Федеральное агентство по рыболовству
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Астраханский государственный технический университет»
Система менеджмента качества в области образования, воспитания, науки и инноваций сертифицирована DQS по международному стандарту ISO 9001:2015

Институт информационных технологий и коммуникаций
Кафедра «Высшая и прикладная математика»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
по выполнению практических работ
ПО ДИСЦИПЛИНЕ «ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В НАУКЕ И ПРОИЗВОДСТВЕ»

Астрахань, 2021

Автор:
к.п.н, доцент кафедры «Высшая и прикладная математика» Л.Б. Аминул

Рецензент:
д.т.н., профессор кафедры «Высшая и прикладная математика» И.Ю. Квятковская

.

Методические указания по выполнению практических работ по дисциплине «Информационные технологии в науке и производстве» утверждены на заседании кафедры «Высшая и прикладная математика» «24» июня 2021 г., протокол № 5.

Методические указания содержат теоретический материал, практические задания, примеры выполнения заданий, контрольные вопросы.

Содержание

Практическая работа №1 (часть1, 2)

Использование программных продуктов семейства Microsoft для реализации инженерных методик расчета

Обработка аналитической информации с использованием MS Excel. Анализ данных. Статистика.

Цель: изучить инструмент Описательная статистика, входящий в надстройку Пакет Анализа. Получить навык расчета показателей выборки.

Полезные материалы

1. *Использование пакета анализа*
2. *Инженерные функции (справка)*
3. *Загрузка пакета анализа*

Задача описательной статистики (descriptive statistics) заключается в том, чтобы с использованием математических инструментов свести сотни значений выборки к нескольким итоговым показателям, которые дают представление о выборке. В качестве таких статистических показателей используются: среднее, медиана, мода, дисперсия, стандартное отклонение и др.

Опишем набор числовых данных с помощью определенных показателей. Для чего нужны эти показатели? Эти показатели позволят сделать определенные статистические выводы о распределении , из которого была взята выборка . Например, если у нас есть выборка значений толщины трубы, которая изготавливается на определенном оборудовании, то на основании анализа этой выборки мы сможем сделать, с некой определенной вероятностью, заключение о состоянии процесса изготовления.

ПРАКТИЧЕСКОЕ ЗАДАНИЕ

Часть 1. *Подключение и загрузка данных*

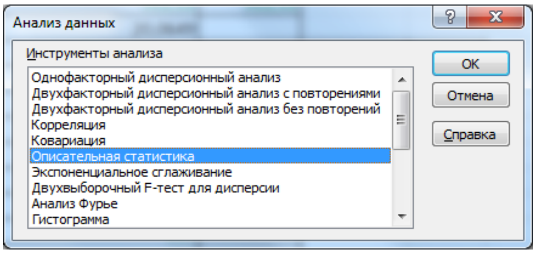
Упражнение 1. Надстройка Пакета Анализа, описательная статистика

Для вычисления статистических показателей одномерных выборок , используем надстройку Пакет анализа . Затем, все показатели, рассчитанные надстройкой, вычислим с помощью встроенных функций MS EXCEL.

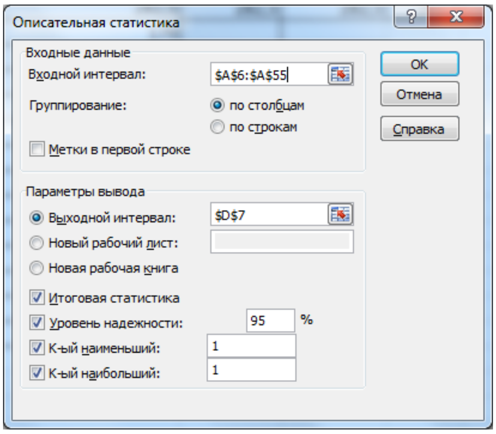
Выборку разместим на листе *примера* в файле примера в диапазоне *A6:A55* (50 значений).

Примечание. Для удобства написания формул для диапазона A6:A55 создан Именованный диапазон Выборка.

В диалоговом окне Анализ данных выберите инструмент Описательная статистика.



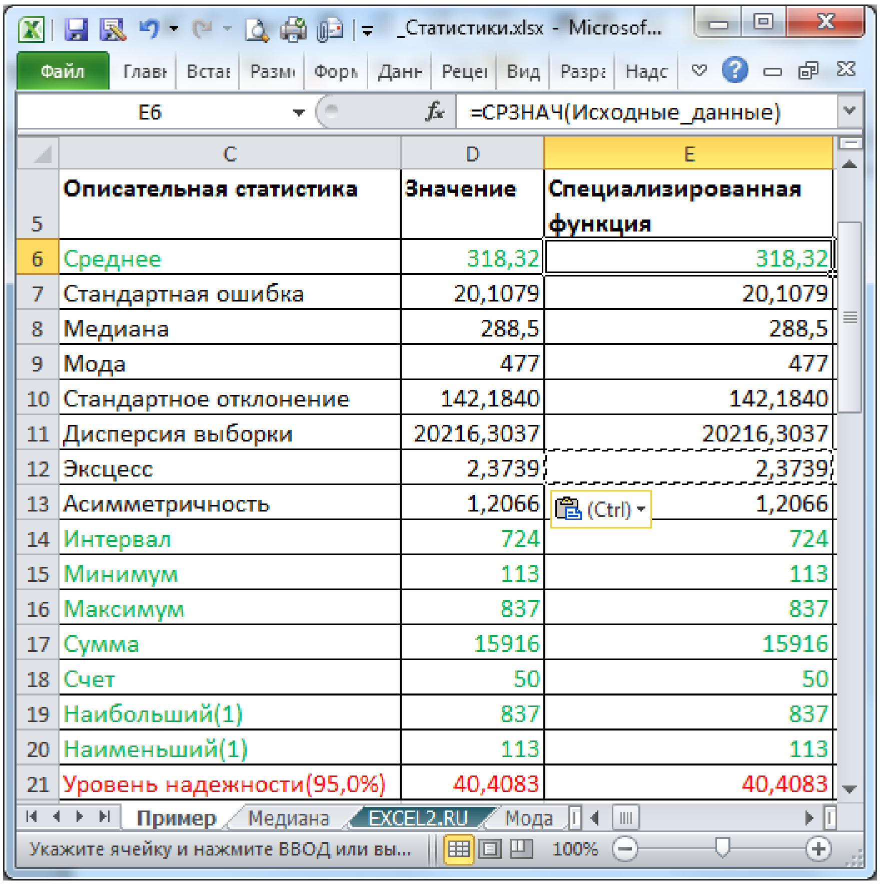
После нажатия кнопки ОК будет выведено другое диалоговое окно,



в котором нужно указать:

- *входной интервал* (Input Range) – это диапазон ячеек, в котором содержится массив данных. Если в указанный диапазон входит текстовый заголовок набора данных, то нужно поставить галочку в поле Метки в первой строке (Labels in first row). В этом случае заголовок будет выведен в Выходном интервале. Пустые ячейки будут проигнорированы, поэтому нулевые значения необходимо обязательно указывать в ячейках, а не оставлять их пустыми;
- *выходной интервал* (Output Range). Здесь укажите адрес верхней левой ячейки диапазона, в который будут выведены статистические показатели;
- *Итоговая статистика* (Summary Statistics) . Поставьте галочку напротив этого поля – будут выведены основные показатели выборки: среднее, медиана, мода, стандартное отклонение и др.;
- Также можно поставить галочки напротив полей Уровень надежности (Confidence Level for Mean) , К-й наименьший (Kth Largest) и К-й наибольший (Kth Smallest).

В результате будут выведены следующие статистические показатели:



	C	D	E
5	Описательная статистика	Значение	Специализированная функция
6	Среднее	318,32	318,32
7	Стандартная ошибка	20,1079	20,1079
8	Медиана	288,5	288,5
9	Мода	477	477
10	Стандартное отклонение	142,1840	142,1840
11	Дисперсия выборки	20216,3037	20216,3037
12	Эксцесс	2,3739	2,3739
13	Асимметричность	1,2066	1,2066
14	Интервал	724	724
15	Минимум	113	113
16	Максимум	837	837
17	Сумма	15916	15916
18	Счет	50	50
19	Наибольший(1)	837	837
20	Наименьший(1)	113	113
21	Уровень надежности(95,0%)	40,4083	40,4083

Рис. 1. Лист Пример (показатели)

Все показатели выведены в виде значений, а не формул. Если массив данных изменился, то необходимо перезапустить расчет.

Если во входном интервале указать ссылку на несколько столбцов данных, то будет рассчитано соответствующее количество наборов показателей. Такой подход позволяет сравнить несколько наборов данных. При сравнении нескольких наборов данных используйте заголовки (включите их во Входной интервал и установите галочку в поле Метки в первой строке). Если наборы данных разной длины, то это не проблема - пустые ячейки будут проигнорированы.

Зеленым цветом на картинке выше и в ФАЙЛЕ ПРИМЕРА выделены показатели, которые не требуют особого пояснения. Для большинства из них имеется специализированная функция:

Интервал (Range) - разница между максимальным и минимальным значениями;

- *Минимум* (Minimum) – минимальное значение в диапазоне ячеек, указанном во Входном интервале;
- *Максимум* (Maximum)– максимальное значение;
- *Сумма* (Sum) – сумма всех значений;
- *Счет* (Count) – количество значений во Входном интервале (пустые ячейки игнорируются;
- *Наибольший* (Kth Largest) – выводится К-й наибольший. Например, 1-й наибольший – это максимальное значение;
- *Наименьший* (Kth Smallest) – выводится К-й наименьший. Например, 1-й наименьший – это минимальное значение.

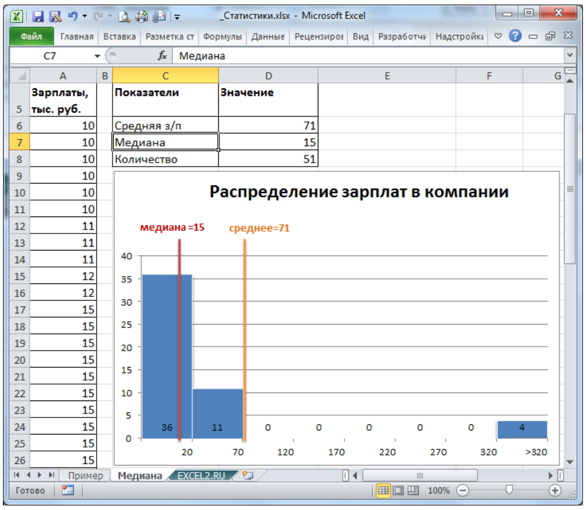
Ниже даны подробные описания остальных показателей.

Среднее выборки. Среднее (mean, average) или выборочное среднее или среднее выборки (sample average) представляет собой арифметическое среднее всех значений массива. В MS EXCEL для вычисления среднего выборки используется функция СРЗНАЧ(). Выборочное среднее является «хорошей» (несмещенной и эффективной) оценкой математического ожидания случайной величины.

Медиана выборки. Медиана (Median) – это число, которое является серединой множества чисел (в данном случае выборки): половина чисел множества больше, чем медиана, а половина чисел меньше, чем медиана. Для определения медианы необходимо сначала отсортировать множество чисел. Например, медианой для чисел 2, 3, 3, 4 , 5, 7, 10 будет 4.

Если множество содержит четное количество чисел, то вычисляется среднее для двух чисел, находящихся в середине множества. Например, медианой для чисел 2, 3, 3, 5, 7, 10 будет 4, т.к. (3+5)/2.

Если имеется длинный хвост распределения, то Медиана лучше, чем среднее значение, отражает «типичное» или «центральное» значение. Например, рассмотрим несправедливое распределение зарплат в компании, в которой руководство получает существенно больше, чем основная масса сотрудников.



Для определения медианы в MS EXCEL существует одноименная функция МЕДИАНА() , английский вариант – MEDIAN().

Медиану также можно вычислить с помощью формул

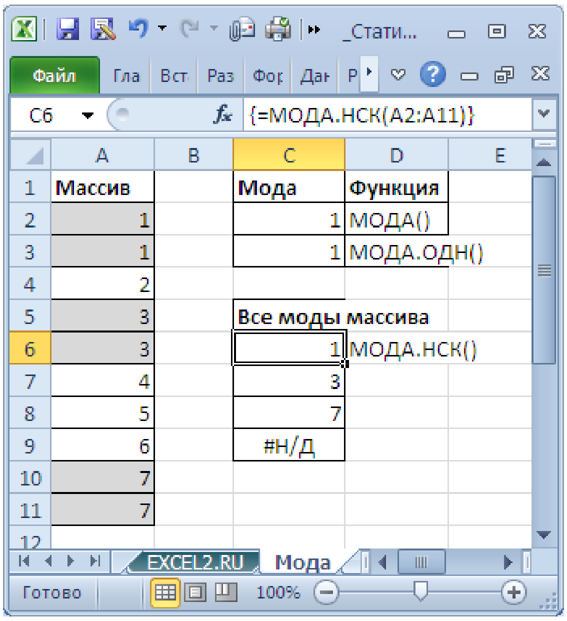
=КВАРТИЛЬ.ВКЛ(Выборка;2) =ПРОЦЕНТИЛЬ.ВКЛ(Выборка;0,5).

Мода выборки. Мода (Mode) – это наиболее часто встречающееся (повторяющееся) значение в выборке. Например, в массиве (1; 1; 2; 2; 2; 3; 4; 5) число 2 встречается чаще всего – 3 раза. Значит, число 2 – это мода. Для вычисления моды используется функция МОДА() , английский вариант MODE().

Примечание: если в массиве нет повторяющихся значений, то функция вернет значение ошибки #Н/Д.

Начиная с MS EXCEL 2010 вместо функции МОДА() рекомендуется использовать функцию МОДА.ОДН() , которая является ее полным аналогом. Кроме того, в MS EXCEL 2010 появилась новая функция МОДА.НСК() , которая возвращает несколько наиболее часто повторяющихся значений (если количество их повторов совпадает).

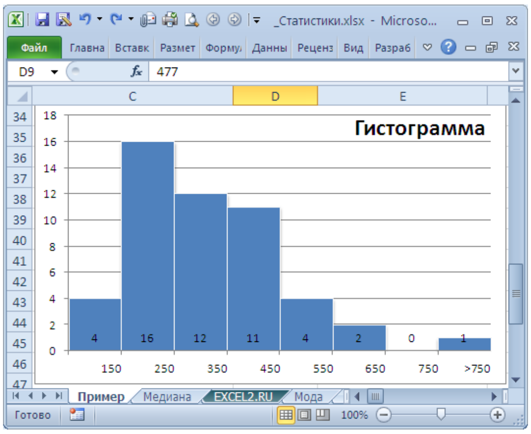
Например, в массиве (1; 1; 2; 2; 2; 3; 4 ; 4 ; 4 ; 5) числа 2 и 4 встречаются наиболее часто – по 3 раза. Значит, оба числа являются модами. Функции МОДА.ОДН() и МОДА() вернут значение 2, т.к. 2 встречается первым, среди наиболее повторяющихся значений (см. файл примера , лист Мода).



Как видно из картинки выше, функция МОДА.НСК() вернула все три *моды* из массива чисел в диапазоне *A2:A11* : 1; 3 и 7. Для этого, выделите диапазон *C6:C9* , в Строку формул введите формулу =МОДА.НСК(A2:A11) и нажмите **CTRL+SHIFT+ENTER** . Диапазон *C 6: C 9* охватывает 4 ячейки, т.е. количество выделяемых ячеек должно быть больше или равно количеству *мод.* Если ячеек больше чем м *о* д, то избыточные ячейки будут заполнены значениями ошибки #Н/Д. Если *мода* только одна, то все выделенные ячейки будут заполнены значением этой *моды.*

Теперь вспомним, что мы определили *моду* для выборки, т.е. для конечного множества значений, взятых из *генеральной совокупности.* Для *непрерывных случайных величин* вполне может оказаться, что выборка состоит из массива на подобие этого (0,935; 1,211; 2,430; 3,668; 3,874; …), в котором может не оказаться повторов и функция МОДА() вернет ошибку.

Модой массива значений является число 477, т.к. оно встречается 2 раза, остальные значения не повторяются. Если рассмотреть гистограмму распределения , построенную для нашего массива, то увидим, что 477 не принадлежит интервалу наиболее часто встречающихся значений (от 150 до 250).



Для того, чтобы получить оценку *моды* распределения, из *генеральной совокупности* которого взята *выборка* , можно, например, построить гистограмму . Оценкой для *моды* будет служить интервал наиболее часто встречающихся значений (самого высокого столбца). Как было сказано выше, в нашем случае это интервал от 150 до 250.

Вывод : Значение *моды* для *выборки* , рассчитанное с помощью функции МОДА() , может ввести в заблуждение, особенно для небольших выборок. Эта функция эффективна, когда случайная величина может принимать лишь несколько дискретных значений, а размер *выборки* существенно превышает количество этих значений.

Например, в рассмотренном примере о распределении заработной платы, *модой* является число 15 (17 значений из 51, т.е. 33%). В этом случае функция МОДА() дает хорошую оценку «наиболее вероятного» значения зарплаты.

Мода и среднее значение

Не смотря на то, что *мода* – это наиболее вероятное значение случайной величины (вероятность выбрать это значение из *Генеральной совокупности* максимальна), не следует ожидать, что среднее значение обязательно будет близко к *моде* .

Примечание: Мода и *среднее* симметричных распределений совпадает (имеется ввиду симметричность плотности распределения).

Представим, что мы бросаем некий «неправильный» кубик, у которого на гранях имеются значения (1; 2; 3; 4; 6; 6), т.е. значения 5 нет, а есть вторая 6. *Модой* является 6, а среднее значение – 3,6666.

Дисперсия выборки

Дисперсия выборки или *выборочная дисперсия (sample variance*) характеризует разброс значений

$$s^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}, \text{ где } \bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$$

или

$$s^2 = \frac{\sum_{i=1}^{n} x_i^2 - n\bar{x}^2}{n-1}$$

или

$$s^2 = \frac{\sum_{i=1}^{n} x_i^2 - \frac{(\sum_{i=1}^{n} x_i)^2}{n}}{n-1}$$

Из формулы №1 видно, что *дисперсия выборки – это* сумма квадратов отклонений каждого значения в массиве от среднего, деленная на размер выборки минус 1.

Дисперсию можно также вычислить непосредственно по нижеуказанным формулам (см. ФАЙЛ ПРИМЕРА): =КВАДРОТКЛ(Выборка)/(СЧЁТ(Выборка)-1) =(СУММКВ(Выборка)-СЧЁТ(Выборка)*СРЗНАЧ(Выборка)^2)/ (СЧЁТ(Выборка)-1) – обычная формула =СУММ((Выборка - СРЗНАЧ(Выборка))^2)/ (СЧЁТ(Выборка)-1) – формула массива

Дисперсия выборки равна 0, только в том случае, если все значения равны между собой и, соответственно, равны *среднему значению.*

Чем больше величина *дисперсии,* тем больше разброс значений в массиве относительно *среднего.*

Размерность *дисперсии* соответствует квадрату единицы измерения исходных значений. Например, если значения в выборке представляют собой измерения веса детали (в кг), то размерность *дисперсии* будет кг 2. Это бывает сложно интерпретировать, поэтому для характеристики разброса значений чаще используют величину равную квадратному корню из *дисперсии* – *стандартное отклонение.*

Стандартное отклонение выборки

Стандартное отклонение выборки (Standard Deviation), как и *дисперсия,* – это мера того, насколько широко разбросаны значения в выборке *относительно их среднего.*

По определению, *стандартное отклонение* равно квадратному корню из *дисперсии:*

$$s = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}}$$

Стандартное отклонение не учитывает величину значений в *выборке,* а только степень рассеивания значений вокруг их *среднего.* Чтобы проиллюстрировать это приведем пример.

Вычислим стандартное отклонение для 2-х *выборок:* (1; 5; 9) и (1001; 1005; 1009). В обоих случаях, s=4. Очевидно, что отношение величины стандартного отклонения к значениям массива у *выборок* существенно отличается.

С версии MS EXCEL 2010 рекомендуется использовать функцию СТАНДОТКЛОН.В().

Стандартное отклонение можно также вычислить непосредственно по нижеуказанным формулам (см. ФАЙЛ ПРИМЕРА): =КОРЕНЬ(КВАДРОТКЛ(Выборка)/(СЧЁТ(Выборка)-1)) =КОРЕНЬ((СУММКВ(Выборка)-СЧЁТ(Выборка)*СРЗНАЧ(Выборка)^2)/(СЧЁТ(Выборка)-1))

Стандартная ошибка

В *Пакете анализа* под термином *стандартная ошибка* имеется ввиду *Стандартная ошибка среднего* (Standard Error of the Mean, SEM). *Стандартная ошибка среднего* – это оценка стандартного отклонения распределения выборочного среднего.

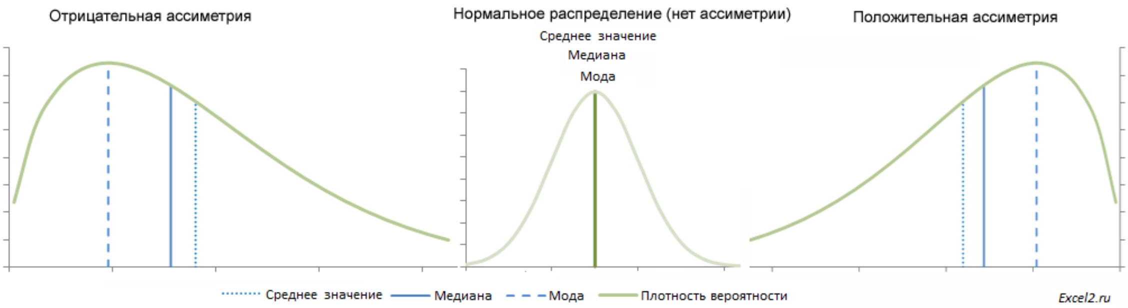
Стандартное отклонение распределения выборочного среднего вычисляется по формуле σ/√n, где n — объём выборки, σ - стандартное отклонение исходного распределения, из которого взята выборка. Т.к. обычно стандартное отклонение исходного распределения неизвестно, то в расчетах вместо σ используют ее оценку s - стандартное отклонение выборки. А соответствующая величина s/√n имеет специальное название - Стандартная ошибка среднего. Именно эта величина вычисляется в Пакете анализа.

В MS EXCEL стандартную ошибку среднего можно также вычислить по формуле =СТАНДОТКЛОН.В(Выборка)/ КОРЕНЬ(СЧЁТ(Выборка))

Асимметричность

Асимметричность или *коэффициент асимметрии* (skewness) характеризует степень несимметричности распределения (плотности распределения) относительно его *среднего.*

Положительное значение *коэффициента асимметрии* указывает, что размер правого «хвоста» распределения больше, чем левого (относительно среднего). Отрицательная асимметрия, наоборот, указывает на то, что левый хвост распределения больше правого. *Коэффициент асимметрии* идеально симметричного распределения или выборки равно 0.

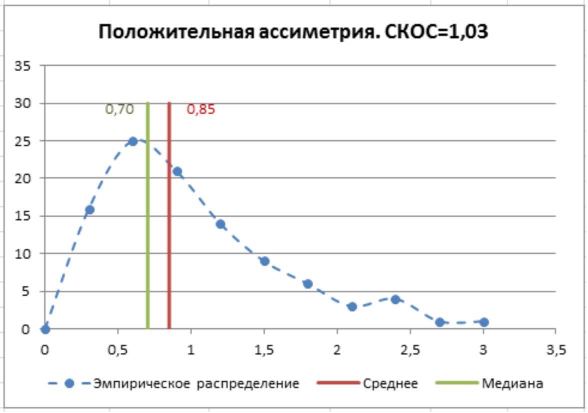


Функция СКОС() , английский вариант SKEW(), возвращает коэффициент *асимметрии выборки* , являющейся оценкой *асимметрии* соответствующего распределения, и определяется следующим образом:

$$G_1 = \frac{n}{(n-1)*(n-2)}\sum_{i=1}^{n}\left(\frac{x_i - \bar{x}}{s}\right)^3$$

где n – размер *выборки,* s – *стандартное отклонение выборки.*

В файле примера на листе Скос приведен расчет коэффициента *асимметрии* на примере случайной выборки, которое имеет значительную положительную *асимметрию* при параметрах распределения W(1,5; 1).



Эксцесс выборки

Эксцесс показывает относительный вес «хвостов» распределения относительно его центральной части.

Для того чтобы определить, что относится к хвостам распределения, а что к его центральной части, можно использовать границы μ +/- σ .

Функция ЭКСЦЕСС() , английский вариант KURT(), вычисляет на основе значений выборки несмещенную оценку *эксцесса распределения* случайной величины и определяется следующим образом:

$$G_2 = \frac{n*(n+1)}{(n-1)*(n-2)*(n-3)} \sum_{i=1}^{n} \left(\frac{x_i-\bar{x}}{s}\right)^4 - \frac{3*(n-1)^2}{(n-2)*(n-3)}$$

Уровень надежности

Уровень надежности - означает вероятность того, что *доверительный интервал* содержит истинное значение оцениваемого параметра распределения.

Вместо термина *Уровень надежности* часто используется термин *Уровень доверия.*

Задав значение *Уровня надежности* в окне *надстройки Пакет анализа* , MS EXCEL вычислит половину ширины доверительного интервала для оценки среднего (дисперсия неизвестна) .

Часть 2. *Расчет доверительного интервала (дисперсия известна)*

Упражнение 2. Доверительный интервал для оценки среднего, статистический вывод

Построим в MS Excel доверительный интервал для оценки среднего значения распределения в случае известного значения дисперсии.

Процесс обобщения данных *выборки,* который приводит к **вероятностным** утверждениям обо всей *генеральной совокупности,* называют статистическим выводом (statistical inference).

Для построения *Доверительного интервала* нам потребуется знание следующих понятий:

- дисперсия и стандартное отклонение,
- выборочное распределение статистики,
- уровень доверия/ уровень значимости,
- стандартное нормальное распределение и его квантили.

Интервал, в котором может находиться неизвестный параметр, совпадает со всей возможной областью изменения этого параметра, поскольку соответствующую *выборку,* а значит и *оценку параметра,* можно получить с ненулевой вероятностью. Поэтому приходится ограничиваться нахождением границ изменения неизвестного параметра с некоторой заданной наперед вероятностью.

Определение: *Доверительным интервалом* называют такой интервал изменения случайной величины, который с заданной вероятностью, накроет истинное значение оцениваемого параметра распределения.

Эту заданную вероятность называют уровнем доверия (или доверительной вероятностью).

Обычно используют значения уровня доверия 90%; 95%; 99%, реже 99,9% и т.д. Например, уровень доверия 95% означает, что дополнительное событие, вероятность которого 1-0,95=5%, исследователь считает маловероятным или невозможным.

Формулировка задачи

Предположим, что из *генеральной совокупности,* имеющей нормальное распределение взята, *выборка* размера n. Предполагается, что *стандартное отклонение* этого распределения известно. Необходимо на основании этой *выборки* оценить неизвестное *среднее значение распределения* (μ, математическое ожидание) и построить соответствующий *двухсторонний доверительный интервал* .

Построение доверительного интервала

Обычно, зная распределение и его параметры, мы можем вычислить вероятность того, что случайная величина примет значение из заданного нами интервала. Сейчас поступим наоборот: найдем интервал, в который случайная величина попадет с заданной вероятностью. Например, из свойств *нормального распределения* известно, что с вероятностью 95%, случайная величина, распределенная по *нормальному закону* , попадет в интервал примерно +/- 2 стандартных отклонения от *среднего значения*. Этот интервал, послужит нам прототипом для *доверительного интервала.*

Теперь разберемся, знаем ли мы распределение, чтобы вычислить этот интервал? Для ответа на вопрос мы должны указать форму распределения и его параметры.

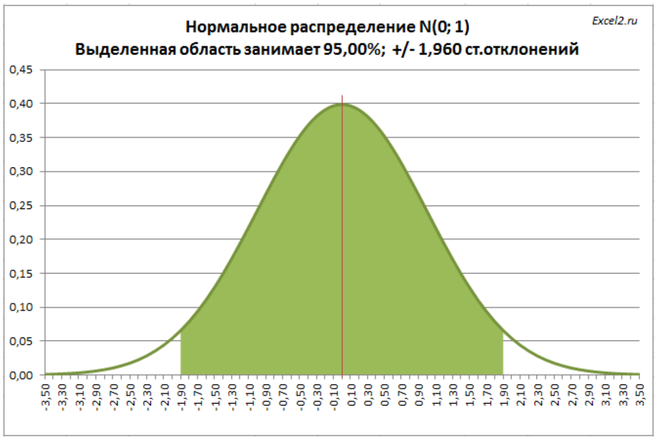
Форму распределения мы знаем – это *нормальное распределение* (напомним, что речь идет о *выборочном распределении статистики X ср).*

Параметр μ неизвестен (его как раз нужно оценить с помощью *доверительного интервала*), но у нас есть его оценка *X ср* , вычисленная на основе *выборки,* которую можно использовать.

Второй параметр – *стандартное отклонение выборочного среднего* будем считать известным, он равен σ/√n.

Т.к. мы не знаем μ, то будем строить интервал +/- 2 *стандартных отклонения* не от *среднего значения,* а от известной его оценки *X ср.* Т.е. при расчете *доверительного интервала* мы НЕ будем считать, что *X ср* попадет в интервал +/- 2 *стандартных отклонения* от μ с вероятностью 95%, а будем считать, что интервал +/- 2 *стандартных отклонения* от *X ср* с вероятностью 95% накроет μ – *среднее генеральной совокупности,* из которого взята *выборка* . Эти два утверждения эквивалентны, но второе утверждение нам позволяет построить *доверительный интервал.*

Кроме того, уточним интервал: случайная величина, распределенная по *нормальному закону,* с вероятностью 95% попадает в интервал +/- 1,960 *стандартных отклонений,* а не+/- 2 *стандартных отклонения.* Это можно рассчитать с помощью формулы =НОРМ.СТ.ОБР ((1+0,95)/2), см. файл примера лист интервал.



Теперь мы можем сформулировать вероятностное утверждение, которое послужит нам для формирования *доверительного интервала:* «Вероятность того, что *среднее генеральной совокупности* находится от *среднего выборки* в пределах 1,960 «стандартных *отклонений выборочного среднего»,* равна 95%».

Значение вероятности, упомянутое в утверждении, имеет специальное название уровень доверия, *который связан с* уровнем значимости α (альфа) простым выражением *уровень доверия = 1* - α . В нашем случае *уровень значимости* α *=1-0,95=0,05* .

Теперь на основе этого вероятностного утверждения запишем выражение для вычисления *доверительного интервала:*

$$\bar{x} - Z_{\alpha/2}\frac{\sigma}{\sqrt{n}} \le \mu \le \bar{x} + Z_{\alpha/2}\frac{\sigma}{\sqrt{n}}$$

где Z α/2 – верхний α/2-квантиль *стандартного нормального распределения* (такое значение случайной величины *z,* что *P (z>= Z α/2) =α/2)*.

В нашем случае при α=0,05, *верхний α/2-квантиль* равен 1,960. Для других уровней значимости α (10%; 1%) *верхний α/2-квантиль Z α/2* можно вычислить с помощью формулы =НОРМ.СТ.ОБР(1-α/2) или, если известен *уровень доверия* , =НОРМ.СТ.ОБР((1+ур.доверия)/2) .

Расчет доверительного интервала в MS EXCEL

Решим задачу. *Время отклика электронного компонента на входной сигнал является важной характеристикой устройства. Инженер хочет построить доверительный интервал для среднего времени отклика при уровне доверия 95%. Из предыдущего опыта инженер знает, что стандартное отклонение время отклика составляет 8 мсек. Известно, что для оценки времени отклика инженер сделал 25 измерений, среднее значение составило 78 мсек.*

Решение: Инженер хочет знать время отклика электронного устройства, но он понимает, что время отклика является не фиксированной, а случайной величиной, которая имеет свое распределение. Так что, лучшее, на что он может рассчитывать, это определить параметры и форму этого распределения.

К сожалению, из условия задачи форма распределения времени отклика нам не известна (оно не обязательно должно быть *нормальным)*. Среднее, т.е. математическое ожидание , этого распределения также неизвестно. Известно только его *стандартное отклонение* σ=8. Поэтому, пока мы не можем посчитать вероятности и построить *доверительный интервал.*

Однако, не смотря на то, что мы не знаем распределение **времени отдельного отклика,** мы знаем, что согласно *ЦПТ, выборочное распределение* **среднего времени отклика** является приблизительно *нормальным* (будем считать, что условия *ЦПТ* выполняются, т.к. размер *выборки* достаточно велик (n=25)) .

Более того, *среднее* этого распределения равно *среднему значению* распределения единичного отклика, т.е. µ. А *стандартное отклонение* этого распределения (σ/√n) можно вычислить по формуле =8/КОРЕНЬ (25).

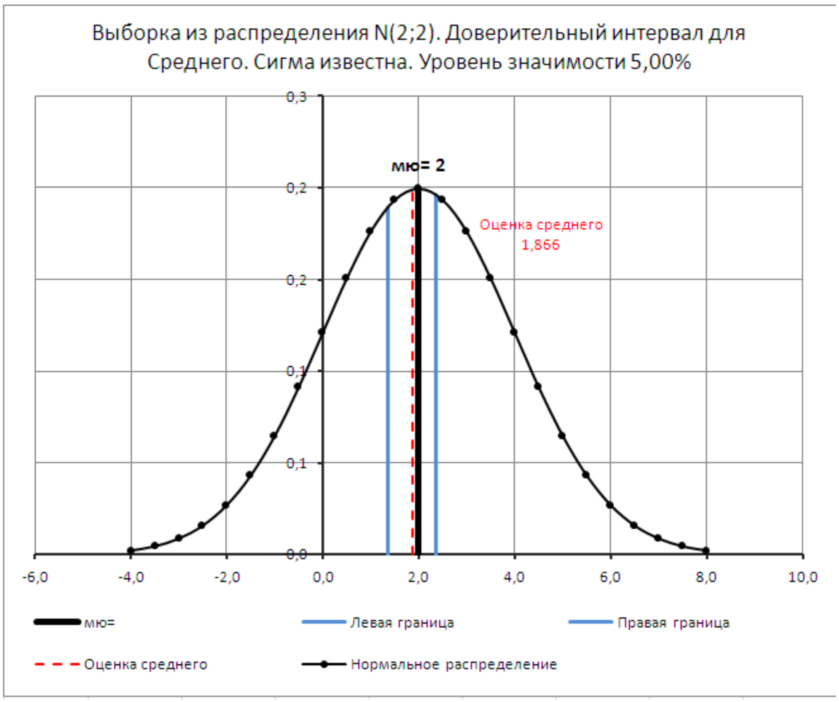
Также известно, что инженером была получена *точечная оценка* параметра µ равная 78 мсек (Х ср). Поэтому, теперь мы можем вычислять вероятности, т.к. нам известна форма распределения (нормальное*)* и его параметры (Х ср и σ/√n).

Инженер хочет знать *математическое ожидание* µ распределения времени отклика. Как было сказано выше, это µ равно *математическому ожиданию выборочного распределения среднего времени отклика.* Если мы воспользуемся *нормальным распределением* N (Х ср; σ/√n), то искомое µ будет находиться в интервале +/-2*σ/√n с вероятностью примерно 95%.

Уровень значимости равен 1-0,95=0,05.
Найдем левую и правую границу *доверительного интервала.*
Левая граница: =78-НОРМ.СТ.ОБР (1-0,05/2) *8/КОРЕНЬ(25) = 74,864
Правая граница: =78+НОРМ.СТ.ОБР (1-0,05/2) *8/КОРЕНЬ(25)=81,136

Ответ: *доверительный интервал* при *уровне доверия 95% и σ =*8 *мсек* равен *78+/-3,136 мсек.*

В файле примера на листе Сигма известна создана форма для расчета и построения *двухстороннего доверительного интервала* для произвольных *выбора* с заданным σ и *уровнем значимости.*



Функция ДОВЕРИТ.НОРМ()

Если значения *выборки* находятся в диапазоне *B20:B79,* а *уровень значимости* равен 0,05; то формула MS EXCEL: =СРЗНАЧ (B20:B79)-ДОВЕРИТ.НОРМ(0,05; σ; СЧЁТ (B20:B79)) вернет левую границу *доверительного интервала.*

Эту же границу можно вычислить с помощью формулы:
=СРЗНАЧ (B20:B79)-НОРМ.СТ.ОБР (1-0,05/2) *σ/КОРЕНЬ(СЧЁТ(B20:B79))

Контрольные вопросы

1. Как подключить надстройку пакет анализа?
2. Функции Квартиль, Процентиль.
3. Дисперсия и стандартное отклонение выборки.
4. Как рассчитать моду для непрерывной случайной величины?
5. Уровень надежности.
6. Эксцесс выборки.
7. Размерность дисперсии.
8. Коэффициент асимметричности.
9. Уровень доверия/уровень значимости.
10. Стандартное и нормальное распределение.

Практическая работа №2.

Программный продукт MathCAD. Предназначение, основные возможности

Математические выражения. Типы данных. Ранжированные переменные. Функции. Основы построения диаграмм. Загрузка и сохранение рабочей области.

Цель: изучить интерфейс Mathcad, научиться загружать и подготавливать данные к анализу.

Mathcad – система компьютерной алгебры из класса систем автоматизированного проектирования, ориентированная на подготовку интерактивных документов с вычислениями и визуальным сопровождением.

С точки зрения пользователя, документ – это чистый лист бумаги, на котором можно размещать блоки трех основных типов: математические выражения, текстовые фрагменты и графические области.

Расположение нетекстовых блоков в документе имеет принципиальное значение – слева направо и сверху вниз.

Математические выражения

К основным элементам математических выражений MathCAD относятся типы данных, операторы, функции и управляющие структуры.

Операторы

Операторы - элементы MathCAD, с помощью которых можно создавать математические выражения. К ним, например, относятся символы арифметических операций, знаки вычисления сумм, произведений, производной и интеграла и т.д.

Оператор определяет:

1. действие, которое должно выполняться при наличии тех или иных значений операндов;
2. сколько, где и какие операнды должны быть введены в оператор.

Операнд – число или выражение, на которое действует оператор. Например, в выражении **5! + 3** число **3** и выражение **5!** – операнды оператора **+** (плюс), а число **5** операнд оператора факториал (**!**). После указания *операндов* операторы становятся исполняемыми по документу блоками.

Типы данных

К *типам данных* относятся числовые константы, обычные и системные переменные, массивы (векторы и матрицы) и данные файлового типа.

Константами называют поименованные объекты, хранящие некоторые значения, которые не могут быть изменены. ***Переменные*** являются поименованными объектами, имеющими некоторое значение, которое может изменяться по ходу выполнения программы. Тип переменной определяется ее значением; переменные могут быть числовыми, строковыми, символьными и т. д. Имена констант, переменных и иных объектов называют *идентификаторами*. Идентификаторы в MathCAD представляют собой набор латинских или греческих букв и цифр.

В MathCAD содержится небольшая группа особых объектов, которые нельзя отнести ни к классу констант, ни к классу переменных, значения которых определены сразу после запуска программы. Их правильнее считать ***системными переменными***, имеющими предопределенные системой начальные значения. Изменение значений системных переменных производят во вкладке **Встроенные переменные** диалогового окна **Math Options** команды **Математика / Опции.**

Обычные переменные отличаются от системных тем, что они должны быть предварительно *определены* пользователем, т. е. им необходимо хотя бы однажды *присвоить значение*. В качестве *оператора присваивания* используется знак :=, тогда как знак = отведен для *вывода значения* константы или переменной.

Если переменной присваивается начальное значение с помощью оператора **:=**, вызывается нажатием клавиши **:** (двоеточие) на клавиатуре, такое присваивание называется *локальным*. До этого присваивания переменная не определена и ее нельзя использовать. Однако с помощью знака **≡** (клавиша ~ на клавиатуре) можно обеспечить *глобальное* присваивание (см. Пример 1 Рисунка 2). MathCAD прочитывает весь документ дважды слева направо и сверху вниз. При первом проходе выполняются все действия, предписанные локальным оператором присваивания (**≡**), а при втором – производятся действия, предписанные локальным оператором присваивания (**:=**), и отображаются все необходимые результаты вычислений (**=**).

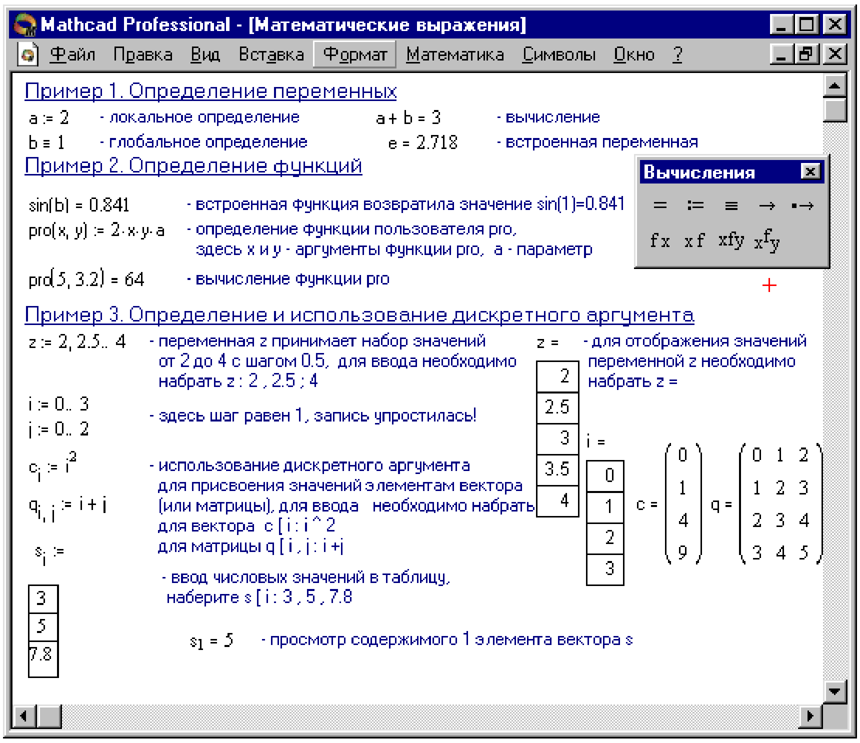


Рис. 2. Математические выражения

Существуют также жирный знак равенства **=** (комбинация клавиш **Ctrl** + =), который используется, например, как оператор приближенного равенства при решении систем уравнений, и символьный знак равенства → (комбинация клавиш **Ctrl + .**).

Дискретные аргументы - особый класс переменных, который в пакете MathCAD зачастую заменяет ***управляющие структуры***, называемые циклами (однако полноценной такая замена не является). Эти переменные имеют ряд фиксированных значений, либо целочисленных (1 способ), либо в виде чисел с определенным шагом, меняющихся от начального значения до конечного (2 способ).

1. *Name* := *Nbegin* .. *Nend*,

где *Name* – имя переменной, *Nbegin* – ее начальное значение, *Nend* – конечное значение, **..** – символ, указывающий на изменение переменной в заданных пределах (вводится клавишей **;**). Если *Nbegin* < *Nend*, то шаг переменной будет равен +1, иначе –1.

2. *Name* := *Nbegin*, (*Nbegin* + *Step*) .. *Nend*

Здесь *Step* – заданный шаг изменения переменной (он должен быть положительным, если *Nbegin* < *Nend*, или отрицательным в обратном случае).

Дискретные аргументы значительно расширяют возможности MathCAD, позволяя выполнять многократные вычисления или циклы с повторяющимися вычислениями, формировать векторы и матрицы (Пример 3 Рисунка 1).

Массив - имеющая уникальное имя совокупность конечного числа числовых или символьных элементов, упорядоченных, некоторым образом, и имеющих определенные адреса. В пакете MathCAD используются массивы двух наиболее распространенных типов:

- одномерные (векторы);
- двумерные (матрицы).

Порядковый номер элемента, который является его адресом, называется *индексом*. Индексы могут иметь только целочисленные значения. Они могут начинаться с нуля или единицы, в соответствии со значением системной переменной **ORIGIN**.

Векторы и матрицы можно задавать различными способами:

- с помощью команды **Вставка / Матрица**, или комбинации клавиш **Ctrl + M**, или щелчком на кнопке  панели **Матрица,** заполнив массив пустых полей для не слишком больших массивов;
- с использованием дискретного аргумента, когда имеется некоторая явная зависимость для вычисления элементов через их индексы (Пример 3 Рисунка 1).

Функции

Функция – выражение, согласно которому проводятся некоторые вычисления с *аргументами* и определяется его числовое значение.

Следует особо отметить разницу между *аргументами* и *параметрами* функции. Переменные, указанные в скобках после имени функции, являются ее *аргументами* и заменяются при вычислении функции значениями из скобок. Переменные в правой части определения функции, не указанные скобках в левой части, являются *параметрами* и должны задаваться *до* определения функции (пример 2, рис. 1).

Главным признаком функции является *возврат значения*, т.е. функция в ответ на обращение к ней по имени с указанием ее аргументов должна возвратить свое значение.

Функции в пакете MathCAD могут быть *встроенные* (см. Приложение 3), т. е. заблаговременно введенные разработчиками, и *определенные пользователем*.

Способы вставки встроенной функции:

1. Выбрать пункт меню **Вставка ⇒ Функция**.
2. Нажать комбинацию клавиш **Ctrl + E**.
3. Щелкнуть на кнопке f(x) .

Текстовые фрагменты

Текстовые фрагменты представляют собой куски текста, которые пользователь хотел бы видеть в своем документе. Существуют два вида текстовых фрагментов:

- *текстовая область* предназначена для небольших кусков текста - подписей, комментариев и т. п. Вставляется с помощью команды **Вставка / Текстовая регион** или комбинации клавиш **Shift + "** (двойная кавычка);
- *текстовый абзац* применяется в том случае, если необходимо работать с абзацами или страницами. Вставляется с помощью комбинации клавиш **Shift + Enter**.

Графические области

Графические области делятся на три основных типа - двумерные графики, трехмерные графики и импортированные графические образы. Двумерные и трехмерные графики строятся самим MathCAD на основании обработанных данных.

Для создания ***декартового графика***:

1. Установить визир в пустом месте рабочего документа.
2. Выбрать команду **Вставка / График / Х-У график,** или нажать комбинацию клавиш **Shift + @,** или щелкнуть кнопку панели **Графики**. Появится шаблон декартового графика.
3. Введите в средней метке под осью X первую независимую переменную, через запятую – вторую и так до 10, например $x1$, $x2$, …
4. Введите в средней метке слева от вертикальной оси Y первую независимую переменную, через запятую – вторую и т. д., например $y1(x1)$, $y2(x2)$, …, или соответствующие выражения.
5. Щелкните за пределами области графика, что бы начать его построение.

Трехмерные, или ***3D-графики***, отображают функции двух переменных вида $Z(X, Y)$. При построении трехмерных графиков в ранних версиях MathCAD поверхность нужно было определить математически (Рисунок 3, способ 2). Теперь применяют функцию MathCAD *CreateMesh.*

CreateMesh(*F* (или *G*, или *f*1, *f*2, *f*3), *x*0, *x*1, *y*0, *y*1, *xgrid*, *ygrid*, *fmap*)

Создает сетку на поверхности, определенной функцией F. $x0$, $x1$, $y0$, $y1$ – диапазон изменения переменных, *xgrid, ygrid* – размеры сетки переменных, *fmap* – функция отображения. Все параметры, за исключением F, - факультативные. Функция *CreateMesh* по умолчанию создает сетку на поверхности с диапазоном изменения переменных от –5 до 5 и с сеткой 20×20 точек.

Пример использования функции *CreateMesh* для построения 3D-графиков приведен на Рисунке 2, способ 1. На Рисунке 2 построена одна и та же поверхность разными способами, с разным форматированием, причем изображены поверхности и под ними те же поверхности в виде контурного графика. Такое построение способно придать рисунку большую наглядность.

Нередко поверхности и пространственные кривые представляют в виде точек, кружочков или иных фигур. Такой график создается операцией **Вставка / График / 3D Точечный**, причем поверхность задается параметрически – с помощью трех матриц (X, Y, Z) (см. Рисунок 3, способ 2), а не одной как в примере на рисунке 3. Для определения исходных данных для такого вида графиков используется функция *CreateSpace* (Рисунок 3, способ 1).

1 способ

$f(x,y) := \sin\left(x^2 + y^2\right)$

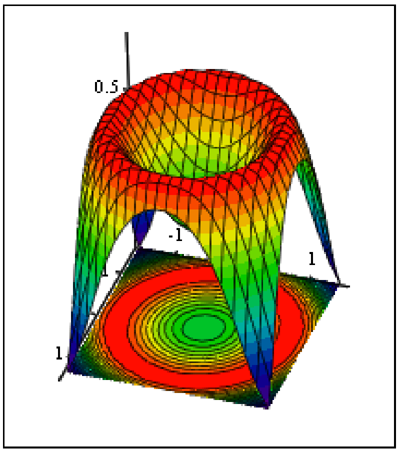
$M := \mathrm{CreateMesh}(f,-1.5,1.5,-1.5,1.5,20,20)$

2 способ

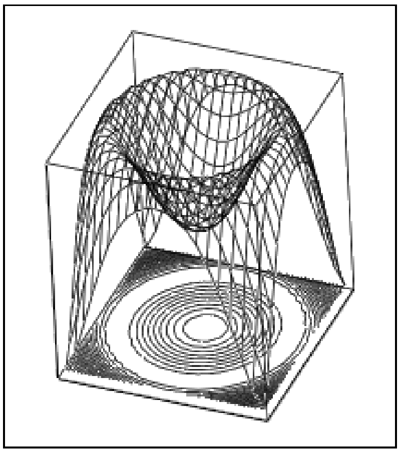
$i := 0..20 \qquad x_i := -1.5 + i \cdot 0.15$

$j := 0..20 \qquad y_j := -1.5 + j \cdot 0.15$

$M1_{i,j} := f\left(x_i, y_j\right)$



M,M



M1,M1

Рис. 3. Пример построения на одном рисунке двух 3D-графиков разного типа

CreateSpace (*F* , *t*0, *t*1, *tgrid*, *fmap*)

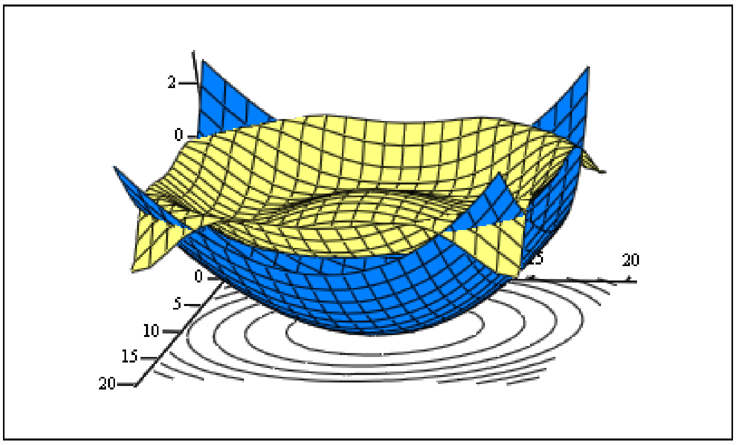
Возвращает вложенный массив трех векторов, представляющих *x*-, *y*-, и *z*-координаты пространственной кривой, определенной функцией *F*. *t*0 и *t*1 – диапазон изменения переменной, *tgrid* – размер сетки переменной, *fmap* – функция отображения. Все параметры, за исключением *F*, - факультативные.

Построение пересекающихся фигур

Особый интерес представляет собой возможность построения на одном графике ряда разных фигур или поверхностей с автоматическим учетом их взаимного пересечения. Для этого надо раздельно задать матрицы соответствующих поверхностей и после вывода шаблона 3D-графика перечислить эти матрицы под ним с использованием в качестве разделителя запятой (Рисунок 4).

$f1(x,y) := -\sin\left(x^2 + y^2\right) \qquad f2(x,y) := x^2 + y^2 - 5 \qquad x := 0..20 \qquad y := 0..20$

$M1_{x,y} := f1\left(\frac{x-10}{5}, \frac{y-10}{5}\right) \qquad M2_{x,y} := f2\left(\frac{x-10}{5}, \frac{y-10}{5}\right)$



M1,M2,M2

Рис. 4. Построение двух пересекающихся поверхностей и одновременно контурного графика

Создание анимационного клипа

MathCAD имеет встроенную переменную FRAME, чье единственное назначение - управление *анимациями*:

- Создайте объект, чей вид зависит от FRAME.
- Убедитесь, что установлен режим автоматического расчета (**Математика ⇒ Автоматическое Вычисление**).
- Выберите **Вид ⇒ Анимация** для вызова одноименного диалогового окна.
- Заключите в выделяющий пунктирный прямоугольник часть рабочего документа, которую нужно анимировать.
- Установите нижние и верхние границы FRAME (поля **От:** и **До:**).
- В поле **Скорость** введите значение скорости воспроизведения (кадров/сек).
- Выберите **Анимация**. Сейчас анимация только создается.
- Сохраните анимацию как AVI файл (**Сохранить как**).
- Воспроизведите сохраненную анимацию **Вид ⇒Воспроизведение.**

ПРАКТИЧЕСКОЕ ЗАДАНИЕ

Упражнение 1. Вычисление выражений

$\sqrt{100} =$ |-10| = 10! = .

Это и все остальные задания снабдить комментариями, используя команду **Вставка ⇒ Текстовая область**.

Определить переменные: a := 3.4, b := 6.22, $c \equiv 0.149$ (причем переменную c - глобально) и выражения:

$Z := \frac{2ab+\sqrt[3]{c}}{\sqrt{(a^2+b^{a+c})\cdot c}}$ $N := e^{\sin c} \cos\frac{a}{b}$.

- Вычислить выражения.
- С помощью команды **Формат/Результат/Формат чисел/Число знаков** изменить точность отображения результатов вычисления *глобально*.

Вывести на экран значение *системной константы* π и установить максимальный формат ее отображения *локально*.

Выполнить следующие операции с комплексными числами:

$Z := -3 + 2i$ $|Z| =$ $Re(Z) =$ $Im(Z) =$ $arg(Z) =$

$\sqrt{Z}=$ $\sqrt{-5}=$ $2 \cdot Z =$ $Z1 := 1 + 2i$ $Z2 := 3 + 4i$

$Z1 + Z2 =$ $Z1 - Z2 =$ $Z1 \cdot Z2 =$ $Z1/Z2 =$

Выполнить следующие операции:

$i := 1 .. 10$ $\sum_i i =$ $\prod_i (i+1)=$ $\int_0^{0.4} x^2 \cdot lg(x+2)dx=$ $\int_{0.8}^{1.2} \frac{\text{ctg } 2x}{(\sin 2x)^2}dx=$

$x := 2$ $\frac{d}{dx}x^5 =$ $\frac{d}{dx}\sin(x)=$

Упражнение 2. Определить векторы *d, S* и *R* через дискретный аргумент *i*.

Отобразить графически таблично заданные функции $S_i(d_i)$ и $R_i(d_i)$, используя команду **ставка⇒График⇒X-Y Зависимость**. Чтобы оформить график, необходимо выполнить следующие команды:

i	d_i	S_i	R_i
0	0.5	3.3	2
1	1	5.9	3.9
2	1.5	7	4.5
3	2	6.3	3.7
4	2.5	4.2	1.2

- Щелкнуть левой клавишей мыши на графике, чтобы выделить его. Затем щелкнуть правой клавишей мыши, при этом появится контекстное меню в котором необходимо выбрать команду **Формат** (появится диалоговое окно **"Formatting Currently Selected X-Y Plot").**
- Нанести линии сетки на график (**Оси X-Y /Вспом. линии**) и отобразить легенду (**След/Скрыть легенду**)
- Отформатировать график так, чтобы в каждой узловой точке графика функции $S_i(d_i)$ стоял знак вида (**След/Символ/box**), а график функции $R_i(d_i)$ отобразить в виде гистограммы (**След/Тип/bar**).

Упражнение 3. Построить декартовы (X-Y Зависимость) и полярные (Полярные Координаты) графики следующих функций:

$$X(\alpha) := cos(\alpha) \cdot sin(\alpha)$$
$$Y(\alpha) := 1.5\, cos(\alpha)^2 - 1$$
$$P(\alpha) := cos(\alpha).$$

Для этого необходимо определить α как дискретный аргумент на интервале от 0 до 2·π с шагом π/30.

Определить по графику **X-Y Зависимость** координаты любой из точек пересечения графиков *Y*(α) и *P*(α)*,* для этого необходимо:

- Выделить график и выбрать из контекстного меню **Масштаб** (появится диалоговое окно **"X-Y Zoom"**) для увеличения части графика в области точки пересечения.
- На чертеже выделить пунктирным прямоугольником окрестность точки пересечения графиков *Y*(α) и *P*(α), которую нужно увеличить.
- Нажать кнопку **Масштаб+**, чтобы перерисовать график.
- Чтобы сделать это изображение постоянным, выбрать ОК.
- Выбрать из контекстного меню Трассировка (появится диалоговое окно "X-Y Trace").
- Внутри чертежа нажать кнопку мыши и переместить указатель мыши на точку, чьи координаты нужно увидеть.
- Выбрать **Copy X** (или **Copy Y**), на свободном поле документа набрать *Xper* := (или *Yper* :=) и выбрать пункт меню **Правка⇒Вставка.**

Вычислить значения функций *X*(α) и *Y*(α) при α:=π⁄2.

Упражнение 4. Создание матрицы, построение графика поверхности

1). Используя команду, **Вставка/Матрица** создать матрицу *Q* размером 6×6, заполнить ее произвольно и отобразить графически с помощью команды **Вставка/График/Поверхности**.

2). Построить график поверхности (**Поверхности**) и карту линий уровня (**Контурный**) для функции двух переменных

$X(t,\alpha) := t \cdot cos\quad(\alpha) \cdot sin\quad(\alpha)$, двумя способами:

1. С помощью функции *CreateMesh* (сетка размером 40 × 40, диапазон изменения *t* от –5 до 5, α - от 0 до 2·π).
2. Задав поверхность математически, для этого:
   - Определить функцию *X*(*t,*α)
   - Задать на осях переменных *t* и α по 41 точке

     $$i:=0..40 \quad j:=0..40$$

     для переменной t_i со значениями, изменяющимися от -5 до 5 с шагом 0.25 $t_i := -5 + 0.25 \cdot i$, а для переменной α_j - от 0 до 2·π с шагом π/20 $\alpha_j := \pi/20 \cdot j$.
   - Определить матрицу $M_{i\,j} := X(t_i,\alpha_j)$ и отобразить ее графически.

С помощью команды **Формат** контекстного меню вызвать диалоговое окно «**Формат 3-D графика»** и изменить:

- характеристики просмотра (**Общее/Вид/Вращение, Наклон**),
- цвета и линии поверхности (**Внешний Вид/Свойства линии, Свойства заливки**),
- параметры осей (**Оси**),
- вид заголовка графика (**Название**).

3) Отобразить графически пересечение поверхностей

$f1(x, y) := \frac{(x+y)^2}{10}$ и $f2(x, y) := 5 \cdot cos\left(\frac{x-y}{3}\right)$.

Матрицы для построения поверхностей задать с помощью функции *CreateMesh*, значения факультативных параметров не указывать. Выполнить однотонную заливку для поверхностей, выбрав из контекстного меню команду **Формат**. Также из контекстного меню выбрать эффекты **Туман**, **Освещение, Перспектива**.

Упражнение 5. Создание анимационных клипов

Используя переменную **FRAME** и команду **Вид ⇒ Анимация**, создать анимационные клипы с помощью данных, приведенных в таблице.

№ варианта	Переменные и функции	FRAME	Тип графика
1	$x := 0, 0.1 .. 30$ $f(x) := x + \text{FRAME}$	от 0 до 20	График Полярные Координаты
2	$i := 0 .. \text{FRAME} + 1$ $g_i := 0.5 \cdot i \cdot \cos(i)$ $h_i := i \cdot \sin(i)$ $k_i := 2 \cdot i$	от 0 до 50	3D точечный график границы на осях Min Max *x* - 50 50 *y* - 50 50 *z* 0 50 В метке для ввода матрицы укажите (*g, h, k*)
3	$i := 0 .. 20$ $j := 0 .. 20$ $f(x,y) := \sin(x^2 + y^2 + \text{FRAME})$ $x_i := -1.5 + 0.15 \cdot i$ $y_j := -1.5 + 0.15 \cdot j$ $M_{i,j} := f(x_i, y_j)$	от 0 до 50	График Поверхности В метке для ввода матрицы укажите *M*
4	$r := \text{FRAME}$ $R := 6$ $n := 0 .. 20$ $m := 0 .. 20$ $v_n := \frac{2 \cdot \pi \cdot n}{r+1}$ $w_m := \frac{2 \cdot \pi \cdot m}{r+1}$ $x_{m\,n} := (R + r \cdot \cos(v_n)) \cdot \cos(w_m)$ $y_{m\,n} := (R + r \cdot \cos(v_n)) \cdot \sin(w_m)$ $z_{m\,n} := r \cdot \sin(v_n)$	от 0 до 20	График Поверхности (границы на всех осях установить от -11 до 11) В метке для ввода матрицы укажите (*x, y, z*)

Контрольные вопросы

1. Чем отличается глобальное и локальное определение переменных?
2. Как изменить формат чисел для всего документа?
3. Какие системные (предопределенные) переменные Вам известны? Как изменить их значение?
4. Какие виды функций в Mathcad Вам известны?
5. Как определить дискретные переменные с произвольным шагом? Какой шаг по умолчанию?
6. Как определить индексированную переменную?
7. Какие виды массивов в Mathcad Вам известны?
8. Какая системная переменная определяет нижнюю границу индексации элементов массива?
9. Опишите способы создания массивов в Mathcad.
10. Как просмотреть содержимое массива, определенного через дискретный аргумент?
11. Как построить несколько графиков в одной системе координат?
12. Как определить координату точки на графике?
13. Как построить гистограмму?
14. Какие функции используются для построения трехмерных графиков?
15. Как создать анимацию в Mathcad?

Практическая работа №3.

Численное решение уравнений для расчета параметров оборудования

Численное решение уравнений для расчета параметров оборудования и выбора технологических схем. Решение систем уравнений: точные и итерационные.

Цель: научиться решать уравнения для расчета параметров оборудования

Как известно, многие уравнения и системы уравнений не имеют аналитических решений. В первую очередь это относится к большинству трансцендентных уравнений. Доказано также, что нельзя построить формулу, по которой можно было бы решить произвольное алгебраическое уравнение степени выше четвертой[1]. Однако такие уравнения могут решаться численными методами с заданной точностью (не более значения заданного системной переменной **TOL**).

Численное решение нелинейного уравнения

Для простейших уравнений вида $f(x) = 0$ решение в Mathcad находится с помощью функции root (Рисунок 5).

root($\boldsymbol{f(x1, x2, \dots), x1, a, b}$ **)**

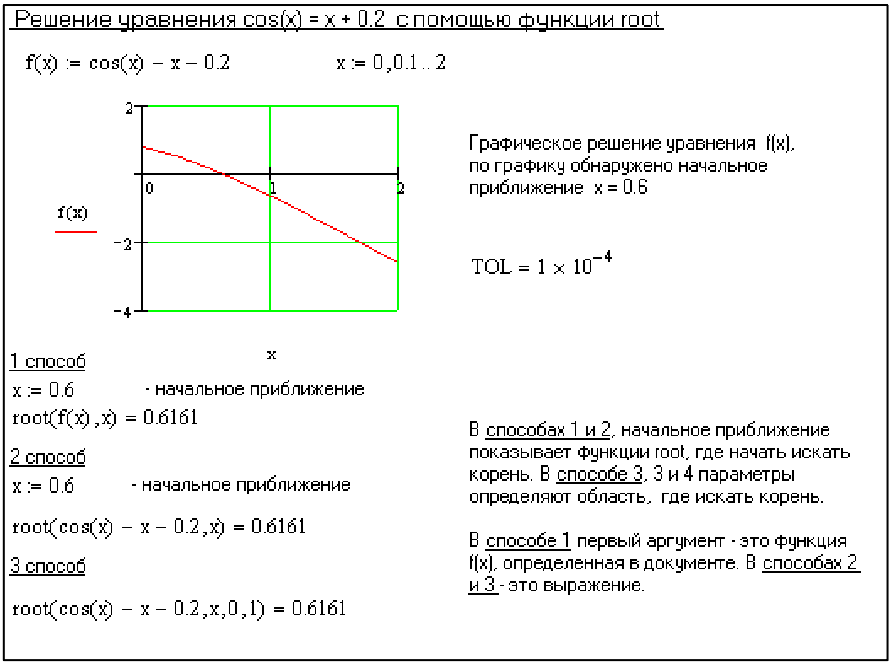


Рис. 5. Решение уравнений средствами Mathcad

Возвращает значение $\boldsymbol{x1}$, принадлежащее отрезку $\boldsymbol{[a, b]}$, при котором выражение или функция $\boldsymbol{f(x)}$ обращается в 0. Оба аргумента этой функции должны быть скалярами. Функция возвращает скаляр.

Аргументы:

$\boldsymbol{f(x1, x2, \dots)}$ **-** функция, определенная где-либо в рабочем документе, или выражение. Выражение должно возвращать скалярные значения.

$\boldsymbol{x1}$ **-** - имя переменной, которая используется в выражении. Этой переменной перед использованием функции *root* необходимо присвоить числовое значение. Mathcad использует его как начальное приближение при поиске корня.

$\boldsymbol{a, b}$ **–** необязательны, если используются, то должны быть вещественными числами, причем $\boldsymbol{a < b}$.

Приближенные значения корней (*начальные приближения*) могут быть:

1. Известны из физического смысла задачи.
2. Известны из решения аналогичной задачи при других исходных данных.
3. Найдены графическим способом.

Наиболее распространен *графический способ* определения начальных приближений. Принимая во внимание, что действительные корни уравнения $f(x) = 0$ - это точки пересечения графика функции $f(x)$ с осью абсцисс, достаточно построить график функции $f(x)$ и отметить точки пересечения $f(x)$ с

[1] Доказательство этого факта связано с именами замечательных математиков Абеля (1802-1829) и Галуа (1811-1832).

осью Ox, или отметить на оси Ox отрезки, содержащие по одному корню. Построение графиков часто удается сильно упростить, заменив уравнение $f(x) = 0$ *равносильным* ему уравнением: $f_1(x) = f_2(x)$, где функции $f_1(x)$ и $f_2(x)$ - более простые, чем функция $f(x)$. Тогда, построив графики функций $y = f_1(x)$ и $y = f_2(x)$, искомые корни получим как абсциссы точек пересечения этих графиков.

Отсутствие сходимости функции root

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

- Уравнение не имеет корней.
- Корни уравнения расположены далеко от начального приближения.
- Выражение имеет локальные *max* и *min* между начальным приближением и корнями.
- Выражение имеет разрывы между начальными приближениями и корнями.
- Выражение имеет комплексный корень, но начальное приближение было вещественным.

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

Рекомендации по использованию функции root

Для изменения точности, с которой функция root ищет корень, нужно изменить значение системной переменной TOL. Если значение TOL увеличивается, функция root будет сходиться быстрее, но ответ будет менее точен. Если значение TOL уменьшается, то функция root будет сходиться медленнее, но ответ будет более точен. Чтобы изменить значение TOL в определенной точке рабочего документа, используйте определение вида TOL:=0.01. Чтобы изменить значение TOL для всего рабочего документа, выберите команду **Математика/Параметры…/Переменные/Допуск сходимости (TOL)**.

- Если два корня расположены близко друг от друга, следует уменьшить TOL, чтобы различить их.
- Если функция $f(x)$ имеет малый наклон около искомого корня, функция $root(f(x), x)$ может *сходиться* к значению r, отстоящему от корня достаточно далеко. В таких случаях для нахождения более точного значения корня необходимо уменьшить значение TOL. Другой вариант заключается в замене уравнения $f(x) = 0$ на $g(x) = 0$
$$g(x) = \frac{f(x)}{\frac{d}{dx}f(x)}$$
- Для выражения $f(x)$ с известным корнем a нахождение дополнительных корней $f(x)$ эквивалентно поиску корней уравнения $h(x) = f(x)/(x - a)$. Подобный прием полезен для нахождения корней, расположенных близко друг к другу. Проще искать корень выражения $h(x)$, чем пробовать искать другой корень уравнения $f(x) = 0$, выбирая различные начальные приближения.

Нахождение корней полинома

Для нахождения корней выражения, имеющего вид

$$v_n x^n + \ldots + v_2 x^2 + v_1 x + v_0,$$

лучше использовать функцию *polyroots*, нежели *root*. В отличие от функции *root*, функция *polyroots* не требует начального приближения и возвращает сразу все корни, как вещественные, так и комплексные.

Polyroots(*v*)

Возвращает корни полинома степени n. Коэффициенты полинома находятся в векторе $\boldsymbol{v}$ длины $n + 1$. Возвращает вектор длины n, состоящий из корней полинома.

Аргументы:

v **–** вектор, содержащий коэффициенты полинома.

Вектор ***v*** удобно создавать использую команду **Символы/Коэффициенты полинома**. Рисунок 6 иллюстрирует определение корней полинома средствами Mathcad.

Решение систем уравнений

MathCAD дает возможность решать также и системы уравнений. Максимальное число уравнений и переменных равно 50. Результатом решения системы будет численное значение искомого корня.

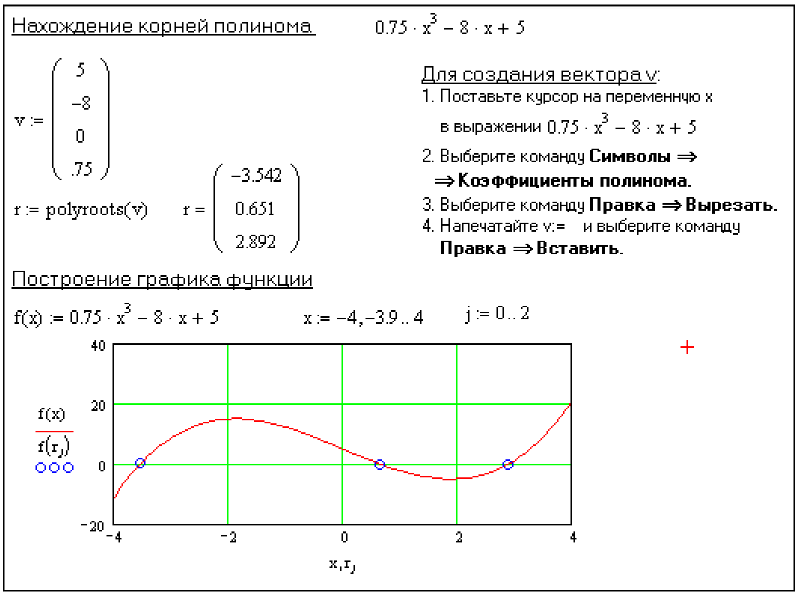


Рис. 6. Определение корней полинома

Для решения системы уравнений необходимо выполнить следующее:

- Задать начальное приближение для всех неизвестных, входящих в систему уравнений. Mathcad решает систему с помощью итерационных методов.
- Напечатать ключевое слово *Given*. Оно указывает Mathcad, что далее следует система уравнений.
- Введите уравнения и неравенства в любом порядке. Используйте **[Ctrl]=** для печати символа =. Между левыми и правыми частями неравенств может стоять любой из символов <, >, ≥ и ≤.

Введите любое выражение, которое включает функцию *Find*, например: *a*:= *Find*(*x, y*).

***Find*(*z*1, *z*2, . . .)**

Возвращает точное решение системы уравнений. Число аргументов должно быть равно числу неизвестных.

Ключевое слово *Given*, уравнения и неравенства, которые следуют за ним, и какое–либо выражение, содержащее функцию *Find*, называют ***блоком решения уравнений***.

Следующие выражения недопустимы внутри блока решения:

- Ограничения со знаком ≠.
- Дискретный аргумент или выражения, содержащие дискретный аргумент в любой форме.
- Неравенства вида $a < b < c$.

Блоки решения уравнений не могут быть вложены друг в друга, каждый блок может иметь только одно ключевое слово *Given* и имя функции *Find*.

Функция, которая завершает блок решения уравнений, может быть использована аналогично любой другой функции. Можно произвести с ней следующие три действия:

- Можно вывести найденное решение, напечатав выражение вида:

$$Find(var1, var2,\ldots) =.$$

- Определить переменную с помощью функции *Find*:

$$a := Find(x) \text{ – скаляр,}$$

$$var := Find(var1, var2,\ldots) \text{ – вектор.}$$

Это удобно сделать, если требуется использовать решение системы уравнений в другом месте рабочего документа.

- Определить другую функцию с помощью *Find*

$$f(a, b, c, \ldots) := Find(x, y, z, \ldots).$$

Эта конструкция удобна для многократного решения системы уравнений для различных значений некоторых параметров *a, b, c,…,* непосредственно входящих в систему уравнений.

Сообщение об ошибке No solution was found. Try changing the guess value or the value of TOL or CTOL. (Решение не найдено) при решении уравнений появляется, когда:

- Поставленная задача может не иметь решения.
- Для уравнения, которое не имеет вещественных решений, в качестве начального приближения взято вещественное число и наоборот.
- В процессе поиска решения последовательность приближений попала в точку локального минимума невязки. Для поиска искомого решения нужно задать различные начальные приближения.
- Возможно, поставленная задача не может быть решена с заданной точностью. Попробуйте увеличить значение TOL.

Пример 1 Рисунка 7 иллюстрирует решение системы уравнений в MathCAD.

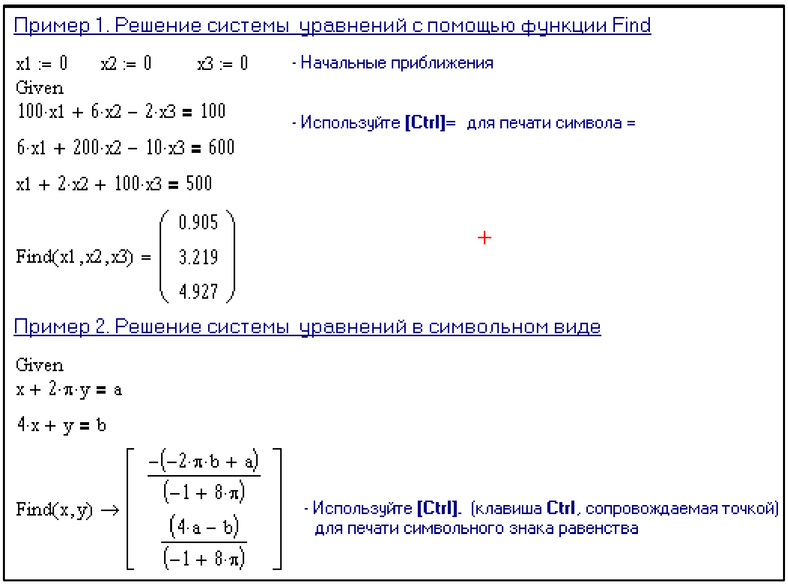


Рис. 7. Решение систем уравнений

Решение матричных[2] уравнений

Рассмотрим систему n линейных алгебраических уравнений относительно n неизвестных x_1, x_2, …, x_n:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1, \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2, \\ \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n = b_n. \end{cases} \qquad (2)$$

В соответствии с правилом умножения матриц рассмотренная система линейных уравнений может быть записана в матричном виде

$$Ax = b, \qquad (3)$$

где:

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{bmatrix}. \qquad (4)$$

Матрица A, столбцами которой являются коэффициенты при соответствующих неизвестных, а строками – коэффициенты при неизвестных в соответствующем уравнении, называется *матрицей системы*; матрица-столбец b, элементами которой являются правые части уравнений системы, называется *матрицей правой части* или просто *правой частью системы*. Матрица-столбец x, элементы

[2] *Матричным уравнением* называется уравнение, коэффициенты и неизвестные которого – прямоугольные матрицы соответствующей размерности.

которой - искомые неизвестные, называется *решением системы*.

Если матрица A - неособенная, то есть $\det A \neq 0$ то система (2), или эквивалентное ей матричное уравнение (3), имеет единственное решение.

В самом деле, при условии $\det A \neq 0$ существует обратная матрица A^{-1}. Умножая обе части уравнения (3) на матрицу A^{-1} получим:

$$A^{-1}Ax = A^{-1}b,$$
$$x = A^{-1}b. \quad (5)$$

Формула (5) дает решение уравнения (3) и оно единственно.

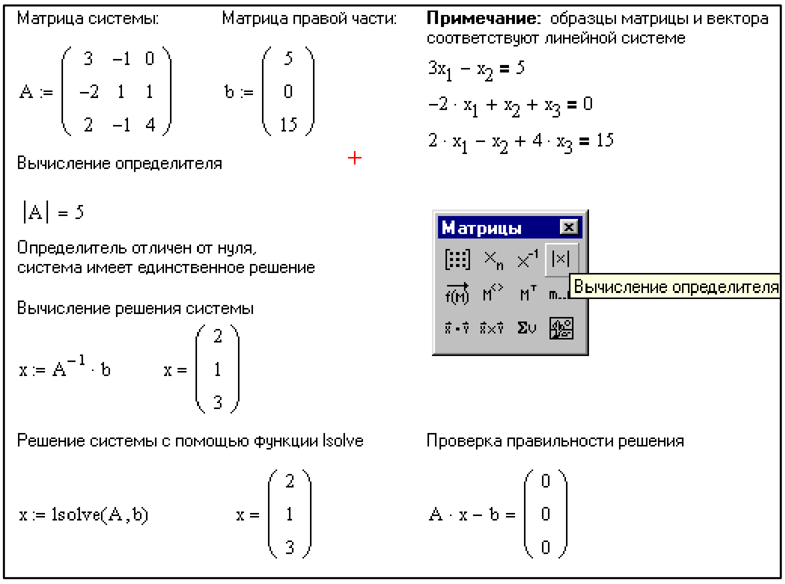


Рис. 8. Решение матричных уравнений

lsolve(*A, b*)
Возвращается вектор решения x такой, что $Ax = b$.
Аргументы:
A - квадратная, не сингулярная матрица.
b - вектор, имеющий столько же рядов, сколько рядов в матрице ***A***.

На Рисунке 8 показано решение системы трех линейных уравнений относительно трех неизвестных.

Приближенные решения

Функция *Minerr* очень похожа на функцию *Find* (использует тот же алгоритм). Если в результате поиска не может быть получено дальнейшее уточнение текущего приближения к решению, *Minerr* возвращает это приближение. Функция *Find* в этом случае возвращает сообщение об ошибке. Правила использования функции *Minerr* такие же, как и функции *Find.*

Minerr(*z*1, *z*2, . . .)

Возвращает приближенное решение системы уравнений. Число аргументов должно быть равно числу неизвестных.

Если *Minerr* используется в блоке решения уравнений, необходимо всегда включать дополнительную проверку достоверности результатов.

Символьное решение уравнений

В Mathcad можно быстро и точно найти численное значение корня с помощью функции *root*. Но имеются некоторые задачи, для которых возможности Mathcad позволяют находить решения в символьном (аналитическом) виде. Решение уравнений в символьном виде позволяет найти точные или приближенные корни уравнения: если решаемое уравнение имеет параметр, то решение в символьном виде может выразить искомый корень непосредственно через параметр. Поэтому вместо того, чтобы решать уравнение для каждого нового значения параметра, можно просто заменять его значение в найденном символьном решении; если нужно найти все комплексные корни полинома со степенью меньше или равной 4, символьное решение даст их точные значения в одном векторе или в аналитическом или цифровом виде.

Команда **Символы/Переменные/Вычислить** позволяет решить уравнение относительно некоторой переменной и выразить его корни через остальные параметры уравнения.

Чтобы решить уравнение символьно необходимо:

- Напечатать выражение (для ввода знака равенства используйте комбинацию клавиш **[Ctrl]=**).
- Выделить переменную, относительно которой нужно решить уравнение, щелкнув на ней мышью.
- Выбрать пункт меню **Символы/Переменные/Вычислить**.

Нет необходимости приравнивать выражение нулю. Если MathCAD не находит знака равенства, он предполагает, что требуется приравнять выражение нулю.

Чтобы решить систему уравнений в символьном виде, необходимо выполнить следующее:

- Напечатать ключевое слово *Given*.
- Напечатать уравнения в любом порядке ниже слова *Given*. Удостоверьтесь, что для ввода знака = используется **[Ctrl]=**.
- Напечатать функцию *Find*, соответствующую системе уравнений.
- Нажать **[Ctrl].** (клавиша CTRL, сопровождаемая точкой). Mathcad отобразит символьный знак равенства →.
- Щелкнуть мышью на функции *Find*.

Пример 2 Рисунка 7 иллюстрирует символьное решение системы уравнений в MathCAD.

ПРАКТИЧЕСКОЕ ЗАДАНИЕ

Упражнение 1. Построение графиков функций, решение уравнений

Построить график функции $f(x)$ (Таблица 2) и приблизительно определить один из корней уравнения. Решить уравнение $f(x)=0$ с точностью $\varepsilon = 10^{-4}$ с помощью встроенной функции *root*;

№ варианта	$f(x)$	№ варианта	$f(x)$
1	$e^{x-1} - x^3 - x$ $x \in [0,\ 1]$	9	$0.25x^3 + x - 2$ $x \in [0,\ 2]$
2	$x - \dfrac{1}{3 + sin(3.6x)}$ $x \in [0,\ 1]$	10	$arccos\dfrac{1 - x^2}{1 + x^2} - x$ $x \in [2, 3]$
3	$arccos\quad x - \sqrt{1 - 0.3x^3}$ $x \in [0,\ 1]$	11	$3x - 4\,ln\,x - 5$ $x \in [2,\ 4]$
4	$\sqrt{1 - 0.4x^2} - arcsin\quad x$ $x \in [0,\ 1]$	12	$e^x - e^{-x} - 2$ $x \in [0,\ 1]$
5	$3x - 14 + e^x - e^{-x}$ $x \in [1,\ 3]$	13	$\sqrt{1 - x} - \mathrm{tg}\ x$ $x \in [0,\ 1]$
6	$\sqrt{2x^2 + 1.2 - cos\,x} - 1$ $x \in [0,\ 1]$	14	$1 - x + sin\,x - ln(1 + x)$ $x \in [0,\ 2]$
7	$cos\left(\dfrac{2}{x}\right) - 2\,sin\left(\dfrac{1}{x}\right) + \dfrac{1}{x}$ $x \in [1,\ 2]$	15	$x^5 - x - 0{,}2$ $x \in [1, 2]$
8	$0.1x^2 - x\,ln\,x$ $x \in [1,\ 2]$		

Для полинома $g(x)$ (Таблица 3) выполнить следующие действия:

1) с помощью команды **Символы/Коэффициенты полинома** создать вектор V, содержащий коэффициенты полинома;
2) решить уравнение $g(x) = 0$ с помощью функции *polyroots*;
3) решить уравнение символьно, используя команду **Символы/Переменные/Вычислить**.

№ варианта	$g(x)$	№ варианта	$g(x)$
1	$x^4 - 2x^3 + x^2 - 12x + 20$	9	$x^4 + x^3 - 17x^2 - 45x - 100$
2	$x^4 + 6x^3 + x^2 - 4x - 60$	10	$x^4 - 5x^3 + x^2 - 15x + 50$
3	$x^4 - 14x^2 - 40x - 75$	11	$x^4 - 4x^3 - 2x^2 - 20x + 25$
4	$x^4 - x^3 + x^2 - 11x + 10$	12	$x^4 + 5x^3 + 7x^2 + 7x - 20$
5	$x^4 - x^3 - 29x^2 - 71x - 140$	13	$x^4 - 7x^3 + 7x^2 - 5x + 100$
6	$x^4 + 7x^3 + 9x^2 + 13x - 30$	14	$x^4 + 10x^3 + 36x^2 + 70x + 75$
7	$x^4 + 3x^3 - 23x^2 - 55x - 150$	15	$x^4 + 9x^3 + 31x^2 + 59x + 60$

Упражнение 2. Решение систем линейных и нелинейных уравнений

1. Решить систему линейных уравнений:
   - используя функцию *Find*;
   - матричным способом;
   - используя функцию *lsolve*.

№ варианта	Система линейных уравнений	№ варианта	Система линейных уравнений
1	$\begin{cases} 2x_1 + x_2 + 2x_3 + 3x_4 = 8 \\ 3x_1 + 3x_3 = 6 \\ 2x_1 - x_2 + 3x_4 = 4 \\ x_1 + 2x_2 - x_3 + 2x_4 = 4 \end{cases}$	9	$\begin{cases} 2x_1 + x_2 - 5x_3 + x_4 = -4 \\ x_1 - 3x_2 - 6x_4 = -7 \\ 2x_2 - x_3 + 2x_4 = 2 \\ x_1 + 4x_2 - 7x_3 + 6x_4 = -2 \end{cases}$
2	$\begin{cases} x_1 + 2x_2 + 3x_3 + 4x_4 = 22 \\ 2x_1 + 3x_2 + x_3 + 2x_4 = 17 \\ x_1 + x_2 + x_3 - x_4 = 8 \\ x_1 - 2x_3 - 3x_4 = -7 \end{cases}$	10	$\begin{cases} x_1 + 2x_2 + 3x_3 + 4x_4 = 26 \\ 2x_1 + 3x_2 + 4x_3 + x_4 = 34 \\ 3x_1 + 4x_2 + x_3 + 2x_4 = 26 \\ 4x_1 + x_2 + 2x_3 + 3x_4 = 26 \end{cases}$
3	$\begin{cases} 6x_1 - x_2 + 10x_3 - x_4 = 158 \\ 2x_1 + x_2 + 10x_3 + 7x_4 = 128 \\ 3x_1 - 2x_2 - 2x_3 - x_4 = 7 \\ x_1 - 12x_2 + 2x_3 - x_4 = 17 \end{cases}$	11	$\begin{cases} 2x_1 - x_2 + 4x_3 + x_4 = 66 \\ 2x_2 - 6x_3 + x_4 = -63 \\ 8x_1 - 3x_2 + 6x_3 - 5x_4 = 146 \\ 2x_1 - 7x_2 + 6x_3 - x_4 = 80 \end{cases}$
4	$\begin{cases} x_1 - 2x_2 + 6x_3 + x_4 = 88 \\ 5x_1 + 2x_3 - 3x_4 = 88 \\ 7x_1 - 3x_2 + 7x_3 + 2x_4 = 181 \\ 3x_1 - 7x_2 + 5x_3 + 2x_4 = 99 \end{cases}$	12	$\begin{cases} 2x_1 - 3x_3 - 2x_4 = -16 \\ 2x_1 - x_2 + 13x_3 + 4x_4 = 213 \\ 3x_1 + x_2 + 2x_3 + x_4 = 72 \\ x_1 - 12x_3 - 5x_4 = -159 \end{cases}$
5	$\begin{cases} x_1 - 2x_2 - 8x_4 = -7 \\ x_1 + 4x_2 - 7x_3 + 6x_4 = -8 \\ x_1 + x_2 - 5x_3 + x_4 = -10 \\ 2x_1 - x_2 + 2x_4 = 7 \end{cases}$	13	$\begin{cases} 7x_1 + 7x_2 - 7x_3 - 2x_4 = 5 \\ 3x_1 + 4x_2 + 5x_3 + 8x_4 = 60 \\ 2x_1 + 2x_2 + 2x_3 + x_4 = 27 \\ 2x_1 - 2x_3 - x_4 = -1 \end{cases}$
6	$\begin{cases} 2x_1 + 2x_2 + 6x_3 + x_4 = 15 \\ -x_2 + 2x_3 + x_4 = 18 \\ 4x_1 - 3x_2 + x_3 - 5x_4 = 37 \\ 3x_1 - 5x_2 + x_3 - x_4 = 30 \end{cases}$	14	$\begin{cases} 6x_1 - 9x_2 + 5x_3 + x_4 = 124 \\ 7x_2 - 5x_3 - x_4 = -54 \\ 5x_1 - 5x_2 + 2x_3 + 4x_4 = 83 \\ 3x_1 - 9x_2 + x_3 + 6x_4 = 45 \end{cases}$
7	$\begin{cases} 4x_1 - 5x_2 + 7x_3 + 5x_4 = 165 \\ 2x_1 + x_2 - 3x_3 - x_4 = -15 \\ 9x_1 + 4x_3 - x_4 = 194 \\ x_1 - x_2 - 2x_3 - 3x_4 = -19 \end{cases}$	15	$\begin{cases} x_1 + 2x_2 + 3x_3 + 4x_4 = 26 \\ 2x_1 + 3x_2 + 4x_3 + x_4 = 34 \\ 3x_1 + 4x_2 + x_3 + 2x_4 = 26 \\ 4x_1 + x_2 + 2x_3 + 3x_4 = 26 \end{cases}$

2. Преобразовать нелинейные уравнения системы из Таблицы 4 к виду $f_1(x) = y$ и $f_2(y) = x$. Построить их графики и определить начальное приближение решения. Решить систему нелинейных уравнений с помощью функции *Minerr*.

№ варианта	**Система нелинейных уравнений**	**№ варианта**	**Система нелинейных уравнений**
1	$\begin{cases} sin\, x + 2y = 2, \\ cos(y-1) + x = 0{,}7. \end{cases}$	9	$\begin{cases} sin\, y + x = -0{,}4, \\ 2y - cos(x+1) = 0. \end{cases}$
2	$\begin{cases} sin(x+0{,}5) - y = 1, \\ cos(y-2) + x = 0. \end{cases}$	10	$\begin{cases} sin(x+2) - y = 1{,}5, \\ cos(y-2) + x = 0{,}5. \end{cases}$
3	$\begin{cases} cos\, x + y = 1{,}5, \\ 2x - sin(y-0{,}5) = 1. \end{cases}$	11	$\begin{cases} cos(x+0{,}5) - y = 2, \\ sin\, y - 2x = 1. \end{cases}$
4	$\begin{cases} cos(x+0{,}5) + y = 0{,}8, \\ sin\, y - 2x = 1{,}6. \end{cases}$	12	$\begin{cases} cos(x-2) + y = 0, \\ sin(y+0{,}5) - x = 1. \end{cases}$
5	$\begin{cases} sin(x-1) = 1.3 - y, \\ x - sin(y+1) = 0.8. \end{cases}$	13	$\begin{cases} cos(x+0{,}5) + y = 1, \\ sin(y+0{,}5) - x = 1. \end{cases}$
6	$\begin{cases} cos(x+0{,}5) + y = 1, \\ sin\, y - 2x = 2. \end{cases}$	14	$\begin{cases} sin(x) - 2y = 1, \\ cos(y+0{,}5) - x = 2. \end{cases}$
7	$\begin{cases} -sin(x+1) + y = 0{,}8, \\ sin(y-1) + x = 1{,}3. \end{cases}$	15	$\begin{cases} 2y - sin(x-0{,}5) = 1, \\ cos(y) + x = 1{,}5. \end{cases}$

3. Символьно решить системы уравнений:

$$\begin{cases} 3x + 4\pi y = a, \\ 2x + y = b. \end{cases} \qquad \begin{cases} 2y - \pi z = a, \\ \pi z - z = b, \\ 3y + x = c. \end{cases}$$

Контрольные вопросы

1. Назовите способы нахождения начального приближения.
2. Какие функции для решения одного уравнения в MathCAD вы знаете? В чем их отличие?
3. Какие аргументы функции *root* не обязательны?
4. В каких случаях MathCAD не может найти корень уравнения?
5. Какая системная переменная отвечает за точность вычислений?
6. Как изменить точность, с которой функция *root* ищет корень?
7. Как системная переменная TOL влияет на решение уравнения с помощью функции root?
8. Назовите функции для решения систем уравнений в MathCAD и особенности их применения.
9. Опишите структуру блока решения уравнений.
10. Какие выражения не допустимы внутри блока решения уравнения?
11. Опишите способы использования функции *Find*.
12. В каких случаях MathCAD не может найти решение системы уравнений?
13. Дайте сравнительную характеристику функциям *Find* и *Minerr*.
14. Какие уравнения называются матричными?
15. Как решать матричные уравнения? Назовите способы решения матричных уравнений.

Практическая работа №4.

Символьные вычисления необходимые для обработки информации, полученной в результате эксперимента

Символьные операции. Инструментальная панель символьных операций. Стиль представления результатов вычислений.

Цель: научиться выполнять аналитические (символьные) вычисления, задавать операторы пользователя

Системы компьютерной алгебры снабжаются специальным процессором для выполнения аналитических (символьных) вычислений. Его основой является ядро, хранящее всю совокупность формул и формульных преобразований, с помощью которых производятся аналитические вычисления. Чем больше этих формул в ядре, тем надежней работа символьного процессора и тем вероятнее, что поставленная задача будет решена, если такое решение существует в принципе (что бывает далеко не всегда).

Ядро символьного процессора системы MathCAD – несколько упрощенный вариант ядра известной системы символьной математики Maple V фирмы Waterloo Maple Software, у которой фирма MathSoft (разработчик MathCAD) приобрела лицензию на его применение, благодаря чему MathCAD стала (начиная с версии 3. 0) системой символьной математики. Символьные вычисления выполняются столь же просто (для пользователя), как вычисление квадрата *x*.

Символьные операции можно выполнять двумя способами:

- Непосредственно в командном режиме (используя операции меню **Символы**);
- С помощью операторов символьного преобразования (используя **палитру инструментов Символы** ).

Выделение выражений для символьных вычислений

Чтобы символьные операции выполнялись, процессору необходимо указать, над каким выражением эти операции должны производиться, т. е. надо выделить выражение. Для ряда операций следует не только указать выражение, к которому они относятся, но и наметить переменную, относительно которой выполняется та или иная символьная операция. Само выражение в таком случае не выделяется.

Таким образом, для выполнения операций с символьным процессором нужно выделить объект (целое выражение или его часть) синими сплошными линиями.

Символьные операции разбиты на пять характерных разделов. Первыми идут наиболее часто используемые операции. Они могут выполняться с выражениями, содержащими комплексные числа или имеющими решения в комплексном виде.

Символьные операции

Операции с выделенными выражениями

Если в документе есть выделенное выражение, то с ним можно выполнять различные операции, представленные ниже:

Расчеты – преобразовать выражение с выбором вида преобразований из подменю;

Символические [Shift] F9 – выполнить символьное преобразование выделенного выражения;

С плавающей запятой… – вычислить выделенное выражение в вещественных числах;

Комплексные – выполнить вычисления в комплексном виде;

Упростить — упростить выделенное выражение с выполнением таких операций, как сокращение подобных слагаемых, приведение к общему знаменателю, использование основных тригонометрических тождеств и т д.;

Расширить – раскрыть выражение [например, для $(X + Y)(X - Y)$ получаем $X^2 - Y^2$];

Фактор – разложить число или выражение на множители [например, $X^2 - Y^2$ даст $(X + Y)(X - Y)$];

Подобные – собрать слагаемые, подобные выделенному выражению, которое может быть отдельной переменной или функцией со своим аргументом (результатом будет выражение, полиномиальное относительно выбранного выражения);

Коэффициенты Полинома – по заданной переменной найти коэффициенты полинома, аппроксимирующего выражение, в котором эта переменная использована.

Операции с выделенными переменными

Для ряда операций надо знать, относительно какой переменной они выполняются. В этом случае необходимо выделить переменную, установив на ней маркер ввода. После этого становятся доступными следующие операции подменю **Переменные**:

Вычислить – найти значения выделенной переменной, при которых содержащее ее выражение становится равным нулю;

Замена – заменить указанную переменную содержимым буфера обмена;

Дифференциалы – дифференцировать выражение, содержащее выделенную переменную, по этой переменной (остальные переменные рассматриваются как константы);

Интеграция – интегрировать все выражение, содержащее переменную, по этой переменной;

Разложить на составляющие... – найти несколько членов разложения выражения в ряд Тейлора относительно выделенной переменной;

Преобразование в Частичные Доли – разложить на элементарные дроби выражение, которое рассматривается как рациональная дробь относительно выделенной переменной.

Операции с выделенными матрицами

Операции с выделенными матрицами представлены позицией подменю **Матрицы**, которая имеет свое подменю со следующими операциями:

Транспонирование – получить транспонированную матрицу;

Инвертирование – создать обратную матрицу;

Определитель – вычислить детерминант (определитель) матрицы.

Результаты символьных операций с матрицами часто оказываются чрезмерно громоздкими и поэтому плохо обозримы.

Операции преобразования

В позиции **Преобразование** содержится раздел операций преобразования, создающий подменю со следующими возможностями:

Фурье – выполнить прямое преобразование Фурье относительно выделенной переменной;

Фурье Обратное – выполнить обратное преобразование Фурье относительно выделенной переменной;

Лапласа – выполнить прямое преобразование Лапласа относительно выделенной переменной (результат – функция переменной s);

Лапласа Обратное – выполнить обратное преобразование Лапласа относительно выделенной переменной (результат – функция
переменной *t*);

Z – выполнить прямое Z-преобразование выражения относительно выделенной переменной (результат – функция переменной *z*);

Обратное Z – выполнить обратное Z-преобразование относительно выделенной переменной.

Стиль представления результатов вычислений

На наглядность вычислений влияет стиль представления их результатов. Следующая команда позволяет задать тот или иной стиль:

Стиль Вычислений... – задать вывод результата символьной операции под основным выражением, рядом с ним или вместо него

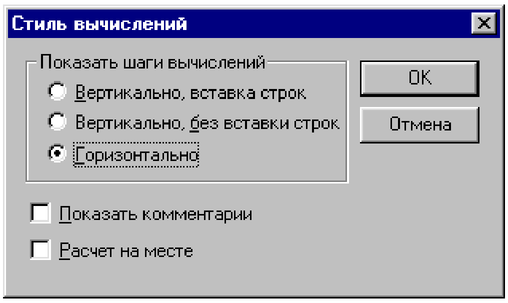


Рис. 9. Стиль представления результатов

Примеры символьных операций в командном режиме

Большинство символьных операций легко выполняются, так что ниже мы остановимся лишь на некоторых примерах. Символьная операция **Расчеты** обеспечивает работу с математическими выражениями, содержащими встроенные в систему функции и представленными в различном виде:

полиномиальном, дробно-рациональном, в виде сумм и произведений, производных и интегралов и т. д. (Рисунок 10). Операция стремится произвести все возможные численные вычисления и представить выражение в наиболее простом виде. Она возможна над матрицами с символьными элементами. Производные и определенные интегралы, символьные значения которых вычисляются, должны быть представлены в своей естественной форме.

Особо следует отметить возможность выполнения численных вычислений с повышенной точностью – 20 знаков после запятой. Для перехода в такой режим вычислений нужно числовые константы в вычисляемых объектах задавать с обязательным указанием десятичной точки, например 10.0 или 3.0, а не 10 или 3. Этот признак является указанием на проведение вычислений такого типа.

На Рисунке 10 показаны типовые примеры действия операции **Расчеты.**

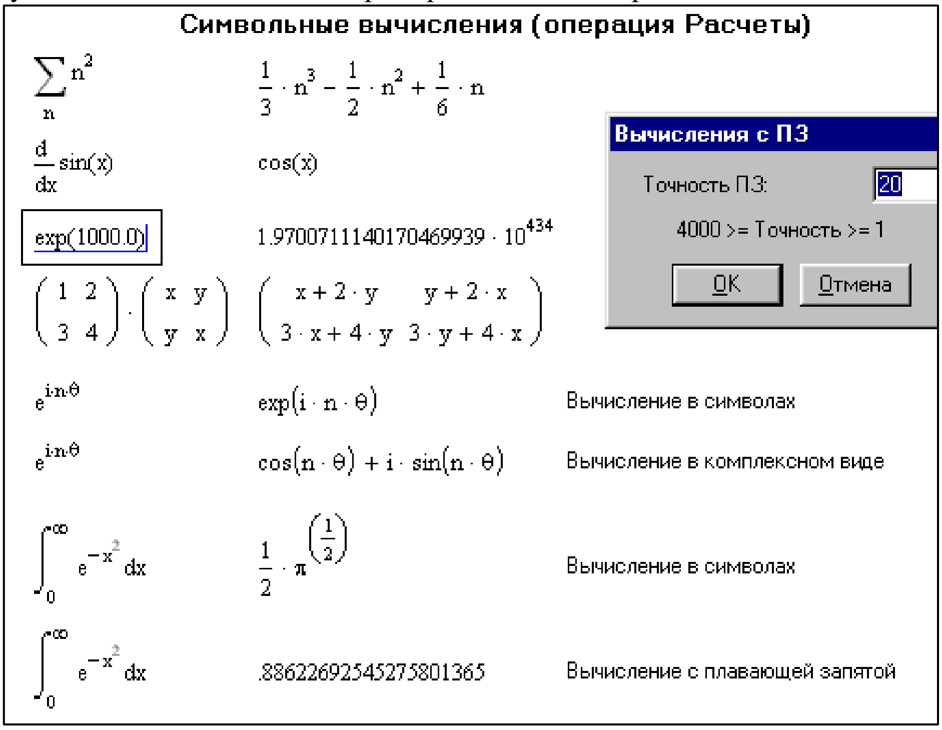


Рис. 10. Символьные вычисления

На рисунке 10 слева показаны исходные выражения, подвергаемые символьным преобразованиям, а справа – результат этих преобразований.

Операция **Расчеты** одна из самых мощных. Как видно из Рисунка позволяет в символьном виде вычислять суммы (и произведения) рядов, производные и неопределенные интегралы, выполнять символьные и численные операции с матрицами.

Эта операция содержит подменю. Команда **Символические** тут наиболее важная. Назначение других команд очевидно: они нужны, если результат требуется получить в форме комплексного или действительного числа. К примеру, если вы хотите вместо числа π получить 3.141..., используйте команду **С плавающей запятой….** В режиме символьных вычислений результат может превосходить машинную бесконечность системы — пример на вычисление exp(1000.0) на рисунке 10. При этом число точных значащих цифр результата практически не ограничено (или, точнее говоря, зависит от емкости ОЗУ).

Операция **Разложить на составляющие...** возвращает разложение в ряд Тейлора выражения относительно выделенной переменной с заданным по запросу числом членов ряда n (число определяется по степеням ряда). По умолчанию задано $n = 6$. В разложении указывается остаточная погрешность разложения. На Рисунке 11 представлено применение этой операции для разложения функции $\frac{sin(x)}{x}$. Минимальная погрешность получается при малых x (см. графическое представление функции и ее ряда).

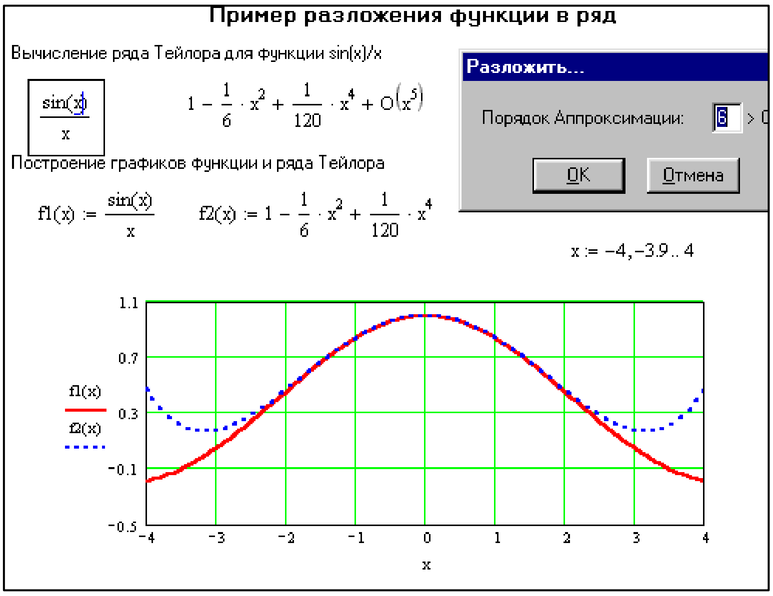


Рис. 11. Разложение функции в ряд Тейлора

Операторы вычисления пределов функций

Для вычисления пределов функций в систему введен символьный оператор limit. Помимо ввода с наборной панели **Матанализ,** его в трех формах можно ввести нажатием следующих комбинаций клавиш:

[Ctrl] **L** — ввод шаблона оператора вычисления предела функции при *x,* стремящемся к заданному значению,

[Ctrl] **A** — ввод шаблона вычисления предела функции слева от заданной точки,

[Ctrl] **B** — ввод шаблона вычисления предела функции справа от заданной точки.

На рисунке 12 показаны примеры вычисления пределов.

Для получения результата установите после блока вычисления предела стрелку с острием, направленным вправо. Предел (если он существует) будет вычислен и появится в шаблоне у острия стрелки. Если функция не имеет предела, вместо результата появится надпись *Undefine*.

Задание операторов пользователя

Еще одна экзотическая возможность, присущая новым версиям системы MathCAD, — задание новых операторов пользователя. Такой оператор задается практически так же, как функция пользователя, но вместо имени выбирается какой-либо подходящий знак. Например, можно задать оператор деления в виде:

$\div(A,B) := \frac{A}{B}$ – задание нового оператора деления;

$\div(6,2) = 3$ – применение функции деления;

$6 \div 2 = 3$ – применение нового оператора деления.

При кажущейся простоте такого задания здесь есть проблемы. Встроенные в систему операторы <u>нельзя переопределить</u>. Поэтому набор доступных знаков для обозначения новых операторов ограничен. Нельзя задать новый оператор деления знаком / (он уже использован), но можно взять знак ÷ , поскольку этот символ системой не используется.

Вторая проблема связана с вводом символа нового оператора. Скорее всего, его напрямую ввести нельзя. Придется воспользоваться типовыми приемами ввода новых символов в документы Windows. Один из этих приемов — использование приложения, выдающего таблицу символов, с возможностью его экспорта из этой таблицы в документ другого приложения (в нашем случае — в документ MathCAD).

Можно также воспользоваться подходящим знаком из набора MATH SYMBOL, имеющегося в составе Шпаргалок, доступ к которым дает Ресурс Центр (**? / Ресурс Центр / Справочный стол и краткое руководство / Дополнительные математические символы**). На Рисунке 12 показан такой вариант задания нового оператора пользователя. Для перетаскивания знака можно скопировать его в буфер обмена с помощью операции **Копировать**, а затем ввести в документ, используя операцию **Вставка.**

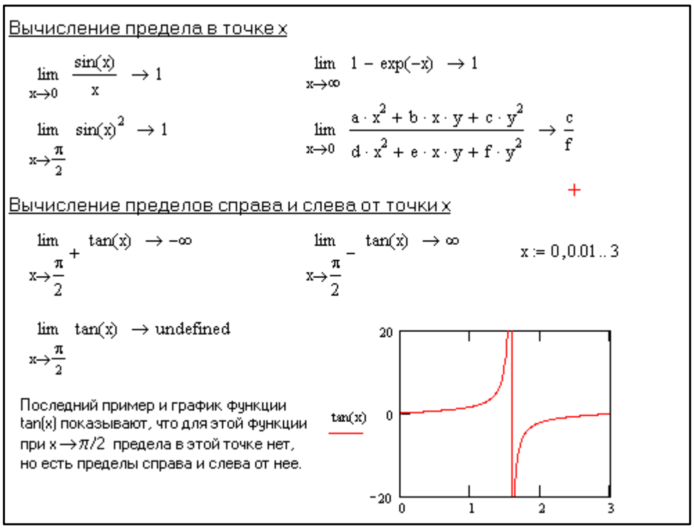


Рис. 12. Вычисление пределов

После того как оператор задан, его можно использовать, как функцию и как оператор. Примеры показаны на рисунке 13. Для применения нового оператора надо вывести его шаблон с помощью панели математических знаков (она также показана Рисунке 13). В нашем случае следует нажать кнопку xfy этой панели — она выводит особый шаблон вида ▪ ▪ ▪. Введите операнды, например 6 и 3 в крайние прямоугольники, а символ оператора — в средний. Поставив после этой конструкции знак равенства, увидите результат — число 2.

Можно задать и другие операторы, например, для работы с одним операндом. Так, вы можете задать оператор для пересчета значения температуры по шкале Цельсия, с тем чтобы определить соответствующее ему значение по шкале Фаренгейта, следующим образом

$$°C(x) := \frac{9}{5} \cdot x + 32 \qquad °F := 1$$

Затем, используя кнопку x f наборной панели символов отношения, можно выполнять операцию пересчета в виде.

$$37\,°C = 98.6\,°F$$

Есть области математики и физики, где задание новых операторов необходимо, поскольку является частью специфического языка их описания.

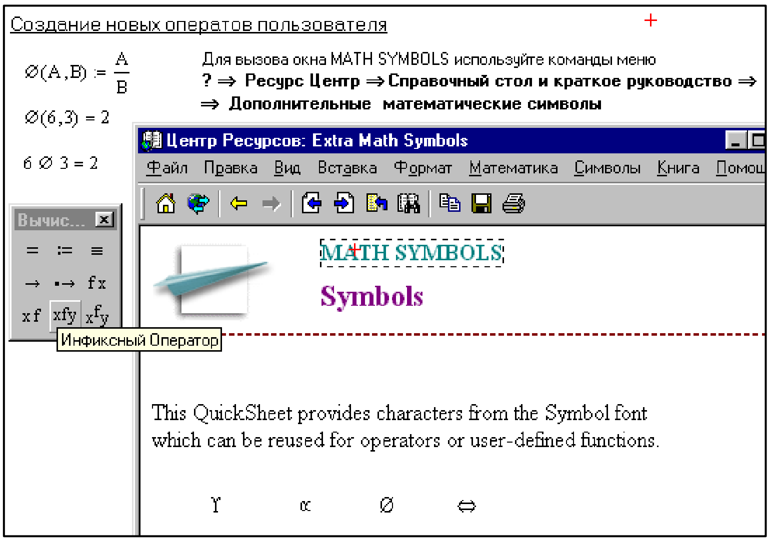


Рис. 13. Задание оператора пользователя с выбором имени из набора знаков

ПРАКТИЧЕСКОЕ ЗАДАНИЕ

Упражнение 1. Использование символьных операций

Используя операцию **Символы/ Расчеты / С плавающей запятой…**, представьте:

1) число π в 7 позициях;
2) число 12, 345667 в 3 позициях.

Выведите следующие числа в комплексной форме, используя операцию **Расчеты** $\Rightarrow$ **Комплексные** меню **Символы**:

1) $\sqrt{-7}$; 2) tg (a $\sqrt{-3}$); 3) $e^{1+\frac{\pi}{4}i}$;

для выражения 3 последовательно выполните операции **Расчеты** / **Комплексные** и **Упростить** меню **Символы.**

Упражнение 2. Преобразования полинома

Для полинома $g(x)$ (таблица) выполнить следующие действия:

1) разложить на множители, используя операцию **Символы / Фактор;**
2) подставьте выражение $x = y + z$ в $g(x)$, используя операцию **Символы / Переменные / Замена** (предварительно скопировав подставляемое выражение в буфер обмена, выделив его и нажав комбинацию клавиш **Ctrl + C**);
3) используя операцию **Символы / Расширить,** разложите по степеням выражение, полученное в 2)**;**
4) используя операцию **Символы / Подобные,** сверните выражение, полученное в 3, по переменной z.

№ варианта	$g(x)$	**№ варианта**	$g(x)$
1	$x^4 - 2x^3 + x^2 - 12x + 20$	9	$x^4 + x^3 - 17x^2 - 45x - 100$
2	$x^4 - 2x^3 + x^2 - 12x + 20$	10	$x^4 - 5x^3 + x^2 - 15x + 50$
3	$x^4 + 6x^3 + x^2 - 4x - 60$	11	$x^4 - 4x^3 - 2x^2 - 20x + 25$
4	$x^4 - 14x^2 - 40x - 75$	12	$x^4 + 5x^3 + 7x^2 + 7x - 20$
5	$x^4 - x^3 + x^2 - 11x + 10$	13	$x^4 - 7x^3 + 7x^2 - 5x + 100$
6	$x^4 - x^3 - 29x^2 - 71x -140$	14	$x^4 + 10x^3 +36x^2 +70x+ 75$
7	$x^4 + 7x^3 + 9x^2 + 13x - 30$	15	$x^4 + 9x^3 + 31x^2 + 59x+ 60$

Упражнение 3. Разложение выражения

Разложите выражения на элементарные дроби используя операцию **Символы / Переменные / Преобразование в частичные доли:**

1) $\frac{6x^2-x+1}{x^3-x}$; 2) $\frac{3x^2-2}{(x^2+x+1)(x+1)}$;

3) $\frac{x+1}{x(x-1)^3}$; 4) $\frac{5x^2-4x+16}{(x^2-x+1)^2(x-3)}$.

Разложите выражения в ряд с заданной точностью, используя операцию **Символы / Переменные / Разложить на составляющие:**

1) ln (1 + x), $x_0 = 0$, порядок разложения 6;
2) sin $(x)^2$, $x_0 = 0$, порядок разложения 6.

Упражнение 4. Задание операторов пользователя

Задайте операторы пользователя:

1) Для пересчета единиц электрической энергии (кВт·ч в Дж, эВ в Дж) если известно, что

   1 кВт·ч = $3{,}6 \cdot 10^6$ Дж;

   1 эВ = $1{,}602 \cdot 10^{-19}$ Дж.

2) Для пересчета единиц магнитной индукции (Вб/см^2 в Т, Гс в Т) если известно, что

1 Вб/см^2 = 1 · 10^4 Т;
1 Гс = 1 · 10^{-4} Т.

3) Для пересчета единиц мощности (эрг/с в Вт, кгс·м/с в Вт) если известно, что
1 эрг/с = 1 · 10^{-7} Вт;
1 кгс·м/с = 9,80665 Вт.

Для пересчета курсов валют (евро в рубли, рубли в доллары, доллары в евро и т.д.)

Контрольные вопросы

1. Назовите способы выполнения символьных операций в MathCAD.
2. Перечислите символьные операции с выделенными выражениями.
3. Перечислите символьные операции с выделенными переменными.
4. Перечислите символьные операции с выделенными матрицами.
5. Перечислите символьные операции преобразования.
6. Какие параметры определяет стиль представления результатов вычислений и где он задается?
7. В каких случаях результат символьных преобразований помещается в буфер обмена?
8. Каким образом можно вычислить предел в MathCAD?
9. Для чего необходимо задание операторов пользователя?
10. Как задать оператор пользователя?

Практическая работа №5 (часть1, 2)

Интерполяция и экстраполяция. Предсказание

Глобальная и локальная интерполяция: линейная, кубическая, сплайн-интерполяция, интерполяция вектора точек. Экстраполяция и предсказание. Импорт и экспорт данных.

Цель: изучить методы интерполяции и экстраполяции для прогнозирования значений

Аппроксимация функций заключается в приближенной замене заданной функции $f(x)$ некоторой функцией $\varphi(x)$ так, чтобы отклонение функции $\varphi(x)$ от $f(x)$ в заданной области было наименьшим. Функция $\varphi(x)$ при этом называется *аппроксимирующей*. Типичной задачей аппроксимации функций является задача *интерполяции*. Необходимость *интерполяции* функций в основном связана с двумя причинами:

1) Функция $f(x)$ имеет сложное аналитическое описание, вызывающее определенные трудности при его использовании (например, $f(x)$ является спецфункцией: гамма-функцией, эллиптической функцией и др.).
2) Аналитическое описание функции $f(x)$ неизвестно, т. е. $f(x)$ задана таблично. При этом необходимо иметь аналитическое описание, приближенно представляющее $f(x)$ (например, для вычисления значений $f(x)$ в произвольных точках, определения интегралов и производных от $f(x)$ и т. п.)

Простейшая задача *интерполяции* заключается в следующем. На отрезке $[a, b]$ заданы $n + 1$ точки $x_i = x_0, x_1, \ldots, x_n$, которые называются *узлами интерполяции,* и значения некоторой функции $f(x)$ в этих точках

$$f(x_0) = y_0,\ f(x_1) = y_1,\ \ldots, f(x_n) = y_n. \qquad (1)$$

Требуется построить функцию $\Phi(x)$ (*интерполяционная функция*), принадлежащую известному классу и принимающую в узлах интерполяции те же значения, что и $f(x)$, т. е. такую, что

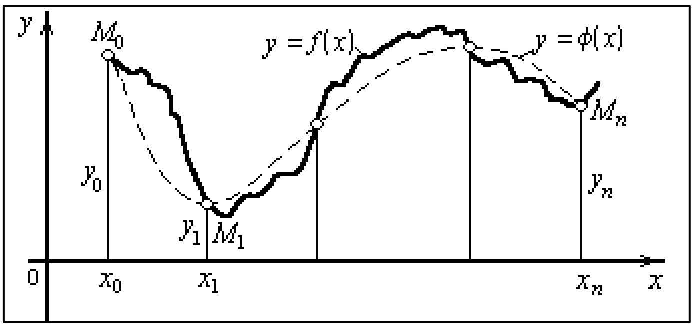


Рис. 14. Геометрическая интерпретация

$$\Phi(x_0) = y_0,\ \Phi(x_1) = y_1,\ \ldots,\ \Phi(x_n) = y_n. \qquad (2)$$

Геометрически это означает, что нужно найти кривую $y = \Phi(x)$ некоторого определенного типа, проходящую через заданную систему точек $M(x_i, y_i)$ ($i = 0, 1, ..., n$) (Рисунок 14).

В такой общей постановке задача может иметь бесконечное множество решений или совсем не иметь решений.

Однако эта задача становится однозначной, если вместо произвольной функции $\Phi(x)$ искать полином.[3] $\varphi(x)$ (*интерполяционный полином*) степени не выше n, удовлетворяющий условиям (2), т. е. такой, что

$$\varphi(x_0) = y_0,\ \varphi(x_1) = y_1,\ \ldots,\ \varphi(x_n) = y_n. \qquad (3)$$

Полученную интерполяционную формулу

$$\phi(x) = a_n x^n + a_{n-1}x^{n-1} + \ldots + a_1 x + a_0 \qquad (4)$$

обычно используют для приближенного вычисления значений данной функции $f(x)$ для значений аргумента x, отличных от узлов интерполяции. Такая операция называется *интерполяцией функций*.

Различают два вида интерполяции:

1) *глобальная* - соединение всех точек $f(x)$ единым интерполяционным полиномом;
2) *локальная* - соединение точек отрезками прямой (по двум точкам), отрезками параболы (по трем точкам).

Глобальная интерполяция

Параболическая интерполяция.[4]

Классический подход основывается на требовании строгого совпадения значений $f(x)$ и $\varphi(x)$ в точках x_i ($i = 0, 1, 2, ... n$) (3).

Будем искать интерполяционную функцию $\varphi(x)$ в виде полинома степени n (4).

Этот полином имеет $n + 1$ коэффициент. Естественно предположить, что $n + 1$ условий (3), наложенные на полином (4), позволяют однозначно определить его коэффициенты. Действительно, требуя для $\varphi(x)$ выполнение условий (3), получаем систему $n + 1$ уравнений с $n + 1$ неизвестными:

$$\sum_{k=0}^{n} a_k x_i^k = y_i\,, \quad (i = 0,\ 1,\ \ldots,\ n). \qquad (5)$$

Решая эту систему относительно неизвестных $a_0, a_1, ..., a_n$ мы получим аналитическое выражение полинома (4). Система (5) всегда имеет единственное решение, т. к. её определитель

$$D = \begin{vmatrix} x_0^n & x_0^{n-1} & \ldots & x_0 & 1 \\ x_1^n & x_1^{n-1} & \ldots & x_1 & 1 \\ . & . & . & . & . \\ x_n^n & x_n^{n-1} & \ldots & x_n & 1 \end{vmatrix},$$

известный в алгебре как *определитель Вандермонда,* отличен от нуля. Отсюда следует, что интерполяционный полином $\varphi(x)$ для функции $f(x)$, заданной таблично, существует и единственен.

Интерполяционная формула Лагранжа

Пусть на отрезке $[a, b]$ даны $n + 1$ различных значений аргумента: $x_0, x_1, ..., x_n$ и известны для функции $y = f(x)$ соответствующие значения выражений (1).

Требуется построить полином $L_n(x)$ степени не выше n, имеющий в заданных узлах $x_0, x_1, ..., x_n$ те же значения, что и функция $f(x)$, т. е. такой, что

$$L_n(x_i) = y_i \quad (i = 0, 1, ..., n)$$

Будем искать $L_n(x)$ в виде:

$$L_n(x) = l_0(x) + l_1(x) + \ldots + l_n(x), \qquad (6)$$

где $l_i(x)$ - полином степени n, причем

[3] *Полиномом* (*многочленом*) или *целой рациональной функцией* от x, называется функция вида (4), где $a_0, a_1, ..., a_n$ – постоянные числа, называемые *коэффициентами*; n – целое неотрицательное число, называемое *степенью полинома*.

[4] В некоторых случаях рассматриваются также задачи *тригонометрической интерполяции*. При этом в качестве класса функций, о котором шла речь выше, берутся тригонометрические полиномы.

$$l_i(x_k) = \begin{cases} y_i, & \textit{если} \quad i = k \\ 0, & \text{если} \quad i \neq k \end{cases} \tag{7}$$

Очевидно, что требование (7) с учетом (6) обеспечивает выполнение условий (3) (Рис. 2).

Так как искомый полином $l_i(x)$ обращается в нуль в n точках $x_0, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n$, то он имеет вид

$$l_i(x) = C_i (x - x_0)(x - x_1)\cdot\ldots\cdot(x - x_{i-1})(x - x_{i+1})\cdot\ldots\cdot(x - x_n) \tag{8}$$

где C_i - постоянный коэффициент. Полагая в формуле (8) $x = x_i$ и учитывая, что $l_i(x_i) = y_i$, получим:

$$y_i = C_i (x_i - x_0)(x_i - x_1)\cdot\ldots\cdot(x_i - x_{i-1})(x_i - x_{i+1})\cdot\ldots\cdot(x_i - x_n).$$

Отсюда

$$C_i = \frac{y_i}{(x_i - x_0)(x_i - x_1)\cdot\ldots\cdot(x_i - x_{i-1})(x_i - x_{i+1})\cdot\ldots\cdot(x_i - x_n)}.$$

Заметим, что ни один из множителей не равен нулю. Подставляя C_i в (8), а также с учетом (6), окончательно имеем:

$$L_n(x) = \sum_{i=0}^{n} y_i \frac{(x - x_0)(x - x_1)\ldots(x - x_{i-1})(x - x_{i+1})\ldots(x - x_n)}{(x_i - x_0)(x_i - x_1)\ldots(x_i - x_{i-1})(x_i - x_{i+1})\ldots(x_i - x_n)}. \tag{9}$$

Для построения *интерполяционной формулы Лагранжа* в MathCAD удобно использовать функцию *if.*

if (*cond* , *x* , *y*)

Возвращает значение *x*, если *cond* отличен от 0 (истина). Возвращает значение *y,* если *cond* равен 0 (ложь).

Локальная интерполяция средствами Mathcad

Как было описано выше, интерполяция использует значения некоторой функции, заданные в ряде точек, чтобы предсказать значения функции между ними. В MathCAD можно соединять точки данных прямыми линиями (линейная интерполяция) или соединять их отрезками кубического полинома (кубическая сплайн-интерполяция).

Функции интерполяции определяют кривую, точно проходящую через заданные точки. Из-за этого результат очень чувствителен к ошибкам данных. Кроме того, убедитесь, что каждый элемент массива, который используется в любой из функций, описанных в этом разделе, содержит определенное значение. Поскольку MathCAD присваивает значение 0 любым элементам, которые явно не определены.

Линейная интерполяция

При линейной интерполяции MathCAD соединяет существующие точки данных прямыми линиями. Это выполняется функцией *linterp*.

linterp$(\boldsymbol{vx}, \boldsymbol{vy}, \boldsymbol{x})$

Использует векторы данных vx и vy, чтобы возвратить интерполируемое значение y, соответствующее третьему аргументу x. Аргументы vx и vy должны быть векторами одинаковой длины. Вектор vx должен содержать вещественные значения, расположенные в порядке возрастания.

Эта функция соединяет точки данных отрезками прямых, создавая таким образом ломаную. Интерполируемое значение для конкретного *x* есть ордината *y* соответствующей точки ломаной.

Для значений x, расположенных перед первой точкой в векторе vx, MathCAD продолжает ломаную прямой линией, проходящей через первые две точки данных. Для значений x, расположенных за последней точкой vx, MathCAD продолжает ломаную прямой линией, проходящей через последние две точки данных.

Для получения наилучших результатов x должно находится между самыми большими и самыми маленькими значениями vx - маловероятно, что будут полезны значения, вычисленные для x вне этого диапазона. Функция *linterp* предназначена для интерполяции, а не для экстраполяции.

Кубическая сплайн-интерполяция

Кубическая сплайн-интерполяция позволяет провести кривую через набор точек таким образом, что первые и вторые производные кривой непрерывны в каждой точке. Эта кривая образуется путем создания ряда кубических полиномов, проходящих через наборы из трех смежных точек. Кубические полиномы затем состыковываются друг с другом, чтобы образовать одну кривую.

MathCAD поставляется с тремя сплайн-функциями:

cspline$(\boldsymbol{vx}, \boldsymbol{vy})$
pspline$(\boldsymbol{vx}, \boldsymbol{vy})$
lspline$(\boldsymbol{vx}, \boldsymbol{vy})$

Они возвращают вектор коэффициентов вторых производных, который мы будем называть vs. Этот вектор vs обычно используется в функции *interp*, описанной ниже. Аргументы vx и vy должны быть вещественными векторами одинаковой длины. Значения вектора vx должны быть расположены в порядке возрастания.

Эти три функции отличаются только граничными условиями:

- функция *lspline* генерирует кривую сплайна, которая приближается к прямой линии в граничных точках;
- функция *pspline* генерирует кривую сплайна, которая приближается к параболе в граничных точках.
- функция *cspline* генерирует кривую сплайна, которая может быть кубическим полиномом в граничных точках.

interp $(\boldsymbol{vs}, \boldsymbol{vx}, \boldsymbol{vy}, \boldsymbol{x})$

Возвращает интерполируемое значение y, соответствующее аргументу x. Вектор vs вычисляется на основе векторов данных vx и vy одной из функций *pspline, lspline* или *cspline.*

Чтобы провести кубический сплайн через набор точек (Рисунок 15):

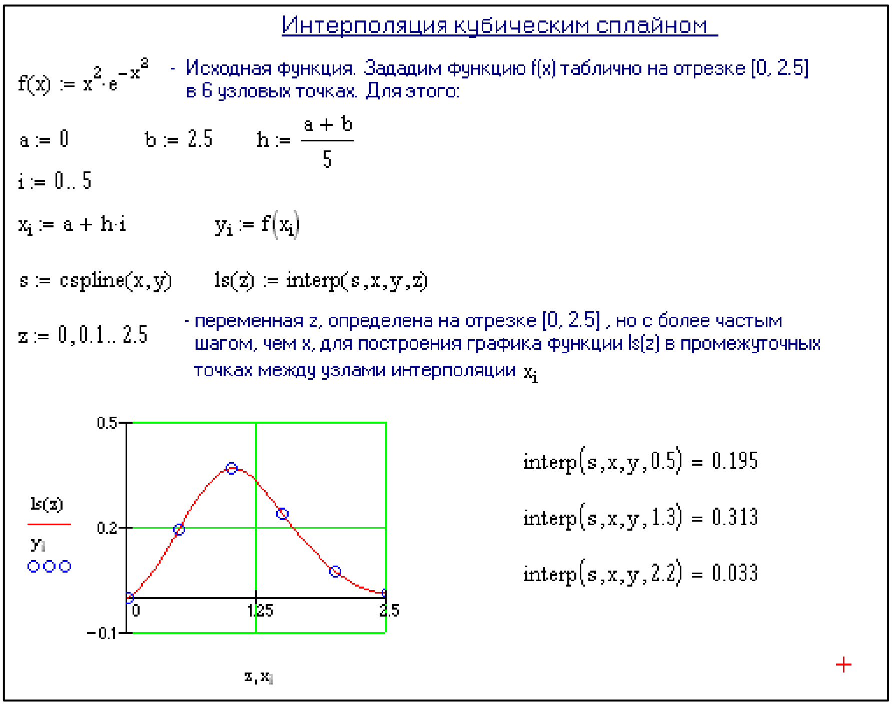


Рис. 15. Кубическая сплайн-интерполяция

- создайте векторы vx и vy, содержащие координаты x и y,через которые нужно провести кубический сплайн. Элементы vx должны быть расположены в порядке возрастания. (Хотя мы используем имена vx, vy и vs, нет никаких ограничений на имена этих переменных; можно использовать любые имена);
- Вычислите вектор $vs := cspline(vx, vy)$.

Вектор vs содержит вторые производные интерполяционной кривой в рассматриваемых точках;

* чтобы найти интерполируемое значение в произвольной точке x_0, вычислите $interp(vs, vx, vy, x_0)$, где vs, vx, vy – векторы, описанные ранее.

Можно сделать то же самое, вычисляя: $interp(cspline(vx, vy), vx, vy, x0)$.

Интерполируемое значение для конкретного *x* есть ордината *y* соответствующей точки сплайна.

Для значений x, расположенных перед первой точкой вектора vx, MathCAD продолжает сплайн первой из составляющих его кубических парабол. Для значений x, расположенных за последней точкой vx, MathCAD продолжает сплайн первой из составляющих его кубических парабол.

Для получения наилучших результатов x должно находится между самыми большими и самыми маленькими значениями vx - маловероятно, что будут полезны значения, вычисленные для x вне этого диапазона. Сплайны предназначены для интерполяции, а не для экстраполяции.

Интерполяция вектора точек

Можно использовать *оператор векторизации*, чтобы вычислить сразу целый набор интерполируемых значений, соответствующих вектору заданных точек. Это можно сделать и с *interp,* с и *linter.*

Пример 1 показывает, как выполнить эту операцию. Чтобы применить оператор векторизации к функции, щелкните мышью на имени функции и нажимайте клавишу **пробел**, пока в рамку не попадет нужная функция. Затем нажмите **Ctrl + -** (минус).

Пример 1. Кубическая сплайн-интерполяция

$f(x) := x^2 \cdot e^{-x^2}$ $\quad a := 0$ $\quad b := 2.5$ $\quad h := \frac{a+b}{5}$

$i := 0..5$ $\quad x_i := a + h \cdot i$ $\quad y_i := f(x_i)$

$s := cspline(x, y)$ $\quad z := 0, 0.1..2.5$

$\overrightarrow{interp(s, x, y, z)} =$

0
0.02
0.052
0.095
0.143
0.195
0.245

Линейное предсказание (экстраполяция)

Иногда необходимо оценить значения формул в точках, находящихся вне области расположения сетки, на которой заданы значения функции. В Mathcad есть функция *predict*, которая позволяет это сделать.

Эта функция использует линейный алгоритм предсказания, который является полезным, когда экстраполируемая функция является гладкой и осциллирующей, хотя не обязательно периодической. Линейное предсказание можно рассматривать как разновидность экстраполяции, но нельзя путать с линейной или полиномиальной экстраполяцией.

predict$(\boldsymbol{v}, \boldsymbol{m}, \boldsymbol{n})$

Возвращает n предсказанных значений, основанных на *m* последовательных значениях вектора данных v. Элементы вектора v должны представлять собой значения, взятые через равные интервалы.

Функция *predict* использует последние *m* исходных значений данных, чтобы вычислять коэффициенты предсказания. Как только это сделано она использует последние m точек, чтобы предсказать координаты $(m + 1)$ - ой точки, фактически создавая скользящее окно шириной в m точек (Рисунок 16).

Двумерная сплайн-интерполяция

MathCAD выполняет интерполяцию кубическими сплайнами функции двух переменных тем же самым образом, как и в одномерном случае, обсужденном ранее. MathCAD проводит через сетку узлов поверхность, составленную из кубических полиномов от *x* до *y*, таким образом, что первые и вторые производные являются непрерывными в каждом узле сетки.

cspline$(\boldsymbol{Mxy},\ \boldsymbol{Mz})$
pspline$(\boldsymbol{Mxy},\ \boldsymbol{Mz})$
lspline$(\boldsymbol{Mxy},\ \boldsymbol{Mz})$

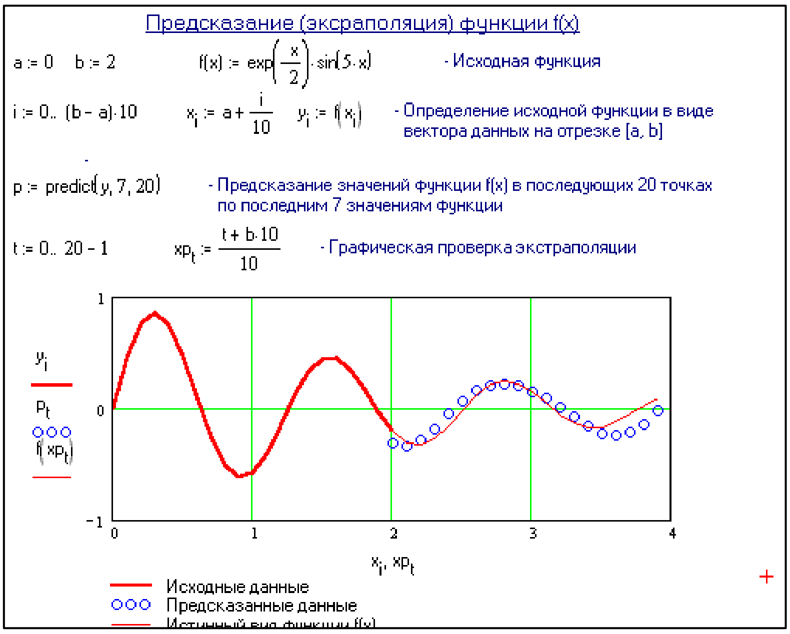


Рис. 16. Экстраполяция

interp $(\boldsymbol{vs}, \boldsymbol{Mxy}, \boldsymbol{Mz}, \boldsymbol{x})$

Возвращает интерполируемое значение *z*, соответствующее точкам $x = v_0$ и $y = v_1$. Вектор vs вычисляется *pspline, lspline* или *cspline* на основе векторов данных *Mxy* и *Mz.*

Первый шаг в двумерной сплайн-интерполяции – точно такой же, как и в одномерном случае: определить узлы, через которые поверхность должна пройти. Однако это потребует больше усилий, потому что теперь нужно определить сетку узлов:

- создайте $n \times 2$ матрицу *Mxy*, чьи элементы, $Mxy_{i,\,0}$ и $Mxy_{i,\,1}$ определяют *x* и *y* координаты *по диагонали* прямоугольной сетки. Эта матрица играет ту же самую роль, что и vx в одномерном случае, описанном ранее. Так как эти узлы описывают диагональ, элементы в каждом столбце *Mxy* должны быть расположены в порядке возрастания ($Mxy_{i,\,k} < Mxy_{j,\,k}$ всякий раз, когда $i < j$);

- создайте $n \times n$ матрицу *Mz*, чей (*ij*)- элемент есть координата z соответствующая точке $x = Mxy_{i,\,0}$ и $y = Mxy_{j,\,1}$. Она играет ту же самую роль, что и *vy* в одномерном случае, описанном ранее;

- вычислите вектор $vs := cspline(Mxy, Mz)$

  Вектор *vs* содержит вторые производные приближающей поверхности в узлах, определенных *Mxy* и *Mz*;

- чтобы найти интерполируемое значение в произвольной точке, скажем (x_0, y_0), вычислите

$$interp\left(vs, Mxy, Mz, \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}\right),$$

  где vs, Mxy, Mz - массивы, описанные ранее. Результатом является значение интерполирующей функции, соответствующее точке (x_0, y_0).

Можно сделать то же самое, вычисляя: $interp\left(cspline\ (Mxy, Mz), Mxy, Mz, \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}\right)$.

Обычно двумерную сплайн-интерполяцию используют для повышения качества построения 3D-графиков. Она позволяет существенно повысить представительность сложных графиков функций, в том числе и контурных.

Пример двумерной сплайн-интерполяции можно посмотреть в Центре Ресурсов MathCAD **? / Ресурс Центр / Обзор и примеры / Анализ ваших данных / Interpolation (Multivariate Interpolation with Spline Functions Using interp)**.

ПРАКТИЧЕСКОЕ ЗАДАНИЕ

Упражнение 1. Вычисление значения функции, интерполяция

1. Вычислить значения заданной функции $y_i = f(x_i)$ в узлах интерполяции $x_i = a + h\,i$, где $h = (b - a)/10$, $i = 0, 1, ..., 10$, на отрезке $[a, b]$ (Таблица).

2. По вычисленной таблице (x_i, y_i) провести *параболическую интерполяцию.*

№ варианта	$f(x)$	$[a, b]$	№ варианта	$f(x)$	$[a, b]$
1	$\sin x^2$	[0, 2]	9	$x \cdot cos(\,x + ln(\,1 + x))$	[1, 5]
2	$\cos x^2$	[0, 2]	10	$10 \cdot ln\,2\,x/(1 + x)$	[1, 5]
3	$e^{sin\,x}$	[0, 5]	11	$sin\,x^2 \cdot e^{-(x/2)^2}$	[0, 3]
4	$1/(0.5 + x^2)$	[0, 2]	12	$cos(x + cos^3 x)$	[0, 2]
5	$e^{-(x + \sin x)}$	[2, 5]	13	$\cos(x + e^{\cos x})$	[3, 6]
6	$1/(1 + e^{-x})$	[0, 4]	14	$cos(2x + x^2)$	[0, 1]
7	$\sin(x + e^{\sin x})$	[0, 3]	15	$e^{\cos x} \cdot \cos x^2$	[0, 2]
8	$e^{-(x + 1/x)}$	[1, 3]			

- Для нахождения коэффициентов искомого полинома (4) необходимо составить систему линейных алгебраических уравнений (5).
- Систему уравнений решить матрично с использованием функции *lsolve*.
- Построить график интерполяционного полинома и отметить на нем узловые точки (x_i, y_i).

3. Провести интерполирование заданной функции с помощью первой и второй интерполяционных формул *Ньютона*.

Построить графики интерполяционных полиномов и отметить на нем узловые точки (x_i, y_i).

4. Провести *линейную интерполяцию* заданной функции с помощью встроенной интерполяционной функции *linterp*.

5. Построить график функции *linterp* и отметить на нем узловые точки (x_i, y_i).

Провести сплайн-интерполяцию с помощью функций lspline, pspline, cspline и interp. Построить график функции *interp* и отметить на нем узловые точки (x_i, y_i).

Упражнение 2. Экстраполяция

Вычислить значения заданной функции $y_i = f(x_i)$ в точках $x_i = a + i/10$, где, $i = 0, 1, ..., 10(b - a)$, на отрезке $[a, b]$.

- С использованием функции *predict* выполнить *предсказание* (*экстраполяцию*) полученного вектора данных y_i в последующих **10** точках по последним **7** значениям функции.
- Отобразить графически имеющиеся данные, предсказанные данные и истинный вид функции $f(x)$.

Контрольные вопросы

1. Что такое аппроксимация функций?
2. Для чего нужна интерполяция функций?
3. Охарактеризуйте виды интерполяции.
4. Чем определяется близость интерполяционного полинома к заданной функции?
5. Чем определяется степень интерполяционного полинома?
6. Какие виды глобальной интерполяции вам известны?
7. Что такое экстраполяция?
8. Какие формулы применяются для экстраполяции?
9. Чем обуславливается выбор способов интерполяции?
10. Какие методы локальной интерполяции вам известны?

Практическая работа №6 (часть1, 2)

Математическая обработка результатов эксперимента

Метод наименьших квадратов. Аппроксимирующие функции. Обработка информации, полученной в результате эксперимента.

Цель: изучить методы обработки информации, полученной в результате эксперимента.

Интерполяционные формулы в точности воспроизводят значения данной функции в узлах интерполяции. Однако в ряде случаев выполнение этого условия затруднительно или даже нецелесообразно:

1. Если заданные величины x и y являются экспериментальными данными, то могут содержать в себе существенные ошибки, т. к. получены в результате измерений или наблюдений. Поэтому построение аппроксимирующего многочлена, воспроизводящего в точности заданное значение функции, означало бы тщательное копирование допущенных при измерениях ошибок.
2. Если имеются точные значения функции в некоторых точках, но число таких точек n весьма велико, то интерполяционный многочлен будет очень высокой степени[5].
3. Между узлами интерполяции значения интерполяционного многочлена могут сильно отклоняться от значений интерполируемой функции.

Пример. Известный из физики закон охлаждения состоит в том, что скорость охлаждения тела пропорциональна избытку температуры тела над температурой среды, т. е. выражается формулой

$$\upsilon = a\,\vartheta,$$

где υ – скорость охлаждения, ϑ - избыток температуры и a – коэффициент пропорциональности (закон Ньютона).

В таблице приведены опытные данные, выражающие зависимость между υ и ϑ для некоторого определенного тела, по которым можно определить значения коэффициента пропорциональности a.

$\vartheta°$	**220**	**200**	**180**	**160**	**140**	**120**	**100**
υ	8,81	7,40	6,10	4,89	3,88	3,02	2,30

По существу, при отсутствии ошибок измерения для определения значения a достаточно иметь лишь одно наблюдение. Однако на самом деле это не так, и, как показывает Таблица 2, каждое из наблюдений дает новое значение a. Например, первое из наблюдений дает $\vartheta = 220$, $\upsilon = 8{,}81$, откуда, следует $a = 0{,}040$. Аналогично следующие наблюдения дают

$$a = 0{,}037;\ 0{,}034;\ 0{,}031;\ 0{,}028;\ 0{,}025;\ 0{,}023.$$

Смотря по тому, каким из наблюдений мы пользуемся, зависимость между скоростью охлаждения и избытком температуры будет иметь тот или иной коэффициент пропорциональности и будет изображаться соответствующей прямой (рис. 17).

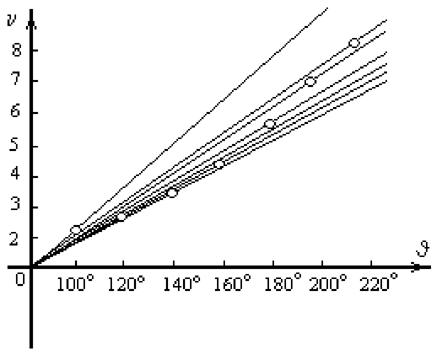


Рис. 17. Опытные данные

Конечно, есть возможность по имеющимся семи наблюдениям, построить интерполяционный многочлен шестой степени, который будет в точности совпадать с наблюденными значениями. Но это

[5] Если только разности не будут становиться постоянными.

совершенно не нужно, так как все эти значения содержат ошибки. Требуется по имеющимся наблюденным данным построить линейную функцию, которая для различных (в том числе и имеющихся в таблице) значений ϑ давала бы наиболее близкие значения υ.

Таким образом, возникает задача построения многочлена некоторой вполне определенной степени, но меньшей чем n - 1, который хотя и не дает точных значений функции в узлах интерполяции, но д о с т а т о ч н о б л и з к о к ним подходит.

Метод наименьших квадратов

В Примере 2 рассматривалась зависимость вида $\upsilon = a\,\vartheta$, причем значения υ и ϑ получаются из наблюдений. Если бы измерения величины υ и ϑ производились без ошибок, то для определения коэффициента a было бы достаточно одного измерения. Если рассматривать более общую зависимость, например, $\upsilon = a\vartheta + b$, то здесь имеется два неизвестных коэффициента, для нахождения которых достаточно было бы двух абсолютно точных измерений.

На самом деле, абсолютно точные измерения чаще всего невозможны. Для того, чтобы исключить влияние ошибок, производится большое число измерений. Каждое измерение дает нам уравнение, связывающее неизвестные коэффициенты. При большом числе измерений мы приходим, следовательно, к системе, число уравнений в которой значительно больше, нежели число неизвестных. Здесь ставится задача отыскания н а и б о л е е   в е р о я т н ы х значений коэффициентов, которые, вообще говоря, не будут точно удовлетворять ни одному из уравнений системы. Эту задачу можно сформулировать в более общем виде. Пусть дана функция

$$y = f(x; a_0, a_1, ..., a_m) \qquad (10)$$

независимой переменной x и m + 1 параметра $a_0, a_1, ..., a_m$. Эти параметры постоянны, но заранее неизвестны и подлежат определению. Для их отыскания производится ряд измерений величин x и y[6]. Подставляя их в равенство (10), мы получаем уравнения между параметрами $a_0, a_1, ..., a_m$, вида

$$y_i = f(x_i, a_0, a_1, ..., a_m) \; (i = 1, 2, \ldots, n), \qquad (11)$$

где x_i и y_i — соответствующие друг другу измеренные значения, а n —число измерений. Если бы значения x и y находились точно, то для отыскания m + 1 параметра достаточно было бы произвести m + 1 измерение.

На самом деле, значения x и y содержат ошибки, и никакие m + 1 измерений не позволят определить истинные значения параметров. Поэтому обычно производится большее число измерений ($n > m + 1$), в результате чего число уравнений (11) будет больше, чем число неизвестных параметров. В этом случае система (11) будет несовместной, т. е. точные решения каких-либо (m + 1) из уравнений системы могут не удовлетворять остальным уравнениям[7]. Требуется найти *наиболее вероятные значения неизвестных параметров.* Эти вероятные значения будут тем более близки к истинным, чем больше число наблюдений.

Так как уравнения (11) удовлетворяются неточно, то будем иметь

$$y_i - f(x_i, a_0, a_1, ..., a_m) = \varepsilon_i \; (i = 1, 2, \ldots, n), \qquad (12)$$

где ε_i — отклонения и з м е р е н н ы х   з н а ч е н и й y_i от в ы ч и с л е н н ы х по формуле (10). Принцип наименьших квадратов утверждает, что *наивероятнейшими значениями параметров будут такие, при которых сумма квадратов отклонений* ε_i, *будет наименьшей*[8], т. е.

$$S = \sum_{i=0}^{n} \varepsilon_i^2 = \sum_{i=0}^{n} [y_i - f(x_i, a_0, a_1, ..., a_m)]^2 = \min. \qquad (13)$$

Функцию S называют *функцией невязки*.

Аппроксимация в виде линейной комбинации функций

Часто функцию $f(x)$ выбирают в виде линейной комбинации подходящих функций

$$f(x) = a_0 f_0(x) + a_1 f_1(x) + \ldots + a_m f_m(x). \qquad (14)$$

Условие минимума S определяется уравнениями

---

[6] Разумеется, так поступают тогда, когда константы $a_0, a_1, ..., a_m$ не поддаются непосредственным измерениям, а величины x и y измерению доступны.

[7] В случае $n < m + 1$ система (2) была бы совместной и всегда имела бы бесчисленное множество решений.

[8] При этом мы считаем, что отклонения ε_i подчиняются нормальному закону распределения.

$$\frac{\partial S}{\partial a_0} = 0,\ \frac{\partial S}{\partial a_1} = 0,\ \ldots,\ \frac{\partial S}{\partial a_m} = 0. \qquad (15)$$

Поскольку

$$S = \sum_{i=0}^{n} [y_i - (a_0 f_0(x) + a_1 f_1(x) + \ldots + a_m f_m(x))]^2, \qquad (16)$$

это условие эквивалентно системе уравнений:

$$\begin{cases} \dfrac{\partial S}{\partial a_0} = 2\sum\limits_{i=0}^{n} [y_i - (a_0 f_0(x) + a_1 f_1(x) + \ldots + a_m f_m(x))]\, f_0(x) = 0, \\ \dfrac{\partial S}{\partial a_1} = 2\sum\limits_{i=0}^{n} [y_i - (a_0 f_0(x) + a_1 f_1(x) + \ldots + a_m f_m(x))]\, f_1(x) = 0, \\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot \\ \dfrac{\partial S}{\partial a_m} = 2\sum\limits_{i=0}^{n} [y_i - (a_0 f_0(x) + a_1 f_1(x) + \ldots + a_m f_m(x))]\, f_m(x) = 0. \end{cases}$$

Эти $m + 1$ уравнений, очевидно, можно представить в виде

$$\begin{bmatrix} \sum_{i=0}^{n} f_0^2(x_i) & \sum_{i=0}^{n} f_0(x_i) f_1(x_i) & \ldots & \sum_{i=0}^{n} f_0(x_i) f_m(x_i) \\ \ldots & \ldots & \ldots & \ldots \\ \sum_{i=0}^{n} f_0(x_i) f_m(x_i) & \ldots & \ldots & \sum_{i=0}^{n} f_m^2(x_i) \end{bmatrix} \begin{bmatrix} a_0 \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^{n} f_0(x_i) y_i \\ \vdots \\ \sum_{i=0}^{n} f_m(x_i) y_i \end{bmatrix}$$

(17)

Так как элементы матрицы в левой части и вектора-столбца в правой определяются табличными данными, то выписанная система $m + 1$ линейных уравнений с $m + 1$ неизвестными может быть решена. Можно выбрать любую функцию $f(x)$, лишь бы она была линейной относительно своих коэффициентов.

Если предполагается, что данные могли бы быть смоделированы в виде линейной комбинации произвольных функций (14), следует использовать функцию *linfit*, чтобы вычислить коэффициенты a_i. Это так называемая аппроксимация линейной *комбинацией функций* (рис. 18).

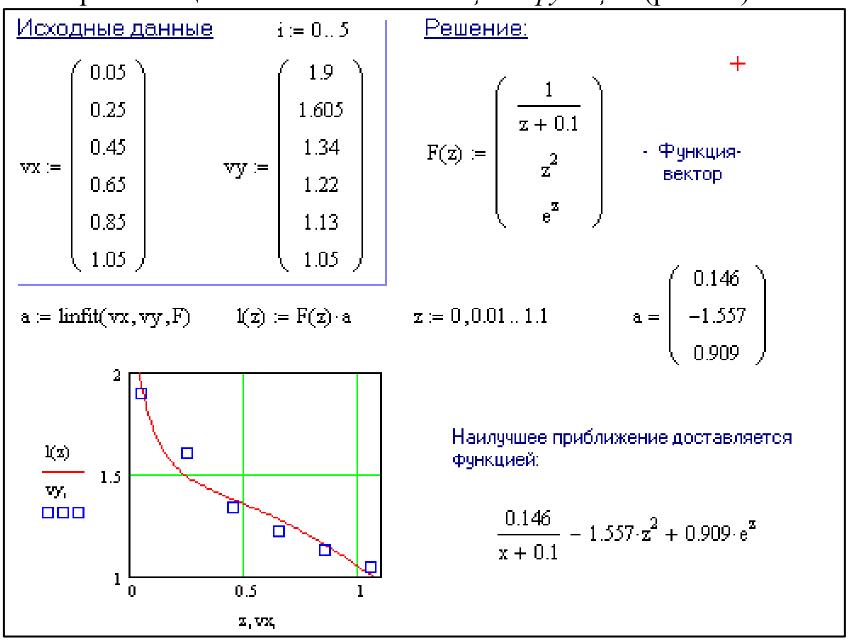


Рис. 18. Использование функции *linfit* для нахождения коэффициентов в линейной комбинации функций

linfit(*vx*, *vy*, *F*)

Возвращает вектор, содержащий коэффициенты, чтобы создать линейную комбинацию функций из F, дающую наилучшую аппроксимацию данных из векторов *vx* и *vy*. F – функция, которая возвращает вектор, состоящий из функций, которые нужно объединить в виде линейной комбинации.

Примеры применения способа наименьших квадратов

Занимаясь интерполированием функций, мы рассматривали, как говорят, *точечную интерполяцию,* т. е. строили интерполяционные многочлены, значения которых точно совпадали со значениями самой функции в узлах интерполяции. Однако было указано, что возможна и другая постановка задачи, когда от интерполяционного многочлена требуется лишь, чтобы он как-то приближал заданную функцию и не обязательно совпадал с ней в отдельных заданных точках. Говоря о заданной функции, мы будем подразумевать при этом, что нам известны лишь отдельные ее значения в некоторых точках[9].

Мы предполагаем, следовательно, что зависимость y от x имеет вид многочлена

$$y = a_0 + a_1 x + \dots + a_m x^m. \qquad (18)$$

с неизвестными коэффициентами.

Нашей задачей является нахождение по результатам наблюдений наиболее вероятных значений коэффициентов $a_0, a_1, \dots, a_m$.

Если бы число наблюдений в точности равнялось числу неизвестных коэффициентов, то мы имели бы дело с задачей интерполяции. Значительно более важным является тот случай, когда число наблюдений n много больше степени многочлена. В этом случае получается обычная задача способа наименьших квадратов[10]. Требуется найти коэффициенты $a_0, a_1, \dots, a_m,$ дающие минимум функции невязки

$$S = \sum_{i=0}^{n} [y_i - (a_0 + a_1 x + \dots + a_m x^m)]^2. \qquad (19)$$

Условие минимума S определяется уравнениями

$$\frac{\partial S}{\partial a_0} = 0, \frac{\partial S}{\partial a_1} = 0, \dots, \frac{\partial S}{\partial a_m} = 0. \qquad 20)$$

Условие (19) с учетом (20) эквивалентно системе уравнений

$$\begin{cases} \dfrac{\partial S}{\partial a_0} = 2\sum\limits_{i=0}^{n} [y_i - (a_0 + a_1 x_i + \dots + a_m x_i^m)], \\ \dfrac{\partial S}{\partial a_1} = 2\sum\limits_{i=0}^{n} [y_i - (a_0 + a_1 x_i + \dots + a_m x_i^m)]\, x_i, \\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot \\ \dfrac{\partial S}{\partial a_m} = 2\sum\limits_{i=0}^{n} [y_i - (a_0 + a_1 x_i + \dots + a_m x_i^m)]\, x_i^m. \end{cases}$$

Эти $m + 1$ уравнений можно представить в виде:

$$\begin{cases} (n+1)a_0 + a_1\sum_{i=0}^{n} x_i + a_2\sum_{i=0}^{n} x_i^2 + \dots + a_m\sum_{i=0}^{n} x_i^m = \sum_{i=0}^{n} y_i, \\ a_0\sum_{i=0}^{n} x_i + a_1\sum_{i=0}^{n} x_i^2 + a_2\sum_{i=0}^{n} x_i^3 + \dots + a_m\sum_{i=0}^{n} x_i^{m+1} = \sum_{i=0}^{n} x_i y_i, \\ \cdot \\ a_0\sum_{i=0}^{n} x_i^m + a_1\sum_{i=0}^{n} x_i^{m+1} + a_2\sum_{i=0}^{n} x_i^{m+2} + \dots + a_m\sum_{i=0}^{n} x_i^{2m} = \sum_{i=0}^{n} x_i^m y_i. \end{cases} \qquad (21)$$

Решая систему уравнений (21), находим коэффициенты $a_0, a_1, \dots, a_m,$ которые являются искомыми параметрами многочлена (18).

Системy (21) можно записать компактно:

[9] Можно говорить и о приближении функций, заданных аналитически.

[10] Нет смысла применять метод наименьших квадратов при $n \le m + 1$. В этом случае система уравнений для определения коэффициентов a_i совместна и сумма квадратов будет равна нулю при любых решениях системы.

$$\begin{cases} b_{00}a_0 + b_{01}a_1 + \ldots + b_{0m}a_m = c_0, \\ b_{10}a_0 + b_{11}a_1 + \ldots + b_{1m}a_m = c_1, \\ \quad \cdot \quad \cdot \quad \cdot \quad \cdot \\ b_{m0}a_0 + b_{m1}a_1 + \ldots + b_{mm}a_m = c_m. \end{cases} \qquad (22)$$

где $b_{k,\,l} = \sum\limits_{i=0}^{n} x_i^{k+l}, \quad c_k = \sum\limits_{i=0}^{n} x_i^k y_i, \quad k, l = 0, 1, \ldots, m\,.$ (23)

Используйте *regress*, когда нужно использовать единственный полином, чтобы приблизить все данные. Функция *regress* допускает использование полинома любого порядка. Однако на практике не следует использовать степень полинома выше $n = 4$ (рис. 19).

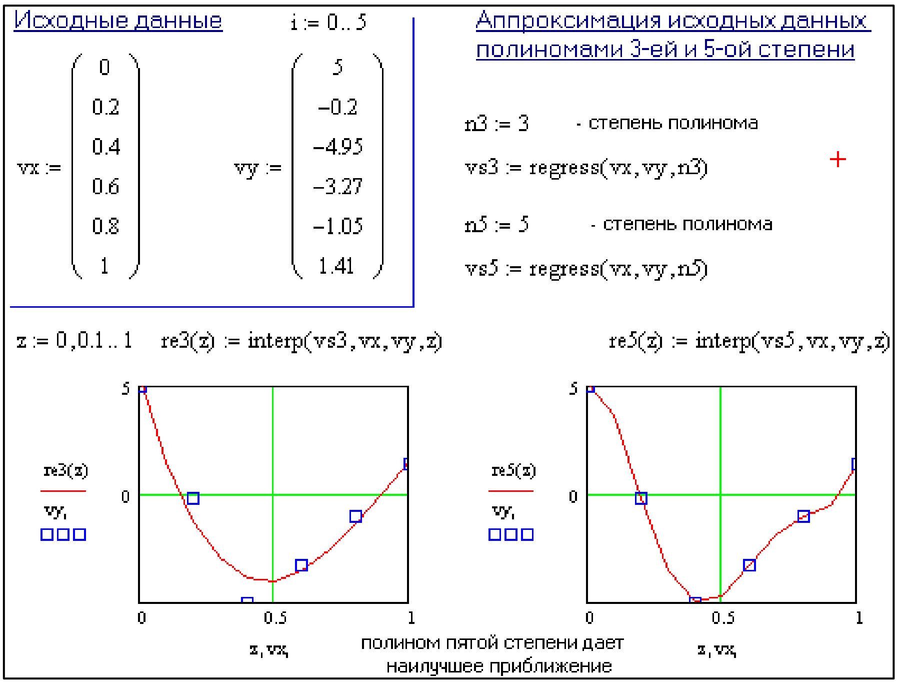


Рис. 19. Аппроксимация полиномами различной степени с помощью функции *regress*

Так как *regress* пытается приблизить все точки данных, используя один полином, это не даст хороший результат, когда данные не связаны единой полиномиальной зависимостью. Например, предположим, ожидается, что y_i зависят линейно от x в диапазоне от x_1 до x_{10} и ведут себя подобно кубическому полиному в диапазоне от x_{11} до x_{20}. Если используется *regress* с $n = 3$, можно получить хорошее приближение для второй половины, но плохое — для первой. Функция *loess* облегчает эти проблемы, выполняя локальное приближение. Вместо создания одного полинома, как это делает *regress*, *loess* создаёт различные полиномы второго порядка в зависимости от расположения на кривой.

Она делает это, исследуя данные в малой окрестности точки, представляющей интерес. Аргумент *span* управляет размером этой окрестности. По мере того, как диапазон становится большим, *loess* становится эквивалентным *regress* с $n = 2$. Значение по умолчанию – *span* = 0,75.

Рисунок 20 показывает, как *span* влияет на приближение, выполненное функцией *loess*. Заметьте, что меньшее значение *span* лучше приближает флуктуации данных. Большее значение *span* сглаживает колебания данных и создает более гладкую приближающую функцию.

regress(*vx*, *vy*, *n*)

Возвращает вектор, требуемый *interp*, чтобы найти полином порядка *n*, который наилучшим образом приближает данные из *vx* и *vy*. *vx* есть *m*-мерный вектор, содержащий координаты *x*. *vy* есть *m*-мерный вектор, содержащий координаты *y,* соответствующие m точкам, определенным в *vx*.

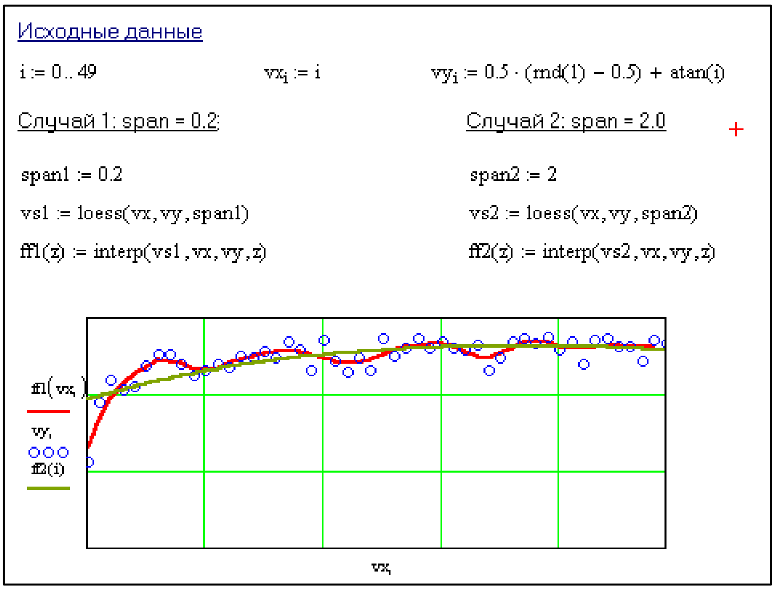


Рис. 20. Влияние различных значений *span* на функцию *loes*

loess(*vx, vy, span*)

Возвращает вектор, требуемый *interp,* чтобы найти набор полиномов второго порядка, которые наилучшим образом приближают определённые окрестности выборочных точек, определенных в векторах *vx* и *vy*. *vx* есть *m*-мерный вектор, содержащий координаты *x*. *vy* есть *m*-мерный вектор, содержащий координаты *y,* соответствующие *m* точкам, определенным в *vx***.** Аргумент *span (span* > 0) определяет, насколько большие окрестности будет использовать *loess* при выполнении локального приближения.

interp(*vs, vx, vy, x*)

Возвращает интерполируемое значение *y,* соответствующее *x.* Вектор vs вычисляется *loess* или *regress* на основе данных из *vx* и *vy*.

Аппроксимирующие функции в Mathcad

В пакете Mathcad представлен широкий набор функций для представления наблюденных данных уравнениями, некоторые из которых уже были рассмотрены в предыдущих разделах. В таблице приведен список функций.

Функция	**Вид аппроксимируемой зависимости**	**Уравнение**
linfit(*vx,vy, F***)**	линейная комбинация функций	$a_1 f_1(x) + \ldots + a_2 f_2(x) + a_m f_m(x)$
regress(*vx, vy, n***)**	Полином степени *n*	$a_0 + a_1 x + \ldots + a_n x^n$
loess(*vx, vy, span***)**	Набор полиномов второй степени	
slope(*vx, vy***)** **intercept(***vx, vy***)**	линейная	$a + b\,x$
line(*vx, vy***)**	линейная	$a + b\,x$
medfit(*vx, vy***)**	линейная	$a + b\,x$
pwrfit(*vx, vy, vg***)**	степенная	$a\,x^{b} + c$
expfit(*vx, vy, vg***)**	показательная	$a\,\mathrm{e}^{bx} + c$
logfit(*vx, vy, vg***)**	логарифмическая	$a \ln(x + b) + c$
sinfit(*vx, vy, vg***)**	синусоидальная	$a \sin(x - b) + c$
lgfit(*vx, vy, vg***)**	логистическая	$\dfrac{a}{1 + b\,e^{-cx}}$
genfit(*vx, vy, vg, F***)**	нелинейная	

Особый интерес представляет собой функция *genfit*, использование которой возможно для аппроксимации в общем случае любой нелинейной зависимости. Все, что можно делать с помощью любой из функций из таблицы, можно также делать, хотя в ряде случаев и менее удобно, с помощью *genfit*.

Однако имеются случаи, когда гибкость функций, представляющих ту или иную математическую зависимость, недостаточна. Например, если данные могут быть смоделированы в виде суммы

$$f(x) = a_1 \sin(2x) + a_2 \operatorname{tg}(3x)$$

и все, что нужно сделать – решить уравнение относительно неизвестных коэффициентов a_1 и a_2, значит эта проблема решается с помощью *linfit*.

В противоположность этому, если данные должны быть смоделированы в виде суммы

$$f(x) = 2 \sin(a_1 x) + 3 \operatorname{tg}(a_2 x)$$

и требуется найти неизвестные параметры a_1 и a_2, то это задача для функции *genfit*.

Таким образом, если данные должны быть смоделированы в виде

$$f(x) = f(x, u_0, u_1, \dots, u_n),$$

нужно использовать функцию *genfit*, чтобы найти неизвестные параметры u_i.

genfit(*vx, vy, vg, F*)

Возвращает вектор, содержащий n параметров $u_0, u_1, \dots, u_{n-1}$, которые обеспечивают наилучшее приближение данных из *vx* и *vy* функцией f, зависящей от x и параметров $u_0, u_1, \dots, u_{n-1}$.

Аргументы:

vx – вектор значений вещественных данных x;

vy – вектор значений вещественных данных y;

F – функция, которая возвращает n + 1-мерный вектор, содержащий f и ее частные производные относительно параметров. Для нахождения частных производных необходимо использовать команду символьных преобразований **Символы / Переменные / Дифференциалы**;

vg – n-мерный вектор начальных значений для n параметров.

На рис. 21 приведен пример использования функции *genfit* для определения коэффициентов степенной функции. Причем результаты расчета, полученные с использованием функции *genfit,* полностью совпадают с результатами использования функции *pwrfit*.

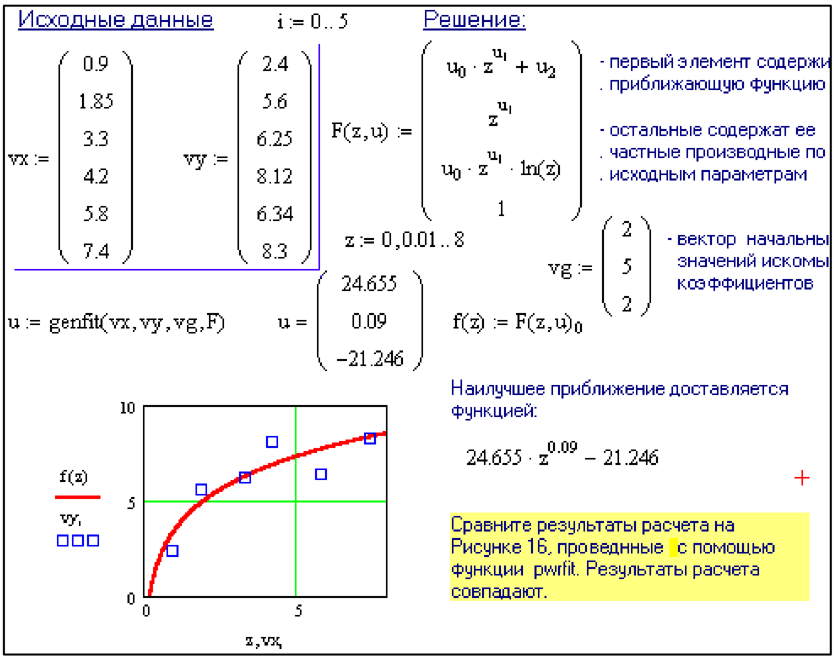


Рис. 21. Использование функции *genfit*

На Рисунке 21 приведен пример использования функции *genfit* для определения коэффициентов степенной функции. Причем результаты расчета, полученные с использованием функции *genfit,* полностью совпадают с результатами использования функции *pwrfit*.

Функции Mathcad для линейной аппроксимации

Встроенные функции *intercept* (to intercept - отложить отрезок на линии) и *slope* (наклон) решают самую простую и распространенную задачу *линейной аппроксимации* экспериментальных данных:

$$f(x) = slope(vx,\ vy)\ x + intercept(vx,\ vy)$$

slope (*vx, vy*)
Возвращает скаляр: наклон линии для данных из векторов *vx* и *vy*.

intercept(vx, vy**)**
Возвращает скаляр: смещение по оси ординат линии для данных из векторов *vx* и *vy*.

Рисунок 22 показывает, как можно использовать эти функции, чтобы провести линию через набор данных.

В последних версиях Mathcad появились две новые функция для определения коэффициентов прямой линии, аппроксимирующей исходный набор точек.

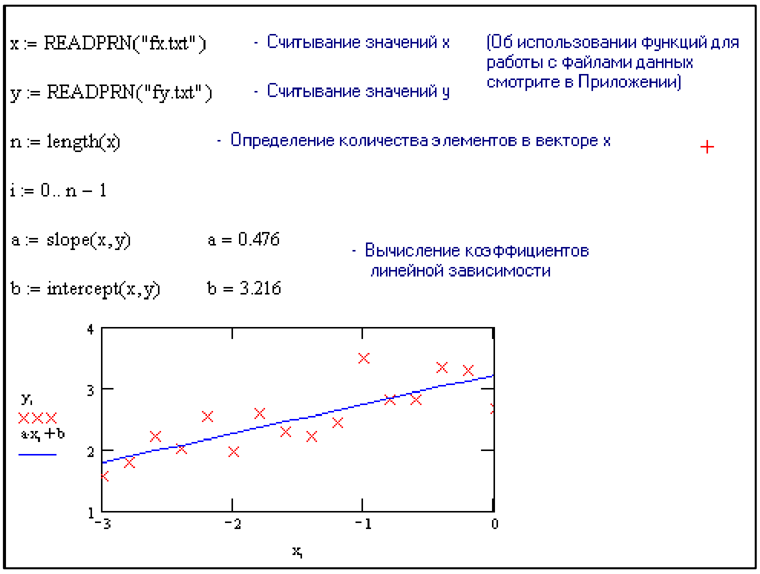


Рис. 22. Использование функций slope и intercept

line(*vx, vy*), **medfit(*vx, vy*)**
Определяют вектор, содержащий коэффициенты линейного уравнения $a + b\,x$.

Рисунок 10 показывает применение функций *line* и *medfit*. А также приведено сравнение результатов вычислений по точности. Как показано на Рисунке 10, функция *line* дает более точное приближение, нежели функция *medfit.*

Нахождение коэффициентов для степенных функций

Формулу вида $y = a + b\,x^{\,m}$ (24)

с заданным показателем степени *m* можно привести к линейному виду, построив на оси *Ox* функциональную шкалу для функции $X = x^m$.

Функция Mathcad *pwrfit* служит для нахождения коэффициентов для степенных функций.

pwrfit(*vx, vy, vg*)
Определяет вектор, содержащий коэффициенты для кривой показателя степени вида $a\,x^{\,b} + c$.

Аргументы:

vx - вектор значений вещественных данных *x*;
vy - вектор значений вещественных данных *y*;
vg - вектор параметров *a, b* и *c* в показательном уравнении.

Векторы *vx* и *vy* должны иметь одно и то же число элементов. Вектор оценок *vg* необходим для инициализации. Уменьшение значения встроенной системной переменной TOL может увеличивать точность аппроксимациис использованием функции *pwrfit*.

На Рисунке 23 поясняется использование функции *pwrfit***.**

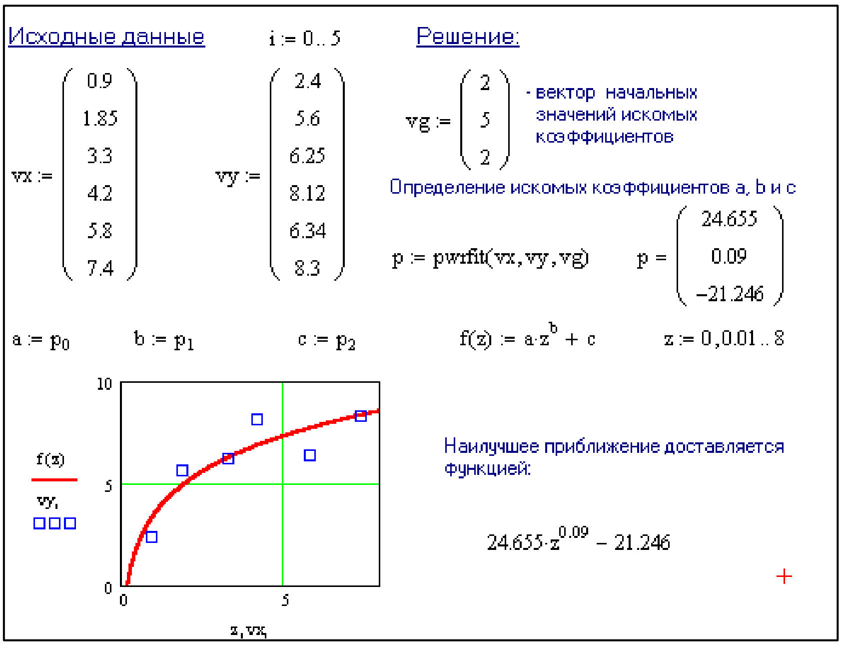


Рис. 23. Аппроксимация данных с использованием функции prwfit

Подбор коэффициентов для показательных функций

Из наиболее часто встречающихся на практике зависимостей осталось рассмотреть зависимость вида

$y = a\,e^{bx}$ (25)

Функция Mathcad *expfit* служит для нахождения коэффициентов для степенных функций.

expfit (*vx, vy, vg*)

Аргументы:

vx - вектор значений вещественных данных *x*;

vy - вектор значений вещественных данных *y*;

vg - вектор параметров *a, b* и *c* в показательном уравнении.

Векторы *vx* и *vy* должны иметь одно и то же число элементов. Вектор оценок *vg* необходим для инициализации. Уменьшение значения встроенной системной переменной TOL может увеличивать точность аппроксимации с использованием функции *expfit*.

На Рисунке 24 поясняется использование функции *expfit***.**

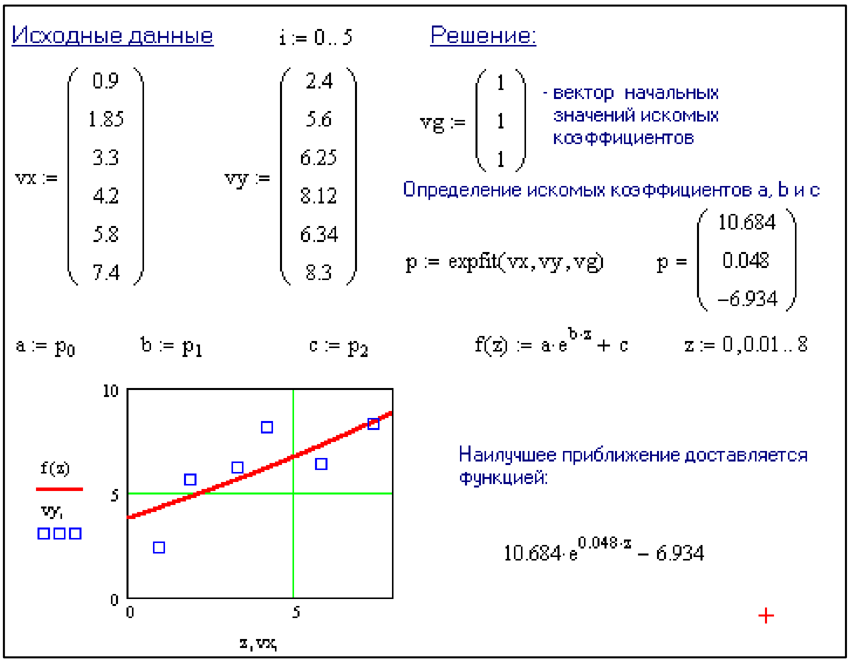


Рис. 24. Аппроксимация данных с использованием функции expfit

Сглаживание данных эксперимента

Сглаживание предполагает использование набора значений y (и, возможно, x) и возвращение нового набора значений y, который является более гладким, чем исходный набор. В отличие от аппроксимации, сглаживание приводит к новому набору значений y, а не к функции, которая может оценивать значения между заданными точками данных.

Функция *supsmooth* использует симметричную линейную процедуру сглаживания Методом наименьших квадратов по правилу k-ближайших точек, чтобы выполнить локальную линейную аппроксимацию исходных данных. *supsmooth* адаптивно выбирает различную ширину полосы сглаживания для различных частей данных.

supsmooth(*vx, vy*)

Возвращает n-мерный вектор, созданный локальным использованием симметричной линейной процедуры сглаживания методом наименьших квадратов по правилу k-ближайших точек, в которой k выбирается адаптивно.

Аргументы:

vx,vy – n-мерные векторы вещественных чисел. Элементы *vx* должны быть расположены в порядке возрастания.

Рисунок 25 показывает пример использования функции *supsmooth* для сглаживания экспериментальных данных.

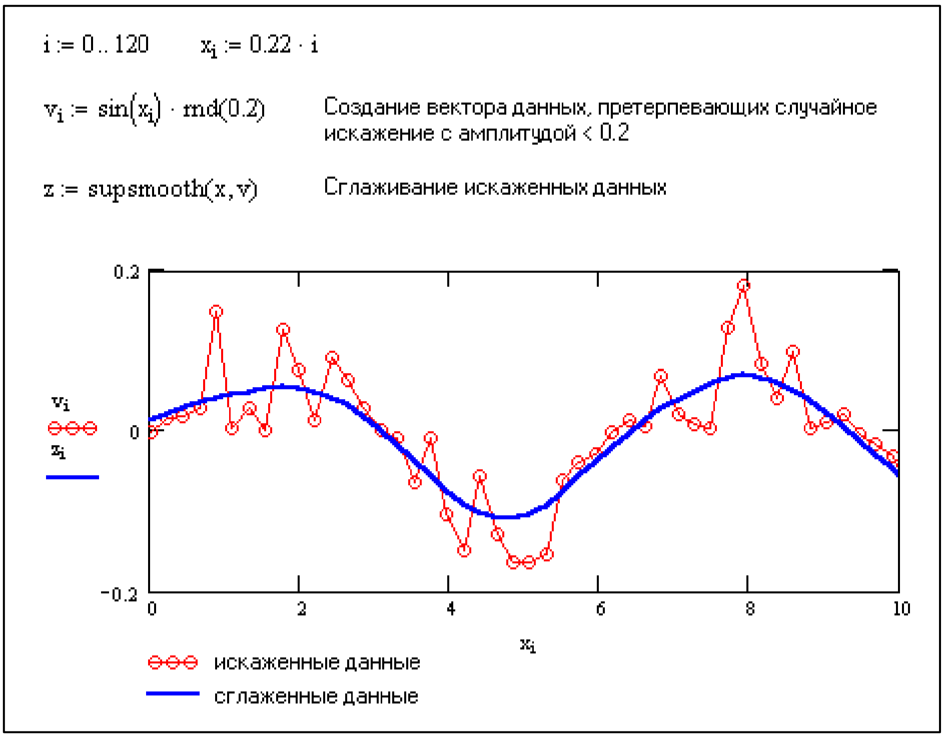


Рис. 25. Сглаживание зашумленных данных с помощью функции supsmooth

В Mathcad имеется еще 2 функции для сглаживания экспериментальных данных. Сглаживание с помощью функции *ksmooth* полезно, когда данные взяты в точках, отделяемых друг от друга интервалами примерно равной ширины.

ПРАКТИЧЕСКОЕ ЗАДАНИЕ

Упражнение 1. Формирование векторов

Из файлов данных **C:\Variant\vx*.txt** и **C:\Variant\vy*.txt** (где * - номер варианта), используя функцию *READPRN*, сформировать векторы x_i и y_i соответственно. С помощью функции *length* определить число элементов n в векторах x и y.

Упражнение 2. Аппроксимация многочленами, определение линейной зависимости

1. Аппроксимировать многочленами 1-ой, 2-ой и 5-ой степеней по *методу наименьших квадратов* функцию, заданную таблицей значений x_i и y_i. Сравнить качество приближений, вычислив сумму квадратов отклонений для каждого случая. Построить графики многочленов и отметить узловые

точки (x_i, y_i).

Указание: составить нормальную систему метода наименьших квадратов и решить ее с помощью встроенной функции *lsolve.*

2. Для сформированных данных (x_i, y_i) определить параметры *линейной зависимости* с использованием встроенных функций Mathcad:

- *slope* и *intercept*;
- *line;*
- *medfit.*

Отобразить графически совокупность точек векторов x_i и y_i и результаты проведенной аппроксимации

3. Аппроксимировать данные из векторов x_i и y_i

- полиномами 2 и 5-ой степени при помощи функций *regress* и *interp*;
- наборами полиномов второго порядка с помощью функций *loess* и *interp*, (при *span* равном 0,6 и 2,0).

Отобразить графически результаты аппроксимации.

Упражнение 3. Аппроксимация экспериментальных данных

Аппроксимировать экспериментальные данные из таблиц значений x_i и y_i линейной комбинацией функций:

$$f(x) = a_1 f_1(x) + a_2 f_2(x) + a_3 f_3(x).$$

Коэффициенты вектора *a* найти с помощью функции *linfit*. Отобразить графически совокупность точек векторов x_i и y_i и результаты проведенной *аппроксимации.*

Варианты упр. 3

№ варианта	$f_1(x)$	$f_2(x)$	$f_3(x)$
1	e^x	$1/\sqrt{1+2\cos^2 x}$	$\sin x$
2	$1/(1+x^2)$	e^x	$\sin(3x)$
3	$1/(1+x^2)$	$e^{\sin x}$	x
4	$\mathrm{arctg}\, x$	$\ln\ln x$	$\sin x$
5	$e^{-x^2/2}$	$1/x$	e^{-x}
6	$(1+x)/(2+x)$	$\cos(x/10)$	$\cos x$
7	$1/(1+e^{x^2})$	$\sqrt{1+x^2}$	$\cos x$
8	$\cos(x/2)$	$2-\cos x$	$\sin(x/2)$
9	$1/(1+e^x)$	$\mathrm{arctg}\sqrt{x}$	$\sin(3x)$
10	$\ln(x+5)$	$\sqrt{1+x}$	$\sin x$
11	$1/x$	$\sqrt{1+x}$	$1/x^2$
12	$\cos x$	$1/(1+x+x^2)$	$1/(1+x)$
13	e^x	$\cos 4x$	$-e^{x/2}$
14	$\sqrt{1+e^{-x}}$	$e^{x/3}$	$\sin^2(3x)$
15	$1/(1+x+x^2)$	$\cos(x/10)$	$\cos(x/10)$

Упражнение 4. Использование функций pwrfit или expfit

Зависимость между величинами *x* и *y* (Таблица 12) описывается функцией $y = f(x, a, b)$, где *a* и *b* – неизвестные параметры. Найти эти параметры с помощью функций *pwrfit* или *expfit* в зависимости от варианта.

Варианты упр. 4

№ варианта	$y = f(x, a, b)$	**Величины x и y**	
1	$a\,x^{b}$	x	0.1; 0.3; 0.4; 0.7; 0.9; 0.95
		y	0.01; 0.13; 0.32; 1.71; 3.65; 4.29
2	$a\,e^{bx}$	x	0.1; 0.4; 0.9; 1.2; 1.6; 2
		y	2.54; 5.22; 17.34; 35.63; 93.05; 243.02
3	$a\,x^{b}$	x	0.1; 0.2; 0.4; 0.5; 0.8; 1
		y	1; 1.52; 2.31; 2.64; 3.5; 4
4	$a\,e^{bx}$	x	1; 1.6; 1.9; 2.2; 2.6; 3
		y	17.93; 44.09; 69.15; 108.45; 197.61; 360.07
5	$a\,x^{b}$	x	0.1; 0.5; 0.9; 1.5; 1.6; 2
		y	0.0005; 0.06; 0.36; 1.69; 2.05; 4
6	$a\,e^{bx}$	x	1; 1.2; 1.4; 1.5; 1.8; 2
		y	0.9; 0.77; 0.65; 0.6; 0.47; 0.4
7	$a\,x^{b}$	x	0.1; 0.2; 0.25; 0.5; 0.9; 1
		y	6.32; 4.47; 4; 2.83; 2.11; 2
8	$a\,e^{bx}$	x	0.1; 0.6; 0.9; 1.2; 1.5; 2
		y	3.12; 3.81; 4.3; 4.85; 5.47; 6.68
9	$a\,x^{b}$	x	0.1; 0.4; 0.9; 1.2; 1.6; 2
		y	0.01; 0.22; 1.55; 3.1; 6.18; 10.56
10	$a\,e^{bx}$	x	0.1; 0.6; 0.9; 1.2; 1.5; 2
		y	2.71; 1.65; 1.22; 0.9; 0.67; 0.41
11	$a\,x^{b}$	x	1; 1.6; 1.9; 2.2; 2.6; 3
		y	4; 8.1; 10.48; 13.05; 16.77; 20.78
12	$a\,e^{bx}$	x	0.1; 0.2; 0.25; 0.5; 0.9; 1
		y	1.9; 1.81; 1.76; 1.56; 1.28; 1.21
13	$a\,x^{b}$	x	0.1; 0.6; 0.9; 1.2; 1.5; 2
		y	1.19; 2.45; 2.88; 3.23; 3.53; 3.96
14	$a\,e^{bx}$	x	0.1; 0.2; 0.4; 0.5; 0.8; 1
		y	4.25; 4.51; 5.08; 5.4; 6.46; 7.29
15	$a\,x^{b}$	x	1; 1.2; 1.4; 1.5; 1.8; 2
		y	2; 1.73; 1.53; 1.45; 1.25; 1.15

Упражнение 5. Сглаживание экспериментальной функции

1. Используя функцию равномерного распределения *rnd,* создать вектор данных y, претерпевающих случайное искажение с амплитудой $0.1v$: $y_i := \cos(x_i)\ \mathrm{rnd}\ (0.1\ v)$

где v - номер варианта, $i := 0..50$, $x_i := 0.1i$.

2. Выполнить сглаживание экспериментальной функции, заданной таблицей значений x_i и y_i, с помощью встроенной функции Mathcad *supsmooth.* Результаты сглаживания отобразить графически.

3. Выполнить сглаживание экспериментальной функции, заданной таблицей значений x_i и y_i, используя встроенные функции Mathcad: *medsmooth* и ***ksmooth.*** Результаты сглаживания отобразить графически.

Контрольные вопросы

1. В каких случаях применение интерполяции нецелесообразно?
2. В чем состоит принцип наименьших квадратов?
3. Влияет ли число наблюденных данных на число параметров в аппроксимируемой функции?
4. С помощью какой функции в Mathcad возможна аппроксимация в виде линейной комбинации функций?
5. Каким образом можно определить вид аппроксимирующей функции?
6. Назовите функции полиномиальной аппроксимации в Mathcad.
7. Влияние различных значений span на функцию loess.
8. Назовите функции Mathcad для линейной аппроксимации.
9. Назовите особенности определения коэффициентов для степенных функций.
10. Назовите особенности определения коэффициентов для показательных функций.