МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«БАРАНОВИЧСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ИНФОРМАТИКА

**Методические указания
для студентов II курса специальностей
1-53 01 01 Автоматизация
технологических процессов и производств,
1-36 01 03 Технологическое оборудование
машиностроительного производства,
1-36 01 01 Технология машиностроения
инженерного факультета**

Барановичи
РИО БарГУ
2012

УДК 004(076)
ББК 32.81я73
И74

Рекомендовано к печати методической комиссией
инженерного факультета

Р а з р а б о т а л и:

О. И. Наранович, Г. М. Раковцы, С. Г. Скобля

Р е ц е н з е н т ы:

Е. Н. Кирюхова, старший преподаватель кафедры
физико-математических дисциплин БарГУ;
О. А. Синдель, инженер-программист ЗАО «АТЛАНТ» БСЗ

И74 **Информатика** [Текст] : метод. указания для студентов II курса специальностей 1-53 01 01 Автоматизация технологических процессов и производств, 1-36 01 03 Технологическое оборудование машиностроительного производства, 1-36 01 01 Технология машиностроения инженерного факультета / разраб.: О. И. Наранович, Г. М. Раковцы, С. Г. Скобля. — Барановичи : РИО БарГУ, 2012. — 87, [5] с. — 165 экз.

Издание содержит материалы для изучения раздела «Численные методы решения инженерных задач» дисциплины «Информатика»; включает пять лабораторных работ с теоретическим материалом и примерами реализации методов в математическом пакете MathCad, а также Excel и Delphi.

Предназначено для студентов II курса инженерного факультета БарГУ.

Табл. 13. Рис. 34. Прил. 2.

УДК 004(076)
ББК 32.81я73

СОДЕРЖАНИЕ

Введение 4

Лабораторная работа 1. Решение систем линейных уравнений 5
Лабораторная работа 2. Решение нелинейных уравнений 19
Лабораторная работа 3. Аппроксимация и интерполирование функций . . . 39
Лабораторная работа 4. Численное интегрирование 61
Лабораторная работа 5. Численное решение дифференциальных уравнений . . . 73

Приложение А. Команды меню и панели инструментов среды MathCad 85
Приложение Б. Математические формулы 89

Список использованных источников 90
Список рекомендованных источников 90

ПРЕДИСЛОВИЕ

Целью данного издания является ознакомление студентов с численными методами решения задач. Лабораторные работы, которые входят в данное пособие, нацелены на выработку навыков, необходимых при решении проектных и научных задач с использованием компьютеров.

Первая лабораторная работа, ознакамливающая студентов с решением систем линейных уравнений методом Гаусса, методом простых итераций и методом Зейделя, отличается невысоким уровнем сложности. Вторая лабораторная работа даёт сведения студентам о методах решения нелинейных уравнений (метод деления отрезка пополам, метод касательных и метод хорд). В третьей работе представлены задачи аппроксимации и интерполирования функций, рассмотрены метод наименьших квадратов и методы Ньютона и Лагранжа. Содержание четвёртой работы — численное интегрирование и дифференцирование. При её выполнении студенты должны освоить методы трапеций и Симпсона. Пятая работа знакомит студентов с методами Эйлера и Рунге — Кутта четвёртого порядка точности решения обыкновенных дифференциальных уравнений.

В издание включены краткие теоретические сведения по каждой рассматриваемой теме. Теоретический материал излагается сжато и рассчитан на студентов, впервые изучающих данную дисциплину. Приведены примеры решения задач с использованием математического пакета MathCAD, среды программирования Delphi и приложения MS Excel.

Для выполнения лабораторных работ студенты должны владеть основами программирования в среде Delphi, а также иметь навыки работы в математическом пакете MathCAD и приложении MS Excel.

Лабораторные работы выполняются каждым студентом индивидуально. Для этого студент получает одно из заданий по указанному преподавателем варианту.

Лабораторные работы считаются выполненными после их защиты. Для этого необходимо представить программы, результаты расчёта в различных средах и отчёт, оформленный в MS Word, содержащий тему и цель работы, формулировки заданий, формы приложений в Delphi, MS Excel и MathCad, тексты модулей, результаты тестирования приложений, выводы по работе.

Л а б о р а т о р н а я   р а б о т а 1
РЕШЕНИЕ СИСТЕМ ЛИНЕЙНЫХ УРАВНЕНИЙ

Цель работы: изучить особенности различных методов решения СЛАУ, приобрести навыки решения СЛАУ с помощью средств MS Excel и MathCad.

Постановка задачи. Система линейных алгебраических уравнений (далее — СЛАУ) имеет вид:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \mathbf{K} + a_{1n}x_n = b_1, \\ a_{21}x_1 + a_{22}x_2 + \mathbf{K} + a_{2n}x_n = b_2, \\ \mathbf{L\,L\,L\,L\,L\,L\,L\,L\,L\,L\,L\,L} \\ a_{s1}x_1 + a_{s2}x_2 + \mathbf{K} + a_{sn}x_n = b_s. \end{cases} \tag{1.1}$$

Рассмотрим только совместные определённые системы, т. е. системы, имеющие единственное решение. Это ограничение приводит к тому, что число уравнений системы (1.1) должно равняться числу неизвестных [6].

Используемые в наше время методы решения СЛАУ разбивают на две группы:

1. *Точные методы* ведутся без округлений и приводят к точным значениям неизвестных. Но так как на практике используемые данные имеют некоторую ограниченную точность, то используемые точные методы решения неизбежно будут давать приближенные значения. К точным методам относится метод Гаусса решения СЛАУ.

2. *Приближённые методы* при вычислении без округлений позволяют получить решение системы лишь с какой-то заданной точностью. Точное решение системы в этом случае может быть получено теоретически, как результат бесконечного процесса. К приближённым методам решения СЛАУ относят методы простой итерации, метод Зейделя и другие [3].

Наиболее эффективно программируемыми на ЭВМ являются метод Гаусса с выбором главного элемента в матрице и метод Зейделя. Если при экспериментальных исследованиях получаются приближённые значения коэффициентов СЛАУ, то сначала решают СЛАУ методом Гаусса с выбором главного элемента, а затем уточняют решение методом Зейделя.

Метод Гаусса. Одним из наиболее распространённых методов решения СЛАУ является метод Гаусса.

Суть метода заключается в следующем. Пусть в первом уравнении коэффициент $a_{11} \neq 0$ (если это не так, то на его место можно поставить уравнение, для которого это условие выполняется). Разделим первое уравнение на a_{11}. Затем умножим первое уравнение на a_{21} и вычтем из второго. Далее умножим первое уравнение на a_{31} и вычтем из третьего и т. д. В результате получим систему вида

$$\begin{cases} x_1 + a_{12}x_2 + \mathbf{K} + a_{1n}x_n = b_1, \\ \qquad a_{22}x_2 + \mathbf{K} + a_{2n}x_n = b_2, \\ \mathbf{KKKKKKKKKKK}\ldots. \\ \qquad a_{n2}x_2 + \mathbf{K} + a_{nn}x_n = b_n. \end{cases} \tag{1.2}$$

Для простоты будем считать, что коэффициенты и свободные члены новой системы мы переобозначили вновь как a и b.

При этом полученная система (1.2) полностью эквивалентна исходной. Продолжим преобразования. Первое уравнение вообще не будем трогать, а разделим второе уравнение на a_{22} и будем последовательно вычитать второе уравнение из оставшихся $n-2$-х уравнений, предварительно умножая его на a_{32}, a_{42}, …, a_{n2}. В результате получим следующую систему:

$$\begin{cases} x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1n}x_n = b_1, \\ \qquad x_2 + a_{23}x_3 + \ldots + a_{2n}x_n = b_2, \\ \qquad\qquad a_{33}x_3 + \ldots + a_{3n}x_n = b_3, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \qquad\qquad a_{n3}x_3 + \ldots + a_{nn}x_n = b_n. \end{cases} \tag{1.3}$$

Продолжая подобные преобразования, получим систему, приведённую к треугольному виду:

$$\begin{cases} x_1 + a_{12}x_2 + a_{13}x_3 + \mathbf{K} + a_{1n}x_n = b_1, \\ \qquad x_2 + a_{23}x_3 + \mathbf{K} + a_{2n}x_n = b_2, \\ \qquad\qquad x_3 + \mathbf{K} + a_{3n}x_n = b_3, \\ \mathbf{KKKKKKKKKKKKKK} \\ \qquad\qquad\qquad a_{nn}x_n = b_n. \end{cases} \tag{1.4}$$

Такие преобразования систем (1.2—1.4) называют прямым ходом метода Гаусса [3].

Далее из последнего уравнения полученной системы (1.4) выражаем x_n, подставляем в предпоследнее уравнение, затем из него выражаем x_{n-1}, подставляем x_n и x_{n-1} в предыдущее уравнение и выражаем x_{n-2} и т. д. (обратный ход метода Гаусса). В итоге получим набор x_1, x_2, …, x_n, являющийся решением исходной системы.

Метод Гаусса с выбором главного элемента. При использовании этого метода на первом шаге среди элементов a_{i1} определяют максимальный по модулю элемент. Первое уравнение системы и уравнение, в котором находят максимальный по модулю элемент, меняют местами. Далее стандартным образом (как в методе Гаусса) производят исключения неизвестных x_1. Аналогичные действия повторяют и с оставшимися уравнениями, т. е. из оставшихся $n - 1$-х уравнений на вторую строку переставляют уравнение с максимальным коэффициентом при x_2. Делят второе уравнение на a_{22} и последовательно вычитают его из оставшихся $n - 2$-х уравнений, предварительно умножая на a_{32}, a_{42} и т. д.

Смысл выбора главного элемента состоит в том, чтобы сделать меньшими коэффициенты при неизвестных в процессе исключения неизвестных и тем самым уменьшить погрешность вычислений [4].

Метод простой итерации решения СЛАУ. Для использования этого метода систему (1.1) необходимо привести к следующему виду (называемому нормальным):

$$\begin{cases} x_1 = \dfrac{b_1}{a_{11}} - \dfrac{a_{12}}{a_{11}} x_2 - \dfrac{a_{13}}{a_{11}} x_3 - \ldots - \dfrac{a_{1n}}{a_{11}} x_n, \\ x_2 = \dfrac{b_2}{a_{22}} - \dfrac{a_{21}}{a_{22}} x_1 - \dfrac{a_{23}}{a_{22}} x_3 - \ldots - \dfrac{a_{2n}}{a_{22}} x_n, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ x_n = \dfrac{b_n}{a_{nn}} - \dfrac{a_{n1}}{a_{nn}} x_1 - \dfrac{a_{n1}}{a_{nn}} x_2 - \ldots - \dfrac{a_{nn-1}}{a_{nn}} x_{n-1}. \end{cases} \tag{1.5}$$

Затем в полученную систему (1.5) подставляют начальное приближение решения, т. е. некоторый набор значений $x_1^0, x_2^0, x_3^0, \ldots, x_n^0$, и вычисляют следующее приближение решения — набор значений $x_1^1, x_2^1, x_3^1, \ldots, x_n^1$. Полученный набор значений снова подставляют в систему (1.5) для вычисления следующего приближения и т. д. Процесс повторяется до тех пор, пока не будет достигнута необходимая точность решения [2].

Сходимость данного метода определяется выполнением одного из следующих условий [6]:

$$\alpha = \max_{1 \le i \le n} \sum_{j=1}^{n} |a_{ij}| < 1, \qquad \alpha = \max_{1 \le j \le n} \sum_{i=1}^{n} |a_{ij}| < 1, \qquad \alpha = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij}^{2}} < 1. \tag{1.6}$$

Метод простой итерации решения СЛАУ сходится, если максимальная из сумм модулей коэффициентов при неизвестных системы (1.5), взятых по строкам меньше единицы, либо если максимальная из сумм модулей коэффициентов при неизвестных системы (1.5), взятых по столбцам меньше единицы, либо если сумма квадратов всех коэффициентов при неизвестных в правой части системы (1.5) меньше единицы. Каждое из этих условий является достаточным для сходимости метода, но не необходимым. Это означает, что метод сходится при выполнении хотя бы одного из этих условий, но он также может сходиться и тогда, когда ни одно из них не выполняется.

Момент прекращения вычислений можно определить, руководствуясь эмпирическим правилом: если в ходе итераций десятичная цифра повторилась три и более раз — её можно считать верной.

Во всех остальных случаях, когда выполняется одно из условий формул (1.6), степень приближения к точному решению оценивается по формуле

$$|x_k - x_{k-1}| < \frac{\varepsilon(1-\alpha)}{\alpha}, \tag{1.7}$$

где α — величина, вычисляемая по одной из формул (1.6), по которой обнаружена сходимость метода.

При выполнении одного из условий формул (1.6) в качестве начального значения решения можно взять любой набор значений.

Решение СЛАУ методом Зейделя. *Метод Зейделя* является модифицированным методом простой итерации.

В данном методе при вычислении k-го приближения неизвестного x_i при $i > 1$ используются уже вычисленные ранее k-е приближения неизвестных $x_1, x_2, \ldots, x_{i-1}$.

Вычислительные формулы при этом примут вид:

$$
\begin{cases}
x_1^{(k)} = b_1 - a_{12}x_2^{(k-1)} - \ldots - a_{1n}x_n^{(k-1)}, \\
x_2^{(k)} = b_2 - a_{21}x_1^{(k)} - \ldots - a_{2n}x_n^{(k-1)}, \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
x_n^{(k)} = b_n - a_{n1}x_1^{(k)} - a_{n2}x_2^{(k)} - \ldots - a_{nn-1}x_{n-1}^{(k)}.
\end{cases} \tag{1.8}
$$

Условия сходимости и условия прекращения итераций для метода простой итерации остаются верными и для метода Зейделя.

Обычно метод Зейделя дает более быструю сходимость. Он удобен при программировании, так как при вычислении x_i^k нет необходимости хранить промежуточные значения x_1^{k-1}, x_2^{k-1}, ..., x_n^{k-1}.

Задания

Задание 1. Используя методы Гаусса, простых итераций и Зейделя, разработайте схемы соответствующих алгоритмов решения СЛАУ в среде MS Excel.

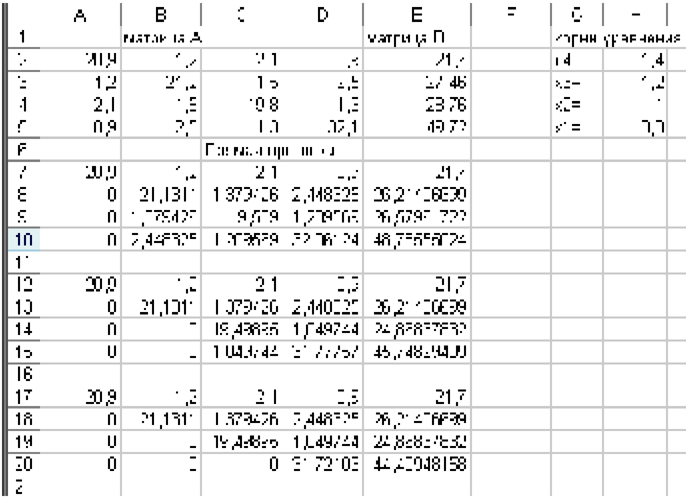
Пример выполнения задания

Найдите решение системы

$$
\begin{cases}
20{,}9x_1 + 1{,}2x_2 + 2{,}1x_3 + 0{,}9x_4 = 21{,}7; \\
1{,}2x_1 + 21{,}2x_2 + 1{,}5x_3 + 2{,}5_4 = 27{,}46; \\
2{,}1x_1 + 1{,}5x_2 + 19{,}8x_3 + 1{,}3x_4 = 28{,}76; \\
0{,}9x_1 + 2{,}5x_2 + 1{,}3x_3 + 32{,}1x_4 = 49{,}72.
\end{cases} \tag{1.9}
$$

1.1. ***Метод Гаусса в среде MS Excel***

1.1.1. В ячейки A2:E5 введите расширенную матрицу системы (рис. 1.1), скопируйте её в диапазоны ячеек A7:E10 [6].



	A	B	C	D	E	F	G	H
1		матрица А			матрица В		корни уравнения	
2	20,9	[illegible]	2,1	[illegible]	21,7		x1	[illegible]
3	1,2	[illegible]	1,5	[illegible]	[illegible]		x2=	[illegible]
4	2,1	1,5	19,8	1,3	23,76		x3=	[illegible]
5	0,9	[illegible]	[illegible]	[illegible]	[illegible]		x4=	[illegible]
6			[illegible]					
7	20,9	[illegible]	2,1	[illegible]	[illegible]			
8	0	[illegible]	[illegible]	[illegible]	[illegible]			
9	0	[illegible]	9,679	[illegible]	[illegible]			
10	0	[illegible]	[illegible]	[illegible]	[illegible]			
11								
12	20,9	[illegible]	2,1	[illegible]	21,7			
13	0	[illegible]	[illegible]	[illegible]	[illegible]			
14	0	[illegible]	[illegible]	[illegible]	[illegible]			
15	0	[illegible]	[illegible]	[illegible]	[illegible]			
16								
17	20,9	1,2	2,1	[illegible]	21,7			
18	0	[illegible]	[illegible]	[illegible]	[illegible]			
19	0	[illegible]	[illegible]	[illegible]	[illegible]			
20	0	[illegible]	0	[illegible]	[illegible]			

Рисунок 1.1 — Вид решения СЛАУ методом Гаусса в Excel

1.1.2. *Прямой ход метода Гаусса:*

а) предположите, что в ячейке A2 не ноль. Если это не так, то переставьте строки таким образом, чтобы число в ячейке A2 было отлично от нуля;

б) выделите диапазон A8:E8 и в строке формул введите формулу

$$=A3:E3–\$A\$2:\$E\$2*A3/\$A\$2,$$

нажмите клавиши **<Ctrl> + <Shift> + <Enter>**. В результате формула примет вид

$$\{=A3:E3–\$A\$2:\$E\$2*A3/\$A\$2\},$$

где фигурные скобки указывают на операции над матрицами;

в) протянув за маркер автозаполнения, скопируйте формулу в ячейки A9:E10. В результате этих операций коэффициенты при x_1 во всех уравнениях, кроме первого, обратятся в ноль;

г) выделите диапазон A7:E10 и скопируйте значения, хранящиеся в нём, в ячейки диапазонов A12:E15. Для копирования значений нужно воспользоваться специальной вставкой. Ей соответствует пункт меню **Правка: Специальная вставка.** После выбора этого пункта появляется диалоговое окно **Специальная вставка**, в котором нужно выбрать **Вставить: Значения** и нажать **ОК**;

д) аналогичным образом обратите в ноль коэффициенты при x_2. В диапазон ячеек B14:E14 введите формулу =B9:E9–$B$8:$E$8*B9/$B$8, нажмите клавиши **<Ctrl> + <Shift> + <Enter>**;

е) протяните маркер автозаполнения этого диапазона так, чтобы заполнить ячейки диапазона B15:E15. Это обратит в ноль коэффициенты при x_2 в последних двух уравнениях;

ж) далее содержимое (только значения!) диапазона A12:E15 скопируйте в ячейки диапазона A17:E20;

и) выделите диапазон C20:E20, введите в него формулу

{=C15:E15–$C$14:$E$14*C15/$C$14},

что обратит в ноль коэффициент при x_3 в последнем уравнении;

к) в результате этих преобразований матрица системы примет треугольный вид.

1.1.3. *Обратный ход метода Гаусса:*

а) в ячейки G2, G3, G4 и G5 введите *x*4, *x*3, *x*2 и *x*1 соответственно, а в ячейки H2:H5 — формулы из таблицы 1.1;

б) в результате этого в диапазоне H2:H5 будет получено решение системы (*см.* рис. 1.1).

1.2. ***Метод простых итераций в среде MS Excel:***

а) в ячейки A2:E5 введите расширенную матрицу системы (рис. 1.2) и сделайте пояснительные записи;

б) приведите систему к нормальному виду (1.8), т. е. все коэффициенты первого уравнения разделите на a_{11}, все коэффициенты второго уравнения — на a_{22} и т. д. Для этого в диапазон ячеек A8:A11 введите надписи: *x*1=, *x*2=, *x*3=, *x*4=. Выделите блок ячеек B8:E8, в строке формул введите формулу

=$B$2:$E$2/$A$2

и нажмите клавиши **<Ctrl> + <Shift> + <Enter>** (операция над матрицей);

Таблица 1.1 — Обратный ход метода Гаусса

Ячейка	Формула
H2	= E20 / D20
H3	= (E19 – D19 * H2) / C19
H4	= (E18 – D18 * H2 – C18 * H3) / B18
H5	= (E17 – D17*H2 – C17 * H3 – B17 * H4) / A17

	A	B	C	D	E	F	G	H	I	J
1		Матрица А			Матрица В		0 приближения			
2	20,9	1,2	2,1	0,9	21,7	x1=	1,038278			
3	1,2	21,2	1,5	2,5	27,46	x2=	1,295283			
4	2,1	1,5	19,8	1,3	28,76	x3=	1,452525			
5	0,9	2,5	1,3	32,1	49,72	x4=	1,54891			
6	Сходимость метода			сходится			модули приведенной матрицы			
7	Норма матрицы			0,247476			0,057416	0,100478	0,043062	
8	x1=	0,057416	0,100478	0,043062	1,038278		0,056604	0,070755	0,117925	
9	x2=	0,056604	0,070755	0,117925	1,295283		0,106061	0,075758	0,065657	
10	x3=	0,106061	0,075758	0,065657	1,452525		0,028037	0,077882	0,040498	
11	x4	0,028037	0,077882	0,040498	1,54891					
12	Метод итераций	1 прибл	2 прибл	3 прибл	4 прибл	5 прибл	6 прибл	7 прибл	8 прибл	9 прибл
13	x1=	0,75126	0,810296	0,79706	0,800449	0,799906	0,80002	0,799996	0,800001	0,8
14	x2=	0,951005	1,011527	0,997719	1,00049	0,999899	1,000021	0,999996	1,000001	1
15	x3=	1,142501	1,211495	1,197542	1,200508	1,199893	1,200022	1,199995	1,200001	1,2
16	x4=	1,360095	1,407501	1,398048	1,400107	1,399929	1,400015	1,399997	1,400001	1,4
17										
18	Метод Зейделя	1 прибл	2 прибл	3 прибл	4 прибл	5 прибл	6 прибл			
19	x1=	0,75126	0,801918	0,800068	0,8	0,8	0,8			
20	x2=	0,967331	0,999571	1,000027	1,000001	1	1			
21	x3=	1,197867	1,199567	1,199991	1,2	1,2	1,2			
22	x4	1,403997	1,399997	1,399996	1,4	1,4	1,4			
23										

Рисунок 1.2 — Вид решения СЛАУ методом простых итераций
и методом Зейделя в Excel

в) в ячейку B9 введите формулу =A3/B3. Далее выделите диапазон ячеек C9:E9 и введите формулу =C3:E3/B3, используя операцию над матрицей;

г) в блок ячеек B10:C10 внесите формулу =A4:B4/C4, а в блок D10:E10 — соответственно =D4:E4/C4, используя операцию над матрицами. Для блока B11:D11 введите формулу =A5:C5/D5, а в ячейку E5 — формулу = E5/D5;

д) из полученной системы определите норму матрицы и признак сходимости метода. Для этого найдите модули полученных коэффициентов и в ячейку G7 введите формулу =ABS(B8), которую скопируйте на блок G7:I10. В ячейке D6 проверьте один из признаков сходимости и введите формулу

=ЕСЛИ(МАКС (G7 + G8 + G9 + G10; H7 + H8 + H9 + H10;
I7 + I8 + I9 + I10) < 1; "сходится"; "не сходится");

е) в ячейке D7 определите норму матрицы по формуле =МАКС(G7 + H7 + I7; G8 + H8 + I8; G9 + H9 + I9; G10 + H10 + I10). Если полученный ответ меньше 1, то метод сходится при любых начальных приближениях. За начальное (нулевое) приближение возьмите полученные свободные члены и внесите их в ячейки G2:G5. Вычислите первые приближённые значения: $x1(1)$, $x2(1)$, $x3(1)$, $x4(1)$ по формулам (табл. 1.2);

ж) таким образом получите первые приближённые значения $x1$, $x2$, $x3$, $x4$. Далее в ячейки C13:C16 введите формулы, используя уже новые полученные приближения из B13:B16 (табл. 1.3);

и) используя полученные вторые приближения, скопируйте формулы из ячеек C13:C16 на блок ячеек D13:D16. В результате получите новые приближённые значения корней. Продолжайте операцию копирования, получая новые приближения. Момент прекращения вычислений определите эмпирическим правилом или в соответствии с формулой (1.7).

1.3. ***Метод Зейделя в среде MS Excel:***

а) в блоке ячеек B19:B22 введите формулы вычисления корней уравнения по методу Зейделя (табл. 1.4);

Таблица 1.2 — Вычисление первых приближённых значений

Ячейка	Формула
B13	= E8 – B8*G3 – C8 * G4 – D8 * G5
B14	= E9 – B9 * G2 – C9 * G4 – D9 * G5
B15	= E10 – B10 * G2 – C10 * G3 – D10 * G5
B16	= E11 – B11 * G2 – C11 * G3 – D11 * G4

Таблица 1.3 — Продолжение вычислений

Ячейка	Формула
C13	= $E$8 – $B$8 * B14 – $C$8 * B15 – $D$8 * B16
C14	= $E$9 – $B$9 * B13 – $C$9 * B15 – $D$9 * B16
C15	= $E$10 – $B$10 * B13 – $C$10 * B14 – $D$10 * B16
C16	= $E$11 – $B$11 * B13 – $C$11 * B14 – $D$11 * B15

Таблица 1.4 — Вычисление корней уравнения по методу Зейделя

Ячейка	Формула
B19	= $E$8 – $B$8 * G3 – $C$8 * G4 – $D$8 * G5
B20	= $E9 – $B$9 * B19 – $C$9 * G4 – $D$9 * G5
B21	= $E$10 – $B$10 * B19 –$C$10 * B20 – $D$10 * G5
B22	= $E$11 – $B$11 * B19 – $C$11 * B20 – $D$11 * B21

б) рассуждая аналогично, введите формулы в ячейки C19:C22, используя полученные данные из B19:B22 соответственно (табл. 1.5).

в) скопируйте формулы из ячеек C19:C22 на блоки: D19:D22, E19:E22 и т. д. Момент прекращения вычислений определяется, как и в методе простых итераций, эмпирическим правилом.

Задание 2. Изучите возможности математического пакета MathCad при решении СЛАУ и разработайте схемы программ решения СЛАУ с использованием средств: lsolve( ), Given … Find( ), метода обратной матрицы и метода Гаусса. (Используйте приложение А.)

Пример выполнения задания

Используя возможности *пакета MathCad,* найдите различные способы решения СЛАУ вида

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

2.1. ***Метод обратной матрицы:***

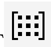
а) установите режим автоматических вычислений, пометив строку **Automatic Calculation** (**Автовычисление** в подменю **Вычисление**) в меню **Tools (Инструменты);**

б) присвойте переменной ORIGIN значение, равное 1 (значение этой переменной определяет номер первой строки (столбца) матрицы. По умолчанию в Mathcad нумерация начинается с 0):

ORIGIN:= 1

Таблица 1.5 — Продолжение вычислений

Ячейка	Формула
C19	=$E$8 – $B$8 * B20 – $C$8 * B21 – $D$8 * B22
C20	=$E$9 – $B$9 * C19 – $C$9 * B21 – $D$9 * B22
C21	=$E$10 – $B$10 * C19 – $C$10 * C20 – $D$10 * B22
C22	=$E$11 – $B$11 * C19 – $C$11 * C20 – $D$11 * C21

в) введите матрицу системы и матрицу-столбец свободных членов, используя инструмент  на панели **Matrix** (или меню **Вставить: Матрица**):

$$A := \begin{pmatrix} 1 & 2 & 3 \\ 1 & -3 & 2 \\ 1 & 1 & 1 \end{pmatrix} \qquad b := \begin{pmatrix} 7 \\ 5 \\ 3 \end{pmatrix}$$

г) вычислите решение системы по формуле $x = A^{-1} \cdot b$:

$$x := A^{-1} \cdot b$$

$$x = \begin{pmatrix} 1 \\ 0 \\ 2 \end{pmatrix}.$$

д) проверьте правильность решения умножением матрицы системы на вектор-столбец решения

$$A \cdot x - b = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

2.2. ***Построение матрицы с помощью функции* lsolve()** [9]: повторив пункты 2.1, а) — в), перейдите к вычислению корней системы:

$$x := \mathrm{lsolve}\ (A, b)$$

$$x = \begin{pmatrix} 1 \\ 0 \\ 2 \end{pmatrix}$$

2.3. ***Использование функции* Given…Find() *для решения СЛАУ***:

а) введите начальные приближения: x1 := 1, x2 := 1, x3 := 1;

б) введите слово Given;

в) ниже введите систему уравнений, используя при этом вместо обычного знака «равно» знак булева равенства (вводится нажатием клавиш **<Ctrl>+<=>**):

$$x1 + 2x2 + 3x3 = 7,$$

$$x1 - 3x2 + 2x3 = 5,$$

$$x1 + x2 + x3 = 3.$$

г) ниже напишите Find(x1,x2,x3)= и получите ответ

$$\mathrm{Find}(x1, x2, x3) = \begin{pmatrix} 1 \\ 0 \\ 2 \end{pmatrix}$$

2.4. ***Метод Гаусса*** [9]:

а) установите режим автоматических вычислений, пометив строку **Automatic Calculation** в меню **Tools**;

б) присвойте переменной ORIGIN значение, равное 1:

ORIGIN:= 1

в) введите матрицу системы и матрицу-столбец свободных членов, используя инструмент  на панели **Matrix**:

$$A := \begin{pmatrix} 1 & 2 & 3 \\ 1 & -3 & 2 \\ 1 & 1 & 1 \end{pmatrix} \qquad b := \begin{pmatrix} 7 \\ 5 \\ 3 \end{pmatrix}$$

г) сформируйте расширенную матрицу системы, используя функцию ***augment(A,b),*** которая формирует матрицу, добавляя к столбцам матрицы системы *A* справа столбец свободных членов ***b:***

$$Ar := \mathrm{augment}(A, b)$$

$$Ar = \begin{pmatrix} 1 & 2 & 3 & 7 \\ 1 & -3 & 2 & 5 \\ 1 & 1 & 1 & 3 \end{pmatrix}$$

д) приведите расширенную матрицу к ступенчатому виду, используя функцию *rref(Ar),* которая приводит расширенную матрицу к ступенчатому виду с единичной матрицей в первых столбцах, т. е. выполняет прямой и обратный ходы метода Гаусса:

$$\mathrm{Ag} := \mathrm{rref}(\mathrm{Ar})$$

$$\mathrm{Ag} = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 2 \end{pmatrix}$$

е) сформируйте столбец решения системы, используя функцию *submatrix*(*Ag*,1, 3, 4, 4), которая выделяет блок матрицы *Ag,* расположенный в строках с первой по третью и в столбцах с четвёртого по четвёртыйй (последний столбец):

$$\mathrm{z} := \mathrm{submatrix}(\mathrm{Ag}, 1, 3, 4, 4)$$

$$\mathrm{z} = \begin{pmatrix} 1 \\ 0 \\ 2 \end{pmatrix}$$

ж) проверьте правильность решения умножением матрицы системы на вектор-столбец решения:

$$\mathrm{A} \cdot \mathrm{z} - \mathrm{b} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

и) полученный результат подтверждает правильность вычислений.

Примечание. В среде MathCad важен порядок расположения операторов, которые выполняются сверху вниз, слева направо.

Задание 3. Получите решение СЛАУ, приведённой в таблице 1.6 (в соответствии со своим вариантом), с использованием алгоритмов, полученных при выполнении заданий 1 и 2.

Т а б л и ц а 1.6 — Варианты заданий

Вариант	Матрица A	Вектор B
1	$\begin{pmatrix} 1{,}02 & -0{,}25 & -0{,}3 \\ -0{,}41 & 1{,}13 & -0{,}15 \\ -0{,}25 & -0{,}14 & 1{,}21 \end{pmatrix}$	$\begin{pmatrix} 0{,}515 \\ 1{,}555 \\ 1{,}21 \end{pmatrix}$
2	$\begin{pmatrix} 10{,}9 & 1{,}2 & 2{,}1 & 0{,}9 \\ 1{,}2 & 11{,}2 & 1{,}5 & 2{,}5 \\ 2{,}1 & 1{,}5 & 9{,}8 & 1{,}3 \\ 0{,}9 & 2{,}5 & 1{,}3 & 12{,}1 \end{pmatrix}$	$\begin{pmatrix} -7 \\ 5{,}3 \\ 10{,}3 \\ 24{,}6 \end{pmatrix}$
3	$\begin{pmatrix} 20{,}9 & 1{,}2 & 2{,}1 & 0{,}9 \\ 1{,}2 & 21{,}2 & 1{,}5 & 2{,}5 \\ 2{,}1 & 1{,}5 & 19{,}8 & 1{,}3 \\ 0{,}9 & 2{,}5 & 1{,}3 & 32{,}1 \end{pmatrix}$	$\begin{pmatrix} 21{,}7 \\ 27{,}46 \\ 28{,}76 \\ 49{,}72 \end{pmatrix}$
4	$\begin{pmatrix} 6{,}1 & 2{,}2 & 1{,}2 \\ 2{,}2 & 5{,}5 & -1{,}5 \\ 1{,}2 & -1{,}5 & 7{,}2 \end{pmatrix}$	$\begin{pmatrix} 16{,}55 \\ 10{,}55 \\ 16{,}8 \end{pmatrix}$
5	$\begin{pmatrix} 3{,}82 & 1{,}02 & 0{,}75 & 0{,}81 \\ 1{,}05 & 4{,}53 & 0{,}98 & 1{,}53 \\ 0{,}73 & 0{,}85 & 4{,}71 & 0{,}81 \\ 0{,}88 & 0{,}81 & 1{,}28 & 3{,}5 \end{pmatrix}$	$\begin{pmatrix} 15{,}655 \\ 22{,}705 \\ 23{,}48 \\ 16{,}11 \end{pmatrix}$
6	$\begin{pmatrix} 30{,}75 & 2{,}11 & 0{,}7 \\ 1{,}21 & 2{,}05 & 0{,}64 \\ 1{,}2 & 1{,}53 & 3{,}21 \end{pmatrix}$	$\begin{pmatrix} -2 \\ 1 \\ 5 \end{pmatrix}$
7	$\begin{pmatrix} 3 & 0{,}4 & 1 \\ 1{,}2 & 4 & 0{,}32 \\ 1 & 0{,}35 & 3 \end{pmatrix}$	$\begin{pmatrix} -1 \\ 2 \\ -0{,}6 \end{pmatrix}$
8	$\begin{pmatrix} 1{,}02 & -0{,}25 & -0{,}3 \\ -0{,}41 & 1{,}13 & -0{,}15 \\ -0{,}25 & -0{,}14 & 1{,}21 \end{pmatrix}$	$\begin{pmatrix} 0{,}51 \\ 1{,}55 \\ 2{,}78 \end{pmatrix}$
9	$\begin{pmatrix} 6{,}1 & 2{,}2 & 1{,}2 \\ 2{,}2 & 5{,}5 & -1{,}5 \\ 1{,}2 & 1{,}5 & 7{,}2 \end{pmatrix}$	$\begin{pmatrix} 16{,}55 \\ 10{,}55 \\ 16{,}8 \end{pmatrix}$
10	$\begin{pmatrix} 2 & -4 & -3 & 1 \\ 3 & 30 & -4 & 8 \\ 1 & -5 & -3 & -2 \\ 2{,}5 & -4 & 2 & -30 \end{pmatrix}$	$\begin{pmatrix} -1 \\ 0{,}9 \\ -0{,}8 \\ -0{,}7 \end{pmatrix}$
11	$\begin{pmatrix} 5 & 1 & 1 & 1 \\ 2 & 8 & 2 & -1 \\ 3 & 2 & 9 & -3 \\ 4 & 2 & -5 & 10 \end{pmatrix}$	$\begin{pmatrix} 10 \\ 2 \\ 2 \\ 1 \end{pmatrix}$
12	$\begin{pmatrix} 9 & 3 & 3 \\ 5 & 10 & -1 \\ 1 & 4 & 5 \end{pmatrix}$	$\begin{pmatrix} -2 \\ -9 \\ -6 \end{pmatrix}$

Задание 4. Вычислите точностные оценки методов по координатам:

$$\delta = \max|x_i - x_i^*|, \text{при } i = 1, 2, ..., n,$$

где x_i* — значение, найденное методом Гаусса.

Задание 5. На основании полученных результатов проведите сравнительный анализ методов по точности и времени решения, сделайте вывод.

Примечание. Погрешность полученных решений не должна превышать ε = 0,0001.

Контрольные вопросы

1. Что называется СЛАУ и её решением?
2. Какие системы называются совместными, какие несовместными?
3. Какую систему называют треугольной? Каковы её особенности?
4. Какие преобразования называют эквивалентными? Приведите пример.
5. Назовите известные вам методы и средства решения СЛАУ.
6. Объясните сущность метода Гаусса с выбором главного элемента (прямой ход, обратный ход).
7. Каковы сущность и особенности сходимости итерационных методов решения СЛАУ?
8. В чём особенности метода Зейделя (как выбрать начальное приближение, чем обосновать полученное по этому методу решение)?

Лабораторная работа 2
РЕШЕНИЕ НЕЛИНЕЙНЫХ УРАВНЕНИЙ

Цель работы: изучить различные методы решения нелинейных (и трансцендентных) уравнений; рассмотреть возможности, предоставляемые для решения нелинейных уравнений математическим пакетом MathCad; научиться создавать приложения для решения нелинейных уравнений в среде Delphi.

Постановка задачи. В практике научных и инженерных расчётов возникает необходимость решения нелинейных уравнений вида $f(x) = 0$. Функция определена и непрерывна на некотором конечном или бесконечном интервалах.

Если функция представляет собой многочлен, то такое уравнение называется **алгебраическим.**

Если x находится под знаком трансцендентной функции (логарифмическая, показательная, тригонометрическая и т. д.), то в таких случаях уравнение называется *трансцендентным* [6].

Если известно значение x^*, при котором выполняется условие $f(x^*)=0$, то x^* называется *корнем уравнения*. Такой корень геометрически представляет собой абсциссу точки пересечения (или касания) графика функции $y=f(x)$ с осью Ox [5]. Будем также полагать, что уравнение $f(x^*)=0$ имеет лишь изолированные корни, т. е. для каждого из корней существует окрестность, не содержащая других корней этого уравнения. В общем случае уравнение $f(x)=0$ не имеет аналитического решения. Так как на практике встречаются уравнения, содержащие коэффициенты с приближёнными значениями, то для решения такого уравнения используют численные приближённые методы, которые позволяют определять приближённое значение корней с заданной точностью.

Процесс решения нелинейного уравнения состоит из двух этапов:

1. *Этап отделения корней,* где производится установление количества корней и поиск интервалов, внутри которых содержится по одному изолированному корню уравнения.

2. *Этап уточнения значений корней,* на котором значения отдельных корней уточняются одним из известных методов до некоторой заданной степени точности [2].

Методы отделения корней. Ограничимся рассмотрением методов поиска лишь действительных корней. Существуют два способа отделения корней:

1. *Графический способ* заключается в обнаружении точек пересечения графика функции $y=f(x)$ с осью *Ox*. Абсциссы точек пересечения и выбираются в качестве начальных приближённых значений корней.

Часто уравнение $f(x)=0$ можно привести к виду $f_1(x)=f_2(x)$. Тогда отделить корни можно, построив графики функций $f_1(x)$ и $f_2(x)$ и обнаружив точки их пересечения.

2. *Численный способ* отделения корней основывается на следующем критерии. Если функция $f(x)$ непрерывна на отрезке *AB*

и принимает на концах этого отрезка значения противоположных знаков, а производная функции на отрезке AB сохраняет постоянный знак, то внутри отрезка существует корень уравнения, и при этом единственный.

Алгоритм численного отделения корней:

1. Найдите производную функции $f'(x)$.
2. Найдите критические точки функции $f(x)$.
3. Составьте таблицу знаков функции $f(x)$ на границах отрезка AB и в критических точках.
4. Определите отрезки, на концах которых функция принимает значения противоположных знаков.
5. Выберите в качестве начальных приближённых значений корней по одному произвольному (если метод уточнения корней, который планируется использовать, не налагает каких-либо дополнительных условий) значению x внутри каждого отрезка, найденного в пункте 4 данного алгоритма.

Следует отметить, что универсальных способов отделения корней, пригодных для любых уравнений, не существует.

Приближённые значения корней уточняют различными итерационными методами. Под *итерационным методом* подразумевается построение числовой последовательности корней, сходящихся к искомому корню x^*.

Одной из важнейших характеристик итерационного метода является его *скорость сходимости*.

Говорят, что последовательность $\{x_k\}$, сходящаяся к пределу x^*, имеет порядок сходимости α, если существуют числа $q \geq 0$ и $k_0 \geq 0$, такие, что для любого $k > k_0$ выполняется неравенство

$$|x_{k+1} - x^*| \leq q|x_k - x^*|^{\alpha}.$$

При $\alpha = 1$ сходимость называется *линейной*, при $\alpha = 2$ — квадратичной, а при $\alpha > 2$ — сверхквадратичной. Чем больше порядок сходимости, тем сложнее вычислительный алгоритм, но выше скорость сходимости итерационной последовательности.

Метод деления отрезка пополам. Для применения этого метода функция $f(x)$ должна быть непрерывна и ограничена на отрезке AB, внутри которого имеется корень. Значения функции на концах отрезка при этом должны иметь разные знаки $f(a)f(b)<0$ (рис. 2.1).

Данный метод предусматривает следующие действия:

1. Отрезок AB делят пополам и находят начальное приближение корня $c=\dfrac{a+b}{2}$.

2. Если $f(c)=0$, то c — корень уравнения, иначе из отрезков $[a; c]$ и $[c; b]$ выбирают тот, на концах которого функция $f(x)$ имеет разные знаки и который, следовательно, содержит искомый корень. Выбор интервала сводится к проверке двух условий. Если $f(a)f(c)<0$, то корень расположен на отрезке $[a; c]$ и $b := c$, если $f(b)f(c)<0$, то корень расположен на отрезке $[c; b]$ и $a := c$.

3. Найденный интервал, который в большинстве случаев удобно повторно переобозначить как $[a; b]$, вновь делят пополам и выполняют поиск отрезка, содержащего корень и т. д. Если требуется определить корень с точностью ε, то деление пополам продолжают до тех пор, пока длина отрезка не станет меньше 2ε. В этом случае середина последнего отрезка и есть корень с требуемой точностью [6].

Метод деления отрезка пополам всегда сходится.

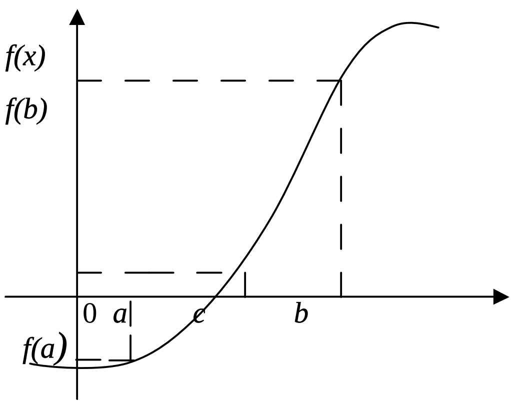


Рисунок 2.1 — Геометрический смысл метода деления отрезка пополам

Метод касательных (метод Ньютона). Пусть на отрезке $[a; b]$ имеется корень уравнения $f(x)=0$. Функция $f(x)$ на этом отрезке непрерывна, а производные функции $f'(x)$ и $f''(x)$ определены, непрерывны и сохраняют постоянные знаки. Таким образом, на отрезке $[a; b]$ функция монотонна и не меняет характера выпуклости.

Данный метод предусматривает следующие действия:

1. Выбирают начальное приближение корня x_0, в качестве которого удобно взять конец отрезка $[a; b]$, для него $f(x_0)f''(x_0)>0$ (в противном случае сходимость метода Ньютона не гарантируется).

2. Далее проводят касательную в точке C_0 к кривой $y=f(x)$ до пересечения с осью абсцисс. Абсциссу x_1 принимают за очередное приближение корня (рис. 2.2). Каждое последующее приближение определяется из рекуррентного соотношения

$$x_k = x_{k-1} - \frac{f(x_{k-1})}{f'(x_{k-1})}.$$

3. Для оценки расстояния очередного приближения x_k до корня x^* воспользуемся следующими рассуждениями. В соответствии с формулой Лагранжа $f(x_k) - f(x^*) = f'(c)(x_k - x^*)$ получаем, что $|x_k - x^*| = \dfrac{|f(x_k)|}{|f'(c)|}$.

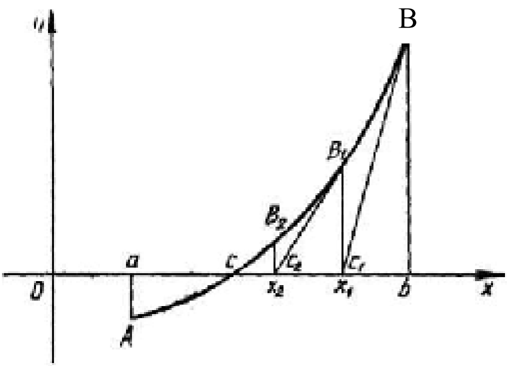


Рисунок 2.2 — Геометрический смысл метода Ньютона

О точке c известно лишь то, что она находится между x_k и x^*. Поэтому оценка погрешности возможна с помощью следующего соотношения:

$$\left|x_k - x^*\right| \le \frac{\left|f(x_k)\right|}{M}, \tag{2.1}$$

где $M = \min\limits_{[a;b]}\left|f'(x)\right|$.

Метод Ньютона имеет квадратичную сходимость. Иногда, если сложно выбрать начальное приближение, начинают решение методом деления отрезка пополам (т. е. берут середину отрезка) и продолжают уточнять с помощью метода Ньютона [5].

Метод хорд. При реализации метода касательных, при каждой итерации нужно вычислять значение как функции $f(x)$, так и её первой производной. Существует вариант метода Ньютона — метод хорд (секущих), который позволяет избежать вычисления производной. Рекуррентное соотношение для вычисления приближённых значений корня в этом методе имеет вид

$$x_k = \frac{cf(x_{k-1}) - x_{k-1}f(c)}{f(x_{k-1}) - f(c)}.$$

В качестве c выбирается конец отрезка (точка a или b), для которого выполняется условие $f(c)f''(c) > 0$. Для начального приближения x_0 выбирается конец отрезка, оставшийся после выбора c (если $c = b$, то $x_0 = a$ и наоборот).

Оценка степени приближения к корню производится так же, как и при использовании метода касательных.

Название методу было дано из-за его геометрического смысла: если $c = b$, а $x_0 = a$, то следующее приближение корня x_1 соответствует точке пересечения хорды, соединяющей концы кривой, с осью абсцисс. Далее на кривой находится точка с абсциссой x_1, проводится следующая хорда и т. д. (рис. 2.3).

Этот метод, как и метод касательных, также имеет квадратичную сходимость [3].

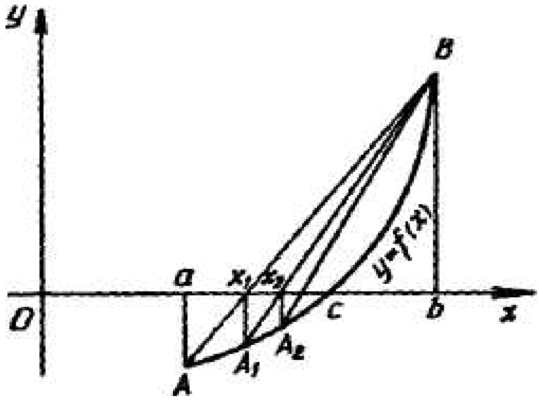


Рисунок 2.3 — Геометрический смысл метода хорд

Сравнение методов. Наибольшей универсальностью обладает метод деления отрезка пополам. С его помощью можно решить любое уравнение вида $f(x) = 0$, если его корни изолированы, а функция $f(x)$ на отрезках, содержащих корни, непрерывна. Методы последовательных приближений, касательных и хорд предъявляют к функции более жёсткие требования. Сходимость этих методов зависит от выбора начального приближения корня. При реализации этих методов необходимо вычислять производные этих функций для организации итерационного процесса и выполнять проверку условия сходимости.

Методы деления отрезка пополам и последовательных приближений имеют линейную сходимость, а методы касательных и хорд — квадратичную.

При выборе метода уточнения корней нужно помнить, что скорость сходимости и быстрота решения задачи — вещи совершенно разные. Поэтому при привлечении ЭВМ для решения нелинейных уравнений частое использование более простого метода с малой скоростью сходимости, например, метода деления отрезка пополам, может дать результат гораздо быстрее, чем использование изощрённого и сложного для понимания и программирования метода уточнения с высокой скоростью сходимости.

Задания

Задание 1. Разработайте алгоритмы решения нелинейных уравнений всеми рассмотренными методами: деления отрезка пополам, хорд, касательных (Ньютона).

Задание 2. Найдите решения нелинейных уравнений, приведённых в таблице 2.1 (в соответствии со своим вариантом), с использованием функции **root**(…) математического пакета MathCad (приложение Б).

Т а б л и ц а 2.1 — Варианты заданий

Вариант	Уравнение	Область, содержащая корень	Метод решения
1-й	$0{,}25x^3 + x - 1{,}2502 = 0$	[0; 2]	Касательных
	$x + \sqrt{x} + \sqrt[3]{x} - 2{,}5 = 0$	[0,4; 1]	Деления пополам
	$x - \dfrac{1}{3 + \sin 3{,}6x} = 0$	[0; 0,85]	Хорд
2-й	$0{,}1x^2 - x\ln x = 0$	[1; 2]	Деления пополам
	$\mathrm{tg}x - \dfrac{1}{3}\mathrm{tg}^3 x + \dfrac{1}{5}\mathrm{tg}^5 x - \dfrac{1}{3} = 0$	[0; 0,8]	Хорд
	$x - 2 + \sin\dfrac{1}{x} = 0$	[1,2; 2]	Касательных
3-й	$e^x + \ln x - 10x = 0$	[3; 4]	Деления пополам
	$3\sin 8x = 0{,}7x - 0{,}9$	[-1; 1]	Хорд
	$1 - x + \sin x - \ln(1 + x) = 0$	[0; 1,5]	Касательных
4-й	$3x - 14 + e^x - e^{-x} = 0$	[1; 3]	Деления пополам
	$\sqrt{1 - x} - \mathrm{tg}x = 0$	[0; 1]	Хорд
	$x + \cos(x^{0,52} + 2) = 0$	[0,5; 1]	Касательных
5-й	$3\ln^2 x + 6\ln x - 5 = 0$	[1; 3]	Деления пополам
	$\sin x^2 + \cos x^2 - 10x = 0$	[0; 1]	Хорд
	$\ln x - x + 1{,}8 = 0$	[2; 3]	Касательных

Окончание табл. 2.1

Вариант	Уравнение	Область, содержащая корень	Методы решения
6-й	$x\operatorname{tg}x - \frac{1}{3} = 0$	[0,2; 1]	Деления пополам
	$\operatorname{tg}\frac{x}{2} - \operatorname{ctg}\frac{x}{2} + x = 0$	[1; 2]	Хорд
	$0,4 + \operatorname{arctg}\sqrt{x} - x = 0$	[1; 2]	Касательных
7-й	$\sqrt{1-x} - \cos\sqrt{1-x} = 0$	[0; 1]	Деления пополам
	$0,6 \cdot 3^{x} - 2,3x - 3 = 0$	[2; 3]	Касательных
	$0,25x^{3} + x - 1,25 = 0$	[0; 2]	Хорд
8-й	$3x - 4 \cdot \ln x - 5 = 0$	[2; 4]	Касательных
	$e^{x} + \ln x - 10x = 0$	[3; 4]	Хорд
	$\ln(0,9146x) - 1,038x + 2,5 = 0$	[1; 6]	Деления пополам
9-й	$1,2755x^{3} - 3,601x^{2} - 1,37x + 6,76 = 0$	[-2; 2]	Деления пополам
	$3\sin\sqrt{x} + 0,35x - 3,8 = 0$	[2; 3]	Хорд
	$e^{x} + e^{-x} - 2 = 0$	[0; 1]	Касательных
10-й	$3\cos\sqrt{x} + 0,45x - 2,8 = 0$	[0; 2]	Деления пополам
	$x - 2 + \sin\frac{1}{x} = 0$	[1,2; 2]	Хорд
	$1 - x + \sin x - \ln(1+x) = 0$	[0; 1,5]	Касательных
11-й	$x^{2} - \ln(1+x) - 3 = 0$	[2; 3]	Касательных
	$0,41 - \frac{1}{3 - \sin 3,6x} = 0$	[0; 0,4]	Хорд
	$\ln x - x + 1,8 = 0$	[2; 3]	Деления пополам
12-й	$0,1x^{2} - x\ln x = 0$	[1; 2]	Касательных
	$x + \cos(x^{0,52} + 2) = 0$	[0,5; 2]	Хорд
	$\sqrt{1 - 0,4x} - \arcsin x = 0$	[0; 1]	Деления пополам

Для простейших уравнений вида *f(x)* = 0 решение в Mathcad находится с помощью функции **root**(…):

1) синтаксис: **root**(*f*(*x*1, *x*2, …), *x*1, *a, b*) — возвращает значение x1, принадлежащее отрезку [a, b], при котором выражение или функция *f(x)* обращается в 0. Оба аргумента этой функции должны быть скалярами. Функция возвращает скаляр;

2) аргументы:

– *f*(*x*1, *x*2, …) — функция, определённая где-либо в рабочем документе, или выражение, равное нулю. Выражение должно возвращать скалярные значения;
– *x*1 — имя переменной, которая используется в выражении. Этой переменной перед использованием функции **root** необходимо присвоить числовое значение. Mathcad использует его как начальное приближение при поиске корня;
– *a, b* — границы интервала, в котором лежит корень уравнения. Внутри интервала не должно быть больше одного корня, так как MathCad выводит на экран лишь один корень, лежащий внутри интервала. Эти параметры необязательны, если используются, то должны быть вещественными числами, причем $a < b$.

Рекомендации по использованию функции **root***(...):*

– для изменения точности, с которой функция **root** ищет корень, нужно изменить значение системной переменной TOL. Если значение TOL увеличивается, функция **root** будет сходиться быстрее, но ответ будет менее точен. Если значение TOL уменьшается, то функция **root** будет сходиться медленнее, но ответ будет более точен. Чтобы изменить значение TOL в определённой точке рабочего документа, используйте определение вида TOL := 0,01. Чтобы изменить значение TOL для всего рабочего документа, выберите команду **Инструменты: Свойства таблицы: Переменные:** Þ **Допуск сходимости (TOL)**;

– если два корня расположены близко друг от друга, следует уменьшить TOL, чтобы различить их;

– если функция *f*(*x*) имеет малый наклон около искомого корня, функция **root**(*f*(*x*), *x*) может *сходиться* к значению *r*, отстоящему от корня достаточно далеко. В таких случаях для нахождения более точного значения корня необходимо уменьшить значение TOL. Другой вариант заключается в замене уравнения $f(x) = 0$ на $g(x) = 0$:

$$g(x) = \frac{f(x)}{d/dx\, f(x)}.$$

Для выражения $f(x)$ с известным корнем a нахождение дополнительных корней $f(x)$ эквивалентно поиску корней уравнения $h(x) = f(x) / (x - a)$. Подобный приём полезен для нахождения корней, расположенных близко друг к другу. Проще искать корень выражения $h(x)$, чем пробовать искать другой корень уравнения $f(x) = 0$, выбирая различные начальные приближения [7].

Отсутствие сходимости функции root(...). Если после многих итераций Mathcad не находит подходящего приближения, то появится сообщение Can't converge to a solution. (отсутствует сходимость). Эта ошибка может быть вызвана следующими причинами:

– уравнение не имеет корней;

– корни уравнения расположены далеко от начального приближения;

– выражение имеет локальные max и min между начальным приближением и корнями;

– выражение имеет разрывы между начальными приближениями и корнями;

– выражение имеет комплексный корень, но начальное приближение было вещественным.

Чтобы установить причину ошибки, исследуйте график $f(x)$. Он поможет выяснить наличие корней уравнения $f(x) = 0$ и, если они есть, определить приблизительно их значения. Чем точнее выбрано начальное приближение корня, тем быстрее будет **root** сходиться [9].

Приближённые значения корней (начальные приближения) могут быть:

– известны из физического смысла задачи;

– известны из решения аналогичной задачи при других исходных данных;

– найдены графическим способом.

Наиболее распространен графический способ определения начальных приближений.

Пример выполнения задания

2.1. Графически отделите корни уравнения $x \cdot \log x = 1$:

а) представьте уравнение в виде двух функций:

$$y1(x) := \log(x) \qquad y2(x) := \frac{1}{x}$$

б) постройте графики функций, выполнив команду меню **Вставка: Графики: Зависимость XY**, и в появившейся рамке слева введите y1(x), y2(x), а снизу введите x (рис. 2.4);

в) корень найдите по графику, увеличив масштаб (рис. 2.5). Точка пересечения двух графиков и будет искомым корнем уравнения. Для этого выполните следующие действия:

– выделите график, щёлкнув левой кнопкой мыши внутри графика;
– в главном меню MathCad выберите команду **Формат: Графики: Масштаб (Format:Graph:Zoom)**;
– при нажатой левой кнопке мыши обведите пунктирной линией область графика вблизи искомого корня, которую надо увеличить;
– в открытом окне **Масштаб** по осям X-Y (X-Y **Zoom**) нажмите кнопку **Zoom**;

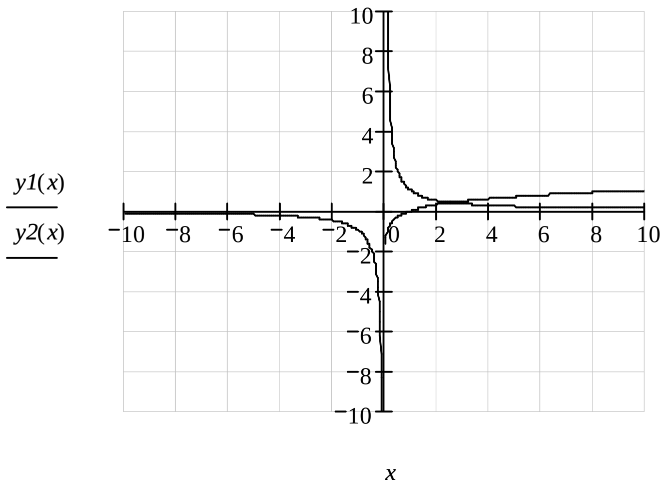


Рисунок 2.4 — Вид графиков в среде MathCad

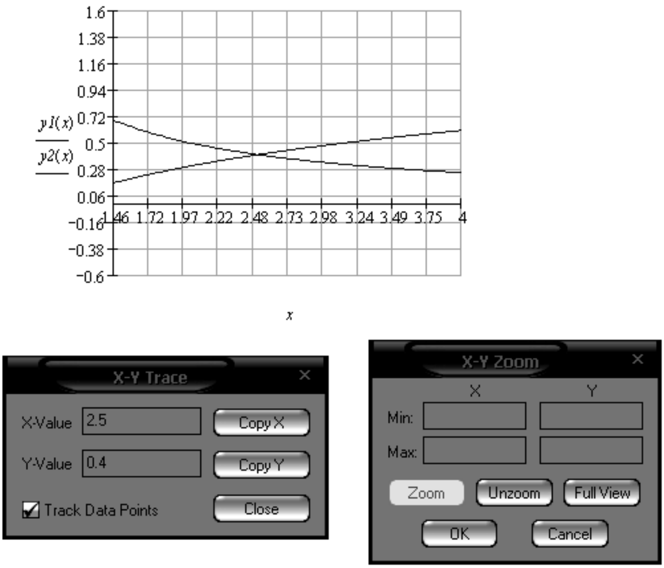


Рисунок 2.5 — Определение корня уравнения

г) прямо с графика передайте в буфер обмена численное значение корня. Для этого выполните следующие действия:

– выделите график, щёлкнув левой кнопкой мыши внутри него;
– в главном меню выберите команду **Формат:Графики: Трассирование (Format:Graph:Trace)** (*см.* рис. 2.5);
– после щелчка левой кнопкой мыши внутри графика появится перекрестье осей, которое с помощью мыши установите на пересечение графиков. При этом численные значения координат перекрестья появляются в открытом окне X-Y **Trace** (**Трассировка** X,Y);
– правильно выбрав положение перекрестья, нажмите кнопки **Copy X** и **Copy Y** — численные значения будут помещены в буфер (*см.* рис. 2.5);
– вне поля графика запишите имя для корня и оператор присваивания :=. Нажмите кнопку **Paste (Вставить)** в стандартном меню MathCad;

д) приближённо нашли единственный корень $x \approx 2{,}5$ уравнения $x \log x = 1$ и определили его содержащий отрезок [2; 3].

2.2. Уточните корни уравнения $x \cdot \log x = 1$ с помощью функции root*(...)* следующим образом:

а) для локализации корней удобно построить график функции. Для этого выполните следующие шаги:

– определите функцию, приравняв её к нулю, т. е. введите $f(x) := x \cdot \log(x) - 1$;
– выберите пункты меню **Вставить:Графики:Зависимость XY;**
– под осью абсцисс введите *x,* слева от оси ординат — *f(x)*;
– при необходимости корректируйте пределы изменения аргумента и функции;

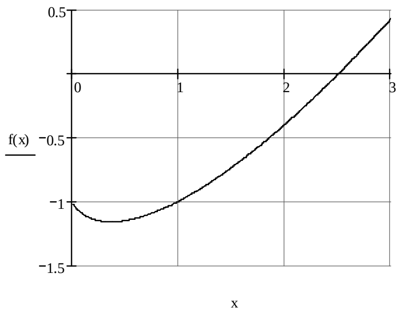
б) основные свойства графика можно настроить, выбрав пункт **Формат** в контекстном меню, вызываемом щелчком правой кнопкой мыши на графике;

в) анализируя график, определите начальные приближения корней уравнения;

г) установите точность вычислений, изменяя значения пункта **Порог сложности** на закладке **Толерантность (Tolerance)** окна, вызываемого командой **Формат-Результат (Format-Result).** На закладке **Формат** номера того же окна выберите способ отображения результата и количество отображаемых десятичных знаков;

д) введите начальное приближение *x,* которое определите визуально по графику. В данном случае *x* := 2. Далее введите функцию **root(***f(x),x)*= и получите ответ, который выводится автоматически самой средой (рис. 2.6).

f(x) := x · log(x) −1



x:=2

root(f(x), x) = 2.506

Рисунок 2.6 — Уточнение корня уравнения с помощью функции **root**(...)

Примечание. В окне **X-Y Trace** (Трассировка X,Y) есть пункт **Track Data Points** (Отмечать расчётные точки). Если установить этот флажок, при перемещении мыши пунктирное перекрестье на графике будет перемещаться скачками, отмечая расчётные значения функции. Если флажок снять, движение перекрестья становится плавным.

Задание 3. Средствами Delphi создайте приложение для решения нелинейных уравнений, приведённых в таблице 2.1 (в соответствии со своим вариантом). Начальные приближения корней выбирайте из указанных областей. (При выполнении данного задания используйте приложение Б).

Пример выполнения задания

Решите нелинейное уравнение вида $f(x) = 3\sin(\sqrt{x}) + \dfrac{x}{15} - 1{,}8$ методами хорд, касательных и делением отрезка пополам.

3.1. Расположите на своей форме основные компоненты среды Delphi, с помощью которых будете решать нелинейное уравнение. Примерный вид формы представлен на рисунке 2.7.

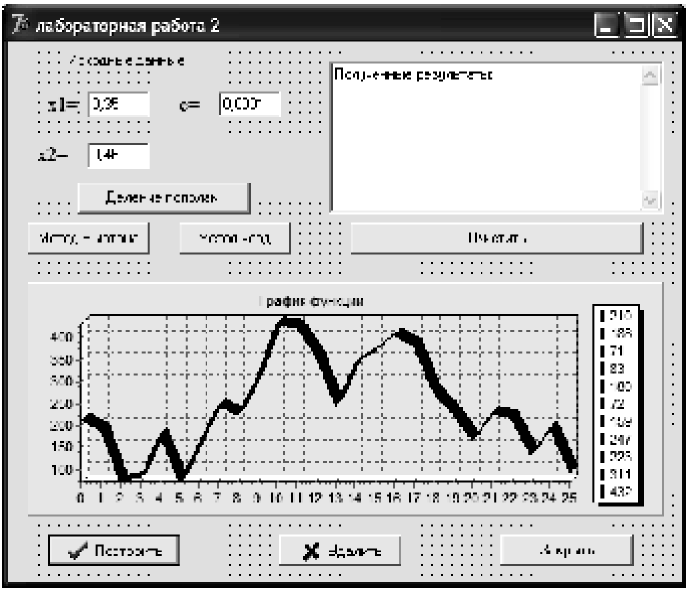


Рисунок 2.7 — Общий вид формы в среде Delphi

3.2. Для каждой кнопки формы создайте процедуры-обработчики событий, предварительно описав в тексте программы функции, определяющие данное уравнение, первые и вторые производные исходной функции. Для функции вида $f(x) = 3\sin(\sqrt{x}) + \frac{x}{15} - 1{,}8$ это выглядит так:

а) исходная функция опишется следующим образом [1]:

```
function f(x:real):real;
begin
f := 3*sin(sqrt(x))+x/15–1.8;
end;
```

б) первая производная опишется так:

```
function p(x:real):real;
begin
p := (3*cos(sqrt(x)))/(2*sqrt(x))+1/15;
end;
```

в) вторая производная опишется следующим образом:

```
function pp(x:real):real;
begin
pp := –3/4*(sin(sqrt(x))/x)–3/4*(cos(sqrt(x))/exp(3/2*ln(x)));
end;
… .
```

Примечание. Часто при нахождении первых и вторых производных сложных функций используются возможности MathCad. Для этого:

а) определите начальную функцию, для данного примера она имеет вид

$$f(x) = 3\sin(\sqrt{x}) + \frac{x}{15} - 1{,}8.$$

Помните, что в среде MathCad при записи числа целая и дробная части отделяются точкой;

б) далее, используя компонент панели **Calculus**  и **Evaluation** , введите выражение

$$\frac{d}{dx} f(x) \rightarrow$$

в) отщёлкнув кнопкой мыши в сторону, получите вид первой производной;

г) для нахождения второй производной необходимо воспользоваться компонентом панели **Calculus**  и **Evaluation**  и выполнить следующие действия: $\frac{d^2}{dx^2} f(x) \rightarrow$;

д) результатом будет вид второй производной:

$$\frac{d^2}{dx^2} f(x) \rightarrow \frac{-3}{4} \frac{\sin(x^{\frac{1}{2}})}{x} - \frac{3}{4} \frac{\cos(x^{\frac{1}{2}})}{x^{\frac{3}{2}}}.$$

3.3. Текст модуля для нахождения корня методом деления пополам (кнопка **Деление пополам**):

```
procedure TForm1.Button1Click(Sender: TObject);
var x1,x2,e,c: real;
begin
e := strtofloat(Edit1.Text);
x1 := strtofloat(Edit2.Text);
x2 := strtofloat(Edit3.Text);
while abs(x2–x1)>e do
begin
c := (x2+x1)/2;
if f(x1)*f(c)<0 then
x2 := c
else x1 := c;
end;
c := (x2+x1)/2;
Memo1.Lines.Add('Корень по методу деления пополам: '+ floattostrf(c,ffgeneral, 5,3));
end;
… .
```

3.4. Текст модуля для нахождения корня методом касательных (кнопка **Метод Ньютона**):

```
procedure TForm1.Button2Click(Sender: TObject);
var x1, x2, e, c, d, b, s: real;
begin
```

```
e := strtofloat(Edit1.Text);
x1 := strtofloat(Edit2.Text);
x2 := strtofloat(Edit3.Text);
b := x2;
if f(b)*pp(b)<0 then b := x1;
repeat
s := b–f(b)/p(b);
if abs(s)>1
then d := abs((s–b)/s)
else d := abs(s–b);
b := s;
until (d<e);
Memo1.Lines.Add('Корень по методу касательных: '+floattostrf (s, ffgeneral,5,3));
end;
… .
```

3.5. Модуль для решения уравнения методом хорд (кнопка **Метод хорд**):

```
procedure TForm1.Button3Click(Sender: TObject);
var s1, x, x0,c,e,x1,x2,:real;
begin
e := strtofloat(Edit1.Text);
x1 := strtofloat(Edit2.Text);
x2 := strtofloat(Edit3.Text);
x := x1;
c := x2;
if f(c)*pp(c)<0 then begin c := x1; x := x2 end;
repeat
x0 := x;
x := (c*f(x0)–x0*f(c))/(f(x0)–f(c));
until abs(f(x))/0.00001<=e;
s1 := x;
Memo1.Lines.Add('Корень по методу хорд:'+floattostrf(s1,ffgeneral, 5,3));
end;
… .
```

3.6. Модуль для очистки окна **Memo** (кнопка **Очистить**):

```
procedure TForm1.Button4Click(Sender: TObject);
begin
Memo1.Clear;
end;
… .
```

3.7. Текст модуля закрытия приложения (кнопка **Закрыть**):

```
procedure TForm1.Button5Click(Sender: TObject);
begin
close;
end;
… .
```

3.8. Модуль для вывода графика (кнопка **Построить**):

```
procedure TForm1.BitBtn1Click(Sender: TObject);
var x1,x2,x,h,f:real;
begin
x1 := strtofloat(Edit2.Text);
x2 := strtofloat(Edit3.Text);
h := strtofloat(Edit4.Text);
x := x1;
while x<=x2 do
 begin
 f := 3*sin(sqrt(x))+x/15–1.8;
 series1.addXY(x,f);
  x := x+h;
  end;
end;
… .
```

3.9. Модуль для удаления графика (кнопка **Удалить**):

```
procedure TForm1.BitBtn2Click(Sender: TObject);
begin
```

```
 series1.Clear;
end;
… .
```

3.10. Запустите приложение и получите корни уравнения разными методами. Сравните результаты.

3.11. Доработайте программу и решите задание согласно своему варианту.

3.12. Самостоятельно продумайте, как вычислить погрешности метода хорд и касательных по формуле (2.1). Реализуйте это в своем приложении.

Примечание. Погрешность полученных решений не должна превышать $\varepsilon = 0{,}0001$.

Задание 4. Оцените полученные результаты в Delphi и MathCad, сделайте вывод.

Контрольные вопросы

1. Какие методы решения нелинейных алгебраических уравнений вы использовали при выполнении работы?
2. Какой метод отделения корней вы применяли?
3. Опишите алгоритм уточнения корней методом:
   а) половинного деления;
   б) хорд;
   в) касательных (Ньютона).
4. Проведите сравнительный анализ различных методов решения нелинейных уравнений.
5. Назовите способы нахождения начального приближения.
6. Какие аргументы функции **root**(...) не обязательны?
7. В каких случаях MathCad не может найти корень уравнения?
8. Какая системная переменная отвечает за точность вычислений?
9. Как изменить точность, с которой функция **root** ищет корень?
10. Как системная переменная TOL влияет на решение уравнения с помощью функции **root**?

Лабораторная работа 3
АППРОКСИМАЦИЯ И ИНТЕРПОЛИРОВАНИЕ ФУНКЦИЙ

Цель работы: изучить основные методы интерполирования и аппроксимации функций; научиться использовать для решения задач интерполирования и аппроксимации табличный процессор MS Excel и математический пакет MathCad; научиться создавать приложения для решения задач интерполирования и аппроксимации в среде Delphi.

Постановка задачи. На практике в результате экспериментальных исследований часто получают набор значений некоторой величины y при фиксированных значениях второй величины x. Аналитическая зависимость между значениями x и y чаще всего неизвестна, что не позволяет, например, вычислить значение величины y в промежуточных точках, отличных от полученных экспериментально. Для нахождения значений в этих промежуточных точках строят приближённую функцию $\varphi(x)$. Значения функции $\varphi(x)$ в имеющихся точках x либо совпадают, либо приближены к экспериментально наблюдаемым значениям y в этих точках. Построение функции $\varphi(x)$ называется *интерполированием* [6]. Таким образом, график функции $\varphi(x)$ должен проходить через все экспериментально полученные точки (x_i, y_i).

К интерполированию прибегают и в тех случаях, когда аналитический вид некоторой функции $f(x)$ известен, но получение её значений в нужных точках требует громоздких вычислений. Чтобы этого избежать, функцию $f(x)$ также заменяют приближенной функцией $\varphi(x)$.

В качестве приближённой функции $\varphi(x)$ часто используется алгебраический многочлен (полином) вида

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

Объясняется это тем, что сравнительно просто автоматизировать вычисления коэффициентов многочлена, легко его интегрировать и дифференцировать. Наряду с многочленами для аппроксимации используют элементарные функции и ряды Фурье.

Интерполяционный многочлен Лагранжа. *Интерполяционный многочлен Лагранжа* степени n, принимающий в точках x_i отрезка $[a; b]$ значения y_i, где $i = \overline{0,n}$, имеет вид

$$P_n(x) = \sum_{i=0}^{n} y_i \prod_{\substack{j=0 \\ i \neq j}} \frac{x - x_j}{x_i - x_j}$$

или

$$P_n(x) = \sum_{i=0}^{n} y_i \frac{(x - x_0)(x - x_1) \cdot \ldots \cdot (x - x_{i-1})(x - x_{i+1}) \cdot \ldots \cdot (x - x_n)}{(x_i - x_0)(x_i - x_1) \cdot \ldots \cdot (x_i - x_{i-1})(x_i - x_{i+1}) \cdot \ldots \cdot (x_i - x_n)}.$$

Погрешность интерполяции определяется разностью $R(x) = y - P_n(x)$.

Таким образом, значение функции y в точке x, отличной от заданных x_i, может быть вычислено по формуле

$$y(x) \approx \sum_{i=0}^{n} y_i \prod_{\substack{j=0 \\ i \neq j}} \frac{x - x_j}{x_i - x_j}.$$

Эту функцию называют функцией Лагранжа.

Интерполяционный многочлен Ньютона. Интерполирование по формуле Лагранжа имеет следующий недостаток: в этой формуле каждое слагаемое представляет собой многочлен n-й степени. Следовательно, при увеличении числа точек x_i и, соответственно, y_i увеличивается степень многочлена Лагранжа, и так как каждое его слагаемое зависит от всех значений аргумента x, то вычисление полинома Лагранжа необходимо производить заново. Указанного недостатка нет при вычислении полинома Ньютона [5].

Введём понятие разделённых разностей.

Разделёнными разностями первого порядка называются значения:

$$f(x_1, x_0) = \frac{y_1 - y_0}{x_1 - x_0} = \frac{\Delta y_0}{x_1 - x_0},$$

$$f(x_2, x_1) = \frac{y_2 - y_1}{x_2 - x_1} = \frac{\Delta y_1}{x_2 - x_1}, \tag{3.1}$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

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

Значение выражения $\Delta y_{i-1} = y_i - y_{i-1}$, где $i = 1 \ldots n$, называется конечными разностями первого порядка.

Разделенными разностями 2-го и более высоких порядков называют значения:

$$f(x_2, x_1, x_0) = \frac{f(x_2, x_1) - f(x_1, x_0)}{x_2 - x_0} = \frac{\Delta^2 y_0}{x_2 - x_0}, \tag{3.2}$$

$$f(x_3, x_2, x_1, x_0) = \frac{f(x_3, x_2, x_1) - f(x_2, x_1, x_0)}{x_3 - x_0} = \frac{\Delta^3 y_1}{x_3 - x_0},$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \tag{3.3}$$

$$f(x_n, x_{n-1}, \ldots, x_1, x_0) = \frac{f(x_n, x_{n-1}, \ldots, x_1) - f(x_{n-1}, x_{n-2}, \ldots, x_0)}{x_n - x_0} = \frac{\Delta^{n-1} y_{n-1}}{x_n - x_0}.$$

Значение выражения $\Delta^2 y_{i-1} = \Delta y_i - \Delta y_{i-1}$, где $i = 1 \ldots n$, называется конечными разностями второго порядка и т. д.

С учётом введённых обозначений *интерполяционный многочлен Ньютона* имеет вид

$$P_n(x) = f(x_0) + (x - x_0) \cdot f(x_1, x_0) + (x - x_0) \cdot (x - x_1) f(x_2, x_1, x_0) +$$
$$+ (x - x_0)(x - x_1) \cdot \ldots \cdot (x - x_{n-1}) \cdot f(x_n, x_{n-1}, \ldots, x_1, x_0).$$

Погрешность рассчитывается по формуле

$$R(x) = \frac{f^{n+1}(\delta)}{(n+1)!}(x - x_0)(x - x_1) \cdot \ldots \cdot (x - x_n),\ \delta \in [a; b].$$

Так как любой i-й член полинома Ньютона зависит только от первых i-х точек x_i-х и от значения функции в них, добавление новых точек вызывает лишь добавление новых слагаемых без изменения первоначальных.

Аппроксимация функций методом наименьших квадратов. Аппроксимация является частным случаем интерполирования и применяется для определения аналитического вида функции, заданной таблично. *Задача аппроксимации* сводится к определению свободного параметра (параметров) функции заданного вида, который обеспечит наилучшее приближение функции, заданной таблично, модельной аналитической функцией [2].

Метод наименьших квадратов. Пусть таблично задана функция $y_i = f(x_i)$. Определить аппроксимационный многочлен вида

$$P_m = \sum_{i=0}^{m} a_i x^i = a_0 + a_1 x + a_2 x^2 + \ldots + a_m x^m. \qquad (3.4)$$

Для определения коэффициентов a_i (где $i = 1, 2, \ldots, m$) используют аппроксимацию методом наименьших квадратов. Функционал находится по формуле $S = \sum_{i=1}^{n} \left[P(x_i) - y_i\right]^2 \to 0$, где n — количество пар значений аргумента x_i и функции y_i. Так как S должен быть минимален, то первые частные производные от S по всем коэффициентам полинома должны равняться нулю.

Вычислим их и приравняем к нулю:

$$\begin{cases} \dfrac{dS}{dq_0} = 2\sum_{i=1}^{n} \left(a_0 + a_1 x_i + \ldots + a_m x_i^m - y_i\right) = 0, \\ \dfrac{dS}{dq_1} = 2\sum_{i=1}^{n} \left(a_0 + a_1 x_i + \ldots + a_m x_i^m - y_i\right) x_i; = 0, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \dfrac{dS}{dq_m} = 2\sum_{i=1}^{n} \left(a_0 + a_1 x_i + \ldots + a_m x_i^m - y_i\right) x_i^m = 0. \end{cases}$$

После преобразования система слегка упростится:

$$\begin{cases} \sum\limits_{i=1}^{n}\left(a_0 + a_1\, x_i + \ldots + a_m\, x_i^{\,m}\right) = \sum\limits_{i=1}^{n} y_i, \\ \sum\limits_{i=1}^{n}\left(a_0\, x_i + a_1\, x_i^{\,2} + \ldots + a_m\, x_i^{\,m+1}\right) = \sum\limits_{i=1}^{n} x_i y_i, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \sum\limits_{i=1}^{n}\left(a_0\, x_i^{\,m} + a_1\, x_i^{\,m+1} + \ldots + a_m\, x_i^{\,2m}\right) = \sum\limits_{i=1}^{n} x_i^{\,m} y_i. \end{cases} \tag{3.5}$$

Если полином первой степени, то получим два уравнения, если шестой степени, то — семь уравнений. Введём следующие обозначения:

$$t_0 = n, \quad t_1 = \sum_{i=1}^{n} x_i, \; \ldots, \; t_{2m} = \sum_{i=1}^{n} x_i^{\,2m},$$

$$C_0 = \sum_{i=1}^{n} y_0, \quad C_1 = \sum_{i=1}^{n} x_i y_i, \; \ldots, C_m = \sum_{i=1}^{n} x_i^{\,m} y_i.$$

С учётом этих обозначений система (3.5) примет следующий вид:

$$\begin{cases} t_0\, a_0 + t_1\, a_1 + \ldots + t_m\, a_m = C_0, \\ t_1\, a_0 + t_2\, a_1 + \ldots + t_{m+1}\, a_m = C_1, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ t_n\, a_0 + t_{m+1}\, a_1 + \ldots + t_{2m}\, a_m = C_m. \end{cases}$$

Решив эту систему одним из известных методов, определим ряд значений $a_0, a_1, \ldots, a_m$, с помощью которого и строится многочлен (3.4).

Среднеквадратичное отклонение вычисляется по формуле

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left[P(x_i) - y_i\right]^2}.$$

Задания

Задание 1. Разработайте схемы алгоритмов интерполирования функций по методам Лагранжа, Ньютона, наименьших квадратов.

Задание 2. Произведите интерполирование и аппроксимацию табличных функций, приведённых в таблице 3.1 (в соответствии со своим вариантом), на отрезке $[a; b]$ с шагом h средствами MS Excel и MathCad. Средствами Delphi создайте приложение, решающее аналогичную задачу.

Таблица 3.1 — Варианты заданий

Вариант	Значение аргумента *x*	Значение функции *y*	Пределы изменения аргумента *[A; B]*	Шаг интерполирования *H*
1-й	–1,060 –0,837 –0,684 –0,315 –0,117 0,000 0,115 0,500	1,220 0,854 0,513 0,271 0,217 0,198 0,218 0,277	[–0,8; 0,4]	0,2
2-й	–2,15 –1,83 –1,62 –1,45 –1,01 –0,72 –0,48 0,00	–2,23 –2,65 –3,10 –3,54 –4,26 –4,38 –4,52 –4,27	[–1,95; 0,1]	0,3
3-й	–0,210 –0,143 –0,099 –0,032 0,114 0,182 0,257 0,380	–12,64 –11,05 –10,25 –9,32 –9,25 –10,00 –11,48 –14,40	[–0,2; 0,28]	0,08

Продолжение табл. 3.1

Вариант	Значение аргумента *x*	Значение функции *y*	Пределы изменения аргумента *[A; B]*	Шаг интерполирования *H*
4-й	0,215 0,441 0,638 0,865 1,050 1,300 1,550 1,820	5,82 4,63 4,10 3,34 3,00 3,29 4,32 5,72	[0,4; 1,6]	0,2
5-й	–1,00 –0,96 –0,86 –0,79 0,22 0,50 0,93 1,10	–1,000 –0,151 0,894 0,986 0,895 0,500 –0,306 –0,510	[–0,98; 1,02]	0,35
6-й	0,50 0,75 1,00 1,25 1,50 1,75 2,00 2,25	0,1915 0,2734 0,3413 0,3944 0,4332 0,4599 0,4773 0,4878	[0,625; 2,126]	0,25
7-й	1,4 1,6 1,8 2,0 2,2 2,4 2,6 2,8	–0,24 –0,24 –0,16 0,00 0,24 0,56 0,96 1,44	[1,5; 2,7]	0,2
8-й	0,43 0,48 0,55 0,62 0,70 0,75 0,78 0,81	1,64 1,73 1,88 2,03 2,23 2,36 2,41 2,78	[0,5; 0,6]	0,02

Окончание табл. 3.1

Вариант	Значение аргумента *x*	Значение функции *y*	Пределы изменения аргумента *[A; B]*	Шаг интерполирования *H*
9-й	0 15 30 45 60 75 90 105	800 718 665 621 586 556 532 510	[68; 72]	0,5
10-й	–2,0 3,0 4,5 12,0 15,0 18,5 20,0 23,0	–17,0 –3,0 1,2 1,8 3,0 4,5 7,0 9,1	[9,8; 11]	0,2
11-й	–2,3 0,0 1,1 4,8 7,3 9,2 11,4 13,0	–12,5 8,6 13,4 15,1 21,4 24,2 28,3 32,1	[1,5; 3]	0,5
12-й	2,34 5,16 7,03 8,42 9,61 10,12 11,35 12,12	15,16 25,03 32,18 37,11 44,82 51,62 50,13 73,16	[3; 6]	0,5

2.1. Произведите интерполирование и аппроксимацию табличных функций на отрезке $[a;\ b]$ с шагом h средствами MS Excel. По результатам выполнения постройте графики функций.

Пример выполнения задания

Для функции, заданной таблицей 3.2, постройте интерполяционные многочлены Лагранжа и Ньютона. Используя полученные многочлены, вычислите приближённое значение функции на отрезке [–3,25; –1] с шагом 0,25.

2.1.1. Загрузите MS Excel. Введите исходные данные и вычислите многочлен Лагранжа для значений указанного отрезка (рис. 3.1). Для этого выполните следующие действия:

а) введите в ячейку B16 формулу

=$B$4*((A16–$A$5)*(A16–$A$6)*(A16–$A$7))/(($A$4–$A$5)*($A$4–$A$6)*($A$4–$A$7))+$B$5*((A16–$A$4)*(A16–$A$6)*(A16–$A$7))/(($A$5–$A$4)*($A$5–$A$6)*($A$5–$A$7))+$B$6*((A16–$A$4)*(A16–$A$5)*(A16–$A$7))/(($A$6–$A$4)*($A$6–$A$5)*($A$6–$A$7))+$B$7*((A16–$A$4)*(A16–$A$5)*(A16–$A$6))/(($A$7–$A$4)*($A$7–$A$5)*($A$7–$A$6));

б) далее копируйте её на блок B17: B25;

в) таким образом, в точке $x0$= –3,25 приближённое значение функции, вычисленное с помощью многочлена Лагранжа, равно $f(x)$= –1,13281;

г) по полученным данным постройте точечные графики функций (рис. 3.2).

2.1.2. Для построения интерполяционного многочлена Ньютона выполните следующие шаги:

а) скопируйте исходные данные на новый лист Excel и сделайте пояснительные записки (рис. 3.3);

Таблица 3.2 — Данные для вычисления приближённого значения функции

x	–4	–3	–2	–1
y	–3	–1	0	7

	A	B	C
1		Многочлен Лагранжа	
2	Исходные данные		
3	x	y	
4	-4	-3	
5	-3	-1	
6	-2	0	
7	-1	7	
8			
9	заданный отрезок		
10	x1=	-3,25	
11	x2=	-1	
12	h=	0,25	
13			
14	Полученные результаты:		
15	x	f(x)	
16	-3,25	-1,13281	
17	-3	-1	
18	-2,75	-0,92969	
19	-2,5	-0,8125	
20	-2,25	-0,53906	
21	-2	0	
22	-1,75	0,914063	
23	-1,5	2,3125	
24	-1,25	4,304688	
25	-1	7	

Рисунок 3.1 — Вычисление многочлена
Лагранжа в среде Excel

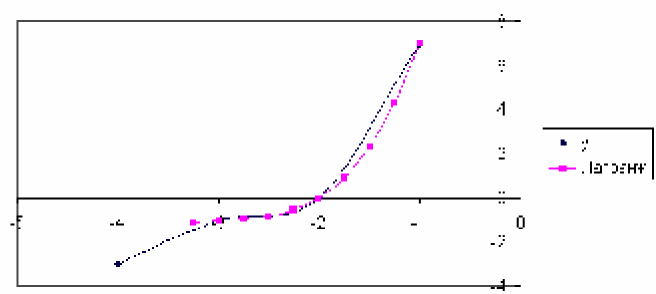


Рисунок 3.2 — Графики исходной и найденной по методу
Лагранжа функций

	A	B	C	D	E
1			Многочлен Ньютона		
2					
3	исходные данные		разделенные разности		
4	x	y	y1	y2	y3
5	-4	-3	=(B6-B5)/(A6-A5)	=(C6-C5)/(A7-A5)	=(D6-D5)/(A8-A5)
6	-3	-1	=(B7-B6)/(A7-A6)	=(C7-C6)/(A8-A6)	
7	-2	0	=(B8-B7)/(A8-A7)		
8	-1	7			
9					
10	заданный отрезок				
11	x1=	-3,25			
12	x2=	-1			
13	h=	0,25			

Рисунок 3.3 — Вид формул для вычисления разделённых разностей в Excel

б) введите разделённые разности первого порядка в ячейки C5:C7 [*см*. формулу (3.1)], разделённые разности второго и третьего порядков [*см*. формулы (3.2) и (3.3)] соответственно в ячейки D5:D6 и E5 (*см*. рис. 3.3);

в) далее получите значения x на заданном интервале, введя формулы в ячейки A16 и A17, как показано на рисунке 3.4;

	A
15	x
16	=B11
17	=A16+$B$13

Рисунок 3.4 — Вид формул для вычисления значений x на заданном интервале

д) формулу из ячейки A17 скопируйте на блок A18: A25;

е) в ячейку B16 введите формулу интерполяционного многочлена Ньютона: =$B$5+(A16–$A$5)*$C$5+(A16–$A$5)*(A16–$A$6)*$D$5+(A16–$A$5)*(A16–$A$6)*(A16–$A$7)*$E$5;

ж) скопируйте её на блок B17:B25. В результате получите числовые значения (рис. 3.5);

и) по полученным данным постройте точечные графики функций (рис. 3.6);

к) таким образом, в точке $x0$= –3,25 приближённое значение функции, вычисленное с помощью многочлена Ньютона, равно $f(x)$= –1,1328125.

14	Многочлен Ньютона на заданном отрезке			
15	x	f(x)		
16	-3,25	-1,1328125		
17	-3	-1		
18	-2,75	-0,9296875		
19	-2,5	-0,8125		
20	-2,25	-0,5390625		
21	-2	0		
22	-1,75	0,9140625		
23	-1,5	2,3125		
24	-1,25	4,3046875		
25	-1	7		

Рисунок 3.5 — Вычисление интерполяционного
многочлена Ньютона в Excel

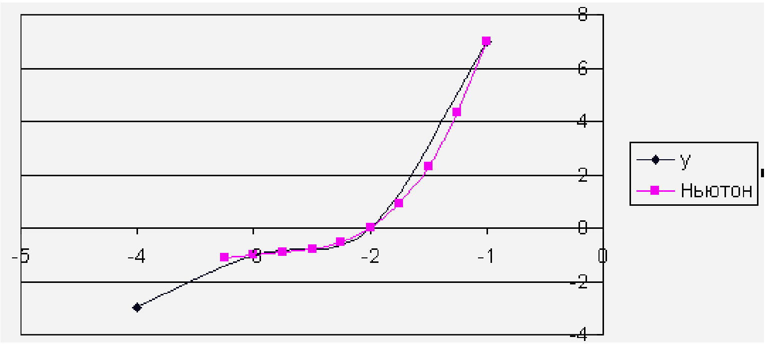


Рисунок 3.6 — Вид построенных графиков функций

2.1.3. Для аппроксимации функции методом наименьших квадратов выполните следующие действия:

а) скопируйте исходные данные на новый лист Excel;

б) постройте по исходным данным график функции и определите, на какую из стандартных функций он внешне наиболее похож. В рассматриваемом примере график напоминает ветвь параболы, а это значит, что аппроксимационный многочлен будет иметь следующий вид: $Pn = a0 + a1x + a2x^2$. Следовательно, необходимо найти значения $a0$, $a1$, $a2$. Для этого проведите вычисления методом наименьших квадратов (рис. 3.7);

	A	B	C	D	E	F	G	H	I
1	Метод наименьших квадратов								
2									
3	x	y		x^2	x^3	x^4			
4	-4	-2		=A4^2	=A4^3	=A4^4			
5	-3	-1		=A5^2	=A5^3	=A5^4			
6	-2	0		=A6^2	=A6^3	=A6^4			
7	-1	[illegible]		=A7^2	=A7^3	=A7^4			
8	Введем необходимые обозначения:								
9	n0	4		s0	=СУММ(B4:F7)	[illegible]	[illegible]	[illegible]	[illegible]
10	n1=	=СУММ(A4:A7)		s1=	=СУММ([illegible])	[illegible]	=A5*B5	[illegible]	[illegible]
11	n2=	=СУММ(D4:D7)		s2=	=СУММ([illegible])	[illegible]	[illegible]	[illegible]	[illegible]
12	n3	=СУММ(E4:E7)				[illegible]	[illegible]	[illegible]	[illegible]
13	n4=	=СУММ(F4:F7)							
14									

Рисунок 3.7 — Аппроксимация функции методом наименьших квадратов

в) после полученных преобразований методом наименьших квадратов создайте систему из трёх уравнений (блок A16:D18). Решите её любым известным способом (*см.* лабораторную работу 1). В примере используется метод Гаусса, т. е. система приводится к треугольному виду (блок A20:D26), и вычисляются корни системы уравнений (блок G16:G18) (рис. 3.8);

г) получив корни уравнения, определите вид многочлена и найдите значения y для заданного интервала x (*см.* рис. 3.8);

J20 ▾ fx =$G$18+$G$17*I20+$G$16*I20^2

	A	B	C	D	E	F	G	H	I	J
15	Составляем систему					корни уравнения				
16	4	-10	30	3		a2=	1,25			
17	-10	30	-100	8		a1=	6,4			
18	30	-100	354	-50		a0=	5,9			
19						вид многочлена:			x	y
20	5	-10	30	3		y=a0+a1*x+a2*x^2			-3,25	-1,696875
21	0	5	-25	15,5					-3	-2,05
22	0	-25	129	-73		заданный отрезок			-2,75	-2,246875
23						x1=	-3,25		-2,5	-2,2875
24	5	-10	30	3		x2=	-1		-2,25	-2,171875
25	0	10	-40	14		h=	0,25		-2	-1,9
26	0	0	4	5					-1,75	-1,471875
27									-1,5	-0,8875
28									-1,25	-0,146875
29									-1	0,75

Рисунок 3.8 — Метод наименьших квадратов для вычисления функции на заданном интервале

д) таким образом, в точке x0= –3,25, согласно методу наименьших квадратов, приближённое значение функции равно f(x)= –1,696875;

е) постройте точечные графики полученных значений функции методом наименьших квадратов и исходных данных (рис. 3.9).

2.2. Произведите интерполирование и аппроксимацию табличных функций на отрезке [*a; b*] с шагом *h* средствами MathCad.

2.2.1. В MathCad можно найти значения функции *y* в промежуточных точках с помощью полинома некоторой степени. Для этого используются следующие функции:

1) **regress**(*vx, vy, k*) (возвращает вектор, использующийся функцией **interp** для нахождения полинома степени *k*, который наилучшим образом приближает значения *x* и *y* данных, хранящихся в векторах *vx, vy*);

2) **interp**(*vs, vx, vy, x*) (возвращает приближённое значение *y*, соответствующее значению *x*, *vs* — вектор, получаемый с помощью **regress**) [7].

Пример выполнения задания

Решите рассматриваемую выше задачу, представленную в пункте 2.1 настоящего методического указания (с. 46), методом наименьших

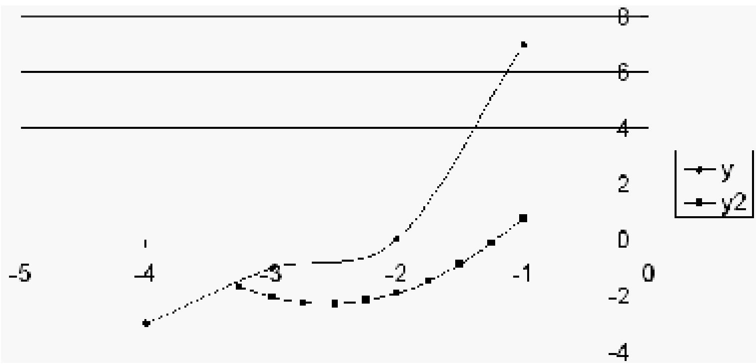


Рисунок 3.9 — Вид построенных графиков функций

квадратов, а затем с помощью встроенных функций MathCad. Для этого выполните следующие действия:

а) введите исходные данные:

$$ExX := \begin{pmatrix} -4 \\ -3 \\ -2 \\ -1 \end{pmatrix} \qquad ExY := \begin{pmatrix} -3 \\ -1 \\ 0 \\ 7 \end{pmatrix}$$

б) так как график исходных точек представляет собой ветвь параболы, то достаточно определить аппроксимирующий полином второй степени. Для этого введём следующие обозначения:

$$n := 3 \quad i := 0 .. n \quad t0 := n + 1$$

$$t1 := \sum_i ExX_i \qquad t2 := \sum_i (ExX_i)^2$$

$$t3 := \sum_i (ExX_i)^3 \qquad t4 := \sum_i (ExX_i)^4 \qquad c0 := \sum_i (ExY_i)$$

$$c1 := \sum_i (ExX_i \cdot ExY_i) \quad c2 := \sum_i (ExX_i)^2 \cdot ExY_i$$

в) определите матрицу коэффициентов и матрицу свободных членов системы:

$$M := \begin{pmatrix} t0 & t1 & t2 \\ t1 & t2 & t3 \\ t2 & t3 & t4 \end{pmatrix} \qquad k := \begin{pmatrix} c0 \\ c1 \\ c2 \end{pmatrix}$$

г) найдите коэффициенты полинома, решив систему с помощью функции **lsolve(..)**:

$$A := \mathrm{lsolve}(M, k)$$

$$A = \begin{pmatrix} 14.75 \\ 9.35 \\ 1.25 \end{pmatrix}$$

д) определите вид полинома:

$$P(x) := \sum_{i=0}^{2} A_i x^i$$

е) укажите количество точек *n*, значения аргумента x_i и его шаг изменения *h*, для построения графиков функций:

$$n := 10 \quad i := 0..n$$

$$h := \frac{(ExX_3 - ExX_0)}{n}$$

$$x_i := ExX_0 + h \cdot i$$

ж) во всех точках x_i для сравнения вычислите значения функции с помощью встроенных средств аппроксимации:

$$VS := regress(ExX, ExY, 3)$$

$$f_i := interp(VS, ExX, ExY, x_i)$$

и) постройте графики в соответствии с рисунком 3.10.

2.2.2. В MathCad имеются средства для сплайновой интерполяции.

Интерполирование производится с помощью функции **interp**(*vs, vx, vy, x*), которая возвращает вектор значений функции *y* в интересующих точках *x*.

Здесь и далее *vx* — вектор исходных значений аргумента, *vy* — вектор исходных значений функции, *vs* — вектор вторых производных, возвращаемых одной из функций **cspline**(*vx, vy*), **pspline**(*vx, vy*), **lspline**(*vx, vy*), которые возвращают вектор *vs* при использовании кубических, параболических и линейных сплайнов соответственно.

Для интерполирования линиями можно воспользоваться функцией **linterp**(*vx, vy, x*) [7].

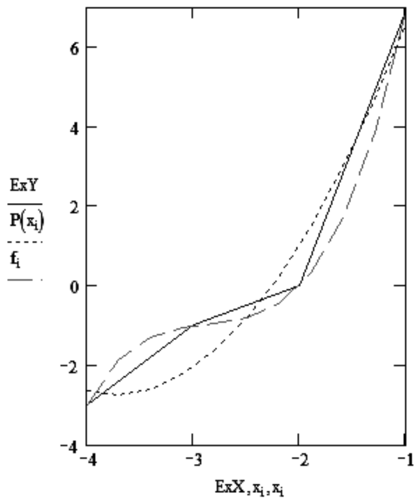


Рисунок 3.10 — Вид графиков в MathCad

Пример выполнения задания

Решите рассматриваемый пример (условие *см.* с. 46), используя средства MathCad для сплайновой интерполяции. Для этого выполните следующие операции:

а) введите исходные данные:

$$ExX := \begin{pmatrix} -4 \\ -3 \\ -2 \\ -1 \end{pmatrix} \qquad ExY := \begin{pmatrix} -3 \\ -1 \\ 0 \\ 7 \end{pmatrix}$$

б) определите количество точек n, значения аргумента x и его шаг изменения h, для построения графиков функций:

$$n := 100 \qquad i := 0..n$$

$$h := \frac{ExX_3 - ExX_0}{n}$$

$$x_i := ExX_0 + h \cdot i$$

в) во всех точках x_i вычислите значения функции с помощью встроенных средств для сплайновой интерполяции:

$$VS := \mathrm{cspline}(ExX, ExY)$$

$$y := \mathrm{interp}(vS, ExX, ExY, x)$$

$$z := \mathrm{linterp}(ExX, ExY, x)$$

г) постройте графики и выведите значения y в соответствии с рисунком 3.11.

2.3. Произведите интерполирование табличных функций методом Лагранжа и Ньютона на отрезке $[a, b]$ с шагом h средствами Delphi.

2.3.1. Расположите на форме основные компоненты (рис. 3.12) согласно рассматриваемому примеру.

y =

	0
0	-3
1	-2.023
2	-1.438
3	-1.133
4	-1
5	-0.93
6	-0.813
7	-0.539
8	0
9	0.914
10	2.313
11	4.305
12	7
13	10.508
14	14.938
15	20.398

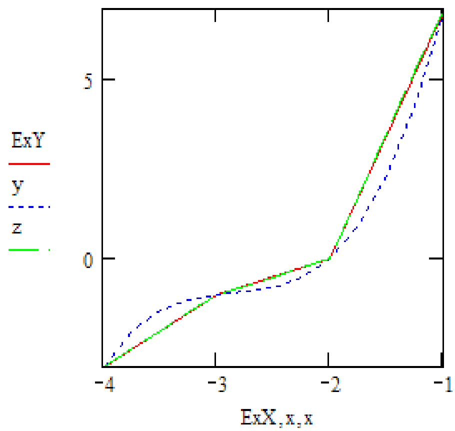


Рисунок 3.11 — Вид графиков интерполирования функций средствами MathCad

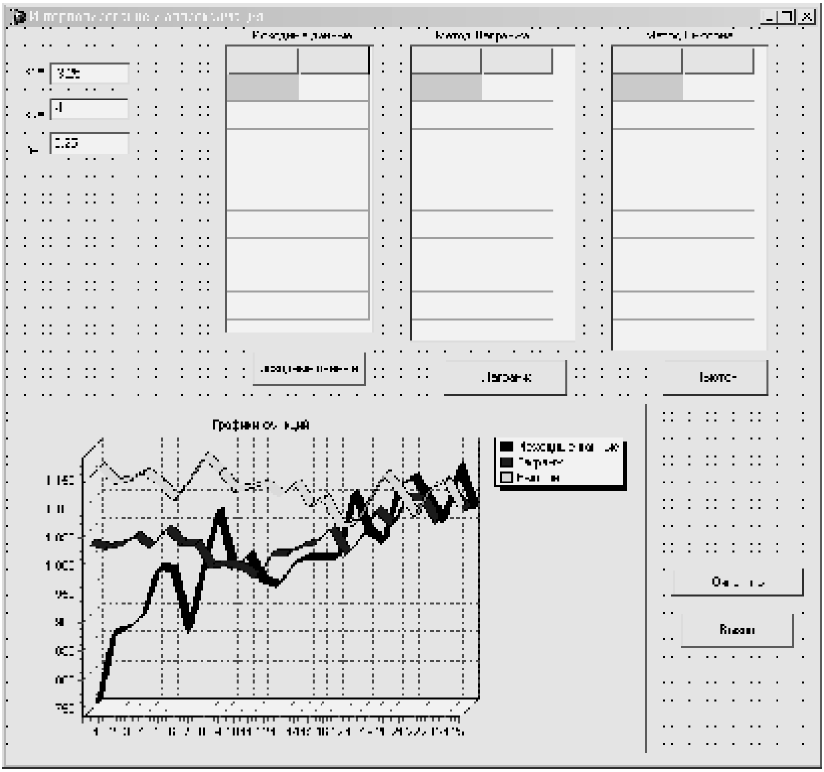


Рисунок 3.12 — Вид формы в среде Delphi

2.3.2. Создайте обработчик события нажатием кнопки **Исходные данные**, в результате чего в таблице должны появиться исходные числовые данные и на координатной плоскости нарисуется график по данным точкам:

```
procedure TForm1.Button1Click(Sender: TObject);
var x,y:array[0...3] of real;
i:integer;
begin
stringgrid1.Cells[0,0] := 'X'; stringgrid1.Cells[1,0] := 'Y';
stringgrid1.Cells[0,1] := '–4'; stringgrid1.Cells[1,1] := '–3';
```

```
stringgrid1.Cells[0,2] := '–3'; stringgrid1.Cells[1,2] := '–1';
stringgrid1.Cells[0,3] := '–2'; stringgrid1.Cells[1,3] := '0';
stringgrid1.Cells[0,4] := '–1'; stringgrid1.Cells[1,4] := '7';
for i := 0 to 3 do begin
x[i] := strtofloat(stringgrid1.Cells[0,i+1]);
y[i] := strtofloat(stringgrid1.Cells[1,i+1]);
 Series1.AddXY(x[i],y[i],'',clTeeColor)
 end;
end;
… .
```

2.3.3. Создайте процедуру-обработчик нажатием кнопки **Лагранж:**

```
procedure TForm1.Button2Click(Sender: TObject);
var x,y: array[0..3] of real;
i,n,k,j: integer;
P,xn,xk,h,a:real;
begin
 xn := strtofloat(edit1.text);
 xk := strtofloat(edit2.text);
 h := strtofloat(edit3.text);
 for i := 0 to 3 do begin
 x[i] := strtofloat(stringgrid1.Cells[0,i+1]);
 y[i] := strtofloat(stringgrid1.Cells[1,i+1]);
end;
k := 0;
while xn<=xk do begin
P := 0;
for i := 0 to 3 do begin
a := 1;
for j := 0 to 3 do
if i<>j then a := a* (xn – x[j])/(x[i] – x[j]);
P := P+a*y[i];
end;
k := k+1;
stringgrid2.RowCount := k+1;
stringgrid2.Cells[0,k] := floattostr(xn);
stringgrid2.Cells[1,k] := floattostrF(P,ffFixed,6,2);
```

```
Series2.AddXY(xn,P,'',clTeeColor);
xn := xn+h;
 end;
 end;
… .
```

2.3.4. Создайте алгоритм вычисления промежуточных значений *x* на интервале [–3,5; –1] методом Ньютона:

```
procedure TForm1.Button3Click(Sender: TObject);
 var x,y: array[0...4] of real;
i,n,k,j,k1: integer;
P,xn,xk,h,a,s1,p1,s,h1:real;
label l;
begin
 xn := strtofloat(edit1.text);
 xk := strtofloat(edit2.text);
 h := strtofloat(edit3.text);
k := 0;
 for i := 0 to 3 do begin
 x[i] := strtofloat(stringgrid1.Cells[0,i+1]);
 y[i] := strtofloat(stringgrid1.Cells[1,i+1]);
end;
while xn<=xk do begin
 s := y[0];
 for i := 1 to 3 do begin
 a := 1; p := 1; s1 := 0;
 for j := 0 to i do begin
 if i=j then goto l;
 p := p*(xn–x[j]);
 l: H1 := 1;
 for k1 := 0 to i do
 begin
 if j<>k1 then H1 := H1*(x[j]–x[k1]);
 end;
 H1 := y[j]/H1; s1 := s1+H1;
 end;
 p := p*s1;
```

```
 s := s+p;
 end;
 k := k+1;
stringgrid3.RowCount := k+1;
stringgrid3.Cells[0,k] := floattostr(xn);
stringgrid3.Cells[1,k] := floattostrF(s,ffFixed,6,2);
Series3.AddXY(xn,S,'',clTeeColor);
xn := xn+h;
end;
end;
… .
```

2.3.5. Для кнопок **Очистить** и **Выход** создайте процедуры-обработчики событий *самостоятельно*.

2.3.6. Запустите проект и протестируйте программу. Получите результаты вычисления (рис. 3.13).

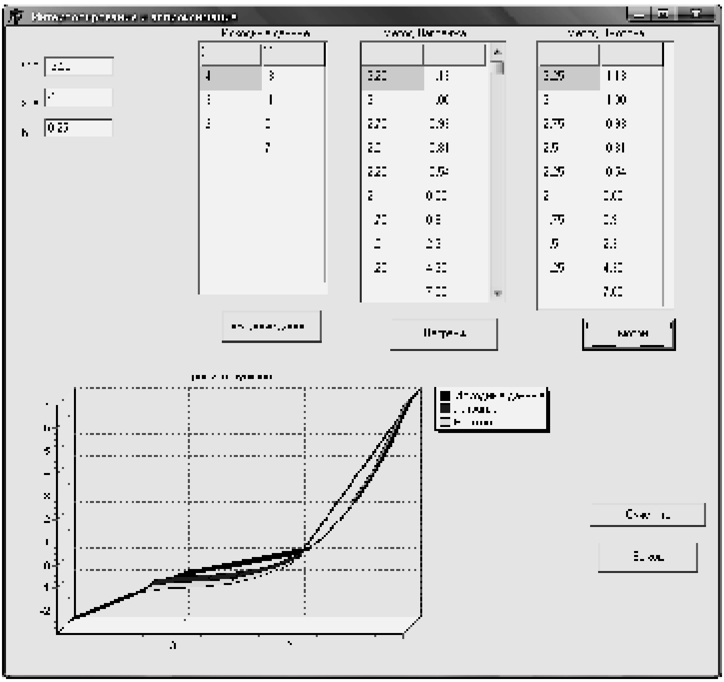


Рисунок 3.13 — Результаты решения задачи в среде Delphi

Задание 3. Произведите сравнительный анализ полученных результатов. Вычислите среднее квадратичное отклонение метода наименьших квадратов в средах Excel и MathCad. Сделайте вывод.

Контрольные вопросы

1. Аппроксимация и интерполирование функций. Постановка задачи.
2. В чём особенность аппроксимации?
3. Опишите особенности метода наименьших квадратов в случае линейной и квадратичной зависимости.
4. Какой из методов полиномиальной зависимости даёт более точный результат в вашем варианте лабораторной работы?
5. Интерполяционный многочлен Лагранжа, его преимущества и недостатки.
6. Интерполяционный многочлен Ньютона, его разновидности.
7. Оценка погрешностей методов.

Лабораторная работа 4
ЧИСЛЕННОЕ ИНТЕГРИРОВАНИЕ

Цель работы: изучить основные методы численного интегрирования; научиться создавать приложения для решения задачи численного интегрирования в среде Delphi; изучить средства пакета MathCad, предоставляемые для решения задачи нахождения определённого интеграла.

Постановка задачи. Значение определённого интеграла вычисляется по формуле Ньютона — Лейбница, если удаётся выразить первообразную формулу через элементарные функции:

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

где $F(a), F(b)$ — первообразные.

Функция $F(x)$ на отрезке $[a; b]$ называется *первообразной* функцией $f(x)$, если на всём интервале справедливо равенство $F'(x) = f(x)$ [3].

Однако для большого числа функций первообразная не выражается через элементарные функции, либо её сложно определить. Также подынтегральная функция $f(x)$ может быть задана графически или в табличной форме. В таких случаях для вычисления определённого интеграла используют приближённые численные методы.

Суть численных методов вычисления определённого интеграла состоит в замене подынтегральной функции $f(x)$ вспомогательной функцией, интеграл от которой легко вычисляется в элементарных функциях. Наиболее часто подынтегральную функцию $f(x)$ заменяют некоторым интерполяционным многочленом. Это приводит к использованию квадратурной формулы

$$\int_a^b f(x)dx = \sum_{i=0}^{n} A_i f(x_i) + R,$$

где x_i — узлы интерполяции;

R — остаточный член (погрешность метода).

В случае отбрасывания R возникает погрешность усечения в процессе вычисления, появляется погрешность округления.

С геометрической точки зрения значение определённого интеграла представляет собой площадь, ограниченную графиком функции $y = f(x)$, осью абсцисс и прямыми $x = a$, $x = b$.

Отрезок интегрирования разбивают на n интервалов x_i, x_{i+1}, где $i = 0, 1, ..., n$. Приближённо определяются значения площадей, соответствующих каждому отрезку. Их сумма даст приблизительное значение интеграла. В зависимости от способа разбиения отрезка узлами интерполяции x_i различают два подхода к построению квадратурных формул:

1. Место положения и длина интервалов разбиения выбираются заранее в начале расчёта. В случае равноотстоящих точек узлов интерполирования $x_i = x_0 + ih,\quad h = \frac{b-a}{n}$. В таком случае квадратурные формулы называются формулами Ньютона — Котеса и различаются степенями используемых итерационных многочленов. Широкое применение находят формулы трапеции и Симпсона.

2. Место положения и длина интервалов подбираются таким образом, чтобы достичь наибольшей точности (формула Гаусса).

Формула трапеций. Отрезок интегрирования $[a;b]$ разбивают на n равных интервалов длиной $h = \frac{b-a}{n}$. В пределах каждого отрезка [x_i; x_{i+1}] функция *f(x)* заменяется интерполяционным многочленом Лагранжа первой степени. Такая замена соответствует замене кривой на секущую (рис. 4.1). Значение примет вид

$$\int_{x_i}^{x_{i+1}} f(x)dx \approx h\left(\frac{f(x_i)+f(x_{i+1})}{2}\right).$$

Суммирование значений интеграла по всем *n* участкам разбиения даёт общую площадь

$$S = h\left(\frac{f(a)+f(b)}{2} + f(x_1) + f(x_2) + \ldots + f(x_{n-1})\right). \qquad (4.1)$$

Если функция *f(x)* на отрезке [*a;b*] непрерывна и имеет вторую производную, то оценка погрешности усечения находится по формуле

$$|R| \le \frac{(b-a)h^2}{12}M, \; M = \max\left[f''(x)\right], a \le x \le b, \qquad (4.2)$$

где *h* — шаг интегрирования;

M — коэффициент, который получается из набора вторых производных для отрезка [*a; b*].

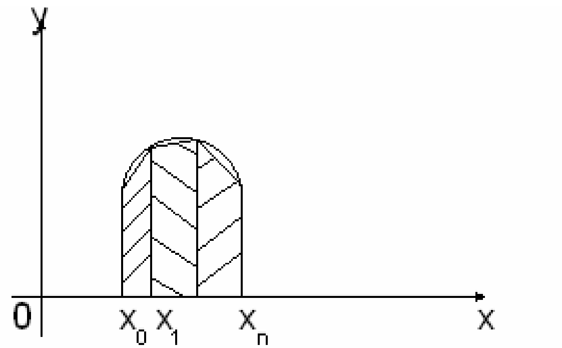


Рисунок 4.1 — Геометрический смысл метода трапеций

Очевидно, что формула трапеции (4.1) даст точное значение интеграла для линейной подынтегральной функции *f(x)* [5].

Формула Симпсона. При замене подынтегральной функции интерполяционным многочленом второй степени и при чётном числе *n* интервалов разбиения квадратурная формула преобразуется к виду

$$\int_a^b f(x)dx \approx \frac{h}{3}\begin{bmatrix} f(a)+4\ f(x_1)+2\ f(x_2)+\ldots \\ \ldots+2f(x_{n-2})+4\ f(x_{n-1})+f(b) \end{bmatrix}. \quad (4.3)$$

Это формула Симпсона либо формула параболы.

Значение функции *f(x)* в нечётных точках разбиения x_1, x_3 и т. д. входят в формулу (4.3) с коэффициентом 4, в чётных точках x_2, x_4 и т. д. — с коэффициентом 2, в граничных точках x_0 и x_n — с коэффициентом 1.

Геометрический смысл формулы Симпсона. Через три последовательные ординаты разбиения проводят параболу и определяют площадь полученной фигуры. Сумма всех фигур, построенных таким образом, даёт примерное значение интеграла (рис. 4.2).

При наличии на отрезке [*a;b*] непрерывной четвёртой производной подынтегральной функции оценка усечения формулы Симпсона следующая:

$$R \le \frac{(b-a)\,h^4}{180} M, \quad M = \mathbf{max}(f''''(x)), a \le x \le b. \quad (4.4)$$

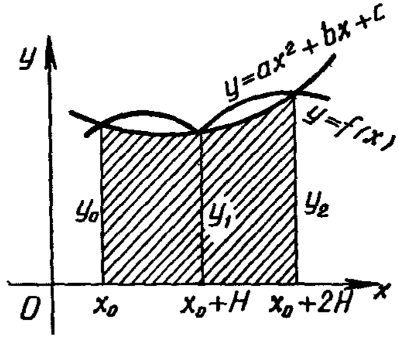


Рисунок 4.2 — Геометрический смысл метода Симпсона

Формула парабол является точной для полиномов третьей степени включительно, так как для них четвёртая производная равна 0. Из-за сложности, а часто и невозможности вычисления четвёртой производной, используют формулу Рунге:

$$|R| = \frac{S_{2n} - S_n}{15},$$

где S_n — значение интеграла при разбиении отрезка интегрирования на n участков (n — чётное);

S_{2n} — значение интеграла при разбиении отрезка интегрирования на $2n$ участков [6].

Сравнение формул интегрирования. Для функции высокой гладкости, т. е. имеющей непрерывные производные достаточно высокого порядка при одинаковом числе узлов, формула Симпсона более точна, чем формула трапеций. При этом для получения одной и той же точности по формуле Симпсона необходимо выполнить меньше операций, чем по формуле трапеций. Если подынтегральная функция задана таблично, то формула Симпсона в достаточной степени точна для умеренного числа узлов интегрирования. Она удобна для программирования на ЭВМ и поэтому получила широкое применение в практических расчётах.

Задания

Задание 1. Разработайте схемы алгоритмов интегрирования функций по методам трапеций и Симпсона.

Задание 2. В среде Delphi создайте приложение для решения задачи численного интегрирования функций, приведённых в таблице 4.1 (в соответствии со своим вариантом), методами трапеций и Симпсона. Предусмотрите в программе вычисление точного значения определённого интеграла через первообразную функцию (*см.* табл. 4.1).

Таблица 4.1 — Варианты заданий

Вари-ант	Подынтегральная функция *f(x)*	Интервал интегрирования [*A*,*B*]	Количество частей разбиения *n*	Шаг *H*	Первообразная функция
1-й	$xe^{x}\sin(x)$	[0; 1]	50	0,0200	$\frac{xe^{x}(\sin x-\cos x)}{2}+\frac{e^{x}\cos x-1}{2}$
2-й	$\frac{1}{\sqrt{9+x^{2}}}$	[0; 2]	200	0,0100	$\ln\left(x+\sqrt{x^{2}}+9\right)-\ln 3$
3-й	$\frac{1}{\sqrt{1+3x+2x^{2}}}$	[0; 1]	40	0,0250	$\frac{1}{\sqrt{2}}\ln\frac{x+0{,}75+\sqrt{(x+0{,}75)^{2}}-0{,}0625}{0{,}75+\sqrt{0{,}5}}$
4-й	$\left(\frac{\ln x}{x}\right)^{3}$	[1; 2]	40	0,0250	$-\frac{(\ln x)^{3}+3(\ln x)^{2}/2+3(\ln x)/2+3/4}{2x^{2}}+\frac{3}{8}$
5-й	$\frac{x^{3}}{3+x}$	[1; 2]	80	0,0125	$9x-\frac{3x^{2}}{2}+\frac{x^{3}}{3}-27\ln(3+x)-\frac{47}{6}+27\ln 4$
6-й	$x\cdot\mathrm{arctg}x$	[0; 2]	50	0,0400	$\frac{x^{2}}{2}\mathrm{arctg}x-\frac{x}{2}+\frac{1}{2}\mathrm{arctg}x$
7-й	$x^{x}(1+\ln x)$	[1; 3]	40	0,0600	x^{x}
8-й	$\frac{1}{\sqrt{1+3x}}$	[0; 1]	40	0,0250	$\frac{2}{3}\sqrt{1+3x}$
9-й	$\sin^{2}x$	[0; $\frac{\pi}{2}$]	60	0,0260	$\frac{x}{2}-\frac{\sin 2x}{4}$
10-й	$\frac{1}{x\lg x}$	[2; 3]	40	0,0250	$2{,}3026(\ln\ln x-\ln\ln 2)$
11-й	$\frac{1}{x^{2}}\sin\frac{1}{x}$	[1; 2,5]	50	0,0300	$\cos\left(\frac{1}{x}\right)$
12-й	$\cos x$	[0; π]	60	0,0260	$\sin x$

Пример выполнения задания

Определите интеграл $\int\limits_{1,0}^{2,6}\left(x^{3}+\frac{1}{x^{2}}\right)dx$ с $n = 40$ и шагом *h*, который вычисляется по формуле *h = (b – a) / n,* используя квадратурные формулы: а) трапеций; б) Симпсона; в) Ньютона — Лейбница. Вид первообразной $F(x) = \frac{x^4}{4} - \frac{1}{x}$. Оцените погрешность всех результатов.

2.1. Создайте приложение в Delphi и разместите необходимые компоненты на форме (рис. 4.3). Для этого используйте компоненты: Label, Edit, Memo, RadioButton, Chart, Button. Введите исходные данные в Edit1 — Edit4, Label1-Label4.

2.2. В начале программы необходимо описать исходную функцию *(f)* и её первообразную *(perv):*

```
function f(x:real):real;
begin
f := exp(3*ln(x))+1/sqr(x);
end;
… .

function perv(x:extended):extended;
begin
perv := exp(4*ln(x))/4–1/x;
end;
```

2.3. Создайте процедуру-обработчик нажатием кнопки **Решить.** Текст процедуры приведите к следующему виду:

```
procedure TForm1.Button1Click(Sender: TObject);
var a,b,h,s,x,s1:real;
j,n,i:integer;
begin
a := StrToFloat(Edit1.Text);
b := StrToFloat(Edit2.Text);
n := StrToint (Edit4.Text);
h := (b–a)/n;
Edit3.Text := FloatToStr(h);
Series1.Clear;
```

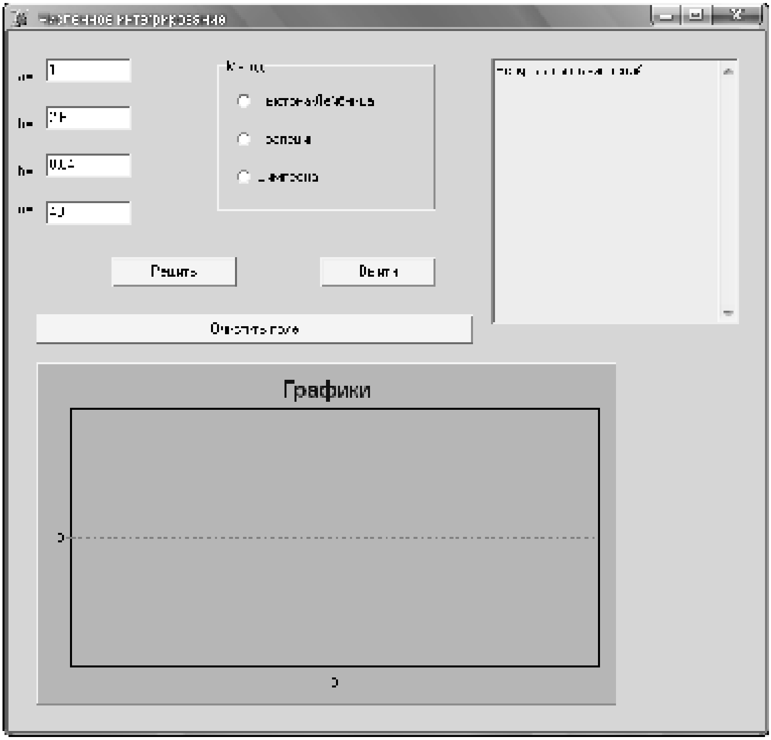


Рисунок 4.3 — Вид формы решения интеграла в Delphi

```
//Метод Ньютона — Лейбница
s1 := perv(b)–perv(a);
x := a;
if RadioButton1.Checked then begin
while x<=b do begin
Series1.AddXY(x,f(x));
x := x+h;
end;
Memo1.Lines.Add('По методу Ньютона — Лейбница ='+FloatToStrF(s1,fffixed,14,7));
end;
```

```
//Метод трапеций
if RadioButton2.Checked then begin
s := (f(a)+f(b))/2;
h := (b–a)/n;
for i := 1 to n–1 do begin
x := a+i*h;
series1.AddXY(x,f(x));
s := s+f(x);
end;
s := s*h;
Memo1.Lines.Add('По методу трапеций='+FloatToStrF(s,fffixed,14,7));
Memo1.Lines.Add('Погрешность='+FloatToStrF(abs(s1–s),fffixed,14,7));
end;

//Метод Симпсона
if RadioButton3.Checked then begin
s := f(a)+f(b);
for j := 1 to n-1 do
begin
x := a+j*h;
if odd(j)=true then s := s+4*f(x)
else s := s+2*f(x);
Series1.AddXY(x,f(x));
end;
s := s*h/3;
Memo1.Lines.Add('По методу Симпсона='+FloatToStrf(s,fffixed,14,7));
Memo1.Lines.Add('Погрешность='+FloatToStrF(abs(s1–s),fffixed,14,7));
end;
end;
… .
```

2.4. В результате при выборе одной из радиокнопок будет строиться график и выводиться решение задачи по выбранному методу (рис. 4.4). Обратите внимание, что полученные результаты близки по значениям, что и подтверждают погрешности вычислений.

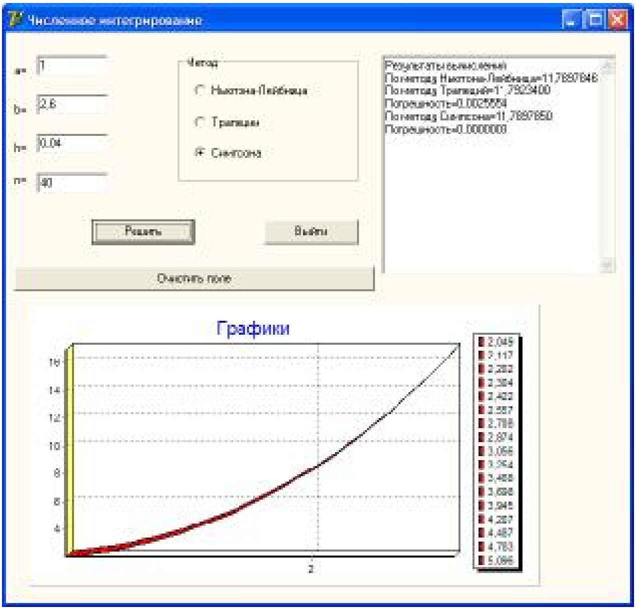


Рисунок 4.4 — Значения интеграла по методам Симпсона, трапеций, Ньютона — Лейбница

2.5. Самостоятельно создайте процедуры-обработчики нажатием кнопок **Выйти** и **Очистить поле**. Продумайте, как вычислить погрешности методов трапеций и Симпсона по формулам (4.2) и (4.4). Реализуйте это в своем приложении.

Задание 3. Проведите интегрирование тех же функций (*см.* табл. 4.1) средствами пакета MathCad и Excel.

Пример выполнения задания

3.1. Вычислите указанный интеграл $\int_{1,0}^{2,6}\left(x^3+\frac{1}{x^2}\right)dx$ с $n = 40$ и шагом h в электронных таблицах Excel с помощью методов трапеций и Симпсона.

3.1.1. Введите исходные данные и формулы, как показано на рисунке 4.5.

3.1.2. Скопируйте формулы A8:D8 на блок A9:D47. Затем в ячейку D47 введите формулу =C7+C47, а в ячейку D48 — формулу =СУММ(D8:D47). Только после этого можно вводить формулы вычисления по методам трапеций и Симпсона.

3.1.3. В результате получите числовые значения, которые отображены на рисунке 4.6.

3.1.4. Таким образом, значение интеграла по формуле трапеций — $S_{тр}$=11,79234003, значение интеграла по формуле Симпсона — S_c=11,78978495.

	A	B	C	D
1	Исходные данные			S_Трапеций
2	a=	1		=B5*(C7/2+C47/2+СУММ(C8:C46))
3	b=	2,6		
4	n=	40		S_Симпсона
5	h=	=(B3-B2)/B4		=B5/3*(D48)
6	№	x	f(x)	
7	0	=B2	=B7^3+1/B7^2	
8	=A7+1	=B7+$B$5	=B8^3+1/B8^2	=ЕСЛИ(ЧЁТН(A8)=A8;2*C8;4*C8)

Рисунок 4.5 — Вычисления значения интеграла в режиме формул

	A	B	C	D
1	Исходные данные			S_Трапеций
2	a=	1		11,79234003
3	b=	2,6		
4	n=	40		S_Симпсона
5	h=	0,04		11,78978495
6	№	x	f(x)	
7	0	1	2	
8	1	1,04	2,04942	8,197680852
9	2	1,08	2,11705	4,234101641
10	3	1,12	2,20212	8,80848751
11	4	1,16	2,30406	4,608117803
12	5	1,2	2,42244	9,689777778

Рисунок 4.6 — Значения интеграла
по формулам Симпсона и трапеций

3.2. Найдите значение заданного интеграла при помощи пакета MathCad по формулам трапеций, Симпсона и прямым способом.

3.2.1. Для этого введите исходные данные, которые будете использовать для всех рассматриваемых методов:

$$\mathrm{F(x)} := x^3 + \frac{1}{x^2} \qquad a := 1 \qquad b := 2.6$$

3.2.2. Вызовите с панели инструментов (или с помощью комбинации клавиш <**Shift**>**+**<**F7**>) значок определенного интеграла, заполните необходимые значения и нажмите знак равенства.

Получите прямой способ нахождения интеграла:

$$\int_a^b F(x)dx = 11.789785$$

3.2.3. Реализуйте метод трапеций в MathCad, выполнив следующие действия:

$$n := 40 \qquad h := \frac{b-a}{n} \qquad h = 0{,}04$$

$$i := 0..n$$

$$x_i := a + h \cdot i$$

$$S := \frac{F(x_0) + F(x_n)}{2} \cdot h + h \cdot \sum_{k=1}^{n-1} F(x_k)$$

$$S = 11.79234$$

3.2.3. Вычислите указанный интеграл, используя формулу Симпсона:

$$S1 := \frac{2}{3}h\left[\frac{F(a)+F(b)}{2} + 2\sum_{i=1}^{\frac{n}{2}} F[a+(2i-1)h] + \sum_{i=1}^{\frac{n}{2}-1} F(a+2\cdot i\cdot h)\right]$$

$$S1 = 11.789785$$

Задание 4. Вычислите абсолютные погрешности методов интегрирования функций по формуле

$$\delta = | I^* - I |,$$

где I^* — точное значение интеграла, вычисленное через первообразную;

I — значение интеграла, полученное в результате применения конкретного численного метода.

Задание 5. На основании результатов, полученных в заданиях 2 и 3, проведите сравнительный анализ методов численного интегрирования.

Контрольные вопросы

1. Назовите сущность и отличия известных вам методов численного интегрирования.
2. Поясните геометрический смысл определённого интеграла.
3. Поясните алгоритм вычисления определённого интеграла методами трапеций и Симпсона.
4. Как оценить точность вычисления определённого интеграла?
5. Каковы преимущества формулы парабол по сравнению с формулой трапеций и следствием чего являются эти преимущества?
6. В каких случаях приближённые формулы трапеций и парабол оказываются точными?
7. Как влияет на точность численного интегрирования величина шага?

Лабораторная работа 5
ЧИСЛЕННОЕ РЕШЕНИЕ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ

Цель работы: изучить наиболее распространённые численные методы решения обыкновенных дифференциальных уравнений; научиться создавать приложения для решения обыкновенных дифференциальных уравнений средствами Delphi; изучить основные средства, предоставляемые пакетом MathCad для решения дифференциальных уравнений (ДУ).

Постановка задачи. Обыкновенные дифференциальные уравнения (ОДУ) или их системы часто используют для построения математических моделей динамических процессов.

Динамические процессы — процессы перехода физических систем из одного состояния в бесконечно близкое другое. Примерами могут служить явления, возникающие в электрических сетях, распространение радиоволн, движение материальных точек, изменение химического состояния вещества и т. д. [4].

Точные методы решения ДУ позволяют выразить решение через элементарные или специальные функции. Однако классы таких уравнений достаточно немногочисленны и охватывают малую часть возникающих на практике задач. В силу этого важное значение приобретают приближённые численные методы решения ДУ [5].

Пусть требуется решить задачу Коши:

$y'(x) = f(x,y)$,
$y0 = y(x_0)$ — начальное условие,
$[x_0, x_n]$ — отрезок,
$y^{(n)} = f(x, y, y', y^n, y^{(n-1)})$.

Для решения ДУ n-го порядка используют метод понижения производной, т. е. уравнение n-го порядка сводят к системе n уравнений n-го порядка способом замены переменных следующим образом:

$$y' = y_1, y'' = y_2, y^{(n-1)} = y_{n-1}$$

$$\begin{cases} y' = y_1; \\ y_1' = y_2; \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots \\ y'_{n-1} = f(x, y, y_1, y_2, ..., y_n). \end{cases}$$

Методы решения одного уравнения первого порядка распространяются на систему уравнений первого порядка.

Сущность численных методов решения ДУ. На отрезке $[x_0, x_n]$ выбирается некоторое множество точек, которое называется *сеткой* $(x_0 < x_1 < x_2 < \dots < x_n)$. В полученных точках вычисляются приближенные значения $y_1, y_2, \dots, y_n$. Эти решения и будут являться решением задачи Коши. Величина $h = \frac{x_n - x_0}{n}$ называется шагом сетки. В большинстве случаев $h = const$, $x_k = x_0 + kh$.

Проблема численного решения ДУ заключается в построении алгоритма вычисления. Различают *одношаговые* и *многошаговые* методы решения ДУ. Метод Эйлера и метод Рунге — Кутта относятся к одношаговым. *Многошаговыми* методами являются методы прогноза и коррекции [6]. Для метода Адамса необходимо знать две предыдущие точки, для метода Милна — три точки. Их преимущество в том, что для них существует формула оценки погрешности.

Метод Эйлера. Методы группы методов Рунге — Кутта отличаются друг от друга объёмом производимых вычислений и получаемой при этом точностью. Метод Эйлера является методом Рунге — Кутта первого порядка точности. *Формула для вычисления методом Эйлера* имеет следующий вид:

$$y_{k+1} = y_k + h\ f(x_k, y_k).$$

Метод Эйлера является наиболее простым, но и наименее точным. Он применяется для получения оценочных решений на небольшом интервале $[x_0;\ x_n]$.

Метод Рунге — Кутта четвёртого порядка точности. Смещение из точки $[x_k;\ y_k]$ в точку $\left[x_{k+1}; y_{k+1}\right]$ происходит не сразу, а через промежуточные точки. На практике наибольшее распространение получил метод четвёртого порядка точности. Значение функции в $i+1$-й точке вычисляется следующим образом:

$$y_{i+1} = y_i + \Delta y_i,$$

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

где $k_1 = hf(x_i, y_i)$,

$$k_2 = hf(x_i + \frac{h}{2}, y_i + \frac{k_1}{2}),$$

$$k_3 = hf(x_i + \frac{h}{2}, y_i + \frac{k_2}{2}),$$

$$k_4 = hf(x_i + h, y_i + k_3).$$

Метод Рунге — Кутта обладает достаточно высокой точностью, легко программируется, так как для вычисления y_{i+1} нужно знать лишь одно значение y_i. С помощью этого метода можно начинать решение ДУ. Величина шага изменения аргумента *x* легко меняется на любом этапе вычисления [3].

Недостатки:

1) необходимость четыре раза вычислять значение функции на каждом шаге;

2) отсутствие легко определяемой оценки ошибки метода.

Для оценки правильности шага рассчитывают $\delta = \left|\frac{k_2 - k_3}{k_1 - k_2}\right|$, которое должно быть меньше либо равно 0,05. В противном случае шаг следует уменьшить в два раза, провести вычисления и снова оценить δ [8].

Метод Рунге — Кутта имеет порядок точности, сопоставимый со значением шага, взятым в четвёртой степени. Для оценки погрешности метода пользуются формулами Рунге: $\varepsilon \sim h^4$, $\varepsilon = \frac{1}{15}(y_{2h} - y_h)$.

Задания

Задание 1. Разработайте схемы алгоритмов решения ОДУ методами Эйлера и Рунге — Кутта четвёртого порядка точности.

Задание 2. В среде Delphi создайте приложение для решения дифференциальных уравнений, приведённых в таблице 5.1 (в соответствии со своим вариантом) методами Эйлера и Рунге — Кутта четвёртого порядка.

Таблица 5.1 — Варианты заданий

Вариант	Уравнение	Начальные данные	Отрезок интегрирования
1-й	$y' = \frac{y}{x} + x^2$	$y(1) = 1$	$x \in [1; 4]$
2-й	$\sqrt{1-x^2}\,y' + \sqrt{1-y^2} = 0$	$y(0) = 0$	$x \in [0; 0{,}8]$

Окончание табл. 5.1

Вариант	Уравнение	Начальные данные	Отрезок интегрирования
3-й	$y' = \frac{2xy}{x^2 + y^2}$	$y(1) = 2$	$x \in [1;\ 3]$
4-й	$y' = -(x^2 + 1)\sqrt{\lvert 1 - y^2 \rvert}$	$y(0) = 0{,}95$	$x \in [0;\ 3]$
5-й	$y' = 2xy - x^3 + x$	$y(0) = 0{,}8$	$x \in [0;\ 5]$
6-й	$x' + 3x = e^{2t}$	$x(0) = -3$	$t \in [-1; 3]$
7-й	$y' = y^2 + \frac{1}{2x^2}$	$y(1) = 0{,}9$	$x \in [1; 1{,}8]$
8-й	$y' = \frac{y}{x + y^3}$	$y(1) = 1{,}5$	$x \in [2;\ 4]$
9-й	$y' = xy^3 + x^2$	$y(0) = 0$	$x \in [-1;\ 1]$
10-й	$y' = x^2 + xy^3$	$y(0) = 0$	$x \in [-2;\ -1]$
11-й	$y' = x^2 + xy + y^2$	$y(1) = 0$	$x \in [0;\ 3]$
12-й	$y' = \frac{1}{3x^2 + y^2}$	$y(1) = 2$	$x \in [1;\ 3]$

Пример выполнения задания

Решите уравнение $y' = \sin xy$ на отрезке [0, π]. Известно, что $y(0)=1$.

2.1. Создайте в среде Delphi форму и расположите на ней необходимые компоненты. Примерный вид представлен на рисунке 5.1.

2.2. В начале программы введите вид функции:

```
function f(x,y:real):real;
 begin
 f := sin(x)*y;
 end; .
```

2.3. Нажатием кнопки **Вычислить**, создайте процедуру-обработчик которая будет иметь вид:

```
procedure TForm1.Button1Click(Sender: TObject);
label Rk ;
var xn,x0,y0,h,x,y,k1,k2,k3,k4,delta,g: extended;
```

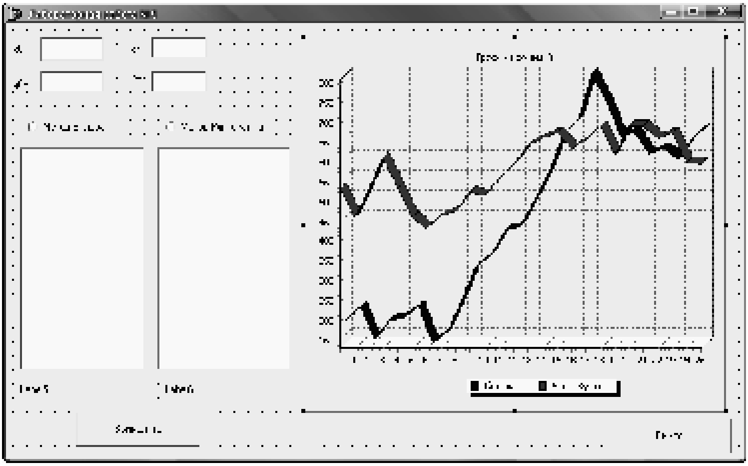


Рисунок 5.1 — Вид формы в среде Delphi

```
n:integer;
begin
if radioButton1.Checked then begin
listbox1.Items.Clear;
x0 := strToFloat(edit1.Text);
y0 := strToFloat(edit2.Text);
xn := strToFloat(edit3.Text);
n := strToint(edit4.Text);
h := (xn–x0)/n;
Series1.Clear;
x := x0;
y := y0;
while x<=xn do begin
y := y+h*f(x,y); x := x+h;
listbox1.Items.Add('x='+floattostrf(x,fffixed,5,4)+'   y='+floattostrf(y,fffixed,5,4));
Series1.AddXY(x,y);
end;
label5.Caption := 'Погрешность='+floattostrf(sqr(h),fffixed,1,7);
end;
```

```
if radiobutton2.Checked then begin
 listbox2.Items.Clear;
x0 := strToFloat(edit1.Text);
y0 := strToFloat(edit2.Text);
xn := strToFloat(edit3.Text);
n := strToint(edit4.Text);
h := (xn–x0)/n;
Series2.Clear;
x := x0;
y := y0;
while x<=xn do begin
Rk:
k1 := h*f(x,y);
k2 := h*f((x+h/2),(y+k1/2));
k3 := h*f((x+h/2),(y+k2/2));
k4 := h*f((x+h),(y+k3));
g := abs((k2–k3)/(k1–k2));
if g>0.05 then begin h := h/2;
goto Rk;
end;
delta := 1/6*(k1+2*k2+2*k3+k4);
listbox2.Items.Add('x='+floattostrf(x,fffixed,5,4)+'  y='+floattostrf(y,fffixed,5,4));
Series2.AddXY(x,y); x := x+h; y := y+delta;
 end;
 label6.Caption:='Погрешность='+ Floattostrf(sqr(sqr(h)),fffixed,1,7);
 end;
 end;
… .
```

2.4. Получите результаты, показанные на рисунке 5.2.

Задание 3. Решите дифференциальное уравнение (в соответствии со своим вариантом) с помощью MathCad, используя встроенные функции и методы Эйлера и Рунге — Кутта четвёртого порядка.

3.1. Для решения линейных обыкновенных дифференциальных уравнений в MathCad функция **odesolve**(*x*,*b*,[*step*]), которая возвращает

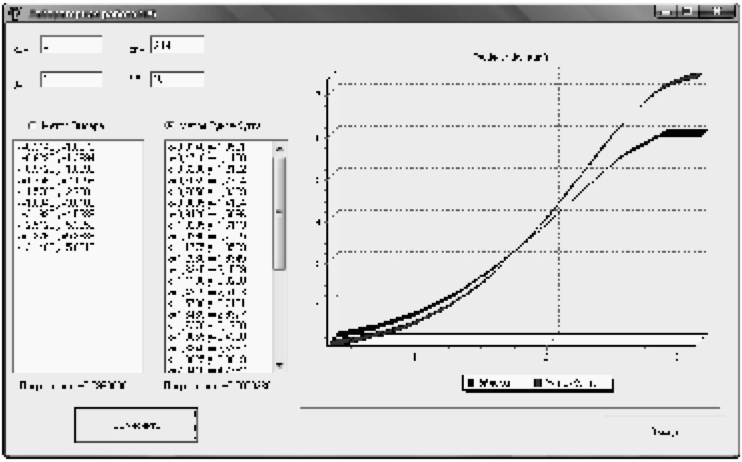


Рисунок 5.2 — Результаты решения дифференциального уравнения в Delphi

значение функции, зависящей от x и являющейся решением линейного ОДУ. Здесь x — аргумент, b — правый конец отрезка, [*step*] — необязательный параметр, количество шагов для нахождения решения. Количество начальных условий должно равняться порядку уравнения [7].

Пример выполнения задания

Решите уравнение $y'' - y'\sin x + y = \dfrac{x}{2\pi}$ на отрезке [0, 4π]. Известно, что $y(0) = 0$, а $y'(0) = 1$.

Решение представлено на рисунке 5.3.

3.2. Решение ОДУ первого порядка вида $y' = f(x, y)$ получите с помощью функций **rkfixed**(y_0,a,b,n,D), которая возвращает матрицу, состоящую из двух столбцов. В первом столбце хранятся значения аргумента, во втором — функции (результаты решения). Здесь y_0 — начальное значение функции y, a — начало отрезка, b — конец отрезка, n — количество отрезков разбиения, D — первая производная от y [8].

Given

$$y''(x) - \sin(x)\cdot y'(x) + y(x) = \frac{x}{2\cdot\pi}$$

$$y(0) = 0 \qquad y'(0) = 1$$

$$y := \mathrm{odesolve}(x, 4\cdot\pi)$$

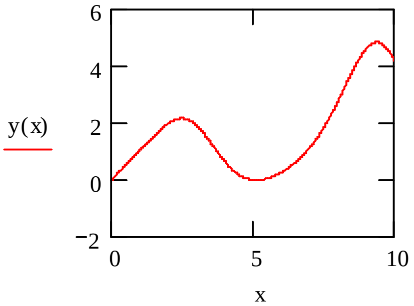


Рисунок 5.3 — Решение уравнения

Примечание. В блоке Given используются знаки булева равенства, вставляемые нажатием клавиш <**Ctrl**>+<=>, и знак производной (штрих), вставляемый нажатием клавиш <**Ctrl**>+<**F7**>.

Пример выполнения задания

Решите уравнение $y' = \sin xy$ на отрезке [0, π]. Известно, что $y(0) = 1$.

Решение представлено на рисунке 5.4.

Примечание. Для решения ОДУ первого порядка так же можно использовать функцию **rkadapt**(), аналогичную рассмотренной выше **rkfixed**(), за исключением того, что решение находится не с фиксированным шагом, а с автоматическим его подбором.

3.3. Решение дифференциального уравнения $y'=f(x,y)$ методом Эйлера.

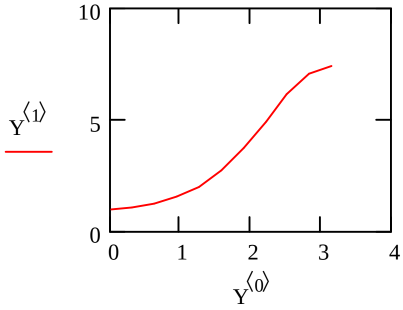
Пример выполнения задания

Решите дифференциальное уравнение $y' = f(x, y)$ методом Эйлера на отрезке $[a;b]$ с шагом h с начальным условием $y(a)=y_0$, где $f(x,y) = (3x - y)/(x^2 + y)$, $a = 2$, $b = 3$, $h = 0{,}1$, $y_0 = 1$.

Решение представлено на рисунке 5.5.

$y_0 := 1 \qquad D(x, y) := \sin(x)\, y_0$

$Y := \mathrm{rkfixed}(y_0, 0, \pi, 10, D)$



$Y =$

	0	1
0	0	1
1	0.314	1.05
2	0.628	1.21
3	0.942	1.51
4	1.257	1.996
5	1.571	2.718
6	1.885	3.702
7	2.199	4.893
8	2.513	6.104
9	2.827	7.036
10	3.142	7.388

Рисунок 5.4 — Решение уравнения

$a := 2 \qquad b := 3 \qquad x_0 := a \qquad y_0 := 1 \qquad h := 0.1$

$f(x, y) := \frac{3 \cdot x - y}{x^2 + y} \qquad i := 0..10$

$x_{i+1} := x_0 + i \cdot h$

$y_{i+1} := y_i + h \cdot \left[\frac{3 \cdot x_i - y_i}{(x_i)^2 + y_i}\right]$

$x =$

	0
0	2
1	2
2	2.1
3	2.2
4	2.3
5	2.4
6	2.5
7	2.6
8	2.7
9	2.8
10	2.9
11	3

$y =$

	0
0	1
1	1.1
2	1.196
3	1.287
4	1.374
5	1.457
6	1.536
7	1.613
8	1.687
9	1.758
10	1.827
11	1.895

Рисунок 5.5 — Решение ОДУ методом Эйлера
в среде MathCad

3.4. Решение дифференциального уравнения *y'=f(x, y)* методом Рунге — Кутта четвёртого порядка.

Пример выполнения задания

Решите дифференциальное уравнение *y'=f(x, y)* методом Рунге — Кутта четвёртого порядка на отрезке $[a, b]$ с шагом h с начальным условием $y(a) = y_0$, где $f(x, y) = (3x - y) / (x^2 + y)$, $a = 2$, $b = 3$, $h = 0{,}1$, $y_0 = 1$.

Решение представлено на рисунке 5.6.

$$a := 2 \qquad b := 3 \qquad x_0 := a \qquad y_0 := 1 \qquad h := 0.1$$

$$f(x,y) := \frac{3 \cdot x - y}{x^2 + y} \qquad i := 0..10$$

$$x_{i+1} := x_0 + i \cdot h \qquad y_{i+1} := y_i + h \cdot f(x_i, y_i)$$

$$k1_i := h \cdot f(x_i, y_i)$$

$$k2_i := h \cdot f\left[\left(x_i + \frac{h}{2}\right), y_i + \frac{k1_i}{2}\right]$$

$$k3_i := h \cdot f\left[\left(x_i + \frac{h}{2}\right), y_i + \frac{k2_i}{2}\right]$$

$$k4_i := h \cdot f\left[(x_i + h), y_i + k3_i\right]$$

$$y_{i+1} := y_i + \frac{(h \cdot f(x_i, y_i)) + 2 \cdot h \cdot f\left[\left(x_i + \frac{h}{2}\right), y_i + \frac{k1_i}{2}\right] + 2 \cdot k3_i + k4_i}{6}$$

y =

	0
0	1
1	1.097
2	1.191
3	1.28
4	1.363
5	1.441
6	1.512
7	1.576
8	1.634
9	1.685
10	1.73
11	1.77

x =

	0
0	2
1	2
2	2.1
3	2.3
4	2.6
5	3
6	3.5
7	4.1
8	4.8
9	5.6
10	6.5
11	7.5

Рисунок 5.6 — Решение ОДУ методом Рунге — Кутта четвёртого порядка

Задание 4. Вычислите погрешности методов решения дифференциальных уравнений.

Задание 5. На основании результатов заданий 2—4 провести сравнительный анализ методов численного решения дифференциальных уравнений.

Контрольные вопросы

1. Что является решением дифференциального уравнения?
2. Почему для решения дифференциального уравнения необходимо иметь начальные условия?
3. Зачем дифференциальное уравнение преобразуют к виду $y' = f(x, y)$?
4. Почему метод Рунге — Кутта четвёртого порядка точнее метода Эйлера?
5. За счёт чего возникает погрешность в методе Эйлера? Как её уменьшить?
6. Как выбирается шаг интегрирования в методе Рунге — Кутта четвёртого порядка точности?

ПРИЛОЖЕНИЕ А

КОМАНДЫ МЕНЮ И ПАНЕЛИ ИНСТРУМЕНТОВ СРЕДЫ MathCad

Таблица А.1 — Команды меню

Меню	Команда	Перевод	Сочетание клавиш	Описание
File (Файл)	New	Создать	<Ctrl>+<N>	Создать новый документ
	Open	Открыть	<Ctrl>+<O>	Открыть существующий документ
	Close	Закрыть	<Ctrl>+<W>	Закрыть активный документ
	Save	Сохранить	<Ctrl>+<S>	Сохранить активный документ
	Save As	Сохранить как		Сохранить активный документ в другом файле
	Send	Отправить		Отправить активный документ по электронной почте
	Page Setup	Параметры страницы		Опции вывода активного документа на печать
	Print Preview	Просмотр		Предварительный просмотр на экране вывода на печать активного документа
	Print	Печать	<Ctrl>+<P>	Распечатать активный документ
	Exit	Выход		Завершение работы с MathCAD
Edit (Правка)	Undo	Отменить	<Ctri>+<Z>	Отменить последнее действие
	Redo	Повторить	<Ctrl>+<Y>	Повторить последнее отмененное действие
	Cut	Вырезать	<Ctrl>+<X>	Вырезать выбранное выражение в буфер
	Copy	Копировать	<Ctrl>+<C>	Копировать выбранное выражение в буфер
	Paste	Вставить	<Ctrl>+<V>	Вставить выражение из буфера
	Paste Special	Специальная вставка		Вставить объект специального формата, находящийся в буфере
	Delete	Удалить	<Ctrl>+<D>	Удалить выбранный регион
	Select All	Выделить всё	<Ctrl>+<A>	Выделить всю рабочую область
	Find	Найти	<Ctrl>+<F>	Поиск текста
	Replace	Заменить	<Ctrl>+<H>	Замена искомого текста другим
	Go to Page	Перейти к странице		Переход к другой странице
	Check Spelling	Проверка орфографии		Проверка орфографии текстовых регионов

Продолжение табл. А.1

Меню	Команда	Перевод	Сочетание клавиш	Описание
	Links	Ссылки		Управление связями OLE с другими приложениям
	Object	Объект		Активизировать вставленный объект OLE
View (Вид)	Toolbars	Панели инструментов		Показать или скрыть панели инструментов
	Status Bar	Строка состояния		Показать или скрыть строку состояния
	Ruler	Линейка		Показать или скрыть линейку
	Regions	Регионы		Показать или скрыть границы регионов
	Zoom	Масштаб		Изменить масштаб отображения документа
	Refresh	Обновить	<Ctrl>+<R>	Обновить документ
	Animate	Анимация		Создать анимацию
	Playback	Воспроизвести		Воспроизвести анимацию
	Preferences			Изменить основные опции
Insert (Вставка)	Graph	График		Вставить график (с выбором типа графика из подменю)
	Matrix	Матрица	<Ctrl>+<M>	Вставить матрицу или вектор
	Function	Функция	<Ctrl>+<E>	Вставить встроенную функцию
	Unit	Единицы измерения	<Ctrl>+<U>	Вставить единицы измерения размерной величины
	Picture	Рисунок	<Ctrl>+<T>	Создать рисунок для отображения матрицы
	Area	Область		Создать зону
	Math Region	Математическая область	<Ctrl>+ +<Shift>+<A>	Создать математическую область в тексте
	Text Region	Текстовая область	<">	Создать текстовую область в документе
	Page Break	Разрыв страницы		Начать новую страницу
	Hyperlink	Гиперссылка	<Ctrl>+<K>	Вставить гиперссылку
	Reference	Ссылка		Вставить ссылку на другой документ
	Component	Компонент		Вставить компонент другого приложения
	Object	Объект		Внедрение объекта
Format (Формат)	Equation	Формула		Форматирование формул
	Result	Результат		Форматирование вывода результатов вычислений

Продолжение табл. А.1

Меню	Команда	Перевод	Сочетание клавиш	Описание
	Text	Текст		Форматирование текста
	Paragraph	Абзац		Изменение разметки абзаца
	Tabs	Табуляции		Установить табуляцию для документа или выделенного участка текста
	Style	Стиль текста		Определить или применить стиль комбинацию настроек текстового формата
	Properties	Свойства		Изменение свойств области
	Graph	График		Изменения в графиках
	Color	Цвет		Изменить цвета
	Separate Regions	Разделить регионы		Разделить перекрывающиеся регионы в документе
	Align Regions	Выравнять регионы		Выравнивание региона по горизонтали или вертикали
	Format Area	Область		Работа с зоной
	Headers/ Footers	Верхние/ нижние колонтитулы		Изменить колонтитулы для распечатываемых страниц
	Repaginate Now	Разбить на страницы		Разбиение документа на страницы
Math (Математика)	Calculate	Пересчитать	<F9>	Вычислить формулы в видимой части документа
	Calculate Worksheet	Пересчитать всё		Вычислить все формулы в документе
	Automatic Calculation	Считать автоматически		Включение/выключение автоматических вычислений
	Optimization	Оптимизация		Включить/выключить режим оптимизации
	Options	Опции		Установка опций математики
Symbolics (Символика)	Evaluate	Вычислить		Вычислить выражение в виде числа, если это возможно
	Simplify	Упростить		Упростить выражение
	Expand	Разложить		Представить выражение в более развёрнутом виде
	Factor	Разложить на множители		Разложить полином или целое число на простые множители
	Collect	Привести подобные		Привести подобные слагаемые

Окончание табл. А.1

Меню	Команда	Перевод	Сочетание клавиш	Описание
	Polynomial Coefficients	Коэффициенты полинома		Вычислить полиномиальные коэффициенты
	Variable	Переменная		Символьные действия с выделенной переменной
	Matrix	Матрица		Символьные действия с матрицей
	Transform	Преобразование		Символьные интегральные преобразования
	Evaluation Style	Стиль вычислений		Изменить показ символьных ответов
Window (Окно)	Cascade	Каскад		Расположить окна документов каскадом
	Tile Horizontal	По горизонтали		Расположить окна документов по горизонтали
	*	(Имя документа)		Активизировать окно *
Help (Справка)	MathCAD Help	Справка	<F1>	Получение справочной информации
	Developer's Reference	Справка для разработчиков		Дополнительная справка для разработчиков
	Author's Reference	Справка для авторов		Дополнительная справка для авторов
	Resource Center	Центр Ресурсов		Доступ **к Центру Ресурсов**
	Tip of the Day	Совет Дня		Включить показ диалогового окна с **Советом Дня**
	Open Book	Открыть книгу		Открыть существующую в виде файла электронную книгу
	MathCAD Update	Обновление MathCAD		Проверить сайт фирмы MathSoft
	About MathCAD	О программе		Информация о текущей версии MathCAD

ПРИЛОЖЕНИЕ Б

МАТЕМАТИЧЕСКИЕ ФОРМУЛЫ

Язык Object Pascal имеет ограниченное количество встроенных математических функций. Поэтому при необходимости использования других функций следует применять известные соотношения. В таблице Б.1 приведены выражения наиболее часто встречающихся функций через встроенные функции языка Object Pascal.

Таблица Б.1 — Встроенные функции языка Object Pascal

Функция	Соотношение	Соотношение на языке Object Pascal
$\log_a(x)$	$\frac{\ln(x)}{\ln(a)}$	Ln(x)/Ln(a)
x^a	$e^{a\cdot\ln(x)}$	Exp(a*Ln(x))
$\mathrm{tg}(x)$	$\frac{\sin(x)}{\cos(x)}$	Sin(x)/Cos(x)
$\mathrm{ctg}(x)$	$\frac{\cos(x)}{\sin(x)}$	Cos(x)/Sin(x)
$\arcsin(x)$	$\mathrm{arctg}(\sqrt{\frac{x}{1-x^2}})$	Arctan(Sqrt(x/(1–sqr(x))))
$\arccos(x)$	$\frac{\pi}{2}-\arcsin(x)$	Pi/2–Arctan(Sqrt(x/(1–sqr(x))))
$\mathrm{arcctg}(x)$	$\frac{\pi}{2}-\mathrm{arctg}(x)$	Pi/2–Arctan(x)
$\mathrm{sh}(x)$	$\frac{e^x-e^{-x}}{2}$	(Exp(x)–Exp(–x))/2
$\mathrm{ch}(x)$	$\frac{e^x+e^{-x}}{2}$	(Exp(x)+Exp(–x))/2
$\csc(x)$	$\frac{1}{\sin(x)}$	1/Sin(x)
$\mathrm{sc}(x)$	$\frac{1}{\cos(x)}$	1/Cos(x)

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Алексеев, В. Е.* Вычислительная техника и программирование практикум по программированию / В. Е. Алексеев, А. С. Ваулин, Г. Б. Петрова ; под ред. А. В. Петрова. — М. : Высш. шк., 1991.
2. *Бородич, Л. И.* Справочное пособие по приближённым методам решения задач высшей математики / Л. И. Бородич, А. И. Герасимович. — М. : Высш. шк., 1986.
3. *Волков, Е. А.* Численные методы : учеб. пособие для вузов / Е. А. Волков. — М. : Наука, 1987.
4. Вычислительная техника и программирование : учеб. для техн. вузов / А. В. Петров [и др.] ; под ред. А. В. Петрова. — М. : Высш. шк., 1990.
5. *Гусак, А. А.* Математический анализ и дифференциальные уравнения : справ. пособие / А. А. Гусак. — 2-е изд. — Минск : ТетраСистем, 2001.
6. *Лапчик, М. П.* Численные методы : учеб. пособие для студентов вузов / М. П. Лапчик, М. И. Рагулина, Е. К. Хеннер ; под ред. М. П. Лапчика. — М. : Академия, 2004.
7. *Макаров, Е.* Инженерные расчеты в Mathcad 14 / Е. Макаров. — СПб. : Питер, 2007.
8. *Наранович, О. И.* Информатика : метод. указания и задания к лаб. работам для студентов специальности 40 01 02, 36 01 03, 36 01 01 : в 3 ч. / О. И. Наранович, С. Г. Скобля. — Барановичи : РИО БарГУ, 2005.
9. *Плис, А. И.* Mathcad. Математический практикум для инженеров и экономистов : учеб. пособие / А. И. Плис, Н. А. Сливина. — 2-е изд., перераб. и доп. — М. : Финансы и статистика, 2003.

СПИСОК РЕКОМЕНДОВАННЫХ ИСТОЧНИКОВ

1. Задачи и упражнения по математическому анализу / под ред. Б. П. Демидовича. — М. : Наука, 1978.
2. *Корн, Г.* Справочник по математике для научных работников и инженеров / Г. Корн, Т. Корн. — М. : Наука, 1984.
3. *Фурунжиев, Р. И.* Применение математических методов и ЭВМ : практикум : учеб. пособие для вузов / Р. И. Фурунжиев, Ф. М. Бабушкин, В. В. Варавко. — Минск : Выш. шк., 1988.
4. *Туркина, Е. П.* Математическая обработка данных с помощью пакета MathCad : сб. лаб. работ для студентов эконом. специальностей / Е. П. Туркина. — Минск : БГЭУ, 2002.

Производственно-практическое издание

ИНФОРМАТИКА

**Методические указания
для студентов II курса специальностей
1-53 01 01 Автоматизация
технологических процессов и производств,
1-36 01 03 Технологическое оборудование
машиностроительного производства,
1-36 01 01 Технология машиностроения
инженерного факультета**

Разработали: *О. И. Наранович, Г. М. Раковцы, С. Г. Скобля*

Ведущий редактор *Е. Г. Хохол*

Технический редактор *Н. В. Иванова*
Корректор *С. А. Березнюк*
Компьютерная вёрстка *В. В Кукреш*

Подписано в печать 19.09.2012.
Формат 60 × 84 1/16. Бумага офсетная.
Гарнитура Таймс. Отпечатано на ризографе.
Усл. печ. л. 5,35. Уч.-изд. л. 3,22.
Заказ 114. Тираж 165 экз.

ЛИ 02330/0552803 от 09.02.2010

Издатель и полиграфическое исполнение:
учреждение образования
«Барановичский государственный университет»,
225404, г. Барановичи, ул. Войкова, 21.

Инженерный факультет БарГУ

Специальности:

- **Технология машиностроения**;
- **Технологическое оборудование машиностроительного производства**;
- **Информационные системы и технологии**;
- **Автоматизация технологических процессов и производств**;
- **Техническое обеспечение процессов сельскохозяйственного производства**;
- **Экономика и организация производства (машиностроение)**;
- **Агроинженер**;
- **Зооинженер**.

Ведущие промышленные предприятия г. Барановичи являются базовыми: станкостроительный завод «Атлант», завод автоматических линий, автоагрегатный завод, завод торгового машиностроения, завод станкопринадлежностей и др. На них студенты проходят производственные и преддипломные практики. Лаборатории, конструкторские бюро и производственные участки предприятий, оснащенные современными техническими средствами, используются для проведения лабораторных работ и научных исследований.

Выпускники распределяются на предприятия республики с учетом уровня теоретической и практической подготовленности.