1. Аппроксимация дискретных данных методом наименьших квадратов

1.1 Цель работы

Целью работы является получение навыков аппроксимации дискретных данных полиномом и исследование погрешностей аппроксимации в зависимости от степени полинома.

1.2 Задание

1. В соответствии с номером варианта задания аппроксимировать таблично заданные значения неизвестной функции методом наименьших квадратов полиномами 1, 2, 3 и 4 степеней. При этом:
   - Для полиномов 1, 2, 3 и 4 степеней найти их коэффициенты путем решения систем линейных уравнений.
   - Для полиномов 1 и 2 степени также найти коэффициенты с помощью функции ***regress***.
   - Для полинома 3-й степени также найти коэффициенты с помощью функции ***linfit***.
   - Для полинома 4-й степени также найти коэффициенты с помощью функции ***interp***.
2. Построить совмещенный график заданных узлов и всех полиномов.
3. Построить график суммы квадратов отклонений полиномов во всех узлах в зависимости от степени полинома.

Варианты заданий

1	x_i	**0**	**0,6**	**1,2**	**1,8**	**2,4**	**3**
	y_i	**2,0**	**1,9**	**4,1**	**8,1**	**6,2**	**-5,0**
2	x_i	**0**	**0,8**	**1,6**	**2,4**	**3,2**	**4**
	y_i	**3,0**	**-0,8**	**-4,0**	**-0,4**	**3,0**	**0,3**
3	x_i	**0**	**0,6**	**1,2**	**1,8**	**2,4**	**3**
	y_i	**1,0**	**-0,3**	**-1,9**	**1,7**	**1,6**	**-1,3**
4	x_i	**0**	**0,8**	**1,6**	**2,4**	**3,2**	**4**
	y_i	**0**	**6,4**	**4,6**	**3,2**	**10,1**	**16,0**
5	x_i	**0**	**0,6**	**1,2**	**1,8**	**2,4**	**3**
	y_i	**4,0**	**5,3**	**5,2**	**2,9**	**0,7**	**2,7**
6	x_i	**0**	**0,6**	**1,2**	**1,8**	**2,4**	**3**
	y_i	**4,0**	**4,0**	**5,0**	**8,0**	**9,4**	**2,5**
7	x_i	**0**	**0,8**	**1,6**	**2,4**	**3,2**	**4**
	y_i	**8,0**	**9,8**	**0,9**	**0**	**0**	**0,1**
8	x_i	**0**	**0,4**	**0,8**	**1,2**	**1,6**	**2**
	y_i	**7,0**	**8,9**	**6,4**	**2,5**	**4,5**	**16,1**

9	x_i	**0**	**0,6**	**1,2**	**1,8**	**2,4**	**3**
	y_i	**0**	**1,4**	**3,7**	**4,3**	**1,9**	**19,3**
10	x_i	**0**	**0,8**	**1,6**	**2,4**	**3,2**	**4**
	y_i	**2,0**	**2,3**	**2,8**	**7,4**	**11,6**	**11,7**
11	x_i	**0**	**0,6**	**1,2**	**1,8**	**2,4**	**3**
	y_i	**0**	**3,1**	**10,1**	**16,0**	**15,1**	**4,0**
12	x_i	**0**	**0,6**	**1,2**	**1,8**	**2,4**	**3**
	y_i	**1,0**	**2,0**	**1,6**	**0,3**	**7,6**	**20,0**
13	x_i	**0**	**0,6**	**1,2**	**1,8**	**2,4**	**3**
	y_i	**0**	**0,5**	**2,7**	**5,2**	**5,3**	**1,4**
14	x_i	**0**	**0,6**	**1,2**	**1,8**	**2,4**	**3**
	y_i	**0**	**3,8**	**3,4**	**3,2**	**4,3**	**5,9**
15	x_i	**0**	**0,8**	**1,6**	**2,4**	**3,2**	**4**
	y_i	**3,0**	**10,8**	**1,7**	**0**	**1,2**	**5,3**
16	x_i	**0**	**0,6**	**1,2**	**1,8**	**2,4**	**3**
	y_i	**0**	**5,3**	**3,4**	**1,8**	**11,0**	**1,1**
17	x_i	**0**	**0,8**	**1,6**	**2,4**	**3,2**	**4**
	y_i	**4,0**	**1,0**	**4,0**	**2,9**	**3,9**	**4,7**
18	x_i	**0**	**0,6**	**1,2**	**1,8**	**2,4**	**3**
	y_i	**0**	**1,6**	**4,6**	**8,9**	**10,5**	**3,4**
19	x_i	**0**	**0,8**	**1,6**	**2,4**	**3,2**	**4**
	y_i	**0**	**0,4**	**0**	**1,3**	**6,8**	**7,3**
20	x_i	**0**	**0,8**	**1,6**	**2,4**	**3,2**	**4**
	y_i	**0**	**1,4**	**3,5**	**6,4**	**8,5**	**6,9**

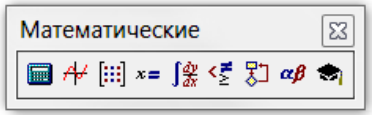
1.3 Теоретические сведения

MathCAD является математическим редактором, позволяющим проводить разнообразные инженерные и научные расчеты, начиная от элементарной арифметики и заканчивая сложными реализациями численных методов.

MathCAD дает возможность:

1. Вычисление численными методами решений уравнений, систем уравнений и неравенств.
2. Вычисление производных и интегралов.
3. Вычисление сумм рядов, произведений.
4. Действия с векторами и матрицами, включая операции матричного умножения, обращения матриц, транспонирование, вычисление определителя матрицы, скалярное и векторное умножение.
5. Проводить символьное решение уравнений, интегрирование и дифференцирование.
6. Разложение выражений на множители и приведение к простейшему виду.

7. Построение графиков различных видов и многое другое.



Панель **Математика** предназначена для вызова на экран еще девяти панелей, с помощью которых, собственно, и происходит вставка математических операций в документы. Чтобы показать какую-либо из них, нужно нажать соответствующую кнопку на панели **Математика.**

1.3.1 Ввод формул

Ввод информации осуществляется в месте расположения курсора. Используются три вида курсоров. Если ни один объект не выбран, используется красный **крестообразный курсор**, определяющий место создания следующего объекта. При вводе формул используется **уголковый курсор**, указывающий текущий элемент выражения. При вводе данных в текстовый блок применяется **текстовый курсор** в виде вертикальной черты.

Формулы – основные объекты рабочего листа. Новый объект по умолчанию является формулой. Чтобы начать ввод формулы, надо установить курсор в нужное место и начать ввод букв, цифр, знаков операций. При этом создается область формулы, в которой появляется уголковый курсор, охватывающий текущий элемент формулы. При вводе **бинарного оператора** по другую сторону знака операции автоматически появляется заполнитель в виде черного прямоугольника. В это место вводят очередной операнд. Местозаполнитель символа — черный прямоугольник; местозаполнитель оператора — черная прямоугольная рамка.

Чтобы изменить формулу, щелкните на ней мышью, поместив таким образом в ее область линии ввода, и перейдите к месту, которое хотите исправить.

1.3.2 Повторяющиеся вычисления

Программа может выполнять повторяющиеся или итерационные вычисления. С этой целью используется специальный тип переменных – дискретные аргументы.

Переменная типа дискретный аргумент принимает диапазон значений. Если в выражении присутствует дискретный аргумент, выражение вычисляется столько раз, сколько значений содержит дискретный аргумент.

Для того чтобы вычислить выражение для диапазона значений сначала надо определить дискретный аргумент. Фактически он представляет собой арифметическую прогрессию. Поэтому надо задать первый, второй и последний члены прогрессии. Если шаг прогрессии равен **1**, то второй член прогрессии можно не задавать.

Примеры задания дискретной переменной с шагом отличным от 1 и с шагом равным 1:

a := 1, 1.5 .. 5

a =
1
1.5
2
2.5
3
3.5
4
4.5
5

b := 1 .. 9

b =
1
2
3
4
5
6
7
8
9

Для задания дискретной переменной удобно пользоваться командой  с панели **Матрица**.

1.3.3 Задание функции

При задании функции после ее имени в скобках перечисляются все независимые переменные.

Пример задания функции одной переменной:

$$y(x) := x^2 + 3x - 4$$

Для вычисления значения функции при конкретном значении независимой переменной, надо обратиться к имени функции указав в скобках числовое значение независимой переменной.

Пример вычисления значения функции при **x**=3:

$$y(3) = 14$$

1.3.4 Вектора и матрицы

Создаются вектора и матрицы с помощью команды  на панели **Матрица**. В диалоговом окне задается число строк и столбцов.

Внимание! Нумерация строк и столбцов матриц и векторов по умолчанию начинается с нуля.

Стартовый индекс массива задается системной переменной **ORIGIN**, которая по умолчанию равна нулю. Если нужно нумеровать элементы векторов и матриц с единицы, присвойте этой переменной значение 1. В этом случае попытка выяснить значение нулевого элемента приводит к ошибке, поскольку его значение не определено.

Можно обращаться к отдельным элементам массива (вектора или матрицы), используя нижние индексы у обозначения массива. Можно также обращаться к отдельному столбцу массива, используя верхний индекс. Для ввода нижнего индекса используется кнопка  на панели **Матрица**, верхнего индекса – кнопка  .

$$M := \begin{pmatrix} 1 & 3 & 5 & 7 \\ 9 & 2 & 4 & 6 \\ 8 & 0 & 10 & 11 \end{pmatrix} \qquad M_{2,3} = 11 \qquad M^{\langle 1 \rangle} = \begin{pmatrix} 3 \\ 2 \\ 0 \end{pmatrix}$$

Транспонирование

Транспортированием называют операцию, переводящую матрицу размерности $M \times N$ в матрицу размерности $N \times M$, делая столбцы исходной матрицы строками, а строки — столбцами. Ввод символа транспонирования M^T осуществляется с помощью панели инструментов **Матрица**.

$$\begin{pmatrix}1\\2\\3\end{pmatrix}^T = (1\ \ 2\ \ 3) \qquad \begin{pmatrix}1 & 2 & 3\\4 & 5 & 6\end{pmatrix}^T = \begin{pmatrix}1 & 4\\2 & 5\\3 & 6\end{pmatrix}$$

Определитель квадратной матрицы

Определитель матрицы обозначается стандартным математическим символом. Чтобы ввести оператор нахождения определителя матрицы можно нажать кнопку $|x|$ на панели инструментов **Матрица**. В результате любого из этих действий появляется местозаполнитель, в который следует поместить матрицу. Чтобы вычислить определитель уже введенной матрицы нужно: переместить курсор в документе таким образом, чтобы поместить матрицу между линиями ввода, ввести оператор нахождения определителя матрицы и ввести знак равенства, чтобы вычислить определитель.

$$\left|\begin{pmatrix}1 & 0 & 3\\5 & 4 & 1\\8 & 6 & 9\end{pmatrix}\right| = 24$$

$$A := \begin{pmatrix}1 & 0 & 3\\5 & 4 & 1\\8 & 6 & 9\end{pmatrix} \qquad |A| = 24$$

Разбиение и слияние матриц

Из матрицы или вектора можно выделить либо подматрицу, либо вектор-столбец, либо отдельный элемент. И обратно, можно соединить несколько матриц в одну.

Выделение подматрицы производится с помощью встроенной функции:

submatrix(A, ir, jr, ic, jc),

где: **A**- имя исходной матрицы или сама матрица; **ir** и **ic** – начальный номер строки и колонки; **jr** и **jc** – конечный номер строки и колонки.

$$A := \begin{pmatrix} 2 & 4 & 6 \\ 8 & 1 & 3 \end{pmatrix}$$

$$\text{submatrix}(A,0,1,0,1) = \begin{pmatrix} 2 & 4 \\ 8 & 1 \end{pmatrix}$$

$$\text{submatrix}\left[\begin{pmatrix} 2 & 4 & 6 \\ 8 & 1 & 3 \end{pmatrix},0,1,0,1\right] = \begin{pmatrix} 2 & 4 \\ 8 & 1 \end{pmatrix}$$

Для того чтобы составить из двух или более матриц одну, предусмотрены две матричные функции:

- $\text{augment}(A,B,■,■)$ - матрица, сформированная слиянием указанных в скобках матриц слева направо;
- $\text{stack}(A,B,■,■)$ - матрица, сформированная слиянием указанных в скобках матриц сверху вниз.

$$A := \begin{pmatrix} 2 & 4 & 6 \\ 8 & 1 & 3 \end{pmatrix} \qquad B := \begin{pmatrix} 0 & 1 & 2 \\ 3 & 4 & 5 \end{pmatrix}$$

$$\text{augment}(A,B) = \begin{pmatrix} 2 & 4 & 6 & 0 & 1 & 2 \\ 8 & 1 & 3 & 3 & 4 & 5 \end{pmatrix} \qquad \text{stack}(A,B) = \begin{pmatrix} 2 & 4 & 6 \\ 8 & 1 & 3 \\ 0 & 1 & 2 \\ 3 & 4 & 5 \end{pmatrix}$$

Вывод размера матриц

Для получения сведений о характеристиках матриц или векторов предусмотрены следующие встроенные функции:

- $\text{rows}(A)$ - число строк;
- $\text{cols}(A)$ - число столбцов.

$$A := \begin{pmatrix} 2 & 4 & 6 \\ 8 & 1 & 3 \end{pmatrix}$$

$\text{rows}(A) = 2$ $\quad \text{cols}(A) = 3$

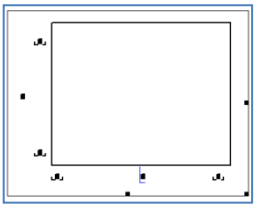
$n := \text{rows}(A)$ $\quad m := \text{cols}(A)$

$n = 2$ $\quad m = 3$

1.3.5 Создание графиков в декартовых координатах

Графики можно создавать как для аналитически заданной функции, так и для таблично заданной.

Чтобы построить график в декартовых координатах, нужно щелкнуть мышью на том свободном месте, где его нужно разместить и выбрать команду  на панели инструментов **График**.

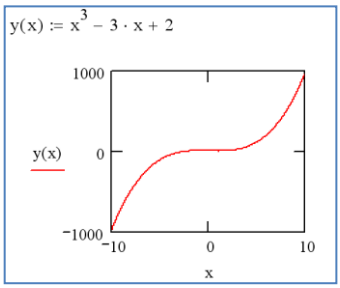


Появится пустой график с полями ввода для выражений, отображаемых по осям графика (одно из них отмечено уголковым курсором). Ввести в местозаполнители имена переменных или функций, которые должны быть изображены на графике. Внизу указывается имя независимой переменной, слева – имя функции или сама функция. Кроме того, по краям осей находятся местозаполнители для задания пределов изменения независимой переменной и функции. Если оставить их пустыми, программа автоматически заполнит их при создании графика. При этом для независимой переменной предел изменения берется от -10 до 10, а для функции – в соответствии с функциональной зависимостью (максимальное и минимальное значение функции для данного диапазона значений независимой переменной).

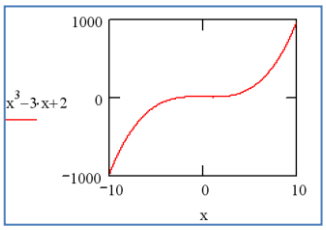
График появится на экране после щелчка на свободном месте листа. Под именем функции появляется образец линии, которой нарисован график.

Созданный график можно изменить, в том числе меняя сами данные, форматируя его внешний вид или добавляя дополнительные элементы оформления.

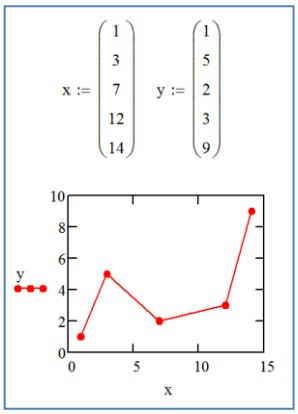
Пример. Построение графика с помощью предварительного задания функции.



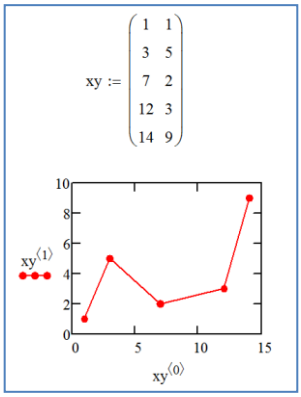
Пример. Построение графика с помощью непосредственного задания функции.



Пример. Построение графика таблично заданной функции (исходные данные заданы в виде двух векторов). При этом в местозаполнителях указываются имена соответствующих векторов.



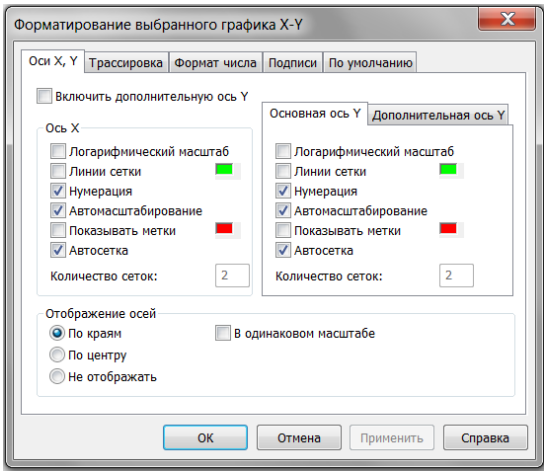
Пример. Построение графика таблично заданной функции (исходные данные заданы в виде матрицы с двумя столбцами: в первом столбце значения независимой переменной, во втором – значения функции). При этом в местозаполнителях указываются имя матрицы и номер соответствующего столбца.



В последних двух примерах маркерами отмечены заданные значения.

Форматирование графика

График обладает некоторыми свойствами, установленными по умолчанию. К ним относятся: деления по осям, отсутствие линий сетки и сплошная линия графика. Их можно изменить, форматируя график. Для этого дважды щелкнуть левой клавишей мыши в пределах графика. Появится диалоговое окно **Форматирование выбранного графика X-Y**, в котором следует перейти на вкладку **Оси X, Y**.



С помощью флажков и переключателей легко поменять внешний вид каждой из осей. Доступные в этом окне опции и их действие:

Логарифмический масштаб — график по данной оси будет нарисован в логарифмическом масштабе. Это удобно, если данные разнятся на несколько порядков.

Линии сетки — показать линии сетки (при этом вокруг графика появляется прямоугольная рамка, выполненная определенным цветом).

Нумерация — показать нумерацию шкалы. Если убрать этот флажок, то числа, размечающие шкалу, пропадут.

Автомасштабирование — выбор диапазона оси производится автоматически.

Показывать метки — выделение определенных значений на осях (не более двух на каждой оси).

Автосетка — разбиение шкалы производится автоматически. Если этот флажок снят, в поле ввода рядом с ним следует указать желаемое количество меток шкалы.

В одинаковом масштабе — оси **X** и **Y** принудительно рисуются в одинаковом масштабе.

По краям — график рисуется в прямоугольной рамке (как показано выше).

По центру — координатные оси в виде двух пересекающихся прямых.

Не отображать — координатные оси не показываются на графике.

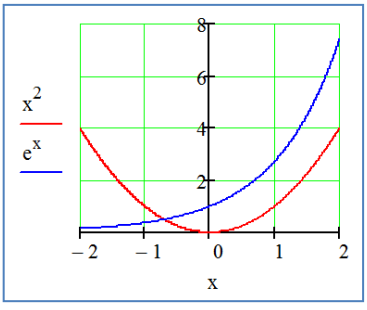
Изменить параметры осей можно и в диалоговом окне **Формат оси**, которое появляется, если щелкнуть дважды на самой оси.

С помощью вкладки **Трассировка** легко установить комбинацию параметров линии и точек для каждого из рядов данных, представленных на графике. Для этого требуется выделить в списке нужный ряд данных (его положение в списке соответствует положению метки зависимости у оси **Y**) и изменить в списках в середине диалогового окна параметры отображения.

Можно построить несколько графиков в одних осях. Графики могут содержать несколько выражений по оси ординат в зависимости от одного выражения по оси абсцисс или нескольких выражений по оси ординат, согласованных с соответствующими выражениями по оси абсцисс.

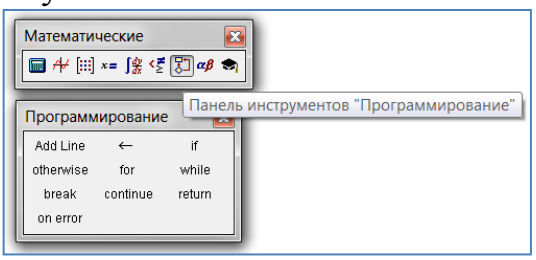
Чтобы представить графически несколько выражений по оси ординат относительно одного выражения по оси абсцисс, ввести первое выражение для оси ординат, сопровождаемое запятой. Под первым выражением появится пустое поле для ввода второго выражения. Поставив после второго выражения запятую, можно получить пустое поле для ввода следующего выражения и т.д.

Например, построим в одних осях графики функций $y = x^2$ и $y = e^x$.



1.3.6 Программирование в MathCAD

Для вставки программного кода в документы в **Mathcad** имеется специальная панель инструментов **Программирование**. Большинство кнопок этой панели выполнено в виде текстового представления операторов программирования, поэтому их смысл легко понятен.

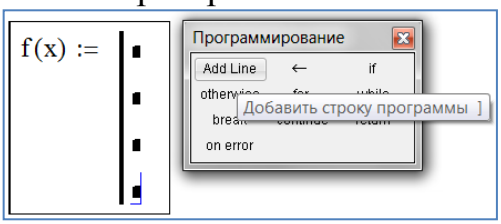


Программный модуль обозначается в **Mathcad вертикальной чертой**, справа от которой последовательно записываются операторы языка программирования.

1.3.6.1 Создание программы (Add Line).

Чтобы создать программный модуль:

- Ввести имя функции и знак присваивания.
- Нажать на панели **Программирование** кнопку **Add Line** (Добавить линию). Появляется вертикальная черта и два местозаполнителя.
- Если приблизительно известно, сколько строк кода будет содержать программа, можно создать нужное количество линий повторным нажатием кнопки **Add Line** (при этом каждый раз добавляется один местозаполнитель) соответствующее число раз. Ниже показан результат трехкратного нажатия.
- В появившиеся местозаполнители ввести желаемый программный код, используя программные операторы.



После того как программный модуль полностью определен и ни один местозаполнитель не остался пустым, функция может использоваться обычным образом, как в численных, так и в символьных расчетах.

Не вводите с клавиатуры имена операторов. Для их вставки пользуйтесь панелью Программирование.

Вставить строку программного кода в уже созданную программу можно в любой момент с помощью той же самой кнопки **Add Line**. Для этого следует предварительно поместить на нужное место внутри программного модуля курсор ввода.

1.3.6.2 Оператор локального определения (←).

Язык программирования позволяет создавать внутри программных модулей локальные переменные, которые "не видны" извне, из других частей документа. Присваивание значения переменной производится с помощью оператора

Локальное определение, который вставляется нажатием кнопки с изображением стрелки ←.

Ни оператор присваивания «:=», ни оператор вывода «=» в пределах программ не применяются.

$f(x) := \left\| \begin{array}{l} z \leftarrow 4 \\ z + x \end{array} \right.$ $f(1) = 5$	Локальное присваивание иллюстрируется примером слева. Переменная **z** существует только внутри программы, выделенной вертикальной чертой. Из других мест документа получить ее значение невозможно.

1.3.6.3 Условный оператор (if, otherwise).

В условном операторе **if** сначала проверяется логическое выражение (условие) справа от него. Если оно истинно, выполняется выражение слева от оператора **if**. Если оно ложно, выполнение программы продолжается переходом к следующей строке.

Оператор **otherwise** используется совместно с оператором **if** и указывает на выражение, которое будет выполняться, если проверяемое условие не выполняется.

Пример 1. Задать функцию $f(x) = |x - 2|$.

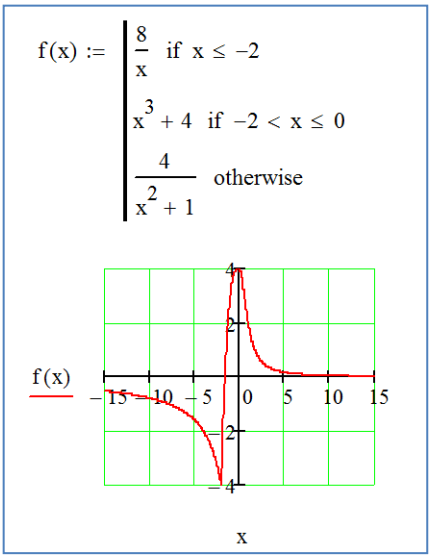
$$f(x) := \left\| \begin{array}{ll} x - 2 & \text{if } x \geq 2 \\ 2 - x & \text{if } x < 2 \end{array} \right.$$

$f(1) = 1 \qquad f(5) = 3$

$f(2) = 0$

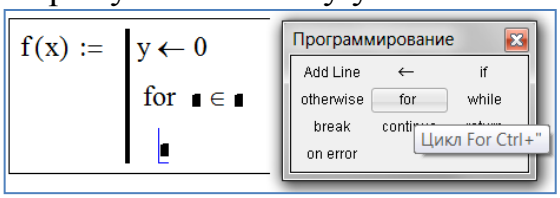
Пример 2. Построить график функции $f(x) = \begin{cases} \dfrac{8}{x} & \text{если } x \leq -2 \\ x^3 + 4 & \text{если } -2 < x \leq 0 \\ \dfrac{4}{x^2 + 1} & \text{если } x > 0 \end{cases}$ на промежутке $x \in [-15;15]$.

$$f(x) := \left\| \begin{array}{ll} \dfrac{8}{x} & \text{if } x \leq -2 \\ x^3 + 4 & \text{if } -2 < x \leq 0 \\ \dfrac{4}{x^2 + 1} & \text{otherwise} \end{array} \right.$$



1.3.6.4 Операторы цикла (for, while).

В языке программирования имеются два оператора цикла: **for** и **while**. Первый из них дает возможность организовать цикл по некоторой переменной, заставляя ее пробегать заданный диапазон значений. Второй создает цикл с выходом из него по некоторому логическому условию.



Диапазон значений переменной в условии цикла **for** можно задать как с помощью дискретной переменной, так и с помощью вектора.

Пример 1. Оператор цикла **for** с дискретной переменной.

```
x := | y ← 0
     | for i ∈ 0..5
     |   y ← y + i

x = 15
```

Пример 2. Оператор цикла **for** с вектором.

```
x := | y ← 0
     | for i ∈ (1  2  3)
     |   y ← y + i

x = 6
```

Пример 3. Оператор цикла **while.**

```
x := | y ← 0
     | while y < 7
     |   y ← y + 2

x = 8
```

Иногда необходимо досрочно завершить цикл, т. е. не по условию в его заголовке, а в некоторой строке в теле цикла. Для этого предназначен оператор **break**.

Пример 4. Оператор **break** внутри цикла **for.**

```
x := | y ← 0
     | for i ∈ 0..5
     |   | y ← y + i
     |   | break if i = 2

x = 3
```

Пример 5. Оператор **break** внутри цикла **while**.

```
x := | y ← 0
     | while y < 7
     |     y ← y + 2
     |     break if y > 5
     x = 6
```

Чтобы четче обозначить границы завершения тела цикла, в его конце может использоваться дополнительная строка с оператором **continue**.

Пример. Программа нахождения вещественных корней квадратного уравнения.

В имени программы в скобках задаются коэффициенты квадратного уравнения. Вначале программы определяется дискриминант квадратного уравнения. Если он меньше нуля, выдается сообщение об отсутствии действительных корней у уравнения и работа программы завершается. Если дискриминант не отрицательный, вычисляются корни уравнения (один или два).

```
Prog(a,b,c) := | discr ← b^2 − 4·a·c
               | if discr < 0
               |     res ← "нет вещественных корней"
               |     return res
               | res ← −b/(2·a)  if discr = 0
               | otherwise
               |     res_0 ← (−b + √discr)/(2·a)
               |     res_1 ← (−b − √discr)/(2·a)
               | res
```

Примеры использования программы.

1. Уравнение $x^2+x+1=0$

$$\text{Prog}(1,1,1) = \text{"нет вещественных корней"}$$

2. Уравнение $x^2+2x+1=0$

$$\text{Prog}(1,2,1) = -1$$

3. Уравнение $x^2+3x+1=0$

$$\text{Prog}(1,3,1) = \begin{pmatrix} -0.382 \\ -2.618 \end{pmatrix}$$

1.3.7 Метод наименьших квадратов

Метод наименьших квадратов применяют, когда необходимо вывести формулу аппроксимирующей кривой, описывающей некоторую зависимость,

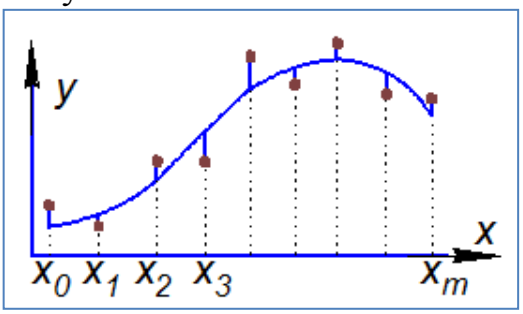
полученную в результате, например, эксперимента. Поскольку экспериментальные данные получают, как правило, с некоторой погрешностью, то обычно находят аппроксимирующую кривую не проходящую через экспериментальные точки. В то же время она учитывает исследуемую закономерность, сглаживая случайные выбросы результатов эксперимента. В качестве аппроксимирующей функции может использоваться линейная комбинация $F(x)$ любых функций:

$$F(x) = a_0 f_0(x) + a_1 f_1(x) + \ldots + a_n f_n(x)$$

где $f_0(x), f_1(x), \ldots f_n(x)$ - набор любых функций, называемых базисными; $a_0, a_1, \ldots, a_n$ - набор коэффициентов.

Довольно часто в качестве базисных функций используют комбинацию из степенной последовательности аргумента $f_0(x) = 1,\ f_1(x) = x, \ldots, f_n(x) = x^n$. При этом аппроксимирующая функция будет являться алгебраическим полиномом. Именно этот случай и рассматривается в работе.

Пусть некоторая функция задана таблицей значений y_i в узлах x_i $(i = 0,1,2,\ldots,m)$, где $(m+1)$ – количество узлов, а m – число интервалов разбиения заданного промежутка.



Требуется определить коэффициенты $a_0, a_1, \ldots, a_n$ полинома

$$P(x) = a_0 + a_1 x + \ldots + a_n x^n$$

таким образом, чтобы сумма S квадратов отклонений полинома $P(x)$ от значений y_i аппроксимируемой функции в заданных точках была бы минимальной, т.е.

$$S = \sum_{i=0}^{m} \left(a_0 + a_1 x_i + \ldots + a_n x_i^n - y_i\right)^2 \to \min$$

Если степень полинома $n = m$, то искомый полином будет интерполяционным и пройдет через все узлы и отклонения будут нулевыми. Стремясь понизить степень полинома, принимают $n < m$, при этом в общем случае аппроксимирующая кривая не будет проходить через заданные точки (x_i, y_i). Сумма квадратов отклонений зависит от коэффициентов полинома $a_0, a_1, \ldots, a_n$. Поскольку функция S принимает только положительные значения и имеет минимум, вычислим частные производные по всем переменным $a_0, a_1, \ldots, a_n$ и приравняем их нулю:

$$\frac{\partial S}{\partial a_i} = 0, \quad i = 0,1,2,\ldots,n$$

Получаем систему уравнений:

$$\begin{cases} \dfrac{\partial S}{\partial a_0} = 2\sum\limits_{i=0}^{m}\left(a_0 + a_1 x_i + \ldots + a_n x_i^n - y_i\right) = 0 \\ \dfrac{\partial S}{\partial a_1} = 2\sum\limits_{i=0}^{m}\left(a_0 + a_1 x_i + \ldots + a_n x_i^n - y_i\right)x_i = 0 \\ \ldots \\ \dfrac{\partial S}{\partial a_n} = 2\sum\limits_{i=0}^{m}\left(a_0 + a_1 x_i + \ldots + a_n x_i^n - y_i\right)x_i^n = 0 \end{cases}$$

Выполнив суммирование, собрав коэффициенты при каждом $a_0, a_1, \ldots, a_n$ и перенеся не содержащие их суммы в правую часть, получим СЛАУ относительно неизвестных $a_0, a_1, \ldots, a_n$:

$$\begin{cases} a_0(m+1) + a_1\sum\limits_{i=0}^{m} x_i + \ldots + a_n\sum\limits_{i=0}^{m} x_i^n = \sum\limits_{i=0}^{m} y_i \\ a_0\sum\limits_{i=0}^{m} x_i + a_1\sum\limits_{i=0}^{m} x_i^2 + \ldots + a_n\sum\limits_{i=0}^{m} x_i^{n+1} = \sum\limits_{i=0}^{m} x_i y_i \\ \ldots \\ a_0\sum\limits_{i=0}^{m} x_i^n + a_1\sum\limits_{i=0}^{m} x_i^{n+1} + \ldots + a_n\sum\limits_{i=0}^{m} x_i^{2n} = \sum\limits_{i=0}^{m} x_i^n y_i \end{cases}$$

В матричной форме эту систему можно записать как $a \cdot C = b$, где:

$$a = \begin{pmatrix} a_0 \\ a_1 \\ \ldots \\ a_n \end{pmatrix};\ C = \begin{pmatrix} m+1 & \sum\limits_{i=0}^{m} x_i & \ldots & \sum\limits_{i=0}^{m} x_i^n \\ \sum\limits_{i=0}^{m} x_i & \sum\limits_{i=0}^{m} x_i^2 & \ldots & \sum\limits_{i=0}^{m} x_i^{n+1} \\ \ldots & \ldots & \ldots & \ldots \\ \sum\limits_{i=0}^{m} x_i^n & \sum\limits_{i=0}^{m} x_i^{n+1} & \ldots & \sum\limits_{i=0}^{m} x_i^{2n} \end{pmatrix};\ b = \begin{pmatrix} \sum\limits_{i=0}^{m} y_i \\ \sum\limits_{i=0}^{m} x_i y_i \\ \ldots \\ \sum\limits_{i=0}^{m} x_i^n y_i \end{pmatrix}$$

Решение этой системы позволяет найти все коэффициенты $a_0, a_1, \ldots, a_n$.

В MathCAD для задания матрицы левой части и вектора правой части системы уравнений можно воспользоваться следующими программами.

```
C(n) := | for j ∈ 0..n
        |   for i ∈ 0..n
        |     C[i,j] ← Σ_{k=0}^{m} (x_k)^(i+j)
        | C
```

```
b(n) := | for i ∈ 0..n
        |   b_i ← Σ_{k=0}^{m} [y_k · (x_k)^i]
        | b
```

Для полиномиальной аппроксимации (регрессии) в **MathCAD**'е можно воспользоваться следующими встроенными функциями:

1. ***regress(x,y,k)*** – возвращает вектор коэффициентов ***a*** полинома, при этом всегда первые три компоненты вектора есть вектор вторых

производных и являются параметрами для описываемой ниже функции ***interp***, а остальные компоненты и есть вектор коэффициентов a, где x - вектор данных аргумента, элементы которого должны быть расположены в порядке возрастания; y - вектор значений того же размера; k - степень полинома (целое положительное число).

Функция ***regress*** позволяет, вычислив коэффициенты $a_0, a_1, \ldots, a_n$, построить полином и уже его использовать, например, для построения графика.

2. ***interp (s,x,y,t)*** – возвращает результат полиномиальной регрессии, где s - вектор вторых производных, созданный, например, предыдущей функцией; x и y - то же, что и в предыдущей функции; t - текущее значение аргумента полинома.

Функция ***interp*** позволяет непосредственно вычислить значение полинома при любом значении аргумента без построения полинома.

3. ***linfit(x,y,F)*** – более универсальная функция, возвращающая вектор коэффициентов a линейной комбинации функций, где x и y - то же, что и в предыдущей функции; F - вектор базисных функций $f_0(x), f_1(x), \ldots f_n(x)$. Например, для нахождения алгебраического полинома степени $n=2$ необходимо записать F в следующем виде:

$$F = \begin{pmatrix} 1 \\ x \\ x^2 \end{pmatrix}$$

Функция ***linfit*** также позволяет непосредственно вычислить значение полинома при любом значении аргумента без построения полинома.

1.4 Пример выполнения работы

Аппроксимировать таблично заданные значения неизвестной функции методом наименьших квадратов полиномами 1, 2, 3 и 4 степеней.

x_i	0	0,8	1,6	2,4	3,2	4
y_i	0	4,6	3,0	1,8	6,8	11,0

Вводим табличные данные в векторы узлов x и y и определяем количество промежутков.

$$x := \begin{pmatrix} 0 \\ 0.8 \\ 1.6 \\ 2.4 \\ 3.2 \\ 4 \end{pmatrix} \qquad y := \begin{pmatrix} 0 \\ 4.6 \\ 3.0 \\ 1.8 \\ 6.8 \\ 11.0 \end{pmatrix}$$

$$m := rows(x) - 1 = 5$$

Программы для нахождения матрицы коэффициентов левой части и вектора правой части системы уравнений:

$$C(n) := \left| \begin{array}{l} \text{for } j \in 0..n \\ \quad \text{for } i \in 0..n \\ \quad\quad C_{i,j} \leftarrow \sum_{k=0}^{m} (x_k)^{i+j} \\ C \end{array} \right.$$

$$b(n) := \left| \begin{array}{l} \text{for } i \in 0..n \\ \quad b_i \leftarrow \sum_{k=0}^{m} \left[y_k \cdot (x_k)^i \right] \\ b \end{array} \right.$$

Определяем коэффициенты и составляем полиномы для различных степеней ***n***.

Полином 1-й степени

Задание степени полинома:

$$n := 1$$

А) Нахождение коэффициентов аппроксимирующего полинома с помощью системы линейных уравнений.

Задание левой и правой части системы уравнений:

$$C(n) = \begin{pmatrix} 6 & 12 \\ 12 & 35.2 \end{pmatrix} \qquad b(n) = \begin{pmatrix} 27.2 \\ 78.56 \end{pmatrix}$$

Решение системы линейных уравнений:

$$a1 := \text{lsolve}(C(n), b(n))$$

$$a1 = \begin{pmatrix} 0.219 \\ 2.157 \end{pmatrix}$$

Задание аппроксимирующего полинома:

$$P11(t) := a1_0 + a1_1 \cdot t$$

Б) Нахождение коэффициентов аппроксимирующего полинома с помощью функции ***regress***.

$$R1 := \text{regress}(x, y, n)$$

$$R1^T = (3 \quad 3 \quad 1 \quad 0.219 \quad 2.157)$$

Значения коэффициентов аппроксимирующего полинома совпадают с полученными ранее.

Задание аппроксимирующего полинома:

$$P12(t) := R1_3 + R1_4 \cdot t$$

Полином 2-й степени

Задание степени полинома:

$$n := 2$$

А) Нахождение коэффициентов аппроксимирующего полинома с помощью системы линейных уравнений.

Задание левой и правой части системы уравнений:

$$C(n) := \begin{pmatrix} 6 & 12 & 35.2 \\ 12 & 35.2 & 115.2 \\ 35.2 & 115.2 & 400.998 \end{pmatrix} \quad b(n) := \begin{pmatrix} 27.2 \\ 78.56 \\ 266.624 \end{pmatrix}$$

Решение системы линейных уравнений:

$$a2 := \text{lsolve}(C(n), b(n))$$

$$a2 = \begin{pmatrix} 1.671 \\ -0.566 \\ 0.681 \end{pmatrix}$$

Задание аппроксимирующего полинома:

$$P21(t) := a2_0 + a2_1 \cdot t + a2_2 \cdot t^2$$

Б) Нахождение коэффициентов аппроксимирующего полинома с помощью функции ***regress***.

$$R2 := \text{regress}(x, y, n)$$

$$R2^T = (3 \quad 3 \quad 2 \quad 1.671 \quad -0.566 \quad 0.681)$$

Значения коэффициентов аппроксимирующего полинома совпадают с полученными ранее.

Задание аппроксимирующего полинома:

$$P22(t) := R2_3 + R2_4 \cdot t + R2_5 \cdot t^2$$

Полученный аппроксимирующий полином позволяет построить его график и найти его числовое значение при разных значениях ***t***. Функция ***interp*** позволяет непосредственно вычислить значение полинома.

Пример нахождения числового значения аппроксимирующего полинома при ***t***=2 двумя способами:

$$\text{interp}(R2, x, y, 2) = 3.262 \qquad P22(2) = 3.262$$

Полином 3-й степени

А) Нахождение коэффициентов аппроксимирующего полинома с помощью системы линейных уравнений.

Задание левой и правой части системы уравнений:

$$C(n) := \begin{pmatrix} 6 & 12 & 35.2 & 115.2 \\ 12 & 35.2 & 115.2 & 400.998 \\ 35.2 & 115.2 & 400.998 & 1.45 \times 10^3 \\ 115.2 & 400.998 & 1.45 \times 10^3 & 5.378 \times 10^3 \end{pmatrix} \quad b(n) := \begin{pmatrix} 27.2 \\ 78.56 \\ 266.624 \\ 966.349 \end{pmatrix}$$

Решение системы линейных уравнений:

$$a3 := \text{lsolve}(C(n), b(n))$$

$$a3 = \begin{pmatrix} 0.438 \\ 6.474 \\ -4.137 \\ 0.803 \end{pmatrix}$$

Задание аппроксимирующего полинома:

$$P31(t) := a3_0 + a3_1 \cdot t + a3_2 \cdot t^2 + a3_3 \cdot t^3$$

Б) Нахождение коэффициентов аппроксимирующего полинома с помощью функции ***linfit***.

Составляем вектор-функцию **F** из линейной комбинации базисных функций.

$$F(x) := \begin{pmatrix} 1 \\ x \\ x^2 \\ x^3 \end{pmatrix}$$

Вычисляем вектор коэффициентов аппроксимирующего полинома.

$$a31 := \text{linfit}(x, y, F)$$

$$a31 = \begin{pmatrix} 0.438 \\ 6.474 \\ -4.137 \\ 0.803 \end{pmatrix}$$

Значения коэффициентов аппроксимирующего полинома совпадают с полученными ранее.

Задание аппроксимирующего полинома:

$$P32(t) := a31_0 + a31_1 \cdot t + a31_2 \cdot t^2 + a31_3 \cdot t^3$$

Функция ***linfit*** позволяет непосредственно вычислить значение полинома без представления его в явном виде.

Пример нахождения числового значения аппроксимирующего полинома при ***t***=3 двумя способами:

$$\text{linfit}(x, y, F) \cdot F(3) = 4.308 \qquad P32(3) = 4.308$$

Полином 4-й степени

Задание степени полинома:

$$n := 4$$

А) Нахождение коэффициентов аппроксимирующего полинома с помощью системы линейных уравнений.

Задание левой и правой части системы уравнений:

$$C(n) = \begin{pmatrix} 6 & 12 & 35.2 & 115.2 & 400.998 \\ 12 & 35.2 & 115.2 & 400.998 & 1.45 \times 10^3 \\ 35.2 & 115.2 & 400.998 & 1.45 \times 10^3 & 5.378 \times 10^3 \\ 115.2 & 400.998 & 1.45 \times 10^3 & 5.378 \times 10^3 & 2.031 \times 10^4 \\ 400.998 & 1.45 \times 10^3 & 5.378 \times 10^3 & 2.031 \times 10^4 & 7.767 \times 10^4 \end{pmatrix} \quad b(n) = \begin{pmatrix} 27.2 \\ 78.56 \\ 266.624 \\ 966.349 \\ 3.61 \times 10^3 \end{pmatrix}$$

Решение системы линейных уравнений:

$$a4 := \text{lsolve}(C(n), b(n))$$

$$a4 = \begin{pmatrix} -0.048 \\ 16.593 \\ -17.734 \\ 6.337 \\ -0.692 \end{pmatrix}$$

Задание аппроксимирующего полинома:

$$P41(t) := a4_0 + a4_1 \cdot t + a4_2 \cdot t^2 + a4_3 \cdot t^3 + a4_4 \cdot t^4$$

Б) Нахождение коэффициентов аппроксимирующего полинома с помощью функции ***interp***.

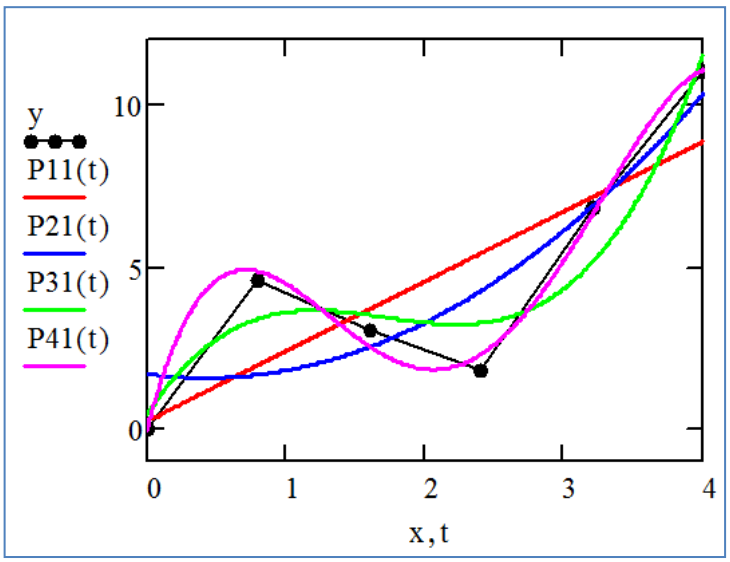
$$R4 := \text{regress}(x, y, n)$$

$$P42(t) := \text{interp}(R4, x, y, t)$$

Пример нахождения числового значения аппроксимирующего полинома при ***t***=1 двумя способами:

$$P41(1) = 4.456 \qquad P42(1) = 4.456$$

Построим совмещенный график заданных узлов и всех полученных полиномов.



Найдем суммы квадратов отклонений ***d*** найденных аппроксимирующих полиномов различных степеней от табличных значений во всех узлах. Для этого составим программу.

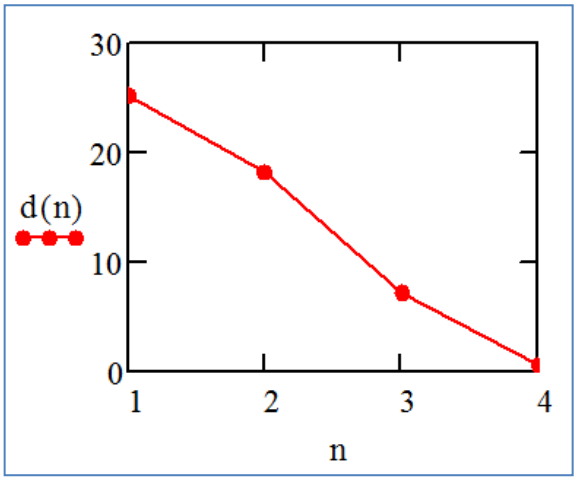
$$d(k) := \left| \begin{array}{l} d \leftarrow 0 \\ \text{for } i \in 0..\text{rows}(x) - 1 \\ \quad d \leftarrow d + \left(y_i - \text{interp}\left(\text{regress}(x, y, k), x, y, x_i\right)\right)^2 \end{array} \right.$$

$$n := 1..4$$

$d(n) =$

25.217
18.129
7.177
0.571

График зависимости суммы квадратов отклонений полиномов во всех узлах от степени полинома.



1.5 Содержание отчета

1. Титульная страница с названием работы.
2. Задание.
3. Цель работы и краткие теоретические сведения.
4. Выполненные задания.
5. Выводы по проделанной работе (сделать самостоятельно).

1.6 Контрольные вопросы

1. Какое должно быть соотношение между порядком аппроксимирующего полинома и количеством узлов таблицы значений функции?
2. Как вычисляются отклонения значений аппроксимирующего полинома и функции?

3. Из каких условий определяются коэффициенты аппроксимирующего полинома?
4. От чего зависит значение суммы квадратов отклонений?
5. Каковы основные этапы решения задачи аппроксимации методом наименьших квадратов?
6. Какие функции **MathCAD** решают задачу аппроксимации методом наименьших квадратов?

1.7 Литература

1. Крылов В.И., Бобков В.В., Монастырный П.И. Вычислительные методы. – М., Наука,1976. –304 с.
2. Копченова Н.В., Марон И.А. Вычислительная математика в примерах и задачах. – М., Наука,1972.
3. Петраш В.И. Вычислительная система MathCAD Ч. 1. Учебное пособие: http://elib.spbstu.ru/dl/2/s17-252.pdf, 2017 год.