



Министерство образования
Республики Беларусь

БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

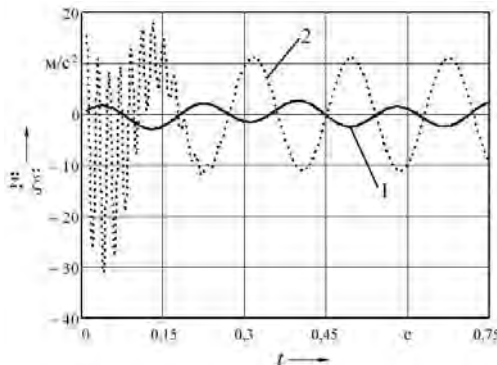
Кафедра «Тракторь»

Ю. Е. Атаманов
В. Н. Плищ
В. В. Равино

**МЕТОДЫ РЕШЕНИЯ ИНЖЕНЕРНЫХ ЗАДАЧ
ПРИМЕНИТЕЛЬНО К ОБЛАСТИ ТРАНСПОРТНОГО
И ТЯГОВОГО МАШИНОСТРОЕНИЯ**

Учебно-методическое пособие
к лабораторным работам по дисциплине «Информатика»

$$\left. \begin{aligned} M\ddot{x} + 2k_p(x - \xi) + 2c_p(z - \xi) &= 0; \\ m_n\ddot{\xi} - 2k_p(x - \xi) + 2k_{ш}(\xi - \varphi) - 2c_p(z - \xi) + 2c_{ш}(\xi - q) &= 0 \end{aligned} \right\}$$



Минск
БНТУ
2015

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
Белорусский национальный технический университет

Кафедра «Тракторы»

Ю. Е. Атаманов
В. Н. Плищ
В. В. Равино

**МЕТОДЫ РЕШЕНИЯ ИНЖЕНЕРНЫХ ЗАДАЧ
ПРИМЕНИТЕЛЬНО К ОБЛАСТИ ТРАНСПОРТНОГО
И ТЯГОВОГО МАШИНОСТРОЕНИЯ**

Учебно-методическое пособие
к лабораторным работам по дисциплине «Информатика»
для студентов специальностей 1-37 01 03 «Тракторостроение»,
1-37 01 04 «Многоцелевые гусеничные и колесные машины»
и 1-37 01 05 «Городской электрический транспорт»

*Рекомендовано учебно-методическим объединением
высших учебных заведений по образованию
в области транспорта и транспортной деятельности*

Минск
БНТУ
2015

УДК 004 (076.5)

ББК 32.97я7

A92

Р е ц е н з е н т ы:

доктор технических наук, заведующий кафедрой «Автомобили»

Белорусского национального технического университета

Ю. Д. Карпиевич;

кандидат технических наук, заместитель начальника

УКЭР-1 РУП «Минский тракторный завод» *В. Г. Ермалёнок*

Атаманов, Ю.Е.

A92 Методы решения инженерных задач применительно к области транспортного и тягового машиностроения : учебно-методическое пособие к лабораторным работам для студентов специальностей 1-37 01 03 «Тракторостроение», 1-37 01 04 «Многоцелевые гусеничные и колесные машины» и 1-37 01 05 «Городской электрический транспорт» / Ю. Е. Атаманов, В. Н. Плищ, В. В. Равино. – Минск : БНТУ, 2015. – 201 с.
ISBN 978-985-550-378-2.

Представлены основные методы численного решения математических задач с помощью компьютера. Описаны наиболее используемые методы численного решения уравнений (алгебраических, трансцендентных, дифференциальных) и их систем. Изложены методы приближения функций, способы приближенного дифференцирования и интегрирования.

УДК 004 (076.5)

ББК 32.97я7

ISBN 978-985-550-378-2

© Атаманов Ю. Е., Плищ В. Н.,
Равино В. В., 2015

© Белорусский национальный
технический университет, 2015

Содержание

Предисловие.....	4
Введение.....	6
Лабораторная работа № 1. Решение систем линейных уравнений.....	8
Лабораторная работа № 2. Решение нелинейных уравнений и их систем	27
Лабораторная работа № 3. Решение обыкновенных дифференциальных уравнений и их систем	72
Лабораторная работа № 4. Численное интегрирование и дифференцирование.....	119
Лабораторная работа № 5. Аппроксимация функций.....	142
Лабораторная работа № 6. Математическая обработка данных	160
Предметный указатель	198
Литература	200

Предисловие

Внедрение персональных вычислительных машин (ПЭВМ) во все сферы инженерной деятельности требует от инженеров-конструкторов овладения навыками использования вычислительной техники для решения различных инженерных задач. С этой целью большое внимание уделяется повышению уровня подготовки студентов вузов, которые уже с первого курса приобщаются к использованию ПЭВМ для решения различных математических задач, не говоря уже о том, что при выполнении курсовых работ, курсовых и дипломных проектов применение вычислительных машин в настоящее время стало нормой.

В связи с этим возникла необходимость создания учебного издания по дисциплине «Информатика», непосредственно связанного с применением вычислительной техники для решения инженерных задач. Отметим, что во втором семестре согласно учебной программе по названной дисциплине студенты как раз и обучаются использованию ПЭВМ при решении инженерных задач. В этом семестре изучаются методы построения и исследования численными методами формализованных технических задач, которые моделируют различные процессы, происходящие в мобильных машинах.

Что касается студентов нематематических специальностей, то для них главной задачей является понимание основных идей численных методов, их особенностей и областей применения. Следует отметить, что указанная категория читателей не обладает достаточными математическими знаниями для подробного исследования технических объектов численными методами. К тому же это не особенно необходимо студенту-нематематику, использующему математические методы как готовый инструмент в своей учебной работе.

В теоретической части предлагаемых лабораторных работ в сжатом виде приводятся основные необходимые сведения о математических методах решения технических задач. Изложение проводится на доступном для студентов указанных специальностей уровне. При необходимости напоминаются основные сведения по математическому методу из курса высшей математики. Для многих рассматриваемых методов приводятся блок-схемы, а также примеры решения задач с применением пакетов MathCAD и Excel и рекомендации по разработке программы на языке Delphi, способствующей

щие лучшему пониманию и усвоению материала. Пособие написано с учетом особенностей применения математических методов при решении технических задач с использованием ПЭВМ.

В список литературы включены некоторые издания по численным методам, которые авторы использовали в работе над пособием. В них читатель может найти более подробные сведения по интересующим его математическим методам. Разумеется, что это далеко не полный перечень литературы по численным методам и их применению. Издание будет полезно студентам и специалистам при первоначальном знакомстве с применением математических методов для решения технических задач. Оно может стать кратким справочником при выполнении расчетных учебных заданий.

Авторы искренне признательны рецензентам: кандидату технических наук, заместителю начальника УКЭР-1 МТЗ В. Г. Ермалёнку и заведующему кафедрой «Автомобили» БНТУ, доктору технических наук Ю. Д. Карпиевичу за ценные замечания по рукописи. Полезные предложения по улучшению содержания работы высказали преподаватели кафедры «Тракторы» БНТУ, прочитавшие рукопись или отдельные ее части. Всем им авторы выражают свою глубокую благодарность.

Компьютерная программа делает то, что вы приказали ей сделать, а не то, что вы хотели, чтобы она сделала.

Третий закон Гривы

Введение

Численные методы – это раздел математики, изучающий методы решения задач с применением компьютера. Название «Численные методы» связано с тем, что решение любой задачи находится в виде числа или нескольких чисел. Так как любое число представляется в компьютере в конечной форме, то численные методы – это методы *приближенного решения* математических задач. Они заключаются в выполнении конечного числа элементарных операций над числами. В численных методах числовая ось заменяется дискретной системой чисел. В результате численные методы сводят решение математических задач к вычислениям, которые могут быть выполнены как вручную, так и на компьютере.

Прикладная техническая задача обычно представляется в виде математической модели некоторого реального объекта (трактор, тягач, троллейбус и т. п.) или процесса (разгон, торможение, колебания и т. п.). Примерами типовых математических задач, решаемых в различных приложениях численными методами, являются решение алгебраических уравнений и их систем, обращение матриц, дифференцирование и интегрирование функций, решение обыкновенных дифференциальных уравнений и их систем и т. п.

Для одних и тех же технических задач существует множество математических методов их численного решения. Выбор того или иного метода зависит от требований к скорости получения решения и особенностей представления исходных данных (например, матрица может быть разреженной).

Развитие численных методов связано прежде всего с бурным развитием средств вычислительной техники, позволяющим расширить использование математических методов при проектировании и исследовании мобильных машин. Появилась возможность решать та-

кие задачи, которые ранее не могли быть решены вручную. Задачи, в которых используются различные функции (тригонометрические, логарифмические, экспоненциальные и др.), в настоящее время решаются на компьютере путем разложения функций в ряды.

Численные методы являются составной частью *вычислительной математики*, которая помимо численных методов выполняет анализ математических моделей и изучает вопросы взаимодействия человека с компьютером (в частности, вопросы программирования). Часто вычислительную математику рассматривают как науку, изучающую теорию численных методов и алгоритмов решения типовых математических задач на компьютере.

ца содержит mn элементов, расположенных в m строках и n столбцах, то она называется *прямоугольной матрицей*.

Используя понятие матрицы A , систему уравнений (1.1) можно записать в матричном виде:

$$AX = B,$$

где X и B – вектор-столбец неизвестных и вектор-столбец правых частей уравнений соответственно:

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix}; \quad B = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{bmatrix}.$$

Определителем (детерминантом) матрицы A n -го порядка называется число D ($\det A$):

$$D = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{vmatrix} = \sum (-1)^k a_{1\alpha} a_{2\beta} \dots a_{n\omega}.$$

Здесь $\alpha, \beta, \dots, \omega$ пробегает все возможные $n!$ перестановок номеров $1, 2, \dots, n$; k – число инверсий в данной перестановке.

Необходимым и достаточным условием существования системы линейных уравнений является условие $D \neq 0$. В случае равенства нулю определителя системы матрица называется *вырожденной*. При этом система линейных уравнений (1.1) либо не имеет решения, либо имеет их бесконечное множество.

О методах решения линейных систем

Методы решения систем линейных уравнений делятся на две группы – прямые и итерационные. *Прямые методы* для вычисления неизвестных используют конечные соотношения (формулы). Они дают решения после выполнения заранее известного числа опера-

ций. Эти методы сравнительно просты и наиболее универсальны, т. е. пригодны для решения широкого класса линейных систем.

Вместе с тем прямые методы имеют ряд недостатков. Существенным недостатком прямых методов является накапливание погрешностей в процессе решения, поскольку вычисления на любом этапе используют результаты предыдущих операций, что особенно опасно для больших систем. В связи с этим прямые методы используются для сравнительно небольших ($n < 200$) систем с плотно заполненной матрицей и неблизким к нулю определителем.

Отметим, что прямые методы решения линейных систем иногда называют *точными*, поскольку решение выражается в виде точных формул через коэффициенты системы.

Итерационные методы – это методы последовательных приближений. В них необходимо задать некоторое *начальное приближение*. После этого с помощью определенного алгоритма проводится один цикл вычислений, называемый *итерацией*. В результате итерации находят новое приближение. Итерации проводятся до получения решения с требуемой точностью. Алгоритмы решения линейных систем с использованием итерационных методов обычно более сложные по сравнению с прямыми методами.

Итерационные методы могут использоваться для уточнения решений, полученных с помощью прямых методов. Такие смешанные алгоритмы эффективны, как правило, для плохо обусловленных систем.

Формулы Крамера

Для системы линейных уравнений (1.1) вводятся обозначения:

$$D = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{vmatrix}; \quad D_1 = \begin{vmatrix} b_1 & a_{12} & \dots & a_{1n} \\ b_2 & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ b_n & a_{n2} & \dots & a_{nn} \end{vmatrix};$$

$$D_2 = \begin{vmatrix} a_{11} & b_1 & \dots & a_{1n} \\ a_{21} & b_2 & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & b_n & \dots & a_{nn} \end{vmatrix}; \dots; \quad D_n = \begin{vmatrix} a_{11} & a_{12} & \dots & b_1 \\ a_{21} & a_{22} & \dots & b_2 \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & b_n \end{vmatrix}.$$

Здесь D – определитель системы (1.1), а D_1, \dots, D_n – определители, получающиеся в результате замены столбца коэффициентов при соответствующем неизвестном столбцом свободных членов.

Если $D \neq 0$, то, как указывалось выше, система (1.1) является определенной, т. е. имеет единственное решение. Это решение находится по следующим формулам:

$$x_1 = \frac{D_1}{D}; \quad x_2 = \frac{D_2}{D}; \dots; \quad x_n = \frac{D_n}{D},$$

которые называются *формулами Крамера*.

Отметим, что при большом числе уравнений (больше трех) требуется выполнить большое число арифметических операций. Поэтому формулы Крамера используются лишь для решения систем, состоящих из нескольких уравнений.

Формулы Крамера применяются не только для решения систем линейных уравнений, но и для определения старших производных в системах дифференциальных уравнений, не решённых относительно старших производных.

Известен также метод решения линейных систем с использованием обратной матрицы. В этом случае система записывается в виде $AX = B$. Тогда, умножая обе части этого векторного на обратную матрицу A^{-1} , получаем $X = A^{-1}B$.

Прямой метод Гаусса

Наиболее распространенными среди прямых методов являются метод исключения Гаусса и его модификации. Метод основан на приведении матрицы системы к треугольному виду. Это достигается последовательным исключением неизвестных из уравнений системы (1.1).

Сначала с помощью первого уравнения из всех остальных уравнений системы исключается x_1 . Затем с помощью второго уравнения из третьего и всех последующих уравнений исключается x_2 . Этот процесс, называемый *прямым ходом метода Гаусса*, продолжается до тех пор, пока в левой части последнего n -го уравнения не останется лишь один член с неизвестным x_n , т. е. матрица системы (1.1) будет приведена к треугольному виду.

Обратный ход метода Гаусса состоит в последовательном вычислении искомых неизвестных. Решая последнее уравнение, нахо-

дят неизвестное x_n . Далее, используя это значение, из предыдущего уравнения вычисляют x_{n-1} и т. д. Последним определяется x_1 из первого уравнения.

На рис. 1.1 приведена блок-схема решения системы n линейных уравнений вида (1.1) методом Гаусса.

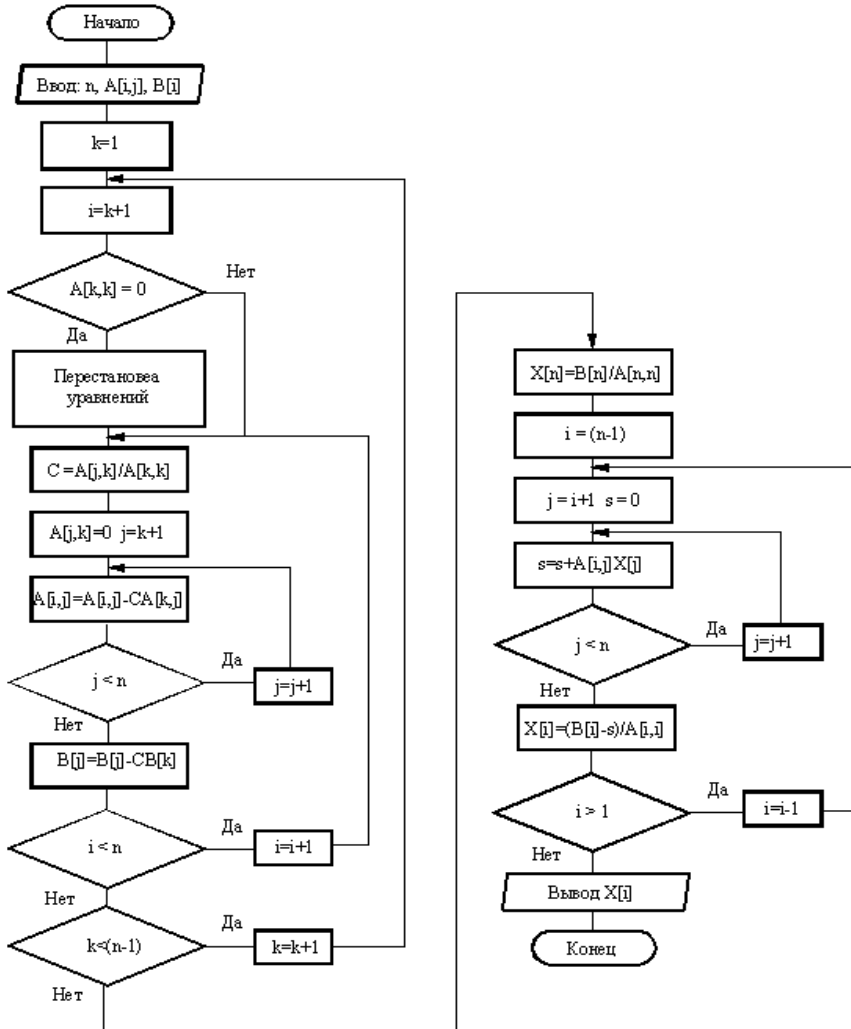


Рис. 1.1. Блок-схема метода Гаусса

Левая часть блок-схемы соответствует прямому ходу.

Поясним смысл индексов:

i – номер уравнения, из которого исключается неизвестное x_i ;

j – номер столбца;

k – номер неизвестного, которое исключается из остальных $n - k$ уравнений, а также номер того уравнения, с помощью которого исключается x_k .

Операция перестановки уравнений, т. е. перестановки соответствующих коэффициентов, служит для предотвращения деления на нулевой элемент. Правая часть блок-схемы описывает процесс обратного хода. Здесь

i – номер неизвестного, которое определяется из i -го уравнения;

$j = i + 1, i + 2, \dots$ – номера уже найденных неизвестных.

Одной из модификаций метода Гаусса является схема с выбором главного элемента, суть которой состоит в том, что требование неравенства нулю диагональных элементов a_{kk} , на которые происходит деление в процессе исключения, заменяется более жестким: из всех оставшихся в k -м столбце элементов нужно выбрать наибольший по модулю и переставить уравнения так, чтобы этот элемент оказался на месте a_{kk} .

Блок-схема алгоритма выбора главного элемента приведена на рис. 1.2. Она дополняет блок-схему метода Гаусса (см. рис. 1.1). На схеме введены новые индексы:

l – номер наибольшего по абсолютной величине элемента матрицы в столбце с номером k , т. е. среди элементов $a_{ik}, \dots, a_{km}, \dots, a_{kn}$;

m – текущий номер элемента, с которым происходит сравнение.

Заметим, что диагональные элементы матрицы называются *ведущими* элементами. Ведущий элемент a_{kk} – это коэффициент при k -м неизвестном в k -м уравнении на k -м шаге исключения.

Благодаря выбору наибольшего по модулю ведущего элемента уменьшаются множители, используемые для преобразования уравнений, что способствует снижению погрешности вычислений, поэтому метод Гаусса с выбором главного элемента обеспечивает приемлемую точность решения сравнительно небольшого числа ($n \leq 100$) уравнений, и только для плохо обусловленных систем решения, полученные этим методом, ненадежны.

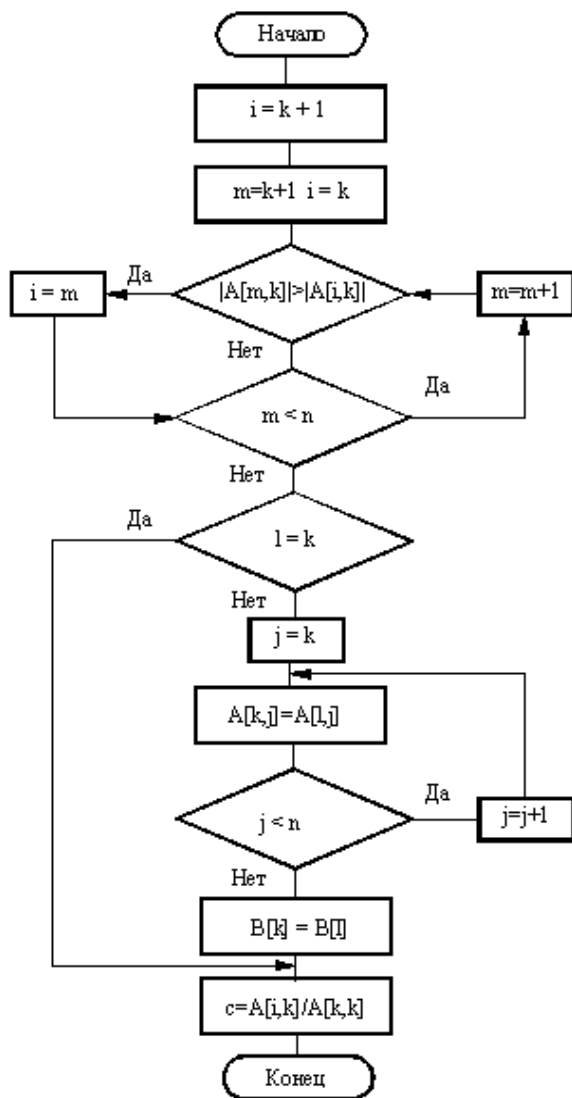


Рис. 1.2. Выбор главного элемента

Пример 1.1

Методом Гаусса решить линейную систему трех уравнений

$$\begin{cases} 10x_1 - 7x_2 = 7; \\ -3x_1 + 2x_2 + 6x_3 = 4; \\ 5x_1 - x_2 + 5x_3 = 6. \end{cases}$$

Решение

Исключим x_1 из второго и третьего уравнений. Для этого умножим первое уравнение на 0,3 и результат прибавим ко второму, а затем умножим первое же уравнение на $-0,5$ и результат прибавим к третьему уравнению. Получим

$$\begin{cases} 10x_1 - 7x_2 = 7; \\ -0,1x_2 + 6x_3 = 6,1; \\ 2,5x_2 + 5x_3 = 2,5. \end{cases}$$

Прежде чем исключать x_3 из третьего уравнения, заметим, что коэффициент при x_2 во втором уравнении (ведущий элемент) мал. Поэтому переставим второе и третье уравнения. Получим

$$\begin{cases} 10x_1 - 7x_2 = 7; \\ 2,5x_2 + 5x_3 = 2,5; \\ -0,1x_2 + 6x_3 = 6,1. \end{cases}$$

Теперь исключим x_2 из третьего уравнения, прибавив к нему второе, умноженное на 0,04. Получим систему в треугольном виде:

$$\begin{cases} 10x_1 - 7x_2 = 7; \\ 2,5x_2 + 5x_3 = 2,5; \\ 6,2x_3 = 6,2. \end{cases}$$

На этом заканчивается прямой ход метода Гаусса.

Обратный ход состоит в последовательном вычислении неизвестных x_3 , x_2 , x_1 соответственно из третьего, второго и первого уравнений. Выполним эти вычисления:

$$x_3 = \frac{6,2}{6,2} = 1; \quad x_2 = \frac{2,5 - 5x_3}{2,5} = -1; \quad x_1 = \frac{7x_2 + 7}{10} = 0.$$

Подставив найденные значения x_3 , x_2 , x_1 в исходную систему уравнений, убеждаемся, что $(0, -1, 1)$ и есть ее решение.

Ответ. Корнями исходной системы уравнений являются $x_1 = 0$, $x_2 = -1$ и $x_3 = 1$.

Для решения линейных уравнений методом Гаусса была разработана программа на языке Delphi в соответствии с алгоритмами, представленными на рис. 1.1 и 1.2. Программа рассчитана на решение системы десяти линейных уравнений с десятью неизвестными. Главная форма разработанной программы показана на рис. 1.3.

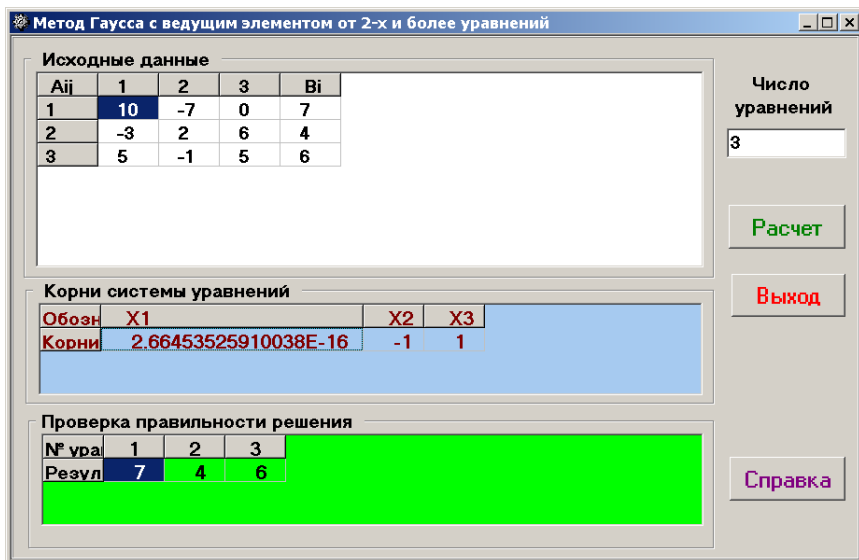


Рис. 1.3. Главная форма для решения линейных систем уравнений методом Гаусса

На главной форме находятся три компоненты *StrinGrid*. В верхней компоненте *StrinGrid* размещаются коэффициенты $A[i, j]$ перед неизвестными x_i . В последнем столбце этой компоненты помещены свободные члены $B[i]$. В средней компоненте *StrinGrid* размещаются результаты решения исходной системы линейных уравнений, т. е. корни уравнений. Найденные корни подставляются в исходные уравнения, и в результате получаем свободные члены этих уравнений, которые помещаются в нижнюю компоненту *StrinGrid*.

Рекомендация. Чтобы в компоненте *StrinGrid* для размещения результатов расчета можно было изменять размеры строк и столбцов, переставлять строки местами и редактировать ячейки, следует в свойстве *Options* установить значение *True* во флагах *goRowSizing*, *goRowMoving*, *goEditing*. Потребность в изменении ширины ячейки видна на рис. 1.3: когда первый корень выводится в экспоненциальной форме и не помещается в обычный размер ячейки, можно прочитать только 2,66 и принять это значение за корень x_1 .

Заполнение ячеек верхней компоненты *StrinGrid* происходит при создании главной формы. Исходные данные считываются из файла исходных данных. Ниже представлена эта подпрограмма:

```
procedure TForm1.FormCreate(Sender: TObject);
Var i,j : integer;
begin
  DecimalSeparator:=',';
  AssignFile(Fr,'A.Dat');
  Reset(Fr);
  readln(Fr,n);
  for i:=1 to n do
    for j:=1 to n do read(Fr,A[i,j]);
  for i:=1 to n do read(Fr,B[i]);
  CloseFile(Fr);
  Edit1.Text:=IntToStr(n);
with StringGrid1 do begin
  ColCount:=n+2;
  RowCount:=n+1;
  Cells[0,0]:=' Aij';
  for i:=1 to n do Cells[i,0]:=' '+IntToStr(i);
  Cells[n+1,0]:=' Bi';
```

```

for i:=1 to n do Cells[0,i]:=' '+IntToStr(i);
  for i:=1 to n do
    for j:=1 to n do Cells[i,j]:=' '+FloatToStr(A[j,i]);
      for i:=1 to n do Cells[n+1,i]:=' '+FloatToStr(B[i]);
        end;
    end;
  end;

with StringGrid2 do begin
  ColCount:=n+1;
  RowCount:=2;
  Cells[0,0]:='Обозн.';
  Cells[0,1]:='Корни';
  for i:=1 to n do Cells[i,0]:=' X'+IntToStr(i);
    end;

  with StringGrid3 do begin
    ColCount:=n+1;
    RowCount:=2;
    Cells[0,0]:='№ уравн.';
    Cells[0,1]:='Результат';
    for i:=1 to n do Cells[i,0]:=' '+IntToStr(i);
      end;
  end;
end;

```

В данной подпрограмме n – число уравнений в исходной системе линейных уравнений. Следует обратить внимание, что ввод данных осуществляется из текстового файла A.Dat, который надо создать заранее любым текстовым редактором (например, *Блокнотом*) и поместить в тот же каталог, где находится исполняемый файл программы. В первой строке файл A.Dat должен содержать количество уравнений, далее – коэффициенты уравнения, записанные через пробел, и последняя строка состоит из свободного коэффициента. В одной строке записываются коэффициенты одного уравнения. Например, для примера, приведенного на рис. 1.3, файл исходных данных A.Dat будет иметь следующую структуру:

3		
10	-7	0
-3	2	6
5	-1	5
7	4	6

Головная программа выглядит еще проще – она обращается к подпрограмме **Gaus** (решение линейных систем уравнений), выполняет проверку правильности полученных корней и размещает результаты в соответствующие компоненты **StrinGrid**. Текст головной программы приведен ниже:

```

procedure TForm1.BitBtn2Click(Sender: TObject);
Var i,j : integer; s : array[1..20] of real;
    C : array[1..20,1..20] of real;
begin
    Gaus(n,A,B,X);
    for i:=1 to n do s[i]:=0;
    for i:=1 to n do
    for j:=1 to n do C[i,j]:=0;
with StringGrid2 do
    for i:=1 to n do Cells[i,1]:= ' '+FloatToStr(X[i]);
    for i:=1 to n do
    for j:=1 to n do C[i,j]:=A[i,j]*X[j];
    for i:=1 to n do
    for j:=1 to n do begin
        s[i]:=s[i]+C[i,j];
        with StringGrid3 do
        Cells[i,1]:= ' '+FloatToStr(s[i]);
    end;
end;
end;

```

Таким образом, с помощью разработанной программы можно решать системы линейных уравнений до десяти уравнений включительно.

Решение линейных систем уравнений с помощью пакета MathCAD

Пакет MathCAD позволяет решать системы линейных уравнений без разработки программ. В этом случае применяется метод решения линейных систем уравнений с использованием обратной матрицы и система записывается в виде $AX = B$. Тогда, умножая обе части этого векторного выражения на обратную матрицу A^{-1} , получаем $X = A^{-1}B$.

Пример 1.2

Систему линейных уравнений, представленную в примере 1.1, решить с помощью пакета MathCAD.

Решение

Данную систему записываем в матричном виде:

$$AX = B,$$

где A – матрица коэффициентов при x_i ;

B – вектор-столбец свободных членов.

Тогда

$$X = A^{-1}B.$$

С использованием панели **Matrix** набираем матрицу коэффициентов при x_i размерностью 3×3 :

$$A := \begin{bmatrix} 10 & -7 & 0 \\ -3 & 2 & 6 \\ 5 & -1 & 5 \end{bmatrix}$$

и вектор-столбец свободных членов

$$B := \begin{bmatrix} 7 \\ 4 \\ 6 \end{bmatrix}.$$

Решим задачу двумя способами.

1-й способ. Ниже столбца B пишем формулу $x = A^{-1}B$ и еще ниже $x =$. Появляется столбец корней решаемой системы

$$x = \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix}.$$

Сказанное выше проиллюстрируем копией экрана (рис. 1.4).

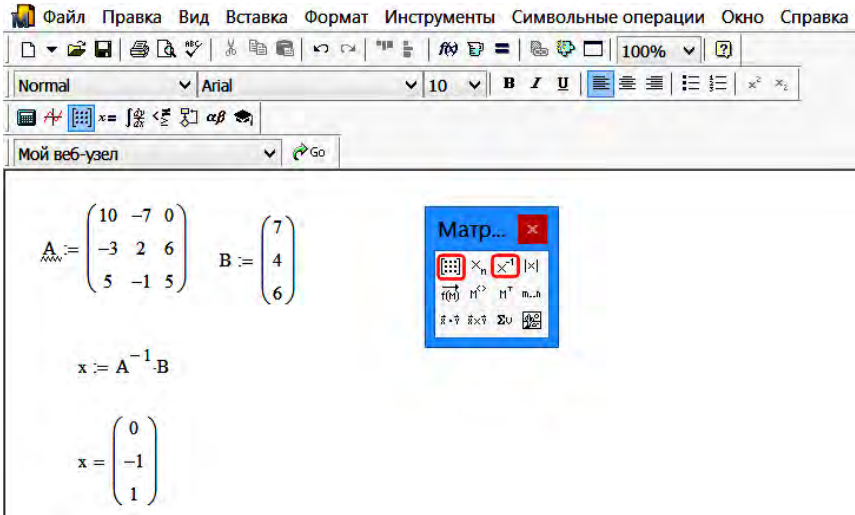


Рис. 1.4. Копия экрана при 1-м способе решения

Обратите внимание, что переменные, используемые в MathCAD, в отличие от Delphi чувствительны к регистру. т. е. x и X – это РАЗНЫЕ переменные.

2-й способ. Используем встроенную функцию

$$\text{lsolve}(A,B).$$

Ниже столбца свободных членов набираем

$$x1 := \text{lsolve}(A,B)$$

и ниже пишем $x1 =$. Пишем $x1$, чтобы MathCAD не «спутал» предыдущий x с новым. Появляется столбец корней решаемой системы уравнений:

$$x1 = \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix}.$$

Ниже на рис. 1.5 представлена копия экрана с использованием второго способа решения линейных систем уравнений.

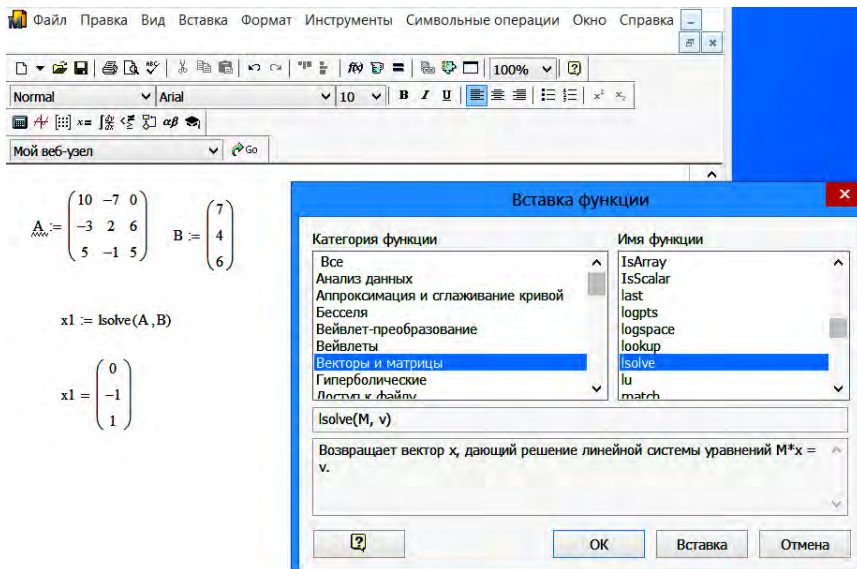


Рис. 1.5. Копия экрана при 2-м способе решения

Ответ. Корнями исходной системы уравнений являются $x_1 = 0$, $x_2 = -1$ и $x_3 = 1$.

Как видно из приведенных примеров, результаты решения заданной системы линейных уравнений полностью совпали независимо от того, с помощью какого пакета получено решение, что говорит о правильности программы, разработанной на языке Delphi.

Решение линейных систем уравнений с помощью пакета Excel

С помощью пакета Excel также можно решать системы линейных алгебраических уравнений. Как и в пакете MathCAD, в пакете Excel исходная система уравнений записывается в матричной форме. Для вычисления корней системы уравнений средствами Excel дополнительно понадобятся следующие функции:

МОБР (матрица) – вычисление обратной матрицы;

МУМНОЖ (матрица1; матрица2) – произведение матриц.

Следует иметь в виду, что эти функции возвращают блок ячеек, поэтому должны вводиться как табличные формулы **CTRL + SHIFT + ENTER**.

Пример 1.3

Решить систему уравнений

$$\begin{cases} x_1 - 0,092x_2 = 64,99; \\ x_2 + x_3 = -9,61; \\ 2,67x_2 + 4,02x_3 = 64,94. \end{cases}$$

Решение

Для удобства расчетов матрицу коэффициентов при неизвестных **A** и вектор-столбец свободных членов **B** следует написать отдельно, как показано на рис. 1.6.

11	Решение системы линейных алгебраических уравнений					
12						
13					Вектор-столбец	
14		Формирование матрицы A			свободных членов	
15			1,00	-0,09	0,00	64,99
16	A=		0,00	1,00	1,00	B= -9,61
17			0,00	2,67	4,02	64,94
18						

Рис. 1.6. Формирование исходных данных в пакете Excel

Присвоим матрице имя **A**, а вектору-столбцу – **B**. Для этого в главном меню выбираем пункт (вкладку) **Формулы**. В появившейся вкладке выбираем пункт **Диспетчер имен**. В пункте *Диспетчер имен* выбираем команду *Присвоить имя*. Появляется диалоговое

окно *Создание имени*, в строке которого *Имя* записывают необходимое имя, рис. 1.7.

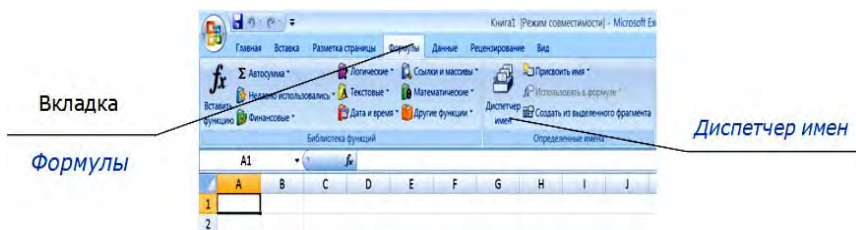


Рис. 1.7. Присвоение имен ячейкам

При этом следует выделить диапазон с B15 по D17 и в строке имен указать букву **A**, а выделив диапазон с F15 по A17 в строке имен, указать букву **B**, рис. 1.8.

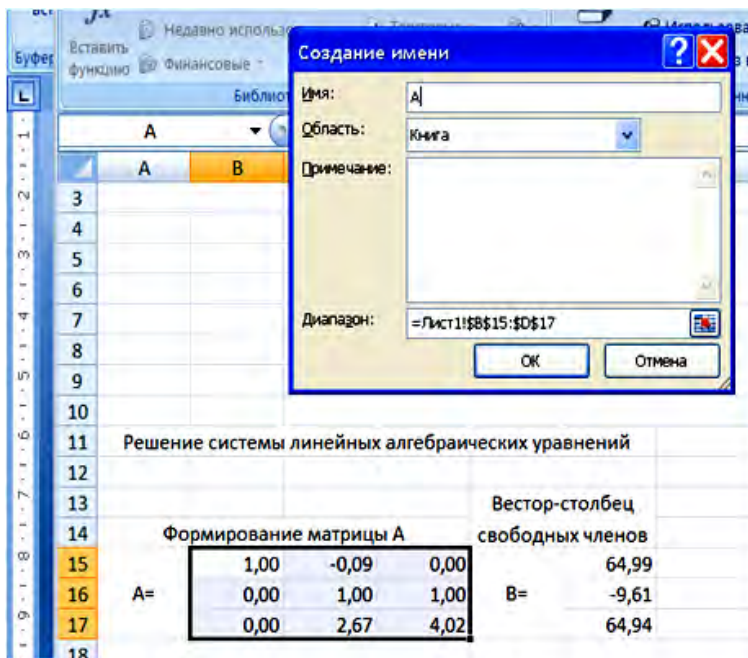


Рис. 1.8. Присвоение имени A матрице коэффициентов перед неизвестными

Чтобы получить решение, необходимо выделить диапазон ячеек, в которые будет помещен результат решения. Выделим ячейки с B20 по B22 и в строке формул запишем выражение

$$=МУМНОЖ(МОБР(A);B).$$

Записанное выражение не текст, а стандартные функции Excel. После этого необходимо нажать комбинацию клавиш

CTRL + SHIFT + ENTER,

и в выделенные ячейки будет выведено решение, рис. 1.9.

	A	B	C	D	E	F
11	Решение системы линейных алгебраических уравнений					
12						
13				Вектор-столбец		
14	Формирование матрицы A			свободных членов		
15		1,00	-0,09	0,00		64,99
16	A=	0,00	1,00	1,00	B=	-9,61
17		0,00	2,67	4,02		64,94
18						
19	Решение					
20		58,08519				
21		-76,7201				
22		67,11015				

Рис. 1.9. Результат решения исходной системы линейных уравнений

Ответ. Корнями исходной системы уравнений являются $x_1 = 58,085$, $x_2 = -76,720$ и $x_3 = 67,110$.

Таким образом, если требуется быстро решить систему линейных уравнений, а программа отсутствует, можно решить заданную систему уравнений с помощью пакетов MathCAD или Excel одним из рассмотренных выше способов.

Варианты заданий

Решить систему линейных уравнений с помощью разработанной программы на языке Delphi, пакетов MathCAD и Excel.

Но- мер вари- ри- анта	Система линейных уравнений	Но- мер вари- ри- анта	Система линейных уравнений
1	$\begin{cases} 0,34x_1 + 0,71x_2 + 0,63x_3 = 2,08 \\ 0,71x_1 - 0,65x_2 - 0,18x_3 = 0,17 \\ 1,17x_1 - 2,35x_2 + 0,75x_3 = 1,38 \end{cases}$	9	$\begin{cases} 0,43x_1 + 0,63x_2 + 1,44x_3 = 2,18 \\ 1,64x_1 - 0,83x_2 - 2,45x_3 = 1,84 \\ 0,58x_1 + 1,55x_2 + 0,18x_3 = 0,74 \end{cases}$
2	$\begin{cases} 3,75x_1 - 0,28x_2 + 0,17x_3 = 0,75 \\ 2,11x_1 - 0,11x_2 - 0,12x_3 = 1,11 \\ 1,22x_1 - 3,17x_2 + 1,81x_3 = 0,05 \end{cases}$	10	$\begin{cases} 0,62x_1 + 0,56x_2 - 0,43x_3 = 1,16 \\ 1,32x_1 - 0,88x_2 + 1,76x_3 = 2,07 \\ 0,73x_1 + 1,42x_2 - 0,34x_3 = 2,18 \end{cases}$
3	$\begin{cases} 3,01x_1 - 0,14x_2 - 0,15x_3 = 1,00 \\ 1,11x_1 + 0,13x_2 - 0,75x_3 = 0,13 \\ 0,17x_1 - 2,11x_2 + 0,71x_3 = 0,17 \end{cases}$	11	$\begin{cases} 0,21x_1 - 0,18x_2 + 0,75x_3 = 0,11 \\ 0,13x_1 + 0,75x_2 - 0,11x_3 = 2,00 \\ 3,01x_1 - 0,33x_2 + 0,11x_3 = 0,13 \end{cases}$
4	$\begin{cases} 1,24x_1 - 0,87x_2 - 3,17x_3 = 0,46 \\ 2,11x_1 - 0,45x_2 + 1,41x_3 = 1,50 \\ 0,48x_1 + 1,25x_2 - 0,63x_3 = 0,35 \end{cases}$	12	$\begin{cases} 0,13x_1 - 0,14x_2 - 2,00x_3 = 0,15 \\ 0,75x_1 + 0,18x_2 - 0,77x_3 = 0,11 \\ 0,28x_1 - 0,17x_2 + 0,39x_3 = 0,12 \end{cases}$
5	$\begin{cases} 0,32x_1 - 0,42x_2 + 0,85x_3 = 1,32 \\ 0,63x_1 - 1,41x_2 - 0,58x_3 = -0,44 \\ 0,84x_1 - 2,23x_2 - 0,52x_3 = 0,64 \end{cases}$	13	$\begin{cases} 0,92x_1 - 0,83x_2 + 0,62x_3 = 2,15 \\ 0,24x_1 - 0,54x_2 + 0,43x_3 = 0,62 \\ 0,73x_1 - 0,81x_2 + 0,67x_3 = 0,88 \end{cases}$
6	$\begin{cases} 0,62x_1 - 0,44x_2 - 0,86x_3 = 0,68 \\ 0,83x_1 + 0,42x_2 - 0,56x_3 = 1,24 \\ 0,58x_1 - 0,37x_2 - 0,62x_3 = 0,87 \end{cases}$	14	$\begin{cases} 0,64x_1 - 0,83x_2 + 4,20x_3 = 2,23 \\ 0,58x_1 - 0,83x_2 + 1,43x_3 = 1,71 \\ 0,86x_1 + 0,77x_2 + 0,88x_3 = -0,54 \end{cases}$
7	$\begin{cases} 0,46x_1 + 1,72x_2 + 2,53x_3 = 2,44 \\ 1,53x_1 - 2,32x_2 - 1,83x_3 = 2,83 \\ 0,75x_1 + 0,86x_2 + 3,72x_3 = 1,06 \end{cases}$	15	$\begin{cases} 0,73x_1 + 1,24x_2 - 0,38x_3 = 0,58 \\ 1,25x_1 + 0,66x_2 - 0,78x_3 = 0,66 \\ 0,75x_1 + 1,22x_2 - 0,83x_3 = 0,92 \end{cases}$
8	$\begin{cases} 4,24x_1 + 2,73x_2 - 1,55x_3 = 1,87 \\ 2,34x_1 + 1,27x_2 + 3,15x_3 = 2,16 \\ 3,05x_1 - 1,05x_2 - 0,63x_3 = -1,25 \end{cases}$	16	$\begin{cases} 1,26x_1 - 2,34x_2 + 1,17x_3 = 3,14 \\ 0,75x_1 + 1,24x_2 - 0,48x_3 = -1,17 \\ 3,44x_1 - 1,85x_2 + 1,16x_3 = 1,83 \end{cases}$

Но- мер вари- ри- анта	Система линейных уравнений	Но- мер вари- ри- анта	Система линейных уравнений
17	$\begin{cases} 2,47x_1 + 0,65x_2 - 1,88x_3 = 1,24 \\ 1,34x_1 + 1,17x_2 + 2,54x_3 = 2,35 \\ 0,86x_1 - 1,73x_2 - 1,08x_3 = 8,15 \end{cases}$	19	$\begin{cases} 1,24x_1 + 0,62x_2 - 0,95x_3 = 1,43 \\ 2,15x_1 - 1,18x_2 + 0,57x_3 = 2,43 \\ 1,72x_1 - 0,83x_2 + 1,57x_3 = 3,88 \end{cases}$
18	$\begin{cases} 0,43x_1 + 1,24x_2 - 0,58x_3 = 2,71 \\ 0,74x_1 + 0,83x_2 + 1,17x_3 = 1,26 \\ 1,43x_1 - 1,58x_2 + 0,83x_3 = 1,03 \end{cases}$	20	$\begin{cases} 1,06x_1 + 0,34x_2 + 1,26x_3 = 1,17 \\ 2,54x_1 - 1,16x_2 + 0,55x_3 = 2,23 \\ 1,34x_1 - 0,47x_2 - 0,83x_3 = 3,26 \end{cases}$

Лабораторная работа № 2

РЕШЕНИЕ НЕЛИНЕЙНЫХ УРАВНЕНИЙ И ИХ СИСТЕМ

Цель работы

1. Освоить методику решения нелинейных уравнений методом половинного деления и методом Ньютона (касательных), а также систем нелинейных уравнений методом Ньютона.
2. Получить навыки в программировании этих методов на языке Delphi.
3. Решить заданные нелинейные уравнения и их системы с помощью пакетов Excel и MathCAD.

Постановка задачи

Одной из наиболее часто встречающихся в инженерной практике задач является решение нелинейных уравнений и их систем. Нелинейные уравнения получаются при расчете собственных частот колебательных систем (частотные уравнения), устойчивости движения мобильных машин, автоматических систем управления (характеристические уравнения) и т. п.

В общем виде уравнение с одним неизвестным записывается так:

$$f(x) = 0, \quad (2.1)$$

где $f(x)$ – некоторая функция от x .

Решая уравнение (2.1), находят такие значения переменной x , при которых функция $f(x)$ равна нулю. Такие значения переменной x называются *корнями уравнения* (2.1).

В общем виде задача нахождения точного решения уравнения (2.1) неразрешима. Даже для алгебраического уравнения выше четвертой степени не существует формулы для нахождения его корней, использующей конечное число алгебраических операций. Поэтому на практике широко распространены численные методы, позволяющие получить приближенное решение уравнения (2.1) с любой наперед заданной точностью ε .

Нахождение приближенного значения корней уравнения (2.1) обычно состоит из двух этапов: отделения корней и уточнения значения корня.

Отделение корней – определение числового интервала, который содержит **один и только один корень** уравнения (2.1). Границы этого интервала можно рассматривать как первое приближенное значение искомого корня (нижняя граница интервала – с недостатком, верхняя – с избытком).

Уточнение значения корня – это процедура сужения границ найденного интервала математическим методом **с целью определения приближенного значения корня уравнения (2.1)** с любой наперед заданной точностью ε .

Ниже рассмотрим некоторые широко известные методы отделения корней.

Отделение корней уравнения

Одним из методов отделения корней уравнения (2.1) является *графический метод*, который заключается в том, что либо непосредственно строится график функции $y = f(x)$, либо функцию $f(x)$ представляют в виде двух и более простых функций, т. е.

$$f(x) = f_1(x) + f_2(x).$$

В первом случае находят пересечение графиком функции $y = f(x)$ оси абсцисс, во втором – точки пересечения функций $y_1 = f_1(x)$ и $y_2 = -f_2(x)$.

Чтобы воспользоваться методами отделения корней нелинейных уравнений, необходимо иметь критерий, позволяющий убедиться в том, что на рассматриваемом интервале $[a, b]$:

1) имеется корень уравнения (2.1);

2) этот корень на данном интервале является единственным.

Следующее утверждение подтверждает такой критерий.

Если на отрезке $[a, b]$ функция $f(x)$ непрерывна и монотонна, а ее значения на концах интервала имеют разные знаки, то на рассматриваемом интервале $[a, b]$ существует один и только один корень уравнения (2.1).

Очевидно, что если непрерывная на отрезке $[a, b]$ функция $f(x)$ монотонна и на концах отрезка ее значения имеют одинаковые знаки, **то на этом отрезке нет ни одного корня уравнения (2.1).**

Таким образом, чтобы отделить все действительные корни уравнения (2.1), необходимо найти все интервалы (их число равно степени уравнения), на которых функция $f(x)$ монотонна и знаки функции на концах каждого такого интервала различны.

Признаком монотонности функции $f(x)$ на некотором интервале является неизменность знака производной от этой функции. Следовательно, для поиска интервалов, содержащих корни уравнения (2.1), необходимо найти такие значения переменной x , для которых производная $f'(x)$ не меняет знак.

Однако может оказаться, что на некотором отрезке $[a, b]$ функция $f(x)$ не меняет знак, но в некоторой точке x_0 обращается в нуль. Тогда производная либо равна нулю, т. е. $f'(x_0) = 0$, либо она не существует. Поэтому задача отделения корней уравнения (2.1) сводится к задаче отыскания точек, в которых производная $f'(x)$ обращается в нуль или имеет разрыв, и к последующему определению знака функции $f(x)$ в найденных точках.

Пример 2.1

Графическим методом отделить корни уравнения

$$x^2 - \sin x = 0.$$

Решение

Так как исходное уравнение – второй степени, то оно имеет два корня, т. е. его график должен два раза пересекать ось абсцисс.

В данном случае удобно построить отдельные графики функций $y_1 = x^2$ и $y_2 = \sin x$ с помощью пакетов Excel или MathCAD на интервале от 0 до 2π (период функции синуса), рис. 2.1.

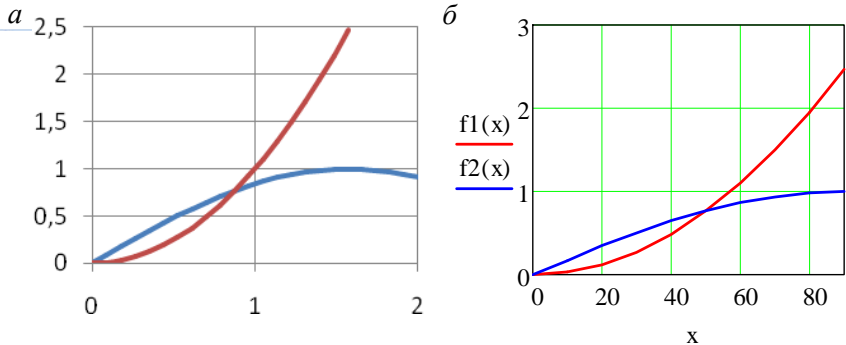


Рис. 2.1. Графики функций $y_1 = x^2$ и $y_2 = \sin x$:
а – построенные с помощью пакета Excel;
б – построенные с помощью пакета MathCAD

На рис. 2.1 аргумент изменялся от 0 до $\pi/2$.

Ответ. Из рис. 2.1 видно, что один из корней определяется точно: $x_1 = 0$, а для второго найден отрезок, содержащий только один корень: $[40^\circ, 60^\circ]$ или примерно $\left[\frac{\pi}{5}, \frac{\pi}{3}\right]$.

Для определения интервала, содержащего все корни алгебраического уравнения, рекомендуется использовать *метод Лагранжа*. В общем виде алгебраическое уравнение с одним неизвестным имеет вид

$$f(x) = a_0 x^n + a_1 x^{n-1} + a_2 x^{n-2} + a_3 x^{n-3} + \dots + a_{n-1} x + a_n = 0, \quad (2.2)$$

где $a_0 > 0$ или $a_0 = 1$;

a_i – коэффициенты многочлена (2.2), некоторые из них могут быть равны нулю.

Для алгебраического уравнения (2.2) с действительными коэффициентами верхняя граница положительных корней $R_{\text{в}}^+$ определяется по формуле Лагранжа

$$R_{\text{в}}^+ = 1 + \sqrt[k]{B/a_0}, \quad (2.3)$$

где $k \geq 1$ – номер (индекс) первого из отрицательных коэффициентов уравнения (2.2), учитывая и коэффициенты, равные нулю;

$B = |-a|_{\text{max}}$ – наибольшее из абсолютных значений отрицательных коэффициентов уравнения (2.2).

Для отыскания нижней границы отрицательных корней в уравнении (2.2) следует x заменить на $-x$ и определить величину R_2 , используя формулу Лагранжа для вспомогательного уравнения:

$$f_2(-x) = 0, \quad R_2 = 1 + \sqrt[k_2]{B_2/a_{02}}, \quad R_{\text{н}}^- = -R_2,$$

где $k_2 \geq 1$ – номер (индекс) первого из отрицательных коэффициентов вспомогательного уравнения $f_2(-x) = 0$, учитывая и коэффициенты, равные нулю;

$B_2 = |-a|_{\text{max}}$ – наибольшее из абсолютных значений отрицательных коэффициентов вспомогательного уравнения $f_2(-x) = 0$.

Следовательно, все вещественные корни алгебраического уравнения (2.2) находятся на отрезке $[a, b]$, где $a = R_{\text{н}}^-$, $b = R_{\text{в}}^+$.

Вычислять значения полинома (2.2) на отрезке $[a, b]$ рекомендуется по схеме Горнера, которая, в общем-то, предназначена для деления исходного многочлена (2.2) на линейный двучлен $(x - c)$. В результате получается многочлен степени, меньшей на единицу по сравнению с исходным многочленом и чаще всего остаток r , т. е.

$$P(x) = (x - c)g(x) + r,$$

где

$$P(x) = a_0x^n + a_1x^{n-1} + a_2x^{n-2} + a_3x^{n-3} + \dots + a_{n-1}x + a_n;$$

$$g(x) = b_0x^{n-1} + b_1x^{n-2} + b_2x^{n-3} + b_3x^{n-4} + \dots + b_{n-2}x + b_{n-1}.$$

Особенность состоит в том, что остаток r численно равен многочлену (2.2), если в него вместо x подставить c , которое следует изменять от нижней границы a до верхней границы b предварительно определенного отрезка $[a, b]$, содержащего все вещественные корни уравнения (2.2). Коэффициенты b_i вычисляются по схеме Горнера, т. е. по коэффициентам исходного многочлена $P(x)$ коэффициенты многочлена $g(x)$ определяются с использованием следующих выражений:

$$\begin{aligned}
 b_0 &= a_0; \\
 b_1 &= b_0c + a_1; \\
 b_2 &= b_1c + a_2; \\
 &\dots\dots\dots \\
 b_{n-1} &= b_{n-2}c + a_n.
 \end{aligned}
 \tag{2.3}$$

Причем остаток r численно равен последнему коэффициенту многочлена $g(x)$, т. е. $r = b_{n-1}$.

Пример 2.2

Методом Лагранжа определить границы вещественных корней уравнения $x^3 - 2x + 1 = 0$ и отделить его корни.

Решение

Перепишем исходное уравнение таким образом, чтобы в нем были представлены все члены, включая члены, содержащие и нулевые коэффициенты:

$$f(x) = x^3 + 0x^2 - 2x + 1 = 0,$$

где $a_0 = 1 > 0$; $a_1 = 0$; $a_2 = -2$; $a_3 = 1$.

Определим верхнюю границу положительных корней, воспользовавшись формулой Лагранжа. Здесь $k = 2$ (индекс первого из отрицательных коэффициентов $a_2 = -2$); $B = 2$ (наибольший по модулю отрицательный коэффициент $a_2 = -2$); $a_0 = 1$.

$$R_B^+ = 1 + \sqrt[k]{B/a_0} = 1 + \sqrt[2]{2/1} = 2,41.$$

Построим вспомогательное уравнение $f_2(-x) = 0$:

$$f_2(-x) = 1(-x)^3 + 0(-x)^2 - 2(-x) + 1 = -1x^3 + 0x^2 + 2x + 1,$$

где $a_0 = -1 > 0$; $a_1 = 0$; $a_2 = 2$; $a_3 = 1$.

Чтобы первый член вспомогательного уравнения был положительным, умножим обе его части на -1 , получим

$$f_2(-x) = x^3 - 0x^2 - 2x - 1 = 0.$$

Определим нижнюю границу отрицательных корней, воспользовавшись вспомогательным уравнением. Здесь $k = 1$ (индекс первого из отрицательных коэффициентов $a_1 = 0$); $B = 2$ (наибольший по модулю отрицательный коэффициент $a_2 = -2$); $a_0 = 1$.

$$R_2 = 1 + \sqrt[k]{B / a_0} = 1 + \sqrt[2]{2 / 1} = 2,41, \quad R_2^- = -R_2 = -2,41.$$

Ответ. Таким образом, для исходного уравнения $x^3 - 2x + 1 = 0$ отрезком исследования, содержащим все вещественные корни, является интервал $[-2,41; 2,41]$.

Пример 2.3

На заданном интервале отделить корни уравнения $x^3 - 2x + 1 = 0$.

Решение

При построении графика функции нижнюю и верхнюю границы найденного отрезка исследования рекомендуется округлять так, чтобы последние цифры границ оканчивались на 0, 2 или 5. Для отделения корней с помощью пакетов Excel и MathCAD воспользуемся графическим методом, изменяя аргумент от $-2,5$ до $2,5$ с шагом 0,5. Результаты представлены на рис. 2.2.

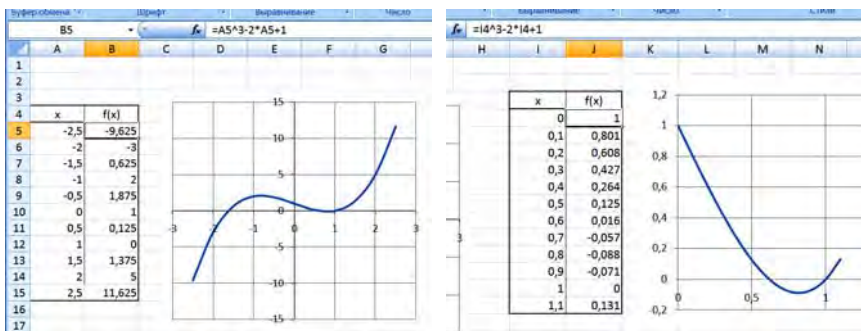


Рис. 2.2. Результаты построения графиков исходного многочлена

Так как на первом варианте графика (рис. 2.2, левый график) пересечения графика оси абсцисс на интервале от 0 до +1 не видны, был построен дополнительный график на этом же интервале с шагом 0,1 (рис. 2.2, правый график).

Выполним эти же действия в пакете MathCAD. Однако, учитывая расчеты, выполненные в пакете Excel, сразу же возьмем интервал от $-2,5$ до $+2,5$ и шаг изменения аргумента примем равным 0,1. Результаты расчета представлены на рис. 2.3, и они полностью совпадают с расчетами, сделанными в пакете Excel.

$x := -2.5, -2.5 + 0.1 .. 2.5$ (Задание пределов изменения аргумента)

$f(x) := x^3 - 2 \cdot x + 1$ (Запись исходного уравнения)

$f(x) =$

-9.625
-8.024
-6.567
-5.248
-4.061
-3
-2.059
-1.232
-0.513
0.104
0.625
...

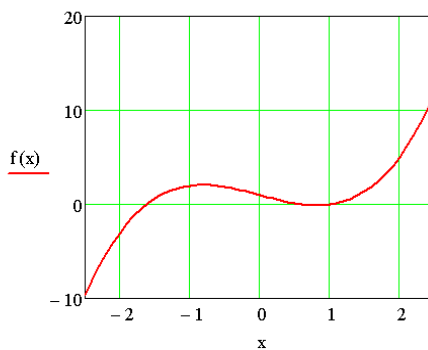


Рис. 2.3. Результаты расчета в пакете MathCAD

Выполним отделение корней заданного уравнения с помощью программы на языке Delphi, в которой для определения значения многочлена используем схему Горнера, представленную выражениями (2.3). Для программы используем три формы, рис. 2.4. На первой форме расположим исходные данные, на второй форме – таблицу результатов расчета, на третьей – график. Кроме того, используем глобальный модуль (Global), в котором разместим глобальные переменные и все подпрограммы.

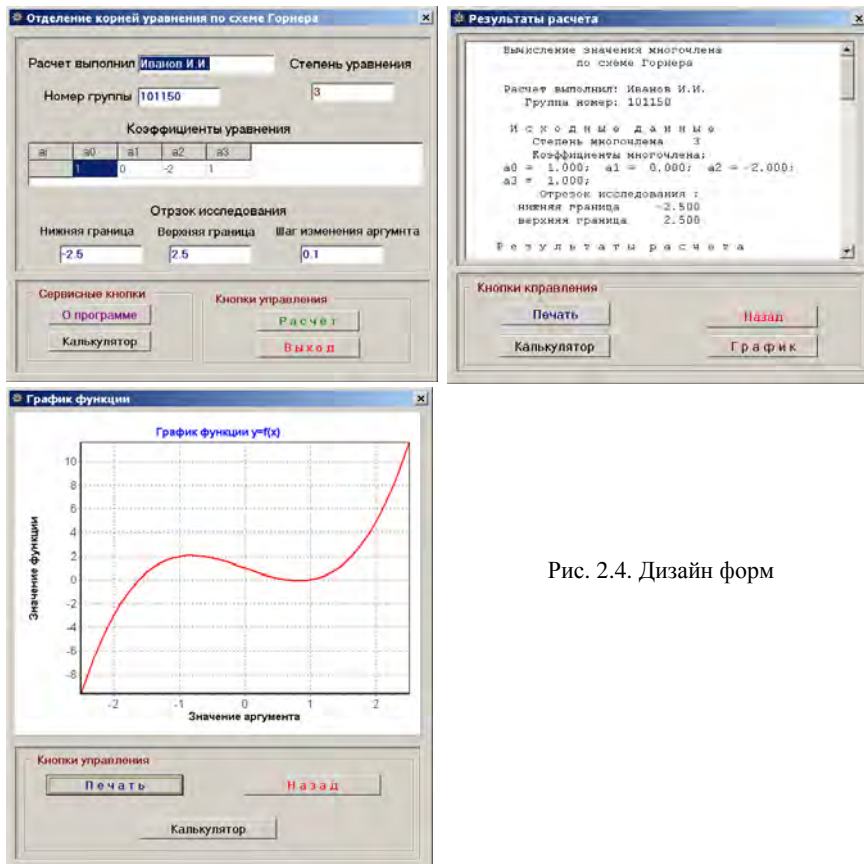


Рис. 2.4. Дизайн форм

В табл. 2.1 представлены результаты расчета по разработанной программе. В выделенных фрагментах результатов показаны отрезки, в которых функция меняет знак, т. е. содержат только один корень.

Таблица 2.1

Результаты расчета

Вычисление значения		
многочлена по схеме Горнера	0.800	2.088
Расчет выполнил: Иванов И.И.	0.700	2.057
Группа номер: 101150	-0.600	1.984
И с х о д н ы е д а н н ы е	-0.500	1.875
Степень многочлена 3	-0.400	1.736
Коэффициенты многочлена:	-0.300	1.573
a0 = 1.000; a1 = 0.000; a2 = -2.000;	-0.200	1.392
a3 = 1.000;	-0.100	1.199
Отрезок исследования:	0.000	1.000
нижняя граница -2.500	0.100	0.801
верхняя граница 2.500	0.200	0.608
Результаты расчета	0.300	0.427
-----	0.400	0.264
C : f(x)	0.500	0.125
-----	0.600	0.016
-2.500 -9.625	0.700	-0.057
-2.400 -8.024	0.800	-0.088
-2.300 -6.567	0.900	-0.071
-2.200 -5.248	1.000	0.000
-2.100 -4.061	1.100	0.131
-2.000 -3.000	1.200	0.328
-1.900 -2.059	1.300	0.597
-1.800 -1.232	1.400	0.944
-1.700 -0.513	1.500	1.375
-1.600 0.104	1.600	1.896
-1.500 0.625	1.700	2.513
-1.400 1.056	1.800	3.232
-1.300 1.403	1.900	4.059
-1.200 1.672	2.000	5.000
-1.100 1.869	2.100	6.061
-1.000 2.000	2.200	7.248
-0.900 2.071	2.300	8.567
	2.400	10.024
	2.500	11.625
	--- *** Конец расчета *** ---	

Ответ. Из таблиц результатов расчета и графиков получены следующие три интервала, содержащие только один вещественный корень уравнения: $[-2; -1,5]$, $[0,6; 0,7]$ и $[0,9; 1,1]$.

Дадим некоторые полезные рекомендации по разработке программы на языке Delphi для отделения корней заданного алгебраического уравнения с использованием схемы Горнера (предварительно должны быть определены нижняя и верхняя границы отрезка исследования).

1. Так как одномерный массив A коэффициентов исходного уравнения передается в подпрограмму, назовем ее **Gorner**, в качестве параметра, то массив следует описывать как в разделе **Type**, так и в разделе **Var** в **interface** части модуля **Global**. Чтобы программа получилась более универсальной, зарезервируем массив A большего размера, чем этого требуется для размещения коэффициентов уравнения 3-й степени, например, при помощи нашей программы дадим возможность отделять корни алгебраического уравнения вплоть до 10-й степени без изменения программы. Такой фрагмент программы показан ниже наряду с описанием других используемых переменных.

```
unit Global;
```

```
interface
```

```
  Type Mas=array[0..10] of real;
```

```
  Var FIO,nGr : string; n : integer; Fr,Fw,Fww : TextFile;
```

```
  A : Mas; Rn,Rv,h,c,b : real;
```

В фрагменте программы, так же как и в тексте, приняты обозначения:

R_n , R_v – нижняя и верхняя границы исследования;

c – переменная, принимающая численные значения от нижней границы R_n до верхней границы R_v с шагом h .

2. Для коэффициентов b_i полинома $g(x)$ можно не отводить одномерный массив. Внимательно посмотрев на выражения (2.3), увидим, что коэффициенты b_i фактически вычисляются по одной и той же формуле независимо от их индекса. Исключение составляет коэффициент b_0 . Тогда подпрограмму **Gorner** можно представить в следующем виде:

```

procedure Gorner(c:real;A:Mas;Var b:real);
var i : integer;
begin
  b:=A[0];
  for i:=1 to n do b:=b*c+A[i];
end;

```

В результате работы подпрограммы *Gorner* в вызывающую программу возвращается значение последнего коэффициента b_i , т. е. численное значение исходного полинома при подстановке в него вместо x значения c , которое записывается как в файл результатов расчета, так и в файл для построения графика.

3. Так как разрабатывается универсальная программа, коэффициенты исходного уравнения удобнее размещать не в окнах, например, *Edit*, а в ячейках компоненты *StringGrid*, которая автоматически изменяет свой объем в зависимости от степени уравнения n . Фрагмент головной программы (*Unit1*) при размещении коэффициентов a_i исходного уравнения в ячейках компоненты *StringGrid* приведен ниже.

```

with StringGrid1 do begin
  ColCount:=n+2; //Число колонок на 2 больше степени уравнения
  RowCount:=2; //Число строк 2
  Cells[0,0]:=' ai';
  for i:=1 to n+2 do Cells[i,0]:=' a'+IntToStr(i-1);
  for i:=0 to n+1 do Cells[i+1,1]:=FloatToStr(A[i]);
end;

```

Результаты работы этого фрагмента видны на головной форме, см. рис. 2.4.

4. В случае неправильной записи в файле исходных данных ошибки обнаружатся при просмотре исходных данных на головной форме. Чтобы можно было исправить некоторые переменные, рекомендуется считывать исходные данные с головной формы после исправления и записывать их в тот же файл исходных данных. Такие фрагменты программ представлены ниже:

считывание с данных из окон *Edit* и из ячеек компоненты *StringGrid* головной формы

```

FIO:=Edit1.Text;
nGr:=Edit2.Text;
n:=StrToInt(Edit3.Text);
Rn:=StrToFloat(Edit4.Text);
Rv:=StrToFloat(Edit5.Text);
h:=StrToFloat(Edit6.Text);
with StringGrid1 do
  for i:=1 to n+1 do A[i-1]:=StrToFloat(Cells[i,1]);

```

запись в файл исходных данных

```

AssignFile(Fw,'A.Dat');
Rewrite(Fw);
  writeln(Fw,FIO);
  writeln(Fw,nGr);
  writeln(Fw,n);
  writeln(Fw,Rn:10:3,Rv:10:3,h:10:3);
  for i:=0 to n do write(Fw,A[i]:8:3); writeln(Fw);
CloseFile(Fw);

```

5. Решаемое трансцендентное уравнение рекомендуется помещать в подпрограмму *Fun* и обращаться к ней в цикле как при табулировании любой функции: от нижней границы R_n исследуемого отрезка до верхней границы R_v с выбранным шагом h . Так, подпрограмма *Fun* для рассмотренного выше трансцендентного уравнения $x^2 - \sin x = 0$ может иметь вид

```

Procedure Fun(x:real;Var y:real);
begin
  y:=x*x-sin(x);
end;

```

Значение переменной x передается в подпрограмму в качестве исходного данного. Результат расчета y возвращается в вызывающую программу. Обе переменные можно описать как глобальные в модуле *Global*.

Таким образом, рассмотренные методы позволяют отделить корни нелинейных уравнений любой степени.

Варианты заданий

Отделить корни уравнений с помощью разработанной программы, пакетов MathCAD и Excel.

Номер варианта	У р а в н е н и я	
	Алгебраическое	Трансцендентное
1	$x^4 + 2x^3 - x - 1 = 0$	$2, 2x - 2^x = 0$
2	$x^3 - x + 1 = 0$	$2^x - 2x^2 - 1 = 0$
3	$4x^{2,1} - 5x - 6 = 0$	$e^x - \frac{1}{x} - 1 = 0$
4	$x^4 + 5x - 3 = 0$	$2^x - 4x = 0$
5	$x^3 + 2x - 4 = 0$	$x^2 - \cos \pi x = 0$
6	$x^3 + x - 3 = 0$	$\ln x - \frac{1}{x^2} = 0$
7	$x^3 - 12x - 8 = 0$	$2 \ln x - \frac{x}{2} + 1 = 0$
8	$x^3 - 2x^2 + x + 1 = 0$	$x^2 - \sin \pi x = 0$
9	$x^3 + 4x - 3 = 0$	$x^2 - 20 \sin x = 0$
10	$x^3 + 8x - 6 = 0$	$\cos 2x - \frac{1}{x} + 5,06 = 0$
11	$3x^4 + 5x - 3 = 0$	$x - \cos x = 0$
12	$x^5 - 5x + 2 = 0$	$1,8x^2 - \sin 10x = 0$
13	$x^3 + 6x - 5 = 0$	$\ln x - \frac{7}{2x + 6} = 0$
14	$x^4 + x - 3 = 0$	$2x \ln x - 1 = 0$
15	$x^3 + 10x - 9 = 0$	$\ln x + (x + 1)^3 = 0$
16	$x^4 + 4x^3 - 8x^2 - 17 = 0$	$\sqrt{x + 1} = \frac{1}{x}$
17	$x^3 - 2x + 7 = 0$	$x - \cos x = 0$
18	$x^3 + 4x^2 - 6 = 0$	$3x - \cos x - 1 = 0$
19	$x^4 - 2x - 1 = 0$	$x + \ln x = 0,5$
20	$x^3 - 6x + 9x - 3 = 0$	$\operatorname{tg}(0,55x + 0,1) = x^2$

Уточнение корней уравнения методом половинного деления

Метод половинного деления – это один из простейших методов уточнения (нахождения) корней нелинейных уравнений, суть которого состоит в следующем.

Пусть имеем отрезок $[a_0, b_0]$, содержащий только один корень исходного уравнения. В качестве первого приближения корня принимается середина этого отрезка, т. е. $x_{cp} = (a_0 + b_0) / 2$.

Далее вычисляем значение функции $F(a_0)$ на левом конце отрезка и $F(x_{cp})$ в середине отрезка. Сравниваем знаки функций $F(a_0)$ и $F(x_{cp})$. Если знаки одинаковы, то на отрезке $[a_0, x_{cp}]$ корень не содержится. Этот отрезок отбрасываем, т. е. сужаем отрезок исследования, принимая $a_1 = x_{cp}$. В противном случае (знаки разные) отрезок $[a_0, x_{cp}]$ содержит корень, а отрезок $[x_{cp}, b_0]$ отбрасываем, т. е. сужаем отрезок исследования, принимая $b_1 = x_{cp}$. В результате первого шага (итерации) отрезок исследования уменьшился вдвое до $[a_1, b_1]$.

Затем за исходный отрезок исследования принимается отрезок $[a_1, b_1]$ и весь процесс расчета повторяется по описанному алгоритму. Следовательно, после каждой итерации отрезок исследования, на котором находится корень уравнения, уменьшается вдвое, т. е. после n итераций отрезок сокращается в 2^n раз. Отсюда этот метод и получил своё название.

Итерационный процесс продолжается до тех пор, пока значение функции $F(x_{cp})$ после n -й итерации не станет меньшим по модулю некоторого наперед заданного малого числа ε (точности решения), т. е. $|F(x_{cp})| < \varepsilon$.

Для проверки знаков функций $F(a)$ и $F(x_{cp})$ используется их произведение

$$p = F(a) F(x_{cp}).$$

Если произведение больше нуля ($p > 0$), отрезок не содержит корня и отбрасывается. В противном случае ($p < 0$) отрезок содержит корень и используется в следующей итерации. Ниже приводится фрагмент программы, поясняющий сказанное:

```

Procedure MPD(a,b:real; Var xsr:real);
Var p : real; i : integer;
begin
  i:=1; //счётчик
  repeat
    xsr:=(a+b)/2;
    Fun(a,fa); //вычисление f(a)
    Fun(xsr,fxsr); //вычисление f(xср)
    P:=fa*fxsr;
    if (P>0) then a:=xsr
      else b:=xsr;
  writeln(Fw,"2,i:3,xsr:13:3,fxsr:10:4);
  writeln(Fww,"2,i:3,xsr:13:3,fxsr:10:4);
  i:=i+1;
  until (abs(fxsr)<=0.001);
end;

Procedure Fun(x:real; Var y:real);
begin
  y:=x*x*x-0.2*x*x+0.4*x-1.4;
end;

```

На рис. 2.5 показана графическая интерпретация метода половинного деления, а на рис. 2.6 – блок-схема этого метода.

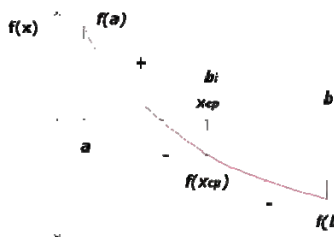


Рис. 2.5. Метод деления отрезка пополам

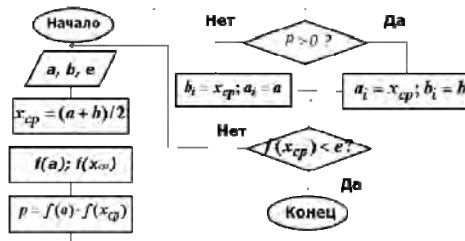


Рис. 2.6. Блок-схема метода деления отрезка пополам

Метод половинного деления довольно медленный, однако он всегда сходится, т. е. при его использовании решение получается всегда, причем с заданной точностью (естественно, в рамках разрядности ПЭВМ). Обычно требуется большее число итераций по

сравнению с некоторыми другими методами не является препятствием к применению этого метода.

Для сравнения по скорости сходимости метода половинного деления с другими методами решения нелинейных уравнений рекомендуется ввести в программу счетчик, который считал бы число итераций до получения решения.

Решаемое уравнение, как и ранее, поместить в подпрограмму *Fun* (см. фрагмент программы).

Варианты заданий

Разработать программу на языке Delphi для нахождения корней нелинейных уравнений и с ее помощью найти корни уравнений в соответствии со своим вариантом (см. варианты заданий) с заданной точностью.

Уточнение корней уравнения методом касательных

Отличие этого метода от предыдущего состоит в том, что на k -й итерации проводится касательная к кривой $y = f(x)$ при $x = x_k$ и ищется точка пересечения касательной и осью абсцисс. При этом не обязательно задавать отрезок $[a, b]$, содержащий корень исходного уравнения, а достаточно найти некоторое начальное приближение корня $x = x_0$, рис. 2.7.

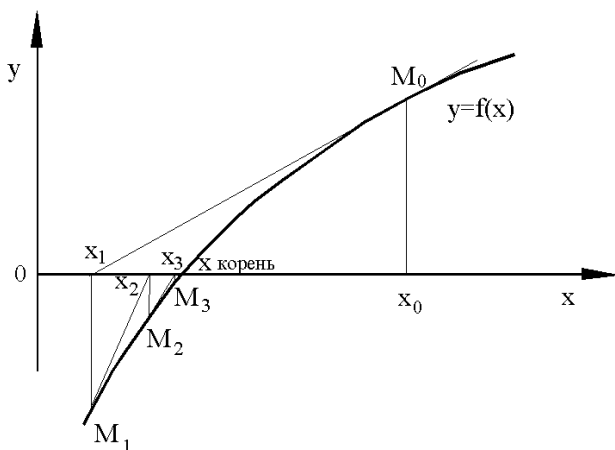


Рис. 2.7. Метод касательных

Блок-схема метода касательных представлена на рис. 2.8.

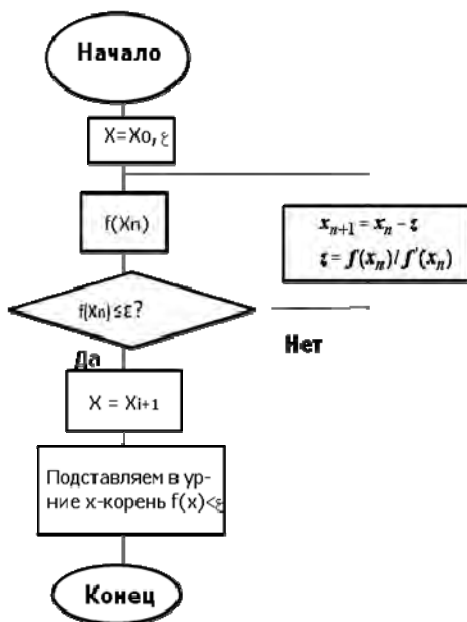


Рис. 2.8. Блок-схема метода касательных

Уравнение касательной, проведенной к кривой $y = f(x)$ в точке M_0 с координатами x_0 и $f(x_0)$, имеет вид

$$y - f(x_0) = f'(x_0)(x - x_0).$$

Отсюда находится следующее приближение корня x_1 как абсцисса точки пересечения касательной с осью x ($y = 0$):

$$x_1 = x_0 - f(x_0) / f'(x_0).$$

Аналогично находятся и следующие приближения как точки пересечения с осью абсцисс касательных, проведенных в точках M_1 , M_2 и т. д. Формула для $n + 1$ -го приближения имеет вид

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}. \quad (2.4)$$

При этом необходимо, чтобы производная $f'(x)$ не равнялась нулю. Для окончания расчета может быть использовано условие или $|f(x_n)| < \varepsilon$, или близости двух последовательных приближений: $|x_{n+1} - x_n| < \varepsilon$. Несмотря на больший, чем в методе половинного деления, объем вычислений, сходимость метода касательных значительно выше.

Рассмотрим некоторые вопросы, связанные со сходимостью метода касательных (Ньютона) и его применением. Известна следующая теорема:

Пусть $x = x_k$ – корень исходного уравнения, т. е. $f(x) = 0$, а $f'(x) \neq 0$ и производная $f'(x)$ непрерывна. Тогда существует окрестность D корня x_k ($x_k \in D$) такая, что если начальное приближение корня x_k принадлежит этой окрестности, то для метода Ньютона последовательность значений $\{x_n\}$ сходится к x_k при $n \rightarrow \infty$. При этом для погрешности корня $\varepsilon = x_n - x$ имеет место соотношение

$$\lim_{n \rightarrow \infty} \frac{\varepsilon_{n+1}}{\varepsilon_n^2} = \frac{f''(x_k)}{2f'(x_k)}.$$

Это означает, что на каждой итерации погрешность возводится в квадрат, т. е. число верных знаков корня удваивается. Если

$$\frac{f''(x_k)}{2f'(x_k)} \approx 1,$$

то при $|\varepsilon_n| < 0,5$ после пяти-шести итераций погрешность станет величиной порядка 2^{-64} . Это наименьшее значение погрешности при вычислениях на современных ПЭВМ даже с удвоенной точностью. Отметим, что для получения такой точности методом половинного деления потребовалось бы более 50 итераций. Трудность в применении метода касательных состоит в выборе начального значения корня x_0 , которое должно находиться в окрестности D (область сходимости). Кроме того, отметим, что начальную точку x_0 следует выбирать так, чтобы в этой точке знаки функции $f(x)$ и второй производной $f''(x)$ совпадали.

Ниже приведены фрагменты головной программы и подпрограммы **Fun**, для решения методом Ньютона уравнения $x^3 - 0,2x^2 + 0,4x - 1,4 = 0$.

```

Procedure Newton(x0:real;Var xk:real;Var i:integer);
Var z,y : real;
begin
  x:=x0; i:=1; //начальные значения корня и счётчика

  repeat
    Fun(x,z,y);
    writeln(Fw,":6,i:5,x:16:3,y:15:5);
    writeln(Fww,":2,i:5,x:10:3,y:10:5);
    x:=x-z;
    xk:=x;
    i:=i+1;
  until (abs(y)<e) or (i>20);
end;

Procedure Fun(x:real;Var z,y:real);
Var y1 : real;
begin
  y:= x*x*x-0.2*x*x+0.4*x-1.4; //уравнение
  y1:=3*x*x-0.4*x+0.4; //производная
  z:=y/y1;
end;

```

В переменной z (см. подпрограмму Fun) вычисляется приращение x .

Учитывая сказанное о выборе начальной точки x_0 , иногда целесообразно использовать смешанный алгоритм. Он состоит в том, что сначала применяется всегда сходящийся метод половинного деления, а затем после нескольких итераций – быстро сходящийся метод Ньютона.

Для сравнения по скорости сходимости методов решения нелинейных уравнений Ньютона и половинного деления рекомендуется ввести в программу счетчик, который считал бы число итераций до получения решения, решаемое уравнение и производную, как и ранее, поместить в подпрограмму **Fun**.

Варианты заданий

Разработать программу на языке Delphi для нахождения корней нелинейных уравнений методом Ньютона и с ее помощью найти корни уравнений в соответствии со своим вариантом (см. варианты заданий) с заданной точностью.

О решении алгебраических нелинейных уравнений

Действительные корни

Рассмотренные выше методы решения нелинейных уравнений пригодны для решения как алгебраических, так и трансцендентных уравнений. Вместе с тем при нахождении корней алгебраических уравнений приходится сталкиваться с некоторыми особенностями. С другой стороны, по сравнению с трансцендентными уравнениями алгебраические уравнения имеют то преимущество, что заранее известно число их корней.

Напомним известные из курса алгебры некоторые свойства алгебраических уравнений вида (2.2).

1. Уравнение степени n имеет всего n корней, среди которых могут быть действительные и комплексные.

2. Комплексные корни образуют комплексно-сопряженные пары, т. е. каждому корню $x = a + bi$ соответствует корень $x = a - bi$.

3. Число положительных действительных корней меньше или равно числу перемен знаков в последовательности коэффициентов a_0, a_1, \dots, a_n . Заменяя в уравнении (2.2) x на $-x$, аналогично можно оценить число отрицательных корней.

Одним из способов решения уравнения (2.2) является *метод понижения порядка*. Он состоит в том, что после нахождения какого-либо корня $x = x_{k1}$ исходное уравнение делят на двучлен $x - x_{k1}$, понижая порядок уравнения до $n - 1$. Однако, применяя этот способ, следует помнить о точности решения, поскольку даже небольшая погрешность в определении первого корня может привести к накоплению погрешности в дальнейших вычислениях.

Рассмотрим применение метода Ньютона к решению уравнения (2.2). В соответствии с формулой (2.5) итерационный процесс для нахождения корня уравнения (2.2) имеет вид

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

$$f(x) = a_0 + a_1x + \dots + a_nx^n;$$

$$f'(x) = a_1 + 2a_2x + \dots + na_nx^{n-1}.$$

Для значений многочленов $f(x)$ и $f'(x)$ в точке $x = x_n$ может быть использован метод Горнера.

Естественно, при использовании метода Ньютона должны выполняться условия сходимости. При их соблюдении в результате решения уравнения получается значение того корня, который находится вблизи начального значения x_0 .

Заметим, что для уменьшения погрешности сначала лучше находить меньшие по модулю корни уравнения и сразу удалять их из него, приводя уравнение к меньшей степени. Поэтому, если нет информации о величинах корней, в качестве начальных приближений рекомендуется принимать числа $0, \pm 1$ и т. д.

Комплексные корни

При использовании ПЭВМ, как правило, имеется возможность работать с комплексными числами, поэтому изложенные методы могут быть применены и для нахождения комплексных корней уравнений. Если в качестве начального приближения x_0 взять комплексное число, то последующие приближения и окончательное значение корня могут оказаться комплексными.

Комплексные корни попарно сопряженные, и при их исключении порядок уравнения уменьшается на 2, поскольку оно делится сразу на квадратный трехчлен, т. е.

$$f(x) = (x^2 + px + q)(b_nx^{n-1} + \dots + b_2) + b_1x + b_0. \quad (2.5)$$

Линейный остаток $b_1x + b_0$ равен нулю, если p и q выразить с помощью найденных корней, т. е.

$$p = -2c; \quad q = c^2 + d^2; \quad x = c \pm id.$$

Решение

Набираем полином, выделяем переменную x и выбираем команды Symbolic-Variable-Solve. Решение (копия экрана) показано на рис. 2.9.

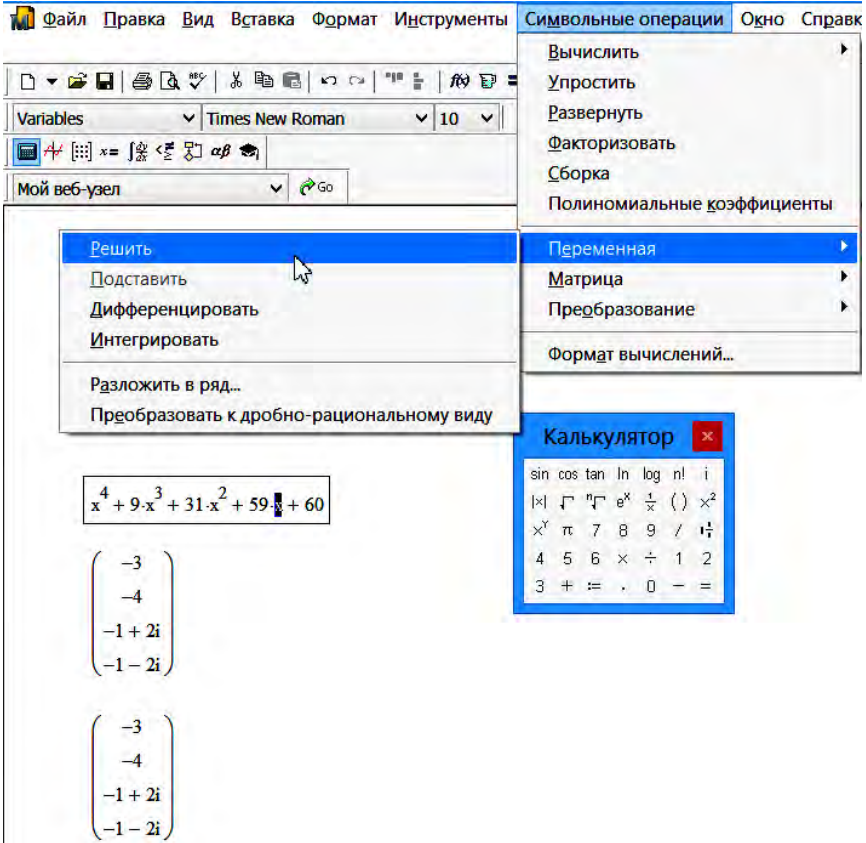


Рис. 2.9. Определение корней полинома

Ответ. Определены корни заданного полинома: два вещественные $x_1 = -3$, $x_2 = -4$ и два комплексных корня $x_3 = -1 + 2i$, $x_4 = -1 - 2i$.

Факторизация выражения

Выделить выражение любым способом и выбрать команду *Symbolic-Factor*.

Пример 2.5

Разложить предыдущий полином на множители.

Решение

Набираем полином и выбираем команды *Symbolic-Factor*. Результат представлен на рис. 2.10 (копия экрана).



$$x^4 + 9x^3 + 31x^2 + 59x + 60 \text{ factor} \rightarrow (x + 3) \cdot (x + 4) \cdot (x^2 + 2x + 5)$$

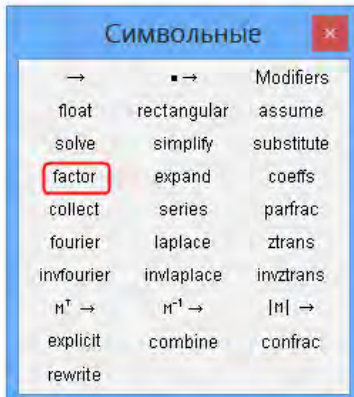


Рис. 2.10. Разложение полинома на множители

Ответ. Полином разложен на два линейных двучлена и квадратный трехчлен.

Нелинейные уравнения

Для уравнений вида $f(x) = 0$ решение находится с помощью стандартной функции:

$root(\text{Выражение}, \text{Имя_переменной})$.

Пример 2.6

Решить уравнение $f(x) = x^3 + (-6)x^2 + 21x - 52$.

Решение

1. Описываем коэффициенты уравнения: $a_3 = 1$; $a_2 = -6$, $a_1 = 21$; $a_0 = -52$.
2. Записываем уравнение:

$$f(x) := a_3x^3 + a_2x^2 + a_1x + a_0.$$

3. Задаем начальное значение « x » и вычисляем корень x_1 :

$$x := 0 \quad x_1 := root(f(x), x) \quad x_1 = 4.$$

4. Вычисляем комплексные корни x_2 и x_3 :

$$i := \sqrt{-1}; \quad x_2 := root\left(\frac{f(x)}{x - x_1}, x\right); \quad x_2 = 1 + 3,464i.$$

При наборе выражения $x := 1 + 1i$ « x » примет значение нуль. Чтобы избавиться от этого, следует удалить выражение $x = 0$, стоящее выше, записать выражение $x := 1 + 1i$ и затем восстановить запись $x = 0$.

5. Вычисляем последний корень x_3 , который должен быть комплексным:

$$x_3 := root\left(\frac{f(x)}{(x - x_1)(x - x_2)}, x\right); \quad x_3 = 1 - 3,464i.$$

На рис. 2.11 (копия экрана) показан описанный выше процесс набора и решения рассматриваемого уравнения в пакете MathCAD.

$$\begin{aligned}
 a3 &:= 1 & a2 &:= -6 & a1 &:= 21 & a0 &:= -52 \\
 f(x) &:= a3 \cdot x^3 + a2 \cdot x^2 + a1 \cdot x + a0 \\
 x &:= 0 & x1 &:= \text{root}(f(x), x) = 4 \\
 i &:= \sqrt{-1} & \underline{x} &:= x1 + 1 \cdot i \\
 x2 &:= \text{root}\left(\frac{f(x)}{x - x1}, x\right) = 1 + 3.464i \\
 x3 &:= \text{root}\left[\frac{f(x)}{(x - x1) \cdot (x - x2)}, x\right] = 1 - 3.464i
 \end{aligned}$$

Рис. 2.11. Решение нелинейного уравнения в пакете MathCAD

При наборе выражения в скобках следует набрать дробь «/», поставить запятую – перед дробью появится маркер. Выделить и вырезать дробь (**ctrl** + **x**) и вставить ее (**ctrl** + **c**) на место маркера перед запятой. На месте второго маркера написать x .

Ответ. Заданное уравнение $x^3 + (-6)x^2 + 21x - 52 = 0$ имеет один действительный корень $x_1 = 4$ и два комплексных корня: $x_2 = 1 + 3,464i$ и $x_3 = 1 - 3,464i$.

Определение корней многочлена степени n

Для решения этой задачи используется стандартная функция MathCAD – **polyroots(v)**, которая возвращает вектор корней многочлена степени n . Коэффициенты многочлена записываются в векторе V , имеющем длину $n + 1$. Корни многочлена могут быть как вещественными, так и комплексными.

Не рекомендуется пользоваться этой функцией, если степень полинома выше 5–6.

Пример 2.7

Определить корни полинома $x^3 - 6x^2 + 21x - 52 = 0$.

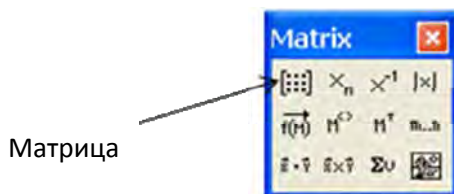
Решение

1. Набираем вектор $\mathbf{V} :=$

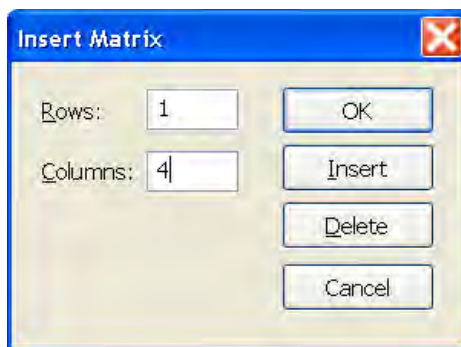
2. На наборной панели **Matrix** щелкаем по пиктограмме \mathbf{M}^T ,

получаем $V := \begin{bmatrix} \\ \\ \end{bmatrix}^T$.

3. На той же панели щелкнем по пиктограмме матрицы



и устанавливаем одну строку и четыре столбца:



получаем

$$v := \begin{bmatrix} & & & \end{bmatrix}^T$$

4. Заполняем вектор \mathbf{V} коэффициентами полинома, начиная со свободного члена:

$$\mathbf{v} := (-52 \quad 21 \quad -6 \quad 1)^T$$

5. Набираем название стандартной функции и получаем результат.

$$\text{polyroots}(\mathbf{v}) = \begin{pmatrix} 1 + 3.464i \\ 1 - 3.464i \\ 4 \end{pmatrix}$$

Весь описанный выше процесс показан на рис. 2.12 (копия экрана).

$$\mathbf{v} := (-52 \quad 21 \quad -6 \quad 1)^T$$

$$\text{polyroots}(\mathbf{v}) = \begin{pmatrix} 1 + 3.464i \\ 1 - 3.464i \\ 4 \end{pmatrix}$$

Рис. 2.12. Определение корней полинома

Ответ. С использованием пакета MathCAD определены корни полинома $x^3 - 6x^2 + 21x - 52 = 0$. Вещественный корень $x_1 = 4$ и два комплексных корня $x_{23} = 1 \pm 3,464i$.

Решение уравнений с помощью пакета Excel

Пример 2.8

Пусть требуется определить буксование δ (в процентах) трактора из уравнения

$$h = \frac{\sigma_0}{k} \cdot \frac{1}{1 - \frac{p}{\sigma_0}} \cdot \frac{1 + \delta}{1 - 0,5\delta} \operatorname{arcth} \frac{p}{\sigma_0},$$

если известно, что глубина колеи $h = 10$ см; давление движителя на почву $p = 80$ кПа; предел прочности грунта на одноосное сжатие $\sigma_0 = 1,54 \cdot 10^6$ Па; коэффициент объемного смятия грунта $k = 1,15 \cdot 10^6$ Н/м³. Аналитическое решение подтвердить графически.

Решение

Исходя из пределов изменения буксования ($0 \leq \delta \leq 1$), решение необходимо искать на интервале от 0 до 1, т. е. [0; 1]. Введем исходные данные. Сделать это можно так, как представлено на рис. 2.13.

	A	B	C	D
1	Решить уравнение	$h = \frac{\sigma_0}{k} \cdot \frac{1}{1 - \frac{p}{\sigma_0}} \cdot \frac{1 + \delta}{1 - 0,5\delta} \cdot \operatorname{arcth} \frac{p}{\sigma_0}$		
2				
3	Глубина колеи h, м	Давление движителя p, Па	Предел прочности грунта σ_0 , Па	Коэффициент объемного смятия грунта k, Н/м ³
4	0,1	80000	1540000	1150000
5				
6	Начальное приближение		0	
7	Значение функции		=A4-C4/D4*1/(1-B4/C4)*(1+C6)/(1-0,5*C6)*АТАНН(B4/C4)	
8	Значение буксования		=C6	

Рис. 2.13. Задание исходных данных и начальных значений

При вводе исходных данных все величины необходимо представить в системе СИ.

В ячейку С6 вводим начальное приближение, относительно которого будем искать решение, в ячейку С7 – решаемое уравнение, предварительно перенеся все члены из правой части в левую. Задачу решим двумя способами: «Подбор параметра» и «Поиск решения».

1-й способ. Для решения заданного уравнения воспользуемся возможностью метода *Подбор параметра* из вкладки *Данные*. В группе *Работа с данными* необходимо выбрать *Анализ «что если»*. Заполняем окно *Подбор параметра* в соответствии с рис. 2.14.

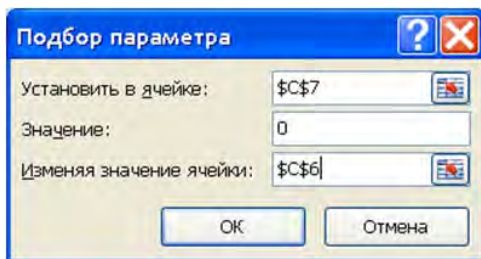
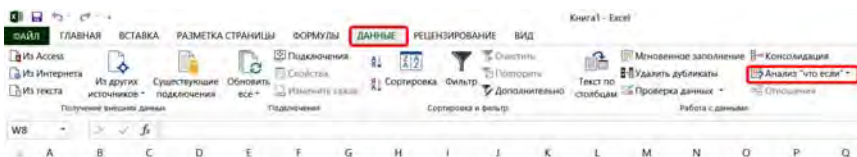


Рис. 2.14. Пример заполнения окна «Подбор параметра»

В поле **Установить в ячейке** даем ссылку на ячейку, в которой содержится решаемое уравнение, т. е. на ячейку **C7**. В поле **Изменение значения ячейки** ссылаемся на ячейку, содержащую начальное приближение, т. е. ячейку **C6**.

После щелчка по кнопке **ОК** получаем решение уравнения. При этом корень будет выведен в ячейку **C6**, т. е. в ячейку, в которой указывалось начальное приближение, рис. 2.15.

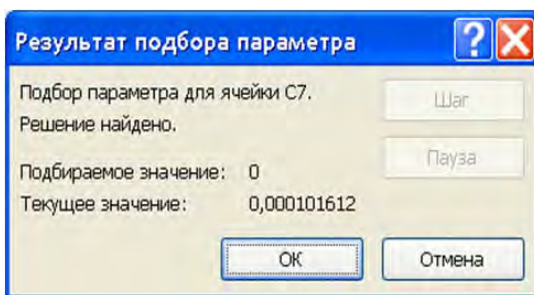


Рис. 2.15. Результат работы мастера **Подбор параметра**

В ячейке **C7** будет выведено значение функции при найденном корне. Очевидно, что это значение должно быть близким к нулю. В ячейке **C8** представлено буксование трактора в процентах, рис. 2.16.

	A	B	C	D
1	Решить уравнение	$h = \frac{\sigma_0}{k} \cdot \frac{1}{1 - \frac{p}{\sigma_0}} \cdot \frac{1 + \delta}{1 - 0.5\delta} \cdot \operatorname{arctgh} \frac{p}{\sigma_0}$		
2				
3	Глубина колеи h, м	Давление двигателя р, Па	Предел прочности грунта σ_0 , Па	Коэффициент объемного смятия грунта k, Н/м ³
4	0,1	80000	1540000	1150000
5				
6	Начальное приближение		0,215127888	
7	Значение функции		9,18342E-07	
8	Значение буксования		21,51%	

Рис. 2.16. Результат определения буксования трактора с использованием мастера подбор параметра

Для получения буксования трактора в процентах установить в ячейке С8 процентный формат этой ячейки: Главная–Число–Процентный формат.

2-й способ. Решить это же уравнение можно, используя мастер *Поиск решения*, который вызывается через меню *Сервис*. Если такого пункта в меню нет, то необходимо вызвать *Сервис-Надстройки* и в появившемся окне установить флажок напротив пункта *Поиск решения*.

Для Excel 2010/2013 надстройка *Поиск решения* находится на вкладке *Данные* в группе *Анализ*. Если ее там нет, то надстройку необходимо загрузить следующим образом:

1. На вкладке *Файл* выбрать команду *Параметры*, а затем — категорию *Надстройки*.

2. В поле *Управление* выбрать значение *Надстройки Excel* и нажать кнопку *Перейти*.

3. В поле *Доступные надстройки* установить флажок рядом с пунктом *Поиск решения* и нажать кнопку *ОК*.

После вызова мастера *Поиск решения* в поле *Установить целевую ячейку* (в более новых версиях Excel поле будет называться *Оптимизировать целевую функцию*) указать на решаемое уравнение, т. е. ячейку *С7* принять равной значению нуль. В поле *Изме-*

няя ячейки необходимо указать на ячейку с начальным приближением *С6*. Остальные поля можно не заполнять. Щелкаем по кнопке **Выполнить** (Найти решение).

Результат решения появляется в диалоговом окне **Результаты поиска решения**, рис. 2.17.

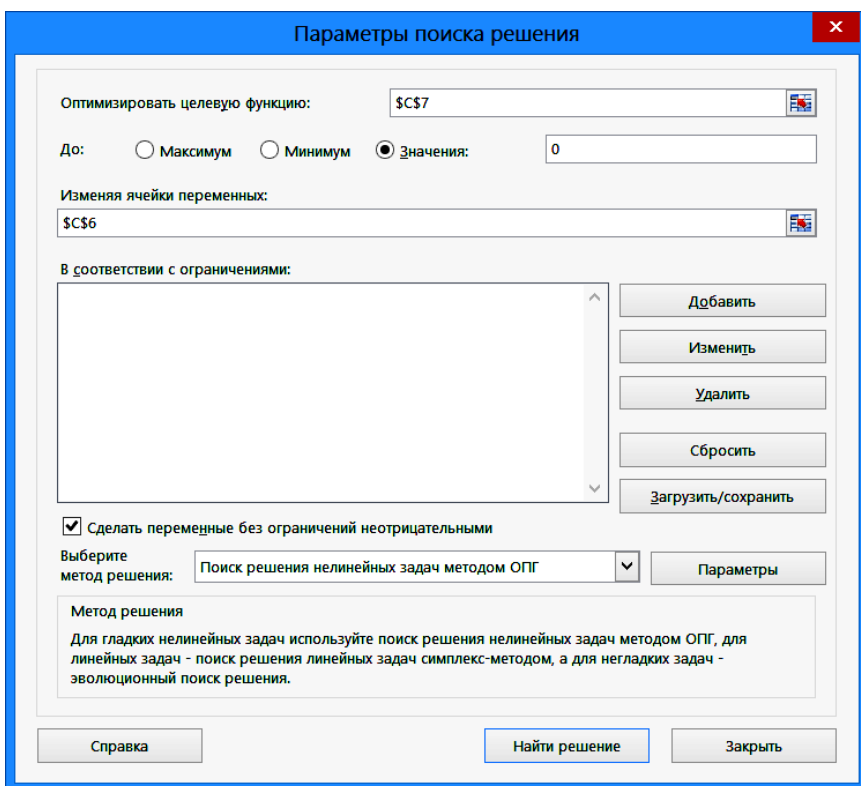


Рис. 2.17. Настройка мастера **Поиск решения**

Результат решения заданного уравнения с применением мастера **Поиск решения** представлен на рис. 2.18.

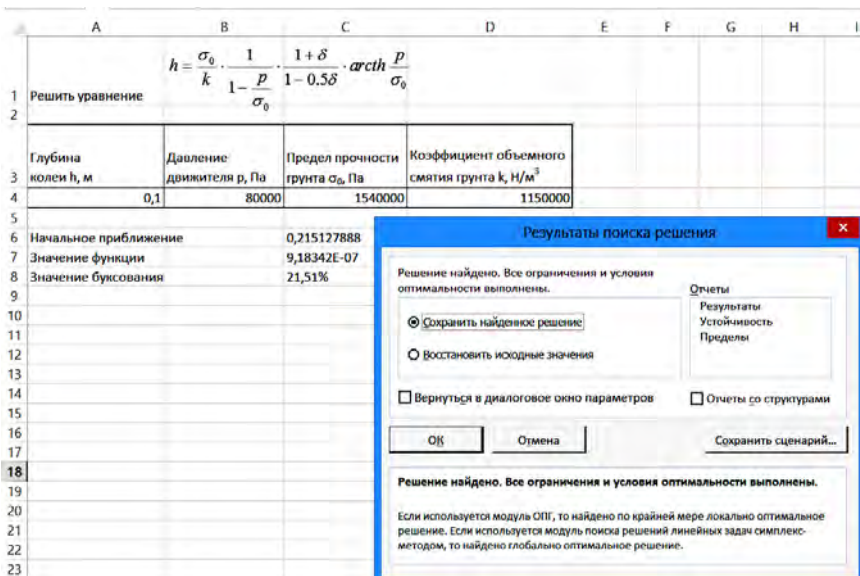


Рис. 2.18. Результат решения заданного уравнения с использованием мастера *Поиск решения*

Ответ. Буксование трактора составляет 21,51 %.

Варианты заданий

Найти корни уравнений в соответствии со своим вариантом (см. варианты заданий) с заданной точностью, с помощью пакетов MathCAD и Excel.

Решение нелинейных систем уравнений методом Ньютона

В лабораторной работе № 1 решались системы линейных уравнений. Однако многие практические технические задачи сводятся к решению системы нелинейных уравнений

$$\begin{aligned}
 f_1(x_1, x_2, \dots, x_n) &= 0; \\
 f_2(x_1, x_2, \dots, x_n) &= 0; \\
 &\dots\dots\dots \\
 f_n(x_1, x_2, \dots, x_n) &= 0.
 \end{aligned}
 \tag{2.7}$$

Значения f_1, f_2, \dots, f_n и их производные вычисляются при x_1, x_2, \dots, x_n . Определителем системы (2.8) является *якобиан*

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}, \dots, \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1}, \dots, \frac{\partial f_2}{\partial x_n} \\ \dots\dots\dots \\ \frac{\partial f_n}{\partial x_1}, \dots, \frac{\partial f_n}{\partial x_n} \end{bmatrix}.$$

Для существования единственного решения системы (2.8) якобиан должен быть отличным от нуля на каждой итерации. Расчет прекращается, если все приращения становятся малыми по абсолютной величине: $\max|\Delta x_i| < \varepsilon$. В противном случае ($\max|\Delta x_i| > \varepsilon$) значения x_i используются как приближенное значение корней и процесс повторяется до тех пор, пока не будет найдено решение или станет ясно, что получить его не удастся. Схема соответствующего алгоритма представлена на рис. 2.19.

В методе Ньютона для обеспечения сходимости решения важен удачный выбор начального приближения x_{0i} . Сходимость ухудшается с увеличением числа уравнений решаемой системы. Итерационный процесс, определяемый системой (2.8), сходится, если

$$\sum_{i,k=1}^n \left[\frac{\partial f_i}{\partial x_k} \right] < 1.$$

Таким образом, итерационный процесс решения системы уравнений (2.7) методом Ньютона состоит в определении приращений $\Delta x_1, \Delta x_2, \dots, \Delta x_n$ к значениям неизвестных x_i на каждой итерации.

Чтобы уяснить алгоритм метода Ньютона для решения систем нелинейных уравнений, рассмотрим пример решения системы двух нелинейных уравнений, который решим вручную.

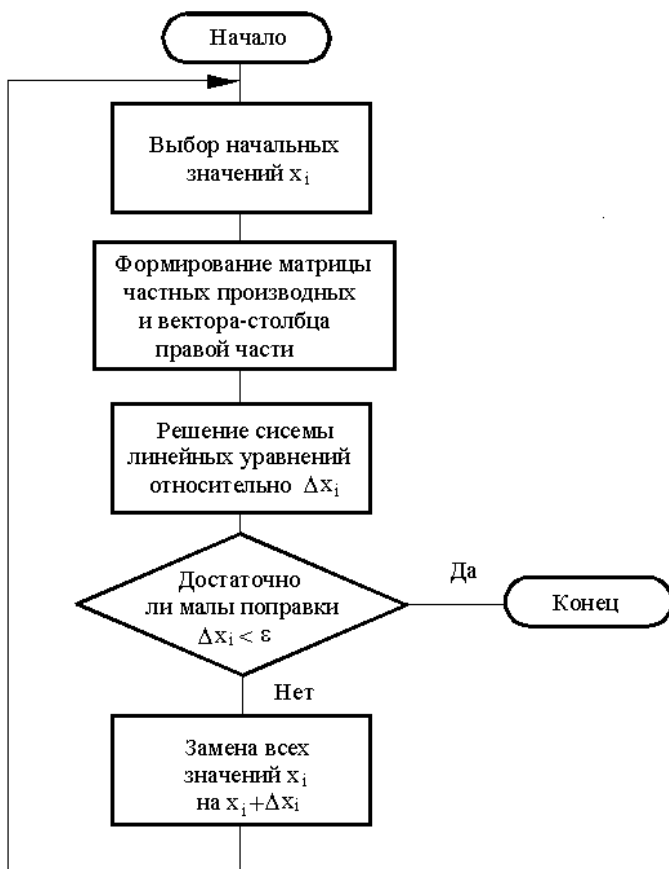


Рис. 2.19. Блок-схема алгоритма метода Ньютона для решения систем нелинейных уравнений

Пример 2.9

Методом Ньютона уточнить приближенные значения $x_0 = 0,4$; $y_0 = 0,9$ системы нелинейных уравнений

$$f_1(x, y) \equiv 4x^2 + y^2 + 2xy - y - 2 = 0;$$

$$f_2(x, y) \equiv 2x^2 + 3xy + y^2 - 3 = 0.$$

Решение

Вычислим частные производные от $f_1(x, y)$ и $f_2(x, y)$ по переменным x и y :

$$\begin{aligned}\frac{\partial f_1}{\partial x} &= 8x + 2y; & \frac{\partial f_1}{\partial y} &= 2x + 2y - 1; \\ \frac{\partial f_2}{\partial x} &= 4x + 3y; & \frac{\partial f_2}{\partial y} &= 3x + 2y.\end{aligned}$$

Исходя из начального приближения, на первой итерации находим x_1 и y_1 :

$$\begin{aligned}f_1(x_0, y_0) &= -0,73; & \frac{\partial f_1(x_0, y_0)}{\partial x} &= 5,0; & \frac{\partial f_1(x_0, y_0)}{\partial y} &= 1,6; \\ f_2(x_0, y_0) &= -0,79; & \frac{\partial f_2(x_0, y_0)}{\partial x} &= 4,3; & \frac{\partial f_2(x_0, y_0)}{\partial y} &= 3,0.\end{aligned}$$

В результате получаем следующую систему линейных алгебраических уравнений:

$$\begin{aligned}5,0\Delta x_0 + 1,6\Delta y_0 &= 0,73; \\ 4,3\Delta x_0 + 3,0\Delta y_0 &= 0,79.\end{aligned}$$

Откуда

$$\begin{aligned}\Delta x_0 &= 0,114; & x_1 &= x_0 + \Delta x_0 = 0,514; \\ \Delta y_0 &= 0,100; & y_1 &= y_0 + \Delta y_0 = 1,000.\end{aligned}$$

На второй итерации определяем x_2 и y_2 :

$$\begin{aligned}f_1(x_1, y_1) &= -0,084784; & \frac{\partial f_1(x_1, y_1)}{\partial x} &= 6,112; & \frac{\partial f_1(x_1, y_1)}{\partial y} &= 2,028; \\ f_2(x_1, y_1) &= -0,070392; & \frac{\partial f_2(x_1, y_1)}{\partial x} &= 5,056; & \frac{\partial f_2(x_1, y_1)}{\partial y} &= 3,542.\end{aligned}$$

$$6,112\Delta x_1 + 2,028\Delta y_1 = -0,084784;$$

$$5,056\Delta x_1 + 3,542\Delta y_1 = -0,070392.$$

$$\Delta x_1 = -0,013826; \quad x_2 = 0,500174;$$

$$\Delta y_1 = -0,000138; \quad y_2 = 0,999862.$$

На третьей итерации находим x_3 и y_3 :

$$f_1(x_2, y_2) = 0,000768; \quad \frac{\partial f_1(x_2, y_2)}{\partial x} = 6,001116; \quad \frac{\partial f_1(x_2, y_2)}{\partial y} = 2,000072;$$

$$f_2(x_2, y_2) = 0,000387; \quad \frac{\partial f_2(x_2, y_2)}{\partial x} = 5,000282; \quad \frac{\partial f_2(x_2, y_2)}{\partial y} = 3,500246.$$

$$6,001116\Delta x_2 + 2,000072\Delta y_2 = -0,000768;$$

$$5,000282\Delta x_2 + 3,500246\Delta y_2 = -0,000387.$$

$$\Delta x_2 = -0,000174; \quad x_3 = 0,5;$$

$$\Delta y_2 = -0,000138; \quad y_3 = 1,0.$$

Для решения данной системы нелинейных уравнений в соответствии с алгоритмом (см. рис. 2.19) была разработана программа на языке Delphi, результаты решения по которой представлены ниже.

Решение нелинейной системы уравнений методом Ньютона

Исходные данные

Расчет выполнил :	Иванов И.И.
Номер группы	каф. «Тракторы»
Число уравнений	2
Точность решения	0.000010

Результаты решения

Итерация	X	Y
0	0.400000	0.900000
1	0.514039	0.999877
2	0.500175	0.999860
3	0.500000	1.000000
4	0.500000	1.000000

----- *** Конец расчета *** -----

В качестве исходных данных задаются начальные приближения переменных x_0 и y_0 , погрешность ε и допустимое число итераций i (заложено в программе $\varepsilon = 1E-6$; $i \leq 20$).

Ответ. В качестве уточненного решения заданной системы нелинейных уравнений принимаем $x = 0,5$; $y = 1$.

Как и ранее, решаемую систему нелинейных уравнений и производные следует размещать в подпрограмме **Fun**:

```

Procedure Fun(X:MasX;Var B:MasX;Var A:MasA);
begin
  B[1]:=-(4*X[1]*X[1]+X[2]*X[2]+2*X[1]*X[2]-X[2]-2);
//уравнения
  B[2]:=-(2*X[1]*X[1]+3*X[1]*X[2]+X[2]*X[2]-3);

  A[1,1]:=8*X[1]+2*X[2]; //производные
  A[1,2]:=2*X[1]+2*X[2]-1;
  A[2,1]:=4*X[1]+3*X[2];
  A[2,2]:=3*X[1]+2*X[2];
end;
```

Так как в дальнейшем для определения приращений переменных на каждой итерации используется подпрограмма **Gaus**, были сохранены наименования массивов. Так, уравнения записываются в массив B со знаком «минус» – вектор-столбец свободных членов. Производные записываются в двумерный массив A – коэффициенты

уравнений перед неизвестными. Одномерный массив X – текущие значения переменных x и y на каждой итерации.

В головной программе организуется обращение к подпрограммам **Fun** и **Gaus**, а также определение новых значений переменных x и y на каждой итерации:

```
i:=0; //счётчик
writeln(Fw,":5,i:9,X[1]:16:6,X[2]:11:6);
repeat
  Inc(i);
  Fun(X,B,A);

  Gaus(n,A,B,dX);

  X[1]:=X[1]+dX[1]; //новое значение x
  X[2]:=X[2]+dX[2]; //новое значение y
  writeln(Fw,":10,i:4,X[1]:16:6,X[2]:11:6);

until((abs(dX[1])<e) and (abs(dX[2])<e)) or (i>20);
```

Чтобы удобнее было пользоваться подпрограммой **Gaus**, одномерный массив результатов решения системы уравнений X переименовали в одномерный массив dX – массив приращений Δx . Наибольшее число итераций равно 20.

Как и ранее, результаты расчета на главной форме желательно размещать в компоненте **StringGrid**.

Решение системы нелинейных уравнений в MathCAD

При решении систем нелинейных уравнений используется специальный вычислительный блок, открываемый служебным словом – директивой **Given** – и имеющий следующую структуру:

Given

Уравнения

Ограничительные условия

Выражения с функциями Find или Minerr.

Например, для системы уравнений, приведенных в примере 2.9, решение будет иметь следующий вид:

$$x := 0.4 \quad y := 0.9$$

Given

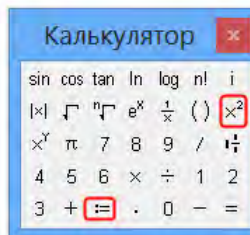
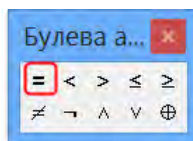
$$4 \cdot x^2 + y^2 + 2 \cdot x \cdot y - y - 2 = 0$$

$$2 \cdot x^2 + 3 \cdot x \cdot y + y^2 - 3 = 0$$

$$\begin{pmatrix} x \\ y \end{pmatrix} := \text{Find}(x, y)$$

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0.5 \\ 1 \end{pmatrix}$$

+



В формулах используется не просто знак = (равно), а = (**жирный равно**). Данный символ набирается комбинацией клавиш **Ctrl+=** или с использованием панели **Булева алгебра**.

В блоке применяется одна из следующих двух функций:

Find(v_1, v_2, \dots, v_n) — возвращает значение одной или ряда переменных для точного решения;

Minerr(v_1, v_2, \dots, v_n) — возвращает значение одной или ряда переменных для приближенного решения.

Между этими функциями существуют принципиальные различия. Первая функция используется, когда решение реально существует (хотя и не является аналитическим). Вторая функция пытается найти максимальное приближение даже к несуществующему решению путем минимизации среднеквадратичной погрешности решения.

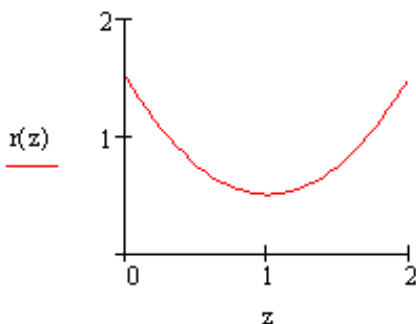
При использовании функции **Minerr** для решения систем нелинейных уравнений надо проявлять известную осторожность и обязательно предусматривать проверку решений. Нередки случаи, когда решения могут оказаться ошибочными, чаще всего из-за того, что из нескольких корней система предлагает нереальный (или не представляющий интереса) корень. Полезно как можно точнее указывать начальные приближения к решению.

Ниже представлен пример, из которого видно, что уравнение не имеет решения и соответственно функция **Find** выдает ошибку, а функция **Minerr** находит максимальное приближение, равное 1, даже к несуществующему решению.

```

r(z) := z2 - 2·z + 1.5
z := 0 Given r(z) = 0 z0 := Find(z)
z := 0 Given r(z) = 0 z0 := MinErr(z)
z := 0,0.05..2 z0 = 1

```



Варианты заданий

Найти корни нелинейных систем уравнений в соответствии со своим вариантом, с заданной точностью, с помощью разработанной программы на языке Delphi. Выполнить проверку.

Номер варианта	Системы нелинейных уравнений	
1	$\begin{cases} \sin(x+1) - y = 1,2; \\ 2x + \cos y = 2 \end{cases}$	$\begin{cases} \operatorname{tg}(xy + 0,4) = x^2; \\ 0,6x^2 + 2y^2 = 1, x > 0, y > 0 \end{cases}$
2	$\begin{cases} \cos(x-1) + y = 0,5; \\ x - \cos y = 3 \end{cases}$	$\begin{cases} \sin(x+y) - 1,6 = 0; \\ x^2 + y^2 = 1, x > 0, y > 0 \end{cases}$

Номер варианта	Системы нелинейных уравнений	
3	$\begin{cases} \sin x + 2y = 2; \\ \cos(y - 1 + x) = 0,7 \end{cases}$	$\begin{cases} \operatorname{tg}(xy + 0,1) = x^2; \\ x^2 + 2y^2 = 1 \end{cases}$
4	$\begin{cases} \cos x + y = 1,5; \\ 2x - \sin 9y - 0,5 = 1 \end{cases}$	$\begin{cases} \sin(x + y) - 1,2x = 0,2; \\ x^2 + y^2 = 1 \end{cases}$
5	$\begin{cases} \sin(x + 0,5) - y = 1; \\ \cos(y - 2) + x = 0 \end{cases}$	$\begin{cases} \operatorname{tg}(xy + 0,3) = x^2; \\ 0,9x^2 + 2y^2 = 1 \end{cases}$
6	$\begin{cases} \cos(x + 0,5) + y = 0,8; \\ \sin y - 2x = 1,6 \end{cases}$	$\begin{cases} \sin(x + y) - 1,3x = 0; \\ x^2 + y^2 = 0 \end{cases}$
7	$\begin{cases} \sin(x - 1) = 1,3 - y; \\ x - \sin(y + 1) = 0,8 \end{cases}$	$\begin{cases} \operatorname{tg}xy = x^2; \\ 0,8x^2 + 2y^2 = 1 \end{cases}$
8	$\begin{cases} 2y - \cos(x + 1) = 0; \\ x + \sin y = -0,4 \end{cases}$	$\begin{cases} \sin(x + y) - 1,5x = 0,1; \\ x^2 + y^2 = 1 \end{cases}$
9	$\begin{cases} \cos(x + 0,5) - y = 2; \\ \sin y - 2x = 1 \end{cases}$	$\begin{cases} \operatorname{tg}xy = x^2; \\ 0,7x^2 + 2y^2 = 1 \end{cases}$
10	$\begin{cases} \sin(x + 2) - y = 1,5; \\ x + \cos(y - 2) - 0,5 \end{cases}$	$\begin{cases} \sin(x + y) - 1,2x = 0,1; \\ x^2 + y^2 = 1 \end{cases}$
11	$\begin{cases} \sin(x + 1) - x = 1,2; \\ 2y + \cos x = 2 \end{cases}$	$\begin{cases} g(xy + 0,2) = x^2; \\ 0,6x^2 + 2y^2 = 1 \end{cases}$
12	$\begin{cases} \cos(y - 1) + x = 0,5; \\ y - \cos x = 3 \end{cases}$	$\begin{cases} \sin(x + y) = 1,5x - 0,1; \\ x^2 + y^2 = 1 \end{cases}$
13	$\begin{cases} \sin y + 2x = 2; \\ \cos(x - 1) + y = 0,7 \end{cases}$	$\begin{cases} \operatorname{tg}(xy + 0,4) = x^2; \\ 0,8x^2 + 2y^2 = 1 \end{cases}$

Номер варианта	Системы нелинейных уравнений	
14	$\begin{cases} \cos y + x = 1,5; \\ 2y - \sin(x - 0,5) = 1 \end{cases}$	$\begin{cases} \sin(x + y) = 1,2x - 0,1; \\ x^2 + y^2 = 1 \end{cases}$
15	$\begin{cases} \sin(x + 0,5) - x = 1; \\ \cos(x - 2) + y = 0 \end{cases}$	$\begin{cases} \operatorname{tg}(xy + 0,1) = x^2; \\ 0,9x^2 + 2y^2 = 1 \end{cases}$
16	$\begin{cases} \cos(x + 0,5) + x = 0,8; \\ \sin x - 2y = 1,6 \end{cases}$	$\begin{cases} \sin(x + y) - 1,4x = 0; \\ x^2 + y^2 = 1 \end{cases}$
17	$\begin{cases} \sin(y - 1) + x = 1,3; \\ y - \sin(x + 1) = 0,8 \end{cases}$	$\begin{cases} g(xy + 0,1) = x^2; \\ 0,5x^2 + y^2 = 1 \end{cases}$
18	$\begin{cases} 2x - \cos(y + 1) = 0; \\ y + \sin x = -0,4 \end{cases}$	$\begin{cases} \sin(x + y) = 1,1x - 0,1; \\ x^2 + y^2 = 1 \end{cases}$
19	$\begin{cases} \cos(x + 0,5) - x = 2; \\ \sin x - 2y = 1 \end{cases}$	$\begin{cases} \operatorname{tg}(x - y) - xy = 0; \\ x^2 + 2y^2 = 1 \end{cases}$
20	$\begin{cases} \sin(x + 2) - x = 1,5; \\ y + \cos(x - 2) = 0,5 \end{cases}$	$\begin{cases} \sin(x - y) - xy = -1; \\ x^2 + y^2 = 3/4 \end{cases}$

Лабораторная работа № 3

РЕШЕНИЕ ОБЫКНОВЕННЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ И ИХ СИСТЕМ

Цель работы

1. Освоить методику понижения порядка дифференциального уравнения любого порядка до первого порядка, метод Рунге–Кутты 4-го порядка для решения дифференциальных уравнений 1-го порядка и их систем.

2. На языке Delphi разработать программу для решения систем уравнений методом Рунге–Кутты 4-го порядка, а также освоить методику решения уравнений и их систем с помощью пакета MathCAD.

Основные понятия

Решение различных геометрических, физических и инженерных задач часто приводят к уравнениям, которые связывают независимые переменные, характеризующие ту или иную задачу, с какой-либо функцией этих переменных и производными этой функции различных порядков.

В качестве примера можно рассмотреть простейший случай равноускоренного движения материальной точки.

Известно, что перемещение (путь) материальной точки при равноускоренном движении является функцией времени и выражается формулой

$$s = v_0 t + \frac{at^2}{2}.$$

В свою очередь, ускорение a является производной по времени t от скорости v , которая также является производной по времени t от перемещения s , т. е.

$$v = \frac{ds}{dt}; \quad a = \frac{dv}{dt} = \frac{d^2s}{dt^2}.$$

Тогда получаем уравнение

$$s = f(t) = v_0 t + \frac{f''(t)t^2}{2},$$

которое связывает функцию $f(t)$ с независимой переменной t и производной второго порядка функции $f''(t)$.

Дифференциальным уравнением называется уравнение, связывающее независимые переменные, их функции и производные (или дифференциалы) этой функции. Если дифференциальное уравнение имеет одну независимую переменную, то оно называется обыкновенным дифференциальным уравнением, если же независимых переменных две или более, то такое дифференциальное уравнение называется дифференциальным уравнением в частных производных.

Наивысший порядок производных, входящих в уравнение, называется *порядком дифференциального уравнения*.

Пример 3.1

Уравнение

$$x^3 y' + 8y - x + 5 = 0$$

представляет собой обыкновенное дифференциальное уравнение 1-го порядка. В общем виде записывается как

$$F(x, y, y') = 0.$$

Уравнение

$$x \frac{d^2 y}{dx^2} + xy \frac{dy}{dx} + x^2 = y$$

является обыкновенным дифференциальным уравнением 2-го порядка. В общем виде записывается как

$$F(x, y, y', y'') = 0.$$

Уравнение

$$y^2 \frac{\partial z}{\partial x} + xy \frac{\partial z}{\partial y} = 0$$

является дифференциальным уравнением в частных производных 1-го порядка.

Общим решением дифференциального уравнения называется такая дифференцируемая функция

$$y = \varphi(x, C),$$

которая при подстановке в исходное уравнение вместо неизвестной функции обращает уравнение в тождество.

Свойства общего решения:

– так как постоянная C – произвольная величина, то дифференциальное уравнение имеет бесконечное множество решений;

– при каких-либо начальных условиях $x = x_0, y(x_0) = y_0$ существует такое значение $C = C_0$, при котором решением дифференциального уравнения является функция $y = \varphi(x, C_0)$.

Решение вида $y = \varphi(x, C_0)$ называется частным решением дифференциального уравнения.

Задачей Коши (Огюстен Луи Коши (1789–1857) – французский математик) называется нахождение любого частного решения дифференциального уравнения вида $y = \varphi(x, C_0)$, удовлетворяющего начальным условиям $y(x_0) = y_0$.

Теорема Коши (теорема о существовании и единственности решения дифференциального уравнения 1-го порядка):

Если функция $f(x, y)$ непрерывна в некоторой области D в плоскости XOY и имеет в этой области непрерывную частную производную $y' = f(x, y)$, то какова бы не была точка (x_0, y_0) в области D , существует единственное решение $y = \varphi(x)$ уравнения $y' = f(x, y)$, определенное в некотором интервале, содержащем точку x_0 , при $x = x_0$ принимающее значение $\varphi(x_0) = y_0$, т. е. существует единственное решение дифференциального уравнения.

Интегралом дифференциального уравнения называется любое уравнение, не содержащее производных, для которого данное дифференциальное уравнение является следствием.

Пример 3.2

Найти общее решение дифференциального уравнения $xy' + y = 0$.

Решение

Общее решение дифференциального уравнения ищется с помощью интегрирования левой и правой частей уравнения, которое предварительно преобразовано следующим образом:

$$x \frac{dy}{dx} + y = 0;$$

$$x dy = -y dx;$$

$$\frac{dy}{y} = -\frac{dx}{x}.$$

Теперь интегрируем:

$$\int \frac{dy}{y} = -\int \frac{dx}{x};$$

$$\ln y = -\ln x + C_0;$$

$$\ln y + \ln x = C_0;$$

$$\ln xy = C_0;$$

$$xy = e^{C_0} = C;$$

$y = \frac{C}{x}$ – это общее решение исходного дифференциального уравнения.

Допустим, заданы некоторые начальные условия $x_0 = 1$; $y_0 = 2$, тогда имеем

$$2 = \frac{C}{1}; \quad C = 2.$$

После подстановки полученного значения постоянной в общее решение получаем частное решение при заданных начальных условиях (решение задачи Коши):

$$y = \frac{2}{x}.$$

Ответ. Решением заданного дифференциального уравнения является функция $y = 2/x$.

Интегральной кривой называется график $y = \varphi(x)$ решения дифференциального уравнения $\varphi(x)$ на плоскости XOY . *Особым решением дифференциального уравнения* называется такое решение, во всех точках которого условие единственности Коши не выполняется, т. е. в окрестности некоторой точки (x, y) существует не менее двух интегральных кривых. Особые решения не зависят от постоянной C , их нельзя получить из общего решения ни при каких значениях постоянной C . Если построить семейство интегральных кривых дифференциального уравнения, то особое решение будет изображаться линией, которая в каждой своей точке касается по крайней мере одной интегральной кривой. Отметим, что не каждое дифференциальное уравнение имеет особые решения.

Пример 3.3

Найти общее решение дифференциального уравнения $y' + y = 0$.
Найти особое решение, если оно существует.

$$\frac{dy}{dx} = -y;$$

$$\frac{dy}{y} = -dx;$$

$$\int \frac{dy}{y} = -\int dx;$$

$$\ln y = -x + C;$$

$$y = e^{-x} e^C.$$

Ответ. Результатом решения исходного дифференциального уравнения является функция $y = e^{-x}e^C$. Уравнение $y = C_1e^{-x}$ имеет также особое решение $y = 0$. Это решение невозможно получить из общего, однако при подстановке в исходное уравнение получаем тождество. Мнение, что решение $y = 0$ можно получить из общего решения при $C_1 = 0$, ошибочно, ведь $C_1 = e^C \neq 0$.

Дифференциальные уравнения первого порядка

Дифференциальным уравнением первого порядка называется соотношение, связывающее функцию, ее первую производную и независимую переменную, т. е. соотношение вида

$$F(x, y, y') = 0.$$

Если такое соотношение преобразовать к виду $y' = f(x, y)$, то это дифференциальное уравнение первого порядка будет называться *уравнением, разрешенным относительно производной*.

Дифференциальное уравнение называется линейным относительно неизвестной функции и ее производной, если оно может быть записано в виде

$$y' + P(x)y = Q(x).$$

При этом если правая часть $Q(x)$ равна нулю, то такое уравнение называется *линейным однородным дифференциальным уравнением*; если правая часть $Q(x)$ не равна нулю, то такое уравнение называется *линейным неоднородным дифференциальным уравнением*. $P(x)$ и $Q(x)$ – функции, непрерывные на некотором промежутке $a < x < b$.

Пример 3.4

Решить дифференциальное уравнение

$$xy' + y = x + 1$$

с начальным условием $y(1) = 0$.

Решение

Имеем линейное неоднородное уравнение. Его решением будет соответствующее однородное уравнение

$$xy' + y = 0;$$

$$\frac{xdy}{dx} = -y;$$

$$\frac{dy}{y} = -\frac{dx}{x};$$

$$\ln|y| = -\ln|x| + \ln C;$$

$$xy = C; \quad y = \frac{C}{x}.$$

Решение неоднородного уравнения будет иметь вид

$$y = \frac{C(x)}{x}.$$

Подставим его в исходное уравнение:

$$x \frac{C'(x)x - C(x)}{x^2} + \frac{C(x)}{x} = x + 1;$$

$$\frac{C'(x)x}{x} = x + 1;$$

$$C'(x) = x + 1;$$

$$C(x) = \frac{x^2}{2} + x + C.$$

Ответ. Общее решение исходного уравнения будет иметь вид

$$y = \frac{x}{2} + 1 + \frac{C}{x}.$$

С учетом начального условия $y(1) = 0$

$$0 = \frac{1}{2} + 1 + C;$$

$$C = -\frac{3}{2};$$

частное решение будет иметь вид

$$y = \frac{x}{2} - \frac{3}{2x} + 1.$$

Численные методы решения дифференциальных уравнений первого порядка

Известные методы точного интегрирования дифференциальных уравнений позволяют найти решение в виде аналитической функции, однако эти методы применимы для очень ограниченного класса функций. Большинство уравнений, встречающихся при решении практических инженерных задач, нельзя проинтегрировать с помощью этих методов.

В таких случаях используются численные методы решения, которые представляют решение дифференциального уравнения не в виде аналитической функции, а в виде таблиц значений искомой функции в зависимости от значения независимой переменной.

Существует несколько методов численного интегрирования дифференциальных уравнений, которые отличаются друг от друга по сложности вычислений и точности результата. Рассмотрим некоторые из них, которые наиболее часто применяются для решения технических задач.

Метод Эйлера

Леонард Эйлер (1707–1783) – швейцарский математик.

Решение уравнения $y' = f(x, y)$ с некоторыми начальными условиями дает интегральную кривую. Если взять последовательность

точек x_0, x_1, x_2, \dots и на получившихся отрезках интегральную кривую заменить на отрезки касательных к ней, то получим ломаную линию (рис. 3.1).

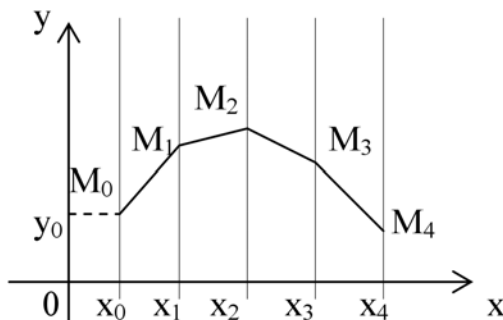


Рис. 3.1. Графическая интерпретация метода Эйлера

При подстановке заданных начальных условий (x_0, y_0) в дифференциальное уравнение $y' = f(x, y)$ получаем угловой коэффициент касательной к интегральной кривой в начальной точке:

$$\operatorname{tg} \alpha_0 = y' = f(x_0, y_0).$$

Заменяв на отрезке $[x_0, x_1]$ интегральную кривую на касательную к ней, получаем

$$y_1 = y_0 + f(x_0, y_0)(x_1 - x_0).$$

Производя аналогичную операцию для отрезка $[x_1, x_2]$, получаем

$$y_2 = y_1 + f(x_1, y_1)(x_2 - x_1).$$

Продолжая подобные действия далее, получаем ломаную кривую, которая называется ломаной Эйлера.

Можно записать общую формулу вычислений

$$y_n = y_{n-1} + f(x_{n-1}, y_{n-1})(x_n - x_{n-1}).$$

Если последовательность точек x_i выбрать так, чтобы они отстояли друг от друга на одинаковое расстояние h , называемое *шагом вычисления*, то получаем формулу

$$y_n = y_{n-1} + f(x_{n-1}, y_{n-1})h.$$

Следует отметить, что точность метода Эйлера относительно невысока. Увеличить точность можно, уменьшив шаг вычислений, однако это приведет к увеличению объема расчетов. Поэтому на практике применяется так называемый уточненный метод Эйлера или формула пересчета.

Суть метода состоит в том, что в формуле

$$y_1 = y_0 + f(x_0, y_0)h$$

вместо значения $y'_0 = f(x_0, y_0)$ берется среднее арифметическое значений $f(x_0, y_0)$ и $f(x_1, y_1)$. Тогда уточненное значение

$$y_1^{(1)} = y_0 + \frac{f(x_0, y_0) + f(x_1, y_1)}{2}h.$$

Затем находится значение производной в точке $(x_1, y_1^{(1)})$. Заменяя $f(x_0, y_0)$ средним арифметическим значений $f(x_0, y_0)$ и $f(x_1, y_1^{(1)})$, находят второе уточненное значение y_1 :

$$y_1^{(2)} = y_0 + \frac{f(x_0, y_0) + f(x_1, y_1^{(1)})}{2}h.$$

Затем третье

$$y_1^{(3)} = y_0 + \frac{f(x_0, y_0) + f(x_1, y_1^{(2)})}{2}h$$

и т. д., пока два последовательных уточненных значения не совпадут в пределах заданной степени точности. Тогда это значение принимается за ординату точки M_1 ломаной Эйлера.

Аналогичная операция производится для остальных значений y . Подобное уточнение позволяет существенно повысить точность результата.

Метод Рунге–Кутты 4-го порядка

Метод Рунге–Кутты 4-го порядка является более точным по сравнению с методом Эйлера. Суть уточнения состоит в том, что искомое решение представляется в виде разложения функции в ряд Тейлора. Если в этой формуле ограничиться двумя первыми слагаемыми, то получим формулу метода Эйлера. Метод Рунге–Кутты 4-го порядка учитывает четыре первых члена разложения:

$$y_{i+1} = y_i + y_i' h + y_i'' \frac{h^2}{2!} + y_i''' \frac{h^3}{3!} = y_i + \Delta y_i.$$

В методе Рунге–Кутты 4-го порядка приращения Δy_i предлагается вычислять по формуле

$$\Delta y_i = \frac{1}{6} \left(k_1^{(i)} + 2k_2^{(i)} + 2k_3^{(i)} + k_4^{(i)} \right),$$

где коэффициенты k_i вычисляются по формулам

$$k_1^{(i)} = hf(x_i, y_i);$$

$$k_2^{(i)} = hf \left(x_i + \frac{h}{2}; y_i + \frac{k_1^{(i)}}{2} \right);$$

$$k_3^{(i)} = hf \left(x_i + \frac{h}{2}; y_i + \frac{k_2^{(i)}}{2} \right);$$

$$k_4^{(i)} = hf \left(x_i + h; y_i + k_3^{(i)} \right).$$

Пример 3.5

Методом Рунге–Кутта 4-го порядка решить дифференциальное уравнение $y' = x + y$ при начальном условии $y(0) = 1$ на отрезке $[0; 0,5]$ с шагом $0,1$.

Для $i = 0$ вычислим коэффициенты k_i :

$$k_1^{(0)} = hf(x_0, y_0) = 0,1(x_0 + y_0) = 0,1(0 + 1) = 0,1;$$

$$k_2^{(0)} = hf\left(x_0 + \frac{h}{2}; y_0 + \frac{k_1^{(0)}}{2}\right) = 0,1(0,05 + 1,05) = 0,11;$$

$$k_3^{(0)} = hf\left(x_0 + \frac{h}{2}; y_0 + \frac{k_2^{(0)}}{2}\right) = 0,1(0,05 + 1,055) = 0,1105;$$

$$k_4^{(0)} = hf\left(x_0 + h; y_0 + k_3^{(0)}\right) = 0,1(0,1 + 1,1105) = 0,1211;$$

$$\Delta y_0 = \frac{1}{6}(k_1^{(0)} + 2k_2^{(0)} + 2k_3^{(0)} + k_4^{(0)}) = \frac{1}{6}(0,1 + 0,22 + 0,221 + 0,1211) = 0,1104;$$

$$x_1 = x_0 + h = 0,1;$$

$$y_1 = y_0 + \Delta y_0 = 1 + 0,1104 = 1,1104.$$

Последующие вычисления приводить не будем, а результаты представим в виде табл. 3.1.

Таблица 3.1

Вычисления методом Рунге–Кутта 4-го порядка для примера 3.5

i	x_i	k		Δy_i	y_i
0	0	1	0,1000	0,1104	1
		2	0,1100		
		3	0,1105		
		4	0,1155		

i	x_i	k		Δy_i	y_i
1	0,1	1	0,1210	0,1325	1,1104
		2	0,1321		
		3	0,1326		
		4	0,1443		
2	0,2	1	0,1443	0,1569	1,2429
		2	0,1565		
		3	0,1571		
		4	0,1700		
3	0,3	1	0,1700	0,1840	1,3998
		2	0,1835		
		3	0,1842		
		4	0,1984		
4	0,4	1	0,1984	0,2138	1,5838
		2	0,2133		
		3	0,2140		
		4	0,2298		
5	0,5				1,7976

Решим этот же пример методом Эйлера.

Применяем формулу

$$y_n = y_{n-1} + hf(x_{n-1}, y_{n-1}).$$

$$x_0 = 0, \quad y_0 = 1, \quad f(x_0, y_0) = x_0 + y_0 = 1;$$

$$hf(x_0, y_0) = h(x_0 + y_0) = 0,1;$$

$$y_1 = y_0 + hf(x_0, y_0) = 1 + 0,1 = 1,1.$$

$$x_1 = 0,1, \quad y_0 = 1,1, \quad f(x_1, y_1) = x_1 + y_1 = 1,2;$$

$$hf(x_1, y_1) = h(x_1 + y_1) = 0,12;$$

$$y_2 = y_1 + hf(x_1, y_1) = 1,1 + 0,12 = 1,22.$$

Производя аналогичные вычисления далее, получаем табл. 3.2 значений.

Таблица 3.2

Вычисления методом Эйлера для примера 3.5

i	0	1	2	3	4	5
x_i	0,0	0,1	0,2	0,3	0,4	0,5
y_i	1	1,1	1,22	1,362	1,528	1,721

Применим теперь уточненный метод Эйлера (табл. 3.3).

Таблица 3.3

Вычисления уточненным методом Эйлера для примера 3.5

i	0	1	2	3	4	5
x_i	0,0	0,1	0,2	0,3	0,4	0,5
y_i	1	1,1	1,243	1,400	1,585	1,799

Для сравнения точности приведенных методов численного решения данного уравнения решим его аналитически и найдем точные значения функции y на заданном отрезке.

Уравнение $y' - y = x$ является линейным неоднородным дифференциальным уравнением первого порядка. Решим соответствующее ему однородное уравнение:

$$y' - y = 0; \quad y' = y; \quad \frac{dy}{dx} = y; \quad \frac{dy}{y} = dx; \quad \int \frac{dy}{y} = \int dx;$$

$$\ln|y| = x + \ln C; \quad \ln \left| \frac{y}{C} \right| = x; \quad y = Ce^x.$$

Решение неоднородного уравнения имеет вид

$$y = C(x)e^x.$$

$$y' = C'(x)e^x + C(x)e^x;$$

$$C'(x)e^x + C(x)e^x = x + C(x)e^x;$$

$$C'(x)e^x = x; \quad C'(x) = xe^{-x};$$

$$C(x) = \int xe^{-x} dx = \left\{ \begin{array}{l} u = x; \quad dv = e^{-x} dx; \\ du = dx; \quad v = -e^{-x}; \end{array} \right\} = -xe^{-x} + \int e^{-x} dx = -xe^{-x} - e^{-x} + C.$$

Общее решение

$$y = Ce^x - x - 1.$$

С учетом начального условия

$$1 = C - 0 - 1; \quad C = 2.$$

Частное решение

$$y = 2e^x - x - 1.$$

Для сравнения полученных результатов составим табл. 3.4.

Таблица 3.4

Сравнение точности вычислений дифференциального уравнения для примера 3.5 различными методами

i	x_i	y_i			
		Метод Эйлера	Уточненный метод Эйлера	Метод Рунге–Кутты	Точное значение
0	0	1	1	1	1
1	0,1	1,1	1,1	1,1104	1,1103
2	0,2	1,22	1,243	1,2429	1,2428
3	0,3	1,362	1,4	1,3998	1,3997
4	0,4	1,528	1,585	1,5838	1,5837
5	0,5	1,721	1,799	1,7976	1,7975

Ответ. Как видно из полученных результатов, метод Рунге–Куты 4-го порядка дает наиболее точный ответ. Точность достигает 0,0001. Кроме того, следует обратить внимание на то, что ошибка (расхождение между точным и приближенным значениями) увеличивается с каждым шагом вычислений. Это обусловлено тем, что во-первых, полученное приближенное значение округляется на каждом шаге, а во-вторых – тем, что в качестве основы вычисления принимается значение, полученное на предыдущем шаге, т. е. приближенное значение. Таким образом происходит накопление ошибки.

Это хорошо видно из табл. 3.4. С каждым новым шагом приближенное значение все более отличается от точного решения.

Дифференциальные уравнения высших порядков

Дифференциальным уравнением порядка n называется уравнение вида

$$F(x, y, y', \dots, y^{(n)}) = 0.$$

В некоторых случаях это уравнение можно разрешить относительно старшей производной $y^{(n)}$:

$$y^{(n)} = f(x, y, y', \dots, y^{(n-1)}).$$

Так же как и уравнение первого порядка, уравнения высших порядков имеют бесконечное количество решений.

Решение $y = \varphi(x)$ удовлетворяет начальным условиям $x_0, y_0, y'_0, \dots, y_0^{(n-1)}$, если $\varphi(x_0) = y_0, \varphi'(x_0) = y'_0, \dots, \varphi^{(n-1)}(x_0) = y_0^{(n-1)}$.

Нахождение решения уравнения $F(x, y, y', \dots, y^{(n)}) = 0$, удовлетворяющего начальным условиям $x_0, y_0, y'_0, \dots, y_0^{(n-1)}$, называется решением задачи Коши.

Теорема Коши (теорема о необходимых и достаточных условиях существования решения задачи Коши): если функция $(n-1)$ -й переменной вида $f(x, y, y', \dots, y^{(n-1)})$ в некоторой области D $(n-1)$ -мерного пространства непрерывна и имеет непрерывные частные про-

изводные по $y, y', \dots, y^{(n-1)}$, то какова бы не была точка $(x_0, y_0, y'_0, \dots, y_0^{(n-1)})$ в этой области, существует единственное решение $y = \varphi(x)$ уравнения $y^{(n)} = f(x, y, y', \dots, y^{(n-1)})$, определенное в некотором интервале, содержащем точку x_0 , удовлетворяющее начальным условиям $x_0, y_0, y'_0, \dots, y_0^{(n-1)}$.

Линейным дифференциальным уравнением n -го порядка называется любое уравнение первой степени относительно функции y и ее производных $y', y'', \dots, y^{(n)}$ вида

$$p_0 y^{(n)} + p_1 y^{(n-1)} + p_2 y^{(n-2)} + \dots + p_{n-1} y' + p_n y = f(x),$$

где p_0, p_1, \dots, p_n – функции от x или постоянные величины, причем $p_0 \neq 0$.

Левую часть этого уравнения обозначим $L(y)$:

$$p_0 y^{(n)} + p_1 y^{(n-1)} + p_2 y^{(n-2)} + \dots + p_{n-1} y' + p_n y = L(y).$$

Если $f(x) = 0$, то уравнение $L(y) = 0$ называется *линейным однородным уравнением*, если $f(x) \neq 0$, то уравнение $L(y) = f(x)$ называется *линейным неоднородным уравнением*; если все коэффициенты $p_0, p_1, p_2, \dots, p_n$ – постоянные числа, то уравнение $L(y) = f(x)$ называется *линейным дифференциальным уравнением высшего порядка с постоянными коэффициентами*.

Отметим одно важное свойство линейных уравнений высших порядков, которое отличает их от нелинейных. Для нелинейных уравнений частный интеграл находится из общего, а для линейных – наоборот, общий интеграл составляется из частных. Линейные уравнения представляют собой наиболее изученный класс дифференциальных уравнений высших порядков. Это объясняется сравнительной простотой нахождения решения. Если при решении каких-либо практических задач требуется решить нелинейное дифференциальное уравнение, то часто применяются приближенные методы, позволяющие заменить такое уравнение «близким» к нему линейным.

Для решения дифференциальных уравнений любого порядка и систем дифференциальных уравнений, которые должны содержать только уравнения 1-го порядка, на кафедре «Тракторы» разработана подпрограмма RK4, текст (листинг) которой приведен ниже.

```

Procedure RK4(n:integer;y,y1,w:real;Var Z,V:Mas);
  Var k : integer;
  ZD:array [1..10] of real;
  KF:array [1..10,1..4] of real;
  Dk:array [1..10] of real;
  begin
  repeat
  for k:=1 to n do ZD[k]:=Z[k];
    FUN(y,V);
  for k:=1 to n do KF[k,1]:=w*V[k];
    y:=y+w/2;
    FUN(y,V);
  for k:=1 to n do KF[k,2]:=w*(V[k]+KF[k,1]/2);
    FUN(y,V);
  for k:=1 to n do KF[k,3]:=w*(V[k]+KF[k,2]/2);
    y:=y+w/2; //Отступаем на полный шаг
    FUN(y,V);
  for k:=1 to n do KF[k,4]:=w*(V[k]+KF[k,3]);
  for k:=1 to n do Dk[k]:=(KF[k,1]+2*KF[k,2]+2*KF[k,3]+KF[k,4])/6;
  for k:=1 to n do Z[k]:=ZD[k]+Dk[k];
  until (y >= y1);
  end; //RK4

```

Обращение к подпрограмме RK4 при решении уравнения 1-го порядка

$$\text{RK4}(1, tn, tp, ci, Z, V),$$

где $n = 1$ – число уравнений;

tn – начальная точка;

tp – конечная точка (точка печати);

ci – шаг интегрирования;

Z, V – массивы переменных и производных.

В головной программе шаг интегрирования h обозначается ci , а в подпрограмме RK4 – через w . Поясним назначения в головной программе переменных tn и tp , которые в подпрограмме RK4 обозначены соответственно y и $y1$:

tn – независимая переменная, которая входит в исходное дифференциальное уравнение, как x или t , и называется **начальной точкой**;

tp – переменная, которая определяет окончание вычислений в подпрограмме RK4, передает управление в головную программу для печати результатов расчета и называется **точкой печати**.

После печати результатов расчета начальная точка tn перемещается в точку печати tp , а точка печати отодвигается на **шаг печати p** :

$$tn := tp; \quad tp := tp + p.$$

Расчет заканчивается, когда точка печати достигнет конца отрезка исследования b или по другому условию, соответствующему смыслу задачи.

Сказанное проиллюстрируем рис. 3.2.

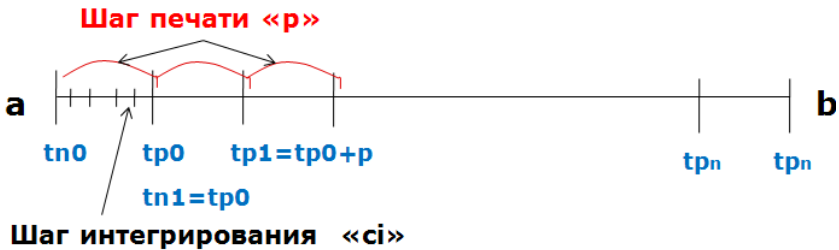


Рис. 3.2. Геометрическая интерпретация понятий «начальная точка tn » и «точка tp печати»

Решаемое дифференциальное уравнение или система уравнений записываются в подпрограмме **FUN**. Все переменные в правой части уравнений должны быть обозначены через Z с соответствующим индексом, а первые производные, стоящие в левых частях уравнений, – через V также со своим индексом.

Для решения дифференциальных уравнений более высокого порядка, чем первый, дифференциальное уравнение n -го порядка не-

обходимо свести к n дифференциальным уравнениям 1-го порядка, следовательно, для решения такого уравнения используется система дифференциальных уравнений 1-го порядка. Например, в дифференциальном уравнении второго порядка

$$\frac{d^2 y}{dx^2} = g\left(x, y, \frac{dy}{dx}\right)$$

можно принять $z = dy/dx$. Тогда $dz/dx = d^2 y/dx^2$ и получаем два уравнения первого порядка (систему двух уравнений 1-го порядка)

$$\frac{dz}{dx} = g(x, y, z); \quad \frac{dy}{dx} = f(x, y, z),$$

где $f(x, y, z) = z$.

В этом случае задача Коши содержит два начальных условия:

$$y(x_0) = y_0 \quad \text{и} \quad z(x_0) = z_0.$$

Приведем алгоритм преобразования дифференциального уравнения второго порядка в систему двух уравнений 1-го порядка, отвечающую требованиям подпрограммы RK4. Исходное уравнение

$$a\ddot{x} + b\dot{x} + cx = 0.$$

Начальные условия: при $t = 0$ $x = x_0$ и $\dot{x} = \dot{x}_0$.

1. Введем обозначения

$$x = z_1; \quad \dot{x} = \dot{z}_1 = z_2; \quad \ddot{x} = \dot{z}_2.$$

2. Решим исходное уравнение относительно старшей производной \ddot{x} :

$$\ddot{x} = -(b\dot{x} + cx) / a.$$

3. Перепишем его, используя принятые обозначения:

$$\begin{cases} \dot{z}_1 = z_2; \\ \dot{z}_2 = -(bz_2 + cz_1) / a. \end{cases}$$

Получили систему двух дифференциальных уравнений 1-го порядка, почти отвечающую требованиям подпрограммы RK4 (все переменные в правых частях уравнений обозначены через Z с соответствующим индексом). Остается заменить 1-е производные на V . Для этого переписываем полученную систему, заменяя 1-е производные на V , сохраняя их индекс:

$$\begin{cases} V[1] := Z[2]; \\ V[2] := -(bZ[2] + cZ[1]) / a. \end{cases}$$

Получили систему уравнений, не только полностью отвечающую требованиям подпрограммы RK4, но и текст подпрограммы **FUN**, в которой описывается решаемая система дифференциальных уравнений.

Остается записать начальные условия в принятых обозначениях:

$$Z[1] := x_0; \quad Z[2] := \dot{x}_0.$$

Таким образом получена система из двух уравнений 1-го порядка, полностью отвечающая требованиям подпрограммы **RK4**, и одновременно текст подпрограммы **Fun**, в которой описывается решаемое исходное дифференциальное уравнение второго порядка, преобразованное в систему двух уравнений первого порядка.

Численные методы решения дифференциальных уравнений в форме задачи Коши в среде MathCAD

Основная форма задания исходных условий, которая используется при моделировании реальных процессов с помощью дифференциальных уравнений, связана с определением значений всех низших производных в начальной точке интервала изменения переменной. Так,

в технике дифференциальные уравнения обычно описывают изменение изучаемой характеристики с течением времени и начальные условия определяются в момент $t = 0$. Заданные таким образом дифференциальные уравнения называются **з а д а ч а м и К о ш и**.

Методы численного решения дифференциальных уравнений в форме задачи Коши разработаны достаточно полно. Самыми популярными из них заслуженно являются алгоритмы Рунге–Кутта, успешно используемые для решения подавляющего большинства дифференциальных уравнений. Однако для некоторых задач они неприменимы (или неэффективны), и тогда приходится применять более тонкие и сложные методы. Функции, реализующие наиболее важные из них, в MathCAD являются системными, что позволяет очень быстро и просто находить решение практически для любого дифференциального уравнения.

Решение линейных дифференциальных уравнений и их систем реализовано в MathCAD в двух формах: в виде вычислительного блока и в виде встроенных функций. Первая форма предпочтительнее с точки зрения наглядности представления решения и технической простоты, вторая же открывает куда более широкие возможности для влияния на ход решения дифференциального уравнения и изучения полученных результатов. Так как, вне зависимости от выбора типа оформления, для решения дифференциальных уравнений система использует одни и те же алгоритмы, то принципиального различия от того, какой из них используется, в результатах вычислений не будет.

Вычислительный блок Given-Odesolve

Самый простой и наглядный способ численного решения линейных дифференциальных уравнений и систем линейных дифференциальных уравнений в MathCAD реализуется с использованием вычислительного блока ***Given-Odesolve***. Чтобы его реализовать, необходимо выполнить следующую последовательность действий:

- задать вводное слово ***Given*** (Дано);
- ниже вводного слова определить вид дифференциального уравнения или системы уравнений. Сделать это нужно, соблюдая следующие правила:

- а) дифференциальное уравнение (в том числе входящее в систему) должно быть строго линейным (т. е. высшая производная в нем

не должна иметь никаких сомножителей или степенных показателей). Подобное ограничение связано с особенностями используемого (по умолчанию) системой при решении дифференциальных уравнений в форме вычислительного блока *Given-Odesolve* метода Рунге–Кутты 4-го порядка. **Если же уравнение нелинейное, то использовать вычислительный блок нельзя,**

б) производные в выражении дифференциального уравнения могут быть заданы с помощью специальных операторов панели *Calculus* (Вычисления) (**Shift+/'** для оператора производной первого порядка и **Shift+Ctrl+/'** для оператора дифференцирования произвольного порядка). Подобная запись дифференциальных уравнений принята в физике, а в математике традиционно производные в дифференциальных уравнениях обозначаются с помощью специальных меток-штрихов. При задании вычислительного блока можно использовать и такую форму отображения дифференциальных уравнений (соответствующий штрих вводится с помощью сочетания **Ctrl+F7**),

в) искомые функции должны быть определены явно с указанием имени переменной, например $f(x)$,

г) в качестве знака равенства внутри вычислительного блока следует использовать только логическое равенство (**Bold Equal–Ctrl+=**);

– помимо самого дифференциального уравнения внутри вычислительного блока необходимо задать и соответствующие начальные условия. В курсе высшей математики доказывается, что для того чтобы получить явное решение дифференциального уравнения n -го порядка, для него следует определить n начальных условий. В случае задачи Коши в качестве начальных условий необходимо задать значения производных порядков $n - 1, n - 2, \dots, 0$ на левой границе интервала изменения переменной. При определении производных в начальной точке используются те же правила, что и при задании выражения самого дифференциального уравнения;

– когда дифференциальные уравнения и начальные условия будут заданы, можно непосредственно приступить к вычислению значений искомой функции на нужном промежутке. Для этого следует задействовать специальную функцию $odesolve([\text{vector}],t,b,[\text{step}])$, где

▪ **vector** – вектор функций, относительно которых решается система дифференциальных уравнений. Вектор должен содержать имена функций без имени переменной. Данный параметр использу-

ется, когда решается система ОДУ, в случае же решения одиночного уравнения его следует опустить,

- t – переменная, от которой зависит искомая функция. Необходимость задания этого параметра связана с тем, что результат использования функции *odesolve* также является функцией, а не вектором некоторых числовых значений;

- b – правая граница интервала поиска решения. Соответственно левая граница определяется при задании начальных условий. Обязательным условием является то, что b должно лежать правее a – это **общее требование для корректно заданной задачи Коши при использовании всех численных методов**,

- **step** – параметр определяет количество шагов, используемых численным методом Рунге–Кутты 4-го порядка. Задание **step** не является обязательным, и по умолчанию этот параметр определяется таким образом, чтобы длина шага была равна 0,1. В случае сложных уравнений правильное определение **step** может играть самую принципиальную роль для получения корректного решения. Однако стремиться во всех случаях сделать **step** минимальным не стоит, так как для большинства дифференциальных уравнений это приведет лишь к резкому увеличению времени расчетов без значимого повышения их точности.

Наряду с дифференциальными уравнениями в блоке *Given-Odesolve* можно задавать и алгебраические ограничения, например, $y(t) + z(t) = w(t)$, что позволяет ввести в систему дополнительную неизвестную функцию $w(t)$, которая должна быть определена *odesolve* наряду с $y(t)$ и $z(t)$. **Ограничения в виде неравенств задавать нельзя.**

Приведем примеры решения дифференциальных уравнений первого и второго порядков. Выберем уравнения таким образом, чтобы для них не было известно аналитическое решение.

Пример 3.6

Решить дифференциальное уравнение первого порядка с помощью пакета MathCAD.

Решение

В MathCAD набираем требуемую информацию и после обращения к функции *odesolve* получаем решение заданного уравнения в виде графика, рис. 3.3.

Given

$$\frac{d}{dt}y(t) = \left(y(t) - y(t)^2 + t\right) \cdot \sin\left(t + \frac{1}{y(t)}\right) \cdot \cos\left(t - \frac{1}{y(t)}\right)$$

$$y(0) = 1$$

y := Odesolve(t, 40)

$$y(6.11115) = 2.593072385234662 \quad y(39.89679) = 6.661252824787725$$

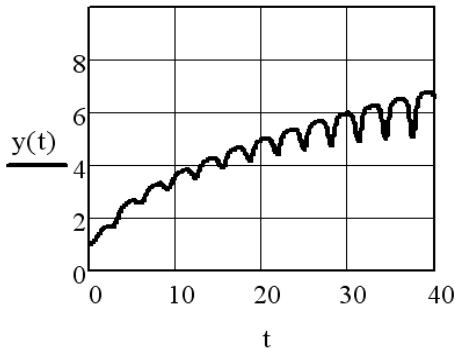


Рис. 3.3. Результат численного решения дифференциального уравнения первого порядка

Ответ. В результате решения заданного уравнения получено его графическое решение.

Пример 3.7

Решить дифференциальное уравнение второго порядка с помощью пакета MathCAD.

Решение

В MathCAD набираем требуемую информацию и после обращения к функции *odesolve* получаем решение заданного уравнения в виде графика, рис. 3.4.

Given

$$\frac{d^2}{dt^2}x(t) + x(t) = \sin(2t)$$

$$x(0) = 0 \quad x'(0) = 0$$

`x := Odesolve(t,10)`

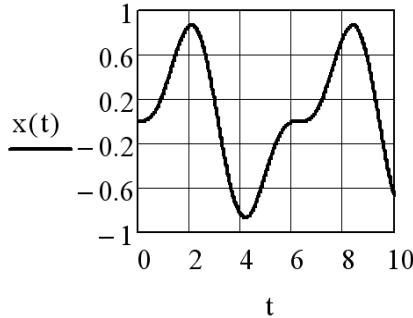


Рис. 3.4. Результат численного решения дифференциального уравнения второго порядка

Ответ. В результате решения заданного уравнения получено его графическое решение.

Следует отметить, что при решении дифференциального уравнения с помощью функции *odesolve* существует возможность определения значений искомой функции и в тех точках, которые не являются узловыми (т. е. непосредственно в ходе работы численного алгоритма эти значения просчитаны не были). Достигается же это за счет задания между каждой парой точек интерполирующего полинома (точнее, кубического сплайна). Очевидно, что точность такого предсказания будет тем выше, чем меньше величина шага и чем более плавно и предсказуемо изменяется функция решения.

График решения дифференциального уравнения вычислительным блоком *Given-Odesolve* задается таким образом, как будто оно является непрерывной функцией (см. рис. 3.3, 3.4).

Аналогичным образом, как и дифференциальные уравнения первого и второго порядков, в MathCAD решаются и линейные дифференциальные уравнения, содержащие производные более высокого порядка.

Пример 3.8

Решить дифференциальное уравнение третьего порядка с помощью пакета MathCAD.

Решение

В MathCAD набираем требуемую информацию и после обращения к функции *odesolve* получаем решение заданного уравнения в виде графика, рис. 3.5.

Given

$$y'''(t) + y''(t) + y'(t) + y(t) = -t \cdot \sin(t)$$

$$y''(0) = 2 \quad y'(0) = 1 \quad y(0) = 0$$

$$y := \text{Odesolve}(t, 30, 30000)$$

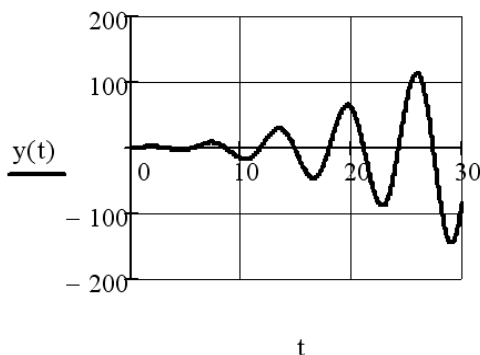


Рис. 3.5. График численного решения дифференциального уравнения третьего порядка

Ответ. В результате решения заданного уравнения получено его графическое решение.

Важным условием успеха при численном решении дифференциальных уравнений является правильный выбор величины шага. Так, если он будет определен недостаточно малым, то найденное решение может быть весьма и весьма далеким от истинного решения. В тех случаях, когда возникают сомнения в верности заданной длины шага, можно руководствоваться следующим прос-

тым правилом: *ее можно принять как корректную, если уменьшение ее величины в 10 раз не приводит к изменению результата в некоторой точке в пределах количества десятичных знаков, отвечающего нужному уровню точности.*

По умолчанию вычислительный блок **Given-Odesolve** использует популярный метод Рунге–Кутта 4-го порядка с постоянным шагом. Однако MathCAD позволяет при необходимости сменить его на так называемый адаптивный алгоритм (*adaptive*) или алгоритм решения жестких систем (*stiff*). Чтобы это сделать, необходимо щелкнуть правой кнопкой мыши на функции **odesolve**. В открывшемся при этом контекстном меню необходимо переместить флажок из строки **Fixed** (фиксированный) в строку **Adaptive** (адаптивный) или **Stiff** (жесткий). Отличие адаптивного метода от простого алгоритма Рунге–Кутта 4-го порядка состоит в том, что используемая им длина шага не постоянна, а зависит от скорости изменения функции результата (что определяется, в частности, величиной первой производной). Такой подход может быть более эффективен в случае не слишком хорошо подобранного шага, а также для решения сложных дифференциальных уравнений с быстро изменяющимися и осциллирующими функциями решений. Если же нужно решить жесткую систему уравнений или систему, содержащую алгебраические ограничения, следует воспользоваться методом **Stiff** (рис. 3.6).



Рис. 3.6. Смена используемого численного метода

Решение дифференциальных уравнений с помощью встроенных функций

В более ранних версиях Mathcad для решения дифференциальных уравнений использовались специальные встроенные функции. Однако с появлением более наглядного и простого способа, связанного с применением вычислительного блока *Given-Odesolve*, они уже не представляют особой практической важности. Несмотря на это, о возможности их применения для решения одинарного дифференциального уравнения нужно иметь четкое представление, так как в некоторых случаях (например, в программировании) использовать вычислительный блок нельзя.

Решения одного дифференциального уравнения с использованием соответствующих встроенных функций рассмотрим на примере.

Пример 3.9

Дифференциальное уравнение решить альтернативным методом с помощью пакета MathCAD.

Решение

В MathCAD набираем требуемую информацию и после обращения к функции *rkfixed* получаем решение заданного уравнения в виде таблицы результатов и графика, рис. 3.7.

```
y0 := 1           M := 100
D(t,y) := sin(t) · y
x1 := 0           x2 := 10
y1 := rkfixed(y0 ,x1 ,x2 ,M,D)   y2 := Bulstoer(y0 ,x1 ,x2 ,M,D)
y3 := Rkadapt(y0 ,x1 ,x2 ,M,D)
```

	0	1
0	0	1
1	0.1	1.005
2	0.2	1.02
3	0.3	1.046
4	0.4	1.082
5	0.5	1.13
6	0.6	1.191
$y^1 = 7$	0.7	1.265
8	0.8	1.354
9	0.9	1.46
10	1	1.584
11	1.1	1.727
12	1.2	1.892
13	1.3	2.08
14	1.4	2.293
15	1.5	...

	0	1
0	0	1
1	0.1	1.005
2	0.2	1.02
3	0.3	1.046
4	0.4	1.082
5	0.5	1.13
6	0.6	1.191
$y^2 = 7$	0.7	1.265
8	0.8	1.354
9	0.9	1.46
10	1	1.584
11	1.1	1.727
12	1.2	1.892
13	1.3	2.08
14	1.4	2.293
15	1.5	...

	0	1
0	0	1
1	0.1	1.005
2	0.2	1.02
3	0.3	1.046
4	0.4	1.082
5	0.5	1.13
6	0.6	1.191
$y^3 = 7$	0.7	1.265
8	0.8	1.354
9	0.9	1.46
10	1	1.584
11	1.1	1.727
12	1.2	1.892
13	1.3	2.08
14	1.4	2.293
15	1.5	...

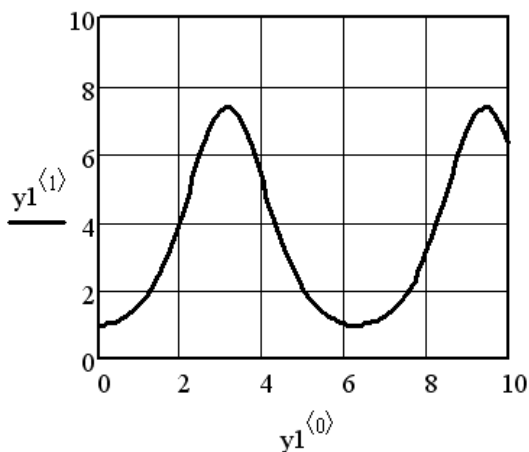


Рис. 3.7. График решения дифференциального уравнения, полученного с помощью встроенной функции *rkfixed*

Ответ. В результате решения заданных уравнений получены таблицы решений и их графические зависимости.

Применять приведенные встроенные функции можно лишь для решения дифференциальных уравнений первого порядка, как линейных, так и нелинейных. Найти же с их помощью решение уравнения более высокого порядка, в отличие от использования блока *Given-Odesolve*, непосредственным образом невозможно. Это можно сделать, сведя указанное уравнение к системе дифференциальных уравнений.

Приведем пример решения дифференциального уравнения второго порядка с помощью вычислительного блока и встроенной функции *rkfixed* (реализующей метод Рунге–Кутты четвертого порядка с фиксированным шагом).

Пример 3.10

Дифференциальное уравнение второго порядка решить с помощью пакета MathCAD альтернативным способом.

Решение

В MathCAD набираем требуемую информацию и после обращения к функциям *odesolve* и *rkfixed* получаем решение заданного уравнения в виде графика, рис. 3.8.

Вычислительный блок *Given-Odesolve*

Given

$$y''(t) + y'(t) = 2y(t)$$

$$y(0) = 1 \quad y'(0) = 3$$

$$z := \text{Odesolve}(t, 5, 100)$$

Встроенная функция *rkfixed*

$$y := \begin{pmatrix} 1 \\ 3 \end{pmatrix} \quad D(t, y) := \begin{pmatrix} y_1 \\ -y_1 + 2 \cdot y_0 \end{pmatrix}$$

$$r := \text{rkfixed}(y, 0, 5, 100, D)$$

Однако в Mathcad имеются специальные встроенные функции, позволяющие находить решения как линейных, так и нелинейных систем дифференциальных уравнений. Всего таких функций три:

1. ***Rkfixed*** (y_0, t_0, t_1, M, D). Задействовав эту системную функцию, можно решить задачу Коши с помощью самого популярного из численных алгоритмов: метода Рунге–Кутта 4-го порядка с фиксированным шагом. Эта функция подходит для качественного и быстрого решения подавляющего большинства систем ОДУ;

2. ***Rkadapt*** (y_0, t_0, t_1, M, D). Эта функция реализует адаптивный алгоритм Рунге–Кутта. Ее принципиальным отличием от ***rkfixed*** является то, что при вычислении соответствующих приближений она использует не постоянный, а зависящий от скорости изменения функций решения шаг дискретизации переменной. Однако, несмотря на это, в качестве ответа ***Rkadapt*** возвращает значения функций в равномерно распределенных, исходя из заданной пользователем величины промежутка, точках интервала. Использовать же рассматриваемую функцию следует в случае более жестких и сложных систем уравнений. Это позволит повысить точность расчета или сэкономить время по сравнению с применением простого метода Рунге–Кутта;

3. ***Bulstoer*** (y_0, t_0, t_1, M, D) – метод Булирша–Штера. Использовать этот алгоритм стоит в том случае, если есть уверенность, что функции решения системы достаточно гладкие и плавно изменяющиеся. При выполнении этого условия функция ***Bulstoer*** позволяет получать более точные решения, чем ***rkfixed***, затрачивая на это меньше времени.

Для корректного использования описанных выше функций система дифференциальных уравнений должна быть записана в векторном виде:

$$Y'(x) = D(Y(x), x),$$

где $Y(x)$ – вектор первых производных системы;

$D(Y(x), x)$ – вектор-функция, каждая строка которой содержит левую часть соответствующего уравнения системы.

Кроме того, в векторной форме должны быть определены начальные условия: $Y(x_0) = y_0$.

Разобравшись с принципом приведения систем дифференциальных уравнений к векторной форме, можно задать и параметры для рассматриваемых встроенных функций:

➤ y_0 – вектор начальных условий; в нем определяются числовые значения искомых функций на левой границе интервала изменения переменной;

➤ t_0 – начальная точка для переменной;

➤ t_1 – конечная точка расчета;

➤ M – количество шагов, при котором численный метод будет решать систему;

➤ D – вектор-функция, содержащая правые части уравнений системы, которая должна быть задана как функция двух переменных: скаляра t и вектора y (т. е. все искомые функции системы должны быть представлены как элементы одного вектора).

Результатом работы приведенных функций является матрица, в первом столбце которой содержатся узловые величины переменной t , а в остальных – значения неизвестных функций системы, рассчитанные в этих точках. При этом порядок расположения столбцов с найденными величинами искомых функций определяется последовательностью, в которой они были занесены в вектор y .

Пример 3.11

Систему дифференциальных уравнений решить с использованием вычислительного блока *Given-Odesolve* с помощью пакета MathCAD.

Решение

В MathCAD набираем требуемую информацию и после обращения к функции *odesolve* получаем решение заданного уравнения в виде графика, рис. 3.9.

Given

$$\frac{d}{dt}y(t) = y(t) - z(t) + \exp(t)$$

$$\frac{d}{dt}z(t) = 2y(t) + z(t) + \exp(3t)$$

$$y(0) = 0 \quad z(0) = 0$$

$$\begin{pmatrix} y \\ z \end{pmatrix} := \text{Odesolve} \left[\begin{pmatrix} y \\ z \end{pmatrix}, t, 1 \right]$$

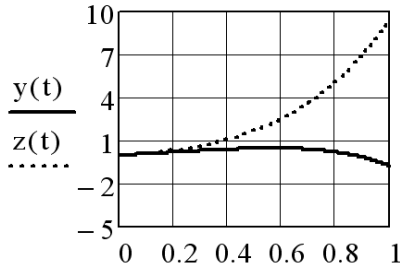


Рис. 3.9. Результат решения системы дифференциальных уравнений с помощью вычислительного блока *Given-Odesolve*

Ответ. В результате решения заданного уравнения получены его графические зависимости.

Задача о нахождении решения дифференциального уравнения $F(x, y, y', y'', \dots, y^{(k)}, \dots, y^{(n)}) = 0$, удовлетворяющего начальным условиям

$$y(x_0) = y_0, y'(x_0) = y'_0, \dots, y^{(n-1)}(x_0) = y_0^{(n-1)},$$

где $y_0, y'_0, \dots, y_0^{(n-1)}$ – заданные числа,

называется задачей Коши для системы дифференциальных уравнений.

Дифференциальное уравнение n -го порядка заменой:

$$y' = z_1;$$

$$y'' = z_2;$$

...

$$y^{(n-1)} = z_{(n-1)}$$

сводится к системе n дифференциальных уравнений первого порядка:

$$y' = z_1;$$

$$y'' = z_2;$$

...

$$y^{(n-1)} = z_{(n-1)}.$$

Например, дифференциальное уравнение второго порядка

$$y'' = -\omega^2 y$$

можно записать в виде двух уравнений:

$$y' = z;$$

$$z' = -\omega^2 y.$$

Программная реализация решения данной системы дифференциальных уравнений в среде MathCAD имеет следующий вид:

Given

$$\frac{d}{dt}y(t) = z(t)$$

$$\frac{d}{dt}z(t) + y(t) = \sin(2t)$$

$$y(0) = 0 \quad z(0) = 0$$

$$\begin{pmatrix} y \\ z \end{pmatrix} := \text{Odesolve} \left[\begin{pmatrix} y \\ z \end{pmatrix}, t, 10 \right]$$

Решение данной системы дифференциальных уравнений представлено на рис. 3.10.

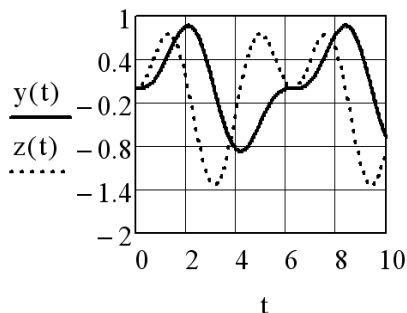


Рис. 3.10. Решение дифференциального уравнения $y'' = -\omega^2 y$ путем сведения его к системе уравнений

Пример 3.12

Дифференциальное уравнение

$$\frac{d^2}{dt^2} y(t) + \beta \frac{d}{dt} y(t) + y(t) = 0$$

решить с помощью пакета MathCAD путем сведения к системе дифференциальных уравнений при помощи подстановки $y_0(t) = y(t)$,

$y_1(t) = \frac{d}{dt} y(t) = \frac{d}{dt} y_0(t)$ с использованием встроенной функции *rkfixed*.

Решение

После применения к исходному уравнению второго порядка указанной подстановки искомая система уравнений примет вид

$$\begin{cases} \frac{d}{dt} y_0(t) = y_1(t); \\ \frac{d}{dt} y_1(t) = -\beta \frac{d}{dt} y_0(t) - y_0(t). \end{cases}$$

В MathCAD набираем требуемую информацию и после обращения к функции *rkfixed* получаем решение заданного уравнения в виде таблицы и графика, рис. 3.11.

$$\beta := 0.2 \quad M := 50$$

$$y_0 := \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$D(t,y) := \begin{pmatrix} y_1 \\ -\beta \cdot y_1 - y_0 \end{pmatrix}$$

$$u := \text{rkfixed}(y_0, 0, 40, M, D)$$

$u =$

	0	1	2
0	0	1	0
1	0.8	0.713	-0.661
2	1.6	0.072	-0.856
3	2.4	-0.514	-0.545
4	3.2	-0.727	0.023
5	4	-0.504	0.493
6	4.8	-0.033	0.619
7	5.6	0.386	0.382
8	6.4	0.528	-0.033
9	7.2	0.355	-0.368
10	8	0.01	-0.448
11	8.8	-0.289	-0.267
12	9.6	-0.383	0.035
13	10.4	-0.25	0.273
14	11.2	$2.616 \cdot 10^{-3}$	0.324
15	12	0.216	...

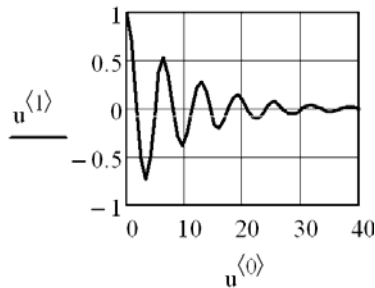


Рис. 3.11. Решение дифференциального уравнения $\frac{d^2}{dt^2} y(t) + \beta \frac{d}{dt} y(t) + y(t) = 0$

с использованием встроенной функции *rkfixed*

Ответ. В результате решения заданного уравнения получены таблица результатов и графическая зависимость.

Далее наглядно представим последовательность действий метода Рунге–Кутты 4-го порядка путем создания программы, реализующей данный метод в среде MathCAD. Программная реализация метода Рунге–Кутты 4-го порядка в среде MathCAD представлена на рис. 3.12.

$$f(t,y) := t^2 - y$$

$$\text{Dif}(f, a, b, y_0, M) := \left(\begin{array}{l} h \leftarrow \frac{b-a}{M} \quad y_0 \leftarrow y_0 \quad t \leftarrow a \\ \text{for } i \in 0..M \\ \quad k1 \leftarrow f(t, y_i) \\ \quad k2 \leftarrow f\left(t + \frac{h}{2}, y_i + \frac{h}{2} \cdot k1\right) \\ \quad k3 \leftarrow f\left(t + \frac{h}{2}, y_i + \frac{h}{2} \cdot k2\right) \\ \quad k4 \leftarrow f(t+h, y_i + h \cdot k3) \\ \quad y_{i+1} \leftarrow y_i + h \cdot \frac{(k1 + 2 \cdot k2 + 2 \cdot k3 + k4)}{6} \quad \text{if } i \leq M-1 \\ \quad \left(\begin{array}{l} T_i \leftarrow t \quad t \leftarrow t+h \end{array} \right) \\ \end{array} \right) \\ \vee \leftarrow \left(\begin{array}{l} T \\ y \end{array} \right)$$

$$D := \text{Dif}(f, 0, 10, 1, 100)$$

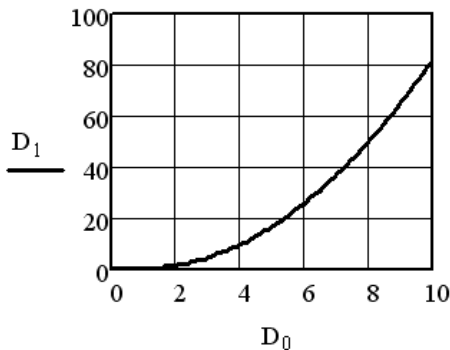


Рис. 3.12. Программная реализация метода Рунге–Кутты 4-го порядка в среде MathCAD

В корректности работы программы легко убедиться, сравнив решения, полученные с помощью встроенных функций и представленного алгоритма.

На каждом шаге алгоритм аппроксимирует дифференциальное уравнение по приведенной схеме, поэтому попытаемся проанализировать, какие факторы влияют на точность полученных с его помощью результатов. Очевидно, возникает ошибка приближения искомой функции параболой при использовании формулы Симпсона, а также погрешность усреднения значений k_2 и k_3 в центральной точке подынтервала. Такой подход к оценке погрешности можно назвать геометрическим. С аналитической точки зрения метод Рунге–Кутта 4-го порядка для нахождения решения использует только первые пять слагаемых, полученных при разложении искомой функции в ряд Тейлора в окрестности заданной точки. Если на отдельном шаге погрешность пренебрежимо мала, то на интервале происходит накопление ошибки, что отражается на точности результата. Поэтому важно выбрать оптимальное соотношение между длиной шага и их количеством, при котором погрешность аппроксимации окажется минимальной, а время расчета приемлемым. Как правило, с уменьшением длины шага точность возрастает до определенной величины, которая не меняется при дальнейшем разбиении интервала. Проиллюстрируем сказанное.

В табл. 3.5 приведены истинные решения задачи Коши $y'(t) = t^2 - y$, $y(0) = 1$ на интервале $[0; 10]$, а также рассчитанные алгоритмом (см. рис. 3.12) при различной длине шага.

Таблица 3.5

Решения задачи Коши $y'(t) = t^2 - y$, $y(0) = 1$ на интервале $[0; 10]$
методом Рунге–Кутта 4-го порядка

t_i	y_i			Истинное решение	Ошибка		
	$h = 1$	$h = 0,5$	$h = 0,1$		$h = 1$	$h = 0,5$	$h = 0,1$
0	1,0000	1,0000	1,0000	1,0000	0,0000	0,0000	0,0000
1	0,6458	0,6329	0,6321	0,6321	0,0137	0,0008	0,0000
2	1,8880	1,8659	1,8647	1,8647	0,0233	0,0012	0,0000
3	4,9788	4,9517	4,9502	4,9502	0,0286	0,0015	0,0000
4	10,0129	9,9833	9,9817	9,9817	0,0312	0,0016	0,0000
5	17,0257	16,9949	16,9933	16,9933	0,0324	0,0016	0,0000

t_i	y_i			Истинное решение	Ошибка		
	$h = 1$	$h = 0,5$	$h = 0,1$		$h = 1$	$h = 0,5$	$h = 0,1$
6	26,0305	25,9992	25,9975	25,9975	0,0330	0,0017	0,0000
7	37,0323	37,0007	36,9991	36,9991	0,0332	0,0016	0,0000
8	50,0329	50,0013	49,9997	49,9997	0,0332	0,0016	0,0000
9	65,0332	65,0015	64,9999	64,9999	0,03330	0,0016	0,0000
10	82,0333	82,0016	82,0000	82,0000	0,0333	0,0016	0,0000

Как видно из табл. 3.5, при длине шага, равной 0,1 (используемой системой по умолчанию), решения оказываются точными до четвертого знака. Уменьшение длины шага до 0,01 увеличивает точность до седьмого знака. Подобная закономерность реализуется далеко не для всех функций. Выбор желаемого уровня точности остается за пользователем, но зачастую оценить величину ошибки при той или иной длине шага просто невозможно. В этом случае альтернативным оказывается метод Рунге–Кутты с переменным шагом (адаптивный метод), лежащий в основе работы функций *rkadapt* и *Rkadapt*. В каждой узловой точке интервала этот алгоритм находит два варианта решения, используя схемы Рунге–Кутты 4-го и 5-го порядков. Если разность между полученными значениями оказывается меньше допустимой величины ошибки (которую можно заранее задать через системную переменную TOL), то длина шага увеличивается. Если же разность превысит ошибку, то следующую точку интервала алгоритм определит с меньшей длиной шага. В случае равенства обоих параметров шаг остается постоянным. Адаптивный метод целесообразно использовать для нахождения неоднородных решений, меняющихся с различной скоростью на том или ином участке интервала. Это позволит сократить время расчета, поскольку для некоторых функций хороший результат может быть получен и при использовании алгоритмом небольшого количества шагов.

Варианты заданий

Решить задачу Коши для заданных дифференциальных уравнений $y' = f(x, y)$ на отрезке $[a, b]$ при заданном начальном условии $y(a) = y_0$ и шаге интегрирования h .

№	$f(x, y)$	$[a, b]$	y_0	h
1	$3x^2 + 0,1xy$	$[0; 1]$	$y(0) = 0,2$	0,1
2	$0,185(x^2 + \cos(0,7x)) + 1,843y$	$[0,2; 1,2]$	$y(0,2) = 0,25$	0,1
3	$x + \cos\left(\frac{y}{3}\right)$	$[1,6; 2,6]$	$y(1,6) = 4,6$	0,1
4	$x + \sin\left(\frac{y+1}{\sqrt{13}}\right)$	$[0,2; 1,2]$	$y(0,2) = 2,5$	0,1
5	$x + \cos\left(\frac{y}{e}\right)$	$[1,4; 2,4]$	$y(1,4) = 2,5$	0,1
6	$x^2 + \cos\left(\frac{y}{\pi}\right)$	$[1,7; 2,7]$	$y(1,7) = 5,7$	0,1
7	$-3y + \sqrt{4x^2 + 1}$	$[2,6; 4,6]$	$y(2,6) = 3,5$	0,2
8	$2 - \sin^2(x + y)$	$[2; 3]$	$y(2) = 2,3$	0,1
9	$1,6x + 0,5y^2$	$[0; 1]$	$y(0) = 0,3$	0,1
10	$x - \cos\left(\frac{y}{\sqrt{5}}\right)$	$[1,8; 2,8]$	$y(1,8) = 2,6$	0,1
11	$x + \cos\left(\frac{y}{\sqrt{5}}\right)$	$[2,1; 3,1]$	$y(2,1) = 2,5$	0,1
12	$e^{2x} + 0,25y^2$	$[0; 0,5]$	$y(0) = 2,6$	0,05
13	$x + \cos\left(\frac{y}{\sqrt{2}}\right)$	$[-2; -1]$	$y(-2) = 3$	0,1
14	$0,133(x^2 + \sin(2x)) + 0,872y$	$[0,2; 1,2]$	$y(0,2) = 0,25$	0,1
15	$\sin(x + y) + 1,5$	$[1,5; 2,5]$	$y(1,5) = 4,5$	0,1
16	$x + \cos\left(\frac{y}{1,25}\right)$	$[0,4; 1,4]$	$y(0,4) = 0,8$	0,1
17	$2,5x + \cos(y + 0,6)$	$[1; 3]$	$y(1) = 1,5$	0,2

№	$f(x, y)$	$[a, b]$	y_0	h
18	$\cos(1,5y + x)^2 + 1,4$	$[1; 2]$	$y(1) = 1,5$	0,1
19	$\frac{1}{1+x^3y} + 2y$	$[1,5; 2]$	$y(1,5) = 2,1$	0,05
20	$\cos y + 3x$	$[0; 2]$	$y(0) = 1,3$	0,1

Варианты заданий

Решить задачу Коши для заданной системы дифференциальных уравнений $\begin{cases} x' = f(x, y) \\ y' = f(x, y) \end{cases}$ на отрезке $[a, b]$ при заданных начальных условиях $x(a) = x_0, y(a) = y_0$ и шаге интегрирования h .

№	$f(x, y)$	$[a, b]$	y_0	h
1	$\begin{cases} \frac{dx}{dt} = 2x + y \\ \frac{dy}{dt} = 3x + 4y \end{cases}$	$[0; 1]$	$x(0) = 0$ $y(0,1) = 0,5$	0,1
2	$\begin{cases} \frac{dx}{dt} = x - y \\ \frac{dy}{dt} = y - 4x \end{cases}$	$[0; 1,5]$	$x(0) = 0$ $y(0,3) = 0,7$	0,1
3	$\begin{cases} \frac{dx}{dt} = -x + 8y \\ \frac{dy}{dt} = x + y \end{cases}$	$[0; 1,4]$	$y(0) = 0$ $x(0,7) = 2,5$	0,05
4	$\begin{cases} \frac{dx}{dt} = x + y \\ \frac{dy}{dt} = 3y - 2x \end{cases}$	$[0; 1,1]$	$y(0) = 0$ $x(0,9) = 1,4$	0,1

№	$f(x, y)$	$[a, b]$	y_0	h
5	$\begin{cases} \frac{dx}{dt} = x - 3y \\ \frac{dy}{dt} = 3x + y \end{cases}$	[0; 1,5]	$y(0) = 0$ $x(0,1) = 1,7$	0,1
6	$\begin{cases} \frac{dx}{dt} = -x - 5y \\ \frac{dy}{dt} = x + y \end{cases}$	[0,5; 1,8]	$y(0,5) = 0$ $x(0,5) = 2$	0,1
7	$\begin{cases} \frac{dx}{dt} = 2x + y \\ \frac{dy}{dt} = 4y - x \end{cases}$	[0,2; 1,6]	$x(0,2) = 0$ $y(0,2) = 0,7$	0,2
8	$\begin{cases} \frac{dx}{dt} = 3x - y \\ \frac{dy}{dt} = 4x - y \end{cases}$	[0,2; 3]	$x(0,2) = 0$ $y(2) = 1,7$	0,1
9	$\begin{cases} \frac{dx}{dt} = 2y - 3x \\ \frac{dy}{dt} = y + 2x \end{cases}$	[0,4; 1,1]	$y(1) = 0$ $x(0,4) = 2,2$	0,1
10	$\begin{cases} \frac{dx}{dt} = 5x + 3y \\ \frac{dy}{dt} = -3x - y \end{cases}$	[1,8; 2,8]	$y(2) = 0$ $x(1,8) = 1,9$	0,1
11	$\begin{cases} \frac{dx}{dt} = x + z - y \\ \frac{dy}{dt} = x + y - z \\ \frac{dz}{dt} = 2x - y \end{cases}$	[1; 2]	$y(0) = 0$ $x(0) = 0,9$ $z(0,9) = 3,4$	0,2

№	$f(x, y)$	$[a, b]$	y_0	h
12	$\begin{cases} \frac{dx}{dt} = x - 2y - z \\ \frac{dy}{dt} = y - x + z \\ \frac{dz}{dt} = x - z \end{cases}$	[0; 2]	$\begin{aligned} y(0) &= 0 \\ z(2) &= 0,8 \\ x(0) &= 3 \end{aligned}$	0,2
13	$\begin{cases} \frac{dx}{dt} = 2x - y + z \\ \frac{dy}{dt} = x + 2y - z \\ \frac{dz}{dt} = x - y + 2z \end{cases}$	[0,7; 2,1]	$\begin{aligned} x(0) &= 0 \\ z(0) &= 0 \\ y(0,7) &= 1 \end{aligned}$	0,1
14	$\begin{cases} \frac{dx}{dt} = 3x - y + z \\ \frac{dy}{dt} = x + y + z \\ \frac{dz}{dt} = 4x - y + 4z \end{cases}$	[0,7; 1;7]	$\begin{aligned} y(0) &= 0 \\ z(0) &= 3 \\ x(0,7) &= 0,1 \end{aligned}$	0,05
15	$\begin{cases} \frac{dx}{dt} = 4y - 2x - 3z \\ \frac{dy}{dt} = z + x \\ \frac{dz}{dt} = 6x - 6y + 5z \end{cases}$	[1,5; 3]	$\begin{aligned} y(0) &= 0 \\ x(0) &= 0 \\ z(1,5) &= 0,5 \end{aligned}$	0,2
16	$\begin{cases} \frac{dx}{dt} = x - y - z \\ \frac{dy}{dt} = x + y \\ \frac{dz}{dt} = 3x + z \end{cases}$	[0,3; 1,2]	$\begin{aligned} z(0) &= 0,5 \\ x(1) &= 0 \\ y(0) &= 0,8 \end{aligned}$	0,1

№	$f(x, y)$	$[a, b]$	y_0	h
17	$\begin{cases} \frac{dx}{dt} = 2x + y \\ \frac{dy}{dt} = x + 3y - z \\ \frac{dz}{dt} = 2y + 3z - x \end{cases}$	$[0,5; 2]$	$\begin{cases} z(0) = 9 \\ y(0,1) = 0 \\ x(0) = 0,1 \end{cases}$	0,2
18	$\begin{cases} \frac{dx}{dt} = 2x + 2z - y \\ \frac{dy}{dt} = x + 2z \\ \frac{dz}{dt} = y - 2x - z \end{cases}$	$[0,7; 1,6]$	$\begin{cases} z(0) = 9 \\ y(0) = 0 \\ x(1,6) = 4 \end{cases}$	0,05
19	$\begin{cases} \frac{dx}{dt} = 4x - y - z \\ \frac{dy}{dt} = x + 2y - z \\ \frac{dz}{dt} = x - y + 2z \end{cases}$	$[0,2; 0,9]$	$\begin{cases} z(0) = 0 \\ x(0) = 0 \\ y(0,2) = 1,3 \end{cases}$	0,05
20	$\begin{cases} \frac{dx}{dt} = 2x - y - z \\ \frac{dy}{dt} = 3x - 2y - 3z \\ \frac{dz}{dt} = 2z - x + y \end{cases}$	$[0,8; 3]$	$\begin{cases} y(0) = 0,4 \\ x(0,2) = 1 \\ z(0) = 3 \end{cases}$	0,2

Лабораторная работа № 4

ЧИСЛЕННОЕ ИНТЕГРИРОВАНИЕ И ДИФФЕРЕНЦИРОВАНИЕ

Цель работы

1. Освоить методику численного интегрирования методами трапеций, парабол (Симпсона), Гаусса, разработать программы для вычисления определённых интегралов названными методами на языке Delphi, а также научиться вычислять интегралы в символьном виде с помощью пакета MathCAD.

2. Ознакомиться с методом вычисления производных по табличным данным.

Основные понятия

Из курса математического анализа известно, что если функция $f(x)$ непрерывна на отрезке $[a, b]$, то определенный интеграл от этой функции в пределах от a до b существует и имеет вид

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

где $F(x)$ – первообразная для функции $f(x)$.

Для большинства элементарных функций первообразную $F(x)$ не удастся выразить через элементарные функции. Кроме того, при практических расчетах подинтегральная функция задается в виде таблиц. Все это приводит к необходимости замены интегрирования численными методами.

Задача численного интегрирования состоит в следующем: найти определенный интеграл на отрезке $[a, b]$, если подинтегральная функция на отрезке $[a, b]$ задана таблично.

Формулы приближенного интегрирования называются *квадратурными формулами*. Рассмотрим простейшие из них.

Метод трапеций

Метод трапеций использует линейную интерполяцию, т. е. график функции $y = f(x)$ представляется в виде ломаной, соединяющей точки (x, y) . В этом случае площадь всей фигуры складывается из площадей элементарных трапеций, рис. 4.1.

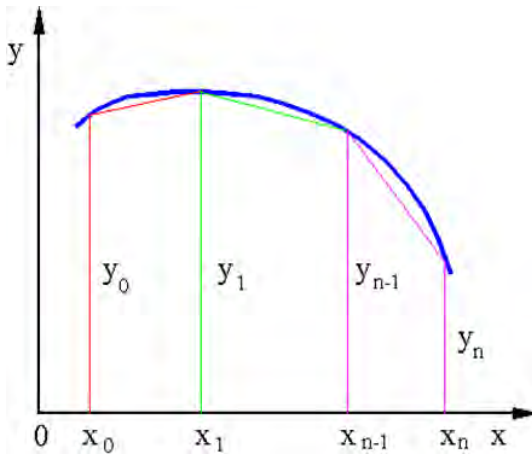


Рис. 4.1. Вычисление определенного интеграла методом трапеций

Площадь каждой такой трапеции равна произведению полусуммы оснований на высоту:

$$A_{\text{трап}} = \frac{y_{i-1} + y_i}{2} h_i, \quad i = 1, 2, \dots, n.$$

Суммируя все эти равенства, получим формулу трапеций для численного интегрирования:

$$\int_a^b f(x) dx = \frac{1}{2} \sum_{i=1}^n h_i (y_{i-1} + y_i).$$

Важным частным случаем формулы трапеции является ее применение при численном интегрировании с **постоянным шагом**

$h_i = h = \text{const}$ ($i = 1, 2, \dots, n$). Формула трапеции в этом случае примет вид

$$\int_a^b f(x)dx = h \left(\frac{y_0 + y_n}{2} + \sum_{i=1}^{n-1} y_i \right) = \frac{b-a}{n} \left(\frac{y_0 + y_n}{2} + \sum_{i=1}^{n-1} y_i \right), \quad (4.1)$$

где n – число разбиений (полос).

Пример 4.1

Пользуясь формулой трапеций (4.1), вычислить интеграл

$$I = \int_0^1 \frac{dx}{1+x^2}$$

при десяти разбиениях ($n = 10$).

Решение

Здесь $f(x) = 1/(1+x^2)$. Определяем шаг интегрирования:

$$h = (b-a)/n = (1-0)/10 = 0,1.$$

Следовательно, $x_0 = 0$; $x_1 = 0,1$; $x_2 = 0,2, \dots, x_{10} = 1$. Тогда по формуле (4.1) получим

$$I = 0,1(0,750000 + 0,9900099 + \dots + 0,552486) = 0,784981.$$

Ответ. $I = \int_0^1 \frac{dx}{1+x^2} = 0,784981.$

Для вычисления определенных интегралов методом трапеций разработана программа на языке Delphi Int_Trap, главная форма которой показана на рис. 4.2.

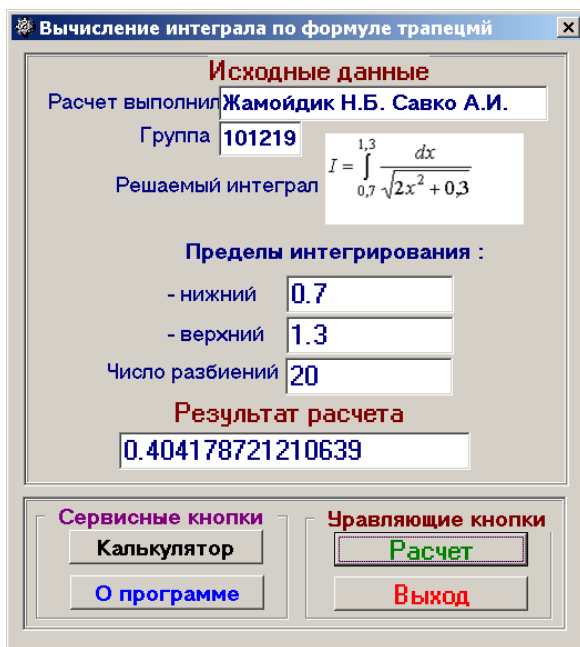


Рис. 4.2. Главная форма программы вычисления определенного интеграла методом трапеций

Дадим некоторые рекомендации программирования вычисления определенного интеграла по методу трапеций.

1. В процессе вычисления интеграла **нет смысла отводить одномерный массив под значения подынтегральной функции**. В нашей программе значение подынтегральной функции при нижнем пределе интегрирования обозначено ua , при верхнем пределе – ub , сумма накапливается в переменной s .

2. В глобальном модуле *Global* в подпрограмме **Int_Trap** описана подынтегральная функция, обращение к которой осуществляется из головной программы:

```

procedure Int_Trap(x:real;Var y:real);
begin
  y:=1/(sqrt(2*x*x+0.3)); //описание подынтегральной функции
end;

```

Текст головной программы в этом случае выглядит следующим образом:

```
procedure TForm1.BitBtn3Click(Sender: TObject);
begin
  a:=StrToFloat(Edit1.Text);
  b:=StrToFloat(Edit2.Text);
  n:=StrToInt(Edit3.Text);
  FIO:=Edit5.Text;
  nGr:=Edit6.Text;
  AssignFile(Fw,'Trap.Dat');
  Rewrite(Fw);
  writeln(Fw,FIO);
writeln(Fw,nGr);
writeln(Fw,n:5);
writeln(Fw,a:10:3,b:10:3);
  CloseFile(Fw);
  h:=(b-a)/n;
  s:=0;
  x:=b;
  Int_Trap(x,yb);
  x:=a;
  Int_Trap(x,ya);
repeat
  x:=x+h;
  Int_Trap(x,y);
  s:=s+y;
until(x>(b-h));
I_Trap:=h*(s+(ya+yb)/2);
  Edit4.Text:=FloatToStr(I_Trap);
end;
```

Метод Симпсона

Точность приближенного интегрирования заметно возрастает, если подынтегральную функцию $y = f(x)$ на отрезке $[a, b]$ заменить интерполяционным многочленом второй степени, рис. 4.3.

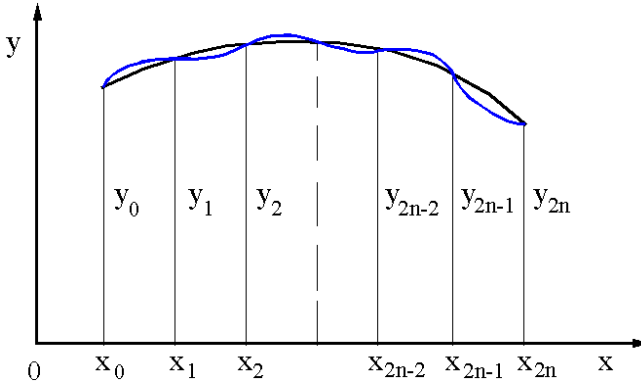


Рис. 4.3. Аппроксимация подынтегральной функции многочленом второй степени

В качестве интерполяционного многочлена используется многочлен Ньютона

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

Тогда

$$\begin{aligned} \int_{x_0}^{x_2} f(x) dx &= \int_{x_0}^{x_2} P_2(x) dx = 2hy_0 + \frac{\Delta y_0}{h} \cdot \frac{4h^2}{2} + \frac{\Delta^2 y_0}{2h^2} \cdot \frac{8h^3}{3} - \frac{\Delta^2 y_0}{2h^2} \cdot h \frac{4h^2}{2} = \\ &= 2hy_0 + 2h\Delta y_0 + \frac{h}{3}\Delta^2 y_0 = \frac{h}{3}(y_0 + 4y_1 + y_2). \end{aligned} \quad (4.2)$$

Соотношение (4.2) называется *формулой Симпсона*. Формула Симпсона обладает повышенной точностью и является точной для многочлена не только второй, но и третьей степени.

Для увеличения точности вычислений отрезок $[a, b]$ разбивают на n пар участков $[x_{2n-2}, x_{2n-1}, x_{2n}]$ и, заменяя подынтегральную функцию интерполяционным многочленом Ньютона второй степени, получают значение интеграла на каждом участке $2h$. Тогда чис-

ленное значение определенного интеграла на отрезке $[a, b]$ будет равно сумме интегралов, т. е.

$$\int_a^b f(x)dx \approx \frac{h}{3} \left[(y_0 + y_{2n}) + 4(y_1 + \dots + y_{2n-1}) + 2(y_2 + \dots + y_{2n-2}) \right]. \quad (4.3)$$

Соотношение (4.3) называется общей формулой Симпсона.

Пример 4.2

По формуле Симпсона вычислить определенный интеграл

$$I = \int_0^1 \frac{dx}{1+x^2}$$

с шагом $h = 0,25$.

Решение

По формуле (4.3) имеем

$$I = \int_0^1 \frac{dx}{1+x^2} = \frac{h}{3} [y_0 + 4y_1 + 2y_2 + 4y_3 + y_4].$$

Подставляя в подынтегральную функцию $f(x) = \frac{dx}{1+x^2}$ значения $x_0 = 0, x_1 = 0,25, x_2 = 0,5, x_3 = 0,75, x_4 = 1$, получим

$$I = \int_0^1 \frac{dx}{1+x^2} = \frac{0,25}{3} \left[1 + \frac{4}{1+0,25^2} + \frac{2}{1+0,5^2} + \frac{4}{1+0,75^2} + \frac{1}{2} \right] \approx 0,785.$$

Ответ. Определенный интеграл $I = \int_0^1 \frac{dx}{1+x^2}$, вычисленный по формуле Симпсона, равен 0,785.

Для вычисления определенного интеграла методом парабол (методом Симпсона) разработана программа на языке Delphi. Главная форма программы представлена на рис. 4.4.

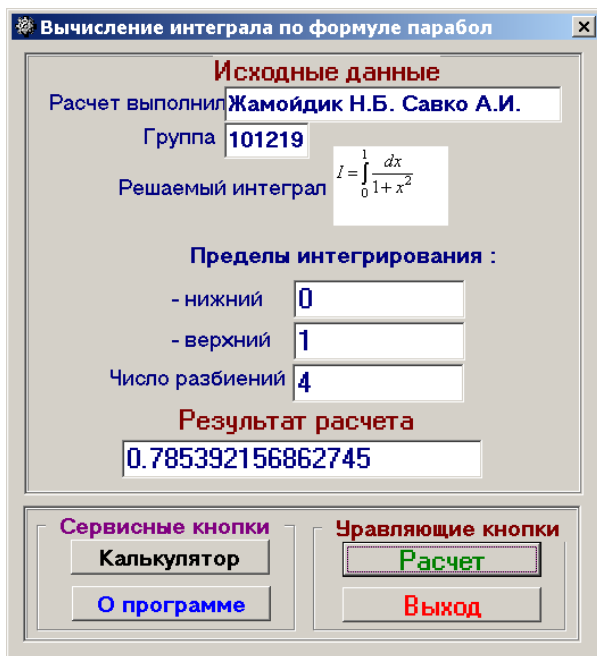


Рис. 4.4. Главная форма программы вычисления определённого интеграла методом парабол

Отметим некоторые особенности этой программы.

1. Подынтегральная функция описывается в подпрограмме **Int_Parabola**, которая размещается в модуле *Global*:

```
procedure Int_Parabola(x:real;Var y:real);
begin
    y:=1/(1+x*x); //описание подынтегральной функции
end;
```

2. В головной программе содержатся некоторые особенности.

```
procedure TForm1.BitBtn3Click(Sender: TObject);
begin
```

```

h:=(b-a)/n;
s1:=0; //обнуление суммы 1
x:=b;

Int_Parabola(x,yb); //вычисление уn
x:=a;
Int_Parabola(x,ya); //вычисление у0
x:=h;

repeat //вычисление суммы нечётных членов
  Int_Parabola(x,y);
  s1:=s1+4*y;
  x:=x+2*h;
until(x>(b-h));
s2:=0; //обнуление суммы 2
x:=2*h; i:=2;

repeat //вычисление суммы чётных членов
  Int_Parabola(x,y);
  s2:=s2+2*y;
  x:=x+2*h;
until(x>(b-h));

I_Parabola:=h*(s1+s2+(ya+yb))/3; //вычисление интеграла
Edit4.Text:=FloatToStr(I_Parabola);
end;

```

2.1. После вычисления шага интегрирования h обнуляется сумма s_1 , в которой накапливается сумма нечетных членов формулы парабол.

2.2. Вычисляются первый $y_0 = y_a$ и последний $y_n = y_b$ члены формулы парабол.

2.3. Отдельно вычисляются суммы нечетных s_1 и четных s_2 членов формулы парабол. Предварительно переменная s_2 обнуляется.

2.4. Определенный интеграл вычисляется по формуле парабол как сумма всех членов, умноженных на шаг интегрирования, деленная на 3.

Квадратурная формула Гаусса

В рассмотренных выше методах вычисления определенного интеграла значения функции должны задаваться с постоянным шагом

по оси x . Если снять это ограничение, то шаг, с которым задаются значения функции, можно выбрать так, чтобы ошибка аппроксимации была как можно меньше. В этом и состоит сущность *метода Гаусса*. Формула интегрирования Гаусса имеет вид

$$I = \int_a^b F(z)dz = \frac{b-a}{2} [c_1 f(x_1) + c_2 f(x_2) + \dots + c_n f(x_n)], \quad (4.4)$$

где

$$x_i = \frac{b+a}{2} - \frac{b-a}{2} t_i, \quad i = 1, 2, \dots, n. \quad (4.5)$$

Значения квадратурных коэффициентов Гаусса c_i и абсцисс t_i приведены в табл. 4.1.

Таблица 4.1

Значения квадратурных коэффициентов Гаусса c_i и абсцисс t_i

n	t_i	c_i
1	$t_1 = 0,5$	$c_1 = 2$
2	$-t_1 = t_2 = 0,577350$	$c_1 = c_2 = 1$
3	$-t_1 = t_3 = 0,774597$ $t_2 = 0$	$c_1 = c_3 = 0,555555$ $c_2 = 0,888889$
4	$-t_1 = t_4 = 0,861136$ $-t_2 = t_3 = 0,339981$	$c_1 = c_4 = 0,347856$ $c_2 = c_3 = 0,652145$
5	$-t_1 = t_5 = 0,906180$ $-t_2 = t_4 = 0,538470$ $t_3 = 0$	$c_1 = c_5 = 0,236927$ $c_2 = c_4 = 0,478629$ $c_3 = 0,568889$
6	$-t_1 = t_6 = 0,932470$ $-t_2 = t_5 = 0,661210$ $-t_3 = t_4 = 0,238620$	$c_1 = c_6 = 0,171324$ $c_2 = c_5 = 0,360761$ $c_3 = c_4 = 0,467914$
7	$-t_1 = t_7 = 0,949108$ $-t_2 = t_6 = 0,741531$ $-t_3 = t_5 = 0,238620$ $t_4 = 0$	$c_1 = c_7 = 0,129485$ $c_2 = c_6 = 0,279705$ $c_3 = c_5 = 0,381630$ $c_4 = 0,417960$
8	$-t_1 = t_8 = 0,960290$ $-t_2 = t_7 = 0,796666$ $-t_3 = t_6 = 0,525532$ $-t_4 = t_5 = 0,183434$	$c_1 = c_8 = 0,101228$ $c_2 = c_7 = 0,222381$ $c_3 = c_6 = 0,313707$ $c_4 = c_5 = 0,362684$

Пример 4.3

Вычислить интеграл $I = \int_0^1 \sqrt{x+1} dx$, применяя квадратурную формулу Гаусса (4.4) с четырьмя ординатами.

Решение

В данном случае имеем $a = 0$, $b = 1$. Пользуясь формулой замены переменных (4.5) и табл. 4.1, найдем абсциссы x_1 , x_2 , x_3 и x_4 :

$$x_1 = \frac{1}{2} + \frac{1}{2}t_1 = 0,5 + \frac{1 \cdot (-0,861136)}{2} = 0,069432;$$

$$x_2 = \frac{1}{2} + \frac{1}{2}t_2 = 0,5 + \frac{1 \cdot (-0,339981)}{2} = 0,330009;$$

$$x_3 = \frac{1}{2} + \frac{1}{2}t_3 = 0,5 + \frac{1 \cdot 0,339981}{2} = 0,669991;$$

$$x_4 = \frac{1}{2} + \frac{1}{2}t_4 = 0,5 + \frac{1 \cdot 0,861136}{2} = 0,930568.$$

Так как квадратурные коэффициенты попарно равны: $c_1 = c_4 = 0,347855$; $c_2 = c_3 = 0,652145$, то формула Гаусса для рассматриваемого случая примет вид

$$\begin{aligned} I = \int_0^1 \sqrt{x+1} dx &= 0,5 \left(0,347855 \left(\sqrt{1,069432} + \sqrt{1,930568} \right) + \right. \\ &\left. + 0,652145 \left(\sqrt{1,300009} + \sqrt{1,669991} \right) \right) = 1,218951. \end{aligned}$$

Ответ. Определенный интеграл $I = \int_0^1 \sqrt{x+1} dx$, вычисленный по формуле Гаусса, равен 1,218951.

Для вычисления определенного интеграла методом Гаусса разработана программа, главная форма которой показана на рис. 4.5.

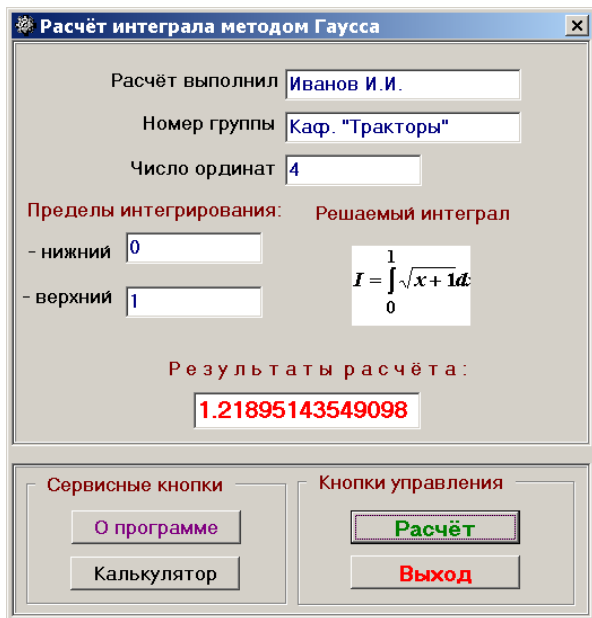


Рис. 4.5. Главная форма программы вычисления определенного интеграла методом Гаусса

Дадим некоторые рекомендации по разработке этой программы.

1. Квадратурные коэффициенты Гаусса размещены в подпрограмме *Koef*, которая находится в модуле *Global*:

```

Procedure Koef(n:integer;Var t:Mas;Var c:Mas);
begin
  Case n of
    1: begin t[1]:=0.5; c[1]:=2 end;
    2: begin
        t[2]:=0.577350; t[1]:=-t[2];
        c[1]:=1; c[2]:=c[1];
        end;
    3: begin
        t[3]:=0.774597; t[1]:=-t[3]; t[2]:=0;
        c[1]:=0.555555; c[3]:=c[1]; c[2]:=0.888889;
        end;
  end;
end;

```

```

4: begin
    t[4]:=0.861136; t[1]:=-t[4]; t[3]:=0.339981; t[2]:=-t[3];
    c[1]:=0.347855; c[4]:=c[1]; c[2]:=0.652145; c[3]:=c[2];
end;
5: begin
    t[5]:=0.906180; t[1]:=-t[5]; t[4]:=-0.538470; t[3]:=-t[4];
    t[3]:=0; c[3]:=0.568889;
    c[1]:=0.236927; c[5]:=c[1]; c[2]:=0.478629;c[4]:=c[2];
end;
end;
end; //Конец Koef

```

В подпрограмму в качестве исходного данного передается число ординат n , а возвращаются квадратурные коэффициенты t и c . Так как в подпрограмме воспроизведена не вся таблица квадратурных коэффициентов (табл. 4.1), а только ее часть, включая пять ординат, в главной программе осуществляется проверка числа вводимых ординат.

```

if (n>5) then begin
    MessageDlg('Число ординат не может быть больше 5',
        mtWarning,[mbOK],0);
    Edit3.SetFocus;
    Exit;
end;

```

Если число ординат больше пяти, выдается сообщение об ошибке и в фокусе оказывается окно, в котором помещается число ординат, рис. 4.6.

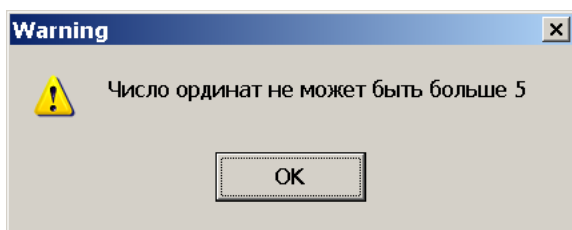


Рис. 4.6. Сообщение об ошибке ввода числа ординат

2. В подпрограмме **Fun** (подпрограмма размещена в глобальном модуле **Global**) помещен вычисляемый интеграл, точнее его подынтегральная функция, которая вычисляется для каждого x_i :

```
Procedure Fun(i:integer;X:Mas;Var Fx:Mas);
begin
  Fx[i]:=Sqrt(X[i]+1); //описание подынтегральной функции
end; //Конец Fun
```

3. Определенный интеграл вычисляется в подпрограмме **Int_Gauss**, помещенной в глобальный модуль **Global**.

```
Procedure Int_Gauss(n:integer;c,t:Mas;Var IntG:real);
  Var i : integer; sint : real;
begin
  for i:=1 to n do X[i]:=(b+a)/2+(b-a)*t[i]/2;
  for i:=1 to n do Fun(i,X,Fx);
  for i:=1 to n do sint:=sint+c[i]*Fx[i]; //sint - сумма
  IntG:=(b-a)*sint/2;
end; //Конец Int_Gauss
```

В качестве исходных данных в подпрограмму передаются число ординат n и квадратурные коэффициенты c и t . Подпрограмма возвращает результат вычисления интеграла $IntG$ методом Гаусса.

4. С использованием указанных подпрограмм головная программа получается весьма короткой:

```
Koef(n,t,c); //Определение коэффициентов t и c
```

```
Int_Gauss(n,c,t,IntG); //Обращение к п/п вычисл.интегр.
```

```
Edit6.Text:=FloatToStr(IntG); //Результат в окне Edit6
```

Таким образом, разработанные программы на языке Delphi позволяют вычислять определенные интегралы различными методами: трапеций, парабол, Гаусса.

Численное дифференцирование

При решении инженерных задач часто приходится вычислять производную, однако во многих практических задачах функция задается таблично и методы дифференциального исчисления к исследованию таких функций применить нельзя. В этом случае прибегают к численному дифференцированию.

Однако численное дифференцирование может дать совершенно бессмысленные результаты, поэтому **по возможности его следует избегать**. Чтобы лучше понять причины подобных затруднений, рассмотрим два их потенциальных источника.

Первая трудность возникает при использовании табличных данных, полученных экспериментальным путем. В любом эксперименте регистрируется полезная информация, сопровождаемая более или менее сильным шумом. Следовательно, результаты измерений истинного сигнала содержат шумовую компоненту, рис. 4.7.

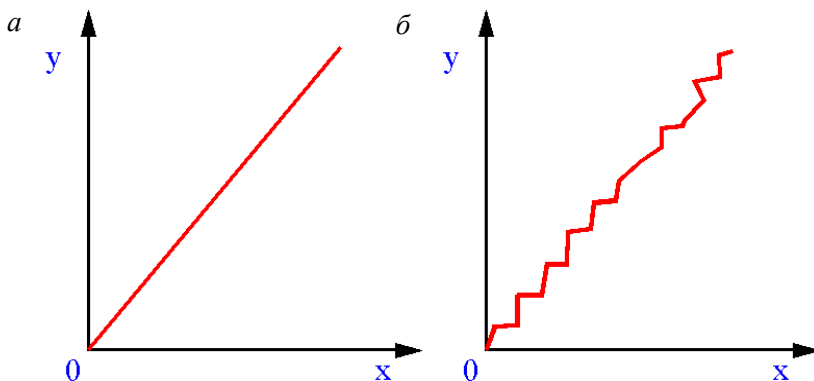


Рис. 4.7. Истинный (а) и измеренный (б) сигналы

Если продифференцировать измеренный сигнал, то погрешность резко возрастает. Наоборот, при интегрировании сигнала, содержащего шум, погрешность убывает, что говорит о более устойчивом процессе численного интегрирования по сравнению с численным дифференцированием, рис. 4.8.

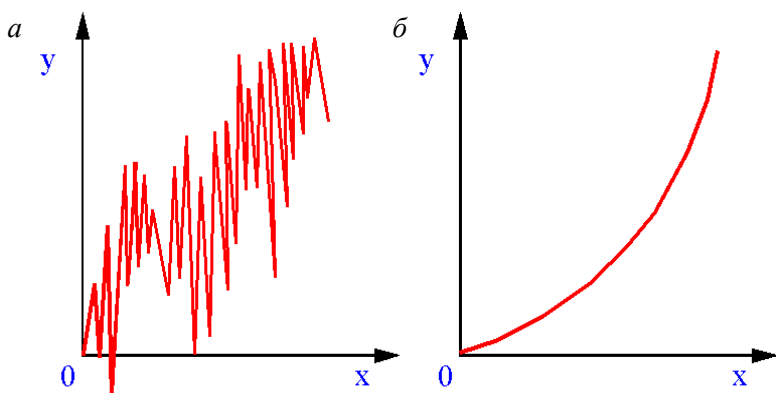


Рис. 4.8. Численное дифференцирование (а) и интегрирование (б) сигнала, содержащего шум

Вторая трудность, встречающаяся при численном дифференцировании, показана на рис. 4.9.

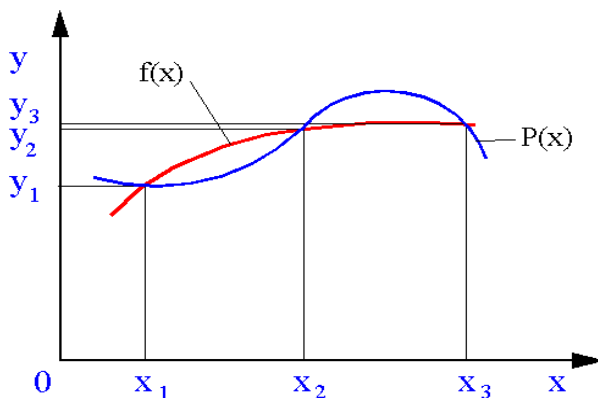


Рис. 4.9. Вторая трудность, встречающаяся при использовании интерполяционных многочленов для численного дифференцирования

Даже в случае когда интерполяционный многочлен $P(x)$ хорошо аппроксимирует табличную функцию в узлах, его производные высших порядков могут не иметь ничего общего с производными аппроксимируемой функции $f(x)$. Это хорошо видно из рис. 4.9 при сравнении наклона и кривизны двух кривых.

Задача численного дифференцирования ставится следующим образом. Пусть функция $y = f(x)$ на отрезке $[a, b]$ задана таблично своими $n + 1$ значениями. Требуется найти аналитическое выражение производной.

Существует ряд формул *численного дифференцирования*. Для построения таких формул заданную функцию $f(x)$ заменяют на отрезке $[a, b]$ интерполирующей функцией $P(x)$, а затем полагают, что исконая производная равна производной функции $P(x)$.

Аналогично поступают при нахождении производных высших порядков. В качестве аппроксимирующей функции чаще всего выбирается интерполяционный многочлен.

Формулы приближенного дифференцирования, основанные на интерполяционных формулах Ньютона

Если узлы интерполяции равноотстоящие, т. е. $x_{i+1} - x_i = h$ ($i = 0, 1, \dots, n$), то для замены исходной функции $f(x)$ используют интерполяционные формулы Ньютона. Тогда для численного дифференцирования получаются следующие формулы:

$$f'(x) \approx \frac{1}{h} \left(\Delta y_0 + \frac{2q-1}{2} \Delta^2 y_0 + \frac{3q^2-6q+2}{6} \Delta^3 y_0 + \frac{2q^3-9q^2+11q-3}{12} \Delta^4 y_0 + \dots \right);$$

$$f''(x) \approx \frac{1}{h^2} \left(\Delta^2 y_0 + (q-1) \Delta^3 y_0 + \frac{6q^3-18q+11}{12} \Delta^4 y_0 \right),$$

где $q = (x - x_0)/h$; $h = x_{i+1} - x_i$, $i = 0, 1, 2, \dots$

Формулы приближенного дифференцирования для определения производных в узлах интерполяции значительно упрощаются:

$$f'(x) \approx \frac{1}{h} \left(\Delta y_0 - \frac{\Delta^2 y_0}{2} + \frac{\Delta^3 y_0}{3} - \frac{\Delta^4 y_0}{4} + \frac{\Delta^5 y_0}{5} + \dots \right); \quad (4.6)$$

$$f''(x) \approx \frac{1}{h^2} \left(\Delta^2 y_0 - \Delta^3 y_0 + \frac{11}{12} \Delta^4 y_0 - \frac{5}{6} \Delta^5 y_0 + \dots \right). \quad (4.7)$$

Чтобы получить значение производной в точке, находящейся в конце таблицы, используют вторую интерполяционную формулу Ньютона. Тогда выражение для приближенного значения производной будет

$$f'(x) \approx \frac{1}{h} \left(\Delta y_{n-1} + \frac{2q+1}{2} \Delta^2 y_{n-2} + \frac{3q^2+6q+2}{6} \Delta^3 y_{n-3} + \frac{2q^3+9q^2+11q+3}{12} \Delta^4 y_{n-4} + \dots \right).$$

Пример 4.4

Функция $y = f(x)$ задана таблично:

x_i	1	2	3	4
y_i	4	9	26	61

Методом численного дифференцирования найти первые две производные этой функции в точке $x = 1$.

Решение

Составим таблицу конечных разностей.

x_i	y_i	Δy_i	$\Delta^2 y_i$	$\Delta^3 y_i$
1	4	5	12	6
2	9	17	18	
3	26	35		
4	61			

Шаг изменения x_i в таблице исходных данных $h = 1$. Согласно формулам (4.6) и (4.7) имеем

$$f'(x) = \frac{1}{1} \left(5 - \frac{12}{2} + \frac{6}{3} \right) = 1;$$

$$f''(x) = \frac{1}{1^2} (12 - 6) = 6.$$

При дифференцировании по заданной переменной достаточно установить на переменной курсор ввода. Выражение при этом отдельно не указывается, поскольку указание в нем на переменную является одновременно и указанием на это выражение. Если выражение содержит другие переменные, то они рассматриваются как константы.

Операция **Differentiate** возвращает символьное значение производной выражения по этой переменной, которая указана курсором. Для вычисления производных высшего порядка (второго и выше) вычисление производной следует повторить необходимое число раз.

Рассмотрим применение операции дифференцирования с использованием пакета MathCAD.

Выражение	Результат
$\sin x$	$\cos x$
$\frac{d}{dx}(\sin x)$	$\cos x$
$(\sin x)^2$	$2 \cos x \cdot \sin x$
$\frac{d}{dx}(\sin x)^2$	$2 \cdot \cos(x) \cdot \sin(x)$
$\sqrt[3]{x}$	$\frac{1}{3x^{\frac{2}{3}}}$
$\sqrt[n]{x}$	$\frac{\frac{1}{n}-1}{x^n}$
$\ln x$	$\frac{1}{x}$

Таким образом, применяя пакет MathCAD, можно вычислять интегралы и дифференцировать в символьном виде или получать численное значение определенного интеграла. Если интеграл не берется, следует разложить подынтегральную функцию в ряд Тейлора (с помощью пакета MathCAD), а затем вычислить интеграл от разложения.

Варианты заданий

1. Разработать программы на языке Delphi для вычисления интегралов методами трапеций с тремя десятичными знаками, парабол (Симпсона) при $n = 8$ и Гаусса при $n = 4$. Эти же интегралы вычислить с помощью пакета MathCAD.

Номер варианта	Задание	
1	$\int_{0,8}^{1,6} \frac{dx}{\sqrt{2x^2 + 1}}$	$\int_{1,2}^{2,0} \frac{\ln(x+2)}{x} dx$
2	$\int_{1,2}^{2,7} \frac{dx}{\sqrt{x^2 + 3,2}}$	$\int_{1,6}^{2,4} (x+1) dx$
3	$\int_1^2 \frac{dx}{\sqrt{2x^2 + 1,3}}$	$\int_{0,2}^{1,0} \frac{\operatorname{tg}(x^2)}{x^2 + 1} dx$
4	$\int_{0,2}^{1,2} \frac{dx}{\sqrt{x^2 + 1}}$	$\int_{0,6}^{1,4} \frac{\cos x}{x+1} dx$
5	$\int_{0,8}^{1,4} \frac{dx}{\sqrt{2x^2 + 3}}$	$\int_{0,4}^{1,2} \sqrt{x} \cos(x^2) dx$
6	$\int_{0,4}^{1,2} \frac{dx}{\sqrt{2+0,5x^2}}$	$\int_{0,8}^{1,2} \frac{\sin(2x)}{x^2} dx$
7	$\int_{1,4}^{2,1} \frac{dx}{\sqrt{3x^2 - 1}}$	$\int_{0,8}^{1,6} \frac{\ln(x^2 + 1)}{x} dx$
8	$\int_{1,2}^{2,4} \frac{dx}{\sqrt{0,5 + x^2}}$	$\int_{0,4}^{1,2} \frac{\cos x}{x+2} dx$
9	$\int_{0,4}^{1,2} \frac{dx}{\sqrt{3 + x^2}}$	$\int_{0,4}^{1,2} (2x + 0,5) \sin x dx$
10	$\int_{0,6}^{1,5} \frac{dx}{\sqrt{1 + 2x^2}}$	$\int_{0,4}^{0,8} \frac{\operatorname{tg}(x^2 + 0,5)}{1 + 2x^2} dx$

Номер варианта	Задание	
11	$\int_{2,0}^{3,5} \frac{dx}{\sqrt{x^2 - 1}}$	$\int_{1,8}^{0,98} \frac{\sin x}{x+1} dx$
12	$\int_{0,5}^{1,3} \frac{dx}{\sqrt{x^2 + 2}}$	$\int_{0,2}^{0,18} \sqrt{x+1} \cos(x^2) dx$
13	$\int_{2,2}^{2,6} \frac{dx}{\sqrt{x^2 + 2}}$	$\int_{1,4}^{3,0} x^2 \ln x dx$
14	$\int_{1,4}^{2,2} \frac{dx}{\sqrt{3x^2 + 1}}$	$\int_{1,4}^{2,2} \frac{\ln(x^2 + 2)}{x+1} dx$
15	$\int_{0,8}^{1,8} \frac{dx}{\sqrt{x^2 + 4}}$	$\int_{0,4}^{1,2} \frac{\cos(x^2)}{x+1} dx$
16	$\int_{1,6}^{2,2} \frac{dx}{\sqrt{x^2 + 2,5}}$	$\int_{0,8}^{1,6} (x^2 + 1) \sin(x - 0,5) dx$
17	$\int_{0,6}^{1,6} \frac{dx}{\sqrt{x^2 + 0,8}}$	$\int_{0,6}^{1,4} x^2 \cos x dx$
18	$\int_{1,2}^{2,0} \frac{dx}{\sqrt{x^2 + 1,2}}$	$\int_{1,2}^{2,0} \frac{\ln(x^2 + 3)}{2x} dx$
19	$\int_{1,4}^{2,0} \frac{dx}{\sqrt{2x^2 + 0,7}}$	$\int_{2,5}^{3,3} \frac{\ln(x^2 + 0,8)}{x-1} dx$
20	$\int_{3,2}^{4,0} \frac{dx}{\sqrt{0,5x^2 + 1}}$	$\int_{0,5}^{1,2} \frac{\operatorname{tg}(x^2)}{x+1} dx$

2. С помощью интерполяционной формулы Ньютона найти значения первой и второй производных при данных значениях аргумента для функции, заданной таблично.

x	y	x	y
2,4	3,526	3,6	4,222
2,6	3,782	3,8	4,331
2,8	3,945	4,0	4,507
3,0	4,043	4,2	4,775
3,2	4,104	4,4	5,159
3,4	4,155	4,6	5,683

Аргумент

- а) $x = 2,4 + 0,05n$;
 б) $x = 3,12 + 0,03n$;
 в) $x = 4,5 - 0,06n$;
 г) $x = 4,04 - 0,04n$;

$N = 1, 3, 5, 7, \dots, 29$

x	y	x	y
1,5		4,5	
2,0		5,0	
2,5		5,5	
3,0		6,0	
3,5		6,5	
4,0		7,0	

Аргумент

- а) $x = 1,6 + 0,08n$;
 б) $x = 3,27 + 0,11n$;
 в) $x = 6,3 - 0,12n$;
 г) $x = 5,85 - 0,09n$;

$N = 2, 4, 6, 8, \dots, 30$

x	y	x	y
0,8	2,857	2,4	6,503
1,2	3,946	2,8	7,010
1,6	4,938	3,2	7,288
2,0	5,801	3,6	7,301

Аргумент

- а) $x = 1,2$;
 б) $x = 2,23$;
 в) $x = 2,76$;
 г) $x = 3,1$

Лабораторная работа № 5

АППРОКСИМАЦИЯ ФУНКЦИЙ

Цель работы

1. Ознакомиться с основными методами аппроксимации функций.
2. Освоить основные методы определения значения функции не в узлах табличных данных (интерполировать).
3. На языке Delphi разработать программы для интерполирования табличных данных различными методами.

Способы задания функций

В практической деятельности инженера постоянно приходится сталкиваться с необходимостью выявления форм связи в процессах и явлениях и необходимостью их описания.

Рассмотрим такие формы связи, для которых некоторая величина y , характеризующая процесс, зависит от совокупности не связанных между собой величин x_1, x_2, \dots, x_n таким образом, что каждому набору x_i соответствует единственное значение величины y_i . Такое однозначное соответствие величины y совокупности независимых переменных x_i называется *функциональной зависимостью*, а сама переменная y – функцией переменных величин x_i , что формально записывается в виде

$$y = f(x_1, x_2, \dots, x_n).$$

Так, выражение силы тяги ведущего колеса

$$F_k = F_{k \max}(1 - e^{-k\delta})$$

является функцией трех переменных: силы тяги по сцеплению $F_{k \max}$, коэффициента k и буксования колеса δ . Если величина y есть функция одной независимой переменной, то эту связь можно представить соотношением вида $y = f(x)$. Например, радиус поворота R_T колесной машины, имеющей базу L , без учета явления увода является функцией независимой переменной α – среднего угла поворота управляемых колес:

$$R_T = L / \operatorname{tg}\alpha.$$

Из курса математического анализа известны три способа задания функциональных зависимостей: аналитический, графический и табличный.

Наиболее удобным способом задания функциональной зависимости $y = f(x)$ является *аналитический*, так как для получения соответствующего значения величины y он прямо указывает действия над независимой переменной x и их последовательность.

Положительным качеством этого способа является возможность получить значения y для любого фиксированного аргумента x с любой точностью. К недостаткам метода относится то, что приходится производить всю последовательность вычислений, кроме того, аналитический метод не обладает наглядностью.

Указанные недостатки аналитического способа устраняются в случае *графического* задания функции. Графиком данной функции $y = f(x)$ называется *геометрическое место точек плоскости xOy , координаты которых удовлетворяют уравнению $y = f(x)$* .

Табличный способ задания функций широко распространен в технике и чаще всего применяется в результате эксперимента. Преимуществом табличного способа задания функции является то, что для каждого значения независимой переменной, помещённой в таблице, можно сразу же, без всяких измерений и вычислений, найти соответствующее значение функции. Недостаток способа состоит в том, что нельзя задать всю функцию, т. е. найти такие значения независимой переменной, которых нет в таблице.

Понятие о приближении функций

Пусть величина y является функцией аргумента x . Вместе с тем на практике связь между ними часто неизвестна, т. е. невозможно записать эту связь в виде некоторой аналитической зависимости $y = f(x)$.

Распространённым и практически важным случаем, когда вид связи между параметрами x и y неизвестен, является задание этой связи, как указывалось выше, в виде некоторой таблицы $y = f(x)$. Но могут понадобиться значения величины y и в других точках, отличных от узлов x . Однако получить эти значения можно лишь путем сложных расчетов или проведением дополнительного дорогостоящего эксперимента.

Таким образом, с целью экономии времени и средств приходим к необходимости использования имеющихся табличных данных для приближенного вычисления искомого параметра y при любом значении их в некоторой области, определяющего параметра x , поскольку точная связь $y = f(x)$ неизвестна. Этой цели и служит задача о приближении (*аппроксимации*) функций: данную функцию $f(x)$ требуется приближенно заменить (аппроксимировать) некоторой функцией $\varphi(x)$ так, чтобы отклонение $\varphi(x)$ от $f(x)$ в заданной области было наименьшим. Функция $\varphi(x)$ при этом называется *аппроксимирующей*. Для практики весьма важен случай аппроксимации функции многочленом.

Если приближение строится на заданном дискретном множестве точек, то аппроксимация называется *точечной*. К ней относятся интерполирование, среднеквадратическое приближение и др. При построении приближения на непрерывном множестве точек, например, на отрезке $[a, b]$, аппроксимация называется *непрерывной* или *интегральной*.

Линейная интерполяция

Простейшим видом интерполяции является линейная интерполяция, в основе которой лежит аппроксимация кривой на участке между точками (x_k, y_k) и (x_{k+1}, y_{k+1}) прямой, проходящей через эти же точки, рис. 5.1.

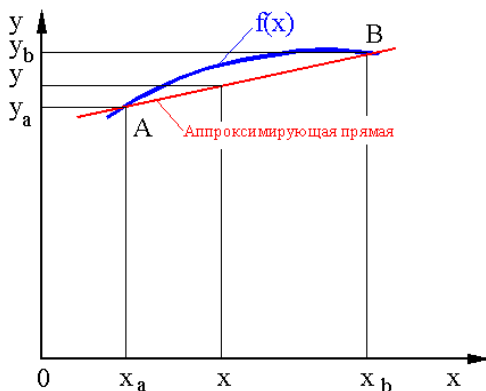


Рис. 5.1. Линейная интерполяция

Уравнение прямой, проходящей через две точки, имеет вид

$$\frac{y - y_k}{x - x_k} = \frac{y_{k+1} - y_k}{x_{k+1} - x_k},$$

откуда

$$y = \frac{(y_{k+1} - y_k)(x_{k+1} - x_k)}{x_{k+1} - x_k} + y_k. \quad (5.1)$$

Таким образом, зная два табличных значения y_k и y_{k+1} , соответствующих x_k и x_{k+1} , с помощью формулы (5.1) можно найти значение функции y при любом значении x в интервале $[x_k, x_{k+1}]$.

Пример 5.1

Определить значение функции, заданной таблично (табл. 5.1), при $x = 2,52$.

Таблица 5.1

x	1	2	3	5
y	1	5	14	81

Решение

В нашем случае $x = 2,52$; $x_k = 2$; $x_{k+1} = 3$; $y_k = 5$; $y_{k+1} = 14$. Подставив эти значения в формулу (5.1), получим

$$y(2,52) = \frac{(14 - 5)(2,52 - 2)}{3 - 2} + 5 = 9,68.$$

Ответ. Значение функции при $x = 2,52$ равно 9,68.

Интерполяционный многочлен Лагранжа

Обычно полагают, что, используя большее число соседних точек и аппроксимируя истинную кривую более сложной линией, можно

уточнить полученный результат. Для достижения этой цели находят многочлен n -й степени $P_n(x)$, аппроксимирующий функцию, заданную таблично, и **проходящий через все заданные в таблице точки** x_i, y_i , где $i = 0, 1, 2, \dots, n$.

Наиболее общей формулой параболического интерполирования является интерполяционная формула Лагранжа. Задача параболического интерполирования в этом случае формулируется следующим образом: на отрезке $[a, b]$ в узлах интерполяции x_0, x_1, \dots, x_n функция $f(x)$ задается своими $n + 1$ значениями: $y_0 = f(x_0), y_1 = f(x_1), \dots, y_n = f(x_n)$.

Требуется так построить многочлен $L(x)$, чтобы в узлах связи интерполяции x_0, x_1, \dots, x_n его значения совпадали со значениями заданной функции, т. е. $L(x_0) = y_0, L(x_1) = y_1, \dots, L(x_n) = y_n$.

Следует заметить, что в такой постановке задачи узлы интерполяции x_0, x_1, \dots, x_n могут произвольно отстоять друг от друга на отрезке $[a, b]$, иными словами, узлы интерполяции неравноотстоящие, т. е. $h = x_{i+1} - x_i \neq \text{const}$ ($i = 0, 1, \dots, n - 1$). Величина h называется шагом интерполяции.

Интерполяционный многочлен Лагранжа имеет вид

$$L_n = y_0 \frac{(x - x_1) \cdot \dots \cdot (x - x_n)}{(x_0 - x_1) \cdot \dots \cdot (x_0 - x_n)} + y_1 \frac{(x - x_1)(x - x_2) \cdot \dots \cdot (x - x_n)}{(x_1 - x_0)(x_1 - x_2) \cdot \dots \cdot (x_1 - x_n)} + \dots + y_n \frac{(x - x_1)(x - x_2) \cdot \dots \cdot (x - x_{n-1})}{(x_n - x_0)(x_n - x_1) \cdot \dots \cdot (x_n - x_{n-1})}. \quad (5.2)$$

В сокращенном виде интерполяционный многочлен Лагранжа записывается так:

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

Пример 5.2

Для функции, заданной таблично, построить интерполяционный многочлен Лагранжа и вычислить ее значение при $x = 2,52$ (исходные данные см. в примере 5.1).

Решение

Так как функция в таблице задается четырьмя значениями, то степень искомого полинома Лагранжа будет не выше третьей. Подставляем исходные данные в формулу (5.2):

$$\begin{aligned}L_3(x) &= 1 \cdot \frac{(x-2)(x-3)(x-5)}{(1-2)(1-3)(1-5)} + 5 \cdot \frac{(x-1)(x-3)(x-5)}{(2-1)(2-3)(2-5)} + \\ &+ 14 \cdot \frac{(x-1)(x-2)(x-5)}{(3-1)(3-3)(3-5)} + 81 \cdot \frac{(x-1)(x-2)(x-3)}{(5-1)(5-2)(5-3)} = \\ &= \frac{17}{12}x^3 - 6x^2 + \frac{145}{12}x - \frac{13}{2}.\end{aligned}$$

Находим значение функции при $x = 2,52$:

$$f(2,52) = \frac{17}{12}2,52^3 - 6 \cdot 2,52^2 + \frac{145}{12}2,52 - \frac{13}{2} = 8,519.$$

Найдем значения интерполяционного многочлена в узлах интерполяции:

$$x = 1, L_3(1) = 1;$$

$$x = 2, L_3(2) = 5;$$

$$x = 3, L_3(3) = 14;$$

$$x = 5, L_3(5) = 81.$$

Видно, что полученный интерполяционный многочлен Лагранжа дает в узлах интерполяции полное приближение заданной функции.

При других значениях аргумента $x = z, z \in [x_0, x_n]$ разность $R_n(z) = f(z) - L_n(z)$ в общем случае отлична от нуля и определяет собой ошибку метода. Полагая, что функция $f(x)$ непрерывна на интервале $[x_0, x_n]$, величину ошибки $R_n(z)$ оценивают по формуле

$$R_n(z) = \frac{f^{n+1}(z)}{(n+1)!} (z-x_0)(z-x_1) \cdot \dots \cdot (z-x_n),$$

где $f^{n+1}(z)$ – производная $(n+1)$ -го порядка.

Пример 5.3

Функция $f(x)$ задана таблично, табл. 5.2.

Таблица 5.2

i	0	1	2	3
x_i	0	1	2	6
y_i	-1	-3	3	1 187

Используя интерполяционный многочлен Лагранжа, найти ее значение в точке $x = 4$.

Решение

Подставив в формулу (5.2) $x = 4$, имеем

$$L_3(4) = -1 \cdot \frac{(4-1)(4-2)(4-6)}{(0-1)(0-2)(0-6)} - 3 \cdot \frac{4(4-2)(4-6)}{1(1-2)(1-6)} +$$

$$+ 3 \cdot \frac{4(4-1)(4-6)}{2(2-1)(2-6)(3-5)} + 1187 \cdot \frac{4(4-1)(4-2)}{6(6-1)(6-2)} = 255.$$

Ответ. При $x = 4$ значение функции $y = 255$.

Если в рассмотренном примере к таблице добавить еще одну точку, то вычисления значения функции при $x = 4$ придется выполнять заново. Кроме того, из рассмотренных примеров видно, что процесс получения приближенного значения функции по интерполяционной формуле Лагранжа связан с большим объемом вычислений.

Это потребовало упрощения вычислительной работы. Для удобства вычислений составляется вспомогательная таблица (табл. 5.3).

Таблица 5.3

$X - X_0$	$X_0 - X_1$	$X_0 - X_2$...	$X_0 - X_n$	D_0
$X_1 - X_0$	$X - X_1$	$X_1 - X_2$...	$X_1 - X_n$	D_1
$X_2 - X_0$	$X_2 - X_1$	$X - X_2$...	$X_2 - X_n$	D_2
...
$X_n - X_0$	$X_n - X_1$	$X_n - X_2$...	$X - X_n$	D_n

где $x_0, x_1, x_2, \dots, x_n$ – узлы интерполяции;

x – значение аргумента, для которого определяется значение функции.

Обозначим произведение элементов первой строки через D_0 :

$$D_0 = (x - x_0)(x_0 - x_1)(x_0 - x_2) \cdot \dots \cdot (x_0 - x_n).$$

В общем виде произведение элементов i -й строки есть

$$D_i = (x_i - x_0)(x_i - x_1) \cdot \dots \cdot (x_i - x_{i-1})(x - x_1)(x - x_{i+1}) \cdot \dots \cdot (x_i - x_n).$$

Произведение строк размещается в правом столбце таблицы. Дополнительно вычисляется произведение элементов, расположенных на главной диагонали:

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

Тогда интерполяционный многочлен Лагранжа можно переписать в виде

$$L_n(x) = \Pi_{n+1}(x) \sum_{i=0}^n \frac{y_i}{D_i}. \quad (5.3)$$

Пользуясь формулой (5.3), снова решим предыдущий пример. Для этого составим вспомогательную таблицу (табл. 5.4) и найдем произведение элементов, расположенных на главной диагонали: $\Pi_4(4) = -48$.

Таблица 5.4

4	-1	-2	-6	-48
1	4-1	1-2	1-6	15
2	-1	4-2	2-6	-16
6	6-1	6-2	4-6	-240

Приближенное значение функции в точке $x = 4$, т. е. $f(x) \approx L_3(4)$ определим по формуле (5.3):

$$L_3(4) = -48 \left(\frac{-1}{-48} + \frac{-3}{15} + \frac{3}{-16} + \frac{1187}{-240} \right) = 255.$$

Следовательно, при значительно меньшем объеме расчетов получен тот же результат.

Пример 5.4

Функция задана таблично, табл. 5.5.

Таблица 5.5

x	0,05	0,10	0,170	0,250	0,30	0,36
y	0,050042	0,100335	0,171657	0,255342	0,309336	0,376403

На языке Delphi разработать программу для вычисления функции с использованием интерполяционного многочлена Лагранжа и определить функцию при $x = 0,263$.

Некоторые рекомендации по разработке программы на языке Delphi

1. Вид файла исходных данных

Жамойдик Н.Б. Савко А.И.

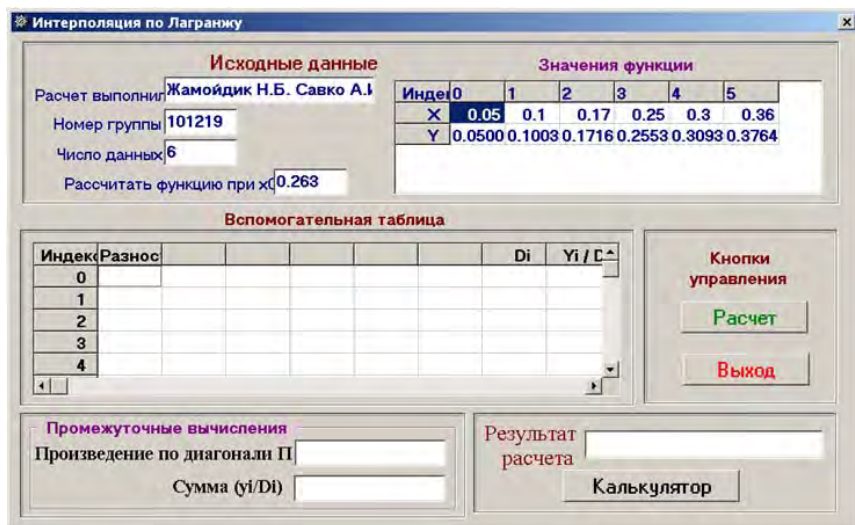
101219

6 0.263 {число данных, рассчитать функцию при 0,263}

0.05 0.1 0.17 0.25 0.30 0.36 {значения x -в}

0.050042 0.100335 0.171657 0.255342 0.309336 0.376403 {значения y -в}

2. Дизайн главной формы с использованием компоненты String-Grid.



3. В данной программе глобальный модуль используется только для описания глобальных переменных и массивов.

```

unit Global;
interface
  type Mas=array[0..10] of real; MasXY=array[0..10,0..10] of real;
  Var   Fr,Fw : TextFile; FIO,nGr : string; n : integer;
        X,Y,D,YD : Mas; XY : MasXY;
x0,Sum,P,Rez : real;
implementation
end.

```

4. Главный модуль.

4.1. Подпрограмма создания главной формы

```

procedure TForm1.FormCreate(Sender: TObject);
Var i : integer;
begin

```



```

DecimalSeparator:='.';
AssignFile(Fr,'A.Dat');
Reset(Fr);
readln(Fr,FIO); //чтение исходных данных из
readln(Fr,nGr); //файла «A.Dat»
readln(Fr,n,x0);
  for i:=0 to (n-1) do read(Fr,X[i]); read(Fr);
  for i:=0 to (n-1) do read(Fr,Y[i]); read(Fr);
CloseFile(Fr);
Edit4.Text:=FIO; //Размещение данных по окнам Edit
Edit5.Text:=nGr;
Edit6.Text:=IntToStr(n);
Edit7.Text:=FloatToStr(x0);
//оформление таблицы StringGrid1
with StringGrid1 do begin
  ColCount:=n+3; //Число столбцов
  RowCount:=n+1; //Число строк
  Cells[0,0]:=' Индекс';
  Cells[n+1,0]:=' Di';
  Cells[n+2,0]:=' Yi / Di';
  for i:=0 to n-1 do Cells[0,i+1]:=' '+IntToStr(i);
  end;

  StringGrid1.Cells[1,0]:='Разности';
//оформление таблицы StringGrid2
with StringGrid2 do begin
  ColCount:=n+1; //Число столбцов
  RowCount:=3; //Число строк
  Cells[0,0]:=' Индекс';
  Cells[0,1]:=' X';
  Cells[0,2]:=' Y';
  //размещение данных в ячейках таблицы StringGrid2
  for i:=0 to n-1 do Cells[i+1,0]:='' +IntToStr(i);
  for i:=0 to n-1 do Cells[i+1,1]:=' '+FloatToStr(X[i]);
  for i:=0 to n-1 do Cells[i+1,2]:='' +FloatToStr(Y[i]);
  end;
end; //конец подпрограммы создания формы

```

4.2. Подпрограмма расчета

```
procedure TForm1.BitBtn1Click(Sender: TObject); //расчет
var i,j:integer;
begin
  for i:=0 to (n-1) do D[i]:=1;
  for i:=0 to (n-1) do //расчет разностей
    for j:=0 to (n-1) do begin
      if (i=j) then XY[i,j]:=x0-X[i]
      else XY[i,j]:=X[i]-X[j];
    end;
  with StringGrid1 do //размещ. разностей по ячейкам String-
Grid1
    for i:=0 to (n-1) do
      for j:=0 to (n-1) do Cells[i+1,j+1]:=FloatToStr(XY[j,i]);
  for i:=0 to (n-1) do //расчет произв. строк таблицы
    for j:=0 to (n-1) do D[i]:=D[i]*XY[i,j];
  with StringGrid1 do //размещ. произв. строк по ячейкам String-
Grid1
    for i:=0 to (n-1) do Cells[n+1,i+1]:=FloatToStr(D[i]);
  for i:=0 to (n-1) do YD[i]:=Y[i]/D[i]; //расчет частного от деления
yi/Di
  with StringGrid1 do //размещ. частного по ячейкам StringGrid1
    for i:=0 to (n-1) do Cells[n+2,i+1]:=FloatToStr(YD[i]);
  Sum:=0; //расчет суммы yi/Di
  for i:=0 to (n-1) do Sum:=Sum+YD[i];
  Edit3.Text:=FloatToStr(Sum);
  P:=1; //расчет произведения значен. ячеек на главн. диа-
гонали
  for i:=0 to (n-1) do P:=P*XY[i,i];
  Edit2.Text:=FloatToStr(P);
  Rez:=P*Sum; //результат вычисления функции
  Edit1.Text:=FloatToStr(Rez);
end; //конец подпрограммы расчет
```

5. Главная форма с результатами расчета

Интерполяция по Лагранжу

Исходные данные: Расчет выполнен: Жамойдик Н.Б. Савко А.И.
 Номер группы: 101219
 Число данных: 6
 Рассчитать функцию при x: 0.263

Значения функции

Индекс	0	1	2	3	4	5
X	0.05	0.1	0.17	0.25	0.3	0.36
Y	0.0500	0.1003	0.1716	0.2553	0.3093	0.3764

Вспомогательная таблица

Индекс	Разнос						Di	Yi / Di^2
0	0.213	-0.05	-0.12	-0.2	-0.25	-0.31	-1.9809E-2526.2	
1	0.05	0.163	-0.07	-0.15	-0.2	-0.26	4.4499E-22547.1	
2	0.12	0.07	0.093	-0.08	-0.13	-0.19	-1.5436E-11120	
3	0.2	0.15	0.08	0.013	-0.05	-0.11	1.716E-148800	
4	0.25	0.2	0.13	0.05	-0.037	-0.06	7.21499428740	

Кнопки управления: Расчет, Выход

Промежуточные вычисления: Произведение по диагонали П: 1.50649247619E-7
 Сумма (yi/Di): 1787173.94568529

Результат расчета: 0.269236410281768
 Калькулятор

Ответ. Значение функции, заданной таблично, при $x = 0,263$ равно $f(x) = 0,269236$.

Разделенные разности

Существует множество разностных методов интерполяции. Однако наиболее распространен метод Ньютона для интерполяции вперед, который в литературе известен как *метод Ньютона–Грегори*. Интерполяционный многочлен этого метода имеет вид

$$P_n(x) = c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) + \dots + c_n(x - x_0)(x - x_1) \cdot \dots \cdot (x - x_{n-1}).$$

Коэффициенты c_i позволяющие записать систему находят из уравнений $P_n(x) = y_i, i = 0, 1, \dots, n$:

$$\begin{aligned} c_0 &= y_0; \\ c_0 + c_1(x_1 - x_0) &= y_1; \\ c_0 + c_1(x_2 - x_0) + c_2(x_2 - x_0)(x_2 - x_1) &= y_2; \\ &\dots \\ c_0 + \dots + c_n(x_n - x_0)(x_n - x_1) \dots (x_n - x_{n-1}) &= y_n. \end{aligned}$$

Это линейная система уравнений с треугольной матрицей. Вычисление коэффициентов c_i по этой системе не вызывает затруднений.

Однако существует и более простой способ определения коэффициентов c_i , основанный на применении правых конечных разностей.

Если значения x заданы через равные промежутки $h = x_{i+1} - x_i$, то в общем случае $x_i = x_0 + ih$, где $i = 1, 2, \dots, n$. Это выражение позволяет привести исходные уравнения к виду

$$\begin{aligned} y_0 &= c_0; \\ y_1 &= c_0 + c_1 h; \\ y_2 &= c_0 + 2hc_1 + 2h^2 c_2; \\ &\dots\dots\dots \\ y_i &= c_0 + c_i h + c_2 i h((i-1)h) + \dots + c_i (i!) h, \end{aligned}$$

откуда для коэффициентов c_i получаются следующие выражения:

$$\begin{aligned} c_0 &= y_0; \\ c_1 &= \frac{y_1 - c_0}{h} = \frac{y_1 - y_0}{h} = \frac{\Delta y_0}{h}. \end{aligned}$$

Здесь Δy_0 называется *первой разностью*. Продолжая вычисления, находим

$$\begin{aligned} c_2 &= \frac{1}{2h^2} (y_2 - c_0 - 2hc_1) = \frac{1}{2h^2} \left(y_2 - y_0 - 2h \frac{y_1 - y_0}{h} \right) = \\ &= \frac{1}{2h^2} (y_2 - y_0 - y_1 - y_1 + y_0 + y_0) = \frac{1}{2h^2} ((y_2 - y_1) - (y_1 - y_0)) = \frac{1}{2h^2} \Delta^2 y_0, \end{aligned}$$

где $\Delta^2 y_0$ – *вторая правая разность*, представляющая собой разность разностей.

Коэффициент c_i можно представить в виде

$$c_i = \frac{\Delta^i y_0}{(i!) h^i}.$$

В общем случае разности более высоких порядков для функции $y = f(x)$ в интервале $x_0 \leq x \leq x_n$ определяются выражением

$$\Delta^j y_i = \Delta^{j-1} y_{i+1} - \Delta^{j-1} y_i,$$

где $i = 0, 1, 2, \dots, n - j$.

Пример 5.5

Пусть имеется следующая таблица данных, табл. 5.6.

Таблица 5.6

Таблица значений функции $y = \sin x$

x_i , град	10	20	30	40	50	60
y_i	0,17365	0,34202	0,50	0,64279	0,76604	0,86603

Требуется определить значение функции при $x = 23^\circ$.

Решение

Расчет выполним в табличном виде, табл. 5.7.

Таблица 5.7

x_i	y_i	Δy_i	$\Delta^2 y_i$	$\Delta^3 y_i$	$\Delta^4 y_i$	$\Delta^5 y_i$
10	0,17365					
		0,16837				
20	0,34202		-0,01039			
		0,15798		-0,0048		
30	0,5		-0,01519		-0,00045	
		0,14279		-0,00435		0,00018
40	0,64279		-0,01954		0,00063	
		0,12325		-0,00372		
50	0,76604		-0,02326			
		0,09999				
60	0,86603					

За x_0 можно принять любой x_i , например $x_0 = 20^\circ$. Необходимые разности стоят на диагонали, идущей от x_0 вниз. Количество используемых разностей высших порядков может быть любым, но чем оно больше, тем выше точность полученного результата.

Одно из достоинств рассматриваемого метода в том, что он позволяет уточнять результат, используя дополнительные разности, причем нет необходимости вычисления начинать сначала. Поэтому в случае, когда неизвестно, сколько членов следует брать, их количество можно увеличивать до тех пор, пока их вклад не станет пренебрежительно мал.

В нашем примере шаг $h = 10^\circ$. Используя только первую разность, найдем

$$y(23)' = y_{20} + \frac{\Delta y_0}{h}(x_{23} - x_0) = 0,34202 + \frac{0,15798}{10}(23 - 20) = 0,38941.$$

Введя вторую разность, получим

$$y(23)'' = y(23)' + \frac{\Delta^2 y_0}{2h^2}(23 - 20) = 0,38941 + \frac{-0,01519}{2 \cdot 10^2} \cdot 3 \cdot (-7) = 0,39100.$$

Наконец, с помощью третьей разности найдем

$$\begin{aligned} y(23)''' &= y(23)'' + \frac{\Delta^3 y_0}{6h^3}(23 - 20)(23 - 30)(23 - 40) = \\ &= 0,39100 + \frac{-0,00435}{6 \cdot 10^3} \cdot 3 \cdot (-7) \cdot (-17) = 0,39074. \end{aligned}$$

Это значение $y(23)$ очень близко к точному значению, равному 0,39073.

Ответ. Значение функции, заданной таблично, при $x = 23^\circ$ равно 0,39074.

Таким образом, рассмотренные методы интерполяции и программы, разработанные на языке Delphi, позволяют вычислять значения функций, заданных таблично, при любом значении аргумента, находящегося в пределах $x_0 \leq x \leq x_n$.

Обратное интерполирование

Это алгоритм, с помощью которого находят значение аргумента x , соответствующего заданному значению функции y , которое лежит между двумя ее значениями, приведенными в таблице.

Чтобы применить рассмотренные выше методы интерполяции, нужно видоизменить исходную таблицу, поменяв в ней x на y .

Недостаток этого метода в том, что теперь значения аргумента не будут расположены через равные интервалы, что исключает использование тех методов, для которых шаг аргумента **обязательно должен быть постоянным**.

Варианты заданий

С помощью линейной интерполяции, интерполяционного многочлена Лагранжа и разделенных разностей, используя разработанные программы на языке Delphi (линейная интерполяция, интерполяционный многочлен Лагранжа), найти приближенное значение функции при данном значении аргумента, табл. 5.8–5.13.

Таблица 5.8

x	y
1,375	5,04192
1,380	5,17744
1,385	5,32016
1,390	5,47069
1,395	5,62968
1,400	5,79788

Номер варианта	Значение аргумента
1	1,3832
7	1,3926
13	1,3862
19	1,3934
25	1,3866

Таблица 5.9

x	y
0,115	8,65729
0,120	8,29329
0,125	7,95829
0,130	7,64893
0,135	7,36235
0,140	7,09613

Номер варианта	Значение аргумента
2	0,1264
8	0,1315
14	0,1232
20	0,1334
26	0,1285

Таблица 5.10

x	y
0,150	6,61659
0,155	6,39989
0,160	6,19558
0,165	6,00551
0,170	5,82558
0,175	5,05583

Номер варианта	Значение аргумента
3	0,1521
9	0,1611
15	0,1662
11	0,1542
27	0,1625

Таблица 5.11

x	y
0,180	5,61548
0,185	5,46693
0,190	5,32634
0,195	5,10304
0,200	5,06649
0,205	4,94619

Номер варианта	Значение аргумента
4	0,1838
10	0,1875
16	0,1944
22	0,1976
28	0,2038

Таблица 5.12

x	y
0,210	4,83170
0,215	4,72261
0,220	4,61855
0,225	4,51919
0,230	4,42422
0,235	4,33337

Номер варианта	Значение аргумента
5	0,2121
11	0,2165
17	0,2332
23	0,2363
29	0,2244

Таблица 5.13

x	y
1,415	0,888551
1,420	0,889599
1,425	0,890637
1,430	0,891667
1,435	0,892687
1,440	0,893698

Номер варианта	Значение аргумента
6	1,4179
12	1,4258
18	1,4396
24	1,4236
30	1,4315

Лабораторная работа № 6

МАТЕМАТИЧЕСКАЯ ОБРАБОТКА ДАННЫХ

Постановка задачи

Анализ технических задач и процессов приводит к необходимости выделения существенных факторов, влияющих на исследуемый объект, а также к выбору вида связи между факторами и оценке параметров полученных уравнений связи.

Будем считать, что некоторый процесс характеризуется двумя варьируемыми параметрами x и y , из которых x выбирается как независимая величина, а y – как зависимая переменная. Предположим, что между переменными x и y существует однозначное соответствие, т. е. каждому значению независимой переменной x соответствует с заданной степенью точности одно значение зависимой переменной y .

Существуют два основных подхода к определению коэффициентов уравнений связи для данных, заданных таблично.

При одном из них требуют, чтобы аппроксимирующая кривая (возможно, кусочно-гладкая) **проходила через все точки, заданные таблицей**. Это удается сделать с помощью методов интерполяции, рассмотренных ранее.

При другом подходе табличные данные аппроксимируют простой функцией, с достаточной точностью аппроксимирующей табличные данные во всем диапазоне, **но не обязательно проходящей через все точки**. Такой подход называется *подгонкой кривой*, которую стремятся провести так, чтобы ее отклонения от табличных данных были минимальными. Обычно сумму квадратов разностей между значениями функции, определяемыми выбранной кривой (расчетные значения) и таблицей (табличные или экспериментальные значения), *стремятся свести к минимуму*. Такой метод подгонки называется методом наименьших квадратов.

Метод наименьших квадратов

В случае использования для приближения функции $y = f(x)$, заданной таблицей, методом наименьших квадратов находят либо линейную функцию

$$g(x) = ax + b,$$

либо квадратическую зависимость

$$g(x) = ax^2 + bx + c.$$

При этом требуется, чтобы сумма квадратов отклонений значений заданной функции $f(x)$ от значений найденной аппроксимирующей функции $g(x)$, как указывалось выше, в заданных точках была наименьшей.

Пусть в таблице задана $n + 1$ точка $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$ и требуется найти аппроксимирующую кривую $g(x)$ в диапазоне $x_0 \leq x \leq x_n$. В этом случае погрешность ε_i в каждой табличной точке будет $\varepsilon_i = g(x_i) - y_i$. Тогда сумма квадратов погрешностей определяется выражением

$$E = \sum_{i=0}^n [g(x_i) - y_i]^2.$$

Обычно функцию $g(x)$ выбирают в виде линейной комбинации подходящих функций

$$g(x) = c_1 g_1(x) + c_2 g_2(x) + \dots + c_n g_n(x).$$

Условие минимума величины E определяется выражением

$$\frac{\partial E}{\partial c_1} = \frac{\partial E}{\partial c_2} = \dots = \frac{\partial E}{\partial c_n} = 0,$$

т. е. равенство нулю частных производных по коэффициентам c_i .

Поскольку

$$E = \sum_{i=0}^n [c_1 g_1(x_i) + c_2 g_2(x_i) + \dots + c_k g_k(x_i) - y_i]^2,$$

то это условие эквивалентно системе k уравнений

$$\frac{\partial E}{\partial c_1} = 2 \sum [c_1 g_1(x_i) + \dots + c_k g_k(x_i) - y_i] g_1(x_i) = 0;$$

.....

$$\frac{\partial E}{\partial c_k} = 2 \sum [c_1 g_1(x_i) + \dots + c_k g_k(x_i) - y_i] g_k(x_i) = 0.$$

По результатам эксперимента определяют коэффициент сцепления φ_{\max} и коэффициент кривизны кривой буксования k .

Таким образом, мы имеем дело со следующей задачей: функциональная зависимость между переменными величинами x и y задана по результатам эксперимента или исходя из тех или иных теоретических соображений. Выражение (формула), описывающее эту зависимость, содержит **постоянные коэффициенты, которые необходимо определить по результатам эксперимента или расчета.**

По сути дела, предыдущие лабораторные работы, в которых рассматривались вопросы интерполяции и аппроксимации, были посвящены решению именно этой задачи лишь в том предположении, что функциональная зависимость имеет линейный или параболический вид, т. е. вид многочлена. Поэтому рассмотрение случаев, когда зависимость между x и y выражается не многочленом, а любой математической зависимостью, будет иметь для нас особый интерес.

Правда, как известно, любую функцию можно с достаточной степенью точности заменить интерполяционным многочленом. Однако чтобы добиться достаточно хорошего совпадения расчетных и табличных данных, может потребоваться разработка многочлена сравнительно высокой степени, не говоря уже о том, что такой многочлен может быть не удобен в его использовании ввиду громоздкости, а коэффициенты такого многочлена, скорее всего, не будут иметь физического смысла.

Рассмотрим применение метода наименьших квадратов для определения коэффициентов в формулах различного вида.

Часто при обработке экспериментальных данных приходится встречаться и с более сложной задачей: в результате эксперимента получен ряд значений переменных x и y . Однако характер функциональной зависимости между ними **остаётся неизвестным**. Требуется по экспериментальным данным найти или подобрать аналитическое выражение, описывающее зависимости между x и y . Такие формулы принято называть *эмпирическими*

После сказанного выше должно быть совершенно ясно, что однозначно восстановить функциональную зависимость между x и y по конечному числу экспериментальных данных не всегда возможно даже в том случае, если бы они не обладали ошибками, свойственными экспериментальным данным. Тем более не следует ожидать,

что это удастся сделать, имея в распоряжении экспериментальные данные, наверняка содержащие случайные ошибки измерений.

Поэтому следует отчетливо представлять, что математическая обработка результатов эксперимента не может ставить перед собой задачу распознать истинный характер зависимостями между имеющимися переменными, а занимается задачей значительно более скромной. Речь идет лишь о том, чтобы описать результаты эксперимента наиболее простой формулой, которая давала бы возможность проводить интерполирование. Это, в свою очередь, позволит применить к экспериментальным данным методы математического анализа.

Построение линейных зависимостей

Построение зависимости $y = ax$

Рассмотрим следующую задачу. Пусть на некотором предприятии в течение рабочего дня поступают сведения о количестве выпускаемых изделий (табл. 6.1). Требуется выявить вид эмпирической зависимости и вычислить коэффициенты этой зависимости.

Таблица 6.1

Текущий час	0	1	1,5	2,5	3	4,5	5	6
Число изделий на i -й час	0	67	101	168	202	301	334	404

Нанесем данные таблицы на график (рис. 6.1), который будет иллюстрировать количество выпускаемых изделий от времени.

Из графика видно, что точки табл. 6.1 могут быть аппроксимированы прямой, проходящей через начало координат. Ее математический вид $y = ax$. Определив вид связи между переменными x и y , вычислим коэффициент a исходной линейной зависимости $y = ax$ из условия минимума суммы квадратов отклонений табличных значений y_i от расчетных y_{pi} :

$$\sum_{i=1}^n (y_i - y_{pi})^2 .$$

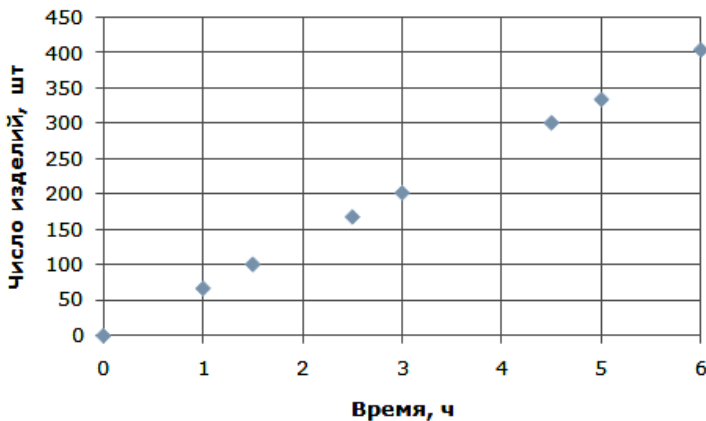


Рис. 6.1. Выпуск изделий во времени

Выразим y_{pi} через его значение ax_i и подставим в предыдущее выражение. Тогда получим

$$\sum_{i=1}^n (y_i - ax_i)^2 = \min. \quad (6.1)$$

Условие (6.1), согласно которому будем вычислять коэффициент a , включает все табличные значения независимой и зависимой переменных величин x_i и y_i . Минимальное значение суммы квадратов отклонений расчетных значений y_{pi} от табличных y_i при вариации коэффициента a найдем из условия равенства нулю производных по параметру (коэффициенту) a от функции

$$F = \sum_{i=1}^n (y_i - ax_i)^2.$$

Дифференцируя это выражение, имеем

$$\frac{\partial F}{\partial a} = 2 \sum_{i=1}^n x_i (y_i - ax_i) = 0.$$

Раскрывая скобки и представляя сумму разностей как разность сумм, получим следующее выражение:

$$\sum_{i=1}^n x_i y_i - a \sum_{i=1}^n x_i^2 = 0,$$

откуда находим выражение для вычисления коэффициента a :

$$a = \frac{\sum_{i=1}^n x_i y_i}{\sum_{i=1}^n x_i^2}. \quad (6.2)$$

Для вычисления коэффициента a по выражению (6.2) составим вспомогательную таблицу (табл. 6.2).

Таблица 6.2

x_i	y_i	$x_i y_i$	x_i^2
1	67,0	67,0	1,0
1,5	101,0	151,5	2,25
2,5	168,0	420,0	6,25
3	202,0	606,0	9,0
4,5	301,0	1354,5	20,25
5	334,0	1670,0	25,0
6	404,0	2424,0	36,0
		$\sum x_i y_i = 6693,0$	$\sum x_i^2 = 99,75$

Теперь по формуле (6.2) находим

$$a = \frac{6693,0}{99,75} = 67,09.$$

Итак, в результате расчетов получили линейную функцию $y = 67,09x$.

На рис. 6.2 показаны результаты расчета по выражению $y = 67,09x$. На этом же графике нанесены табличные данные. На рис. 6.3 изобра-

жен график изменения погрешности аппроксимации. Если из рис. 6.2 можно заключить, что табличные и расчетные значения имеют хорошие совпадения, то рис. 6.3 также подтверждает это заключение – погрешность не превышает 0,45 %.

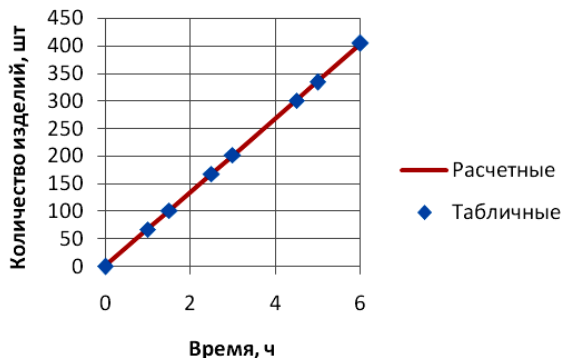


Рис. 6.2. Результаты аппроксимации



Рис. 6.3. Погрешность

Теперь определим коэффициент a для линейной функции $y = ax$, используя в пакете Excel метод «Поиск решения». В ячейках A3:A10 разместим значения x , а в ячейках B3:B10 – значения y (исходные данные). В ячейку A13 поместим предварительное значение коэффициента $a = 1$. В ячейках C3:C10 поместим результат вычисления значения $y = \$A\$13*A3$.

В столбце D помещаем разность между табличными данными и расчетными, т. е. формулу B3–C3. В ячейке C13 помещаем сумму

погрешностей, т. е. формулу = СУММКВ(D3:D10), и выполняем настройку решателя «Поиск решения»:

- в качестве целевой ячейки указываем ячейку C13;
- устанавливаем **минимальное значение** целевой ячейки;
- в качестве изменяемой ячейки указываем ячейку \$A\$13 и нажимаем кнопку «**Выполнить**» (рис. 6.4).

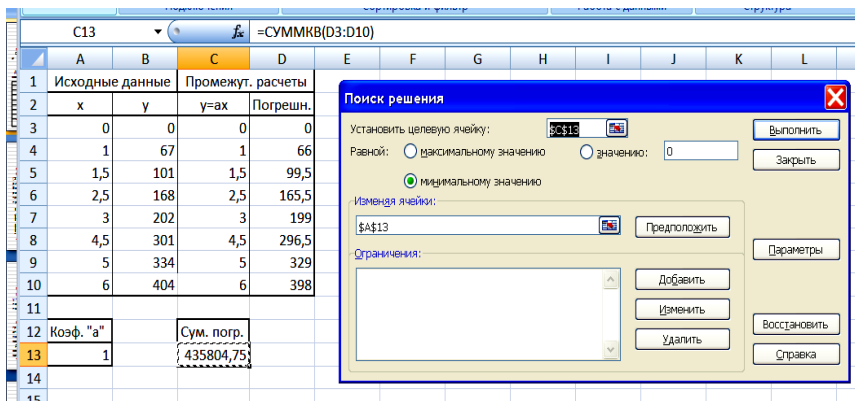


Рис. 6.4. Настройка метода «Поиск решения»

Результат расчета коэффициента a , рис. 6.5, помещается в ячейку A13 ($a = 67,09774$), в ячейке C13 – сумма квадратов погрешностей между табличными и расчетными данными ($\sum \Delta^2 = 5,79699$).

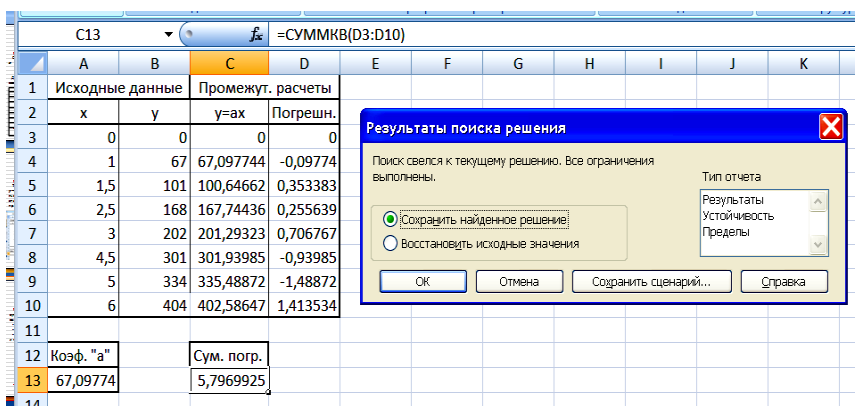


Рис. 6.5. Результат решения

Построение линейной зависимости общего типа $y = ax + b$

Рассмотрим следующую задачу. Требуется подобрать аппроксимирующую формулу табличных данных, полученных в результате эксперимента, табл. 6.3.

Таблица 6.3

x_i	1,0	1,5	2,0	2,5	3,0	3,5
y_i	0,12	1,26	2,32	3,25	4,35	5,52

Табличные данные, как и в первом случае, нанесем на график (рис. 6.6) для определения типа аппроксимирующего выражения. Из графика видно, что имеет место линейная зависимость, не проходящая через начало координат. Ее аналитическое выражение $y = ax + b$.

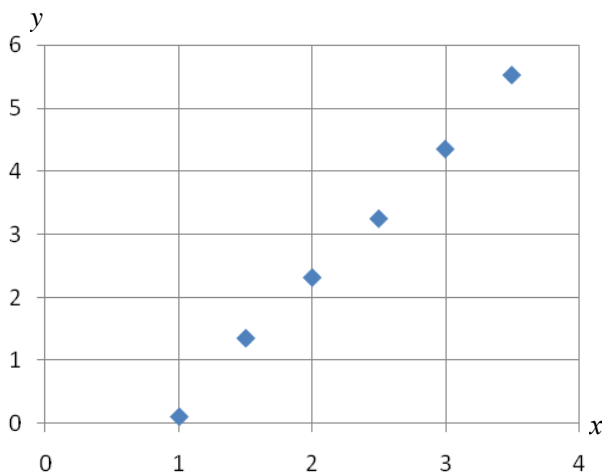


Рис. 6.6. График экспериментальной зависимости

Для расчета коэффициентов a и b , как и прежде, воспользуемся методом наименьших квадратов, согласно которому для выбранной прямой должно выполняться условие

$$\sum_{i=1}^n [y_i - (ax_i + b)]^2 = \min.$$

Используя необходимое условие экстремума функции нескольких переменных при вариации параметров a и b , получим

$$\frac{\partial}{\partial a} \sum_{i=1}^n (y_i - ax_i - b)^2 = 0;$$

$$\frac{\partial}{\partial b} \sum_{i=1}^n (y_i - ax_i - b)^2 = 0$$

или

$$\begin{cases} \sum_{i=1}^n 2(y_i - ax_i - b)(-x_i) = 0; \\ \sum_{i=1}^n 2(y_i - ax_i - b)(-1) = 0, \end{cases}$$

откуда

$$\begin{cases} \sum_{i=1}^n x_i y_i = a \sum_{i=1}^n x_i^2 + b \sum_{i=1}^n x_i; \\ \sum_{i=1}^n y_i = a \sum_{i=1}^n x_i + ab, \end{cases}$$

где n – число наблюдений (точек).

Чтобы вычислить коэффициенты линейной функции для нашего примера, выполним ряд вспомогательных расчетов: $\sum x_i = 13,5$, $\sum y_i = 16,82$. Далее имеем выражения, представленные в табл. 6.4.

Таблица 6.4

x_i^2	1,0	2,25	4,0	6,25	9,0	12,25
$x_i y_i$	0,12	1,89	4,64	8,16	13,05	19,32

Таким образом, получили систему уравнений

$$\begin{cases} 34,75a + 13,5b = 47,18; \\ 13,5a + 6b = 16,82, \end{cases}$$

из которой находим $a = 2,13$; $b = -1,99$. Следовательно, искомая функция имеет вид

$$g(x) = 2,13x - 1,99 \text{ (рис. 6.7),}$$

и в каждой точке вычислим отклонения расчетных значений от табличных данных (рис. 6.8).

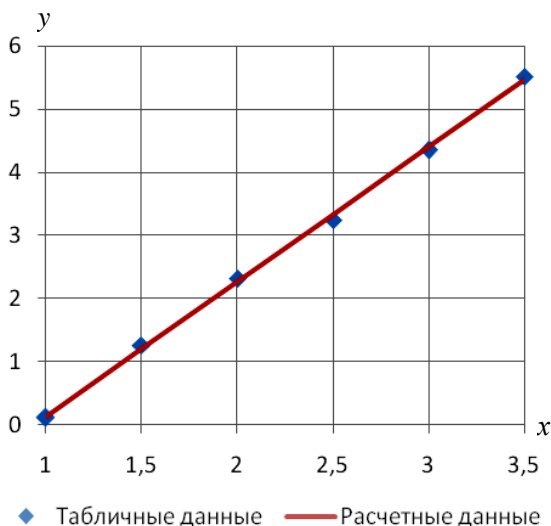


Рис. 6.7. Результаты аппроксимации

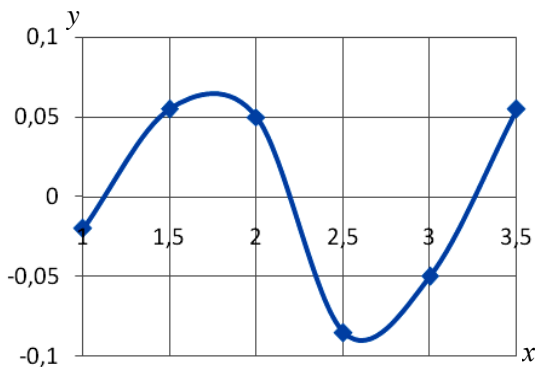


Рис. 6.8. Погрешность

Из рис. 6.8 видно, что наибольшее отклонение расчетных данных от экспериментальных составляет 0,090, т. е. результаты аппроксимации вполне удовлетворительные.

Выполним аппроксимацию с помощью метода «Поиск решения» пакета Excel. Настройки показаны ниже, рис. 6.9.

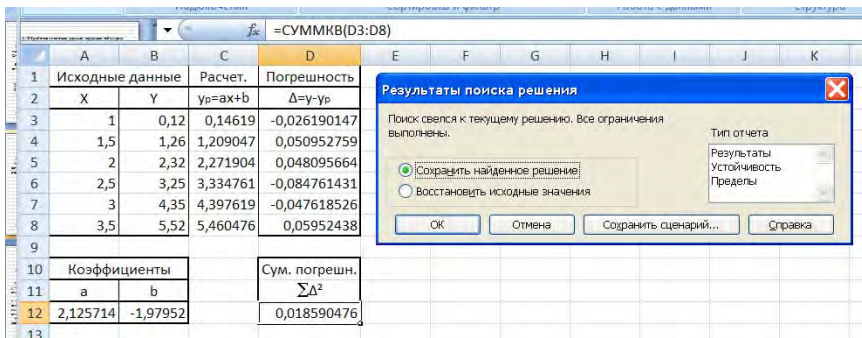


Рис. 6.9. Настройки метода «Поиск решения»

Результаты выполнения «Поиска решения» следующие:

– коэффициенты

$$b = -1,97952,$$

$$a = 2,125714;$$

– сумма квадратов отклонений

$$\Sigma\Delta^2 = 0,018590476.$$

Таким образом, как видно из полученных результатов, коэффициенты аппроксимации, вычисленные ручным методом (рекомендуется использовать пакет MathCAD) и с помощью пакета Excel, имеют хорошее совпадение.

Нелинейная зависимость

Построение квадратической зависимости общего вида

$$y = ax^2 + bx + c$$

Как и ранее, запишем отклонения расчетных значений от табличных:

$$v_i = y_i - (ax_i^2 + bx_i + c), \quad i = 0, 1, \dots, n,$$

и полагаем, что искомая функция $g(x)$ будет незначительно отличаться от табличной функции $f(x)$, если достигается минимум суммы квадратов отклонений расчетных и табличных значений $F = \sum_{i=0}^n v_i^2$.

Найдем частные производные от функции $g(x)$ по коэффициентам a, b, c и приравняем их к нулю:

$$\frac{\partial F}{\partial a} = \sum_{i=0}^n 2 \left[y_i - (ax_i^2 + bx_i + c) \right] (-x_i^2) = 0;$$

$$\frac{\partial F}{\partial b} = \sum_{i=0}^n 2 \left[y_i - (ax_i^2 + bx_i + c) \right] (-x_i) = 0;$$

$$\frac{\partial F}{\partial c} = \sum_{i=0}^n 2 \left[y_i - (ax_i^2 + bx_i + c) \right] (-1) = 0.$$

Получаем систему нормальных уравнений

$$\begin{cases} a \sum_{i=0}^n x_i^4 + b \sum_{i=0}^n x_i^3 + c \sum_{i=0}^n x_i^2 = \sum_{i=0}^n x_i^2 y_i; \\ a \sum_{i=0}^n x_i^3 + b \sum_{i=0}^n x_i^2 + c \sum_{i=0}^n x_i = \sum_{i=0}^n x_i y_i; \\ a \sum_{i=0}^n x_i^2 + b \sum_{i=0}^n x_i + cn = \sum_{i=0}^n y_i. \end{cases} \quad (6.3)$$

Решив систему уравнений (6.3), найдем искомые коэффициенты a, b и c функции $g(x)$.

Пример 6.1

Подобрать аппроксимирующую функцию для данных, заданных таблично, табл. 6.5.

Таблица 6.5

x_i	1,0	1,5	2,0	2,5	3,0	3,5
y_i	2,62	3,09	4,17	5,87	8,18	11,11

Нанесем табличные данные на график, рис. 6.10.

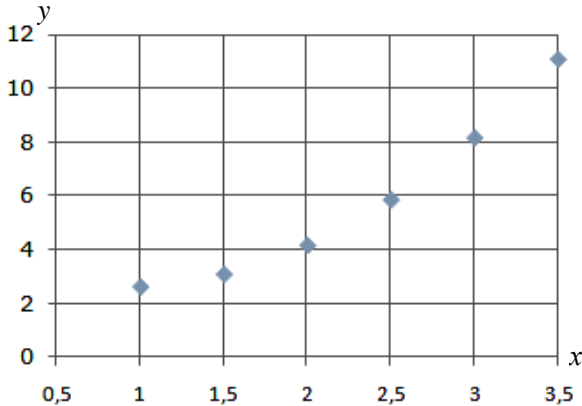


Рис. 6.10. График экспериментальных данных

Из графика видно, что точки располагаются примерно по параболической зависимости. Следовательно, их можно аппроксимировать квадратным полиномом.

Из нормальной системы уравнений (6.3) следует, что для определения коэффициентов a , b и c необходимо вычислить $\sum x_i = 13,5$; $\sum y_i = 35,054$. Для дальнейших вычислений составим вспомогательную таблицу, табл. 6.6.

Таблица 6.6

x_i^2	1,0	2,25	4,0	6,25	9,0	12,25
x_i^3	1,0	3,37	8,0	15,62	27,0	42,87
x_i^4	1,0	5,05	16,0	39,05	81,0	150,04
$x_i y_i$	2,62	4,63	8,34	14,67	24,54	38,88
$x_i^2 y_i$	2,62	6,95	16,68	36,69	73,62	136,10

Воспользовавшись таблицей, найдем суммы: $\sum x_i^2 = 34,75$; $\sum x_i^3 = 97,875$; $\sum x_i^4 = 292,1875$; $\sum x_i y_i = 93,695$; $\sum x_i^3 y_i = 272,6575$.

Следовательно, имеем следующую систему линейных алгебраических уравнений:

$$\begin{cases} 292,19a + 97,88b + 34,75c = 272,66; \\ 97,88a + 34,75b + 13,50c = 93,70; \\ 34,75a + 13,50b + 6c = 35,05. \end{cases}$$

Решая эту систему, находим $a = 1,25$; $b = -2,25$; $c = 3,64$.

Таким образом, искомая функция имеет вид

$$g(x) = 1,25x^2 - 2,25x + 3,64.$$

Построим график функции $g(x)$ и нанесем на него табличные данные, рис. 6.11.

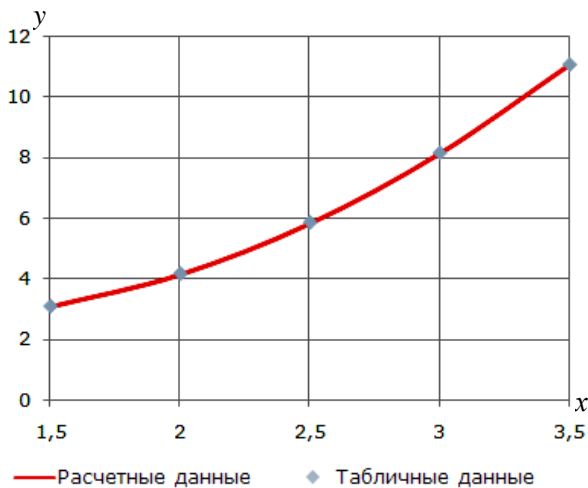


Рис. 6.11. Результат аппроксимации

Кроме того, вычислим величины отклонений для каждой точки и построим график отклонений расчетных значений функции $g(x)$ от табличных значений функции $f(x)$, рис. 6.12.

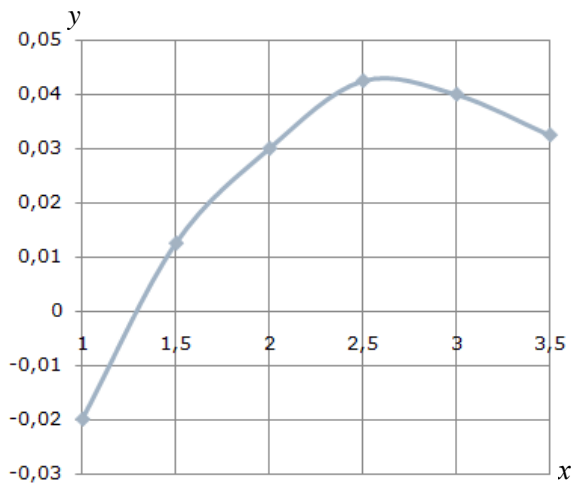


Рис. 6.12. Погрешность

Анализ графических зависимостей на рис. 6.11 и 6.12 позволяет сделать заключение о хорошей аппроксимации табличных данных $f(x)$ функцией $g(x)$. Отклонение расчетных и табличных данных – не более 0,042.

Повторим вычисления коэффициентов a , b и c функции $y = ax^2 + bx + c$ с помощью пакета Excel, используя метод «Поиск решения». Настройка метода «Поиск решения» и результат расчёта показаны на рис. 6.13 и 6.14.

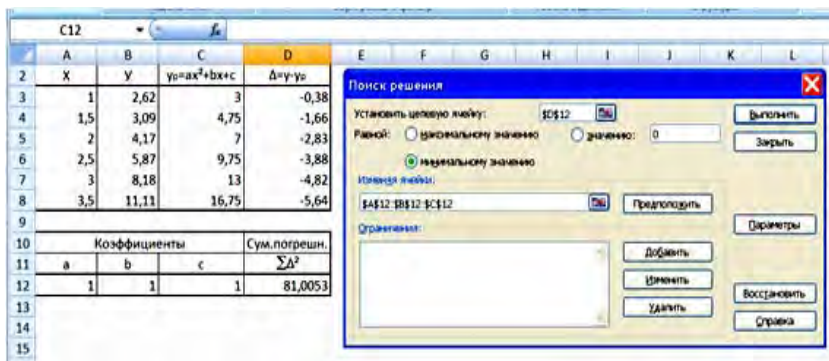


Рис. 6.13. Настройка метода

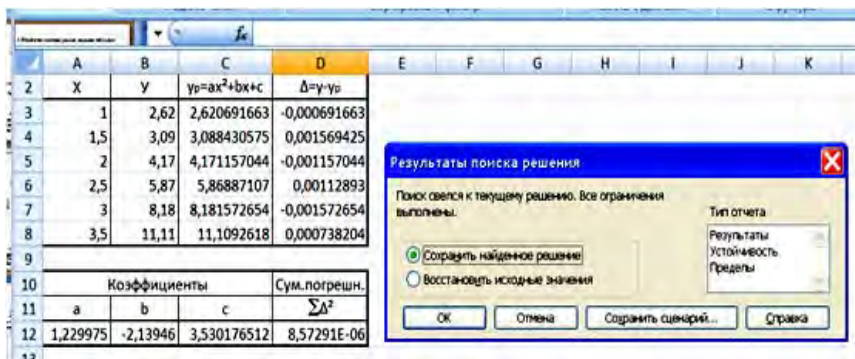


Рис. 6.14. Результаты решения

Результаты решения системы уравнений следующие:

$$a = 1,229975; b = -2,13946; c = 3,5301765; \Sigma \Delta^2 = 8,57291E-06.$$

Следовательно, с помощью пакета Excel получен более точный результат, так как $\Sigma \Delta^2 \approx 0$.

Выбор вида эмпирической формулы

Ранее были рассмотрены линейные зависимости, для которых построенный график определял выбор функциональной зависимости. Так, для прямой, проходящей через начало координат, аналитическим выражением являлась функция $y = ax$, а для прямой общего положения – функция $y = ax + b$. Однако графическое построение нелинейной зависимости не дает ответа на вопрос о том, какой аналитический вид имеет эта функция, т. е. будет ли эта зависимость степенной, дробно-рациональной, логарифмической или иметь другой вид.

Пусть y есть функция одной переменной с двумя коэффициентами a и b . В качестве набора функций, из которых будем выбирать эмпирическую зависимость, рассмотрим:

- линейную функцию $y = ax + b$;
- показательную функцию $y = ab^x$;
- дробно-рациональную функцию $y = 1/(ax + b)$;
- логарифмическую функцию $y = a \ln(x) + b$;

– степенную функцию $y = ax^b$ (она определяет параболическую зависимость при $b > 0$ и гиперболическую зависимость, если $b < 0$, если же параметр $b = 0$, то зависимость вырождается в линейную);

– гиперболическую функцию вида $y = a + b/x$;

– дробно-рациональную функцию вида $y = x/(ax + b)$.

Для наилучшего выбора подбираемой функции $y = f(x, a, b)$, соответствующей табличным данным и построенному графику, выполним следующие промежуточные вычисления. На заданном отрезке изменения независимой переменной x выберем точки, достаточно надежные и по возможности далеко отстоящие друг от друга. Будем считать, что это точки x_1 и x_n , т. е. первая и последняя точки. Вычислим их среднее арифметическое $x_{\text{ср}} = (x_1 + x_n)/2$, среднее геометрическое $x_{\text{ср.геом}} = \sqrt{x_1 x_n}$ и среднее гармоническое $x_{\text{ср.гарм}} = 2x_1 x_n / (x_1 + x_n)$. По вычисленным значениям независимой переменной x из построенного графика найдем соответствующие значения зависимой переменной y : $x_{\text{ср}} \rightarrow y_1^*$; $x_{\text{ср.геом}} \rightarrow y_2^*$; $x_{\text{ср.гарм}} \rightarrow y_3^*$.

1. Если наименьшей среди всех абсолютных погрешностей окажется ε_1 , то в качестве аналитической зависимости для данного графика хорошим приближением служит линейная функция $y = ax + b$.

2. Если наименьшей абсолютной погрешностью является ε_2 , то в качестве эмпирической зависимости следует брать показательную функцию $y = ab^x$.

3. В том случае, когда наименьшая из абсолютных погрешностей есть ε_3 , искомая эмпирическая зависимость определяется дробно-рациональной функцией вида $y = 1/(ax + b)$.

4. Если наименьшей из абсолютных погрешностей будет ε_4 , то хорошим приближением служит логарифмическая функция $y = a \ln x + b$.

5. Для случая, когда наименьшей абсолютной погрешностью является ε_5 , в качестве эмпирической зависимости выбирается степенная функция $y = ax^b$.

6. Если наименьшей из абсолютных погрешностей окажется ε_6 , то за искомую зависимость следует выбирать гиперболическую $y = a + b/x$.

7. Наконец, в случае, когда наименьшая из абсолютных погрешностей есть ε_7 , в качестве аналитической зависимости выбирается дробно-рациональная функция вида $y = x/(ax + b)$.

Эмпирические формулы, содержащие два параметра

Пример 6.2

Для функции, заданной таблично (табл. 6.7), подобрать эмпирическую зависимость.

Таблица 6.7

x_i	1	2	3	4	5	6	7	8	9
y_i	521,0	308,0	240,5	204,0	183,0	171,0	159,0	152,0	147,0

Решение

1. Предположим, что в таблице крайние данные достаточно надежны. Выполним вспомогательные вычисления и для крайних значений независимой переменной $x = 1$ и $x = 9$ найдем:

среднее арифметическое

$$x_{\text{ср}} = (1 + 9)/2 = 5,$$

среднее геометрическое

$$x_{\text{ср.геом}} = \sqrt{1 \cdot 9} = 3$$

и среднее гармоническое

$$x_{\text{ср.гарм}} = 2 \cdot 1 \cdot 9 / (1 + 9) = 1,8.$$

2. По табличным данным построим график (рис. 6.15), из которого найдем приблизительные значения функции, т. е. y^* для $x_{\text{ср}}$, $x_{\text{ср.геом}}$ и $x_{\text{ср.гарм}}$: $y_1^* \approx 180$; $y_2^* \approx 240$; $y_3^* \approx 341$.

3. Выполним дополнительные расчеты для зависимой переменной y . Для крайних значений найдем:

среднее арифметическое

$$y_{\text{ср}} = \frac{y_1 + y_9}{2} = \frac{521 + 147}{2} = 334,$$

среднее геометрическое

$$y_{\text{ср.геом}} = \sqrt{y_1 y_9} = \sqrt{521 \cdot 147} = 274$$

и среднее гармоническое

$$y_{\text{ср.гарм}} = \frac{2y_1y_9}{y_1 + y_9} = \frac{2 \cdot 512 \cdot 147}{512 + 147} = 228,419.$$

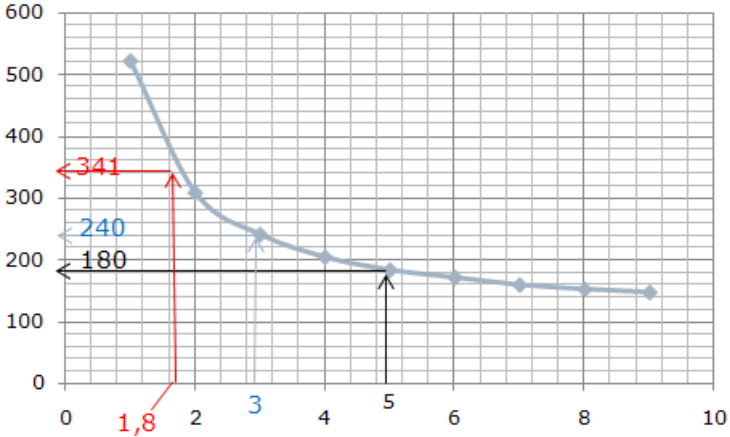


Рис. 6.15. Табличные данные

4. Сравним найденные по графику значения зависимой переменной y_i^* с $y_{\text{ср}}$, $y_{\text{ср.геом}}$ и $y_{\text{ср.гарм}}$:

$$\varepsilon_1 = |y_1^* - y_{\text{ср}}| = |180 - 334| = 154;$$

$$\varepsilon_2 = |y_1^* - y_{\text{ср.геом}}| = |180 - 274| = 94;$$

$$\varepsilon_3 = |y_1^* - y_{\text{ср.гарм}}| = |180 - 228| = 48;$$

$$\varepsilon_4 = |y_2^* - y_{\text{ср}}| = |240 - 334| = 94;$$

$$\varepsilon_5 = |y_2^* - y_{\text{ср.геом}}| = |240 - 274| = 34;$$

$$\varepsilon_6 = |y_3^* - y_{\text{ср}}| = |341 - 334| = 7;$$

$$\varepsilon_7 = |y_3^* - y_{\text{ср.гарм}}| = |341 - 228| = 113.$$

Поскольку наименьшей из рассчитанных погрешностей оказалась ε_6 , следует выбрать гиперболическую зависимость

$$y = a + \frac{b}{x}.$$

Используя пакет Excel и метод «Поиск решения», для рассмотренного примера определим коэффициенты a и b гиперболической функции. Результаты решения представлены на рис. 6.16 и 6.17.

X	Y _{таб}	Y _{рас} =a+b/x	Δ=(Y _{таб} -Y _{рас}) ²
1	521	520,28675	0,7132529
2	308	309,86155	-1,86155
3	240,5	239,71982	0,7801823
4	204	204,64895	-0,648952
5	183	183,60643	-0,606432
6	171	169,57809	1,4219146
7	159	159,55784	-0,557838
8	152	152,04265	-0,042652
9	147	146,19751	0,802492

Коэффициенты		ΣΔ ²
a	b	8,3505175
99,43635	420,8504	

Рис. 6.16. Определение коэффициентов гиперболической функции

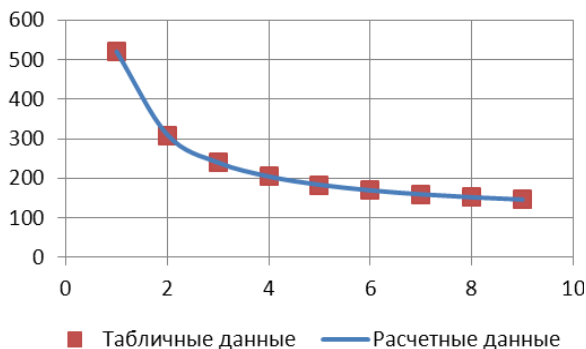


Рис. 6.17. График функции

Ответ. В результате расчета получили следующие коэффициенты гиперболической функции: $a = 99,43653$; $b = 420,8504$.

Преобразование координат

В системе координат xOy рассмотрим некоторую нелинейную зависимость $y = f(x, a, b)$, непрерывную на отрезке $[x_1, x_n]$. Перейдем к новым переменным $q = \varphi(x)$ и $z = \psi(y)$ так, чтобы в новой системе qOz эмпирическая зависимость стала линейной $z = b + a$.

Точки N_i с координатами $[\varphi(x_i), \psi(y_i)]$ в плоскости qOz должны лежать практически на прямой линии. Справедливо и обратное заключение: если при построении на плоскости qOz окажется, что точки лежат на прямой линии, то между переменными q и z имеет место линейная зависимость $\psi(y) = a \varphi(x) + b$.

Покажем, как рассмотренные выше нелинейные зависимости путем преобразования координат можно свести к линейным функциям.

1. Дробно-линейную зависимость $y = 1/(ax + b)$ можно преобразовать в линейную функцию следующим образом. Найдя обратную зависимость $1/y = ax + b$ и введя новые переменные $q = x$, $z = 1/y$, получаем линейную зависимость вида $z = aq + b$.

2. Для показательной функции вида $y = ab^x$, логарифмируя, имеем $\ln y = x \ln b + \ln a$. Полагая $\ln b = B$, $\ln a = A$, $\ln y = z$, $x = q$, в плоскости qOz получим уравнение прямой линии $z = Bq + A$.

3. Рассмотрим логарифмическую зависимость $y = a \ln x + b$. Снова введем новые переменные $q = \ln x$, $z = y$ и получаем линейную зависимость $z = aq + b$.

4. Рассмотрим степенную функцию $y = bx^a$, где $a > 0$ и $b > 0$. Логарифмируя, находим $\ln y = a \ln x + \ln b$. Откуда, полагая $z = \ln y$, $q = \ln x$, $B = \ln b$, получаем $z = aq + B$. Прямую этого вида удобно строить, если оси q и z в плоскости qOz взять в логарифмическом масштабе. Началом координат логарифмической сетки служат точки $q = 0$, $z = B$.

5. В выражении $y = a + b/x$ произведем замену переменных $q = 1/x$, $z = y$. Тем самым заданная гиперболическая зависимость преобразуется в линейную $z = a + bq$.

6. Рассмотрим дробно-рациональную функцию $y = x/(ax + b)$. Функция, обратная данной, имеет вид $1/y = a + b/x$. Введя новые координаты $z = 1/y$, $q = 1/x$, снова получаем линейную зависимость $z = a + bq$.

Пример 6.3

Получены экспериментальные данные, табл. 6.8.

Таблица 6.8

x	1	2	3	4	5	6	7	8	9
y	0,33	0,49	0,59	0,65	0,71	0,755	0,77	0,81	0,82

Подобрать эмпирическую формулу.

Решение

1. Как и раньше, выполним вспомогательные вычисления, принимая, что крайние табличные данные достаточно надежны. Для $x_1 = 1$ и $x_9 = 9$: среднее арифметическое $x_{\text{ср}} = (1 + 9)/2 = 5$; среднее геометрическое $x_{\text{ср.геом}} = \sqrt{1 \cdot 9} = 3$ и среднее гармоническое $x_{\text{ср.гарм}} = 2 \cdot 1 \cdot 9 / (1 + 9) = 1,8$.

2. Построим график (рис. 6.18) и по нему определим значения зависимой переменной y для трех значений независимой переменной: $x_{\text{ср}} = 5$, $y_1^* = 0,7$; $x_{\text{ср.геом}} = 3$, $y_2^* = 0,581$ и $x_{\text{ср.гарм}} = 1,8$, $y_3^* = 0,468$.

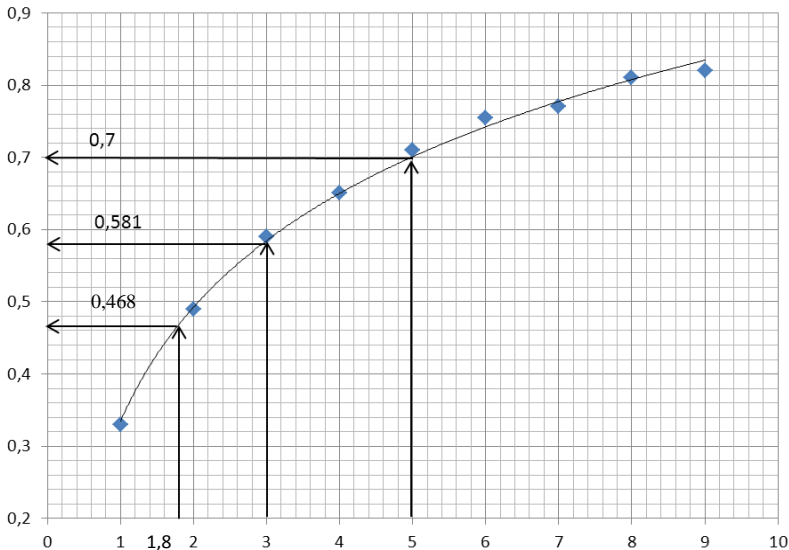


Рис. 6.18. График функции

3. Выполним дополнительные расчеты для зависимой переменной y . Рассчитаем для крайних значений:

среднее арифметическое $y_{\text{ср}} = (0,33 + 0,82)/2 = 0,575$;

среднее геометрическое

$$y_{\text{ср.геом}} = \sqrt{0,33 \cdot 0,82} = 0,52;$$

среднее гармоническое

$$y_{\text{ср.гарм}} = 2 \cdot 0,33 \cdot 0,82 / (0,33 + 0,82) = 0,471.$$

4. Сравним найденные по графику значения зависимой переменной y с $y_{\text{ср}}$, $y_{\text{ср.геом}}$ и $y_{\text{ср.гарм}}$:

$$\varepsilon_1 = |y_1^* - y_{\text{ср}}| = |0,7 - 0,575| = 0,125;$$

$$\varepsilon_2 = |y_1^* - y_{\text{ср.геом}}| = |0,7 - 0,52| = 0,18;$$

$$\varepsilon_3 = |y_1^* - y_{\text{ср.гарм}}| = |0,7 - 0,471| = 0,23;$$

$$\varepsilon_4 = |y_2^* - y_{\text{ср}}| = |0,581 - 0,575| = 0,006;$$

$$\varepsilon_5 = |y_2^* - y_{\text{ср.геом}}| = |0,581 - 0,52| = 0,061;$$

$$\varepsilon_6 = |y_3^* - y_{\text{ср}}| = |0,468 - 0,575| = 0,107;$$

$$\varepsilon_7 = |y_3^* - y_{\text{ср.гарм}}| = |0,468 - 0,471| = 0,003.$$

5. Так как наименьшей погрешностью является $\varepsilon_7 = 0,003$, то в качестве искомой функции выбираем дробно-рациональную функцию $y = x/(ax + b)$, коэффициенты которой определим с помощью пакета Excel и его метода «Поиск решения», рис. 6.19.

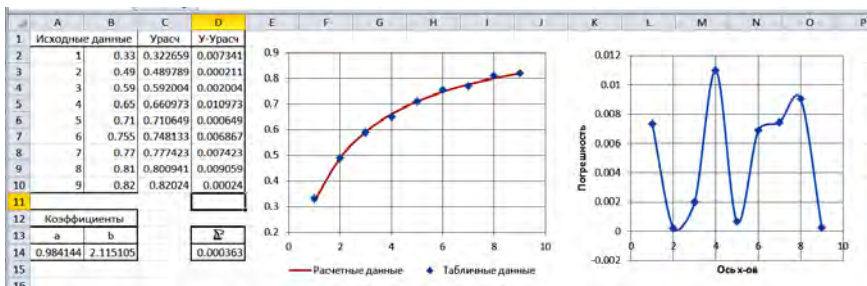


Рис. 6.19. Результат расчета коэффициентов, график аппроксимирующей дробно-рациональной функции и график погрешности

Ответ. По результатам расчета имеем: коэффициенты $a = 0,984144$; $b = 2,115105$; сумма квадратов погрешностей $\sum \Delta^2 = 0,000363$; аппроксимирующее уравнение $y = x / (0,984144x + 2,115105)$.

Эмпирические формулы, содержащие три параметра

Для заданных значений табличных данных (x_i, y_i) рассмотрим важнейших представителей эмпирических формул, содержащих три параметра (коэффициента) вида

$$y = f(x, a, b, c),$$

где a, b, c – коэффициенты, подлежащие определению.

Квадратическая зависимость

Пусть для заданных табличных данных подобрана аналитическая зависимость

$$y = ax^2 + bx + c.$$

Определение коэффициентов для этой функции методом наименьших квадратов рассмотрены ранее.

Степенная зависимость

Положим $y = ax^b + c$. Тогда можем записать

$$y - c = ax^b.$$

Логарифмируя это выражение, будем иметь

$$\ln(y - c) = \ln a + b \ln x.$$

Отсюда, полагая $\ln(y - c) = Y$ и $\ln x = X$, получим линейную функцию

$$Y = bX + \ln a.$$

Определение параметров исходной формулы следует начинать с нахождения коэффициента c . Для этого рассчитывается среднее геометрическое независимой переменной

$$x_{\text{ср.геом}} = \sqrt{x_1 x_n},$$

где x_1 и x_n – крайние табличные значения переменной x .

Построив по табличным данным график, найдем соответствующее значение $y_{\text{ср.геом}}$. Полагая, что точки $M_1(x_1, y_1)$, $M_2(x_{\text{ср.геом}}, y_{\text{ср.геом}})$ и $M_n(x_n, y_n)$ расположены на исходной кривой, будем иметь три равенства:

$$y_1 = c + ax_1^b;$$

$$y_{\text{ср.геом}} = c + ax_{\text{ср.геом}}^b;$$

$$y_n = c + ax_n^b.$$

Возводя $x_{\text{ср.геом}}$ в степень b и умножая на a , получим

$$ax_{\text{ср.геом}}^b = \sqrt{ax_1^b ax_n^b}.$$

Подставим значение $x_{\text{ср.геом}}$ в выражение для $y_{\text{ср.геом}}$, имеем

$$\begin{aligned} y_{\text{ср.геом}} = c + ax_{\text{ср.геом}}^b &= c + \sqrt{ax_1^b ax_n^b} = c + \sqrt{\frac{a(y_1 - c)}{a} \cdot \frac{a(y_n - c)}{a}} = \\ &= \sqrt{(y_1 - c)(y_n - c)}. \end{aligned}$$

Решая последнее равенство относительно коэффициента c , находим

$$c = \frac{y_1 y_n - y_{\text{ср.геом}}^2}{y_1 + y_n - 2y_{\text{ср.геом}}}.$$

Когда коэффициент c определен, строим точки $N_i(X_i, Y_i)$, где $X_i = \ln x_i$, $Y_i = \ln(y_i - c)$. Если эти точки лягут на прямую или почти на прямую, то аппроксимация заданных табличных данных степенной функцией оправдана, причем коэффициенты a и b находятся методом наименьших квадратов.

Пример 6.4

Для переменных x и y дана таблица экспериментальных значений, табл. 6.9.

Таблица 6.9

x	250	500	900	1200	1600	2000
y	0,10	0,28	0,80	1,38	2,56	4,0

Подобрать эмпирическую формулу, связывающую эти переменные выражением $y = ax^b + c$.

Решение

Определяем среднее геометрическое крайних значений независимой переменной:

$$x_{\text{ср.геом}} = \sqrt{x_1 x_6} = \sqrt{250 \cdot 2000} = 707,1.$$

По табличным данным построим график и по графику приблизительно найдем численное значение $y_{\text{ср.геом}}$ для $x_{\text{ср.геом}}$ $y_{\text{ср.геом}} = 0,507$.

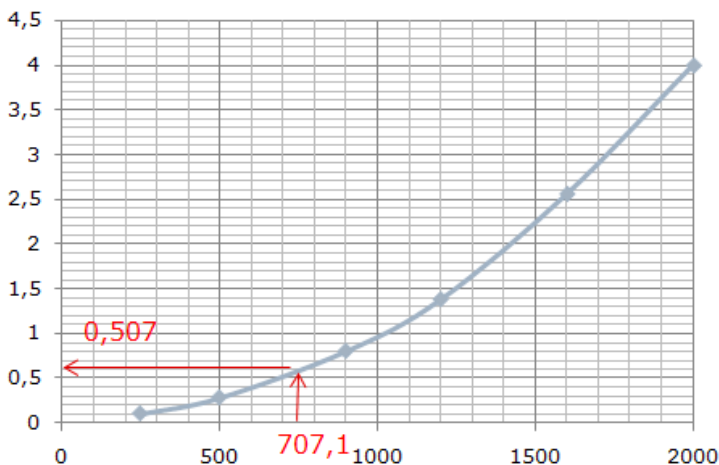


Рис. 6.20. График функции

Теперь можно рассчитать коэффициент c :

$$c = \frac{y_1 y_n - y_{\text{ср.геом}}^2}{y_1 + y_n - 2 y_{\text{ср.геом}}} = \frac{0,10 \cdot 4,10 - (0,507)^2}{0,10 + 4,10 - 2 \cdot 0,507} = 0,048.$$

Коэффициенты a и b найдем методом наименьших квадратов. Построим новую таблицу (табл. 6.10), используя исходную таблицу данных, в которой

$$X_i = \ln x_{i,\text{таб}}; \quad Y_i = \ln(y_{i,\text{таб}} - c),$$

где $i = 1, 2, \dots, 6$.

Таблица 6.10

X_i	5,521	6,245	6,802	7,090	7,378	7,601
Y_i	-2,957	-1,461	-0,287	0,287	0,921	1,370

По данным новой таблицы построим график, рис. 6.21.

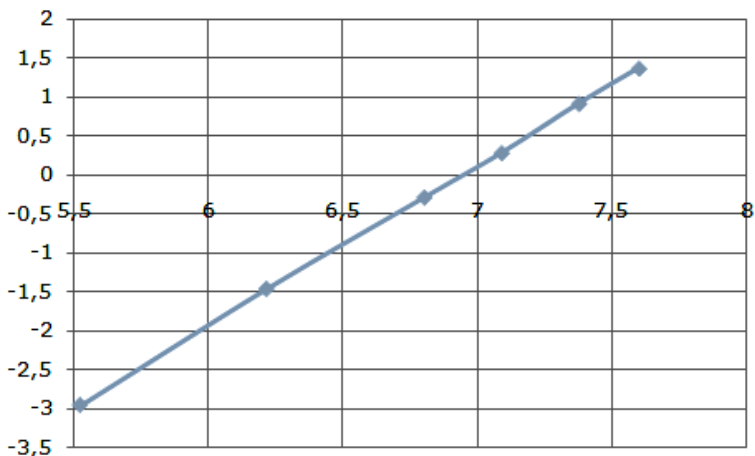


Рис. 6.21. График функции в новых координатах

Из графика видно, что данные новой таблицы хорошо ложатся на прямую. Это говорит о правильности выбранного вида эмпирической формулы (степенная зависимость) для заданных данных.

Определим коэффициенты a и b для полученной прямой, используя систему нормальных уравнений. Для этого выполним вспомогательные вычисления: $\sum x_i^2 = 277,855$; $\sum x_i = 40,607$; $\sum x_i y_i = -8,069$; $\sum y_i = -2,121$. Таким образом, получили систему уравнений

$$\begin{cases} 277,855a^* + 40,607b^* = -8,069; \\ 40,607a^* + 6b^* = -2,121, \end{cases}$$

из которой находим коэффициенты $a^* = 2,072$ и $b^* = -14,376$. Следовательно, имеем уравнение прямой вида

$$Y = a^* X + b^* \quad \text{или} \quad \ln(y_i - 0,048) = \ln a + b \ln x_i.$$

Подставив в последнее равенство значения x_i и y_i из исходной таблицы, получим следующие уравнения:

$$\begin{aligned} -2,957 &= \ln a + 5,521b; & 0,287 &= \ln a + 7,090b; \\ -1,461 &= \ln a + 6,215b; & 0,921 &= \ln a + 7,378b; \\ -0,285 &= \ln a + 6,802b; & 1,399 &= \ln a + 7,601b. \end{aligned}$$

$$-4,703 = 3\ln a + 18,538b; \quad 2,607 = 3\ln a + 22,069b.$$

Решая полученную систему двух уравнений, находим $\ln a = -14,36$ или $a = 5,801 \cdot 10^{-7}$; $b = 2,07$.

Сравнение расчетных значений с исходными данными показано на рис. 6.22 и 6.23.

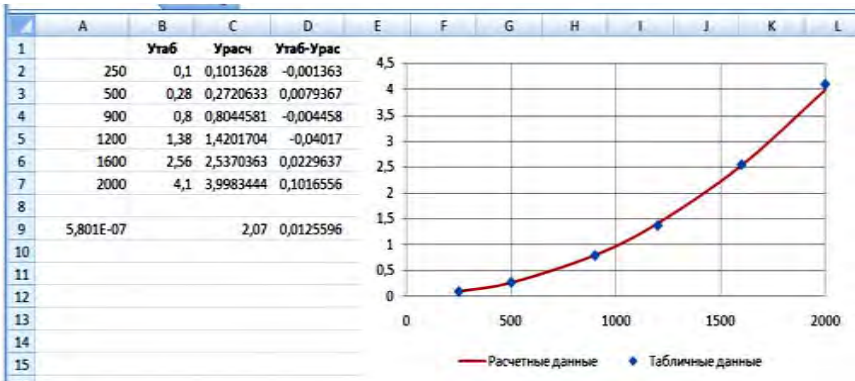


Рис. 6.22. Результаты аппроксимации

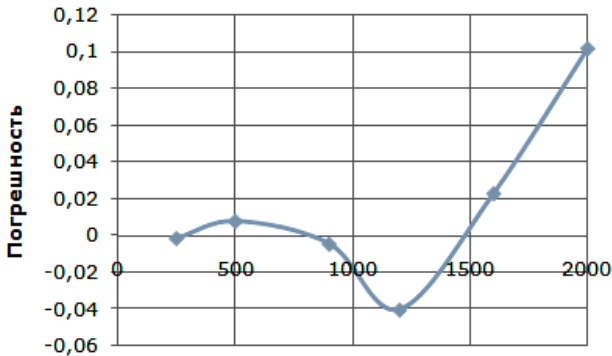


Рис. 6.23. Погрешность аппроксимации

Сумма квадратов погрешности $\sum \Delta^2 = 0,0125596$.

Ответ. Искомая эмпирическая формула имеет вид

$$y = 5,801 \cdot 10^{-7} x^{2,07} + 0,048.$$

Показательная функция

Пусть для табличных данных требуется подобрать коэффициенты показательной функции

$$y = ae^{bx} + c. \quad (6.4)$$

Перенеся свободный член c влево и логарифмируя, получим

$$\ln(y - c) = \ln a + bx.$$

Следовательно, имеем следующее уравнение:

$$Y = \ln(y - c) = \ln a + bx.$$

Как и в предыдущем случае, вначале определим коэффициент c . Для этого выберем крайние значения зависимой переменной x и рассчитаем ее среднее арифметическое $x_{\text{ср}} = (x_1 + x_n)/2$. Для $x_{\text{ср}}$ определим соответствующее значение $y_{\text{ср}}$ (по графику или применяя линейную интерполяцию).

Подставляя значения $x_{\text{ср}}$ и $y_{\text{ср}}$ в исходную эмпирическую формулу (6.4), будем иметь

$$y_1 = ae^{bx_1} + c;$$

$$y_n = ae^{bx_n} + c;$$

$$y_{\text{ср}} = ae^{bx_{\text{ср}}} + c.$$

Отсюда $y_1 - c = ae^{bx_1}$; $y_n - c = ae^{bx_n}$ и $y_{\text{ср}} - c = ae^{bx_{\text{ср}}}$. Умножим первое выражение на второе и возведем в квадрат третье выражение, получим

$$(y_1 - c)(y_n - c) = a^2 e^{b(x_1 + x_n)};$$

$$(y_{\text{ср}} - c)^2 = a^2 e^{\frac{b(x_1 + x_n)}{2}} = a^2 e^{b(x_1 + x_n)}.$$

Замечаем, что правые части обоих выражений равны, следовательно, имеем

$$(y_1 - c)(y_n - c) = (y_n - c)^2.$$

Решив последнее уравнение относительно c , получим

$$c = \frac{y_1 y_n - y_{\text{ср}}^2}{y_1 + y_n - 2y_{\text{ср}}}. \quad (6.5)$$

Если обнаружится линейная зависимость уравнения (6.5), то коэффициенты a и b находятся обычными приемами.

Предполагая, что значения x_i равноотстоящие, т. е. $\Delta x_i = h = \text{const}$, для расположения точек $M_i(x_i, \ln \Delta y_i)$, где $\Delta y_i = y_{i+1} - y_i$, $i = 1, 2, \dots, n-1$, на прямой необходимо, чтобы $\Delta(\ln \Delta y_i) = \text{const}$, где $i = 1, 2, \dots, n-2$.

Пример 6.5

Для переменных x и y дана следующая таблица данных, табл. 6.11.

Таблица 6.11

X	0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1,0
Y	1,30	1,44	1,89	1,78	1,97	2,19	2,46	2,74	3,06	3,42	3,84

Найти эмпирическую формулу для зависимости между x и y .

Решение

Составим таблицу разностей Δy_i , табл. 6.12.

Таблица 6.12

x	y	Δy	$\ln(\Delta y)$	$\Delta(\ln \Delta y)$
0,0	1,30	-0,854	-1,96611	-0,06899
0,1	1,44	-0,824	-1,89712	-0,23639
0,2	1,59	-0,721	-1,66073	0
0,3	1,78	-0,721	-1,66073	-0,1466
0,4	1,97	-0,658	-1,51413	-0,20479
0,5	2,19	-0,569	-1,30933	-0,03637
0,6	2,46	0,28	-1,27297	-0,13353
0,7	2,74	0,32	-1,13943	-0,11778
0,8	3,06	0,36	-1,02165	-0,15415
0,9	3,42	0,42	-0,8675	-0,06899
1,0	3,84			

Так как разность Δy монотонно возрастает (из отрицательных значений переходит в положительные), то зависимость между исходными данными x и y не является линейной.

Проверяем критерий, отмеченный в замечании для показательной функции (6.4). Из табл. 6.12 видно, что для $\Delta(\ln \Delta y)$ имеется значительный разброс, поэтому формулу (6.4) можно принять в качестве грубого приближения.

Найдем средние значения $x_{\text{ср}}$ для крайних значений x_1 и x_n , а также $y_{\text{ср}}$:

$$x_{\text{ср}} = (x_1 + x_{10})/2 = (0 + 1)/2 = 0,5.$$

Из таблицы исходных данных для $x = 0,5$ имеем $y_{\text{ср}} = 2,19$. Теперь по формуле (6.5) можем рассчитать коэффициент c :

$$c = \frac{3,84 \cdot 1,30 - 2,19^2}{1,30 + 3,84 - 2 \cdot 2,19} = 0,258.$$

Коэффициенты a и b определим, как и ранее, воспользовавшись уравнением

$$\ln(y_i - 0,258) = \ln a + bx_i, \quad i = 0, 1, \dots, 10.$$

Составим два уравнения:

$$\begin{cases} \sum_{i=0}^5 \ln(y_i - 0,258) = 6 \ln a + b \sum_{i=0}^5 x_i; \\ \sum_{i=6}^{10} \ln(y_i - 0,258) = 5 \ln a + b \sum_{i=6}^{10} x_i. \end{cases} \quad (6.6)$$

Подставив в систему уравнений (6.6) значения x_i и y_i , получим новую систему уравнений

$$\begin{cases} 6 \ln a + 1,5b = 2,111; \\ 5 \ln a + 4b = 5,037. \end{cases}$$

Решая эту систему, находим коэффициенты $a = 1,055$ и $b = 1,192$. Следовательно, искомая эмпирическая формула имеет вид

$$y = 1,055e^{1,192x} + 0,258.$$

Согласованность полученной формулы с исходными данными показана на графиках, рис. 6.24 и 6.25.

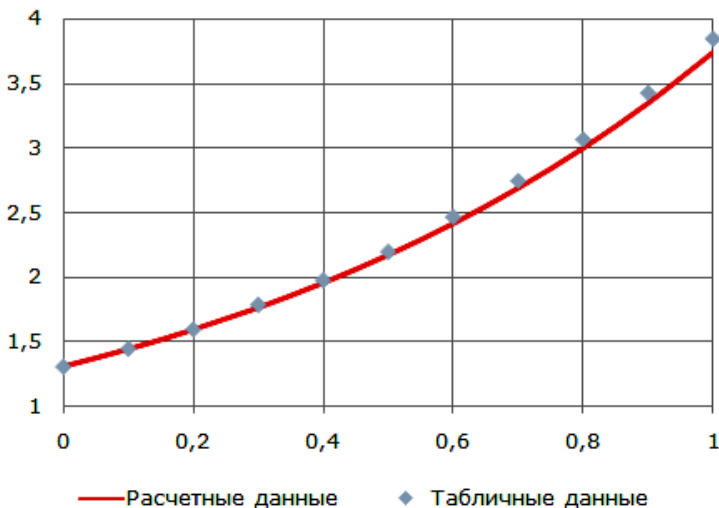


Рис. 6.24. Результаты аппроксимации

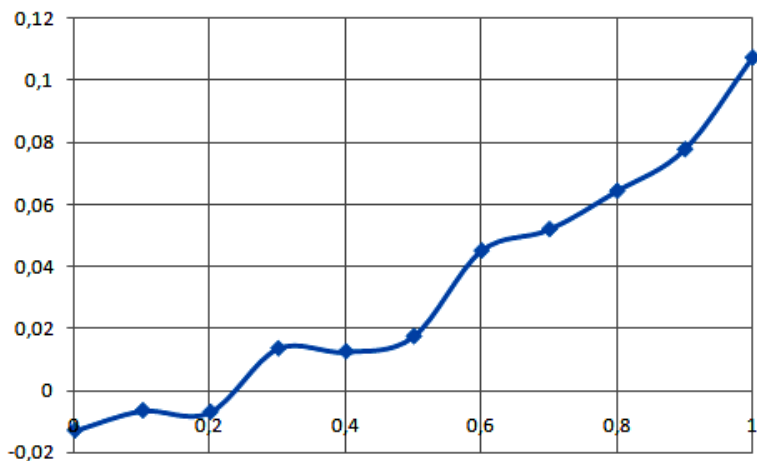


Рис. 6.25. Погрешность аппроксимации

Сумма квадратов погрешностей $\sum \Delta^2 = 0,027$.

Ответ. Искомая эмпирическая формула имеет вид $y = 1,055e^{1,192x} + 0,258$.

Варианты заданий

Подобрать эмпирическую формулу для табличных данных, табл. 6.13.

Таблица 6.13

	x	2	4	6		8	
				0,725	0,947	2,5	3,0
Вариант 1	x	0,350	0,573				
	y	0,5	1,0	2,0			
Вариант 2	x	0,62	1,64	3,70			
	y	1,0	2,0	4,0	5,02		6,04
Вариант 3	x	1,06	1,33	1,52	1,68	1,81	1,91
	y	0,0	0,2	0,5	1,0	2,0	3,5
Вариант 4	x	1,0	0,833	0,667	0,540	0,400	0,333
	y	7	12	17	22	27	32
Вариант 5	x	80,7	72,9	63,2	54,7	47,5	41,4
	y	14,5	30,0	64,5	74,5	86,7	94,5
Вариант 6	x	0,0	0,004	0,018	0,029	0,051	0,073
	y	273	283	288	293	313	333
Вариант 7	x	29,4	33,3	35,2	37,2	45,8	55,2
	y	3,334	1,630	0,8657	0,4323	0,2646	0,1699
Вариант 8	x	0,482	1,034	2,027	4,247	7,164	11,48
	y	250	900	900	1200	1600	2000
Вариант 9	x	0,10		0,28	0,80	1,38	2,56
	y	0,0	0,1	0,2	0,3	0,4	0,5
Вариант 10	x	1,30	1,44	1,59	1,78	1,97	2,19
	y	8	10	10	15	20	30
Вариант 11	x	1,30		14,0	15,4	16,3	17,2
	y	0,0	1,0	2,0	3,0	3,5	4,0
Вариант 12	x	2,01		0,74	0,45	0,347	0,270
	y	0,0	0,2	0,6	1,2	1,6	2,0
Вариант 13	x	2,05		1,638	0,907	0,423	0,028
	y						

Вариант 14	x	0,15	0,30	0,40	0,45	0,50	0,55	0,60
	y	2,17	3,63	5,07	5,929	6,883	7,78	9,069
Вариант 15	x	0,0	5,0	10,0	15,0	20,0	25,0	
	y	0,0	106,0	182,0	234,0	261,0	275,0	
Вариант 16	x	0,75	1,50	2,25	3,0	3,75	4,0	
	y	2,50	1,20	1,12	2,25	4,28	5,66	
Вариант 17	x	0,0	5,0	10,0	15,0	20,0	25,0	
	y	5,0	150,0	560,0	1200,0	2100,0	3300,0	
Вариант 18	x	1956	1958	1960	1962	1964	1966	1968
	y	50000	68500	92500	11000	132500	152000	175000
Вариант 19	x	1,0	2,0	3,0	4,0	5,0	6,0	7,0
	y	10,1	15,1	19,7	25,3	31,5	37,6	42,7
Вариант 20	x	1,0	2,0	3,0	4,0	5,0	6,0	
	y	104,0	113,0	119,0	122,0	130,0	137,0	

Предметный указатель

Аппроксимирующая функция.....	144
Ведущий элемент	13
Вторая правая разность.....	155
Вырожденная матрица	9
Вычислительная математика.....	7
График функции	143
Графический метод	28
Дифференциальное уравнение	73
Дифференциальное уравнение в частных производных	74
Дифференциальное уравнение первого порядка	73
Дифференциальное уравнение порядка n	87
Задача Коши.....	74
Задача численного дифференцирования	135
Задача численного интегрирования.....	119
Интеграл дифференциального уравнения.....	74
Интегральная кривая.....	76
Интерполяционный многочлен Лагранжа	146
Квадратная матрица	8
Квадратурные формулы.....	119
Корни уравнения	28
Линейное дифференциальное уравнение.....	77
Линейное дифференциальное уравнение с постоянными коэффициентами.....	88
Линейное неоднородное дифференциальное уравнение.....	77
Линейное неоднородное уравнение.....	88
Линейное однородное дифференциальное уравнение.....	77
Линейное однородное уравнение.....	88
Метод Лагранжа	30
Метод Лина	49
Метод Ньютона–Грегори.....	154
Метод понижения порядка	47
Начальное приближение	10
Непрерывная или интегральная аппроксимация	144
Нормальная система дифференциальных уравнений	103
Обратный ход метода Гаусса	11
Общая формула Симпсона	125

Общее решение дифференциального уравнения.....	74
Обыкновенное дифференциальное уравнение	73
Определитель.....	9
Особое решение дифференциального уравнения.....	76
Отделение корней.....	28
Первая разность	155
Подгонка кривой	160
Приближенное решение.....	6
Признак монотонности функции	29
Прямой ход метода Гаусса	11
Прямоугольная матрица	9
Прямые методы.....	9
Рекомендации по разработке программы вычисления определённого интеграла методом Гаусса.....	130
Рекомендации по программированию вычислений определенного интеграла методом Симпсона	126
Рекомендации по программированию вычисления определенного интеграла методом трапеций	122
Свойства общего решения	74
Сущность метода Гаусса.....	128
Схема Горнера	31
Схема с выбором главного элемента.....	13
Теорема Коши.....	74
Точечная аппроксимация.....	144
Уравнение, разрешенное относительно производной	77
Уточнение корня.....	28
Формула Лагранжа	31
Формула пересчета.....	81
Формулы Крамера	10
Функциональная зависимость	142
Численные методы	6
Эмпирические формулы	163
Якобиан	62

Литература

1. Амосов, А. А. Вычислительные методы для инженеров : учебное пособие / А. А. Амосов, Ю. А. Дубинский, Н. В. Копченова. – М. : Высшая школа, 1994. – 544 с.
2. Архангельский, А. Я. Приемы программирования в Delphi на основе VCL / А. Я. Архангельский. – М. : БИНОМ-Пресс, 2006. – 944 с.
3. Бахвалов, Н. С. Численные методы в задачах и упражнениях : учебное пособие / Н. С. Бахвалов, А. В. Лапин, Е. В. Чижонков ; под ред. В. А. Содовниченко. – М. : Высшая школа, 2000. – 190 с.
4. Вержбицкий, В. М. Численные методы (математический анализ и обыкновенные дифференциальные уравнения) : учебное пособие для вузов / В. М. Вержбицкий. – М. : Высшая школа, 2001. – 382 с.
5. Гайдышев, И. П. Решение научных и инженерных задач средствами Excel, VBA и C/C++ / И. П. Гайдышев. – СПб. : БХВ-Петербург, 2004. – 512 с.
6. Исаков, В. Б. Элементы численных методов : учебное пособие для вузов / В. Б. Исаков. – М. : Академия, 2003. – 192 с.
7. Киреев, В. И. Численные методы в примерах и задачах : учебное пособие / В. И. Киреев, А. В. Пантелеев. – 3-е изд., стер. – М. : Высшая школа, 2008. – 400 с.
8. Кирьянов, Д. В. Mathcad 13 / Д. В. Кирьянов. – СПб. : БХВ-Петербург, 2006. – 608 с.
9. Копченова, Н. В. Вычислительная математика в примерах и задачах : учебное пособие / Н. В. Копченова, И. А. Марон. – 3-е изд., стер. – СПб. : Лань, 2009. – 368 с.
10. Крылов, А. Н. Лекции о приближенных вычислениях / А. Н. Крылов. – М. : Гос. изд-во технико-теоретической лит-ры, 1954.
11. Макаров, Е. Г. Инженерные расчеты в Mathcad 15 : учебный курс / Е. Г. Макаров. – СПб. : Питер, 2011 – 400 с.
12. Нинул, А. С. Оптимизация целевых функций: Аналитика. Численные методы. Планирование эксперимента / А. С. Нинул. – М. : Изд-во физико-математической литературы, 2009. – 336 с.
13. Охфузин, В. А. Прикладная математика в системе MATHCAD : учебное пособие / В. А. Охфузин. – 2-е изд., испр. и доп. – СПб. : Лань, 2008. – 352 с.

14. Плотников, А. Д. Численные методы : учебное пособие / А. Д. Плотников. – Минск : Новое знание, 2007. – 174 с.

15. Поршнев, С. В. Численные методы на базе Mathcad / С. В. Поршнев, И. В. Беленкова. – СПб. : БХВ-Петербург, 2005. – 464 с.

16. Ракитин, В. И. Практическое руководство по методам вычисления с приложением программ для персональных компьютеров : учебное пособие / В. И. Ракитин, В. Е. Первушин. – М. : Высшая школа, 1998. – 383 с.

17. Рашиков, В. И. Численные методы. Компьютерный практикум : учебно-методическое пособие / В. И. Рашиков. – М. : МИФИ, 2010. – 133 с.

18. Тарасевич, Ю. Ю. Численные методы на Mathcad'е / Ю. Ю. Тарасевич. – Астрахань : Астраханский гос. пед. ун-т, 2000. – 70 с.

19. Турчак, Л. И. Основы численных методов : учебное пособие / Л. И. Турчак, П. В. Плотников. – 2-е изд., перераб. и доп. – М. : ФИЗМАТЛИТ, 2003. – 304 с.

20. Формалев, В. Ф. Численные методы / В. Ф. Формалев, Д. Л. Резвизников. – М. : ФИЗМАТЛИТ, 2004. – 400 с.

Учебное издание

АТАМАНОВ Юрий Евгеньевич
ПЛИЩ Владимир Николаевич
РАВИНО Виктор Валерьевич

**МЕТОДЫ РЕШЕНИЯ ИНЖЕНЕРНЫХ ЗАДАЧ
ПРИМЕНИТЕЛЬНО К ОБЛАСТИ ТРАНСПОРТНОГО
И ТЯГОВОГО МАШИНОСТРОЕНИЯ**

Учебно-методическое пособие
к лабораторным работам по дисциплине «Информатика»
для студентов специальностей 1-37 01 03 «Тракторостроение»,
1-37 01 04 «Многоцелевые гусеничные и колесные машины»
и 1-37 01 05 «Городской электрический транспорт»

Редактор *Т. Н. Микулик*
Компьютерная верстка *Н. А. Школьниковой*

Подписано в печать 30.10.2015. Формат 60×84 ¹/₁₆. Бумага офсетная. Ризография.
Усл. печ. л. 11,74. Уч.-изд. л. 9,18. Тираж 100. Заказ 1069.

Издатель и полиграфическое исполнение: Белорусский национальный технический университет.
Свидетельство о государственной регистрации издателя, изготовителя, распространителя
печатных изданий № 1/173 от 12.02.2014. Пр. Независимости, 65. 220013, г. Минск.