов Чебышева для решения задачи приближения функции $f(x)$ заключается в следующем.

Поскольку система функций $T_0(x), T_1(x), \ldots, T_n(x)$ образует базис, то на [-1,1] любую функцию можно представить как линейную комбинацию $T_i(x)$:

$$f(x) = \alpha_0 \cdot T_0(x) + \alpha_1 \cdot T_1(x) + \ldots + \alpha_n \cdot T_n(x). \qquad (2.7)$$

Коэффициенты разложения можно определить, используя ортогональности (2.6) полиномов Чебышева. Для определения α_0 почленно умножим правую и левую часть выражения (2.7) на $\frac{T_0(x)}{\sqrt{1-x^2}}$. Проинтегрировав по x, получим

$$\int_{-1}^{1} f(x)\frac{T_0(x)}{\sqrt{1-x^2}}dx = \alpha_0 \int_{-1}^{1}\frac{T_0^2(x)}{\sqrt{1-x^2}}dx + \ldots + \alpha_n \int_{-1}^{1}\frac{T_n(x)\cdot T_0(x)}{\sqrt{1-x^2}}dx .$$

Учитывая ортогональность, имеем

$$\int_{-1}^{1} f(x)\frac{T_0(x)}{\sqrt{1-x^2}}dx = \alpha_0 \cdot \pi$$

или $\alpha_0 = \frac{1}{\pi}\cdot\int_{-1}^{1} f(x)\frac{T_0(x)}{\sqrt{1-x^2}}dx$.

Аналогично можно вычислить все остальные коэффициенты разложения (2.7):

$$\alpha_k = \frac{2}{\pi}\cdot\int_{-1}^{1} f(x)\frac{T_k(x)}{\sqrt{1-x^2}}dx, \qquad k = 1,2,\ldots n \qquad (2.8)$$

Единственной проблемой разложения функции $f(x)$ по полиномам Чебышева является вычисление достаточно сложных интегралов (2.8).

2.5. Экономизация степенных рядов

Полиномы Чебышева дают очень хорошее приближение функции, но эти приближения довольно сложно вычислять. Существует относительно простой метод корректировки традиционного разложения функции $f(x)$ в степенной ряд (например, в ряд Тейлора) до разложения функции по полиномам Чебышева.

Пусть дан отрезок степенного ряда функции на [-1,1]:

$$f(x) = a_0 + a_1 \cdot x + \ldots + a_n \cdot x^n. \qquad (2.9)$$

Используя представление полиномов Чебышева степеней 0,1,2,

..., можно построить таблицу для вычисления степенных функций $1, x, x^2$ в разложении по полиномам Чебышева:

$$1 = T_0(x), \qquad x^3 = \frac{1}{4}\left(3T_1(x) + T_3(x)\right),$$

$$x = T_1(x), \qquad x^4 = \frac{1}{8}\left(3T_0(x) + 4T_2(x) + T_4(x)\right) \qquad (2.10)$$

$$x^2 = \frac{1}{2}\left(T_0(x) + T_2(x)\right) \quad \ldots$$

Подстановка (2.10) в (2.9) превращает степенной ряд (2.9) в ряд, разложенный по полиномам Чебышева:

$$f(x) = b_0 + b_1 \cdot T_1(x) + b_2 \cdot T_2(x) + \ldots + b_n \cdot T_n(x). \qquad (2.11)$$

Для широкого класса функций разложение функции по чебышевским полиномам сходится много быстрее. Следовательно, мы можем надеяться, что b_k в формуле (2.11) убывают гораздо быстрее, чем a_k в (2.9). Тогда, не уменьшая общую точность приближения функции $f(x)$ (за счет улучшения структуры ее представления), мы можем понизить степень многочлена, удалив из (2.11) последний член разложения, если $\left|b_n \cdot T_n(x)\right| < e_{0\Gamma}$.

Последнее означает, что с помощью многочлена меньшей степени мы имеем возможность вычислять функцию без ущерба точности ее представления в ЭВМ.

2.6. Рациональные приближения

Некоторые функции нельзя достаточно точно приблизить многочленами. Например, функции, принимающие бесконечные значения в некоторой точке числовой оси, или функции, имеющие горизонтальные асимптоты. Рациональные функции способствуют получению хороших приближений.

В настоящее время теория приближения рациональными функциями находится в запутанном, но быстро развивающемся состоянии.

Наиболее сложную проблему составляет вопрос выбора вида рациональной функции: $\frac{N(x)}{D(x)}$ для заданной функции $f(x)$, где $N(x)$ и $D(x)$- многочлены. При выборе вида рациональной функции следует учитывать всю имеющуюся априорную информацию: симметрию приближаемой функции, точки разрыва,

нули функции и т.д.

Проиллюстрируем идею метода на конкретном примере. Пусть вид рациональной функции определен в виде частного от деления двух полиномов третьего порядка:

$$\frac{b_0 + b_1 \cdot x + b_2 \cdot x^2 + b_3 \cdot x^3}{1 + c_1 \cdot x + c_2 \cdot x^2 + c_3 \cdot x^3} \qquad (2.12)$$

Для функции $f(x)$ по формуле Тейлора получено разложение ее в степенной ряд:

$$f(x) = a_0 + a_1 \cdot x + a_2 \cdot x^2 + ... + a_7 \cdot x^7 + ... \qquad (2.13)$$

Приравнивая (2.12) и (2.13) и освобождаясь от знаменателя, получим

$$b_0 + b_1 \cdot x + b_2 \cdot x^2 + b_3 \cdot x^3 = \left(1 + c_1 \cdot x + c_2 \cdot x^2 + c_3 \cdot x^3\right) \cdot \left(a_0 + a_1 \cdot x + a_2 \cdot x^2 + ...\right).$$

Раскрывая скобки и приравнивая коэффициенты при одинаковых степенях x, получаем систему уравнений, из которой определим неизвестные коэффициенты $b_0, b_1, b_2, c_1, c_2, c_3$:

$$\begin{cases} b_0 = a_0, \\ b_1 = a_1 + a_0 \cdot c_1, \\ b_2 = a_2 + a_1 \cdot c_1 + a_0 \cdot c_2, \\ b_3 = a_3 + a_2 \cdot c_1 + a_1 \cdot c_2 + a_0 \cdot c_3, \\ 0 = a_4 + a_3 \cdot c_1 + a_2 \cdot c_2 + a_1 \cdot c_3, \\ 0 = a_5 + a_4 \cdot c_1 + a_3 \cdot c_2 + a_2 \cdot c_3, \\ 0 = a_6 + a_5 \cdot c_1 + a_4 \cdot c_2 + a_3 \cdot c_3. \end{cases}$$

Последние три уравнения отражают тот факт, что в числителе рационального приближения коэффициенты многочлена при степенях x, выше третьей, равны нулю.

При использовании данного метода возникает методическая ошибка метода, которую можно оценить следующим образом.

Вычислим, каким был бы коэффициент b_7, если бы он был включен в это приближение, и разделим его на величину знаменателя:

$$e_{огр} = \frac{\left(a_7 + a_6 \cdot c_1 + a_5 \cdot c_2 + a_4 \cdot c_3\right) \cdot x^7}{1 + c_1 \cdot x + c_2 \cdot x^2 + c_3 \cdot x^3}.$$

Для повышения точности приближения функции $f(x)$ рациональной функцией целесообразно степенной ряд (2.13) предварительно подвергнуть экономизации.

2.7. Интерполяция функций

Довольно часто информация о функции $f(x)$ задана не аналитически, а в виде конечного множества ее узловых точек (дискретно). В то же время для реализации вычислительных алгоритмов обычно требуется восстановить функцию с дискретного множества точек на непрерывную область ее определения x.

Приближенная замена функции $f(x)$, заданной на множестве отдельных точек x_i, $i = 0,...n$ функцией $F(x)$ некоторого класса, значения которой в точках x_i совпадают с соответствующими значениями функции $f(x)$, называется **интерполяцией функции** $f(x)$ функцией $F(x)$.

Точки x_i называют **узлами интерполяции**, а $F(x)$- **интерполянтом** или интерполирующей функцией. При этом предполагается, что значения функции $f(x)$ в узлах x_i известны точно. Множество узлов интерполяции

$$x_i < x_1 < x_2 <...< x_n \qquad (2.14)$$

называют **интерполяционной сеткой**.

Пусть на интерполяционной сетке (2.14) задана своими значениями некоторая функция:

$y_0 = f(x_0), y_1 = f(x_1),...y_n = f(x_n)$.

Требуется построить некую функцию $F(x)$, чтобы выполнялись условия:

$$\begin{cases} F(x_0) = y_0, \\ F(x_1) = y_1, \\ ... \\ F(x_n) = y_n. \end{cases} \qquad (2.15)$$

Геометрически процесс интерполирования дискретно (таблично) заданной функции $f(x)$ интерполянтом $F(x)$ заключается в проведении графика функции $F(x)$ через все узловые точки x_i функции $f(x)$ (рис.2.2).

Если класс функции $f(x)$ не указан, т.е. не описаны ее свойства, то задача интерполирования функции не является корректной. Действительно, отсутствие информации о характере поведения функции $f(x)$ между узлами x_i может привести к непредсказуемым погрешностям при построении интерполянта

$F(x)$, и задача не может быть решена точно.

На ЭВМ в качестве интерполянтов $F(x)$ обычно используются полиномы степени n. В этом случае система уравнений (2.15) превращается в систему линейных уравнений относительно неизвестных коэффициентов $a_0, a_1, \ldots a_n$:

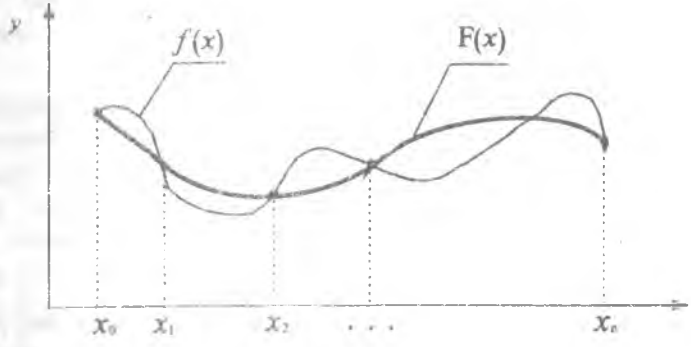


Рис.2.2. Интерполирование функции $f(x)$

$$\begin{cases} a_0 + x_0 \cdot a_1 + \ldots + x_0^n \cdot a_n = y_0 , \\ a_0 + x_1 \cdot a_1 + \ldots + x_1^n \cdot a_n = y_1 , \\ \ldots \\ a_0 + x_n \cdot a_1 + \ldots + x_n^n \cdot a_n = y_n . \end{cases} \qquad (2.16)$$

Система уравнений (2.16) состоит из $n+1$ уравнения для $n+1$ неизвестных, и если $x_i \neq x_j$, то можно показать, что система (2.16) имеет единственное решение.

Решив системы (2.16), мы находим неизвестные коэффициенты для интерполянта вида

$$F(x) \equiv a_0 + a_1 \cdot x + a_2 \cdot x^2 + \ldots + a_n \cdot x^n .$$

Единственность решения означает, что разные по форме способы построения интерполяционного многочлена дают в результате один и тот же полином .

2.8. Полином Лагранжа

Для построения интерполяционного многочлена прямым методом необходимо предварительно решить систему линейных

уравнений (2.16). Интерполяционная формула Лагранжа не требует решения системы уравнений (2.16). В общем виде полином Лагранжа можно представить формулой

$$L_n(x)=\sum_{k=0}^{n} y_k \frac{(x-x_0)\cdot(x-x_1)\ldots(x-x_{k-1})\cdot(x-x_{k+1})\ldots(x-x_n)}{(x_k-x_0)\cdot(x_k-x_1)\ldots(x_k-x_{k-1})\cdot(x_k-x_{k+1})\ldots(x_k-x_n)}, \quad (2.17)$$

где $x_0, x_1, \ldots x_n$ - узлы интерполяционной сетки, $y_0, y_1, \ldots y_n$ - значения функции $f(x)$ в узловых точках.

Каждый из слагаемых формулы (2.17), как нетрудно убедиться, является полиномом степени n, следовательно $L_n(x)$ - также есть полином n-й степени (как сумма многочленов n-й степени). Структура формулы (2.17) построена таким образом, чтобы выполнялось условие $L_n(x_k)=y_k$, в чем нетрудно убедиться.

Если функция $f(x)$ достаточно гладкая, т.е. имеет непрерывные производные $f'(x),\ldots,f^{(n+1)}(x)$ вплоть до $(n+1)$-порядка включительно, то погрешность интерполяции (остаточный член), определяемую формулой

$$R_n(x)=f(x)-L_n(x),$$

можно оценить следующим образом:

$$|R_n(x)| \le \frac{M_{n+1}}{(n+1)!}|\Pi_{n+1}(x)|, \quad (2.18)$$

где

$$M_{n+1}=\max_{x_0\le x\le x_n}\left|f^{(n+1)}(x)\right|,$$

$$\Pi_{n+1}(x)=(x-x_0)\cdot(x-x_1)\ldots(x-x_n).$$

Полином Лагранжа полезен тем, что в явном виде содержит значение функции y_i.

2.9. Интерполяционная формула Ньютона

Рассмотрим регулярную интерполяционную сетку с равноотстоящими узлами: $x_i=x_0+i\cdot h$, где h- шаг интерполяции, $i=0,1,\ldots n$.

Интерполяционную формулу будем искать в виде

$$P_n(x)=a_0+a_1\cdot(x-x_0)+a_2\cdot(x-x_0)\cdot(x-x_1)+\ldots+a_n\cdot(x-x_0)\cdot(x-x_1)\ldots(x-x_{n-1}). \quad (2.19)$$

Предварительно составим таблицу конечных разностей

функции $f(x)$ (таблица 2.1). В таблице приняты следующие обозначения:

$\Delta y_k = y_{k+1} - y_k$,

$\Delta^2 y_k = \Delta\, y_{k+1} - \Delta\, y_k$,

$\Delta^n y_0 = \Delta^{n-1} y_1 - \Delta^{n-1} y_0$

Таблица 2.1

Конечные разности функции *f(x)*

x	y	$\Delta\, y$	$\Delta^2 y$		$\Delta^n y$
x_0	y_0				
x_1	y_1	$\Delta\, y_0$	$\Delta^2 y_0$		
x_2	y_2	$\Delta\, y_1$	$\Delta^2 y_1$		
x_3	y_3	$\Delta\, y_2$	...		$\Delta^n y_0$
...	...	...			
x_n	y_n	$\Delta\, y_{n-1}$	$\Delta^2 y_{n-2}$		

Можно показать, что для того, чтобы выполнялись условия интерполяции $P_n(x_i) = y_i$, $i = 0,1,...n$, необходимо и достаточно, чтобы

$$\Delta^i P_n(x_0) = \Delta^i y_0, \quad i = 0,1,...n \tag{2.20}$$

или

$\Delta^0 P_n(x_0) = a_0 = y_0$,

$\Delta^1 P_n(x_0) = P_n(x_1) - P_n(x_0) = a_0 + a_1(x_1 - x_0) - a_0 = a_1 \cdot h = \Delta y_0$,

$\Delta^2 P_n(x_0) = \Delta P_n(x_1) - \Delta P_n(x_0) = P_n(x_2) - 2P_n(x_1) + P_n(x_0) = a_0 + a_1(x_2 - x_0) + a_2(x_2 - x_0)$,

$(x_2 - x_1) - 2a_0 - 2a_1(x_1 - x_0) + a_0 = a_1 \cdot 2h + a_2 \cdot 2h^2 - 2a_1 h = 2h^2 a_2 = \Delta^2 y_0$,

...

$\Delta^n P_n(x_0) = n! h^n a_n = \Delta^n y_0$,

откуда нетрудно получить формулы для вычисления коэффициентов a_i:

$$a_0 = y_0, a_1 = \frac{\Delta y_0}{h}, ...a_k = \frac{\Delta^k y_0}{k! h^k}, ...a_n = \frac{\Delta^n y_0}{n! h^k} \tag{2.21}$$

Подставив (2.21) в (2.19), получим

$$P_n(x) = y_0 + \frac{\Delta y_0 (x - x_0)}{h} + \frac{\Delta^2 y_0 (x - x_0)(x - x_1)}{2! h^2} + \ldots + \frac{\Delta^n y_0 (x - x_0)(x - x_1)\ldots(x - x_{n-1})}{n! h^n}. \quad (2.22)$$

Выражение (2.22) является *первой интерполяционной формулой Ньютона*. Погрешность интерполяции для формулы Ньютона можно вычислить следующим образом.

Предположим, что функция $f(x)$ $(n+1)$ раз дифференцируема. Введем переменную $q = \frac{(x - x_0)}{h}$, тогда ошибка метода может быть вычислена по формуле

$$R_n(x) = h^{n+1} \frac{q(q-1)\ldots(q-n)}{(n+1)!} \cdot f^{(n+1)}(\xi), \quad (2.23)$$

где $\xi \in [x_0, x_n]$. Если известна конечная разность $\Delta^{n+1} y_0$, то погрешность интерполяционной формулы можно оценить приближенно по формуле

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

Интерполяционная формула Ньютона представляет собой просто другой способ составления интерполяционного многочлена. Она полезна, поскольку число используемых узлов может быть легко увеличено или уменьшено без перевычисления остальных коэффициентов полинома в форме Ньютона. Интерполяционная формула Ньютона используется только для регулярных сеток.

2.10. Интерполяционные сплайн-функции

Использование интерполяционных многочленов при восполнении дискретно заданных функций с конечной и невысокой гладкостью имеет свои недостатки.

1. При большом количестве узлов интерполяции наблюдается осцилляция многочлена между узловыми точками.
2. Большое количество арифметических операций, свойственное многочленам высоких степеней, с одной стороны, увеличивает величину погрешностей результатов вычислений на ЭВМ (за счет накопления ошибок округления), с другой - приводит к значительным затратам машинного времени.

Можно избежать практически всех перечисленных выше недостатков, если в качестве интерполянта использовать сплайн-функции.

Определение. *Интерполяционным сплайном называют функцию, гладко склеенную из кусков функций некоторого класса и проходящую через узлы интерполяции.*

Если в качестве носителя сплайн-функции используются полиномы, то сплайн называется полиномиальным. На практике обычно применяют полиномиальные сплайн-функции. Пусть задана интерполяционная сетка

$a = x_0 < x_1 < x_2 < \ldots < x_n = b$.

Функцию $S_n(x)$ будем называть полиномиальным сплайном, если

а) $S_n(x) = P_n^i(x)$, где $x \in (x_i, x_{i+1}]$,

б) $S_n(x)$- принадлежит классу непрерывных функций на [a,b] вместе со своими производными, вплоть до *n-1* порядка;

с) $S_n(x_i) = y_i$.

2.11. Линейный сплайн

Пусть носителем сплайн-функции является полином 1-ой степени $P_1(x) = a + b \cdot x$, т.е. сплайн составлен из кусочков прямых линий, соединяющихся в узлах интерполяции (см. рис.2.3). Рассмотрим следующее формальное представление линейной сплайн-функции:

$$S_1(x) = y_0 + c_0 \cdot (x - x_0) + \sum_{k=1}^{n-1} c_k \cdot (x - x_k)_+ , \qquad (2.24)$$

где $(x - x_k)_+ = \begin{cases} 0, & x \le x_k \\ (x - x_k), & x > x_k \end{cases}$

Для определения неизвестных коэффициентов $c_0, c_1, \ldots, c_{n-1}$ линейного сплайна (2.24) составим систему линейных уравнений

$$\begin{cases} S_1(x_1) = y_0 + c_0(x_1 - x_0) = y_1 , \\ S_1(x_2) = y_0 + c_0(x_2 - x_0) + c_1(x_2 - x_1) = y_2 , \\ \ldots \\ S_1(x_n) = y_0 + c_0(x_n - x_0) + \ldots + c_{n-1}(x_n - x_{n-1}) = y_n . \end{cases} \qquad (2.25)$$

Система уравнений (2.25) имеет треугольную матрицу коэффициентов, что, с одной стороны, позволяет сделать вывод о единственности решения системы уравнений, с другой - организовать простой итерационный процесс вычисления

коэффициентов сплайн-функции.

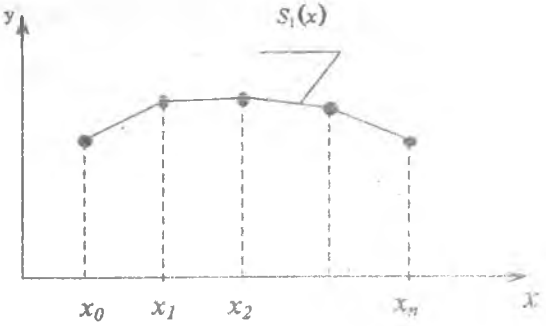


Рис.2.3. Линейный сплайн $S_1(x)$

Формула (2.24) удобна для аналитических исследований линейного сплайна, но достаточно трудоемка при реализации вычислительного алгоритма на ЭВМ. В тех случаях, когда требуется большее быстродействие программ при вычислении функции, целесообразно использовать альтернативную форму представления сплайн-функции, основанную на определении сплайна. Поскольку на каждом частичном участке интерполяционной сетки (между узлами интерполяции) линейный сплайн является полиномом 1-й степени $P_1(x) = a + b \cdot x$, то, используя формулу (2.24), несложно вычислить коэффициенты линейной функции и получить следующее представление сплайна:

$$S_1(x) = \begin{cases} a_0 + b_0 \cdot x, & x \in (-\infty, x_1] \\ a_1 + b_1 \cdot x, & x \in (x_1, x_2] \\ \dots \\ a_{n-1} + b_{n-1} \cdot x, & x \in (x_{n-1}, +\infty) \end{cases} \qquad (2.26)$$

Полученная формула (2.26) при ее реализации на ЭВМ требует большего объема памяти для хранения коэффициентов $a_0, b_0, a_1, b_1, \dots, a_{n-1}, b_{n-1}$, но в этом случае при вычислении сплайна производится всего лишь две арифметических операции: сложение и умножение.

2.12 Параболический сплайн

Рассмотрим в качестве носителя сплайн-функции параболу $P_2(x) = a + b \cdot x + c \cdot x^2$, тогда сплайн будет образован из кусков парабол, гладко склеенных между собой по первой производной. Остановимся на случае, когда узлы «склейки» парабол совпадают с узлами интерполяции функции $f(x)$:

$$S_2(x) = y_0 + u \cdot (x - x_0) + c_0 \cdot (x - x_0)^2 + \sum_{k=1}^{n-1} c_k \cdot (x - x_k)_+^2, \qquad (2.27)$$

где $(x - x_k)_+^2 = \begin{cases} 0, & x \le x_k \\ (x - x_k)^2, & x > x_k \end{cases}$.

Выражение (2.27) для параболического сплайна «сконструировано» таким образом, чтобы в узловых точках $x_1, x_2, \ldots, x_{n-1}$ куски параболы были гладко склеены по первой производной, т.е. выполнялись условия:

$$\frac{dS_2(x_i - 0)}{dx} = \frac{dS_2(x_i + 0)}{dx}, \quad i = 1,2,\ldots,n-1. \qquad (2.28)$$

Действительно:

$$\frac{dS_2(x_i - 0)}{dx} = \frac{d}{dx}\left(y_0 + u(x_i - x_0) + c_0(x_i - x_0)^2 + \ldots + c_{i-1}(x_i - x_{i-1})^2\right) = u + 2c_0(x_i - x_0) + \ldots$$

$$+ 2c_{i-1}(x_i - x_{i-1});$$

$$\frac{dS_2(x_i + 0)}{dx} = \frac{d}{dx}\left(y_0 + u(x_i - x_0) + c_0(x_i - x_0)^2 + \ldots + c_{i-1}(x_i - x_{i-1})^2 + c_i(x_i - x_i + 0)^2\right) =$$

$$u + 2c_0(x_i - x_0) + \ldots + 2c_{i-1}(x_i - x_{i-1}) + 2c_i(x_i - x_i + 0),$$

откуда следует справедливость формулы (2.28).

Формула (2.27) содержит ровно $n+1$ неизвестных коэффициентов $u, c_0, c_1, \ldots c_{n-1}$, для определения которых имеется n условий интерполяции. Для «замыкания» системы линейных уравнений в случае параболического сплайна требуется еще одно дополнительное условие. Обычно это значение производной функции $f(x)$ на границе интерполяционной сетки. Пусть нам известна $y_0' = f'(x_0)$, тогда для определения неизвестных коэффициентов сплайна можно составить следующую систему линейных уравнений:

$$\begin{cases} \dfrac{dS(x_0)}{dx} = u = y_0', \\ S(x_1) = y_0 + u\cdot(x_1 - x_0) + c_0\cdot(x_1 - x_0)^2 = y_1, \\ S(x_2) = y_0 + u\cdot(x_2 - x_0) + c_0\cdot(x_2 - x_0)^2 + c_1\cdot(x_2 - x_1)^2 = y_2, \\ \ldots \\ S(x_n) = y_0 + u\cdot(x_n - x_0) + c_0\cdot(x_n - x_0)^2 + c_{n-1}\cdot(x_n - x_{n-1})^2 = y_n. \end{cases} \quad (2.29)$$

Выбор граничных условий параболического сплайна

Начиная с параболического сплайна, все сплайн-функции имеют недоопределенную систему уравнений для вычисления их коэффициентов, т.е. количество условий интерполяции меньше, чем количество неизвестных коэффициентов. Для параболического сплайна недостает одного условия, для кубического сплайна двух и т.д. Выбор граничных условий существенно отражается на виде сплайн-функции, т.е. качестве решения исходной задачи интерполирования дискретной функции $f(x)$.

Например, для параболического сплайна (2.27) коэффициент u можно интерпретировать как тангенс угла наклона касательной к графику функции в узловой точке x_0. Как видно из рисунка 2.4, неправильные граничные условия (случай u_2) приводят к эффекту осцилляции сплайна $S_2(x,u_2)$ между узлами интерполяции, а следовательно и к большим погрешностям в решении задачи интерполирования исходной функции $f(x)$.

К сожалению, из постановки задачи редко бывают известны дополнительные граничные условия, следовательно, при построении сплайн-функции их необходимо доопределять. Наибольшее распространение получили следующие способы доопределения граничных условий.

1. На границе интерполяционной сетки задают нулевые значения для вторых производных функции $f(x)$, т.е. полагают, что

$$\frac{d^2 f(x)}{dx^2} = 0 \text{ при } x = a \quad (x = b).$$

Идея метода заключается в линеаризации сплайн-функции на границах интерполяционной сетки. Полагают, что за пределами

интервала интерполирования функции $f(x)$ сплайн ведет себя как линейная функция. Такой подход к решению проблемы выбора граничных условий используется в большинстве стандартных программ сплайн-интерполяции. Однако практика показывает, что этот метод не дает устойчивых результатов. Достаточно часто наблюдается эффект осцилляции функции между узлами интерполяции, а для кубического сплайна – на границах интерполяционной сетки.

2. Вторая группа методов основывается на идее численной оценки (по имеющейся информации $y_0, y_1, \ldots y_n$) недостающих граничных условий, например с помощью интерполяционных многочленов. Недостаток этой группы методов заключается в неизбежных погрешностях в оценке недостающих граничных условий, что,естественно,ухудшает решение исходной задачи.

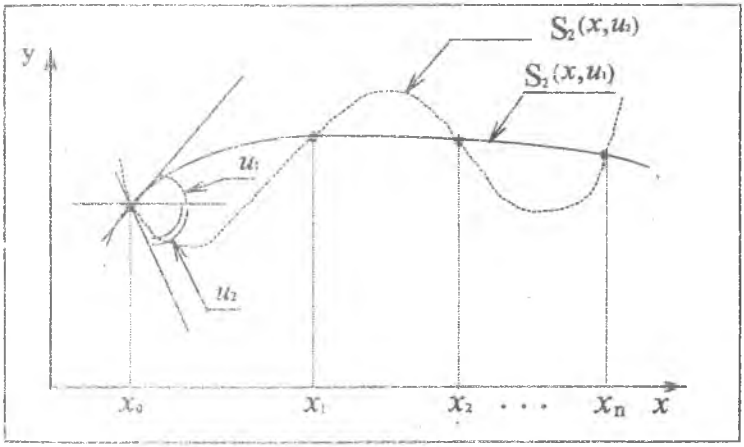


Рис.2.4. Влияние выбора граничных условий на решение задачи интерполирования функции *f(x)*

3. Наиболее перспективным способом решения проблемы граничных условий можно считать вариационный подход к выбору граничных условий на основе дополнительной информации об исходной функции $f(x)$.

Например, предположим, что функция $f(x)$ является унимодальной выпуклой функцией. Граничное условие $u\left(y_0' = u = \frac{dS(x_0)}{dx}\right)$ для параболического сплайна (2.27) будем

считать свободной (варьируемой) переменной, выбор которой будем осуществлять, оптимизируя некоторый критерий оценки качества решения задачи интерполирования функции $f(x)$. Например, угол наклона касательной (рис.2.4) будем подбирать таким образом, чтобы сплайн-функция $S_2(x,u)$ проходила между узлами интерполяции достаточно гладко.

В качестве критериев оценки качества решения задачи интерполяции функции $f(x)$ можно рассмотреть следующие:

$$\text{а)}\ \int_{x_0}^{x_n}\left(\frac{d^2S(x,u)}{dx^2}\right)^2 dx \xrightarrow[u]{} \min\,, \qquad (2.30)$$

$$\text{б)}\ \int_{x_0}^{x_n}\sqrt{1+\left(\frac{dS(x,u)}{dx}\right)^2}\,dx \xrightarrow[u]{} \min\,. \qquad (2.31)$$

Первый критерий (2.30) дает оценку среднеквадратичной кривизны формируемого сплайна, второй - определяет общую длину сплайна, проведенного через узлы интерполяции. И первый, и второй критерий помогают избежать эффекта осцилляции сплайн-функции.

Сходимость параболического сплайна

Остаточный член параболического сплайна для класса непрерывных на *[a,b]* функций, имеющих непрерывную первую производную, можно оценить по формуле

$$|R_2| = |S_2(x) - f(x)| < \frac{2\bar{h}w(f')}{(3\sqrt{3})}, \quad \text{где} \qquad \bar{h} = \max(x_i - x_{i-1}),$$

$i = 1,\dots n$,

$$w(f') = \max_{i=1,\dots n-1} w_i(f');$$

$$w_i(f') = \max_{x^*,x^{**}\in[x_i,x_{i+1}]} \left|f'(x^{**}) - f'(x^*)\right|.$$

2.13 Кубические сплайн-функции

Если носителем сплайн-функции является кубическая парабола $P_3(x)a + bx + cx^2 + dx^3$, то можно построить два кубических сплайна:

$$S_3(x) = y_0 + u(x - x_0) + C(x - x_0)^2 + D_0(x - x_0)^3 + \sum_{k=1}^{n-1} D_k(x - x_0)_+^3\,, \qquad (2.32)$$

$$S_3(x)y_0 + u(x-x_0) + C_0(x-x_0)^2 + D(x-x_0)^3 + \sum_{k=1}^{n-1} C_k(x-x_0)_+^2 . \quad (2.33)$$

В первом случае (2.32) условию гладкого сопряжения кубических парабол подвергаются первые и вторые производные $S_3(x)$, во втором случае (2.33) кубические параболы в узлах сплайна сопрягаются только по первой производной.

2.14 Интерполирование многомерных функций

На примере интерполирования функции двух переменных $z = f(x,y)$ рассмотрим общую схему решения задачи интерполирования многомерных функций.

Пусть задана прямоугольная интерполяционная сетка (рис.2.5):

Δx: $x_0 < x_1 < \ldots < x_n$,

Δy: $y_0 < y_1 < \ldots < y_m$,

в узлах которой известны значения функции $z_{ij} = f(x_i, y_j)$. Рассмотрим следующую схему интерполяции функции $f(x,y)$. На

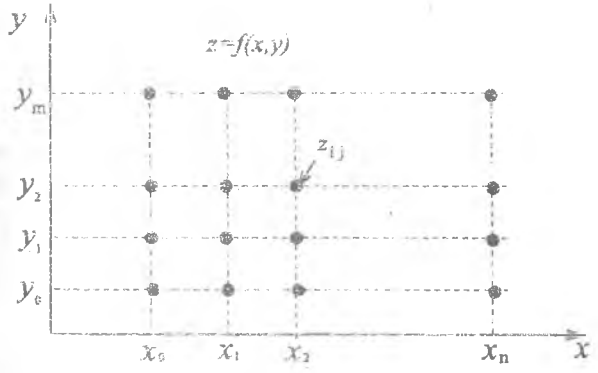


Рис. 2.5 Интерполяционная сетка для функций двух переменных

первом этапе, например, по переменной x при фиксированных значениях второй переменной $y = y_0, y = y_1, \ldots y = y_m$, решим $m+1$ задачу интерполирования функции от одной переменной

$f(x, y_j)$ $(j = 0,1,...m)$.

Пусть $y = y_j$, тогда можно поставить задачу интерполирования функции $f(x, y_j)$ в узловых точках $(x_0, y_j), (x_1, y_j), ...(x_n, y_j)$, где функция $f(x)$ принимает значения $z_{0j}, z_{1j}, ... z_{nj}$, полиномом степени n:

$$P_n(x, y_j) = a_0(y_j) + a_1(y_j) \cdot x + ... + a_n(y_j) \cdot x^n. \qquad (*)$$

Решив m задач типа (*), мы получим систему многочленов степени n для различных по y сечений интерполяционной сетки:

$$y = y_0, \quad P_n(x, y_0) = a_0(y_0) + a_1(y_0) \cdot x + ... + a_n(y_0) \cdot x^n,$$
$$...$$
$$y = y_m, \quad P_n(x, y_m) = a_0(y_m) + a_1(y_m) \cdot x + ... + a_n(y_m) \cdot x^n. \qquad (2.34)$$

Из (2.34) видно, что общее решение задачи интерполирования функции следует искать как полином степени n по переменной x с переменными коэффициентами:

$$P_n(x, y) = a_0(y) + a_1(y) \cdot x + ... + a_n(y) \cdot x^n. \qquad (2.35)$$

Для определения функциональных коэффициентов $a_i(y)$ $(i = 0,1,...n)$ рассмотрим $n+1$ интерполяционных задач. Например, для определения $a_i(y)$ можно поставить задачу о построении многочлена степени m на интерполяционной сетке $y_0 < y_1 < ... < y_m$ со значениями $a_i(y_0), a_i(y_1), ... a_i(y_m)$ в ее узловых точках соответственно.

Решение поставленной задачи найдем в виде многочлена степени m по переменной y: $a_i(y) = b_{0i} + b_{1i} y + ... + b_{mi} y^m$. Решив $n+1$ таких задач, мы получим выражения для вычисления коэффициентов $a_i(y)$:

$$a_0(y) = b_{00} + b_{10} y + ... + b_{m0} y^m,$$
$$...$$
$$a_m(y) = b_{0n} + b_{1n} y + ... + b_{mn} y^m. \qquad (2.36)$$

Подставив (2.36) в (2.35), в итоге получаем двумерный интерполяционный многочлен $P_{nm}(x, y)$ решения задачи интерполирования функции 2 переменных.

Предложенную схему можно распространить и на решение задачи интерполирования функций многих переменных.

2.15. Многомерный интерполяционный сплайн

Если вместо полиномов в предложенной выше схеме использовать сплайн-функции, то аналогично можно построить двумерный интерполяционный сплайн. Однако существуют и другие схемы построения многомерных интерполяционных сплайнов. Рассмотрим одну из них.

Определение. *Системой фундаментальных сплайнов* $S_i(x)$ *на интерполяционной сетке* Δx: $x_0 < x_1 < \ldots < x_n$ *называются сплайн-функции, удовлетворяющие условиям* :

$$S_i(x_k) = \delta_{ik} = \begin{cases} 0 & i \neq k \\ 1 & i = k,\ i,k = 0,1,\ldots n \end{cases} \tag{2.37}$$

В таблице 2.2. приведены условия интерполяции для фундаментальных сплайнов $S_i(x)$. Как видно из таблицы, каждый интерполяционный сплайн в узлах интерполяции n раз пересекает координатную ось (нули сплайна) и лишь в одном узле, совпадающем по номеру с номером сплайна, равен 1. На рис.2.6 показана система интерполяционных сплайнов для сетки $x_0 < x_1 < x_2 < x_3$

Таблица 2.2

Интерполяционные условия для фундаментальных сплайнов

$S_i(x)$	x_0	x_1	x_2	$\ldots$	x_n
$S_0(x)$	1	0	0	$\ldots$	0
$S_1(x)$	0	1	0	$\ldots$	0
$S_2(x)$	0	0	1	$\ldots$	0
$S_n(x)$	0	0	0	$\ldots$	1

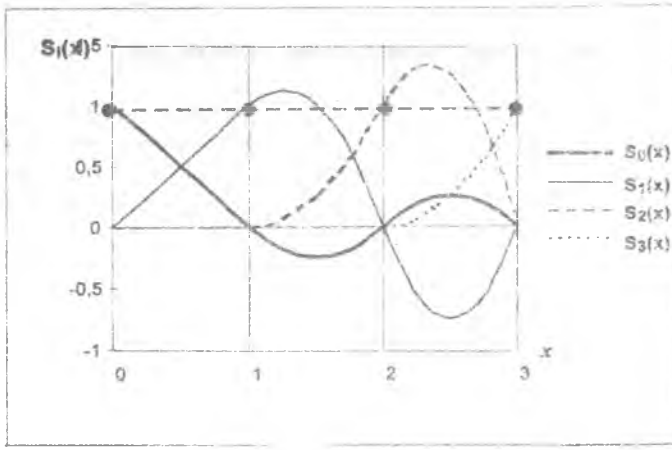


Рис. 2.6. Система фундаментальных интерполяционных сплайнов

Аналогичным образом можно построить систему фундаментальных сплайнов для независимой переменной y $S_j(y)$:

$$S_j(y) = \delta_{jk}; \quad j,k = 0,1,\ldots m, \tag{2.38}$$

тогда двумерный интерполяционный сплайн $S(x,y)$ можно представить следующим образом:

$$S(x,y) = \sum_{i=1}^{n}\sum_{j=1}^{m} f_{ij} \cdot S_i(x) \cdot S_j(y), \tag{2.39}$$

где f_{ij} - значение функции в узловых точках интерполяционной сетки. $\Delta x \times \Delta y$.

Учитывая свойства одномерных фундаментальных сплайнов (2.37) -(2.38), нетрудно убедиться, что двумерный сплайн (2.39) решает исходную задачу интерполирования функции $f(x,y)$, т.е. $S(x_i,y_j) = f_{ij}$. В формуле (2.39) автоматически заложены условия гладкого сопряжения кусков поверхностей, из которых состоит сплайн-функция, по линиям «склейки» этих поверхностей (в нашем случае по границам частичных прямоугольников интерполяционной сетки).

При построении фундаментальных сплайнов (2.37)-(2.38) требуются дополнительные граничные условия. Последние можно получить, используя дополнительные сведения о поведении функции $f(x,y)$ на границе интерполяционной сетки, либо в результате оптимизации некоторого критерия оценки качества

решения интерполяционной задачи.

Нетрудно показать, что при построении сплайна $S(x,y)$ с освобожденными граничными условиями, когда его вторые частные производные равны 0 на границе интерполяционной сетки, в общем случае требуется, чтобы вторые производные фундаментальных сплайнов на концах отрезков интерполяции также были равны 0.

2.16. Аппроксимация функции среднеквадратичная

Предположим, что функция $f(x)$ известна своими значениями на некотором конечном множестве точек $x_0, x_1, \ldots x_N$. Причем в узловых точках x_i функция $f(x)$ задана с некоторой погрешностью, т.е. $y_i = f(x_i) + \varepsilon_i$, где ε_i-некоторая ошибка измерения. Требуется по исходной информации, насколько это возможно, восстановить функцию $f(x)$. Такая задача часто возникает при обработке результатов испытаний технических систем, при вводе в ЭВМ графической информации, при описании геометрических образов объектов и т.д.

Определение. *Среднеквадратичная аппроксимация функции - это нахождение для заданной функции другой функции из некоторого класса функций, для которой среднеквадратичное отклонение от заданной функции минимально.*

Функцию, с помощью которой решают задачу аппроксимации, иногда называют аппроксимантом.

Пусть $\varphi(x, \alpha_0, \alpha_1, \ldots \alpha_n)$ - аппроксимант, $\alpha_0, \alpha_1, \ldots \alpha_n$ - его варьируемые параметры, тогда можно дать формальное описание задачи аппроксимации функции $f(x)$:

$$\sum_{k=0}^{N} \left(y_k - \varphi(x_k, \alpha_0, \alpha_1, \ldots \alpha_n) \right)^2 \xrightarrow[\alpha_0, \alpha_1 \ldots \alpha_n]{} \min \quad . \qquad (2.40)$$

Задачу аппроксимации можно интерпретировать как проведение аппроксимирующей функции $\varphi(x)$ между узлами аппроксимации исходной функции так, чтобы сумма квадратов отклонений функции $\varphi(x)$ от точек (x_k, y_k) была наименьшей (рис.2.7).

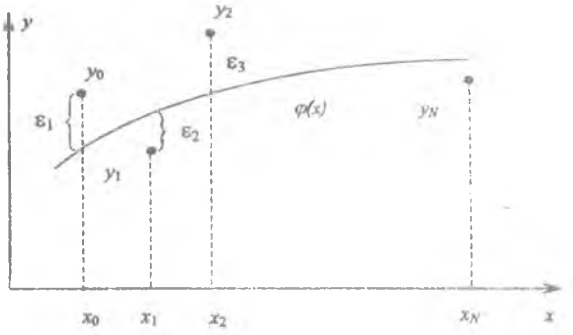


Рис.2.7. Задача аппроксимации функции $f(x)$

2.17. Полиномиальная аппроксимация

Пусть в качестве аппроксиманта используется полином степени n:

$$\varphi(x) = P_n(x) = \alpha_0 + \alpha_1 \cdot x + \ldots + \alpha_n \cdot x^n .$$

Для табличной функции, заданной конечным множеством точек $(x_0, y_0), (x_1, y_1), \ldots (x_N, y_N)$, запишем условие аппроксимации:

$$L(\alpha_0, \alpha_1, \ldots \alpha_n) = \sum_{k=0}^{N} \left(y_k - \left(\alpha_0 + \alpha_1 \cdot x_k + \ldots \alpha_n \cdot x_k^n \right) \right)^2 \longrightarrow \min , \quad (2.41)$$

Для нахождения неизвестных коэффициентов $\alpha_0, \alpha_1, \ldots \alpha_n$ воспользуемся необходимым условием экстремума функции $L(\alpha_0, \alpha_1, \ldots \alpha_n)$:

$$\begin{cases} \dfrac{dL}{d\alpha_0} = -2 \sum\limits_{k=0}^{N} \left(y_k - \left(\alpha_0 + \alpha_1 \cdot x_k + \ldots + \alpha_n \cdot x_k^n \right) \right) = 0 , \\ \dfrac{dL}{d\alpha_1} = -2 \sum\limits_{k=0}^{N} \left(y_k - \left(\alpha_0 + \alpha_1 \cdot x_k + \ldots + \alpha_n \cdot x_k^n \right) \right) \cdot x_k = 0 , \\ \ldots \\ \dfrac{dL}{d\alpha_n} = -2 \sum\limits_{k=0}^{N} \left(y_k - \left(\alpha_0 + \alpha_1 \cdot x_k + \ldots + \alpha_n \cdot x_k^n \right) \right) \cdot x_k^n = 0 . \end{cases} \quad (2.42)$$

Систему (2.42) несложно преобразовать к следующей системе

линейных уравнений относительно неизвестных параметров $\alpha_0, \alpha_1, \ldots \alpha_n$:

$$\begin{cases} N\cdot\alpha_0+\left(\sum\limits_{k=0}^{N}x_k\right)\cdot\alpha_1+\left(\sum\limits_{k=0}^{N}x_k^2\right)\cdot\alpha_2+\ldots+\left(\sum\limits_{k=0}^{N}x_k^n\right)\cdot\alpha_n=\sum\limits_{k=0}^{N}y_k, \\ \left(\sum\limits_{k=0}^{N}x_k\right)\cdot\alpha_0+\left(\sum\limits_{k=0}^{N}x_k^2\right)\cdot\alpha_1+\left(\sum\limits_{k=0}^{N}x_k^3\right)\cdot\alpha_2+\ldots+\left(\sum\limits_{k=0}^{N}x_k^{n+1}\right)\cdot\alpha_n=\sum\limits_{k=0}^{N}x_k\cdot y_k, \\ \ldots \\ \left(\sum\limits_{k=0}^{N}x_k^n\right)\cdot\alpha_0+\left(\sum\limits_{k=0}^{N}x_k^{n+1}\right)\cdot\alpha_1+\left(\sum\limits_{k=0}^{N}x_k^{n+2}\right)\cdot\alpha_2+\ldots+\left(\sum\limits_{k=0}^{N}x_k^{2n}\right)\cdot\alpha_n=\sum\limits_{k=0}^{N}x_k^n\cdot y_k. \end{cases} \quad (2.43)$$

Если $N>n$, то система (2.43) имеет единственное решение, поскольку по построению функция (2.41) является выпуклой унимодальной функцией (направленной выпуклостью вниз), следовательно, условия (2.42) являются необходимым и достаточным признаком минимума функции $L(\alpha_0,\alpha_1,\ldots\alpha_n)$, у которой имеется единственный экстремум.

2.18 Сплайн аппроксимация

Пусть в качестве аппроксиманта в (2.40) используется интерполяционный сплайн $S(x)$, заданный на интерполяционной сетке $\bar{x}_0\langle\bar{x}_1\langle\ldots\langle\bar{x}_n$. Построим систему фундаментальных сплайнов $S_i(x)$:

$$S_i(\bar{x}_k)=\delta_{ik},$$

$i,k=0,1,\ldots n$, тогда сплайн $S(x)$ можно представить в разложении по фундаментальным сплайнам :

$$S(x)=\sum_{i=0}^{n}\alpha_i\cdot S_i(x), \quad (2.44)$$

где α_i- неизвестное значение функции $f(x)$ в узле интерполяции x_i.

Для аппроксимации исходной функции $f(x)$, заданной своими значениями на конечном множестве точек $(x_0,y_0),(x_1,y_1),\ldots(x_N,y_N)$, построим целевую функцию:

$$L(\alpha_0,\alpha_1,\ldots\alpha_n)=\sum_{k=0}^{N}\left(y_k-\sum_{i=0}^{n}\alpha_i\cdot S_0(x_k)\right)^2 \xrightarrow[\alpha_0,\alpha_1,\ldots\alpha_n]{} \min. \quad (2.45)$$

Используя необходимый признак экстремума функции $L(\alpha_0,\alpha_1,\ldots\alpha_n)$ для определения $\alpha_0,\alpha_1,\ldots\alpha_n$, имеем следующую систему уравнений :

$$\begin{cases} \dfrac{dL}{d\alpha_0} = -2\sum\limits_{k=0}^{N}\left(y_k - \sum\limits_{i=0}^{n}\alpha_i \cdot S_i(x_k)\right)S_0(x_k) = 0, \\ \ldots \\ \dfrac{dL}{d\alpha_n} = -2\sum\limits_{k=0}^{N}\left(y_k - \sum\limits_{i=0}^{n}\alpha_i \cdot S_i(x_k)\right)S_n(x_k) = 0, \end{cases}$$

которую можно преобразовать к системе линейных уравнений относительно неизвестных $\alpha_0,\alpha_1,\ldots\alpha_n$:

$$\begin{cases} \left(\sum\limits_{k=0}^{N}S_0^2(x_k)\right)\cdot\alpha_0 + \left(\sum\limits_{k=0}^{N}S_0(x_k)S_1(x_k)\right)\cdot\alpha_1 + \ldots + \left(\sum\limits_{k=0}^{N}S_0(x_k)S_n(x_k)\right)\cdot\alpha_n = \sum\limits_{k=0}^{N}y_kS_0(x_k), \\ \ldots \\ \left(\sum\limits_{k=0}^{N}S_0^2(x_k)S_n(x_k)\right)\cdot\alpha_0 + \left(\sum\limits_{k=0}^{N}S_n(x_k)S_1(x_k)\right)\cdot\alpha_1 + \ldots + \left(\sum\limits_{k=0}^{N}S_n^2(x_k)\right)\cdot\alpha_n = \sum\limits_{k=0}^{N}y_kS_n(x_k). \end{cases}$$

Задача аппроксимации функции $f(x)$ сплайнами решается несколько сложнее, чем с помощью полиномов, поскольку необходимо решить три достаточно сложные проблемы:

1) Определиться с количеством узлов интерполяции сплайна $S(x)$. Чем больше узлов интерполяции, тем больше степеней свободы предоставляется для сплайн-функции, тем формально точнее может быть решена задача аппроксимации функции $f(x)$. Однако в этом случае ухудшаются сглаживающие свойства сплайна, поскольку он начинает интерполировать функцию $f(x_i)$ вместе с ее погрешностями измерения ε_i. При выборе количества узлов интерполяции следует искать разумный компромисс между точностью приближения и гладкостью формируемой зависимости.

2) Построить внутри аппроксимационной сетки $x_0,x_1,\ldots,x_N$ интерполяционную сетку $\bar{x}_1,\bar{x}_2,\ldots,\bar{x}_n$ сплайна. Расположение узлов интерполяции достаточно сильно сказывается на решении задачи аппроксимации. Обычно используют интерполяционные сетки с равномерным расположением узлов интерполяции.

3) Выбрать дополнительные граничные условия для построения системы фундаментальных сплайнов $S_i(x)$.

ГЛАВА 3

Численное решение нелинейных уравнений

3.1. Введение

Нахождение корней уравнений - это одна из древнейших математических проблем, не потерявшая своей актуальности и в наши дни. Она часто встречается в самых разнообразных областях науки и техники. Чаще всего в приложениях используются трансцендентные уравнения. Нередко решается задача о нахождении всех корней алгебраического многочлена.

Методы решения линейных и квадратных уравнений были известны еще древним грекам. Решение уравнений третьей и четвертой степеней было получено усилиями итальянских математиков в XV веке. Затем наступила пора поиска формул для корней уравнений 5-й и более высоких степеней. Безуспешные попытки продолжались около 300 лет и завершились в 20-х годах XIX в. благодаря работам норвежского математика Н. Абеля. Он доказал, что общее уравнение 5-й и более высоких степеней неразрешимо в радикалах.

Задача нахождения решения уравнения ставится следующим образом: для некоторой функции $F(x)$ необходимо найти такие значения аргумента x, для которых

$$F(x) = 0. \qquad (3.1)$$

Функция $F(x)$ может быть алгебраической или трансцендентной. Мы будем полагать, что она дифференцируема. В общем случае будем считать, что уравнение (3.1) не имеет аналитических формул для вычисления корней, поэтому приходится использовать приближенные методы его решения.

Приближенные методы решения уравнения (3.1), как правило, содержат два этапа.

1. Отыскание приближенного значения корня.

2. Уточнение приближенного значения до некоторой заданной степени точности.

Основное внимание этой главы будет уделено различным численным методам, относящимся ко второму этапу.

Практически все приближенные методы нахождения корней уравнений относятся к классу итерационных методов.

Определение. *Методом итерации назовем численный метод, который последовательно, шаг за шагом, уточняет первоначальное, грубое значение корня.*

Каждый шаг в таком методе называется **итерацией**. Важным свойством итерационных методов, которое присуще им или нет, является **сходимость итерационных методов**.

Пусть некоторый итерационный метод генерирует последовательность новых приближенных значений решений уравнения (3.1), которое начинается из точки x_0: $x_0, x_1, \ldots, x_n, \ldots$. Если по мере увеличения числа шагов разница между приближенным значением корня x_n и точным x^* $\left(\left|x_n - x^*\right|\right)$ уменьшается, то говорят, что метод итераций сходится.

3.2 Метод последовательных приближений

Уравнение (3.1) приведем к виду

$$x = f(x). \tag{3.2}$$

Это преобразование можно сделать следующим образом. Прибавив к левой и правой частям (3.1) x и поменяв их местами, получим

$$x = x + F(x),$$

тогда, обозначив через $f(x) = x + F(x)$, имеем (3.2).

Итерационный процесс в методе последовательных приближений строится по следующей простой формуле:

$$x_n = f(x_{n-1}), \tag{3.3}$$

при этом предполагается, что начальное приближение корня уравнения (3.1) нам известно (x_0). Начиная с x_0 последовательно подставлять найденные новые

приближенные значения корня в (3.3), мы реализуем итерационный метод последовательных приближений. Основным вопросом этого метода является вопрос о сходности x_n к решению уравнения (3.2).

Достаточное условие сходимости метода простой итерации

Пусть x^* корень уравнения (3.2), т.е.

$$x^* = f(x^*). \tag{3.4}$$

Из равенства (3.3) вычтем равенство (3.4), получим

$$x_n - x^* = f(x_{n-1}) - f(x^*). \tag{3.5}$$

Умножая (3.5) на $\frac{(x_{n-1} - x^*)}{(x_{n-1} - x^*)}$, имеем

$$x_n - x^* = \frac{f(x_{n-1}) - f(x^*)}{(x_{n-1} - x^*)} \cdot (x_{n-1} - x^*),$$

тогда по теореме о среднем для непрерывной и дифференцируемой функции $f(x)$, имеющей непрерывную производную, справедливо:

$$x_n - x^* = f'(\xi) \cdot (x_{n-1} - x^*),$$

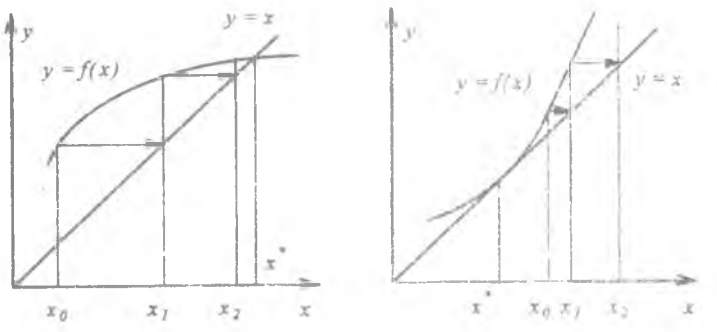
где точка ξ находится между точками x_{n-1} и x^*.

Если $f'(\xi) < 1$ во всем рассматриваемом интервале, т.е. в интервале, включающим точки $x_0, x_1, \ldots, x_n, x^*$, то $|x_n - x^*| < |x_{n-1} - x^*|$. Последнее означает, что новое значение x_n ближе к x^*, чем приближенное значение корня x_{n-1}, полученное на предыдущем шаге, т.е. метод простой итерации сходится.

Таким образом, если $|f'(x)| < 1$, то процесс сходится, если же $|f'(x)| > 1$, то процесс расходится.

На рис. 3.1 представлены геометрические интерпретации сходящегося и расходящегося

интерполяционных процессов для метода простой итерации.



а) $|f(x)| < 1$ б) $|f(x)| > 1$

Рис.3.1 Геометрическое представление сходящегося (а) и расходящегося (б) процессов метода простой итерации

3.3. Метод Ньютона-Рафсона

Метод простой итерации медленно сходится к решению x^*. Скорость сходимости можно существенно увеличить, если «движение» к новому приближению корня (рис.3.1а) делать не параллельно оси 0x до пересечения с прямой $y = x$, а по касательной к графику кривой $y = f(x)$.

Пусть итерационный процесс достиг точки x_{n-1}. Составим уравнение касательной к графику функции $y = f(x)$ в точке x_{n-1}:

$$y - f(x_{n-1}) = f'(x_{n-1}) \cdot (x - x_{n-1}).$$

Найдем точку пересечения касательной с прямой $y = x$:

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

Построим итерационный процесс по формуле

$$x_n = \frac{f(x_{n-1}) - x_{n-1}\, f'(x_{n-1})}{1 - f'(x_{n-1})} \qquad (3.5)$$

Это и есть знаменитый метод Ньютона-Рафсона (рис. 3.2). Для

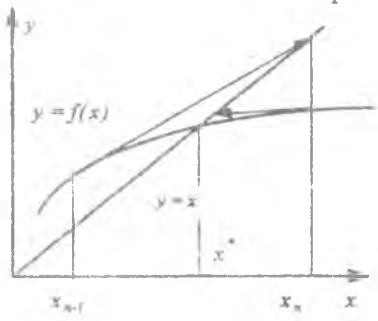


Рис.3.2.Метод касательных

уравнения (3.1) итерационную формулу после несложных преобразований можно представить в виде

$$x_{n+1} = x_n - \frac{F(x_n)}{F'(x_n)}. \qquad (3.6)$$

Сходимость метода Ньютона-Рафсона

Введем в обращение функцию $g(x) = \dfrac{f(x) - x\, f'(x)}{1 - f'(x)}$,

тогда с помощью функции $g(x)$ итерационная формула (3.5) преобразуется к виду

$$x_{n+1} = g(x_n). \qquad (3.7)$$

Итерационный метод (3.7) сходится, если

$$|g'(x)| < 1$$

или

$$|g'(x)| = \left| \frac{f''(x)[f(x) - x]}{[1 - f'(x)]^2} \right| < 1. \qquad (3.8)$$

Несложный анализ выражения (3.8) показывает, что метод Ньютона – Рафсона сходится, если:

1) x_0 выбрано достаточно близко к x^*;

2) производная $f''(x)$ не становится слишком большой;

3) производная $f'(x)$ не слишком близка к единице.

3.4. Ошибки округления в итерационных методах

Вычислительные процессы на ЭВМ практически всегда связаны с возникновением ошибок округления. Во многих случаях эти ошибки накапливаются. Так, например, при вычислении степенных рядов на ЭВМ накапливаются ошибки арифметических операций. Итерационные методы являются приятным исключением, когда достаточно длинные итерационные процессы не только не накапливают ошибки округления, но и уменьшают их.

В сходящихся итерационных процессах общая ошибка округления равна ошибке, возникшей в последней итерации, и не зависит от арифметических операций, выполненных на предыдущих итерациях. Причина этого явления ясна — каждое новое приближение, включая и предпоследнее, можно рассматривать как исходное приближение. Ошибка округления при вычислении последнего приближения зависит, таким образом, только от арифметических операций, с помощью которых это последнее приближение получается из предпоследнего.

3.5. Вычисление корней многочленов

Рассмотрим важный практический случай, когда $F(x)$ представляет собой многочлен степени m:

$$F(x) = a_0 + a_1 \cdot x + \ldots + a_m \cdot x^m. \qquad (3.9)$$

Применим метод Ньютона—Рафсона согласно формуле $x_{n+1} = x_n - \dfrac{F(x_n)}{F'(x_n)}$. Из многочлена $F(x)$ выделим линейный множитель $(x - x_n)$:

$$F(x) = (x - x_n)\cdot\left(b_1 + b_2 \cdot x + \ldots + b_m \cdot x^{m-1}\right) + b_0 \ . \qquad (3.10)$$

Сравнивая коэффициенты при одинаковых степенях в выражениях (3.9) и (3.10), нетрудно получить рекуррентные формулы для вычисления $b_0, b_1, \ldots b_m$:

$$\begin{cases} b_m = a_m, \\ b_k = a_k + x_n \cdot b_{k+1}, \quad k = m-1, m-2, \ldots 0. \end{cases} \tag{3.11}$$

Из (3.10) находим, что: $F(x_n) = b_0$

Положим

$G(x) = b_1 + b_2 \cdot x + \ldots + b_m \cdot x^{m-1}$, тогда

$F(x) = (x - x_n) \cdot G(x) + b_0$, откуда

$F'(x) = (x - x_n) \cdot G'(x) + G(x)$, следовательно

$F'(x_n) = G(x_n)$.

В многочлене $G(x)$ степени $m-1$ выделим линейный множитель $(x - x_n)$:

$$G(x) = (x - x_n) \cdot \left(c_2 + c_3 \cdot x + \ldots + c_m \cdot x^{m-2}\right) + c_1. \tag{3.12}$$

Сравнивая коэффициенты в многочлене $G(x)$ и выражении (3.12), имеем следующие рекуррентные формулы для вычисления коэффициентов $c_1, c_2, \ldots c_m$:

$$\begin{cases} c_m = b_m, \\ c_k = b_k + x_n \cdot c_{k+1}, \quad k = m-1, \ldots 1. \end{cases} \tag{3.13}$$

и соответственно $F'(x_n) = G(x_n) = c_1$

Подставляя найденные значения $F(x_n)$ и $F'(x_n)$ в формулу метода Ньютона – Рафсона, получаем

$$x_{n+1} = x_n - \frac{b_0}{c_1}, \tag{3.14}$$

где b_0 и c_1 вычисляются по формулам (3.11) и (3.13).

Следует отметить, что в формуле (3.14) коэффициенты b_0 и c_1 необходимо перевычислять на каждом шаге итерационного процесса.

3.6. Выбор начального приближения

Для «запуска» итерационного метода необходимо задать начальное приближение значения корня x_0. Иногда такое начальное приближение известно из физических соображений. Довольно часто его можно найти с помощью грубого анализа функции $F(x)$. В некоторых случаях помогает графический метод решения уравнения. В общем случае не существует эффективного алгоритма отыскания

начального приближения x_0. Приведем некоторые соображения, полезные при нахождении исходного приближения.

Любая функция $F(x)$ имеет свою область определения. В качестве примера на рис. 3.3. приведена область

Рис. 3.3. Выбор начального приближения решения уравнения F(x) = 0

определения некоторой функции $F(x)$ – отрезок [A,B]. Считается, что за пределами отрезка [A,B] функция не определена. Внутрь отрезка [A,B] вложен еще один отрезок [C,D], который описывает область адекватности математической модели $F(x)=0$.

Область адекватности – *это множество таких точек, для которых функция $F(x)$ с достаточной достоверностью (т.е. с заданной точностью) описывает реальные физические процессы или явления.*

Область адекватности математической модели для сложных физических объектов значительно уже области определения функции $F(x)$, что обычно связано с большим количеством упрощающих предположений, принятых на начальном этапе построения математической модели

объекта. Внутри области адекватности (отрезка [C,D], рис.3.3.) можно обнаружить область сходимости итерационного метода (отрезок [E,F], рис. 3.3.). Размеры области сходимости зависят от используемого итерационного метода. Очевидно, что идеальным случаем является выбор x_0 из области сходимости итерационного метода, т.е. когда $x_0 \in [E,F]$. Попадание начального приближения x_0 за пределы области сходимости, но в область адекватности может вызвать расходимость итерационного метода. Попадание x_0 за пределы области адекватности математической модели приводит, либо к нахождению несуществующих корней функции $F(x)$ (см. [A,C], рис 3.3), либо к аварийному завершению программы (переполнение разрядной сетки, корень из отрицательного числа и т.д.).

При разработке метода нахождения начального приближения следует учитывать еще одно обстоятельство. Границы перечисленных выше областей обычно неизвестны, а ограниченность разрядной сетки ЭВМ вызывает их серьезное сужение. Задача определения границ областей адекватности модели и сходимости итерационного метода по своей трудоемкости значительно сложнее задачи отыскания корня уравнения $F(x)=0$. На практике целесообразно использовать достаточно простые методы, учитывающие перечисленные факторы.

Предположим, что из физических соображений известен отрезок $[a,b]$, содержащий решение x^*. Разыграем серию случайных величин $\xi_1, \xi_2, \ldots \xi_m$ при $\xi_k \in [a,b]$, равномерно распределенных на отрезке $[a,b]$. В качестве начального приближения можно взять такую точку ξ_k, для которой $|F(\xi_k)|$ минимально, т.е. $x_0 = \xi_k$. Итерационный процесс запускается из точки x_0, если при этом метод расходится, то необходимо разыграть еще одну серию случайных величин и т.д., пока не будет найдено решение x^*. При реализации предложенного метода выбора начального приближения x_0 необходимо изыскать возможность перехвата программных прерываний в тех случаях, когда ξ_k выходит за пределы области определения $F(x)$.

В тех случаях, когда область адекватности модели $F(x)$ достаточно велика, решить проблему нахождения начального приближения x_0 позволяет **метод продолжения по параметру**.

Метод продолжения по параметру

Из условия (3.8) сходимости метода Ньютона–Рафсона видно, что метод сходится, если начальное приближение x_0 достаточно близко к решению уравнения x^*. Идея метода продолжения по параметру заключается в искусственном приведении условий сходимости итерационного метода к случаю, когда он сходится.

Пусть x_0 – начальное приближение, из которого итерационный метод для задачи

$$F(x)=0 \qquad (3.15)$$

расходится.

Построим новую функцию, зависящую от параметра t $H(x,t)$, и рассмотрим задачу о нахождении корней уравнения:

$$H(x,t)=F(x)+(t-1)F(x_0)=0 \qquad (3.16)$$

при заданном t.

Из (3.16) видно, что при $t=0$ уравнение

$$H(x,0)=F(x)-F(x_0)=0$$

имеет тривиальное решение $x^*=x_0$ и в этом случае итерационный метод всегда сходится. С другой стороны, при $t=1$ решение уравнения

$$H(x,1)=F(x)=0,$$

совпадает с решением исходного уравнения (3.15).

Выбирая промежуточные значения параметра $t\in[0,1]$, можно добиться, чтобы значение корня уравнения

$$H(x,t_1)=0$$

было бы достаточно близко к начальному приближению x_0. В результате находим решение x_1^*.

На рис. 3.4 решение x_1^* соответствует значению параметра $t_1=0{,}4$. Новое значение x_1^* используется в качестве начального приближения для решения другой

задачи при $t_2 = 0{,}7$, в результате получаем решение x_2^*, которое уже достаточно близко от решения исходной задачи. На последнем этапе решается исходная задача $F(x)=0$ при $t=1$ для последнего найденного начального приближения x_2^*.

Таким образом, вместо решения исходного уравнения $F(x)=0$ из точки x_0 в методе продолжения по параметру предлагается построить последовательность задач для функций $H(x,t_1), H(x,t_2),\ldots$, в каждой из которых начальное приближение расположено достаточно близко от своего решения. Тогда предпоследнее из найденных решений будет находиться в окрестностях решения исходной задачи, а алгоритм за конечное число шагов вычислит корень уравнения (3.15).

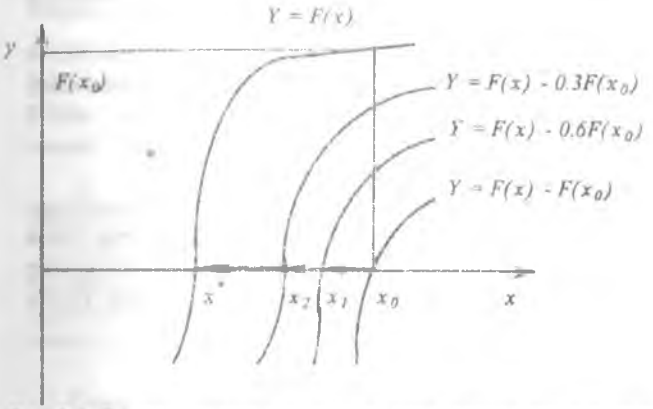


Рис. 3.4. Геометрическая интерпретация метода продолжения по

ГЛАВА 4

Численные методы решения систем уравнений

4.1. Введение

Проблемы решения систем уравнений возникают на практике достаточно часто. Например, с помощью системы нелинейных уравнений описываются математические модели большинства объектов исследования. С другой стороны, задача решения системы линейных уравнений возникает при построении интерполяционных многочленов, в процессе аппроксимации функции, при решении дифференциальных уравнений в частных производных и т.д. Проблема решения систем уравнений привлекает внимание ученых многих поколений. В результате в настоящее время имеется множество методов её решения.

В этой главе основное внимание сосредоточим на наиболее типичных представителях методов решения систем уравнений, наглядно раскрывающих главную идею метода, полагая, что с различными модификациями этих методов читатель может познакомиться в специальной литературе.

Практически все известные методы решения систем нелинейных уравнений относятся к классу итерационных методов. В связи с чем рассмотрим общие вопросы организации итерационного численного метода решения системы нелинейных уравнений. Пусть нам задана система нелинейных уравнений:

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

Требуется найти такие значения $\left(x_1^*, x_2^*, \dots x_n^*\right)$, которые преобразуют уравнения (4.1) в систему тождеств. Систему (4.1) удобно записать в компактной матричной форме:

$$F(X) = 0, \tag{4.2}$$

где $X = (x_1, x_2, \dots x_n)^T$ - вектор неизвестных переменных, $F(X) = (\ f_1(X), f_2(X), \dots f_n(X)\ )^T$ - векторная функция. Через

X^* обозначим решение системы уравнений (4.2).

Численные методы решения системы (4.2) сводятся к нахождению последовательности векторов $X^0, X^1, \ldots X^k$, которая сходится к точному решению X^*. Вектор X^0 называется начальным приближением.

Большинство итерационных методов решения системы нелинейных уравнений можно представить следующей обобщенной итерационной формулой:

$$X^{k+1} = G\left(X^k, H^k\right), \tag{4.3}$$

где H^k - вектор-параметр итерационного процесса, зависящий от результатов выполнения предыдущих операций, $G(X,H)$-итерационная вектор-функция, вид которой зависит от способа построения итерационного процесса. Если H^k не зависит от k, то метод будет стационарным с обобщенной итерационной формулой вида

$$X^{k+1} = G\left(X^k\right). \tag{4.4}$$

К основным характеристикам итерационных методов относятся:

1.*Сходимость метода*, определяющая его алгоритмическую надежность.

2.*Скорость сходимости*, определяющая точность и экономичность метода.

Сходимость итерационных методов

Итерации называются сходящимися если выполняются следующие условия:

$$\lim_{k\to\infty} X^k = X^*.$$

Пусть вектор-функция $G(X)$ определена и непрерывна вместе со своей матричной производной $G'_X(X) = \dfrac{dG}{dX}$, тогда, если X^k не выходит из области определения вектор-функций $F(X)$ и $G(X)$ и если *спектральный радиус* ρ *матрицы* $G'_X(X)$ *меньше 1*, т.е.

$$\rho\left(G'_X(X)\right)\langle 1, \tag{4.5}$$

то процесс итераций (4.4) сходится к решению X^*.

Условие (4.5) определяет *достаточное условие сходимости итерационного процесса*. Здесь под матричной производной $G'_X(X)$ подразумевается следующее выражение:

$$G'_X(X) = \begin{vmatrix} \frac{\partial g_1(X)}{\partial x_1} & \frac{\partial g_1(X)}{\partial x_2} & \dots & \frac{\partial g_1(X)}{\partial x_n} \\ \frac{\partial g_2(X)}{\partial x_1} & \frac{\partial g_2(X)}{\partial x_2} & \dots & \frac{\partial g_2(X)}{\partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial g_n(X)}{\partial x_1} & \frac{\partial g_n(X)}{\partial x_2} & \dots & \frac{\partial g_n(X)}{\partial x_n} \end{vmatrix}$$

Определение. *Спектральным радиусом* $\rho(A)$ *квадратной матрицы A называется максимальный из модулей ее собственных значений:*

$$\rho(A) = \max\{|\lambda_1|, |\lambda_2|, \dots, |\lambda_n|\}. \qquad (4.6)$$

где $\lambda_1, \lambda_2, \dots, \lambda_n$ *- собственные значения (числа) матрицы A.*

Условие сходимости (4.5) итерационного метода (4.4) напоминает достаточный признак сходимости метода последовательных приближений решения нелинейных уравнений (см. п. 3.2), с той лишь разницей, что вместо $f'(x)$ в (4.5) используется матричная производная векторной функции $G(X)$, а операция взятия модуля заменяется оператором вычисления спектрального радиуса.

Условия сходимости итерационного метода сформулировано для произвольных значений X из области определения $G(X)$ и $F(X)$, т.е. для любого начального приближения X^0, и относится к глобальной теореме сходимости.

Однако доказать сходимость итерационных процессов на основании этой теоремы и дать практические рекомендации к обоснованию сходимости итераций во многих реальных случаях не представляется возможным. В связи с этим нашло применение условие локальной сходимости, в котором предполагается, что решение X^* существует, а начальное приближение X^0 выбрано достаточно близко к X^*. В этом случае достаточно оценить спектральный радиус матрицы $G(X)$ только в точке X^*, а

результаты распространить на окрестности точки X^*. Такая оценка позволяет теоретически оценить поведение итерационного процесса в окрестностях точного решения.

При необходимости с помощью условия (4.5) можно проверить, будет ли сходиться метод из произвольной точки X^k.

Скорость сходимости итераций

Выполнение условия (4.5) обеспечивает уменьшение расстояния между X^k и X^* $\left(\left|X^k - X^*\right|\right)$ с увеличением номера итераций k.

Скорость, с которой уменьшается $\left|X^k - X^*\right|$ в ближайшей окрестности точного решения, называется скоростью сходимости итераций. Скорость сходимости можно оценить по формуле

$$\left|X^{k+1} - X^*\right| = c\left|X^k - X^*\right|^m \qquad (4.7)$$

где m целое число, c -константа ($|c|<1$).

Если m =1, то итерационный метод имеет линейную скорость сходимости, так как $\left|X^{k+1} - X^*\right|$ является линейной функцией от $\left|X^k - X^*\right|$.

Если m=2, то метод обладает квадратичной скоростью сходимости.
Если $1<m<2$, то говорят о сверхлинейной скорости сходимости.
Если $c>1$, то итерационный метод расходится.

4.2. Метод простой итерации

В большинстве итерационных методов решения систем нелинейных уравнений структура итерационной функции $G(X)$ имеет следующий вид:

$$G(X) = X + B \cdot F(X) \qquad (4.8)$$

где B- некоторая итерационная матрица размера nxn. В результате имеем вместо (4.4) следующую обобщенную итерационную формулу:

$$X^{k+1} = X^k + B \cdot F\left(X^k\right) \qquad (4.9)$$

В методе простой итерации итерационная матрица имеет вид

$$B = h \cdot E, \qquad (4.10)$$

где h - скалярная величина, E - единичная матрица размера $n \times n$.

Тогда итерационная формула метода простой итерации может быть представлена формулой:

$$X^{k+1} = X^k + h \cdot F\left(X^k\right). \qquad (4.11)$$

Сходимость метода простой итерации

Вычислим $G'_X(X)$ для метода простой итерации:

$$\frac{\partial G(X)}{\partial X} = \frac{\partial}{\partial X}\left[X + h \cdot F(X)\right] = \frac{\partial X}{\partial X} + h \cdot \frac{\partial F(X)}{\partial X} \qquad (4.12)$$

Для векторной функции $F(X)$ матрица

$$\frac{\partial F(X)}{\partial X} = \begin{vmatrix} \frac{\partial f_1(X)}{\partial x_1} & \frac{\partial f_1(X)}{\partial x_2} & \dots & \frac{\partial f_1(X)}{\partial x_n} \\ \frac{\partial f_2(X)}{\partial x_1} & \frac{\partial f_2(X)}{\partial x_2} & \dots & \frac{\partial f_2(X)}{\partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial f_n(X)}{\partial x_1} & \frac{\partial f_n(X)}{\partial x_2} & \dots & \frac{\partial f_n(X)}{\partial x_n} \end{vmatrix}$$

получила название матрицы Якоби. Тогда, учитывая обозначение $Я = \frac{\partial F(X)}{\partial (X)}$, для (4.12) имеем

$$G'_X(X) = E + h \cdot Я \ . \qquad (4.13)$$

Для сходимости метода простой итерации требуется, чтобы

$\rho(E + h \cdot Я) < 1$

Пусть $\lambda_1, \lambda_2, \dots, \lambda_n$ собственные значения матрицы Я. Для единичной матрицы E собственные значения $\mu_1 = \mu_2 = \dots = \mu_n = 1$. Покажем, что собственные значения матрицы $E + h \cdot Я$ будут: $1 + h\lambda_1, 1 + h\lambda_2, \dots, 1 + h\lambda_n$.

Первоначально найдем собственные значения матрицы h Я

Если λ_i - собственные значения матрицы Я, то из определения собственного значения следует, что

$$\left| Я - \lambda_i \cdot E \right| = 0 .$$

Тогда справедливо:

$$\left| h \cdot Я - h\lambda_i \cdot E \right| = h \left| Я - \lambda_i \cdot E \right| = 0 \quad \text{, т.е.}$$

$h\lambda_i$ -собственные значения матрицы h Я.

Рассмотрим:

$$\left|\left(E+h\cdot Я\right)-\left(1+h\lambda_i\right)E\right|=\left|E+h\cdot Я-E-h\lambda_i\cdot E\right|=\left|h\cdot Я-h\lambda_i\cdot E\right|=0,$$

из чего следует, что $1+h\lambda_i$ - собственные значения матрицы $E+h\cdot Я$.

Используя определение спектрального радиуса, можно записать условия сходимости для нашего случая:

$$\rho(E+h\cdot Я)=\max\left\{\left|1+h\lambda_1\right|,\ldots,\left|1+h\lambda_n\right|\right\}<1 \qquad (4.14)$$

Условие (4.14) можно заменить системой неравенств:

$$\begin{cases}\left|1+h\lambda_1\right|<1,\\ \left|1+h\lambda_2\right|<1,\\ \ldots\\ \left|1+h\lambda_n\right|<1.\end{cases} \qquad (4.15)$$

Анализ условий (4.15) показывает, что имеет место три случая соотношений собственных чисел λ_i, определяющие сходимость или расходимость метода простой итерации.

Случай 1. Пусть все действительные части собственных значений λ_i отрицательны, т.е. $\mathrm{Re}(\lambda_i)<0$, тогда система неравенств (4.15) будет выполняться, если

$$\left(1+h\,\mathrm{Re}(\lambda_i)\right)^2+h^2\left(\mathrm{Im}(\lambda_i)\right)^2<1,\ i=1,2,\ldots,n,$$

где $\mathrm{Im}(\lambda_i)$-мнимая часть собственного значения λ_i, из чего следует, что параметр h необходимо выбирать положительным из системы неравенств:

$$0<h<\frac{2\left|\mathrm{Re}(\lambda_i)\right|}{\left[\mathrm{Re}(\lambda_i)\right]^2+\left[\mathrm{Im}(\lambda_i)\right]^2},\qquad i=1,2,\ldots,n. \qquad (4.16)$$

Случай 2. Пусть все действительные числа собственных значений положительны. В этом случае метод сходится, если параметр h выбрать по формуле (4.16) со знаком минус.

Случай 3. Если действительные части собственных значений имеют разные знаки, то метод простой итерации расходится.

Метод простой итерации имеет линейную скорость сходимости, в связи с чем медленно сходится к решению системы нелинейных уравнений X^*. На практике используются различные модификации

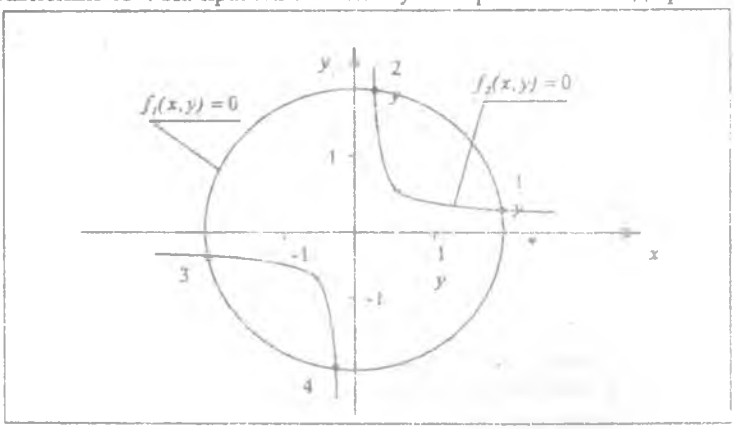


Рис.4.1 Геометрический образ системы нелинейных уравнений.

этого метода, ускоряющие итерационный процесс.

Пример

Пусть требуется найти хотя бы одно решение следующей системы уравнений:

$$\begin{cases} f_1(x,y)=0 \\ f_2(x,y)=0 \end{cases}, \text{ где} \qquad \begin{aligned} f_1(x,y) &= x^2+y^2-4, \\ f_2(x,y) &= xy-1. \end{aligned} \tag{4.17}$$

Четыре возможных варианта решения системы (4.17) образуются как точки пересечения окружности радиуса 2 и двух веток гиперболы $y=1/x$ (рис. 4.1). В данном случае векторная функция $F(X)$ имеет вид

$$F(X)=\begin{Vmatrix} x^2+y^2-4 \\ xy-1 \end{Vmatrix}.$$

В векторной форме метод простой итерации сможет быть представлен:

$$\begin{Vmatrix} x_{k+1} \\ y_{k+1} \end{Vmatrix} = \begin{Vmatrix} x_k \\ y_k \end{Vmatrix} + h \begin{Vmatrix} x_k^2 + y_k^2 - 4 \\ x_k y_k - 1 \end{Vmatrix}.$$

Произведя соответствующие действия над матрицами, получим координатную форму метода простой итерации:

$$\begin{cases} x_{k+1} = x_k + h\left(x_k^2 + y_k^2 - 4\right), \\ y_{k+1} = y_k + h\left(x_k y_k - 1\right). \end{cases} \tag{4.18}$$

Исследуем сходимость метода для нашего примера. Предварительно построим матрицу Якоби:

$$\text{Я} = \begin{Vmatrix} \dfrac{\partial f_1(X)}{\partial x_1} & \dfrac{\partial f_1(X)}{\partial x_2} \\ \dfrac{\partial f_2(X)}{\partial x_1} & \dfrac{\partial f_2(X)}{\partial x_2} \end{Vmatrix} = \begin{Vmatrix} 2x & 2y \\ y & x \end{Vmatrix} \tag{4.19}$$

Для нахождения собственных значений матрицы Я составим характеристическое уравнение $|\ \text{Я} - \lambda \cdot E\ | = 0$:

$$\begin{vmatrix} 2x - \lambda & 2y \\ y & x - \lambda \end{vmatrix} = \lambda^2 - 3\lambda x + 2x^2 - 2y^2 = 0, \text{ откуда}$$

$$\lambda_{1,2} = \frac{3x \pm \sqrt{x^2 + 8y^2}}{2}. \tag{4.20}$$

Решение системы (4.17) будем искать для действительных x и y.

Для сходимости итерационного процесса (4.18) требуется, чтобы λ_1 и λ_2 имели одинаковые знаки.

Пусть $x>0$, тогда λ_1 и λ_2 положительны, если

$$3x > \sqrt{x^2 + 8y^2}.$$

Из уравнения $9x^2 = x^2 + 8y^2$ можно найти границу положительности λ_1 и λ_2, она описывается уравнением

$$x^2 = y^2. \tag{4.21}$$

Нетрудно показать, что тем же уравнением описывается граница отрицательности λ_1 и λ_2. Уравнение (4.21) представляет две скрещивающиеся прямые, проходящие под углом $\pm 45°$ к оси Ох.

На рисунке 4.2 штриховкой показана область сходимости метода простой итерации для системы уравнений (4.17). Как видно из рисунка, в области сходимости находится всего два решения (1 и 3) из 4 возможных вариантов. Выбор начального приближения

X^0 и параметра h необходимо производить с учетом полученных зон сходимости метода. Величину шага можно определить из условий (4.16). Приведенный пример демонстрирует причины потери "работоспособности" итерационных методов в случае неудачного выбора начального сближения. Например, для X^0=(0,2;1,9), несмотря на близость X к решению 2 системы (4.17), метод простой итерации не только не приблизится к решению, а удаляется от него на значительное расстояние.

К сожалению, полное исследование областей сходимости итерационных методов - задача на порядок сложнее, чем нахождение решения систем уравнений. Поэтому на практике такие исследования не проводятся.

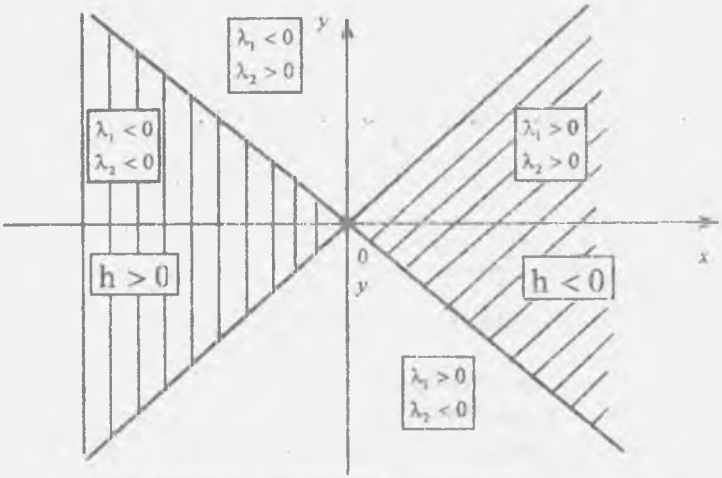


Рис. 4.2 Область сходимости метода простой итерации

Значительно проще ввести в программу критерий проверки сходимости метода, например, сравнивая расстояния между соседними итерациями:

$$\left|X^k - X^{k-1}\right| \geq \left|X^{k+1} - X^k\right|. \qquad (4.22)$$

В случае расхождения итерационного процесса (невыполнения условия (4.22)) необходимо пересмотреть начальное приближение.

Методы коррекции начального приближения были рассмотрены в разделе для методов решения нелинейных уравнений.

4.3. Метод Зейделя

Метод Зейделя рассмотрим на примере решения системы линейных уравнений вида

$$F(X) = AX + B = 0. \tag{4.23}$$

В рассмотренном выше методе простой итерации элементы вектора X^{k+1} вычисляются по формуле

$$x_i^{k+1} = x_i^k + h\left(\sum_{j=1}^{n} a_{ij}x_j^k + b_i\right), \quad i = 1,2,\ldots,n.$$

В методе Зейделя поэлементное нахождение x_i^{k+1} производится с учетом уже вычисленных значений $x_1^{k+1}, x_2^{k+1}, \ldots, x_{i-1}^{k+1}$

Запишем первое уравнение системы (4.23) в виде

$$a_{11}x_1 + \sum_{j=2}^{n} a_{1j}x_j^k + b_1 = 0 \ .$$

Из этого уравнения определим x_1. Положим $x_1^{k+1} = x_1$ и подставим это значение во второе уравнение системы:

$$a_{22}x_2 + a_{21}x_1^{k+1} + \sum_{j=3}^{n} a_{2j}x_j^k + b_2 = 0 \ .$$

Действуя таким же способом по отношению к другим уравнениям, в результате для каждой i – й переменной будем иметь линейное уравнение с одной неизвестной вида

$$\sum_{j=1}^{i-1} a_{ij}x_j^{k+1} + a_{ij}x_i + \sum_{j=i+1}^{n} a_{ij}x_j^k + b_i = 0 \ .$$

Чтобы записать итерационную формулу метода Зейделя, представим матрицу A в виде $A = D + L + U$, где D - диагональная матрица, L - строго нижнетреугольная, U - строго верхнетреугольная матрица (т.е. все диагональные элементы L и U равны нулю).

Тогда рассмотренную выше последовательность вычислений можно записать в виде

$$(D + L)X^{k+1} + UX^k + B = 0 \ . \tag{4.24}$$

Из (4.24) несложно получить итерационную формулу метода Зейделя в виде

$$X^{k+1} = X^k - (D + L)^{-1}F(X^k). \tag{4.25}$$

Сходимость метода Зейделя

Для метода Зейделя итерационная матрица имеет следующий вид :

$$G(X) = X - (D + L)^{-1} F(X).$$

Тогда условия глобальной сходимости метода Зейделя определяются неравенством $\rho(G'_X(X))<1$. Можно показать, что это условие выполняется, если матрица обладает свойством диагонального преобладания:

$$|a_{jj}| > \sum_{j \neq i} |a_{ij}| \qquad (4.26)$$

для всех $i = 1,2,...,n$. В этом случае говорят, что матрица A является матрицей с диагональным преобладанием. Условие (4.26) является достаточным условием сходимости метода Зейделя.

Метод Зейделя, как и метод простой итерации, имеет линейную скорость сходимости, но большую, чем у метода простой итерации. Ускорить сходимость метода Зейделя можно, введя в итерационную матрицу параметр релаксации ω:

$$X^{k+1} = X^k - \omega(D + \omega L)^{-1} F(X^k). \qquad (4.27)$$

При ω=1 релаксационный метод (4.27) совпадает с методом Зейделя, при $\omega > 1$ итерационный процесс схождения к решению X^* значительно ускоряется. При правильном выборе ω можно получить 20 кратную экономию машинного времени.

В процессе вывода итерационной формулы метода Зейделя рассматривался частный случай системы линейных уравнений, при этом матрица A представлялась суммой матриц D, L, U . В общем случае (решения системы нелинейных уравнений) в формуле (4.25) вместо матрицы A рассматривается разложение матрицы Якоби в соответствующей точке итерационного процесса по формуле

$$Я(X^k) = D + L + U.$$

4.4 .Метод Ньютона

Пусть итерационная матрица B в формуле (4.8) имеет вид $B = -Я^{-1}$,

тогда итерационный процесс метода Ньютона строится в соответствии с формулой

$$X^{k+1} = X^k - Я^{-1}F(X^k). \qquad (4.28)$$

Метод итераций, реализующий итерационный процесс по формуле (4.28), получил название метода Ньютона. Метод Ньютона в окрестности решения системы нелинейных уравнений X^* имеет очень высокую скорость сходимости. Как правило, это квадратичная скорость сходимости. В среднем метод Ньютона сходится за число шагов, приблизительно равное размерности пространства переменных.

Сходимость метода Ньютона

Итерационная функция метода Ньютона имеет вид

$$G(X) = X - Я^{-1}F(X). \qquad (4.29)$$

Чтобы метод Ньютона сходился к точному решению X^* из произвольного начального приближения X^0, достаточно, чтобы

$$\rho\left(G_X'(X)\right) < 1 \qquad \text{,т.е. } \rho\left(E - \left(Я^{-1}F(X)\right)_X'\right) < 1$$

(матричная производная берется от произведения матриц $Я^{-1}F(X)$).

К сожалению, глобальная сходимость метода Ньютона достигается при выполнении жестких ограничений на функцию $F(X)$. В частности, она должна быть непрерывной, дифференцируемой и строго выпуклой (вогнутой) в области определения $F(X)$.

Для изучения вопроса локальной сходимости метода Ньютона необходимо исследовать поведение $G'_X(X)$ в окрестности решения X^*.

Полагая матрицу Якоби в окрестности X^* постоянной, имеем

$$\rho\left(E - Я^{-1}\left(X^*\right)F_X'(X)\right) = \rho\left(E - Я^{-1}\left(X^*\right)Я\left(X^*\right)\right) = 0, \qquad (4.30)$$

т.к. $F_X'(X) = Я\left(X^*\right)$.

Из (4.30) следует, что для метода Ньютона всегда есть некоторая окрестность точки X^*, в которой выполняются условия сходимости итераций $\rho\left(G_X'(X)\right) < 1$. Другими словами, если начальное приближение выбрано достаточно близко к X^*, то метод Ньютона всегда сходится к точному решению X^*. Для выбора или коррекции начального приближения можно рекомендовать методы,

предложенные в п. 3.6.

Особенно эффективен метод продолжения по параметру. Учитывая свойство локальной сходимости метода Ньютона, всегда можно подобрать такую последовательность параметров $t_1, t_2, ..., t_m$, что начальные приближения частных задач $H(X;t_i)=0$: $X^0, X^1, ..., X^m$ будут находиться в локальных окрестностях их решений $X^1, X^2, ..., X^*$.

В качестве условия перехода к корректировке начального приближения можно предложить следующий надежный, универсальный критерий.

Если не удалось найти решение за 6,7 итераций с заданной точностью, то необходимо выбрать новое начальное приближение.

Метод Ньютона наиболее эффективен при аналитическом вычислении элементов матрицы Якоби. В тех случаях, когда найти аналитические выражения элементов матрицы Я не удается, частные производные $\frac{\partial f_i(x)}{\partial x_j}$ можно заменить конечно-разностными аппроксимациями:

$$\frac{\partial f_i(x)}{\partial x_j} \approx \frac{f_i(x_1,...,x_j+\Delta x_j,...,x_n)-f(x_1,...,x_j,...,x_n)}{\Delta x_j} \qquad (4.31)$$

или

$$\frac{\partial f_i(x)}{\partial x_j} \approx \frac{f_i(x_1,...,x_j+\Delta x_j,...,x_n)-f(x_1,...,x_j-\Delta x_j,...,x_n)}{2\Delta x_j}, \qquad (4.32)$$

где Δx_j - заданный параметр дискретизации.

С учетом ошибок округления формула (4.31) более предпочтительна. С другой стороны, формула (4.32) требует меньшее количество дополнительных вычислений функций $f_1(x), f_2(x), ..., f_n(x)$. Дискретный метод Ньютона требует многократных вычислений каждой из нелинейных функций $f_i(x)$ на каждом итерационном шаге метода. От точности вычисления частных производных зависит не только скорость сходимости метода, но и сама сходимость метода.

Проблемы точности вычисления частных производных будут рассмотрены в главе 5.

4.5. Решение систем линейных уравнений

Задача решения системы линейных уравнений возникает практически в каждом разделе прикладной математики. Например, многие методы решения обыкновенных дифференциальных уравнений сводятся к задаче решения систем линейных уравнений. Как было показано выше, решение задачи среднеквадратической аппроксимации функции приводит к необходимости решения системы линейных уравнений и т.д.

Рассмотрим методы решения систем линейных уравнений общего вида:

$$A\,X = B, \tag{4.33}$$

где A - матрица коэффициентов системы размера (nxn);

X - вектор-столбец неизвестных переменных;

B - вектор-столбец свободных членов.

Для решения системы (4.33) можно использовать метод простой итерации, однако он характеризуется малой скоростью сходимости. Поэтому для решения (4.33) часто используют прямые методы решения, из которых наибольшее распространение получил метод Гаусса.

Метод Гаусса основан на последовательном исключении неизвестных в уравнении системы до тех пор, пока не останется только одно уравнение с одним неизвестным в левой части. Последнее уравнение решается, полученное значение неизвестной подставляется в предыдущее уравнение с двумя неизвестными, и находится следующая неизвестная. Полученные значения неизвестных подставляются в уравнение с тремя неизвестными, и т.д., в обратном порядке, пока не будут найдены все элементы вектора. Таким образом, метод Гаусса состоит из двух этапов:

1) прямого хода;

2) обратного хода.

ЭТАП 1. Прямой ход выполняется за n-1 стадий, где на каждой $k-$ й стадии исключается очередная известная x_k по формулам:

$$m_{ik} = a_{ik}^{(k)} \Big/ a_{kk}^{(k)};$$

$$\begin{cases} a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik} a_{kj}^{(k)} \\ b_i^{(k+1)} = b_i^{(k)} - m_{ik} b_k^{(k)} \end{cases} \quad \text{при} \quad \begin{matrix} j = k, k+1, \ldots, n; \\ i = k+1, \ldots, n; \end{matrix} \tag{4.34}$$

В результате выполнения прямого хода метода Гаусса система (4.33) эквивалентными преобразованиями приводится к виду

$$UX = \widetilde{B}, \qquad (4.35)$$

где U - верхнетреугольная матрица, все элементы которой ниже главной диагонали равны 0;

$\widetilde{B}$-преобразованный вектор-столбец свободных членов.

ЭТАП 2. Обратный ход состоит из $(n+1)$ стадий обратных подстановок снизу вверх. Если решение системы линейных уравнений существует и единственно, то последнее уравнение системы (4.35) дает

$$x_n^* = \widetilde{b}_n^{(n)} / u_{nn} \ . \qquad (4.36)$$

Остальные неизвестные находятся последовательно по формуле

$$x_i^* = \frac{\widetilde{b}_i^{(n)} - \sum_{j=i+1}^{n} u_{ij} x_j^*}{u_{ii}}; \qquad i = n-1,\ldots,1 \ . \qquad (4.37)$$

Ошибки округления метода Гаусса

Для устойчивой реализации метода Гаусса необходимо, чтобы все диагональные элементы $a_{kk}^{(k)}$ отличались от нуля. В случае получения нулевого элемента на $k-$й стадии необходимо выбрать из оставшихся $k, k+1,\ldots,n$ уравнений уравнение с ненулевым элементом на соответствующем месте и поменять местами уравнения.

Количество "длинных" операций (умножений и делений), необходимых для однократного решения системы, может быть вычислено по формуле

$$N = \left(n^3 + 3n^2 - n\right)/3 \ .$$

При реализации таких операций на ЭВМ возникают ошибки арифметических операций, которые неизбежно влияют на результат решения задачи.

Пусть на $k-$й стадии мы собираемся исключить переменную x_k. Положим для простоты, что относительные погрешности округления при выполнении операций деления, умножения и вычитания одинаковы: $\delta = 5 \cdot 10^{-t}$. Относительные ошибки, содержащиеся в коэффициентах $a_{ij}^{(k)}$, обозначим α_{ij}. В соответствии с методом Гаусса коэффициенты $a_{ij}^{(k+1)}$ вычитаются по формуле

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik} a_{kj}^{(k)},$$

где m_{ik} вычисляется по формуле (4.39). Граф этого вычислительного процесса приведен на рисунке 4.3.

Определим теперь относительную ошибку округления при вычислении коэффициента $a_{ij}^{(k+1)}$. Обозначим через e_{ij} абсолютную ошибку коэффициента $a_{ij}^{(k+1)}$, тогда на основании графа рисунок 4.3 имеем

$$\frac{e_{ij}}{a_{ij}^{(k+1)}} = -\frac{m_{ik} a_{kj}^{(k)}}{a_{ij}^{(k+1)}} \left(\alpha_{ik} - \alpha_{kk} + \alpha_{kj} + \delta + \delta\right) + \frac{a_{ij}^{(k)}}{a_{ij}^{(k+1)}} \alpha_{ij} + \delta,$$

откуда, положив $\alpha_{ij} \le K \cdot 10^{-t}$ (K>5), получаем следующую оценку:

$$\left|e_{ij}\right| \le \left[3(K+5) \cdot \left|a_{kj}^{(k)}\right| \cdot \left|m_{ik}\right| + 10\left|a_{ij}^{(k)}\right|\right] 10^{-t}. \qquad (4.38)$$

Как видно из формулы (4.38), единственным фактором, с помощью которого мы можем повлиять на величину ошибки e_{ij}, является коэффициент $m_{ik} = a_{ik}^{(k)} / a_{kk}^{(k)}$. Уменьшение $|m_{ik}|$ уменьшает и величину ошибки e_{ij}. Но для того, чтобы сделать $|m_{ik}|$ как можно меньшим, необходимо, чтобы $|a_{kk}^{(k)}|$ было возможно большим.

Последнего можно добиться путем перестановки оставшихся уравнений так, чтобы наибольший по модулю коэффициент при x_k попал на главную диагональ. Описанный способ решения линейных уравнений часто называют методом главного элемента. Ошибку округления можно еще уменьшить, если переставлять не только строки системы уравнений, но и столбцы. В этом случае необходимо искать наибольший по абсолютной величине коэффициент среди всех оставшихся ниже k – й строки матрицы A и правее k-го столбца.

Метод LU-разложения

В тех случаях, когда приходится решать большое количество систем линейных уравнений с одинаковой матрицей A и разными векторами B, определенное преимущество имеет метод LU-разложения.

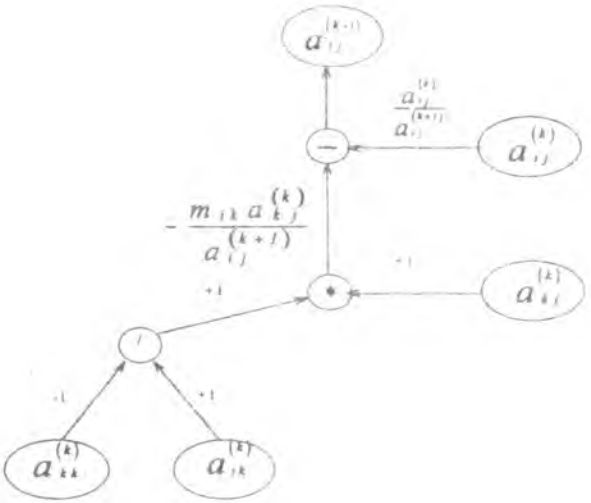


Рис.4.3.Граф вычислительного процесса при решении системы уравнений методом Гаусса

Идея метода заключается в разложении матрицы A в виде произведения нижнетреугольной матрицы L и верхнетреугольной матрицы U, все диагональные элементы которой равны 1, т.е. $A=LU$.

Тогда

$$L(UX)=B, \tag{4.39}$$

положив

$$UX=Y, \tag{4.40}$$

имеем

$$LY=B. \tag{4.41}$$

Решение системы уравнений (4.39) после разложения матрицы производится в два этапа:

На первом этапе решается система (4.41) с помощью алгоритма, аналогичного обратному ходу Гаусса (только сверху вниз). На втором этапе для найденного Y решается система уравнений (4.40) с верхнетреугольной матрицей U по формуле обратного хода Гаусса. Разложение матрицы A выполняется за n стадий. Элементы матриц L и U размещаются в ходе разложения на месте элементов матрицы A следующим образом (рассмотрим случай n=4):

$$A_{LU} = \begin{Vmatrix} l_{11} & u_{12} & u_{13} & u_{14} \\ l_{21} & l_{22} & u_{23} & u_{24} \\ l_{31} & l_{32} & l_{33} & u_{34} \\ l_{41} & l_{42} & l_{43} & l_{44} \end{Vmatrix}$$

На стадии 1 первый столбец матрицы A остается без изменений, т.е. $l_{i1} = a_{i1} \quad (i = 1,2,...n)$, а первая строка рассчитывается по формуле

$$u_{1j} = a_{1j} / l_{11}; \quad j = 2,3,...n. \tag{4.42}$$

На каждой следующей стадии разложения последовательно пересчитываются элементы a_{ij} очередного j-го столбца по формулам

$$l_{ij} = a_{ij} - \sum_{k=1}^{j-1} l_{ik} u_{kj}, \qquad i = j, j+1,...n, \tag{4.43}$$

оставшиеся элементы a_{ij} очередной j-й строки по формулам

$$u_{ji} = \left(a_{ji} - \sum_{k=1}^{i-1} l_{ik} u_{kj} \right) \Big/ l_{ii}, \qquad j = i+1,...,n. \tag{4.44}$$

По количеству "длинных" операций и по затратам памяти ЭВМ метод LU-разложения эквивалентен методу Гаусса.

Метод Гаусса-Зейделя

В тех случаях, когда матрица A сильно разряжена (имеет большое количество нулевых элементов), хорошо себя зарекомендовал итерационный метод Гаусса-Зейделя. По существу, это метод Зейделя, описанный в 4.3, примененный для решения систем линейных уравнений.

В заключение отметим, что в настоящее время существует достаточно большое количество численных методов решения систем линейных уравнений, учитывающих разнообразные структурные особенности матрицы A.

ГЛАВА 5

Численные методы интегрирования и дифференцирования функций

5.1. Введение

Задачи, в которых требуется вычисление интегралов или производных функций, возникают почти во всех областях прикладной математики. Например, проблема численного дифференцирования функций встречается в методе Ньютона решения систем нелинейных уравнений, в методах решения обыкновенных дифференциальных уравнений, в задачах отыскания экстремумов функций одной и многих переменных и т.д. С другой стороны, многие критерии оценки качества проектируемого изделия вычисляются с помощью определенных интегралов. В теории вероятности интеграл от функции плотности вероятности определяет величину вероятности некоторого события. С помощью интегралов вычисляются геометрические характеристики объектов и т.д.

Иногда удается найти аналитическую формулу для вычисления определённого интеграла или дифференциала функции, но значительно чаще этого сделать не удается. В таких ситуациях приходится применять различные методы численного интегрирования или дифференцирования функций.

Задача численного интегрирования функции заключается в вычислении определенного интеграла на основании ряда значений подынтегральной функции.

В настоящее время разработано достаточно большое количество методов численного интегрирования функций, учитывающих различные особенности в постановке задачи. В этой главе будут рассмотрены общие подходы к решению указанных задач.

5.2. Правило трапеций

Рассмотрим задачу вычисления определенного интеграла:

$$I = \int_a^b f(x)dx, \qquad (5.1)$$

где a,b - конечны; $f(x)$ - непрерывная функция на отрезке

интегрирования

Общий подход к решению задачи заключается в разбиении отрезка $[a,b]$ на множество отрезков меньших размеров и вычислении интеграла как суммы приближенно вычисленных площадей полосок, получившихся при таком разбиении (рис. 5.1).

Разобьем отрезок $[a,b]$ на n равных частей точками $x_0 < x_1 < x_2 < ... < x_n$, каждая длиной $h = (b-a)/n$. Рассмотрим один из интервалов, представленный на рис. 5.2. Площадь, лежащую под кривой $y = f(x)$ между точками x_i и x_{i+1}, будем приближённо вычислять как площадь трапеции ABCD, так что

$$J_i = \int_{x_i}^{x_{i+1}} f(x)dx \approx S_{ABCD} = \frac{1}{2}h(y_i + y_{i+1}). \qquad (5.2)$$

Тогда величину определённого интеграла можно оценить.

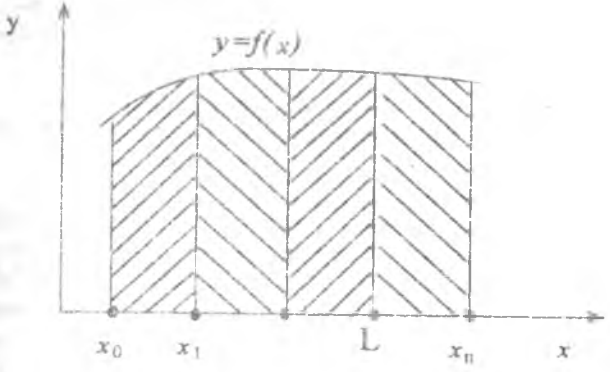


Рис. 5.1. Вычисление определенного интеграла

$$J = \sum_{n=0}^{n-1} J_i \approx \frac{h}{2}(y_0 + 2y_1 + 2y_2 + ..+2y_{n-1} + y_n) . \qquad (5.3)$$

Эта формула описывает хорошо известное правило трапеций для численного интегрирования. Это один из простейших методов численного интегрирования.

Ошибка ограничения для метода трапеций

Предположим, что $y = f(x) \in C^2_{[a,b]}$- непрерывная вместе со своими первой и второй производными на $[a,b]$ функции. Остаточный член на *i*- м участке равен:

$$R_i = \int_{x_i}^{x_{i+1}} y dx - \frac{h}{2}(y_i + y_{i+1}) \quad \text{или}$$

$$R(h) = \int_{x_i}^{x_i+h} y dx - \frac{h}{2}\left(y(x_i) + y(x_{i+1})\right). \qquad (5.4)$$

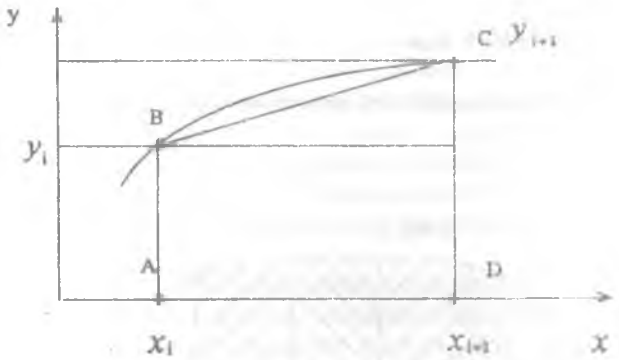


Рис.5.2. Вычисление площади частичного отрезка

Дифференцируя формулу (5.4.) по h последовательно два раза, получим

$$R'(h) = y(x_i + h) - \frac{1}{2}\left(y(x_i) + y(x_{i+1} + h)\right) - \frac{h}{2}y'(x_i + h) = \frac{1}{2}\left(y(x_i + h) - y(x_i)\right) - \frac{h}{2}y'(x_i + h), \qquad (5.5)$$

$$R''(h) = \frac{1}{2}y'(x_i + h) - \frac{1}{2}y'(x_i + h) - \frac{h}{2}y''(x_i + h) = -\frac{h}{2}y''(x_i + h), \qquad (5.6)$$

причём очевидно, что $R(0) = R'(0) = 0$.

Формулу (5.6) проинтегрируем по h и, используя теорему о среднем, получим

$$R'(h) = R'(0) + \int_0^h R''(t)dt = -\frac{1}{2}\int_0^h t y''(x_0 + t)dt = -\frac{1}{2}y''(\xi 1)\int_0^h t dt = -\frac{h^2}{4}y''(\xi 1), \qquad (5.7)$$

где $\xi 1 \in (x_i, x_{i+1})$.

Аналогично:

$$R(h) = R(0) + \int_0^h R'(t)dt = -\frac{1}{4}\int_0^h t^2 y''(\xi)dt = -\frac{1}{4}y''(\xi)\int_0^h t^2 dt = -\frac{h^3}{12}y''(\xi), \quad (5.8)$$

где $\xi \in (x_i, x_{i+1})$.

Таким образом, окончательно имеем

$$R_i(h) = -\frac{h^3}{12}y''(\xi), \quad (5.9)$$

где $\xi \in (x_i, x_{i+1})$.

Тогда остаточный член на всём отрезке $[a,b]$ можно вычислить по формуле

$$R(h) = \sum_{i=0}^{n-1} R_i(h) = -\frac{h^3}{12}\sum_{i=0}^{n} y''(\xi_i) = \frac{n \cdot h^3}{12}y''(\xi) = -\frac{(b-a)h^2}{12}y''(\xi), \quad (5.10)$$

где $\xi \in [a,b]$.

Ошибка округления правила трапеций

Для вычисления ошибки округления составим граф вычислительного процесса правила трапеций (рис.5.3.). Предполагается, что все члены суммы, которые необходимо умножить на 2, сначала складываются, а их сумма умножается на 2.

Пусть $\delta_i (i = 0,1,...n)$ - относительные ошибки каждой величины y_i. Пусть α - относительная ошибка операции сложения, μ - относительная ошибка операции умножения. Тогда относительную ошибку округления для формулы вычисления интеграла по правилу трапеций (5.3) в соответствии с графом вычислительного процесса можно вычислить:

$$\frac{e_I}{I}=\mu+\alpha+\alpha\frac{y_0+y_n}{y_0+2y_1+\ldots+y_n}+\delta_0\frac{y_0}{y_0+2y_1+\ldots+y_n}+\delta_n\frac{y_n}{y_0+2y_1+\ldots+y_n}+\mu\frac{2y_1+\ldots+2y_{n-1}}{y_0+2y_1+\ldots+y_n}+$$

$$+\alpha\frac{2y_1+\ldots+2y_{n-1}}{y_0+2y_1+\ldots+y_n}+\alpha\frac{2(y_1+\ldots+y_{n-2})}{y_0+2y_1+\ldots+y_n}+\ldots+\alpha\frac{2(y_1+y_2)}{y_0+2y_1+\ldots+y_n}+\ldots+\delta_{n-1}\frac{2y_{n-1}}{y_0+2y_1+\ldots+y_n}+\ldots+$$

$$+\delta_3\frac{2y_3}{y_0+2y_1+\ldots+y_n}+\delta_2\frac{2y_2}{y_0+2y_1+\ldots+y_n}+\delta_1\frac{2y_1}{y_0+2y_1+\ldots+y_n}.$$

Абсолютная ошибка равна:

$$e_I=h\left(\delta_0\frac{y_0}{2}+\delta_1y_1+\ldots+\delta_{n-1}y_{n-1}+\delta_n\frac{y_n}{2}\right)+h\left[\alpha(y_1+y_2)+\alpha(y_1+y_2+y_3)+\ldots+\alpha(y_1+y_2+\ldots+y_{n-1})\right]+$$

$$+\frac{h}{2}\left[\alpha(y_0+y_n)+\alpha(y_0+2y_1+\ldots+2y_{n-1}+y_n)+\mu(2y_1+\ldots+2y_{n-1})+\mu(y_0+2y_1+\ldots+2y_{n-1}+y_n)\right]=$$

$$=[\text{Положим}\,\delta=\delta_0=\delta_1=\ldots=\delta_n]=$$

$$=\frac{h}{2}\left(\frac{y_0+2y_1+\ldots+2y_{n-1}+y_n}{n}\right)n\delta+\frac{h}{2}\alpha\left(\frac{y_0+2y_1+\ldots+2y_{n-1}+y_n}{n}n+\ldots+\frac{y_0+y_n}{2}2\right)+h\alpha\left(\frac{y_1+y_2}{2}2+\right.$$

$$\left.+\frac{y_1+y_2+y_3}{3}3+\ldots+\frac{y_1+\ldots+y_{n-1}}{n-1}(n-1)\right)+h\mu\left(\frac{y_0+2y_1+\ldots+2y_{n-1}+y_n}{n}n+2\frac{y_1+\ldots+y_{n-1}}{n-1}(n-1)\right)$$

Положим $|\alpha|=|\mu|\le\varepsilon$, $|\delta|\le\varphi\varepsilon$, где φ-коэффициент пропорциональности;

$|\vec{y}|=\frac{1}{n}\sum_{i=0}^{n}y_i$ среднее из всех y_i. Тогда

$$|e_I|\le h|\vec{y}|n\varphi\varepsilon+h\alpha|\vec{y}|\sum_{j=2}^{n-1}j+\frac{h}{2}|\vec{y}|\varepsilon\left(n+2+2(n-1)\right)\le h|\vec{y}|n\varphi\varepsilon+h|\vec{y}|\varepsilon\sum_{j=2}^{n-1}j+\frac{h}{2}|\vec{y}|\varepsilon 3n. \tag{5.11}$$

Учитывая, что $\sum_{j=1}^{m}j=\frac{m(m+1)}{2}$, имеем $\sum_{j=2}^{n-1}j=\frac{n^2-n}{2}-1$.

Тогда

$$|e_I|\le\frac{h}{2}|\vec{y}|n\varphi\varepsilon+\frac{h}{2}|\vec{y}|n^2\varepsilon-\frac{h}{2}|\vec{y}|n\varepsilon-\frac{h}{2}|\vec{y}|\varepsilon+\frac{h}{2}|\vec{y}|\varepsilon 3n=\frac{h}{2}|\vec{y}|\varepsilon\left(n^2+(n\varphi+2n)-1\right). \tag{5.12}$$

При малом h (большом n) главная часть ошибки округления в формуле (5.12) заключена в члене n^2, поэтому можно дать следующую верхнюю оценку e_I:

$$|e_I|\le\frac{h}{2}|\vec{y}|\varepsilon n^2=\frac{|\vec{y}|\varepsilon(b-a)^2}{2h} \tag{5.13}$$

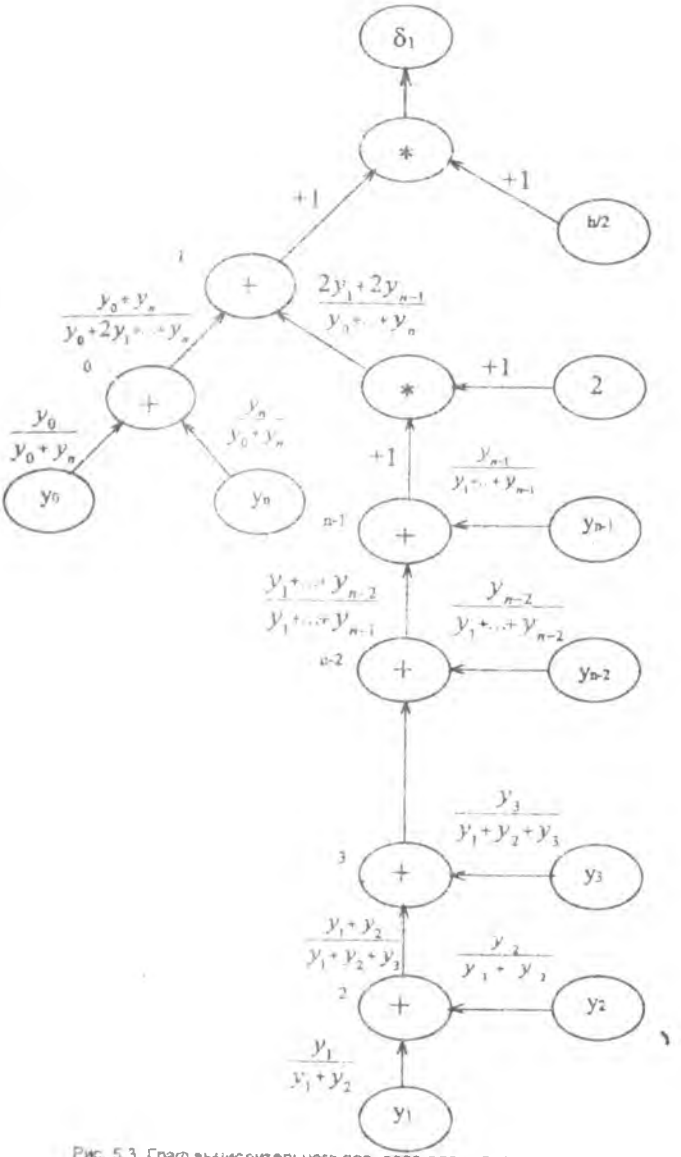


Рис. 5.3. Граф вычислительного процесса для интегрирования функции по правилу трапеций

Полученный результат показывает, что ошибка ограничения с уменьшением h не уменьшается, как это имеет место для ошибки ограничения, а увеличивается. Следовательно, влияние ошибок округления и ограничения в методах численного интегрирования имеет противоположную направленность. Из чего можно заключить, что существует некоторая величина шага интегрирования h_{opt} (n_{opt}), при которой суммарная ошибка вычисления интеграла наименьшая. На рисунке 5.4. представлены графики ошибок ограничения и округления, а также суммарная ошибка вычисления интеграла.

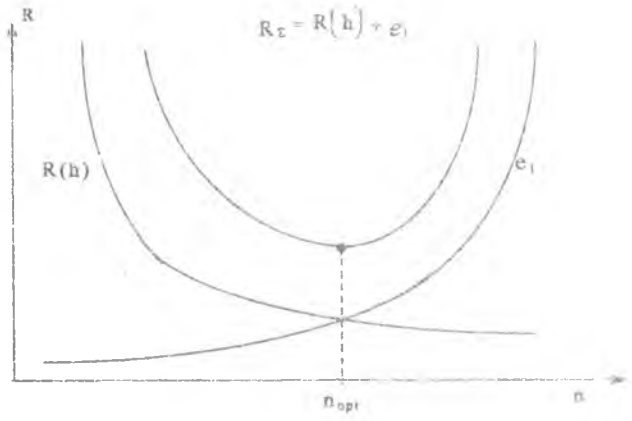


Рис. 5.4. Зависимость суммарной ошибки (округления и ограничения) от количества интервалов интегрирования

5.3. Правило Симпсона

Интегрирование функции $f(x)$ по правилу трапеций можно интерпретировать как замену исходной функции $f(x)$ некоторой кусочно-линейной функцией (после разбиения общего интервала интегрирования на множество отрезков, на каждом из которых функция заменяется прямой линией), от которой и вычисляется приближенное значение искомого интеграла. Ошибка метода в этом случае определяется грубостью предложенного способа аппроксимации функции. Естественно допустить, что если исходную функцию $f(x)$ приближать на отрезках не линейными

функциями, а полиномами более высоких порядков, то ошибка метода интегрирования должна уменьшиться. Для правила Симпсона в качестве функции, с помощью которой осуществляется приближение исходной функции на частичных отрезках интегрирования, выбрана парабола.

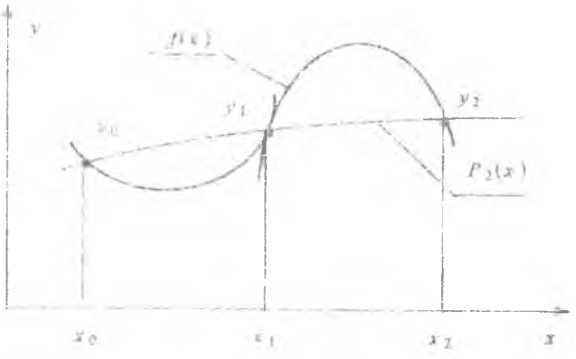


Рис 5.5 Геометрическое представление правила Симпсона на $[x_0, x_2]$

Отрезок $[a,b]$ разобьем на $2n$ равных частей. Рассмотрим частичный отрезок интегрирования $[x_0, x_2]$ (рис. 5.5.).

Для исходной функции $f(x)$ на $[x_0, x_2]$ по трем точкам можно построить интерполяционный многочлен

$$P_2(x) = y_0 + \Delta y_0 \frac{x - x_0}{h} + \frac{\Delta^2 y_0}{2} \frac{(x - x_0)(x - x_1)}{h^2},$$

тогда справедливо приближенное равенство

$$J_0 = \int_{x_0}^{x_2} f(x)dx \approx \int_{x_0}^{x_2} P_2(x)dx = \frac{h}{3}(y_0 + 4y_1 + y_2). \qquad (5.14)$$

Искомое значение интеграла от функции $f(x)$ на $[a,b]$ можно найти по формуле

$$J = \int_a^b f(x)dx \approx \sum_{k=0}^{n} J_{2k}, \qquad (5.15)$$

где

$J_{2k} = \frac{h}{3}\left(y_{2k} + 4y_{2k+1} + y_{2k+2}\right)$, тогда имеем

$$J \approx \frac{h}{3}\left[\left(y_0 + y_{2n}\right) + 2\left(y_2 + y_4 + \ldots + y_{2n-2}\right) + 4\left(y_1 + y_3 + \ldots + y_{2n-1}\right)\right], \quad (5.16)$$

Рис.5.6. Зависимость суммарной ошибки (ограничения и округления) от количества интервалов интегрирования

Формула (5.16) получила название метода интегрирования функции по правилу Симпсона и является одним из наиболее распространенных и применяемых методов интегрирования. Правило Симпсона удачно сочетает простоту метода и высокую точность. Остаточный член формулы Симпсона можно оценить формулой

$$R(h) = -\frac{(b-a)h^4}{180} y^{(4)}(\xi), \text{ где } \xi \in [a,b].$$

На рис. 5.6. для сравнения представлены графики суммарных ошибок интегрирования функции $y = \sin x$ для правила трапеции и метода Симпсона. Из графиков видно несомненное превосходство правила Симпсона.

5.4. Экстраполяционный переход к пределу

Оригинальную идею повышения точности интегрирования функции предложил Ричардсон. Рассмотрим применение этой идеи для метода трапеции.

Для метода трапеции ошибка ограничения оценивается

величиной $R(h) = ch^2$, где $c = -\frac{(b-a)}{12} y''(\xi)$, $\xi \in [a,b]$. Пусть $y''(\xi) = const$, тогда коэффициент c также является постоянной величиной. Откуда справедливо:

$$J = J_h + ch^2, \tag{5.17}$$

где J_h -значение интеграла, вычисленное по правилу трапеции с шагом интегрирования $h = (b-a)/n$.

Предположим теперь, что выбрана некоторая другая величина шага разбиения $k = (b-a)/m$, причем $m \neq n$, тогда интеграл J может быть вычислен по формуле

$$J = J_k + ck^2. \tag{5.18}$$

Из выражения (5.17) и (5.18) определим величину константы c:

$$c = \frac{J_h - J_k}{k^2 - h^2}. \tag{5.19}$$

Величину интеграла с учетом поправки можно вычислить по формуле

$$J = J_h + \frac{J_h - J_k}{\frac{k^2}{h^2} - 1}. \tag{5.20}$$

Вычисленное таким образом значение интеграла является лучшим приближением, чем J_h и J_k. Если же $y''(x)$ действительно постоянна при $\xi \in [a,b]$, то ошибка ограничения в формуле (5.20) равна 0.

5.5. Численное интегрирование с использованием сплайн-функции

В методе трапеции для интегрирования функций применялась идея кусочно-линейной аппроксимации функции $f(x)$. С точки зрения теории сплайн-функций кусочно-линейная функция является линейным сплайном. С этих позиций метод трапеции можно интерпретировать, как приближение исходной функции $f(x)$ линейным сплайном $S_1(x)$, построенным на интерполяционной сетке $x_0 < x_1 < \ldots < x_n$. Процедура взятия определенного интеграла от функции $f(x)$ заменяется интегрированием $S_1(x)$:

$$J \approx \int_a^b S_1(x)dx \ .$$

В методе Симпсона (значительно более эффективного, чем правило трапеций) на двойных отрезках функция $f(x)$ заменялась параболой. Однако построенная таким образом кусочно-полиномиальная функция не является параболическим сплайном по следующим причинам:

1) нет гладкого сопряжения кусков парабол между собой;

2) каждая парабола метода Симпсона распространяется на два интервала интерполирования функции.

Естественно предположить, что на той же интерполяционной сетке $x_0 < x_1 < \ldots < x_{2n}$, которая используется для метода Симпсона, параболический сплайн $S_2(x)$ приближает исходную функцию $f(x)$ с более высокой точностью. Тогда приближенное вычисление интеграла по формуле

$$J \approx \int_a^b S_2(x)dx \qquad (5.21)$$

имеет меньшую величину ошибки ограничения $R(h)$, чем для метода Симпсона. Для вычисления интеграла (5.21) удобно использовать следующую форму представления сплайна

$$S_2(x) = \begin{cases} a_0x^2 + b_0x + c_0, & x \le x_1, \\ a_1x^2 + b_1x + c_1, & x_1 \le x \le x_2, \\ \ldots & \ldots \\ a_kx^2 + b_kx + c_k, & x_k < x \le x_{k+1}, \\ \ldots & \ldots \\ a_{2n-1}x^2 + b_{2n-1}x + c_{2n-1}, & x > x_{2n-1}. \end{cases} \qquad (5.22)$$

Частичный интеграл J_k можно вычислить на $[x_k, x_{k+1}]$ по формуле:

$$J_k = \int_{x_k}^{x_{k+1}} (a_kx^2 + b_kx + c_k)dx = \frac{a_kh^3}{3} + \frac{b_kh^2}{2} + c_kh \ ,$$

тогда

$$J \approx \sum_{k=0}^{2n-1} J_k = \frac{h^3}{3}[a_0 + a_1 + \ldots + a_{2n-1}] + \frac{h^2}{2}[b_0 + b_1 + \ldots + b_{2n-1}] + h(c_0 + c_1 + \ldots + c_{2n-1}$$

(5.23)

5.6. Численные методы дифференцирования функций

При решении практических задач часто нужно найти производные указанных порядков от функции $y=f(x)$. Возможно, что в силу сложности аналитического выражения функции $f(x)$ непосредственное её дифференцирование затруднено. В этих случаях обычно используют приближённые численные методы дифференцирования функций.

Идея всех методов численного дифференцирования функций сводится к замене исходной функции $f(x)$ некоторой функцией $P(x)$, её интерполирующей (чаще всего полиномом или сплайном). Затем полагают:

$$f'(x) \approx P'(x) \tag{5.24}$$

при $x \in [a,b]$.

Если для интерполирующей функции известна погрешность $R(x)=f(x)-P(x)$, то погрешность вычисления производной функции $f(x)$ может быть вычислена по формуле

$$r(x)=f'(x)-P'(x)=R'(x). \tag{5.25}$$

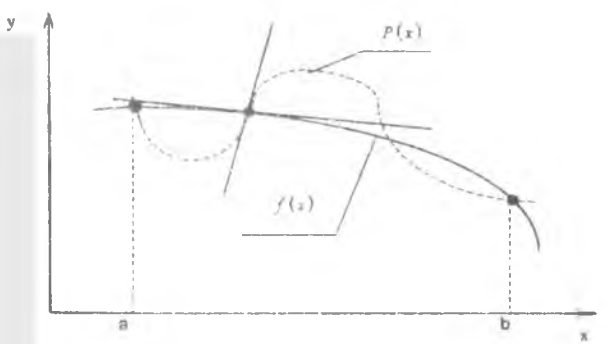


Рис.5.7. Погрешность вычисления производной функции

Следует отметить, что численное дифференцирование представляет собой операцию менее точную, чем интегрирование. Близость друг к другу ординат двух кривых $y=f(x)$ и $y=P(x)$ на $[a,b]$ еще не гарантирует близость на этом отрезке их производных,

то есть малого расхождения угловых коэффициентов касательных к графикам рассматриваемых кривых (рис.5.7.).

Использование сплайнов со специально выбранными граничными условиями, уменьшающими осцилляцию сплайна между узлами интерполяции, во многих случаях может существенно повысить точность вычисления производной функции.

5.7. Использование первой интерполяционной формулы Ньютона для вычисления производных функции

Пусть имеем функцию $y = f(x)$, заданную в равноотстоящих точках $x_i (i = 0,1,...n)$ $h = x_{i+1} - x_i$. Введем переменную $q = \frac{x - x_0}{h}$, тогда интерполяционная формула Ньютона примет вид

$$y(x) = y_0 + q\Delta y_0 + \frac{q(q-1)}{2!}\Delta^2 y_0 + ... + \frac{q(q-1)...(q-n+1)}{n!}\Delta^n y_0 \quad (5.26)$$

или

$$y(x) = y_0 + q\Delta y_0 + \frac{q^2 - q}{2!}\Delta^2 y_0 + \frac{q^3 - 3q^2 + 2q}{3!}\Delta^3 y_0 + ... \quad (5.27)$$

Учитывая $\frac{\partial y}{\partial x} = \frac{\partial y}{\partial q} = \frac{\partial q}{\partial x} = \frac{1}{h}\frac{\partial y}{\partial q}$ из (5.27) имеем

$$y'(x) = \frac{1}{h}\left[\Delta y_0 + \frac{2q-1}{2}\Delta^2 y_0 + \frac{3q^2 - 6q + 2}{6}\Delta^3 y_0 + ...\right]. \quad (5.28)$$

Для вычисления второй производной, дифференцируя (5.28), получим

$$y''(x) = \frac{1}{h^2}\left[\Delta^2 y_0 + (q-1)\Delta^3 y_0 + \frac{6q^2 - 18q + 11}{12}\Delta^4 y_0 + ...\right]. \quad (5.29)$$

Аналогично можно получить формулы для вычисления производных более высоких порядков.

Если производная функции вычисляется в точке x_0, то, учитывая, что q=0, имеем следующие формулы вычисления дифференциалов функции $f(x)$:

$$y'(x_0)=\frac{1}{h}\left[\Delta y_0-\frac{\Delta^2 y_0}{2}+\frac{\Delta^3 y_0}{3}+\dots\right],$$
$$y''(x_0)=\frac{1}{h^2}\left[\Delta^2 y_0-\Delta^3 y_0+\frac{11}{12}\Delta^4 y_0+\dots\right]. \qquad (5.30)$$

Оценка погрешности вычисления производных функции

В п.5.6 было показано, что: $r(x)=f'(x)-P'(x)=R'(x)$. Для формулы Ньютона имеем

$$R_n(x)=\frac{q(q-1)\dots(q-n)}{(n+1)!}\Delta^{n+1}y_0,$$

тогда $R_n'(x)=\frac{1}{h}\left[\frac{\partial}{\partial q}\left[q(q-1)\dots(q-n)\right]\Delta^{n+1}y_0\right]\frac{1}{(n+1)!}$,

а при $x=x_0$

$$R_n'(x_0)=\frac{1}{h}(-1)^n\frac{\Delta^{n+1}y_0}{(n+1)}. \qquad (5.31)$$

Аналогично может быть найдена погрешность $R_n''(x_0)$, возникающая при вычислении второй производной функции $f(x)$, и т.д.

5.8. Вычисление частных производных

Вычисление частных производных функции рассмотрим на примере функции двух переменных $Z=f(x,y)$. В общем случае для вычисления частных производных можно воспользоваться идеей, изложенной в п.5.6. На первом этапе интерполировать функцию $f(x,y)$ некоторой другой функцией (например, полиномом или сплайном) $P(x,y)$, а затем считать, что любая частная производная функции $f(x,y)$ приближенно совпадает с частной производной функции $P(x,y)$. Таким способом следует пользоваться, когда требуется произвести разовую операцию определения частной производной с высокой степенью точности. Во многих приложениях, например, в итерационных методах решения системы нелинейных уравнений методом Ньютона, частные производные вычисляются многократно. Здесь на первое

место выходят вопросы быстродействия методов вычисления производных. Наибольшее распространение получили следующие простейшие конечно-разностные методы вычисления частных производных:

$$\frac{\partial z}{\partial x} \approx \frac{f(x_0+h, y_0)-f(x_0, y_0)}{h}$$

$$\frac{\partial z}{\partial y} \approx \frac{f(x_0, y_0+k)-f(x_0, y_0)}{k}$$

$$\frac{\partial^2 z}{\partial x^2} \approx \frac{f(x_0+h, y_0)-2f(x_0, y_0)+f(x_0-h, y_0)}{h^2}$$

$$\frac{\partial^2 z}{\partial y^2} \approx \frac{f(x_0, y_0+k)-2f(x_0, y_0)+f(x_0, y_0-k)}{k^2}$$

$$\frac{\partial^2 z}{\partial x \partial y} \approx \frac{f(x_0+h, y_0+k)-f(x_0, y_0+k)+f(x_0+h, y_0)-f(x_0, y_0)}{hk}$$

ГЛАВА 6

Численные методы решения обыкновенных дифференциальных уравнений

6.1. Введение

В настоящее время большинство математических моделей сложных технических систем описывается системой обыкновенных дифференциальных уравнений. Для разных постановок задач разработано огромное количество численных методов решения дифференциальных уравнений. Укрупнённо их можно разбить на две большие группы методов: конечно-разностные методы (методы сеток) и методы конечного элемента. Учитывая теоретическую сложность второй группы методов, в данной главе будут рассмотрены конечно-разностные схемы решения дифференциальных уравнений применительно к уравнениям, содержащим первую производную от функции одной переменной. Несмотря на явное ограничение класса решаемых задач, идеи численных методов, рассмотренные в этой главе, полностью или с небольшими изменениями пригодны для решения многих и более сложных дифференциальных уравнений. Для более глубокого изучения методов решения дифференциальных уравнений необходимо обратиться к специальной литературе.

Рассмотрим обыкновенное дифференциальное уравнение первого порядка, заданное в нормальной форме Коши:

$$\frac{dy}{dx} = f(x, y), \tag{6.1}$$

с начальными условиями

$$y_0 = y(x_0). \tag{6.2}$$

Функцию $y^*(x)$, удовлетворяющую начальным условиям (6.2) и обращающую в тождество равенство (6.1), будем называть решением уравнения (6.1) при начальных условиях (6.2).

Например, для уравнения $\frac{dy}{dx} = y(x)$ решение с точностью до постоянного множителя описывается функцией $y(x) = ae^x$.

Такая функция описывает целое семейство интегральных кривых решения исходного уравнения (см. рис. 6.1.). Задание начального приближения позволяет конкретизировать выбор единственного решения (интегральной кривой, проходящей через

точку (x_0, y_0)).

Существует множество приёмов для нахождения решения дифференциальных уравнений через элементарные или специальные функции. Однако очень часто в практических задачах классические методы решения дифференциальных уравнений либо вообще неприменимы, либо приводят к таким сложным решениям, что затраты труда на их получение превосходят все допустимые пределы.

Например, внешне простое уравнение $\frac{dy}{dx} = x^2 + y^2$ не имеет элементарного решения.

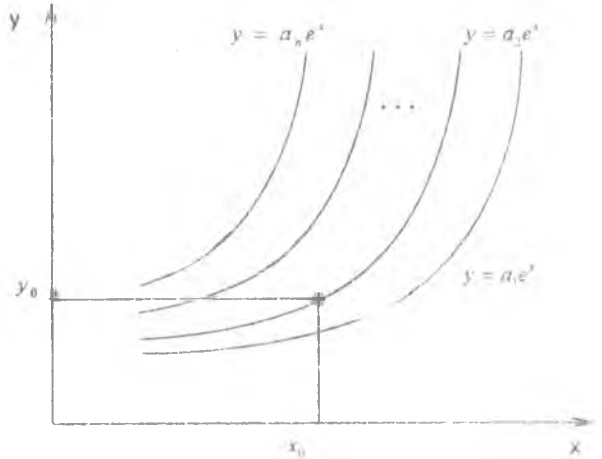


Рис. 6.1. Семейство интегральных кривых решения уравнения

Процесс нахождения решения дифференциального уравнения (6.1) часто называют интегрированием уравнения. К сожалению, на ЭВМ невозможно найти решение уравнения в виде функции $y^*(x)$ непрерывного аргумента, поэтому на начальном этапе реализации метода производится дискретизация пространства независимых переменных (в нашем случае переменной x). Для этого на интересующем нас интервале интегрирования выделяется конечное число точек $x_0, x_1, x_2, ..x_n$. Интервал между соседними точками называют шагом интегрирования $h_m = (x_{m+1} - x_m)$. Вместо

непрерывного решения $y^*(x)$ ищется её дискретное приближение (табличная функция), то есть функция $y_i = y(x_i)$, заданная своими значениями в узлах x_i, $i = 0,1,...n$.

Большинство численных методов интегрирования дифференциального уравнения строится по следующей формуле:

$$y_{m+1} = g\left(y_m, y_{m-1}, \ldots y_{m-p+1}, y'_{m+1}, \ldots y'_{m-p+1}\right), \qquad (6.3)$$

где g - некоторая функция, зависящая от численного способа интегрирования дифференциального уравнения; p - количество предыдущих точек, которые используются в формуле интегрирования.

Основными характеристиками численных методов решения дифференциальных уравнений, от которых зависит их эффективность, являются **точность** и **устойчивость** методов.

Точность интегрирования или точность решения дифференциального уравнения можно оценить, проанализировав ошибки, возникающие на каждом шаге интегрирования.

Основными составляющими полной ошибки интегрирования являются:

1. *ошибки ограничения* (или *ошибки метода*) - погрешности, вызванные заменой в дифференциальном уравнении производных функции их конечно-разностными аналогами. Эта погрешность возникает в связи с заменой функции непрерывного аргумента её дискретным аналогом;
2. *ошибки вычислений* - e_m^b, связанные с ошибками округления чисел в ЭВМ;
3. *ошибки накопления* - e_m^H, связанные с увеличением полной ошибки интегрирования предыдущего шага ввиду того, что на шаге m для вычисления значения y_m используется не точное значение $y(x_{m-1})$, а приближённое y_{m-1}.

Практика расчёта разных уравнений на ЭВМ показывает, что ошибками вычислений - e_m^b, даже в расчётах с обычной точностью, можно пренебречь.

Для качественного анализа ошибок ограничения сравнивают формулу интегрирования (6.3) используемого метода с разложением в ряд Тейлора в окрестностях точки x_m точного решения дифференциального уравнения при $h_m \to 0$.

Ошибку можно определить путём оценки суммы оставшихся

членов ряда Тейлора. Число совпадений формулы интегрирования с первыми членами ряда Тейлора определяет порядок точности метода.

Анализ накопленной ошибки вводит важное для дифференциальных уравнений понятие устойчивости метода. Устойчивость связана с характером изменения накопленной ошибки. Если e_m^H в ходе интегрирования уравнения не возрастает с увеличением шагов, то используемый метод называется численно устойчивым. Если даже при небольших ошибках метода или вычислений ошибка накопления e_m^H растёт шаг от шага, то метод интегрирования будет неустойчивым при выбранной величине шага интегрирования и результаты таких вычислений бесполезны.

6.2. Решение с помощью рядов Тейлора

Рассмотрим метод решения дифференциального уравнения, представляющего ценность, поскольку он помогает во многих случаях оценить порядок точности для практически значимых методов интегрирования дифференциальных уравнений.

Пусть ранее было уже сделано m шагов интегрирования дифференциального уравнения. В точке x_m найдено решение y_m. Функцию $y(x)$ в окрестностях точки x_m разложим в ряд Тейлора:

$$y(x) = y_m + y'_m(x - x_m) + \frac{1}{2}y''_m(x - x_m)^2 + \frac{1}{6}y'''_m(x - x_m)^3 + \ldots, \qquad (6.4)$$

где $x_m = x_0 + mh$; h - шаг интегрирования. Тогда

$$y_{m+1} = y(x_0 + (m+1)h) = y_m + y'_m h + \frac{1}{2}y''_m h^2 + \frac{1}{6}y'''_m h^3 \qquad (6.5)$$

Из уравнения (6.1) имеем $y'_m = f(x_m, y_m)$, откуда

$$y''_m = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y}\frac{dy}{dx} = f'_x + ff'_y. \qquad (6.7)$$

Учитывая (6.7), для выражения (6.5) имеем

$$y_{m+1} = y_m + y'_m h + \frac{h^2}{2}(f'_x + ff'_y) + O(h^3)$$

или

$$y_{m+1} = y_m + h\left(f + \frac{h}{2}(f'_x + ff'_y) + O(h^3)\right), \qquad (6.8)$$

где $O(h^3)$ - остаточный член, означающий, что в следующие члены ряда h входит в степени не ниже третьей. Иначе говоря , если

для нахождения приближённого решения уравнения (6.1) будет использована формула (6.8) без $O(h^3)$, то ошибка ограничения будет приблизительно равна Kh^3, где K - некоторая постоянная.

Для практического использования этот метод малопригоден, поскольку очень трудно (иногда невозможно) найти частные производные f'_x и f'_y. С другой стороны, именно в этом методе удаётся явно оценить порядок точности метода.

6.3. Методы Рунге - Кутта

Большую группу численных методов решения дифференциальных уравнений образуют методы Рунге-Кутта. Методы Рунге-Кутта обладают следующими отличительными свойствами:

1. Эти методы являются одноступенчатыми: чтобы найти y_{m+1}, нужна информация только о предыдущей точке (x_m, y_m).
2. Они имеют высокий порядок точности.
3. Они не требуют вычислений производных от $f(x,y)$, а только вычисления самой функции.

Метод Эйлера

Наиболее простым представителем этой группы методов является метод Эйлера решения дифференциального уравнения.

Идея метода заключается в следующем. Зная начальное приближение $y_0 = y(x_0)$, т.е. точку (x_0, y_0), лежащую на искомой интегральной кривой, а также функцию $f(x,y)$ дифференциального уравнения, мы можем на первом шаге определить угол наклона касательной к кривой $y^*(x)$ $\quad y'_0 = f(x_0, y_0)$.

Для небольшой величины шага h можно допустить, что следующая точка решения уравнения y_1 лежит на касательной прямой:

$$y_1 = y_0 + y'_0 h = y_0 + hf(x_0, y_0) \qquad (6.9)$$

На следующем шаге интегрирования вычисляется новое значение функции y_2 в точке $x_2 = x_1 + h$, лежащей на касательной, проведённой под углом $y'_1 = f(x_1, y_1)$, и т.д. В итоге, интегральная кривая $y^*(x)$ заменяется ломаной (см. рис. 6.2). Формула интегрирования (6.3) для метода Эйлера имеет вид

$$y_{m+1} = y_m + hf(x_m, y_m), \qquad (6.10)$$

где $h = x_{m+1} - x_m$ - постоянный шаг интегрирования, а значение начальной точки берётся из начальных условий дифференциального уравнения (6.2).

Если сравнить формулу (6.10) и формулу интегрирования с использованием рядов Тейлора, то видно, что в формуле (6.10) согласуются только два первых члена ряда, а остаточный член пропорционален h, из чего следует, что метод Эйлера имеет первый порядок точности.

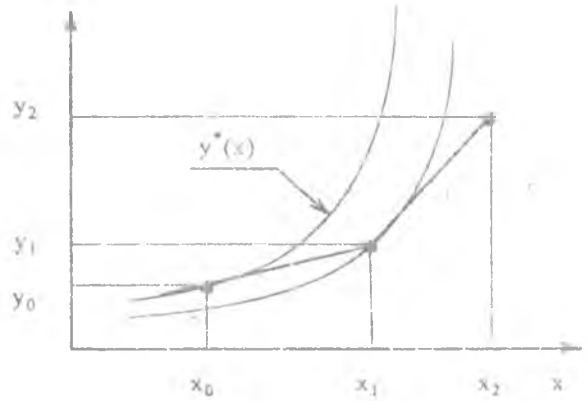


Рис. 6.2 Графическое представление метода Эйлера

Исправленный метод Эйлера

Метод Эйлера имеет довольно большую ошибку ограничения; кроме того, он очень часто оказывается неустойчивым, поэтому на практике используют более совершенные методы. Однако метод Эйлера можно значительно усовершенствовать.

Пусть на некотором шаге интегрирования получена точка (x_m, y_m). Для определения нового значения функции y_{m+1} в точке $x_{m+1} = x_m + h$ поступим следующим образом:

1. На предварительном этапе в направлении к касательной L_1 (рис. 6.3), проведённой в точке (x_m, y_m), сделаем шаг длины h, чем найдём новую точку $(x_m + h, y_m + hy'_m)$, как это делается в методе Эйлера.

2. В точке $(x_m + h, y_m + hy'_m)$ найдём новое положение касательной

L_2 . Новое положение касательной указывает направление "закругления" интегральной кривой. Усреднение двух тангенсов углов наклона прямых L_1 и L_2 даёт прямую $\bar{L}$.

3. При малых значениях h касательная L_1 всегда отклоняется от искомой интегральной кривой $y^*(x)$, скорректированное направление секущей, проведённой из точки (x_m, y_m) в направлении прямой, параллельной $\bar{L}$, может уточнить искомое значение функции:

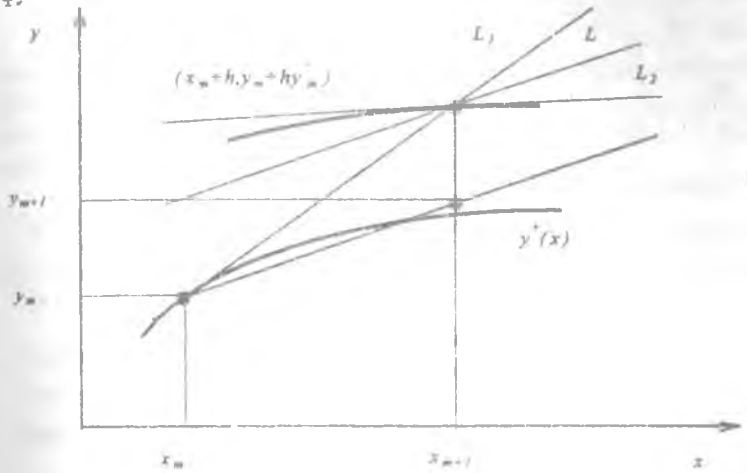


Рис. 6.3 Геометрическое представление исправленного метода Эйлера

$$y_{m+1} = y_m + \frac{h}{2}\left[f(x_m, y_m) + f(x_m + h, y_m + hy'_m)\right]. \qquad (6.11)$$

Формула(6.11)описывает исправленный метод Эйлера.

Для того, чтобы выяснить, насколько хорошо этот метод согласуется с разложением в ряд Тейлора, разложим в ряд Тейлора функцию $f(x, y(x))$ в окрестностях точки (x_m, y_m):

$$f(x,y) = f(x_m, y_m) + \frac{\partial f}{\partial x}(x - x_m) + \frac{\partial f}{\partial y}(y - y_m) + \ldots$$

или $$f(x_m + h, y_m + hy'_m) = f + hf'_x + hff'_y + O(h^3). \qquad (6.12)$$

Подставив (6.12) в(6.11)имеем

$$y_{m+1} = y_m + hf + \frac{h^2}{2}\left(f_x + ff'_y\right) + O(h^3). \qquad (6.13)$$

Как видно из (6.13), исправленный метод Эйлера согласуется с рядом Тейлора вплоть до членов степени h^2, являясь, таким образом, методом Рунге-Кутта второго порядка.

Модифицированный метод Эйлера

В исправленном методе Эйлера для корректировки направления касательной усреднялись наклоны двух прямых L_1 и L_2. Другой способ модификации метода Эйлера сводится к следующему. Пусть мы на предварительном этапе находим точку $\left(x_m + \frac{h}{2}, y_m + \frac{h}{2}y'_m\right)$, лежащую на касательной L_1 на половинном шаге интегрирования $\frac{h}{2}$. Новое значение функции y_{m+1} в точке x_{m+1} будем иметь в направлении прямой, параллельной к прямой L_2 (рис.6.4):

$$y_{m+1} = y_m + hf\left(x_m + \frac{h}{2}, y_m + \frac{h}{2}y'_m\right). \qquad (6.14)$$

Модифицированный метод Эйлера согласуется с разложением в ряд Тейлора вплоть до членов степени h^2, то есть величина остаточного члена пропорциональна kh^3 и этот метод является ещё одним методом Рунге-Кутта второго порядка.

Метод Рунге-Кутта четвёртого порядка

Метод Рунге-Кутта четвёртого порядка применяется на практике настолько широко, что в литературе он получил название "метода Рунге-Кутта" без указания порядка метода.

Этот классический метод описывается системой следующих соотношений:

$$y_{m+1} = y_m + \frac{h}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right), \qquad (6.15)$$

где

$$k_1 = f(x_m, y_m),$$

$$k_2 = f\left(x_m + \frac{h}{2}, y_m + \frac{hk_1}{2}\right),$$

$$k_3 = f\left(x_m + \frac{h}{2}, y_m + \frac{hk_2}{2}\right),$$

$$k_4 = f\left(x_m + h, y_m + hk_3\right).$$

Ошибка ограничения для этого метода пропорциональна kh^5.

При использовании этого метода функцию $f(x,y)$ приходится вычислять 4 раза. При выборе шага интегрирования h для достижения заданной точности решения дифференциального уравнения достаточно часто оказывается полезным грубое оценочное правило, если

$$\left|\frac{k_2 - k_3}{k_1 - k_2}\right| > 0{,}03,$$

то шаг интегрирования следует уменьшить.

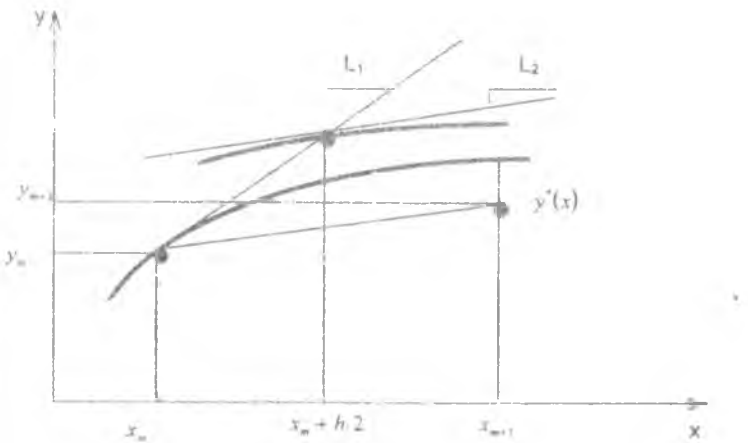


Рис. 6.4 Геометрическое представление модифицированного метода Эйлера

6.4. Метод прогноза и коррекции

Отличительной чертой методов Рунге-Кутта является то, что

при вычислении следующей точки (x_{m+1}, y_{m+1}) используется информация только о точке (x_m, y_m), но не о предшествующих точках из числа полученных . Это положение представляется нерациональным, поскольку информация о поведении функции $y^*(x)$ на предшествующих шагах $y_0, y_1, \ldots y_{m+1}$ может помочь более точно вычислить значение функции на следующем шаге интегрирования y_{m+1}. Метод прогноза и коррекции относится к группе методов решения дифференциальных уравнений, в котором используется "предыстория" процесса интегрирования дифференциального уравнения.

Каждый очередной шаг интегрирования в таких методах производится в два этапа. На первом этапе на основании информации о поведении функции $y^*(x)$ на предшествующих шагах интегрирования "предсказывается" значение y_{m+1}. На втором этапе организуется итерационный процесс корректировки предсказанного значения y^l_{m+1} до достижения требуемого уровня точности вычисления y_{m+1}. Первый этап работы таких методов называют "прогнозом", второй "коррекцией".

Для запуска метода прогноза и коррекции кроме начальных условий $y_0 = y^*(x_0)$ требуется знать решение дифференциального уравнения ещё в нескольких точках (в нашем случае в одной точке) $y_1 = y^*(x_1)$. Собственными силами метод прогноза и коррекции получить такую дополнительную информацию неспособен. Поэтому на предварительном этапе для "раскрутки" метода тем или иным методом (обычно методом Рунге-Кутта 4-го порядка) вычисляют значение функции из начального приближения ещё в нескольких точках.

Пусть мы имеем следующую информацию:

(x_0, y_0) - начальное условие дифференциального уравнения;

(x_1, y_1) - дополнительное условие, полученное, например, методом Рунге-Кутта.

Далее процесс интегрирования дифференциального уравнения развивался и достиг точки (x_m, y_m). Рассмотрим методику вычисления значения y_{m+1} на $(m+1)$ шаге интегрирования

Этап прогноза.

Этап прогноза заключается в нахождении начального приближения $y_{m+1}^{(0)}$ значения y_{m+1}, которое вычисляется по предшествующим двум точкам (x_{m-1}, y_{m-1}) и (x_m, y_m) модифицированным методом Эйлера:

$$y_{m+1}^{(0)} = y_{m-1} + 2hf(x_m, y_m). \tag{6.16}$$

Этап коррекции

При реализации этапа коррекции в методе прогноза и коррекции используется итерационная формула, заимствованная из исправленного метода Эйлера:

$$y_{m+1}^{(i)} = y_m + \frac{h}{2}\left[f(x_m, y_m) + f\left(x_{m+1}, y_{m+1}^{(i-1)}\right)\right], \tag{6.17}$$

где в качестве начального приближения берётся значение $y_{m+1}^{(0)}$, полученное при выполнении этапа прогноза. Процесс коррекции продолжается до тех пор, пока не выполнится условие

$$\left|y_{m+1}^{(i+1)} - y_{m+1}^{(i)}\right| < \varepsilon, \tag{6.18}$$

где ε- заданная точность решения уравнения.

Сходимость итерационного процесса

Применив теорему о среднем к формуле

$$y_{m+1}^{(i+1)} - y_{m+1}^{(i)} = \frac{h}{2}\left[f\left(x_m, y_{m+1}^{(i)}\right) - f\left(x_m, y_{m+1}^{(i-1)}\right)\right],$$

получим

$$y_{m+1}^{(i+1)} - y_{m+1}^{(i)} = \frac{h}{2} f_y'(x_m, y)\left[y_{m+1}^{(i)} - y_{m+1}^{(i-1)}\right], \tag{6.19}$$

где y лежит между $y_{m+1}^{(i)}$ и $y_{m+1}^{(i-1)}$.

Пусть $\left|f_y'(x_m, y)\right| \le M$, т.е. функция $f(x,y)$ имеет ограниченную первую производную, тогда

$$\left|y_{m+1}^{(i+1)} - y_{m+1}^{(i)}\right| \le \frac{hM}{2}\left|y_{m+1}^{(i)} - y_{m+1}^{(i+1)}\right|. \tag{6.20}$$

Из неравенства (6.20) видно, что итерационный этап коррекции сходится, если $\frac{hM}{2} < 1$ или

$$h<\frac{2}{M}. \qquad (6.21)$$

Необходимо чётко представлять себе, что мы доказали в данном случае, что итерационный процесс сходится к некоторому определённому значению, но вовсе не обязательно к точному решению уравнения.

Разница между точным решением и значением, к которому сходится итерационная формула (6.17), образует ошибку ограничения метода прогноза и коррекции. Величину этой ошибки можно оценить по формуле

$$e_{\text{огр}} = \frac{1}{5}\left[y_m^{(0)} - y_m^{(i)}\right] = -\frac{h^3}{12}y''' . \qquad (6.22)$$

Устойчивость метода прогноза и коррекции

Свойство устойчивости характеризует отсутствие зависимости ошибок, полученных на текущем шаге численного метода, от ошибок предшествующих шагов работы алгоритма и имеет место в тех случаях, когда ошибка не накапливается. Исследуем устойчивость для метода прогноза и коррекции.

Из (6.17) имеем

$$y_{m+1} = y_m + \frac{h}{2}[f(x_m, y_m) + f(x_{m+1}, y_{m+1})] .$$

Пусть известно точное значение функции Y_m в точке x_m, тогда

$$Y_{m+1} = Y_m + \frac{h}{2}[f(x_m, Y_m) + f(x_{m+1}, Y_{m+1})] + e_m ,$$

где e_m - методическая ошибка.

Полную ошибку на m-м шаге алгоритма можно вычислить по формуле $\varepsilon_m = Y_m - y_m$. Вычитая первое уравнение из второго, имеем

$$\varepsilon_{m+1} = \varepsilon_m + \frac{h}{2}\{[f(x_m, Y_m) + f(x_m, y_m)] + [f(x_{m+1}, Y_{m+1}) + f(x_{m+1}, y_{m+1})]\} + e_m .$$

Из последнего уравнения, используя теорему о среднем, получим

$$\varepsilon_{m+1} = \varepsilon_m + \frac{h}{2}[f_y'(x_{m+1}, \xi_1)\varepsilon_{m+1} + f_y'(x_m, \xi_2)\varepsilon_m] + e_m ,$$

откуда после преобразований имеем

$\varepsilon_{m+1} = \mu\varepsilon_m + \delta$, где

$$\mu = (1 + \frac{hf_y'}{2}) / (1 - \frac{hf_y'}{2}) \quad \text{и} \quad \delta = e_m / (1 - \frac{hf_y'}{2}) .$$

На первом шаге алгоритма погрешность может быть вычислена

по формуле

$$\varepsilon_1 = \mu\varepsilon_0 + \delta.$$

На последующих:

$$\varepsilon_2 = \mu(\mu\varepsilon_0 + \delta) + \delta = \mu^2\varepsilon_0 + \mu\delta + \delta,$$

$$\varepsilon_3 = \mu^3\varepsilon_0 + \mu^2\delta + \mu\delta + \delta,$$

...

$$\varepsilon_n = \mu^n\varepsilon_0 + \delta(1 + \mu + \mu^2 + \cdots + \mu^{n-1}).$$

Если положить $|\mu| < 1$, то, учитывая формулу для сходящейся геометрической прогрессии, получим

$$\lim_{n\to\infty} \varepsilon_n = \lim_{n\to\infty} (\mu^n\varepsilon_0 + \delta\frac{1}{1-\mu}) = \delta\frac{1}{1-\mu}.$$

Из чего следует, что ошибки метода прогноза и коррекции по мере его работы не накапливаются и ограничены «сверху» величиной $\varepsilon\frac{1}{1-\mu}$, если $|\mu| < 1$. Последнее возможно, если выполняется условие сходимости метода $\left|\frac{hf'_y}{2}\right| < 1$ и производная правой части дифференциального уравнения по функции меньше нуля ($f'_y < 0$). В этом случае $0 < \mu < 1$ и метод прогноза и коррекции обладает свойством абсолютной устойчивости. Можно показать, что метод прогноза и коррекции имеет относительную устойчивость и при $f'_y > 0$

ГЛАВА 7

Методы численной оптимизации

7.1. Введение

Проблема оптимизации встречается в любой сфере человеческой деятельности. Экономическое планирование, управление, распределение ограниченных ресурсов, анализ производственных процессов, проектирование сложных объектов всегда должно быть направлено на поиск наилучшего варианта решения с точки зрения намеченной цели. При большом разнообразии задач оптимизации только математика может дать общие методы их решения. Однако для того, чтобы воспользоваться математическим аппаратом, необходимо сформулировать интересующую нас проблему как математическую задачу, придав количественные оценки возможным вариантам, количественный смысл слова "лучше", "хуже". В основе подобных правил предпочтения лежит целевая функция (критерий оценки качества), количественно выражающая качество объекта. Формирование целевой функции всегда выполняется с учётом различных выходных параметров проектируемого изделия, изменяющихся в зависимости от входных параметров. Тогда задача выбора наилучшего варианта изделия сводится к максимизации или минимизации целевой функции в зависимости от смысловой нагрузки критерия качества.

Таким образом, поиск рационального технического решения сводится к отысканию экстремума (максимума или минимума) некоторой целевой функции $f(x_1, x_2, \ldots x_n)$, где $x_1, x_2, \ldots x_n$ - независимые переменные (входные, варьируемые параметры изделия). Такие задачи часто ещё называют задачами *параметрической оптимизации*.

Задачи параметрической оптимизации можно разбить на два больших класса: задачи *безусловной и условной оптимизации*.

Задача безусловной оптимизации ставится как задача отыскания такого сочетания оптимизируемых параметров $x^* = \left(x_1^*, x_2^*, \ldots x_n^*\right)^T$, которые доставляют минимум целевой функции:

$$\min_x f(x), \qquad (7.1)$$

Для определённости будем считать, что нашей целью является отыскание минимума целевой функции. В противном случае, простая замена целевой функции $f(x) = -\varphi(x)$ преобразует задачу отыскания максимума целевой функции $\varphi(x)\left(\max_x \varphi(x)\right)$ к задаче (7.1).

Если на пространство оптимизируемых переменных наложены ограничения, то задача безусловной оптимизации превращается в задачу условной оптимизации (или задачу оптимизации с ограничениями):

$$\min_x f(x),$$
$$g_i(x) \le 0, \quad i = 1,2,...m \ , \tag{7.2}$$

где $g_i(x)$- некоторые функции, связывающие оптимизируемые параметры. В зависимости от свойств целевой функции $f(x)$ и ограничений $g_i(x)$ методы поиска экстремума можно разбить на большое количество классов.

Если $f(x)$ зависит от одного параметра и является унимодальной функцией, то рассматривают методы одномерной оптимизации. Если функция $f(x)$- линейна вместе со своими ограничениями $g_i(x)$, то такую задачу условной оптимизации называют задачей линейного программирования.

Известны также разделы математического программирования, которые рассматривают частные случаи постановок задач оптимизации. К таким методам относятся методы выпуклого, геометрического и т.п. программирования.

В зависимости от типа искомого экстремума различают методы локальной оптимизации, условной глобальной или безусловной оптимизации.

Широта спектра только постановок задач оптимизации указывает на огромное количество разнообразных численных методов их решения. В данном пособии остановимся лишь на наиболее ярких представителях численных методов оптимизации, предоставив читателю найти в обширной специальной литературе возможность более подробную информацию.

Наиболее многочисленную группу составляют методы безусловной оптимизации. Некоторое представление о широко применяемых методах этой группы даёт рис. 7.1.

В методах нулевого порядка (прямых методах) информация о производных целевой функции не используется. Для методов первого порядка необходимо вычислять как значение функции качества, так и её первые частные производные. В методах второго порядка организация поиска экстремума ведётся с учётом значений целевой функции, её первых и вторых производных.

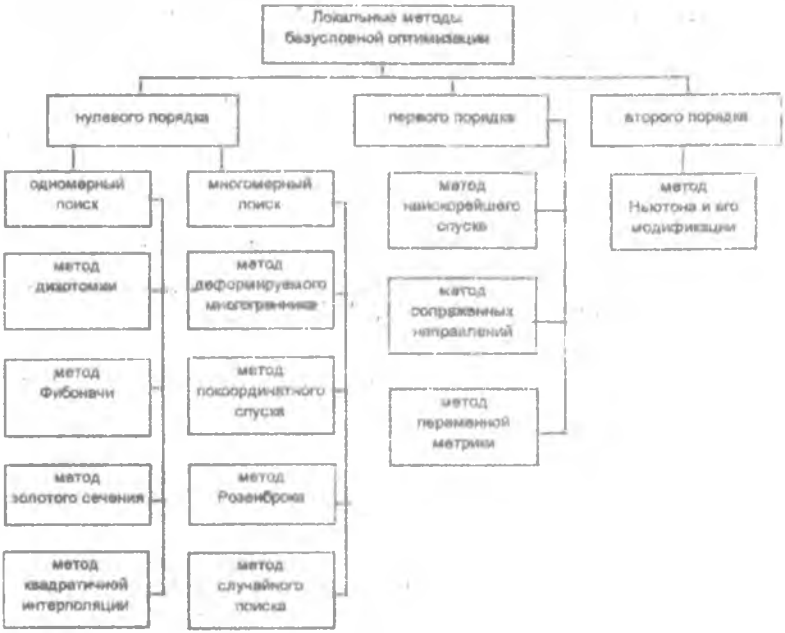


Рис. 7.1. Классы методов безусловной оптимизации

7.2. Метод золотого сечения

Метод золотого сечения относится к классу методов одномерной оптимизации и является одним из наиболее эффективных методов.

Пусть $f(x)$ – унимодальная функция одного аргумента на $[a,b]$. К функции $f(x)$ не предъявляются требования дифференцируемости или непрерывности. Предполагается, что для любого $x \in [a,b]$ значение $f(x)$ может быть вычислено.

Для реализации метода золотого сечения строится следующий итерационный процесс.

Обозначим a_k - левую границу интервала неопределённости, а через b_k - правую границу интервала неопределённости на k – м шаге итерационного процесса. Интервалом неопределённости будем называть отрезок числовой оси, на котором находится экстремум целевой функции $f(x)$. Первоначально предполагается, что интервал неопределённости совпадает с $[a,b]$, т.е. $a_0 = a, b_0 = b$. Идея метода золотого сечения заключается в организации процедуры последовательного деления интервала неопределённости с использованием решающего правила, позволяющего выделить новый, уменьшенный интервал неопределённости.

Пусть к настоящему моменту уже сделано k шагов метода золотого сечения. Интервал неопределённости ограничен отрезком $[a_k, b_k]$. На этом отрезке в точках

$$x_1^{(k)} = a_k + 0{,}382(b_k - a_k),$$
$$x_2^{(k)} = b_k - 0{,}382(b_k - a_k) \qquad (7.3)$$

вычислены значения целевой функции $y_1 = f\left(x_1^{(k)}\right)$ и $y_2 = f\left(x_2^{(k)}\right)$.

На рис. 7.2. показаны три возможных варианта соотношений y_1 и y_2.

1. Если $y_1 < y_2$, то искомое значение минимума функции $f(x)$, очевидно, находится на отрезке $[a_k, x_2]$ и, следовательно, интервал неопределённости $[a_k, b_k]$ уменьшается до $[a_k, x_2]$.

Очевидно, что тогда

$$a_{k+1} = a_k, b_{k+1} = x_2^{(k)}$$

В методе золотого сечения пропорции выбраны таким образом, что точка $x_1^{(k)}$ (отстоящая от a_k на 0,382 относительные единицы длины отрезка $[a_k, b_k]$) на отрезке $[a_{k+1}, b_{k+1}]$ будет отстоять на те же 0,382 относительные единицы влево от точки b_{k+1}. Тогда

$$x_2^{(k+1)} = x_1^{(k)}$$

и для выполнения $(k+1)$-го шага метода золотого сечения нам потребуется вычислить

$$x_1^{(k+1)} = a_{k+1} + 0{,}382(b_{k+1} - a_{k+1}).$$

2. Если $y_1 > y_2$, то искомое значение минимума функции $f(x)$

лежит на отрезке $[x_1, b_k]$, в этом случае необходимо выполнить следующие действия:

$$a_{k+1} = x_1^{(k)},$$

$$b_{k+1} = b_k,$$

$$x_1^{(k+1)} = x_2^{(k)},$$

$$x_2^{(k+1)} = b_{k+1} - 0{,}382(b_{k+1} - a_{k+1}).$$

3. Равенство $y_1 = y_2$ означает, что точка минимума функции $f(x)$ расположена между $x_1^{(k)}$ и $x_2^{(k)}$, тогда

$$a_{k+1} = x_1^{(k)},$$

$$b_{k+1} = x_2^{(k)},$$

$$x_1^{(k+1)} = a_{k+1} + 0.382(b_{k+1} - a_{k+1}),$$

$$x_2^{(k+1)} = b_{k+1} - 0{,}382(b_{k+1} - a_{k+1}).$$

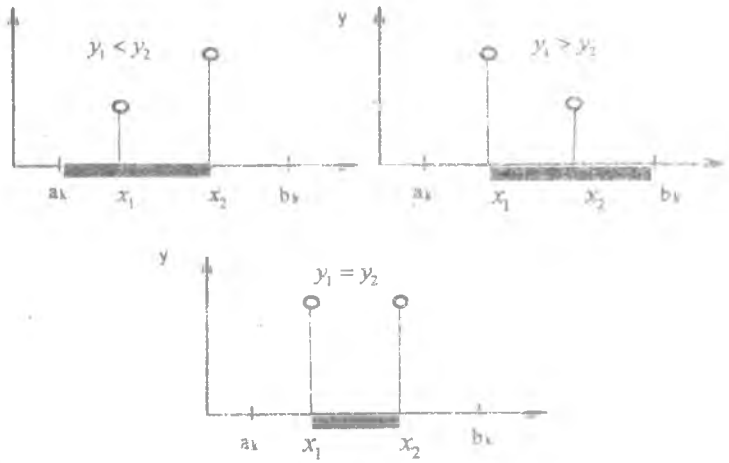


Рис. 7.2. Выбор интервала неопределённости для метода золотого сечения

Метод золотого сечения хорош тем, что для выполнения каждого шага итерации необходимо выполнить лишь одно дополнительное вычисление функции $f(x)$. При этом интервал неопределённости уменьшается на 38,2%, а в случае равенства y_1 и y_2 на 76,4%. Величина оставшегося интервала неопределённости

может служить критерием останова алгоритма оптимизации, то есть если выполняется условие $|b_{k+1} - a_{k+1}| < \varepsilon$, то можно считать, что искомый оптимум функции найден с точностью ε. Из уравнения (7.4) несложно подсчитать число шагов алгоритма, необходимого для определения $x_{\min}$ с заданной точностью

$$d0{,}382^n = \varepsilon \,, \tag{7.4}$$

где $d = b_0 - a_0$- первоначальный размер интервала неопределённости, ε- абсолютная точность оптимизации, *n*- число шагов метода золотого сечения.

7.3. Метод деформированного многогранника

Среди прямых методов многомерного поиска метод деформируемого многогранника выделяется высокой эффективностью, помехозащищённостью и чаще всего применяется на практике. Метод деформируемого многогранника Нелдера и Мида легко адаптируется к особенностям оптимизируемой функции, не "замечает" отдельные шероховатости функции (вызванные ошибками вычисления), а скорость сходимости алгоритма не слишком сильно зависит от регулярности целевой функции. Очень часто этот метод оптимизации конкурирует с такими мощными методами оптимизации, как метод Ньютона.

Метод деформируемого многогранника является модификацией симплексного метода. Симплексом называют регулярный многогранник в *n*- мерном евклидовом пространстве. Для случая 2 переменных симплекс представляет собой равносторонний треугольник; 3 переменных - тетраэдр и т.д. Для *n* -мерного пространства симплекс всегда имеет *n*+1 вершину.

Координаты вершин регулярного симплекса можно определить с помощью матрицы размером $n \times (n+1)$:

$$R = \begin{Vmatrix} 0 & r_1 & r_2 & \dots & r_2 \\ 0 & r_2 & r_1 & \dots & r_2 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & r_2 & r_2 & \dots & r_1 \end{Vmatrix}, \tag{7.5}$$

где

$$r_1 = \left[l / \left(n\sqrt{2} \right) \right] \left(\sqrt{n+1} + n - 1 \right), \tag{7.6}$$

$$r_2 = \left[l / \left(n\sqrt{2} \right) \right] \left(\sqrt{n+1} - 1 \right), \tag{7.7}$$

l - параметр, отождествляемый с расстоянием между двумя вершинами.

Элемент r_{ij} матрицы R равен i – й координате j – й вершины симплекса.

Поиск минимума функции симплексным методом ведётся следующим образом:

1. В каждой вершине симплекса вычисляется значение функции $y_i = f(x_i)$.

2. Определяется вершина с наибольшим (наихудшим) значением $f(x)$.

3. Через эту вершину и центральную точку симплекса проводится прямая, на которой на некотором удалении от центра C устанавливается новая вершина (см. рис. 7.3).

4. Вершина с наибольшим значением $F(x)$ удаляется. Симплекс, по существу, "переворачивается" через грань, противоположную наихудшей вершине.

5.Далее процесс повторяется, начиная с п.1.

Важной особенностью симплексного метода поиска является то, что для реализации каждого последующего шага итерации необходимо вычислить функцию $f(x)$ лишь в одной новой точке симплекса. Сама же оптимизация этим алгоритмом ассоциируется с процессом "кантования" симплекса вниз по поверхности функции $f(x)$ в направлении её минимума.

Регулярный метод симплексного поиска склонен к зацикливанию, поэтому Нелдер и Мид, нарушив регулярность, устранили указанный недостаток.

Обозначим X_k^A - вершину многогранника (первоначального симплекса), которая даёт максимальное значение $f(x)$ на k-- м шаге, а X_k^B - минимальную оценку функции $f(x)$. Определим вектор координат X_k^C центра многогранника по следующей формуле:

$$x_i^C = \frac{1}{n}\left[\left(\sum_{j=1}^{n+1} x_i^j\right) - x_i^A\right], \qquad (7.8)$$

$i = 1,2,...n$,

где i - номер координаты, j -номер вершины симплекса, k - номер шага итерации.

В методе деформируемого многогранника над многогранником выполняются операции отражения, растяжения, сжатия и редукции.

1. *Отражение* есть проецирование X_k^A через центр X_k^C в соответствии с соотношением

$$X_k^0 = X_k^C + \alpha\left(X_k^C - X_K^A\right), \qquad (7.9)$$

где $\alpha > 0$ - коэффициент отражения, X_k^0 - вектор координат новой (отражённой) вершины .

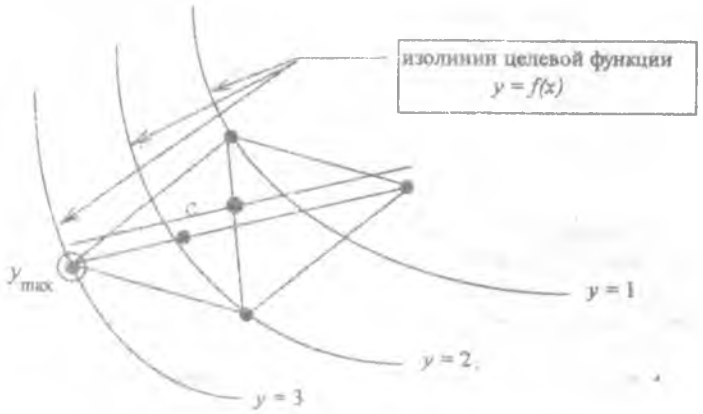


Рис. 7.3. Геометрическая интерпретация симплексного поиска

2. *Растяжение* применяется в том случае, когда отражение оказалось удачным, то есть значение функции в новой точке меньше, чем в наилучшей из вершин многогранника:

$$f\left(X_k^0\right) \le f\left(X_k^B\right),$$

при этом вектор $X_k^0 - X_k^C$ растягивается и получается новая точка

$$X_k^P = X_k^C + \gamma\left(X_k^0 - X_k^C\right), \qquad (7.10)$$

где $\gamma > 1$ - коэффициент растяжения.

3. *Сжатие* выполняется, когда в результате отражения значение функции в точке X_k^0 оказалось больше, чем во всех вершинах многогранника, кроме вершины X_k^A, то есть:

$$f\left(X_k^0\right)<f\left(X_k^A\right),$$

$$f\left(X_k^0\right)>f\left(X_k^j\right), \quad j \neq A,$$

тогда вектор $X_k^A - X_k^C$ сжимается так, что

$$X_k^C = X_k^C + \beta\left(X_k^A - X_k^C\right),$$

где $0<\beta<1$ - коэффициент сжатия

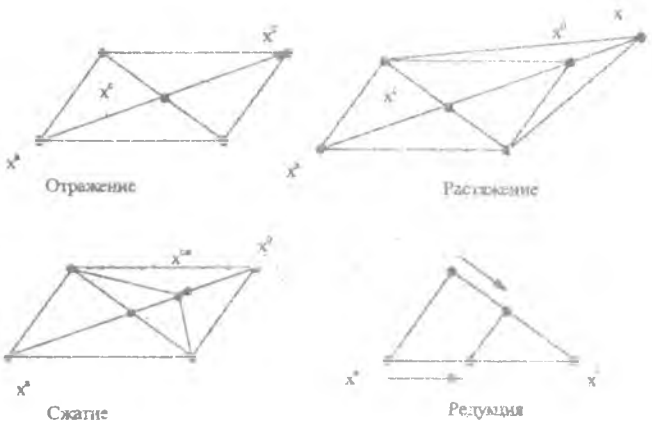


Рис. 7.4 Основные операции метода деформируемого многогранника

4. *Редукция*, то есть сжатие симплекса в два раза по отношению к вершине с наименьшим значением $f(x)$: $f\left(x_k^B\right)$.

Редукция применяется, если $f\left(X_k^0\right)>f\left(X_k^A\right)$ и выполняется по формуле

$$X_k^j = X_k^B + 0{,}5\left(X_k^j - X_k^B\right), \text{ при } j=1,2,\ldots n+1.$$

На рис.7.4 схематично показаны перечисленные операции

Метод деформируемого многогранника прекращает свою работу, если выполняются условия

$$\left[\frac{1}{n+1}\sum_{j=1}^{n+1}\left[f\left(x_k^j\right) - f\left(x_k^C\right)\right]^2\right]^{1/2} \leq \varepsilon,$$

где $\varepsilon>0$-малое число, определяющее ε-окрестность поиска экстремума.

7.4. Метод наискорейшего спуска

Численные методы прямого поиска используют лишь значение функции в вычисляемых точках. В этом смысле они позволяют решать задачу оптимизации для более широкого класса функции, однако, как правило, имеют невысокую скорость сходимости (за исключением метода деформированного многогранника). Использование дополнительной информации часто позволяет существенно увеличить скорость сходимости метода. В методах первого порядка или, как их ещё называют, в градиентных методах используются первые частные производные для организации итерационного процесса поиска оптимума функции.

Известно, что градиент ортогонален к гиперповерхности отклика целевой функции в точке его вычисления и это направление указывает локальное направление наибыстрейшего возрастания целевой функции. Зная такое направление на каждом шаге итерационного метода, несложно организовать поиск оптимума функции по траектории наискорейшего спуска или подъёма к минимуму или максимуму функции.

Все градиентные методы используют указанные особенности поведения градиента, а их стратегия поиска строится на рекуррентном выражении вида

$$X_{k+1} = X_k + hE_k, \tag{7.11}$$

где h- величина шага; E_k -единичный вектор направления поиска на k– м шаге.

Для метода наискорейщего спуска

$$E_k = -grad\left(f(x_k)\right) \Big/ \left|grad\left(f(x_k)\right)\right|,$$

$$grad\left(f(x_k)\right) = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots \frac{\partial f}{\partial x_n}\right)^T. \tag{7.12}$$

Элементы вектор-градиента могут вычисляться по аналитическим зависимостям или численными методами. В последнем случае говорят о квазиградиентных методах.

В градиентных методах для выбора и модификации шага h следует задаться алгоритмом. Наиболее простым методом модификации шага градиентного метода является метод половинного деления шага в случае, если в направлении антиградиента не удаётся найти меньшее значение функции. На

рисунке 7.5. приведена геометрическая иллюстрация работы метода наискорейшего спуска.

7.5. Метод Ньютона

Методы второго порядка используют первые и вторые производные целевой функции. Среди этих методов наиболее известен метод Ньютона и его многочисленные модификации, получившие своё название в связи с тем, что в нём необходимое условие безусловного экстремума функции $grad\left(F\left(X\right)\right)=0$ рассматривается как система алгебраических уравнений, которая решается методом Ньютона.

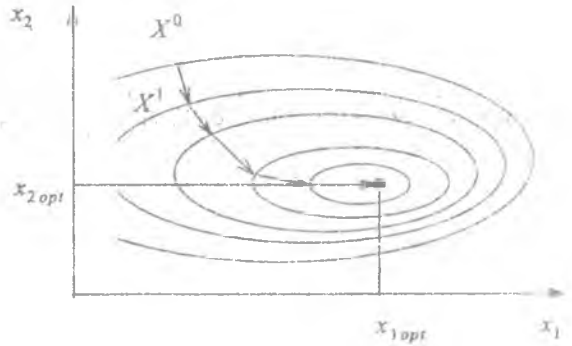


Рис. 7.5. Метод наискорейшего спуска

Обозначим через H_k-матрицу Гессе целевой функции $F(X)$:

$$H_k=\begin{Vmatrix} \frac{d^2f}{dx_1^2} & \frac{d^2f}{dx_1dx_2} & \cdots & \frac{d^2f}{dx_1dx_n} \\ \frac{d^2f}{dx_2dx_1} & \frac{d^2f}{dx_2^2} & \cdots & \frac{d^2f}{dx_2dx_n} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{d^2f}{dx_ndx_1} & \frac{d^2f}{dx_ndx_2} & \cdots & \frac{d^2f}{dx_n^2} \end{Vmatrix}, \qquad (7.13)$$

вычисленную в точке X_k, тогда итерационная формула метода Ньютона поиска оптимума функций имеет вид

$$X_{k+1}=X_k-H_k^{-1}\cdot grad\left(F\left(X_k\right)\right). \qquad (7.14)$$

Как видно из (7.14), для реализации каждого шага по методу

Ньютона нет необходимости определять величину шага k.

Если целевая функция $F(X)$ квадратичная и матрица Гессе положительно определённая, то метод Ньютона позволяет найти минимум целевой функции за один шаг независимо от выбора начальной точки. В противном случае экстремум будет найден за большее число шагов.

Скорость сходимости метода Ньютона выше, чем скорость сходимости методов нулевого и первого порядка. Однако, если матрица Гессе имеет знакопеременные значения, то может оказаться, что метод Ньютона не будет сходиться. Главный недостаток метода Ньютона заключается в отсутствии простых и экономичных алгоритмов вычисления матрицы Гессе и в отсутствии твёрдых гарантий его сходимости.

ГЛАВА 8

Метод конечных разностей решения уравнения в частных производных

8.1. Введение

Математические модели в настоящее время становятся средством познания закономерностей разнообразной природы (физических или химических процессов, функционирования технических систем, биологических процессов и т.д.). Значение математического моделирования возрастает в связи с естественной тенденцией к оптимизации технических устройств и технологических схем.

Одним из универсальных методов построения математических моделей является представление последних в виде систем уравнений.

Если независимыми переменными в таких моделях выступают пространственные координаты и время, то уравнения в таких системах представляются дифференциальными уравнениями в частных производных. Эти уравнения составляют основу математического моделирования на микроуровне и позволяют описывать процессы с высоким уровнем точности.

Например, уравнение

$$U''_{xx} + U''_{yy} = -2 \qquad (8.1)$$

с граничными условиями

$$U(x,y)\big|_{(x,y)\in C} = 0, \qquad (8.2)$$

получившее название уравнения Пуассона, описывает изменение напряжений, возникающих при упругом кручении цилиндрического стержня. Сечение такого стержня представлено на рисунке 8.1.

Функция U(x,y) - "потенциальная" функция, связанная с напряжением сдвига в направлении осей x и y (τ_x τ_y) следующими соотношениями:

$$\tau_x = \alpha U'_x,$$

$$\tau_y = \alpha U'_y.$$

К уравнению Пуассона приводит также задача определения распределения потенциалов на проводящей пластине при заданных потенциалах на границе пластины.

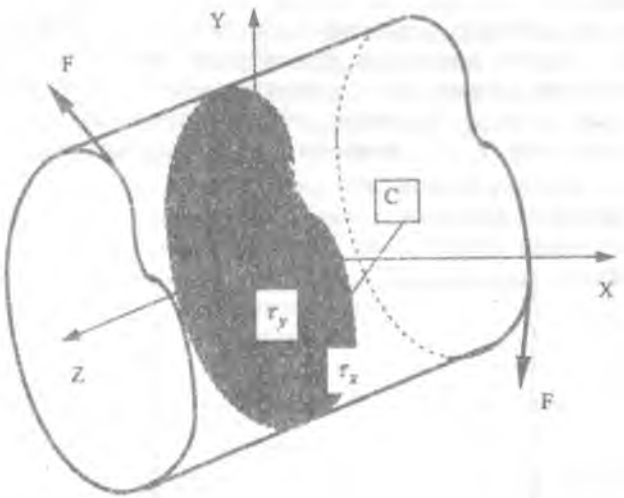


Рис.8.1.Сечение цилиндрического стержня

Решить уравнение (8.1), удовлетворяющее граничным условиям (8.2), значит, найти такую функцию U(x,y), которая удовлетворяя условию (8.2), обращает в тождество уравнение (8.1).

Следует отметить, что аналитические методы решения известны лишь для сравнительно небольшого числа уравнений в частных производных, поэтому на практике для их решения применяют численные методы.

8.2. Метод конечных разностей

Численные методы решения дифференциальных уравнений в частных производных (ДУЧП) основаны на *дискретизации* пространства переменных и *алгебраизации* задачи. **Дискретизация** заключается в замене непрерывных переменных конечным множеством их значений в пространственных или временных интервалах. **Алгебраизация** - в замене частных производных соответствующими алгебраическими соотношениями.

Применяют различные способы *дискретизации* и *алгебраизации* переменных при решении дифференциальных уравнений в частных производных. Большинство используемых методов относится либо

к методам конечных разностей, либо к методам конечных элементов.

В методе конечных разностей *дискретизация* заключается в покрытии области изменения независимых переменных сеткой равноотстоящих линий (с соответствующими шагами и по переменным x и y) и замене области изменений конечным множеством точек (x_i, y_j), являющихся узлами сетки (см. рис.8.2).

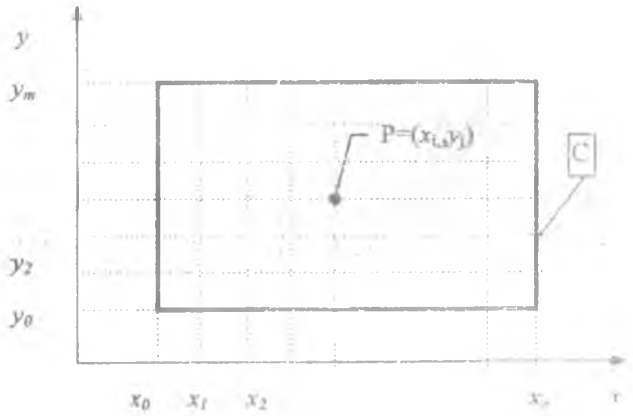


Рис.8.2. Дискретизация пространства независимых переменных методом конечных разностей

Метод конечных разностей рассмотрим применительно к численному решению линейных дифференциальных уравнений в частных производных:

$$A\frac{\partial^2 U}{\partial x^2} + B\frac{\partial^2 U}{\partial x \partial y} + C\frac{\partial^2 U}{\partial y^2} + D\frac{\partial U}{\partial x} + E\frac{\partial U}{\partial y} + FU = G, \qquad (8.3)$$

с соответствующими граничными или начальными условиями.

Линейные уравнения в частных производных (8.3) в зависимости от значений коэффициентов уравнения классифицируют следующим образом:

1. Уравнение относится к *эллиптическому* типу, если

$$B^2 - 4AC < 0;$$

2. Уравнение относится к *параболическому* типу, если

$$B^2 - 4AC = 0;$$

3. Уравнение относится к *гиперболическому* типу, если

$$B^2 - 4AC > 0.$$

Классификация достаточно условная, поскольку если коэффициенты уравнения зависят от фазовых переменных, то уравнение (8.3) может превратиться в уравнение смешанного типа. Например, в одной области гиперболического типа, в другой - параболического типа. Однако классификация оправдана, поскольку, как мы увидим далее, тип уравнения во многом определяет метод его решения.

Для простоты предположим, что *область решения* (некоторая конечная или бесконечная область, имеющая границу, внутри которой ищется решение ДУЧП) представлена прямоугольником (см. рис. 8.2). В этом случае легко реализуется дискретизация пространства переменных сеткой линий:

$\Delta_x: x_0, x_1, \ldots, x_n;$

$\Delta_y: y_0, y_1, \ldots, y_m;$

где $x_i = x_0 + h$; $y_j = y_0 + k$, h, k - соответствующие приращения по переменным x и y. Решение уравнения ищется в узлах сетки $U_{ij} = U(x_i, y_j)$, которое, в известном смысле, заменяет истинное решение $U(x,y)$. Для краткости записи математических выражений далее будем пользоваться "индексированной" формой записи решения ДУЧП.

Алгебраизация задачи заключается в замене частных производных их конечно-разностными аналогами (см. п.5.8), а также в формировании конечно-разностного аналога ДУЧП. Частные производные в этом случае можно заменить выражениями

$$U''_{xx}(x_i, y_j) = \frac{U_{i+1\,j} - 2U_{i\,j} + U_{i-1\,j}}{h^2} + O(h^2),$$

$$U''_{yy}(x_i, y_j) = \frac{U_{i\,j+1} - 2U_{i\,j} + U_{i\,j-1}}{k^2} + O(k^2),$$

$$U'_x(x_i, y_j) = \frac{U_{i+1\,j} - U_{i\,j}}{h} + O(h),$$

$$U'_y(x_i, y_j) = \frac{U_{i\,j+1} - U_{i\,j}}{k} + O(k).$$

Например, для уравнения Лапласа $U''_{xx} + U''_{yy} = 0$ мы получим конечно-разностный аналог:

$$\frac{U_{i+1\,j} - 2U_{i\,j} + U_{i-1\,j}}{h^2} + \frac{U_{i\,j+1} - 2U_{i\,j} + U_{i\,j-1}}{k^2} = 0,$$

который при $h, k \to 0$ превращается в ДУЧП.

Следует помнить, что при $h, k \to 0$ стремление конечно-разностного уравнения (КРУ) к дифференциальному

($КРУ \xrightarrow[h,k\to 0]{} ДУЧП$) вовсе не означает, что решение КРУ будет стремиться к решению ДУЧП. Однако в большом количестве случаев это справедливо.

8.3. Решение уравнения эллиптического типа

Одним из простейших примеров уравнения эллиптического типа является уравнение Пуассона:

$$U''_{xx} + U''_{yy} = -f(x,y), \tag{8.4}$$

$$U(x,y)\big|_{(x,y)\in\Gamma} = \varphi(x,y), \tag{8.5}$$

когда решение ищется в прямоугольной области G, имеющей границу Γ. Граничные условия (8.5) определяют первую краевую задачу (задачу Дирихле).

Если на достаточно гладкой границе Г, имеющей нормаль, задано условие

$$\frac{\partial U(x,y)}{\partial n}\Big|_{(x,y)\in\Gamma} = \varphi(x,y), \tag{8.6}$$

то говорят, что задана вторая краевая задача или задача Неймана.

Третья краевая задача представляется уравнением

$$\frac{\partial U(x,y)}{\partial n}\Big|_{(x,y)\in\Gamma} + \alpha(x,y)U(x,y) = \varphi(x,y). \tag{8.7}$$

В формулах (8.6)-(8.7) n - нормаль к границе Γ.

Дискретизация пространства переменных производится разбиением прямоугольной области сеткой параллельных линий (рис. 8.3).

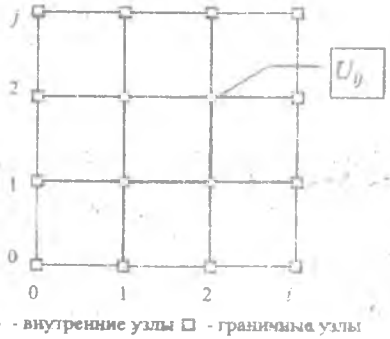


Рис. 8.3

Заменив частные производные в (8.4) разностными соотношениями, получим КРУ:

$$\frac{U_{i+1\,j} - 2U_{i\,j} + U_{i-1\,j}}{h^2} + \frac{U_{i\,j+1} - 2U_{i\,j} + U_{i\,j-1}}{k^2} = -f_{ij}$$

Умножив последнее уравнение на k^2 и введя параметр $\lambda = k/h$, получим

$$\lambda^2 U_{i+1\,j} - 2(1+\lambda^2)U_{i\,j} + \lambda^2 U_{i-1\,j} + U_{i\,j+1} + U_{i\,j-1} = -k^2 f_{ij}. \quad (8.8)$$

Составив конечно-разностные уравнения для всех внутренних точек прямоугольника, (при $\lambda = 1$) получим систему линейных уравнений:

$$\begin{cases} -4U_{11} + U_{21} + U_{12} = -k^2 f_{11} - U_{10} - U_{01}, \\ U_{11} - 4U_{21} + U_{22} = -k^2 f_{21} - U_{20} - U_{31}, \\ U_{11} - 4U_{12} + U_{22} = -k^2 f_{12} - U_{02} - U_{13}, \\ U_{21} + U_{12} - 4U_{22} = -k^2 f_{22} - U_{23} - U_{32}. \end{cases} \quad (8.9)$$

где $U_{10} = \varphi_{10}, U_{01} = \varphi_{01}, U_{20} = \varphi_{20}, U_{31} = \varphi_{31}, \ldots, U_{32} = \varphi_{32}$ - граничные условия.

Выбор метода решения системы КРУ

Систему линейных уравнений (8.9) можно решать любым из известных численных методов. Например, прямым методом Гаусса. Однако несложные подсчеты показывают, метод Гаусса в этом случае малопригоден. Для "скромной" по плотности сетки размера 100*100 число неизвестных значений функции равно 10000, а матрица системы уравнений приобретает размеры 10000*10000. При решении такой системы уравнений возникают проблемы с памятью ЭВМ, необходимой для размещения матрицы системы, а также серьезную проблему составляют ошибки округления арифметических операций.

Учитывая значительную "разряженность" матрицы системы (8.9), значительно выгоднее использовать итерационные методы, например, метод Гаусса-Зейделя. В этом случае система уравнений (8.9) приобретает вид

$$\begin{cases} U_{11}^{(k+1)} = \frac{1}{4}(k^2 f_{11} + \varphi_{10} + \varphi_{01} + U_{21}^{(k)} + U_{12}^{(k)}), \\ U_{21}^{(k+1)} = \frac{1}{4}(k^2 f_{21} - \varphi_{20} - \varphi_{31} + U_{11}^{(k+1)} + U_{22}^{(k)}), \\ U_{12}^{(k+1)} = \frac{1}{4}(k^2 f_{12} - \varphi_{02} - \varphi_{13} + U_{11}^{(k+1)} + U_{22}^{(k)}), \\ U_{22}^{(k+1)} = \frac{1}{4}(k^2 f_{22} + \varphi_{23} + \varphi_{32} + U_{21}^{(k+1)} + U_{12}^{(k+1)}). \end{cases}$$

При составлении уравнений удобно пользоваться пятиточечным шаблоном КРУ (см. рис. 8.4). Начальное приближение $U_{ij}^{(0)}$ можно задавать произвольно, однако чем ближе

$U_{ij}^{(0)}$ к решению системы уравнений, тем раньше закончится итерационный процесс решения системы уравнений.

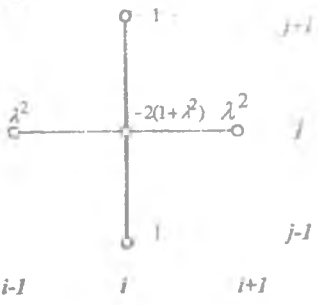


Рис. 8.4

Критерием остановки итерационного алгоритма Гаусса-Зейделя может служить условие достижения заданной точности ε при определении матрицы решения системы уравнений U_{ij}^{*}:

$$\max_{ij} \frac{\left|U_{ij}^{(k+1)} - U_{ij}^{(k)}\right|}{\left|U_{ij}^{(k+1)}\right|} < \varepsilon,$$

если $\left|U_{ij}^{(k+1)}\right| > 1$ и

$$\max_{ij} \left|U_{ij}^{(k+1)} - U_{ij}^{(k)}\right| < \varepsilon, \text{ если } \left|U_{ij}^{(k+1)}\right| \le 1.$$

Перед использованием метода Гаусса-Зейделя необходимо убедиться в его *сходимости*. Как известно (см. п. 4.3), метод Гаусса-Зейделя сходится, если матрица системы уравнений обладает диагональным преобладанием. В нашем случае для произвольного λ и произвольной точки сетки выполняется условие

$$\left|-2(1+\lambda^2)\right| = \left|\lambda^2\right| + \left|\lambda^2\right| + 1 + 1,$$

а для приграничных точек сетки, когда часть переменных U_{ij} (известных из граничных условий) переносится в правую часть уравнения, условие диагонального преобладания выполняется строго, следовательно, достаточный признак сходимости метода Гаусса-Зейделя *выполнен* и его можно применять для решения системы уравнений.

φ_{03}	φ_{13}	φ_{23}	φ_{33}
φ_{02}	$U_{12}^{(0)}$	$U_{22}^{(0)}$	φ_{32}
φ_{01}	$U_{11}^{(0)}$	$U_{21}^{(0)}$	φ_{31}
φ_{00}	φ_{10}	φ_{20}	φ_{30}

Рис.8.5

Алгоритмическая сторона вопроса при реализации метода Гаусса-Зейделя выглядит еще проще. Для нашего примера создается двумерный массив решения системы линейных уравнений, в который заносятся граничные условия

и начальное приближение решения системы (см. рис.8.5). Используя пятиточечный шаблон и организуя циклический обход внутренних точек, производится пересчет значений U_{ij}, до тех пор пока не выполнится условие достижения заданной точности решения системы линейных уравнений.

8.4 Решение уравнений гиперболического типа

Простейшим примером уравнения гиперболического типа (волнового уравнения) является уравнение плоского колебания струны (см. рис. 8.6)

$$U''_{xx} + aU''_{tt} = 0 \qquad (8.10)$$

при следующих граничных и начальных условиях:

$$\begin{aligned} &U(0,t) = U(L,t) = 0; \quad t \geq 0; \\ &U(x,0) = f(x), \quad 0 \leq x \leq L; \\ &U'(x,0) = \varphi(x) \quad 0 < x < L. \end{aligned} \qquad (8.11)$$

Первое условие закрепляет концы струны, второе описывает начальное положение струны, третье - начальную скорость точек струны. Здесь $a = \frac{w}{Tg}$, w - вес струны на единицу длины, T - натяжение струны, g - ускорение свободного падения.

Далее для простоты положим a=1.

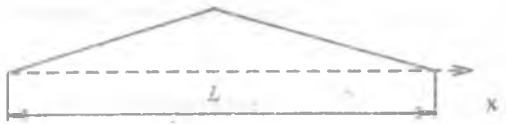


Рис.8.6

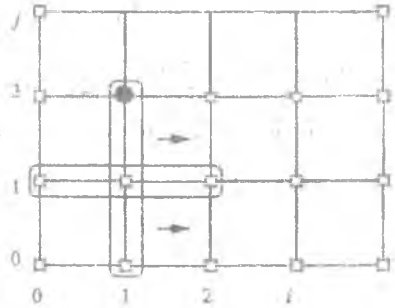
Схема решения уравнений гиперболического типа во многом аналогична уравнению эллиптического типа.

Пусть $h=L/n$ - шаг по переменной x. Приращение времени введем формулой $k = \lambda h$, тогда конечно-разностное уравнение будет выглядеть так:

$$\lambda^2 U_{i+1\,j} + 2(1-\lambda^2)U_{i\,j} + \lambda^2 U_{i-1\,j} - U_{i\,j+1} - U_{i\,j-1} = 0. \qquad (8.12)$$

Однако составить систему линейных уравнений нам не удастся, поскольку при бесконечности времени мы получим бесконечное число уравнений с бесконечным числом неизвестных. Поступим иначе. для второго начального условия (8.11) составим конечно-разностное уравнение:

$$\frac{U_{i1} - U_{i0}}{k} = g_i \quad \text{или} \quad U_{i1} = f_i + g_i. \tag{8.13}$$



○ - внутренние узлы □ - граничные узлы

Рис. 8.7

С помощью уравнения (8.13) вычислим второй временной слой при $t=k$. Тогда, разрешив уравнение (8.12) относительно узла $(i,j+1)$, получим явное конечно-разностное уравнение:

$$U_{i\,j+1} = \lambda^2 U_{i+1\,j} + 2(1-\lambda^2)U_{i\,j} + \lambda^2 U_{i-1\,j} - U_{i\,j-1}. \tag{8.14}$$

Используя уравнение (8.14), несложно подсчитать третий временной слой (см. рис. 8.7), на основе первого и второго - четвертый и т.д.

Для уравнений гиперболического типа доказано, что метод сходится, если $\lambda < 1$.

8.5 Решение уравнений параболического типа

Простейшим уравнением параболического типа является одномерное уравнение теплопроводности

$$U''_{xx} = U'_t, \tag{8.15}$$

при следующих граничных и начальных условиях

$$\begin{aligned} &U(0,t) = T_0, U(L,t) = T_L; \quad t \ge 0; \\ &U(x,0) = f(x), \quad 0 \le x \le L; \end{aligned} \tag{8.16}$$

Если для U'_t применить правую конечную разность $U'_t \approx \frac{U_{i\,j+1} - U_{i\,j}}{k}$, то конечно-разностное уравнение выглядит так:

$$U_{i\,j+1} = \lambda U_{i+1\,j} + (1-2\lambda)U_{i\,j} + \lambda U_{i-1\,j}, \tag{8.17}$$

где $\lambda = k / h^2$.

Уравнение (8.17) имеет четырехточечный шаблон, который показан на рисунке 8.8.

Рис. 8.8.

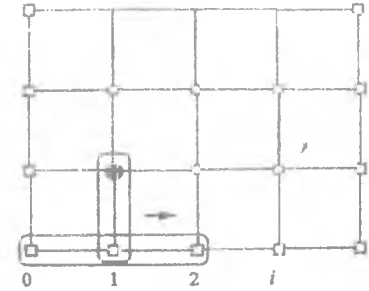


Рис. 8.9

Используя явное конечно-разностное уравнение (8.15) и начальные условия (8.16), слой за слоем, можно вычислить $U_{i\,j}$ в произвольных временных слоях. Схема расчета показана на рисунке 8.9.

Явная схема решения параболического уравнения сходится, если $\lambda < 1$.

Упражнения

1. Практическое вычисление функций

1.1. В предположении, что a есть положительное действительное число (соответствующим образом округленное) и что число b можно записать в ЭВМ точно, оценить максимально возможные относительные ошибки для выражений

$$u = a + b \quad и \quad v = \frac{\left(a^2 - b^2\right)}{a - b}.$$

Построить графы вычислительных процессов.

1.2. Пусть числа a, b и c положительны, заданы точно и $a \approx b$. Сравнить между собой максимально возможные относительные ошибки выражений

$$u = \frac{a - b}{c} \quad и \quad v = \frac{a}{c} - \frac{b}{c}.$$

1.3. Исследовать максимально возможные ошибки решения системы уравнений

$$\begin{cases} ax + by = 0, \\ dx + ey = f. \end{cases}$$

1.4. Разложить по формуле Тейлора функцию $y = arctg\, x$. Определить радиус сходимости, построенного ряда Тейлора. Определить число членов ряда, необходимого для вычисления функции с ошибкой ограничения не более чем $5*10^{-5}$ в точке $x = 2$ (ошибкой округления пренебречь).

1.5. Используя формулу оценки остаточного члена ряда Тейлора, оценить верхнюю грань абсолютной величины ошибки ограничения для функции

$$\ln x = (x-1) - \frac{(x-1)^2}{2} + \frac{(x-1)^3}{3} - \ldots,$$

в точке $x = 2$, вычисленной по первым семи членам ряда.

1.6. Найти первые пять коэффициентов разложения по полиномам Чебышева функции

$$f(x) = \sqrt{1 - x^2}.$$

Представить функцию $f(x)$ в разложении по полиномам Чебышева. Сколько полиномов Чебышева требуется для вычисления функции $f(x)$ с ошибкой ограничения не более $5*10^{-5}$ в точке $x = 0{,}5$?

1.7. Функцию $f(x) = 6x^4 - 2x^3 + x^2 - x + 4$ выразить через полиномы Чебышева.

1.8. Проведите экономизацию степенного ряда

$$e^x \approx 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \frac{x^6}{6!} + \frac{x^7}{7!}.$$

Сведите его к полиному, в котором наивысшая степень x будет равна шести.

1.9. Для табличной функции

x_i	0	1	2	3
y_i	1	2	4	8

построить интерполяционный многочлен Лагранжа. Полученную непрерывную функцию табулировать с шагом h=0.25

1.10. Для табличной функции

x_i	0	1	2	3
y_i	1	2	4	8

построить интерполяционный многочлен Ньютона. Оценить абсолютную величину ошибки ограничения (остаточного члена формулы Ньютона). Табулировать функцию с шагом h=0.25.

1.11. Для табличной функции

x_i	0	1	2	3	4
y_i	0	0.258	0.5	0.707	0.866

построить интерполяционный линейный сплайн $S_1(x)$.

1.12. Для табличной функции $f(x)$ (см. задачу 1.11.) построить интерполяционный параболический сплайн $S_2(x)$. В качестве

граничных условий использовать $S_2'(x_0) = 0$. Протабулировать функцию $S_2(x)$ с шагом h=0.25.

1.13. Для табличной функции $f(x)$ (см. задачу 1.11.) построить интерполяционный кубический сплайн $S_3(x)$. В качестве граничных условий использовать $S_3''(x_0) = S_3''(x_4) = 0$.

1.14. Для табличной функции $f(x)$ построить интерполяционный параболический сплайн $S_2(x)$. Подобрать оптимальные граничные условия по критерию оценки среднеквадратической кривизны функции :

x_i	0	1	2	3	4	5
y_i	0	1	2	2	1	0

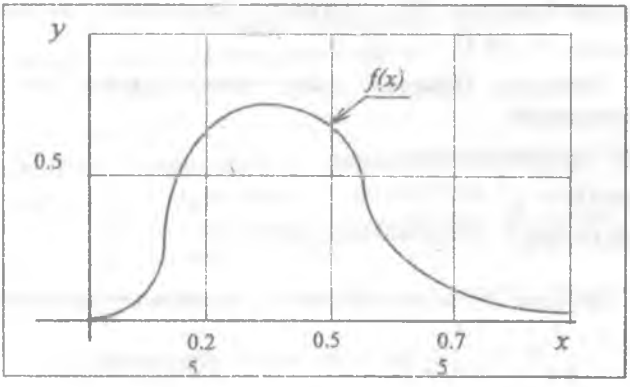
1.15. Для функции $u = f(x,y)$ построить двумерный параболический сплайн $S_2(x,y)$. В качестве граничных условий выбрать нулевые значения вторых частных производных на границе интерполяционной сетки сплайна.

Функция $u = f(x,y)$ задана таблично:

y \ x	0	1	2	3
0	0	0	0	0
1	1	2.72	7.39	20.1
2	2	5.44	14.8	40.2
3	3	8.15	22.2	60.3

1.16. Выбрать рациональный вид аппроксимирующей функции и аппроксимировать экспериментальную зависимость

многочленом степени m. Оценить погрешность аппроксимации. Построить график.



1.17. В условиях задачи 1.16. построить аппроксимирующий параболический сплайн.

2. Методы численного решения уравнений и систем уравнений

2.1. Вычислить отрицательное значение квадратного корня из 0.5 методом последовательных приближений. Начальное приближение взять: $x_0 = -0.25$. Можно ли тем же методом найти положительное значение корня?

2.2. Методом Ньютона-Рафсона найти с точностью до третьего знака все корни уравнения

$$x^3 - 1.473x^2 - 5.738x + 6.763 = 0 .$$

2.3. Попытайтесь использовать метод Ньютона-Рафсона для решения уравнения $x^3 - 2x^2 - 3x + 10 = 0$, причем в качестве x_0 возьмите 1.9. Чем объясняется странное поведение последних значений корня?

2.4. Методом простой итерации найти положительное решение системы уравнений

$$\begin{cases} x^2 + y^2 = 1, \\ xy = 1. \end{cases}$$

Начальное приближение $X^0 = (0.5; 0.05)$. Исследовать сходимость метода в точке $X = (0, 1)$.

2.5. Методом Ньютона найти положительное решение системы уравнений

$$\begin{cases} 2x^2 - xy - 5x + 1 = 0, \\ x + 3\ln x - y^2 = 0, \end{cases}$$

исходя из начального приближения $x_0=1$, $y_0=0$.

2.6. Методом Гаусса решить систему линейных уравнений

$$\begin{cases} x - y + z = -4, \\ 5x - 4y + 3z = -12, \\ 2x + y + z = 11. \end{cases}$$

2.7. Решить систему линейных уравнений методом Гаусса-Зейделя с точностью до 0.02:

$$\begin{cases} 10x + 2y + 6z = 28, \\ x + 10y + 9z = 7, \\ 2x - 7y - 10z = -17. \end{cases}$$

2.8. Методом Гаусса-Зейделя решить систему линейных уравнений

$$\begin{cases} 20x + 2y + 6z = 38, \\ x + 20y + 9z = -23, \\ 2x - 7y - 20z = -57. \end{cases}$$

3. Численное дифференцирование и интегрирование функций

3.1. Используя конечно-разностную формулу первого порядка, найти производные функций в точке $x = 1$:

а) $y = \sin(x)/x$,

б) $y = x^2 e^x$.

Исследовать зависимость погрешности вычисления производной функции от h (h -приращение по аргументу).

3.2. Используя конечно-разностные формулы второго и третьего порядка найти первую и вторую производные функции $y = \sqrt{1 + x^2}$ в точке $x = 1$. Оценить погрешности вычислений.

3.3. Вычислить точное значение интеграла $\int_0^1 f(x)dx$, а также приближенные значения по формуле трапеций при n=20 и по формуле Симпсона с 2n=20. В качестве $f(x)$ возьмите следующие функции:

а) $f(x) = x\sqrt{1 + x^2}$,

б) $f(x) = \dfrac{1}{x\sqrt{1 + x^3}}$,

в) $f(x) = \dfrac{x}{\sin^2 x}$.

Оценить погрешность вычислений и остаточные члены формул трапеции и Симпсона.

4. Решение обыкновенных дифференциальных уравнений

4.1. Дано уравнение $\dfrac{y'}{2x} + 3xy = e^{-2x^3}$ и начальное условие y(0)=5. Решить это уравнение, используя исправленный метод Эйлера и модифицированный метод Эйлера, принимая h=0.1 и продолжая решение до x=2. Сравнить результат с точным решением. Исследовать сходимость и устойчивость исправленного метода Эйлера.

4.2. Дано уравнение $y' - \frac{y}{2x} = \frac{x^2}{2y}$ и начальное условие y(1)=2. Вычислите y(x), используя модифицированный метод Эйлера и метод Рунге-Кутта 4-го порядка. Шаг интегрирования h=0.1 ($x \in [1;3]$).

4.3. Дано уравнение $y' - 2xy = 2x^2 y^2$ и начальное условие y(0)=1. Проинтегрировать уравнение, используя метод прогноза и коррекции на [0; 3].

5. Методы численной оптимизации

5.1. Методом деформируемых многогранников найти экстремум функции $z=(x^2-y)^2+(x-1)^2$. В качестве начального приближения выбрать точку $x_0=10$, $y_0=5$.

5.2. Найти экстремум функции $y = \frac{(x+1)^3}{4(x-2)^2}$ методом Ньютона. В качестве начального приближения выбрать точку $x_0=10$.

ПРИЛОЖЕНИЕ 1. Задания для самостоятельных работ

Рабочая программа курса "Численные методы" предполагает самостоятельное решение набора индивидуальных задач, закрепляющих освоение студентами основных разделов курса. Задачи сгруппированы в четыре блока. По каждому блоку (после решения задач) оформляется пояснительная записка, которая сдается преподавателю на проверку. Оценки, полученные за выполнение блоков задач, учитываются при выставлении экзаменационной оценки. Студенты, не выполнившие самостоятельной работы,до экзаменов не допускаются.

Пояснительная записка составляется в соответствии со стандартами оформления научно-технической документации. Записка должна содержать:

- титульный лист с реквизитами наименования учреждения, наименования предмета, номера группы, ФИО исполнителя и т.д;
- полное описание постановок задач;
- описание выбранных методов решения задач;
- описание результатов вычислительных экспериментов в виде таблиц и графиков;
- при необходимости приводится оценки погрешностей вычислений;
- выводы по результатам проделанной работы.

Исследования, описанные в блоках задач, проводятся над одной и той же функцией f(x) (см. таблицу П1).

Блок 1

1. В предположении, что аргумент функции f(x) и ее коэффициенты заданы неточно, оценить величину абсолютной и относительной погрешностей. Предполагается, что относительные погрешности арифметических операций и представления чисел равны $0.5*10^{-t}$, где t - количество значащих цифр в разрядной сетке ЭВМ. Построить граф вычислительного процесса. Вычислить функцию с пятью верными значащими цифрами в точке x0.

2. Разложить по формуле Тейлора функцию f(x). Выбрать точку разложения. Определить радиус сходимости построенного ряда. Оценить величину погрешностей $\varepsilon_{мет}$ и $\varepsilon_{окр}$ (методической ошибки и ошибки округления). Определить число членов ряда, необходимое для вычисления функции в точке x=1.5 с точностью $\varepsilon_{п} = \varepsilon_{мет} + \varepsilon_{окр} = 0.5 * 10^{-5}$.

3. Для полученного разложения функции f(x) в ряд Тейлора произвести экономизацию степенного ряда, который вычислял бы функцию f(x) с точностью $\varepsilon_{п} < 0.5*10^{-3}$.

Блок 2

4. Произвести табуляцию функции f(x) на [a,b]. Шаг табуляции функции выбрать из соображений достижения заданной точности для интерполяционного многочлена Лагранжа не выше третьего порядка. На любом участке отрезка построить интерполяционный многочлен, вычисляющий функцию $\forall x \in [a,b]$ с 4-я верными значащими цифрами.

5. Используя интерполяционную сетку полинома Лагранжа, построить интерполяционный многочлен Ньютона.

6. Для табличной функции, полученной в задании 5, построить:

- линейный интерполяционный сплайн;
- параболический сплайн;
- кубический сплайн.

Указание: при построении параболического и кубического сплайнов в качестве дополнительных условий использовать $f'(a)=0$ и $f'(a)=f'(b)=0$.

Сравнить величины абсолютных погрешностей при интерполировании функции f(x) сплайнами и полиномом между собой. Построить графики ошибок интерполирования функций.

7. Протабулировать функцию f (x) в 15 точках, начиная с x_0 =a с шагом h. На вычисленные значения y_i наложить случайную погрешность ε_i ($\varepsilon_i \leq 10\% \ \delta_{yi}$). Выбрать рациональный вид аппроксимирующей функции и аппроксимировать

экспериментальную зависимость многочленом. Оценить погрешность аппроксимации. Построить график.

Блок 3

8. Найти хотя бы один корень уравнения $f(x) = c$ с точностью до 5 значащих цифр. Исследовать сходимость выбранного численного метода. Для определения начального приближения использовать графический метод решения задачи.

9. Найти приближенное решение системы нелинейных уравнений

$$\begin{cases} y^2 + f(x) = c \\ x^2 - y = 0 \end{cases}$$

с точностью до 5 значащих цифр. Для определения начального приближения использовать графический метод решения задачи.

10.Вычислить приближенное значение интеграла $\int_a^b f(x)dx$ с точностью до 5 значащих цифр. (При оценке погрешности учитывать не только методическую ошибку, но и ошибки округления).

11. Используя конечно-разностные формулы первого и второго порядка, найти производные функции в точке x0 . Исследовать зависимость погрешности вычисления производной от шага приращения аргумента h.

Блок 4

12. Дано уравнение $y' + cy = f(x)$ и начальное условие $y(x0) = b$. Найти численное решение дифференциального уравнения на отрезке [x0; b]. Исследовать сходимость и устойчивость численного метода.

Таблица П1

№	f(x)	Параметры x0, c, a, b
1	2	3
1	$\pi(7-2x)^3$	$\frac{\pi}{4}$; 10; 0; 2
2	$\pi\left(9-\frac{x}{3}\right)^4$	π; 10; 20; 27
3	$6(\pi+2x)^{\frac{1}{2}}$	$\frac{\pi}{4}$; 10; -1,5; 6
4	$4x^2(3\pi-2x)^{\frac{1}{2}}$	$\frac{\pi}{4}$; 10, 0; 4,5
5	$5x(5\pi+2x)^{-\frac{1}{4}}$	$\frac{\pi}{4}$; 5; 0; 6
6	$8\pi(12+\pi x)^{-\frac{1}{2}}$	$\frac{\pi}{4}$; 5; 0; 3,5
7	$\pi x\left(10+\frac{x}{2}\right)^{-1}$	π; 2; 0; 10

1	2	3
8	$\pi(10+\pi x)^{-2}+x^2$	$\frac{\pi}{2}$; 5; 0; 3,1
9	$8(12+\pi x)^{-3}-\pi x$	π; -4; 2; 3,5
10	$\pi\frac{\sin 8x}{x}+x^2$	$\frac{\pi}{7}$; 5; 0,1; 0,85
11	$\pi x\cdot\sin(8x+10)$	$\frac{\pi}{7}$; 1; 0; 0,75
12	$x\pi\cdot\cos 8x+x^3$	$\frac{\pi}{7}$; 1; 0; 0,75
13*	$\frac{\pi}{x}\cdot tg\left(\frac{3x}{8}\right)+x$	$\frac{\pi}{3}$; 5; 0,1; 4
14*	$\pi x^2\cdot arctg\left(\frac{5x}{11}\right)+x^3$	$\frac{\pi}{3}$; 20; 0,1; 4
15*	$\pi x^2\cdot ctg\left(\frac{3x}{8}\right)-x$	$\frac{\pi}{3}$; 5; 0,1; 6

1	2	3
16*	$\pi x \cdot th\left(\frac{5x}{11}\right) - x^2$	$\frac{\pi}{3}$; -5; 0,1; 6
17	$\pi x^2 \cdot e^{\frac{5x-1}{2}} + \frac{1}{x}$	$\frac{\pi}{3}$; 6; 0,1; 1
18	$\pi x \cdot e^{\frac{7x+3}{4}} + x^3$	π; 10; 0; 2
19*	$\frac{\pi x}{e^{3x} - 1}$	$\frac{\pi}{3}$; 0,5; 0; 2
20*	$\frac{\pi x}{e^{\pi x} - 1} + x^2$	$\frac{\pi}{3}$; 1; 0; 2
21	$\frac{\pi}{x}\ln(8 - 3x) + x^2$	$\frac{\pi}{3}$; 10; 0; 2
22	$\pi x \cdot \ln(8 + 3x) - x^2$	$\frac{\pi}{3}$; 2; 0; 6
23*	$\pi \ln\lvert\sin x\rvert + x$	$\frac{\pi}{3}$; 1; 0,1; 3

Продолжение табл. П1

1	2	3
24*	$\frac{1}{\pi x}\ln(\cos x) - x^3$	$\frac{\pi}{4}$; -1; 0,1; 1,3
25*	$\frac{\pi}{x}\ln\lvert tgx\rvert + x$	$\frac{\pi}{4}$; -10; 0,1; 1,3
26*	$\frac{\pi}{x}\arcsin\left(\frac{3x}{8}\right) + x^2$	$\frac{\pi}{4}$; 2; 0,1; 2,5
27*	$\pi x \cdot \arccos\left(\frac{5x}{11}\right) - x$	$\frac{\pi}{4}$; 2; 0; 2
28	$\frac{\pi x - 3}{(x-1)^2}$	$\frac{\pi}{4}$; -10; 0; 0,9
29	$\pi x e^{-2x} + x^2$	$\frac{\pi}{4}$; 1; 0; 1,5
30	$\pi \sin 3x + x \cdot \cos 3x$	$\frac{\pi}{4}$; 1; 0; 1,5
31	$\pi \ln\left(1 + x - 2x^2\right)$	$\frac{\pi}{4}$; 0,25; 0; 0,75
32	$\ln\left(x + \sqrt{1 + x^2}\right)$	$\frac{\pi}{4}$; 4; 0; 15
33	$\sin^2 x \cdot \cos^2 x$	$\frac{\pi}{4}$; 0,5; 0; 1,5

ПРИЛОЖЕНИЕ 2

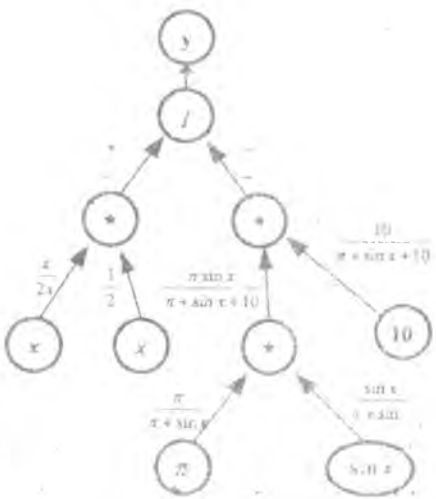
Примеры оформления заданий для самостоятельных работ

1. *В предположении, что аргумент функции* $y = \dfrac{x^2}{10\,\pi \sin x}$ *и ее коэффициенты заданы неточно, оценить величину абсолютной и относительной погрешностей. Предполагается, что относительные погрешности арифметических операций и представления чисел равны* $0.5*10^{1}$, *где* t *- количество значащих цифр в разрядной сетке ЭВМ. Построить граф вычислительного процесса. Вычислить функцию с пятью верными значащими цифрами в точке* $x0=2.15$.

Решение.

Положим $\delta_x = \delta_* = \delta_/ = \delta$ - относительные погрешности представления чисел в ЭВМ и соответствующих арифметических операций; $\delta_{\sin}$ - относительная погрешность вычисления функции.

Вычислительный процесс представим графом вычислительного процесса:



По формуле (1.17) вычислим относительную погрешность вычисления $\sin(x)$:

$$\delta_{\sin x} = (\ln \sin x)' \Delta x = \frac{\cos x}{\sin x} \Delta x = \frac{x \cos x}{\sin x} \delta_x .$$

Используя граф вычислительного процесса, выведем формулу для относительной погрешности вычисления функции на ЭВМ. При этом положим, что $\delta_{10} = 0$:

$$|\delta_y| = 1 \cdot (\tfrac{1}{2}|\delta_x| + \tfrac{1}{2}|\delta_x| + \delta_{*1}) +$$
$$+ 1 \cdot (|\frac{x \sin x}{x+\sin x+10}| (|\frac{x}{x+\sin x}| \delta_\pi + |\frac{\sin x}{x+\sin x}| \delta_\pi + \delta_{*2}) + \delta_{*3}) + \delta_{/1};$$

приведя подобные члены, имеем

$$|\delta_y| = (1 + |\frac{x \sin 2x}{2(x+\sin x+10)(x+\sin x)}|)\delta_x + |\frac{x \sin x}{(x+\sin x+10)(x+\sin x)}| \delta_\pi +$$
$$+ (3 + |\frac{x \sin x}{(x+\sin x+10)}|)\delta. \qquad (П.1)$$

Если вычисления производятся на ЭВМ и число π задано с "машинной" точностью, то $\delta_x = \delta_\pi \approx \delta = 0.5\ 10^{-t}$ и в точке x=2.15 при t=15 имеем $|\delta_y| = 2.14\ 10^{-14}$, а $|\Delta y| = |f(x)| |\delta_y| = 3.7\ 10^{-15}$.

Вычисление функции с заданным числом верных значащих цифр несколько отличается от оценки погрешностей вычисления функции на ЭВМ.

Говорят, что число x задано с *n верными значащими цифрами*, если абсолютная погрешность этого числа не превышает половины единицы разряда, выражаемого n-й значащей цифрой, считая слева направо. Для относительной погрешности представления приближенного числа число верных значащих цифр определяет параметр t.

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

В нашем случае t=5, а функция имеет два аргумента x и π. При "ручном" способе вычисления функции погрешности арифметических операций не учитываются, т.е. в формуле (П.1) можно положить $\delta = 0$.

Простейшее решение обратной задачи основывается на принципе "равных влияний". Согласно этому принципу предполагается, что все погрешности одинаково влияют на образование общей относительной погрешности:

$K_1 \delta_x = K_2 \delta_\pi = \frac{\delta_y}{2}$, отсюда

$$\delta_x = \frac{\delta_y}{2K_1}, \quad \delta_\pi = \frac{\delta_y}{2K_2},$$

где

$$K_1 = 1 + \left| \frac{x \sin 2x}{2(x+\sin x+10)(x+\sin x)} \right|,$$

$$K_2 = \left| \frac{x \sin x}{(x+\sin x+10)(x+\sin x)} \right|$$

Исходя из условия задачи $|\delta_y| < 0.510^{-5}$ имеем

$|\delta_x| \approx 0.24410^{-7}$,

$|\delta_\pi| \approx 0.510^{-6}$

Полученный результат означает, что число "пи" следует "брать" с шестью значащими цифрами, аргумент функции x с семью. Исходное значение аргумента - 2.15 , содержащее три верных значащих цифры в этом смысле задано точно. Итак, x=2.15, π=3.14159, а по результатам расчетов y=0.17581.

2. *Разложить по формуле Тейлора функцию* $y = \frac{x^2}{10\pi \sin x}$

Выбрать точку разложения. Определить радиус сходимости построенного ряда. Оценить величину погрешностей $\varepsilon_{мет}$ *и* $\varepsilon_{окр}$ *(методической ошибки и ошибки округления). Определить число членов ряда, необходимое для вычисления функции в точке x=1.5 с точностью* $\varepsilon_n = \varepsilon_{мет} + \varepsilon_{окр} = 0.5 * 10^{-5}$.

Решение

Поскольку ЭВМ способна непосредственно вычислять только функции, содержащие арифметические операции (полиномы, дробно-рациональные функции, полиномиальные сплайны и т.д.), то большое количество элементарных функций реализуется на вычислительной машине путем замены исходной функции ее приближенным аналогом. Обычно используются полиномиальные функции: $f(x) = P_n(x) + \varepsilon_n$. Если полная погрешность вычисления на заданном интервале не превышает заданную величину, то приближение исходной функции считается удовлетворительным.

В нашем примере с помощью арифметических операций невозможно вычислить только функцию sin(x). Разложим функцию sin(x) в ряд Тейлора. Из курса математического анализа известно, что в окрестностях точки разложения $x_0 = 0$ функция sin(x) раскладывается в ряд Тейлора следующим образом:

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \ldots + R_n(x),$$

где $R_n(x)$ - остаточный член формулы Тейлора.

Замечание. При разложении более сложных функций в ряд Тейлора необходимо воспользоваться стандартной методикой. При этом точку разложения выбирать из соображений простоты вычисления коэффициентов формулы Тейлора и принадлежности точки разложения интервалу вычисления функции. Радиус сходимости разложения можно определить с помощью признака Даламбера.В отдельных случаях помогают признаки сходимости Коши и Лейбница.

В нашем примере остаточный член может быть оценен по формуле

$$|R_n(x)| = \frac{\left|f^{(n+1)}(\xi)\right|}{(n+1)!}|x|^{n+1} \le \frac{\left|(\sin x)^{(n+1)}\right|}{(n+1)!}|x|^{n+1} < \frac{|x|^{n+1}}{(n+1)!}.$$

Определимся с интервалом разложения функции. Исходная функция является нечетной функцией, поскольку $f(x) = -f(-x)$, кроме того, она терпит разрывы второго рода в точках, где sin(x)=0. На рисунке П.1 представлен график функции, построенный с помощью электронной таблицы Excel.

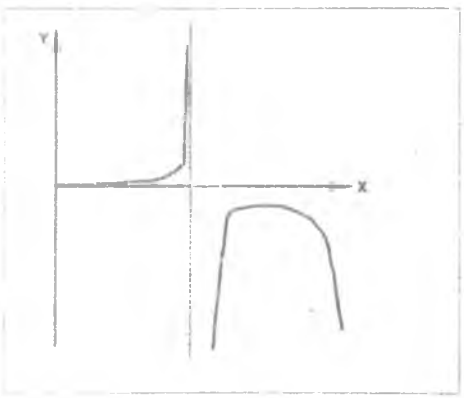


Рис. П.1.

В точке x=0 функция терпит устранимый разрыв первого рода, поскольку $\lim\limits_{x\to 0}\frac{x^2}{10\pi\sin x} = \frac{1}{10\pi}(\lim\limits_{x\to 0} x)(\lim\limits_{x\to 0}\frac{x}{\sin x}) = 0$.

Будем считать, что вычисления производятся на центральном участке непрерывности функции $(-\pi, \pi)$. Учитывая центральную

симметрию функции, рассмотрим интервал $(0, \pi)$. Рассмотрим, каким образом изменяется оценка остаточного члена формулы Тейлора при n=5 (см. рисунок П.2).

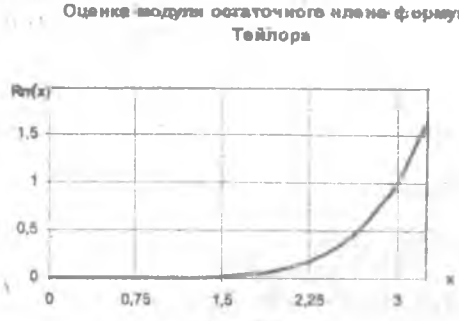


Рис. П.2.

Из графика видно, что при удалении x от точки разложения величина остаточного члена резко увеличивается. Установим связь между полной ошибкой приближенного представления вычисляемой функции и остаточным членом разложения sin(x) по формуле Тейлора. При замене sin(x) имеем

$$y = \frac{x^2}{10\pi(P_n(x) + R_n(x))} = \frac{x^2}{10\pi P_n(x)} + \varepsilon_{мет}.$$

Методическая ошибка в связи с заменой синуса полиномиальной функцией содержится в знаменателе вычисляемой функции, следовательно, по формуле оценки ошибки от деления получим

$$e_y = \frac{e_{x^2}}{10\pi \sin x} - \frac{x^2}{(10\pi \sin x)^2} 10\pi R_n(x),$$

учитывая, что $e_{x^2} = 0$, имеем

$$|e_{мет}| = \frac{x^2}{(10\pi \sin x)^2} 10\pi |R_n(x)|.$$

Если не учитывать погрешности округления, то для оценки числа членов ряда разложения функции sin(x), обеспечивающие заданную точность вычисления функции, можно воспользоваться неравенством

$$|\varepsilon_{мет}| = \frac{x^2}{(10\pi \sin x)^2} 10\pi |R_n(x)| = \frac{x^2 |x|^{n+1}}{10\pi (\sin x)^2 (n+1)!} < \quad (П.2)$$

$$< \begin{cases} f(x) \cdot 0.5\,10^{-5}, \text{если } |f(x)| > 1 \\ 0.5\,10^{-5}, \quad \text{если } |f(x)| < 1. \end{cases}$$

В последней формуле учитывается, что в условии задачи задана величина относительной погрешности вычисления функции.

Из рисунка П.2 видно, что чем ближе x к точке разрыва, тем больше величина методической погрешности, поэтому аргумент функции выберем достаточно близко к π, однако на некотором разумном удалении, поскольку в окрестностях точки разрыва следует применять иные методы приближения, например, дробно-рациональные функции. Пусть x=3.

Таблица П2

n	εмет
10	0,0638401
11	0,01596002
12	0,00368308
13	0,00078923
14	0,00015785
15	2,9596E-05
16	5,2229E-06

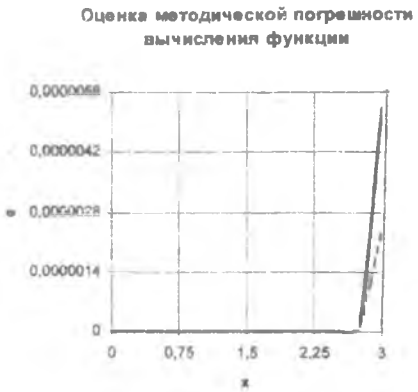


Рис. П.3

Из таблицы П.2 видно, что для достижения заданной точности вычисления функции в интервале (0, 3) необходимо в формуле Тейлора взять 16 членов ряда разложения функции sin(x).На рисунке П.3 жирной линией показана мажорантная оценка методической погрешности вычисления заданной функции при n=16, пунктирной линией - величина истинной погрешности.

При вычислении приближенной функции для n=16 потребуется порядка 132 арифметических операций. Проверим, не окажет ли этот фактор влияния на полную погрешность вычисления функции?

Основную долю погрешностей арифметических операций вносит приближенное вычисление функции $\sin(x)$. В общем случае представим формулу Тейлора многочленом общего вида:

$$P_n(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n = y_0 + y_1 + y_2 + \ldots + y_n.$$

Построим граф вычислительного процесса для вычисления $y_k = a_k \cdot \underbrace{x \cdot x \cdot \ldots \cdot x}_{k}$ (см. рисунок П.4, а)).

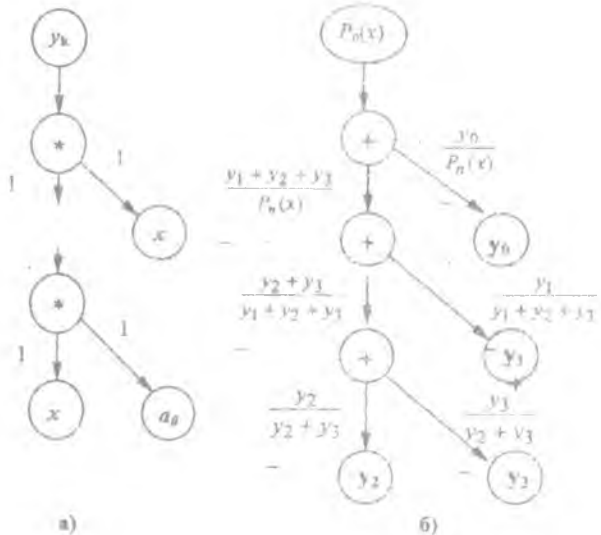


Рис. П.4

Тогда для δ_{y_k} имеем

$$\delta_{y_k} = (\ldots((\delta_x + \delta_{a_k} + \delta_{*1}) + \delta_x + \delta_{*2}) + \ldots + \delta_x + \delta_{*k} =$$

$$= k\delta_x + \delta_{a_k} + k\delta_* = k(\delta_x + \delta_*) + \delta_{a_k}$$

Используя граф вычислительного процесса (рис. П.4, б), выведем формулу для оценки относительной погрешности многочлена $P_3(x)$:

$$\delta_{P_3} = \frac{y_1 + y_2 + y_3}{P_3}\left(\frac{y_2 + y_3}{y_1 + y_2 + y_3}\left(\frac{y_2}{y_2 + y_3}\delta_{y_2} + \frac{y_3}{y_2 + y_3}\delta_{y_3} + \delta_{+1}\right) + \right.$$

$$\left. + \frac{y_1}{y_1 + y_2 + y_3}\delta_{y_1} + \delta_{+2}\right) + \frac{y_0}{P_3}\delta_{y_0} + \delta_{+3} =$$

$$= \frac{y_0}{P_3}\delta_{y_0} + \frac{y_1}{P_3}\delta_{y_1} + \frac{y_2}{P_3}\delta_{y_2} + \frac{y_3}{P_3}\delta_{y_3} + \frac{\delta_+}{P_3}(P_3 + y_1 + 2y_2 + 2y_3).$$

Модуль абсолютной погрешности можно оценить сверху величиной

$$|e_{P_3}| = |P_3\delta_{P_3}| < |y_0|\delta_{y_0}| + |y_1|\delta_{y_1}| + |y_2|\delta_{y_2}| + |y_3|\delta_{y_3}| + |\delta_+|(|P_3| + |y_1| + 2|y_2| + 2|y_3|) <$$
$$< |y_0|\delta_{y_0}| + |y_1|\delta_{y_1}| + |y_2|\delta_{y_2}| + |y_3|\delta_{y_3}| + |\delta_+|(|P_3| + 4\bar{y}),$$

где $\bar{y} = (|y_0| + |y_1| + |y_2| + |y_3|)/4$.

Подставив выражения для δ_{y_i}, получим

$$|e_{P_3}| < |y_0|\delta_{a_0}| + |y_1|\delta_{a_1} + \delta_x + \delta_*| + |y_2|\delta_{a_2} + 2(\delta_x + \delta_*)| + +|y_3|\delta_{a_3} + 3(\delta_x + \delta_*)| + 4\cdot 3\bar{y}|\delta_+| <$$
$$< 4\bar{y}|\delta_a| + (|y_1| + 2|y_2| + 3|y_3|)|\delta_x| + (|y_1| + 2|y_2| + 3|y_3|)|\delta_*| <$$
$$< 4\bar{y}|\delta_a| + 3\cdot 4\bar{y}|\delta_x| + 3\cdot 4\bar{y}|\delta_*| + 4\cdot 3\bar{y}|\delta_+| < 3\cdot 4\bar{y}\delta(\frac{1}{3} + 3) \approx 3\cdot 4\bar{y}\delta\cdot 3.$$

В данной формуле были сделаны упрощающие предположения о том, что $\delta_x = \delta_+ = \delta_a = \delta_* = \delta$. Применяя метод математической индукции, можно показать, что

$$|e_{P_n}| < 3n(n-1)\bar{y}\delta.$$

В примере при x=3, n=16, t=15 имеем $\bar{y} = 0.626$, $|\varepsilon_{окр}| = |e_{P_{16}}| = 2.2510^{-13}$. Из чего следует, что $\varepsilon_{окр} << \varepsilon_{мет}$ и $\varepsilon_n \approx \varepsilon_{мет} \approx 5.2210^{-6}$.

3. *Для полученного разложения функции f(x) в ряд Тейлора произвести экономизацию степенного ряда, который вычислял бы функцию f(x) с точностью* $\varepsilon_n < 0.5*10^{-3}$.

Решение

Экономизация ряда заключается в уменьшении числа арифметических операций при условии сохранения заданной точности вычисления функции. Данный прием использует свойство полиномов Чебышева сводить к минимуму максимальную ошибку приближения. Экономизация основывается на изменении структуры ряда в сторону увеличения сходимости, при этом в отдельных случаях удается уменьшить число членов ряда.

В нашем примере наибольшая погрешность возникает на правой границе интервала приближения функции, с другой стороны, полиномы Чебышева действуют на интервале (-1, 1), поэтому вычисления будем производить для x=0,99.

Замечание. Если вычисление необходимо произвести на более широком интервале (a, b), то предварительно следует произвести линейное преобразование системы координат, т.е. ввести новую переменную

$\tilde{x} = \frac{2x - b - a}{b - a}$, которая принадлежит интервалу (-1, 1), а все вычисления производить для функции $f(\tilde{x})$.

Из формулы П.2 и условий задачи оценим методическую погрешность,с которой необходимо вычислять функцию sin(x):

$$|R_n(x)| < \frac{10\pi \sin^2 x}{x^2} 0.5\,10^{-3}$$

При x=0.99 имеем $|R_n(x)| < 0.0112$.

Метод экономизации основывается на замене степенных функций x^n разложениями по полиномам Чебышева, например, $x^3 = \frac{1}{2^2}(3T_1(x) + T_3(x))$. Можно показать, что при таких заменах коэффициент при старшем члене степенной функции будет равен $\frac{a_n}{2^{n-1}} T_n(x)$, где a_n- коэффициент при старшем члене степенной функции в формуле Тейлора. Если для знакопеременного степенного ряда $\left|\frac{a_n}{2^{n-1}} T_n(x)\right| < |R_n(x)|$, член $b_n = \frac{a_n}{2^{n-1}} T_n(x)$ можно отбросить.

Таблица П.3

n	bn	Rn(x)
2	0,25	0,1617165
3	0,0416667	0,0400248
4	0,0052083	0,0079249
5	0,0005208	0,0013076
6	4,34E-05	0,0001849
7	3,1E-06	2,289E-05
8	1,938E-07	2,517E-06
9	1,076E-08	2,492E-07

В таблице П.3 верхние оценки остаточного члена в формуле Тейлора при x=0,99 и оценки коэффициентов при старших членах экономизированного ряда. Из таблицы видно, что для функции sin(x) экономизация возможна начиная n=5 на один член ряда. В нашем случае при $|R_n(x)| < 0.0112$ экономизация невозможна.

Тем не менее, покажем технику экономизации в тех случаях, если это возможно. Пусть n=5. В этом случае при x=0.99 $|R_n(x)| < 0.00137$.

Для $\sin x \approx x - \frac{x^3}{3!} + \frac{x^5}{5!}$, произведя замену $x = T_1$, $x^3 = \frac{1}{4}(3T_1 + T_3)$, $x^5 = \frac{1}{16}(10T_1 + 5T_3 + T_5)$, после приведения подобных членов получим

$\sin x \approx 0.8802087_1 - 0.03906T_3 + 0.000521T_5$. Отбросив последний член ряда, имеем

$\sin x \approx 0.8802087_1 - 0.03906T_3$.

Степень последнего многочлена на два порядка меньше, чем в формуле Тейлора, и в то же время удовлетворяет заданной точности вычисления. На рис.П.5 показана абсолютная погрешность вычисления функции sin(x) с помощью экономизированного ряда.

Рис. П.5

4. *Произвести табуляцию функции f(x) на [a,b]. Шаг табуляции функции выбрать из соображений достижения заданной точности для интерполяционного многочлена Лагранжа не выше третьего порядка. На любом участке отрезка построить интерполяционный многочлен, вычисляющий функцию* $\forall x \in [a,b]$ *с 4 верными значащими цифрами.*

Решение.

Если интерполяционная сетка заранее не задана, то будем считать, что задачей интерполирования функции является составление таблицы для исходной функции. Табулирование функций заключается в построении такой таблицы, чтобы восстановление исходной функции $f(x)$ с помощью вспомогательной функции $\varphi(x)$ было бы возможно с заданной наперед точностью. При построении табличных функций шаг табулирования h подбирают таким образом, чтобы на любом участке таблицы имелась возможность обеспечить заданную точность вычисления функции с помощью полиномов Лагранжа не выше третьей степени.

В задании указано, что вычисления производить с точностью до четырех значащих цифр, т.е. $|\delta_{мет}| < 0.5\,10^{-4}$ или $|e_{мет}| = |R_n(x)| < f(x) \cdot 0.5\,10^{-4}$. Для формулы Лагранжа (см. п.2.8)

$$|R_n(x)| \le \frac{M_{n+1}}{(n+1)!}|\Pi_{n+1}(x)|,$$

где $M_{n+1} = \max\limits_{x_0 \le x \le x_n} |f^{(n+1)}(x)|$,

$$\Pi_{n+1}(x) = (x - x_0)\cdot(x - x_1)\ldots(x - x_n)$$

Из формул видно, что методическая погрешность существенно зависит от (n+1)-й производной функции. Определим первые три производные исходной функции:

$$y' = \frac{1}{10\pi}\left(2x\frac{1}{\sin x} + x^2\left(\frac{1}{\sin x}\right)'\right),$$

$$y'' = \frac{1}{10\pi}\left(2\frac{1}{\sin x} + 4x\left(\frac{1}{\sin x}\right)' + x^2\left(\frac{1}{\sin x}\right)''\right),$$

$$y''' = \frac{1}{10\pi}\left(6\left(\frac{1}{\sin x}\right)' + 6x\left(\frac{1}{\sin x}\right)'' + x^2\left(\frac{1}{\sin x}\right)'''\right),$$

где

$$\left(\frac{1}{\sin x}\right)' = -\frac{\cos x}{\sin^2 x},$$

$$\left(\frac{1}{\sin x}\right)'' = \frac{1+\cos^2 x}{\sin^3 x},$$

$$\left(\frac{1}{\sin x}\right)''' = -\frac{\cos x(5+\cos^2 x)}{\sin^4 x}.$$

На рисунке П.6 показаны графики первых трех производных

Рис. П.6.

функции. Из рисунка видно, что в окрестностях точки разрыва, производные функции резко возрастают, а следовательно уменьшается точность интерполирования функции. В таких случаях целесообразно воспользоваться дробно-рациональными функциями. Для полиномиальных интерполяционных формул рассмотрим такой отрезок числовой оси, где производные функции принимают по абсолютной величине не слишком большие значения, например, отрезок [1.5; 3]. В окрестностях нуля [0; 1.5] все три производные близки к нулю и здесь не возникает проблемы с интерполированием функции.

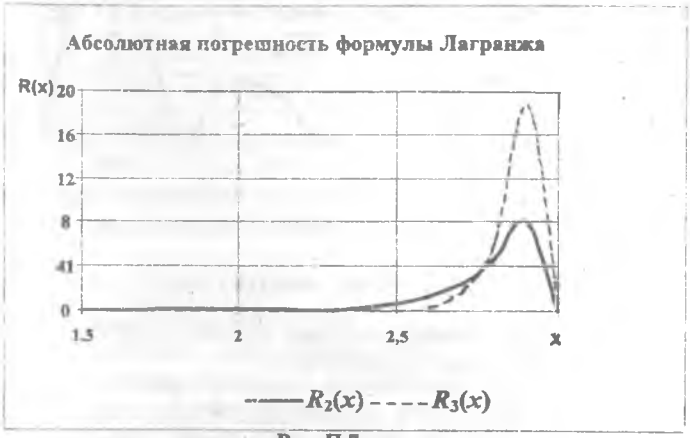


Рис. П.7

На рисунке П.7 показана зависимость погрешностей интерполяционной формулы Лагранжа для полиномов второго и третьего порядка. Видно, что около точки x=3 полином третьего порядка описывает исходную функцию даже хуже, чем полином второго порядка. Последнее связано с близостью точки разрыва второго рода, поэтому сузим отрезок интерполирования до [1,5; 2,5]. Выберем шаг интерполяции до достижения заданной точности для полинома Лагранжа третьего порядка. Для этого зафиксируем правую точку отрезка интерполирования, где абсолютная величина погрешности формулы Лагранжа принимает наибольшее значение. Пусть $x_3 = 2.5$; $\bar{x} = x_3 - h/2$, h выберем из таблицы, описывающей зависимость $R_3(\bar{x})$ от h. Как видно из таблицы, при h=0,08 достигается заданная точность.

h	$R_3(\bar{x})$
0.2	0,001349
0,14	0,000404
0,08	5,43E-05
0,02	2,7E-07

Таким образом, поставленная задача решается табуляцией исходной функции с шагом h=0.08. На рисунке П.8 показана зависимость абсолютной погрешности формулы Лагранжа третьего порядка на правой границе отрезка табуляции.

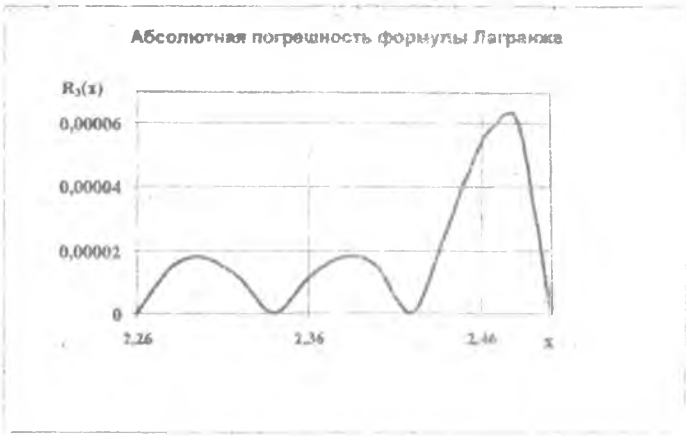
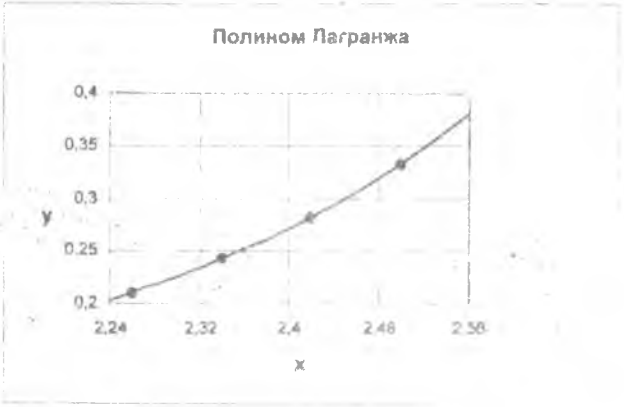


Рис. П.8

На данном отрезке построим интерполяционный многочлен Лагранжа, для этого составим таблицу:

Xi	2,26	2,34	2,42	2,5
Yi	0,210663	0,242592	0,282198	0,3324193

Используя интерполяционную формулу Лагранжа и возможности электронной таблицы Excel, построим график интерполяционной функции.



5. Используя интерполяционную сетку полинома Лагранжа, построить интерполяционный многочлен Ньютона.

Решение.

Пусть задана интерполяционная сетка:

Xi	2,26	2,34	2,42	2,5
Yi	0,210663	0,242592	0,282198	0,3324193

На ее основе построим таблицу конечных разностей:

x	y	Δy_i	$\Delta^2 y_i$	$\Delta^3 y_i$
2,26	*0,210663*	*0,031929*	*0,007678*	*0,002936*
2,34	0,242592	0,039607	0,010614	
2,42	0,282198	0,050221		
2,5	0,332419			

Используя полученную таблицу и первую интерполяционную формулу Ньютона, вычислим коэффициенты многочлена (см. п.2.9):

$$P_3(x) = 0.210663 + 0.399108\cdot(x-2.26) + 0.599836\cdot(x-2.26)(x-2.34) + 0.955853\cdot(x-2.26)(x-2.34)(x-2.42).$$

Для одинаковых интерполяционных сеток полиномы Лагранжа и Ньютона ничем не отличаются друг от друга (с точностью до погрешности округления), которая зависит от числа операций. Поэтому график многочлена Ньютона совпадает с графиком многочлена Лагранжа. В данной задаче оценим величину

Рис. П.9.

истинной погрешности интерполирования т.е $e_{мет} = |f(x) - P_3(x)|$. На рисунке П.9 представлена эта зависимость, как видно из рисунка на отрезке интерполирования функции действительно обеспечивается заданная точность, однако за пределами отрезка ошибка резко возрастает.

6. *Для табличной функции, полученной в задании 5, построить:*

- *линейный интерполяционный сплайн;*
- *параболический сплайн;*
- *кубический сплайн.*

Указание: при построении параболического и кубического сплайнов в качестве дополнительных условий использовать $f'(a)=0$ и $f'(a)=f'(b)=0$. Сравнить величины абсолютных погрешностей при интерполирования функции $f(x)$ сплайнами и полиномом между собой. Построить графики ошибок интерполирования функций.

Решение.

Пусть задана табличная функция:

Xi	2,26	2,34	2,42	2,5
Yi	0,210663	0,242592	0,282198	0,3324193

В соответствии с методикой п.2.11 составим систему линейных уравнений для вычисления коэффициентов линейного сплайна:

$$S_1(x) = y_0 + C_0(x - x_0) + C_1(x - x_1)_+ + C_2(x - x_2)_+$$

расширенная матрица которой имеет вид

$$A = \left[\begin{array}{ccc|c} (x_1 - x_0) & 0 & 0 & y_1 - y_0 \\ (x_2 - x_0) & (x_2 - x_1) & 0 & y_2 - y_0 \\ (x_3 - x_0) & (x_3 - x_1) & (x_3 - x_2) & y_3 - y_0 \end{array}\right],$$

или в нашем случае

$$A = \left[\begin{array}{ccc|c} 0.08 & 0 & 0 & 0.0319 \\ 0.16 & 0.08 & 0 & 0.0715 \\ 0.24 & 0.16 & 0.08 & 0.1219 \end{array}\right].$$

Решив систему уравнений получим, C_0=0.3991, C_1=0.09597, C_2=0.13267, откуда

$$S_1(x) = 0.21066 + 0.3991(x - 2.26) + 0.09597(x - 2.34)_+ + 0.1367(x - 2.42)_+.$$

Построим параболический сплайн вида

$$S_2(x) = y_0 + u(x - x_0) + C_0(x - x_0)^2 + C_1(x - x_1)_+^2 + C_2(x - x_2)_+^2.$$

Для данного типа сплайна число неизвестных коэффициентов на единицу больше, чем число условий интерполяции. Необходимо одно дополнительное условие. В качестве такового возьмем $f'(a)=0$:

$$S_2'(x) = u + 2C_0(x-x_0) + 2C_1(x-x_1)_+ + 2C_2(x-x_2)_+,$$

$$S_2''(x) = 2C_0 + 2C_{1+} + 2C_{2+}.$$

В точке $x=x_0$ имеем $S_2''(x_0) = 2C_0 = 0$. Таким образом, расширенная матрица системы линейных уравнений имеет вид

$$A = \left[\begin{array}{cccc|c} 0 & 2 & 0 & 0 & 0 \\ (x_1-x_0) & (x_1-x_0)^2 & 0 & 0 & y_1-y_0 \\ (x_2-x_0) & (x_2-x_0)^2 & (x_2-x_1)^2 & 0 & y_2-y_0 \\ (x_3-x_0) & x_3-x_0)^2 & (x_3-x_1)^2 & (x_3-x_2)^2 & y_3-y_0 \end{array}\right].$$

Решив систему уравнений, получим коэффициенты параболического сплайна: u=0.3991, C_0=0, C_1=1.1996, C_2=-0.7408.

Кубический сплайн

$$S_3(x) = y_0 + u(x-x_0) + C_0(x-x_0)^2 + D_0(x-x_0)^3 + D_1(x-x_1)_+^3 + D_2(x-x_2)_+^3$$

строится аналогично с учетом того факта, что в данном случае используются два дополнительных условия: равенство нулю вторых производных на концах отрезка:

$$S_3'(x) = u + 2C_0(x-x_0) + 3D_0(x-x_0)^2 + 3D_1(x-x_1)_+^2 + 3D_2(x-x_2)_+^2,$$

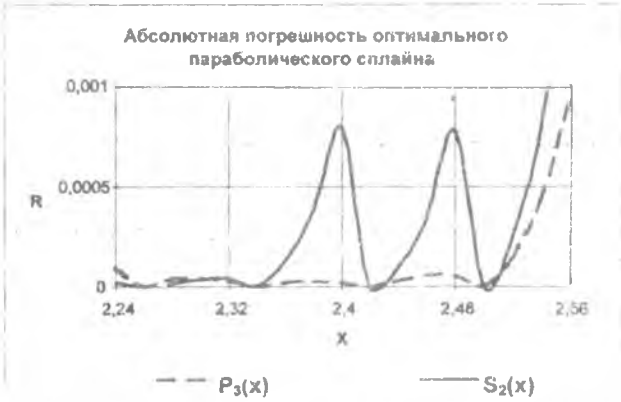
$$S_3''(x) = 2C_0 + 6D_0(x-x_0) + 6D_1(x-x_1)_+ + 6D_2(x-x_2)_+.$$

Рис. П.10.

На левой границе имеем $S_3''(x_0) = 2C_0 = 0$, откуда $C_0=0$ и этот коэффициент можно исключить из системы уравнений. С учетом дополнительного условия на правой границе имеем следующую расширенную матрицу:

$$A = \left[\begin{array}{cccc|c} (x_1-x_0) & (x_1-x_0)^3 & 0 & 0 & y_1-y_0 \\ (x_2-x_0) & (x_2-x_0)^3 & (x_2-x_1)^3 & 0 & y_2-y_0 \\ (x_3-x_0) & x_3-x_0)^3 & (x_3-x_1)^3 & (x_3-x_2)^3 & y_3-y_0 \\ 0 & 6(x_3-x_0) & 6(x_3-x_1) & 6(x_3-x_2) & 0 \end{array}\right].$$

Построим график погрешностей построенных интерполяционных сплайнов (см. рис. П.10). Как видно из рисунка, все интерполяционные сплайны дают более плохой результат по сравнению с интерполяционным многочленом. Более того, параболический сплайн интерполирует функцию даже хуже, чем линейный сплайн. Причина заключается в некорректности задания дополнительных граничных условий. Методом подбора коэффициента C_0 для параболического сплайна для нашего примера удалось подобрать такое значение коэффициента, для которого величина абсолютной погрешности принимает наименьшее значение.



7. Протабулировать функцию $f(x)$ в 16 точках, начиная с $x_0 = a$ с шагом h. На вычисленные значения y_i наложить случайную погрешность ε_i ($\varepsilon_i \le 10\%\ \delta_y$). Выбрать рациональный вид аппроксимирующей функции и аппроксимировать

экспериментальную зависимость многочленом. Оценить погрешность аппроксимации Построить график.

Решение

Составим таблицу функции с ошибками измерения:

x	f	f+e	e
0	0	0	0
0,2	0,006409	0,006242	-0,00017
0,4	0,013078	0,012398	-0,00068
0,6	0,020295	0,018549	-0,00175
0,8	0,028398	0,029989	0,00159
1	0,037828	0,040703	0,00287
1,2	0,049179	0,049867	0,00069
1,4	0,06331	0,057232	-0,00608
1,6	0,081522	0,081359	-0,00016
1,8	0,105902	0,102937	-0,00297
2	0,140025	0,136944	-0,00308
2,2	0,190554	0,185218	-0,00534
2,4	0,271436	0,281753	0,01031
2,6	0,417414	0,447468	0,03005
2,8	0,744967	0,803074	0,05811
3	2,030037	1,835154	-0,19488

В качестве полинома аппроксимации рассмотрим квадратный трехчлен:

$$P_2(x) = a + bx + cx^2.$$

По формулам (2.43) составим систему уравнений

$$\begin{cases} 16a + (\sum x_i)b + (\sum x_i^2)c = \sum y_i, \\ (\sum x_i)a + (\sum x_i^2)b + (\sum x_i^3)c = \sum y_i x_i, \\ (\sum x_i^2)a + (\sum x_i^3)b + (\sum x_i^4)c = \sum y_i x_i^2. \end{cases}$$

При использовании электронной таблицы Excel для подсчета коэффициентов системы линейных уравнений удобно строить таблицы вида:

x	y	x^2	x^3	x^4	yx	yx^2
0	0	0	0	0	0	0
0,2	0,006242	0,04	0,008	0,0016	0,001248	0,00025
0,4	0,012398	0,16	0,064	0,0256	0,004959	0,001984
0,6	0,018549	0,36	0,216	0,1296	0,01113	0,006678
0,8	0,029988	0,64	0,512	0,4096	0,023991	0,019193
1	0,040702	1	1	1	0,040703	0,040703
1,2	0,049867	1,44	1,728	2,0736	0,059841	0,071809
1,4	0,057232	1,96	2,744	3,8416	0,080125	0,112175
1,6	0,081359	2,56	4,096	6,5536	0,130174	0,208279
1,8	0,1029367	3,24	5,832	10,4976	0,185286	0,333515
2	0,136944	4	8	16	0,273888	0,547776
2,2	0,1852182	4,84	10,648	23,4256	0,40748	0,896456
2,4	0,2817528	5,76	13,824	33,1776	0,676207	1,622896
2,6	0,4474678	6,76	17,576	45,6976	1,163416	3,024882
2,8	0,8030739	7,84	21,952	61,4656	2,248607	6,296099
3	1,8351538	9	27	81	5,505461	16,51638
Σ 24	4,9888871	49,6	115,2	285,2992	10,81252	29,69908

Решив систему уравнений, получим
a=0,196878; b=-0,63611; c=0,326721.

На рис.П.11 показаны результаты аппроксимации исходной функции на отрезке [0; 3]. Как видно из рисунка, аппроксимация исходной функции $f(x)+e$ параболой не дает удовлетворительного результата. Последнее связано с близостью точки разрыва второго рода. Даже использование в качестве аппроксиманта кубической параболы принципиально ничего не меняет (см. рис.П.11).

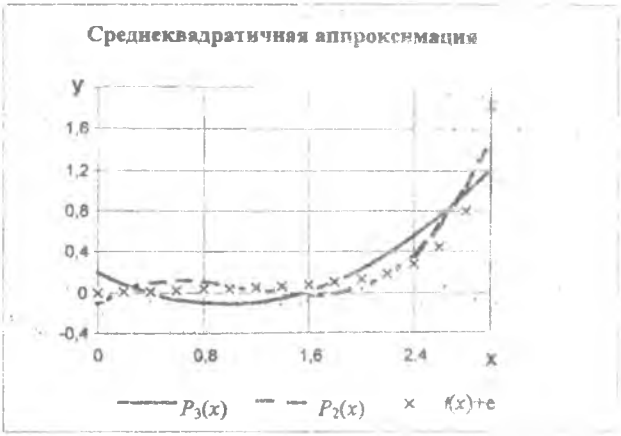


Рис. П.11

Если исключить две последние точки таблицы "экспериментальных" данных (x_{15}=2,8; x_{16}=3), то результаты аппроксимации существенно изменятся (см. рис.П.12).

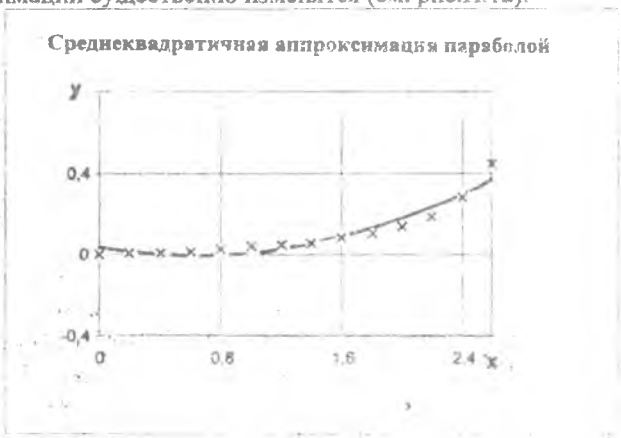
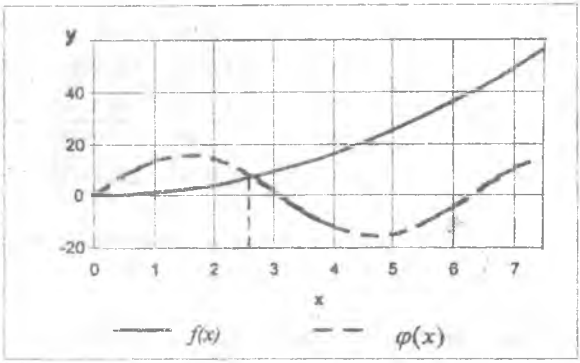


Рис. П.12

8. Найти хотя бы один корень уравнения $f(x)=c$ с точностью до 5- значащих цифр. Исследовать сходимость выбранного численного метода. Для определения начального приближения использовать графический метод решения задачи.

Решение.

Пусть задано уравнение $\frac{x^2}{10\pi \sin x} = 0.5$, представим его в общем виде $F(x) = \frac{x^2}{10\pi \sin x} - 0.5 = 0$.



Графическим методом произведем отделение корней уравнения. Действительные корни уравнения можно определить как абсциссы функции $y = F(x)$. На практике, при графическом способе отделения корней, часто бывает выгодно использовать равносильное уравнение $f(x) = \varphi(x)$. Например, в нашем случае, для того чтобы избавиться от особенностей функции на этапе отделения корней, целесообразно рассматривать уравнение: $x^2 = 0.5 \cdot 10 \cdot \pi \cdot \sin x$. Как видно из рисунка, при $x>=0$ функция имеет два коня $x = 0$ и $x \approx 2.5$.

При выполнении задания можно использовать любой из описанных в пособии метод решения нелинейного уравнения. Воспользуемся методом Ньютона-Рафсона. Предварительно выведем формулу для $F'(x)$:

$$F'(x) = \frac{2x \sin x - x^2 \cos x}{10\pi \sin^2 x}.$$

Тогда итерационная формула метода Ньютона-Рафсона имеет вид

$$x_{n+1} = x_n - \frac{x_n^2}{10\pi \sin x_n} \cdot \frac{10\pi \sin^2 x_n}{2x_n \sin x_n - x_n^2 \cos x_n} = x_n - \frac{x_n^2 \sin x_n}{2x_n \sin x_n - x_n^2 \cos x_n}$$

Момент завершения вычислений можно определить по формуле $\frac{|x_{n+1} - x_n|}{|x_n|} \le \delta_x = 0{,}5\,10^{-5}$. В качестве начального приближения возьмем точку x_0=2,5. В таблице приведены результаты вычислений:

x_n	$F(x_n)$	$F'(x_n)$	x_{n+1}	$\delta_x(x_n)$
2,5	-0,16758	0,710928	2,735721	0,086164
2,735721	0,103386	1,845215	2,679692	0,020909
2,679692	0,012892	1,413078	2,670569	0,003416
2,670569	0,000258	1,35698	2,670378	7,13E-05
2,670378	1,08E-07	1,355844	2,670378	2,99E-08

Таким образом, одним из решений уравнения с точностью до пяти значащих цифр является точка x=2.670378.

9. *Найти приближенное решение системы нелинейных уравнений*

$$\begin{cases} y^2 + f(x) = c \\ x^2 - y = 0 \end{cases}$$

с точностью до 5 значащих цифр. Для определения начального приближения использовать графический метод решения задачи.

Решение.

Пусть задана система нелинейных уравнений:

$$\begin{cases} y^2 + \dfrac{x^2}{10\pi \sin x} - 0.5 = 0, \\ x^2 - y = 0. \end{cases}$$

Построим графики функций $f_1(x,y) = 0$ и $f_2(x,y)$, в нашем случае это $y = \pm\sqrt{0.5 - x^2/(10\pi \sin x)}$; $y = x^2$ (см. рисунок П.13). Как видно из рисунка, при x>0 система уравнений имеет по крайней мере, один корень (0,8; 0,5). Для решения системы уравнений воспользуемся более эффективным методом Ньютона:

$$X^{n+1} = X^n - Я^{-1}(X^n)F(X^n),$$

где

$$X = \begin{pmatrix} x \\ y \end{pmatrix}, \quad F(X) = \begin{pmatrix} f_1(x,y) \\ f_2(x,y) \end{pmatrix}, \quad Я^{-1} = \begin{pmatrix} \frac{\partial}{\partial x} f_1(x,y) & \frac{\partial}{\partial y} f_1(x,y) \\ \frac{\partial}{\partial x} f_2(x,y) & \frac{\partial}{\partial y} f_2(x,y) \end{pmatrix}^{-1}$$

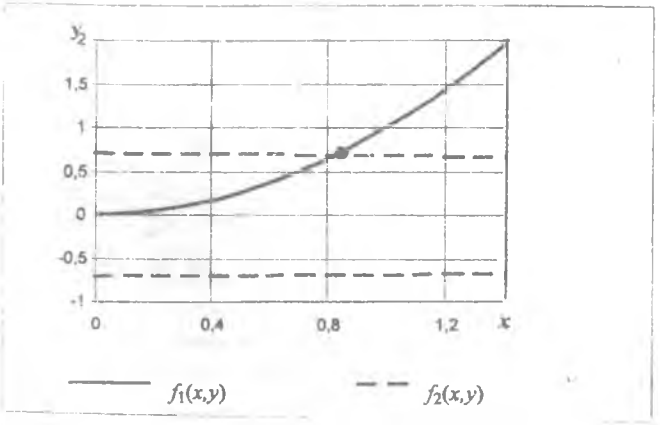


Рис. П.13

Построим матрицу Якоби:

$$Я = \begin{pmatrix} c & 2y \\ 2x & -1 \end{pmatrix}, \text{ где } c = \frac{2x\sin x - x^2\cos x}{10\pi\sin^2 x}$$

Найдем обратную матрицу $Я^{-1}$:

$\det Я = \begin{vmatrix} c & 2y \\ 2x & -1 \end{vmatrix} = -c - 4xy = -(c + 4xy)$. Определим алгебраические дополнения матрицы Я:

$A_{11} = -1, \quad A_{12} = -2x,$

$A_{21} = -2y, \quad A_{22} = c,$

тогда

$$Я^{-1} = \frac{1}{c + 4xy}\begin{pmatrix} 1 & 2y \\ 2x & -c \end{pmatrix}.$$

Теперь итерационную формулу метода Ньютона можно представить следующим образом:

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} x_n \\ y_n \end{pmatrix} - \frac{1}{c_n + 4x_n y_n}\begin{pmatrix} 1 & 2y_n \\ 2x_n & -c_n \end{pmatrix}\begin{pmatrix} f_1(x_n, y_n) \\ f_2(x_n, y_n) \end{pmatrix}$$ или в координатной форме:

$$x_{n+1} = x_n - \frac{f_1(x_n, y_n) + 2y_n f_2(x_n, y_n)}{c_n + 4x_n y_n},$$

$$y_{n+1} = y_n - \frac{2x_n f_1(x_n, y_n) - c_n f_2(x_n, y_n)}{c_n + 4x_n y_n}.$$

Итерационный процесс завершается, если будет выполнено условие: $\max\left\{\left|\frac{x_{n+1} - x_n}{x_{n+1}}\right|, \left|\frac{y_{n+1} - y_n}{y_{n+1}}\right|\right\} < \delta = 0{,}5\,10^{-5}$.

Xn	Yn	Xn+1	Yn+1	δ_x	δ_y
0,8	0,5	0,849654	0,719446	0,05844	0,305021
0,849654	0,719446	0,828874	0,686601	0,025069	0,047837
0,828874	0,686601	0,828151	0,685833	0,000874	0,00112
0,828151	0,685833	0,82815	0,685832	6,87E-07	6,1E-07

Как видно из таблицы, метод Ньютона сошелся к решению системы уравнений за четыре шага, при этом получено решение (x=0.82815; y=68583).

10. *Вычислить приближенное значение интеграла* $\int_a^b f(x)dx$ *с точностью до 5 значащих цифр. (При оценке погрешности учитывать не только методическую ошибку, но и ошибки округления).*

Решение.

Для вычисления определенного интеграла воспользуемся общей формулой Симпсона. Предварительно определим шаг интегрирования. Для формулы Симпсона остаточный член может быть оценен величиной

$$R = -\frac{(b-a)h^4}{180} y^{(4)}(\xi), \quad \xi \in [a,b].$$

Если задана предельная допустимая погрешность ε, то обозначив $M = \max\left|y^{(4)}(x)\right|$ для определения шага, будем иметь неравенство

$$h < \sqrt[4]{\frac{180\varepsilon}{(b-a)M}}.$$

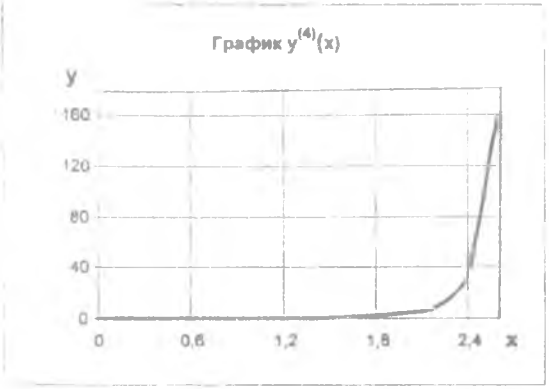


Рис. П.14

Для определения величины допустимой абсолютной погрешности интегрирования необходимо знать значение интеграла, поскольку $\varepsilon = I \cdot \delta_I = I \cdot 0.5\,10^{-5}$. Из рисунка П.2 видно, что исходная функция на отрезке [0; 2.5] является монотонно возрастающей, выпуклой функцией (выпуклостью вниз), причем при x=0, y=0, а при x=2.5, y=0.3324, поэтому справедлива оценка

$$I \le \frac{1}{2}(0.3324 - 0) \cdot 2.5 \approx 0.4156 \text{ и } \varepsilon \approx 2.08\,10^{-6}.$$

Таким образом, $h \approx 0.0369$. Для удобства в качестве шага интегрирования выберем $h \approx 0.05$, что соответствует n=50.

Используя правило Симпсона, получим

$$I_{50} = \int_0^{2.5} \frac{x^2}{10\pi \sin x}\,dx \approx 0.20285819.$$

Для проверки точности вычислений попытаемся "взять" интеграл аналитически. Из таблицы неопределенных интегралов (см. [11]) для заданной функции имеем

$$\frac{1}{10\pi}\int \frac{x\,dx}{\sin x} = \frac{1}{10\pi}\left(x + \frac{x^3}{3 \cdot 3!} + \frac{7x^5}{3 \cdot 5\ 5!} + \ldots + \frac{2(2^{2n-1}-1)}{(2n+1)!}B_n x^{2n+1} + \ldots\right),$$

где B_n - числа Бернулли. Однако данный ряд очень медленно сходится и в этом смысле мало пригоден для вычисления интеграла. Попытаемся "практически" оценить реальную величину погрешности интегрирования. Для этого вычислим интеграл с половинным шагом при n=100 (I_{100}=0.20285783). Тогда имеем

$$I_{ист} - I_{100} = -\frac{(b-a)y^{(4)}(\xi)}{180}h^4 = Ch^4 = e_{мет},$$

$$I_{ист} - I_{50} = -\frac{(b-a)y^{(4)}(\xi)}{180}(2h)^4 = 16Ch^4,$$

вычитая из первого выражения второе, имеем

$$I_{100} - I_{50} = 15Ch^4 \quad \text{или} \quad e_{мет} = \frac{1}{15}(I_{100} - I_{50}).$$

В нашем случае $|e_{мет}| \approx 2.4\,10^{-8}$, что обеспечивает требуемый уровень точности вычисления интеграла.

Поскольку при вычислении интеграла на ЭВМ выполняется большое количество арифметических операций, то необходимо оценить величину погрешности округления. Для грубой оценки можно использовать формулу ошибки округления для правила трапеций (5.13). Для простоты положим $\bar{y} = \frac{2.5-0}{2} \approx 1.25$, h=0.025,

тогда $$|e_{окр}| \approx \frac{\bar{y}(b-a)^2\delta}{2h} = \frac{1.25 \cdot 2.5^2 \cdot 0.5\,10^{-15}}{2 \cdot 0.025} \approx 7.8\,10^{-14}$$

Полученный результат показывает, что при n=100 погрешности округления пренебрежимо малы по сравнению с методической погрешностью, в связи с чем $I = 0.2028578 \pm 2.4\,10^{-8}$.

11. *Используя конечно-разностные формулы первого и второго порядка, найти производные функции в точке x_0. Исследовать зависимость погрешности вычисления производной от шага приращения аргумента h.*

Решение

Пусть x_0=2,5. Как показано в разделе 5.7 данного пособия, конечно-разностные формулы вычисления первой производной функции имеют вид

$$y'(x_0) \approx \frac{\Delta y_0}{h},$$

$$y'(x_0) \approx \frac{1}{h}\left(\Delta y_0 - \frac{\Delta^2 y_0}{2}\right).$$

Исследуем зависимости поведения погрешностей вычисления первой производной функции для этих двух случаев от шага интерполирования функции. При этом погрешности будем оценивать относительно аналитической формулы вычисления $y'(x)$:

h	e_1	e_2
0,1	0,139019	0,063711
0,01	0,011899	0,000388
0,001	0,001173	3,72E-06
0,0001	0,000117	3,7E-08
0,00001	1,17E-05	3,56E-10
0,000001	1,17E-06	4,69E-10
1E-07	1,18E-07	2,5E-09
1E-08	1,36E-08	8,05E-09
1E-09	1,14E-07	1,41E-07
1E-10	6,69E-07	9,46E-07
1E-11	3,44E-06	3,44E-06
1E-12	0,000114	0,000142

$$|e_1| = \left| y'(x_0) - \frac{\Delta y_0}{h} \right|,$$

$$|e_2| = \left| y'(x_0) - \frac{1}{h}\left(\Delta y_0 - \frac{\Delta^2 y_0}{2}\right) \right|.$$

На рисунке П.15 показаны графики зависимостей погрешностей вычисления первой производной функции. Как видно из графиков, как для первого, так и второго случаев погрешности вычисления производной имеют характерный оптимум по h. Как и следовало ожидать, более высокий уровень точности достигается для конечно-разностной формулы второго порядка, причем даже при больших значениях шага интерполирования.

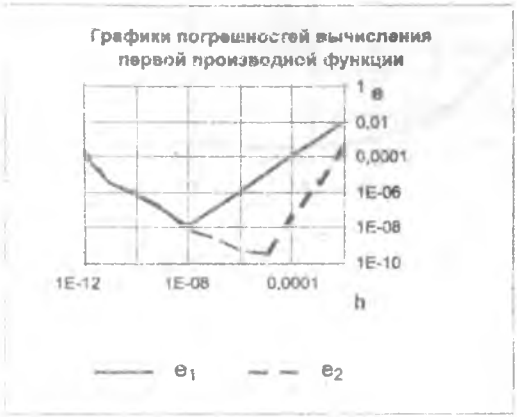


Рис. П.15

12. *Дано уравнение* $y' + cy = f(x)$ *и начальное условие* $y(x0) = b$. *Найти численное решение дифференциального уравнения на отрезке* $[x0; b]$. *Исследовать сходимость и устойчивость численного метода.*

Решение.

Пусть дано дифференциальное уравнение первого порядка $y' + 2y = \frac{x^2}{10\pi \sin x}$ при начальных условиях $y(0) = 1$. Найдем решение уравнения на отрезке [0; 2.5].

Воспользуемся наиболее употребительным методом Рунге-Кутта четвертого порядка (см. п.6.3). В задании не оговаривается

точность решения задачи. Из-за незначительности количества арифметических операций, выполняемых при реализации метода Рунге-Кутта, можно пренебречь ошибками арифметических операций. Тогда полную погрешность практически определяет методическая ошибка, которая зависит от величины шага интегрирования h. Построим графики решения (см. рис.П.16) для разных значений шага(h=0.2; h=0.1; h=0.05).

Рис. П.16

Как видно из рисунка, графики решений дифференциального уравнения при разных значениях шага интегрирования практически совпадают. Методическую погрешность решения уравнения исследуем практически.

Очевидно, что

$$Y(x) = \tilde{y}_h(x) + e_{h\ мет}(x),$$

$$Y(x) = \tilde{y}_{2h}(x) + e_{2h\ мет}(x),$$

где $Y(x)$- истинное решение уравнения, $\tilde{y}_h(x)$, $\tilde{y}_{2h}(x)$, $e_{h\ мет}(x)$, $e_{2h\ мет}(x)$ - приближенные решения уравнения и погрешности при различных значениях шага интегрирования. Вычитая одно равенство из другого, получим

$$|R_{мет}(x,h)| \approx |\tilde{y}_h(x) - \tilde{y}_{2h}(x)| = |e_{h\ мет}(x) - e_{2h\ мет}(x)|.$$

При условии, что $e_{h\ мет}(x) \neq e_{2h\ мет}(x)$, можно считать, что $R_{мет}(x,h)$ приближенно оценивает величину методической

погрешности. На рисунке П.17 представлены графики методических погрешностей решения уравнения.

Рис. П.17

Как видно из рисунка, абсолютная величина погрешности с увеличением x не возрастает, следовательно для данной задачи не наблюдается эффект накапливания погрешностей, полученных на предыдущих шагах интегрирования, и для данного примера можно утверждать, что имеет место *абсолютная устойчивость* метода Рунге-Кутта. Уменьшение величины погрешности с уменьшением h говорит о сходимости приближенного решения к истинному. На рисунке П.18 показана зависимость абсолютной величины погрешности от h в точке x=0,6

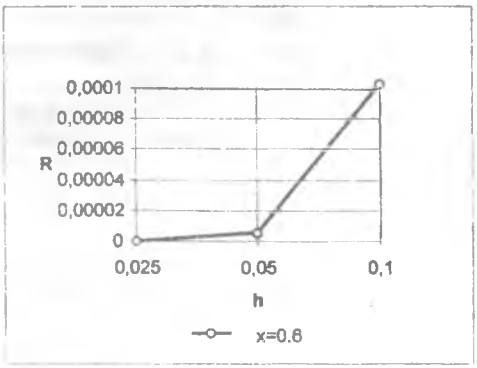


Рис. П.18

Построив подобные графики, можно определить величину шага h, обеспечивающего заданную точность решения дифференциального уравнения.

Основная литература

1. Д. Мак-Кракен, У. Дорн Численные методы и программирование на ФОРТРАНЕ. - М. :Мир, 1977.-584 с.

2. Демидович Б.П., Марон И.А. Основы вычислительной математики. - М. :Наука, 1970. - 664 с.

Используемая литература

3. Хемминг Р.В. Численные методы. - М. :Наука, 1972. - 398 с.

4. Завьялов Ю.С., Квасов Б.И., Мирошниченко В.Л. Методы сплайн-функций.-М. :Наука, 1980. - 352 с.

5. Тихонов А.Н., Костомаров Д.П. Вводные лекции по прикладной математике .-М. :Наука, 1984. - 190 с.

6. Кузьмин П.К., Маничев В.Б Автоматизация функционального проектирования- М. :Высшая школа. 1986. -141 с.

Рекомендуемая литература

7. Марчук Г.И. Методы вычислительной математики. М. :Наука, 1980.-535 с.

8. Стечкин С.В., Субботин Ю.Н. Сплайны в вычислительной математике. М. :Наука, 1976.

9. Химмельблау Д. Прикладное нелинейное программирование. М. :Мир, 1975.- 534 с.

10. Ортеш Дж., Рейболт В. Итерационные методы решения нелинейных систем уравнений со многими неизвестными. М. :Мир, 1975.-558 с.

11 Бронштейн И.Н., Семендяев К.А. Справочник по математике. М. :Наука, 1980.-974 с.

12 Коварцев А.Н. Автоматизация разработки и тестирования программных средств . Самара: Самар. гос. аэрокосм ун-т, 1999. - 150 с.

Учебное издание

Коварцев Александр Николаевич

АЛГОРИТМИЧЕСКИЕ ЯЗЫКИ И ПРОГРАММИРОВАНИЕ
Курс лекций для студентов заочной формы обучения

Редактор Т.К. Кретинина
Корректор Т.К. Кретинина

Лицензия ЛР № 020301 от 30.12.96 г.
Подписано в печать 05.09.2000. . Формат 60 х 84 1/16.
Бумага офсетная. Печать офсетная.
Усл. печ. л. 10,46. Усл. кр - отт. 10,58 Уч - изд. л. 11,25.
Тираж 200 экз. Заказ 6 52 Арт. С-И(03)/2000.

Самарский государственный аэрокосмический университет
имени академика С.П. Королева
443086 Самара, Московское шоссе, 34.

ИПО Самарского государственного аэрокосмического университета.
443001 Самара, ул. Молодогвардейская, 151.