

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**  
**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**  
**«БАРАНОВИЧСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**Кафедра информационных технологий  
и безопасности жизнедеятельности**

**ИНФОРМАТИКА**

**Методические указания и задания к лабораторным работам**

**Для студентов 2-го курса дневной формы обучения  
специальностей 1-40 01 02, 1-36 01 03, 1-36 01 01, 1-53 01 01**

**Часть 3**

**Барановичи 2006**

Составители: **О.И. Наранович, С.Г. Скобля**

Рецензенты: **Д.А. Ционенко, Т.Р. Якубович**

Рекомендованы кафедрой информационных технологий и безопасности жизнедеятельности (протокол № 10 от 15.06.2005 г.)

Рассмотрены и рекомендованы к утверждению методической комиссией инженерного факультета (протокол № 1 от 09.09.2005 г.)

**Информатика:** Метод. указания и задания к лабораторным работам для студентов 2-го курса дневной формы обучения специальностей 1-40 01 02, 1-36 01 03, 1-36 01 01, 1-53 01 01/ Барановичский гос. ун-т; [Сост. О.И. Наранович, С.Г. Скобля]. – Барановичи: БарГУ, 2006. – 36 с.

Методические указания и задания к лабораторным работам посвящены изучению раздела «Численные методы решения инженерных задач» дисциплины «Информатика» и содержат пять лабораторных работ с теоретическим материалом и примерами реализации методов в математическом пакете MathCad. Настоящие указания предназначены для студентов 2-го курса инженерного факультета дневной и заочной форм обучения, а также рекомендуются для слушателей факультета повышения квалификации и переподготовки кадров.

## СОДЕРЖАНИЕ

Лабораторная работа № 1. Решение систем линейных уравнений.....	4
Лабораторная работа № 2. Решение нелинейных уравнений.....	8
Лабораторная работа № 3. Аппроксимация и интерполирование функций.....	14
Лабораторная работа № 4. Численное интегрирование.....	20
Лабораторная работа № 5. Численное решение дифференциальных уравнений ..	23
Литература .....	27
Приложение .....	28
Решение СЛАУ средствами MathCad .....	28
Решение нелинейных уравнений с помощью MathCad. ....	29
Интерполирование функций средствами MathCad .....	30
Аппроксимация зависимостей с помощью MathCad .....	32
Вычисление определенного интеграла и производной средствами MathCad ....	34
Решение ОДУ средствами MathCad .....	35

Библиотека БарГУ



0000 4582

## ЛАБОРАТОРНАЯ РАБОТА № 1

### РЕШЕНИЕ СИСТЕМ ЛИНЕЙНЫХ УРАВНЕНИЙ

**Цель:** изучить особенности различных методов решения систем линейных алгебраических уравнений (далее СЛАУ), приобрести навыки решения СЛАУ с помощью средств MS Excel и MathCad.

#### ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

**Постановка задачи.** Система линейных алгебраических уравнений имеет вид:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1, \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= b_2, \\ &\dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n &= b_m. \end{aligned}$$

Мы будем рассматривать только совместные определенные системы, т.е. системы, имеющие единственное решение. Это ограничение приводит к тому, что число уравнений системы должно равняться числу неизвестных.

Используемые в наше время методы решения СЛАУ разбивают на 2 группы:

1) *точные методы* – это методы, в которых вычисления 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 решения. К точным методам относится метод Гаусса решения СЛАУ;

2) *приближенные методы* – это методы, которые даже 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 приближенным методам решения СЛАУ относят методы простой итерации, метод Зейделя и др.

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

**Метод Гаусса.** Одним из наиболее распространенных 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:

$$\begin{aligned} x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1, \\ a_{22}x_2 + \dots + a_{2n}x_n &= b_2, \\ &\dots \\ a_{m2}x_2 + \dots + a_{mn}x_n &= b_m. \end{aligned}$$

Для простоты будем считать, что коэффициенты и свободные члены новой системы мы переобозначили вновь как  $a_{ii}$  и  $b_i$ .

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

$$x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n = b_1,$$

$$x_2 + a_{23}x_3 + \dots + a_{2n}x_n = b_2,$$

$$a_{33}x_3 + \dots + a_{3n}x_n = b_3,$$

...

$$a_{n3}x_3 + \dots + a_{nn}x_n = b_n.$$

Продолжая подобные преобразования, в итоге получим систему вида:

$$x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n = b_1,$$

$$x_2 + a_{23}x_3 + \dots + a_{2n}x_n = b_2,$$

$$x_3 + \dots + a_{3n}x_n = b_3,$$

...

$$x_n = b_n.$$

То есть система оказывается приведенной к треугольному виду. Такие преобразования называют прямым ходом метода Гаусса.

Далее из последнего уравнения полученной системы 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  $x_1, x_2, \dots, x_n$ , 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. То есть из оставшихся  $n-1$  уравнений на вторую строку переставляют уравнение с максимальным по модулю коэффициентом при  $x_2$ . Делят второе уравнение на  $a_{22}$  и последовательно вычитают его из оставшихся  $n-2$ -х, 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.

**Метод простой итерации решения СЛАУ.** Для использования этого метода систему, имеющую вид:

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1,$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2,$$

...

$$a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = b_m,$$

нужно привести к следующему виду (называемому нормальным):

$$\begin{aligned}
 x_1 &= a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n + b_1, \\
 x_2 &= a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n + b_2, \\
 &\dots \\
 x_n &= a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n + b_n.
 \end{aligned}
 \tag{1}$$

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

Данный метод сходится при выполнении одного из следующих условий:

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

То есть метод простой итерации решения СЛАУ сходится, если максимальная из сумм модулей коэффициентов при неизвестных системы (1), взятых по строкам, меньше единицы либо если максимальная из сумм модулей коэффициентов при неизвестных системы (1), взятых по столбцам, меньше единицы, либо если сумма квадратов всех коэффициентов при неизвестных в правой части системы (1) 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 некоторая десятичная цифра повторилась три и более раз – ее можно считать верной.

Во всех остальных случаях (когда выполняется одно из условий (2)) степень приближения к точному решению оценивается по формуле:

$$|x_k - x_{k-1}| < \frac{\varepsilon(1-\alpha)}{\alpha}
 \tag{3}$$

При выполнении одного из условий (2) в качестве начального значения решения можно взять любой набор значений, где  $\alpha$  – величина, вычисляемая по одной из формул (2), по которой обнаружена сходимость метода.

**Метод Зейделя решения СЛАУ.** 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  $x_1, x_2, \dots, x_{i-1}$ .

Вычислительные формулы при этом примут вид:

$$\begin{aligned}
 x_1^{(k)} &= a_{11}x_1^{(k-1)} + a_{12}x_2^{(k-1)} + \dots + a_{1n}x_n^{(k-1)} + b_1, \\
 x_2^{(k+1)} &= a_{21}x_1^{(k)} + a_{22}x_2^{(k-1)} + \dots + a_{2n}x_n^{(k-1)} + b_2, \\
 &\dots \\
 x_n^{(k+1)} &= a_{n1}x_1^{(k)} + a_{n2}x_2^{(k)} + \dots + a_{nn}x_n^{(k-1)} + b_n.
 \end{aligned}$$

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

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

### ЗАДАНИЯ

1. Используя методы Гаусса и Зейделя, разработать схемы соответствующих алгоритмов решения СЛАУ в среде MS Excel.

2. Разработать схемы программ решения СЛАУ с использованием средств математического пакета MathCad: `Isolve()`, `Given ... Find()`, метода обратной матрицы.

3. Получить решение СЛАУ, приведенное в табл. 1 (в соответствии со своим вариантом), с использованием алгоритмов, полученных при выполнении пп. 1 и 2.

4. Вычислить точностные оценки методов по координатам:

$$\delta = \max |x_i - x_i^*|, \quad i = 1..n,$$

где  $x_i^*$  – значение, найденное методом Гаусса.

5. На основании результатов, полученных в пп. 3, 4, провести сравнительный анализ методов по точности и времени решения, сделать вывод.

*Примечание.* Погрешность полученных решений не должна превышать  $\epsilon = 0,0001$ .

Таблица 1

Вариант	Матрица А	Вектор В
1	2	3
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	$\begin{pmatrix} 3.82 & 1.02 & 0.75 & 0.81 \\ 1.05 & 4.53 & 0.98 & 1.53 \\ 0.73 & 0.85 & 4.71 & 0.81 \\ 0.88 & 0.81 & 1.28 & 3.5 \end{pmatrix}$	$\begin{pmatrix} 15.655 \\ 22.705 \\ 23.480 \\ 16.11 \end{pmatrix}$
6	$\begin{pmatrix} 3075 & 211 & 0.7 \\ 1.21 & 205 & 0.64 \\ 1.2 & 153 & 3.21 \end{pmatrix}$	$\begin{pmatrix} -2.1 \\ 1 \\ 5 \end{pmatrix}$

Продолжение таблицы 1

1	2	3
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	$\begin{pmatrix} 1.02 & -0.25 & -0.3 \\ -0.41 & 1.13 & -0.15 \\ -0.25 & -0.14 & 1.21 \end{pmatrix}$	$\begin{pmatrix} 0.51 \\ 1.35 \\ 2.78 \end{pmatrix}$
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	$\begin{pmatrix} 9 & 3 & 3 \\ 5 & 10 & -1 \\ 1 & 4 & 5 \end{pmatrix}$	$\begin{pmatrix} -2 \\ -9 \\ -6 \end{pmatrix}$

### КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что называется системой линейных алгебраических уравнений (СЛАУ) и ее решением?
2. Назовите известные Вам методы и средства решения СЛАУ.
3. Объясните сущность метода Гаусса с выбором главного элемента (прямой ход, обратный ход).
4. Каковы сущность и особенности сходимости итерационных методов решения СЛАУ?
5. В чем особенности метода Зейделя (как выбрать начальное приближение, чем обосновать полученное по этому методу решение)?

### ЛАБОРАТОРНАЯ РАБОТА № 2 РЕШЕНИЕ НЕЛИНЕЙНЫХ УРАВНЕНИЙ

**Цель:** 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.

#### ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

**Постановка задачи.** В практике научных и инженерных расчетов возникает необходимость в решении 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*.

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  $Ox$ . 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 приближенными значениями, то для решения такого уравнения используют численные приближенные методы, которые позволяют определять приближенное значение корней с заданной точностью.

Процесс решения нелинейного уравнения состоит из двух этапов.

1. *Этап отделения корней*, на котором 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.

**Методы отделения корней.** Мы ограничимся рассмотрением методов поиска лишь действительных корней. Существуют 2 способа отделения корней.

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

Часто уравнение  $f(x) = 0$  можно привести к виду  $f_1(x) = f_2(x)$ . Тогда можно отделить корни, 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. Найти производную функции  $f'(x)$ .
2. Найти критические точки функции  $f(x)$ .
3. Составить таблицу знаков функции  $f(x)$  на границах отрезка  $AB$  и в критических точках.
4. Определить отрезки, на концах которых функция принимает значения противоположных знаков.
5. Выбрать в качестве начальных приближенных 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 п. 4.

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

Приближенные 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  $\alpha$ , если существуют такие числа  $q \geq 0$  и  $k_0 \geq 0$ , 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  $\alpha > 2$  – сверхкватричной. Чем больше порядок сходимости, тем сложнее вычислительный алгоритм, но тем 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  $f(a)f(b) < 0$ .

*Суть метода.*

1. Отрезок  $AB$  делят пополам и находят начальное приближение корня  $x_0 = \frac{a+b}{2}$ .

2. Если  $f(x_0) = 0$ , то  $x_0$  – корень уравнения; в противном случае из отрезков  $[a; x_0]$  и  $[x_0; b]$  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  $f(a)f(x_0) < 0$ , то корень расположен на отрезке  $[a; x_0]$ ; если  $f(b)f(x_0) < 0$ , то корень расположен на отрезке  $[x_0; b]$ .

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

Метод деления отрезка пополам всегда сходится.

**Метод последовательных приближений.** Часто этот метод называют методом простых итераций. Для данного метода уравнение  $f(x) = 0$  должно быть представлено в итерационном виде  $x = \varphi(x)$ .

*Суть метода.*

В качестве начального приближения корня  $x_0$  выбирают любую точку, расположенную внутри отрезка  $[a; b]$ . Последующие приближения вычисляются с помощью итерационной процедуры:

$$x_1 = \varphi(x_0);$$

$$x_2 = \varphi(x_1)$$

...

$$x_k = \varphi(x_{k-1}), k = 0, 1, 2, \dots$$

Процесс продолжают до тех пор, пока не будет найдено приближенное значение корня, имеющее заданную точность.

*Теорема сходимости итерационного процесса:* если интервал  $(a, b)$  является интервалом изоляции корня уравнения  $x = \varphi(x)$  и во всех точках этого интервала  $|\varphi'(x)| \leq q < 1$ , то итерационный процесс сходится. Вычисления прекращаются при выполнении условия  $|x_k - x_{k-1}| \leq \frac{1-q}{q} \varepsilon$ , где  $\varepsilon$  – заданная абсолютная погрешность вычисления корня.

Преобразование уравнения  $f(x) = 0$  к виду  $x = \varphi(x)$  можно произвести различными способами, простейшие из них:

1) выделяют  $x$  из  $f(x) = 0$ , а все остальное переносят в правую часть;

2) выполняют преобразование  $x = x + cf(x)$ , где  $c$  – произвольная постоянная. Тогда  $\varphi(x) = x + cf(x)$ . Подбором величины  $c$  добиваются получения возможного меньшего значения константы  $q$ .

Метод последовательных приближений имеет преимущество в том, что при его использовании не накапливается погрешность вычисления. Любое найденное  $x_k$  можно использовать в качестве нового начального приближения. То есть если в процессе вычислений допускались ошибки, то они не влияют на окончательный результат в том случае, когда заданная погрешность нахождения корня существенно превышает погрешность вычислений (в т.ч. машинных). Это свойство метода последовательных приближений делает его наиболее надежным методом уточнения корней.

**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.

*Суть метода.*

Выбирают начальное приближение корня  $x_0$ , в качестве которого удобно взять конец отрезка  $[a; b]$ , для которого  $f(x_0)f''(x_0) > 0$  (в противном случае сходимость метода Ньютона не гарантируется). 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. Каждое последующее приближение определяется из рекуррентного соотношения:

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

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  $|x_k - x^*| = \frac{|f(x_k)|}{|f'(c)|}$ . О точке  $c$  известно лишь то, что она находится между  $x_k$  и  $x^*$ . Поэтому оценка погрешности возможна с помощью следующего соотношения:  $|x_k - x^*| \leq \frac{|f(x_k)|}{M}$ ,

где  $M = \min_{[a; b]} |f'(x)|$ .

Метод Ньютона имеет квадратичную сходимость. Если сложно выбрать начальное приближение, то иногда начинают решение методом деления отрезка пополам (т.е. берут середину отрезка) и продолжают уточнять с помощью метода Ньютона.

**Метод хорд.** При реализации метода касательных, при каждой итерации нужно вычислять значение как функции  $f(x)$ , так и ее первой производной. Существует вариант метода Ньютона – метод хорд (секущих), который позволяет избежать вычисления производной. Рекуррентное соотношение для вычисления приближенных 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  $f(c)f'(c) > 0$ . В качестве 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.

Этот метод, как и метод касательных, также имеет квадратичную сходимость.

**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 жесткие 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 использование более простого метода с малой скоростью сходимости, например, метода деления отрезка пополам, может дать результат гораздо быстрее, чем использование изощренного и сложного для понимания и программирования метода уточнения с высокой скоростью сходимости.

## ЗАДАНИЯ

1. Разработать алгоритмы решения нелинейных уравнений методами: деления отрезка пополам, хорд, касательных (Ньютона).

2. Найти решения нелинейных уравнений, приведенных в таблице 2 (в соответствии со своим вариантом), с использованием функции `root(...)` математического пакета MathCad.

3. Средствами Delphi создать приложение для решения нелинейных уравнений, приведенных в таблице. Начальные приближения корней выбрать из указанных областей.

4. Оценить полученные результаты, сделать вывод.

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

Таблица 2

Вариант	Уравнение	Область, содержащая корень	Методы решения
1	2	3	4
1	$0.25x^3 + x - 1.2502 = 0$	[0; 2]	Касательных
	$x + \sqrt{x} + \sqrt[3]{x} - 2,5 = 0$	[0.4; 1]	Деления пополам
	$x - \frac{1}{3 + \sin 3.6x} = 0$	[0; 0.85]	Хорд
2	$0.1x^2 - x \ln x = 0$	[1; 2]	Деления пополам
	$\operatorname{tg} x - \frac{1}{3} \operatorname{tg}^3 x + \frac{1}{5} \operatorname{tg}^5 x - \frac{1}{3} = 0$	[0; 0.8]	Хорд
	$x - 2 + \sin \frac{1}{x} = 0$	[1.2; 2]	Касательных
3	$e^x + \ln x - 10x = 0$	[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	$3x - 14 + e^x - e^{-x} = 0$	[1; 3]	Деления пополам
5	$\sqrt{1-x} - \operatorname{tg} x = 0$	[0; 1]	Хорд
	$x + \cos(x^{0.52} + 2) = 0$	[0.5; 1]	Касательных
6	$3 \ln^2 x + 6 \ln x - 5 = 0$	[1; 3]	Деления пополам
	$\sin x^2 + \cos x^2 - 10x = 0$	[0; 1]	Хорд
	$\ln x - x + 1.8 = 0$	[2; 3]	Касательных
7	$x \operatorname{tg} x - \frac{1}{3} = 0$	[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
8	$\sqrt{1-x} - \cos \sqrt{1-x} = 0$	[0; 1]	Деления пополам
	$0.6 * 3^x - 2.3x - 3 = 0$	[2; 3]	Касательных
	$0.25x^3 + x - 1.25 = 0$	[0; 2]	Хорд
9	$3x - 4 \ln x - 5 = 0$	[2; 4]	Касательных
	$e^x + \ln x - 10x = 0$	[3; 4]	Хорд
10	$\ln(0.9146x) - 1.038x + 2.5 = 0$	[0; 6]	Деления пополам
11	$1.2755x^3 - 3.601x^2 - 1.37x + 6.76 = 0$	[-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
12	$3 \sin \sqrt{x} + 0.35x - 3.8 = 0$	[2; 3]	Деления пополам
	$x - 2 + \sin(1/x) = 0$	[1.2; 2]	Хорд

Продолжение табл. 2

13	$1 - x + \sin x - \ln(1+x) = 0$	[0; 1.5]	Касательных
14	$x^2 - \ln(1+x) - 3 = 0$	[2; 3]	Касательных
	$1 - \frac{3}{3 + \sin 3.6x} = 0$	[-1; 2]	Хорд
	$\ln x - x + 1.8 = 0$	[2; 3]	Деления пополам
15	$0.1x^2 - x \ln x = 0$	[1; 2]	Касательных
	$x + \cos(x^{0.52} + 2) = 0$	[0.5; 1]	Хорд
	$\sqrt{1 - 0.4x} - \arcsin x = 0$	[0; 1]	Деления пополам

### 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.

### ЛАБОРАТОРНАЯ РАБОТА № 3

#### АППРОКСИМАЦИЯ И ИНТЕРПОЛИРОВАНИЕ ФУНКЦИЙ

**Цель:** 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.

#### ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

**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 приближенную 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*. 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 ее значений в нужных точках

требует громоздких вычислений. Чтобы этого избежать, функцию  $f(x)$  также заменяют приближенной функцией  $\varphi(x)$ .

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

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

Объясняется это тем, что сравнительно просто автоматизировать вычисления коэффициентов многочлена, просто 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:

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

или

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

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:

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

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  $x_i$  (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.

Введем понятие разделенных разностей.

Разделенными разностями 1-го порядка называются значения:

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

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

...

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

$\Delta y_{i-1} = y_i - y_{i-1}$  — называются конечными разностями 1-го порядка.

Разделенными разностями 2-го и более высоких порядков называют:

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

$$f(x_3, x_2, x_1, x_0) = \frac{f(x_3, x_2) - f(x_2, x_1)}{x_3 - x_1} = \frac{\Delta^2 y_1}{x_3 - x_1},$$

...

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

$\Delta^2 y_{i-1} = \Delta y_i - \Delta y_{i-1}$  – конечная разность 2-го порядка и т.д.

С учетом введенных обозначений *интерполяционный многочлен Ньютона* имеет вид:

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

*Погрешность:*

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

Так как любой  $i$ -й член полинома Ньютона зависит только от  $x_i$  и  $n$  первых  $i$ -точек и от значения функции в них, то 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.

*Метод наименьших квадратов.*

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

$$P_m = \sum_{i=0}^m a_i \cdot x^i = a_0 + a_1 \cdot x + a_2 \cdot x^2 + \dots + a_m \cdot x^m. \quad (1)$$

Для определения коэффициентов  $a_i$  (где  $i=1, 2, \dots, m$ ) используют аппроксимацию методом наименьших квадратов. Функционал находится по формуле  $S = \sum_{i=1}^n [P(x_i) - y_i]^2 \rightarrow 0$ , где  $n$  – количество пар значений аргумента  $x_i$  и функции  $y_i$ .

Так как  $S$  должен быть минимальным, то первые частные производные от  $S$  по всем коэффициентам полинома должны равняться нулю.

Вычислим их и приравняем к нулю:

$$\begin{cases} \frac{\partial S}{\partial a_0} = 2 \cdot \sum_{i=1}^n (a_0 + a_1 \cdot x_i + \dots + a_m \cdot x_i^m - y_i) = 0; \\ \frac{\partial S}{\partial a_1} = 2 \cdot \sum_{i=1}^n (a_0 + a_1 \cdot x_i + \dots + a_m \cdot x_i^m - y_i) \cdot x_i = 0 \\ \dots \\ \frac{\partial S}{\partial a_m} = 2 \cdot \sum_{i=1}^n (a_0 + a_1 \cdot x_i + \dots + a_m \cdot x_i^m - y_i) \cdot x_i^m = 0. \end{cases} \quad (2)$$

После преобразования система слегка упростится:

$$\begin{cases} \sum_{i=1}^n (a_0 + a_1 \cdot x_i + \dots + a_m \cdot x_i^m) = \sum_{i=1}^n y_i; \\ \sum_{i=1}^n (a_0 \cdot x_i + a_1 \cdot x_i^2 + \dots + a_m \cdot x_i^{m+1}) = \sum_{i=1}^n x_i y_i; \\ \dots \\ \sum_{i=1}^n (a_0 \cdot x_i^m + a_1 \cdot x_i^{m+1} + \dots + a_m \cdot x_i^{2m}) = \sum_{i=1}^n x_i^m y_i. \end{cases} \quad (3)$$

Если полином первой степени, то получим 2 уравнения; если шестой степени, то – 7 уравнений. Введем следующие обозначения:

$$t_0 = n, \quad t_1 = \sum_{i=1}^n x_i, \quad \dots, \quad t_m = \sum_{i=1}^n x_i^{2m},$$

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

С учетом этих обозначений система (2) переписывается следующим образом:

$$\begin{cases} t_0 \cdot a_0 + t_1 \cdot a_1 + \dots + t_m \cdot a_m = C_0; \\ t_1 \cdot a_0 + t_2 \cdot a_1 + \dots + t_{m+1} \cdot a_m = C_1; \\ \dots \\ t_m \cdot a_0 + t_{m+1} \cdot a_1 + \dots + t_{2m} \cdot a_m = C_m. \end{cases} \quad (4)$$

Решив эту систему одним из известных методов, определится ряд значений  $a_0, a_1, \dots, a_m$ , с помощью которого и строится многочлен (1).

Среднеквадратичное отклонение вычисляется по формуле:

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

## ЗАДАНИЯ

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

2. Произвести интерполирование и аппроксимацию табличных функций, приведенных в таблице 2 (в соответствии со своим вариантом), на отрезке  $[a, b]$

с шагом  $h$  средствами MS Excel и MathCad. Средствами Delphi создать приложение, решающее аналогичную задачу.

3. По результатам выполнения п. 2 построить графики функций.

4. Произвести сравнительный анализ полученных результатов.

Таблица 3

Вариант	Значение аргумента $x$	Значение функции $y$	Пределы изменения аргумента $[a, b]$	Шаг интерполирования $h$
1	-1.06 -0.837 -0.684 -0.315 -0.117 -0.0 0.115 0.5	1.22 0.854 0.513 0.271 0.217 0.198 0.218 0.277	[-0.8; 0.4]	0.2
2	-2.15 -1.83 -1.62 -1.45 -1.01 -0.72 -0.48 0.0	-2.23 -2.65 -3.1 -3.54 -4.26 -4.38 -4.52 -4.27	[-1.95; 0.1]	0.3
3	-0.21 -0.143 -0.099 -0.032 0.114 0.182 0.257 0.38	-12.64 -11.05 -10.25 -9.32 -9.25 -10.0 -11.48 -14.4	[-0.2; 0.28]	0.08
4	0.215 0.441 0.638 0.865 1.05 1.30 1.55 1.82	5.82 4.63 4.10 3.34 3.0 3.29 4.32 5.72	[0.4; 1.6]	0.2
5	-1.0 -0.96 -0.86 -0.79 0.22 0.50 0.93 1.10	-1.00 -0.151 0.894 0.986 0.895 0.50 -0.306 -0.51	[-0.98; 1.02]	0.35
6	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

Продолжение табл. 3

7	1.4	-0.24	[1.5; 2.7]	0.2
	1.6	-0.24		
	1.8	-0.16		
	2.0	0.0		
	2.2	0.24		
	2.4	0.56		
	2.6	0.96		
2.8	1.44			
8	0.43	1.64	[0.5; 0.6]	0.02
	0.48	1.73		
	0.55	1.88		
	0.62	2.03		
	0.70	2.23		
	0.75	2.36		
	0.78	2.41		
0.81	2.78			
9	0	800	[68; 72]	0.5
	15	718		
	30	665		
	45	621		
	60	586		
	75	556		
	90	532		
105	510			
10	-2.0	-17.0	[9.8; 11]	0.2
	3.0	-3.0		
	4.5	1.2		
	12.0	1.8		
	15.0	3.0		
	18.5	4.5		
	20.0	7.0		
23.0	9.1			
11	-2.3	-12.3	[1.5; 3]	0.5
	0.0	8.6		
	1.1	13.4		
	4.8	15.1		
	7.3	21.4		
	9.2	24.2		
	11.4	28.3		
13.0	32.1			
12	2.34	15.16	[3; 6]	0.5
	5.16	25.03		
	7.03	32.18		
	8.42	37.11		
	9.61	44.82		
	10.12	51.62		
	11.35	50.13		
12.12	73.16			

### КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Аппроксимация и интерполирование функций. Постановка задачи.
2. В чем особенность аппроксимации?
3. Опишите особенности метода наименьших квадратов в случае линейной и квадратичной зависимости.
4. Какой из методов полиномиальной интерполяции зависимости дает 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 Лагранжа и Ньютона.

## ЛАБОРАТОРНАЯ РАБОТА № 4

### ЧИСЛЕННОЕ ИНТЕГРИРОВАНИЕ

**Цель:** 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 определенного интеграла.

#### ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

**Постановка задачи.** Значение определенного интеграла вычисляется по формуле Ньютона-Лейбница, если удастся выразить первообразную формулу через элементарные функции:

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

где  $F(x)$  – первообразная.

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

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

Суть численных методов вычисления определенного 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 квадратурных формул:

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

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 определенного 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  $i = 0, 1, \dots, n$ . Приближенно определяются значения площадей, соответствующих каждому отрезку. Сумма этих площадей даст приблизительное значение интеграла. В зависимости от способа разбиения отрезка узлами интерполяции  $x_i$  различают два подхода к построению квадратурных формул:

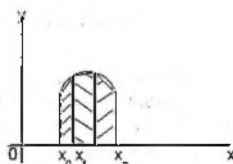
1. Местоположения и длины интервалов разбиения выбираются заранее в начале расчета. В случае равноотстоящих точек узлов интерполирования: шаг

$h = \frac{b-a}{n}$ , а узлы  $X_i = X_{0+i} \cdot h$ . В таком случае квадратурные формулы называются формулами Ньютона-Котеса. Квадратурные формулы Ньютона-Котеса различаются степенями используемых итерационных многочленов. Широкое применение находят формулы трапеции и Симпсона.

2. Местоположения и длины 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. Значение

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



Суммирование значений интеграла по всем  $n$  участкам разбиения дает общую площадь:  $S = h \cdot \left( \frac{f(a) + f(b)}{2} + f(x_1) + f(x_2) + \dots + f(x_{n-1}) \right)$ .

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

$$|R| \leq \frac{(b-a) \cdot h^2}{12} \cdot M, \quad M = \max(f''(x)), a \leq x \leq b,$$

где  $h$  – шаг интегрирования,  $M$  – коэффициент, который получается из набора вторых производных для отрезка  $[a; b]$ .

Очевидно, что формула трапеции даст точное значение интеграла для линейной подынтегральной функции  $f(x)$ .

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

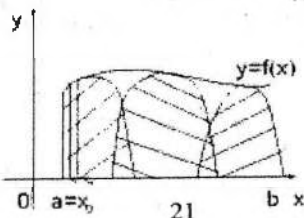
$$\int_a^b f(x) dx \approx \frac{h}{3} \cdot (f(a) + 4 \cdot f(x_1) + 2 \cdot f(x_2) + \dots + 2 \cdot f(x_{n-2}) + 4 \cdot f(x_{n-1}) + f(b)) - \text{формула}$$

Симпсона либо формула парабол.

Значения функции  $f(x)$  в нечетных точках разбиения  $x_1, x_3$  и т.д. входят в формулу с коэффициентом 4. В четных 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. Сумму всех фигур, построенных таким образом, дает примерное значение интеграла.



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

$$R \leq \frac{(b-a) \cdot h^4}{180} \cdot M, \quad M = \max(f^{(4)}(x)), a \leq x \leq b.$$

Формула парабол является точной для полиномов третьей степени включительно, т.к. для них четвертая производная равна 0. Из-за сложности, а часто и невозможности вычисления четвертой производной используют формулу Рунге:

$$|R| = \frac{S_{2n} - S_n}{15}, \quad \text{где } S_n - \text{значение интеграла при разбиении отрезка интегрирования}$$

на  $n$  участков ( $n$  – четное);  $S_{2n}$  – значение интеграла при разбиении отрезка интегрирования на  $2n$  участков.

**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 расчетах.

### ЗАДАНИЯ

1. Разработать схемы алгоритмов интегрирования функций по методам трапеций и Симпсона.

2. В среде Delphi создать приложение для решения задачи численного интегрирования функций, приведенных в табл. 4 (в соответствии со своим вариантом), методами трапеций и Симпсона. Предусмотреть в программе вычисление точного значения определенного интеграла через первообразную функцию (табл. 4).

3. Провести интегрирование тех же функций средствами пакета MathCad.

4. Вычислить 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.

4. На основании результатов пп. 2, 3 провести сравнительный анализ методов численного интегрирования.

Таблица 4

Вариант	Подынтегральная функция $f(x)$	Интервал интегрирования $[a, b]$	Кол-во частей разбиения $n$	Шаг $h$	Первообразная функция
1	2	3	4	5	6
1	$xe^x \sin(x)$	$[0; 1]$	50	0.02	$\frac{xe^x(\sin x - \cos x)}{2} + \frac{e^x \cos x - 1}{2}$

Продолжение табл. 4

2	$\frac{1}{\sqrt{9+x^2}}$	[0; 2]	200	0.01	$\ln(x + \sqrt{x^2+9}) - \ln 3$
3	$\frac{1}{\sqrt{1+3x+2x^2}}$	{0; 1}	40	0.025	$\frac{1}{\sqrt{2}} \ln \frac{x+0.75 + \sqrt{(x+0.75)^2 - 0.0625}}{0.75 + \sqrt{0.5}}$
4	$\left(\frac{\ln x}{x}\right)^2$	{1; 2}	40	0.025	$\frac{(\ln x)^2 + 3(\ln x)^2/2 + 3(\ln x)/2 + 3/4}{2x^2}$
5	$\frac{x^3}{3+x}$	{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	$\arctg x$	{0; 2}	50	0.04	$\frac{x^2}{2} \arctg x - \frac{x}{2} + \frac{1}{2} \arctg x$
7	$\frac{1}{x \lg x}$	{2; 3}	40	0.025	$2.3026(\ln \ln x - \ln 2)$
8	$\frac{1}{x} \sin \frac{1}{x}$	{1; 2.5}	50	0.03	$\cos\left(\frac{1}{x}\right)$
9	$\cos x$	{0; $\pi$ }	60	0.026	$\sin x$
10	$x^2(1 + \ln x)$	{1; 3}	40	0.06	$x^4$
11	$\frac{1}{\sqrt{1+3x}}$	{0; 1}	40	0.025	$\frac{2}{3} \sqrt{1+3x}$
12	$\sin^2 x$	$\left[0; \frac{\pi}{2}\right]$	60	0.026	$\frac{x}{2} - \frac{\sin 2x}{4}$

### КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Назовите сущность и отличия известных Вам методов численного интегрирования.
2. Поясните геометрический смысл определенного интеграла.
3. Расскажите сущность и поясните алгоритм вычисления определенного интеграла методами трапеций и Симпсона.
4. Как оценить точность вычисления определенного интеграла?
5. Приведите примеры использования определенных интегралов.

### ЛАБОРАТОРНАЯ РАБОТА № 5

#### ЧИСЛЕННОЕ РЕШЕНИЕ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ

**Цель:** изучить наиболее распространенные 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.

#### ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

**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 таких процессов могут

служить явления, возникающие в электрических сетях, распространение радиоволн, движение материальных точек, изменение химического состояния вещества и т.д.

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 приближенные численные методы решения ДУ.

Задача Коши формулируется следующим образом:  $y'(x) = f(x, y)$ ,  $y(0) = y(x_0)$  – начальное условие,  $[x_0, x_n]$  – отрезок.

Для решения ДУ  $n$ -го порядка  $y^{(n)} = f(x, y, y', y'', \dots, y^{(n-1)})$  используют метод понижения производной, т.е. уравнение  $n$ -го порядка сводят к системе уравнений  $n$ -го порядка способом замены переменных.

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

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

Методы решения одного уравнения 1-го порядка распространяются на систему уравнений 1-го порядка.

**Сушность численных методов решения ДУ.** На отрезке  $[x_0, x_n]$  выбирается некоторое множество точек. Это множество точек 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  $h = \text{const}$ ,  $x_n = x_0 + kh$ .

Проблема численного решения ДУ заключается в построении алгоритма вычисления. Различают *одношаговые* и *многошаговые* методы решения ДУ. Метод Эйлера и метод Рунге-Кутты относятся к одношаговым. *Многошаговые* методы – это методы прогноза и коррекции. Для метода Адамса необходимо знать две предыдущие точки, для метода Милна – три. Их преимущество в том, что для них существует формула оценки погрешности.

**Метод Эйлера.** Различные методы группы методов Рунге-Кутта различаются друг от друга объемом производимых вычислений и получаемой при этом точностью. Метод Эйлера является методом Рунге-Кутта первого порядка точности. *Формула для вычисления методом Эйлера:*

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

Следовательно:

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

...

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

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

Метод Рунге-Кутты 4-го порядка точности. Смещение из точки  $[x_k; y_k]$  в точку  $[x_{k+1}; y_{k+1}]$  происходит не сразу, а через промежуточные точки. На практике наибольшее распространение получил метод 4-го порядка точности. Значение функции в  $i+1$ -й точке вычисляется следующим образом:

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

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

$$\text{где: } k_1 = h \cdot f(x_i, y_i),$$

$$k_2 = h \cdot f\left(x_i + \frac{h}{2}, y_i + \frac{k_1}{2}\right),$$

$$k_3 = h \cdot f\left(x_i + \frac{h}{2}, y_i + \frac{k_2}{2}\right),$$

$$k_4 = h \cdot f(x_i + h, y_i + k_1).$$

Метод Рунге-Кутты обладает достаточно высокой точностью, легко программируется, т.к. для вычисления  $y_{i+1}$  нужно знать лишь одно значение  $y_i$ . С помощью этого метода можно начинать решение ДУ. Величину шага изменения аргумента  $x$  можно легко менять на любом этапе вычисления.

Недостатки:

- 1) необходимость 4 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 2 раза, провести вычисления и снова оценить  $\delta$ .

Метод Рунге-Кутты имеет порядок точности, сопоставимый со значением шага, взятым в 4-й степени. Для оценки погрешности метода пользуются формулой Рунге:

$$\varepsilon \sim h^4, \quad \varepsilon = \frac{1}{15} \cdot (y_{2h} - y_h).$$

## ЗАДАНИЯ

1. Разработать схемы алгоритмов решения обыкновенного дифференциального уравнения методами Эйлера и Рунге-Кутты 4-го порядка точности.

2. В среде Delphi создать приложение для решения дифференциальных уравнений, приведенных в табл. 5 (в соответствии со своим вариантом), методами Эйлера и Рунге-Кутты 4-го порядка.

3. Решить дифференциальное уравнение (в соответствии со своим вариантом) с помощью MathCad.

4. Вычислить погрешности методов решения дифференциальных уравнений.

5. На основании результатов пп. 2, 3, 4 провести сравнительный анализ методов численного решения дифференциальных уравнений.

Таблица 5

Вариант	Уравнение	Начальные данные	Отрезок интегрирования
1	$y' = \frac{y}{x} + x^2$	$y(1) = 1$	$x \in [1; 4]$
2	$\sqrt{1-x^2}y' + \sqrt{1-y^2} = 0$	$y(0) = 0$	$x \in [0; 0.8]$
3	$y' = \frac{2xy}{x^2 + y^2}$	$y(1) = 2$	$x \in [1; 3]$
4	$y' = -(x^2 + 1)\sqrt{1-y^2}$	$y(0) = 0.95$	$x \in [0; 3]$
5	$y' = 2xy - x^3 + x$	$y(0) = 0.8$	$x \in [0; 5]$
6	$x^2 + 3x = e^{2x}$	$x(0) = -3$	$t \in [-1; 3]$
7	$y' = y^2 + \frac{1}{2x^2}$	$y(1) = 0.9$	$x \in [1; 5]$
8	$y' = \frac{y}{x + y^3}$	$y(1) = 1.5$	$x \in [2; 4]$
9	$y' = xy^3 + x^2$	$y(0) = 0$	$x \in [-1; 1]$
10	$y' = x^2 + xy^3$	$y(0) = 0$	$x \in [-2; 2]$
11	$y' = x^2 + xy + y^2$	$y(1) = 0$	$x \in [0; 3]$
12	$y' = \frac{1}{3x^2 + y^2}$	$y(1) = 2$	$x \in [1; 3]$

### 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-Кутты 4-го порядка точнее метода Эйлера?
5. За счет чего возникает погрешность в методе Эйлера? Как ее уменьшить?
6. Как выбирается шаг интегрирования в методе Рунге-Кутты 4-го порядка точности?

## РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

### Основная

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. Воробьева Г.Н., Данилова А.Н. Практикум по вычислительной математике: Учеб. пособ. для техникумов. – М.: Высш. шк., 1990.
3. Волков Е.А. Численные методы: Учеб. пособ. для вузов. – М.: Наука, 1987.
4. Вычислительная техника и программирование: Учеб. для технических вузов / А.В. Петров, В.Е. Алексеев, А.С. Ваулин и др.; Под ред. А.В. Петрова. – М.: Высш. шк., 1990.
5. Гусак А.А. Математический анализ и дифференциальные уравнения: Справ. пособ. – Изд. 2-е. – Мн.: Тетрасистем, 2001.
6. Дьяконов В. MathCad 2001: Спец. справ. – СПб.: Питер, 2002.
7. Кренкель Т.Э. и др. Персональные ЭВМ в инженерной практике: Справ. / Т.Э. Кренкель, А.Г. Кочан, А.М. Тараторин. – М.: Радио и связь, 1989.
8. Методические указания к выполнению курсовой работы «Вычислительная техника, программирование и математическое моделирование» для студентов-заочников машиностроительных специальностей. – Мн.: БГПА, 1994.
9. Справочное пособие по приближенным методам решения задач высшей математики / Л.И. Бородич, А.И. Герасимович. – М.: Высш. шк., 1986.

### Дополнительная

1. Задачи и упражнения по математическому анализу / Под ред. Б.П. Демидовича. – М.: Наука, 1978.
2. Корн Г., Корн Т. Справочник по математике для научных работников и инженеров. – М.: Наука, 1984.
3. Фурунжиев Р.И., Бабушкин Ф.М., Варавко В.В. Применение математических методов и ЭВМ: Практикум: Учеб. пособ. для вузов. – Мн.: Высш. шк., 1988.
4. Туркина Е.П. Математическая обработка данных с помощью пакета MathCad: Сб. лабор. работ для студ. эконо. специальностей. – Мн.: БГЭУ, 2002.

## ПРИЛОЖЕНИЕ

### РЕШЕНИЕ СЛАУ СРЕДСТВАМИ MATHCAD

MathCad – это программная среда для выполнения на компьютере разнообразных математических и технических расчетов, предоставляющая пользователю инструменты для работы с формулами, числами, графиками и текстами, снабженная простым в освоении графическим интерфейсом.

Окно программы имеет стандартный для Windows-приложений вид. Строка меню содержит следующие пункты: *Файл, Редактирование, Вид, Вставка, Формат, Инструменты, Символы, Окно, Справка*.

Включение необходимых панелей инструментов производится в *Вид-Панели инструментов*.

#### Решение с помощью функции *Isolve()*

1. Задаем матрицу системы *A* и матрицу свободных членов *B*. Для этого пишем *A*, вводим двоеточие (которое автоматически заменится на знак присваивания «:=»), затем на панели инструментов для работы с матрицами и векторами нажимаем кнопку создания матрицы или вектора  $\begin{bmatrix} \vdots \\ \vdots \\ \vdots \end{bmatrix}$  и вводим матрицу системы. Аналогично вводим столбец свободных членов.

2. Вводим  $\text{Isolve}(A,B)=$

**Пример:**

$$A := \begin{pmatrix} 1 & 2 & 3 \\ 9 & -3 & 2 \\ 2 & 5 & 1 \end{pmatrix} \quad B := \begin{pmatrix} 7 \\ 5 \\ 3 \end{pmatrix}$$

$$\text{Isolve}(A,B) = \begin{pmatrix} 0.1 \\ 0.115 \\ 2.223 \end{pmatrix}$$

#### Решение с помощью функции *Given...Find*

1. Вводим начальные приближения:  $x1:=1, x2:=1, x3:=1$ .

2. Вводим слово *Given*.

3. Ниже вводим систему уравнений, используя при этом вместо обычного знака равно знак булева равенства (вводится нажатием  $\text{Ctrl}+=$ ).

4. Ниже пишем:  $\text{Find}(x1,x2,x3)=$

**Пример:**

$$x1:=1 \quad x2:=1 \quad x3:=1$$

*Given*

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

$$9x1 - 3x2 + 2x3 = 5$$

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

$$\text{Find}(x1,x2,x3) = \begin{pmatrix} 0.1 \\ 0.115 \\ 2.223 \end{pmatrix}$$

### Решение СЛАУ методом обратной матрицы

1. Вводим матрицу системы и матрицу свободных членов, как и при решении с помощью `isolve()`.

2. Вводим  $x := A^{-1}B$ . Для ввода обозначения обратной матрицы нужно воспользоваться соответствующей кнопкой панели инструментов «Матрицы».

3. Вводим  $x =$

Пример:

$$A := \begin{pmatrix} 1 & 2 & 3 \\ 9 & -3 & 2 \\ 2 & 5 & 1 \end{pmatrix} \quad B := \begin{pmatrix} 7 \\ 5 \\ 3 \end{pmatrix}$$

$$x := A^{-1}B$$

$$x = \begin{pmatrix} 0.1 \\ 0.115 \\ 2.223 \end{pmatrix}$$

## РЕШЕНИЕ НЕЛИНЕЙНЫХ УРАВНЕНИЙ С ПОМОЩЬЮ MATHCAD

### Локализация корней

Предположим, нам нужно уточнить корни уравнения  $f(x) := x \cos(x)$ .

Для локализации корней удобно построить график функции. Для этого необходимо:

- 1) определить функцию, т.е. ввести  $f(x) := x \cos(x)$ ;
- 2) выбрать *Вставить-Графики-Зависимость XY*;
- 3) под осью абсцисс ввести  $x$ , слева от оси ординат ввести  $f(x)$ ;
- 4) при необходимости откорректировать пределы изменения аргумента и функции.

Основные свойства графика можно настроить, выбрав пункт *Формат* в контекстном меню, вызываемом щелчком правой кнопкой мыши на графике.

В результате анализа графика определяются начальные приближения корней уравнения.

### Уточнение корней

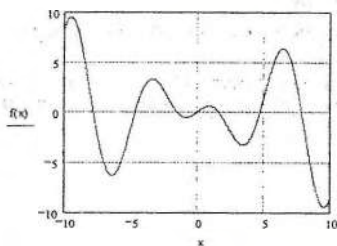
Уточнение корней производится с помощью функции `root()`:

1) устанавливаем точность вычислений, изменяя значения пункта *Порог сложности* на закладке *Толерантность* окна, вызываемого командой *Формат-Результат*. На закладке *Формат номера* того же окна выбираем способ отображения результата и количество отображаемых десятичных знаков;

2) вводим `root(f(x),x)=`

Пример:

$$f(x) := x \cos(x)$$



$X := -7$

$$\text{root}(f(x), x) = -7.854 \times 10^0$$

## ИНТЕРПОЛИРОВАНИЕ ФУНКЦИЙ СРЕДСТВАМИ MATHCAD

В MathCad имеются средства для сплайновой интерполяции.

Интерполирование производится с помощью функции  $\text{interp}(v_x, v_y, x)$ , которая возвращает вектор значений функции  $y$  в интересующих точках  $x$ .

Здесь и далее  $v_x$  – вектор исходных значений аргумента,  $v_y$  – вектор исходных значений функции,  $vs$  – вектор вторых производных, возвращаемых одной из функций  $\text{cspline}(v_x, v_y)$ ,  $\text{pspline}(v_x, v_y)$ ,  $\text{lspline}(v_x, v_y)$ , которые возвращают вектор  $vs$  при использовании кубических, параболических и линейных сплайнов соответственно.

Для интерполирования линиями можно воспользоваться функцией  $\text{interp}(v_x, v_y, x)$ .

Конечно, используя вычислительные возможности MathCad, всегда можно воспользоваться другим методом интерполяции (например, с помощью полиномов Ньютона и Лагранжа).

### Пример.

Имеются экспериментально полученные точки:

Значение аргумента $x$	Значение функции $y$
-1.06	1.22
-0.837	0.854
-0.684	0.513
-0.315	0.271
-0.117	0.217
-0.0	0.198
0.115	0.218
0.5	0.277

Построить график функции  $y(x)$ , используя точки, отличные от экспериментально полученных.

Решение

$$ExX := \begin{pmatrix} -1.06 \\ -0.837 \\ -0.684 \\ -0.315 \\ -0.117 \\ 0 \\ 0.115 \\ 0.5 \end{pmatrix} \quad ExY := \begin{pmatrix} 1.22 \\ 0.854 \\ 0.513 \\ 0.271 \\ 0.217 \\ 0.198 \\ 0.218 \\ 0.277 \end{pmatrix}$$

vs := cspline (ExX, ExY)

n := 100

i := 0..n

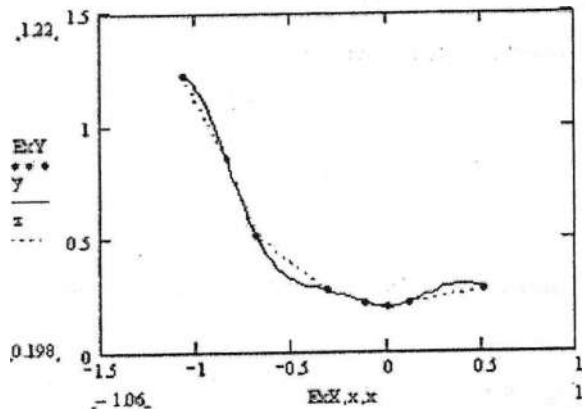
$$h := \frac{ExX_7 - ExX_0}{n}$$

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

z := linterp (ExX, ExY, x)

y := interp (vs, ExX, ExY, x)

	D
0	1.22
1	1.214
2	1.203
3	1.189
4	1.171
5	1.15
6	1.127
7	1.1
8	1.072
9	1.041
10	1.008
11	0.974
12	0.939
13	0.902
14	0.865
15	0.827



## АППРОКСИМАЦИЯ ЗАВИСИМОСТЕЙ С ПОМОЩЬЮ MATHCAD

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

$\text{regress}(vx, vy, k)$  – возвращает вектор, который использует функция  $\text{interp}$  для нахождения полинома степени  $k$ , который наилучшим образом приближает значения  $x$  и  $y$  данных, хранящихся в векторах  $vx, vy$ .

$\text{interp}(vs, vx, vy, x)$  – возвращает приближенное значение  $y$ , соответствующее значению  $x$ , где  $vs$  – вектор, получаемый с помощью  $\text{regress}$ .

### Пример.

Имеются экспериментально полученные точки:

Значение аргумента	Значение функции
$x$	$y$
0	0
1	1,5
2	3,7
3	10
4	15

Определить вид аппроксимирующего полинома 4-й степени и с его помощью построить график функции  $y(x)$ .

### Решение.

Решим задачу методом наименьших квадратов, а затем с помощью встроенных функций MathCad.

Введем исходные данные

$$ExX := \begin{pmatrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{pmatrix} \quad ExY := \begin{pmatrix} 0 \\ 1.5 \\ 3.7 \\ 10 \\ 15 \end{pmatrix}$$

Вычислим коэффициенты системы уравнений

$$\begin{aligned} m &:= 1 & n &:= 4 & i &:= 0..n \\ t_0 &:= n + 1 & t_1 &:= \sum_i ExX_i & t_2 &:= \sum_i (ExX_i)^2 \\ t_3 &:= \sum_i (ExX_i)^3 & t_4 &:= \sum_i (ExX_i)^4 \\ t_5 &:= \sum_i (ExX_i)^5 & t_6 &:= \sum_i (ExX_i)^6 & t_7 &:= \sum_i (ExX_i)^7 & t_8 &:= \sum_i (ExX_i)^8 \end{aligned}$$

$$C_0 := \sum_i ExY, \quad C_1 := \sum_i ExX_i ExY, \quad C_2 := \sum_i (ExX_i)^2 ExY,$$

$$C_3 := \sum_i (ExX_i)^3 ExY, \quad C_4 := \sum_i (ExX_i)^4 ExY,$$

Определим матрицу коэффициентов и матрицу свободных членов системы:

$$M := \begin{pmatrix} t_0 & t_1 & t_2 & t_3 & t_4 \\ t_1 & t_2 & t_3 & t_4 & t_5 \\ t_2 & t_3 & t_4 & t_5 & t_6 \\ t_3 & t_4 & t_5 & t_6 & t_7 \\ t_4 & t_5 & t_6 & t_7 & t_8 \end{pmatrix} \quad K := \begin{pmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \\ C_4 \end{pmatrix}$$

Найдем коэффициенты полинома, решив систему:

$$A := \text{solve}(M, K)$$

$$A = \begin{pmatrix} 1.32 \times 10^{-13} \\ 4.483 \\ -5.383 \\ 2.767 \\ -0.367 \end{pmatrix}$$

Определим вид полинома:  $P(x) := \sum_{i=0}^4 A_i x^i$

Построим график  $P(x)$ :

$$n := 10$$

$$i := 0..n$$

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

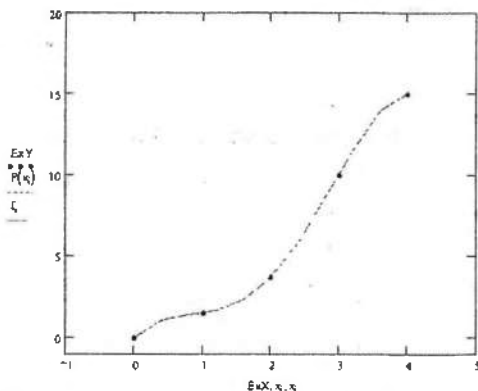
$$x_i := ExX_0 + i \cdot h$$

Во всех точках  $x$  для сравнения вычислим значения функции с помощью встроенных средств аппроксимации:

$$vs := \text{regress}(ExX, ExY, 4)$$

$$f_1 := \text{interp}(vs, ExX, ExY, x_i)$$

Строим графики



Как видно, графики совпадают.

## ВЫЧИСЛЕНИЕ ОПРЕДЕЛЕННОГО ИНТЕГРАЛА И ПРОИЗВОДНОЙ СРЕДСТВАМИ MATHCAD

Предположим, нужно вычислить интеграл функции  $f(x) = e^x \cos(x)$  на отрезке  $[0; 2]$  и производную этой функции в точке  $x=1$ . Решение такой задачи в Math-Cad имеет вид:

$$F(x) := e^x \cos(x)$$

$$a := 0 \quad b := 2$$

$$\int_a^b F(x) dx = 1.322$$

$$x := 1$$

$$\frac{d}{dx} F(x) = -0.819$$

Интеграл может быть кратным, подынтегральная функция может быть комплексной или функцией нескольких переменных.

Для вычисления производных более высоких порядков используется специальная конструкция:

$$\frac{d^n}{da^n}$$

## РЕШЕНИЕ ОДУ СРЕДСТВАМИ MATHCAD

Для решения линейных обыкновенных дифференциальных уравнений в MathCad функция  $odesolve(x, b, \{step\})$ , которая возвращает 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.

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

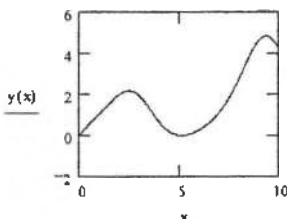
**Решение:**

Given

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

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

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



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

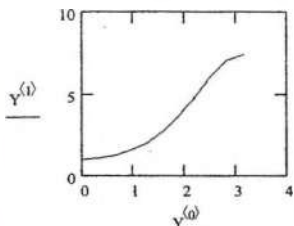
Решение ОДУ первого порядка вида  $y' = f(x, y)$  может быть получено с помощью функции  $rkfixed(y_0, a, b, n, D)$ , которая возвращает матрицу, состоящую из двух столбцов. В первом столбце хранятся значения аргумента, во втором – функции (результаты решения). Здесь  $y_0$  – начальное значение функции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

**Пример.** Решим уравнение  $y' = \sin xy$  на отрезке  $[0, \pi]$ . Известно, что  $y(0)=1$ .

**Решение:**

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

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



	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

Для решения ОДУ первого порядка также можно использовать функцию  $gkadapt()$ , аналогичную рассмотренной выше  $rkfixed()$ , за исключением того, что решение находится не с фиксированным шагом, а с автоматическим его подбором.