**Федеральное государственное автономное образовательное учреждение
высшего профессионального образования
«Национальный исследовательский технологический университет
«МИСиС»
Новотроицкий филиал**

**Д. Д. Изаак,
А. В. Швалева**

ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА

Учебно-методическое пособие

Орск 2012

УДК 518.12
ББК 22.19
И32

Научный редактор

Бонди И. Л., *кандидат физико-математических наук*

Рецензенты:

Гюнтер Д. А., *кандидат физико-математических наук, доцент кафедры высшей математики Орского гуманитарно-технологического института (филиала) ОГУ;*

Соколов А. А., *кандидат физико-математических наук, доцент кафедры общеобразовательных и профессиональных дисциплин филиала СамГУПС в г. Орске*

И32 Изаак, Д. Д. Вычислительная математика : учебно-методическое пособие / Д. Д. Изаак, А. В. Швалева. – Орск : Издательство Орского гуманитарно-технологического института (филиала) ОГУ, 2012. – 97 с. – ISBN 978-5-8424-0615-9.

Данное пособие предназначено в первую очередь для студентов очной формы обучения технических вузов, изучающих вычислительную математику, а также может быть интересно студентам заочной формы обучения.

Рекомендовано методическим советом НФ НИТУ «МИСиС».

ISBN 9785-8424-0615-9

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Современные электронно-вычислительные машины (ЭВМ) дали в руки исследователей эффективное средство для математического моделирования сложных задач науки и техники. Именно поэтому количественные методы исследования в настоящее время проникают практически во все сферы человеческой деятельности, а математические модели становятся средством познания. Реализация математических моделей на ЭВМ осуществляется с помощью методов вычислительной математики, которая непрерывно совершенствуется вместе с прогрессом в области электронно-вычислительной техники.

Данное пособие предназначено для студентов технических вузов дневной и заочной форм обучения, изучающих вычислительную математику. Пособие включает в себя курс лекций и лабораторные работы, составленные в соответствии с программой курса «Вычислительная математика», утвержденной для специальности «Прикладная информатика в экономике».

Теоретическая часть составлена в соответствии с изданным ранее курсом лекций «Вычислительная математика» Д. Д. Изаака.

Практикум содержит четыре лабораторные работы, для каждой из которых предлагается 30 вариантов заданий. Каждая лабораторная работа расположена после соответствующей теоретической части и описывается по определенной схеме: цель, используемое программное обеспечение, основное задание, теоретические комментарии к выполнению задания, дополнительное задание и комментарии к дополнительному заданию. В ходе выполнения лабораторных работ студенты рассматривают важнейшие методы и приемы вычислительной математики, знание которых необходимо современному программисту при разработке алгоритмов решения практических задач. Однако, прежде чем приступить к выполнению лабораторных работ, авторы предлагают решить задачи на теорию погрешностей, которые представлены в конце первой главы.

Пособие охватывает следующие разделы: Погрешности; Численные методы решения уравнений; Численные методы решения систем уравнений; Аппроксимация функций; Численное интегрирование.

Авторы выражают благодарность К. М. Жумабековой и А. Н. Шакалову за помощь в создании данного пособия.

ГЛАВА 1. ПОГРЕШНОСТИ

1.1. Математические модели и численные методы. Устойчивость, корректность, сходимость

Математические модели и численные методы

Научные исследования предполагают выделение наиболее существенных черт в изучаемом явлении. Часто выделение таких черт позволяет перейти к более простому объекту, который правильно отражает основные закономерности явления и дает возможность получать о нем новую информацию. Такой объект и называется моделью.

Основное требование, предъявляемое к математической модели, – адекватность рассматриваемому явлению, то есть она должна достаточно точно (в рамках допустимых погрешностей) отражать характерные черты явления. Вместе с тем она должна обладать сравнительной простотой и доступностью исследования.

При построении математических моделей получают некоторые математические соотношения (как правило, уравнения).

Примеры

1. Уравнения $h = h_0 - v_0 t - \frac{gt^2}{2}$, $v = v_0 + gt$ описывают свободное падение тела, которое находилось на высоте h_0 и стало двигаться с начальной скоростью v_0.

2. В гидродинамике известна модель на основе уравнений в частных производных Навье – Стокса, описывающая движение вязкой сжимаемой жидкости.

Имеются математические модели и для описания задач экономики, социологии, медицины и так далее.

С помощью математического моделирования решение научно-технической задачи сводится к решению математической задачи, являющейся ее моделью. А это, в свою очередь, часто достигается средствами вычислительной математики, так как аналитическое решение задачи не всегда бывает возможно.

Вычислительная математика – дисциплина, изучающая численные (приближенные) методы решения математических задач.

Численные методы разрабатываются высококвалифицированными специалистами-математиками. Что касается подавляющей час-

ти студентов, то для них главной задачей является понимание основных идей методов, особенностей и областей применения.

Что же такое численные методы? Под численными методами подразумеваются методы решения задач, сводящиеся к арифметическим действиям над числами, то есть к тем действиям, которые выполняют ЭВМ.

Решение, полученное численным методом, обычно является приближенным, то есть содержит некоторую погрешность. Источниками погрешностей приближенного решения являются:

1. Несоответствие математической модели изучаемому реальному явлению (неадекватность математической модели).
2. Погрешность исходных данных (входных параметров).
3. Погрешность метода решения.
4. Погрешности округлений в арифметике и других действиях над числами.

Погрешность в решении, вызванная первыми двумя источниками, называется неустранимой. Она может присутствовать, даже если решение поставленной математической задачи найдено точно.

Численные методы в большинстве случаев сами по себе являются приближенными, то есть даже при отсутствии погрешностей во входных данных и при идеальном выполнении арифметических действий они дают решение исходной задачи с некоторой погрешностью, называемой погрешностью метода. Это происходит потому, что численным методом обычно решается другая, более простая задача, аппроксимирующая (приближающая, заменяющая) исходную задачу.

Устойчивость, корректность, сходимость

Рассмотрим погрешности исходных данных. Поскольку это так называемые неустранимые погрешности и с ними вычислитель не может бороться, то нужно хотя бы иметь представление об их влиянии на точность окончательных результатов.

Пусть в результате решения задачи по исходному значению x находится значение искомой величины y. Пусть исходная величина имеет абсолютную погрешность Δx, а решение имеет погрешность Δy. Задача называется устойчивой по исходному параметру x, если малое приращение исходной величины Δx приводит к малому приращению искомой величины Δy. Другими словами, малые погрешно-

сти в исходных величинах приводят к малым погрешностям в результатах расчетов.

Отсутствие устойчивости означает, что даже незначительные погрешности в исходных данных приводят к большим погрешностям в решении или вовсе – к неверному результату. О подобных неустойчивых задачах также говорят, что они чувствительны к погрешностям исходных данных.

Примеры

1. Рассмотрим задачу нахождения корней многочленов вида $(x-a)^n=\varepsilon$, $0<\varepsilon<1$. Изменение правой части на величину порядка ε приводит к погрешности корней порядка $\varepsilon^{1/n}$, что при больших n гораздо больше ε. Например, если правую часть уравнения $x^6=10^{-6}$ увеличить на $7\cdot10^{-6}$, то есть рассмотреть уравнение $x^6=8\cdot10^{-6}$, то корень увеличится примерно на $4\cdot10^{-2}$ (с 0,10 до 0,14).

2. Пример Уилкинсона. Рассмотрим многочлен

$$P(x)=(x-1)(x-2)\ldots(x-20)=x^{20}-210x^{19}+\ldots$$

Очевидно, что корнями многочлена являются $x_1=1$, $x_2=2$, …, $x_n=20$. Предположим, что один из коэффициентов многочлена вычислен с некоторой малой погрешностью. Например, коэффициент -210 при x^{19} увеличим на $2^{-23}\left(10^{-7}\right)$. В результате вычислений с точностью до 11 значащих цифр получим существенно другие значения корней, и половина корней станет мнимыми. Причина такого явления – неустойчивость самой задачи, так как вычисления выполнялись очень точно (11 разрядов) и погрешность округлений не могла привести к таким результатам.

Задача называется поставленной корректно, если для всех значений исходных данных из некоторого класса ее решение существует, единственно и устойчиво по исходным данным. Рассмотренные выше примеры неустойчивых задач является некорректно поставленными. Применять для решения таких задач численные методы, как правило, нецелесообразно, так как возникающие в расчетах погрешности округлений будут сильно возрастать в ходе вычислений, что приведет к значительному искажению результатов.

Однако в настоящее время разработаны методы решения некоторых некорректных задач. Это, в основном, так называемые методы регуляризации, которые основываются на замене исходной задачи корректно поставленной задачей.

При анализе точности вычислений процесса одной из важнейших характеристик является сходимость численного метода. Она означает близость получаемого численного решения задачи к исходному решению. Различают сходимость итерационного процесса и сходимость в методах дискретизации.

Рассмотрим понятие сходимости итерационного процесса. Этот процесс состоит в том, что для решения некоторой задачи строится метод последовательных приближений. В результате многократного повторения этого процесса (итераций) получаем последовательность значений x_1, x_2, …, x_n, … . Говорят, что эта последовательность сходится к точному значению $x = a$, если при неограниченном возрастании числа итераций предел этой последовательности существует и равен a. То есть $\lim\limits_{n\to\infty} x_n = a$. В этом случае имеем сходящийся численный метод. (Например, метод Ньютона для численного решения уравнений.)

Рассмотрим теперь понятие сходимости, используемое в методах дискретизации. Эти методы заключаются в замене задачи с непрерывными параметрами на задачу, в которой значения функции вычисляются в фиксированных точках. Здесь под сходимостью понимается стремление значений решения дискретной модели к соответствующим значениям решения исходной задачи при стремлении к нулю параметра дискретизации. (Например, квадратурные формулы.)

При рассмотрении сходимости важными понятиями являются вид сходимости, ее порядок и другие характеристики. В общем виде эти понятия рассматривать нецелесообразно, мы будем обращаться к ним при изучении конкретного численного метода.

Таким образом, для получения решения задачи с некоторой точностью ее постановка должна быть корректной, а используемый численный метод должен обладать сходимостью.

1.2. Абсолютная и относительная погрешности. Погрешность суммы и разности приближенных чисел

Абсолютная и относительная погрешности

На практике обычно числа, над которыми производятся вычисления, являются приближенными значениями тех или иных величин. Для краткости речи приближенное значение величины называют

приближенным числом. Истинное значение величины называют точным числом.

Приближенное число имеет практическую ценность лишь тогда, когда мы можем определить, с какой степенью точности оно дано, то есть оценить его погрешность, отличие от точного числа. Будем обозначать через x точное число, через a – приближенное число.

Определение 1.2.1. Истинной погрешностью приближенного числа a называется разность между точным числом x и его приближенным значением a.

Таким образом, истинная погрешность приближенного числа a равна $x - a$. Истинная погрешность может быть числом положительным, отрицательным или равным нулю.

Определение 1.2.2. Абсолютной погрешностью приближенного числа a называется модуль разности между точным числом x и его приближенным значением a.

Абсолютную погрешность приближенного числа a будем обозначать Δa. То есть, $\Delta a = |x - a|$.

Точное число x чаще всего бывает неизвестно, поэтому найти истинную и абсолютную погрешности не представляется возможным. С другой стороны, бывает необходимо оценить абсолютную погрешность, то есть указать число, которого не может превысить абсолютная погрешность. Другими словами, нужно знать границу абсолютной погрешности. Эту границу будем называть предельной абсолютной погрешностью.

Определение 1.2.3. Предельной абсолютной погрешностью приближенного числа a называется положительное число Δ_a такое, что $|x - a| \le \Delta_a$.

Последнюю формулу можно записать в виде: $a - \Delta_a \le x \le a + \Delta_a$. Таким образом, $a - \Delta_a$ есть приближенное значение числа x по недостатку, $a + \Delta_a$ – по избытку. Используют также такую запись: $x = a \pm \Delta_a$.

Очевидно, что предельная абсолютная погрешность выбирается неоднозначно: если какое-то число является предельной абсолютной погрешностью, то любое большее число тоже есть предельная абсолютная погрешность. На практике стараются выбирать возможно меньшее и простое по записи (с 1-2 значащими цифрами) число Δ_a, удовлетворяющее неравенству $|x - a| \le \Delta_a$.

Пример

Определить истинную, абсолютную и предельную абсолютную погрешности числа $a = 0{,}17$, взятого в качестве приближенного значения числа $x = \frac{1}{6}$.

Истинная погрешность: $\frac{1}{6} - 0{,}17 = \frac{1}{6} - \frac{17}{100} = -\frac{1}{300}$.

Абсолютная погрешность: $\Delta a = \left| -\frac{1}{300} \right| = \frac{1}{300}$.

За предельную абсолютную погрешность можно принять число $\frac{1}{300}$ и любое большее число. В десятичной записи будем иметь $\frac{1}{300} = 0{,}00333\ldots$. Заменяя это число большим и возможно более простым по записи, примем: $\Delta_a = 0{,}004$.

Часто говорят, что a есть приближенное значение числа x с точностью до Δ_a.

Так как термины «истинная погрешность» и «абсолютная погрешность» в дальнейшем практически не будут использоваться, предельную абсолютную погрешность будем называть просто абсолютной погрешностью. Слово «погрешность» употребляется, когда речь идет о действиях над числами. Когда говорят об измерениях, вместо слова «погрешность» употребляют слово «ошибка».

Знания абсолютной погрешности недостаточно для характеристики качества измерения или вычисления.

Определение 1.2.4. Относительной погрешностью δ приближенного значения a называется отношение абсолютной погрешности Δa к модулю числа x: $\delta = \frac{\Delta a}{|x|}$. Так как точное число обычно бывает неизвестно, его заменяют приближенным числом: $\delta = \frac{\Delta a}{|a|}$.

Определение 1.2.5. Предельной относительной погрешностью приближенного числа a называется положительное число δ_a такое, что $\frac{\Delta a}{|a|} \le \delta_a$.

Так как $\Delta a \le \Delta_a$, то $\delta_a = \frac{\Delta_a}{|a|}$.

Для краткости вместо предельной относительной погрешности будем говорить просто относительная погрешность. Относительную погрешность обычно выражают в процентах.

Пример

При взвешивании тела получен результат: $p = 23{,}4 \pm 0{,}2$ г. Имеем $\Delta_p = 0{,}2$; $\delta_p = \dfrac{0{,}2}{23{,}4}$. Произведя деление и округляя в большую сторону, получаем $\delta_p = 0{,}9\%$.

Определение 1.2.6. Значащими цифрами числа называются все цифры его десятичного представления, кроме нулей, стоящих перед первой цифрой, отличной от нуля.

Определение 1.2.7. Цифра α называется верной в узком смысле, если абсолютная погрешность приближенного числа не превосходит половины единицы того разряда, в котором записана цифра α.

Пример

Даны приближенные числа, все цифры которых верны в узком смысле: $a = 3{,}8$; $b = 0{,}0283$; $c = 4260$. Предельные абсолютные погрешности этих чисел: $\Delta_a = 0{,}05$; $\Delta_b = 0{,}00005$; $\Delta_c = 0{,}5$.

Определение 1.2.8. Цифра α называется верной в широком смысле, если абсолютная погрешность приближенного числа не превосходит единицы того разряда, в котором записана цифра α.

Принимается за правило при десятичной записи приближенного числа писать только верные в широком смысле цифры. То есть при правильной записи абсолютная погрешность не превышает единицы низшего разряда.

Определение 1.2.9. Погрешностью округления называется погрешность, возникающая при округлении.

Пример

Округляя точное число $2{,}378$ до двух значащих цифр, получим $2{,}4$. Погрешность округления менее $0{,}03$. Округляя точное число $\sqrt{3} \approx 1{,}73205\ldots$ до трех значащих цифр, получим приближенное значение $\sqrt{3} \approx 1{,}73$. Погрешность округления не превышает $0{,}003$.

На первый взгляд, последнее определение может показаться «примитивным». Однако это не так. Чтобы пояснить почему, рассмотрим задачу.

Задача. Вычислить приближенно сумму ряда $\sum\limits_{n=0}^{\infty} \dfrac{(-1)^n}{6^n}$ с точностью $\varepsilon = 0{,}001$. Из курса анализа известно, что, если ряд является

знакочередующимся и его члены убывают по абсолютной величине, то достаточно найти тот член ряда, который по модулю меньше, чем ε, и, сложив все предыдущие члены, мы получим приближенное значение суммы с необходимой точностью. Наш ряд удовлетворяет этим условиям. $a_0 = 1$, $a_1 = -0{,}166667$, $a_2 = 0{,}027778$, $a_3 = -0{,}004630$, $a_4 = 0{,}000772$, $a_5 = -0{,}000129$. (Мы здесь округляли промежуточные значения до шестого знака после запятой для того, чтобы промежуточные округления не повлияли бы на окончательный ответ.) Модуль четвертого члена меньше, чем ε, поэтому

$$S \approx \overline{S}_1 = a_0 + a_1 + a_2 + a_3 = 0{,}856480.$$

Округляя результат до третьего знака после запятой, чтобы в ответе оставить только значащие цифры в широком смысле, получаем: $S \approx \overline{S}_2 = 0{,}856$. Мы можем оценить абсолютные погрешности этих двух приближений, так как числовой ряд представляет бесконечно убывающую геометрическую прогрессию, и мы можем посчитать точное значение суммы.

$$S = \frac{1}{1 + \frac{1}{6}} = \frac{6}{7} = 0{,}857143;$$

$$\Delta_{S_1} = |\, S - \overline{S}_1 \,| = 0{,}000663 < \varepsilon;$$

$$\Delta_{S_2} = |\, S - \overline{S}_2 \,| = 0{,}001143 > \varepsilon.$$

Видно, что абсолютная погрешность первого приближения, как и должно быть, меньше ε. Абсолютная же погрешность второго, округленного приближения, вследствие погрешности округления, больше заданной точности.

Найти какую-либо величину (корень уравнения, интеграл и т. п.) с точностью ε, означает найти такое приближенное значение величины, абсолютная погрешность которого строго меньше ε. При этом обычно берут $\varepsilon = 10^{-n}$, где $n \in N$. Причем окончательный результат округляют до n знаков после запятой. Таким образом, появляется погрешность округления, которая меньше либо равна $\frac{\varepsilon}{2}$. Поэтому договоримся при решении любой задачи каким-либо численным методом вместо ε брать величину $\frac{\varepsilon}{2}$ так, чтобы окончательный округленный результат имел абсолютную погрешность, строго меньшую ε.

Вернемся к задаче. Поэтому сначала будем искать сумму ряда с точностью $\frac{\varepsilon}{2} = 0{,}0005$. $|a_4| > \frac{\varepsilon}{2}$, а $|a_5| < \frac{\varepsilon}{2}$. Тогда

$$S \approx \overline{S}_1 = a_0 + a_1 + a_2 + a_3 + a_4 = 0{,}857250.$$

Округляя результат до третьего знака после запятой, получаем: $S \approx \overline{S}_2 = 0{,}857$. Причем $\Delta_{S_2} = |S - \overline{S}_2| = 0{,}000143 < \varepsilon$.

Погрешность суммы и разности приближенных чисел

Теорема 1.2.1. Предельная абсолютная погрешность суммы нескольких приближенных чисел равна сумме предельных абсолютных погрешностей слагаемых.

Теорема 1.2.2. Предельная абсолютная погрешность разности приближенных чисел равна сумме предельных абсолютных погрешностей уменьшаемого и вычитаемого.

Таким образом, если $u = a + b$, $v = a - b$, то $\Delta_u = \Delta_a + \Delta_b$, $\Delta_u = \Delta_a + \Delta_b$.

Примеры

Здесь и в последующих задачах будем считать, что даны приближенные числа, но в записи которых все цифры верные в широком смысле.

1. $u = 2{,}72 + 3{,}00 + 2{,}11$; $u = 7{,}83$; $\Delta_u = 0{,}01 + 0{,}01 + 0{,}01 = 0{,}03$.
2. $a = 1{,}374$; $b = 0{,}921$; $u = a - b = 0{,}453$. $\Delta_u = 0{,}001 + 0{,}001 = 0{,}002$.

1.3. Погрешность произведения и частного приближенных чисел. Погрешность возведения в степень и извлечения корня из приближенных чисел. Оценка погрешности результата вычислений по формуле

Погрешность произведения и частного приближенных чисел

Теорема 1.3.1. Предельная относительная погрешность произведения нескольких приближенных чисел равна сумме предельных относительных погрешностей сомножителей.

Теорема 1.3.2. Предельная относительная погрешность частного от деления двух приближенных чисел равна сумме предельных относительных погрешностей делимого и делителя.

Таким образом, если $u = a \cdot b$, $v = \frac{a}{b}$, то $\delta_u = \delta_a + \delta_b$, $\delta_v = \delta_a + \delta_b$.

Пример

Найдем относительную погрешность произведения двух приближенных чисел: $a = 6{,}32$; $b = 0{,}783$. Из записи чисел определяем

$$\delta_a = \frac{\Delta_a}{|a|} = \frac{0{,}01}{6{,}32} = 0{,}001583;\ \delta_b = \frac{\Delta_b}{|b|} = \frac{0{,}001}{0{,}783} = 0{,}001278.$$

Значит, $\delta_{a\cdot b} = \delta_a + \delta_b = 0{,}29\%$. Мы здесь брали число знаков с запасом в промежуточных результатах, но, округляя в большую сторону, и окончательный результат немного увеличили, чтобы сделать его более простым по записи.

Погрешность возведения в степень и извлечения корня из приближенных чисел

Теорема 1.3.3. Предельная относительная погрешность степени приближенного числа равна произведению показателя степени на предельную относительную погрешность основания.

Теорема 1.3.4. Предельная относительная погрешность корня из приближенного числа равна предельной относительной погрешности подкоренного числа, деленной на показатель корня.

Таким образом, $\delta_{a^n} = n \cdot \delta_a$, $\delta_{\sqrt[n]{a}} = \frac{\delta_a}{n}$.

Пример

Требуется найти относительную погрешность объема куба $V = a^3$, где a – длина ребра, измеренная с погрешностью не более 1%.

Предельная относительная погрешность $\delta_v = 3 \cdot \delta_a = 3\%$.

Оценка погрешности результата вычислений по формуле

В практике вычислений очень часто приходится оценивать погрешность числового значения величины, полученной в результате вычислений по формуле, которая содержит не одно, а несколько действий. Для оценки погрешности в этом случае следует последовательно применять теоремы о погрешностях.

Во всех задачах, рассмотренных в данном параграфе, будет задана формула, все составляющие которой – приближенные числа (за редким исключением), а найти нужно приближенный результат и записать его так, чтобы в записи все цифры были верными в широком смысле, а также надо определить абсолютную и относительную погрешности окончательного результата.

Примеры

1. $u = \dfrac{\sqrt{3}}{\pi}$, где $\sqrt{3}$ и π вычислены приближенно с четырьмя верными значащими цифрами: $\sqrt{3} \approx 1{,}732$; $\pi = 3{,}142$.

Из записи числа определяем

$$\delta_{\sqrt{3}} = \frac{0{,}001}{1{,}732} = 0{,}000578;\ \delta_{\pi} = \frac{0{,}001}{3{,}142} = 0{,}000319.$$

Значит,

$$\delta_u = \delta_{\sqrt{3}} + \delta_{\pi} = 0{,}000897,\ u = 0{,}551241,\ \Delta_u = \delta_u \cdot u = 0{,}000495.$$

Мы округляли промежуточные результаты в сторону увеличения. Запишем ответ так, чтобы в записи все цифры были верные: $u = 0{,}551$. Действительно,

$$\Delta_u = 0{,}000495 + 0{,}000241 = 0{,}000736 < 0{,}001.\ \delta_u = \frac{0{,}000736}{0{,}551} = 0{,}001336.$$

Ответ: $u = 0{,}551$, $\Delta_u = 0{,}0008$, $\delta_u = 0{,}14\%$.

2. $u = a^4$, где $a = 1{,}30$.

$$\delta_a = \frac{0{,}01}{1{,}30} = 0{,}007693; \qquad \delta_u = 4 \cdot \delta_a = 0{,}030772; \qquad u = a^4 = 2{,}8561;$$

$\Delta_u = \delta_u \cdot u = 0{,}087888$. Округляя так, чтобы сохранить только верные цифры, получаем: $u = 3$. Действительно, $\Delta_u = 0{,}087888 + 0{,}1439 = 0{,}231788 < 1.\ \delta_u = \dfrac{0{,}231778}{3} = 0{,}077263$. Ответ: $u = 3$, $\Delta_u = 0{,}24$, $\delta_u = 7{,}8\%$.

3. Вычисляют объем цилиндра по формуле $V = \pi R^2 H$. При этом принимают $\pi \approx 3{,}142$; $R \approx 28{,}70$ см; $H \approx 84{,}3$ см.

Так как в формуле участвует только умножение, то проще начать с оценки относительной погрешности. Из записи приближенных чисел видно:

$$\delta_{\pi} = \frac{0{,}001}{3{,}142} = 0{,}000319,\ \delta_R = \frac{0{,}01}{28{,}70} = 0{,}000349,\ \delta_H = \frac{0{,}1}{84{,}3} = 0{,}001187.$$

Отсюда получаем: $\delta_V = \delta_{\pi} + 2\delta_R + \delta_H = 0{,}002204$. $V = 218171{,}25$; $\Delta_V = \delta_V \cdot V = 480{,}8495$. Округляя так, чтобы сохранить только верные цифры, получаем: $V = 218 \cdot 10^3$. Действительно,

$$\Delta_V = 480{,}8495 + 171{,}25 = 652{,}0995 < 1000.\ \delta_V = \frac{652{,}0995}{218 \cdot 10^3} = 0{,}002992.$$

Ответ: $V = 218 \cdot 10^3$, $\Delta_V = 653$, $\delta_V = 0{,}30\%$.

4. Вычисляют период колебания маятника по формуле $T = 2\pi\sqrt{\frac{l}{g}}$. При этом полагают $\pi \approx 3{,}142$; $l \approx 120{,}00$ см; $g \approx 981{,}32 \frac{\text{см}}{\text{сек}^2}$.

Из записи приближенных чисел видно: $\delta_\pi = \frac{0{,}001}{3{,}142} = 0{,}000319$, $\delta_l = \frac{0{,}01}{120{,}00} = 0{,}000084$, $\delta_g = \frac{0{,}01}{981{,}32} = 0{,}000011$. Отсюда получаем:

$$\delta_T = \delta_\pi + \frac{1}{2}(\delta_l + \delta_g) = 0{,}000367.$$

$$T = 2{,}197461,\ \Delta_T = \delta_T \cdot T = 0{,}000807.$$

Округляя так, чтобы сохранить только верные цифры, получаем: $T = 2{,}20$. Действительно,

$$\Delta_T = 0{,}002539 + 0{,}000807 = 0{,}003346 < 0{,}01.\ \delta_T = \frac{0{,}003346}{2{,}20} = 0{,}001521.$$

Ответ: $T = 2{,}20$, $\Delta_T = 0{,}0034$, $\delta_T = 0{,}16\%$.

5. Вычислить $u = \frac{ab}{c+d}$, где $a = 5{,}64$; $b = 7{,}26$; $c = 2{,}33$; $d = 1{,}64$.

Здесь нужно произвести сложение, при котором легко определяется абсолютная погрешность, и, кроме того, умножение и деление, при которых легко определяется относительная погрешность. Поэтому при оценке погрешности надо будет переходить от одного вида погрешности к другому.

Найдем $\Delta_{c+d} = \Delta_c + \Delta_d = 0{,}01 + 0{,}01 = 0{,}02$. Для дальнейшего придется найти относительную погрешность знаменателя: $\delta_{c+d} = \frac{0{,}02}{3{,}97} = 0{,}005038$. Теперь можем найти относительную погрешность искомого числа u.

$$\delta_u = \delta_a + \delta_b + \delta_{c+d} = 0{,}001773 + 0{,}001378 + 0{,}005038 = 0{,}008189.$$

$u = 10{,}313954$, $\Delta_u = \delta_u \cdot u = 0{,}084461$. Округляя так, чтобы сохранить только верные цифры, получаем: $u = 10{,}3$. Действительно,

$$\Delta_u = 0{,}013954 + 0{,}084461 = 0{,}098415 < 0{,}1.$$

Ответ: $u = 10{,}3$, $\Delta_u = 0{,}1$, $\delta_u = 1\%$.

1.4. О вычитании «близких чисел». Обратная задача теории приближенных вычислений. О вычислениях без строгого учета погрешностей

О вычитании «близких чисел»

Пусть даны приближенные числа $a = 58{,}345$ и $b = 58{,}321$, требуется найти их разность. Находим: $u = a - b = 0{,}024$. Заметим, что $\delta_a = \delta_b = 0{,}000018 = 0{,}0018\%$, то есть относительные погрешности данных чисел очень малы. Найдем теперь относительную погрешность разности.

$$\delta_u = \frac{\Delta_a + \Delta_b}{|a - b|} = \frac{0{,}002}{0{,}024} = 0{,}083333 = 8{,}4\%.$$

Как видим, относительная погрешность разности почти в 5000 раз больше, чем погрешность вычитаемого и уменьшаемого. Говорят, что при вычитании близких чисел происходит потеря точности.

Точность результата при вычитании близких чисел можно повысить, если перед выполнением вычислений произвести тождественные преобразования так, чтобы избежать вычитания близких чисел. Например, пусть $a = \sqrt{2{,}01} - \sqrt{2}$ (под знаком радикала стоят точные числа). Это разность близких чисел. Во избежание потери точности преобразуем выражение так:

$$a = \sqrt{2{,}01} - \sqrt{2} = \frac{0{,}01}{\sqrt{2{,}01} + \sqrt{2}} = \frac{0{,}01}{1{,}42 + 1{,}41} = 0{,}003534.$$

Относительная погрешность этого приближения равна: $\delta_a = \frac{0{,}02}{2{,}83} = 0{,}71\%$. Если бы мы не проводили тождественных преобразований, то: $a = 1{,}42 - 1{,}41 = 0{,}01$, $\delta_a = \frac{0{,}02}{0{,}01} = 200\%$!

Обратная задача теории приближенных вычислений

Рассмотрим задачу. Требуется вычислить объем конуса по формуле $V = \frac{1}{3}\pi R^2 H$ так, чтобы погрешность не превышала 1%. С какой точностью следует измерить радиус основания и высоту, чтобы обеспечить требуемую точность результата?

Для решения задачи следует знать грубые приближенные значения R и H по недостатку. Пусть $R \approx 30$ см $(R > 30)$; $H \approx 80$ см $(H > 80)$. На основании теоремы об умножении приближенных чисел составляем неравенство: $\delta_\pi + 2\delta_R + \delta_H \le 0{,}01$. Число π мы можем взять с любой степенью точности, то есть δ_π можно взять сколь угодно малым. Положим пока $\pi = 3{,}142$, то есть $\delta_\pi = 0{,}00032$. Неравенство будет таково: $2\delta_R + \delta_H \le 0{,}00968$. Мы получили одно неравенство с двумя неизвестными. Но мы можем наложить на δ_R и δ_H некоторые дополнительные условия. Например, мы можем считать, что измерения R и H будут проведены при одинаковой точности измерительных инструментов. Значит, можно положить $\Delta_R = \Delta_H$. Так как

$$\delta_R = \frac{\Delta_R}{30}\ (R > 30),\ \delta_H = \frac{\Delta_H}{80}\ (H > 80),$$

то находим: $\frac{19}{240}\Delta_R \le 0{,}00968$. Отсюда $\Delta_R = \Delta_H = 0{,}1223(\text{см}) = 1{,}223(\text{мм})$.

Значит, для получения требуемой точности достаточно произвести измерения R и H с погрешностью, не превышающей 1 мм, так как

$$\Delta_R = \Delta_H \le 1\text{мм} < 1{,}223\ \text{мм}.$$

Итак, в теории приближенных вычислений рассматриваются две основные задачи.

Прямая задача. Указаны действия, которые следует выполнить над приближенными числами (например, произвести вычисления по данной формуле), и заданы предельные погрешности этих чисел. Требуется оценить погрешность результата.

Обратная задача. Указаны действия, которые нужно выполнить над приближенными числами (например, провести вычисления по данной формуле). Требуется установить, каковы должны быть допустимые погрешности приближенных чисел, чтобы полученный результат имел наперед заданную предельную погрешность.

О вычислениях без строгого учета погрешностей

Приведенные теоремы позволяют проводить строгий учет погрешности. Применяя, например, теорему о сложении приближенных чисел, мы можем гарантировать, что при вычислении суммы 10 слагаемых, каждое из которых имеет абсолютную погрешность, не превышающую 0,005, погрешность суммы не превзойдет 0,05. Однако найденная таким образом граница погрешности обычно бывает зна-

чительно завышенной, она получается с большим «запасом». В действительности при сложении 10 слагаемых погрешность возрастает (в большинстве случаев) не в 10 раз, а лишь немного превышает погрешность слагаемых. Поэтому при выполнении арифметических действий над приближенными числами в тех случаях, когда не требуется строгого учета точности, установлены правила, позволяющие быстро, без громоздких исследований определить, как нужно проводить вычисления, чтобы получить результат нужной точности. Эти правила не столь точны, как правила вычислений со строгим учетом погрешностей, но во многих вычислениях вполне достаточны. Они называются правилами верных цифр. При формулировке этих правил будем считать, что число данных чисел невелико.

Правила верных цифр

1. При сложении и вычитании приближенных чисел в результате младший сохраненный десятичный разряд должен быть тот, который соответствует наименее точному из слагаемых. (Например: 2,173+1,22=3,393=3,39.)
2. При умножении и делении в результате следует сохранять столько значащих цифр, сколько их имеет приближенное данное с наименьшим числом значащих цифр. (Например: $3000 \cdot 0{,}21 = 6{,}3 \cdot 10^2$.)
3. При возведении в квадрат и куб в результате следует сохранять столько значащих цифр, сколько их имеет возводимое в степень приближенное число.
4. При извлечении квадратного и кубического корней в результате следует брать столько значащих цифр, сколько их имеет приближенное значение подкоренного числа.
5. При вычислениях по формуле во всех промежуточных результатах следует сохранять одной цифрой больше, чем рекомендуют предыдущие правила. В окончательном результате эта «запасная» цифра отбрасывается.
6. Если какое-нибудь данное имеет больше десятичных знаков (при сложении и вычитании) или больше значащих цифр (при умножении, делении, возведении в степень, извлечении корня), чем другие, то его предварительно следует округлить, сохраняя одну лишнюю цифру.

1.5. Задачи на вычисление погрешностей

Во всех задачах, рассмотренных в данной главе, будет задана формула, все составляющие которой – приближенные числа за исключением коэффициентов, а найти нужно приближенный результат и записать его так, чтобы в записи все цифры были верными в широком смысле, а также надо определить абсолютную и относительную погрешности окончательного результата.

1. $u = a + b - 2c$; $a = 0{,}98861$; $b = 3{,}012$; $c = 2{,}281$.
2. $u = a - 3b + 2c$; $a = 1{,}21$; $b = 0{,}92$; $c = 5{,}10$.
3. $u = a\sqrt{b}$; $a = 2{,}364$; $b = 1{,}28$.
4. $u = \dfrac{\sqrt{a}}{b}$; $a = 4{,}0223$; $b = 0{,}982$.
5. $u = a + b\sqrt{c}$; $a = 2{,}875$; $b = 7{,}022$; $c = 2{,}9277$.
6. $u = \dfrac{a}{b + c}$; $a = 0{,}11$; $b = 0{,}90$; $c = 1{,}12$.
7. $u = \dfrac{a}{b + ac}$; $a = 4{,}271$; $b = 4{,}821$; $c = 0{,}827$.
8. $u = \dfrac{c}{a^3 + b}$; $a = 1{,}322$; $b = 0{,}48$; $c = 7{,}18$.

Ответы

1. $u = -0{,}56$; $\Delta_u = 0{,}005$; $\delta_u = 0{,}8\%$.
2. $u = 9$; $\Delta_u = 0{,}41$; $\delta_u = 4{,}6\%$.
3. $u = 2{,}7$; $\Delta_u = 0{,}038$; $\delta_u = 1{,}4\%$.
4. $u = 2{,}04$; $\Delta_u = 0{,}005$; $\delta_u = 0{,}22\%$.
5. $u = 14{,}89$; $\Delta_u = 0{,}003$; $\delta_u = 0{,}02\%$.
6. $u = 0{,}05$; $\Delta_u = 0{,}01$; $\delta_u = 19{,}9\%$.
7. $u = 0{,}511$; $\Delta_u = 0{,}0008$; $\delta_u = 0{,}16\%$.
8. $u = 2{,}6$; $\Delta_u = 0{,}045$; $\delta_u = 1{,}72\%$.

ГЛАВА 2. ЧИСЛЕННЫЕ МЕТОДЫ РЕШЕНИЯ УРАВНЕНИЙ

2.1. Постановка задачи. Метод половинного деления

Постановка задачи

Пусть задана некоторая функция $y = f(x)$, являющаяся левой частью уравнения $f(x) = 0$.

Задача решения уравнения $f(x) = 0$ заключается в отыскании такого значения $x^* \in X \subset R$, где X – область определения $f(x)$, что $f(x^*) = 0$. Ясно, что аналитическое, или, как принято говорить, «точное», решение уравнения $f(x) = 0$ возможно получить, лишь когда удается аналитически представить $x^* = f^{-1}(0)$, где f^{-1} – обратная для f функция.

В общем случае задача $f(x) = 0$ решается с использованием специально разработанных методов, позволяющих отыскивать некоторое приближение для x^* с гарантированной погрешностью. При этом, как правило, исходная задача сводится к решению двух последовательных задач: задаче локализации (отделения) корня и задаче уточнения корня.

Рассмотрим каждую из задач более подробно. Задача локализации (отделения) корня уравнения состоит в следующем: найти отрезок, содержащий единственный корень уравнения $f(x) = 0$. В основе аналитического способа отделения корней лежит следующая теорема существования.

Теорема 2.1.1 (теорема Коши). Если $f(x) \in C_{[x_1, x_2]}$ и $f(x_1) \cdot f(x_2) \le 0$, то существует точка $x^* \in [x_1, x_2]$ такая, что $f(x^*) = 0$.

Идея метода заключается в том, что исходный отрезок $[a, b]$ разбивается на *n* частей – элементарных отрезков и в каждом отрезке исследуется знак функции на его концах.

Теорема Коши не гарантирует единственности корня. Единственность корня следует из монотонности функции на отрезке $[x_1, x_2]$. То есть справедлива следующая теорема единственности.

Теорема 2.1.2. Если существует точка $x^* \in [x_1, x_2]$ такая, что $f(x^*) = 0$ и $f(x)$ монотонна на отрезке $[x_1, x_2]$, то x^* – единственный корень на отрезке $[x_1, x_2]$.

На практике при аналитическом отделении корней шаг разбиения отрезка берут достаточно малым.

Кроме аналитического отделения корней существует и графический способ, основанный на построении качественного графика функции $f(x)$ и приближенного (на глаз) определения точек пересечения графика с осью *Ox*. Если график функции $f(x)$ построить трудно, то представляют уравнение $f(x)=0$ в виде $\varphi(x)=g(x)$. И тогда решением уравнения будут абсциссы точек пересечения графиков $\varphi(x)$ и $g(x)$.

Пример. $x^2 - e^x = 0 \Leftrightarrow x^2 = e^x$.

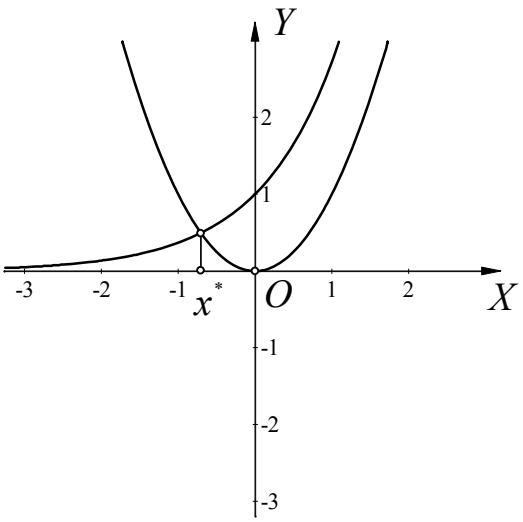


Рис. 2.1.1

Задача уточнения корней заключается в том, чтобы найти точку $\bar{x} \in [x_1, x_2]$ такую, что $|\bar{x} - x^*| < \varepsilon$, где $\varepsilon > 0$ – заданная точность решения уравнения $f(x)=0$.

Метод половинного деления

Если функция $f(x)$ является функцией общего вида и никакой дополнительной информацией о ней мы не располагаем, то метод половинного деления является оптимальным методом перебора. Суть его заключается в следующем. Пусть функция $f(x)$ непрерывна на отрезке $[a,b]$ и имеет на концах этого отрезка значения разных знаков. Предположим также, что задача локализации уже решена, то есть отрезок $[a,b]$ содержит единственный корень уравнения $f(x)=0$. В качестве начального приближения корня c_0 принимаем середину отрез-

ка $[a,b]$. То есть $c_0 = \dfrac{a+b}{2}$. Если $f(c_0) = 0$, то c_0 – точный корень уравнения. В противном случае находим знаки функции $f(x)$ на концах отрезков $[a,c_0]$ и $[c_0,b]$. Тот из них, на концах которого $f(x)$ принимает значения разных знаков, содержит искомый корень, поэтому его принимают в качестве нового отрезка $[a_1,b_1]$. Вторую половину отрезка $[a,b]$, на которой знак $f(x)$ не меняется, отбрасываем. В качестве первой итерации c_1 принимаем середину нового отрезка $[a_1,b_1]$ и так далее.

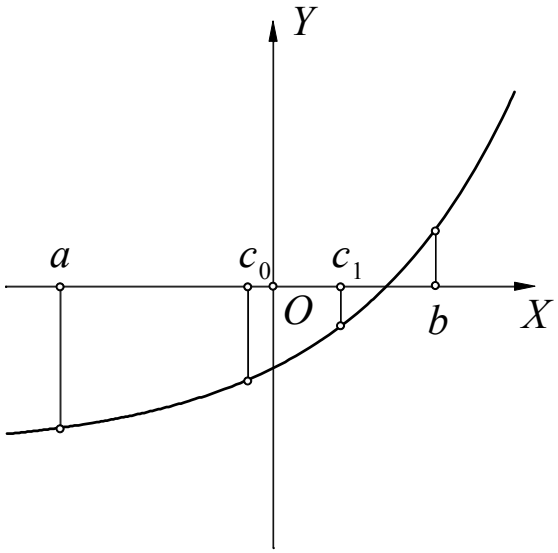


Рис. 2.1.2

Таким образом, после каждой итерации отрезок, на котором располагается корень, уменьшается вдвое, то есть после *n* итераций он сокращается в 2^n раз. Итерационный процесс следует прекратить, если $|b_n - a_n| < 2\varepsilon$. За приближенное значение корня надо принять c_n.

Заметим, что при рассмотрении практически любого численного метода следует четко знать ответы на следующие четыре вопроса.

1. Что взять в качестве нулевого приближения?
2. По какому принципу строить итерационный процесс?
3. Когда его следует остановить?
4. Что брать за приближенное решение задачи в случае остановки итерационного процесса?

При рассмотрении метода половинного деления мы ответили на все четыре вопроса.

2.2. Понятие метрического пространства. Теоретическое обоснование метода простых итераций

Понятие метрического пространства

Общая идея многих итерационных методов заключается в непосредственном построении последовательных приближений с помощью некоторого рекуррентного соотношения при заданном начальном приближении. При этом важным моментом является проблема сходимости, для исследования которой нам потребуются некоторые факты из функционального анализа.

Определение 2.2.1.Множество X называется метрическим пространством, если для любых x, y, принадлежащих X, определено число $\rho(x,y)\geq 0$, удовлетворяющее следующим условиям:

1. $\rho(x,y)=0 \Leftrightarrow x=y$ (аксиома тождества).
2. $\rho(x,y)=\rho(y,x)$ (аксиома симметрии).
3. $\rho(x,y)\leq \rho(x,z)+\rho(z,y)$ для любого $z\in X$ (аксиома треугольника).

Число $\rho(x,y)$ называется метрикой, или расстоянием между элементами x и y, которое определяет сходимость в пространстве X. Последовательность элементов $x_n\in X$ сходится к некоторому элементу $x\in X$, если $\rho(x,x_n)\to 0$ при $n\to\infty$.

Определение 2.2.2. Последовательность $\{x_n\}$ точек метрического пространства называется фундаментальной или сходящейся в себе, если $\rho(x_n,x_m)\to 0$ при $n\to\infty, m\to\infty$.

Очевидно, что всякая сходящаяся последовательность является фундаментальной, так как $\rho(x_n,x_m)\leq\rho(x_n,x)+\rho(x,x_m)$ (по правилу треугольника).

Обратное утверждение верно не всегда. Например, последовательность рациональных чисел $y_n=\left(1+\frac{1}{n}\right)^n$ является фундаментальной в метрике $\rho(r_1,r_2)=|r_1-r_2|$, но в пространстве рациональных чисел не является сходящейся, так как $\lim\limits_{n\to\infty} y_n=e$ – иррациональное число.

Определение 2.2.3. Метрическое пространство X называется полным метрическим пространством, если в нем каждая фундаментальная последовательность является сходящейся.

Примеры

1. Арифметическое n-мерное пространство R^n с метрикой $\rho(x,y)=\sqrt{\sum_{i=1}^{n}|\xi_i-\eta_i|^2}$, где $x=(\xi_1,\xi_2,\ldots,\xi_n)$, $y=(\eta_1,\eta_2,\ldots,\eta_n)$, является полным метрическим пространством (по критерию Коши).

2. Пространство непрерывных на $[a,b]$ функций $C_{[a,b]}$ с чебышевской метрикой $\rho(x,y)=\max\limits_{t\in[a,b]}|x(t)-y(t)|$ есть полное метрическое пространство (по критерию Коши равномерной сходимости последовательности функций).

3. Любое замкнутое подмножество полного метрического пространства является, в свою очередь, полным метрическим пространством (так, например, отрезок $[a,b]$ – полное метрическое пространство).

4. Метрическим, но неполным пространством является пространство рациональных чисел.

5. Полуинтервал $(0,1]$ не является полным метрическим пространством, так как фундаментальная последовательность $\left\{\frac{1}{n}\right\}$ не сходится в нем.

Теоретическое обоснование метода простых итераций

Определение 2.2.4. Отображение $\varphi: X\to X$ называют сжатым отображением в себе, если для любых x, y, принадлежащих X, выполняется условие $\rho(\varphi(x),\varphi(y))\le k\cdot\rho(x,y)$, где $0\le k<1$.

Число k называется коэффициентом сжатия отображения φ. Отображение φ может не быть сжатым на всем пространстве X, а лишь на некоторой его части $D\subset X$, когда неравенство $\rho(\varphi(x),\varphi(y))\le k\cdot\rho(x,y)$, $0\le k<1$, выполняется при всех $x,y\in D$.

Пусть $\bar{x}\in X$.

Определение 2.2.5. ε-окрестностью точки $\bar{x}$ называется множество точек x пространства X, которые удовлетворяют условию $\rho(x,\bar{x})\le\varepsilon$, $\varepsilon>0$.

Математическая запись этого определения выглядит следующим образом: $O_\varepsilon(\bar{x})=\{x\in X\,|\,\rho(x,\bar{x})\le\varepsilon\}$.

Определение 2.2.6. Неподвижной точкой отображения $\varphi: X \to X$ называется такая точка x^*, что $\varphi(x^*) = x^*$.

Замечание. Если представить уравнение $f(x) = 0$ в виде $x = \varphi(x)$, то решение уравнения сведется к поиску неподвижной точки отображения φ.

Теорема 2.2.1. Если отображение φ является сжатым и имеет неподвижную точку x^*, то любая ε-окрестность неподвижной точки отображается сама в себя, то есть $O_\varepsilon(x^*) \xrightarrow{\varphi} O_\varepsilon(x^*)$, для любого $\varepsilon > 0$.

Доказательство

Пусть дана $O_\varepsilon(x^*)$ – произвольная ε-окрестность точки x^*, и пусть точка $x \in O_\varepsilon(x^*)$ – произвольная точка из этой окрестности.

Докажем, что $\varphi(x)$ также принадлежит $O_\varepsilon(x^*)$. Действительно, по определению сжатых отображений можно записать

$$\rho(\varphi(x), x^*) = \rho(\varphi(x), \varphi(x^*)) \le k \cdot \rho(x, x^*), \text{ где } 0 \le k < 1.$$

$\rho(x, x^*) \le \varepsilon$, так как $x \in O_\varepsilon(x^*)$, учитывая, что $0 \le k < 1$, можно утверждать, $\rho(\varphi(x), x^*) \le \varepsilon$, то есть $\varphi(x) \in O_\varepsilon(x^*)$.

Принцип Банаха сжатых отображений устанавливает достаточное условие существования и единственности неподвижной точки сжатого отображения $\varphi: X \to X$, когда X является полным метрическим пространством.

Теорема 2.2.2 (принцип Банаха). Пусть $\varphi: X \to X$ – сжатое отображение полного метрического пространства X в себя с коэффициентом сжатия k. Тогда φ имеет одну неподвижную точку x^*, причем

1. $x^* = \lim\limits_{n \to \infty} x_n$, где x_0 – произвольная точка пространства X и $x_n = \varphi(x_{n-1}),\ n = 1,2,3,\ldots$
2. имеет место оценка для всех n:

$$\rho(x^*, x_n) \le \frac{k}{1-k} \rho(x_n, x_{n-1}).$$

Доказательство

I. Докажем, что существует не более одной неподвижной точки.

Допустим противное, то есть пусть существуют точки y_1 и y_2 такие, что $y_1 \ne y_2$, $y_1 = \varphi(y_1)$ и $y_2 = \varphi(y_2)$. Тогда

$$\rho(y_1, y_2) = \rho(\varphi(y_1), \varphi(y_2)) \le k \cdot \rho(y_1, y_2).$$

Получили противоречие, так как условие $\rho(y_1, y_2) \le k \cdot \rho(y_1, y_2)$, при $0 \le k < 1$ выполняться не может. Наше предположение было неверно.

II. Докажем фундаментальность последовательности $\{x_n\}$.

Не нарушая общности рассуждений, будем считать, что $n<m$, оценим $\rho(x_n,x_m)$.

$$\rho(x_n,x_m)=\rho(\varphi(x_{n-1}),\varphi(x_{m-1}))\le k\cdot\rho(x_{n-1},x_{m-1})\le$$
$$\le k^2\cdot\rho(x_{n-2},x_{m-2})\le\dots\le k^n\cdot\rho(x_0,x_{m-n})\le$$
$$\le k^n\left[\rho(x_0,x_1)+\rho(x_1,x_2)+\rho(x_2,x_3)+\dots+\rho(x_{m-n-1},x_{m-n})+\dots\right]\le$$
$$k^n\left(1+k+k^2+\dots\right)\cdot\rho(x_0,x_1)=\frac{k^n}{1-k}\cdot\rho(x_0,x_1).$$

Последнее равенство получается по формуле суммы бесконечной геометрической прогрессии с первым членом, равным 1: $1+k+k^2+\dots=\frac{1}{1-k}$, при $0\le k<1$. $\left(S_\infty=\frac{b_1}{1-q}\right)$.

Так как $\frac{\rho(x_0,x_1)}{1-k}=\text{const}$, $k^n\to 0$ при $n\to\infty$, то отсюда следует, что $\rho(x_n,x_m)\to 0$, при $n\to\infty,m\to\infty$. То есть $\{x_n\}$ – фундаментальная последовательность.

Так как X – полное метрическое пространство, то последовательность $\{x_n\}$ имеет в X предел, который мы обозначим через x^*.

III. Докажем, что x^* – неподвижная точка. Отображение $\varphi(x)$, будучи сжатым отображением, является непрерывным отображением. В равенстве $x_n=\varphi(x_{n-1})$ перейдем к пределу, получим $x^*=\varphi(x^*)$, то есть x^* – неподвижная точка отображения φ.

IV. Докажем теперь оценку $\rho(x^*,x_n)\le\frac{k}{1-k}\rho(x_n,x_{n-1})$.

Ранее было доказано, что $\rho(x_n,x_m)\le\frac{k^n}{1-k}\rho(x_0,x_1)$. Перейдем в этом неравенстве к пределу при $m\to\infty$. Получим $\rho(x_n,x^*)\le\frac{k^n}{1-k}\rho(x_0,x_1)$. Переобозначим: $\rho(x_l,x^*)\le\frac{k^l}{1-k}\rho(x_0,x_1)$. Так как за начальное приближение можно взять любую точку из X, возьмем в качестве x_0 значение x_{n-1} ($n-1$-ое приближение), тогда $x_1=x_n$, $x_l=x_{l+n-1}$. Имеем, таким образом, $\rho(x_{l+n-1},x^*)\le\frac{k^l}{1-k}\rho(x_{n-1},x_n)$. Это неравенство верно при любом натуральном l, а значит и при $l=1$, то есть $\rho(x^*,x_n)\le\frac{k}{1-k}\rho(x_n,x_{n-1})$.

Теорема доказана полностью.

Замечание. Принцип Банаха сжатых отображений имеет очень важное значение. Он утверждает, что если $\varphi(x)$ является сжатым отображением полного метрического пространства в себя, то неподвижную точку этого отображения можно найти с любой степенью точности, построив итерационную последовательность x_0, $x_1 = \varphi(x_0)$, $x_2 = \varphi(x_1),\ldots,\ x_n = \varphi(x_{n-1}),\ \ldots$.

Оценить степень приближения можно так:

$$\rho(x_n, x^*) \le \varepsilon \Leftarrow \frac{k}{1-k} \cdot \rho(x_{n-1}, x_n) < \varepsilon \Leftrightarrow \rho(x_{n-1}, x_n) < \frac{\varepsilon(1-k)}{k},$$

то есть если нужно найти приближение к неподвижной точке с точностью ε, то следует строить итерационный процесс до тех пор, пока расстояние между двумя приближениями не станет меньше $\frac{\varepsilon(1-k)}{k}$.

2.3. Метод простых итераций

Перейдем непосредственно к задаче решения уравнения $f(x) = 0$ методом простой итерации, основным моментом которого является сведение исходного уравнения к эквивалентному уравнению вида $x = \varphi(x)$.

$$f(x) = 0 \Leftrightarrow x = \varphi(x).$$

Пусть задача локализации корня уже решена, то есть известно, что единственный корень x^* уравнения $x = \varphi(x)$ находится в отрезке $[a,b]$. Используя принцип сжатых отображений Банаха, можно построить итерационный процесс $x_{n+1} = \varphi(x_n), n = 0,1,2,\ldots$ с любым начальным приближением $x_0 \in [a,b]$. Предел последовательности $\{x_n\}$ будет единственной неподвижной точкой отображения, то есть решением уравнения $x = \varphi(x)$, а значит, и уравнения $f(x) = 0$. Однако для этого нужно, чтобы отображение $\varphi(x)$ в уравнении $x = \varphi(x)$ было сжатым отображением $[a,b] \to [a,b]$.

В общем случае оставляем открытым вопрос о том, как свести уравнение $f(x) = 0$ к виду $x = \varphi(x)$ так, чтобы отображение $\varphi(x)$ было сжатым $[a,b] \to [a,b]$. Однако если функция $f(x)$ непрерывна вместе со своей первой производной на отрезке $[a,b]$ и $0 < m < f'(x) < M$ на $[a,b]$, то сведение уравнения $f(x) = 0$ к виду $x = \varphi(x)$ осуществляют следующим образом:

$$f(x)=0$$
$$\lambda f(x)=0$$
$$x=x+\lambda f(x)$$
$$x=\varphi(x),\ \text{где}\ \varphi(x)=x+\lambda f(x),$$
$$\varphi'(x)=1+\lambda f'(x)$$

Если в качестве константы λ взять $\lambda=-\frac{1}{M}$, то

$$\varphi'(x)=1+\lambda f'(x)=1-\frac{1}{M}f'(x)<1-\frac{m}{M}<1,$$
$$\varphi'(x)=1+\lambda f'(x)=1-\frac{1}{M}f'(x)>1-\frac{M}{M}=0$$

То есть $0<\varphi'(x)<k=1-\frac{m}{M}<1$.

1. Докажем сперва, что φ есть отображение $[a,b]\to[a,b]$, то есть что для любого $x\in[a,b]$ $\varphi(x)$ также принадлежит $[a,b]$. Так как $\varphi'(x)>0$ на отрезке $[a,b]$, то $\varphi(x)$ возрастает на $[a,b]$. Так как

$$a\le x^*\le b$$

(отрезок $[a,b]$ содержит корень x^* по предположению), то

$$\varphi(a)\le\varphi(x^*)\le\varphi(b).$$

$$x^*-\varphi(a)=\varphi(x^*)-\varphi(a)=[\text{по т. Лагранжа}\ \exists\xi\in[a,b]\,]=\varphi'(\xi)(x^*-a)\le x^*-a$$

То есть $\varphi(a)$ находится от x^* не далее, чем a.

$$\varphi(b)-x^*=\varphi(b)-\varphi(x^*)=[\ \text{по т. Лагранжа}\ \exists\eta\in[a,b]\,]=\varphi'(\eta)(b-x^*)\le b-x^*$$

То есть $\varphi(b)$ находится от x^* не далее, чем b.

Следовательно, $[\varphi(a),\varphi(b)]\subseteq[a,b]$. Так как $x\in[a,b]$ и $\varphi(x)$ возрастает на $[a,b]$, то $\varphi(a)\le\varphi(x)\le\varphi(b)$, то есть $\varphi(x)\in[\varphi(a),\varphi(b)]$. А значит, $\varphi(x)\in[a,b]$. Итак, отображение φ отображает отрезок $[a,b]$ в $[a,b]$.

2. Докажем теперь, что φ есть сжатое отображение $[a,b]\to[a,b]$. По теореме Лагранжа, для любых $x,y\in[a,b]$ найдется точка $\xi\in[a,b]$ такая, что $|\varphi(x)-\varphi(y)|=|\varphi'(\xi)|\cdot|x-y|$, то есть $\rho(\varphi(x),\varphi(y))=|\varphi'(\xi)|\rho(x,y)$.

Так как $0<\varphi'(x)<k<1$ на $[a,b]$, то $\rho(\varphi(x),\varphi(y))\le k\cdot\rho(x,y)$, где $0<k<1$. Итак, условие сжатости отображения $[a,b]\to[a,b]$ доказано.

Заметим, что если функция $f(x)$ будет удовлетворять условию $-M<f'(x)<-m<0$, то аналогично можно показать, что в качестве коэффициента λ следует взять $\lambda=\frac{1}{M}$.

Пример

Написать программу на языке «Паскаль» для решения уравнения $x^3 + 2x^2 + 2 = 0$ методом простых итераций с точностью $\varepsilon = 0{,}001$, если известно, что корень уравнения находится на отрезке $[-3,-2]$.

Легко показать, что $f'(x) = 3x^2 + 4x > 0$ на рассматриваемом отрезке, а значит, в задаче можно применять описанную выше схему. Абсцисса $-\frac{2}{3}$ вершины параболы $y = 3x^2 + 4x$ не входит в отрезок $[-3,-2]$, а значит,

$$m = f'(-2) = 4,\ M = f'(-3) = 15,\ \lambda = -\frac{1}{M} = -\frac{1}{15},\ k = 1 - \frac{m}{M} = \frac{11}{15}.$$

Составим программу.

```
Uses crt;
Const e=0.001; la=-1/15; k=11/15;
Var x1,x2:real;
Function f(x:real):real;
    Begin
    F:=x*sqr(x)+2*sqr(x)+2;
    End;
Function fi(x:real):real;
    Begin
    Fi:=x+la*f(x);
    End;

Begin
ClrScr;
X2:=-2.5;
Repeat
    x1:=x2;
    x2:=fi(x1);
Until abs(x2-x1)<e*(1-k)/k;
WriteLn(x2:3:3);
ReadKey;
End.
```

После выполнения программы будет получен ответ $-2{,}360$.

2.4. Метод Ньютона. Оценка погрешности метода Ньютона

Метод Ньютона

Рассмотрим уравнение $f(x)=0$, причем $f(x)$ удовлетворяет следующим условиям: $f(x)\in C^2_{[a,b]}$, $f(a)\cdot f(b)<0$, то есть функция $f(x)$ принимает на концах отрезка $[a,b]$ значения с противоположными знаками, производные $f'(x)$ и $f''(x)$ сохраняют знак в отрезке $[a,b]$.

При этих условиях возможны четыре случая, указанные на рисунках 2.4.1, 2.4.2, 2.4.3, 2.4.4.

Заметим, что на рисунке 2.4.1 функция $f(x)$ возрастает и вогнута, на рисунке 2.4.2 – убывает и вогнута, на рисунке 2.4.3 – возрастает и выпукла, на рисунке 2.4.4 – убывает и выпукла.

1. $f'(x)>0$, $f''(x)>0$.

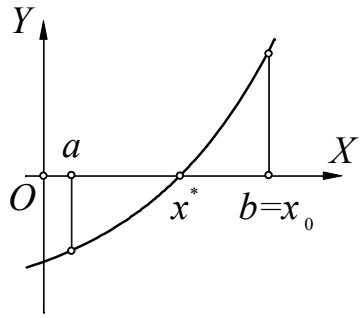


Рис. 2.4.1

2. $f'(x)<0$, $f''(x)>0$.

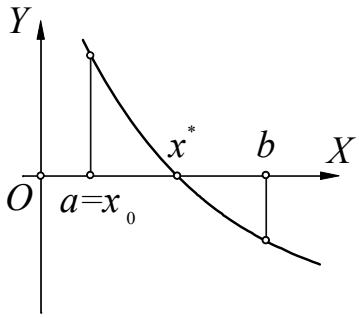


Рис. 2.4.2

3. $f'(x)>0$, $f''(x)<0$.

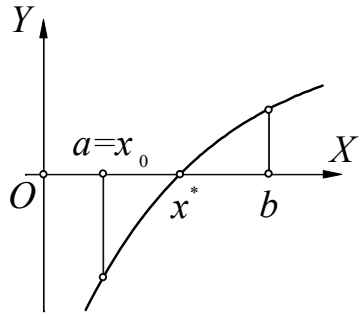


Рис. 2.4.3

4. $f'(x)<0$, $f''(x)<0$.

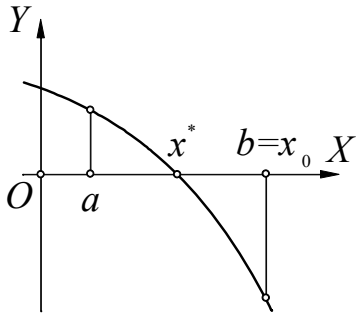


Рис. 2.4.4

Примем за x_0 тот конец отрезка $[a,b]$, в котором функция $f(x)$ имеет тот же знак, что и $f''(x)$. Метод Ньютона, называемый также методом касательных, состоит в следующем. Рассмотрим в точке x_0 касательную к кривой $y=f(x)$, задаваемую уравнением

$y = f(x_0) + (x - x_0) \cdot f'(x_0)$. Положив $y = 0$, найдем точку пересечения касательной с осью *Ox*.

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}.$$

Построив касательную в точке x_1, получаем по аналогичной формуле точку пересечения этой касательной с осью *Ox*.

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

Продолжая этот процесс, получаем:

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

Полученная итерационная последовательность является убывающей (возрастающей) и ограниченной снизу (сверху). По соответствующей теореме из анализа эта последовательность имеет предел $\bar{x}$. Покажем, что он равен корню уравнения.

Перейдем в равенстве $x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})}$ к пределу, пользуясь непрерывностью $f(x)$ и $f'(x)$. Имеем $\bar{x} = \bar{x} - \frac{f(\bar{x})}{f'(\bar{x})}$. Из последнего равенства следует, что $f(\bar{x}) = 0$, то есть $\bar{x} = x^*$.

Оценка погрешности метода Ньютона

Для характеристики приближенных методов решения уравнений вводится понятие порядка сходимости метода.

Определение 2.4.1. Говорят, что метод имеет *k*-й порядок сходимости, если существуют $c_1 > 0$, $c_2 > 0$ такие, что $\rho(x_n, x^*) \le c_2 \left[\rho(x_{n-1}, x^*)\right]^k$ при условии $\rho(x_{n-1}, x^*) \le c_1$.

Очевидно, что чем больше *k*, тем быстрее сходится процесс итераций. В вычислительной практике широко распространены методы второго порядка.

Теорема 2.4.1. Для метода Ньютона имеет место следующая оценка:

$$\left|x_n - x^*\right| \le \frac{M_2}{2m_1} \cdot \left|x_n - x_{n-1}\right|^2, \text{ где } M_2 = \max_{a \le x \le b} \left|f''(x)\right|,\ m_1 = \min_{a \le x \le b} \left|f'(x)\right|.$$

Доказательство

По формуле конечных приращений Лагранжа, $f'(c) = \frac{f(b) - f(a)}{b - a}$, для некоторого $c \in [a,b]$ найдется точка $\xi \in \left[x_n, x^*\right]$ (или $\xi \in \left[x^*, x_n\right]$) та-

кая, что $\left|f(x_n)-f(x^*)\right|=\left|f'(\xi)\right|\cdot\left|x_n-x^*\right|$. Учитывая, что $f(x^*)=0$, получим $\left|f(x_n)\right|=\left|f'(\xi)\right|\cdot\left|x_n-x^*\right|$. Выразим из последней формулы модуль разности между n-м приближением и истинным корнем: $\left|x_n-x^*\right|=\dfrac{\left|f(x_n)\right|}{\left|f'(\xi)\right|}\le\dfrac{\left|f(x_n)\right|}{m_1}$.

Запишем формулу Тейлора в окрестности точки x_{n-1}.

$$f(x)=f(x_{n-1})+f'(x_{n-1})\cdot(x-x_{n-1})+\frac{f''(\theta)}{2!}\cdot(x-x_{n-1})^2.$$

Тогда

$$f(x_n)=f(x_{n-1})+f'(x_{n-1})\cdot(x_n-x_{n-1})+\frac{f''(\theta)}{2!}\cdot(x_n-x_{n-1})^2,$$

где $\dfrac{f''(\theta)}{2!}\cdot(x_n-x_{n-1})^2$ – остаточный член формулы Тейлора, $\theta\in[x_n,x_{n-1}]$ (или $\theta\in[x_{n-1},x_n]$).

В силу самого метода Ньютона $f(x_{n-1})+f'(x_{n-1})\cdot(x_n-x_{n-1})=0$. Тогда $\left|f(x_n)\right|\le\dfrac{M_2}{2}\cdot\left|x_n-x_{n-1}\right|^2$. Используя полученное неравенство и выведенное ранее $\left(\left|x_n-x^*\right|\le\dfrac{\left|f(x_n)\right|}{m_1}\right)$, можно записать $\left|x_n-x^*\right|\le\dfrac{M_2}{2m_1}\cdot\left|x_n-x_{n-1}\right|^2$.

Теорема 2.4.2. Метод Ньютона является методом второго порядка сходимости.

Доказательство

Полученную выше оценку погрешности метода Ньютона запишем в виде: $\rho(x_n,x^*)\le\dfrac{M_2}{2m_1}\cdot\rho^2(x_n,x_{n-1})$.

По неравенству треугольника имеем $\rho(x_n,x_{n-1})\le\rho(x_n,x^*)+\rho(x^*,x_{n-1})$. Так как $\rho(x^*,x_{n-1})\ge\rho(x^*,x_n)$, то $\rho(x_n,x_{n-1})\le2\cdot\rho(x_{n-1},x^*)$. Возвращаясь к первоначальной оценке метода Ньютона, имеем:

$$\rho(x_n,x^*)\le\frac{M_2}{2m_1}\cdot4\cdot\rho^2(x_{n-1},x^*).$$

Обозначив $c_2=\dfrac{M_2}{2m_1}\cdot4$, получаем $\rho(x_n,x^*)\le c_2\cdot\rho^2(x_{n-1},x^*)$. Таким образом, метод Ньютона является методом второго порядка и сходится гораздо быстрее, чем другие методы приближенного решения уравнений.

Из оценки метода Ньютона $\left|x_n-x^*\right|\le\dfrac{M_2}{2m_1}\cdot\left|x_n-x_{n-1}\right|^2$ можно заключить, что итерации следует завершать, если выполнилось условие

$\frac{M_2}{2m_1}|x_n - x_{n-1}|^2 < \varepsilon$, то есть $|x_n - x_{n-1}| < \sqrt{\frac{2m_1\varepsilon}{M_2}}$. Однако для использования этого условия приходится находить величины m_1 и M_2, что не всегда бывает просто. Поэтому иногда это условие нестрого упрощают и записывают в виде $|x_n - x_{n-1}| < \varepsilon$. Очевидно, что

$$|x_n - x_{n-1}| < \varepsilon \Rightarrow |x_n - x_{n-1}| < \sqrt{\frac{2m_1\varepsilon}{M_2}},$$

если $\varepsilon^2 < \frac{2m_1\varepsilon}{M_2}$, то есть, если $\varepsilon < \frac{2m_1}{M_2}$.

Пример

Определить условие выхода из цикла при решении уравнения $x^3 + 2x^2 + 2 = 0$ методом Ньютона, зная, что корень уравнения находится на отрезке $[-3,-2]$.

Пусть $f(x) = x^3 + 2x^2 + 2$, тогда $f'(x) = 3x^2 + 4x$, $f''(x) = 6x + 4$. Абсцисса вершины параболы $y = 3x^2 + 4x$ не принадлежит рассматриваемому отрезку, а значит, $f'(x)$ достигает наибольшего и наименьшего значений на концах отрезка: $f'(-3) = 15$, $f'(-2) = 4$, $m_1 = \min\limits_{[-3,-2]} |f'(x)| = 4$. $f''(x)$ – линейная функция, а, значит, и она достигает наибольшего и наименьшего значений на концах отрезка: $f''(-3) = -14$, $f''(-2) = -8$, $M_2 = \max\limits_{[-3,-2]} |f''(x)| = 14$. $\sqrt{\frac{2m_1\varepsilon}{M_2}} = \sqrt{\frac{4}{7}\varepsilon}$, а значит, условие выхода из цикла:

$$|x_n - x_{n-1}| < \sqrt{\frac{4}{7}\varepsilon}.$$

Заметим, что поскольку $\frac{4}{7} > 0{,}1$, то, упростив условие выхода из цикла до

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

мы бы не ошиблись при приближенном нахождении корня с точностью 0,1 и любой лучшей.

2.5. Лабораторная работа «Методы решения нелинейных уравнений»

(4 часа)

Цель работы: научиться строить графики функций одной независимой переменной в собственных программах и в программе MathCad, научиться численно и символически решать уравнения в

программе MathCad, научиться реализовывать методы половинного деления, простых итераций и Ньютона в собственных программах.

Используемое программное обеспечение: Borland Pascal (или Delphi) и MathCad.

Основное задание:

1. Отделите корни каждого из уравнений графически с точностью до 1 в программе MathCad; уточните корни каждого из уравнений в программе MathCad с точностью до 0,0001; решите в программе MathCad символически уравнение $x^2 - nax + 2 = 0$, где n – номер варианта, a – параметр (задания по вариантам смотрите в таблице 2.5.1).

2. Уточните корни каждого из уравнений методом половинного деления с точностью до 0,001.

3. Уточните какой-нибудь корень одного из уравнений методом Ньютона с точностью до 0,001.

Все уравнения следует рассматривать лишь на отрезке $[-10,10]$.

Таблица 2.5.1

Задания к лабораторной работе

(для тридцати вариантов)

№	**Задание**	**№**	**Задание**
1	*2*	*3*	*4*
1	1) $2^x + 5x - 3 = 0$ 2) $3x^4 + 4x^3 - 12x^2 - 5 = 0$	2	1) $[\log_2(-x)] \cdot (x+2) = -1$ 2) $2x^3 - 9x^2 - 60x + 1 = 0$
3	1) $5^x + 3x = 0$ 2) $x^4 - x - 1 = 0$	4	1) $x \cdot \log_3(x+1) = 1$ 2) $2x^4 - x^2 - 10 = 0$
5	1) $3^{x-1} - 2 - x = 0$ 2) $3x^4 + 8x^3 + 6x^2 - 10 = 0$	6	1) $x^2 \cdot 2^x = 1$ 2) $x^4 - 18x^2 + 6 = 0$
7	1) $0{,}5^x - 1 = (x+2)^2$ 2) $x^4 + x^3 - 8x^2 - 17 = 0$	8	1) $5^x - 6x - 3 = 0$ 2) $x^4 - x^3 - 2x^2 + 3x - 3 = 0$
9	1) $(x-2)^2 \cdot 2^x = 1$ 2) $3x^4 + 4x^3 - 12x^2 + 1 = 0$	10	1) $2\lg x - \frac{x}{2} + 1 = 0$ 2) $3x^4 - 8x^3 - 18x^2 + 2 = 0$
11	1) $3^x + 2x - 2 = 0$ 2) $2x^4 - 4x^3 + 8x^2 - 1 = 0$	12	1) $[\log_2(x+2)](x-1) = 1$ 2) $2x^4 + 8x^3 + 8x^2 - 1 = 0$
13	1) $3^x + 2x - 5 = 0$ 2) $x^4 - 4x^3 - 8x^2 + 1 = 0$	14	1) $x\log_3(x+1) = 2$ 2) $3x^4 + 4x^3 - 12x^2 - 5 = 0$
15	1) $3^{x-1} - 4 - x = 0$ 2) $2x^3 - 9x^2 - 60x + 1 = 0$	16	1) $(x-1)^2 2^x = 1$ 2) $x^4 - x - 1 = 0$

Окончание таблицы 2.5.1

1	*2*	*3*	*4*
17	1) $0{,}5^x - 3 = (x+2)^2$ 2) $2x^4 - x^2 - 10 = 0$	18	1) $3^x - 2x - 5 = 0$ 2) $3x^4 + 8x^3 + 6x^2 - 10 = 0$
19	1) $(x-2)^2 2^x = 1$ 2) $x^4 - 18x^2 + 6 = 0$	20	1) $2\lg x - \frac{x}{2} + 1 = 0$ 2) $x^4 + 4x^3 - 8x^2 - 17 = 0$
21	1) $2^x - 3x - 2 = 0$ 2) $x^4 - x^3 - 2x^2 + 3x - 3 = 0$	22	1) $(x+2)\log_2(x) = 1$ 2) $3x^4 + 4x^3 - 12x^2 + 1 = 0$
23	1) $3^x + 2x - 3 = 0$ 2) $3x^4 - 8x^3 - 18x^2 + 2 = 0$	24	1) $x\log_3(x+1) = 1$ 2) $3x^4 + 4x^3 - 12x^2 - 5 = 0$
25	1) $3^x + 2 + x = 0$ 2) $2x^3 - 9x^2 - 60x + 1 = 0$	26	1) $(x-1)^2 2^x = 1$ 2) $x^4 - x - 1 = 0$
27	1) $0{,}5^x - 3 = -(x+1)^2$ 2) $2x^4 - x^2 - 10 = 0$	28	1) $3^x - 2x - 5 = 0$ 2) $3x^4 + 8x^3 + 6x^2 - 10 = 0$
29	1) $(x-2)^2 2^x = 1$ 2) $x^4 - 18x^2 + 6 = 0$	30	1) $3^x + 5x - 2 = 0$ 2) $3x^4 + 4x^3 - 12x^2 + 1 = 0$

Комментарии к основному заданию

1. Для выполнения работы в программе MathCad понадобятся панели инструментов «*Графики*» и «*Арифметика*». Вывести их на экран можно через пункт меню «*Вид / Панель инструментов*». Программу желательно составлять так, чтобы она обладала универсальностью. В том случае, если придется изменить уравнение, то пусть в программе при этом придется сделать минимум изменений.

Программа MathCad различает большие и маленькие буквы. Построить декартов график можно с помощью команды панели графиков «*Декартов график*». При этом следует указать в нижнем поле ввода имя независимой переменной и в левом – функцию. При наборе функции удобно пользоваться кнопками на панели «*Арифметика*».

Наша цель: с помощью графика решить задачу локализации корней, поэтому после того, как график будет построен, следует привести его к такому виду, чтобы можно было с уверенностью назвать отрезки длиной 1, на которых находится каждый из корней уравнения. Для этого можно изменить диапазон изменения аргумента и функции в области отображения графика и изменить свойства отображения графика: показать координатные оси, масштабную сетку и т. д. Это можно сделать, два раза «кликнув» по графику. Если известно, что корень уравнения $f(x) = 0$ находится на отрезке $[a,b]$, то его мож-

но найти командой `root`. Первый ее параметр – функция $f(x)$, второй – имя независимой переменной, в нашем случае x, третий и четвертый параметры – границы отрезка a и b. Так, например, один из корней уравнения $x^2 = 3$ можно найти так: `root(`x^2`-3, x, 1, 2)`. Заметим, что программа MathCad может проводить два типа вычислений: численные и символьные. Если результатом какой-либо операции является число и нас устраивает, чтобы оно отобразилось в виде десятичной дроби, то речь идет о численных вычислениях. В этом случае после того, как математическое выражение будет набрано, следует подвести курсор синего цвета в его конец и нажать клавишу «=». Так, например, можно поступить в случае приближенного нахождения корней указанных уравнений. Если же результатом какой-либо операции является не число, а, например, функция, то речь идет о символьных вычислениях. В этом случае также следует подвести курсор в конец выражения, но затем выбрать пункт меню «*Символы / Расчеты / Символически*» или нажать комбинацию клавиш «Shift+F9». Точно так же следует воспользоваться символическими вычислениями, если результатом некоторой операции является число, но мы хотим, чтобы оно было показано точно, в символах, например, «$\sqrt{3}$», а не «1.732». Поэтому для выполнения последнего задания первого пункта следует воспользоваться командой `root` без третьего и четвертого параметров и провести символьные расчеты.

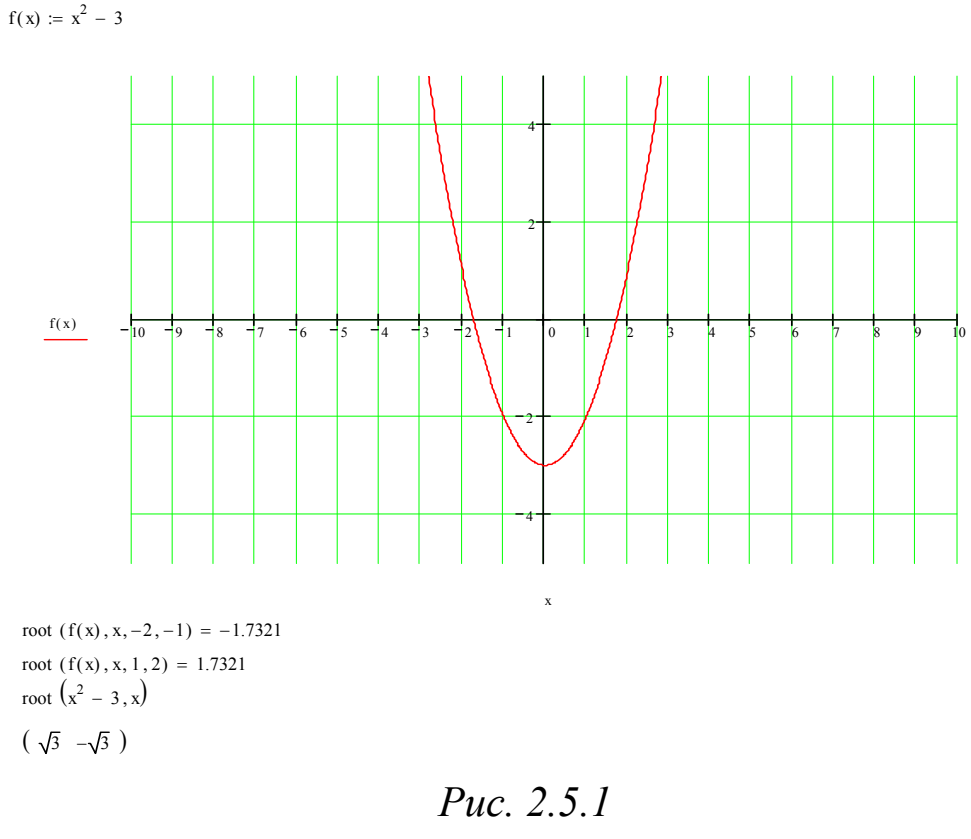


Рис. 2.5.1

По умолчанию программа MathCad указывает результат, округленный до тысячных. Изменить требуемую точность результата можно, два раза «кликнув» по нему. В появившемся окошке следует указать количество десятичных знаков после запятой и желательно сделать активной ячейку «Показать конечные нули».

Выше приведен листинг программы для нахождения корней уравнения $x^2 = 3$ (рис. 2.5.1).

2. При выполнении задания 2 подразумевается, что задача локализации корней уже решена. Для удобства тестирования программы можно задать точность вычислений и границы отрезка, на котором находится искомый корень, константами. Функцию $f(x)$ желательно описать как функцию с помощью служебного слова «function». Окончательный результат следует округлить до нужного числа знаков после запятой. Напомним, что при этом возникает погрешность, которую тоже следует учесть. После того, как программа выдаст некоторый результат, сравните его с результатом, полученным в программе MathCad. Заметим, что ваш ответ *может* отличаться от истинного, но на величину, *не большую, чем заданная точность*. Так, например, если при одинаковой точности вычислений $\varepsilon = 0{,}0001$ программой MathCad и вашей программой были получены соответственно ответы «1.7321» и «1.7320», это не означает, что ваш ответ неверен. Но, увеличив точность, в программе MathCad вы можете получить практически точный ответ. И если ваш ответ будет отличаться от него на величину, большую ε, это означает, что ваша программа содержит ошибку.

3. При выполнении задания 3 также подразумевается, что задача локализации уже решена и, таким образом, известен отрезок, на котором находится единственный искомый корень уравнения. Перед написанием программы следует убедиться, что ваше уравнение на рассматриваемом отрезке удовлетворяет требованиям метода Ньютона. При желании это можно было бы сделать программно, однако, например, по графику, построенному в MathCad'е, это будет сделать намного проще. Если вдруг не все требования выполняются (например, рассматриваемый отрезок содержит точку экстремума), то можно попытаться сузить отрезок. Функцию $f(x)$, а также ее первую и вторую производные желательно описать как отдельные функции. Производные можно найти аналитически и просто запрограммировать полученные функции. Можно производные считать приближенно программно, используя определение производной. Так, значение первой производной функции $f(x)$ в точке x можно запрограммировать как

$$f1(x) := \frac{f(x+h) - f(x)}{h},$$

где h – некоторая малая величина. Тогда вторая производная функции $f(x)$ в точке x будет равна

$$f2(x) := \frac{f1(x+h) - f1(x)}{h}.$$

Однако при каждом обращении к таким функциям будет возникать ошибка, которые в совокупности могут исказить окончательный результат. В этом случае и если мы будем пользоваться упрощенным условием выхода из цикла в методе Ньютона $|x_n - x_{n-1}| < \varepsilon$, окончательный результат в принципе может отличаться от истинного на величину, большую ε.

Дополнительное задание

1. Напишите собственную программу для построения графиков в декартовой системе координат.

2. Напишите собственную программу, которая табулировала бы функцию на отрезке [–10,10] с шагом 1 и, сравнивая знаки функции на концах каждого элементарного отрезка, решала бы программно задачу локализации корней.

3. Решите одно из нижеуказанных уравнений методом простых итераций (табл. 2.5.2).

Таблица 2.5.2

Дополнительные задания

1	$x^3 - 3x^2 + 9x - 10 = 0$	2	$x^3 - 2x + 2 = 0$
3	$x^3 + 3x - 1 = 0$	4	$x^3 + x - 3 = 0$
5	$x^3 + 0{,}4x^2 + 0{,}6x - 1{,}6 = 0$	6	$x^3 - 0{,}2x^2 + 0{,}4x - 1{,}4 = 0$
7	$x^3 - 0{,}1x^2 + 0{,}4x + 2 = 0$	8	$x^3 + 3x^2 + 12x + 3 = 0$
9	$x^3 - 0{,}2x^2 + 0{,}5x - 1 = 0$	10	$x^3 - 0{,}1x^2 + 0{,}4x + 1{,}2 = 0$

4. Усовершенствуйте свою программу для построения графиков, добавив в нее следующие возможности: динамическое изменение масштабов координатных осей, динамическое задание функций (например, с помощью готовых компонентов или модулей); построение графиков функций в полярной системе координат и заданных параметрически; построение графиков функций, заданных неявно (при этом придется численно решать ряд уравнений). В этом случае вы

будете в одном шаге от написания программы для построения поверхностей методом сечений.

Комментарии к дополнительному заданию

1. При написании данной программы следует иметь в виду, что вам придется работать с двумя системами координат: «экранной», которая «состоит» из пикселей, и «истинной», или «бумажной», переводя координаты из одной в другую. «Ядро» программы на языке «Паскаль» может выглядеть так:

```
for i:=0 to 640 do {перебираем «экранные» пиксели}
  begin
  x:=(i-320)/32; {переводим «экранную» координату x в}
                 {истинную, считая, что одна единица}
                 {по оси x содержит 32 пикселя}
  y:=f(x); {вычисляем «истинное» значение функции}
  j:=round(240-24*y); {переводим «истинную»}
                      {координату y в «экранную»,}
                      {считая, что одна единица по}
                      {оси y содержит 24 пикселя}
  putpixel(i,j,10); {изображаем точку на экране}
                    {зеленым цветом}
  end;
```

2. Следует обратить внимание на тот случай, когда один из корней в точности совпадает с концом элементарного отрезка.

3. Напомним, что перед написанием программы следует преобразовать уравнение $f(x)=0$ к виду $x=\varphi(x)$ так, чтобы отображение φ $[a,b]\to[a,b]$ было сжатым.

4. Помощь по «вычислению строки» есть, например, в системе управления архивом статей «Delphi World 6». Что же касается построения графиков функций, заданных неявно, заметим следующее. Если функция задана уравнением $F(x,y)=0$, то при фиксированном $x=x_0$ мы получаем уравнение с одним неизвестным $F(x_0,y)=0$, решив которое, мы найдем точку (точки) для изображения на экране.

ГЛАВА 3. ЧИСЛЕННЫЕ МЕТОДЫ РЕШЕНИЯ СИСТЕМ УРАВНЕНИЙ

3.1. Постановка задачи. Метод Гаусса с выбором главного элемента

Постановка задачи

Так как существует целый класс задач, сводящихся к решению систем линейных алгебраических уравнений, в которых число уравнений совпадает с числом неизвестных, а определитель системы отличен от нуля, то, прежде всего, следует научиться решать такие системы, тем более, что системы нелинейных уравнений часто сводятся к системам линейных уравнений.

Итак, пусть требуется решить систему линейных алгебраических уравнений $A \cdot X = B$, где A – матрица коэффициентов системы $m \times m$, $X = (x_1, x_2, \ldots, x_m)^T$ – искомый вектор, $B = (b_1, b_2, \ldots, b_m)^T$ – заданный вектор правых частей. Предположим, что определитель матрицы A отличен от 0 $(\det(A) \neq 0)$ и, следовательно, решение X существует и единственно.

Для большинства вычислительных задач характерным является большой порядок матрицы A. Из курса алгебры известно, что систему $A \cdot X = B$ можно решить, по крайней мере, тремя способами: по формулам Крамера, матричным или методом последовательного исключения неизвестных Гаусса. Из них наиболее удобным для реализации на ЭВМ является метод Гаусса.

Методы численного решения систем $A \cdot X = B$ делятся на две группы: прямые и итерационные. В прямых (или точных) методах решение X системы $A \cdot X = B$ находится за конечное число арифметических действий. Примером прямого метода является метод Гаусса. Отметим, что вследствие погрешностей округлений при решении задач на ЭВМ, прямые методы на самом деле не приводят к точному решению системы $A \cdot X = B$ и называть их точными можно, только отвлекаясь от погрешностей округления.

Итерационные методы (методы последовательных приближений) состоят в том, что решение x – системы $A \cdot X = B$ находится как предел при $n \to \infty$ последовательности приближений $x^{(n)}$, где n – номер итерации. Как правило, за конечное число итераций этот предел не достигается. Обычно задается некоторое малое число ε, и вычисления проводятся до тех пор, пока не будет

выполнено условие $\left\|x^{(n)}-x\right\|<\varepsilon$, где $\|x\|$ – одна из норм в пространстве R^m, например: $\|x\|=\max\limits_{1\le i\le m}|x_i|$ или $\|x\|=\sqrt{\sum\limits_{i=1}^{m}x_i^2}$.

Прямые методы на практике применяются для матриц умеренного порядка (m – порядка 100). Для матриц высокого порядка предпочтительнее итерационные методы.

Метод Гаусса с выбором главного элемента

При «обычном» применении метода Гаусса на k-том шаге исключается k-тое неизвестное. Это возможно, если коэффициент при k-том неизвестном отличен от нуля. Условием применимости такого метода является неравенство нулю всех угловых миноров матрицы A, то есть $a_{11}\neq 0$, $\begin{vmatrix}a_{11} & a_{12}\\ a_{21} & a_{22}\end{vmatrix}\neq 0$, $\begin{vmatrix}a_{11} & a_{12} & a_{13}\\ a_{21} & a_{22} & a_{23}\\ a_{31} & a_{32} & a_{33}\end{vmatrix}\neq 0$,..., $\det(A)\neq 0$. Однако, может оказаться, что система $A\cdot X=B$ имеет единственное решение, хотя какой-либо из угловых миноров матрицы A равен 0. Кроме того, заранее обычно неизвестно, все ли угловые миноры матрицы A отличны от 0. Отметим также, что если в процессе вычислений встречаются ведущие элементы, которые достаточно малы по сравнению с другими элементами матрицы, то это способствует увеличению погрешностей округлений (при делении на маленькие числа погрешность возрастает).

Избежать указанных недостатков «обычного» метода Гаусса позволяет метод Гаусса с выбором главного элемента.

Пусть, как и прежде, дана система $A\cdot X=B$. Сначала добиваются выполнения условий $\left|a_{11}^{(11)}\right|\ge\left|a_{i1}^{(11)}\right|$, $i=\overline{2,m}$ путем перестановки в случае необходимости двух уравнений системы. Найденный максимальный по модулю коэффициент, обозначенный при перенумерации через $a_{11}^{(11)}$, называют первым главным элементом. Исключив x_1 из всех уравнений, начиная со второго, получим систему:

$$\begin{cases}a_{11}^{(11)}x_1+a_{12}^{(11)}x_2+a_{13}^{(11)}x_3+\ldots+a_{1m}^{(11)}x_m=b_1^{(11)}\\ a_{22}^{(12)}x_2+a_{23}^{(12)}x_3+\ldots+a_{2m}^{(12)}x_m=b_2^{(12)}\\ a_{32}^{(12)}x_2+a_{33}^{(12)}x_3+\ldots+a_{3m}^{(12)}x_m=b_3^{(12)}\\ \ldots\\ a_{m2}^{(12)}x_2+a_{m3}^{(12)}x_3+\ldots+a_{mm}^{(12)}x_m=b_m^{(12)}\end{cases}.$$

Далее с полученной системой без первого уравнения поступим аналогично, как и со всей системой $A \cdot X = B$. А именно, осуществив, если нужно, перестановку двух уравнений и произведя соответствующую перенумерацию, обеспечиваем выполнение неравенств $\left|a_{22}^{(21)}\right| \geq \left|a_{i2}^{(21)}\right|$, $i = \overline{3,m}$. Найденный максимальный по модулю коэффициент, обозначенный $a_{22}^{(21)}$, называется вторым главным элементом. Исключив x_2 из всех уравнений, начиная с третьего, получим систему:

$$\begin{cases} a_{11}^{(11)}x_1 + a_{12}^{(11)}x_2 + a_{13}^{(11)}x_3 + \ldots + a_{1m}^{(11)}x_m = b_1^{(11)} \\ a_{22}^{(21)}x_2 + a_{23}^{(21)}x_3 + \ldots + a_{2m}^{(21)}x_m = b_2^{(21)} \\ a_{33}^{(22)}x_3 + \ldots + a_{3m}^{(22)}x_m = b_3^{(22)} \\ \ldots \\ a_{m3}^{(22)}x_3 + \ldots + a_{mm}^{(22)}x_m = b_m^{(22)} \end{cases}.$$

Если определитель системы $A \cdot X = B$ отличен от нуля, то после $(m-1)$-го шага будет получена система вида:

$$\begin{cases} a_{11}^{(11)}x_1 + a_{12}^{(11)}x_2 + a_{13}^{(11)}x_3 + \ldots + a_{1m}^{(11)}x_m = b_1^{(11)} \\ a_{22}^{(21)}x_2 + a_{23}^{(21)}x_3 + \ldots + a_{2m}^{(21)}x_m = b_2^{(21)} \\ a_{33}^{(31)}x_3 + \ldots + a_{3m}^{(31)}x_m = b_3^{(31)} \\ \ldots \\ a_{mm}^{(m-1,2)}x_m = b_m^{(m-1,2)} \end{cases}.$$

Заметим, что можно на *k*-м шаге искать главный элемент не только в *k*-ом столбце на местах ниже диагонального, а во всех столбцах, начиная с *k*-го и кончая *m*-м, и во всех строках, начиная с *k*-ой, кончая *m*-ой. Преимущество этой модификации заключается в том, что погрешность округлений будет еще меньшей, однако этот метод не очень удобен для реализации вследствие перестановок столбцов, что приведет к перенумерации не только коэффициентов, но и неизвестных.

Описанные выше рассуждения называют прямым ходом метода Гаусса.

Обратный ход будет заключаться в последовательном определении x_m, x_{m-1}, …, x_1.

Заметим, что применение метода Гаусса с выбором главного элемента позволяет вычислить определитель матрицы *A* по формуле:

$$\det(A) = (-1)^k \cdot a_{11}^{(11)} \cdot a_{22}^{(21)} \cdot a_{33}^{(31)} \ldots \cdot a_{mm}^{(m-1,2)},$$

где k – число перестановок строк и столбцов на всех шагах приведения матрицы к треугольному виду. Заметим также, что описанный выше алгоритм можно использовать в различных задачах линейной алгебры: при вычислении рангов матриц, при нахождении обратной матрицы и так далее.

Если определитель матрицы равен нулю, то это обстоятельство выяснится при вычислениях, так как на некотором шаге окажется равным нулю главный элемент или элемент $a_{mm}^{(m-1,2)}$.

3.2. Метод прогонки решения систем алгебраических уравнений с трехдиагональной матрицей. Достаточные условия применимости метода прогонки. Итерационные методы. Метод простых итераций

Метод прогонки решения систем алгебраических уравнений с трехдиагональной матрицей

Метод прогонки является модификацией метода Гаусса для частного случая разряженных систем трехдиагональной матрицы, которые возникают при моделировании некоторых инженерных и краевых задач.

Рассмотрим следующую задачу:

$$a_i y_{i-1} - c_i y_i + b_i y_{i+1} = -F_i, \qquad (3.2.1)$$

где $i = \overline{1, N-1}$, $y_0 = \xi_1 y_1 + \eta_1$, $y_N = \xi_2 y_{N-1} + \eta_2$, $a_i \neq 0$, $b_i \neq 0$ для всех $i = \overline{1, N-1}$. Матрица этой системы

$$\begin{pmatrix} 1 & -\xi_1 & 0 & 0 & \ldots & 0 & 0 & 0 \\ a_1 & -c_1 & b_1 & 0 & \ldots & 0 & 0 & 0 \\ 0 & a_2 & -c_2 & b_2 & \ldots & 0 & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & 0 & \ldots & a_{N-1} & -c_{N-1} & b_{N-1} \\ 0 & 0 & 0 & 0 & \ldots & 0 & -\xi_2 & 1 \end{pmatrix}$$

содержит нули везде, кроме главной диагонали и двух соседних, и является трехдиагональный. Это система линейных алгебраических уравнений относительно величин y_0, y_1, …, y_N.

Будем находить неизвестные y_i по следующей формуле: $y_i = \alpha_{i+1} y_{i+1} + \beta_{i+1}$ $(i = \overline{0, N-1})$ с неизвестными коэффициентами прогонки α_{i+1} и β_{i+1} $(i = \overline{1, N-1})$. Подставим $y_{i-1} = \alpha_i y_i + \beta_i$ в (3.2.1).

$$a_i(\alpha_i y_i + \beta_i) - c_i y_i + b_i y_{i+1} = -F_i,$$

$$y_i(a_i\alpha_i - c_i) + a_i\beta_i + b_i y_{i+1} = -F_i,$$

$$(\alpha_{i+1} y_{i+1} + \beta_{i+1})(a_i\alpha_i - c_i) + a_i\beta_i + b_i y_{i+1} = -F_i,$$

$$y_{i+1}[(a_i\alpha_i - c_i)\alpha_{i+1} + b_i] + (a_i\alpha_i - c_i)\beta_{i+1} + a_i\beta_i = -F_i.$$

Так как это равенство выполняется для любого y_{i+1}, то $(a_i\alpha_i - c_i)\alpha_{i+1} + b_i = 0$ и $(a_i\alpha_i - c_i)\beta_{i+1} + a_i\beta_i + F_i = 0$. Итак для $i = \overline{1, N-1}$

$$\alpha_{i+1} = \frac{b_i}{c_i - a_i\alpha_i} \tag{3.2.2}$$

$$\beta_{i+1} = \frac{a_i\beta_i + F_i}{c_i - a_i\alpha_i} \tag{3.2.3}$$

Это прогоночные коэффициенты. Для определения α_1 и β_1 заметим, что $y_0 = \xi_1 y_1 + \eta_1$ и $y_0 = \alpha_1 y_1 + \beta_1$. Отсюда видно, что $\alpha_1 = \xi_1$ и $\beta_1 = \eta_1$. Зная α_1 и β_1, из (3.2.2) и (3.2.3) определим все прогоночные коэффициенты. Этот процесс называется прямым ходом прогонки: $y_i = \alpha_{i+1} y_{i+1} + \beta_{i+1}$.

Далее заметим, что по условию $y_N = \xi_2 y_{N-1} + \eta_2$. С другой стороны, $y_{N-1} = \alpha_N y_N + \beta_N$. Итак,

$$y_N = \xi_2(\alpha_N y_N + \beta_N) + \eta_2,\ y_N = \frac{\xi_2\beta_N + \eta_2}{1 - \xi_2\alpha_N}.$$

Теперь, зная y_N, можно найти все y_i $(i = \overline{0, N-1})$. Этот процесс называется обратным ходом прогонки.

Метод прогонки является точным методом, а следовательно, результат, отвлекаясь от погрешностей вычислений, можно считать точным.

Достаточные условия применимости метода прогонки

Теорема 3.2.1. Достаточными условиями применимости метода прогонки являются: $|c_i| \geq |a_i| + |b_i|$ для любого $i = \overline{1, N-1}$, $|\xi_1| \leq 1$, $|\xi_2| \leq 1$, $|\xi_1| + |\xi_2| < 2$. (3.2.4)

Доказательство

При выводе формул прогонки мы делили на некоторые величины, не задумываясь, не равны ли они нулю, а именно на

$c_i - \alpha_i a_i$ $(i = \overline{1, N-1})$ и $1 - \alpha_N \xi_2$. Докажем, что при выполнении условий (3.2.4) эти величины не равны нулю.

Докажем сперва, что $|\alpha_i| \le 1$, для любого $i = \overline{1, N}$. Доказательство проведем методом математической индукции. При $i = 1$ условие выполняется: $|\alpha_1| = |\xi_1| \le 1$. Сделаем индуктивное предположение, что $|\alpha_i| \le 1$, и докажем, что $|\alpha_{i+1}| \le 1$. Рассмотрим разность

$$|c_i - \alpha_i a_i| - |b_i| \ge |c_i| - |\alpha_i||a_i| - |b_i| \ge |a_i| + |b_i| - |\alpha_i||a_i| - |b_i| =$$

$$= |a_i|(1 - |\alpha_i|) \ge 0, \text{ так как } |\alpha_i| \le 1.$$

Из полученного результата и (3.2.2) следует, что $|\alpha_{i+1}| = \dfrac{|b_i|}{|c_i - \alpha_i a_i|} \le 1$. По принципу математической индукции все величины $|\alpha_i| \le 1$, а следовательно, $|c_i - \alpha_i a_i| - |b_i| \ge 0$ для всех $i = \overline{1, N-1}$. Так как $b_i \ne 0$ по условию, то $|c_i - \alpha_i a_i| > 0$, а следовательно, $c_i - \alpha_i a_i \ne 0$ для $i = \overline{1, N-1}$.

Исследуем теперь величину $|1 - \alpha_N \xi_2|$. Докажем, что она строго больше нуля. Отметим, что $|1 - \alpha_N \xi_2| \ge 1 - |\alpha_N||\xi_2|$. Из условий (3.2.4) следует, что ξ_1 и ξ_2 не могут равняться 1 одновременно, то есть либо $|\xi_1| < 1$, либо $|\xi_2| < 1$.

Пусть $|\xi_1| < 1$, тогда $|\alpha_1| < 1$. Из $|\alpha_{i+1}| = \dfrac{|b_i|}{|c_i - \alpha_i a_i|}$ следует, что все $|\alpha_i| < 1$, следовательно, и $|\alpha_N| < 1$, а также $|\alpha_N||\xi_2| < 1$. Значит, $|1 - \alpha_N \xi_2| > 0$. Пусть $|\xi_2| < 1$. Так как $|\alpha_N| \le 1$, то $|\alpha_N||\xi_2| < 1$. Значит, $|1 - \alpha_N \xi_2| > 0$.

Итак, доказано, что $1 - \xi_2 a_N \ne 0$. Теорема доказана полностью.

Итерационные методы. Метод простых итераций

Вновь рассмотрим систему линейных алгебраических уравнений $A \cdot X = B$, где A – матрица коэффициентов системы $m \times m$, $X = (x_1, x_2, \ldots, x_m)^T$ – искомый вектор, $B = (b_1, b_2, \ldots, b_m)^T$ – заданный вектор правых частей.

Соотношение $A \cdot X = B$ задает отображение $A : R^m \to R^m$, где R^m – m-мерное арифметическое пространство. Зададим на нем метрику, например, так:

$$\forall x, y \in R^m \left(\rho(x, y) = \max_i |x_i - y_i| \right).$$

Выразим неизвестные $x_1, x_2, \ldots, x_m$ из уравнений системы $A \cdot X = B$.

$$\begin{cases} x_1 = -\dfrac{a_{12}}{a_{11}} \cdot x_2 - \dfrac{a_{13}}{a_{11}} \cdot x_3 - \ldots - \dfrac{a_{1m}}{a_{11}} \cdot x_m + \dfrac{b_1}{a_{11}} \\ x_2 = -\dfrac{a_{21}}{a_{22}} \cdot x_1 - \dfrac{a_{23}}{a_{22}} \cdot x_3 - \ldots - \dfrac{a_{2m}}{a_{22}} \cdot x_m + \dfrac{b_2}{a_{22}} \\ \ldots \\ x_m = -\dfrac{a_{m1}}{a_{mm}} \cdot x_1 - \dfrac{a_{m2}}{a_{mm}} \cdot x_2 - \ldots - \dfrac{a_{mm-1}}{a_{mm}} \cdot x_{m-1} + \dfrac{b_m}{a_{mm}} \end{cases}$$

Полученную систему можно записать в компактной форме: $X = B \cdot X + C$, где $B = \|b_{ij}\|$, $i, j = \overline{1,m}$, $C = (c_1, c_2, \ldots, c_m)^T$ и $b_{ij} = -\dfrac{a_{ij}}{a_{ii}}$, $i, j = \overline{1,m}$, $j \neq i$, $b_{ii} = 0$, $c_i = \dfrac{b_i}{a_{ii}}$.

Если отображение $B : R^m \to R^m$ является сжатым отображением в себя, то для решения системы $X = B \cdot X + C$ можно применить принцип сжатых отображений Банаха.

Найдем условия сжатости отображения B.

$$\rho(Bx, By) = \max_i \left| \sum_{j=1}^m b_{ij} x_j - \sum_{j=1}^m b_{ij} y_j \right| = \max_i \left| \sum_{j=1}^m b_{ij} (x_j - y_j) \right| \le \max_i \sum_{j=1}^m |b_{ij}| \cdot |x_j - y_j| \le$$

$$\le \max_i \left(\sum_{j=1}^m |b_{ij}| \cdot \max_j |x_j - y_j| \right) = \max_i \left(\sum_{j=1}^m |b_{ij}| \cdot \rho(x, y) \right) = \left(\max_i \sum_{j=1}^m |b_{ij}| \right) \cdot \rho(x, y).$$

Следовательно, оператор B будет оператором сжатия, если $\max\limits_i \sum\limits_{j=1}^m |b_{ij}| < 1$ или $\max\limits_i \sum\limits_{\substack{j=1 \\ j \neq i}}^m \dfrac{|a_{ij}|}{|a_{ii}|} < 1$, или для всех i $\sum\limits_{\substack{j=1 \\ j \neq i}}^m \dfrac{|a_{ij}|}{|a_{ii}|} < 1$. Следовательно, для всех i $|a_{ii}| > \sum\limits_{j=1}^m |a_{ij}|$, $i \neq j$. Здесь мы пользовались тем, что $\left(\max\limits_{1 \le i \le m} s_i < 1\right) \Leftrightarrow \left(\forall_{1 \le i \le m} (s_i < 1)\right)$.

Последнее условие называют условием преобладания диагональных коэффициентов, оно означает, что любой диагональный коэффициент системы по модулю больше, чем сумма модулей остальных коэффициентов строки.

Пример

$$\begin{cases} 2x_1 + 0{,}5x_2 + 0{,}2x_3 = 1 \\ x_2 + 0{,}3x_3 = 2 \\ 0{,}5x_1 - 0{,}7x_2 + 2x_3 = 3 \end{cases}$$

Таким образом, если в исходной системе диагональные коэффициенты преобладают, то решение системы можно получить, построив итерационный процесс $X^{(n+1)} = B \cdot X^{(n)} + C$, взяв за начальное приближение любую точку пространства R^m. Чаще всего, в качестве начального приближения берут вектор правых частей системы.

Более подробно итерационный процесс можно записать так:

$$\begin{cases} x_1^{(n+1)} = -\dfrac{a_{12}}{a_{11}} \cdot x_2^{(n)} - \dfrac{a_{13}}{a_{11}} \cdot x_3^{(n)} - \ldots - \dfrac{a_{1m}}{a_{11}} \cdot x_m^{(n)} + \dfrac{b_1}{a_{11}} \\ x_2^{(n+1)} = -\dfrac{a_{21}}{a_{22}} \cdot x_1^{(n)} - \dfrac{a_{23}}{a_{22}} \cdot x_3^{(n)} - \ldots - \dfrac{a_{2m}}{a_{22}} \cdot x_m^{(n)} + \dfrac{b_2}{a_{22}} \\ \ldots \\ x_m^{(n+1)} = -\dfrac{a_{m1}}{a_{mm}} \cdot x_1^{(n)} - \dfrac{a_{m2}}{a_{mm}} \cdot x_2^{(n)} - \ldots - \dfrac{a_{mm-1}}{a_{mm}} \cdot x_{m-1}^{(n)} + \dfrac{b_m}{a_{mm}} \end{cases}$$

Процесс построения приближений продолжается до тех пор, пока $\rho\left(x^{(n)}, x^{(n+1)}\right) = \max\limits_i \left| x_i^{(n)} - x_i^{(n+1)} \right| < \varepsilon \cdot \dfrac{1-k}{k}$, где ε – заданная точность вычислений, $k = \max\limits_i \sum\limits_{j=1}^{m} |b_{ij}|$. (Однако часто это условие нестрого упрощают, заменяя $\varepsilon \cdot \dfrac{1-k}{k}$ просто на ε.) За решение системы принимают вектор $x^{(n+1)}$.

Если в исходной системе не выполняется условие диагонального преобладания, то следует путем линейных преобразований привести систему к требуемому виду.

3.3. Метод Зейделя. Метод Ньютона решения систем нелинейных уравнений

Метод Зейделя

Пусть вновь дана система линейных алгебраических уравнений вида $A \cdot X = B$, где $A = \left\| a_{ij} \right\|$, $i, j = \overline{1, m}$ – матрица коэффициентов системы, $X = (x_1, x_2, \ldots, x_m)^T$ – искомый вектор, $B = (b_1, b_2, \ldots, b_m)^T$ – заданный вектор свободных членов.

Метод Зейделя является своеобразной модификацией метода простой итерации и при прочих равных условиях дает более быструю сходимость.

Преобразуем систему $A \cdot X = B$ к виду $X = B \cdot X + C$, где $B = \left\| b_{ij} \right\|$, $i, j = \overline{1, m}$, $C = (c_1, c_2, \ldots, c_m)^T$ и $b_{ij} = -\dfrac{a_{ij}}{a_{ii}}$, $i, j = \overline{1, m}$, $j \neq i$, $b_{ii} = 0$, $c_i = \dfrac{b_i}{a_{ii}}$.

Построим итерационный процесс следующим образом:

$$\begin{cases} x_1^{(n+1)} = -\dfrac{a_{12}}{a_{11}} \cdot x_2^{(n)} - \dfrac{a_{13}}{a_{11}} \cdot x_3^{(n)} - \ldots - \dfrac{a_{1m}}{a_{11}} \cdot x_m^{(n)} + \dfrac{b_1}{a_{11}} \\ x_2^{(n+1)} = -\dfrac{a_{21}}{a_{22}} \cdot x_1^{(n+1)} - \dfrac{a_{23}}{a_{22}} \cdot x_3^{(n)} - \ldots - \dfrac{a_{2m}}{a_{22}} \cdot x_m^{(n)} + \dfrac{b_2}{a_{22}} \\ x_3^{(n+1)} = -\dfrac{a_{31}}{a_{33}} \cdot x_1^{(n+1)} - \dfrac{a_{32}}{a_{33}} \cdot x_2^{(n+1)} - \dfrac{a_{34}}{a_{33}} \cdot x_4^{(n)} - \ldots - \dfrac{a_{3m}}{a_{33}} \cdot x_m^{(n)} + \dfrac{b_3}{a_{33}} \\ \ldots \\ x_m^{(n+1)} = -\dfrac{a_{m1}}{a_{mm}} \cdot x_1^{(n+1)} - \dfrac{a_{m2}}{a_{mm}} \cdot x_2^{(n+1)} - \ldots - \dfrac{a_{mm-1}}{a_{mm}} \cdot x_{m-1}^{(n+1)} + \dfrac{b_m}{a_{mm}} \end{cases}$$

В качестве начального приближения берут вектор правых частей системы. Итерации следует производить до достижения заданной точности, то есть до выполнения условия $\max\limits_i \left| x_i^{(n+1)} - x_i^{(n)} \right| < \varepsilon \cdot \dfrac{1-k}{k}$. Тогда за решение системы можно принять вектор $x^{(n+1)}$.

Отметим, что для сходимости итерационного процесса, по Зейделю, достаточным является преобладание диагональных коэффициентов. Однако это условие не является необходимым, то есть иногда даже при несоблюдении этого условия процесс может сойтись к решению. Однако для достижения гарантированной сходимости рекомендуется систему привести к диагональному преобладанию путем линейных преобразований.

Метод Ньютона решения систем нелинейных уравнений

Пусть дана система нелинейных уравнений:

$$\begin{cases} F_1(x_1, x_2, \ldots, x_m) = 0 \\ F_2(x_1, x_2, \ldots, x_m) = 0 \\ \ldots \\ F_m(x_1, x_2, \ldots, x_m) = 0 \end{cases} \qquad (3.3.1)$$

Идея метода Ньютона заключается в том, что в окрестности имеющегося приближения к решению системы (3.3.1) задача заменяется некоторой вспомогательной линейной задачей.

В основе метода Ньютона для системы уравнений (3.3.1) лежит использование разложения функций $F_i(x_1, x_2, \ldots, x_m)$ в ряд Тейлора, причем члены, содержащие производные второго и более высоких порядков, отбрасывают.

Пусть приближенные значения неизвестных системы (3.3.1) (например, полученные в результате предыдущей итерации) равны соответственно $a_1, a_2, \ldots, a_m$. Задача состоит в нахождении приращений (поправок) к этим значениям $\Delta x_1, \Delta x_2, \ldots, \Delta x_m$, благодаря которым решение системы (3.3.1) запишется в виде:

$$x_1 = a_1 + \Delta x_1, x_2 = a_2 + \Delta x_2, \ldots, x_m = a_m + \Delta x_m \qquad (3.3.2)$$

Проведем разложение левых частей уравнений (3.3.) с учетом (3.3.2) в ряд Тейлора, ограничиваясь лишь линейными членами относительно приращений:

$$\begin{cases} F_1(x_1, x_2, \ldots, x_m) \approx F_1(a_1, a_2, \ldots, a_m) + \sum_{i=1}^{m} \frac{\partial F_1(a_1, a_2, \ldots, a_m)}{\partial x_i} \cdot \Delta x_i \\ F_2(x_1, x_2, \ldots, x_m) \approx F_2(a_1, a_2, \ldots, a_m) + \sum_{i=1}^{m} \frac{\partial F_2(a_1, a_2, \ldots, a_m)}{\partial x_i} \cdot \Delta x_i \\ \ldots \\ F_m(x_1, x_2, \ldots, x_m) \approx F_m(a_1, a_2, \ldots, a_m) + \sum_{i=1}^{m} \frac{\partial F_m(a_1, a_2, \ldots, a_m)}{\partial x_i} \cdot \Delta x_i \end{cases}$$

Поскольку в соответствии с (3.3.1) левые части этих выражений должны обращаться в нуль, то приравниваем к нулю и правые части. Получим следующую систему линейных алгебраических уравнений относительно приращений $\Delta x_1, \Delta x_2, \ldots, \Delta x_m$.

$$\begin{cases} \sum_{i=1}^{m} \frac{\partial F_1(a_1, a_2, \ldots, a_m)}{\partial x_i} \cdot \Delta x_i = -F_1(a_1, a_2, \ldots, a_m) \\ \sum_{i=1}^{m} \frac{\partial F_2(a_1, a_2, \ldots, a_m)}{\partial x_i} \cdot \Delta x_i = -F_2(a_1, a_2, \ldots, a_m) \\ \ldots \\ \sum_{i=1}^{m} \frac{\partial F_m(a_1, a_2, \ldots, a_m)}{\partial x_i} \cdot \Delta x_i = -F_m(a_1, a_2, \ldots, a_m) \end{cases} \qquad (3.3.3)$$

Определителем системы (3.3.3) является якобиан

$$\begin{vmatrix} \frac{\partial F_1}{\partial x_1} & \frac{\partial F_1}{\partial x_2} & \dots & \frac{\partial F_1}{\partial x_m} \\ \frac{\partial F_2}{\partial x_1} & \frac{\partial F_2}{\partial x_2} & \dots & \frac{\partial F_2}{\partial x_m} \\ \dots & \dots & \dots & \dots \\ \frac{\partial F_m}{\partial x_1} & \frac{\partial F_m}{\partial x_2} & \dots & \frac{\partial F_m}{\partial x_m} \end{vmatrix}$$

Частные производные вычислены в точке $(a_1, a_2, \dots, a_m)$. Для существования и единственности решения системы (3.3.3) он должен быть отличен от нуля (на каждой итерации).

Таким образом, итерационный процесс решения системы уравнений (3.3.1) методом Ньютона состоит в определении приращений $\Delta x_1, \Delta x_2, \dots, \Delta x_m$ к значениям неизвестных на каждой итерации. Процесс прекращается, если все приращения становятся малыми по абсолютной величине: $\max_i |\Delta x_i| < \varepsilon$, где ε – заданная точность. При определенных условиях, наложенных на функции $F_i(x_1, x_2, \dots, x_m)$, и при удачном выборе начального приближения каждый шаг итерационного процесса выполним и последовательность итераций сходится к решению системы (3.3.1), причем этот метод имеет второй порядок сходимости.

3.4. Лабораторная работа «Методы решения систем линейных уравнений»

(4 часа)

Цель работы: научиться решать системы линейных уравнений в программе MathCad и реализовывать методы Гаусса и Зейделя в собственных программах.

Используемое программное обеспечение: Borland Pascal (или Delphi) и MathCad.

Основное задание:

1. Решите систему, с номером, соответствующим вашему варианту, в программе MathCad (задания по вариантам смотрите в таблице 3.4.1).

2. Решите систему методом Гаусса с выбором главного элемента.

Таблица 3.4.1

Задания к лабораторной работе

(для тридцати вариантов)

№	Задание	№	Задание
1	*2*	*3*	*4*
1	$\begin{cases} 3,7x_1 + 3,3x_2 + 1,3x_3 = 2,1; \\ 3,5x_1 - 1,7x_2 + 2,8x_3 = 1,7; \\ 4,1x_1 + 5,8x_2 - 1,7x_3 = 0,8. \end{cases}$	2	$\begin{cases} 1,7x_1 + 2,8x_2 + 1,9x_3 = 0,7; \\ 2,1x_1 + 3,4x_2 + 1,8x_3 = 1,1; \\ 4,2x_1 - 1,7x_2 + 1,3x_3 = 2,8. \end{cases}$
3	$\begin{cases} 3,1x_1 + 2,8x_2 + 1,9x_3 = 0,2; \\ 1,9x_1 + 3,1x_2 + 2,1x_3 = 2,1; \\ 7,5x_1 + 3,8x_2 + 4,8x_3 = 5,6. \end{cases}$	4	$\begin{cases} 9,1x_1 + 5,6x_2 + 7,8x_3 = 9,8; \\ 3,8x_1 + 5,1x_2 + 2,8x_3 = 6,7; \\ 4,1x_1 + 5,7x_2 + 1,2x_3 = 5,8. \end{cases}$
5	$\begin{cases} 3,3x_1 + 2,1x_2 + 2,8x_3 = 0,8; \\ 4,1x_1 + 3,7x_2 + 4,8x_3 = 5,7; \\ 2,7x_1 + 1,8x_2 + 1,1x_3 = 3,2. \end{cases}$	6	$\begin{cases} 7,6x_1 + 5,8x_2 + 4,7x_3 = 10,1; \\ 3,8x_1 + 4,1x_2 + 2,7x_3 = 9,7; \\ 2,9x_1 + 2,2x_2 + 3,8x_3 = 7,8. \end{cases}$
7	$\begin{cases} 3,2x_1 - 2,5x_2 + 3,7x_3 = 6,5; \\ 0,5x_1 + 0,34x_2 + 1,7x_3 = -0,24; \\ 1,.6x_1 + 2,3x_2 - 1,5x_3 = 4,3. \end{cases}$	8	$\begin{cases} 5,4x_1 - 2,3x_2 + 3,4x_3 = -3,5; \\ 4,2x_1 + 1,7x_2 - 2,3x_3 = 2,7; \\ 3,4x_1 + 2,4x_2 + 7,4x_3 = 1,9. \end{cases}$
9	$\begin{cases} 3,6x_1 + 1,8x_2 - 4,7x_3 = 3,8; \\ 2,7x_1 - 3,6x_2 + 1,9x_3 = 0,4; \\ 1,5x_1 + 4,5x_2 + 3,3x_3 = -1,6. \end{cases}$	10	$\begin{cases} 5,6x_1 + 2,7x_2 - 1,7x_3 = 1,9; \\ 3,4x_1 - 3,6x_2 - 6,7x_3 = -2,4; \\ 0,8x_1 + 1,3x_2 + 3,7x_3 = 1,2. \end{cases}$
11	$\begin{cases} 2,7x_1 + 0,9x_2 - 1,5x_3 = 3,5; \\ 4,5x_1 - 2,8x_2 + 6,7x_3 = 2,6; \\ 5,1x_1 + 3,7x_2 - 1,4x_3 = -0,14. \end{cases}$	12	$\begin{cases} 4,5x_1 - 3,5x_2 + 7,4x_3 = 2,5; \\ 3,1x_1 - 0,6x_2 - 2,3x_3 = -1,5; \\ 0,8x_1 + 7,4x_2 - 0,5x_3 = 6,4. \end{cases}$
13	$\begin{cases} 3,8x_1 + 6,7x_2 - 1,2x_3 = 5,2; \\ 6,4x_1 + 1,3x_2 - 2,7x_3 = 3,8; \\ 2,4x_1 - 4,5x_2 + 3,5x_3 = -0,6. \end{cases}$	14	$\begin{cases} 5,4x_1 - 6,2x_2 - 0,5x_3 = 0.52; \\ 3,4x_1 + 2,3x_2 + 0,8x_3 = -0,8; \\ 2,4x_1 - 1,1x_2 + 3,8x_3 = 1,8. \end{cases}$
15	$\begin{cases} 7,8x_1 + 5,3x_2 + 4,8x_3 = 1,8; \\ 3,3x_1 + 1,1x_2 + 1,8x_3 = 2,3; \\ 4,5x_1 + 3,3x_2 + 2,8x_3 = 3,4. \end{cases}$	16	$\begin{cases} 3,8x_1 + 4,1x_2 - 2,3x_3 = 4,8; \\ -2,1x_1 + 3,9x_2 - 5,8x_3 = 3,3; \\ 1,8x_1 + 1,1x_2 - 2,1x_3 = 5,8. \end{cases}$
17	$\begin{cases} 1,7x_1 - 2,2x_2 + 3,0x_3 = 1,8; \\ 2,1x_1 + 1,9x_2 - 2,3x_3 = 2,8; \\ 4,2x_1 + 3,9x_2 - 3,1x_3 = 5,1. \end{cases}$	18	$\begin{cases} 2,8x_1 + 3,8x_2 - 3,2x_3 = 4,5; \\ 2,5x_1 - 2,8x_2 + 3,3x_3 = 7,1; \\ 6,5x_1 - 7,1x_2 + 4,8x_3 = 6,3. \end{cases}$
19	$\begin{cases} 3,3x_1 + 3,7x_2 + 4,2x_3 = 5,8; \\ 2,7x_1 + 2,3x_2 - 2,9x_3 = 6,1; \\ 4,1x_1 + 4,8x_2 - 5,0x_3 = 7,0. \end{cases}$	20	$\begin{cases} 7,1x_1 + 6,8x_2 + 6,1x_3 = 7,0; \\ 5,0x_1 + 4,8x_2 + 5,3x_3 = 6,1; \\ 8,2x_1 + 7,8x_2 + 7,1x_3 = 5,8. \end{cases}$

Окончание таблицы 3.4.1

1	*2*	*3*	*4*
21	$\begin{cases} 3{,}7x_1 + 3{,}1x_2 + 4{,}0x_3 = 5{,}0; \\ 4{,}1x_1 + 4{,}5x_2 - 4{,}8x_3 = 4{,}9; \\ -2{,}1x_1 - 3{,}7x_2 + 1{,}8x_3 = 2{,}7. \end{cases}$	22	$\begin{cases} 4{,}1x_1 + 5{,}2x_2 - 5{,}8x_3 = 7{,}0; \\ 3{,}8x_1 - 3{,}1x_2 + 4{,}0x_3 = 5{,}3; \\ 7{,}8x_1 + 5{,}3x_2 - 6{,}3x_3 = 5{,}8. \end{cases}$
23	$\begin{cases} 3{,}7x_1 - 2{,}3x_2 + 4{,}5x_3 = 2{,}4; \\ 2{,}5x_1 + 4{,}7x_2 - 7{,}8x_3 = 3{,}5; \\ 1{,}6x_1 + 5{,}3x_2 + 1{,}3x_3 = -2{,}4. \end{cases}$	24	$\begin{cases} 6{,}3x_1 + 5{,}2x_2 - 0{,}6x_3 = 1{,}5; \\ 3{,}4x_1 - 2{,}3x_2 + 3{,}4x_3 = 2{,}7; \\ 0{,}8x_1 + 1{,}4x_2 + 3{,}5x_3 = -2{,}3. \end{cases}$
25	$\begin{cases} 1{,}5x_1 + 2{,}3x_2 - 3{,}7x_3 = 4{,}5; \\ 2{,}8x_1 + 3{,}4x_2 + 5{,}8x_3 = -3{,}2; \\ 1{,}2x_1 + 7{,}3x_2 - 2{,}3x_3 = 5{,}6. \end{cases}$	26	$\begin{cases} 0{,}9x_1 + 2{,}7x_2 - 3{,}8x_3 = 2{,}4; \\ 2{,}5x_1 + 5{,}8x_2 - 0{,}5x_3 = 3{,}5; \\ 4{,}5x_1 - 2{,}1x_2 + 3{,}2x_3 = -1{,}2. \end{cases}$
27	$\begin{cases} 2{,}4x_1 + 2{,}5x_2 - 2{,}9x_3 = 4{,}5; \\ 0{,}8x_1 + 3{,}5x_2 - 1{,}4x_3 = 3{,}2; \\ 1{,}5x_1 - 2{,}3x_2 + 8{,}6x_3 = -5{,}5. \end{cases}$	28	$\begin{cases} 5{,}4x_1 - 2{,}4x_2 + 3{,}8x_3 = 5{,}5; \\ 2{,}5x_1 + 6{,}8x_2 - 1{,}1x_3 = 4{,}3; \\ 2{,}7x_1 - 0{,}6x_2 + 1{,}5x_3 = -3{,}5. \end{cases}$
29	$\begin{cases} 2{,}4x_1 + 3{,}7x_2 - 8{,}3x_3 = 2{,}3; \\ 1{,}8x_1 + 4{,}3x_2 + 1{,}2x_3 = -1{,}2; \\ 3{,}4x_1 - 2{,}3x_2 + 5{,}2x_3 = 3{,}5. \end{cases}$	30	$\begin{cases} 3{,}2x_1 - 11{,}5x_2 + 3{,}8x_3 = 2{,}8; \\ 0{,}8x_1 + 1{,}3x_2 - 6{,}4x_3 = -6{,}5; \\ 2{,}4x_1 + 7{,}2x_2 - 1{,}2x_3 = 4{,}5. \end{cases}$

Комментарии к основному заданию

1. Для выполнения работы в программе MathCad понадобится панель инструментов «*Матрицы*» и ее кнопка «*Создать матрицу или вектор*». Команда `lsolve(a,b)` возвращает вектор, являющийся решением системы линейных уравнений с матрицей системы *a* и вектором свободных членов *b*.

2. Укажем блок-схему для программной реализации метода Гаусса с выбором главного элемента (рис. 3.4.1). Переменная *a* – двумерный массив размерности $n \times (n+1)$, в котором изначально хранятся все коэффициенты системы, включая столбец свободных членов. Заметим, что вся схема состоит из двух относительно независимых блоков: первый реализует прямой ход метода Гаусса (цикл по *k*), второй – обратный ход (цикл по *i*). Первый блок состоит из двух «подблоков». В первом реализуется выбор главного элемента, во втором – обнуляются все элементы *k*-го столбца, расположенные ниже диагонального элемента. Процедура swap меняет местами значения соответствующих переменных. Эту процедуру следует написать самостоятельно. После написания программы и сверки ответов с программой MathCad желательно протестировать свою программу для

какой-нибудь системы, у которой на k-ом шаге на диагональном месте оказывается число 0.

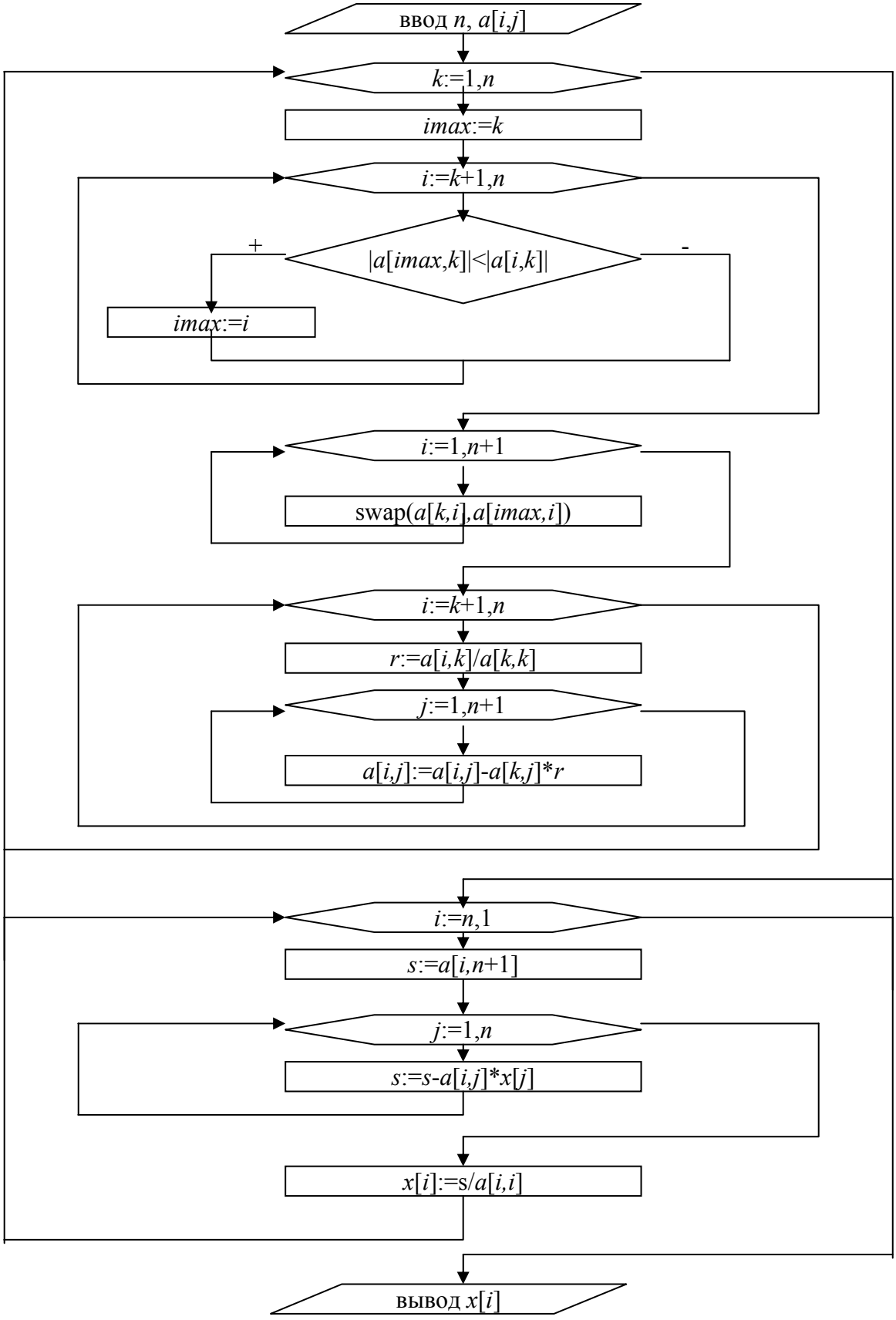


Рис. 3.4.1. Блок-схема: метод Гаусса

Дополнительное задание

1. Дополните программу, реализующую метод Гаусса с выбором главного элемента следующими возможностями: а) в том случае, если определитель системы будет равен нулю, пусть программа выдаст соответствующее сообщение; б) наряду с решением системы пусть программа выдает величину определителя системы.
2. Напишите программу для решения системы методом Зейделя.

Комментарии к дополнительному заданию

1. Если на очередном шаге главный элемент оказывается равным нулю, то определитель системы равен нулю. В этом случае последняя либо не имеет решений, либо имеет бесконечно много решений.
2. Перед тем, как приступить к реализации метода Зейделя, следует привести систему к преобладанию диагональных коэффициентов, например, с помощью программы Excel, используя пункт меню «*Правка / Специальная вставка…*». Укажем блок-схему для программной реализации метода Зейделя (рис. 3.4.2).

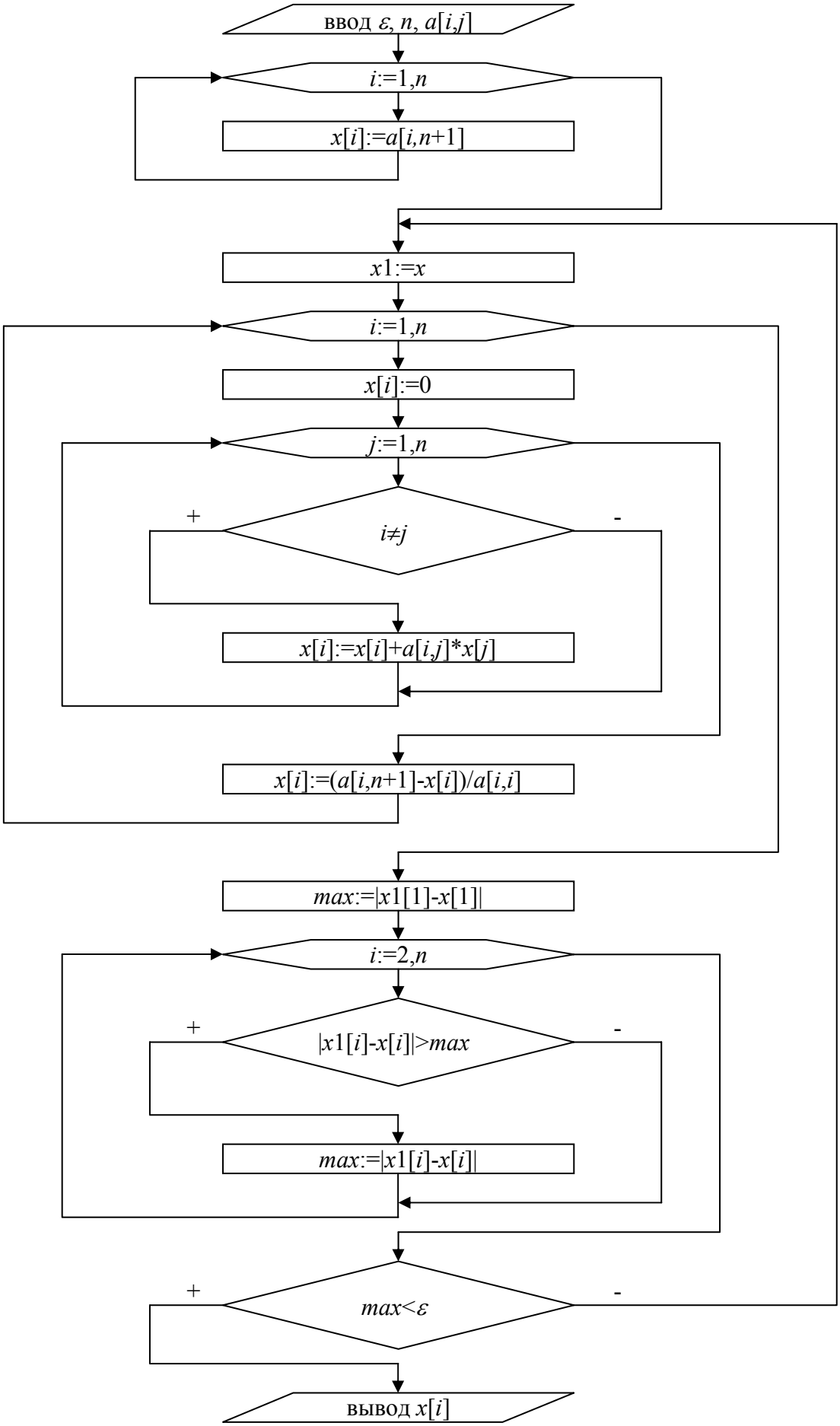


Рис. 3.4.2. Блок-схема: метод Зейделя

ГЛАВА 4. АППРОКСИМАЦИЯ ФУНКЦИЙ

4.1. Понятие об аппроксимации функций. Вычисление значений многочленов. Интерполирование функции многочленом

Понятие об аппроксимации функций

Определение 4.1.1. Аппроксимация – это приближенное выражение каких-либо объектов через другие более простые объекты.

Пусть величина y является функцией аргумента x. Это означает, что любому значению x из области определения поставлено в соответствие единственное y. Вместе с тем на практике часто неизвестна явная связь в виде некоторой зависимости $y = f(x)$. В некоторых случаях даже при известной зависимости $y = f(x)$ она настолько громоздка (например, содержит сложные интегралы), что ее использование на практике затруднительно.

Наиболее распространенным и практически важным случаем, когда вид связи между параметрами x и y неизвестен, является задание этой связи в виде некоторой таблицы $\{x_i, y_i\}$. Это означает, что дискретному множеству значений аргумента $\{x_i\}$ поставлено в соответствие множество соответствующих значений функции $\{y_i\}$ ($i = \overline{0,n}$). Эти значения могут быть либо результатами расчетов, либо экспериментальными данными. Такую функцию называют сеточной функцией. На практике могут понадобиться значения величины y и в других точках, отличных от узлов x_i. Однако получить эти значения можно лишь путем очень сложных расчетов или проведением дорогостоящих экспериментов.

Таким образом, с точки зрения экономии времени и средств мы приходим к необходимости использования имеющихся табличных данных для приближённого вычисления искомого параметра y при любом значении определяющего параметра из некоторой области определения.

Этой цели и служит задача о приближении (аппроксимации) функций: данную функцию $f(x)$ требуется аппроксимировать (приближенно заменить) некоторой функцией $\varphi(x)$ так, чтобы отклонение (в некотором смысле) $\varphi(x)$ от $f(x)$ в заданной области было наименьшим. Функцию $\varphi(x)$ называют при этом аппроксимирующей.

Если приближение строится на заданном дискретном множестве точек $\{x_i\}$, то аппроксимация называется точечной. При построении приближения на непрерывном множестве точек аппроксимация называется непрерывной.

Вычисление значений многочленов

Очень часто в качестве аппроксимирующей функции $\varphi(x)$ берут многочлен. Это связано с тем, что множество многочленов всюду плотно во множестве непрерывных функций. То есть имеет место следующая теорема.

Теорема 4.1.1 (теорема Вейерштрасса). Если $f(x) \in C_{[a,b]}$, то для любого $\varepsilon > 0$ существует многочлен $P(x)$ такой, что $|f(x) - P(x)| < \varepsilon$ при всех $x \in [a,b]$.

Кроме того, и это немаловажно, значения многочлена легко вычисляются. Рассмотрим алгебраический многочлен

$$P_n(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n,$$

где a_0, a_1, …, a_n – числовые коэффициенты, n – степень многочлена. Если проводить вычисления в «лоб», то есть находить значения каждого члена и суммировать их, то при больших n потребуется выполнить большое число операций: $\frac{n^2+n}{2}$ умножений и n сложений. Кроме того, это может привести к потере точности за счет погрешности округлений.

Запишем многочлен в следующем виде:

$$P_n(x) = a_0 + x(a_1 + x(a_2 + \ldots + x(a_{n-1} + xa_n))).$$

Согласно этой формуле, вычисление значения $P_n(x)$ сводится к последовательному нахождению следующих величин:

$b_n = a_n$;

$b_{n-1} = a_{n-1} + xb_n$;

$b_{n-2} = a_{n-2} + xb_{n-1}$;

…

$b_1 = a_1 + xb_2$;

$b_0 = a_0 + xb_1 = P_n(x)$.

Способ нахождения значения многочлена по вышеописанным формулам называется схемой Горнера. Для реализации этой схемы требуется n умножений и n сложений, то есть всего $2n$

арифметических действий. Схема Горнера является в общем случае самым оптимальным способом вычисления значения многочлена. Использование этой схемы не только экономит машинное время, но и повышает точность вычислений за счет уменьшения погрешности округления. Схема Горнера удобна также для реализации на ЭВМ благодаря цикличности вычислений и необходимости сохранять кроме коэффициентов многочлена и значения аргумента только одно значение промежуточной величины, а именно b_i при текущем $i = \overline{n,0}$.

Интерполирование функции многочленом

Теорема Вейерштрасса не дает способа построения аппроксимирующего многочлена, она устанавливает лишь принципиальную возможность этого построения. Для построения приближающих многочленов разработано много способов. Один из них – интерполирование, который заключается в следующем.

Пусть имеется таблица значений некоторой функции $y = f(x)$, причем если $i \neq j$, то $x_i \neq x_j$ (табл. 4.1.1).

Таблица 4.1.1

x	x_0	x_1	x_2	...	x_n
y	y_0	y_1	y_2	...	y_n

Задача состоит в том, чтобы найти такой многочлен степени не выше n, который в заданных точках x_i принимает те же значения y_i, что и функция $f(x)$. Таким образом, близость интерполяционного многочлена для заданной функции состоит в том, что их значения совпадают на заданной системе точек (сетке).

Различают интерполяцию глобальную и локальную (или кусочную). Если один многочлен $\varphi(x) = a_0 + a_1 x + \ldots + a_n x^n$ используется для интерполяции функции $f(x)$ на всем рассматриваемом интервале изменения аргумента x, то говорят о глобальной интерполяции. В этом случае максимальная степень интерполяционного многочлена равна n, то есть на единицу меньше количества узлов интерполирования.

С геометрической точки зрения, задача глобальной интерполяции заключается в построении такого многочлена степени не выше n, график которого проходит через данные точки (x_0, y_0), (x_1, y_1), ..., (x_n, y_n) кривой $y = f(x)$.

Если интерполяционный многочлен строится отдельно для разных частей рассматриваемого интервала изменения x, то имеет место локальная интерполяция. Например, можно по трем лежащим рядом точкам построить кусочки парабол.

Теорема 4.1.2. Глобальный интерполяционный многочлен существует и единственен.

Доказательство

Пусть $\varphi(x) = a_0 + a_1 x + \ldots + a_n x^n$ – глобальная интерполяция функции $f(x)$ по системе узлов $\{x_0, x_1, \ldots, x_n,\}$.

Учитывая, что $\varphi(x_i) = y_i$ $(i = \overline{0,n})$, можем записать:

$$\begin{cases} a_0 + a_1 x_0 + \ldots + a_n x_0^n = y_0 \\ a_0 + a_1 x_1 + \ldots + a_n x_1^n = y_1 \\ \ldots \\ a_0 + a_1 x_n + \ldots + a_n x_n^n = y_n \end{cases} \qquad (4.1.1)$$

Это система для определения коэффициентов интерполяционного многочлена a_0, a_1, …, a_n. Как известно, многочлен однозначно задается системой своих коэффициентов.

Определитель системы (4.1.1) является определителем Вандермонда.

$$\Delta = \begin{vmatrix} 1 & x_0 & \cdots & x_0^n \\ 1 & x_1 & \cdots & x_1^n \\ \cdots & \cdots & \cdots & \cdots \\ 1 & x_n & \cdots & x_n^n \end{vmatrix}$$

Этот определитель не равен 0, если среди чисел x_0, x_1, …, x_n нет равных. Так как при постановке задачи интерполирования мы потребовали, чтобы узлы были различны, то $\Delta \neq 0$.

Тогда система (4.1.1) имеет единственное решение. Таким образом, для данной системы узлов $\{x_0, x_1, \ldots, x_n\}$ существует единственный глобальный интерполяционный многочлен.

Из приведенных рассуждений следует способ построения интерполяционного многочлена: нужно составить и решить систему (4.1.1).

Пример

x	0	1	2
y	0	1	4

$$y = ax^2 + bx + c,\ \begin{cases} a\cdot 0^2 + b\cdot 0 + c = 0 \\ a\cdot 1^2 + b\cdot 1 + c = 1 \\ a\cdot 2^2 + b\cdot 2 + c = 4 \end{cases},\ \begin{cases} a = 1 \\ b = 0 \\ c = 0 \end{cases},\ y = x^2.$$

Однако для практического применения этот способ является неудобным. Существуют способы, позволяющие построить интерполяционный многочлен более экономичными методами.

4.2. Интерполяционный многочлен в форме Лагранжа. Остаточный член интерполирования

Интерполяционный многочлен в форме Лагранжа

Будем искать глобальный интерполяционный многочлен в форме

$$L_n(x) = y_0\Phi_0(x) + y_1\Phi_1(x) + \ldots + y_n\Phi_n(x),$$

где y_0, y_1, …, y_n – значения интерполируемой функции $f(x)$ в узлах x_0, x_1, …, x_n, а $\Phi_0(x)$, …, $\Phi_n(x)$ – некоторые многочлены степени не выше n.

Задача интерполирования в этом случае сводится к построению многочленов $\Phi_i(x)$ с таким расчетом, чтобы $L_n(x_j) = y_j$ при $j = \overline{0,n}$. Очевидно, они должны обладать свойством:

$$\Phi_i(x_j) = \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases}.$$

Действительно, тогда $L_n(x_j) = \sum\limits_{i=0}^{n} y_i\Phi_i(x_j) = y_j$.

Так как $\Phi_i(x_j) = 0$ $(i \neq j)$, то все узлы за исключением x_i являются корнями многочлена $\Phi_i(x)$.

По теореме из курса алгебры о разложении многочлена по корням, имеем $\Phi_i(x) = A_i(x - x_0)\ldots(x - x_{i-1})(x - x_{i+1})\ldots(x - x_n)$, где A_i – некоторый коэффициент. Найдем его из условия $\Phi_i(x_i) = 1$.

$$A_i(x_i - x_0)\ldots(x_i - x_{i-1})(x_i - x_{i+1})\ldots(x_i - x_n) = 1,$$

следовательно,

$$A_i = \frac{1}{(x_i - x_0)\ldots(x_i - x_{i-1})(x_i - x_{i+1})\ldots(x_i - x_n)}.$$

Таким образом, $\Phi_i(x) = \dfrac{(x - x_0)\ldots(x - x_{i-1})(x - x_{i+1})\ldots(x - x_n)}{(x_i - x_0)\ldots(x_i - x_{i-1})(x_i - x_{i+1})\ldots(x_i - x_n)}$.

Тогда интерполяционный многочлен будет выглядеть так:

$$L_n(x)=\sum_{i=0}^{n} y_i \frac{(x-x_0)\dots(x-x_{i-1})(x-x_{i+1})\dots(x-x_n)}{(x_i-x_0)\dots(x_i-x_{i-1})(x_i-x_{i+1})\dots(x_i-x_n)}. \qquad (4.2.1)$$

Это и есть интерполяционный многочлен в форме Лагранжа.

Введем обозначение $\omega_n(x)=(x-x_0)(x-x_1)\dots(x-x_n)$.

Найдем производную $\omega_n(x)$:

$$\omega_n'(x)=(x-x_0)(x-x_1)\dots(x-x_{i-1})(x-x_{i+1})\dots(x-x_n)+$$
$$+(x-x_i)\left[(x-x_0)(x-x_1)\dots(x-x_{i-1})(x-x_{i+1})\dots(x-x_n)\right]'.$$

Тогда $\omega_n'(x_i)=(x_i-x_0)(x_i-x_1)\dots(x_i-x_{i-1})(x_i-x_{i+1})\dots(x_i-x_n)$, и многочлен (4.2.1) можно записать более компактно:

$$L_n(x)=\sum_{i=0}^{n} y_i \frac{\omega_n(x)}{(x-x_i)\omega_n'(x_i)}.$$

Остаточный член интерполирования

Пусть $L_n(x)$ – интерполяционный многочлен, построенный для функции $f(x)$ по узлам x_0, x_1, …, x_n. В узлах значения $L_n(x)$ и $f(x)$ равны между собой. В точках x, не совпадающих с узлами, вообще говоря, $L_n(x)\neq f(x)$. Всегда можно написать равенство $f(x)=L_n(x)+R_n(x)$, где $R_n(x)$ – остаточный член, то есть погрешность интерполяции, которая характеризует точность приближения функции $f(x)$ интерполяционным многочленом $L_n(x)$.

Заметим, что если относительно функции $f(x)$ ничего не известно, кроме её значений y_i в узлах интерполяции, то никаких полезных рассуждений относительно остаточного члена $R_n(x)$ провести нельзя. В предположении, что $f(x)\in C^{(n+1)}_{[a,b]}$, где $[a,b]$ – отрезок, содержащий все узлы интерполяции x_i, $i=\overline{0,n}$, например, $a=\min\{x_0,\dots,x_n\}$, $b=\max\{x_0,\dots,x_n\}$, можно оценить погрешность интерполяции. Оценим остаточный член в произвольной точке $x\in[a,b]$, не совпадающей ни с одним из узлов. Имеет место следующая теорема об остаточном члене интерполирования.

Теорема 4.2.1. Если $f(x)\in C^{(n+1)}_{[a,b]}$, то для всякого $x\in[a,b]$ найдется точка $\xi\in[a,b]$ такая, что

$$R_n(x)=\frac{f^{(n+1)}(\xi)}{(n+1)!}(x-x_0)(x-x_1)\dots(x-x_n). \qquad (4.2.2)$$

Доказательство

Рассмотрим функцию

$$\varphi(t) = f(t) - L_n(t) - k \cdot \omega_n(t), \qquad (4.2.3)$$

где k – некоторое число, $t \in [a,b]$, $\omega_n(t) = (t-x_0)(t-x_1)\ldots(t-x_n)$.

Очевидно, что $\varphi(t)$ имеет на $[a,b]$ производные до порядка $(n+1)$ включительно. Узлы x_0, x_1, …, x_n являются корнями $\varphi(t)$. Возьмем произвольную точку $x \in [a,b]$, не совпадающую ни с одним из узлов, и подберем k так, чтобы $\varphi(x) = 0$.

Полагая в (4.2.3) $t = x$ и $\varphi(x) = 0$, получим, что k следует взять равным

$$k = \frac{f(x) - L_n(x)}{\omega_n(x)}. \qquad (4.2.4)$$

При этом значении k функция $\varphi(t)$ обращается в нуль в $(n+2)$ точках x, x_0, x_1, …, x_n. В анализе доказывается теорема Ролля: если функция $f(x)$ непрерывна на $[x_1,x_2]$ и дифференцируема на (x_1,x_2) и на концах отрезка принимает равные значения, то существует хотя бы одна точка η внутри отрезка $[x_1,x_2]$, в которой производная $f'(\eta) = 0$.

Применяя эту теорему к функции $\varphi(t)$, получаем, что её производная обращается в нуль, по крайней мере, в $(n+1)$ точках отрезка $[a,b]$. Вновь применяя теорему Ролля, но уже к функции $\varphi_t'(t)$, получаем, что производная функции φ_t' обращается в нуль, по крайней мере, в n точках отрезка $[a,b]$. Продолжая эти рассуждения дальше, имеем, что $\varphi_t^{(n+1)}(t)$ обращается в нуль, по крайней мере, в одной точке. Обозначим ее через $\xi \in [a,b]$. Отметим, что $(n+1)$-я производная $\omega_n(t)$, то есть $\omega_n^{(n+1)}(t) = (n+1)!$, так как $\omega_n(t)$ есть многочлен степени $n+1$ со старшим коэффициентом 1.

Из $\varphi(t) = f(t) - L_n(t) - k \cdot \omega_n(t)$ имеем $\varphi_t^{(n+1)}(t) = f^{(n+1)}(t) - k \cdot (n+1)!$. Полагая в этом равенстве $t = \xi$ и учитывая, что $\varphi_t^{(n+1)}(\xi) = 0$, находим $k = \dfrac{f^{(n+1)}(\xi)}{(n+1)!}$.

Подставив найденное значение k в формулу (4.2.4), найдем:

$$\frac{f^{(n+1)}(\xi)}{(n+1)!} = \frac{f(x) - L_n(x)}{\omega_n(x)}.$$

Откуда

$$f(x)-L_n(x)=\frac{f^{(n+1)}(\xi)}{(n+1)!}\cdot\omega_n(x)$$

или

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

Теорема доказана.

Если $M_{n+1}=\max\limits_{[a,b]}\left|f^{(n+1)}(x)\right|$, то

$$\left|R_n(x)\right|\le\frac{M_{n+1}}{(n+1)!}\cdot\left|(x-x_0)(x-x_1)\ldots(x-x_n)\right|. \qquad (4.2.5)$$

Это и есть оценка погрешности интерполяции.

4.3. Минимизация оценки погрешности интерполяции. Многочлены Чебышева. Локальная интерполяция. Сплайны

Минимизация оценки погрешности интерполяции. Многочлены Чебышева

Поставим перед собой проблему оптимизировать оценку (4.2.5), то есть попытаемся сделать такой выбор узлов x_i, чтобы $\max\limits_{[a,b]}|\omega_n(x)|$ был минимальным.

Рассмотрим отрезок $[-1,1]$. Для задачи минимизации понадобятся многочлены Чебышева.

Определение 4.3.1. Многочленом Чебышева $T_n(x)$ при $n\ge 0$ называется следующая функция $T_n(x)=\cos(n\cdot\arccos x)$.

Очевидно, что $T_0(x)=1$, $T_1(x)=x$. Остальные многочлены Чебышева можно получить по следующему рекуррентному соотношению:

$$T_{n+1}(x)=2xT_n(x)-T_{n-1}(x), \qquad (4.3.1)$$

где $n\ge 1$. Действительно, используя формулы косинуса суммы и разности, получаем:

$$T_{n+1}(x)+T_{n-1}(x)=\cos((n+1)\arccos x)+\cos((n-1)\arccos x)=$$
$$=\cos(n\arccos x)\cdot\cos(\arccos x)-\sin(n\arccos x)\cdot\sin(\arccos x)+$$
$$+\cos(n\arccos x)\cdot\cos(\arccos x)+\sin(n\arccos x)\cdot\sin(\arccos x)=2T_n(x)\cdot x.$$

Тогда $T_2(x)=2x^2-1$, $T_3(x)=2x(2x^2-1)-x=4x^3-3x$ и так далее.

Укажем свойства многочленов Чебышева.

1. $T_n(x)$ – многочлен степени n на $[-1,1]$; $|T_n(x)|\le 1$. При четном (нечетном) n многочлен $T_n(x)$ содержит только четные (нечетные) степени x. (Это следует из формулы (4.3.1).)

2. Старший коэффициент $T_n(x)$ равен 2^{n-1} $(n\ge 1)$.

3. На отрезке $[-1,1]$ $T_n(x)$ имеет ровно n корней, которые имеют вид:

$$x_i=\cos\frac{(2i+1)\pi}{2n},\ i=\overline{0,n-1}.$$

Действительно,

$$T_n(x)=0\Leftrightarrow\cos(n\arccos x)=0\Leftrightarrow n\arccos x=\frac{2i+1}{2}\pi\Leftrightarrow$$

$$\Leftrightarrow[\,i=\overline{0,n-1},\text{ так как }0\le n\arccos x\le \pi n\,]\Leftrightarrow$$

$$\Leftrightarrow\arccos x=\frac{2i+1}{2n}\pi\ (i=\overline{0,n-1})\Leftrightarrow x=\cos\frac{2i+1}{2n}\pi\ (i=\overline{0,n-1}).$$

4. На отрезке $[-1,1]$ $|T_n(x)|$ принимает максимальное значение 1 ровно в $(n+1)$ точке, которые имеют вид $x_m=\cos\frac{m\pi}{n}$, $m=\overline{0,n}$.

$$|T_n(x)|=1\Leftrightarrow|\cos(n\arccos x)|=1\Leftrightarrow n\arccos x=m\pi\Leftrightarrow$$

$$\Leftrightarrow[\,m=\overline{0,n},\text{ так как }0\le n\arccos x\le n\pi\,]\Leftrightarrow$$

$$\Leftrightarrow\arccos x=\frac{m\pi}{n}\ \left(m=\overline{0,n}\right)\Leftrightarrow\ x=\cos\frac{m\pi}{n}\ \left(m=\overline{0,n}\right).$$

Отметим также, что

$$T_n(x_m)=\cos(n\arccos x_m)=\cos m\pi=(-1)^m,$$

а следовательно, знаки многочлена $T_n(x)$ в точках x_m чередуются. Таким образом, $\max|T_n(x)|=1$.

5. Обозначим через $\overline{T_n(x)}$ многочлен, который получается из многочлена Чебышева $T_n(x)$ нормированием, то есть приведением к виду, в котором старший коэффициент равен 1. Тогда $\overline{T_n(x)}=\frac{1}{2^{n-1}}T_n(x)$.

Докажем, что многочлен $\overline{T_n(x)}$ имеет на отрезке $[-1,1]$ наименьшее значение максимума модуля среди всех многочленов n-ой степени со старшим коэффициентом 1. Допустим противное, то есть что существует многочлен $\overline{P_n(x)}$ степени n со старшим коэффициентом 1, причем

$$\max_{[-1,1]}|\overline{P_n(x)}|<\max_{[-1,1]}|\overline{T_n(x)}|=\frac{1}{2^{n-1}}. \qquad (4.3.2)$$

Рассмотрим разность $\overline{T_n(x)} - \overline{P_n(x)}$. Это многочлен степени не выше, чем $(n-1)$. По свойству 4 многочленов Чебышева и благодаря (4.3.2) эта разность на отрезке $[-1,1]$ имеет $(n+1)$ значение с чередующимися знаками в точках $x_m = \cos\frac{m\pi}{n}$, $m = \overline{0,n}$. Следовательно, функция $\overline{T_n}(x) - \overline{P_n}(x)$, по крайней мере, в n точках имеет значение, равное нулю. Итак, многочлен степени не выше, чем $(n-1)$, имеет n корней. Полученное противоречие доказывает свойство 5.

Построим графики первых четырех многочленов Чебышева. Они иллюстрируют свойства этих многочленов (см. рис. 4.3.1).

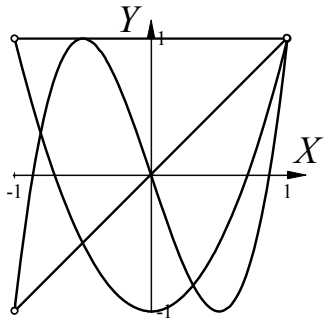


Рис. 4.3.1

Возьмем в качестве узлов интерполяции корни многочлена $T_{n+1}(x)$. Тогда многочлен $\omega_n(x) = (x-x_0)(x-x_1)...(x-x_n)$ будет пропорционален $T_{n+1}(x)$, причем он получается из $T_{n+1}(x)$ следующим образом: $\omega_n(x) = T_{n+1}(x)\cdot 2^{-n} = \overline{T_{n+1}}(x)$. В этом случае оценка погрешности интерполяции будет выглядеть так:

$$|R_n(x)| \le \frac{M_{n+1}}{(n+1)!}|\omega_n(x)| \le \frac{M_{n+1}}{(n+1)!}\max_{[-1,1]}|\omega_n(x)| = \frac{M_{n+1}}{(n+1)!\cdot 2^n}.$$

Так как $\overline{T_{n+1}}(x)$ имеет на отрезке $[-1,1]$ наименьшее значение максимума модуля, то вышеуказанную оценку за счет другого выбора узлов интерполяции улучшить невозможно.

В случае произвольного отрезка $[a,b]$ нужно преобразовать отрезок $[a,b]$ в отрезок $[-1,1]$ и в качестве узлов интерполяции x_i взять точки, соответствующие корням многочлена Чебышева $T_{n+1}(t)$ на отрезке $[-1,1]$.

$$x = \frac{1}{2}((b-a)t + b + a),\ -1 \le t \le 1,\ a \le x \le b.$$

В качестве узлов интерполяции выбираем точки

$$x_i = \frac{1}{2}\left((b-a)\cos\frac{(2i+1)\pi}{2n+2} + b + a\right).$$

Локальная интерполяция. Сплайны

Сплайном называется функция, «склеенная» из «кусков» многочленов. Точнее, сплайном $S(x)$ порядка m с узлами x_i, $a = x_0 < \ldots < x_i < \ldots < x_n = b$ называется непрерывная функция, которая на каждом из элементарных отрезков $[x_i, x_{i+1}]$, $i = \overline{0, n-1}$, является многочленом степени не выше m, причем некоторая производная $S^{(k)}(x)$, $1 \le k \le m$ может быть разрывной.

Из определения следует, что основными характеристиками сплайна являются:

– наивысший порядок многочленов, из которых склеен сплайн;

– количество и расположение узлов интерполяции;

– степень гладкости склейки.

Для характеристики гладкости склейки сплайна в узлах применяется понятие дефекта сплайна. Говорят, что сплайн $S(x)$ во внутреннем узле x_i $(i = \overline{1, n-1})$ имеет дефект k_i, если в точке x_i функции $S'(x)$, $S''(x)$, …, $S^{(m-k_i)}(x)$ являются непрерывными, а $S^{(m-k_i+1)}(x)$ в этой точке терпит разрыв. Таким образом, дефект сплайна определяется в каждом внутреннем узле. Число $k = \max\limits_{1 \le i \le n-1} k_i$ называется дефектом сплайна.

4.4. Линейная интерполяция. Квадратичная интерполяция. Интерполяция кубическими сплайнами. Обратная интерполяция с помощью многочлена Лагранжа. Эмпирические зависимости. Метод наименьших квадратов

Линейная интерполяция

Суть ее заключается в том, что на каждом элементарном отрезке $[x_i, x_{i+1}]$, $i = \overline{0, n-1}$, функция аппроксимируется линейной функцией. Геометрически это означает, что через точки табличной функции проводится ломаная. На каждом из элементарных отрезков линейный сплайн задается определенным уравнением. Найдем уравнение

сплайна на произвольном отрезке $[x_i, x_{i+1}]$. Здесь следует воспользоваться уравнением прямой, проходящей через две точки (x_i, y_i) и (x_{i+1}, y_{i+1}).

$$\frac{x - x_i}{x_{i+1} - x_i} = \frac{y - y_i}{y_{i+1} - y_i}$$

$$y = y_i + \frac{y_{i+1} - y_i}{x_{i+1} - x_i}(x - x_i)$$

Таким образом, для построения линейной интерполяции следует найти промежуток $[x_i, x_{i+1}]$, в который попадает значение x, а затем с использованием вышеуказанной формулы посчитать значение функции.

Квадратичная интерполяция

Шаблоном для квадратичной интерполяции являются три узла x_{i-1}, x_i, x_{i+1}. Суть ее заключается в том, что на элементарном отрезке $[x_{i-1}, x_{i+1}]$, содержащем три узла интерполяции, функция аппроксимируется квадратичной (параболической) функцией $y = a_i x^2 + b_i x + c_i$. Найдем неизвестные a_i, b_i, c_i из условия интерполяции, то есть из условия совпадения значений сплайна в узловых точках x_{i-1}, x_i, x_{i+1} со значениями табличной функции.

$$\begin{cases} a_i x_{i-1}^2 + b_i x_{i-1} + c_i = y_{i-1} \\ a_i x_i^2 + b_i x_i + c_i = y_i \\ a_i x_{i+1}^2 + b_i x_{i+1} + c_i = y_{i+1} \end{cases}.$$

Чтобы определить значение функции при определенном x при квадратичной интерполяции, нужно провести следующие вычисления:

– определить три ближайших к x узла интерполяции;

– составить и решить вышеуказанную систему;

– по полученным коэффициентам вычислить значение функции в точке x.

Заметим, что ответы могут различаться в зависимости от того, $x \in (x_{i-1}, x_i)$ или $x \in (x_i, x_{i+1})$.

Интерполяция кубическими сплайнами

Определим сплайн $S(x)$ так, чтобы между любыми двумя узлами интерполяции он был многочленом третьей степени:

$$S(x) = a_i + b_i(x - x_{i-1}) + c_i(x - x_{i-1})^2 + d_i(x - x_{i-1})^3,$$

где $x_{i-1} \le x \le x_i$, $i = \overline{1,n}$. Для определения всех коэффициентов a_i, b_i, c_i, d_i нужно $4n$ уравнений. Часть из этих уравнений мы получим из условия прохождения сплайна через узлы интерполяции: $S(x_{i-1}) = y_{i-1}$, $S(x_i) = y_i$, $i = \overline{1,n}$. Получаем $2n$ уравнений:

$$a_i = y_{i-1},\ i = \overline{1,n}; \tag{4.4.1}$$

$$a_i + b_i h_i + c_i h_i^2 + d_i h_i^3 = y_i,\ i = \overline{1,n}, \tag{4.4.2}$$

где $h_i = x_i - x_{i-1}$.

В качестве дополнительных условий возьмем условия непрерывности первой и второй производных во внутренних узлах, то есть условия гладкости. Так как

$$S'(x) = b_i + 2c_i(x - x_{i-1}) + 3d_i(x - x_{i-1})^2,$$

$$S''(x) = 2c_i + 6d_i(x - x_{i-1}),$$

получаем

$$b_i + 2c_i h_i + 3d_i h_i^2 = b_{i+1},\ i = \overline{1,n-1} \tag{4.4.3}$$

$$c_i + 3d_i h_i = c_{i+1},\ i = \overline{1,n-1}. \tag{4.4.4}$$

(Для получения этих уравнений нужно приравнять производные в точке x_i, вычисленные через левый и правый интервал от x_i, $i = \overline{1,n-1}$.) Получаем еще $(2n-2)$ уравнения. Последние два уравнения можно получить, приравняв к нулю вторые производные в концевых точках: $S''(x_0) = 0$, $S''(x_n) = 0$.

$$c_1 = 0, \tag{4.4.5}$$

$$c_n + 3d_n h_n = 0. \tag{4.4.6}$$

Решив систему уравнений (4.4.1)-(4.4.6), мы получим коэффициенты a_i, b_i, c_i, d_i. С целью экономии памяти ЭВМ эту систему можно представить в более компактном виде.

1. Прежде всего заметим, что из (4.4.1) можно найти все a_i.
2. Далее выразим из (4.4.4) и (4.4.6) d_i.

$$\begin{cases} d_i = \dfrac{c_{i+1} - c_i}{3h_i}, \ i = \overline{1, n-1} \\ d_n = -\dfrac{c_n}{3h_n} \end{cases}. \qquad (4.4.7)$$

Заметим, что первой из этих формул можно пользоваться и при $i = n$, если положить $c_{n+1} = 0$.

3. Исключим a_i и d_i из (4.4.2).

$$y_{i-1} + b_i h_i + c_i h_i^2 + \frac{c_{i+1} - c_i}{3h_i} h_i^3 = y_i$$

$$b_i h_i = y_i - y_{i-1} - c_i h_i^2 - \frac{c_{i+1}}{3} h_i^2 + \frac{c_i}{3} h_i^2$$

$$b_i = \frac{y_i - y_{i-1}}{h_i} - \frac{h_i}{3}(2c_i + c_{i+1}), \ i = \overline{1, n}. \qquad (4.4.8)$$

4. Из уравнений (4.4.3) исключим b_i и d_i, пользуясь (4.4.7) и (4.4.8).

$$\frac{y_i - y_{i-1}}{h_i} - \frac{h_i}{3}(c_{i+1} + 2c_i) + 2c_i h_i + 3\frac{c_{i+1} - c_i}{3h_i} h_i^2 = \frac{y_{i+1} - y_i}{h_{i+1}} - \frac{h_{i+1}}{3}(c_{i+2} + 2c_{i+1}),$$

$$i = \overline{1, n-1}.$$

Как видим, записанное выше уравнение связывает значения коэффициентов c_i в трех соседних узлах. Найдем коэффициенты при неизвестных.

При c_i: $-\frac{2}{3}h_i + 2h_i - h_i = \frac{h_i}{3}$.

При c_{i+1}: $-\frac{h_i}{3} + h_i + \frac{2}{3}h_{i+1} = \frac{2}{3}(h_i + h_{i+1})$.

При c_{i+2}: $\frac{h_{i+1}}{3}$.

Свободный член: $\frac{y_{i+1} - y_i}{h_{i+1}} - \frac{y_i - y_{i-1}}{h_i}$.

Произведем переиндексацию:

$$i - 1 \to i - 2, \ i + 1 \to i, \ i \to i - 1, \ i + 2 \to i + 1.$$

$$\frac{y_{i-1} - y_{i-2}}{h_{i-1}} - \frac{h_{i-1}}{3}(c_i + 2c_{i-1}) + 2c_{i-1}h_{i-1} + 3\frac{c_i - c_{i-1}}{3h_{i-1}} h_{i-1}^2 = \frac{y_i - y_{i-1}}{h_i} - \frac{h_i}{3}(c_{i+1} + 2c_i)$$

Коэффициенты:

При c_{i-1}: $\frac{h_{i-1}}{3}$.

При c_i: $\frac{2}{3}(h_{i-1}+h_i)$.

При c_{i+1}: $\frac{h_i}{3}$.

Свободный член: $\frac{y_i - y_{i-1}}{h_i} - \frac{y_{i-1} - y_{i-2}}{h_{i-1}}$.

В результате получаем следующую систему из $(n+1)$ уравнения:

$$\begin{cases} c_1 = 0 \\ F(c_{i-1}, c_i, c_{i+1}) = F_i; \quad i = \overline{2,n} \\ c_{n+1} = 0 \end{cases} \tag{4.4.9}$$

Эта система решается прогонкой.

Таким образом, чтобы произвести кубическую сплайн-интерполяцию, следует:

1. Определить коэффициенты a_i, b_i, c_i, d_i, $i = \overline{1,n}$, в следующем порядке: a_i – из уравнения (4.4.1), c_i – решая (4.4.9) прогонкой, d_i – с помощью (4.4.7), b_i – с помощью (4.4.8).

2. Определить интервал $[x_{i-1}, x_i]$, который содержит аргумент x, и в качестве приближенного значения функции в этой точке взять значение сплайна:

$$y(x) \approx S(x) = a_i + b_i(x - x_{i-1}) + c_i(x - x_{i-1})^2 + d_i(x - x_{i-1})^3.$$

Обратная интерполяция с помощью многочлена Лагранжа

Пусть задана табличная функция $y = f(x)$. Если $f(x)$ непрерывна и монотонна на отрезке $[a,b]$, где $a = \min\{x_i\}$, $b = \max\{x_i\}$, то, по теореме об обратной функции, на отрезке $[c,d]$, где $c = \min\{y_i\}$, $d = \max\{y_i\}$ определена обратная функция $x = g(y)$, также монотонная и непрерывная. Пусть следует определить по таблице такое $\bar{x}$, что $\bar{y} = f(\bar{x})$. Эту задачу можно решить с помощью обратного интерполирования. Если известна таблица (4.1.1), значит, известна таблица для $x = g(y)$ (строка значений равна строке аргументов). Поэтому можно построить интерполяционный многочлен:

$$L_n(y) = \sum_{i=0}^{n} x_i \frac{(y-y_0)(y-y_1)...(y-y_{i-1})(y-y_{i+1})...(y-y_n)}{(y_i-y_0)(y_i-y_1)...(y_i-y_{i-1})(y_i-y_{i+1})...(y_i-y_n)}.$$

Тогда в качестве $\bar{x} = g(\bar{y})$ можно принять значение многочлена в точке $\bar{y}$, то есть $\bar{x} = g(\bar{y}) \approx L_n(\bar{y})$.

Эмпирические зависимости. Метод наименьших квадратов

При интерполяции требуется совпадение значений аппроксимируемой функции $f(x)$ и аппроксимирующей функции $\varphi(x)$ в узлах: $f(x_i)=\varphi(x_i)$. Между тем значения функции могут быть найдены путем экспериментов, измерений, которые могут содержать ошибки. Поэтому совпадение значений в узлах будет означать повторение ошибок эксперимента. Задача поиска эмпирической зависимости отличается от задачи интерполирования. Графически это выглядит так, как показано на рисунке 4.4.1.

То есть график эмпирической зависимости не обязательно проходит через узловые точки.

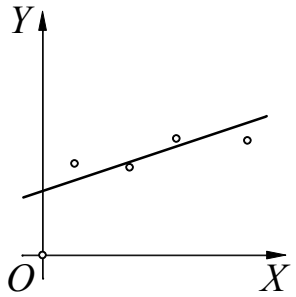


Рис. 4.4.1

Поставим задачу более четко. Пусть дана таблица значений функции $y=f(x)$ (см. табл. 4.4.1).

Таблица 4.4.1

x	x_0	x_1	…	x_n
y	y_0	y_1	…	y_n

Следует найти такую функцию $y=\varphi(x)$ из определенного класса функций, которая бы в узлах $\{x_i\}$ мало отличалась бы от значений функции $f(x)$. Для характеристики близости $\varphi(x)$ к $f(x)$ вводится понятие отклонения функции в узле x_i, а именно $\varepsilon_i=\varphi(x_i)-f(x_i)$. Различные способы минимизации модулей этих величин определяют тот или иной метод приближения функции. Например, можно минимизировать величины $\max\limits_i|\varepsilon_i|$, $\sum\limits_{i=0}^{n}|\varepsilon_i|$, $\sum\limits_{i=0}^{n}|\varepsilon_i|^2$. Минимизация последней величины определяет метод наименьших квадратов, который является наиболее оптимальным.

Задача поиска эмпирической зависимости решается в два этапа.

1. Подбор вида эмпирической зависимости $y = \varphi(x, a_0, a_1, ..., a_m)$, где a_0, …, a_m – параметры эмпирической зависимости, а $m << n$. Например, в зависимости $y = cx^{\alpha}$ два параметра – $a_0 = c$, $a_1 = \alpha$.

2. Определение параметров эмпирической зависимости с помощью того или иного способа минимизации модулей отклонений.

Заметим, что линейная зависимость $y = ax + b$ содержит два параметра. И часто функциональную зависимость с двумя параметрами можно свести к линейной. Например:

$$y = cx^{\alpha},$$

$$\ln y = \ln c + \alpha \ln x.$$

Если обозначить

$$Y = \ln y,\ C = \ln c,\ X = \ln x,$$

то получим линейную зависимость $Y = C + \alpha X$.

Часто вид эмпирической зависимости известен из физических соображений. Если вид заранее неизвестен, то бывает полезным экспериментальные данные нанести на координатную плоскость и по графику угадать вид зависимости путем сравнения этого графика с ранее известными (см. рис. 4.4.2).

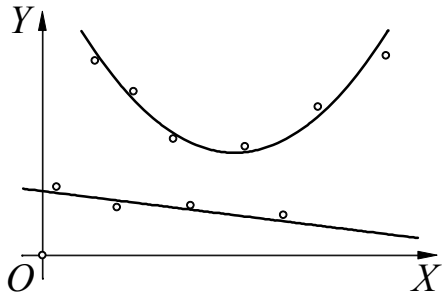


Рис. 4.4.2

Пусть вид эмпирической зависимости найден $y = \varphi(x, a_0, ..., a_m)$. Надо определить параметры a_0, …, a_m. Как было сказано ранее, они определяются путем минимизации величин $|\varepsilon_i|$, $i = \overline{0,n}$. Рассмотрим метод наименьших квадратов.

Будем минимизировать величину $s = \sum_{i=0}^{n} |\varepsilon_i|^2 = \sum_{i=0}^{n} (\varphi(x_i, a_0, ..., a_m) - y_i)^2$. Сделаем предположение, что функция $y = \varphi(x, a_0, ..., a_m)$ является линейной относительно параметров:

$$\varphi(x, a_0, ..., a_m) = a_0\varphi_0(x) + ... + a_m\varphi_m(x).$$

Так как функция s есть функция m+1 переменных, то стационарную точку найдем из условия одновременного равенства нулю всех частных производных:

$$\frac{\partial s}{\partial a_0}=0,\ \frac{\partial s}{\partial a_1}=0,\ \ldots,\ \frac{\partial s}{\partial a_m}=0.$$

Решая эту систему из m+1 уравнения с m+1 неизвестным, получаем искомые параметры. (Если все функции $\varphi_i(x)$ $(i=\overline{0,m})$ линейно независимы, то можно показать, что решение системы будет единственно, и найденная точка будет являться точкой минимума.)

Применим МНК для частного случая, когда вид эмпирической зависимости такой: $\varphi(x,a,b,c)=ax^2+bx+c$. (Заметим, что эта зависимость является линейной относительно параметров a, b и c и функции $\varphi_0(x)=x^2$, $\varphi_1(x)=x$, $\varphi_2(x)=1$ линейно независимы.)

$$s(a,b,c)=\sum_{i=0}^{n}(ax_i^2+bx_i+c-y_i)^2$$

$$\begin{cases}\dfrac{\partial s}{\partial a}=2\sum\limits_{i=0}^{n}(ax_i^2+bx_i+c-y_i)x_i^2=0\\ \dfrac{\partial s}{\partial b}=2\sum\limits_{i=0}^{n}(ax_i^2+bx_i+c-y_i)x_i=0\\ \dfrac{\partial s}{\partial c}=2\sum\limits_{i=0}^{n}(ax_i^2+bx_i+c-y_i)=0\end{cases}$$

$$\begin{cases}a\sum\limits_{i=0}^{n}x_i^4+b\sum\limits_{i=0}^{n}x_i^3+c\sum\limits_{i=0}^{n}x_i^2=\sum\limits_{i=0}^{n}x_i^2y_i\\ a\sum\limits_{i=0}^{n}x_i^3+b\sum\limits_{i=0}^{n}x_i^2+c\sum\limits_{i=0}^{n}x_i=\sum\limits_{i=0}^{n}x_iy_i\\ a\sum\limits_{i=0}^{n}x_i^2+b\sum\limits_{i=0}^{n}x_i+c(n+1)=\sum\limits_{i=0}^{n}y_i\end{cases}$$

4.5. Лабораторная работа «Аппроксимация функций»

(5 часов)

Цель работы: научиться в программе MathCad аппроксимировать функции, заданные таблично, с помощью локальной и глобальной интерполяции и по методу наименьших квадратов; научиться реализовывать эти виды аппроксимации в собственных программах.

Используемое программное обеспечение: Borland Pascal (или Delphi) и MathCad.

Основное задание:

Первые два столбца в таблицах каждого варианта задают табличную функцию (таблицы 4.5.1-4.5.6). В четвертом столбце напротив соответствующего варианта указана точка, в которой требуется посчитать значение функции с помощью того или иного вида аппроксимации. Приближающая функция дана для метода наименьших квадратов.

1. В программе MathCad определите значение функции в точке t с помощью глобального интерполяционного многочлена, найденного как в каноническом виде, так и в форме Лагранжа, методом наименьших квадратов и с помощью локальной линейной интерполяции.
2. Напишите программу для вычисления значения функции в точке t с помощью интерполяционного многочлена, найденного в каноническом виде.
3. Напишите программу для вычисления значения функции в точке t с помощью интерполяционного многочлена Лагранжа.
4. Напишите программу для вычисления значения функции в точке t методом наименьших квадратов.
5. Напишите программу для вычисления значения функции в точке t с помощью локальной линейной интерполяции.

Полученные значения округляйте до 0,000001.

Таблица 4.5.1
приближающая функция $y = ax^m$

x	y	№ варианта	t
0,43	1,63597	1	0,702
0,48	1,73234	7	0,512
0,55	1,87686	13	0,645
0,62	2,03345	19	0,736
0,70	2,22846	25	0,608
0,75	2,35973		

Таблица 4.5.2
приближающая функция $y = ae^{mx}$

x	y	№ варианта	t
0,02	1,02316	2	0,102
0,08	1,09590	8	0,114
0,12	1,14725	14	0,125
0,17	1,21483	20	0,203
0,23	1,30120	26	0,154
0,30	1,40976		

Таблица 4.5.3

приближающая функция $y = \dfrac{1}{ax+b}$

x	y	№ варианта	t
0,35	2,73951	3	0,526
0,41	2,30080	9	0,453
0,47	1,96864	15	0,482
0,51	1,78776	21	0,552
0,56	1,59502	27	0,436
0,64	1,34310		

Таблица 4.5.4

приближающая функция $y = a \cdot \ln x + b$

x	y	№ варианта	t
0,41	2,57418	4	0,616
0,46	2,32513	10	0,478
0,52	2,09336	16	0,665
0,60	1,86203	22	0,537
0,65	1,74926	28	0,673
0,72	1,62098		

Таблица 4.5.5

приближающая функция $y = \dfrac{a}{x} + b$

x	y	№ варианта	t
0,68	0,80866	5	0,896
0,73	0,89492	11	0,812
0,80	1,02964	17	0,774
0,88	1,20966	23	0,955
0,93	1,34087	29	0,715
0,99	1,52368		

Таблица 4.5.6

приближающая функция $y = \dfrac{x}{ax+b}$

x	y	№ варианта	t
0,11	9,05421	6	0,314
0,15	6,61659	12	0,235
0,21	4,69170	18	0,332
0,29	3,35106	24	0,275
0,35	2,73951	30	0,186
0,40	2,36522		

Комментарии к основному заданию

1. Для выполнения работы в программе MathCad понадобятся панели инструментов «*Матрицы*», «*Матанализ*», «*Арифметика*», «*Графики*», «*Булево*», «*Программирование*». Вновь будем по возможности составлять программу так, чтобы она обладала универсальностью. В том случае, если придется изменить условие, то пусть в программе при этом придется сделать минимум изменений. Укажем некоторые команды, которые нам понадобятся для выполнения работы.

`length(x)` – длина массива x.

$x:=x^{T}$ – транспонирование массива x. Некоторые команды программы MathCad применимы к массивам-строкам, некоторые – к массивам-столбцам. С помощью команды транспонирования (панель «*Матрицы*») можно преобразовывать рассматриваемый массив.

`linterp(x,y,x0)` – значение интерполируемой функции, вычисленное в точке x_0 с помощью локальной линейной интерполяции на основе значений векторов x и y.

Программа MathCad позволяет искать практически любые регрессионные модели, даже не линейные относительно параметров. Для

этого необходимо задать вектор, координатами которого являются приближающая функция и ее частные производные по параметрам (в примере – это вектор *F*). Затем надо задать начальное приближение для параметров (в примере – это вектор *vb*). Начальное приближение можно брать наугад, но затем всегда следует проверять правильность построенной модели графически, так как в случае неудачно подобранного начального приближения программа может выдать неправильный результат. Далее воспользуемся командой `genfit(x,y,vb,F)`, которая возвратит коэффициенты регрессионной модели. Напомним, что индексация в массивах в MathCad'е начинается с нуля.

Пример: «screenshot» с программой (для выполнения 30-го варианта).

$x := (\ 0.11 \quad 0.15 \quad 0.21 \quad 0.29 \quad 0.35 \quad 0.40\ )$

$y := (\ 9.05421 \quad 6.61659 \quad 4.69170 \quad 3.35106 \quad 2.73951 \quad 2.36522\ )$

$t := 0.186$

$n := \text{length}(x^T) \quad n = 6 \qquad x := x^T \qquad y := y^T$

Интерполяционный многочлен в каноническом виде

$$a := \left| \begin{array}{l} \text{for } i \in 0..n-1 \\ \quad \text{for } j \in 0..n-1 \\ \quad\quad a_{i,j} \leftarrow (x_i)^j \\ a \end{array} \right. \qquad b := \left| \begin{array}{l} \text{for } i \in 0..n-1 \\ \quad b_i \leftarrow y_i \\ b \end{array} \right.$$

$c := \text{lsolve}(a, b) \qquad L1 := \sum_{i=0}^{n-1} c_i \cdot t^i \qquad L1 = 5.305042$

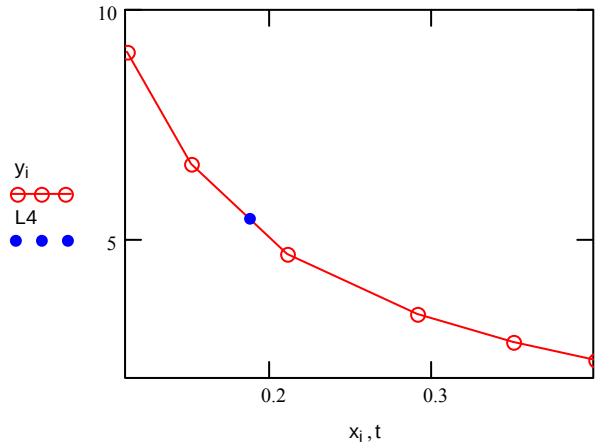
Интерполяционный многочлен в форме Лагранжа

$$L2 := \left| \begin{array}{l} L \leftarrow 0 \\ \text{for } i \in 0..n-1 \\ \quad P \leftarrow 1 \\ \quad P \leftarrow \left| \begin{array}{l} \text{for } j \in 0..n-1 \\ \quad P \leftarrow P \cdot \dfrac{t - x_j}{x_i - x_j} \quad \text{if } i \neq j \\ P \end{array} \right. \\ \quad L \leftarrow L + P \cdot y_i \\ L \end{array} \right.$$

$L2 = 5.305042$

Локальная линейная интерполяция

$L4 := linterp(x, y, t)$ $\quad L4 = 5.461656$ $\quad i := 0 .. n - 1$



Метод наименьших квадратов

$$F(w, u) := \begin{bmatrix} \dfrac{w}{u_0 \cdot w + u_1} \\ \dfrac{-w^2}{(u_0 \cdot w + u_1)^2} \\ \dfrac{-w}{(u_0 \cdot w + u_1)^2} \end{bmatrix} \qquad vb := \begin{pmatrix} 1 \\ 1 \end{pmatrix} \qquad p := genfit(x, y, vb, F) \qquad p = \begin{pmatrix} 0.385 \\ -0.031 \end{pmatrix}$$

$g(r) := F(r, p)_0$ $\quad i := 0 .. n - 1$

Ответ y=x/(0.385*x-0.031)

$g(t) = 4.529315$

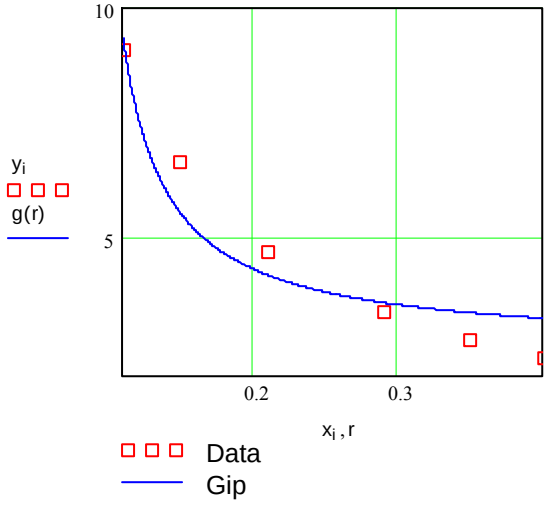


Рис. 4.5.1. Различные виды аппроксимаций в программе MathCad

2. Основная часть программы 2 должна содержать три блока. В первом блоке следует задать коэффициенты системы для определения коэффициентов интерполяционного многочлена. Это можно сделать практически так же, как в программе MathCad. Во втором блоке сле-

дует решить составленную систему линейных уравнений методом Гаусса. Для этого следует написать отдельную процедуру или функцию для решения произвольной системы линейных уравнений, поместить ее в отдельный модуль и просто сослаться на нее. В третьем блоке следует вычислить значение найденного многочлена в заданной точке.

3. Внешний вид Паскаль-программы для решения задачи 3 практически ничем не отличается от соответствующей программы MathCad'a, приведенной в «screenshot'е».

4. Перед написанием программы для решения задачи 4 следует сначала на бумаге привести вашу приближающую функцию к линейному виду и, применив к ней метод наименьших квадратов, составить систему линейных уравнений для определения коэффициентов приближающей функции. Сама программа так же, как и программа 2, будет состоять из трех блоков. В первом задаем коэффициенты системы, во втором – решаем ее, ссылаясь уже на готовый модуль с методом Гаусса, в третьем – вычисляем значение приближающей функции в заданной точке.

5. При написании программы 5 следует организовать цикл для определения отрезка, в который попадает данная точка, и затем вычислить по двум соседним узлам значение линейной функции.

Ответы в программах 2, 3, 5 должны в точности совпадать с ответами в MathCad'е. Ответ в программе 4 может отличаться от ответа, полученного в MathCad'е, так как MathCad не приводит приближающую функцию к линейному виду.

Дополнительное задание

1. Напишите программу для вычисления значения функции в указанной точке с помощью локальной квадратичной интерполяции.
2. В программе MathCad найдите значение функции в указанной точке с помощью интерполяции кубическим сплайном.
3. В программе MathCad найдите значение функции в указанной точке, построив линейную и квадратичную регрессионные модели.

Комментарии к дополнительному заданию

1. Заметим, что ответы могут различаться в зависимости от того, как будут выбраны узловые точки для определения коэффициентов квадратичной функции: данная точка может попасть как в первый, так и во второй элементарный отрезок.

2. Укажем необходимые команды.

`cspline(x,y)` – возвращает вектор коэффициентов, используемый функцией `interp` для построения кубического сплайна, который интерполирует значения, представленные в векторах *x* и *y*. При этом на поведение сплайна на границе области никаких ограничений не накладывается.

`pspline(x,y)` – то же, что и `cspline`, но создаваемый сплайн на границе области имеет равную нулю третью производную.

`lspline(x,y)` – то же, что и `cspline`, но создаваемый сплайн на границе области имеет равные нулю вторую и третью производные.

`interp(s,x,y,`x_0`)` – возвращает интерполируемое значение в точке x_0. Вектор *s* есть результат, возвращаемый одной из функций `cspline, pspline, lspline`.

3. Укажем необходимые команды.

`slope(x,y)` – возвращает коэффициент при *x* в линейной регрессионной модели.

`intercept(x,y)` – возвращает свободный член в линейной регрессионной модели.

`regress(x,y,l)` – возвращает вектор, последние *l*+1 координат которого – коэффициенты степенной регрессионной модели степени *l*.

ГЛАВА 5. ЧИСЛЕННОЕ ИНТЕГРИРОВАНИЕ

5.1. Понятие определенного интеграла. Формулы прямоугольников. Формула трапеций. Формула Симпсона. Оценка погрешности квадратурных формул

Понятие определенного интеграла

Пусть функция $f(x)$ задана на отрезке $[a,b]$. Разобьем этот отрезок точками $x_0 = a < \ldots < x_i < \ldots < x_n = b$, $\Delta x_i = x_{i+1} - x_i$. На каждом отрезке выберем произвольную точку $q_i \in [x_i, x_{i+1}]$, составим произведение $f(q_i)\Delta x_i$ и составим сумму всех таких произведений, то есть интегральную сумму $\sum_{i=0}^{n-1} f(q_i)\Delta x_i$. Определенным интегралом $\int_a^b f(x)dx$ называется предел интегральной суммы при $\max_i \Delta x_i \to 0$, то есть

$$\int_a^b f(x)dx = \lim_{\max_i \Delta x_i \to 0} \sum_{i=0}^{n-1} f(q_i)\Delta x_i .$$

Причем этот предел не зависит от выбора точек q_i. В частном случае, когда функция $f(x) > 0$ на отрезке $[a,b]$, определенный интеграл имеет следующий геометрический смысл. Интегральная сумма представляет собой площадь ступенчатой фигуры. Если $\max_i \Delta x_i \to 0$, ломаная стремится занять положение кривой $y = f(x)$, следовательно, определенный интеграл равен площади криволинейной трапеции.

Из курса математического анализа известно, что если существует первообразная функция $F(x)$ для функции $y = f(x)$, то определенный интеграл есть приращение первообразной на отрезке $[a,b]$, то есть

$$\int_a^b f(x)dx = F(b) - F(a).$$

Это формула Ньютона – Лейбница. Но в некоторых случаях этой формулой воспользоваться нельзя. Например, существует целый класс «неберущихся» интегралов, то есть таких, для которых первообразные не вычисляются в элементарных функциях. Между тем некоторые из них играют большую роль в математике, как, например,

$\int_0^x e^{-t^2} dt$. Кроме того, функция может быть задана таблично, или в принципе интеграл можно посчитать по формуле Ньютона – Лейбница, но практически это сделать очень сложно, как, например, в случае $\int_0^x t^{100} \sin 2t dt$.

Суть численного интегрирования заключается в том, что подынтегральная функция $f(x)$ заменяется более простой функцией $\varphi(x)$, интеграл от которой вычислить легко, и тогда

$$\int_a^b f(x)dx \approx \int_a^b \varphi(x)dx.$$

Формулы численного интегрирования называются квадратурными.

Формулы прямоугольников

В основу использования формул прямоугольников положено понятие интегральной суммы, а именно в этих формулах интеграл $\int_a^b f(x)dx$ буквально заменяется интегральной суммой, в которой выбор точек q_i производится вполне определенным образом.

Разобьем отрезок $[a,b]$ на n равных частей с шагом $h = x_{i+1} - x_i$ $(i = \overline{0,n-1})$. Если в качестве точек q_i взять левые концы элементарных отрезков, то получим формулу левых прямоугольников:

$$\int_a^b f(x)dx \approx h\sum_{i=0}^{n-1} f(x_i).$$

Взяв в качестве точек q_i правые концы прямоугольников, получим формулу правых прямоугольников:

$$\int_a^b f(x)dx \approx h\sum_{i=0}^{n-1} f(x_{i+1}).$$

Если в качестве точек q_i взять середины элементарных отрезков, получим формулу средних прямоугольников:

$$\int_a^b f(x)dx \approx h\sum_{i=0}^{n-1} f\left(\overline{x_i}\right), \text{ где}$$

$$\overline{x_i} = \frac{x_i + x_{i+1}}{2} = x_i + \frac{h}{2}.$$

Заметим, что формулы левых и правых прямоугольников используются очень редко, так как точность формулы средних прямоугольников гораздо выше. Заметим также, что в формулах прямоугольников подынтегральная функция аппроксимируется многочленом нулевой степени, то есть константой.

Формула трапеций

Пусть вновь требуется вычислить интеграл $\int_a^b f(x)dx$. Разобьем отрезок $[a,b]$ на n равных частей с шагом $h = x_{i+1} - x_i$ $(i = \overline{0,n-1})$. Подынтегральную функцию $f(x)$ на каждом элементарном отрезке аппроксимируем линейной функцией $g(x)$, то есть сплайном первой степени. Таким образом, площадь криволинейной трапеции мы заменяем суммарной площадью обычных трапеций.

$$\int_a^b f(x)dx \approx \frac{f(x_0)+f(x_1)}{2}h + \frac{f(x_1)+f(x_2)}{2}h + \frac{f(x_2)+f(x_3)}{2}h + \ldots +$$

$$+ \frac{f(x_{n-1})+f(x_n)}{2}h = h\left(\frac{f(x_0)+f(x_n)}{2} + f(x_1) + f(x_2) + \ldots + f(x_{n-1})\right) =$$

$$= h\left(\frac{f(x_0)+f(x_n)}{2} + \sum_{i=1}^{n-1} f(x_i)\right)$$

Мы проводили рассуждение в предположении, что функция $f(x)$ является положительной на отрезке $[a,b]$, однако формула остается верной и в том случае, когда функция $f(x)$ меняет знак на отрезке интегрирования.

Формула Симпсона

Отрезок интегрирования $[a,b]$ разобьем на четное число элементарных отрезков равной длины точками $x_0 = a < x_1 < \ldots < x_{2n-1} < x_{2n} = b$ с шагом $h = x_{i+1} - x_i$ $(i = \overline{0,2n-1})$. На каждом отрезке $[x_{i-1}, x_{i+1}]$ аппроксимируем многочленом второй степени подынтегральную функцию, которая на этом отрезке имеет вид $g_i(x) = a_i x^2 + b_i x + c_i$. Заметим, что i принимает здесь только нечетные значения от 1 до $2n-1$. Таким образом, подынтегральная функция аппроксимируется совокупностью квадратных многочленов или сплайном второй степени.

$$\int_a^b f(x)dx \approx \int_{x_0}^{x_2} g_1(x)dx + \int_{x_2}^{x_4} g_3(x)dx + \ldots + \int_{x_{i-1}}^{x_{i+1}} g_i(x)dx + \ldots + \int_{x_{2n-2}}^{x_{2n}} g_{2n-1}(x)dx .$$

Вычислим произвольный интеграл из правой части.

$$\int_{x_{i-1}}^{x_{i+1}} g_i(x)dx = \int_{x_{i-1}}^{x_{i+1}} (a_i x^2 + b_i x + c_i)dx = \frac{a_i x^3}{3}\Bigg|_{x_{i-1}}^{x_{i+1}} + \frac{b_i x^2}{2}\Bigg|_{x_{i-1}}^{x_{i+1}} + c_i x\Bigg|_{x_{i-1}}^{x_{i+1}} =$$

$$= \frac{a_i}{3}(x_{i+1}^3 - x_{i-1}^3) + \frac{b_i}{2}(x_{i+1}^2 - x_{i-1}^2) + c_i(x_{i+1} - x_{i-1}) =$$

$$= \frac{x_{i+1} - x_{i-1}}{6}\left[2a_i(x_{i+1}^2 + x_{i+1}x_{i-1} + x_{i-1}^2) + 3b_i(x_{i+1} + x_{i-1}) + 6c_i\right] \qquad (5.1.1)$$

Коэффициенты a_i, b_i и c_i могут быть найдены из условия интерполяции, то есть из уравнений:

$$y_{i-1} = a_i x_{i-1}^2 + b_i x_{i-1} + c_i ,$$
$$y_i = a_i x_i^2 + b_i x_i + c_i ,$$
$$y_{i+1} = a_i x_{i+1}^2 + b_i x_{i+1} + c_i .$$

Заметим, что точка x_i является серединой отрезка $[x_{i-1}, x_{i+1}]$, следовательно, $x_i = \frac{x_{i-1} + x_{i+1}}{2}$. Подставим это выражение во второе уравнение интерполяции:

$$y_i = a_i\left(\frac{x_{i+1} + x_{i-1}}{2}\right)^2 + b_i\left(\frac{x_{i+1} + x_{i-1}}{2}\right) + c_i .$$

Умножим это уравнение на 4 и сложим с остальными:

$$y_{i-1} + 4y_i + y_{i+1} = a_i[x_{i-1}^2 + (x_{i+1} + x_{i-1})^2 + x_{i+1}^2] + b_i[x_{i-1} + 2(x_{i+1} + x_{i-1}) + x_{i+1}] + 6c_i =$$
$$= 2a_i[x_{i-1}^2 + x_{i-1}x_{i+1} + x_{i+1}^2] + 3b_i[x_{i+1} + x_{i-1}] + 6c_i .$$

Последнее выражение в точности совпадает с выражением, стоящим в квадратных скобках формулы (5.1.1). Следовательно,

$$\int_{x_{i-1}}^{x_{i+1}} g_i(x)dx = \frac{x_{i+1} - x_{i-1}}{6}(y_{i-1} + 4y_i + y_{i+1}) = \frac{h}{3}(y_{i-1} + 4y_i + y_{i+1}) .$$

А значит,

$$\int_a^b f(x)dx \approx \frac{h}{3}(y_0 + 4y_1 + y_2 + y_2 + 4y_3 + y_4 + \ldots + y_{2n-2} + 4y_{2n-1} + y_{2n}) =$$

$$= \frac{h}{3}(y_0 + y_{2n} + 4(y_1 + y_3 + \ldots + y_{2n-1}) + 2(y_2 + y_4 + \ldots + y_{2n-2})) .$$

Таким образом, формула Симпсона имеет вид:

$$\int_a^b f(x)dx \approx \frac{h}{3}(y_0 + y_{2n} + 4(y_1 + y_3 + \ldots + y_{2n-1}) + 2(y_2 + y_4 + \ldots + y_{2n-2})) .$$

Оценка погрешности квадратурных формул

Оценим погрешность при использовании метода средних прямоугольников в предположении, что функция $f(x)$ бесконечно дифференцируема.

$$R = \int_a^b f(x)dx - h\sum_{i=0}^{n-1} f(\overline{x_i})$$

Разложим подынтегральную функцию $f(x)$ в ряд Тейлора в окрестности точки $\overline{x_i}$, $i = \overline{0, n-1}$.

$$f(x) = f(\overline{x_i}) + f'(\overline{x_i})(x - \overline{x_i}) + \frac{f''(\overline{x_i})}{2!}(x - \overline{x_i})^2 + \ldots$$

Тогда

$$\int_{x_i}^{x_{i+1}} f(x)dx = f(\overline{x_i})x\Big|_{x_i}^{x_{i+1}} + f'(\overline{x_i})\frac{(x-\overline{x_i})^2}{2}\Big|_{x_i}^{x_{i+1}} + \frac{f''(\overline{x_i})}{2}\cdot\frac{(x-\overline{x_i})^3}{3}\Big|_{x_i}^{x_{i+1}} + \ldots =$$

$$= f(\overline{x_i})h + f'(\overline{x_i})\left(\frac{h^2}{8} - \frac{h^2}{8}\right) + \frac{f''(\overline{x_i})}{2}\left(\frac{h^3}{24} + \frac{h^3}{24}\right) + \ldots =$$

$$= f(\overline{x_i})h + f''(\overline{x_i})\frac{h^3}{24} + \ldots$$

Последний ряд содержит лишь нечетные степени *x*. Тогда

$$R = \int_a^b f(x)dx - h\sum_{i=0}^{n-1} f(\overline{x_i}) = \sum_{i=0}^{n-1}\left(\int_{x_i}^{x_{i+1}} f(x)dx - f(\overline{x_i})h\right) = \sum_{i=0}^{n-1}\left(f''(\overline{x_i})\frac{h^3}{24} + \ldots\right)$$

При малой величине шага *h* основной вклад в погрешность *R* будет вносить величина $R_0 = \frac{h^3}{24}\sum_{i=0}^{n-1} f''(\overline{x_i})$, называемая главным членом погрешности *R*.

Применим метод средних прямоугольников к функции $f''(x)$ на отрезке $[a,b]$ с шагом *h*. Тогда

$$R_0 = \frac{h^2}{24}\cdot h\cdot\sum_{i=0}^{n-1} f''(\overline{x_i}) \approx \frac{h^2}{24}\int_a^b f''(x)dx.$$

Итак, $R \approx R_0 \approx \frac{h^2}{24}\int_a^b f''(x)dx = C\cdot h^2$, где $C = \frac{1}{24}\int_a^b f''(x)dx$ – постоянная величина. Погрешность в приближенном равенстве $R \approx C\cdot h^2$ есть величина бесконечно малая высшего порядка по сравнению с h^2 при $h \to 0$.

Степень шага h, которой пропорционален остаток R, называется порядком точности метода интегрирования. Метод средних прямоугольников имеет второй порядок точности.

Оценим погрешность при использовании метода трапеций также в предположении, что функция $f(x)$ бесконечно дифференцируема.

$$R = \int_a^b f(x)dx - h\sum_{i=0}^{n-1}\frac{f(x_i)+f(x_{i+1})}{2}$$

Разложим подынтегральную функцию в ряд Тейлора в окрестности точки x_i $(i=\overline{0,n-1})$.

$$f(x) = f(x_i) + f'(x_i)(x-x_i) + \frac{f''(x_i)}{2}(x-x_i)^2 + \ldots$$

$$\int_{x_i}^{x_{i+1}} f(x)dx = f(x_i)x\Big|_{x_i}^{x_{i+1}} + f'(x_i)\frac{(x-x_i)^2}{2}\Big|_{x_i}^{x_{i+1}} + \frac{f''(x_i)}{2}\cdot\frac{(x-x_i)^3}{3}\Big|_{x_i}^{x_{i+1}} + \ldots =$$

$$= f(x_i)h + f'(x_i)\frac{h^2}{2} + \frac{f''(x_i)}{2}\cdot\frac{h^3}{3} + \ldots$$

$$f(x_{i+1}) = f(x_i) + f'(x_i)(x_{i+1}-x_i) + \frac{f''(x_i)}{2}(x_{i+1}-x_i)^2 + \ldots =$$

$$= f(x_i) + f'(x_i)h + \frac{f''(x_i)}{2}h^2 + \ldots$$

$$h\frac{f(x_i)+f(x_{i+1})}{2} = f(x_i)h + f'(x_i)\frac{h^2}{2} + f''(x_i)\frac{h^3}{4} + \ldots$$

$$\int_{x_i}^{x_{i+1}} f(x)dx - h\frac{f(x_i)+f(x_{i+1})}{2} = -f''(x_i)\frac{h^3}{12} + \ldots$$

$$R = \int_a^b f(x)dx - h\sum_{i=0}^{n-1}\frac{f(x_i)+f(x_{i+1})}{2} = \sum_{i=0}^{n-1}\left(-f''(x_i)\frac{h^3}{12} + \ldots\right)$$

Главный член погрешности R:

$$R_0 = -\frac{h^3}{12}\sum_{i=0}^{n-1} f''(x_i).$$

Применяя метод левых прямоугольников к функции $f''(x)$ на отрезке $[a,b]$ с шагом h, получаем

$$R \approx R_0 = -\frac{h^2}{12}\cdot h\sum_{i=0}^{n-1} f''(x_i) = -\frac{h^2}{12}\int_a^b f''(x)dx = Ah^2,$$

где

$$A = -\frac{1}{12}\int_a^b f''(x)dx.$$

Итак, метод трапеций также имеет второй порядок точности.

Аналогично можно показать, что методы левых и правых прямоугольников имеют первый, метод Симпсона – четвертый порядок точности.

5.2. Правило Рунге практической оценки погрешности. Понятие об адаптивных алгоритмах. Особые случаи численного интегрирования. Метод ячеек. Вычисление кратных интегралов

Правило Рунге практической оценки погрешности

Пусть некоторый метод интегрирования имеет порядок точности k, то есть $R_h \approx A \cdot h^k$, где R_h – погрешность, A – коэффициент, зависящий от метода интегрирования и подынтегральной функции, h – шаг разбиения. Тогда

$$J = J_h + R_h \approx J_h + A \cdot h^k,$$

а при шаге $\frac{h}{2}$

$$J = J_{h/2} + R_{h/2} \approx J_{h/2} + A \cdot \frac{h^k}{2^k},$$

$$J_h + A \cdot h^k \approx J_{h/2} + A \cdot \frac{h^k}{2^k}$$

$$J_{h/2} - J_h \approx \frac{A \cdot h^k}{2^k}(2^k - 1)$$

$$J_{h/2} - J_h \approx R_{h/2}(2^k - 1)$$

$$R_{h/2} \approx \frac{J_{h/2} - J_h}{2^k - 1}$$

Выведенная формула называется **первой формулой Рунге**. Она имеет большое практическое значение. Если нужно вычислить интеграл с точностью ε, то мы должны вычислять приближенные значения интеграла, удваивая число элементарных отрезков, пока не добьемся выполнения неравенства:

$$|J_{h/2} - J_h| < \varepsilon \cdot (2^k - 1).$$

Тогда, пренебрегая бесконечно малыми величинами, можно считать, что

$$|J - J_{h/2}| = |R_{h/2}| < \varepsilon.$$

Если мы хотим получить более точное значение искомого интеграла, то за уточненное значение J мы можем принять вместо $J_{h/2}$ сумму

$$J \approx J_{h/2} + \frac{J_{h/2} - J_h}{2^k - 1}.$$

Это **вторая формула Рунге**. К сожалению, погрешность этого уточненного значения остается неопределенной, но обычно она на порядок выше, чем точность первоначального метода (когда за значение J мы принимаем $J_{h/2}$).

Для примера рассмотрим метод трапеций. Как было показано выше, порядок точности k этого метода равен 2.

$$J_h = \frac{1}{2} h \sum_{i=0}^{n-1} (f(x_i) + f(x_{i+1}))$$

$$J_{h/2} = \frac{1}{2} \cdot \frac{h}{2} \sum_{i=0}^{n-1} ((f(x_i) + f(\overline{x_i})) + (f(\overline{x_i}) + f(x_{i+1}))),$$

где $\overline{x_i} = x_i + \frac{h}{2}$. По второй формуле Рунге

$$J \approx J_{h/2} + \frac{J_{h/2} - J_h}{2^2 - 1} = \frac{4}{3} J_{h/2} - \frac{1}{3} J_h =$$

$$= \frac{1}{2} h \sum_{i=0}^{n-1} \left(\left(\frac{4}{6} f(x_i) + \frac{4}{6} f(\overline{x_i}) \right) + \left(\frac{4}{6} f(\overline{x_i}) + \frac{4}{6} f(x_{i+1}) \right) - \left(\frac{1}{3} f(x_i) + \frac{1}{3} f(x_{i+1}) \right) \right) =$$

$$= \frac{1}{2} h \sum_{i=0}^{n-1} \left(\frac{1}{3} f(x_i) + \frac{4}{3} f(\overline{x_i}) + \frac{1}{3} f(x_{i+1}) \right) = J^*_{h/2},$$

где $J^*_{h/2}$ есть приближенное значение интеграла, найденное методом Симпсона с шагом $\frac{h}{2}$. Так как порядок этого метода равен 4, то в данном примере применение второй формулы Рунге увеличило порядок точности на 2.

Понятие об адаптивных алгоритмах

Точность интегрирования зависит не только от шага дискретизации, но и характера поведения функции на отрезке интегрирования. А поэтому было бы неплохо иметь алгоритм, который приспосабливался бы к поведению функции на отрезке интегрирования. На практике часто встречаются случаи, когда подынтегральная функция ведет себя по-разному на различных участках отрезка интегрирования.

Суть адаптивного алгоритма заключается в следующем. Отрезок интегрирования первоначально делится на n равных частей. Затем каждый элементарный отрезок подвергается делению до тех пор, пока на этом отрезке не будет достигнута заданная точность.

Пусть для интеграла $\int_{x_{i-1}}^{x_i} f(x)dx$ получены два приближения: $J_i^{(1)}$ и $J_i^{(2)}$ – интегралы, вычисленные для двух разбиений отрезка $[x_{i-1}, x_i]$ при шаге h и $h/2$. Используя правило Рунге, проверим выполнимость неравенства $|J_i^{(1)} - J_i^{(2)}| < (2^k - 1)\varepsilon / n$. Если неравенство выполняется, то за значение интеграла по элементарному отрезку берем значение $J_i^{(2)}$. В противном случае увеличиваем число разбиений в два раза, находим новое приближение $J_i^{(3)}$. И так далее до тех пор, пока на каком-то этапе не будет выполнено неравенство:

$$|J_i^{(l)} - J_i^{(l+1)}| < (2^k - 1)\varepsilon / n. \qquad (5.2.1)$$

Подобный процесс можно провести для всех элементарных отрезков. Интеграл будет вычислен с точностью ε, так как для каждого элементарного интеграла J_i были выполнены неравенства (5.2.1).

Особые случаи численного интегрирования

К особым случаям численного интегрирования относятся следующие.

1. Несобственные интегралы, то есть хотя бы один из пределов интегрирования равен бесконечности или подынтегральная функция хотя бы в одной точке участка интегрирования обращается в бесконечность.

2. Подынтегральная функция терпит разрыв.

Рассмотрим случай разрывной функции. Если в точке $x = c$ имеет место разрыв первого рода, то есть левосторонний и правосторонний пределы существуют и конечны в точке разрыва, то

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

Если имеет место разрыв второго рода, то задача сводится к вычислению несобственного интеграла.

Общего метода численного нахождения несобственных интегралов не существует.

Рассмотрим случай, когда один из пределов интегрирования, например, верхний, равен бесконечности, то есть $\int_a^{\infty} f(x)dx$. Можно попытаться определить такое число B – верхний предел интегрирования, для которого $\left|\int_B^{\infty} f(x)dx\right| < \varepsilon$. Тогда

$$\int_a^{\infty} f(x)dx = \int_a^{B} f(x)dx + \int_B^{\infty} f(x)dx \approx \int_a^{B} f(x)dx.$$

Можно попытаться провести замену переменной так, чтобы после преобразований промежуток интегрирования стал конечным. Например, преобразование $x = \frac{a}{1-t}$ позволяет свести промежуток $[a,+\infty)$ к отрезку $[0,1]$. Но нужно следить за тем, чтобы при такой замене подынтегральная функция оставалась бы ограниченной.

Рассмотрим случай несобственного интеграла, когда подынтегральная функция обращается в бесконечность в некоторой точке интегрирования. Можно попытаться выделить особенность, то есть представить подынтегральную функцию в виде $f(x) = \varphi(x) + \psi(x)$ так, чтобы неограниченность была сосредоточена на функции $\psi(x)$, а несобственный интеграл от $\psi(x)$ можно было вычислить аналитически; функция $\varphi(x)$ ограничена, и к ней можно применить методы численного интегрирования. То есть

$$\int_a^{b} f(x)dx = \int_a^{b} \varphi(x)dx + \int_a^{b} \psi(x)dx,$$

причем первый интеграл вычисляем численно, второй – аналитически.

Метод ячеек. Вычисление кратных интегралов

Рассмотрим метод ячеек на примере двойного интеграла. Сделав очевидные обобщения, его можно распространить и на случай интегралов большей кратности. Рассмотрим интеграл $\iint_G f(x,y)dxdy$, где G – прямоугольник, $a \le x \le b$, $c \le y \le d$. Из курса математического анализа

известна теорема о среднем. Если подынтегральная функция $f(x,y)$ непрерывна и интегрируема, то существует такая точка $(\xi,\eta)\in G$, что $\iint\limits_G f(x,y)dxdy = f(\xi,\eta)\cdot S$, где S – площадь фигуры G. Если среднее значение функции заменить на значение функции в центре прямоугольника, то получим приближенную формулу:

$$\iint\limits_G f(x,y)dxdy \approx f\left(\frac{a+b}{2},\frac{c+d}{2}\right)\cdot(b-a)(d-c). \qquad (5.2.2)$$

Точность этой формулы можно повысить, если область G разбить на части (на элементарные ячейки) и к каждой ячейке применить формулу типа (5.2.2). То есть если областью интегрирования является прямоугольник, дискретизируем задачу.

$$\Delta x_i = x_i - x_{i-1},\ \Delta y_i = y_i - y_{i-1},$$
$$\iint\limits_{G_{ij}} f(x,y)dxdy \approx f(\bar{x}_i,\bar{y}_j)\Delta x_i \Delta y_j,$$
$$\bar{x}_i = \frac{x_{i-1}+x_i}{2},\ \bar{y} = \frac{y_{i-1}+y_i}{2}.$$

Просуммируем:

$$\iint\limits_G f(x,y)dxdy = \sum_{i=1}^{M}\sum_{j=1}^{N} f(\bar{x}_i,\bar{y}_j)\Delta x_i \Delta y_j.$$

Здесь M – количество элементарных отрезков по горизонтали, N – по вертикали.

В случае непрямоугольной области следует преобразовать независимые переменные так, чтобы область интегрирования стала прямоугольником. Пусть G – криволинейный четырехугольник.

Если сделать замену $x_1 = x,\ y_1 = \dfrac{y-\varphi_1(x)}{\varphi_2(x)-\varphi_1(x)}$, то область G перейдет в область $a \le x_1 \le b,\ 0 \le y_1 \le 1$.

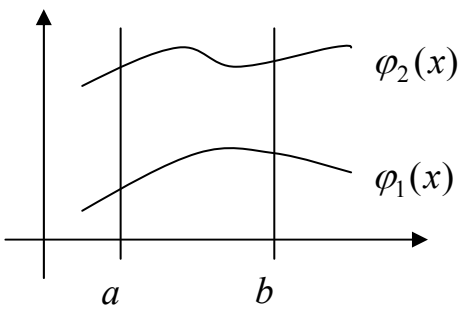


Рис. 5.2.1

Метод ячеек имеет второй порядок точности, как по направлению x, так и по направлению y. Поэтому сгущение сетки по обоим направлениям следует проводить одинаково, то есть удваивать количество элементарных отрезков нужно так, чтобы отношение $\frac{M}{N}$ оставалось постоянным.

5.3. Лабораторная работа «Численное интегрирование»

(4 часа)

Цель работы: научиться вычислять интегралы в программе MathCad численно и символически, научиться реализовывать в собственных программах формулы прямоугольников, трапеций и Симпсона для приближенного вычисления интегралов с заданной точностью.

Используемое программное обеспечение: Borland Pascal (или Delphi) и MathCad.

Основное задание:

1. Вычислите интеграл в программе MathCad с точностью 0,0001 (задания по вариантам смотрите в таблице 5.3.1). Вычислите также $\int ln(x)dx$.

2. Вычислите интеграл по формуле средних прямоугольников при $n = 10$.

3. Вычислите интеграл по формуле трапеций при $n = 8$ и $n = 16$.

4. Вычислите интеграл по формуле Симпсона с погрешностью 0,0001. Точность вычислений контролируйте правилом Рунге.

Таблица 5.3.1

Задания к лабораторной работе 4

(для тридцати вариантов)

№	Задание	№	Задание
1	*2*	*3*	*4*
1	$\int_{0,6}^{1,4} \frac{\sqrt{x^2+5}dx}{2x+\sqrt{x^2+0,5}}$	2	$\int_{0,4}^{1,2} \frac{\sqrt{0,5x+2}dx}{\sqrt{2x^2+1}+0,8}$
3	$\int_{0.8}^{1.8} \frac{\sqrt{0.8x^2+1}dx}{x+\sqrt{1.5x^2+2}}$	4	$\int_{1,0}^{2,2} \frac{\sqrt{1,5x+0,6}dx}{1,6+\sqrt{0,8x^2+2}}$
5	$\int_{1,2}^{2,0} \frac{\sqrt{2x^2+1,6}dx}{2x+\sqrt{0,5x^2+3}}$	6	$\int_{1,3}^{2,5} \frac{\sqrt{x^2+0,6}dx}{1,4+\sqrt{0,8x^2+1,3}}$
7	$\int_{1,2}^{2,6} \frac{\sqrt{0,4x+1,7}dx}{1,5x+\sqrt{x^2+1,3}}$	8	$\int_{0,8}^{1,6} \frac{\sqrt{0,3x^2+2,3}dx}{1,8+\sqrt{2x+1,6}}$
9	$\int_{1,2}^{2} \frac{\sqrt{0,6x+1,7}dx}{2,1x+\sqrt{0,7x^2+1}}$	10	$\int_{0,8}^{2,4} \frac{\sqrt{0,4x^2+1,5}dx}{2,5+\sqrt{2x+0,8}}$
11	$\int_{1.2}^{2.8} \frac{\sqrt{1.2x+0.7}dx}{1.4x+\sqrt{1.3x^2+0.5}}$	12	$\int_{0,6}^{2,4} \frac{\sqrt{1,1x^2+0,9}dx}{1,6+\sqrt{0,8x^2+1,4}}$
13	$\int_{0,7}^{2,1} \frac{\sqrt{0,6x+1,5}dx}{2x+\sqrt{x^2+3}}$	14	$\int_{0,8}^{2,4} \frac{\sqrt{1,5x+2,3}dx}{3+\sqrt{0,3x+1}}$
15	$\int_{1,9}^{2,6} \frac{\sqrt{2x+1,7}dx}{2,4+\sqrt{1,2x^2+0,6}}$	16	$\int_{0,5}^{1,9} \frac{\sqrt{0,7x^2+2,3}dx}{3,2+\sqrt{0,8x+1,4}}$
17	$\int_{1}^{2,6} \frac{\sqrt{0,4x+3}dx}{0,7x+\sqrt{2x^2+0,5}}$	18	$\int_{0,7}^{2,1} \frac{\sqrt{1,7x^2+0,5}dx}{1,4+\sqrt{1,2x+1,3}}$
19	$\int_{0,6}^{2,2} \frac{\sqrt{1,5x+1}dx}{1,2x+\sqrt{x^2+1,8}}$	20	$\int_{1,2}^{3} \frac{\sqrt{2x^2+0,7}dx}{1,5+\sqrt{0,8x+1}}$
21	$\int_{1,3}^{2,7} \frac{\sqrt{1,3x^2+0,8}dx}{1,7x+\sqrt{2x+0,5}}$	22	$\int_{0,6}^{1,4} \frac{\sqrt{x^2+0,5}dx}{2x+\sqrt{x^2+2,5}}$
23	$\int_{0,4}^{1,2} \frac{\sqrt{2x^2+1}dx}{0,8x+\sqrt{0,5x+2}}$	24	$\int_{0,8}^{1,8} \frac{\sqrt{1,5x^2+2}dx}{x+\sqrt{0,8x^2+1}}$

Окончание таблицы 5.3.1

1	*2*	*3*	*4*
25	$\int_{1}^{2,2} \frac{\sqrt{0,8x^2+2}dx}{1,6+\sqrt{1,5x+0,6}}$	26	$\int_{1,2}^{2,0} \frac{\sqrt{0,5x^2+3}dx}{2x+\sqrt{2x^2+1,6}}$
27	$\int_{1,3}^{2,5} \frac{\sqrt{0,8x^2+1,3}dx}{1,4+\sqrt{x^2+0,6}}$	28	$\int_{1,2}^{2,6} \frac{\sqrt{x^2+1,3}dx}{1,5x+\sqrt{0,4x+1,7}}$
29	$\int_{0,8}^{1,6} \frac{\sqrt{2x+1,6}dx}{1,8+\sqrt{0,3x^2+2,3}}$	30	$\int_{1,2}^{2} \frac{\sqrt{0,7x^2+1}dx}{2,1x+\sqrt{0,6x+1,7}}$

Комментарии к основному заданию

1. Напомним, что программа MathCad может проводить два типа вычислений: численные и символьные.

2. Заметим, что значение функции f в середине i-го отрезка удобно запрограммировать следующим образом: $f(a+(i+0.5)*h)$, где a – левый конец отрезка интегрирования, h – длина элементарного отрезка.

3. Заметим, что значение функции f в правом конце i-го отрезка удобно запрограммировать следующим образом: $f(a+i*h)$, где a – левый конец отрезка интегрирования, h – длина элементарного отрезка.

В заданиях 1 и 2 мы не контролируем точность вычислений, поэтому ответы могут отличаться от ответов программы MathCad.

4. Программирование формулы Симпсона

$$\int_a^b f(x)dx \approx \frac{h}{3}(y_0 + y_{2n} + 4(y_1 + y_3 + ... + y_{2n-1}) + 2(y_2 + y_4 + ... + y_{2n-2}))$$

Заключается, главным образом, в нахождении значений первой и второй скобок. Заметим, что число слагаемых в этих скобках различно, поэтому, на наш взгляд, целесообразно использовать разные циклы для их нахождения, например, так:

```
for i:=1 to 2*n-1 do if i mod 2 = 1 then s1:=s1+f(a+i*h);
for i:=2 to 2*n-2 do if i mod 2 = 0 then s2:=s2+f(a+i*h);
```

Заметим, что при обосновании правила Рунге пренебрегают бесконечно малыми слагаемыми более высокого порядка, чем h^k, где k – порядок точности метода. Поэтому есть небольшая вероятность того, что полученный ответ будет отличаться от истинного на величину, чуть большую заданной точности.

Дополнительное задание

1. Вычислите интеграл по формулам левых и правых прямоугольников.

2. Вычислите интеграл по формулам средних прямоугольников и трапеций с заданной точностью, контролируя точность вычислений правилом Рунге.

Комментарии к дополнительному заданию

1. Заметим, что структура программы будет такой же, что и для формулы средних прямоугольников.

2. Напомним, что порядок точности формул средних прямоугольников и трапеций равен 2, в отличие от формулы Симпсона, для которой он равен 4.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Бахвалов, Н. С. Численные методы в задачах и упражнениях [Текст] : учеб. пособие / Н. С. Бахвалов, А. В. Лапин, Е. В. Чижонков. – М. : Высшая школа, 2000. – 190 с.

2. Васильков, Ю. В. Компьютерные технологии вычислений в математическом моделировании [Текст] : учеб. пособие / Ю. В. Васильков, Н. Н. Василькова. – М. : Финансы и статистика, 2002. – 256 с.

3. Вержбицкий, В. М. Численные методы. Линейная алгебра и нелинейные уравнения [Текст] : учеб. пособие для вузов / В. М. Вержбицкий. – М. : Высшая школа, 2000. – 266 с.

4. Воробьева, Г. Н. Практикум по вычислительной математике [Текст] : пособие / Г. Н. Воробьева, А. Н. Данилова. – М. : Высшая школа, 1990. – 208 с.

5. Изаак, Д. Д. Вычислительная математика [Текст] : курс лекций / Д. Д. Изаак – Орск : Маркет-Сервис, 2009. – 92 с.

6. Измайлов, А. Ф. Численные методы оптимизации [Текст] : учеб. пособие / А. Ф. Измайлов, М. В. Солодов. – М. : Физматлит, 2003. – 304 с.

7. Лабораторный практикум по курсу: «Основы вычислительной математики» [Текст]. – М. : МЗ Пресс, 2001. – 192 с.

8. Очан, Ю. С. Математический анализ [Текст] : учеб. пособие для пед. институтов / Ю. С. Очан, В. Е. Шнейдер. – М. : Изд-во министерства просвещения РСФСР, 1961. – 880 с.

9. Иванов, В. Д. Лабораторный практикум по курсу «Основы вычислительной математики» [Текст] : практикум / В. Д. Иванов. – М. : МЗ-Пресс, 2001. – 189 с.

10. Мудров, А. Е. Численные методы для ПЭВМ на языках Бейсик, Фортран и Паскаль [Текст] : учеб. / А. Е. Мудров. – Томск : МП «Раско», 1991. – 272 с.

11. Чертежи сделаны с помощью авторской программы Изаака Д. Д. «Януш» версии 2.1.

Учебное издание

Дмитрий Давидович Изаак,

Анна Викторовна Швалева

ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА

Учебно-методическое пособие

Ведущий редактор
Е. В. Кондаева

Старший корректор
Е. А. Феонова

Ведущий инженер
Г. А. Чумак

Подписано в печать 26.12.2011 г.
Формат 60x84 1/16. Усл. печ. 5,9.
Тираж 50 экз. Заказ _____

**Издательство Орского гуманитарно-технологического института (филиала)
Федерального государственного бюджетного образовательного учреждения
высшего профессионального образования
«Оренбургский государственный университет»**

462403, г. Орск Оренбургской обл., пр. Мира, 15 А