Министерство образования РФ

Томский государственный университет систем управления и радиоэлектроники

Кафедра радиотехнических систем (РТС)

Учебное пособие
«Начальные сведения о MathCAD»
для студентов технических вузов

2016

Министерство образования РФ

Томский государственный университет систем управления и радиоэлектроники

Кафедра радиотехнических систем (РТС)

Утверждаю:
Зав. кафедрой РТС, проф., д.т.н.
_______________ С.В. Мелихов
______________________ 2016 г.

Учебное пособие «Начальные сведения о MathCAD» для студентов технических вузов

Разработчик:
Ст. преподаватель каф. РТС
_____________ Ноздреватых Д.О.
_____________________2016 г.

2016

Аннотация

Учебное пособие **«Начальные сведения о MathCAD»** для студентов технических вузов представляет собой теоретический материал для работы в математическом пакете **MathCAD.**

Каждый раздел заканчивает заданиями для самостоятельной работы и закрепления пройденного материала (для преподавателей вузов могут быть заданиями для лабораторных работ).

Содержание

Введение

Одной из основных областей применения ПК являются математические и научно-технические расчеты. Сложные вычислительные задачи, возникающие при моделировании технических устройств и процессов, можно разбить на ряд элементарных: вычисление интегралов, решение уравнений, решение дифференциальных уравнений и т. д. Для таких задач уже разработаны методы решения, созданы математические системы, доступные для изучения студентам младших курсов втузов.

Первый этап – компьютер, оправдывая свое название (в переводе с англ. «вычислитель»), работал как мощный программируемый калькулятор, способный быстро и автоматически (по веденной программе) выполнять сложные и громоздкие арифметические и логические операции над числами.

Успехи вычислительной математики и постоянно совершенствующиеся численные методы позволяют решить таким способом любую математическую задачу применительно к любой отрасли знаний. Важно отметить, что результат вычислений при этом представляется одним конечным числом в арифметическом виде, то есть при помощи десятичных цифр. Иногда результат представляется множеством (массивом, матрицей) таких чисел, но существо представления от этого не меняется – результат в виде конечного десятичного арифметического числа.

Однако такой результат часто не удовлетворял профессиональных математиков, и вот почему. Подавляющее большинство результатов нетривиальных математических вычислений в классической математике традиционно записывается в символьной форме: с использованием специальных общеизвестных чисел: π, e, C, а иррациональные значения – с помощью радикала. Считается, что в противном случае имеет место принципиальная потеря точности.

Другой классический пример, вызывающий замечание математика – выражение, знакомое любому школьнику:

$$sin^2(x) + cos^2(x)$$

всегда равное единице; а в компьютере либо будет предпринята попытка вычислить это выражение (с неизбежными ошибками округления), либо будет выдано сообщение о неопределенности аргумента X и всякие дальнейшие действия будут прекращены.

На этом первый этап завершился.

Естественно, вслед за стремительным совершенствованием компьютерных систем человеку в компьютерных расчетах захотелось большего: почему бы не заставить компьютер выполнять преобразования традиционными для математики способами (дробно-рациональные преобразования, подстановки, упрощения, решение уравнений, дифференцирование и т.п.).

Их принято называть преобразованиями в символьном виде или аналитическими преобразованиями, а результат получать не как раньше – в виде одного числа, а в виде формулы.

К этому моменту практически все области человеческой деятельности оказались охваченными каждая своим собственным математическим аппаратом и обзавелись собственными пакетами прикладного программного обеспечения (ППО). При этом всем понадобился универсальный математический инструмент, ориентированный на широкий круг пользователей, которые не являются ни профессионалами в математике, ни программистами, воспитанными на узкоспециальных, малопонятных большинству конечных пользователей компьютерных языках.

Это привело к созданию компьютерных систем символьной математики, рассчитанных на широкие круги пользователей – непрофессионалов в математике. Так началась с середины 60-х годов XX века эра **систем компьютерной математики** (СКМ), по-английски CAS – Computer algebra system.

В конце 60-х годов в России на отечественных ЭВМ серии "Мир", разработанных под руководством академика В. Глушкова, была реализована СКМ на языке программирования "Аналитик", обладающая всеми возможностями символьных вычислений, впрочем, с весьма скромными, по нынешним понятиям, характеристиками.

Конечно, даже самые простые неинтеллектуальные компьютерные математические справочники представляют большой практический интерес – ведь ни один самый способный человек не в состоянии вместить в своей голове все математические законы и правила, созданные за многовековую историю человечества.

Данные об особенностях существующих СКМ приведены в табл. В1.

Таблица В1. Современные СКМ и их возможности

Система	**Назначение и возможности**	**Недостатки**
MathCAD 13, MathCAD 14	Система универсального назначения в основном для непрофессиональных математиков и целей образования всех ступеней. Продуманный интерфейс представления данных в традиционной математической форме и изумительная графика на всех этапах работы, включая ввод. Ввод с помощью выбора из панелей инструментов или из меню практически без	Достаточно примитивные средства программирования. Дороговизна электронных книг и библиотек, отсутствие русифицированных версий самого пакета и дополнительных библиотек (книг). Затруднена символьная обработка дифференциальных уравнений. Не создается итоговый исполняемый *.exe-файл; для запуска документа необходимо наличие пакета

	использования клавиатуры. Мощный и исчерпывающий набор операторов и функций. Множество примеров, электронных книг и библиотек, готовых решений практических задач. Ядро символьных вычислений импортировано из СКМ Maple. Предоставление серверных услуг профессионального пакета. Легкость переноса документа в другие приложения	СКМ MathCAD. Затруднения при выполнении тригонометрических преобразований
Maple V R4/R5/R6	Университетское высшее образование и научные расчеты. Мощное ядро символьных вычислений – возможности аналогичны СКМ MathCAD, содержащее до 3000 функций. Мощнейшая графика. Удобная справочная система. Средства форматирования документов	Повышенные требования к аппаратным ресурсам. Отсутствие синтеза звуков. Ориентация на опытных пользователей и специалистов по математике. Все недостатки аналитических действий аналогичны СКМ MathCAD
Mathematica 5/7	Высшее образование и научные расчеты. Наиболее развитая система символьной математики. Единственная СКМ, обеспечивающая символьное решение дифференциальных уравнений. Совместимость с разными компьютерными платформами. Уникальная трехмерная графика. Поддержка синтеза звука. Развитые средства форматирования документов. Программный синтез звуков.	Высокие требования к аппаратным ресурсам. Чрезмерная защита от копирования. Слабая защита от некорректных задач. Ориентация на опытных пользователей. Ввод задач на уникальном языке функционального программирования. Непривычная индикация функций запуска вычислений.
MATLAB 7.*	Образование (в том числе техническое), научные расчеты, численное моделирование, и расчеты, ориентированные на применение матричных	Очень высокие требования к аппаратным ресурсам. Практически отсутствует возможность символьных вычислений. Относительно

	методов, при этом скаляр рассматривается как матрица 1x1. Уникальные матричные средства, обилие численных методов, описательная (дескрипторная) графика, высокая скорость вычислений, легкость адаптации к задачам пользователя благодаря множеству пакетов расширения системы. Развитый язык программирования с возможностями объектно-ориентированного программирования (ООП), совместимость с алгоритмическим языком Java	высокая стоимость. Ввод задач на уникальном языке программирования

Рассмотрим внутреннюю архитектуру СКМ на примере наиболее мощной, по мнению ряда авторитетных специалистов, СКМ Mathematica, обладающей наиболее развитой системой символьной математики. На рис.В.1 представлена ее программная архитектура.

Рис. В.1. – Архитектура обобщенной СКМ

Центральная часть – ядро (Kernel) системы СКМ реализует алгоритм функционирования СКМ, обеспечивает совместное функционирование всех ее частей, организует прием и интеллектуальную обработку запроса пользователя, а затем – вызов нужной процедуры решения. В ядре помещается большое количество встроенных функций и операторов системы. Их количество в современных СКМ может достигать многих тысяч. Например, ядро системы Mathematica 4 содержит данные более чем 5000 одних только интегралов, хотя для интегрирования используются только несколько встроенных функций.

Поиск и выполнение функций и процедур, встроенных в ядро СКМ, выполняется быстро, если их там не слишком много. Поэтому объем ядра ограничивают, но к нему добавляют встроенные в СКМ библиотеки процедур и функций, использующихся относительно редко. При этом общее число доступных пользователю математических функций ядра и этих встроенных библиотек достигает многих тысяч.

Кардинальное расширение возможностей СКМ и их приспособленность к нуждам конкретных пользователей для углубленного решения определенного круга задач (например, задач теоретической и прикладной статистики, векторного анализа) достигается за счет установки внешних пакетов расширения. Эти пакеты, приобретаемые отдельно, делают возможности СКМ практически безграничными.

Все эти библиотеки, пакеты расширений и справочная система современных СКМ (назовем их инструментами СКМ) содержат не только и не просто знания в области математики, накопленные за много веков ее развития (этим никого не удивишь: именно такие возможности характерны для широко распространенного класса ИПО – информационно-поисковых систем). Но восхищает, что эти инструменты удивительным образом автоматически и творчески используют такие знания для решения задач, где нужно выбрать и уметь применить один, единственный из многих десятков, неочевидный метод решения. Например, СКМ могут мгновенно найти неопределенный интеграл либо сразу же сообщить о невозможности его представления элементарными функциями – задача непростая, даже для профессионального математика. Не менее впечатляет и то, что если после получения искомой формулы перейти к началу документа и задать входящим в эту формулу параметрам конкретные числовые значения, мгновенно будет получен ее численный результат. В состав любой СКМ входит набор редакторов (на рис. В.1 они названы редакторами по направлениям): текстовый, формульный, графический редакторы, средства поддержки работы в сети и HTML(XML)-средства, пакеты анимации и аудиосредства.

Благодаря всем этим возможностям СКМ могут быть отнесены к программным продуктам самого высокого на сегодняшний день уровня – интеллектуального. Такие программы в настоящее время объединяются термином "базы знаний". Современные СКМ, по мнению признанных авторитетов, предоставляет неискушенному пользователю возможности выпускника математического вуза в областях численных методов расчета, математического анализа, теории матриц и других общих разделах высшей математики, позволяющих получить конструктивные результаты.

Конечно, в абстрактных разделах математики, типа функционального анализа или вопросов "существования и единственности..." СКМ пока вряд ли могут быть полезны (кроме как для предоставления нужной справки, что очень даже немало), но в прикладных задачах, для которых СКМ и создавались, такие разделы математики обычно не задействованы.

Цель пособия – научить пользоваться простейшими методами вычислений с использованием современных информационных технологий. Наиболее подходящей для этой цели является одна из самых мощных и эффективных математических систем – MathCAD, которая занимает особое место среди множества таких систем (Matlab, Maple, Mathematica и др.).

MathCAD – это мощная и в то же время простая универсальная среда для решения задач в различных отраслях науки и техники, финансов и экономики, физики и астрономии, математики и статистики… MathCAD остается единственной системой, в которой описание решения математических задач задается с помощью привычных математических формул и знаков. MathCAD позволяет выполнять как численные, так и аналитические (символьные) вычисления, имеет чрезвычайно удобный математико-ориентированный интерфейс и прекрасные средства научной графики.

MathCAD является программным средством, предназначенным для выполнения математических расчетов. Вид уравнений на экране MathCAD совпадает с первичной математической записью. Как в электронных таблицах, любое изменение содержимого рабочего документа MathCAD вызывает обновление всех зависимых результатов, в том числе перерисовку графиков. Текстовый процессор MathCAD поддерживается множеством шрифтов.

Система MathCAD существует в нескольких основных вариантах:

- MathCAD Standard – идеальная система для повседневных технических вычислений. Предназначена для массовой аудитории и широкого использования в учебном процессе;
- MathCAD Professional – промышленный стандарт прикладного использования математики в технических приложениях. Программа ориентирована на математиков и научных работников, проводящих сложные и трудоемкие расчеты.
- MathCAD Professional Academic – пакет программ для профессионального использования математического аппарата с электронными учебниками и ресурсами.

Данное пособие ориентировано на использование пакета MathCAD Professional.

Назначения и возможности MathCAD:

- ввод на компьютере разнообразных математических выражений (для дальнейших расчетов или создания документов, презентаций, Web-страниц);
- проведение математических расчетов;
- подготовка графиков с результатами расчетов;
- ввод исходных данных и вывод результатов в текстовые файлы или файлы с базами данных в других форматах;

- подготовка отчетов работы в виде печатных документов;
- подготовка Web-страниц и публикация результатов в Интернете;
- получение различной справочной информации из области математики, и многое другое.

Особенности работы в среде MathCAD:

- в отличие от большинства других математических приложений, построен по принципу «что видите, то и получаете»;
- математические выражения и текст вводятся с помощью формульного редактора MathCAD, который по возможностям и простоте использования не уступает, к примеру, редактору формул, встроенному в Microsoft Word;
- математические расчеты производятся немедленно, в соответствии с введенными формулами;
- графики различных типов (по выбору пользователя) с богатыми возможностями форматирования вставляются непосредственно в документы;
- возможен ввод или вывод данных в файлы различных форматов;
- документы могут быть распечатаны непосредственно в MathCAD в том виде, который пользователь видит на экране компьютера, или сохранены в формате RTF для последующего редактирования в более мощных текстовых редакторах (например, Microsoft Word);
- возможно полноценное сохранение документов MathCAD в формате Web-страниц (генерация вспомогательных графических файлов происходит автоматически);
- имеется опция объединения разрабатываемых вами документов в электронные книги, которые, с одной стороны, позволяют в удобном виде хранить математическую информацию, а с другой – являются полноценными MathCAD-программами, способными осуществлять расчеты;
- символьные вычисления позволяют осуществить аналитические преобразования, а также мгновенно получать разнообразную справочную математическую информацию.

MathCAD имеет свою справочную систему. Обращение к справочной системе: клавиша [F1]; меню HELP; комбинация [Shift]+[F1] – для открытия контекстной, т.е. обозначенной указателем, справки по теме.

MathCAD позволяет решать в одном рабочем документе текст, математические формулы и строить необходимые графики. Основные приемы в MathCAD такие же, как и в любом приложении Windows.

1 Интегрированная Среда СКМ Mathcad

Интегрированная Среда СКМ MathCAD является системой СКМ универсального назначения и наиболее приспособлена для решения широкого спектра, а точнее – практически любых математических задач, в основном непрофессиональными математиками, а также для эффективного использования во всех областях сферы образования.

По сей день они остаются единственными математическими системами, в которых описание решения математических задач дается с помощью привычных математических формул и знакомых символов. Такой же вид имеют и результаты вычислений. СКМ MathCAD не очень подходит для серьезной профессиональной научной деятельности математиков, она больше предназначена для решения не слишком изощренных математических задач, выполнения технических расчетов любой сложности, а главное – не имеет конкурентов в области образования. Благодаря высоким характеристикам, СКМ MathCAD полностью оправдывает термин «CAD» в своем названии (Computer Aided Design), подтверждающий принадлежность к классу наиболее сложных и совершенных систем автоматического проектирования – САПР. Система MathCAD является типичной интегрированной системой, то есть объединяющей в своем составе несколько обособленных программных средств для решения определенного круга самостоятельных задач.. Первоначально она была предназначена для сугубо численных вычислений и ориентирована под MS-DOS, но, начиная с версии 3.0 (1990 г.), работает под ОС Windows и имеет достаточно широкий набор средств для символьных и графических вычислений.

Все действия в СКМ MathCAD сразу оформляются в виде документа, состоящего из рабочих листов, на которых помещается описание алгоритма, рабочие формулы, комментарии, иллюстрации, графики, таблицы. Форма такого документа максимально приспособлена для печати, передачи по сети Internet и не требует дополнительного редактирования. С другой стороны, этот документ, имеющий расширение .mcd, содержит в скрытом виде всю программу вычислений. Он может быть импортирован как для целей издания, так и для продолжения и совершенствования программных вычислений. Весь документ или отдельные его части могут быть заблокированы для редактирования путем задания пароля.

На рис.1.1 приведена архитектура СКМ MathCAD. Центральным блоком являются два ядра: собственно ядро СКМ и ядро символьных вычислений, аналогичное СКМ Maple, приобретенное у разработчика – фирмы Waterloo Maple.

Встроенные в среду MathCAD электронные книги (e-Books) содержат примеры, справки и типовые расчеты из различных областей науки, техники, экономики. Любой фрагмент из этих книг можно скопировать на рабочий лист документа и выполнить.

Библиотеки и пакеты расширений, ориентированные на решение различных прикладных задач, поставляются и устанавливаются разработчиком отдельно.

Мощный интерфейс СКМ MathCAD не требует программирования при вводе заданий и индикации результатов – все это выполняется в традиционной форме на общепринятом языке математических символов и формул без применения каких-либо специальных команд или операторов. Показательно, что в каждом алгоритмическом языке простое возведение в степень, в меру фантазий разработчиков языка, выполняется при помощи уникальных собственных условных обозначений – всевозможных стрелочек, крышечек, двойных звездочек и Бог знает чего еще, а то и вовсе отсутствует и требует вызова специальных функций – как в языках семейства Си. В MathCAD эта операция имеет привычный вид.

Интерфейс является визуальным – то есть практически любые действия в СКМ можно выполнять без помощи клавиатуры, просто выбирая нужные пункты меню или инструменты на панелях. В этом интерфейсе реализован принцип «WYSIWYG» – что видим на экране, то и получаем в работе и при выводе.

Интерфейс интеллектуален – конечно, здесь далеко до интеллекта Visual Studio-2010, но во многих случаях он не допустит ошибочных действий пользователя.

Упомянутый входной язык ввода является интерпретирующим, то есть промежуточные результаты появляются по мере ввода очередной формулы. Сама же СКМ MathCAD написана на одном из самых мощных языков – С++. По мере того, как пользователь набирает на рабочем листе текст алгоритма вычислений, среда сама составляет скрытую программу на промежуточном языке связи, которая затем сохраняется в виде файла с расширением .mcd. К сожалению, исполняемого файла с расширением .exe пакет MathCAD не формирует – для работы с импортированным документом необходимо наличие установленного приложения MathCAD. А вот вставить образ документа либо отдельный его фрагмент в текстовый редактор, например, MS WORD, через системный буфер никакого труда не представляет. Именно так и вставлялись все иллюстрации в этой главе. Рекомендую после такой вставки фрагмента вызвать на нем контекстное меню – пункт «Формат рисунка…/Размер» и установить в окне «Масштаб по высоте» 128% – для шрифта 12-го кегля наиболее подходящий.

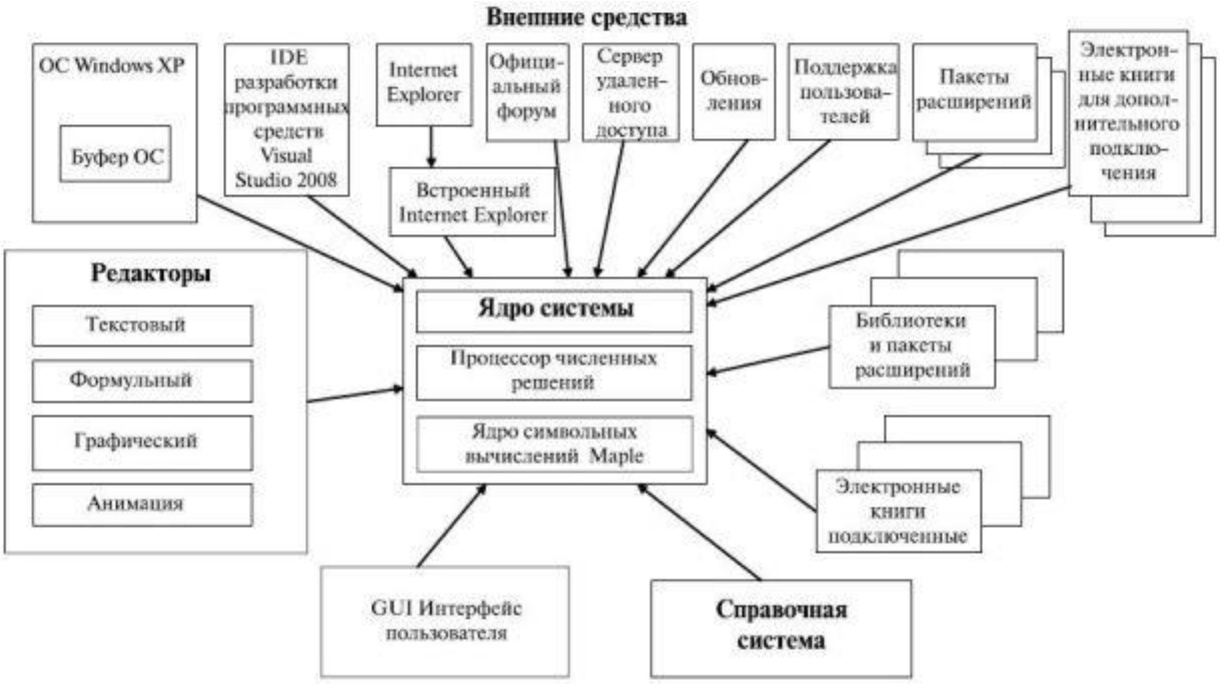


Рис. 1.1 – Архитектура СКМ MathCad

Объектами рабочего листа могут быть формульные текстовые или графические блоки. Действия над блоками выполняются в строгом порядке слева направо, сверху вниз. Блоки, готовящие операции, должны предшествовать выполнению этих операций. При этом организована сквозная передача данных от одного объекта к другому. Изменение входных данных мгновенно обеспечивает пересчет результатов.

Контент (содержание) этой СКМ можно рассматривать в качестве исключительно мощного справочного средства по математике. Кроме того, в СКМ MathCAD интегрированы формульный, текстовый и графический редакторы, позволяющие упростить ввод многоэтажных сложнейших формул и получить итоговый документ. Промежуточные действия в ходе символьных преобразований в СКМ MathCAD скрыты от пользователя, но не следует забывать, что для получения конечного результата используются сложнейшие рекурсивные алгоритмы, мало знакомые широкому пользователю и зачастую не оптимальные на взгляд математика. При этом никто не запрещает пользователю пошаговое выполнение и индикацию знакомых из литературы алгоритмов, что значительно упрощает решение при известном конечном результате. В СКМ MathCAD не создается итоговый исполняемый *.exe файл, значит, для просмотра готового (например, импортированного) документа требуется наличие установленного пакета СКМ MathCAD.

Перечислим основные возможности Среды MathCAD.

Общие возможности

1. Разработка и редактирование документов, содержащих как математические формулы любой сложности, так и все встроенные

инструменты Среды MathCAD. Подготовка этих документов к изданию или передаче по сети Internet.

2. Использование общепринятого расширяемого языка разметки XML как универсального способа организации обмена данными с другими приложениями. Это позволяет преобразовывать файлы MathCAD в HTML-страницы и в формат PDF.

3. Возможность вставки в документ широкого спектра объектов (см. рис.12.3.)

4. Разработка веб-документов и сетевые возможности по их пересылке, получению обновлений и поддержки.

5. Получение документов MathCAD по сети и выполнение расчетов в этих документах.

6. Получение через Internet и подключение новых книг расширения для реализации дополнительных возможностей среды MathCAD.

7. Доступный официальный форум.

8. Использование серверных услуг среды MathCAD (MathCAD Application Server) – удаленное подключение к пакету MathCAD в полной комплектации через стандартный веб-браузер Internet, даже если MathCAD не установлен на компьютере.

9. Импорт документов из MS Access и MS Excel и математическая обработка данных из этих документов.

10. Наличие обширных справочных материалов по математике и основных физических констант, а также большое количество задач с готовыми решениями по многим разделам науки и техники.

11. Выполнение вычислений любой сложности – использование среды MathCad в качестве сверхмощного научного интеллектуального калькулятора с применением богатой библиотеки встроенных функций (более 680; для сравнения, в MS Excel их около 200), с точностью до 17 значащих цифр (а при использовании специальных операторов – и до 250) и с неограниченными возможностями запоминания промежуточных результатов. При этом имеется возможность вычислений как по введенной в документ формуле целиком, так и по отдельному, выделенному фрагменту формулы.

12. Использование графического редактора для построения двумерных и трехмерных графиков любой сложности, наглядных диаграмм и не только для простого построения, но и для связи графика с формулой, при которой изменение параметра сразу отражается на кривой графика. Имеется также возможность создание объектов движущейся анимации и просмотра импортированных файлов, например, видеофильмов в формате AVI при помощи встроенного в среду MathCAD проигрывателя Playback.

13. Действия с размерностями.

Численные методы вычислений

- Решение уравнений и систем уравнений, как линейных, так и нелинейных. Нахождение корней многочлена.

- Решение неравенств.
- Вычисление определенного интеграла.
- Вычисление несобственных интегралов.
- Вычисление кратных интегралов.
- Численные методы дифференцирования.
- Численное решение обыкновенных дифференциальных уравнений – задача Коши.
- Численное решение обыкновенных дифференциальных уравнений – решение краевой задачи.
- Решение дифференциальных уравнений в частных производных.
- Вычисление суммы и произведения членов ряда.
- Исследование функций и численное определение экстремумов функций одной и нескольких переменных, построение асимптот.
- Решение оптимизационной задачи методом линейного программирования.

Символьные вычисления

1. Выполнение точных вычислений с представлением результатов в традиционной математической форме – с записью ответа в форме радикала и специальных иррациональных чисел π, e, C.
2. Символьные преобразования математических выражений целиком или их фрагментов:
    1. разложение выражений в более простые;
    2. приведение подобных;
    3. разложение на множители;
    4. приведение к общему знаменателю;
    5. вынесение общего множителя;
    6. разложение на элементарные дроби;
    7. вычисление коэффициентов полиномов;
    8. выполнение подстановок.
3. Аналитическое решение уравнений и систем уравнений.
4. Дифференцирование в символьной форме определение производных любых порядков.
5. Аналитическое определение первообразной.
6. Построение касательной и нормали к плоской кривой и к поверхности.
7. Аналитическое вычисление определенного интеграла.
8. Символьное вычисление кратных интегралов.
9. Решение неравенств.
10. Аналитическое вычисление предела.
11. Аналитическое вычисление суммы ряда конечного или бесконечного.
12. Аналитическое вычисление произведения членов ряда конечного или бесконечного.

13. Аналитическое вычисление суммы/произведения членов ряда конечного или бесконечного, когда пределы и шаг изменения индекса члена ряда задаются (например, сложить четные числа от 10 до $10 + к$.)

14. Разложение в ряд Тейлора.

15. Разложение в ряд Фурье.

16. Символьное преобразование Фурье и Лапласа – прямое и обратное.

17. Операции с матрицами в символьной форме: умножение и сложение матриц, поиск обратной матрицы, вычисление определителя, поиск собственных значений и собственных векторов.

Работа с матрицами и матричные вычисления

1. Элементарные матричные действия: создание, импорт, заполнение матриц, задание матриц специального вида, умножение, сложение, транспонирование и сортировка матрицы в целом или ее фрагмента. Выполнение векторизации – однотипных действий над всеми элементами матрицы.

2. Вычисление определителя, размерности, ранга и следа матрицы, скалярное и векторное умножение векторов, вычисление якобиана, например, для перехода к другим системам координат в тройном интеграле. Вычисление собственных значений и собственных векторов, поиск максимального и минимального элемента матрицы.

3. Матричные преобразования: скалярное и векторное умножение векторов, поиск обратной матрицы и решение системы алгебраических линейных уравнений, всевозможные разложения матрицы на произведение матриц специального вида: двух треугольных – верхней и нижней (LU-преобразование), треугольной и ее же транспонированной (разложение Холецкого), ортогональной и верхней треугольной (QR-разложение), сингулярное разложение.

4. Интегрирование среды MathCad с матричной математической системой MATLAB и возможность использования ее аппарата открывает удивительные возможности эффективного решения матричных задач неограниченной сложности.

Решение дифференциальных уравнений

Программирование

Составление программ и выполнение расчетов на упрощенном процедурном алгоритмическом языке с возможностью использования всех процедурных конструкций: условных операторов, циклов, массивов, модуль-функций, модуль-процедур.

Комплексные числа

1. Представление комплексных чисел в традиционной форме, возможность выполнения основных арифметических действия с ними.

2. Возможность автоматического получения результатов многих вычислений в виде комплексного числа (например, всех корней многочлена).

3. Возможность задания комплексного аргумента для многих библиотечных функций и получение математически корректного результата.

Обработка данных и финансовые расчеты

Теория вероятностей и математическая статистика

Математическое моделирование

Специальные возможности по прикладным инженерным и научным расчетам

1. Обработка электрических сигналов и расчет электронных устройств.
2. Виртуальная генерация электрических сигналов и их обработка.

2 Интерфейс Mathcad

MathCAD работает с *документами*. 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*.

Для создания документа в среде MathCAD необходимо в требуемой папке, например, на рабочем столе в папке ***«Петров»***, выбрать из контекстного меню команду ***Создать – MathCAD Document***, с клавиатуры ввести название этого документа, например, **«Задание к ЛР-1 по MathCAD»** и открыть документ, дважды щелкнув на созданный файл. На экране откроется окно (рис. 2.1)

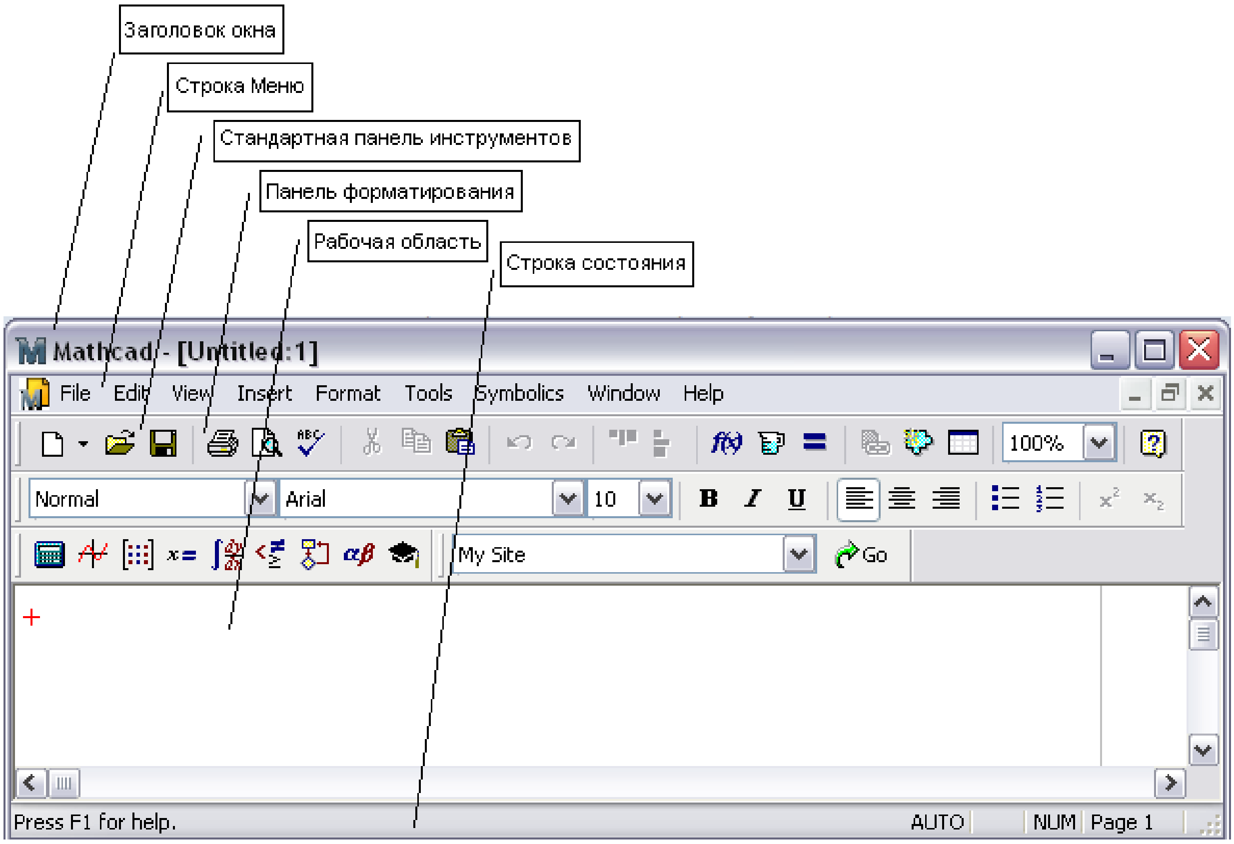


Рис. 2.1 – Созданный документ в MathCAD

Таким образом, оконный интерфейс MathCAD содержит:

1. Заголовок;
2. Строку меню, пункты которого доступны при помощи мыши и клавиатуры (краткое нажатие на клавишу ALT, затем стрелки и ENTER);
3. Стандартной панели инструментов (на них вынесены наиболее часто употребляемые команды). Доступны только при помощи мыши. Активизируются по необходимости из меню инструментов ***View-Toolbars***;
4. Линейка горизонтальная (активизируется ***View-Rules***). Доступна только при помощи мыши;
5. Горизонтальная и вертикальная полосы прокрутки. Доступны при помощи мыши;
6. Рабочая область;
7. Контекстное меню (активизируется щелчком правой кнопки мыши для конкретного положения курсора);
8. Строка состояния (активизируется ***View-Status Bar***). Отображает (слева направо):

- контекстно-зависимая подсказка о готовящемся действии,
- режим вычислений (**AUTO**) или задаваемый вручную (**Calc F9**),
- текущий режим раскладки клавиатуры **NUM**,
- номер страницы, на которой находится курсор.

2.1 Меню

Строка меню располагается в самой верхней части окна MathCAD. Она содержит девять заголовков, щелчок мышью на каждом из которых выводит соответствующее меню с перечнем команд:

- **File** (файл) – команды, связанные с созданием, открытием, сохранением, пересылкой по электронной почте и распечаткой на принтере файлов с документами;
- **Edit** (Правка) – команды, относящиеся к правке текста (копирование, вставка, удаление фрагментов и т.п.);
- **View** (Вид) – команды, управляющие внешним видом документа в окне редактора MathCAD, а также команды, создающие файлы анимации;
- **Insert** (Вставка) – команды вставки различных объектов в документы;
- **Format** (Формат) – команды формирования текста, формул и графиков;
- **Tools** (Инструменты) – команды управления расположением окон с различными документами на экране;
- **Symbolics** (Символика) – команды символьных вычислений;
- **Window** (Окно) – команды управления расположением окон с различными документами на экране;
- **Help** (Справка) – команды вызова справочной информации, сведений о версии программы, а также доступа к ресурсам и электронным книгам.

2.2 Справка по Mathcad

При необходимости получения справочных сведений во время работы можно использовать один из следующих ресурсов.

- Кнопка ***F1*** служит для быстрого контекстного доступа к справочной системе;
- **MathCAD Help** (Справка) – система справки или технической поддержки.

Кроме того, в зависимости от версии пакета возможно получение дополнительной справочной информации по ***Developer's Reference*** (Справке для разработчиков), ***Author's Reference*** (Справке для авторов) или из ***Resource Center*** (Центра описания ресурсов).

В случаях, когда назначение команды или кнопки неизвестно или требуется узнать больше о каком-либо параметре в диалоговом окне, нужные сведения можно получить с помощью всплывающих подсказок. Всплывающие подсказки содержат сведения о различных элементах, отображаемых на экране.

2.3 Панели инструментов

Панели инструментов служат для быстрого (за один щелчок мыши) выполнения наиболее часто применяемых команд. Все действия, которые можно выполнить с помощью панелей инструментов, доступны и через верхнее меню. На рисунке 2.2 изображено окно MathCAD, содержащее три основные панели инструментов, расположенные непосредственно под строкой меню. Кнопки в панелях сгруппированы по сходному действию команд:

- **Standard (Стандартные)** – служит для выполнения большинства операций, таких, как действия с файлами, редакторская правка, вставка объектов и доступ к справочным системам;
- **Formatting (Форматирование)** – для формирования (изменения типа и размера шрифта, выравнивания и т.п.) текста и формул;
- **Math (Математика)** – для вставки математических символов и операторов в документы;

Панель **Math** предназначена для вызова на экран еще девяти панелей (рис. 2.2), с помощью которых, собственно, и происходит вставка математических операций в документы. В прежних версиях MathCAD эти математические панели инструментов назывались *палитрами* (palettes) или *наборными панелями*. Чтобы показать какую-либо из них, нужно нажать соответствующую кнопку на панели **Math**. Перечислим назначение математических панелей:

- **Calculator (Арифметика)** – служит для вставки основных математических операций;
- **Graph (Графики)** – для вставки графиков;
- **Matrix (Матрицы)** – для вставки матриц и матричных операторов;
- **Evaluation (Оценка)** – для вставки операторов управления вычислениями;
- **Calculus (Исчисления)** – для вставки операторов интегрирования, дифференцирования, суммирования;
- **Boolean (Булевы операторы)** – для вставки логических (булевых) операторов;
- **Programming (Программирование)** – для программирования средствами MathCAD;
- **Greek (Греческие символы)** – для вставки греческих символов;
- **Symbolic (Символы)** – для вставки символьных операторов.

При наведении указателя мыши на многие из кнопок математических панелей появляется всплывающая подсказка, содержащая еще и сочетание «горячих клавиш», нажатие которых приведет к эквивалентному действию.

«Горячие клавиши» приведены в Приложении

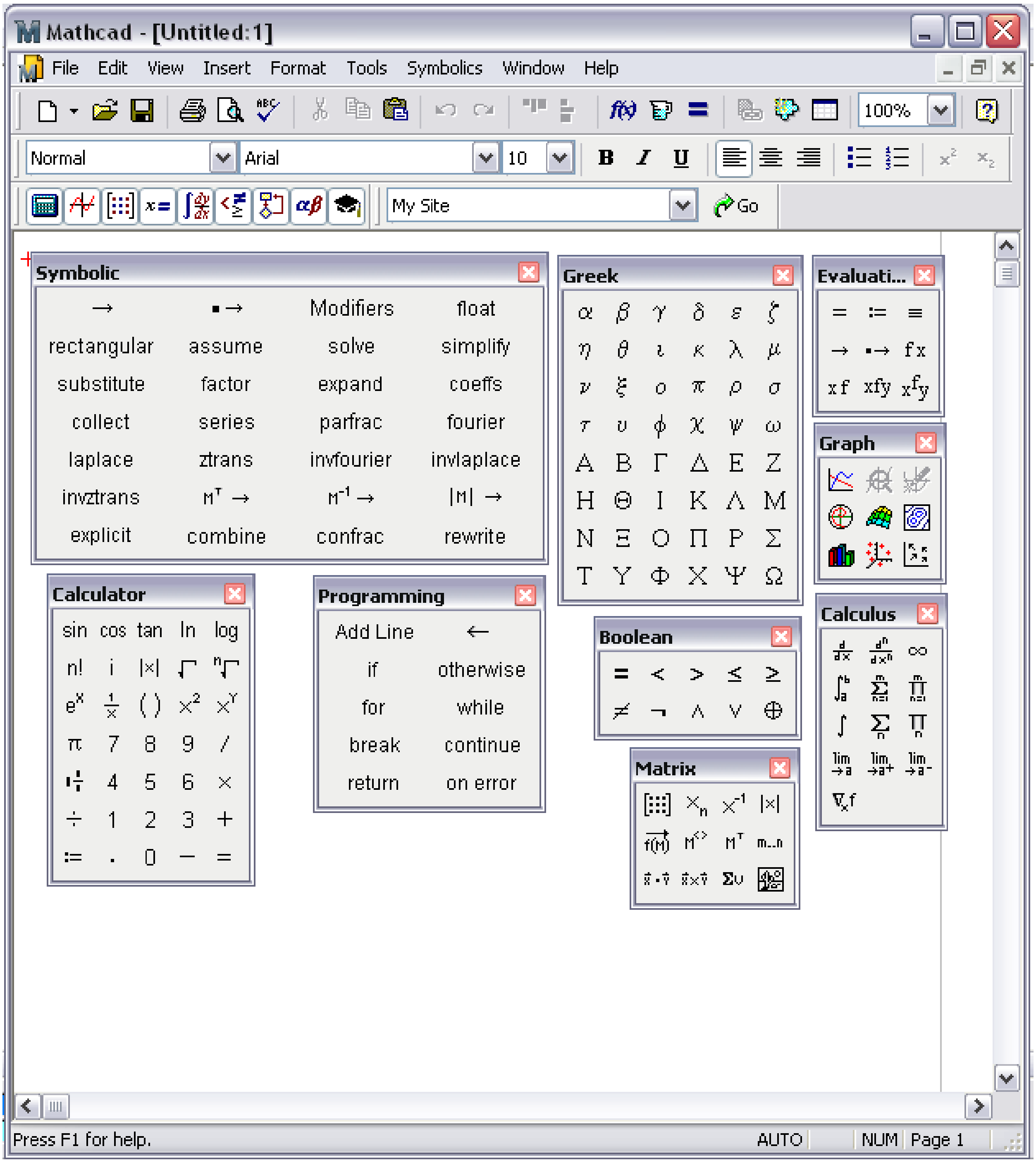


Рис. 2.2 – Панели в пакете MathCAD

Текстовые области можно размещать в любом месте документа командой меню Insert/Text Region или вводом с клавиатуры символа ["].

Для ввода математических формул и графиков можно применять панели (палитры) кнопок в клавиатуру. Панели приведены на следующем рисунке 2.2.

Если при запуске нового файла вы не обнаружили общую панель инструментов **Math** (рис. 2.3), то включить ее можно следующим образом (рис. 2.4):

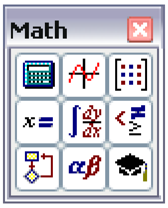


Рис. 2.3 – Общая панель инструментов Math

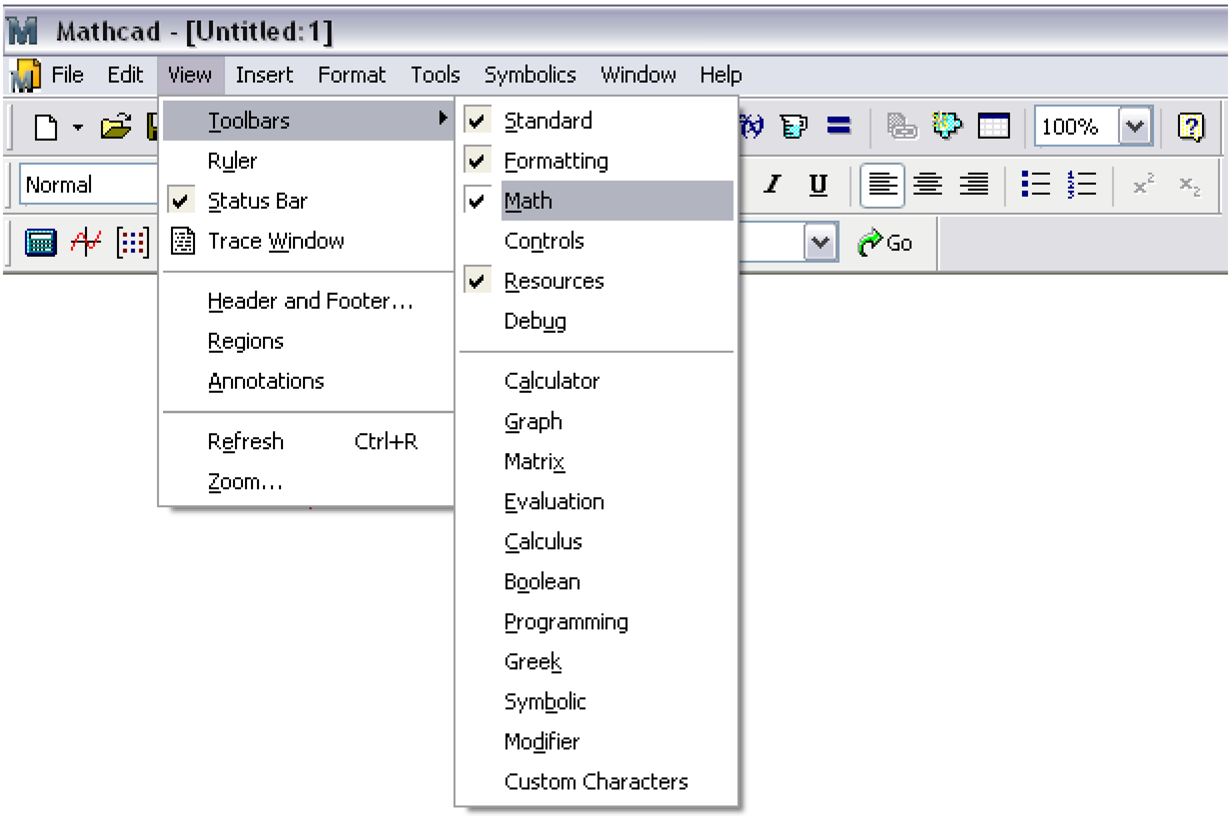


Рис. 2.4 – Включение панели инструментов Math

3 Основы работы с Mathcad

В MathCAD все расчеты организуются на рабочих областях, или «листах» (worksheets), изначально пустых, на которые можно добавлять формулы и текст. Здесь и далее будем называть рабочий лист *документом* MathCAD. Это не совсем точно передает смысл английского термина «worksheet», зато более привычно с точки зрения терминологии Windows-приложений. Каждый Документ представляет собой независимую серию математических расчетов и сохраняется в отдельном файле. Документ является одновременно и листингом MathCAD-программы, и результатом исполнения этой программы, и отчетом, пригодным для распечатки на принтере или публикации в Web.

После запуска пакета в рабочей области окна создается новый пустой документ.

Управление документом осуществляется достаточно просто. Манипулируя меню File можно:

- создать новый документ (New), комбинация клавиш Ctrl+N;
- открыть для редактирования уже имеющийся документ (Open), комбинация клавиш Ctrl+O;
- закрыть документ (Close), комбинация клавиш Ctrl+W;
- сохранить документ (Save), комбинация клавиш Ctrl+S;
- сохранить документ под новым именем или типом (Save As);
- распечатать документ (Print), комбинация клавиш Ctrl+P;
- отослать документ по e-mail (Send);
- назначить параметры распечатки страниц документа (Page Setup);
- просмотреть документ (Print Preview);
- закрыть приложение (Exit).

Кроме того, открыв меню File можно увидеть открытые ранее документы MathCAD.

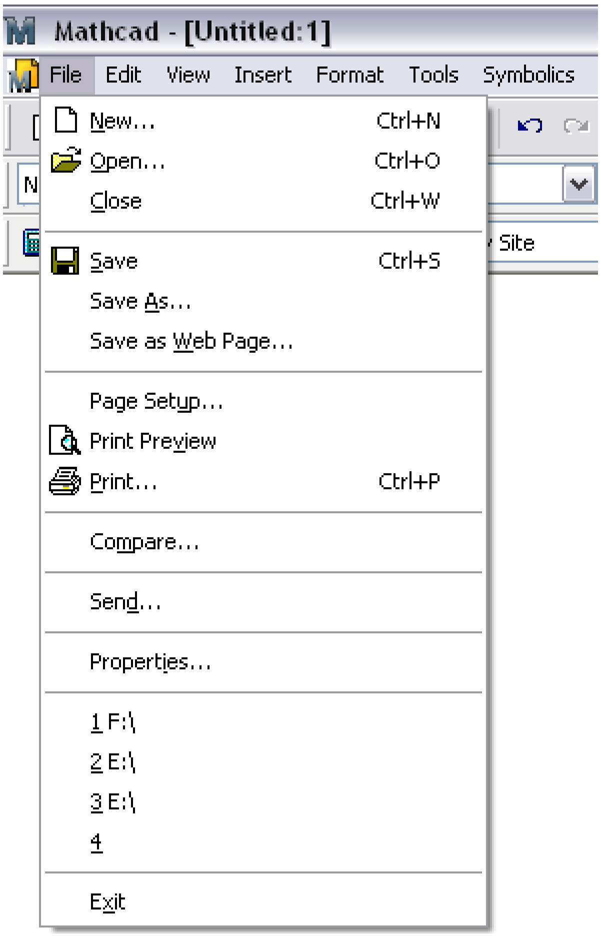


Рис. 3.1 – Вид меню File

3.1 Создания формул и редактирование документа

Язык, на котором изъясняются в MathCAD для изображения констант, переменных величин, операторов, функций, уравнений и иных математических записей, практически полностью совпадает с общепринятым в математике.

Символами этого языка являются: малые и заглавные буквы латинского и греческого алфавита; арабские цифры от 0 до 9; знаки математических операций (+, -, *, /, = и т.д.); имена функций (cos, sin, tan, log, n!, т.д.) и некоторые специальные знаки.

В математике различают константы (целые и вещественные) и переменные величины. Значение константы остается неизменным в процессе выполнения программы, значение переменной – может изменяться.

Пример записи целых констант:

1, -5, 0, 769, -3 и т.д.

Примеры записи вещественных констант:

0.564, -89.439, 7.72 и т.д.

Вещественные константы можно также представить в сжатой форме при большом числе знаков с использованием буквы E в качестве основания 10: aEn, где а – целое или дробное число, E – основание 10, n – целое число, являющееся показателем степени 10.

Большую роль в наборе чисел играет расположение уголка (на экране он синего цвета). Например (рисунок 3.2.а), если уголок расположен, как показано в левой части рисунка, то любые знаки операций (сложение, вычитание и т.п.) будут добавляться к показателю степени, если же уголок расположен, как показано в правой части рисунка, то они будут добавляться ко всему выражению.

3.1.1 Создание формул

Формульный редактор MathCAD позволяет быстро и эффективно вводить и изменять математические выражения. Тем не менее, некоторые аспекты его применения не совсем интуитивны, что связано с необходимостью избежать ошибок при расчетах по этим формулам. Поэтому не пожалейте немного времени на знакомство с особенностями формульного редактора, и впоследствии при реальной работе вы сэкономите гораздо больше.

Элементы интерфейса редактора MathCAD:

- указатель мыши (mouse pointer) – играет обычную для приложений Windows роль, следуя за движениями мыши;
- курсор – обязательно находится внутри документа в одном из трех видов:
- курсор ввода (crosshair) – крестик красного цвета, который отмечает пустое место в документе, куда можно вводить текст или формулу;
- линии ввода (editing lines) – горизонтальная (underline) и вертикальная (insertion line) линии синего цвета, выделяющие в тексте или формуле определенную часть (рис. 3.2б);
- линии ввода текста (text insertion point) – вертикальная красная линия, аналог линий ввода для текстовых областей (рис. 3.3).

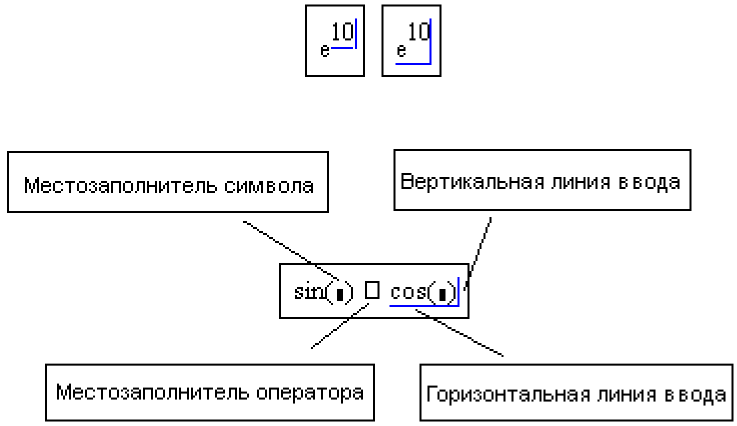


Рис. 3.2 а, б – Линии ввода

Рис. 3.3 – Линия ввода текста

- местозаполнители (placeholders) – появляются внутри незавершенных формул в местах, которые должны быть заполнены символом или оператором:
- местозаполнитель символа – черный прямоугольник;
- местозаполнитель оператора – черная прямоугольная рамка.

MathCAD выполняет расчеты по формулам, которые близки к естественной математической символике. *Формула* – это математическое выражение, которое состоит из операндов, соединенных знаками математических операций. *Операндами* могут быть числа, переменные, встроенные и определенные пользователем функции.

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. В Приложении 2 данного пособия приведен список наиболее часто используемых операторов.

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.

Для ввода формул можно использовать пиктограммы всевозможных меню из математической панели инструментов. Пиктограммы позволяют вводить знаки математической символики, греческие буквы, графические и текстовые блоки следующим образом:

- указать позицию ввода: указатель мыши переместить в нужное место экрана и щелкнуть левой кнопкой мыши, в результате это место отметится крестиком;
- выбрать нужную пиктограмму и щелкнуть на ней левой кнопкой мыши (объект, обозначенный на пиктограмме, появится на месте крестика).

Математические выражения имеют строго определенную структуру. Выражения не просто печатаются, а строятся по правилам старшинства операций и другим дополнительным правилам, которые упрощают ввод знаменателей, показателей степени, выражений в радикалах. Скобки вставляются везде, где это необходимо, чтобы указать нужный порядок вычислений. Ключевым шагом в построении и редактировании формул является заключение части выражения в выделяющуюся рамку. При этом то, что заключено в выделяющую рамку, становится операндом следующего вводимого оператора. Управлять выделяющей рамкой можно мышью, клавишами управления курсором и клавишей [Space].

При редактировании необходимо следить за правильным размещением выделяющей рамки. Если нужно вставить оператор перед существующим выражением, то его заключают в рамку и нажимают клавишу [Ins].

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 (см. Приложение 1). 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.

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 (=).

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.

Ввести математическое выражение можно в любом пустом месте документа MathCAD. Для этого поместите курсор ввода в желаемое место документа, щелкнув в нем мышью, и просто начинайте вводить формулу, нажимая клавиши на клавиатуре. При этом в документе создается *математическая область* (math region), которая предназначена для хранения формул, интерпретируемых процессором MathCAD. Продемонстрируем последовательность действий на примере ввода выражения x^{5+x}.

1. Щелкните мышью, обозначив место ввода.

2. Нажмите клавишу <**x**> - в этом месте вместо курсора ввода появятся регион с формулой, содержащей один символ **x**, причем он будет выделен линиями ввода.

3. Введите оператор возведения в степень, нажав сочетание клавиш Shift+^, либо выбрав кнопку возведения в степень на панели инструментов Calculator – в формуле появится местозаполнитель для введения значения степени, а линии ввода выделят этот местозаполнитель.

4. Последовательно введите остальные символы <**5**>, <**+**>, <**x**>.

Если же далее в конце выражения поставить знак равенства, а перед самим выражением присвоить переменной **x** значение 11, то получим вычисленное математическое выражение (рис. 3.4).

$$x := 11$$

$$x^{5+x} = 4.595 \times 10^{16}$$

Рис. 3.4 – Пример вычисления

Таким образом, поместить формулу в документ можно, просто начиная вводить символы, числа и операторы, например + или /. Во всех этих случаях на месте курсора ввода создается математическая область, иначе называемая *регионом*, с формулой, содержащей и линии ввода. В последнем случае, если пользователь начинает ввод формулы с оператора, в зависимости от его типа, автоматически появляются и местозаполнители, без заполнения которых формула не будет восприниматься процессором MathCAD.

Чтобы изменить формулу, щелкните на ней мышью, поместив таким образом в ее область линии ввода, и перейдите к месту, которое хотите исправить. Перемещайте линии ввода в пределах формулы одним из двух способов:

- щелкая в нужном месте мышью;
- нажимая на клавиатуре клавиши со стрелками, пробел и <Ins>:
- клавиши со стрелками имеют естественное назначение, переводя линии ввода вверх, вниз, влево и вправо;
- клавиша <Ins> переводит вертикальную линию ввода с одного конца горизонтальной линии ввода на противоположный;
- пробел предназначен для выделения различных частей формулы. Если раз за разом нажимать клавишу пробела в формуле, то линии ввода будут циклически изменять свое положение.

В MathCAD возможно вырезание (Edit-Cut или Ctrl+X), копирование (Edit-Copy или Ctrl+C) выделенной формулы или ее части с последующей вставкой (Edit-Paste или Ctrl+V).

И так, математические выражения содержат, как правило, самые различные, в том числе специфические символы, набор которых выполняется в MathCAD не так, как в большинстве текстовых процессоров. При этом для вставки символов доступны следующие инструменты:

- клавиатура;
- панель инструментов Greek для ввода греческих символов;
- панель инструментов Calculator для ввода операторов;
- команда Insert-Function для ввода имен функций (возможен ввод с клавиатуры);
- клавиша <'> (апостроф) для выделения скобками уже введенной части формулы, помещенной между линиями ввода.

Некоторые операторы, например оператор присваивания (:=) или знак умножения имеют множественное отображение. Для выбора нужного их отображения в документе следует использовать контекстное меню, вызываемое нажатием правой кнопки мыши после установления курсора на знаке оператора. Выбор отображения таких операторов по умолчанию осуществляется настройками вкладки Display (Отображение) диалогового окна Worksheet Options (Опции документа), воспользовавшись меню Tools-Worksheet Options.

3.1.2 Ввод и редактирование текста

Хотя MathCAD и является математическим редактором, он обладает развитыми средствами по оформлению текста. Текст в MathCAD используется в качестве комментариев к производимым вычислениям, или как элемент оформления документов для создания качественных отчетов. В MathCAD текст можно вводить, редактировать и импортировать.

Для ввода текста необходимо указать программе, что создается не формульный, а текстовый регион: для этого перед вводом первого символа текста нажмите клавишу <">. При этом курсор принимает вид вертикальной линии красного цвета, который называется *линией ввода текста*.

Для редактирования текста необходимо щелкнуть левой клавишей мыши на область текста, что приведет к выделению текстового региона, в котором размещается данный текст. Далее следует изменить текст, используя курсор и клавиатуру.

Для импортирования текста (например, из Блокнота или MS Word) необходимо:

- находясь в другом приложении, скопировать нужный фрагмент в буфер обмена;

- перейти в окно MathCAD и пометить курсором ввода место вставки;

- ввести текст.

Ввод импортируемого текста возможен двумя способами:

- в текстовом регионе (для этого создается регион ввода и нажимается клавиша <”>, после чего вставляется содержимое буфера обмена нажатием сочетания клавиш <Ctrl>+<V>);

- вставкой импортируемого текста, как объекта OLE (Object Linking and Embedding) нажатием сочетания клавиш <Ctrl>+<V> (при этом для каждого последующего редактирования текста будет вызываться приложение, в котором был создан импортированный фрагмент текста).

Для вставки математических выражений внутри текстовой области необходимо войти внутрь текстового региона и в нужное место текстового региона вставить как обычно математический регион (меню ***Insert – Math Region*** или сочетание клавиш <Ctrl>+<Shift>+<A>), в котором ввести необходимое математическое выражение. При этом надо помнить, что введенные формулы являются не комментариями, а полноправными вычислительными элементами документа, переопределяющими введенные ранее (до ввода текстового региона) значения математических переменных.

Для вставки гиперссылок на какое-либо место внутри активного документа, другого документа MathCAD или сайт в Интернет используется команда ***Insert-Hyperlink***.

3.1.3 Правка документа

При правке документа используются следующие операции:

- выделение части документа;
- удаление части документа;
- вырезка, копирование, вставка и перемещение части документа;
- выравнивание регионов;
- обновление вида и документа;
- поиск и замена;
- проверка англоязычной орфографии.

Для выделения части документа MathCAD используется стандартная процедура выделения элементов окна принятая в Windows: вне крайнего из них нажимается левая клавиша мыши (определяется курсором начальная точка выделения) и протаскивается ее указатель через все выделяемые регионы. При этом все выделяемые регионы помечаются пунктирными границами. Для снятия выделения достаточно щелкнуть мышью в любой части документа.

Для удаления части документа необходимо сначала выделить эту часть, а затем нажать клавишу <Del> или <Ctrl>+<Del>.

Текущий регион удаляется нажатием сочетания клавиш <Ctrl>+<Del> или командой ***Edit-Delete***.

Для вырезки, копирования выделенных регионов и вставки их из буфера используются меню ***Edit***, кнопки правки на панели инструментов ***Standard*** (Стандартная) или сочетания горячих клавиш:

- <Ctrl> + <X> (вырезать),
- <Ctrl>+<C> (копировать в буфер обмена),

- <Ctrl>+<V> (вставить из буфера обмена).

Для перемещения выделенных регионов в другое место можно использовать прием перетаскивания: если поместить курсор на выделенный регион, он приобретает форму «руки». Далее зафиксировав «руку» нажатой левой клавишей мыши, можно переместить регион в другое место документа.

Для выравнивая регионов документа необходимо воспользоваться командой меню ***Format-Align Regions*** с опциями ***Across*** (Горизонтально), ***Down*** (вертикально).

Для обновления вида документа (при появлении «мусора») необходимо воспользоваться командой обновления документа ***View-Refresh***.

Поиск символа, фрагмента или слова документа осуществляется командой ***Edit-Find*** или сочетанием клавиш <Ctrl>+<F>. Команда имеет окно с кнопками ***Find Next*** (Искать далее), ***Cancel*** (Закончить поиск), окно ***Find what*** (Что искать) и опциями:

- Match whole word only (совпадение слов целиком),
- Match case (учитывать регистр),
- Find in Text Regions (в текстовых областях),
- Find in Math Regions (в математических областях),
- Up (вверх),
- Down (вниз).

Замена символа, фрагмента или слова документа осуществляется командой ***Edit-Replace*** или сочетанием клавиш <Ctrl>+<H>. Команда имеет окно с кнопками ***Find Next*** (Искать далее), ***Replace*** (Заменить), ***Replace All*** (Заменить все), ***Cancel*** (закончить поиск) и опциями:

- Match whole word only (совпадение слов целиком),
- Match case (учитывать регистр),
- Find in Text Regions (в текстовых областях),
- Find in Math Regions (в математических областях).

3.1.4 Печать и посылка документа по электронной почте

Готовый документ можно распечатать на активном принтере, воспользовавшись командой ***File-Print*** (<Ctrl>+<P>). Перед этим можно установить параметры страницы по команде ***File-Print Setup*** и осуществить предварительный просмотр по команде ***File-Print Preview***.

Документ также возможно переслать по электронной почте, воспользовавшись командой ***Edit-Send***.

3.2 Вычисления

3.2.1 Числа

MathCAD выполняет вычисления над вещественными и комплексными числами. Числа можно задавать в десятичной, восьмеричной и шестнадцатеричной системах счисления. Признаком восьмеричных чисел является буква «о», шестнадцатеричных чисел – буква «h».

Для обозначения мнимой единицы в комплексных числах используются буквы «i» и «j». Пользователь может выбрать обозначение мнимой единицы по своему усмотрению. Для этого нужно использовать главное меню ***Format-Number***. Нельзя использовать для обозначения мнимой единицы буквы «i» и «j» сами по себе. Следует печатать 1i или 1j. Чтобы ввести числа в экспоненциальном представлении, нужно умножить мантиссу на степень десяти.

Пример:

769.07

-22.34578

$1.237 \cdot 10^4$

156o – восьмеричное число

4ABh – шестнадцатеричное число

47.09-0.7i – комплексное число

В вычислениях можно использовать числа, представленных в различных системах счислениях. Систему счисления числового результата можно установить командой ***Format-Number***.

Пример:

156o + 12 = 122

14h + 156o = 130

3.2.2 Переменные

При помощи переменных обозначаются скалярные величины, векторы, матрицы и функции. Имена переменных могут быть любой длины и состоять из латинских, греческих букв, цифр от 0 до 9, символа подчеркивания, символа процента, символа бесконечности (∞). Переменная может быть набрана в любом шрифте, но все символы одного имени должны быть набраны одним шрифтом. MathCAD различает имена, набранные в разных регистрах и в разных шрифтах, например, F, f и курсивное *f* обозначают различные переменные.

Имя не может начинаться с цифры, символа подчеркивания (_), символа процента (%), или символа штриха (`); символ бесконечности (∞) может быть только первым символом в имени.

Пример:

A0123 αβ_120 n% V AB Sum12_k

В именах можно использовать символ (.). Все символы, набранные после нажатия точки (.), представляют нижний индекс. Например, формулу R_a:=11+R_b следует набрать так: R.a:11+R.b. Заметим, что набранные таким образом переменные не являются элементами массивов.

Некоторые переменные в пакете MathCAD имеют предопределенные значения. Эти переменные и их значения приведены в таблице 3.1.

Таблица 3.1 – Переменные с предопределенными значениями

Имена предопределенных переменных	Значение	Пояснения
π	3.141…	Число π используется с учетом 15 значащих цифр. В символьных вычислениях сохраняется точное значение. Для ввода можно использовать комбинацию <Ctrl>+<P>.
e	2.718…	Основание натуральных логарифмов используется с учетом 15 значащих цифр. В символьных вычислениях сохраняется точное значение.
∞		Бесконечность. В численных расчетах – число 10^{307}. В символьных вычислениях – бесконечность. Для ввода ∞ можно использовать комбинацию <Ctrl>+<z>.
%	0.01	Процент. В выражениях, подобных 10*%, или как масштабный множитель в поле, отводимом для единиц размерности.
TOL	0.001	Допускаемая погрешность для различных приближенных алгоритмов и аппроксимаций.
ORIGIN	0	Начало массива. Определяет индекс первого элемента массива.
PRNCOLWIDTH	8	Ширина столбца, используемая при записи файлов функцией WRITEPRN.
PRNPRECISION	4	Число значащих цифр, используемых при записи файлов функцией WRITEPRN.
FRAME	0	Используется в качестве счетчика при управлении анимациями.

3.2.3 Операции

Вычисления в MathCAD проводятся через вызов оператора или вызов функции. Операторы вводятся нажатием кнопок с их изображением или комбинацией клавиш. Ниже в таблице 3.2 приведены основные операторы и способ их задания.

Таблица 3.2 – Основные операторы и способ их задания

Обозначение	Клавиши	Пояснения
X+Y	+	Сложение. Сложение, если X и Y – скалярные выражения. Сложение элементов массивов, если X и Y – массивы (векторы или матрицы) одинаковой размерности.

		Если X-массив, а Y-скаляр, то сложение каждого элемента массива X с Y.
X-Y	-	Вычитание. Вычитание, если X и Y – скалярные выражения. Вычитание элементов массивов, если X и Y – массивы (векторы и матрицы) одинаковой размерности. Если X-массив, а Y-скаляр, то вычитание из каждого элемента массива X скаляра Y.
$X \cdot Y$	*	Умножение. Умножение, если X и Y – скалярные выражения. Скалярное произведение векторов, если X и Y – вектора. Произведение матриц, если X и Y – матрицы соответствующих размерностей. Если X-массив, а Y-скаляр, то умножение каждого элемента массива X на скаляр Y.
$\frac{X}{Y}$	/	Деление. Если X-скаляр и Y-ненулевой скаляр, то деление X и Y. Если X-массив и Y-ненулевой скаляр, то деление каждого элемента массива X и Y.
Z^w	^	Возведение в степень. Возводит Z в степень w, Z и w могут быть вещественными и комплексными.
$\sqrt{Z}$	\	Квадратный корень. При неотрицательном Z возвращает положительное значение квадратного корня; для отрицательных и комплексных – главное значение $\sqrt{Z}$.
$\sqrt[n]{Z}$	Ctrl+\	Корень n-й степени Для натуральных n возвращает значение корня $\sqrt[n]{Z}$
$\lvert Z \rvert$	|	Абсолютное значение Для комплексного Z вычисляется $\sqrt{Re(Z)^2 + Im(Z)^2}$
n!	!	Факториал Для целого $n \geq 0$ вычисляется $n \cdot (n-1) \cdot \ldots \cdot 2 \cdot 1$
$\frac{d}{dx} f(x)$ $\frac{d^n}{dx^n} f(x)$	? Ctrl+?	Производные Операторы применяются для вычисления производной от функции $f(x)$ в точке. В этом случае нужно задать значение аргумента и, кроме того, все переменные в $f(x)$, должны быть определены. Значение n должно быть целым от 0

		до 5. Эти операторы применяются и для символьного дифференцирования, здесь n может быть целым положительным.
$\int_a^b f(x)dx$	&	Определенный интеграл Численное интегрирование, вычисляется значение определенного интеграла. Пределы интегрирования a и b и все переменные в функции $f(x)$ должны быть определены.

Mathcad допускает использование следующих операций сравнения. Результаты представлены в таблице 3.3.

Таблица 3.3 – Операции сравнения

Обозначение	Клавиши	Пояснения
x<y	<	Результат сравнения равен 1, если x<y, в других случаях – 0.
x>y	>	Результат сравнения равен 1, если x>y, в других случаях – 0.
x≤y	Ctrl+(	Результат сравнения равен 1, если x≤y, в других случаях – 0.
x≥y	Ctrl+)	Результат сравнения равен 1, если x≥y, в других случаях – 0.
z≠w	Ctrl+#	Результат сравнения равен 1, если z≠w, в других случаях – 0.
z=w	Ctrl+=	Результат сравнения равен 1, если z=w, в других случаях – 0.

Для операций больше, меньше, больше или равно и меньше или равно операндами могут быть только действительные числа. В случае операций равно и неравно операндами могут быть вещественные, комплексные числа и массивы. Результат операции сравнения равен 1, если операнды таковы, что для них выполняется операция сравнения, и равен 0 в противном случае.

3.2.3 Операторы присваивания

Для вычисления значений переменной применяются операторы присваивания, имеющие вид:

<имя переменной> <операция присваивания> <выражение>.

Mathcad имеет в своем распоряжении две операции присваивания: (:=) и (≡). Операция присваивания (:=) появляется, если ввести символ двоеточие (:), а вторая после ввода символа «тильда» (~). Кроме того, что для ввода операций присваивания можно применять и пиктограммы из математических панелей инструментов.

Итак, Mathcad имеет два вида оператора присваивания. Разница между ними состоит в следующем. Mathcad все вычисления выполняет за два просмотра. На каждом просмотре документ просматривается слева направо и сверху вниз. Во время первого просмотра Mathcad выполняет вычисления, заданные операторам присваивания (≡), а при втором просмотре – вычисления, определенные операторами со знаком (:=). Таким образом, вычисления, заданные оператором (≡), являются глобальными по отношению к остальным. Это значит, что значения переменных, вычисленных таким оператором, имеют силу на всем документе независимо от его расположения. А значения переменных, вычисленных оператором (:=), распространяются только влево и вниз по документу.

Пример:

$$x := 3.175$$

$$z := 8.512 + 5.26i$$

$$J := \int_0^{10} x^2 \cdot \cos(x)\, dx$$

$$R \equiv \sum_{n=1}^{12} \frac{1}{3 \cdot n + 4}$$

$$A := \begin{pmatrix} 1.1 & 2.2 & 3.3 \\ 4.1 & 5.2 & 6.3 \\ 7.1 & 8.2 & 9.3 \end{pmatrix}$$

3.2.4 Функции

Встроенные функции Mathcad – это основной набор функций, который поставляется вместе с Mathcad. Просмотреть список всех встроенных функций Mathcad можно командой ***Insert-Insert Function*** или использовать сочетание клавиш <Ctrl+E>. Список всех встроенных функций представлен на рисунке 3.5.

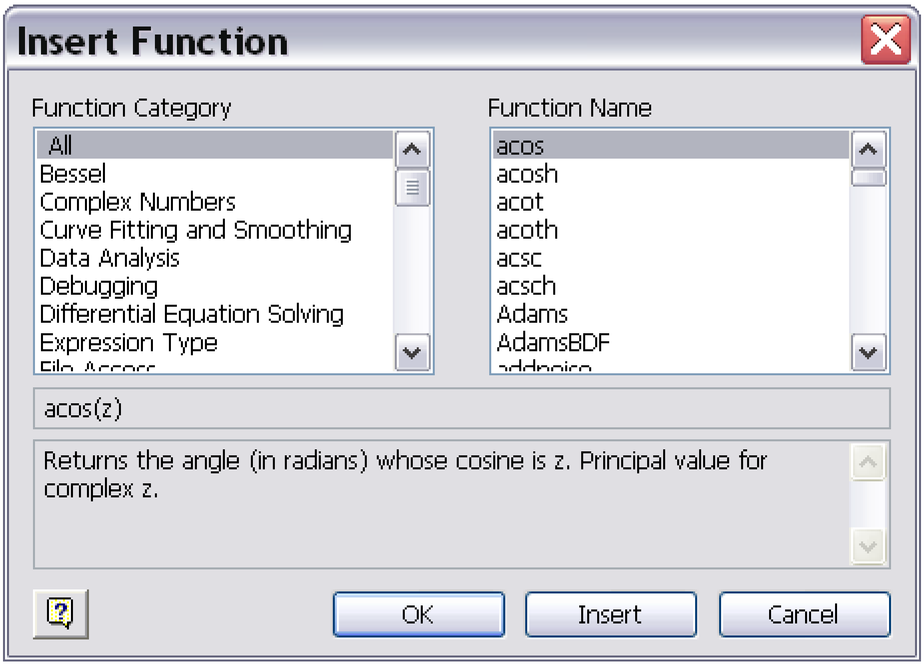


Рис. 3.5 – Список всех встроенных функций

В диалогом окне этой команды выводится имя функции с аргументами, здесь же дано краткое описание выбранной функции. Имена встроенных функций могут вводиться обычным набором на клавиатуре. Имена встроенных функций не чувствительны к шрифту, но чувствительны к регистру – их следует печатать в точности, как они приведены. Поэтому рекомендуется для ввода функций использовать «мастер функций» из меню ***Insert-Insert Function***. Набор встроенных функций включает тригонометрические, гиперболические, логарифмические, экспоненциальную и другие функции, которые записываются в привычном виде. Ниже приведены функции, которые служат для извлечения части своего аргумента, которые приведены в таблице 3.4.

Таблица 3.4 – Функции, которые служат для извлечения части своего аргумента

Функция	Назначения
Re(z)	Вещественная часть комплексного числа z.
Im(z)	Мнимая часть комплексного числа z.
arg(z)	Аргумент комплексного числа z.
floor(x)	Наибольшее целое число $\le$x (x – вещественное число).
ceil(x)	Наименьшее целое число $\ge$x (x – вещественное число).
mod(x,y)	Остаток от деления x на y. Аргументы – вещественные числа.
angle(x,y)	Угол в радианах между положительной полуосью x и вектором (x,y) в плоскости x-y. Аргументы – вещественные числа, результат между 0 до 2π.

Встроенные функции можно предопределить.

3.2.5 Функции пользователя

Определение функции имеет вид:

<имя функции> (<список аргументов>) <оператор присваивания> <выражение>

Аргументами функции могут быть скаляры, векторы, матрицы и функции.

Пример:

Определение

$$f(x,y) := \sqrt{x^2 + y^2}$$

$$g(x,y) := x \cdot y$$

$$I(f,a,b) := \int_a^b f(x)\,dx$$

Применение

$$f(4,3) = 5$$

$$g(2.25, 4.11 \cdot e - 5) = 13.887$$

$$g\left(2 - 8i, \sqrt{8 - 4i}\right) = 0.324 - 24.66i$$

$$g\left[\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}, \begin{pmatrix} 0 & 1 - 5i \\ 5 - 4i & 2 \end{pmatrix}\right] = \begin{pmatrix} 10 - 8i & 5 - 5i \\ 20 - 16i & 11 - 15i \end{pmatrix}$$

$$F(x) := \frac{1}{\sqrt{1 + x + x^2 + x^3}}$$

$$I(F,3,5) = 0.224$$

3.2.6 Простейшие вычисления

Для получения численного результата необходимо набрать на экране формулу справа от нее знак (=) или выполнить команду меню ***Math-Calculate***. Mathcad вычислит результат и напечатает его справа от знака равенства.

Пример:

$$\sqrt[9]{456.3} = 1.975$$

$$(2.01 + 5.1i)^{-1} = 0.067 - 0.17i$$

$$4^7 - \frac{8774.1}{\left(2 \cdot \pi - \sqrt{2 - \sin(0.5)^2}\right)} = 1.461 \times 10^4$$

$$\int_0^{\pi} \frac{\sin(x + 0.954 \cdot \pi)}{\cos(x - \pi)}\,dx = -0.5$$

$$\sum_{n=1}^{12} n! = 5.23 \times 10^{8}$$

$$\prod_{x=0}^{2} (\sin(x) + \cos(x)) = 0.681$$

3.2.7 Задание 1 для самостоятельной проработки материала

Выполнить вычисления по следующим формулам:

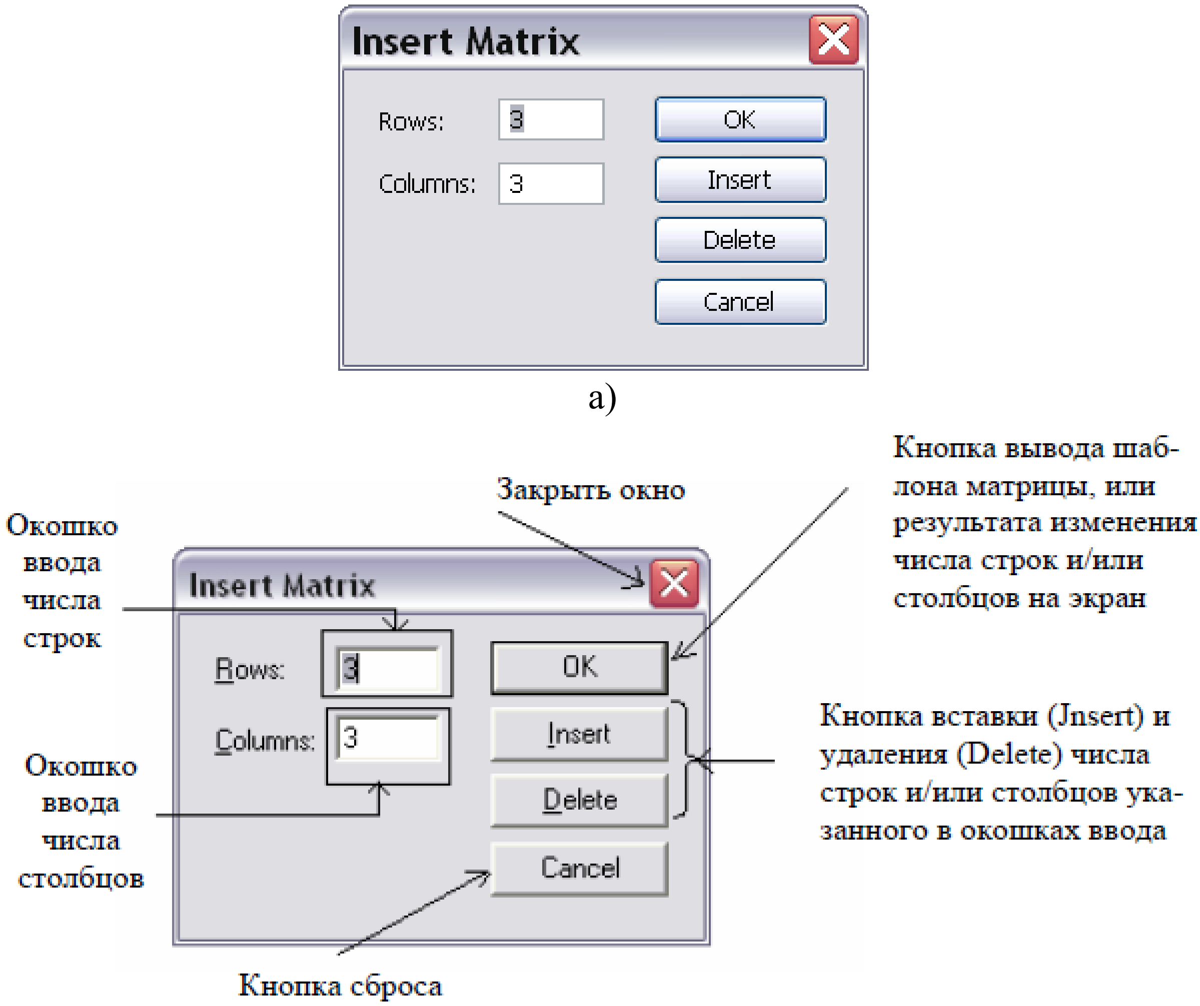
1. $\sqrt[4]{\frac{12,34+3,47i}{|\sin(0,365)-\cos(0,5)|}} + \ln(14,85)$
2. $\sum_{k=1}^{12} k \cdot \cos(k \cdot \frac{\pi}{6})$
3. $\sum_{k=0}^{11} \frac{\sin(\frac{7}{4}+\frac{k}{11})}{k!\cdot(5+k)!}$
4. $\prod_{n=1}^{6} n^3 + \sin(n \cdot \frac{\pi}{1,14})$
5. $\coprod_{n=1}^{12}(1 - \frac{0,72}{\sqrt{n}}) \cdot e^{\frac{0,72}{\sqrt{n}}}$
6. $\sum_{n=1}^{20} \frac{\prod_{n=1}^{20} \ln(n+1)}{\ln(n+1+0,3)}$
7. $\int_{2,45}^{9,68} \frac{1+2,43\cdot x+4,71\cdot x^2}{\sqrt[5]{1+x}} dx$
8. $y = \frac{x\cdot(1+G)}{(1+G)-\frac{23,76}{x}x^3}$, где $G = \sum_{n=1}^{3} \frac{u^k}{(2k-1)\cdot(2k+1)}, u = \frac{\pi}{5,2}$
9. $\sin^{-1} x \frac{2,34}{\cos^{-1} 2}$
10. $\iint_{-10}^{10} \sin(x) \cdot \cos(y) dxdy$

3.3 Векторы и матрицы

Mathcad выполняет операции над массивами: векторами и матрицами. Создать вектор или матрицу можно тремя способами:

1. Ввод массивов с клавиатуры. Для этого необходимо:

- вызвать диалоговое окно Insert-Matrix (рис. 3.6а,б): это можно сделать либо командой Insert-Matrix либо клавишами <Ctrl>+<M> либо с помощью кнопки на панели математических инструментов;



б)

Рис. 3.6 а,б – Диалоговое окно для ввода массива с клавиатуры

- задать размерность в диалоговом окне: количество строк (rows), столбцов (columns);
- в появившемся макете заполнить пустые позиции (рис. 3.7) числами или формулами, при этом для перехода к другой позиции можно использовать клавишу Tab или клавиши со стрелками.

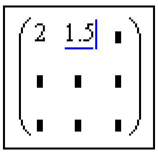


Рис. 3.7 – Пример заполнения массива элементами

Таким способом можно ввести массив, общее число элементов которого не больше 100.

Это же окно используется и для изменения размеров ранее созданного массива: можно вставлять или удалять строки и столбы правее и ниже отмеченного курсором элемента массива. Для изменения размеров массива необходимо:

- отметить курсором элемент массива;

- в полях диалогового окна Insert-Matrix нужно указать количество вставляемых (удаляемых) столбцов и строк;

- нажать на кнопку Insert или Delete.

2. Вектор или матрицу, размеры которых ограничивается лишь доступной памятью компьютера, можно создать, непосредственно вычисляя элементы матрицы по некоторой явной формуле;

3. Для создания массива достаточно также ввести его элементы из файла. При этом нет необходимости делать предварительное описание массива.

Для обращения к элементам массива используются нижние индексы. Чтобы напечатать нижний индекс используют кнопку Xn на панели математических инструментов или клавишу «[». Чтобы ввести оператор верхнего индекса, нужно щелкнуть по кнопке M<> или нажать <Ctrl>+<6> и поместить в поле целое число.

Для присвоения значений элементами массивов применяется дискретный аргумент, т.е. ряд значений, отделяемых одинаковыми шагами. Дискретный аргумент определяется как:

<переменная>:=<первое значение>,<второе значение>..<последнее значение>

Здесь многоточие (..) появляется, если нажать клавишу точка с запятой [;]. Если переменная принимает целочисленное значение с шагом единица, то ***<второе значение>*** можно не писать.

Пример:

1.

$n := 0..5$

$x_n := \cos(10 \cdot n) + n^2$

$x_n =$

1
0.161
4.408
9.154
15.333
25.965

2.

$i := 0..3 \qquad j := 0..3$

$$A_{(i,j)} := \frac{i \cdot (j+7)}{(i+j+1)}$$

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 3.5 & 2.667 & 2.25 & 2 \\ 4.667 & 4 & 3.6 & 3.333 \\ 5.25 & 4.8 & 4.5 & 4.286 \end{pmatrix}$$

Обращение к элементам матрицы

$A_{0,0} = 0 \qquad A_{1,0} = 3.5$

$A_{1,1} = 2.667 \qquad A_{1,2} = 2.25$

Обращение к столбцам матрицы

$$A^{\langle 1 \rangle} = \begin{pmatrix} 0 \\ 2.667 \\ 4 \\ 4.8 \end{pmatrix} \quad A^{\langle 2 \rangle} = \begin{pmatrix} 0 \\ 2.25 \\ 3.6 \\ 4.5 \end{pmatrix}$$

Обращение к строкам матрицы

$$\left(A^T\right)^{\langle 1 \rangle} = \begin{pmatrix} 3.5 \\ 2.667 \\ 2.25 \\ 2 \end{pmatrix} \qquad \left(A^T\right)^{\langle 2 \rangle} = \begin{pmatrix} 4.667 \\ 4 \\ 3.6 \\ 3.333 \end{pmatrix}$$

Нижние границы индексов элементов векторов и матриц задает переменная ORIGIN. По умолчанию ее значение равно нулю. Другие начальные значения, в том числе и отрицательные, можно установить, заменив значение ORIGIN через меню ***Math-Options-Built In Variable*** или при помощи одного глобального оператора присваивания. Значение переменной ORIGIN, установленное этими способами применяется ко всем массивам. Невозможно сделать для разных массивов разные начальные значения индексов.

Возникающие в результате вычислений массивы могут оказаться громоздкими для отображения на экране. Поэтому Mathcad массивы, имеющие большое количество элементов, отображает в виде таблицы с полосами прокрутки. Увидеть элемент можно, используя полосы прокрутки или увеличив границы таблицы с помощью мыши.

Пример:

$$ORIGIN := 1$$

$$M := \begin{pmatrix} 1.23 & 2.23 & 3.33 \\ 5.12 & 6.12 & 7.12 \\ 1.23 + 5.12i & 2.23 + 6.12i & 3.33 + 7.12i \end{pmatrix}$$

$M_{1,1} = 1.23$ $M_{1,2} = 2.23$ $M_{1,3} = 3.33$

$M_{2,1} = 5.12$ $M_{2,2} = 6.12$ $M_{2,3} = 7.12$

$M_{3,1} = 1.23 + 5.12i$ $M_{3,3} = 3.33 + 7.12i$

$M_{3,2} = 2.23 + 6.12i$

$$M^{\langle 1\rangle} = \begin{pmatrix} 1.23 \\ 5.12 \\ 1.23 + 5.12i \end{pmatrix} \qquad \left(M^T\right)^{\langle 1\rangle} = \begin{pmatrix} 1.23 \\ 2.23 \\ 3.33 \end{pmatrix}$$

$$M^{\langle 2\rangle} = \begin{pmatrix} 2.23 \\ 6.12 \\ 2.23 + 6.12i \end{pmatrix} \qquad \left(M^T\right)^{\langle 2\rangle} = \begin{pmatrix} 5.12 \\ 6.12 \\ 7.12 \end{pmatrix}$$

$$M^{\langle 3\rangle} = \begin{pmatrix} 3.33 \\ 7.12 \\ 3.33 + 7.12i \end{pmatrix} \qquad \left(M^T\right)^{\langle 3\rangle} = \begin{pmatrix} 1.23 + 5.12i \\ 2.23 + 6.12i \\ 3.33 + 7.12i \end{pmatrix}$$

3.3.1 Векторные и матричные операции

Некоторые из операторов Mathcad имеют особые значения в применении к векторам и матрицам. Например, символ умножения «·» при применении к векторам означает скалярное умножении и умножение матриц – когда применяется к матрицам.

Векторные и матричные операторы доступны из палитры символов или математической панели. Если результатом операции является вектор, то это обязательно вектор-столбец, а не вектор-строка.

Выше, при рассмотрении операторов, обращалось внимание на их особенности при работе с массивами. В следующей таблице 3.5 приведены операторы, имеющие место для массивов и дискретных переменных.

Таблица 3.5 – Операторы, имеющие место для массивов и дискретных переменных

Обозначение	Клавиши	Пояснения
$X \times Y$	Ctrl+*	Векторное произведение трехмерных векторов X и Y
A^n	^	Степень матрицы

		Для квадратной матрицы A и целого положительного n вычисляется n-я степень обратной матрицы A.
$\|v\|$	|	Длина вектора Для вектора v с вещественными элементами вычисляется длина вектора $\sqrt{v \cdot v}$. Для вектора с комплексными элементами вычисляется $\sqrt{v \cdot \bar{v}}$.
$\|A\|$	|	Определитель матрицы Для квадратной матрицы A вычисляется определитель.
$\sum_{a} X$	$	Суммирование по переменной Для любого выражения X вычисляется сумма по дискретным значениям переменной a.
$\sum v$	Ctrl+$	Суммирование элементов одномерного массива Вычисляет сумму элементов массива v.
$\sum_{i=n}^{m} X$	Ctrl+Shift+#	Суммирование Вычисляет сумму значений X для i=n,n+1,..,m,где X – любое выражение, n,m – целые.
$\prod_{a} X$	#	Произведение Для любого выражения X вычисляется произведение по дискретным значениям переменной a.
$\prod_{i=n}^{m} X$	Ctrl+Shift+#	Произведение Вычисляет произведение значений X для i=n,n+1,..,m,где X – любое выражение, n,m – целые.

Пример:

Транспонирование матрицы

$$\begin{pmatrix} 2.01 & 1.2 & 3.4 & 7.01 & -52 \\ 41 & -2.3 & 1.25 & 4.28 & 7.2 \\ 7 & -8.1-4.2 & 12 & 14.2 & 14.8 \\ 0.67 & 9.7 & -5.8 & 4.13 & 0.2 \\ 0 & 4 & 1 & 2 & 3 \end{pmatrix}^{T} = \begin{pmatrix} 2.01 & 41 & 7 & 0.67 & 0 \\ 1.2 & -2.3 & -12.3 & 9.7 & 4 \\ 3.4 & 1.25 & 12 & -5.8 & 1 \\ 7.01 & 4.28 & 14.2 & 4.13 & 2 \\ -52 & 7.2 & 14.8 & 0.2 & 3 \end{pmatrix}$$

Определитель матрицы и сумма элементов первого столбца матрицы

$$R := \begin{pmatrix} -7 & 9 & -9.7 \\ 4 & 2 & 0 \\ 19 & 67 & 4.25 \end{pmatrix}$$

$$|R| = -2.443 \times 10^3 \qquad \sum R^{\langle 1 \rangle} = 78$$

Векторные и матричные функции представлены в таблице 3.6.

Таблица 3.6 – Векторные и матричные функции

Имя функции	Результат
rows(A)	Число строк в массиве.
cols(A)	Число столбцов в массиве.
length(v)	Число элементов в векторе.
last(v)	Индекс последнего элемента вектора.
max(A)	Максимальный элемент массива.
min(A)	Минимальный элемент массива.
identity(n)	Единичная матрица размера $n \times n$.
rank(A)	Ранг вещественной матрицы.
augment(A,B)	Массив, сформированный расположением A над B бок о бок. Массивы A и B должны иметь одинаковое количество столбцов.
stak(A,B)	Массив, сформированный расположением A над B. Массивы A и B должны иметь одинаковое количество столбцов.
submatrix(A, ir, jr, ic, jc)	Подматрица, состоящая из всех элементов, содержащихся в строках ir≤jr и ic≤jc столбцах.
csort(A,n)	Переставляет строки матрицы A по возрастанию элементов n-го столбца.
rsort(A,n)	Переставляет столбцы матрицы A по возрастанию элементов n-ой строки.
reverse(v) reverse(A)	Обращает порядок элементов вектора или матрицы.

Пример:

Переставить столбцы матрицы D= $\begin{bmatrix} 1 & 5 & 3 \\ -1 & -3 & 8 \\ 32 & 8 & 7 \end{bmatrix}$ по элементам первой строки

$$D := \begin{pmatrix} 1 & 5 & 3 \\ -1 & -3 & 8 \\ 32 & 8 & 7 \end{pmatrix}$$

$$rsort(D,0) = \begin{pmatrix} 1 & 3 & 5 \\ -1 & 8 & -3 \\ 32 & 7 & 8 \end{pmatrix} \quad csort(D,1) = \begin{pmatrix} -1 & -3 & 8 \\ 1 & 5 & 3 \\ 32 & 8 & 7 \end{pmatrix}$$

$$max(D) = 32 \qquad min(D) = -3$$

В профессиональной версии Mathcad включен ряд дополнительных матричных функций. Они представлены в таблице 3.7.

Таблица 3.7 – Дополнительные матричные функции

Имя функции	Результат
eigenvals (M)	Возвращает вектор, содержащий собственные значения матрицы M
eisenvec (M,Z)	Для указанной матрицы M и заданного собственного значения Z возвращает принадлежащий этому собственному значению вектор
eigenvecs (M)	Возвращает матрицу, столбцами которой являются собственные векторы матрицы M (порядок расположения собственных векторов соответствует порядку собственных значений, возвращаемых функцией eigenvals)
genvals (M,N)	Возвращает вектор обобщенных собственных значений v, соответствующий решению уравнения M·x=vi-N-x (матрицы M и N должны быть вещественными)
+ lu (M)	Выполняет треугольное разложение матрицы M: P·M=L ·U, L и U – соответственно нижняя и верхняя треугольные матрицы. Все четыре матрицы квадратные, одного порядка
+ qr (A)	Дает разложение матрицы A, A=Q·R, где Q – ортогональная матрица и R – верхняя треугольная матрица
+ svd (A)	Дает сингулярное разложение матрицы A размером n·m: $A=U\cdot S\ \cdot V^T$ где и – ортогональные матрицы размером m·m и n·n соответственно, S – диагональная матрица, на диагонали которой расположены сингулярные числа матрицы A
+ svds (A)	Возвращает вектор, содержащий сингулярные числа матрицы A размером m·n, где m≥ n
Egeninv (A)	Возвращает матрицу левую обратную к матрице A. L·A=E, где E – единичная матрица размером n·n, L – прямоугольная матрица размером n·m, A – прямоугольная матрица размером m·n

3.3.2 Добавление и удаление столбцов и строк

Пусть, например, нужно удалить третий столбец матрицы A, в результате чего должна образоваться матрица размером 3×3. Для этого необходимо установить синий указатель на третьем элементе первой строки матрица А так, чтобы угол указателя был справа и вызвать окно «Матрицы» (<Ctrl>+<M>), далее указать число удаляемых строк (в нашем случае равно 0) и столбцов (равно 1). После этого необходимо нажать клавишу ***Delete*** окна «Матрицы». Должно получиться так, как показано на рисунке 3.7.

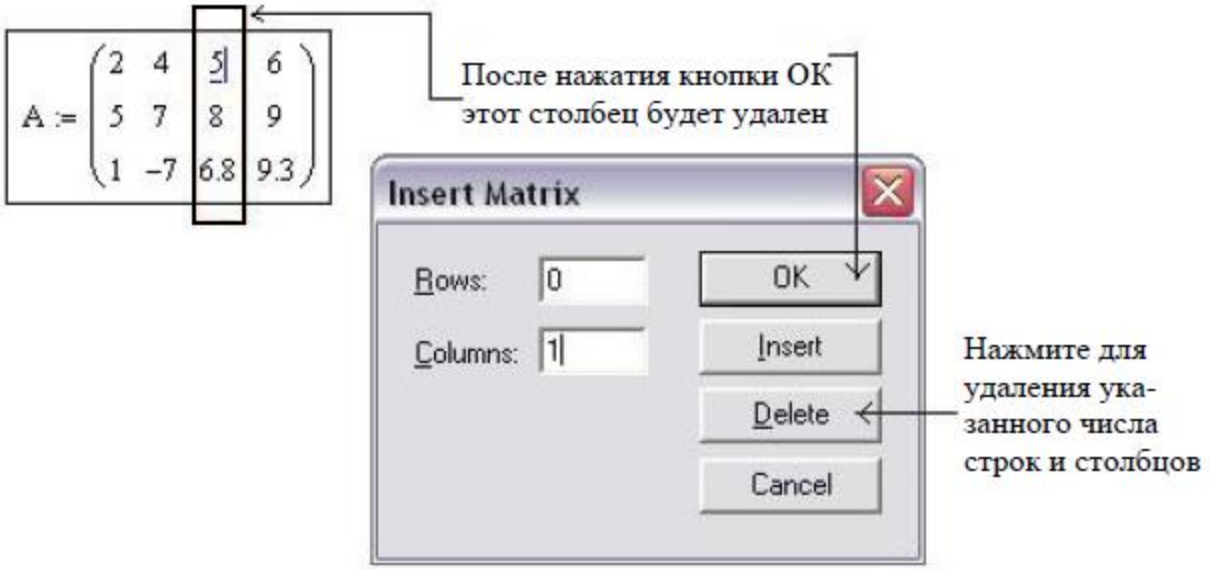


Рис. 3.7 – Пример удаления строк и столбцов матрицы

Для добавления столбцов и строк матрицы необходимо воспользоваться кнопкой ***Insert***, показанной на рисунке 3.8.

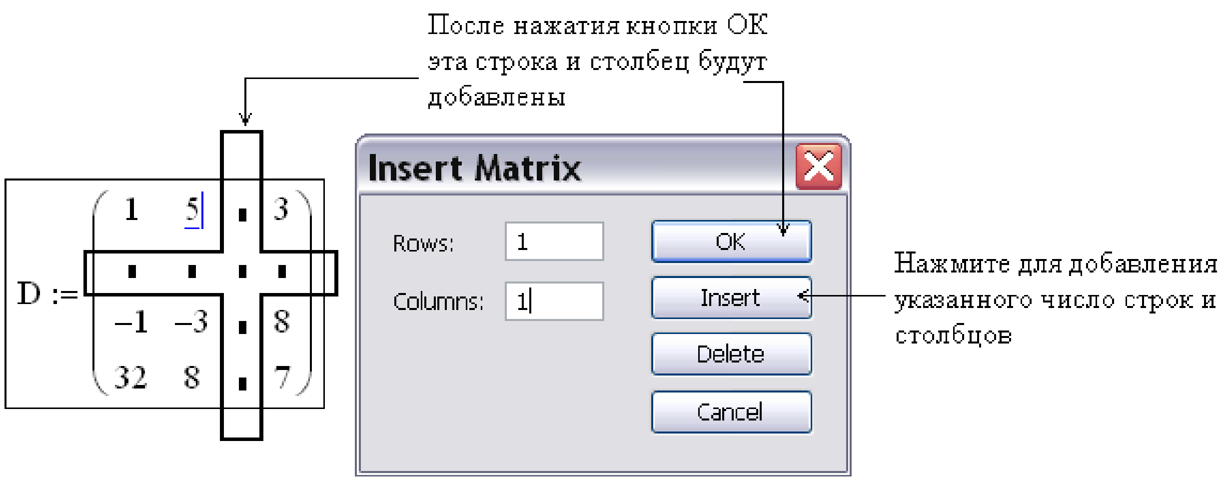


Рис. 3.8 – Пример добавления строк и столбцов матрицы

3.3.3 Операции над матрицами в аналитической (символьной) форме

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, разумеется, если такое решение существует в принципе (что бывает далеко не всегда).

Ядро символьного процессора Mathcad – несколько упрощенный вариант ядра известной системы символьной математики Maple V фирмы Waterloo Maple Software , у которой MathSoft (разработчик Mathcad) приобрела лицензию на его применение, благодаря чему Mathcad стал системой символьной математики.

Введение в систему Mathcad символьных вычислений придает ей качественно новые возможности. Символьные вычисления выполняются, в конечном счете, столь же просто для пользователя, как, скажем, вычисление квадрата x.

Операции, относящиеся к работе символьного процессора, содержатся в подменю позиции **Symbolic** (Символика) главного меню.

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, ведь и так ясно, что если маркер ввода выделяет переменную какого-либо выражения, то это выражение уже отмечено наличием в нем выделяемой переменной

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

Символьный процессор системы Mathcad обеспечивает проведение в символьном виде трех наиболее распространенных матричных операций транспонирования и обращения матриц, а также вычисления их детерминанта.

При символьных вычислениях, прежде всего, следует вызвать панель символьных вычислений нажатием кнопки  на математической панели.

После этого появится панель символьных вычислений, показанная на рисунке 3.9.

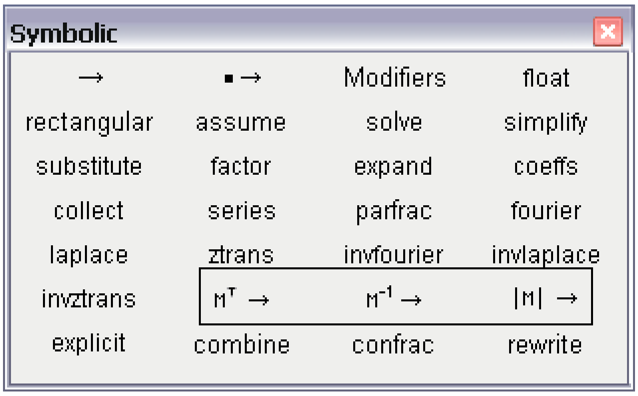


Рис. 3.8 – Панель символьных вычислений

Для символьных операций над матрицами нам понадобится только выделенная часть этого окна, с помощью кнопок которой и производятся транспонирование, обращение матрицы и нахождение ее определителя.

Пример:

$$A := \begin{pmatrix} 1 & 2 & 3 \\ 5 & 2 & 9 \\ 1 & 8 & 4 \end{pmatrix} \qquad A^T \rightarrow \begin{pmatrix} 1 & 5 & 1 \\ 2 & 2 & 8 \\ 3 & 9 & 4 \end{pmatrix} \qquad A^{-1} \rightarrow \begin{pmatrix} -\frac{16}{7} & \frac{4}{7} & \frac{3}{7} \\ -\frac{11}{28} & \frac{1}{28} & \frac{3}{14} \\ \frac{19}{14} & -\frac{3}{14} & -\frac{2}{7} \end{pmatrix}$$

$$|A| \rightarrow 28$$

3.3.4 Основные аксиомы и теоремы теории матриц

Основные аксиомы и теоремы теории матриц представлены в таблице 3.8.

Таблица 3.8 – Основные аксиомы и теоремы теории матриц

$(A \cdot B)^T = B^T \cdot A^T$	A, B – прямоугольные матрицы согласованных размерностей. Для ввода знака транспонирования – клавиши <Ctrl>+<1>.
$(A \cdot B)^{-1} = B^{-1} \cdot A^{-1}$	A, B – квадратные матрицы, определители которых не равны 0.
A+(B+C)=(A+B)+C=A+B+C	A, B, C – прямоугольные матрицы согласованных размерностей.
A·(B+C)=A·B+A·C	A, B, C – прямоугольные матрицы согласованных размерностей.
A·(B·C)=(A·B)·C=A·B·C	A, B, C – прямоугольные матрицы согласованных размерностей.
$A^T \cdot A$ – симметричная матрица	A – прямоугольная матрица произвольной размерности.
$a^T \cdot b = b^T \cdot a$ – скалярное произведение переместительно	a, b – векторы одинаковой размерности.

3.3.5 Применение матриц и векторов для решения систем линейных уравнений

Традиционно решение систем линейных уравнений рассматривается в терминах матричной алгебры. Допустим, задана система из n линейных уравнений. Коэффициенты этих уравнений – не обязательно постоянные, они могут зависеть от какого-либо внешнего параметра (на практике чаще всего – от времени), но не должны зависеть от искомых переменных (в этом случае уравнения были бы нелинейными).

Система уравнений, записанная в обычной форме, имеет вид:

$$a_{11}(t)\cdot x_1 + a_{12}(t)\cdot x_2 + \cdots a_{1n}(t)\cdot x_n = b_1(t)$$
$$a_{21}(t)\cdot x_1 + a_{22}(t)\cdot x_2 + \cdots a_{2n}(t)\cdot x_n = b_2(t)$$
$$\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots$$
$$a_{n1}(t)\cdot x_1 + a_{n2}(t)\cdot x_2 + \cdots a_{nn}(t)\cdot x_n = b_n(t)$$

Здесь t – некоторый параметр, от которого зависят коэффициенты уравнения.

Для решения системы линейных уравнений формируется квадратная n×n матрица из коэффициентов левой части и n-мерный вектор из значений $b_i(t)$, I =1,…,n, входящих в правую часть. Эти матрицы будут иметь вид:

$$A(t) = \begin{vmatrix} a_{11}(t) & a_{12}(t)\dots & a_{1n}(t) \\ a_{21}(t) & a_{22}(t)\dots & a_{2n}(t) \\ a_{n1}(t) & a_{n2}(t) & a_{nn}(t) \end{vmatrix}, B(t) = \begin{vmatrix} b_1(t) \\ b_2(t) \\ b_3(t) \end{vmatrix}$$

Согласно правилам матричной алгебры решение системы линейных уравнений для случая, когда правая часть – не нулевой вектор, возможно тогда и только тогда, когда определитель матрицы не равен 0 ни для одного значения параметра t из диапазона, на котором он задан. В этом случае решение будет n-мерным вектором, зависящим от параметра t:

$$x(t) = [A(t)]^{-1}\cdot B(t).$$

Как видим, для получения решения используются следующие матричные операции:

- обращение квадратной матрицы;
- умножение матрицы на вектор;
- диагностика отличия определителя матрицы от 0 (для установления существования решения).

Пример:

Найдем значения двумерного вектора x(t) при изменении параметра t в диапазоне 0,…,5 с шагом 0,01. Пусть каждый элемент матрицы A(t) – экспонента с заданным показателем степени, а каждый элемент вектора B(t) – синусоидальная функция.

Задаем матрицу и вектор коэффициентов.

$$A = \begin{bmatrix} 1 & 1{,}5 \\ 2 & 4 \end{bmatrix}, B = \begin{bmatrix} 5 \\ 7 \end{bmatrix}$$

Далее проводим векторизацию матрицы А. Каждый элемент ее превратится в экспоненту. Показателем будет соответствующий элемент матрицы А.

$$A(t) = \overrightarrow{e^{A \cdot t}}$$

Термин «векторизация» означает применение одной и той же функции тотально ко всем элементам матрицы. Этот прием часто экономит время при вводе, и при вычислениях за счет параллельных действий над всеми элементами матрицы сразу. Для того чтобы над всеми элементами матрицы выполнялась одна и та же операция, нужно:

- ввести матрицу с постоянными элементами, которые будут использоваться как коэффициенты в формулах для операций с матрицей;

- ввести идентификатор матрицы с указанием ее зависимости от параметра (можно – тот же, что и для матрицы коэффициентов), присвоить ему значение функции, которую мы хотим применить ко всем элементам матрицы;

- результат охватить синим контуром, и нажать на клавиши <Ctrl>+<-(минус)>. Над функцией появится горизонтальная стрелка – символ векторизации.

Для контроля проведем вычисление экспонент непосредственно (должен быть получен тот же результат, что и при векторизации):

$$A1(t) = \begin{bmatrix} e^{1 \cdot t} & e^{1,5 \cdot t} \\ e^{2 \cdot t} & e^{4 \cdot t} \end{bmatrix}$$

Проводим векторизацию вектора В. Каждый его элемент должен превратиться в синус с сомножителем-элементом вектора В:

$$B(t) = \overrightarrow{\sin(B \cdot t)}$$

Для контроля вводим функцию синуса непосредственно:

$$B1(t) = \begin{bmatrix} \sin(5 \cdot t) \\ \sin(7 \cdot t) \end{bmatrix}$$

Проверяем, действительно ли выполнилась векторизация, например, при t=0,5.

$$A := \begin{pmatrix} 1 & 1.5 \\ 2 & 4 \end{pmatrix} \quad B := \begin{pmatrix} 5 \\ 7 \end{pmatrix}$$

$$A(t) := \overrightarrow{\left(e^{A\cdot t}\right)}$$

$$A1(t) := \begin{pmatrix} e^{1\cdot t} & e^{1.5\cdot t} \\ e^{2\cdot t} & e^{4\cdot t} \end{pmatrix}$$

$$B(t) := \overrightarrow{\sin(B\cdot t)}$$

$$B1(t) := \begin{pmatrix} \sin(5\cdot t) \\ \sin(7\cdot t) \end{pmatrix}$$

$$t := 0.5$$

$$A(t) = \begin{pmatrix} 1.649 & 2.117 \\ 2.718 & 7.389 \end{pmatrix} \qquad A1(t) = \begin{pmatrix} 1.649 & 2.117 \\ 2.718 & 7.389 \end{pmatrix}$$

$$B(t) = \begin{pmatrix} 0.598 \\ -0.351 \end{pmatrix} \qquad B1(t) = \begin{pmatrix} 0.598 \\ -0.351 \end{pmatrix}$$

3.3.6 Задание 2 для самостоятельной проработки материала

1. Ввести с клавиатуры матрицы A, B и вектор b.

$$A=\begin{bmatrix} 41 & 23 & 11 \\ -17 & 58 & 30 \\ 10 & -23 & 67 \end{bmatrix}, B=\begin{bmatrix} 2 & -1 & 3 \\ 5 & 1 & 2 \\ 7 & -1 & 4 \end{bmatrix}, b=\begin{bmatrix} -1 \\ 2 \\ 3 \end{bmatrix}.$$

Вычислить

$|A|, A^T, B^{-1}, A^3, (2A+B)(A-B^2), A_{1,1}$ и так далее, $B_{1,1}$ и так далее.

Найти максимальный и минимальный элемент матрицы C=A·B.

Решить систему линейных алгебраических уравнений A·x=b по формулам Крамера:

$$x_0 = \frac{\Delta_1}{\Delta}, x_1 = \frac{\Delta_2}{\Delta}, x_2 = \frac{\Delta_3}{\Delta},$$

где

$$\Delta= \begin{vmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{vmatrix}, \Delta_1= \begin{vmatrix} b_0 & a_{01} & a_{02} \\ b_1 & a_{11} & a_{12} \\ b_2 & a_{21} & a_{22} \end{vmatrix},$$

$$\Delta_2= \begin{vmatrix} a_{00} & b_0 & a_{02} \\ a_{10} & b_1 & a_{12} \\ a_{20} & b_2 & a_{22} \end{vmatrix}, \Delta_2= \begin{vmatrix} a_{00} & a_{01} & b_0 \\ a_{10} & a_{11} & b_1 \\ a_{20} & a_{21} & b_2 \end{vmatrix}.$$

Решить систему уравнений A·X=B при помощи обратной матрицы.

2. Построить функцию пользователя для вычисления значений характеристического полинома матрицы A=$\begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{bmatrix}$, используя формулу $p(\lambda) = \lambda^3 - p_1\lambda^2 + p_2\lambda - p_3$, где

$$p_1 = \sum_{i=0}^{2} a_{ii}, p_2 = \begin{vmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{vmatrix} + \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} + \begin{vmatrix} a_{00} & a_{02} \\ a_{10} & a_{22} \end{vmatrix},$$

$$p_3 = \begin{vmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{vmatrix}.$$

Затем вычислить таблицу значений характеристического полинома матрицы B из пункта 1 для $\lambda \in [-4{,}4]$ с шагом 0,5.

3. Уравнение окружности, проходящей через три точки $M_1(x_1, y_1), M_2(x_2, y_2), M_3(x_3, y_3)$ определяется уравнением

$$\begin{vmatrix} x^2 + y^2 & x & y & 1 \\ (x_1)^2 + (y_1)^2 & x_1 & y_1 & 1 \\ (x_2)^2 + (y_2)^2 & x_2 & y_2 & 1 \\ (x_3)^2 + (y_3)^2 & x_3 & y_3 & 1 \end{vmatrix} = 0$$

Составить функцию пользователя, которая определяет, лежит ли точка $M(x, y)$ на окружности.

4. Составить примеры самостоятельно таким образом, чтобы они доказывали и подтверждали аксиомы и теоремы теории матриц, приведенных в таблице 3.7.

5. Пояснить на примерах роль переменной ORIGIN.

6. Провести векторизацию. Найти значения двумерного вектора x(t) при изменении параметра t в диапазоне 0,…,2 с шагом 0,05. Пусть каждый элемент матрицы A(t) – экспонента с заданным показателем степени, а каждый элемент вектора B(t) – синусоидальная функция.

Задаем матрицу и вектор коэффициентов.

$$A = \begin{bmatrix} 1{,}3 & 5 \\ 7 & 14 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 7 \end{bmatrix}$$

7. Вычислить следующие функции многочлена:

$$y = 4 \cdot x^2 + 5 \cdot x + 8 \text{ для } x = 1,2, \dots, 10$$

$$y = 3 \cdot x^4 - 7 \cdot x^3 + 4 \cdot x^2 - 9 \cdot x + 2 \text{ для } x = 0,0.05,1,1.5 \dots, 3$$

8. Вычислить функцию двух переменных:

$$z = 3x^2 + 4y^2 + 8 \text{ для } x = 1,1.5,2, \dots, 5; y = 0,0.5,1, \dots, 5$$

3.4 Построение графиков

Mathcad позволяет строить графики 7 типов: в прямоугольной и полярных системах координат, графики поверхностей, контурные линии, трехмерные гистограммы, точечные графики и графики векторных полей. Для графиков допустимы следующие операции: создание, удаление, перемещение, изменение размеров графического блока и формата графика.

Панель графиков вызывается нажатием кнопки с изображением графиков на математической панели (рисунок 3.9).

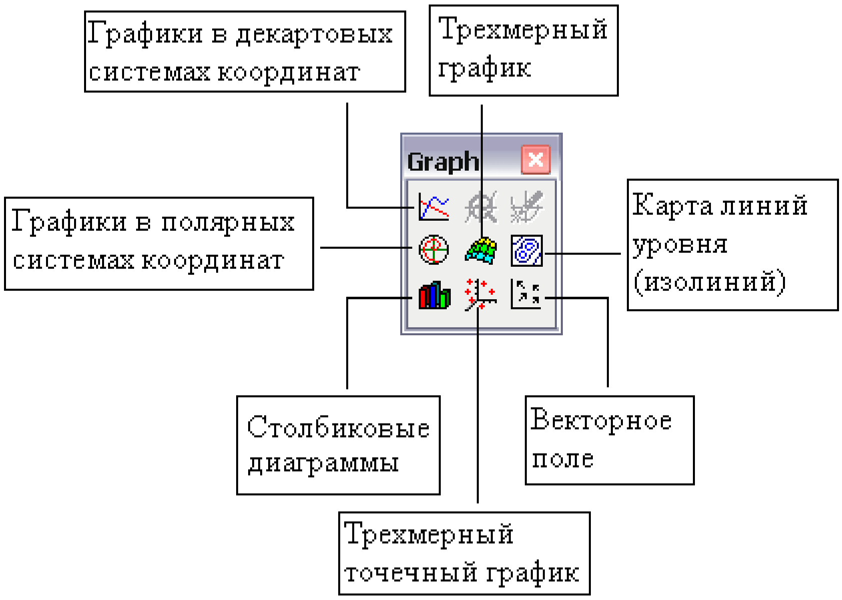


Рис. 3.9 – Панель графиков

На панели графиков расположены девять кнопок с изображением различных типов графиков (название графиков каждой кнопки высвечивается при подводе к ней курсора и ожидании в течение 3-5 секунд): X-Y Plot (<Shift>+<2>) – графики в декартовых координатах, Polar Plot (<Ctrl>+<7>) – графики в полярных координатах, 3D Bar Plot – столбиковые диаграммы, Surface Plot (<Ctrl>+<2>) – трехмерный график, Contour Plot – карта линий уровня (изолиний), Vector Field Plot – векторное поле, 3D Scatter Plot – трехмерный точечный график.

Для удаления, перемещения или формирования графической области необходимо выделить графический блок. Чтобы выделить графический блок, нужно щелкнуть левой кнопкой мыши в области графика.

Для изменения размера графика нужно переместить курсор мыши на границу выделенного блока (при этом он изменится на двунаправленную стрелку) и перемещать мышь, нажав на левую кнопку. Размер блока будет меняться в направлении перемещения курсора. Достигнув необходимого размера, отпустить кнопку мыши.

Перемещать выделенный графический блок можно с помощью мыши или воспользоваться командами меню.

Mathcad позволяет достаточно широко варьировать формат изображения графиков. Можно изменять такие параметры как масштаб по осям, задавать координатную сетку и значения координат по осям, менять внешний вид графика, цвет и толщину линий, точку зрения и т.д. параметры изменяют через диалоговое окно, содержащее несколько полей. Для вызова диалогового окна, в котором можно форматировать график, следует дважды щелкнуть левой кнопкой мыши на графическом блоке или воспользоваться командой ***Format-Graph***.

3.4.1 Прямоугольная система координат на плоскости

Декартова система координат (прямоугольная) хорошо известна. И всё же сформулируем подробнее, каким образом она задаётся на плоскости, и какие величины в результате однозначно определяют положение точки на плоскости. Не будем, однако, слишком углубляться в терминологию, т.к. используемые понятия просты и подробно изучаются в курсе средней школы.

Задать декартову систему координат на плоскости означает зафиксировать, во-первых, точку начала координат, а во-вторых, две перпендикулярные направленные оси (так называемые, оси координат). Причём, эти оси занумерованы. И, конечно, понадобится единичный отрезок, чтобы численно обозначать расстояние между двумя точками.

Таким образом, положение любой точки на плоскости однозначно определено двумя числами: первое число – величина проекции точки на первую ось (взятая с плюсом, если проекция попала на «положительную» часть оси, или с минусом, если на «отрицательную»), а второе – величина проекции на вторую ось. Прямоугольная система координат на плоскости представлена на рисунке 3.10.

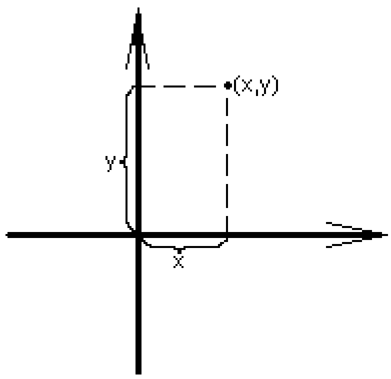


Рисунок 3.10 – Прямоугольная система координат

Стандартным образом декартова система координат обозначается Oxy, оси нумеруются таким образом, что поворот от первой оси ко второй осуществляется против часовой стрелки. Координаты точки – (x,y).

Для построения графиков в прямоугольной системе координат на плоскости необходимо:

- щелкнуть левой кнопкой мыши в том месте экрана, где нужно построить график;

- создать графический блок, для этого следует выполнить команду меню ***Insert-Graph-X/Y Plot*** (ввести символ @) или воспользоваться кнопкой из палитры кнопок) или нажать на кнопку с изображением декартовых графиков с панели графиков;

- в появившемся графическом блоке заполнить пустые поля, т.е. задать в среднем поле по горизонтальной оси аргумент, а в среднем поле по вертикальной – функцию. Еще четыре пустых поля можно использовать для указания границ по осям координат.

Пример:

1. Построим график функции $f(x) = x^2 \cdot \cos(x) + \sin(2x) + 5$ на отрезке $[-\pi;\pi]$. Для этого зададим функцию пользователя для вычисления f(x) и область значений переменной x (рис. 3.11).

$$f(x) := x^2 \cdot \cos(x) + \sin(2x) + 5$$

$$x := -\pi, -\pi + 0.1 .. \pi$$

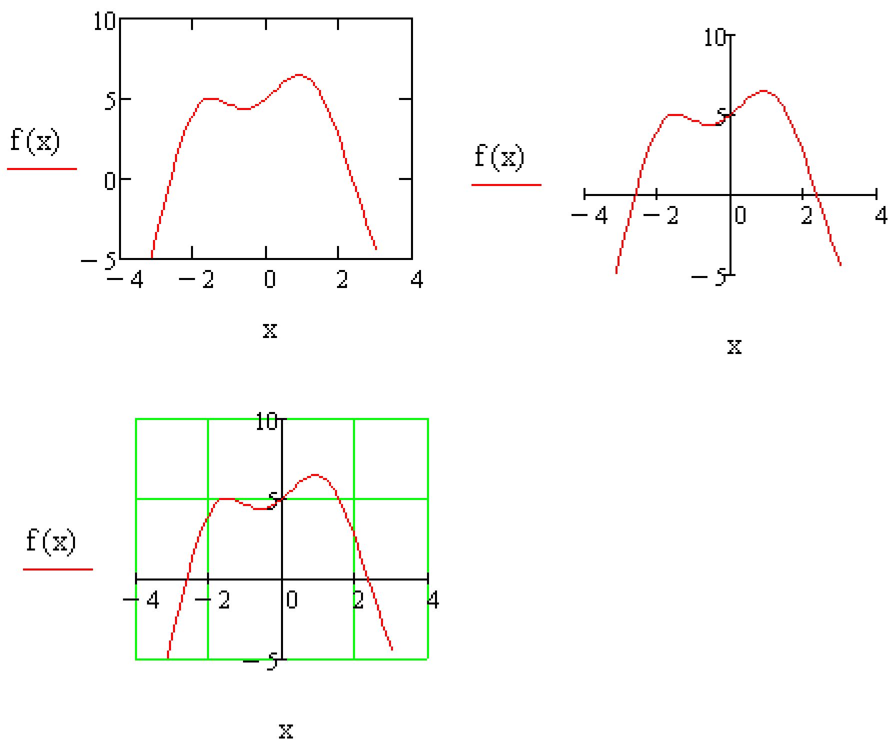


Рисунок 3.11 – Построение графика, с применением различных режимов форматирования

2. Построим графики функций $f(x) = \frac{32}{3+x^2}$ и $g(x) = (5 + x) \cdot \sin(x), x \in [-5; 11]$ в одном графическом блоке. В среднем поле вертикальной оси укажем функции, отделив их запятой. Для аргумента укажем границы -5 и 11 и подберем границы для значений функции так,

чтобы оба графика рисовались с наименьшим искажением. Пример показан на рисунке 3.12.

$$f(x) := \frac{32}{3 + x^2}$$

$$g(x) := (5 + x)\cdot\sin(x)$$

$$x := -5, -5 + 0.05 .. 11$$

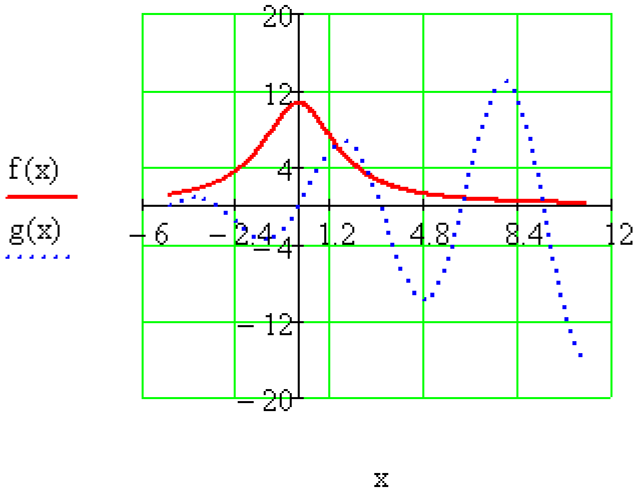


Рисунок 3.12 – Построение двух графиков функций в одном графическом окне

3. Построим графики функций с различными переменными: $f(x) = x^2 + 2 \cdot x + 4$ для $x \in [-10; 8]$ и $g(z) = 3 \cdot \sin(z) + 7$ для $z \in [-7; 10]$, которые изображены на рисунке 3.13.

$$f(x) := x^2 + 2x + 4$$

$$x := -10, -10 + 0.02 \,..\, 8$$

$$g(z) := 3\sin(z) + 7$$

$$z := -7, -7 + 0.05 \,..\, 10$$

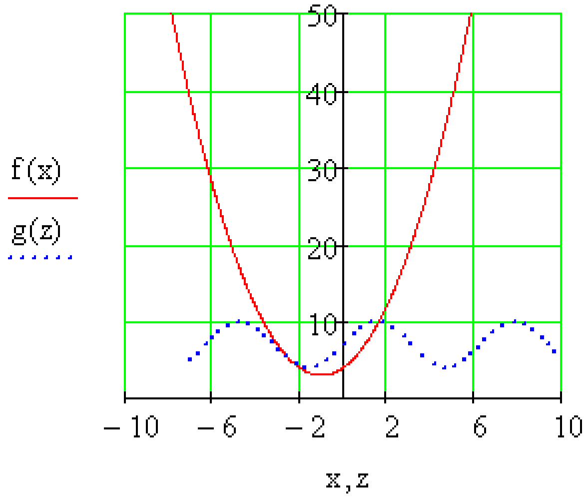
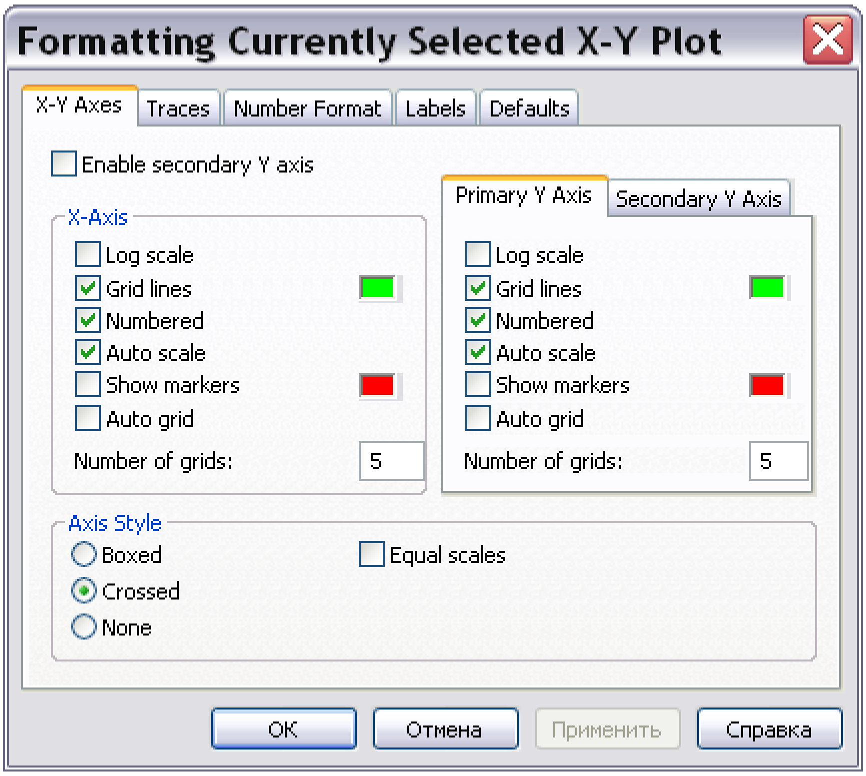
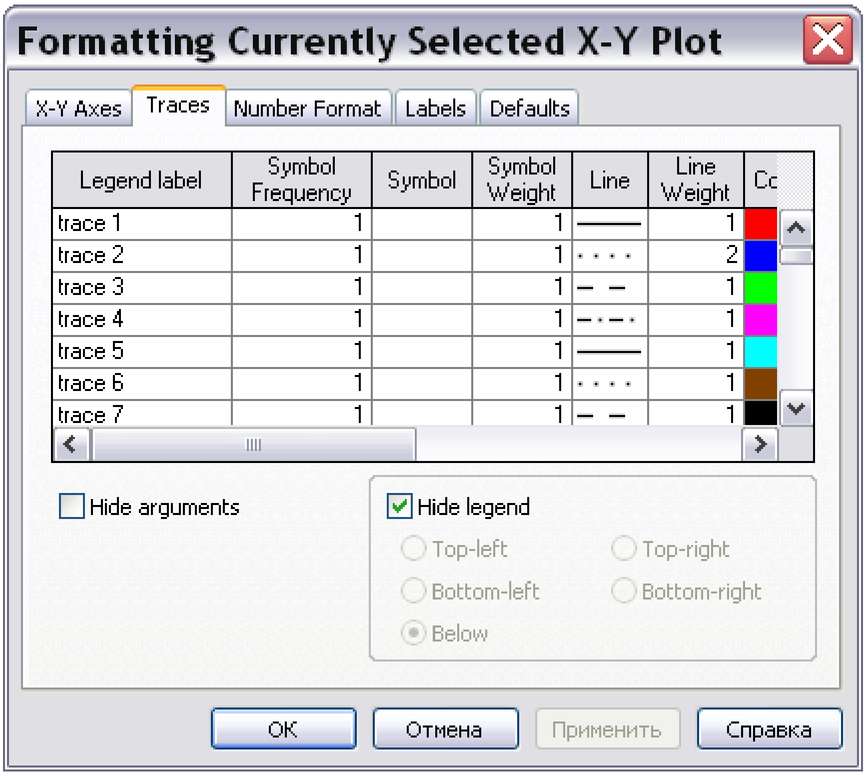


Рисунок 3.13 – Построение двух графиков функций с различными переменными в одном графическом окне

При помещении курсора внутрь графика и щелкнув левой клавишей мыши появится окно, показанное на рисунке 3.14 а, б. Оно состоит из пяти страниц. На рисунке 3.14 а, б представлены первые две страницы.



а)



б)

Рисунок 3.14 – Окна формирования графиков

В левом нижнем углу первой страницы имеются точки Boxed (коробочка), Crossed (оси), None (нет). Нажав на первую точку, введем в график оси координат.

На первой странице имеются строки X-Axes (ось X) и Primary Y-Axes (первая ось У), а под ними ряд надписей, левая часть которых относится к оси X, а правая – к оси У: Log Scale (логарифмическая шкала) вводит логарифмический масштаб для соответствующей оси; Grid lines (сетка) – ее нажатие вводит сетку на график; Numbered(оцифровка) – оцифровка сетки; Auto scale (автоматическая оцифровка); Show markers (показ маркеров); Auto grid (автоматическое разбиение сетки).

Наличие надписей Enable secondary Y-Axes (возможность второй оси У) и Secondary YAxes (вторая ось У) дает возможность формировать графики различного масштаба для различных функций.

Из ее левого столбца (trace (след) 1, trace 2 и т.д.) второй страницы следует, что на одном графике можно наносить до 16 различных функций.

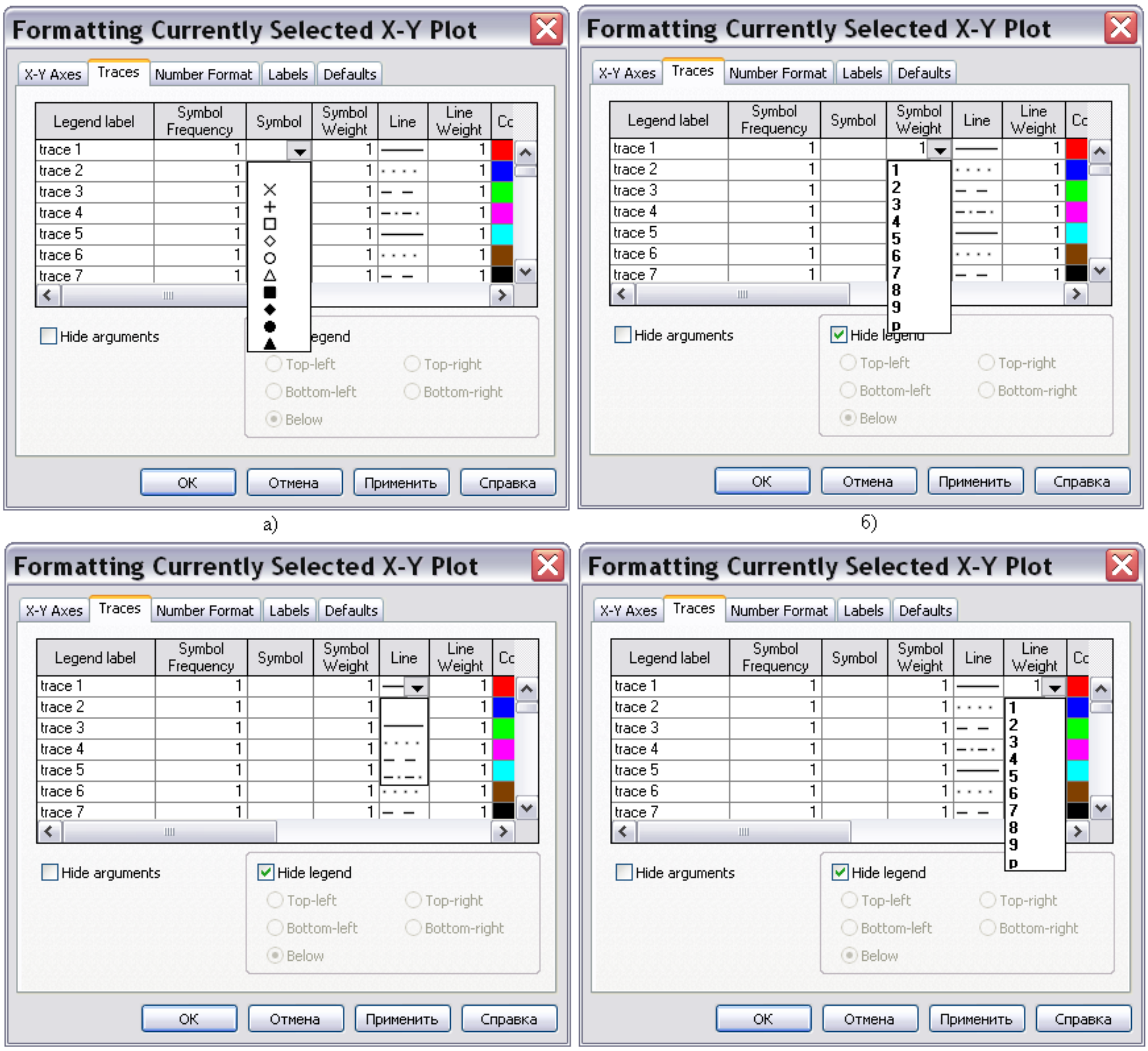
Вводя соответствующие значения в остальные столбцы, можно изменять вид (сплошная линия, пунктир, точки), цвет, толщину и т.д. каждой функции.

На третьей странице окна задается заголовок (Title), место его расположения Above(сверху), Belove (снизу), наименования осей (Axis Labels).

Выбрав те или иные требования к графику, нажмем ОК и получим желаемый график.

Чтобы увидеть сделанные изменения, нужно, не закрывая диалоговое окно, нажать кнопку «Применить». Если все в порядке, можно закрывать окно.

Вторая страница окна формирования графиков в полном объеме показана на рисунке 3.15.



а) б)

в) г)

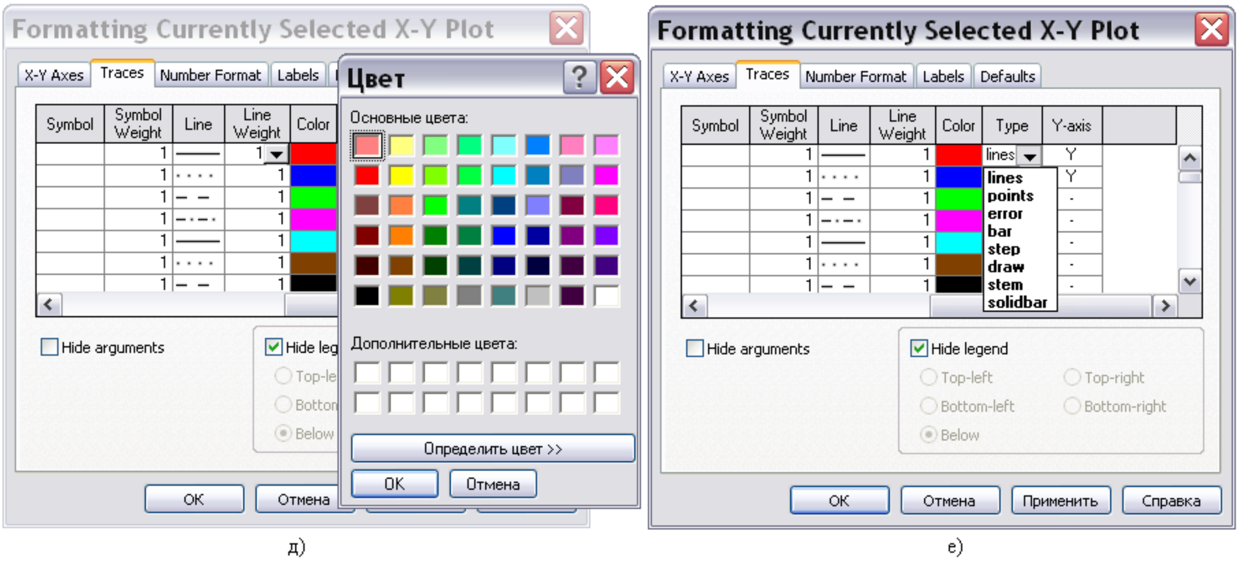


Рисунок 3.15 – Вторая страница окна формирования графиков
а) «структура» линии графика; б) толщина «структуры» линии графика; в) вид линии; г) толщина линии; д) выбор цвета линии; е) тип линии

3.4.2 Поверхности

Декартовы координаты в пространстве (график поверхности) задаются с помощью точки начала координат и трёх взаимно-перпендикулярных направленных прямых. Прямые занумерованы, задан единичный отрезок. Положение любой точки в пространстве однозначно определено тремя числами: первое число – величина проекции точки на первую ось, второе – величина проекции на вторую ось, третье – на третью. Декартова система координат в пространстве представлена на рисунке 3.16.

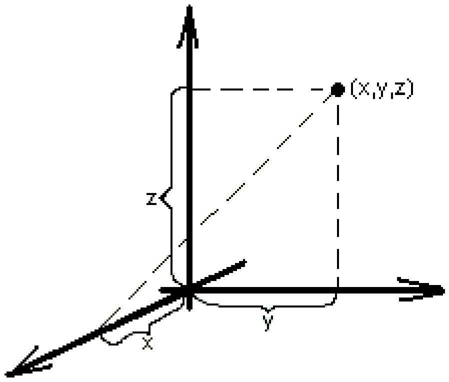


Рисунок 3.16 – Декартова система координат в пространстве

В Mathcad можно строить различные трехмерные графики: поверхности, уровней, столбиковые диаграммы и т.п. мы рассмотрим примеры построения поверхности.

Для построения поверхности необходимо:

- записать исходную функцию;
- определить узлы, в которых будет вычисляться функция;
- установить связь аргументов *x* и *y* с узлами;
- вычислить матрицу значений функций двух переменных;

- вызвать графический блок командой меню ***Insert-Graph-Surface Plot*** или кнопкой из палитры кнопок /;

- в появившемся графическом блоке в единственном пустом поле указать имя матрицы.

Рассмотрим примеры построения графиков поверхностей различными способами.

Пример:

1. Построить график поверхности

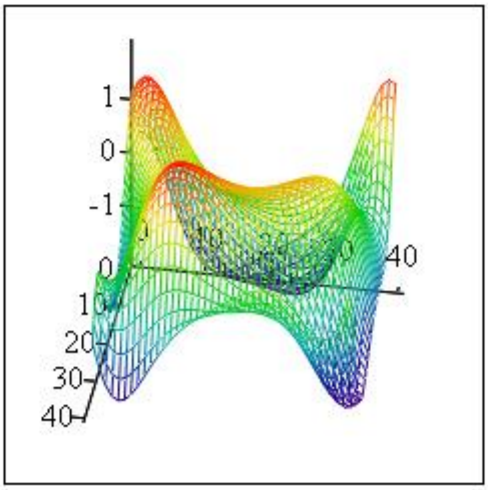
$f(x,y) = \sin(y^2 + x) + \cos(x^2 + y)$ для x от -2 до $2, y$ от -2 до 2

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

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

$$x_i := -2 + 0.1 \cdot i \quad y_j := -2 + 0.1 \cdot j$$

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



A

Рисунок 3.17 – Построение поверхности в Mathcad

Трехмерные графики можно изменять. Двойной щелчок левой кнопкой мыши на графическом блоке или команда ***Format-Graph-3D Plot*** открывает окно (рисунок 3.18), которое имеет девять вкладок, часть из которых описаны ниже:

- Axes – форматирование осей (в виде непосредственно осей, в виде плоскостей, отсутствие осей). На странице Axes (оси) нанесены те же, что и для случая плоских графиков надписи, но теперь уже для трех осей: Gird Lines – нанесение сетки координат, Numbered – оцифровка соответствующей оси, Autoscale – автоматическая разметка осей, Shove Markers – показать метки по осям, Autogird – автоматический показ сетки координат, Numbers of Grids - оцифровка сетки.;

- Title – редактирование заголовка;

- Appearance – изменение цвета и линии поверхности;

- Backplanes – показ заднего плана (показывать (Show), заполнять цветом поверхность (Fill Color) и кромку(Edge Color) и угол, под которым показываются оси координат).

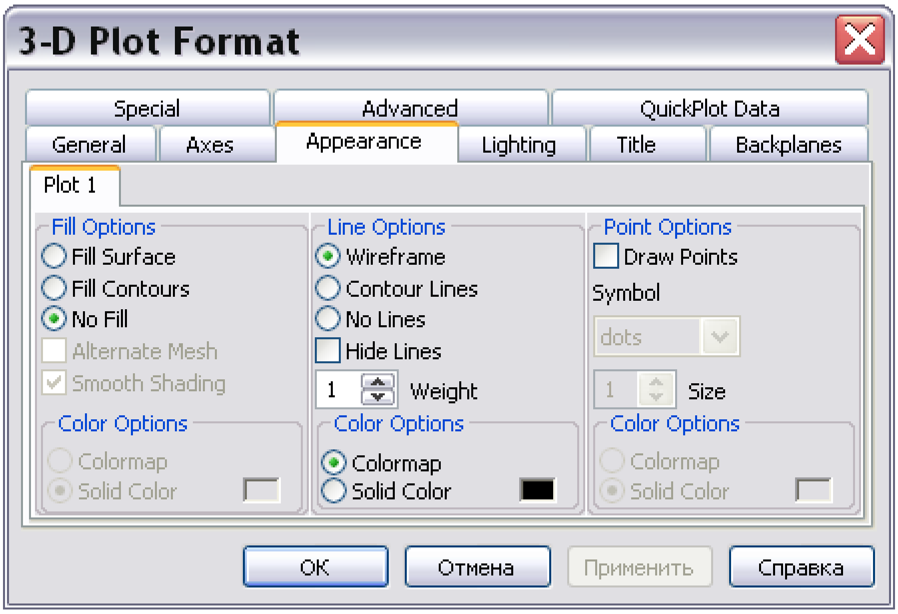


Рисунок 3.18 – Окно для изменения трехмерного графика

2. Рассмотрим другой пример – построение шара.

Построение любой объемной фигуры начинается с того, что мы задаем сетку значений на осях *x* и *y*.

- Пусть на каждой оси будет по 25 значений. N=25.

- Задаем размещение индексов от 0 до N на осях. Пусть по оси *x* индекс будет обозначен i, а на оси *y* – обозначен j.

- Рассчитаем значения углов α_i, α_j к сечению шара плоскостью, параллельной координатной плоскости XOY, и в этом сечении φ_i до пересечения с поверхностью шара – как функции i и j.

- Задаем радиус шара R.

- Рассчитаем координаты точки $x_{i,j}, y_{i,j}, z_{i,j}$ на шаре по соответствующим формулам.

- Выводим графическое поле для построения поверхности и указываем координаты *x*, *y*, *z* в скобках.

Не забываем, что Mathcad различает строчные и прописные литеры, так что если использовались прописные X, Y, Z в формулах, то так и набираются они в слотах.

$N := 25$

$i := 0..N$

$i := 0..N$

$\alpha_j := j \cdot 2 \cdot \frac{\pi}{N}$

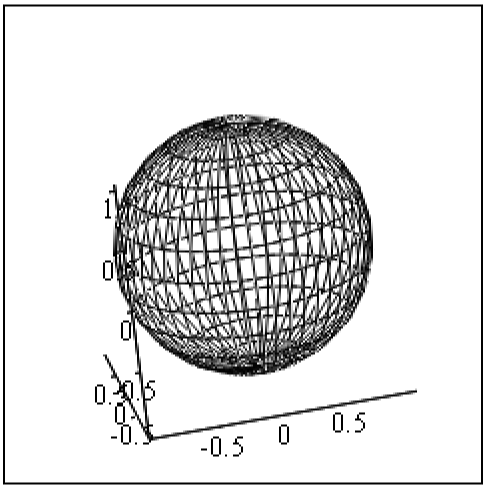
$\phi_i := i \cdot 2 \cdot \frac{\pi}{N}$

$R := 1$

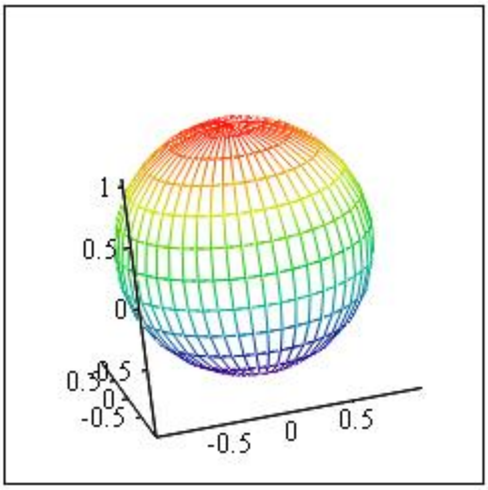
$x_{i,j} := R \cdot \sin(\alpha_j) \cdot \cos(\phi_i)$

$y_{i,j} := R \cdot \sin(\alpha_j) \cdot \sin(\phi_i)$

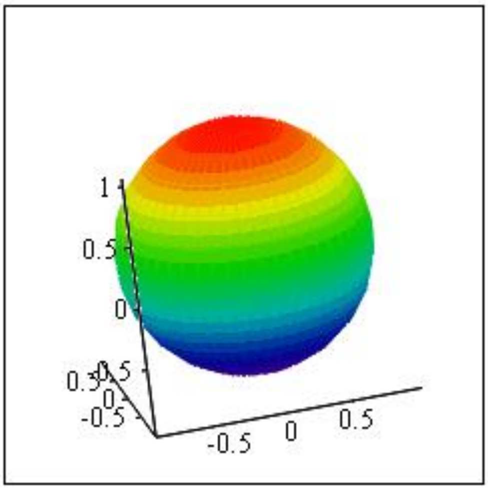
$z_{i,j} := R \cdot \cos(\alpha_j)$



(x, y, z)



(x,y,z)



(x,y,z)

Рисунок 3.19 – Трехные графики: а) опции графика – по умолчанию; б) раскраска; опция Appearance – Colormap – Fill Surface; в) раскраска; опция Appearance – Colormap – Fill Contours.

В Mathcad имеется очень удобная функция (CreateMesh), позволяющая задать диапазоны изменения аргументов и частоту нанесения сетки. Рассмотрим действие этой функции на примере построения полусферы.

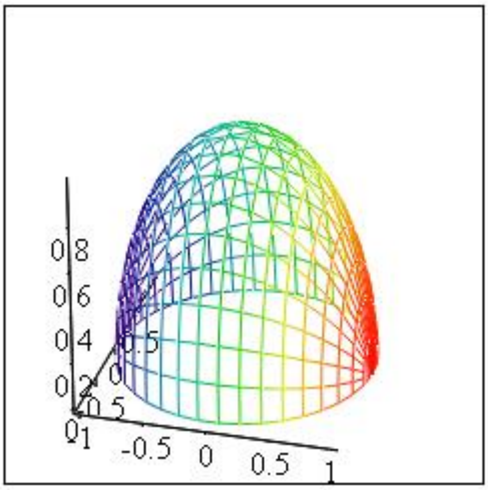
Пример:

Создадим вектор с тремя элементами, в которые впишем координаты точек сферы. Далее используем функцию CreateMesh для показа графика, задавая имя формулы для графика, диапазон изменения аргумента 1, диапазон изменения аргумента 2, число шагов сетки по аргументу 1, число шагов сетки по аргументу 2 соответственно. Далее выводим графическое поле и экспериментируем с пределами изменения аргумента, с шагом сетки, с цветом и другими опциями графика.

$R := 1$

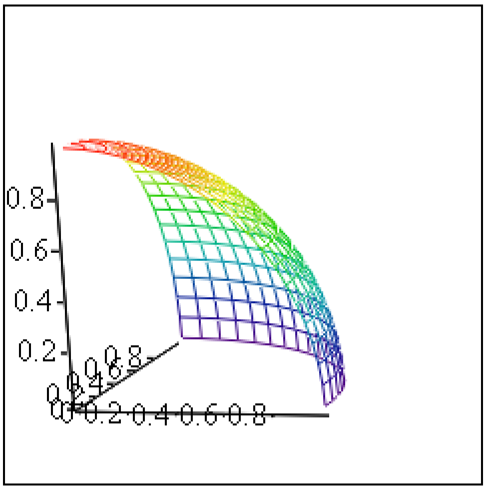
$$\mathrm{Sphere}(\alpha, \phi) := \begin{pmatrix} R\cdot\sin(\alpha)\cdot\cos(\phi) \\ R\cdot\sin(\alpha)\cdot\sin(\phi) \\ R\cdot\cos(\alpha) \end{pmatrix}$$

$H := \mathrm{CreateMesh}(\mathrm{Sphere}, 0, \pi, 0, \pi, 20, 20)$



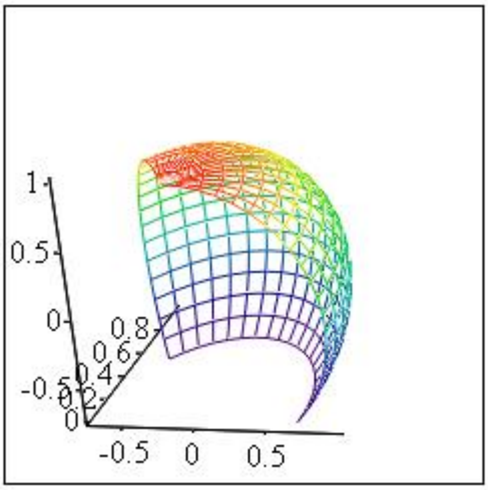
H

$$H := \mathrm{CreateMesh}\left(\mathrm{Sphere}, 0, \frac{\pi}{2}, 0, \frac{\pi}{2}, 20, 20\right)$$



H

$$H := \mathrm{CreateMesh}\left(\mathrm{Sphere}, 0, \frac{3\pi}{4}, 0, \frac{3\pi}{4}, 20, 20\right)$$



н

Рисунок 3.20 – Примеры графиков, построенных с помощью функции CreateMesh

3.4.3 Полярная система координат на плоскости

Для того, чтобы задать полярную систему координат на плоскости, надо зафиксировать, во-первых, точку начала координат, а во-вторых, луч, выходящий из этой точки. Необходимо также определить единичный отрезок и положительное направление отсчета угла между лучом и отрезком, соединяющим начало координат с какой-либо точкой плоскости.

Положение точки на плоскости задаётся двумя числами. Первое – расстояние от точки до начала координат, а второе – угол между зафиксированным лучом и отрезком, соединяющим точку и начало координат (рис. 3.21).

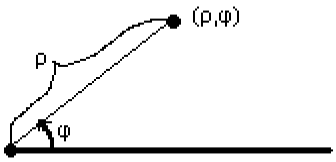


Рисунок 3.21 – Определение положения точки на плоскости

Обычно направление отсчета угла выбирают против часовой стрелки. Стандартное обозначение координат точки в полярной системе – (ρ, φ). Очевидно, $\rho \geq 0$.

Существуют формулы перехода между заданными стандартным образом декартовой и полярной системами координат. Если они друг другу соответствуют (т.е. должны совпадать начала координат в обеих системах, луч полярной системы координат должен совпадать с «положительной» частью первой оси декартовой системы, должны быть одинаковыми единичные отрезки), то

$x = \rho \cdot \cos\varphi$,

$y = \rho \cdot \sin\varphi$.

В других случаях формулы зависят от постановки задачи, но получить их легко из геометрических соображений.

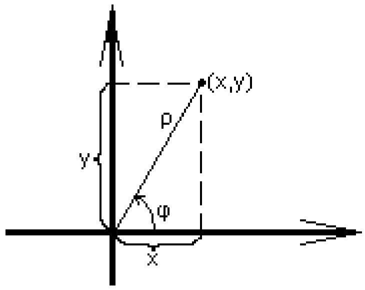


Рисунок 3.22 – Положение точки на плоскости в декартовой системе координат

С помощью этих формул можно осуществлять переход между двумя системами координат, преобразовывать координаты точек, уравнения кривых и т.д.

В полярной системе координат очень просто выглядят уравнения прямых, проходящих через начало координат и окружностей с центром в этой точке. Кроме того, уравнения многих стандартных, часто используемых, кривых принято (с точки зрения простоты) записывать в полярных координатах.

Для построения графиков в полярных системах координат необходимо выбрать на панели графиков – Poral Plot или воспользоваться «горячими» клавишами – <Ctrl>+<7>.

Пример:

Построим графики функций в полярных системах координат:

$y(x)=\sin(x)$; $z(x)=\cos(x)$ для $x\in[0; 2\pi]$.

$x := 0, 0.05 .. 2\pi$

$y(x) := \sin(x)$

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

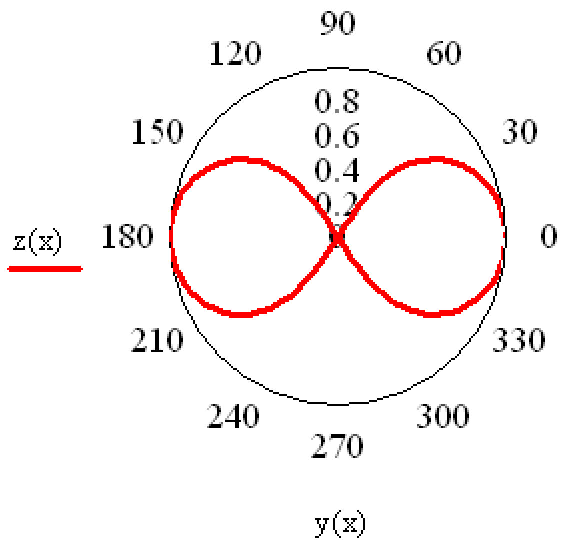


Рисунок 3.23 – Фигура Лисажу в полярной системе координат

Форматирование полярных графиков. Если вы хотите отредактировать график в полярных координатах, необходимо выделить график (щелчком левой кнопки мыши) и выполнить команду *Format-Graph-Polar Plot* или выполнить двойной щелчок на выделенном графике. При этом откроется окно *Formatting Currently Selected Polar Plot* (форматирование полярного графика). Это окно содержит те же вкладки, что и для графиков в декартовой системе координат.

Вкладка *Polar Axes* содержит следующие элементы:

- *Log Scale* (логарифмическая шкала) - используется для создания логарифмической r-оси.
- *Grid Lines* (вспомогательные линии) - отображает сетку линий, соответствующих уравнениям *r=const* и *q=const*.
- *Numbered* (нумерация) - линии *r=const* и *q=const* снабжаются подписями.
- *Show Markers* (показать метки) - при помощи этой опции можно снабдить график двумя дополнительными пунктирными окружностями *r=const*. Для этого надо ввести нужные значения радиуса в появившиеся ячейки. Кроме того, справа от графика указывается минимальный и максимальный радиус; можно увеличить или уменьшить график, введя в эти ячейки собственные значения.
- *Auto Grid* (автосетка) - при установке этой опции число линий

сетки определяет Mathcad.

Остальные значения параметров на этой вкладке и на других вкладках те же самые, что и в случае команды *X-Y Plot*.

3.4.4 Задание 3 для самостоятельной проработки материала

1. Построить график функции $f(x) = x^2 + 1{,}2 \cdot x + 3{,}1$ на отрезке $[-2; 3]$.

2. Построить в одном графическом блоке графики функций: $f(x) = x \cdot \sin(4x)$, $g(x) = \cos(2x) + 0{,}5$ на отрезке $[-\pi; \pi]$.

3. Построить на одном чертеже графики 3-х различных функций с различными аргументами: $r(b) = 2b - \lg(b) + 1{,}2$, $r(t) = t - \cos(t)$, $f(x) = x + 2x^2$.

4. Выполнить форматирование построенных графиков: изменить толщину и стиль линий, масштаб, нанести дополнительные линии, ввести надписи.

5. Создать график, путем увеличения некоторой области графика из пункта 3. Прочитать координаты выбранной точки.

6. Построить график функции $z(x, y) = \sin(x^2 + y^2)$ на области $[-1; 1] \times [-1; 1]$.

7. Отформатировать полученный график, изменяя свойства обзора и способы отображения поверхности, добавить заголовок и отформатировать оси.

8. Построить контурный график функции из пункта 6.

9. Построить график функции $z(x, y) = 55x^2 + 25x^2$, область выбрать самостоятельно. Выберите зеленый цвет поверхности, розовый цвет заднего плана, угол представления 45 градусов и оцифрованные оси.

10. Построить график уровней и столбиковую диаграмму для функции из пункта 9.

11. Выведите уравнения (аналогично выводу уравнения шара) для конуса и постройте чертеж половины конуса (рисунок 3.24), высота конуса равна 4, угол при вершине конуса равен π/3.

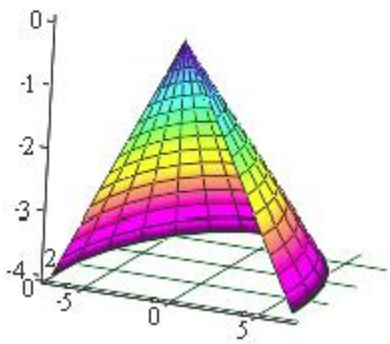


Рисунок 3.24 – Половина конуса

12. Выведите уравнения (аналогично выводу уравнения шара) для цилиндра и постройте чертеж одной третьей части цилиндра, как показано на рисунке 3.25. высота цилиндра равна 4.

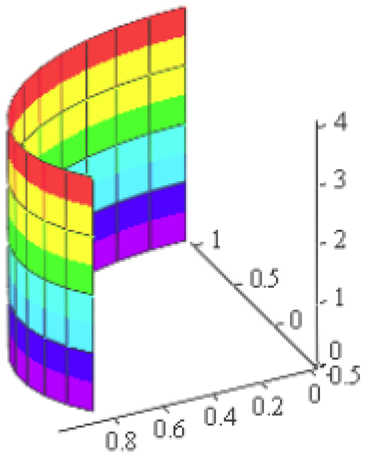


Рисунок 3.25 – Одна третья часть цилиндра

13. Построить графики заданных функций в полярной системе координат

$r(\theta) = 1 + sin\left(2 \cdot \theta + 3 \cdot \frac{\pi}{2}\right)$, $d(\theta) = 1 + \frac{\sin(3 \cdot \theta + \pi)}{2}$, если $\theta \epsilon [0; 2 \cdot \pi]$ с шагом $N = 15$.

14. Построить кривую на плоскости с заданными параметрами:

$x_i = \sin\left(2 \cdot \pi \cdot \frac{i}{N}\right)$, $y_i = \sin\left(4 \cdot \pi \cdot \frac{i}{N}\right)$

15. Построить кривую в пространстве.

$x_i = cos\left(\frac{i}{N} \cdot 6 \cdot \pi\right)$, $y_i = sin\left(\frac{i}{N} \cdot 6 \cdot N\right)$, $z_i = \frac{i}{N} \cdot 3$

15*. Построить векторное и градиентное поле следующей функции:

$f(x, y) = \sin(x) \cdot \cos(y)$.

16*. Построить график функции $f(x) = x \cdot \sin(x)^2$. Получить вид поверхности путем вращения кривой вокруг оси Ox и Oy (использовать функцию *CreateMesh*).

3.5 Решение уравнений

Алгебраические уравнения в Mathcad решаются как численными, так и аналитическими методами.

3.5.1 Численное решение нелинейных алгебраических уравнений

В Mathcad корни алгебраических уравнений и систем определяется с помощью встроенных функций.

Для решения одного уравнения с одним неизвестным используется функция *root(f(x),x)*, возвращая значение аргумента *x*, при котором значение функции *f(x)* обращается в ноль. Эта функция для поиска корня использует метод секущих. Поэтому перед использованием функции *root(f(x),x)* необходимо переменной *x* присвоить начальное значение.

Важно запомнить, что само уравнение в Mathcad не набирается.

Пример:

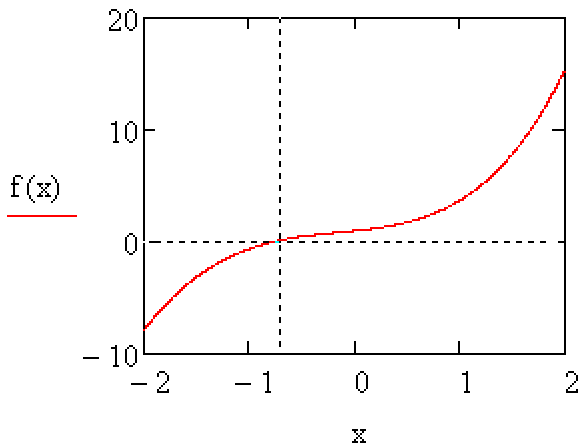
$f(x) := x^3 + e^x$

$x := 3$

$s1 := root(f(x), x)$

$s1 = -0.773$

Графическое решение:



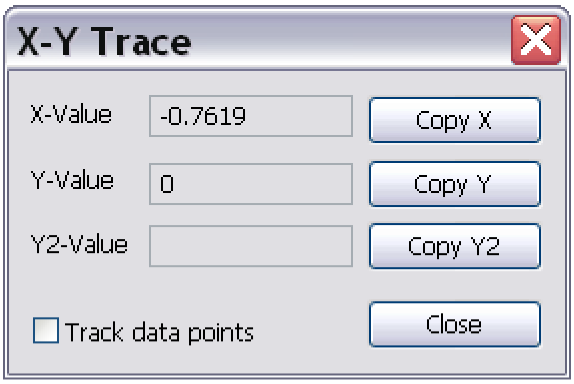


Рисунок 3.26 – Графическое решение уравнения с одной неизвестной функции $f(x) = x^3 + e^x$

Если корней несколько, то потребуется столько же раз вызывать функцию, меняя начальное приближение. Полезно для отделения корней исследовать график функции *f(x)*. Тип найденного решения (вещественный, комплексный) зависит от начального приближения.

Пример:

Найти корни уравнения $x^3 - 10 \cdot x + 2 = 0$

Предварительно построим график:

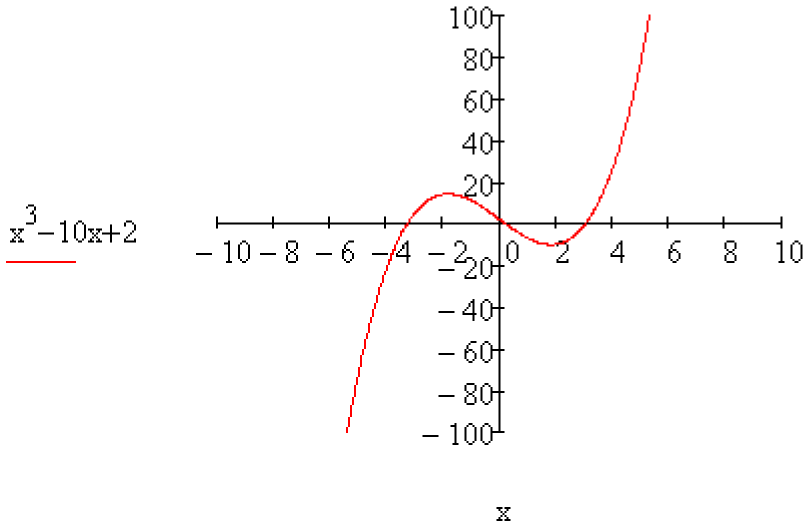


Рисунок 3.27 – Графическое решение уравнения с одной неизвестной функции $f(x) = x^3 - 10 \cdot x + 2$

Зададим функцию пользователя $f(x) = x^3 - 10 \cdot x + 2$. Вычислим корни уравнения, трижды задавая начальное приближение.

$$f(x) := x^3 - 10x + 2$$

$x := 0 \qquad \mathrm{root}(f(x), x) = 0.201$

$x := -5 \qquad \mathrm{root}(f(x), x) = -3.258$

$x := 3 \qquad \mathrm{root}(f(x), x) = 3.057$

Для увеличения или уменьшения точности, с которой ищется корень, можно изменить значение встроенной переменной TOL. При уменьшении значения этой переменной ответ станет более точным, но время вычислений увеличивается.

Если решение с заданной точностью не находится, то на экране появится сообщение «отсутствует сходимость».

Функция *root* может использоваться для решения уравнений с параметром. Для этого определяют данное уравнение как функцию нескольких переменных и решают уравнение, присвоив начальное значение параметрам и переменной.

Для нахождения корней полинома лучше использовать функцию *polyroots(v)*, которая не требует начального приближения и возвращает сразу все корни (вещественные и мнимые).

Решение уравнений с помощью функции *polyroots(v)* можно разбить на несколько этапов:

1. Вводим функцию для исходного уравнения;
2. Составляем вектор столбец *v* коэффициентов уравнения;
3. Записываем функцию *r=polyroots(v)*;

4. Выводим результат на экран.

Пример:

Найти решение уравнения $x^3 - 10 \cdot x + 2 = 0$.

Количество решений можно определить по максимальной степени полинома n. В нашем примере количество корней равно 3.

Введем функцию для исходного уравнения и вектор коэффициентов:

Размерность вектора определяется как *n+1* (к максимальной степени полинома прибавляется единица).

$$1 \cdot x^3 + 0 \cdot x^2 - 10 \cdot x^1 + 2 \cdot x^0 = 0$$

$g(x) := x^3 - 10x + 2$

$$v := \begin{pmatrix} 2 \\ -10 \\ 0 \\ 1 \end{pmatrix}$$

$r := \mathrm{polyroots}(v)$

$$r = \begin{pmatrix} -3.258 \\ 0.201 \\ 3.057 \end{pmatrix}$$

Построим график многочлена *g(x)* и выведем координаты корней *g(x)*.

$$j := 0, 1 .. 2$$

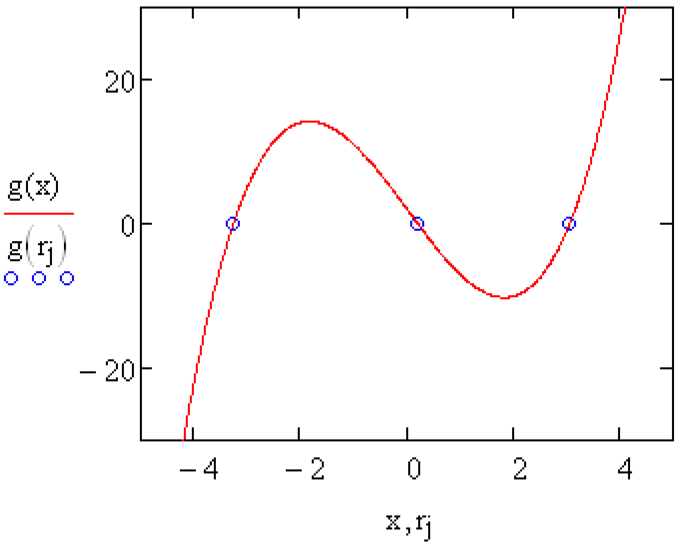


Рисунок 3.28 – Построение графика многочлена и выведение координат корней

Следует обратить внимание, на то, что первый элемент вектора соответствует коэффициенту уравнения при свободном члене.

3.5.2 Численное решение системы линейных алгебраических уравнений

При численном решении систем линейных уравнений используется специальный вычислительный блок, открываемый служебным словом – директивой Given. Блок имеет следующую структуру:

1. Задание начальных приближений;
2. Given;
3. Уравнения;
4. Ограничительные условия выражения с функцией Find.

Mathcad решает системы уравнений при помощи Given, в которых применяются итерационные методы. Для решения системы уравнений нужно выполнить следующее:

1. Задать начальные приближения для всех переменных, значения которых находим. На основе начального приближения строится последовательность, сходящаяся к искомому решению;
2. Напечатать ключевое слово Given;
3. Ниже Given ввести уравнения и неравенства, задающие систему. Левые и правые части уравнений связываются знаками операций сравнения >, ≥, <, ≤ или символом равно (=), который вводится комбинацией клавиш [Ctrl]+[=].
4. Ввести выражение, которое включает функцию Find. В качестве аргументов этой функции нужно указать переменные, значения которых находим. Функция Find возвращает в виде вектора решение системы уравнений.

Пример:

$x1 := 0 \qquad x2 := 0 \qquad x3 := 1 \qquad x4 := 1$

Given

$0.63 \cdot x1 - 9.72 \cdot x2 - 0.62 \cdot x3 + 0.48 \cdot x4 = 1.03$

$21.08 \cdot x1 + 0.64 \cdot x2 + 1.95 \cdot x3 + 1.54 \cdot x4 = -9.47$

$0.88 \cdot x1 - 0.72 \cdot x2 + 1.36 \cdot x3 + 0.68 \cdot x4 = -0.85$

$3.58 \cdot x1 + 1.48 \cdot x2 + 0.82 \cdot x3 + 1.58 \cdot x4 = 1.02$

$$\begin{pmatrix} x1 \\ x2 \\ x3 \\ x4 \end{pmatrix} := \mathrm{Find}(x1, x2, x3, x4)$$

$$\begin{pmatrix} x1 \\ x2 \\ x3 \\ x4 \end{pmatrix} = \begin{pmatrix} -0.493 \\ 0.079 \\ -1.497 \\ 2.466 \end{pmatrix}$$

Подобным образом можно решать и нелинейные уравнения. Однако они имеют несколько корней. Задавшись начальными приближениями, мы найдем в лучшем случае один корень, ближайший к начальному приближению. Таким способом имеет смысл искать корни трансцендентных уравнений, имеющие, как известно, бесконечное количество корней.

Пример:

Найти корень трансцендентного уравнения.

X sin(x)+ cos(x) =25, ближайший x=1.

Набираем задачу описанным выше способом и находим значение x. Однако получить решение при начальном приближении 10 нам не удастся.

Mathcad позволяет решать системы линейных алгебраических уравнений в матричной форме. Решение можно получить двумя способами.

1 способ.

Как известно, система линейных алгебраических уравнений в матричной форме имеет вид:

A·X=B,

где A – квадратная матрица коэффициентов,

X – вектор – столбец неизвестных,

B – вектор – столбец свободных членов системы.

Решение системы в матричной форме: $X = A^{-1}B$.

Решим в матричной форме систему:

$$11\cdot x + 12\cdot y + 31\cdot z = 9$$

$$4\cdot x + 52\cdot y + 69\cdot z = 8$$

$$7\cdot x + 86\cdot y + 93\cdot z = 7$$

1) Наберем ORIGIN:=1. Как говорилось выше, это означает, что счет элементов будет производиться не от нуля, а с единицы.

2) Введем матрицу А.

3) Введем вектор – столбец В.

4) Набор выражения для Х желательно выполнять, используя соответствующую кнопку матричной панели. После этого наберем Х= и сразу получим вектор ответа.

$$A := \begin{pmatrix} 1 & 12 & 31 \\ 4 & 52 & 69 \\ 7 & 86 & 93 \end{pmatrix} \qquad B := \begin{pmatrix} 9 \\ 8 \\ 7 \end{pmatrix}$$

$$X := A^{-1}\cdot B \qquad X = \begin{pmatrix} 7.694 \\ -1.016 \\ 0.435 \end{pmatrix}$$

2 способ

Возможно получения решения матричного уравнения с помощью специальной функции lsolve.

$$ORIGIN := 1$$

$$A := \begin{pmatrix} 1 & 12 & 31 \\ 4 & 52 & 69 \\ 7 & 86 & 93 \end{pmatrix} \qquad B := \begin{pmatrix} 9 \\ 8 \\ 7 \end{pmatrix}$$

$$X := \mathrm{lsolve}(A, B)$$

$$X = \begin{pmatrix} 7.694 \\ -1.016 \\ 0.435 \end{pmatrix}$$

Уравнения и неравенства, которые следуют за словом Given, называются ограничениями. Ключевое слово Given, ограничения, выражение, содержащее функцию Find, называются блоком решения уравнений. В этом блоке могут появляться выражения строго определенного типа. Нельзя использовать ограничения, содержащие двойное неравенство вида $a<d<c$ или строгое неравенство ≠, операторы присваивания или дискретные аргументы.

Блоки решений не могут быть сложенными друг в друга.

Если в результате решения системы уравнений будет выдано сообщение об ошибке типа «Решение не найдено», это означает, что на каком-то этапе итераций не может быть найдено приемлемое приближение к искомому решению. В этом случае полезно исследовать графики, связанные с системой, для определения области начального приближения. Можно также изменять значение переменной TOL.

Системы уравнений в Mathcad можно также решать, используя функцию Minerr. Эта функция использует тот же алгоритм, что и функция Find. Различие заключается в следующем. Если в случае поиска решения не может быть получено дальнейшее уточнение текущего приближения к решению, Minerr возвращает это приближение, а не сообщение об ошибке. Правила использования функции Minerr такие же, как и для функции Find. Поэтому при использовании функции Minerr необходимо всегда включать дополнительную проверку достоверности получаемых результатов.

Пример:

$x := 1 \qquad y := 0$

Given

$\sin(x+1) - y = 1.2$

$2 \cdot x + \cos(y) = 2$

$$\mathrm{minerr}(x,y) = \begin{pmatrix} 0.51 \\ -0.202 \end{pmatrix}$$

3.5.3 Решение алгебраических уравнений в аналитической (символьной) форме

Mathcad предоставляет возможность решения алгебраических уравнений в символьной (аналитической) форме. Преимуществом символьного решения является возможность решения уравнений с буквенными значениями коэффициентов. Правда, более или менее сложные уравнения символьно в Mathcad не решаются, поэтому приходится обращаться к численным методам.

3.5.4 Решение систем линейных уравнений

Символьное решение линейных систем алгебраических уравнений производится с помощью двух операций: solve (решить) и lsolve. Рассмотрим на примерах различные методы решения.

Пример:

Система линейных алгебраических уравнений задана матрицей M коэффициентов и вектором v правых частей. Найти аналитическое решение.

Сначала вводим матрицу и вектор.

$$M := \begin{pmatrix} 0.3 & 0.2 & 6.6 & -1.1 \\ 4.5 & -1.8 & -0.3 & 6.5 \\ -7.3 & 9.7 & 10.9 & -4.1 \\ 8.1 & -2.7 & 8.7 & 8.9 \end{pmatrix} \qquad V := \begin{pmatrix} 1 \\ 0.1 \\ 0.01 \\ 1\cdot 10^{-3} \end{pmatrix}$$

А) Решение с использованием встроенной функции lsolve. Функция набирается с клавиатуры или из окна встроенных функций. Стрелка набирается с панели символьных решений.

$$\mathrm{lsolve}(M,V) \rightarrow \begin{pmatrix} -3.9371585952195628203 \\ -2.9752573457871181164 \\ 0.7459060209508983589 5 \\ 1.9516188095933060907 \end{pmatrix}$$

Б) Решение с использованием оператора solve

$$M\cdot\begin{pmatrix} w \\ x \\ y \\ z \end{pmatrix} - V \ \left|\begin{array}{l} \mathrm{solve}, w, x, y, z \\ \mathrm{float}, 4 \end{array}\right. \rightarrow (-3.937 \ -2.975 \ \ .7459 \ \ 1.952)$$

Здесь помимо оператора lsolve использован оператор float (плавающая точка) и задана точность решения – 4 знака.

Операторы solve и float набираются последовательно.

В) Решение в скалярной форме

$$\begin{pmatrix} 0.3\cdot w + 0.2\cdot x + 6.6\cdot y - 1.1\cdot z = 1 \\ 4.5\cdot w - 1.8\cdot x - 0.3\cdot y + 6.5\cdot z = 0.1 \\ -7.3\cdot w + 9.7\cdot x + 10.9\cdot y - 4.1\cdot z = 0.01 \\ 8.1\cdot w - 2.7\cdot x + 8.7\cdot y + 8.9\cdot z = 1\cdot 10^{-3} \end{pmatrix} \left|\begin{array}{l} \mathrm{solve}, w, x, y, z \\ \mathrm{float}, 4 \end{array}\right. \rightarrow (-3.937 \ -2.975 \ \ .7459 \ \ 1.952)$$

Г) Решение с созданием решающего блока и директивы Given. Директива Given и оператор Find набираются с клавиатуры.

$$\text{Given}$$

$$3\cdot x1+4\cdot x2+4\cdot x3+6x4+8x5=21$$

$$5\cdot x1+7\cdot x2+8\cdot x3-7x4-9x5=34$$

$$9\cdot x1+12\cdot x2+9\cdot x3+8x4+4x5=41$$

$$13\cdot x1+3\cdot x2+19\cdot x3+18x4+9x5=141$$

$$23\cdot x1+13\cdot x2+9\cdot x3+8x4+19x5=241$$

$$\text{Find}(x1,x2,x3,x4,x5)\rightarrow\begin{pmatrix}\dfrac{30965}{2539}\\ \dfrac{-58955}{10156}\\ \dfrac{4366}{2539}\\ \dfrac{-15189}{5078}\\ \dfrac{23741}{10156}\end{pmatrix}$$

Д) Решение системы с буквенными коэффициентами

$$\text{Given}$$

$$a1\cdot x+b1\cdot y+c1\cdot z=d1$$

$$a2\cdot x+b2\cdot y+c2\cdot z=d2$$

$$a3\cdot x+b2\cdot y+c2\cdot z=d3$$

$$\text{Find}(x,y,z)\rightarrow\begin{bmatrix}\dfrac{(-d3)+d2}{(-a3)+a2}\\ \dfrac{-[(-a2)\cdot c2\cdot d1+a2\cdot d3\cdot c1+c2\cdot a1\cdot d2-c2\cdot a1\cdot d3+c2\cdot d1\cdot a3-d2\cdot a3\cdot c1]}{[(-a3)+a2]\cdot(b1\cdot c2-c1\cdot b2)}\\ \dfrac{-[(-a1)\cdot b2\cdot d2-b1\cdot a2\cdot d3+b1\cdot a3\cdot d2+a1\cdot d3\cdot b2-d1\cdot a3\cdot b2+d1\cdot b2\cdot a2]}{b1\cdot a2\cdot c2-b1\cdot a3\cdot c2+c1\cdot a3\cdot b2-c1\cdot b2\cdot a2}\end{bmatrix}$$

3.5.5 Символьное решение нелинейных алгебраических уравнений

Решение уравнения четвертой степени с численными коэффициентами с использованием оператора solve.

Пример:

$$7x^2 - 9x + 1 \text{ solve}, x \rightarrow \begin{pmatrix} \frac{9}{14} - \frac{\sqrt{53}}{14} \\ \frac{\sqrt{53}}{14} + \frac{9}{14} \end{pmatrix}$$

Решение квадратного уравнения с буквенными коэффициентами.

$$a \cdot x^2 + b \cdot x + c = 0 \text{ solve}, x \rightarrow \begin{pmatrix} -\frac{b - \sqrt{b^2 - 4 \cdot a \cdot c}}{2 \cdot a} \\ -\frac{b + \sqrt{b^2 - 4 \cdot a \cdot c}}{2 \cdot a} \end{pmatrix}$$

Решение квадратного уравнения с буквенными коэффициентами с формированием решающего блока

Given

$$x^2 + y^2 = r^2$$

$$(x - \alpha)^2 + y^2 = r^2$$

$$M(\alpha, r) := \mathbf{Find}(x, y) \rightarrow \begin{pmatrix} \frac{\alpha}{2} & \frac{\alpha}{2} \\ \frac{\sqrt{4 \cdot r^2 - \alpha^2}}{2} & -\frac{\sqrt{4 \cdot r^2 - \alpha^2}}{2} \end{pmatrix}$$

Но уже кубическое уравнение с буквенными коэффициентами не решается!

3.5.6 Задание 4 для самостоятельной проработки материала

Решить уравнения с помощью функции root(f(x),x)

1. $2^x + 5 \cdot x - 3 = 0$
2. $0.5^x = (x - 2)^2$
3. $(x - 3) \cdot \cos(x) = 1$
4. $\mathrm{arctg}(x) + \frac{1}{3 \cdot x^3} = 0$
5. $\sin(x + \frac{\pi}{3}) - 0.5 \cdot x = 0$
6. $5^x - 3 \cdot x = 0$
7. $(x - 1)^2 \cdot \lg(x + 1) = 0$

Решить уравнения с помощью функции polyroots(v)

8. $5 \cdot x^5 + 6 \cdot x^3 + 8 \cdot x^2 + 2 \cdot x = 0$

9. $x^4 - x^3 - 2 \cdot x^2 + 3 \cdot x - 3 = 0$
10. $3 \cdot x^4 - 8 \cdot x^3 - 8 \cdot x^2 = -2$
11. $2 \cdot x^4 - x^2 - 10 = 0$
12. $5 \cdot x^4 + 8 \cdot x^3 + 3 \cdot x^2 + 9 \cdot x + 8 = 0$

Решить систему уравнений с помощью Find

13. 5x+6y-9z+2v-7w=90
    3x-4y+5z-3v+4w=12
    9x+y +3z-2v +9w=51
    7x+2y-8z+v +10w=32
    6x+5y-4z+3v-2w= 87
14. 4.5x+7.9y-2.1v+6.75w+7.9u= 43
    5.6x+7.2y+9.8z+3.9v+3.4w+8.3u=12.54
    5.6x+98.5y+43.7z+67.85v+4.9w+21.5u = 54.98
    65.75x+54.32y-78.32z-565.9v+32w+78.54u = 55.5
    54.2x+76.45y+32.23z+ 45.71v+43.43w+ u = 65.21
    8.9x+9.8y-5.6z+6.5v-4.5w+2.1u = 0
15. 0.46x1+1.72x2+2.53x3=2.44
    1.53x1 – 2.32x2 – 1.83x3=2.83
    0.75x1+0.86x2+3.73x3=1.06
16. 4.24x1+2.73x2 – 1.55x3=1.87
    2.34x1+1.27x2 – 3.83x3=2.16
    3.05x1 – 1.05x2 – 0.63x3= – 1.26

Решить систему уравнений с помощью Minerr

17. sin(x – 6) – y=1.6
    3x – cos(y)=0.9
18. $x^2 + y^2 = 1$
    sin(x+y) – 1.1x=0.1
19. $e^{x+y} - x^2 + y = 2$
    $(x + 0.5)^2 + y^2 = 1$
20. tg(y-x)+x·y=0.3
    $x^2 + y^2 = 1.5$

3.5.7 Дифференциальные уравнения

Mathcad имеет ряд встроенных функций для решения обыкновенных дифференциальных уравнений. Некоторые из этих функций используют специфические свойства конкретного дифференциального уравнения, чтобы обеспечить достаточное быстродействие и точность при поиске решения. Другие полезны, когда требуется получить решение и построить его график. В результате решения получается матрица, содержащая значения функции, вычисленные на некотором множестве точек (на некоторой сетке значений).

Каждая из встроенных функций, предназначенных для решения ОДУ, требует чтобы было задано следующее:

- Начальные условия.
- Множество точек, в которых необходимо найти решение.
- Само дифференциальное уравнение.

Наиболее употребляемой для решения ОДУ является функция rkfixed, которая используется для поиска решения методом Рунге-Кутта четвертого порядка. В результате решения получается матрица, имеющая два столбца. Первый столбец – точки, в которых ищется решение ОДУ. Второй – это столбец значений найденного решения в соответствующих точках. Функция rkfixed(y,x1,x2,npoints,D) имеет следующие аргументы:

y – вектор начальных условий размерности n, где n – порядок дифференцированного уравнения или число уравнений в системе. Для уравнения первого порядка этот вектор вырождается в точку.

x1, x2 – граничные точки интервала, на котором ищется решение.

npoints – число точек (не считая начальной), в которых ищется приближенное решение. Число строк в возвращаемой матрице решения определяется как (1+ npoints).

D(x,y) – функция, которая возвращает значение в виде вектора из n элементов, содержащих первые производные неивестных функций.

Иногда (особенно в случае нелинейных дифференциальных уравнений) не удается просто определить функцию D. В таких случаях можно попытаться разрешить уравнение относительно $y'(x)$ в символьном виде и подставить это решение в определение для функции D(x,y), используя для этого команду меню Symbolics/Variable/Solve.

3.5.8 Дифференциальные уравнения первого порядка

Дифференциальные уравнения первого порядка не содержат производных выше первого порядка от неизвестной функции.

Пример:

Решить уравнение $y' = \frac{2y}{x} + x$ с начальным условием y(1)=0 на отрезке $[1; 1.5]$.

Зададим начальные условия:

$$y_0 := 0$$

Определим функцию, задающую производную:

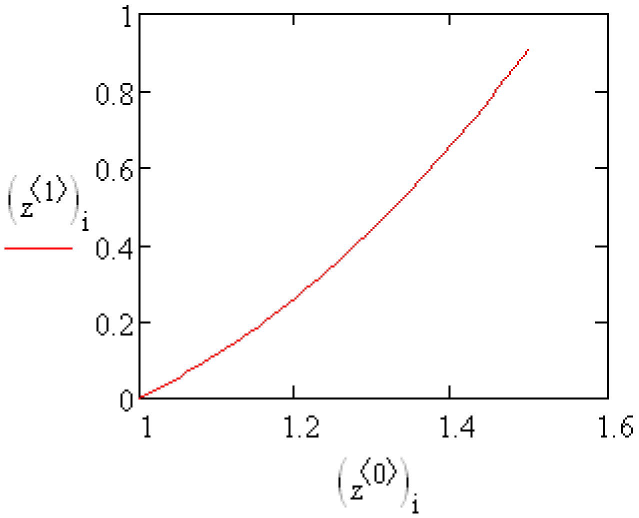
$$D(x,y) := x + \frac{2 \cdot y_0}{x}$$

Находим решение в 50 точках отрезка $[1; 1.5]$:

$$z := \mathrm{rkfixed}(y, 1, 1.5, 50, D)$$

$$i := 0 .. \mathrm{rows}(z) - 1$$

Выводим решение и график решения:



z =

	0	1
0	1	0
1	1.01	0.01
2	1.02	0.021
3	1.03	0.031
4	1.04	0.042
5	1.05	0.054
6	1.06	0.065
7	1.07	0.077
8	1.08	0.09
9	1.09	0.102
10	1.1	0.115
11	1.11	0.129
12	1.12	0.142
13	1.13	0.156
14	1.14	0.17
15	1.15	...

Рисунок 3.29 – Вывод решения и его графика

Далее выводи решение на концах отрезка:

$z_{50,0} = 1.5$

$z_{50,1} = 0.912$

3.5.9 Дифференциальные уравнения второго порядка

Для решения дифференциального уравнения второго порядка необходимо задать:

- Вектор начальных условий состоит из двух элементов: значений функции и ее первой производной в начальной точке интервала.
- Функция D(x,y) – вектор с двумя элементами.
- $D(x, y) = \begin{bmatrix} y'(x) \\ y''(x) \end{bmatrix}$.
- Матрица решения содержит три столбца: первый – значения *x*, в которых ищется решение, второй – *y(x)* и третий – $y'(x)$.

Пример:

Решить дифференциальное уравнение второго порядка $y'' = -y' + 2y$ со следующими начальными условиями y(0)=1, $y'(0) = 3$ на отрезке $[0; 0.5]$.

Решение

$$y := \begin{pmatrix} 1 \\ 3 \end{pmatrix} \qquad D(x,y) := \begin{bmatrix} y_1 \\ (2 \cdot y_0) - y_1 \end{bmatrix}$$

$$Z := \mathrm{rkfixed}(y, 0, 0.5, 400, D)$$

$$n := 0 .. 400$$

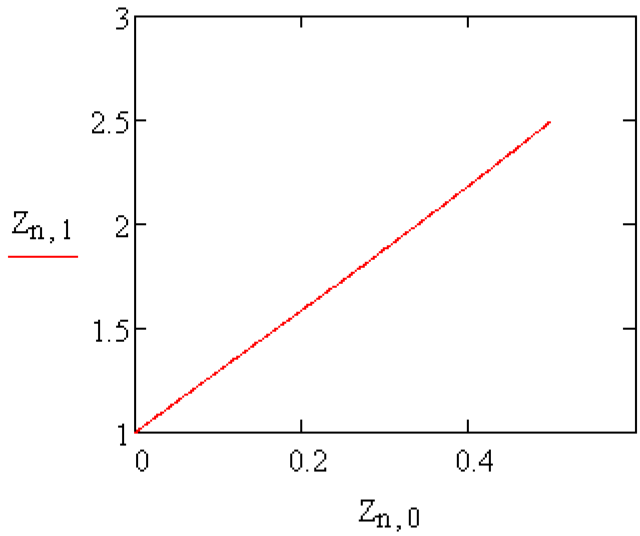


Рисунок 3.30 – Решение дифференциального уравнения второго порядка

3.5.10 Задание 5 для самостоятельной проработки материала

Найти решение дифференциальных уравнений на отрезке $[a, b]$ при следующих начальных условиях.

1. $y' = \frac{x \cdot y}{1+x^2}, y(0) = 2, a = 0, b = 0.3$
2. $y' = y + (1 + x) \cdot y^2, y(0) = 1, a = 0, b = 0.5$
3. $y' = \frac{x^2 \cdot y^2 - (2 \cdot x + 1) \cdot y + 1}{x}, y(1) = 0, a = 1, b = 1.5$
4. $y' = \frac{1 + e^{\frac{x}{y}}}{e^{\frac{x}{y}} \cdot \left(\frac{x}{y} - 1\right)}, y(0) = 1, a = 0, b = 0.3$
5. $y'' + 0.1 \cdot (y')^2 + (1 + 0.1 \cdot x) \cdot y = 0, y(0) = 1, y'(0) = 2, a = 0, b = 0$
6. $y'' + x \cdot y' + y = 0, y(0) = 1, y'(0) = 1, a = 0, b = 1.2$
7. $y'' - x \cdot y' + y = 1 - \cos(x), y(0) = 1, y'(0) = 1, a = 0, b = 0.5$
8. $y'' + x \cdot y' + 2 \cdot y = 12, y(0) = 5, y'(0) = 2, a = 0, b = 0.5$
9. $y'' + x \cdot y' = e^{-x^2}, y(0) = 1, y'(0) = 0, a = 0, b = 0.7$
10. $y'' + y \cdot \cos(x) = 0, y(0) = 1, y'(0) = 0, a = 0, b = 1.2$

3.5.11 Системы дифференциальных уравнений

Для решения системы обыкновенных дифференциальных уравнений первого порядка необходимо:

- Определить вектор, содержащий начальные значения для каждой неизвестной функции.
- Определить функцию, которая возвращает значение в виде вектора из n элементов, содержащих первые производные каждой из неизвестных функций.
- Выбрать точки для поиска приближенного решения.
- Задать функцию rkfixed для этой информации. Функция вернет матрицу, первый столбец которой – значения аргумента, а остальные столбцы содержат значения найденных приближенных решений в соответствующих точках.

Пример:

Решить систему двух ОДУ

$$x_0'(t) = \mu \cdot x_0(t) - x_1(t) - [[x_0(t)]^2 + [x_1(t)]^2] \cdot x_0(t)$$
$$x_1'(t) = \mu \cdot x_1(t) - x_0(t) - [[x_0(t)]^2 + [x_1(t)]^2] \cdot x_1(t)$$

с начальными условиями $x_0(0) = 0 \quad x_1(0) = 1$

Решение

$$x := \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad \mu := -0.2 \quad d(t,x) := \begin{bmatrix} \mu \cdot x_0 - x_1 - \left[(x_0)^2 + (x_1)^2\right] \cdot x_0 \\ \mu \cdot x_1 - x_0 - \left[(x_0)^2 + (x_1)^2\right] \cdot x_1 \end{bmatrix}$$

$$z := \mathrm{rkfixed}(x, 0, 20, 100, d)$$

$$n := 0 .. 100$$

График решения для t=0..20 представлен на рисунке 3.31

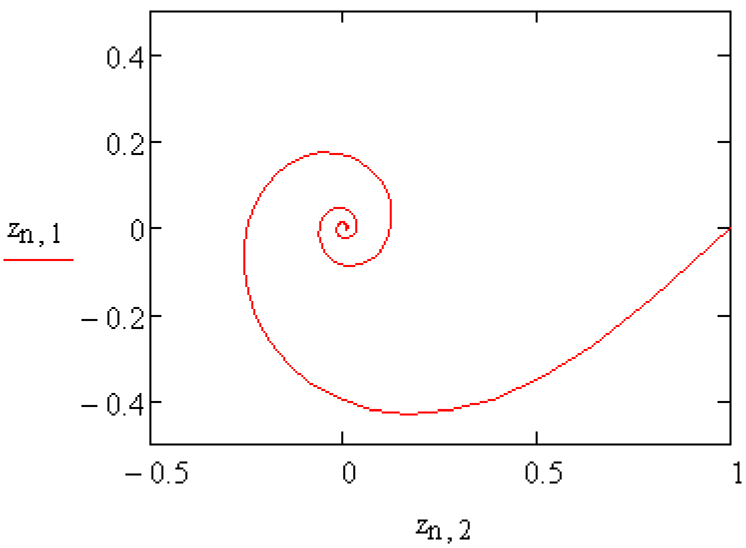


Рисунок 3.31 – Решение системы обыкновенных дифференциальных уравнений

3.5.12 Задание 6 для самостоятельной проработки материала

Решить на отрезке $[a, b]$ системы дифференциальных уравнений со следующими начальными условиями.

1. $y' = y \cdot \cos(x) - z \cdot \sin(x)$ $y(0) = 0,\ z(0) = 0,\ a = 0,\ b = 0.5$
   $z' = y \cdot \sin(x) + z \cdot \cos(x)$
2. $y' = y \cdot x + z$ $y(0) = 0,\ z(0) = 0,\ a = 0,\ b = 0.7$ $z' = y - z$
3. $y' = x + z^2$ $y(0) = 1,\ z(0) = -1,\ a = 0,\ b = 0.7$ $z' = y \cdot x$
4. $y' = y \cdot z + x$ $y(0) = 1,\ z(0) = 0.5,\ a = 0,\ b = 0.5$ $z' = x^2 + y^2$
5. $y' = \cos(y + 2 \cdot z)$ $y(0) = 1,\ z(0) = 2,\ a = 0,\ b = 0.4$ $z' = \frac{2}{4 \cdot y + x} + x + 1$
6. $y' = 2 \cdot x^2 + y$ $y(0) = 1,\ x(0) = 0.5,\ a = 0,\ b = 0.5$ $x' = e^{-(x^2 + y^2)} + 3.5 \cdot z$
7. $z' = y + x$ $y(0) = 0,\ z(0) = 1,\ a = 0,\ b = 0.3$ $y' = x - z^2$
8. $z' = y \cdot z$ $y(0) = 0,\ z(0) = 0.5,\ a = 0,\ b = 0.3$ $y' = y - z$
9. $z' = -z^2 + \frac{2.5}{x}$ $y(0) = 0,\ z(0) = 0.41,\ a = 0,\ b = 0.3$
   $y' = y \cdot z + \frac{\sin(x)}{x}$
10. $x' = \ln(2 \cdot t + \sqrt{9 \cdot t^2 + y^2})$ $x(0) = 1,\ y(0) = 0.5,\ a = 0,\ b = 0.3$
    $y' = \sqrt{9 \cdot t^2 + x^2}$.

4 Программирование в Mathcad

4.1 Введение. О программировании в среде Mathcad

Вплоть до появления 7 версии системы Mathcad возможности программирования в них были крайне ограниченными. Фактически Mathcad позволяла реализовать лишь линейные программы, в основе которых лежит понятие функции. Функция if и ранжированные переменные в отдельных случаях могли заменить условные выражения и циклы, но с серьезными ограничениями. Отсутствовала возможность задания завершенных программных модулей.

Отметим, что возможность составлять программы реализована только в версии Professional. Все эти Mathcad-программы с точки зрения программиста представляют собой подпрограммы-функции, которые могут возвращать в качестве результата число, вектор или матрицу. Функции могут вызывать сами себя (рекурсивно определенные функции) или другие подпрограммы-функции, определенные выше в том же Mathcad-документе.

Mathcad содержит собственные (правда, очень скромные) средства программирования; имеется также возможность встраивания программ, написанных на C++.

Перечислим некоторые особенности программирования в Mathcad.

1. Операторы программы нельзя набирать в ручную. Их вводят щелчком мыши по наименованиям операторов из окна программирования (рис. 4.1). Окно вызывается из меню ***View-Toolbars-Programming***.

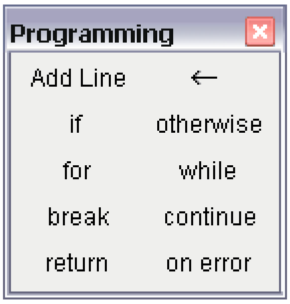
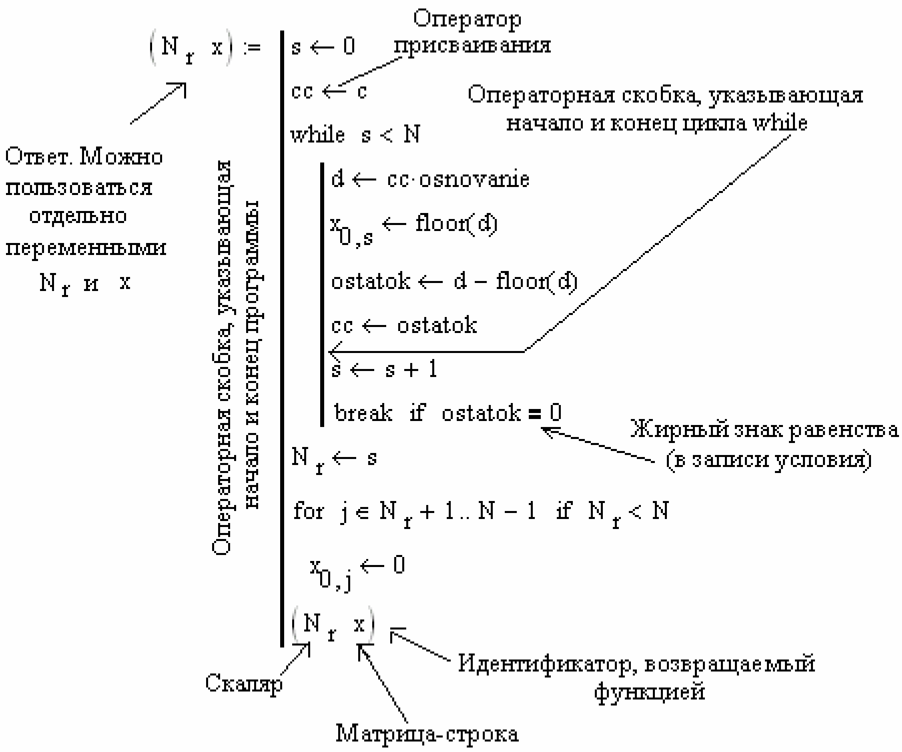


Рисунок 4.1 – Окно программирования в Mathcad

Add Line – добавление линии разделителя операторных скобок (for – конец цикла, if – конец условия и т.п.), break – команда прерывания цикла, ← – символьный знак присваивания

2. Все программы являются функциями (не процедурами!).
3. Переменные, объявленные выше и/или левее текста программы, передаются в программу как значения (by value) и не изменяются, если по ходу программы им присваиваются другие значения (часто это обстоятельство забывается и служит источником ошибок).
4. Программы пишутся с указанием начала и конца операторных скобок (for…, while…, if…); такое указание производится с помощью вертикальных линий.

Пример:



5. В арифметических выражениях используется специфический знак присваивания: стрелка справа налево (←).
6. В логических выражениях и для организации циклов используются:

- знаки отношения:

а) равно – жирный знак равенства (ввод <Ctrl>+<=>);
б) больше (ввод клавишей >);
в) меньше (ввод клавишей <);
г) больше или равно (ввод <Ctrl>+<круглая закрывающая скобка>);
д) меньше или равно (ввод <Ctrl>+<круглая открывающая скобка>);

- знаки логических действий:

условия «И» и «ИЛИ» (ввод знаком умножения (*) и сложения (+)).

7. Функция должна заканчиваться идентификатором возвращаемых переменных (простая переменная, матрица или структура – т.е. некоторые из переменных, указанных в идентификаторе, являются простыми переменными, а другие – векторами или матрицами).
8. Программы могут содержать рекурсии.

Ниже в таблице 4.1 приведены операторы, находящиеся на кнопках панели программирования.

Таблица 4.1 – Команды панели Symbolic (символы)

Команда	Функция	Пример
Add Line	Добавляет новую строку под/над (зависит от выделения) текущей строкой.	
←	Присваивание значения локальной переменной.	$y \leftarrow 0$
if	Условный оператор (оператор ветвления) *if*; условие должно стоять после if, а оператор, который исполняется, если выполнено заданное условие,- перед if.	$f(x) := \left\vert \begin{array}{l} -x \ \text{ if } \ x < 0 \\ x \ \text{ otherwise} \end{array} \right.$ $f(-3) = 3 \quad f(4) = 4$
otherwise	Обозначает оператор, который должен быть исполнен, если условие оператора *if* не выполняется.	$f(x) := \left\vert \begin{array}{l} -x \ \text{ if } \ x < 0 \\ x \ \text{ otherwise} \end{array} \right.$ $f(-3) = 3 \quad f(4) = 4$

for	Цикл *for*; за ключевым словом *for* следует переменная-счетчик, а после символа принадлежности вводится промежуток изменения этой переменной. Внутренние операторы цикла сдвинуты немного вправо.	$\mathrm{Sum}(n) := \left\vert \begin{array}{l} s \leftarrow 0 \\ \mathbf{for}\ \ i \in 1..n \\ \quad s \leftarrow s + i \end{array} \right.$ $\mathrm{Sum}(4) = 10$
while	Цикл *while*; внутренние операторы цикла будут исполняться до тех пор, пока будет истинным условие, следующее за ключевым словом *while*. Пример показывает применение цикла для нахождения нулей функции методом касательных Ньютона.	$N(x, f, f_x) := \mathbf{while}\ \ \lvert f(x) \rvert > 10^{-6}$ $\qquad x \leftarrow x - \dfrac{f(x)}{f_x(x)}$ $N(2, \sin, \cos) = 3.142$
break	Служит для преждевременного завершения цикла, чтобы, например, избежать зацикливания или слишком продолжительных вычислений.	$\mathbf{break}\ \ \text{if}\ \ i \ge 10$
continue	Служит для преждевременного завершения текущей итерации цикла; сам цикл при этом продолжается.	$\mathbf{continue}\ \ \text{if}\ \ x \ge 10$

return	Преждевременное завершение программы; указанное в ячейке значение будет возвращено.	return y
on error	Если при вычислении выражения expr2 возникла ошибка, вычисляется выражение expr1.	expr1 on error expr2

4.2 Техника программирования в Mathcad

При освоении техники программирования (любой, не только в Mathcad) наилучшим способом является решение конкретной задачи и (по его ходу) освоение операторов и приемов. Такой способ имеет особое название DYI («do it yourself» – «делай сам»). Разберем этот способ более подробно.

Для начала рассмотрим этапы, относительно которых будем изучать программирование:

1. Этап 1. Составить требования к входным данным и к форме выдачи результата.
2. Этап 2. Разработать способ получения результата, описанный в литературе, или самому разработать способ получения результата. Рассчитать контрольные примеры, иллюстрирующие разные варианты окончания работы алгоритма.
3. Этап 3. Составить алгоритм и представить его в схематичной форме – блок-схема.
4. Этап 4. Рассчитать контрольные примеры применительно к алгоритму.
5. Этап 5. Составить программу, отладить ее на контрольных примерах.

Необходимо также учесть, что готовую схему или алгоритм вы сразу получите. Не бойтесь делать ошибки, самое главное – во время их обнаружить.

Пример:

Требуется составить программу для перевода правильной десятичной дроби в новую систему счисления и представить результат в слове с заданным числом позиций.

Решение задачи будем разбирать относительно наших этапов.

4.2.1 Этап 1. «Выработка требований к исходным данным и к форме выдачи результата»

Требования к исходным данным. В начале определяем, что в данной задаче представляют собой исходные данные. Согласно заданию в состав исходных данных входят:

1. Основание новой системы счисления, в которую должна переводиться дробь;
2. Число разрядов (если переводится в двоичную систему) или позиций (если система не двоичная), отведенных для представления результата;
3. Правильная десятичная дробь, подлежащая переводу в новую системы счисления. Поскольку программа имеет форму функции, исходные данные укажем в виде абстрактных аргументов (формальных параметров). Примем следующие обозначения:
   - osnovanie – основание новой системы счисления;
   - n_bit – число разрядов или позиций для представления результата;
   - fraction – правильная десятичная дробь, подлежащая переводу в новую систему счисления;
   - Z – идентификатор функции, программу для которой предстоит разработать.

В этих обозначениях функция с аргументами будет записана так:

Z(osnovanie, n_bit, fraction)

«Требования к форме выдачи результата»

Согласно содержательной постановке задачи функция Z(osnovanie, n_bit, fraction) должна возвращать массив, j-й элемент которого представляет собой сомножитель, с которым слагаемое $\text{Разряд}_j \cdot osnovanie^{-j}$ входит в представление дроби fraction в новой системе счисления.

Для того чтобы прочесть и проверить результат, нужно перевести его снова в десятичную форму:

$$\text{Дес}_{\text{знач}} = [\text{разряд}]_1 \cdot [\text{основание}]^{-1} + [\text{разряд}]_2 \cdot [\text{основание}]^{-2} + \cdots + [\text{разряд}]_{n\_bit} \cdot [\text{основание}]^{-n\_bit}$$

Например, пусть задано значение основания osnovanie = 2 новой системы счисления. Пусть десятичная дробь, подлежащая переводу, равна: fraction = 0.625. Пусть число разрядов в слове равно n_bit = 8.

Тогда функция должна возвратить массив – строку (1 0 1 0 0 0 0 0). Для того чтобы прочесть это число (т.е. перевести его в десятичную форму), рассчитываем:

$1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} + 0 \cdot 2^{-4} + 0 \cdot 2^{-5} + 0 \cdot 2^{-6} + 0 \cdot 2^{-7} + 0 \cdot 2^{-8} = 0.625$

Важно помнить, что не всякую десятичную дробь можно точно представить в любой системе счисления, если число позиций для выдачи результата ограничено. Поэтому полезно дополнить состав переменных, возвращаемых функцией, сведениями:

– о погрешности представления дроби fraction в новой системе счисления;

– о числе значащих разрядов (если дробь представима точно, то часть разрядов справа будет нулями). В примере, рассмотренном выше, число значащих разрядов равно 3.

Таким образом, определяем перечень переменных, которые должна возвратить функция Z(osnovanie, n_bit, fraction):

- Число значащих разрядов N_r;
- Массив-строку x из n_bit элементов (столбцов) с результатами перевода дроби в новую систему счисления;
- Погрешность ε от перевода десятичной дроби fraction в новую систему счисления.

Состав возвращаемых переменных неоднороден: часть переменных (N_r, ε) являются простыми переменными, другая часть (x) – массив с индексом $j \in \{0, \dots, n_{bit} - 1\}$. Как известно, смешанные типы данных называются структурами.

На этом этап первый завершен.

4.2.2 Этап 2. «Разбор способа получения результата. Расчет примеров»

Алгоритм перевода десятичной дроби в новую систему счисления состоит из последовательности умножения дроби на новое основание, выделения целой части (она записывается в очередную позицию слова-ответа), вычисления остатка и повторного умножения его на основание – до тех пор, пока (а) либо не будет получен остаток, равный нулю, либо (б) не будут исчерпаны позиции, отведенные для хранения числа.

Нужно составить примеры для обоих вариантов окончания работы алгоритма.

Пример 1. Пусть число разрядов в слове n_bit = 8. В результате вычислений, число 0.78_{10} в двоичной системе представляется не точно: $(0.11000111\dots)_2$. Его дробная часть читается «сверху вниз» в ходе последовательных умножений остатка от вычисления целой части на 2.

Проверка:

$$1 \cdot \frac{1}{2} + 1 \cdot \frac{1}{4} + 0 \cdot \frac{1}{8} + 0 \cdot \frac{1}{16} + 0 \cdot \frac{1}{32} + 1 \cdot \frac{1}{64} + 1 \cdot \frac{1}{128} + 1 \cdot \frac{1}{256} = 0.77734375$$

Погрешность $\varepsilon = (0.78 - 0.77734375) = 0.00265625$

Это же значение можно прочесть и при выводе результата: остаток после последнего шага равен 0.68, номер последнего разряда равен 8, тогда погрешность равна: $\varepsilon = 0.68 \cdot 2^{-8} = 0.00265625$.

Число значащих разрядов получилось равным числу разрядов слова.

Пример2. Перевод правильной десятичной дроби к такой, которая представима в новой системе счисления без погрешности. Переводим 0.625 в двоичную систему. $0.625 \cdot 2 = 1.25,$ остаток $0.25 \cdot 2 = 0.5,$ остаток $0.5 \cdot 2 = 1,$ остаток равен 0. Ответ имеет вид: (0.10100000), число значащих разрядов равно 3, погрешность равна 0.

Проверка:

$$1 \cdot \frac{1}{2} + 0 \cdot \frac{1}{4} + 1 \cdot \frac{1}{8} = 0.625$$

На этом этап второй завершен.

4.2.3 Этап 3. «Составление алгоритма и представление его в схематичной форме»

При разборе алгоритма полезно следить за операциями по примеру (этап 2).

Дано: создать функцию, переводящую заданную десятичную правильную дробь в новую систему счисления с заданным основанием. Число позиций для представления результата задано.

Идентификаторы аргументов:

- osnovanie – основание новой системы счисления;
- n_bit – число позиций для представления результата;
- fraction – правильная десятичная дробь, подлежащая переводу в новую систему счисления.

Получить: функция Z(osnovanie, n_bit, fraction) возвращает структуру $\{N_r \;\; x \;\; \varepsilon\}$.

Комментарии:

- Nr – число значащих разрядов (позиций) в слове длиной n_bit;
- x – матрица-строка с числом элементов, равным n_bit. Элемент $x_j \cdot (osnovanie)^{-(j+1)}$ входит в формулу расчета значения дроби; j=0, …, n_bit – 1 (нумерация элементов матрицы – с нуля, как это принято в Mathcad);
- $\varepsilon -$ погрешность представления правильной десятичной дроби fraction в системе счисления с основанием osnovanie в слове с числом позиций n_bit.

Для ОБРАЩЕНИЯ функции Z(osnovanie, n_bit, fraction)

ЗАДАТЬ числовые значения аргументов (osnovanie, n_bit, fraction);
АЛГОРИТМ вычисления функции Z(osnovanie, n_bit, fraction);
НАЧАЛО расчета значения функции Z(osnovanie, n_bit, fraction).
ПРИСВОИТЬ начальное значение (нуль) счетчику значащих разрядов N_r.
ПРИСВОИТЬ начальное значение (fraction) переменной «c», обозначающий остаток;
ЦИКЛ ПОКА значение числа значащих разрядов $N_r < n\_bit$ ВЫПОЛНЯТЬ

ВЫЧИСЛИТЬ $c \cdot osnovanie$;

ПРИСВОИТЬ результат вычисления переменной d;

ПРИСВОИТЬ элементу массива-строки x_0, N_r[целую часть d];

ВЫЧИСЛИТЬ $ostatok$ как РАЗНОСТЬ между d и [целую часть d];

ПРИСВОИТЬ переменной c значение переменной $ostatok$;

УВЕЛИЧИТЬ на 1 значение счетчика значащих разрядов N_r;

ЕСЛИ $ostatok = 0$ ТО ПРЕРЫВАТЬ ЦИКЛ и ПЕРЕЙТИ к КОМАНДЕ за скобкой КОНЕЦ ЦИКЛА

ПРИСВОИТЬ новое значение $ostatok$ переменной c;

КОНЕЦ ЦИКЛА ПОКА…

ЕСЛИ число значащих разрядов N_r меньше заданного числа n_bit разрядов ТО

Комментарий: разряды, оставшиеся свободными, заполнить нулями.

ЦИКЛ ДЛЯ КАЖДОГО $j = N_r + 1, \dots, n_{bit} - 1$ ВЫПОЛНЯТЬ

ПРИСВОИТЬ нуль $j -$ му элементу массива-строки $x_{0,j}$;

КОНЕЦ ЦИКЛА ДЛЯ КАЖДОГО…

КОНЕЦ ЕСЛИ

ПРИСВОИТЬ переменной ε значение $ostatok \cdot osnovanie^{-n\_bit}$

ПРИСВОИТЬ возвращаемые значения структуре $\{N_r \cdot \varepsilon\}$

КОНЕЦ АЛГОРИТМА

На этом этап третий завершен.

4.2.4 Этап 4. «Расчет контрольных примеров применительно к алгоритму»

В качестве контрольных примеров используем примеры, рассчитанные на этапе 2.

Пусть число разрядов n_bit = 8, основание новой системы счисления $osnovanie = 2$.

Пример 1 (для проверки правильности перевода в новую систему счисления с погрешностью): перевести десятичную правильную дробь 0.78 в двоичную систему счисления.

ФУНКЦИЯ Z(2,8,0.78) должна возвратить:

$Z(2{,}8{,}0.78)_{0,0} = 8$ (число значащих разрядов)

$Z(2{,}8{,}0.78)_{0,1} = (11000111)$ (дробная часть числа 0.78 в двоичной системе счисления)

$Z(2{,}8{,}0.78)_{0,2} = 0.00265625$ (погрешность)

Пример 2 (для проверки правильности перевода в новую систему счисления без погрешности): перевести десятичную правильную дробь 0.65 в двоичную систему счисления.

ФУНКЦИЯ Z(2,8,0.625) должна возвратить:

$Z(2{,}8{,}0.625)_{0,0} = 3$ (число значащих разрядов)

$Z(2{,}8{,}0.625)_{0,1} = (10100000)$ (дробная часть числа 0.625 в двоичной системе счисления)

$Z(2{,}8{,}0.625)_{0,2} = 0$ (погрешность)

На этом этап четвертый завершен.

4.2.5 Этап 5. «Составление программы, отладка на контрольных примерах»

Составление программы будем проводить шаг за шагом. Удобно следить за реализацией алгоритма по этапу 3.

Шаг 1. Вводим идентификатор функции, в скобках перечисляем аргументы, нажимаем знак присваивания (вводится двоеточием). Получается так:

$$Z(\mathrm{osnovanie}, \mathrm{n\_bit}, \mathrm{fraction}) := \blacksquare$$

Шаг 2. На панели «Программирование» (если ее нет на экране – вызовите из Меню *View – Toolbars - Programming*) найдем икону *Add Line* («добавить линию») и щелкнем по ней левой кнопкой мыши. Введется операторная скобка «начало программы – конец программы». Получится так:

$$Z(\mathrm{osnovanie}, \mathrm{n\_bit}, \mathrm{fraction}) := \left| \begin{array}{l} \blacksquare \\ \blacksquare \end{array} \right.$$

Шаг 3. Установим указатель мыши на верхний слот и щелкнем левой кнопкой. Получится так:

$$Z(\mathrm{osnovanie}, \mathrm{n\_bit}, \mathrm{fraction}) := \left| \begin{array}{l} \blacksquare \\ \blacksquare \end{array} \right.$$

Шаг 4. Щелчком левой кнопки мыши вводим с панели программирования знак присваивания. Получится так:

$$Z(\mathrm{osnovanie}, \mathrm{n\_bit}, \mathrm{fraction}) := \left| \begin{array}{l} \blacksquare \leftarrow \blacksquare \\ \blacksquare \end{array} \right.$$

Шаг 5. Пользуясь мышью, присваиваем начальное значение (нуль) счетчику числа значащих разрядов N_r (он же будет использоваться и как счетчик циклов). Необходимо учесть, что *r* – не индекс. Этот параметр вводится косметической точкой (*N* точка *r*). Получится так:

$$Z(\mathrm{osnovanie}, \mathrm{n\_bit}, \mathrm{fraction}) := \left| \begin{array}{l} N_{.r} \leftarrow 0 \\ \blacksquare \end{array} \right.$$

Шаг 6. Аналогичными действиями присваиваем исходное значение переменной *с*, которую согласно алгоритму будем в цикле умножать на основание системы счисления. Исходное значение равно той десятичной дроби, которую мы переводим в новую систему счисления. Получится так:

$$Z(\mathrm{osnovanie}, \mathrm{n\_bit}, \mathrm{fraction}) := \left| \begin{array}{l} N_{.r} \leftarrow 0 \\ c \leftarrow \mathrm{fraction} \end{array} \right.$$

Шаг 7. Длина вертикальной линии (операторной скобки), установленная по умолчанию, исчерпана. Нужно ее увеличить.

Предварительная операция: охватить синим контуром нижнюю строку программы. Получится так:

```
Z(osnovanie,n_bit,fraction) := | N.r ← 0
                               | c ← fraction
```

Шаг 8. Теперь щелчком левой кнопки мыши по иконке *Add Line* на панели «Программирование» удлиним операторную скобку. Получится так:

```
Z(osnovanie,n_bit,fraction) := | N.r ← 0
                               | c ← fraction
                               | ■
```

Шаг 9. Оформим условие цикла: делать вычисления, пока не будут исчерпаны разряды слова (записаны в составе аргументов функции, n_bit). Для этого щелкнем левой кнопкой мыши по оператору *while* на панели «Программирование» (НЕЛЬЗЯ набирать от руки!). Получится так:

```
Z(osnovanie,n_bit,fraction) := | N.r ← 0
                               | c ← fraction
                               | while ■
                               |   ■
```

Шаг 10. Введем условие вхождения в цикл. Получится так:

```
Z(osnovanie,n_bit,fraction) := | N.r ← 0
                               | c ← fraction
                               | while N.r < n_bit
                               |   ■
```

Шаг 11. Образуем операторную скобку операторы цикла *while* (щелчком в слоте оператора *while* введем скобку с панели «Программирование», иконка *Add Line*), установим указатель в верхнем слоте внутри операторной скобки цикла. Получится так:

```
Z(osnovanie,n_bit,fraction) := | N.r ← 0
                               | c ← fraction
                               | while N.r < n_bit
                               |   | ■
                               |   | ■
```

Шаг 12. Согласно алгоритму перевода десятичной дроби в другую систему счисления присвоим переменной d значение произведения дроби на основание, выделим целую часть произведения и присвоим его N_r-му элементу строки x, т.е. элементу с индексом $x_{0,s}$ (напоминание: нумерация элементов массива в Mathcad начинается с нуля, а не с единицы). При первом входе в цикл значение индекса N_r равно 0.

Подготовимся к удлинению операторной скобки (охватим нижнюю строку программы целиком синим контуром). Не забудьте, что r – не индекс (набор через косметическую точку). Получится так:

$$
Z(osnovanie, n\_bit, fraction) := \left|\begin{array}{l} N_{.r} \leftarrow 0 \\ c \leftarrow fraction \\ \text{while } N_{.r} < n\_bit \\ \quad \left|\begin{array}{l} d \leftarrow c \cdot osnovanie \\ x_{0, N_{.r}} \leftarrow floor(d) \end{array}\right. \end{array}\right.
$$

Шаг 13. Чтобы каждый раз не тратить время на удлинение операторной скобки, «нащелкаем» с помощью иконки *Add Line* (панель «Программирование») сразу несколько слотов. Не бойтесь образовывать лишние слоты – их легко будет удалить. Получится так:

$$
Z(osnovanie, n\_bit, fraction) := \left|\begin{array}{l} N_{.r} \leftarrow 0 \\ c \leftarrow fraction \\ \text{while } N_{.r} < n\_bit \\ \quad \left|\begin{array}{l} d \leftarrow c \cdot osnovanie \\ x_{0, N_{.r}} \leftarrow floor(d) \\ \blacksquare \\ \blacksquare \\ \blacksquare \\ \blacksquare \end{array}\right. \end{array}\right.
$$

Шаг 14. Согласно алгоритму следует вычислить остаток (разность между результатом умножения дроби и его целой частью, присваивается переменной *ostatok*) и подготовимся к новому циклу (присвоим цикловой переменной значение $N_r + 1$). Получится так:

```
Z(osnovanie, n_bit, fraction) := | N.r ← 0
                                 | c ← fraction
                                 | while N.r < n_bit
                                 |   | d ← c·osnovanie
                                 |   | x0,N.r ← floor(d)
                                 |   | N.r ← N.r + 1
                                 |   | ■
                                 |   | ■
                                 |   | ■
```

Шаг 15. Оформление досрочного выхода из цикла. Для этого введем щелчком кнопки мыши оператор условия *if* (иконка *if* на панели «Программирование»). Получится так:

```
Z(osnovanie, n_bit, fraction) := | N.r ← 0
                                 | c ← fraction
                                 | while N.r < n_bit
                                 |   | d ← c·osnovanie
                                 |   | x0,N.r ← floor(d)
                                 |   | N.r ← N.r + 1
                                 |   | ■ if ■
                                 |   | ■
                                 |   | ■
```

Шаг 16. Введем условие досрочного прекращения цикла: ЕСЛИ *ostatok=0*. Знак равенства носит название жирное равно и ставится при использовании сочетания клавиш <Ctrl>+<=>. Условие вводится справа от оператора *if* – вот так:

$$
Z(osnovanie, n\_bit, fraction) := \left|
\begin{array}{l}
N_{.r} \leftarrow 0 \\
c \leftarrow fraction \\
\text{while } N_{.r} < n\_bit \\
\quad \left|
\begin{array}{l}
d \leftarrow c \cdot osnovanie \\
x_{0, N_{.r}} \leftarrow \text{floor}(d) \\
N_{.r} \leftarrow N_{.r} + 1 \\
\blacksquare \;\; \text{if } \; ostatok = 0 \\
\blacksquare \\
\blacksquare
\end{array}
\right.
\end{array}
\right.
$$

Шаг 17. Показываем, что должно быть сделано, если *ostatok=0*. В данном случае должен быть приведен цикл (оператор *break*, вводится щелчком по соответствующей иконке на панели «Программирование»). Получится так:

$$
Z(osnovanie, n\_bit, fraction) := \left|
\begin{array}{l}
N_{.r} \leftarrow 0 \\
c \leftarrow fraction \\
\text{while } N_{.r} < n\_bit \\
\quad \left|
\begin{array}{l}
d \leftarrow c \cdot osnovanie \\
x_{0, N_{.r}} \leftarrow \text{floor}(d) \\
N_{.r} \leftarrow N_{.r} + 1 \\
\text{break} \;\; \text{if } \; ostatok = 0 \\
\blacksquare \\
\blacksquare
\end{array}
\right.
\end{array}
\right.
$$

Шаг 18. Если досрочный выход из цикла не произошел, нужно присвоить новое значение остатка *ostatok* переменной *c*. Получится так:

```
Z(osnovanie,n_bit,fraction) := | N.r ← 0
                               | c ← fraction
                               | while N.r < n_bit
                               |   | d ← c·osnovanie
                               |   | x0,N.r ← floor(d)
                               |   | N.r ← N.r + 1
                               |   | break if ostatok = 0
                               |   | c ← ostatok
                               |   | ▪
```

Шаг 19. Дальше текст программы должен записываться вне операторной скобки *while*. Чтобы выйти из нее, охватите синим контуром часть программы до оператора цикла *while* включительно. Угол контура должен быть справа (иначе слот для ввода продолжения программы появится выше *while*). Переброску угла синего контура удобно делать клавишей <Insert>, а увеличение охвата – клавишей <Пробел>. Получится так:

```
Z(osnovanie,n_bit,fraction) := | N.r ← 0
                               | c ← fraction
                               | while N.r < n_bit
                               |   | d ← c·osnovanie
                               |   | x0,N.r ← floor(d)
                               |   | N.r ← N.r + 1
                               |   | break if ostatok = 0
                               |   | c ← ostatok
```

Шаг 20. Теперь, если щелкать левой кнопкой мыши по иконке *Add Line*, то появятся слоты для продолжения программы вне цикла. Заготовим несколько таких слотов. Получится так:

```
Z(osnovanie, n_bit, fraction) := | N.r ← 0
                                 | c ← fraction
                                 | while N.r < n_bit
                                 |   | d ← c·osnovanie
                                 |   | x0,N.r ← floor(d)
                                 |   | N.r ← N.r + 1
                                 |   | break if ostatok = 0
                                 |   | c ← ostatok
                                 | ▪
                                 | ▪
                                 | ▪
```

Шаг 21. Теперь нужно циклически присвоить нули тем разрядам, которые остались незаполненными (если цикл *while* оказался прерванным досрочно, т.е. если для представления дроби *fraction* достаточно меньшего, чем n_bit числа разрядов). Вводим условный оператор и справа указываем условие, при котором следует дописывать незаполненные позиции. Получится так:

```
Z(osnovanie, n_bit, fraction) := | N.r ← 0
                                 | c ← fraction
                                 | while N.r < n_bit
                                 |   | d ← c·osnovanie
                                 |   | x0,N.r ← floor(d)
                                 |   | N.r ← N.r + 1
                                 |   | break if ostatok = 0
                                 |   | c ← ostatok
                                 | ▪ if N.r < n_bit
                                 | ▪
                                 | ▪
```

Шаг 22. Показываем, что должно быть сделано, если $N_r < n\_bit$. Согласно алгоритму следует циклически присваивать нулевые значения разрядам числа, начиная $N_r + 1$-го до последнего разряда (его номер *n_bit-1*, учитывая Mathcad-нумерацию с нуля, а не с единицы). Оператор *for* введем с соответствующей иконки щелчком мыши в слот слева от оператора *if*. Внутреннюю операторную скобку цикла *for* Mathcad оформит сам. Знак «Две точки» ставится с помощью клавиши точка с запятой. Получится так:

$$Z(osnovanie, n\_bit, fraction) := \left| \begin{array}{l} N_{.r} \leftarrow 0 \\ c \leftarrow fraction \\ \text{while } \; N_{.r} < n\_bit \\ \quad \left| \begin{array}{l} d \leftarrow c \cdot osnovanie \\ x_{0,N_{.r}} \leftarrow floor(d) \\ N_{.r} \leftarrow N_{.r} + 1 \\ \text{break } \; \text{if } \; ostatok = 0 \\ c \leftarrow ostatok \end{array}\right. \\ \text{for } \; j \in N_{.r} + 1 .. n\_bit - 1 \; \text{ if } \; N_{.r} < n\_bit \\ \quad x_{0,j} \leftarrow 0 \\ \blacksquare \\ \blacksquare \end{array}\right.$$

В этом случае в цикле – только один оператор, поэтому Mathcad не вводит линию-границу операторной скобки (считает, что и так поймем).

Шаг 23. В ходе вычисление получены данные для определения погрешности (если данная дробь *fraction* не может быть представлена в новой системе точно), см. пример 1. Присваиваем значение погрешности переменной ε, входящей в состав структуры, возвращаемой функцией Z(osnovanie, n_bit, fraction). Показатель степени вводится клавишами <Shift>+<^>. Получится так:

```
Z(osnovanie ,n_bit ,fraction) := | N.r ← 0
                                 | c ← fraction
                                 | while  N.r < n_bit
                                 |     | d ← c·osnovanie
                                 |     | x0,N.r ← floor(d)
                                 |     | N.r ← N.r + 1
                                 |     | break  if  ostatok = 0
                                 |     | c ← ostatok
                                 | for  j ∈ N.r + 1 .. n_bit − 1   if  N.r < n_bit
                                 |   x0,j←0
                                 | ε ← ostatok·osnovanie^(− n_bit)
                                 | ▪
```

Шаг 24 (последний). Все составляющие структуры (N_r x ε), которые должны возвращаться функцией Z(osnovanie, n_bit, fraction), получены. После окончания цикла или после его досрочного прерывания значение переменной N_r окажется равным числу значащих разрядов в слове длины n_bit. Значащие разряды – такие, за которыми:

- либо следуют все нули, если заданная дробь *fraction* может быть представлена без потерь точности меньшим, чем n_bit, числом разрядов;
- либо число разрядов окажется исчерпанным с потерей точности представления дроби *fraction*. В этом случае число значащих разрядов равно *n_bit*.

Рассчитаны также элементы матрицы-строки x с результатом перевода дроби в новую систему счисления и погрешность ε, возникающая при переводе. Осталось записать структуру, возвращаемую функцией.

Оформляем перечень значений, возвращаемых функцией Z(osnovanie, n_bit, fraction). Согласно заданию функция должна возвращать:

- число значащих разрядов (в обозначениях программы N_r);
- массив значений из 8-ми элементов (матрица-строка x), содержащий перечень значений в разрядах числа;
- погрешность ε от представления заданной дроби *fraction* в новой системе счисления с основанием *osnovanie*.

В данном случае нужно, чтобы функция возвратила разнородную структуру данных (в ней и простые переменные, и массив). Такая структура данных так и называется – структура (в отличие от простой переменной или массива).

Для ввода перечня возвращаемых значений в слот программы используем шаблон матрицы. Можно использовать меню *Insert-Matrix*, но лучше запомнить сочетание клавиш <Ctrl>+<M> (мнемоника M от matrix). Указываем число строк (rows) 1, число столбцов (columns) 3 по числу выводимых элементов структуры. В каждый из слотов матрицы вводим идентификаторы переменных. Если остались лишние слоты – удалим их. Получится так:

```
Z(osnovanie ,n_bit ,fraction) := | N.r ← 0
                                 | c ← fraction
                                 | while  N.r < n_bit
                                 |    | d ← c·osnovanie
                                 |    | x0,N.r ← floor(d)
                                 |    | N.r ← N.r + 1
                                 |    | break  if  ostatok = 0
                                 |    | c ← ostatok
                                 | for  j ∈ N.r + 1 .. n_bit − 1   if  N.r < n_bit
                                 |   x0,j ← 0
                                 | ε ← ostatok·osnovanie^(− n_bit)
                                 | (N.r  x  ε)
```

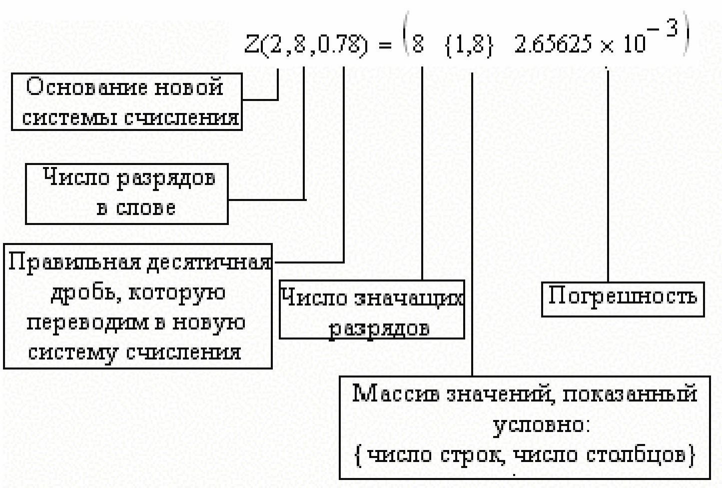
Разработка программы завершена. Мы получили функцию в форме матрицы-строки, первый элемент которой – число значащих разрядов, второй – строка представления заданной дроби в новой системе счисления, третий – погрешность этого представления.

4.2.6 «Тестирование программы на контрольных примерах»

Сначала посмотрим, что возвратит функция при обращении к ней. Набираем:

$$\mathbf{Z}(2, 8, 0.78) = \blacksquare$$

Выведется структура:



Чтобы прочесть позиции в массиве {1,8}, нужно обратиться к соответствующему элементу структуры. Сделаем это в ходе тестирования программы, для чего используем заранее рассчитанные примеры. Для удобства чтения скопируем результаты, которые должна показать программа.

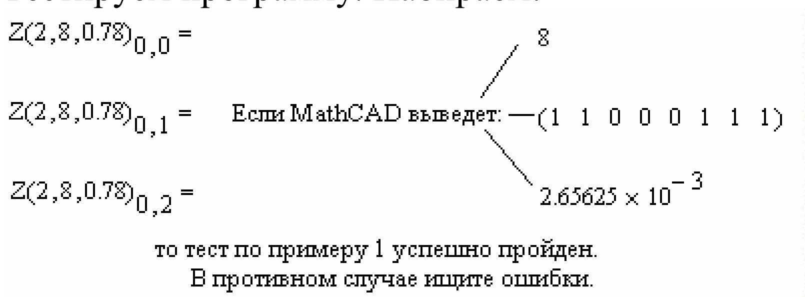
Пример 1 (для проверки правильности перевода в новую систему счисления с погрешностью): перевести десятичную правильную дробь 0.78 в двоичную систему счисления:

ФУНКЦИЯ $Z(2{,}8{,}0.78)_{0,0} = 8$ (число значащих разрядов)

$Z(2{,}8{,}0.78)_{0,1} = (11000111)$ (дробная часть числа 0.78 в двоичной системе счисления)

$Z(2{,}8{,}0.78)_{0,2} = (0.00265625)$ (погрешность)

Тестируем программу. Набираем:



Пример 2 (для проверки правильности перевода в новую систему счисления без погрешности): перевести десятичную правильную дробь 0.625 в двоичную систему счисления.

ФУНКЦИЯ $Z(2{,}8{,}0.625)$ должна возвратить:

$Z(2{,}8{,}0.625)_{0,0} = 3$ (число значащих разрядов)

$Z(2{,}8{,}0.625)_{0,1} = (10100000)$ (дробная часть числа 0.65 в двоичной системе счисления)

$Z(2{,}8{,}0.625)_{0,2} = 0$ (погрешность)

Тестируем программу. Набираем:

$Z(2,8,0.625)_{0,0} =$ 3

$Z(2,8,0.625)_{0,1} =$ Если MathCAD выведет: (1 0 1 0 0 0 0 0)

$Z(2,8,0.625)_{0,2} =$ 0

то тест по примеру 2 успешно пройден.
В противном случае ищите ошибки.

После успешного прохождения тестов этап 5 завершен.

Разработка программы закончена.

4.2.7 Эксперименты с программой

Любопытно, что программа позволяет переводить десятичные дроби в системы счисления с нецелыми основаниями (не только положительными, но и отрицательными). И сама правильная должна десятичная дробь не обязательно должна быть положительной. Почти так всегда и бывает: с помощью программы почти всегда удается сделать больше, чем предусмотрено заданием.

Например, переведем десятичную дробь 0.625 в систему счисления с основанием π. Получим:

$Z(\pi,8,0.625)_{0,1} = (1\ \ 3\ \ 0\ \ 0\ \ 0\ \ 2\ \ 1\ \ 2)$

Проверка:

$$1\cdot\frac{1}{\pi}+3\cdot\frac{1}{\pi^2}+0\cdot\frac{1}{\pi^3}+0\cdot\frac{1}{\pi^4}+0\cdot\frac{1}{\pi^5}+2\cdot\frac{1}{\pi^6}+1\cdot\frac{1}{\pi^7}+2\cdot\frac{1}{\pi^8}=0.6249$$

Можно записать проверку компактнее:

$$\sum_{j=1}^{8}\pi^{-j}\cdot\left[\left(Z(\pi,8,0.625)_{0,1}\right)^{\langle j-1\rangle}\right]_0 = 0.6249$$ Почти 0.625

Так же можно проверить и провести еще какие-либо эксперименты, например: попытаться перевести отрицательную дробь, неправильную дробь, использовать отрицательное основание системы счисления.

4.2.8 Задание 7 для самостоятельной проработки материала

1. Создайте программный блок, вычисляющего факториал при помощи цикла *for*.

2. При помощи алгоритма Евклида определить наибольший общий делитель. Для реализации алгоритма использовать цикл с ключевым словом *while*.

3. Составьте программу-функцию, вычисляющую *n* коэффициентов разложения функции *f(x)*, заданной на отрезке *[-L,L]*, в ряд Фурье – т.е. параметрами являются *f*, *n*, *L*, причем в результате должна выдаваться матрица значений: нулевая по счету строка с коэффициентами A_n, а первая – с коэффициентами B_n. Для выделения этих коэффициентов используйте транспонированную матрицу и из нее будет выбираться нулевой столбец для коэффициентов A_n и первый – для B_n.

4. Разработайте функцию, генерирующую импульсы прямоугольной формы (рис. 4.2). Прямоугольный импульс характеризуется четверкой параметров: {продолжительность импульса; верхний уровень (максимум); нижний уровень (минимум); начальное значение}. Эти параметры, показанные на рисунке 4.2, должны принимать значения, указанные пользователем. Им же задается максимальная продолжительность серии импульсов (на рисунке 4.2. – максимальное число тактов отсчета дискретного времени).

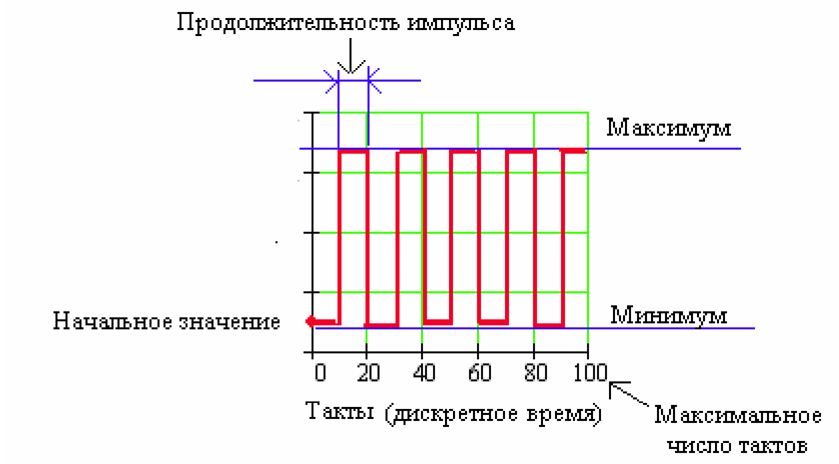


Рисунок 4.2 – Иллюстрация к содержательной постановке задачи

ДАНО:

Xmax – максимальное значение импульса (действительное число);

Xmin – минимальное значение импульса (действительное число);

Δs – продолжительность импульса (число тактов дискретного времени, в течение которого значение импульса не изменяется (целое число);

N – число тактов дискретного времени, для которого должна быть получена серия импульсов;

Xstart – начальное состояние (стартовое значение), задается для первого импульса серии.

ПОЛУЧИТЬ: функция X(Xmax, Xmin, Δs, N, Xstart), возвращает вектор X с элементами X_s, показывающими значение импульса в s-м такте, s = 0,...,N. Значение элемента вектора X_0 должно быть равно заданному аргументу функции X_{start}.

Помощь при выполнении задания.

Позиция 1. «Нахождение идеи алгоритма»

Зададим себе вопрос – чем отличается номер такта дискретного времени, в который происходит «переброска» импульса с минимума на максимум и наоборот, от других номеров тактов, в которых импульс не изменяет своего значения? Пусть, например, заданная пользователем продолжительность импульса составляет 11 тактов, а номер текущего такта 28. Ясно, что число 28 не удается представить как целое число продолжительностей. Поэтому в текущем такте импульс будет иметь то же значение, что и в предшествующем, 27-м.

Если номер текущего такта равен, например, 44, то (поскольку в 44 такта укладывается целое число 11-тактных продолжительностей) нужно изменять импульс с выполнением следующего правила:

1. если на предшествующем такте импульс имел максимальное значение, то его значение нужно заменить на минимальное;
2. если же на предшествующем такте импульс имел минимальное значение, то его значение нужно заменить на максимальное.

Отсюда следует, что можно использовать такой «ключ» переброски импульса с минимума на максимум и наоборот:

Пусть имеет место s-й такт дискретного времени. Тогда, если $\frac{s}{\Delta s}$ – целое число, значение импульса нужно «перебросить», а в противном случае – сохранить прежнее значение.

Позиция 1 закончена.

Позиция 2. «Подбор встроенных Mathcad-функций».

Полный список функций Mathcad можно просмотреть с помощью иконки *f(x)* на строке *Standard* (если эта строка на мониторе не показана – включите ее из меню *View-Toolbars-Standard*). Подходящие функции – округление до целого числа сверху *ceil ()* («до потолка») или снизу *floor()* («от пола»). Обе они могут быть использованы для определения момента переброски значения импульса. Остановимся на функции округления снизу *floor()*. Ее действие легко понять на примере:

$$\mathbf{floor}\left(\frac{18}{11}\right) = 1 \quad \mathbf{floor}\left(\frac{43}{11}\right) = 3 \quad \mathbf{floor}\left(\frac{44}{11}\right) = 4 \quad \mathbf{floor}\left(\frac{4}{11}\right) = 0$$

Мы видим, что если номер такта кратен Δs, то $\frac{s}{\Delta s} = floor(\frac{s}{\Delta s})$, например, $floor\left(\frac{44}{11}\right) = 4$. Для номеров тактов, не кратных Δs, $floor\left(\frac{s}{\Delta s}\right) < \frac{s}{\Delta s}$, например, $floor\left(\frac{43}{11}\right) = \frac{44}{11} = 3 < \frac{43}{11}$.

Это свойство можно использовать как ключ для изменения импульса с минимума на максимум и наоборот.

Позиция 2 закончена.

Позиция 3. «К разработке алгоритма»

ЕСЛИ $\frac{s}{\Delta s} = floor(\frac{s}{\Delta s})$ И значение импульса X_{s-1} в (s-*1*)-м такте было меньше или равно ($\leq$) минимальному значению X_{min} ТО присвоить импульсу X_s в s-м такте максимальное значение;

ЕСЛИ $\frac{s}{\Delta s} = floor(\frac{s}{\Delta s})$ И значение импульса X_{s-1} в (s-*1*)-м такте было больше или равно ($\geq$) максимальному значению X_{max} ТО присвоить импульсу X_s в s-м такте минимальное значение;

В ОТСТАЛЬНЫХ СЛУЧАЯХ присвоить импульсу X_s в s-м такте то же значение X_{s-1}, что и в предыдущем, (*s*-*1*)-м такте.

КОНЕЦ ЕСЛИ

Позиция 3 закончена.

Позиция 4. «Некоторые детали»

1. Запись условий с логическим оператором И делается с использованием знака умножения (*).
2. В записи условий нужно использовать жирный знак равенства.
3. Пример записи условия показан на рисунке 4.3.

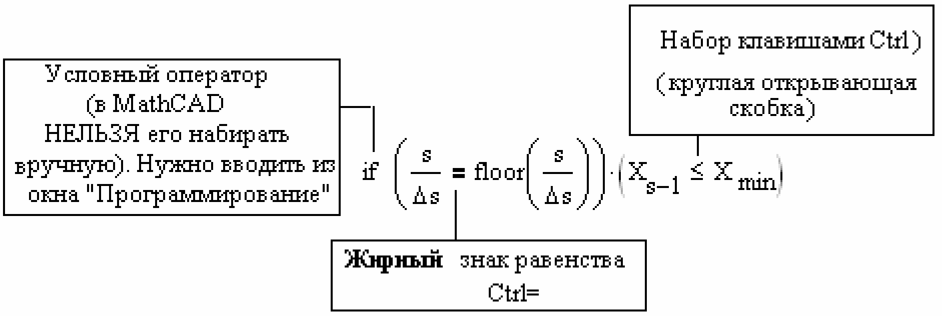
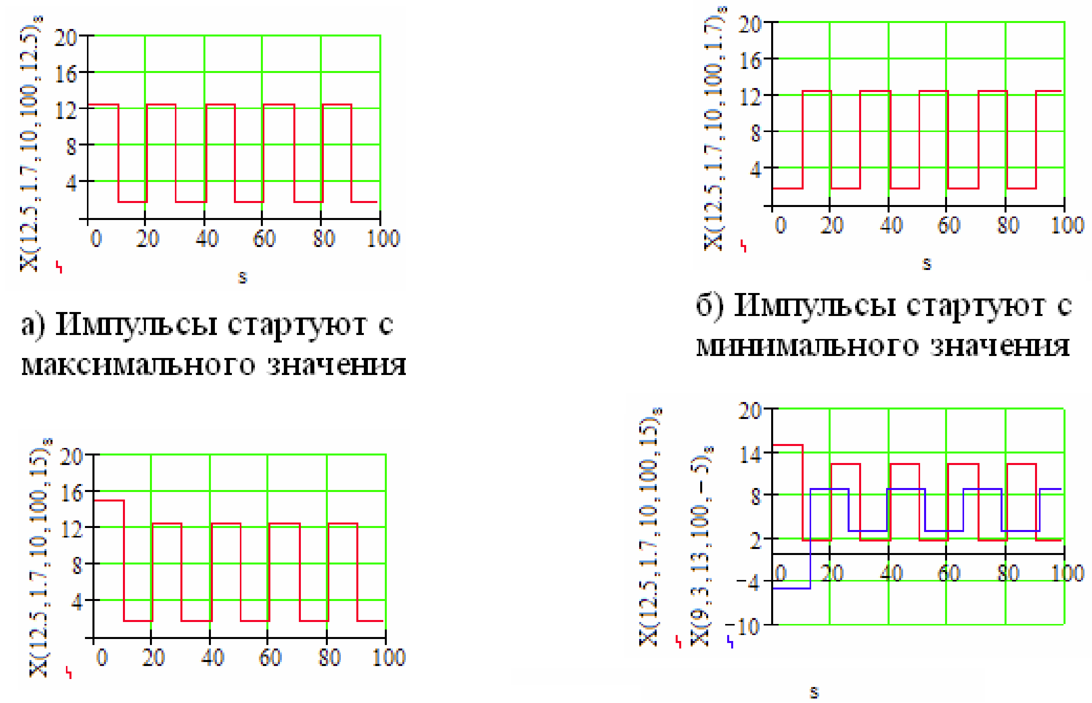


Рисунок 4.3 – Пояснения к записи условий переключения импульсов

Позиция 4 закончена.

Позиция 5. «Иллюстрация графиков»

Так будут выглядеть графики (рис. 4.4), если программа будет работать правильно. Опция графика, выводящего прямоугольные импульсы: *Trace-Type-Step*.



а) Импульсы стартуют с максимального значения

б) Импульсы стартуют с минимального значения

в) Импульсы стартуют со значения, несовпадающего ни с минимальным, ни с максимальным

г) Две серии, показанные на одном графике. Параметры импульсов различны

Рисунок 4.4 – Вывод результата работы программы на график

Позиция 5 закончена.

Помощь при выполнении задания закончена.

5 Символьные преобразования в Mathcad

5.1 Введение. Сведения о символьных преобразованиях в Mathcad

Mathcad содержит замечательные средства работы не только с числами, но и с формулами (т.е. может оказать помощь в упрощении сложных выражений, приведении подобных слагаемых, нахождении пределов, интегрировании, дифференцировании, решении уравнений в виде формулы зависимости от коэффициентов и во многих других вопросах…). Символьная математика, встроенная в Mathcad, заимствована из известного пакета программ для символьных вычислений Maple.

Операции, относящиеся к работе символьного процессора, содержатся в подменю *Symbolics* (Символика) главного меню (рис. 5.1).

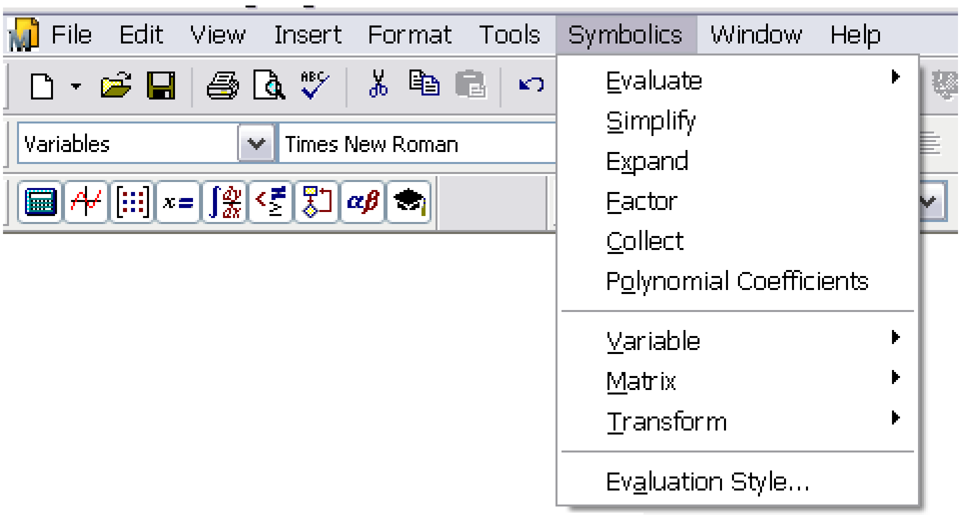


Рисунок 5.1 – Символьный процессор подменю *Symbolics* главного меню

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: ведь и так ясно, что если маркер ввода выделяет переменную какого-либо выражения, то это выражение уже отмечено наличием в нем выделяемой переменной.

Следует отметить некоторые особенности при работе с командами меню *Symbolics*.

Для символьных вычислений выражения необходимо указывать явно. Например, недопустимо вводить некоторую функцию пользователя *f(x)* и пытаться найти ее производные или интеграл. Это существенное ограничение, о котором надо всегда помнить. Однако оно преодолимо при выполнении вычислений с помощью функций системы *SmartMath*, которая описывается в дальнейшем; главное в том, что для вывода символьных вычислений в этом случае используется оператор →.

Иногда результат вычислений содержит встроенные в систему специальные математические функции; в этом случае результат помещается в буфер обмена. Используя команду *Paste* или клавишу *F4*, можно вставить содержимое буфера обмена в документ как ТЕКСТ и проанализировать полученный результат.

К недостаткам работы с командами меню *Symbolics* следует отнести то, что это ручная работа, одношаговая. При дальнейшем использовании результатов символьных вычислений необходимо с помощью операций *Copy* и *Past* присвоить этот результат некоторой переменной или функции. Кроме того, при изменении формулы, которая подвергается символьному преобразованию, результат (даже при установленном Автоматическом режиме вычислений) не пересчитывается.

Если операция невыполнима – система выводит сообщение об ошибке или просто повторяет выделенное выражение (без изменений).

В качестве минуса при работе с командами меню *Symbolics* следует отметить не очень наглядную визуализацию процесса вычислений. Хотя при установке определенных параметров стиля эволюции выражений можно вывести некоторые комментарии.

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

5.1.1 Операции с выделенными выражениями

К операциям с выделенными выражениями относятся следующие (табл. 5.1):

Таблица 5.1 – Операции с выделенными выражениями

Вид операции	Назначение операции
Evaluate (Вычислить)	Преобразовать выражение с выбором вида преобразований из подменю
Simplify (Упростить)	Упростить выделенное выражение с выполнением таких операций, как сокращение подобных слагаемых, приведение к общему знаменателю, использование основных тригонометрических тождеств и т. д.
Expand (Разложить по степеням)	Раскрыть выражение [например, для $(X+ Y)\cdot(X- Y)$ получаем X^2-$У^2$]
Factor (Разложить на множители)	Разложить число или выражение на множители [например, X^2- $У^2$ даст $(X+ Y)$ $(X- Y)$]
Collect (Разложить по подвыражению)	Собрать слагаемые, подобные выделенному выражению, которое может быть отдельной переменной или функцией со своим аргументом (результатом будет выражение, полиномиальное относительно выбранного выражения)
Polynomial Coefficients (Полиномиальные коэффициенты)	Найти коэффициенты полинома по заданной переменной, приближающего выражение, в котором эта переменная использована

5.1.2 Операции с выделенными переменными

Следующая группа символьных операций выполняется с выражениями, требующими указания переменной, по отношению к которой выполняется операция. Для этого достаточно установить на переменной курсор ввода. Само выражение при этом не указывается отдельно, поскольку указание в нем на переменную является одновременно и указанием на само выражение. Если выражение содержит другие переменные, то они рассматриваются как константы.

К числу операций с выделенными переменными относятся (табл. 5.2):

Таблица 5.2 – Операции с выделенными переменными

Вид операции	Назначение операции
Solve (Решить относительно переменной)	Найти значения выделенной переменной, при которых содержащее ее выражение становится равным нулю (решить уравнение или неравенство относительно выделенной переменной)
Substitute (Заменить переменную)	Заменить указанную переменную содержимым буфера обмена
Differentiate (Дифференцировать переменной)	Дифференцировать все выражение, содержащее выделенную переменную, по отношению по к этой переменной (остальные переменные рассматриваются как константы)
Integrate (Интегрировать по переменной)	Интегрировать все выражение, содержащее выделенную переменную, по этой переменной
Expand to Series... (Разложить в ряд)	Найти несколько членов разложения выражения в ряд Тейлора относительно выделенной переменной
Convert to Partial Fraction (Разложить на элементарные дроби)	Разложить на элементарные дроби выражение, которое рассматривается как рациональная дробь относительно выделенной переменной

5.1.3 Операции с выделенными матрицами

Операции с выделенными матрицами представлены позицией подменю *Matrix* (матричные операции), которая имеет свое подменю со следующими операциями:

Transpose (транспонировать) – получить транспонированную матрицу;

Invert (обратить) – создать обратную матрицу;

Determinant (определитель) – вычислить детерминант (определитель) матрицы.

5.1.4 Операции преобразования

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

Fourier Transform (Преобразование Фурье) – выполнить прямое преобразование Фурье относительно выделенной переменной;

Inverse Fourier Transform (Обратное преобразование Фурье) – выполнить обратное преобразование Фурье относительно выделенной переменной;

Laplace Transform (Преобразование Лапласа) – выполнить прямое преобразование Лапласа относительно выделенной переменной (результат – функция от переменной s),

Inverse Laplace Transform (Обратное преобразование Лапласа) – выполнить обратное преобразование Лапласа относительно выделенной переменной (результат – функция от переменной t);

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

Inverse Z (Обратное Z-преобразование) – выполнить обратное Z-преобразование относительно выделенной переменной (результат – функция от переменной *n*).

5.1.5 Стиль эволюции

В меню *Transform* находится команда *Evaluation Style...* (рис. 5.2), при исполнении которой на экран выводится диалоговое окно, позволяющее установить вид вывода результатов при работе с символьным процессором Mathcad.

В группе переключателей *Show envaluation steps* (стиль эволюции) – можно задать вывод результата символьной операции под основным выражением (со вставкой между исходным выражением и результатом пустой строки и без нее) или рядом с ним. Кроме того можно установить опции *Show Comments* для вставки комментариев и задать вывод результирующего выражения вместо исходного *Evaluate in Place*.

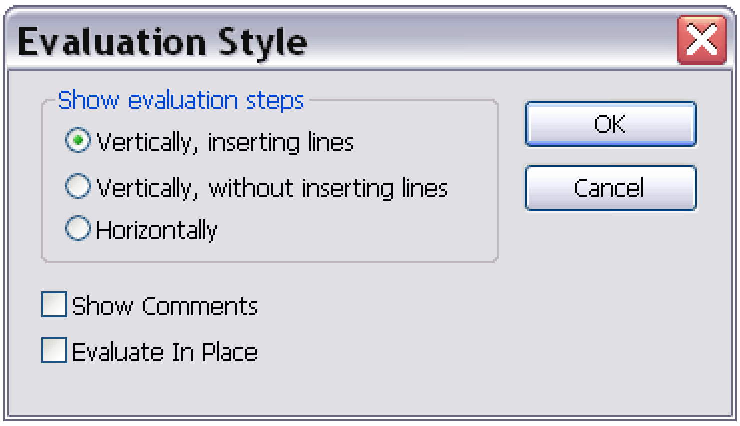


Рисунок 5.2 – Команда *Evaluation Style...* в меню *Transform*

5.2 Осваиваем операции символьной математики

Операции символьной математики проводятся из Меню *Symbolics* (рис. 5.3). Освоим ряд его позиций.

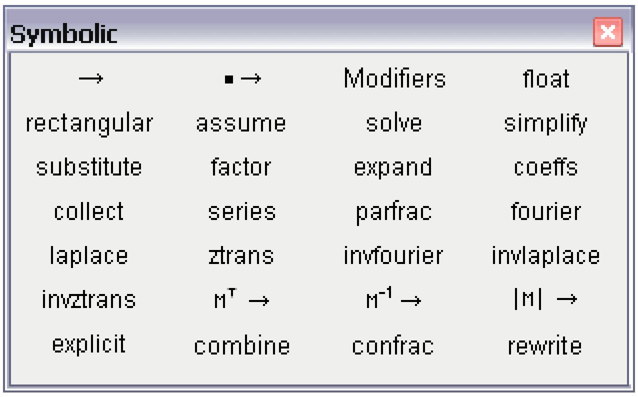


Рисунок 5.3 – Окно символьной математики в Mathcad

5.2.1 Опция Evaluate-Symbolically

Эта операция содержит подменю со следующими командами:

- *Evaluate Symbolically* <Shift>+<F9> (вычислить в символах) – выполнить символьное вычисление выражения;
- *Floating Point Evaluation...* (с плавающей точкой) – выполнить арифметические операции в выражении с результатом в форме числа с плавающей точкой;
- *Complex Evaluation* (в комплексном виде) – выполнить преобразование с представлением в комплексном виде.

Опция *Evaluate-Symbolically* позволяет сделать с формулой то, что Вы сами обычно бы сделали для ее приведения к «красивой форме». Обратите внимание: понятие «красивая формула» – совершенно неформальное. Но, поскольку символьную математику разрабатывали профессиональные научные работники, в понятие «красота формулы» они вложили свои, привычные представления. Например: все подобные слагаемые должны быть приведены; формула должна быть как можно короче; если есть слагаемые с дробными коэффициентами, не представимыми точно десятичной дробью, то ответ нужно представить с коэффициентами в форме дробей и т.п. Поэтому чаще всего (но не всегда!) результат использования опции *Evaluate-Symbolically* будет таким же, как Вы сами бы его получили (если бы умели считать быстро и без ошибок).

Команда *Evaluate-Symbolically* 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 *p* получить 3.141..., используйте команду *Floating Point Evaluation*. В режиме символьных вычислений результат может превосходить машинную бесконечность системы. При этом число точных значащих цифр результата практически не ограничено (или, точнее говоря, зависит от емкости ОЗУ).

Символьная операция *Evaluate Symbolically* <Shift>+<F9> (Вычислить) 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. 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.

Пример:

Вычислить выражение:

$$4 \cdot x + 15 \cdot y^2 + \frac{4}{7} \cdot x + 19 \cdot x + 99 \cdot x^2 - \frac{1}{19} \cdot x$$

Для этого необходимо выбрать опцию *Evaluate-Symbolically* из меню *Symbolics* (или нажать <Shift>+<F9>). Получается:

$$99 \cdot x^2 + \frac{3128 \cdot x}{133} + 15 \cdot y^2$$

Это же выражение можно преобразовать и другим способом: необходимо ввести знак символьного равенства (→) или сочетание клавиш: <Ctrl>+<Shift>+<точка> (клавиша точка в американском варианте называется period).

$$4 \cdot x + 15 \cdot y^2 + \frac{4}{7} \cdot x + 19 \cdot x + 99 \cdot x^2 - \frac{1}{19} \cdot x \rightarrow 99 \cdot x^2 + \frac{3128 \cdot x}{133} + 15 \cdot y^2$$

Преобразование таким образом позволяет добавлять необходимые элементы в вводимое выражение и Mathcad будет проделывать вычисления автоматически.

$$16 \cdot x^2 + 4 \cdot x + 15 \cdot y^2 + \frac{4}{7} \cdot x + 19 \cdot x + 99 \cdot x^2 - \frac{1}{19} \cdot x \rightarrow 115 \cdot x^2 + \frac{3128 \cdot x}{133} + 15 \cdot y^2$$

Знак символьного преобразования носит названия режима *life symbolics*.

Пример:

Пользуясь режимом *live symbolics*, найдите предел, к которому стремится выражение:

$$\left(1 + \frac{1}{x}\right)^x, \text{если } x \rightarrow 0$$

Для ввода предела используем панель Calculus из меню *View-Toolbars*, на появившемся панели найдем кнопку пределов (*lim*) со слотом для ввода имени переменной, по которой ищется предел.

$$\lim_{x \rightarrow 0} \left(1 + \frac{1}{x}\right)^x \rightarrow 1$$

5.2.2 Опция Symbolics-Simplify

Символьная операция *Simplify* (упростить) – одна из самых важных. Эта операция позволяет упрощать математические выражения, содержащие алгебраические и тригонометрические функции, а также выражения со степенными многочленами (полиномами).

Опция *Symbolics-Simplify* позволяет упрощать и приводить к «красивой форме» более сложные выражения, чем алгебраические. В такие выражения могут входить дроби, для сокращения которых нужно искать корни уравнения, элементарные функции (тригонометрические, логарифмические …) и т.п.

Упрощение означает замену более сложных фрагментов выражений на более простые. Приоритет тут отдается простоте функций. К примеру, функция *tan(x)* считается более сложной, чем функции *sin(x)* и *cos(x)*. Поэтому *tan(x)* упрощается так, что получает представление через соотношение этих функций, что несколько неожиданно, так как в некоторых пакетах символьной математики, например *Derive*, ситуация иная: они заменяют отношение *sin(x)/cos(x)* функцией *tan(x)*.

Эта команда открывает широкие возможности для упрощения сложных и плохо упорядоченных алгебраических выражений.

Система Mathcad содержит встроенную функцию для вычисления значений определенных интегралов приближенным численным методом. Ею целесообразно пользоваться, когда нужно просто получить значение определенного интеграла в виде числа. Однако команда *Simplify* применительно к вычислениям определенных интегралов делает гораздо больше – она ищет аналитическое выражение для интеграла. Более того, она способна делать это и при вычислении кратных интегралов, пределы которых – функции.

Операцию *Simplify* можно использовать и для вычисления сумм и произведений символьных последовательностей. Результат операции, как и следовало ожидать, получается в символьной форме (если она существует).

Приведенные примеры могут создать впечатление, что Mathcad лихо справляется со всеми производными, интегралами, суммами и произведениями с помощью операции *Simplify*. К сожалению, это далеко не так. Нередко система не справляется с кажущимися простыми справочными примерами. Надо помнить, что символьный процессор системы Mathcad обладает заметно урезанной библиотекой функций и преобразований (в сравнении с библиотекой системы Maple V). Поэтому часто система не находит решение в замкнутом виде, хотя оно и приводится в справочнике. Тогда система повторяет введенное выражение или сообщает об ошибке.

В результате преобразований могут появляться специальные функции – как встроенные в систему (функции Бесселя, гамма-функция, интеграл вероятности и др.), так и ряд функций, дополнительно определенных при загрузке символьного процессора (интегральные синус и косинус, интегралы Френеля, эллиптические интегралы и др.). Последние нельзя использовать при создании математических выражений.

Пример:

Вычислить выражение:

$$2^{\log(x,2)}$$

$$2^{\log(x,2)}$$

$$x$$

Вычисления проводятся с помощью опции *Simplify* из меню *Symbolics*. *Simplify* переводится как упрощай.

$$2^{\log(x,2)} \text{ simplify } \rightarrow x$$

5.2.3 Опция Symbolics-Expand

Действие операции *Expand* (разложить по степеням) в известном смысле противоположно действию операции *Simplify*. Подвергаемое преобразованию выражение расширяется с использованием известных (и введенных в символьное ядро) соотношений, например алгебраических разложений многочленов, произведений углов и т. д. Разумеется, расширение происходит только в том случае, когда его результат однозначно возможен. Иначе нельзя считать, что действие этой операции противоположно действию операции *Simplify*. К примеру, операция *Simplify* преобразует сумму квадратов синуса и косинуса в 1, тогда как обратное преобразование многозначно и потому в общем виде невыполнимо.

Опция *Symbolics-Expand* позволяет раскрывать скобки в сложных выражениях.

При преобразовании выражений операция *Expand Expression* старается более простые функции представить через более сложные, свести алгебраические выражения, представленные в сжатом виде, к выражениям в развернутом виде и т. д.

Пример:

$$\left(x^2 + a\cdot x + b\right)^3$$

$$a^3\cdot x^3 + 3\cdot a^2\cdot b\cdot x^2 + 3\cdot a^2\cdot x^4 + 3\cdot a\cdot b^2\cdot x + 6\cdot a\cdot b\cdot x^3 + 3\cdot a\cdot x^5 + b^3 + 3\cdot b^2\cdot x^2 + 3\cdot b\cdot x^4 + x^6$$

Для использования режима *live symbolic* Mathcad нужна информация о том, какие из переменных относятся к коэффициентам, а какие обозначают переменные. Поэтому следует при наборе использовать для переменных прописные литеры.

Пример:

$$\left(X^2 + a\cdot X + c\right)^2 \text{ expand } \rightarrow X^4 + 2\cdot X^3\cdot a + X^2\cdot a^2 + 2\cdot X^2\cdot c + 2\cdot X\cdot a\cdot c + c^2$$

Если заменить показатель степени на 7, то получится моментальный результат («вручную» подобный пример займет очень много времени).

5.2.4 Опция Symbolics-Factor

Операция *Factor Expression* (разложить на множители) используется для факторизации – разложения выражений или чисел на простые множители. Она способствует выявлению математической сущности выражений; к примеру, наглядно выявляет представление полинома через его действительные корни, а в том случае, когда разложение части полинома содержит комплексно-сопряженные корни, порождающее их выражение представляется квадратичным трехчленом.

В большинстве случаев (но не всегда) операция факторизации ведет к упрощению выражений. Термин *факторизация* не является общепризнанным в отечественной математической литературе, но мы его оставляем в связи с созвучностью с англоязычным именем этой операции.

Пример: разложить многочлен на множители

$$x^3 - 6 \cdot x^2 + 11 \cdot x - 6$$

$$(x - 3) \cdot (x - 1) \cdot (x - 2)$$

Пример: разложить число на множители

$$123456$$

$$2^6 \cdot 3 \cdot 643$$

5.2.5 Опция Symbolics-Collect

Операция *Collect* (разложить по подвыражению) обеспечивает замену указанного выражения выражением, скомплектованным по базису указанной переменной, если такое представление возможно. В противном случае появляется окно с сообщением о невозможности комплектования по указанному базису.

Эта команда особенно удобна, когда заданное выражение есть функция ряда переменных и нужно представить его в виде функции заданной переменной имеющей вид степенного многочлена. При этом другие переменные входят в сомножители указанной переменной, представленной в порядке уменьшения ее степени.

В том случае, когда комплектование по базису указанной переменной невозможно, система выдает сообщение об этом. Оно выводится в отдельном небольшом информационном окошке.

Пример:

$$(a + b + c)^2$$

$$c^2 + (2\cdot a + 2\cdot b)\cdot c + (a + b)^2 \quad \text{по a}$$

$$b^2 + (2\cdot a + 2\cdot c)\cdot b + (a + c)^2 \quad \text{по b}$$

$$a^2 + (2\cdot b + 2\cdot c)\cdot a + (b + c)^2 \quad \text{по c}$$

5.2.6 Опция Symbolics-Polynomial Coefficients

Операция *Polynomial Coefficients* (полиномиальные коэффициенты) в ранних версиях Mathcad отсутствующая, служит для вычисления коэффициентов полинома.

Операция применяется, если заданное выражение – полином (степенной многочлен) или может быть представлено таковым относительно выделенной переменной. Результатом операции является вектор с коэффициентами полинома. Операция полезна при решении задач полиномиальной аппроксимации и регрессии.

Пример:

$$a\cdot x^3 + b\cdot x^2 + c\cdot x + d$$

$$\begin{pmatrix} d \\ c \\ b \\ a \end{pmatrix}$$

5.2.7 Опция Symbolics-Variable

Опция *Symbolics-Variable* содержит весьма полезные позиции: символьное решение уравнений; символьное дифференцирование; символьное интегрирование и др.

Опция *Symbolics-Variable-Solve* служит для нахождения решений уравнений в символьной форме. Как и в описанных выше позициях, возможно однократное («мертвое») решение путем выделения искомой переменной и использования позиций меню, а также возможен режим *live symbolic*, в котором искомые переменные обозначаются прописными литерами.

Если задано некоторое выражение *f(x)* и отмечена переменная *x,* то операция *Solve* (Решить) возвращает символьные значения указанной переменной *x,* при которых *f(x)=0.* Это очень удобно для решения алгебраических уравнений, например квадратных и кубических, или для вычисления корней полинома.

Ранее отмечалось, что усложнение уравнения, например переход от квадратного уравнения к кубическому, может вызвать и существенное усложнение результата. Тогда система представляет решение в более компактном виде (но без общепринятой математической символики) и предлагает занести его в буфер обмена.

С помощью операции *Paste* (вставить) в позиции *Edit* (правка) главного меню можно перенести решение в основное окно системы, но оно имеет уже тип текстового комментария, а не математического выражения, пригодного для дальнейших преобразований. Впрочем, часть его можно (опять-таки с помощью буфера обмена) ввести в формульные блоки для последующих преобразований и вычислений.

Более того, форма представления результата в таком случае отличается от принятой в системе Mathcad (например, в качестве знака деления используется косая черта и т. д.). Это сделано ради компактности представления результатов вычислений.

В случаях, описанных выше, пользователю надо реально оценить свои силы в упрощении решения. Это придется сделать вручную. При технических расчетах специалист нередко знает, какие из параметров решения несущественны и может отбросить их. Однако для строгих математических расчетов это не всегда возможно, поэтому даже громоздкий результат может быть весьма полезным с познавательной точки зрения.

Пример: решение уравнения

$$ax^2 + b\cdot x + c = 0 \text{ solve}, x \rightarrow \begin{pmatrix} -\dfrac{b - \sqrt{b^2 - 4\cdot a\cdot c}}{2\cdot a} \\ -\dfrac{b + \sqrt{b^2 - 4\cdot a\cdot c}}{2\cdot a} \end{pmatrix}$$

Пример: решение системы уравнений

$$\begin{pmatrix} 2\cdot x^2 + 3\cdot y = 10 \\ 7\cdot x - 19\cdot y = 14 \end{pmatrix} \text{solve}, \begin{pmatrix} x \\ y \end{pmatrix} \rightarrow \begin{pmatrix} \dfrac{\sqrt{35705}}{76} - \dfrac{21}{76} & \dfrac{7\cdot\sqrt{35705}}{1444} - \dfrac{1211}{1444} \\ -\dfrac{\sqrt{35705}}{76} - \dfrac{21}{76} & -\dfrac{7\cdot\sqrt{35705}}{1444} - \dfrac{1211}{1444} \end{pmatrix}$$

Опция *Symbolics-Variable-Substitute* возвращает новое выражение, полученное путем постановки на место указанной переменной некоторого другого выражения. Последнее должно быть подготовлено и скопировано (операциями *Cut* или *Copy*)в буфер обмена. Наряду с получением результата в символьном виде эта команда позволяет найти и числовые значения функции некоторой переменной путем замены ее на числовое значение. Операция *Substitute* переводится как «Заменить переменную».

Подстановки и замены переменных довольно часто встречаются в математических расчетах, что делает эту операцию весьма полезной. Кроме того, она дает возможность перейти от символьного представления результата к числовому.

Пример:

$y - a$ в буфер обмена

$x^3 + 2 \cdot x^2 + 1$ исходное выражение

$3 \cdot a^2 \cdot y - a^3 + 2 \cdot a^2 - 3 \cdot a \cdot y^2 - 4 \cdot a \cdot y + y^3 + 2 \cdot y^2 + 1$ результат

$x^3 + 2 \cdot x^2 + 1$ исходное выражение

2 в буфер обмена

17 результат

Опция *Symbolics-Variable-Differentiate* служит для дифференцирования в символьной форме. Как и в описанных выше позициях, возможно однократное («мертвое») дифференцирование путем выделения переменной, по которой ищется производная, и использования позиций меню, а также возможен режим *live symbolic,* в котором переменная дифференцирования обозначается прописными литерами.

Нахождение символьного значения производной – одна из самых распространенных задач в аналитических вычислениях. Операция *Differentiate* (дифференцировать по переменной) возвращает символьное значение производной выражения по той переменной, которая указана курсором. Для вычисления производных высшего порядка (свыше 1) нужно повторить вычисление необходимое число раз.

Пример: нахождение корней производных

$$\frac{d}{dX} a^X \cdot \sin(X)^2$$

$$\frac{d}{dX} a^X \cdot \sin(X)^2 \rightarrow a^X \cdot \sin(X)^2 \cdot \ln(a)$$

Пример: нахождение производных высших порядков

$$\frac{d^3}{dX^3} a^X \cdot \sin(X) \rightarrow a^X \cdot \sin(X) \cdot \ln(a)^3$$

Опция *Symbolics-Variable-Integrate* служит для интегрирования в символьной форме. Как и в описанных выше позициях, возможно однократное («мертвое») интегрирование путем выделения переменной, по которой ищется интеграл, и использование позиций меню, а также возможен режим *live symbolic*.

Другая не менее важная операция при символьных вычислениях – вычисление интегралов (или нахождение первообразных) для аналитически заданной функции. Для этого используется операция *Integrate* (интегрировать по переменной). Она возвращает символьное значение неопределенного интеграла по указанной курсором ввода переменной. Выражение, в состав которого входит переменная, является подынтегральной функцией.

Как и для операции дифференцирования, в состав исходных выражений и результатов символьного интегрирования могут входить встроенные в систему специальные математические функции.

Пример: нахождение неопределенного интеграла

$$e^{-\alpha\cdot x}\cdot\cos(\beta\cdot x)$$

$$e^{-\alpha\cdot x}\cdot\cos(\beta\cdot x)$$

$$-\frac{e^{-\alpha\cdot x}\cdot(\alpha\cdot\cos(\beta\cdot x)-\beta\cdot\sin(\beta\cdot x))}{\beta^2+\alpha^2}$$

Пример: нахождение определенного интеграла в режиме *live symbolic*

$$\int_0^\infty e^{\frac{-x^2}{2}}\,dx$$

$$\int_0^\infty e^{\frac{-x^2}{2}}\,dx \rightarrow \frac{\sqrt{2}\cdot\sqrt{\pi}}{2}$$

Опция *Symbolics-Variable-Expand to Series...* служит для разложения в ряд Тейлора. Операция *Expand to Series...* (разложить в ряд) возвращает разложение в ряд Тейлора выражения относительно выделенной переменной с заданным по запросу числом членов ряда п (число определяется по степеням ряда). По умолчанию задано n=6 <\p>.

Разложение возможно для функции заданной переменной. В разложении указывается остаточная погрешность разложения. На рисунке представлено применение этой операции для разложения функции *sin(x)/x*. Минимальная погрешность получается при малых *x*.

Символьные операции нередко можно комбинировать для решения сложных задач.

Пример:

$$e^z$$

$$1 + z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24} + \frac{z^5}{120} + \frac{z^6}{720} + \frac{z^7}{5040} \qquad n=8$$

$$1 + z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24} + \frac{z^5}{120} \qquad n=6$$

Опция *Symbolics-Variable-Convert to Partial Fraction* (разложить на элементарные дроби) возвращает символьное разложение выражения, представленное относительно заданной переменной в виде суммы правильных целых дробей.

Пример:

$$\frac{x + a}{x + x^2 + b \cdot x}$$

$$\frac{x}{x + x^2 + b \cdot x} + \frac{a}{x + x^2 + b \cdot x} \qquad \text{по переменной a}$$

$$\frac{a}{x \cdot (b + 1)} - \frac{\frac{a}{b + 1} - 1}{b + x + 1} \qquad \text{по переменной x}$$

5.2.8 Опция Transform-Fourier

Пример:

$$a \cdot t$$

$$-2i \cdot \pi \cdot t \cdot \Delta(1, \omega)$$

$$-2i \cdot \pi \cdot a \cdot \Delta(1, \omega)$$

5.2.9 Опция Transform-Laplace Transform

Пример:

$$1 - e^{\frac{-t}{r}}$$

$$\frac{1}{s \cdot (r \cdot s + 1)}$$

5.2.10 Опция Transform-Z

Пример:

$$a \cdot t$$

$$\frac{a \cdot z}{(z-1)^2}$$

5.2.11 Задание 8 для самостоятельной проработки материала

1. Найдите предел функции:

$$\frac{sin(2x)}{\sin(3x)}\text{ , если } x \to 0$$

2. Упростите выражение:

$\frac{\sin(x)}{tg} \cdot \cos(x) + \sin(x) \cdot \mathrm{tg}(x) \cdot \cos(x) \cdot \frac{1+4\cdot x+4\cdot x^2}{1+2\cdot x}$, $\lim_{x\to\infty}\left[\frac{x+1}{x+3}\right]^{2\cdot x+1}$

3. Раскрыть скобки:

$(x^3 + 7 \cdot x^2 + 3 \cdot x - 4)^4$, $(a \cdot x^2 + b \cdot x + c)^2$

4. Решить уравнение:

$4 \cdot x^2 - 5 \cdot x + 6 = 0$, $3 \cdot \mathrm{x}^2 + 2 \cdot \mathrm{x} - 1 = 0$

5. Решить систему уравнений:

$\begin{pmatrix} 4 \cdot x^3 - 5 \cdot y = 11 \\ 2 \cdot x + 7 \cdot y^2 = -4 \end{pmatrix}$, $\begin{pmatrix} -x + y^2 + 5 \cdot z = -3 \\ 2 \cdot x + 3 \cdot y - z = 2 \\ x^2 - y + 2 \cdot z = -1 \end{pmatrix}$

6. Найти корни производных:

$\frac{d^3}{d(x)^3}\cos(x)$, $\frac{d}{dx}\cos(x)$

7. Используя интегральное преобразование Лапласа (и обратное преобразование Лапласа) решите в символьном виде дифференциальное уравнение:

$$\frac{d}{dt}y(t) - k \cdot y(t) = \sin(t)$$

8. Используя палитру символьных преобразований, найдите точку, в которой функция двух переменных имеет экстремум

$$\frac{d}{dt}y(t) - k \cdot y(t) = \sin(t)$$

9. Вычислить в символьном виде интеграл:

$$x \cdot \int_0^\infty \exp(-c \cdot t)\,dt$$

10. Вычислите коэффициенты полинома по степеням *x* и *y*:

$$x^2 \cdot y^2 + 5 \cdot x^2 \cdot y - 6 \cdot x \cdot y^2 + 7 \cdot x \cdot y - 8 \cdot x + 9 \cdot y - 10$$

11. Исследуйте аналитически функцию *f(x)*: определите точки максимума и минимума, точки перегиба, постройте графики *f(x)*, $f''(x)$, $f'''(x)$

$$f(x) = \frac{2 \cdot x}{x^2 + 1}$$

12. Разложить на множители:
$x^3 - 6 \cdot x^2 + 21 \cdot x - 5,10244785$
13. Разложить по подвыражению:
$(y + z) \cdot (z + y)$по $y,$ по z
14. Разложить полином относительно c
$a \cdot x^3 + b \cdot x^2 + c \cdot x + d$
15. Разложить в ряд Тейлора (n=6)
$\frac{\sin(x)}{x}$
16. Найти преобразование Фурье, обратное преобразование Фурье, преобразование Лапласа, обратное преобразование Лапласа от выражения: $\sin(x)$
17. Найти Z преобразование и обратное Z преобразование от $\frac{1}{x}$.

5.3 Создание анимационных клипов

Начиная с 6-ой версии, в Mathcad появилась возможность создавать анимации. Для ее создания строим график функции (рис. 5.4) командой *X-Y Plot* из подменю *Graph* меню *Insert*. Для анимации задается промежуток изменения целочисленного параметра *FRAME* (по умолчанию от 0 до 9).

Этот параметр должен входить в определение функции, график которой вы хитите пронаблюдать при изменении какого-то параметра (на самом деле вы можете определить свой параметр произвольным образом, лишь бы в нем присутствовал счетчик кадров *Frame*).

$$r := FRAME$$

$$a := 2$$

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

$$F(x) := \frac{(f(x - a \cdot r)) + f(x + a \cdot r)}{2}$$

$$f1(x) := \frac{f(x - a \cdot r)}{2}$$

$$f2(x) := \frac{f(x + a \cdot r)}{2}$$

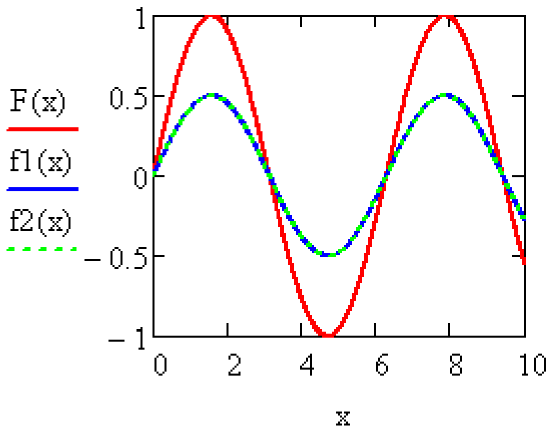
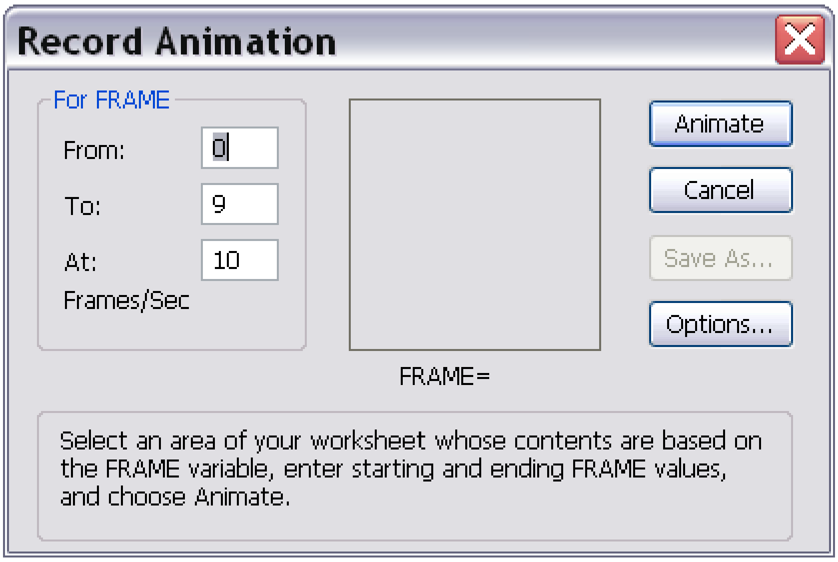


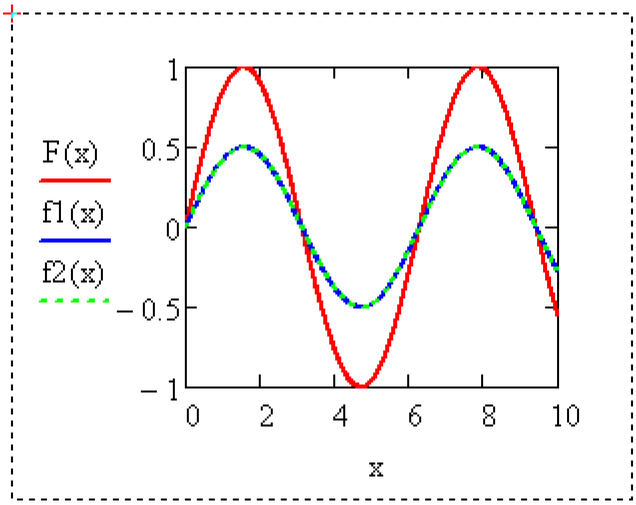
Рисунок 5.4 – График исследуемой функции

Теперь для создания анимации необходимо выполнить следующие действия:

- Выбрать команду *Animation* из меню *Tools* и далее выбрать *Record*…. При этом появится диалоговое окно *Record Animation* (рис. 5.5, а).
- Заключить построенный график в маркировочный прямоугольник, это можно сделать используя левую клавиши мыши (рис. 5.5, б).



а)



б)

Рисунок 5.5 – Работа с окном анимации

а) внешний вид кона анимации, б) исследуемый график функции с выделением

Необходимо учесть, что в разных версиях программы Mathcad это окно можно вызвать выбрав команду *Animate* из меню *View*. При этом появится диалоговое окно *Animate*.

- Задать минимальное и максимальное значения параметра FRAME (поля *From* и *To*).
- Задать в поле *At* количество воспроизводимых кадров в секунду.
- Выполнить щелчок на кнопке *Animate*. При этом в диалоговом окне вы увидите анимационные кадры (рис. 5.6).

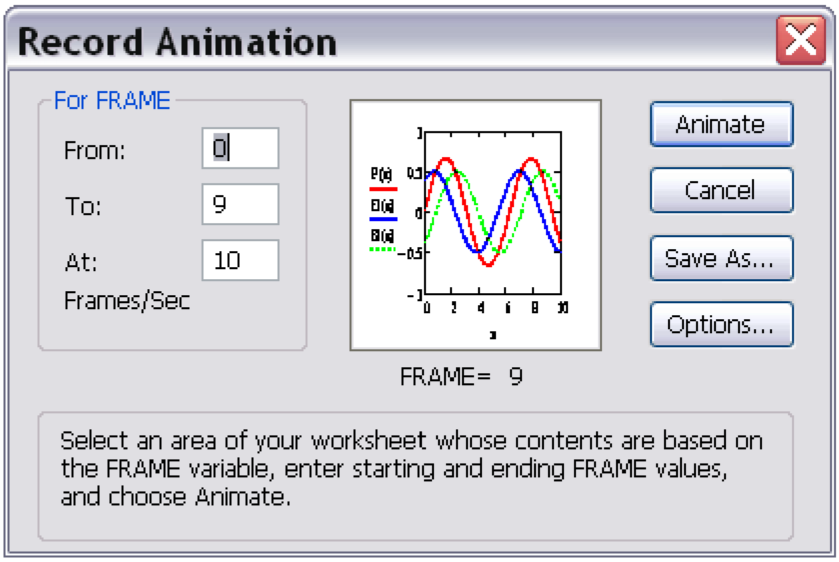
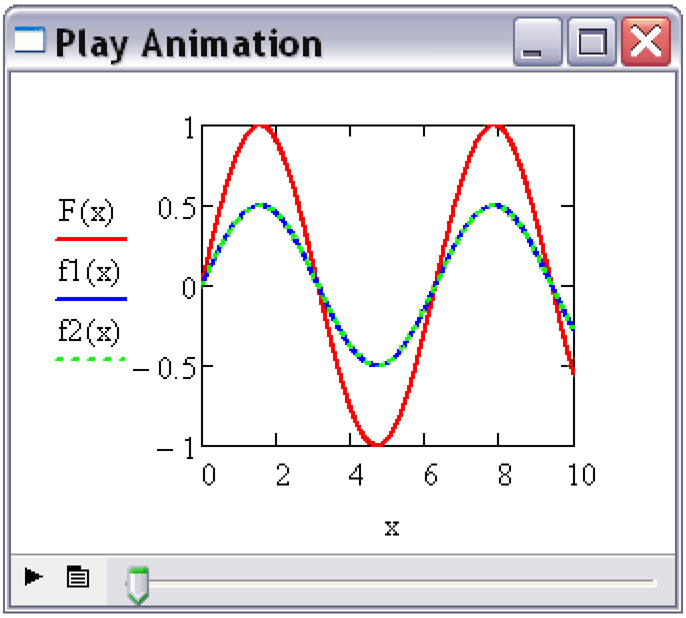


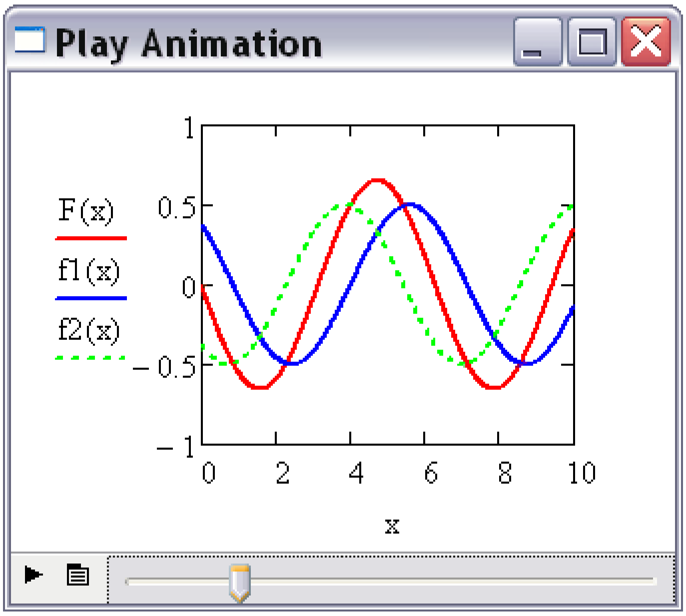
Рисунок 5.6 – Запуск программы с помощью кнопки *Animate*

- Чтобы воспроизвести анимацию щелкните на кнопке *Play* в появившемся окне *Playback* (проигрыватель) (рис. 5.7 а, б, в).
- Чтобы внести изменения в анимацию выполнить щелчок на кнопке открытия меню в окне *Playback*.

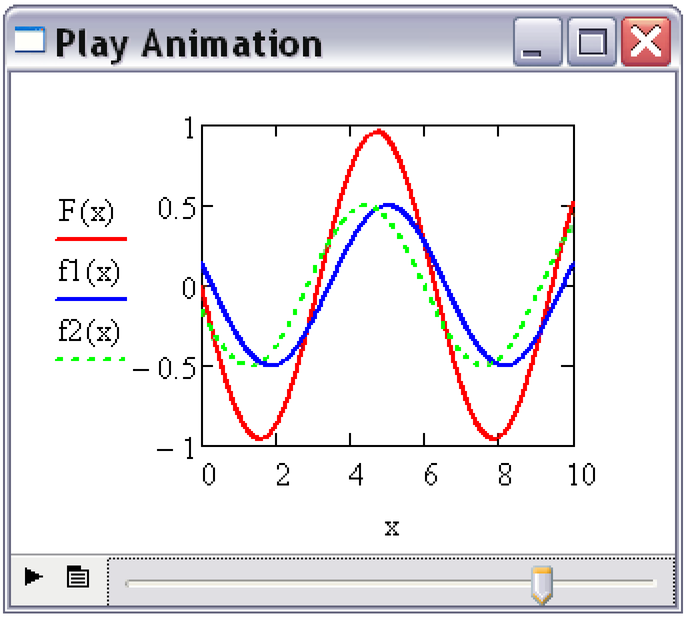
- При помощи команды *Save As* можно сохранить анимацию в файле с расширением AVI.



а)



б)



в)

Рисунок 5.7 – Окно проигрывателя анимации в нескольких режимах переменной FRAME

Встраивание анимации в Mathcad-документ производится при помощи *Windows Explorer*. Для этого необходимо:

- Запустить *Windows Explorer*.
- Выполнить в окне *Windows Explorer* щелчок на имени AVI-файла.
- Перетащить AVI-файл в соответствующий Mathcad-документ.
- Воспроизвести анимацию можно посредством двойного щелчка в графической области.

Анимацию можно также воспроизвести выполнив двойной щелчок на динамически связанной с соответствующим AVI-файлом пиктограмме (рис. 5.8). Для того чтобы встроить такую пиктограмму в Mathcad-документ необходимо:

- Выбрать команду *Object* из меню *Insert*.
- Установить опцию *Создать из файла*.
- Выбрать нужный AVI-файл при помощи кнопки *Обзор*.
- Установить опции *Связь* и *В виде значка*, после чего выполнить щелчок на кнопке *OK*.

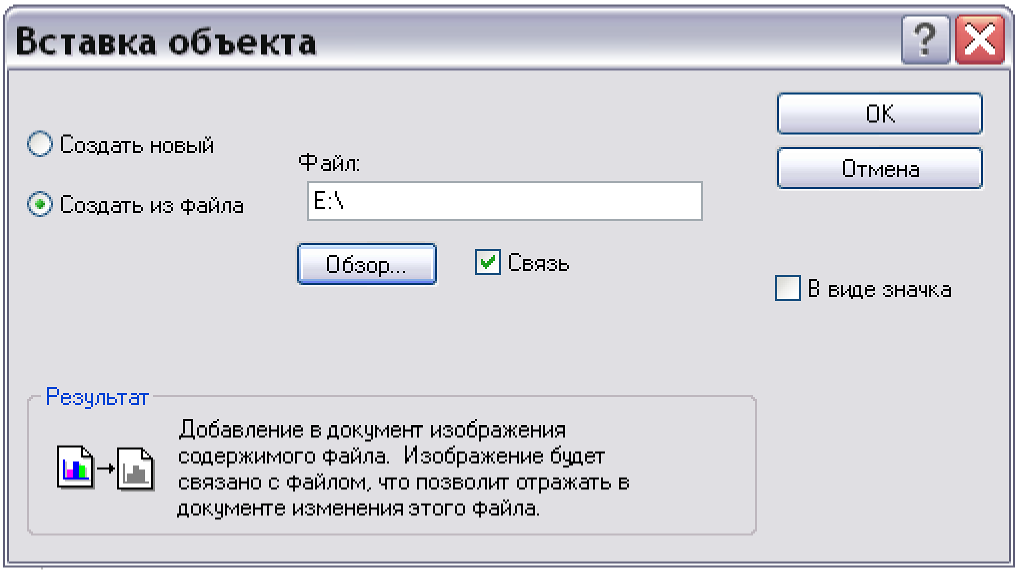


Рисунок 5.8 – Вставка объекта

При создании анимационных картинок надо отключить все опции автоматического масштабирования графиков и перейти к ручному заданию масштаба. Автоматическое изменение масштаба может привести к скачкообразному изменению размеров графика, хотя на деле он должен меняться без скачков, с дискретностью, определяемой только изменением FRAME=0, 1, 2... и т.д.

6 Интерполяция и регрессия, функции сглаживания данных и предсказания

6.1 Функции линейной и сплайновой аппроксимации

К функциям линейной и сплайновой аппроксимации относятся:

- Одномерная линейная аппроксимация;
- Одномерная сплайн-интерполяция и сплайн-аппроксимация
- Двумерная линейная сплайн-интерполяция и сплайн-аппроксимация

6.1.1 Одномерная линейная аппроксимация

При проведении научно-технических расчетов часто используются зависимости вида *y(x)*, причем число точек этих зависимостей ограничено. Неизбежно возникает задача получения приемлемой представительности функций в промежутках между узловыми точками (интерполяция) и за их пределами (экстраполяция). Эта задача решается аппроксимацией исходной зависимости, т.е. ее подменой какой-либо достаточно простой функцией. Система Mathcad предоставляет возможность аппроксимации двух типов: кусочно-линейной и сплайновой.

При кусочно-линейной интерполяции, или аппроксимации, вычисления дополнительных точек выполняются по линейной зависимости. Графически это означает просто соединение узловых точек отрезками прямых, для чего используется следующая функция: *linterp(VX, VY, x)*.

Для заданных векторов *VX* и *VY* узловых точек и заданного аргумента *x* эта функция возвращает значение функции при ее линейной аппроксимации. При экстраполяции используются отрезки прямых с наклоном, соответствующим наклону крайних отрезков при линейной интерполяции.

Пример:

$$m := \begin{pmatrix} 0 & 0 \\ 4 & 4.5 \\ 5 & 7 \\ 7 & 6.5 \\ 10 & 9 \\ 11 & 11 \\ 1.5 & 5 \\ 8.5 & 9.5 \\ 12.5 & 9 \end{pmatrix}$$

$$m := \mathrm{csort}(m, 0)$$

$$X := m^{\langle 0 \rangle} \quad Y := m^{\langle 1 \rangle}$$

$$f(x) := \mathrm{linterp}(X, Y, x)$$

$$scale := 100 \quad j := 0 .. scale$$

$$x_j := \min(X) + j \cdot \frac{\max(X) - \min(X)}{scale}$$

$$i := 0 .. \mathrm{length}(X) - 1$$

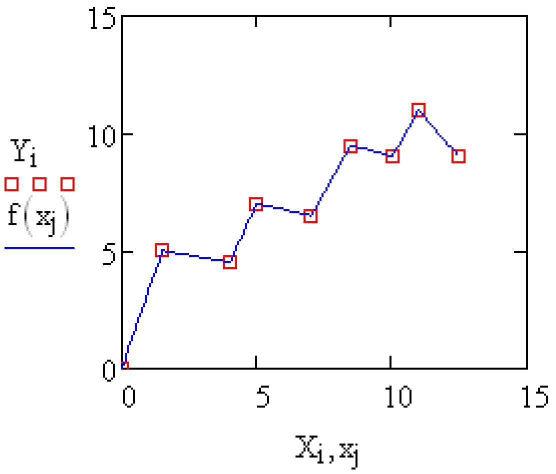


Рисунок 6.1 – Вид кусочно-линейной интерполяции

6.1.2 Одномерная сплайн-интерполяция и сплайн-аппроксимация

При небольшом числе узловых точек (менее 10) линейная интерполяция оказывается довольно грубой. Для целей экстраполяции функция *linterp (VX, VY, x)* не предназначена и за пределами области определения может вести себя непредсказуемо.

Гораздо лучшие результаты дает сплайн-аппроксимация. При ней исходная функция заменяется отрезками кубических полиномов, проходящих через три смежные узловые точки. Коэффициенты полиномов рассчитываются так, чтобы непрерывными были первая и вторая производные.

Для осуществления сплайновой аппроксимации система Mathcad предлагает четыре встроенные функции. Три из них служат для получения векторов вторых производных сплайн-функций при различном виде интерполяции:

- *cspline(VX, VY)* – возвращает вектор *VS* вторых производных при приближении в опорных точках к кубическому полиному;
- *pspline(VX, VY)* – возвращает вектор *VS* вторых производных при приближении к опорным точкам параболической кривой;
- *lspline(VX, VY)* – возвращает вектор *VS* вторых производных при приближении к опорным точкам прямой.
- *interp(VS, VX, VY, x)* – возвращает значение *y (x)* для заданных векторов *VS, VX*, *VY* и заданного значения *x*.

Таким образом, сплайн-аппроксимация проводится в два этапа. На первом с помощью функций *cspline*, *pspline* или *Ispline* отыскивается вектор вторых производных функции *y (x),* заданной векторами *VX* и *VY* ее значений (абсцисс и ординат). Затем, на втором этапе для каждой искомой точки вычисляется значение *y (x)* спомощью функции *interp***.**

Пример:

$$m := \begin{pmatrix} 0 & 0 \\ 4 & 4.5 \\ 5 & 7 \\ 7 & 6.5 \\ 10 & 9 \\ 11 & 11 \\ 1.5 & 5 \\ 8.5 & 9.5 \\ 12.5 & 9 \end{pmatrix} \qquad m := \mathrm{csort}(m,0) \qquad X := m^{\langle 0 \rangle} \quad Y := m^{\langle 1 \rangle}$$

$$S1 := \mathrm{cspline}(X,Y) \qquad f1(x) := \mathrm{interp}(S1,X,Y,x)$$

$$S2 := \mathrm{pspline}(X,Y) \qquad f2(x) := \mathrm{interp}(S2,X,Y,x)$$

$$S3 := \mathrm{lspline}(X,Y) \qquad f3(x) := \mathrm{interp}(S3,X,Y,x)$$

Построение графиков: $\mathrm{scale} := 100 \quad j := 0..\mathrm{scale}$

$$x_j := \min(X) + j \cdot \frac{\max(X) - \min(X)}{\mathrm{scale}} \qquad i := 0..\mathrm{length}(X) - 1$$

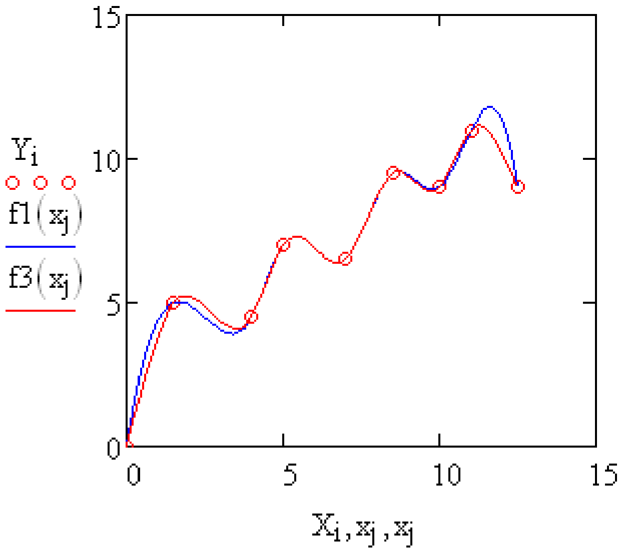


Рисунок 6.2 – Вид одномерной сплайн-интерполяции и сплайн-аппроксимации

6.1.3 Двумерная линейная сплайн-интерполяция и сплайн-аппроксимация

Для повышения качества построения 3D-графиков имеется возможность осуществления двумерной сплайн-интерполяции. Это позволяет существенно повысить представительность сложных графиков SD-функций, в том числе контурных.

Интерполяция функции 2-х переменных проводится также в два этапа:

1. Вычисляется вектор *VS* вторых производных в опорных точках с помощью функций:

- *cspline(Mxy,Mz),*
- *pspline(Mxy,Mz),*
- *lspline(Mxy,Mz)*

Здесь *Mxy* – матрица размера *n*2*, строки которой определяют по диагонали *(x,y)* координаты прямоугольной сетки,

Mz – матрица размера *n*n* значений функции в узлах вышеопределенной сетки.

2. Вычисление с помощью функции *interp(VS,Mxy,Mz,V)*. Здесь *V* – вектор координат *(x,y)*.

На рисунке 6.3 справа показан график функции 2-х переменных после проведения двумерной сплайн-интерполяции, а слева – без нее.

Пример:

$$Mz := \begin{pmatrix} 0.18 & 0.14 & -0.14 & -0.51 & -0.29 & 0.33 \\ 0.93 & 0.17 & -0.76 & -0.99 & -0.31 & -0.83 \\ -0.65 & -0.33 & -0.66 & 0.24 & 0 & 0 \\ -0.55 & -0.22 & 0.47 & 0.74 & -0.11 & 0 \\ -0.98 & 0.17 & 0.37 & 0.81 & 0.39 & -0.87 \\ -0.71 & 0.13 & 0.76 & 0.31 & 0.3 & -0.16 \end{pmatrix}$$

$$n := \mathrm{rows}(Mz)$$

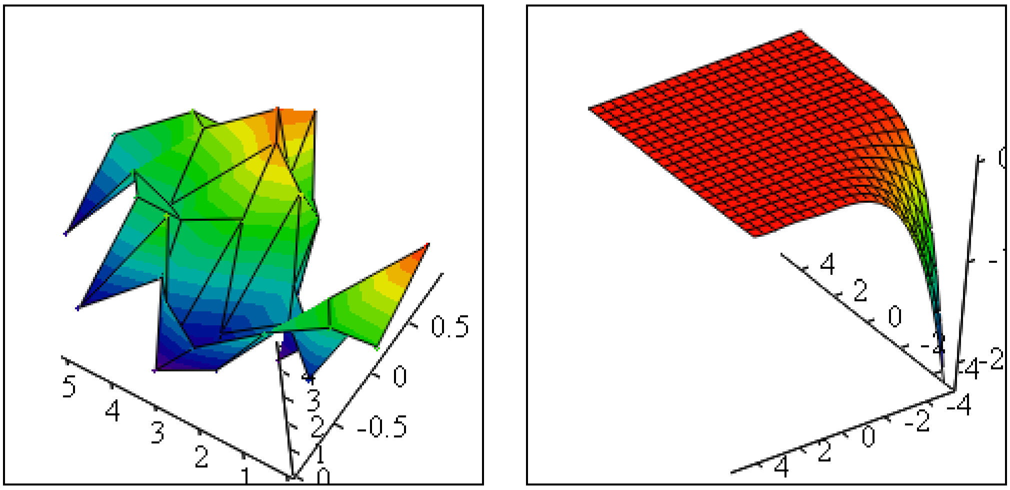
$$X := \begin{pmatrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{pmatrix} \qquad Y := \begin{pmatrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{pmatrix}$$

$$Mxy := \mathrm{augment}(\mathrm{sort}(X), \mathrm{sort}(Y))$$

$$Mxy = \begin{pmatrix} 0 & 0 \\ 1 & 1 \\ 2 & 2 \\ 3 & 3 \\ 4 & 4 \\ 5 & 5 \end{pmatrix}$$

$$S := \mathrm{cspline}(Mxy, Mz)$$

$$f(x,y) := \mathrm{interp}\left[S, Mxy, Mz, \begin{pmatrix} x \\ y \end{pmatrix}\right]$$



Mz f

Рисунок 6.3 – Вид двумерной линейной сплайн-интерполяции и сплайн-аппроксимации

6.2 Функции для проведения регрессии

К функциям для проведения регрессии относятся:

- Функция для линейной регрессии:
- Функция для линейной регрессии общего вида;
- Функции для одномерной и многомерной полиномиальной регрессии;
- Функция для нелинейной регрессии общего вида

6.2.1 Функции для линейной регрессии

Другой широко распространенной задачей обработки данных является представление их совокупности некоторой функцией *y(x)*. Задача регрессии заключается в получении параметров этой функции такими, чтобы функция приближала облако исходных точек (заданных векторами *VX* и *VY*) с наименьшей среднеквадратичной погрешностью.

Чаще всего используется линейная регрессия, при которой функция *y(x)* имеет вид: $y(x) = a + b \cdot x$ и описывает отрезок прямой. К линейной регрессии можно свести многие виды нелинейной регрессии при двупараметрических зависимостях *y(x)*.

Для проведения линейной регрессии в систему встроен ряд приведенных ниже функций:

- corr(VX, VY) – возвращает скаляр – коэффициент корреляции Пирсона;
- *intercrpt(VX, VY)* – возвращает значение параметра *a* (смещение линии регрессии по вертикали);
- *slope(VX, VY)* – возвращает значение параметра *b* (наклона линии регрессии).

Пример:

$$m := \begin{pmatrix} 0 & 0 \\ 4 & 4.5 \\ 5 & 7 \\ 7 & 6.5 \\ 10 & 9 \\ 11 & 11 \\ 1.5 & 5 \\ 8.5 & 9.5 \\ 12.5 & 9 \end{pmatrix}$$

$$m := \mathrm{csort}(m, 0)$$

$$X := m^{\langle 0 \rangle}$$

$$Y := m^{\langle 1 \rangle}$$

$$n := 0 .. \mathrm{rows}(m) - 1$$

$$a := \mathrm{intercept}(X, Y)$$

$$b := \mathrm{slope}(X, Y)$$

$$f(x) := a + b \cdot x$$

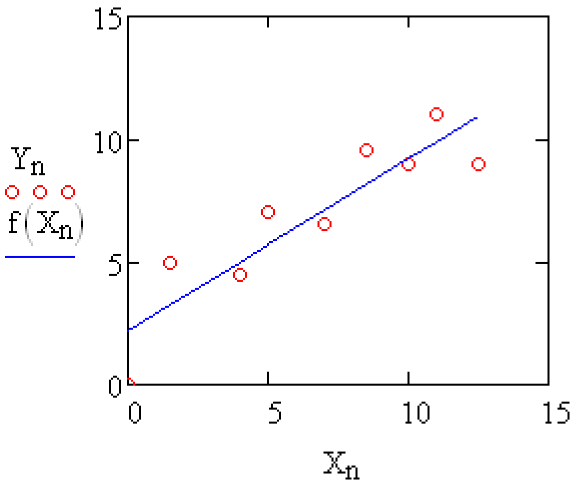


Рисунок 6.4 – Вид функции для линейной регрессии

6.2.2 Функция для линейной регрессии общего вида

В Mathcad реализована возможность выполнения линейной регрессии общего вида. При ней заданная совокупность точек приближается функцией вида:

F(x, K1, K2,..., Kn)=K1·F1(x)+K2·F2(x)+…+Kn·Fn(x).

Таким образом, функция регрессии является линейной комбинацией функций *F1(x), F2(x),..., Fn(x),* причем сами эти функции могут быть нелинейными, что резко расширяет возможности такой аппроксимации и распространяет ее на нелинейные функции.

Для реализации линейной регрессии общего вида используется функция *linfit(VX,VY,F)*. Эта функция возвращает вектор коэффициентов линейной регрессии общего вида *К*, при котором среднеквадратичная погрешность приближения облака исходных точек, если их координаты хранятся в векторах *VX* и *VY*, оказывается минимальной. Вектор *F* должен содержать функции *F1(x), F2(x),..., Fn(x),* записанные в символьном виде.

Расположение координат точек исходного массива может быть любым, но вектор *VX* должен содержать координаты, упорядоченные в порядке их возрастания, а вектор *VY* – ординаты, соответствующие абсциссам в векторе *VX*.

Пример:

$$F(x) := \begin{pmatrix} 1 \\ x \\ 2\cdot x^2 - 1 \end{pmatrix} \qquad S2 := \mathrm{linfit}(X, Y, F) \qquad S2 = \begin{pmatrix} 1.028 \\ 1.347 \\ -0.026 \end{pmatrix}$$

$$g2(t) := F(t)\cdot S2$$

$$t := 0, 0.1 .. 12.5$$

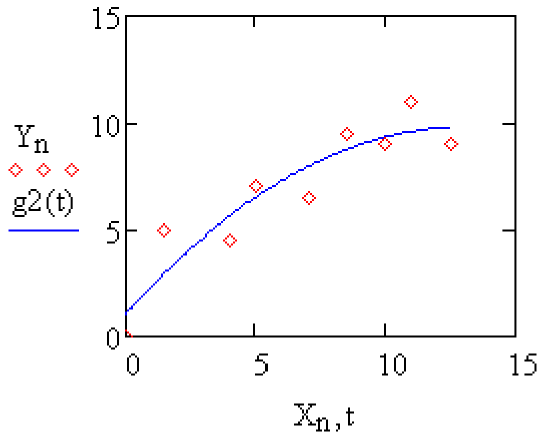


Рисунок 6.5 – Вид функции для линейной регрессии общего вида

6.2.3 Функции для одномерной и многомерной полиномиальной регрессии

Введена в новую версию Mathcad и функция для обеспечения полиномиальной регрессии при произвольной степени полинома регрессии:

regress(VX,VY, n)

Она возвращает вектор *VS,* запрашиваемый функцией *interp(VS,VX,VY,x),* содержащий коэффициенты многочлена *n*-й степени, который наилучшим образом приближает «облако» точек с координатами, хранящимися в векторах *VX* и *VY*.

Для вычисления коэффициентов полинома регрессии используется функция *submatrix*.

На практике не рекомендуется делать степень аппроксимирующего полинома выше четвертой – шестой, поскольку погрешности реализации регрессии сильно возрастают.

Функция *regress* создает единственный приближающий полином, коэффициенты которого вычисляются по всей совокупности заданных точек, т.е. глобально. Иногда полезна другая функция полиномиальной регрессии, дающая локальные приближения отрезками полиномов второй степени, – *loess(VX, VY, span)*. Эта функция возвращает используемый функцией *interp(VS,VX,VY,r)* вектор *VS*, дающий наилучшее приближение данных (с координатами точек в векторах *VX* и *VY*) отрезками полиномов второй степени. Аргумент *span>0* указывает размер локальной области приближаемых данных (рекомендуемое начальное значение – 0,75).

Чем больше *span*, тем сильнее сказывается сглаживание данных. При больших *span*< эта функция приближается к *regress(VX,VY,2)*.

Пример:

$$vs1 := \mathrm{regress}(X, Y, 2)$$

$$f1(x) := \mathrm{interp}(vs1, X, Y, x)$$

$$vs2 := \mathrm{loess}(X, Y, 0.75)$$

$$f2(x) := \mathrm{interp}(vs2, X, Y, x)$$

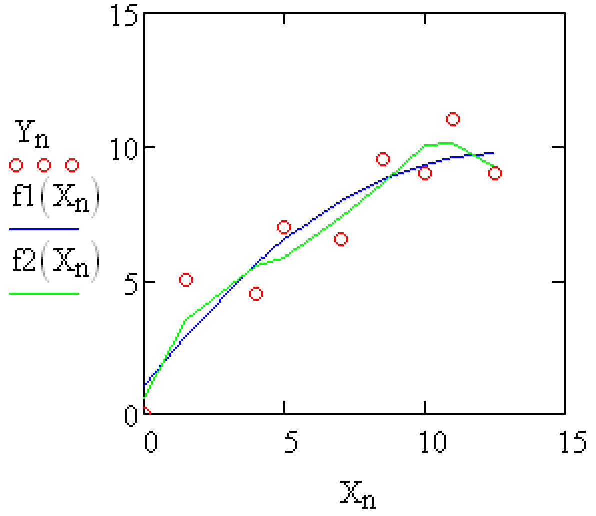


Рисунок 6.6 – Вид функции для одномерной полиномиальной регрессии

Mathcad позволяет выполнять также многомерную регрессию, самый типичный случай которой – приближение трехмерных поверхностей. Их можно характеризовать массивом значений высот *z*, соответствующих двумерному массиву *Mxy* координат точек *(x,y)* на горизонтальной плоскости.

Новых функций для этого не задано. Используются уже описанные функции в несколько иной форме:

- *Regress(Mxy,Vz,n)* – возвращает вектор, запрашиваемый функцией *interp*(*VS,Mxy,Vz,V*) для вычисления многочлена *n*-й степени, который наилучшим образом приближает точки множества *Mxy* и *Vz*. *Mxy* – матрица *mx2,* содержащая координаты *x* и *y*.
- *Vz* – *m*-мерный вектор, содержащий *z*-координат, соответствующих *m* точкам, указанным в *Mxy*;
- *loess(Mxy,Vz,span)* – аналогична *loes(VX,VY,span),* но в многомерном случае;
- *interp (VS,Mxy,Vz,V)* – возвращает значение *z* по заданным векторам *VS* (создается функциями *regress* или *loess*) и *Mxy*, *Vz* и *V* (вектор координат *x* и *y* заданной точки, для которой находится *z*).

Пример:

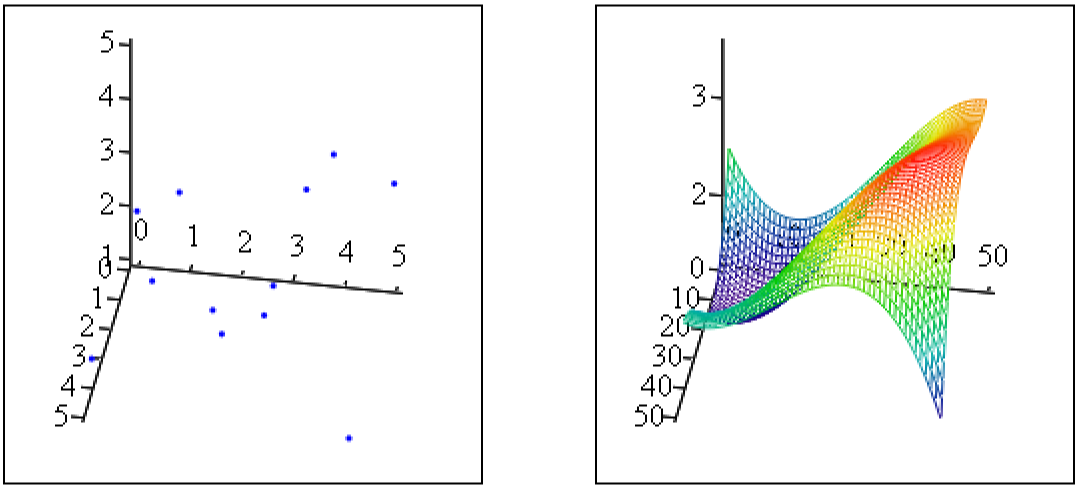
$$Mxy := \begin{pmatrix} 2 & 2 \\ 2 & 3 \\ 3 & 2 \\ 3 & 3 \\ 1 & 1 \\ 1 & 4 \\ 4 & 1 \\ 4 & 4 \\ 0 & 0 \\ 0 & 5 \\ 5 & 0 \\ 5 & 5 \end{pmatrix} \qquad VZ := \begin{pmatrix} 1 \\ 2 \\ 2 \\ 2 \\ 3 \\ 4 \\ 3 \\ 5 \\ 2 \\ 3 \\ 2 \\ 1 \end{pmatrix} \qquad VS := \mathbf{regress}(Mxy, VZ, 3)$$

$$i := 0..50 \qquad j := 0..50$$

$$x_i := \frac{i}{10} \qquad y_j := \frac{j}{10}$$

$$Z3_{i,j} := \mathbf{interp}\left[VS, Mxy, VZ, \begin{pmatrix} x_i \\ y_j \end{pmatrix}\right]$$

$$X := Mxy^{\langle 0 \rangle} \qquad Y := Mxy^{\langle 1 \rangle} \qquad Z := VZ$$

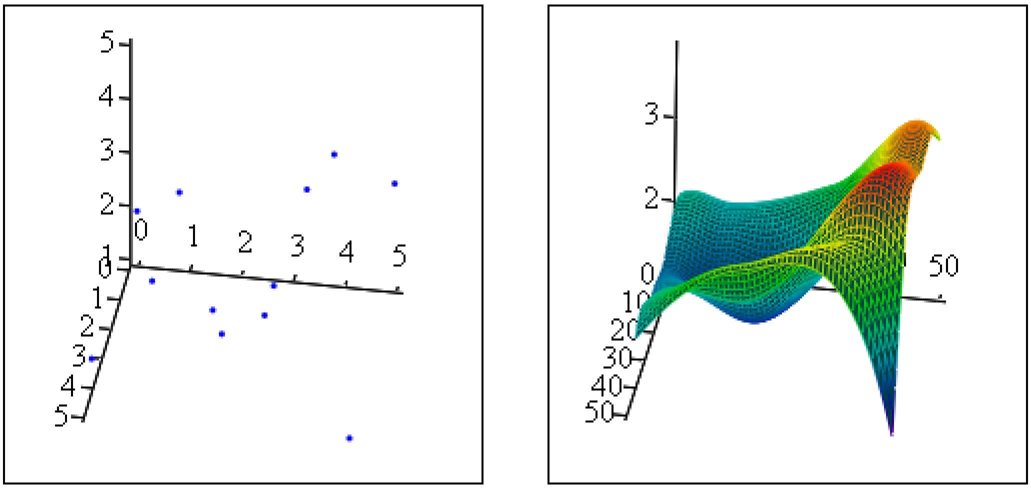


(X ,Y ,Z)   Z3

Рисунок 6.7 – Вид функции *Regress(Mxy, Vz, n)* для многомерной полиномиальной регрессии

$$VS1 := \mathrm{loess}(Mxy, VZ, 0.85)$$

$$Z1_{i,j} := \mathrm{interp}\left[VS1, Mxy, VZ, \begin{pmatrix} x_i \\ y_j \end{pmatrix}\right]$$



(X ,Y ,Z)   Z1

Рисунок 6.8 – Вид функции *loess(Mxy,Vz,span)* для многомерной полиномиальной регрессии

6.2.4 Функция для нелинейной регрессии общего вида

Под нелинейной регрессией общего вида подразумевается нахождение вектора *K* параметров произвольной функции *F(x,K1,K2,...,Kn)*, при котором обеспечивается минимальная среднеквадратичная погрешность приближения облака исходных точек.

Для проведения нелинейной регрессии общего вида используется функция *genfit(VX, VY, VS, F)*. Эта функция возвращает вектор *К* параметров функции *F*, дающий минимальную среднеквадратичную погрешность приближения функцией *F(x, Kl, K2, ..., Kn)* исходных данных.

F должен быть вектором с символьными элементами, содержащими уравнение исходной функции и ее производных по всем параметрам. Вектор *VS* должен содержать начальные значения элементов вектора *К*, необходимые для решения системы нелинейных уравнений регрессии итерационным методом.

Пример:

$$XX := \begin{pmatrix} 1.5 \\ 4 \\ 5 \\ 7 \\ 8.5 \\ 10 \\ 11 \\ 12.5 \end{pmatrix} \qquad YY := \begin{pmatrix} 5 \\ 4.5 \\ 7 \\ 6.5 \\ 9 \\ 11 \\ 5 \\ 9.5 \end{pmatrix}$$

$$ff(x,r,w) := r\cdot\ln(w\cdot x)$$

$$\left(\frac{d}{dr} ff(x,r,w) \rightarrow \right) \ln(w\cdot x)$$

$$\left(\frac{d}{dw} ff(x,r,w) \rightarrow \right) \frac{r}{w}$$

$$F(x,U) := \begin{pmatrix} U_0\cdot\ln(U_1\cdot x) \\ \ln(U_1\cdot x) \\ \dfrac{U_0}{U_1} \end{pmatrix} \qquad VG := \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$k := 0\,..\,\mathrm{rows}(XX) - 1$$

$$U := \mathrm{genfit}(XX, YY, VG, F) \qquad U := \begin{pmatrix} 2.051 \\ 5.288 \end{pmatrix}$$

$$y(t) := U_0\cdot\ln(U_1\cdot t)$$

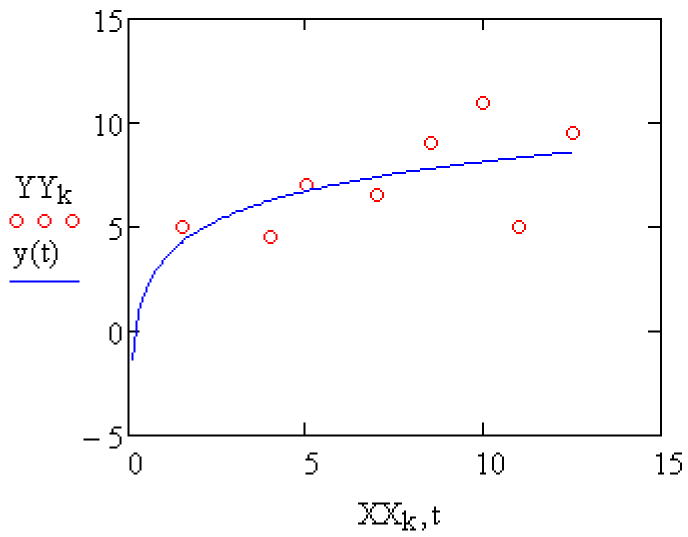


Рисунок 6.9 – Вид функции для нелинейной регрессии общего вида

6.3 Функция сглаживания данных

Данные большинства экспериментов имеют случайные составляющие погрешности. Поэтому часто возникает необходимость статистического сглаживания данных. Ряд функций Mathcad предназначен для выполнения операций сглаживания данных различными методами. Вот перечень этих функций:

- *medsmooth(VY,n)* – для вектора с *m* действительными числами возвращает *m*-мерный вектор сглаженных данных по методу скользящей медианы, параметр *n* задает ширину окна сглаживания (*n* должно быть нечетным числом, меньшим *m*);
- *ksmooth(VX, VY, b)* – возвращает *n*-мерный вектор сглаженных *VY*, вычисленных на основе распределения Гаусса. *VX* и *VY* – *n*-мерные векторы действительных чисел. Параметр *b* (полоса пропускания) задает ширину окна сглаживания (*b* должно в несколько раз превышать интервал между точками по оси *x*);
- *supsmooth(VX, VY)* – возвращает *n*-мерный вектор сглаженных *VY*, вычисленных на основе использования процедуры линейного сглаживания методом наименьших квадратов по правилу *k*-ближайших соседей с адаптивным выбором *k*. *VX* и *VY* – *n*-мерные векторы действительных чисел. Элементы вектора *VX* должны идти в порядке возрастания.

Пример:

$i := 1..50$

$VX_i := i$

$VY_i := rnd(1)$

$n := rows(VY)$

$n = 51$

$SM1 := medsmooth(VY, 9)$

$SM2 := ksmooth(VX, VY, 5)$

$SM3 := supsmooth(VX, VY)$

$i := 0..n-1$

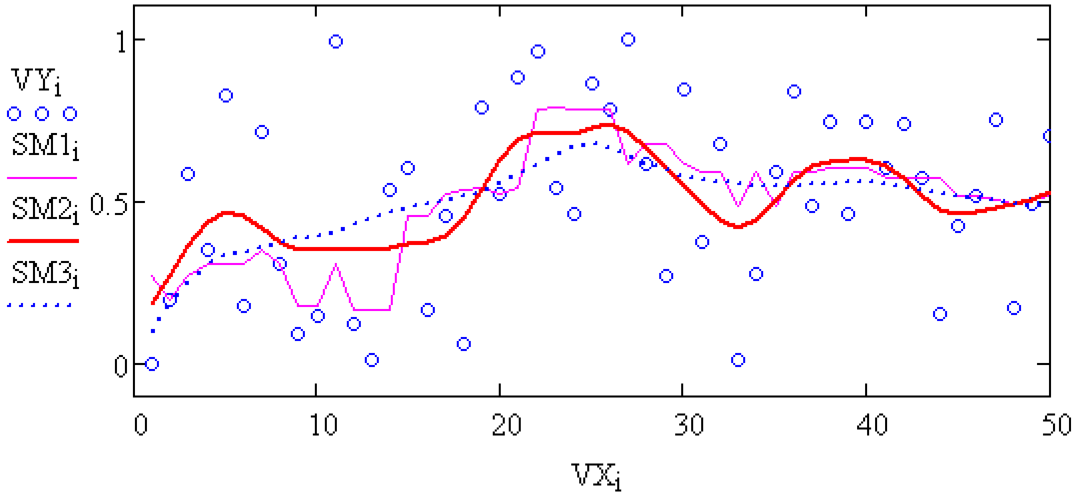


Рисунок 6.10 – Вид для функции сглаживания данных

6.4 Функция предсказания

Весьма интересной является функция предсказания *predikt(data,k,N)*, где *data* – вектор данных, где *data* – вектор данных, *k* – число последних точек существующих данных, на основе которых происходит расчет предсказываемых точек; и *N* – число точек, в которых необходимо предсказать данные. Она по ряду заданных равномерно расположенных точек позволяет рассчитать некоторое число *N* последующих точек, т. е. по существу выполняет экстраполяцию произвольной (но достаточно гладкой и предсказуемой) зависимости.

Функция предсказания обеспечивает высокую точность при монотонных исходных функциях или функциях, представляемых полиномом невысокой степени.

Пример:

$k := 0..100$

$$DAT_k := \exp\left(\frac{-k}{200}\right)\cdot\sin\left(\frac{k}{10}\right)$$

$P := \mathrm{predict}(DAT, 3, 100)$

$i := 0, 5..99$

$n := 100, 105..200$

$$DANN_n := \exp\left(\frac{-n}{200}\right)\cdot\sin\left(\frac{n}{10}\right)$$

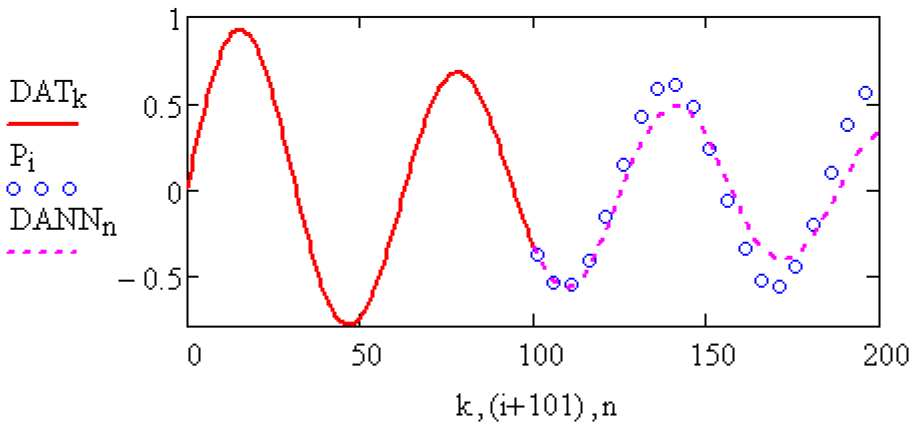


Рисунок 6.11 – Вид для функции предсказания

6.5 Задание 9 для самостоятельной проработки материала

1. Введите матрицу координат точек плоскости:

x	0.1	5	1.1	0.5	3.4	2.7	1.9	4	4.7
y	10	1.3	5	7	3	4.5	6	2.5	1.5

а) Постройте линейную и сплайновую интерполяцию для данного облака точек;

б) Постройте функции линейной, полиномиальной и обобщенной регрессии для этих же точек.

2. С помощью функции *rnd* введите 50 случайных чисел из отрезка [0,2]. Постройте функции сглаживания данных (с помощью различных встроенных функций).

3. С помощью функции *predict* предскажите поведение функции $f(x) = \sin(2 \cdot x)$ на $[2\pi; 4\pi]$, если предположить, что она задана на отрезке $[0; 2\pi]$. Изменится ли предсказанное поведение, если в качестве исходных данных взять функцию, определенную на $[0; \pi]$?

7 Статистические функции

7.1 Статистики совокупностей

Mathcad содержит шесть функций для вычисления статистических оценок случайных совокупностей (таблица 7.1). В последующих описаниях m и n представляют число рядов и столбцов рассматриваемых массивов. В используемых далее формулах переменная ORIGIN по умолчанию принята равной нулю.

Таблица 7.1 – Функции для вычисления статистических оценок случайных совокупностей

№	Вид функции	Описание функции
1	*mean (A)*	Возвращает среднее значение элементов массива **A** размерности $m \times n$ согласно формуле: $$mean(A) = \frac{1}{m \cdot n} \cdot \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} A_{i,j}$$
2	*median(A)*	Возвращает медиану элементов $m \times n$ массива A. Медианой называется величина, выше и ниже которой в вариационном ряду находится равное количество членов. Если A имеет четное число элементов, медиана определяется как среднее арифметическое двух центральных величин.
3	*var(A)*	Возвращает дисперсию элементов массива A размерности $m \times n$ согласно формуле: $$var(A) = \frac{1}{m \cdot n} \cdot \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \left|A_{i,j} - mean(A)\right|^2$$
4	*cvar(A,B)*	Возвращает ковариацию элементов массивов A и B размерности $m \times n$ согласно формуле: $$cvar(A) = \frac{1}{m \cdot n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \left[A_{i,j} - mean(A)\right]$$ $\cdot \left[\overline{B_{i,j} - mean(B)}\right]$, где черта указывает на комплексно-сопряженную величину.
5	*stdev(A)*	Возвращает среднеквадратичное отклонение (квадратный корень из дисперсии) элементов $m \times n$ массива A: $$stdev(A) = \sqrt{var(A)}$$
6	*corr(A,B)*	Возвращает скаляр: коэффициент корреляции для двух $m \times n$ массивов A и B.

7.2 Распределение вероятности

Mathcad использует несколько функций для работы с распространенными плотностями вероятности. Эти функции распадаются на три класса:

- Плотности распределения вероятности: вероятность того, что случайная величина будет находиться в окрестности определенной точки, пропорциональна плотности распределения вероятности случайной величины в этой точке;
- Функции распределения (вероятности): они дают вероятность, что случайная величина будет принимать значение, меньшее или равное определенной величине. Они получены просто интегрированием (или суммированием, когда это необходимо) соответствующей плотности вероятности по подходящему интервалу значений;
- Обращения функций распределения: они позволяют по заданной вероятности вычислить такое значение, что вероятность того, что случайная величина будет меньше или равна этому значению, будет равна вероятности, заданной в качестве аргумента.

Mathcad PLUS поставляется со всеми функциями, перечисленными в следующих трех разделах. Если Вы не используете Mathcad PLUS, Вы будете иметь все функции, связанные со следующими законами распределения вероятности: нормальным, *хи*-квадрат, *t*-распределением Стьюдента, *F*, биномиальным, Пуассона и равномерным.

7.2.1 Плотности распределения вероятности

Эти функции показывают отношение вероятности того, что случайная величина попадает в малый диапазон значений с центром в заданной точке, к величине этого диапазона. Функции плотности вероятности – производные соответствующих функций распределения, обсуждаемых в следующем разделе.

Функции плотности распределения вероятности представлены в таблице 7.2.

Таблица 7.2 – Функции плотности распределения вероятности

№	Вид функции	Описание функции
1	*dbeta*(x,s_1,s_2)	Возвращает плотность вероятности бэта-распределения: $\frac{\Gamma(s_1+s_2)}{\Gamma(s_1)\cdot\Gamma(s_2)}\cdot x^{s_1-1}\cdot(1-x)^{s_2-1}$, где $(s_1, s_2 > 0)$ являются параметрами формы. $(0 < x < 1)$.
2	*dbinom*(k,n,p)	Возвращает P(X = k), когда случайная величина *X* имеет биномиальное распределение: $\frac{n!}{k!\cdot(n-k)!}\cdot p^k\cdot(1-p)^{n-k}$,

		в котором n и k являются целыми числами, удовлетворяющими условию $0 \le k \le n, 0 \le p \le 1$.
3	*dcauchy*(*x*,*l*,*s*)	Возвращает плотность вероятности распределения Коши: $\left(p \cdot s\left(1+\left(\frac{(x-1)}{s}\right)^2\right)\right)^{-1}$, в котором / является параметром расположения, а $s > 0$ есть параметр масштаба.
4	*dchisq*(*x*,*d*)	Возвращает плотность вероятности для *хи*-квадрат распределения: $\frac{e^{-\frac{x}{2}}}{2\Gamma\cdot\left(\frac{d}{2}\right)} \cdot \left(\frac{x}{2}\right)^{\left(\frac{d}{2}-1\right)}$, в котором $d > 0$ является числом степеней свободы, и $x > 0$.
5	*dexp*(*x*,*r*)	Возвращает плотность вероятности экспоненциального распределения: $r \cdot e^{-r \cdot x}$, в котором $r > 0$ является параметром, и $x > 0$.
6	*dF*(*x*,d_1,d_2)	Возвращает плотность вероятности *F*-распределения: $$\frac{d_1^{1.5 \cdot d} \cdot d_2^{1.5 \cdot d} \cdot \Gamma\left(\frac{(d_1+d_2)}{2}\right)}{\Gamma\left(\frac{d_1}{2}\right) \cdot \Gamma\left(\frac{d_2}{2}\right)} \cdot \frac{x^{0.5 \cdot (d_1-2)}}{(d_2+d_1 \cdot x)^{0.5 \cdot (d_1-2)}},$$ в котором $d_1, d_2 > 0$ являются числами степеней свободы и $x > 0$
7	*dgamma*(*x*,*s*)	Возвращает плотность вероятности Гамма-распределения: $$\frac{x^{s-1} \cdot e^{-x}}{\Gamma(s)},$$ в котором $s > 0$ являются параметром формы, и $x \ge 0$.
8	*dgeom*(*k*,*p*)	Возвращает, когда случайная величина *X* подчиняется геометрическому распределению $$p \cdot (1-p)^k,$$ в котором $0 < p \le 1$ является вероятностью успеха в отдельном испытании, k есть неотрицательное целое число.

9	*dlnorm*(*x*,*m*,*s*)	Возвращает плотность вероятности логнормального распределения: $$\frac{1}{\sqrt{2\pi}\cdot\sigma\cdot x}\cdot exp\left[-\frac{1}{2\cdot\sigma^2}\cdot(ln(x)-\mu)^2 right\right],$$ в котором m равно натуральному логарифму среднего значения, $s > 0$ равно натуральному логарифму среднеквадратичного отклонения, и $x > 0$.
10	*dlogis*(*x*,*l*,*s*)	Возвращает плотность вероятности логистического распределения: $$\frac{exp\left(-\frac{(x-l)}{s}\right)}{s\cdot\left(1+exp\left(-\frac{(x-l)}{s}\right)\right)^2},$$ в котором / является параметром расположения, и $s > 0$ есть параметр масштаба.
11	*dnbinom*(*k*,*n*,*p*)	Возвращает *P(X=k)*, когда случайная величина *X* имеет отрицательное биномиальное распределение: $$\binom{n-k-1}{k}\cdot p^k\cdot(1-p)^{n-k},$$ в котором $0 < p \leq 1,$ а n и k являются целыми числами, $n > 0$ и $k > 0$.
12	*dnorm*(*x*,*m*,*s*)	Возвращает плотность вероятности нормального распределения: $$\frac{1}{\sqrt{2\pi}\cdot\sigma\cdot x}\cdot exp\left[-\frac{1}{2\cdot\sigma^2}\cdot(x-\mu)^2 right\right],$$ в котором m и s есть среднее значение и среднеквадратичное отклонение. $s > 0$.
13	*dpois(k,l)*	Возвращает *P(X=k)*, когда случайная величина *X* имеет распределение Пуассона: $$\frac{\lambda^k}{k!}\cdot e^{-\lambda},$$ в котором $l > 0,$ а k является неотрицательным целым числом.
14	*dt*(*x*,*d*)	Вычисляет плотность вероятности *t* - распределения Стьюдента:

		$$\frac{\Gamma\left(\frac{(d+1)}{2}\right)}{\Gamma\left(\frac{d}{2}\right)\cdot\sqrt{\pi\cdot d}}\cdot\left(1+\frac{x^2}{d}\right)^{-0.5\cdot(d+1)},$$ в котором d является числом степеней свободы, $d > 0,$ а x есть вещественное число.
15	*dunif*(x,a,b)	Вычисляет плотность вероятности равномерного распределения: $$\frac{1}{b-a},$$ в котором b и a являются граничными точками интервала, $a < b$ и $a \leq x \leq b$.
16	*dweibull*(x,s)	Вычисляет плотность вероятности распределения Вейбулла: $$s\cdot x^{s-1}\cdot\exp(-x^s),$$ в котором $s > 0$ есть параметр формы и $x > 0$.

7.2.2 Функции распределения

Эти функции возвращают вероятность того, что случайная величина меньше или равна определенному значению. Функция распределения вероятности $-$ просто функция плотности вероятности, проинтегрированная от $-\infty$ до определенного значения. Для целочисленных случайных величин интеграл заменен суммированием по соответствующим индексам.

Рисунок 7.1 иллюстрирует связь между плотностью вероятности и функцией распределения случайной величины.

Функции распределения представлены в таблице 7.3.

Таблица 7.3 – Функции распределения

№	Вид функции	Описание функции
1	*cnorm(x)*	Возвращает стандартную нормальную функцию распределения. Эквивалент *pnorm(x,0,1)*.
2	*pbeta(x,*s*1,*s*2)*	Возвращает функцию *бэта*-распределения с параметрами формы *s*1 и *s*2. (*s*1,*s*2>0).
3	*pbinom(k,n,p)*	Возвращает функцию биномиального распределения для *k* успехов в *n* испытаниях: *n* есть натуральное число; *p* есть вероятность успеха, $0 \leq p \leq 1$.
4	*pcauchy(x,l,s)*	Возвращает функцию распределения Коши с параметром масштаба s и параметром расположения *l*. $s > 0$.
5	*pchisq(x,d)*	Возвращает функцию распределения *хи*-квадрат, в котором $d > 0$ равно числу степеней свободы.
6	*pexp(x,r)*	Возвращает функцию экспоненциального распределения, в котором $r > 0$ является параметром.

7	*pF(x,d_1,d_2)*	Возвращает функцию F-распределения, в котором $d_1, d_2 > 0$ являются числами степеней свободы.
8	*pgamma(x,s)*	Возвращает функцию *Гамма*-распределения, в котором $s > 0$ является параметром формы.
9	*pgeom(k,p)*	Возвращает функцию геометрического распределения. p есть вероятность успеха в одиночном испытании. $0 < p \leq 1$.
10	*plnorm(x,m,s)*	Возвращает функцию логнормального распределения, в котором m равно логарифму среднего значения, а $s > 0$ есть логарифм среднеквадратичного отклонения.
11	*plogis(x,l,s)*	Возвращает функцию логистического распределения. l есть параметр расположения. $s > 0$ – параметр масштаба.
12	*pnbinom(k,n,p)*	Возвращает функцию отрицательного биномиального распределения, в котором $0 < p \leq 1$, n – натуральное.
13	*pnorm(x,m,s)*	Возвращает функцию нормального распределения со средним m и среднеквадратичным отклонением s. $s > 0$.
14	*ppois(k,l)*	Возвращает функцию распределения Пуассона. $l > 0$.
15	*pt(x,d)*	Возвращает функцию t-распределения Стьюдента. d есть число степеней свободы. $d > 0$.
16	*punif(x,a,b)*	Возвращает функцию равномерного распределения. b и a есть граничные точки интервала. $a < b$.
17	*pweibull(x,s)*	Возвращает функцию распределения Вейбулла. $s > 0$.

7.2.3 Обращения функций распределения

Эти функции принимают вероятность p как аргумент и возвращают значение *x* такое, что $P(X \leq x) = p$.

Обращения функций распределения представлены в таблице 7.4.

Таблица 7.4 – Обращение функций распределения

№	Вид функции	Описание функции
1	*qbeta(p,s_1,s_2)*	Обращает *бета*-распределение с параметрами формы s_1 и s_2. $(0 \leq p \leq 1)$ $(s_1, s_2 > 0)$.
2	*qbinom(p,n,r)*	Возвращает число успехов в n испытаниях схемы Бернулли при условии, что вероятность успехов не превышает p и r – вероятность успеха на одиночном испытании. $0 \leq r \leq 1$ и $0 \leq p \leq 1$. n есть натуральное число.

3	*qcauchy(p,l,s)*	Обращает распределение Коши с параметром масштаба s и параметром расположения l. $s > 0, 0 < p < 1$.
4	*qchisq(p,n)*	Обращает *хи*-квадрат распределение, в котором $d > 0$является числом степеней свободы. $0 \leq p < 1$.
5	*qexp(p,r)*	Обращает экспоненциальное распределение, в котором $r > 0$ является параметром. $0 \leq p < 1$.
6	*qF(p,d_1,d_2)*	Обращает F -распределение, в котором $d_1, d_2 > 0$ являются числами степеней свободы. $0 \leq p < 1$
7	*qgamma(p,s)*	Обращает *Гамма*-распределение, в котором $s > 0$ является параметром формы. $0 \leq p < 1$
8	*qgeom(p,r)*	Обращает геометрическое распределение. r – вероятность успеха при одиночном испытании. $0 < r < 1$ и $0 < p < 1$.
9	*qlnorm (p,m,s)*	Обращает логнормальное распределение, в котором m является натуральным логарифмом среднего значения, $s > 0$ – натуральный логарифм среднеквадратичного отклонения. $0 \leq p < 1$.
10	*qlogis(p,l,s)*	Обращает логистическое распределение. l – параметр расположения, $s > 0$ – параметр масштаба. $0 < p < 1$.
11	*qnbinom(p,n,r)*	Обращает отрицательное биномиальное распределение с числом испытаний n и вероятностью успеха в одиночном испытании r. $0 < r \leq 1$ и $0 \leq p \leq 1$.
12	*qnorm(p,m,s)*	Обращает нормальное распределение со средним m и среднеквадратичным отклонением s. $0 < p < 1$ и $s > 0$.
13	*qpois(p,l)*	Обращает распределение Пуассона. $l > 0$ и $0 \leq p \leq 1$.
14	*qt(p,d)*	Обращает t-распределение Стьюдента. d – число степеней свободы. $d > 0$ и $0 < p < 1$.
15	*qunif(p,a,b)*	Обращает равномерное распределение. b и a – граничные точки интервала. $a < b$ и $0 \leq p \leq 1$.
16	*qweibull(p,s)*	Обращает распределение Вейбулла. s$>$ 0 и $0 < p < 1$.

Пример:

$x := -10, -9.8 .. 10$

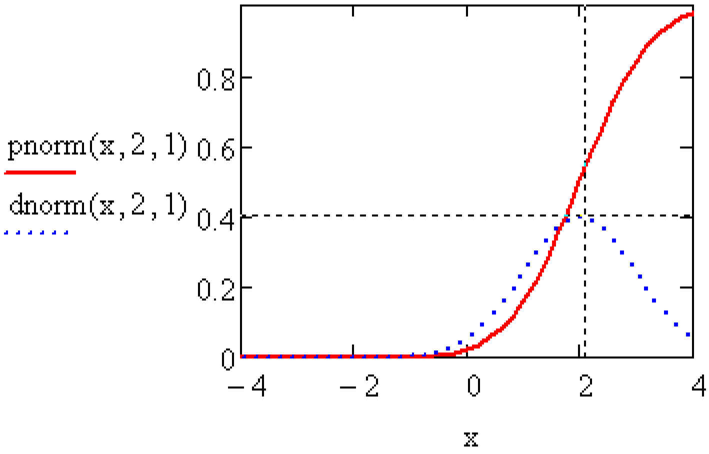


Рисунок 7.1 – Связь между плотностями вероятности, функциями распределения и их обратными функциями

Функция *dnorm* создает колоколообразную кривую, имеющую в данном случае ось симметрии, проходящую через точку $x = 2$. Функция *pnorm* есть интеграл от функции *dnorm*.

$$\mathrm{qnorm}(0.5, 2, 1) = 2$$

$$\mathrm{pnorm}(2, 2, 1) = 0.5$$

Из примера наглядно видно, что *qnorm* и *pnorm* – взаимно обратные функции.

7.3 Функция *hist*

Mathcad включает одну функцию, *hist*, для вычисления частотного распределения, применяемого для построения гистограмм:

$$hist(int, A)$$

Возвращает вектор, представляющий частоты, с которыми величины, содержащиеся в векторе A, попадают в интервалы, представляемые вектором *int*. Элементы в A и *int* и должны быть вещественными. Кроме того, элементы *int* должны быть расположены в порядке возрастания. Возвращаемый результат – вектор, содержащий на один элемент меньше, чем *int*.

Mathcad интерпретирует *int* как набор точек, определяющих последовательность интервалов в гистограмме. Значения в *int* должны быть расположены в порядке возрастания. Результатом этой функции является вектор *f*, в котором есть число значений в A, удовлетворяющих условию:

$$int_i \le value < int_{i+1}$$

Mathcad игнорирует данные, меньшие, чем первое значение в *int*, или большие, чем последнее значение в *int*. Рисунок 7.2 показывает, как использовать гистограммы в Mathcad.

Пример:

$N := 10$

$j := 0 .. N$

$k := 0 .. N - 1$

$$intervals_j := 1 + \frac{j}{2}$$

$f := hist(intervals, data)$

$min(data) = 0.04$

$max(data) = 11.4$

$mean(data) = 3.109$

$$data := \begin{pmatrix}
0.75 & 0.7 & 0.6 & 0.36 & 0.66 & 0.55 & 0.57 & 0.48 & 0.42 & 0.12 & 0.11 & 1.1 & 1.14 & 7.8 & 9.1 \\
11.4 & 8.45 & 9.19 & 6.9 & 7.8 & 7.55 & 4.15 & 3.33 & 3.21 & 1.22 & 6.1 & 8.4 & 5.1 & 4.33 & 4.7 \\
7.15 & 0.34 & 0.38 & 1.42 & 0.66 & 0.87 & 6.5 & 8.5 & 1.24 & 0.57 & 1.68 & 8.65 & 4.66 & 2.13 & 0.33 \\
0.69 & 0.68 & 0.57 & 0.63 & 0.52 & 4.15 & 0.65 & 0.45 & 4.27 & 4.25 & 4.65 & 5.45 & 2.23 & 1.25 & 4.15 \\
0.14 & 0.87 & 0.79 & 0.56 & 1.28 & 1.26 & 4.25 & 2.36 & 4.58 & 5.69 & 4.23 & 3.24 & 3.21 & 4.58 & 1.85 \\
4.69 & 5.12 & 4.26 & 2.36 & 3.31 & 2.3 & 4.36 & 4.26 & 1.19 & 3.56 & 2.35 & 3.25 & 1.36 & 4.1 & 5.68 \\
7.18 & 4.68 & 4.21 & 0.25 & 0.68 & 9.6 & 4.15 & 7.36 & 4.12 & 6.08 & 0.14 & 1.14 & 1.21 & 4.69 & 1.25 \\
4.65 & 1.14 & 4.21 & 1.03 & 0.55 & 0.87 & 0.64 & 1.25 & 1.47 & 5.21 & 4.32 & 0.77 & 0.77 & 0.69 & 1.1 \\
2.21 & 4.51 & 1.7 & 7.4 & 4.65 & 0.23 & 0.74 & 1.14 & 6.25 & 3.11 & 0.89 & 0.66 & 2.37 & 0.48 & 0.99 \\
4.15 & 7.1 & 5.21 & 1.4 & 4.51 & 1.23 & 4.52 & 0.12 & 1.24 & 0.33 & 2.65 & 3.26 & 1.47 & 1.26 & 0.87 \\
2.21 & 4.12 & 4.01 & 6.3 & 4.2 & 4.21 & 8.51 & 4.23 & 6.25 & 7.12 & 0.11 & 2.14 & 0.33 & 2.36 & 0.25 \\
4.26 & 3.05 & 0.21 & 1.25 & 2.3 & 4.12 & 0.04 & 1.47 & 8.62 & 4.28 & 1.02 & 3.01 & 0.25 & 6.35 & 4.68 \\
3.25 & 0.05 & 1.24 & 0.47 & 4.67 & 8.41 & 4.44 & 5.55 & 6.25 & 5.47 & 4.58 & 4.25 & 1.24 & 4.16 & 7.81 \\
7.15 & 4.61 & 1.64 & 2.48 & 7.58 & 4.18 & 2.01 & 0.11 & 0.13 & 0.48 & 4.65 & 1.02 & 4.69 & 0.33 & 4.75 \\
6.14 & 4.15 & 2.31 & 6.15 & 4.28 & 2.36 & 0.25 & 0.44 & 1.25 & 0.65 & 1.24 & 4.21 & 3.24 & 0.68 & 4.15
\end{pmatrix}$$

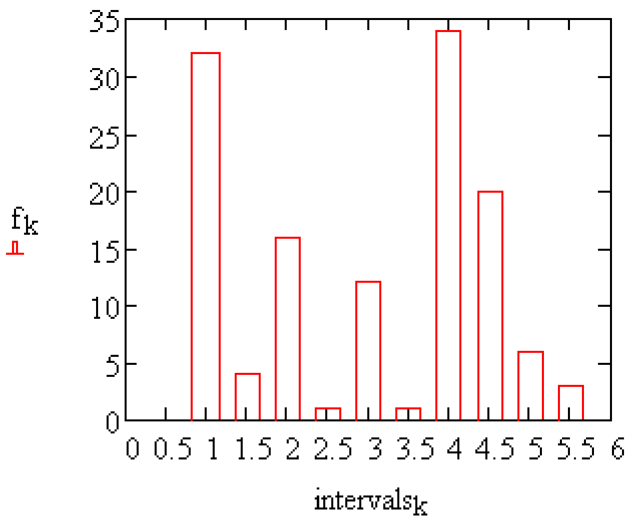


Рисунок 7.2 – Построение гистограммы

7.4 Случайные числа

Mathcad поставляется с рядом функций для генерирования случайных чисел, имеющих разнообразные распределения вероятностей. Функциональные формы распределений, связанных с приведенными ниже функциями даны в подразделе «Распределения вероятности».

Mathcad PLUS поставляется со всеми функциями, перечисленными в этом разделе. Если Вы не используете Mathcad PLUS, Вы будете иметь все генераторы случайных чисел, связанные со следующими законами распределения вероятностей: нормальным, *хи*-квадрат, t-распределением Стьюдента, F, биномиальным, Пуассона и равномерным.

Случайные числа представлены в таблице 7.5.

Таблица 7.5 – Случайные числа

№	Вид функции	Описание функции
1	$rbeta(m,s_1,s_2)$	Возвращает вектор m случайных чисел, имеющих бэта-распределение. $s_1, s_2 > 0$ есть параметры формы.
2	$rbinom(m,n,p)$	Возвращает вектор m случайных чисел, имеющих биномиальное распределение. $0 \leq p \leq 1$. n есть натуральное число.
3	$rcauchy(m,l,s)$	Возвращает вектор m случайных чисел, имеющих распределение Коши. $s > 0$ есть параметр масштаба. l – параметр расположения.
4	$rchisq(m,d)$	Возвращает вектор m случайных чисел, имеющих распределение *хи*-квадрат. $d > 0$ есть число степеней свободы.
5	$rexp(m,r)$	Возвращает вектор m случайных чисел, имеющих экспоненциальное распределение. $r > 0$ – параметр распределения.
6	$rF(m,d_1,d_2)$	Возвращает вектор m случайных чисел, имеющих F -распределение. $d_1, d_2 > 0$ есть числа степеней свободы.
7	$rgamma(m,s)$	Возвращает вектор m случайных чисел, имеющих гамма- распределение, $s > 0$ есть параметр формы.
8	$rgeom(m,p)$	Возвращает вектор m случайных чисел, имеющих геометрическое распределение. $0 < p \leq 1$
9	$rlnorm(m,m,s)$	Возвращает вектор m случайных чисел, имеющих логнормальное распределение, в котором m является натуральным логарифмом среднего значения, а $s > 0$ есть натуральный логарифм среднеквадратичного отклонения.
10	$rlogis(m,l,s)$	Возвращает вектор m случайных чисел, имеющих

		логистическое распределение, в котором l является параметром расположения, а $s > 0$ есть параметр масштаба.
11	*rnbinom(m,n,p)*	Возвращает вектор m случайных чисел, имеющих отрицательное биномиальное распределение. $0 < p \le 1$. n есть натуральное число.
12	*rnorm(m,m,s)*	Возвращает вектор m случайных чисел, имеющих нормальное распределение. $s > 0$.
13	*rpois(m,l)*	Возвращает вектор m случайных чисел, имеющих распределение Пуассона. $l > 0$.
14	*rt(m,d)*	Возвращает вектор m случайных чисел, имеющих t-распределение Стьюдента. $d > 0$.
15	*runif(m,a,b)*	Возвращает вектор m случайных чисел, имеющих равномерное распределение, в котором b и a являются граничными точками интервала. $a < b$.
16	*rnd(x)*	Возвращает равномерно распределенное случайное число между 0 и x. Эквивалент *runif(1,0,x)*.
17	*rweibull(m,s)*	Возвращает вектор m случайных чисел, имеющих распределение Вейбулла, в котором $s > 0$ является параметром формы.

Каждый раз, когда повторно вычисляется выражение, содержащее одну из этих функций, Mathcad генерирует новые случайные числа. Чтобы заставить Mathcad генерировать новые случайные числа, щёлкните мышью на выражении, содержащем функцию, и нажмите <F9>. Рисунок 7.3 показывает пример того, как использовать генератор случайных чисел Mathcad. Рисунок 7.4 показывает, как создать большой вектор случайных чисел, имеющих заданное распределение.

Каждая из этих функций в действительности создаёт последовательность псевдослучайных чисел, связанную с некоторым задаваемым стартовым значением. Каждое нажатие <F9> заставляет функцию выдать новое значение из этой последовательности. Одно и то же стартовое значение производит одинаковые последовательности чисел. Изменение стартового значения приводит к смене последовательности случайных чисел, выдаваемых функцией.

Чтобы изменить стартовое значение, выберите *Генератор случайных чисел...* из меню *Математика* и измените стартовое значение в диалоговом окне. Убедитесь, что введено целое число.

Чтобы перезапустить генератор случайных чисел Mathcad, не изменяя стартового значения, выберите *Генератор случайных чисел...* из меню *Математика* и нажмите «OK», чтобы принять текущее значение. Затем щёлкните мышью на выражении с функцией, генерирующей случайное число, и нажмите <F9>.

$i := 0..350$

$\theta_i := \mathrm{rnd}(2\pi)$

$r_i := \mathrm{rnd}(1)$

$i =$	$\theta_i =$	$r_i =$
0	$7.97 \cdot 10^{-3}$	0.855
1	1.215	0.307
2	3.676	0.473
3	2.201	0.192
4	5.17	0.995
5	1.094	0.648
6	4.464	0.174
7	1.91	0.275
8	0.574	0.839
9	0.926	0.371
10	6.211	0.233
11	0.748	0.232
12	0.056	0.971
13	3.341	0.023
14	3.781	0.152
...	...	...

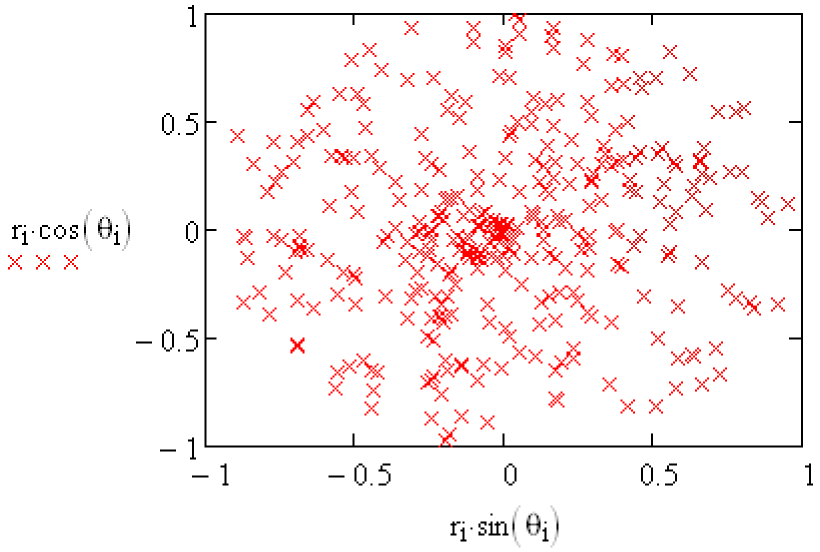


Рисунок 7.3 – Равномерно распределения случайные числа

Примечание: так как генератор случайных чисел производит каждый раз различные числа, маловероятно, что удастся в точности воспроизвести этот пример.

$v := rnorm(8000, 0, 1)$ — Создает вектор случайных чисел, имеющих стандартное нормальное распределение

$m := 0..20$

$int_m := -1 + 0.1 \cdot m$

$h := hist(int, v)$ — Подсчитывает число попаданий случайной величины в заданные интервалы

$n := 0..19$

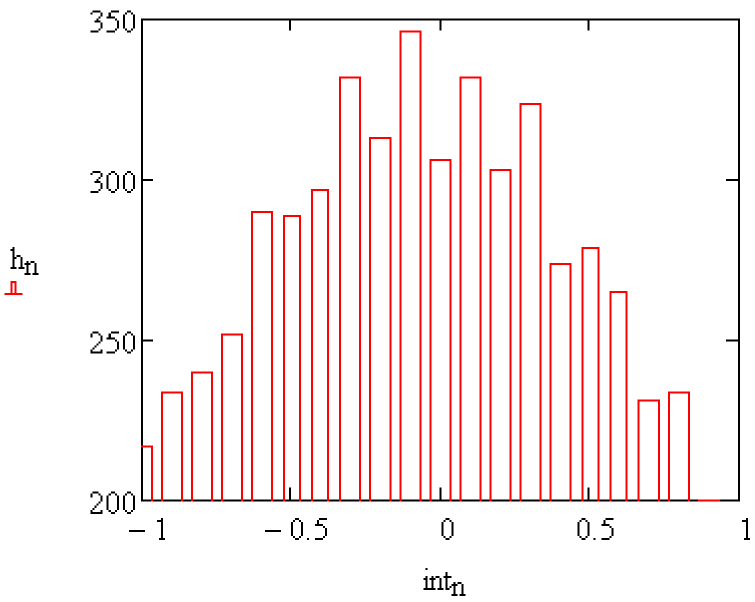


Рисунок 7.4 – Вектор случайных чисел, распределенных по нормальному закону

Так как генератор случайных чисел был сброшен, Mathcad будет производить те же самые случайные числа, которые производились бы после перезапуска Mathcad.

Если нужно несколько раз использовать одну и ту же последовательность случайных чисел, сбросьте генератор случайных чисел между вычислениями, как описано выше.

Чтобы получить новый набор случайных чисел, измените стартовое значение, как описано выше. Это заставит Mathcad генерировать набор случайных чисел, отличный от того, который создаётся после перезапуска Mathcad. Каждый раз при необходимости получить новую последовательность случайных чисел следует переустанавливать стартовое значение, как описано выше.

8 Файлы данных

8.1 Файлы данных и функции доступа к ним

Файл данных Mathcad должен быть просто файлом в ASCII формате. Mathcad читает файлы, которые состоят из чисел, отделяемых запятыми, пробелами или возвратами каретки. Ниже описаны примеры некоторых файлов, которые Mathcad может читать, в предположении, что они записаны в ASCII формате:

- Файл, содержащий экспериментальные данные, фиксируемые аппаратными средствами и программным обеспечением сбора данных.
- Файл, созданный выводом данных из электронной таблицы на диск.
- Столбец чисел, набитых в текстовом процессоре и сохраненных в ASCII формате.
- Результат работы программы, написанной на языке БЕЙСИК.
- Данные, экспортированные из базы данных.

Числа в файлах данных могут быть целыми числами, подобными 3 или -1, числами с плавающей запятой, подобными 2.54, или иметь экспоненциальную запись, как 4.51E -4 (для $4.5 \cdot 10^{-4}$). Например, следующий список чисел был бы допустимой строкой в файле данных Mathcad:

$$200, 50\ 25.1256, 16E - 2, - 16.125E15$$

Mathcad также сохраняет данные в ASCII файлы. Файлы данных, сохраненные Mathcad, содержат числа, отделяемые пробелами и возвратами каретки. Обратите внимание: документы Mathcad сами по себе не являются файлами данных в указанном понимании. Единственный способ создать файл данных из Mathcad – использование функций доступа к файлам, описанное в этой главе.

8.1.1 Функции доступа к файлам

В Mathcad есть шесть функций доступа к файлам *READ*, *WRITE*, *APPEND*, *READPRN*, *WRITEPRN* или *APPENDPRN*. Их свойства:

- Имя функции должно печататься заглавными буквами. Можно также использовать пункт *Вставить функцию* меню *Математика*, чтобы перенести имя функции из раскрывающегося списка двойным щелчком мыши. *Insert Function command*;
- Если Mathcad не может найти файл данных, он отмечает соответствующую функцию доступа сообщением об ошибке «файл не найден». Если Mathcad пытается прочитать файл неподходящего формата, он отмечает функцию сообщением «ошибка файла»;
- Левая часть оператора присваивания, использующего одну из функций *WRITE, APPEND, WRITEPRN* или *APPENDPRN* и должна больше ничего содержать;
- Каждое новое равенство с использованием функций доступа заново открывает файл данных. При считывании данных,

например, каждое новое равенство начинает читать в начале файла;

- В одном равенстве файл может быть открыт только единожды. Это означает, что, если функция *READ* используется с одним аргументом-именем файла дважды в одном уравнении (это возможно при использовании дискретного аргумента), во второй раз *READ* начнет читать с того места, где закончила в первый. Поскольку *READPRN* читает весь файл целиком, это означает, что *READPRN* нельзя использовать с одним аргументом дважды в одном равенстве – во второй раз *READPRN* не останется ничего читать;
- Если два равенства в рабочем документе используют *WRITE* или *WRITEPRN* с одним аргументом, данные из второго равенства запишутся поверх данных из первого. Используйте *APPEND* или *APPENDPRN,* если нужно сохранить первую порцию данных. Эти функции дописывают новые данные к существующему файлу;

Таблица 8.1 ниже описывает эти шесть функций. В этой таблице:

- А обозначает массив (вектор или матрицу);
- *v*i обозначают отдельные элементы вектора v;
- *file* – любое допустимое имя переменной Mathcad;
- *i* – дискретный аргумент.

Функции *Read* , *Write* и *Append* могут использоваться с дискретными аргументами, остальные – нет.

Таблица 8.1 – Описание функций данных

№	Функция	Значение
1	*READ*(*file*)	Считывает значение из файла данных. Возвращает скаляр. Обычно используется следующим образом: v_i:=*READ*(*file*)
2	*WRITE*(*file*)	Записывает значение в файл данных. Если файл уже существует, заменяет его на новый файл. Должна использоваться в определениях следующего вида: *WRITE*(*file*):=v_i
3	*APPEND*(*file*)	Дописывает значение к существующему файлу. Должна использоваться в определениях следующего вида: *APPEND*(*file*):=v_i
4	*READPRN*(*file*)	Читает структурированный файл данных. Возвращает матрицу. Каждая строка в файле данных становится строкой в матрице. Число элементов в каждой строке должно быть одинаковым. Обычно используется следующим образом: *A*:=*READPRN*(*file*)

5	*WRITEPRN(file)*	Записывает матрицу в файл данных. Каждая строка матрицы становится строкой в файле. Должна использоваться в определениях следующего вида: *WRITEPRN*(*file*):=*A*
6	*APPENDPRN*(*file*)	Дописывает матрицу к существующему файлу. Каждая строка в матрице становится новой строкой в файле данных. Должна использоваться в определениях следующего вида: *APPENDPRN(file):=A*. Существующий файл должен иметь столько же столбцов, как и матрица *A*.

8.1.2 Аргументы функций доступа к файлам

Аргументы всех функций в предыдущей таблице называются файловыми переменными. *file variable* в отличие от других переменных файловая переменная содержит в себе не числовое значение, а имя файла. За исключением случаев, описанных ниже в разделе «Импортирование данных из других каталогов», имя файла, соответствующего файловой переменной, есть просто имя этой переменной с расширением *.dat* или *.prn*.

Выбор расширения зависит от функции, используемой с файловой переменной. Например, если имя файловой переменной – *papageno*, и используется *READPRN* или *WRITEPRN*, то Mathcad будет работать с файлом, называемом *papageno.prn* и находящемся в текущем каталоге. Если же использовать *READ* либо *WRITE*, Mathcad будет работать с файлом, называемым *papageno.dat* и находящемся в текущем каталоге.

Можно отменять расширения по умолчанию *.dat* и *.prn*, используя в качестве файловой переменной имя файла с расширением. Например, чтобы читать из файла *papageno.txt*, просто напечатайте имя с расширением в качестве файловой переменной. Точка исчезнет, и появится запись «$papageno_{txt}$».

Будьте внимательны, используя слишком длинные имена файловых переменных. Хотя Mathcad допускает создание длинных имен файловых переменных вроде $resistance_{parallel}$, они не могут соответствовать именам файлов, потому что содержат слишком много символов. В этом случае Mathcad ищет файл, чьё имя получается усечением имени файловой переменной. Например, READPRN($resistance_{parallel}$) указывает Mathcad читать из файла с названием *resistan.par*.

Если файл, с которым нужно работать, размещен в каталоге ином, нежели каталог по умолчанию, нельзя просто напечатать полный путь к файлу между круглыми скобками одной из функций доступа. Дело в том, что функция доступа есть математическое выражение, и в результате Mathcad истолковывает нажатие клавиш \ или : при попытке указать символ дисковода или каталог как указание создать квадратный корень или оператор определения.

Чтобы обойти эту трудность, Mathcad позволяет присвоить имя файла файловой переменной подобно тому, как число присваивается обычной переменной. Эта возможность обсуждена в следующем разделе.

8.2 Импортирование данных из других каталогов

При работе с документом Mathcad каталогом по умолчанию считается тот, из которого документ загружен, или в котором он последний раз сохранялся.

Функции Mathcad доступа к файлам работают с файлами из каталога по умолчанию. Обычно имя требуемого файла есть просто имя файловой переменной с добавленным к нему расширением *.dat* либо *.prn*. Таким образом, если используется *READPRN*(*papageno*), то Mathcad ищет файл *papageno.prn* в каталоге по умолчанию. Если используется *READ*(*papageno*), Mathcad ищет *papageno.dat* в каталоге по умолчанию.

Можно указать функции доступа на файл, находящийся в произвольном месте, не обязательно в каталоге по умолчанию. Следующий пример показывает, как обратиться к файлу с названием *papageno.dat* в каталоге *C:\rem*:

- Выберите *Присоединить к файлу* из меню *Файл*.
- В поле «Переменная Mathcad» введите имя файловой переменной. Если, например, для доступа к данным используется конструкция *READPRN*(*Aviarium*), то напечатайте *Aviarium*. Имейте в виду, что имена переменных в Mathcad являются чувствительными к регистру. Нужно напечатать имя переменной точно так, как оно написано в рабочем документе.
- Задайте путь к файлу данных. В этом примере путь есть *C:\rem\papageno.prn*. Можно использовать при этом списки директорий и имен файлов, расположенные в диалоговом окне.
- Нажмите «Присоединить». Mathcad будет теперь обращаться к файлу *C:\rem\papageno.prn* всякий раз, когда ему встретится функция доступа с аргументом *Aviarium*.
- Выберите *Пересчитать все* из меню *Математика*, чтобы обновить документ.

В отличие от других равенств равенства, содержащие функции доступа к файлам, не обрабатываются автоматически. Если изменить файл данных или присоединить другой файл к файловой переменной, это не повлияет на вычисления. Чтобы произошло соответствующее обновление вычислений, нужно либо щелкнуть в равенстве и нажать <F9>, либо выбрать *Пересчитать все* из меню *Математика*. Можно считать, что функции доступа к файлам всегда находятся в ручном режиме вычислений: *update*; *file access functions*.

Поскольку Mathcad способен одновременно держать открытыми только конечное число файлов, может понадобится разорвать связь с файловой переменой, когда она уже больше не нужна (это может быть необходимо, например, при сообщении об ошибке «слишком много файлов»). Для этого выберите *Присоединить к файлу* из меню *Файл*, и напечатайте имя переменной, связь с которой нужно отменить. Затем нажмите «Отсоединить».

8.3 Неструктурированные файлы

В этом разделе обсуждается использование функций *READ, WRITE* и *APPEND* для работы с неструктурированными файлами. Неструктурированный файл данных – файл, который содержит числа, расположенные не обязательно в строках и столбцах.

8.3.1 Чтение данных функцией *READ*

Рассмотрим два способа использования функции *READ* для чтения данных из файла.

Первое равенство с использованием *READ* присваивает переменой *N* первое значение из файла данных *sizefile.dat*. Второе равенство с использованием *READ caiieiyao iannea y* первыми 100 числами из файла данных *papageno.dat*.

Когда Mathcad читает данные с помощью *READ*:

Каждое новое равенство заново открывает файл и начинает читать из его начала. Невозможно считать два последовательных набора данных из одного файла, используя два раздельных равенства с *READ*.

Если равенство содержит *READ* и дискретную переменную, Mathcad считывает по одному значению из файла для каждого значения дискретного аргумента. Если данных меньше, чем значений дискретного аргумента, Mathcad перестает считывать данные для лишних значений дискретного аргумента. Если значений дискретного аргумента меньше, чем данных, Mathcad игнорирует лишние данные в файле.

Равенство не может включать более чем одну функцию *READ*.

Чтобы читать из файла с именем, которое не совпадает с именем файловой переменной, либо из файла, находящегося не в каталоге по умолчанию, выберите *Присоединить к файлу* из меню *Файл*.

8.3.2 Запись данных с помощью функций *WRITE* и *APPEND*

При использовании функции *WRITE*, чтобы записать данные в файл необходимо следующее.

Первое равенство с использованием *WRITE* записывает число в файл данных *sizefile.dat*. Второе равенство с *WRITE caienuaaao N* чисел в файл данных *papageno.dat*, по одному числу для каждого значения дискретного аргумента *i*. Когда Mathcad записывает данные в файл, он отделяет последовательные значения пробелами, а также вставляет прерывания строки, сохраняя длину строк менее 80 символов. При использовании *WRITE* все значения передаются файлу с максимальной точностью, независимо от глобального формата документа.

Mathcad игнорирует единицы размерности при записи данных в файл.

Подобно функции *READ*, функция *WRITE* заново открывает файл и вновь переходит к его началу в каждом новом равенстве.

Если нужно записывать данные в файл из различных уравнений, используйте функцию *APPEND* вместо *WRITE* во втором и последующих уравнениях.

Пример: Запись данных с помощью *WRITE*.

$$N := 100$$

$$\mathrm{WRITE}(\mathrm{sizefile}) := N$$

$$i := 0 .. N-1$$

$$y_i := 2.75 \cdot i + 115 + \mathrm{rnd}(i)$$

$$\mathrm{WRITE}(\mathrm{papageno}) := y_i$$

Примечание: Если функция *WRITE* используется с одним аргументом в двух равенствах, данные из второго равенства будет записаны поверх данных из первого.

8.4 Структурированные файлы

В этом разделе обсуждается использование функций *READPRN*, *WRITEPRN* и *APPENDPRN* для работы со структурированными файлами. Структурированный файл данных – файл с фиксированным числом значений на строке. Например, если экспортировать прямоугольную область из электронной таблицы в текстовый файл, возникающие в результате строки и столбцы чисел сформируют структурированный файл.

8.4.1 Считывание матрицы с помощью функции *READPRN*

Предположим, что имеется ASCII-файл, содержащий данные, показанные ниже. Эти данные могут быть взяты из электронной таблицы или из любого другого источника.

Функция *READPRN* читает весь файл данных, определяет число строк и столбцов, и создает матрицу из этих данных.

Когда Mathcad читает данные с помощью функции *READPRN*:

- Каждый раз *READPRN* читает файл данных целиком.
- Все строки в файле данных должны содержать одинаковое количество значений. (Строки, не содержащие значений, игнорируются). Если строки в файле имеют различное число значений, Mathcad, отмечает функцию *READPRN* сообщением «ошибка файла».
- Функция *READPRN* игнорирует текст в файле данных.
- Результатом чтения файла данных является $m \times n$ матрица, где m есть число строк, содержащих данные в файле, и n есть число значений в строке. Чтобы создать матрицу из значений в файле

данных, используйте равенство, подобное *M*:=*READPRN*(*file*). Не используйте нижние индексы для *M*. Функция *READPRN* возвращает матрицу, поэтому нижние индексы излишни.

- Чтобы читать из файла с именем, которое не совпадает с именем файловой переменной, либо из файла, находящегося не в каталоге по умолчанию, выберите *Присоединить к файлу* из меню *Файл*. Подробнее см. раздел «Импортирование данных из других каталогов».

Примечание: Каждая строка в файле данных должна содержать одинаковое число значений. Если оставить промежутки там, где Mathcad ожидает значение, функция *READPRN* не сможет прочитать файл. Mathcad определяет конец одного и начало следующего значения, ища пробелы или запятые.

Иногда каждый столбец значений в файле данных представляет отдельную переменную.

8.4.2 Запись данных при помощи функций WRITEPRN и APPENDPRN

В отличие от *WRITE* функция *WRITEPRN* записывает данные в виде столбцов. Обратите внимание, что, поскольку для *PRNPRECISION* установлено значение четыре, числа записываются с четырьмя знаками после запятой. Поскольку значение *PRNCOLWIDTH* равно восьми, каждый столбец имеет ширину в восемь символов. Так как *PRNPRECISION* и *PRNCOLWIDTH* могут изменяться независимо, нужно не упустить из виду, что ширина столбца должна быть такой, чтобы разместились все необходимые цифры вместе с пробелом, разделяющим отдельные значения.

Пример: Запись данных в структурированный файл данных.

$$i := 0..5$$

$$j := 0..7$$

$$A_{i,j} := i \cdot \sin(j \cdot 5)$$

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.959 & -0.544 & 0.65 & 0.913 & -0.132 & -0.988 & -0.428 \\ 0 & -1.918 & -1.088 & 1.301 & 1.826 & -0.265 & -1.976 & -0.856 \\ 0 & -2.877 & -1.632 & 1.951 & 2.739 & -0.397 & -2.964 & -1.285 \\ 0 & -3.836 & -2.176 & 2.601 & 3.652 & -0.529 & -3.952 & -1.713 \\ 0 & -4.795 & -2.72 & 3.251 & 4.565 & -0.662 & -4.94 & -2.141 \end{pmatrix}$$

$$\mathrm{WRITEPRN}(sines) := A$$

Когда используется функция *WRITEPRN*:

- Равенства, содержащие *WRITEPRN*, должны быть следующего вида. Слева записывается *WRITEPRN*(file), где *file* – имя файловой переменной, затем следует символ определения (:=) и выражение, возвращающее матрицу. Не используйте дискретные аргументы или нижние индексы с *WRITEPRN*.
- Каждое новое равенство записывает новый файл. Если два равенства пишут в один файл, данные, написанные вторым равенством уничтожат данные, написанные первым. Используйте *APPENDPRN*, если нужно дописать значения к файлу вместо того, чтобы перезаписать файл. *APPENDPRN function*
- Встроенные переменные *PRNCOLWIDTH* и *PRNPRECISION* определяют формат файла данных, создаваемого Mathcad. Текущее значение *PRNCOLWIDTH* определяет ширину столбцов (в символах). Текущее значение *PRNPRECISION* определяет число используемых значащих цифр. По умолчанию *PRNCOLWIDTH*=8 и *PRNPRECISION*=4. Чтобы изменить эти значения, выберите *Встроенные переменные* из меню *Математика* или поместите соответствующие определения в документ Mathcad выше места использования *WRITEPRN*.
- Если записываемый массив является составным (чьи элементы являются сами массивами), либо комплекснозначным, то тогда *WRITEPRN* создаст не просто ASCII-файл, а файл специального формата, который навряд ли будет читаем другими прикладными программами. Этот файл может, однако, читаться функцией *READPRN*.

Пример: показывает документ, который создает файл данных со столбцами шириной в 10 символов, содержащий числа с 5 значащими цифрами.

$i := 0 .. 5$

$j := 0 .. 7$

$A_{(i,j)} := i \cdot \sin\left(j \cdot \frac{\pi}{5}\right)$

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.588 & 0.951 & 0.951 & 0.588 & 0 & -0.588 & -0.951 \\ 0 & 1.176 & 1.902 & 1.902 & 1.176 & 0 & -1.176 & -1.902 \\ 0 & 1.763 & 2.853 & 2.853 & 1.763 & 0 & -1.763 & -2.853 \\ 0 & 2.351 & 3.804 & 3.804 & 2.351 & 0 & -2.351 & -3.804 \\ 0 & 2.939 & 4.755 & 4.755 & 2.939 & 0 & -2.939 & -4.755 \end{pmatrix}$$

PRNPRECISION := 5

PRNCOLWIDTH := 10

WRITEPRN(sinevals) := A

Используя функцию *augment*, можно объединять отдельные переменные в массивы, и записывать их все в файл данных.

Пример: Запись некоторых векторов, объединенных вместе.

$i := 0 .. 99$

$x_i := \sin\left(\frac{i}{10}\right)$

$y_i := \cos\left(\frac{i}{10}\right)$

$z_i := \tan\left(\frac{i}{10}\right)$

Можно использовать три вектора в один файл с тремя столбцами:

WRITEPRN(Trigs) := augment[(x,y),z]

8.4.3 Преимущества использования READPRN и WRITEPRN

Использование *READPRN*, как правило, предпочтительнее использования *READ*. Когда данные структурированы в столбцы, *READPRN* переводит данные в Mathcad в легкодоступной форме.

Если некоторые строки в файле данных имеют большее количество значений, чем другие, данные могут быть потеряны. Используйте текстовый редактор, чтобы заменить пропущенные значения на нули перед использованием *READPRN*.

Функция *READ* используется для файлов, в которых значения одной переменной разбросаны по нескольким строкам. Таковы файлы, созданные *WRITE*, которая располагает столько чисел на строке, сколько строка может вместить.

Помните: используйте нижний индекс, являющийся дискретным аргументом, чтобы читать с помощью *READ;* не используйте нижний индекс, чтобы читать с помощью *READPRN*.

Обычно *WRITEPRN* производит более читаемые файлы, чем *WRITE*, поскольку данные в них аккуратно расположены в строках и столбцах. С другой стороны, *WRITE* производит меньшие файлы, чем *WRITEPRN*, потому что ей не нужно добавлять пробелы для выравнивания данных.

Используйте *WRITE* вместо *WRITEPRN*, когда требуется напихать так много значений, сколько возможно, в малый файл данных.

9 Учебная задача «Разработка и исследование алгоритма оценки временного положения сигнала при наличии шума»

В современном мире задача обнаружения объектов каким-либо устройством с помощью различных сигналов (определение расстояния до объекта, скорость объекта и т.д.) играет большую роль и часто оказывается очень сложной. Из-за наличия всевозможных шумов и помех, большого расстояния до цели достаточно трудно оценить характеристики сигнала. Шумы обусловливают случайный характер результатов наблюдения, и поэтому наиболее подходящим способом изучения этих явлений является статистический метод. Каждое приемное устройство является фильтром, выделяющим полезную информацию из принимаемой смеси сигнала и шума.

Целью данной учебной задачи является оценка временного положения сигнала при наличии шума, нахождение энергии полезного сигнала при заданных значениях, определение среднеквадратичной ошибки оценки (СКО), построение гистограммы распределения оценок, улучшение навыков работы в среде Mathcad.

9.1 Полезный сигнал и его параметры

Пусть полезный сигнал задается уравнением:

$$s(t) = a \cdot (t - t0)^m \cdot exp\left[\frac{-(t-t0)^q}{2 \cdot b}\right], t \geq t0,$$

где a – константа, которая задается пользователем (она определяет амплитуду сигнала, то есть его максимальное значение);

$t0$ – временное положение сигнала;

t – временной параметр функции, определяющий сигнал.

$$s(t) := \begin{cases} \left[a \cdot (t - t0)^m \cdot \exp\left[\dfrac{-(t - t0)^q}{2 \cdot b} \right] \right] & \text{if } t \ge t0 \\ 0 & \text{otherwise} \end{cases}$$

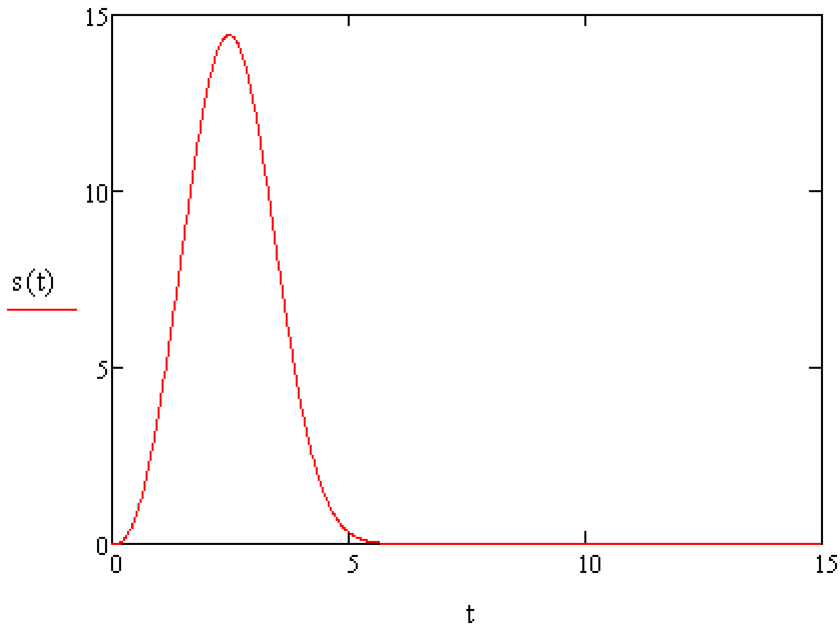


Рисунок 9.1 – График полезного сигнала

Пока речь идет о постоянном времени, на самом деле для наложения шума на сигнал и определения временного положения сигнала нужно перейти к дискретному времени.

Сигнал в дискретном времени будет выглядеть следующим образом:

$$s(k) = a \cdot (k \cdot \Delta t - v \cdot \Delta t)^m \cdot exp\left[\frac{-(k \cdot \Delta t - v \cdot \Delta t)^q}{2 \cdot b}\right], k \ge \frac{t0}{\Delta t},$$

где k – дискретное время;

v – дискретное значение временного положения;

Δt – шаг, с которым идет разбиение интервала.

Переход к дискретному времени выполнить самостоятельно.

9.1.1 Длительность сигнала

Важной характеристикой сигнала является его длительность. Для начала нужно определить *tu*: для этого надо взять уровень 0.1 от максимального, методом перебора определить точки пересечения t_1 и t_2, тогда длительность сигнала будем определять как разность между этими значениями. Максимальный уровень сигнала найдем, взяв производную от функции сигнала и приравняв ее к нулю, получим значения t, подставив в функцию, найдем максимум.

$$L = 0.1 \cdot a$$

Решить простое уравнение:

$$s(t) = L \text{ или } a \cdot (t - t0)^m \cdot exp\left[\frac{-(t-t0)^q}{2 \cdot b}\right] = 0.1 \cdot a$$

Решить данное уравнение можно несколькими способами:

- Используя функцию *root(f(x),x)*;
- Используя функцию *solve*.

Находим корни уравнения: t_1 и t_2. Далее находим $tu = t_1 - t_2$.

9.1.2 Энергия сигнала

Как и любая электромагнитная волна, сигнал имеет энергию. Эта энергия определяется по следующей формуле (как площадь подынтегральной кривой):

$$E(s) = \int_{t_1}^{t_2} (s(t))^2 \, dt$$

9.2 Случайный шумовой процесс

В реальности вместе с полезным сигналом всегда присутствует шум. Из-за него и возникают помехи, трудности передачи и соответственно затрудняется прием, также нужно учесть, что шум всегда разный, следовательно, смесь сигнала с шумом будет всегда разной – в этом и состоит вся сложность данной работы.

9.2.1 Параметры, определяющие шум

Шум имеет следующий вид:

$$n_{(k+1)} = (1 - p) \cdot n_k + \sigma \cdot \sqrt{2 \cdot p} \cdot x_k,$$

где p – интенсивность шумового процесса;

σ – амплитуда шумового процесса;

x_k – случайная составляющая шумового процесса, задается с помощью функции rnd.

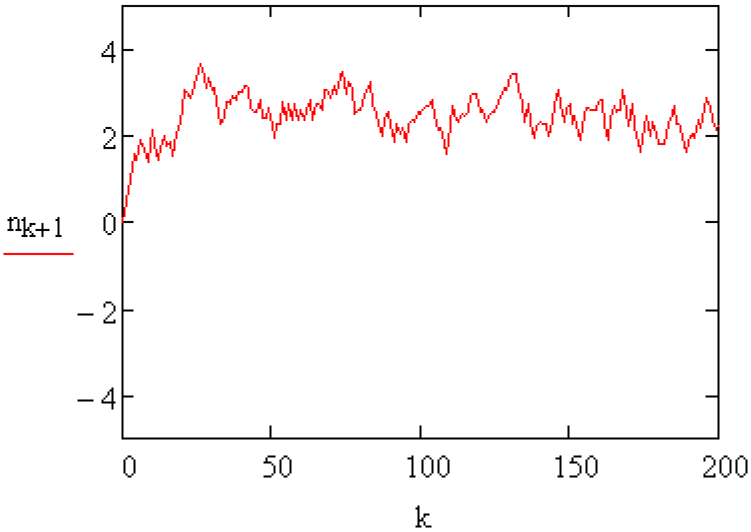


Рисунок 9.2 – График шумового процесса

9.3 Генерация смеси сигнала и шума

Как было сказано раньше, всегда имеется смесь положительного сигнала с шумом. Смесь определяется по следующей формуле:

$$z(k) = s(k) + n_{(k+1)},$$

где $s(k)$ – полезный сигнал;

$n_{(k+1)}$ – шумовой процесс.

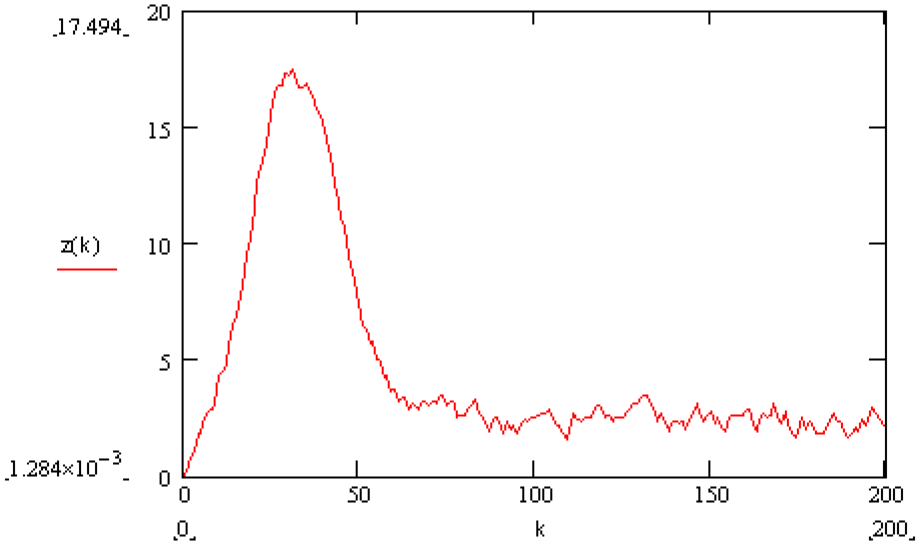


Рисунок 9.3 – График смеси полезного сигнала с шумом

Получившаяся смесь трудна для дальнейшего анализа из-за влияния шума. Можно увеличить полезный сигнал (амплитуду) или уменьшить шум, тогда влияние шума будет меньше, но в действительности этого не происходит, т.к. пока еще не научились передавать полезный сигнал, значительно превышающий шум, и полностью избавляться от влияния шума, поэтому применяют сглаженную смесь, определяемую следующей формулой:

$$z(k) = \frac{1}{m} \cdot \sum_{j=1}^{m} z(k-j),$$

где j – начальное значение отсчета;

m – значение, которое определяет «качество» сглаживания. Чем оно больше, тем сглаживание лучше.

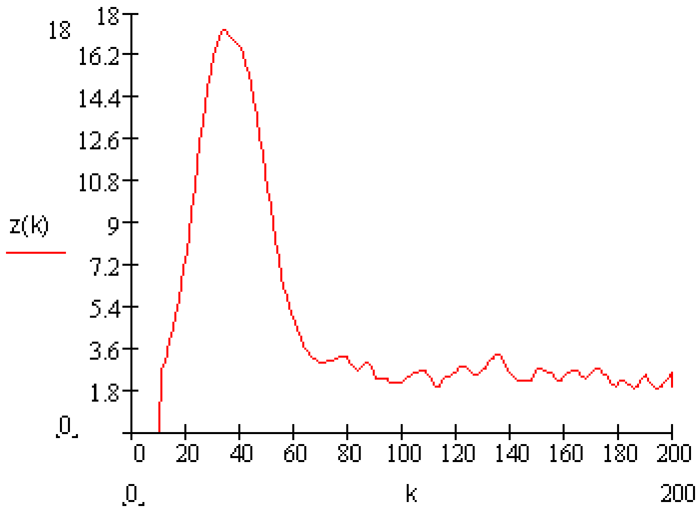


Рисунок 9.4 – График сглаженной функции

9.4 Оценка временного положения полезного сигнала

9.4.1 Алгоритм нахождения временного положения полезного сигнала

Существует несколько способов определения временного положения, самый качественный – метод наименьших квадратов. Кстати, от того, на сколько хорош используемый метод, зависит качество определения временного положения.

Предложить свой метод обработки сигнала самостоятельно.

В результате должно получится одно временное положение. Конечно, по одному измерению нельзя сделать соответствующие выводы и расчеты. Поэтому данный алгоритм повторяется много раз, т.е. составляется большой цикл (так называемые *N* реализации). В результате работы программы должен получится массив значений, размер которого определяется числом измерений (реализаций). Эти значения далее подвергаются обработке.

9.5 Статистическая обработка результатов временного положения сигнала

Получив массив, состоящий из значений временного положения смеси сигнала с шумом, можно начать его анализ, а именно: построить гистограмму распределения оценок временного положения и рассчитать среднеквадратичную погрешность вычислений.

9.5.1 Построение гистограммы

В результате реализаций шума *N* раз получаются различные оценки положения сигнала. Все значения попадают в интервалы, лежащие рядом с истинной оценкой, определенной нами из графика функции сигнала. Для того чтобы увидеть, в какие интервалы попадают значения оценок, необходимо построить гистограмму распределения. Воспользуемся функцией *histogram(M)*, которая определяет интервал между минимальным и максимальным значениями оценок, затем разбивает этот интервал на *M* интервалов и показывает, какое количество оценок положения в какой интервал попало. Гистограмма представлена на рисунке 9.5.

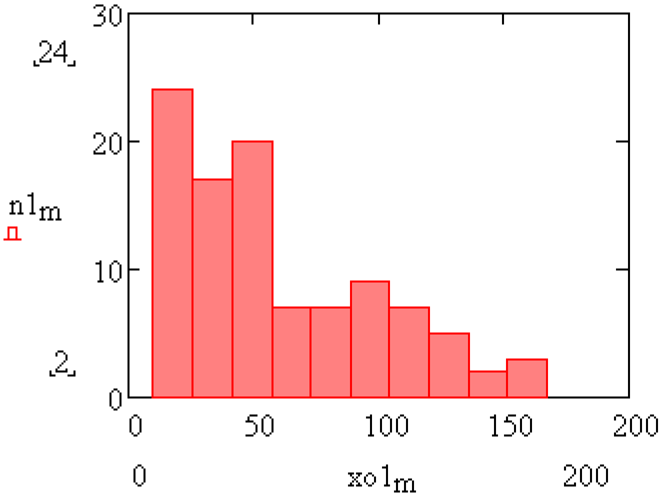


Рисунок 9.5 – Гистограмма распределения оценок положения сигнала

9.5.2 Расчет среднеквадратичного отклонения (СКО)

Т.к. временное положение смеси сигнала с шумом всегда разное, из-за влияние шума, то найденные значения будут «колебаться» возле действительного, определяемого из графика сигнала, следовательно, временное положение будет определяться с некоторой погрешностью, определяемой по следующей формуле:

$$\delta = \sqrt{\frac{1}{N} \cdot \sum_{i=1}^{N} (\varepsilon_i)^2},$$

где N – число опытов;

ε_i – погрешность измерения одного опыта.

9.5.3 Зависимость СКО от отношения сигнал/шум

Увеличивая амплитуду сигнала, увеличивается его максимальное значение, поэтому выделить сигнал из шума становится легче, следовательно, ошибка определения положения сигнала должна уменьшаться, что и показано на рисунке 9.6.

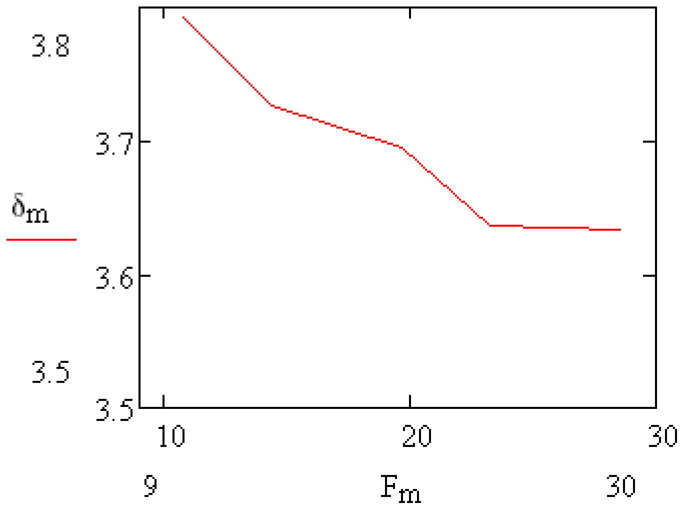


Рисунок 9.6 – Зависимость СКО от отношения сигнал/шум

9.6 Задание 10 для самостоятельной проработки материала

1. Исследовать сигнал, в п. 9.1
2. Исследовать следующие виды сигналов:

$$s(t) =$$

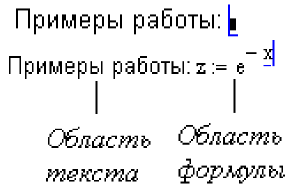
Составить блок-схему программы

Приложение 1

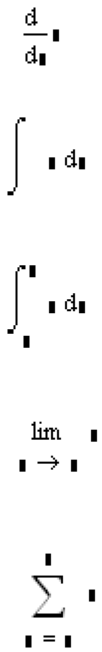
Приемы работы в Mathcad (краткий справочник)

№	Наименование	Пояснения	Ввод данных	Вывод на экран
		1. Общие приемы		
1	Курсор	Указывает позицию ввода с клавиатуры. Перемещается клавишами *Enter* и стрелочными клавишами	Щелчком левой кнопки	+
2	Объекты Mathcad, операции с ними	Формула или ее часть Группа объектов для проведения общих действий (перемещение, копирование, удалении…). Копировать: <Ctrl>+<C> Вырезать: <Ctrl>+<X> Вставлять из буфера: <Ctrl>+<V>	Охват синим контуром появляется автоматически. При необходимости область охвата расширяется клавишей *Пробел* Обводка контура мышью при нажатой левой кнопке мыши. После этого можно копировать, перетаскивать мышью, удалять	$f(x,y) := \sin\left(x^2\right) + \cos\left(\sqrt{y}\right)$ $x := 3$ $y := 7$ Given $2\cdot x + 6\cdot y = 1$ $3\cdot x - 2\cdot y = -3$ $z := \mathrm{Find}(x,y)$ $z = \begin{pmatrix} -0.727 \\ 0.409 \end{pmatrix}$

3	Идентификаторы простых переменных	Латинские, греческие или русские буквы, цифры. Можно показать часть идентификатора на 0.5 строки ниже (для красоты). Строчные и прописные литеры различаются	R точка 1	X x R1 R_1 (набор единицы через "косметическую" точку. Это НЕ индекс!)
4	Греческие буквы	Часто используются как идентификаторы	Ввести похожую латинскую букву, затем нажать <Ctrl>+<g> или *View-Toolbars-Greek*	α β γ δ ε
5	Операция присваивания	Идентификатор := значение Область действия – правее и/или ниже оператора присваивания, до строки присваивания этому оператору нового значения	Клавишей *ДВОЕТОЧИЕ*	$R_1 := 12$
6	Операция присваивания глобального значения	Идентификатор ≡ значение Область действия – весь лист Mathcad, независимо от места на листе, где это присваивание осуществлено	Из меню *View-Toolbars-Evaluation*, иконка со знаком ≡	$X \equiv 15$
7	Вставка текста (комментариев)	В среде Mathcad можно создавать описания, методические указания, электронные книги.	Нажать клавишу *КАВЫЧКИ* и набирать текст как обычно в текстовом	Примеры работы

		Текст нужно вводить в специальные области (называются text-regions – текстовые регионы)	редакторе. Или из меню *Insert-Text-Regions*	
8	Вставка формул в текстовый регион	Можно вписывать «живые» формулы прямо в поясняющий текст	Находясь в области текста, из меню *Insert-Math- Region*	
2. Арифметические, алгебраические и логические операторы				
1	Знаки алгебраических действий	Умножение Деление Возведение числа или матрицы в степень Извлечения корня Абсолютное значение числа	* ½ 2^n \\ Shift :	$\blacksquare\cdot\blacksquare$ $\frac{1}{2}$ 2^n $\sqrt{\blacksquare}$ $\lvert\blacksquare\rvert$
2	Знаки отношения	Больше или равно Меньше или равно Не равно Равно (знак для уравнений, условий…), жирный знак равенства	<Ctrl>+<)> <Ctrl>+<(> <Ctrl>+<3> <Ctrl>+<=>	$\blacksquare \geq \blacksquare$ $\blacksquare \leq \blacksquare$ $\blacksquare \neq \blacksquare$ $\blacksquare = \blacksquare$
3	Число *π*	В расчетах Mathcad автоматически воспринимает *π* как 3.1415926	Набрать латинскую букву *p* (строчную) и нажать <Ctrl>+<g>	π
4	Число *e* (основание натуральных логарифмов)	В расчетах Mathcad автоматически воспринимает *e* как 2.718281828	Обычный набор латинских букв	*e*, можно *exp*
5	Комплексные числа	В Mathcad реализована алгебра комплексных чисел	y:a<Ctrl>+<g>+1*i+b<Ctrl>+<g> 1i набирается без пробела	$\alpha := 0.5$ $\beta := -5$ $y := \alpha + i\cdot\beta$

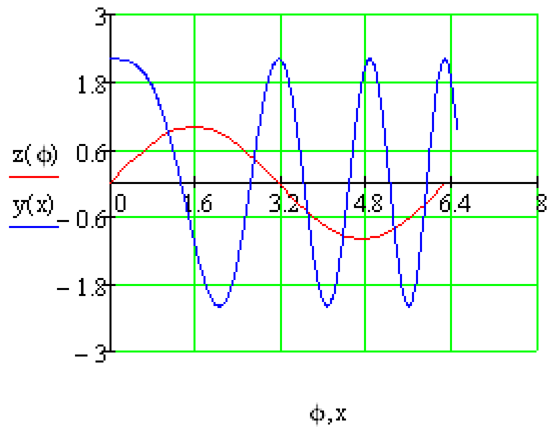
3. Переменные с индексами (элементы массива, область значений аргументов функций)				
1	Присваивание области определения переменных	Определяется диапазон изменения переменной с указанием шага. Если шаг не указать, то по умолчанию Mathcad считает его равным 1	Идентификатор *ДВОЕТОЧИЕ* начальное значение *ЗАПЯТАЯ* шаг *ТОЧКА_С_ЗАПЯТОЙ* конечное значение. Если нужен шаг, равный 1, то Идентификатор *ДВОЕТОЧИЕ* начальное значение *ТОЧКА_С_ЗАПЯТОЙ* конечное значение	$x := 0, 0.001 .. 10$ $Y := Y_{min}, Y_{min} + \Delta Y .. Y_{max}$ $j := 0 .. N$
2	Индексы массива	Если массив – вектор, указывается только номер элемента. Если массив двумерный или массив является строкой, нужно указать два индекса. Нумерация – с нуля.	Идентификатор *КВАДРАТНАЯ ОТКРЫВАЮЩАЯСЯ СКОБКА* [индексы через *ЗАПЯТЫЕ*	$i := 0 .. 25$ $j := 0 .. 100$ $X_j := 2 \cdot j$ $A_{i,j} := \frac{j}{i+1}$

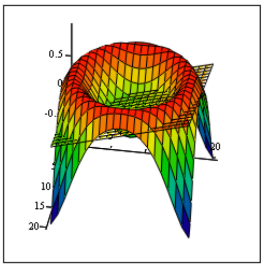
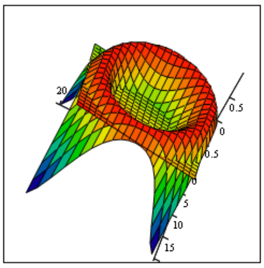
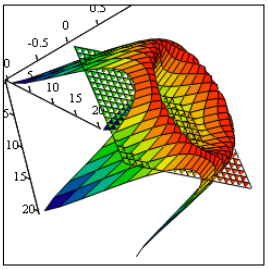
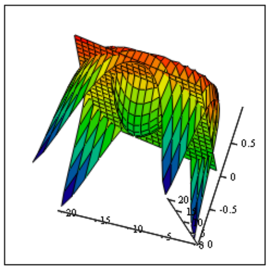
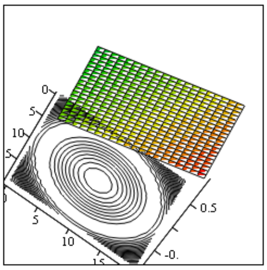
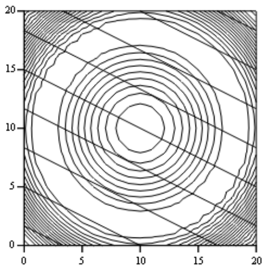
3	Вывод таблицы значений элементов массива	Массив выводится в форме таблицы. Если таблица имеет больше строк, чем принято по умолчанию (15), автоматически появляется линейка прокрутки	x ДВОЕТОЧИЕ 0 ЗАПЯТАЯ 0.1 ТОЧКА_С_ЗАПЯТОЙ 1 y(x) ДВОЕТОЧИЕ sin(x) x= y(x)=	x := 0,0.1..1 y(x) := sin(x) x = 0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1 y(x) = 0 0.1 0.199 0.296 0.389 0.479 0.565 0.644 0.717 0.783 0.841
4. Шаблоны операторов математического анализа				
1	Знаки операций математического анализа	Шаблон производной Шаблон неопределенного интеграла Шаблон определенного интеграла Шаблон предела	<Shift>+<?> или меню *View-Toolbars-Calculus* – иконка со знаком производной <Ctrl>+<i> или меню *View-Toolbars-Calculus* – иконка со значком интеграла <Shift>+<&> или меню *View-Toolbars-Calculus* – иконка со значком интеграла <Ctrl>+<L> или меню *View-Toolbars-Calculus* – иконка со	

		Шаблон суммы	значком lim <Ctrl>+<Shift>+<4> или меню *View-Toolbars-Calculus* – иконка со значками $\sum$	
5. Шаблоны для матричной алгебры				
1	Знаки матричной алгебры	Шаблон матрицы	<Ctrl>+<m> или меню *Insert-Matrix*. В окошке указать число строк (rows) и столбцов (columns)	$A := \begin{pmatrix} \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare \end{pmatrix}$ $A := \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$
		Транспонирование	Идентификатор матрицы охватить синим контуром, нажать <Ctrl>+<l>	$A^T = \begin{pmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{pmatrix}$ $A^{\langle 0 \rangle} = \begin{pmatrix} 1 \\ 4 \\ 7 \end{pmatrix}$
		Выделение одного из столбцов матрицы в форме вектора. Нумерация столбцов матрицы – с нуля	Набрать идентификатор матрицы, затем <Ctrl>+<^>	$v := \begin{pmatrix} 3 \\ 4 \end{pmatrix}$ $\lvert v \rvert = 5$
		Длина вектора	Набрать идентификатор вектора, затем <Shift>+<|>	$A := \begin{pmatrix} 1 & 2 \\ 5 & 6 \end{pmatrix}$
		Определитель квадратной матрицы	Набрать идентификатор матрицы, затем <Shift>+<|>	$\lvert A \rvert = -4$
6. Часто употребляющиеся функции				
1	Тригонометрическ	*sin*(*x*), *cos*(*x*), *tan*(*x*) (тангенс). По умолчанию область		

	ие функции	определения тангенса – от $-\frac{\pi}{2}$ до $+\frac{\pi}{2}$, область определения функций синус и косинус – от 0 до $2 \cdot \pi$		
2	Логарифмы с заданным основанием	По умолчанию основание полагается равным 10	Основание z := log(10, 2) u := log(2) Число, логарифм которого нужно определить	
3	Экспоненциальная функция	Для ее ввода не требуется задавать числовое значение основанию натуральных логарифмов	Для набора степени используйте клавиши <Shift>+<^>	$e^{-0.1\cdot x}$ или exp(x)
4	Условная функция	Выдает различные значения (*ОТВЕТ_1* или *ОТВЕТ_2*) в зависимости от выполнения или невыполнения условия *УСЛОВИЕ*: if(*УСЛОВИЕ*, *ОТВЕТ_1*, *ОТВЕТ_2*)	В записи условий используются знаки отношения (жирный знак равенства <Ctrl>+<=> и др.) Условная функция начисления стипендии: y(ball) := if(ball ≥ 95, 1000, if(ball ≥ 90, 800, if(ball ≥ 80, 600, 0))) y(79) = 0 y(86) = 600 y(90) = 800 y(96) = 1×10^3	
5	Округление снизу floor(); сверху ceil() и отсечение от дробной части trunc()	Округление снизу floor() и сверху ceil() до ближайшего целого числа. Функция trunc() – отсечение дробной части	z := 21.1548 y := −21.1548 floor(z) = 21 floor(y) = −22 trunc(z) = 21 trunc(y) = −21 ceil(z) = 22 ceil(y) = −21 Функция floor() и trunc() возвращают одинаковые результаты только для положительных чисел	
7. Функция для работы с матрицами и векторами				
1	Минимальный и максимальный элемент массива, матрицы, вектора. Число строк и столбцов с массиве	max(A), min(A) rows(A) – число строк; cols(A) – число столбцов	Вывод значений – с помощью обычного (нежирного) знака неравенства	$A := \begin{pmatrix} -2 & 0 \\ 1 & 3 \\ 8 & 19 \end{pmatrix}$

				max(A) = 19 min(A) = −2 rows(A) = 3 cols(A) = 2
2	Единичная матрица	Автоматически формируется диагональная матрица с единицами в диагонали	Вывод матрицы на экран – с помощью обычного (нежирного) знака равенства	n := 3 $\text{identity}(n) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$
3	Собственные числа и собственные векторы квадратной матрицы. Пример: найдем для матрицы $A := \begin{pmatrix} -17 & -30 & 1 \\ -50 & 90 & 0 \\ -70 & 0 & -9 \end{pmatrix}$	Автоматически решается уравнение $A \cdot x = \alpha x$ для собственных чисел α_i, *I=1,…,n* матрица A размерности $n \times n$. Автоматически определяются собственные нормированные векторы (единичной длины)	Вектор собственных чисел $\text{eigenvals}(A) = \begin{pmatrix} 102.488 \\ -25.785 \\ -12.703 \end{pmatrix}$ Матрица, столбцы которой - собственные векторы $\text{eigenvecs}(A) = \begin{pmatrix} -0.24 & 0.232 & 0.053 \\ 0.959 & 0.1 & 0.026 \\ 0.15 & 0.968 & 0.998 \end{pmatrix}$	
4	Объединение матриц «сверху вниз»: Получение единой матрицы из двух: Первая – верхний блок, вторая – нижний Дано: $A := \begin{pmatrix} 1 & 3 & 5 \\ 7 & 9 & 11 \end{pmatrix}$ $B := \begin{pmatrix} 2 & 4 & 6 \\ 8 & 10 & 12 \end{pmatrix}$ Получить: $C = \begin{pmatrix} \boxed{\begin{matrix} 1 & 3 & 5 \\ 7 & 9 & 11 \end{matrix}} \\ \boxed{\begin{matrix} 2 & 4 & 6 \\ 8 & 10 & 12 \end{matrix}} \end{pmatrix}$		$A := \begin{pmatrix} 1 & 3 & 5 \\ 7 & 9 & 11 \end{pmatrix}$ $B := \begin{pmatrix} 2 & 4 & 6 \\ 8 & 10 & 12 \end{pmatrix}$ C := stack(A, B) $C = \begin{pmatrix} 1 & 3 & 5 \\ 7 & 9 & 11 \\ 2 & 4 & 6 \\ 8 & 10 & 12 \end{pmatrix}$	

<table>
<tr><td>5</td><td colspan="2">Объединение матриц «слева направо»:
Получение единой матрицы из двух:
Первая – левый блок, вторая – правый
Дано:
$A := \begin{pmatrix} 1 & 3 & 5 \\ 7 & 9 & 11 \end{pmatrix}$ $B := \begin{pmatrix} 2 & 4 & 6 \\ 8 & 10 & 12 \end{pmatrix}$
Получить: $D = \left(\boxed{\begin{matrix} 1 & 3 & 5 \\ 7 & 9 & 11 \end{matrix}} \; \boxed{\begin{matrix} 2 & 4 & 6 \\ 8 & 10 & 12 \end{matrix}} \right)$</td><td>$A := \begin{pmatrix} 1 & 3 & 5 \\ 7 & 9 & 11 \end{pmatrix}$ $B := \begin{pmatrix} 2 & 4 & 6 \\ 8 & 10 & 12 \end{pmatrix}$
D := augment(A,B)
$D = \begin{pmatrix} 1 & 3 & 5 & 2 & 4 & 6 \\ 7 & 9 & 11 & 8 & 10 & 12 \end{pmatrix}$</td></tr>
<tr><td>6</td><td colspan="2">Вырезание блока из матрицы. Получение новой матрицы – части исходной
Дано:
$A := \left(\begin{matrix} 1 \\ 7 \end{matrix} \; \boxed{\begin{matrix} 3 & 5 & 2 \\ 9 & 11 & 8 \end{matrix}} \; \begin{matrix} 4 & 6 \\ 10 & 12 \end{matrix} \right)$
Получить: $B = \begin{pmatrix} 3 & 5 & 2 \\ 9 & 11 & 8 \end{pmatrix}$</td><td>$A := \begin{pmatrix} 1 & 3 & 5 & 2 & 4 & 6 \\ 7 & 9 & 11 & 8 & 10 & 12 \end{pmatrix}$
Границы вырезаемого блока задаются номерами элементов «по диагонали»: (верхний левый)-(нижний правый)
Номер верхнего левого элемента (строка =0, столбец =1): 0,1
Номер нижнего правого элемента (строка =1, столбец =3): 1,3
B := submatrix(A,0,1,1,3)
$B = \begin{pmatrix} 3 & 5 & 2 \\ 9 & 11 & 8 \end{pmatrix}$</td></tr>
<tr><td colspan="4">8. Графики</td></tr>
<tr><td>1</td><td>График плоскостной
Пример: построить на одном рисунке графики функций $z(\varphi)=\sin(\varphi)$, $y(x) = 2.2 \cdot \cos\left(\sqrt{x^{3.2}}\right)$</td><td>Нужно задать область изменения аргумента функции. На одном графике можно показывать разные функции одного или разных аргументов. График можно формировать щелчком левой мыши по его полю</td><td>$z(\phi) := \sin(\phi)$ $\phi := 0, 0.1 \cdot \pi .. 2\pi$
$y(x) := 2.2 \cdot \cos\left(\sqrt{x^{3.2}}\right)$ $x := 0, 0.01 .. 6.5$
Шаблон графика выводится клавишами <Shift>+<2> или из меню Insert-Graph-XY Plot
</td></tr>
</table>

2	График двумерный Пример: построить на одном рисунке графики функций $f(x,y) = \sin(x^2 + y^2)$, $Z(X,Y) = 0.1 \cdot X + 0.2 \cdot Y + 0.5$ для областей определения аргументов: $-1.5 \leq x \leq 1.5$, $-1.5 \leq y \leq 1.5$, $-1.5 \leq X \leq 1.5$, $-1.5 \leq Y \leq 1.5$.	Нужно задать целочисленные индексы, с помощью которых образовать массивы данных о значениях функций в ряде точек внутри области определения их аргументов. График можно формировать щелчком левой кнопки мыши по его полю. Готовый график можно рассматривать под различными ракурсами (на рисунке справа показаны 4 ракурса)	$f(x,y) := \sin(x^2 + y^2)$ $Z(X,Y) := 0.1 \cdot X + 0.2 \cdot Y + 0.5$ Индексы $i := 0..20$ $j := 0..20$ Области определения аргументов $x_i := -1.5 + 0.15 \cdot i$ $X_i := -1.5 + 0.15 \cdot i$ $y_j := -1.5 + 0.15 \cdot j$ $Y_j := -1.5 + 0.15 \cdot j$ Массивы точек графика $F_{i,j} := f(x_i, y_j)$ $P_{i,j} := Z(X_i, Y_j)$   F,P F,P   F,P F,P
3	Графики линий равного уровня (Contour Plot) в виде проекций на плоскость аргументов	Нужно сформировать график, щелчком левой кнопки мыши вызвать окно форматирования графика, выбрать вкладку *General* и включить кнопку *Contour Plot*. Можно один из графиков показать как обычный, а другой – в виде	  F,P F,P Оба графика в виде линий равного уровня Первый график - в виде линий равного уровня, второй - обычный

		линий равного уровня	
9. Решение уравнений, нахождение корней уравнения n-го порядка			
1	Нахождение всех корней уравнения n-го порядка. Пример: найти корни уравнения $2 \cdot x^7 + 8 \cdot x^5 + 4 \cdot x^4 - x + 1 = 0$ (жирный знак равенства, <Ctrl>+<=>)	Для нахождения всех корней (включая комплексные) нужно сформировать вектор из коэффициентов уравнения (в порядке возрастания степеней, начиная со свободного слагаемого). Если такая-то степень отсутствует – нужно ввести нуль	$2x^7 + 8x^5 + 4x^4 - x + 1 = 0$ $D := \begin{array}{l} \begin{pmatrix} 1 \\ -1 \\ 0 \\ 0 \\ 4 \\ 8 \\ 0 \\ 2 \end{pmatrix} \end{array} \begin{array}{l} \\ \text{1-й степени} \\ \text{2-й степени} \\ \text{3-й степени} \\ \text{4-й степени} \\ \text{5-й степени} \\ \text{6-й степени} \\ \text{7-й степени} \end{array}$ $z := \mathrm{polyroots}(D)$
2	Решение уравнений, систем уравнений, систем уравнений и неравенств. Пример: найти решение системы уравнений: $0.5 \cdot x + 2 \cdot y^2 = 9$, $-3 \cdot x \cdot y - 2 \cdot y^2 = 15$.	Используется жирный знак равенства (<Ctrl>+<=>) и другие операторы отношений. Если уравнение нелинейное, то решений может быть несколько. При заданном начальном приближении найдется только одно из них. Решение может отсутствовать.	Нужно задать начальное приближение к решению $x := 70$ $y := 5$ Операторная скобка Given-find Given Система уравнений и неравенств $0.5 \cdot x + 2 \cdot y^2 = 9$ $-3 \cdot x \cdot y - 2 \cdot y^2 = 15$ $z := \mathrm{find}(x, y)$ Решение: $z = \begin{pmatrix} 17.673 \\ -0.286 \end{pmatrix} \begin{array}{l} \text{— x} \\ \text{— y} \end{array}$
3	Приближенное решение уравнений, систем	Если использовать скобку *Given-find*, то Mathcad не сможет найти решение. Выведется сообщение о невозможности нахождения точного решения. В этом	

	уравнений, систем уравнений и неравенств, если точное решение найти не удается. Пример: найти решение системы уравнений: $x^3+2\cdot\sqrt{y}+6=0$, $y^3-2\cdot\sqrt{x}+8=0$.	случае можно использовать операторную скобку *Given-Minnerr*, помощью которой находится решение с наилучшими приближением $x := 9 \quad y := 7$ Given $x^3+2\cdot\sqrt{y}+6=0$ $y^3-2\cdot\sqrt{x}+8=0$ Find(x,y) = Точное решение не найдено $x := 9 \quad y := 7$ Given $x^3+2\cdot\sqrt{y}+6=0$ $y^3-2\cdot\sqrt{x}+8=0$ $\mathrm{Minerr}(x,y)=\begin{pmatrix}0.159\\-1.915\end{pmatrix}$ Приближенное решение найдено. В этом случае обязательна проверка (подстановка найденного решения в уравнения и оценка степени расхождения между левой и правой частями уравнения)

Приложение 2

Алфавитный список встроенных функций

В этом приложении перечисляются встроенные функции Mathcad вместе с их краткими описаниями. Эти функции подробно описаны в главах.

Значком E обозначены функции, реализованные только в Mathcad PLUS.

В приведенной таблице:

x и *y* обозначают вещественные числа.

z обозначает число, вещественное либо комплексное.

m, n, i, j и *k* обозначают целые числа.

v, u и все имена, начинающиеся с v, обозначают векторы.

A и *B* обозначают матрицы либо векторы.

M и *N* обозначают квадратные матрицы.

F обозначает векторнозначную функцию.

file обозначает либо имя файла, либо файловую переменную, присоединённую к имени файла.

Все углы измеряются в радианах. Многозначные функции и функции с комплексным аргументом всегда возвращают главное значение.

Имена приведенных функций не чувствительны к шрифту, но чувствительны к регистру – их следует печатать в точности, как они приведены.

№	Функция	Возвращает…
1	*acos(z)*	Арккосинус
2	*acosh(z)*	Обращение гиперболического косинуса
3	*angle(x,y)*	Угол между осью абсцисс и вектором с координатами (*x,y*)
4	*APPEND(file)*	Добавляет число в файл данных *file*
5	*APPENDPRN(file)*	Добавляет матрицу в структурированный файл данных *file*
6	*arg(z)*	Угол в комплексной плоскости между положительным направлением вещественной оси и числом *z*
7	*asin(z)*	Арксинус
8	*asinh(z)*	Обращение гиперболического синуса
9	*atan(z)*	Арктангенс
10	*atanh(z)*	Обращение гиперболического тангенса
11	*augment(A,B)*	Объединяет матрицы-аргументы бок о бок. *A* и *B* должны иметь одинаковый размер.
12	*ceil(x)*	Наименьшее целое $\geq x$. *x* должен быть вещественным
13	*cfft(A)*	Быстрое преобразование Фурье комплексных данных. Возвращает массив того же размера, что

		и аргумент
14	*CFFT(A)*	То же, что и *cfft(A)*, но использует другую форму преобразования Фурье
15	*cholesky(M)*	Нижняя треугольная матрица L такая, что $L \cdot L^T$. Матрица *M* должна быть симметричной.
16	*cnorm(x)*	Функция нормального распределения
17	*cols(A)*	Число столбцов массива *A*. Возвращает скаляр
18	*cond1(M)*	Число обусловленности матрицы *M*, основанное на норме L_1
19	*cond2(M)*	Число обусловленности матрицы *M*, основанное на норме L_2
20	*conde(M)*	Число обусловленности матрицы *M*, основанное на евклидовой норме
21	*condi(M)*	Число обусловленности матрицы *M*, основанное на равномерной норме
22	*corr(A,B)*	Корреляция массивов *A* и *B*, имеющих одинаковый размер
23	*cos(z)*	Косинус
24	*cosh(z)*	Гиперболический косинус
25	*cot(z)*	Котангенс
26	*coth(z)*	Гиперболический котангенс
27	*csc(z)*	Косеканс
28	*csch(z)*	Гиперболический косеканс
29	*csort(A,n)*	Сортирует столбцы по возрастанию элементов в строке *n*
30	*cspline(vx,vy)*	Коэффициенты кубического сплайна. *vx* и *vy* вещественные векторы одного размера. Элементы *vx* должны идти в возрастающем порядке
31	*cspline(Mxy,Mz)*	Вектор, используемый функцией *interp* для интерполяции данных из *Mxy* и *Mz*
32	*cvar(A,B)*	Ковариация элементов из *A* и *B*. *A* и *B* должны быть одного размера
33	*diag(v)*	Диагональная матрица, имеющая на диагонали элементы из *v*
34	E $dbeta(x,s_1,s_2)$	Плотность *бэта*-распределения
35	*dbinom(k,n,p)*	*P(X=k)*, когда *X* имеет биномиальное распределение
36	E *dcauchy(x,l,s)*	Плотность распределения Коши
37	*dchisq(x,d)*	Плотность *хи*-квадрат распределения
38	E *dexp(x,r)*	Плотность экспоненциального распределения
39	$dF(x,d_1,d_2)$	Плотность *F*-распределения

40	E *dgamma(x,s)*	Плотность *Гамма*-распределения
41	E *dgeom(k,p)*	*P(X=k)*, когда *X* имеет геометрическое распределение
42	E *dlnorm(x,m,s)*	Плотность логнормального распределения
43	E *dlogis(x,l,s)*	Плотность логистического распределения
44	E *dnbinom(k,n,p)*	*P(X=k)*, когда случайная величина *X* имеет отрицательное биномиальное распределение
45	*dnorm(x,m,s)*	Плотность нормального распределения
46	*dpois(k,l)*	*P(X=k)*, когда случайная величина *X* имеет распределение Пуассона
47	*dt(x,d)*	Плотность распределения Стьюдента
48	*dunif(x,a,b)*	Плотность равномерного распределения
49	E *dweibull(x,s)*	Плотность распределения Вейбулла
50	*eigenvals(M)*	Вектор из собственных значений матрицы *M*
51	*eigenvec(M,z)*	Нормированный собственный вектор матрицы *M*, соответствующий её собственному значению *z*
52	*eigenvecs(M)*	Матрица, чьими столбцами являются собственные векторы матрицы *M*. Порядок расположения собственных векторов соответствует порядку собственных значений, возвращаемых *eigenvals*
53	*exp(z)*	Экспонента e^z
54	*find(var1,var2,…)*	Значения *var1,var2,...*, доставляющие решение системе уравнений. Число возвращаемых значений равно числу аргументов
55	*fft(v)*	Быстрое преобразование Фурье вещественных данных. *v* должен быть вещественным вектором с 2^n элементами, где *n* есть целое число. Возвращает вектор размера $2^{n-1}+1$
56	*FFT(v)*	То же, что и *fft(v)*, но использует другую форму преобразования Фурье. *v* должен быть вещественным вектором с 2^n элементами, где *n* есть целое число. Возвращает вектор размера $2^{n-1}+1$
57	*floor(x)*	Наибольшее целое $\leq x$. *x* должен быть вещественным
58	E *genfit(vx,vy,vg,F)*	Возвращает вектор, содержащий *n* параметров $u_0,u_1,...u_{n-1}$, которые обеспечивают наилучшее приближение данных из *vx* и *vy* функцией *f*, зависящей от *x* и параметров $u_0,u_1,...u_{n-1}$. *F* – функция, которая возвращает *n*+1-мерный вектор, содержащий *f* и ее частные производные

		относительно параметров. vg есть n-мерный вектор начальных значений для n параметров
59	*geninv(A)*	Возвращает левую обратную к A матрицу
60	*genvals(M,N)*	Возвращает вектор v обобщенных собственных значений, соответствующих решению задачи $M \cdot x = v_i \cdot N \cdot x$. Матрицы M и N должны быть вещественными
61	*genvecs(M,N)*	Матрица, столбцы которой являются нормированными обобщёнными собственными векторами. Вектор x, находящийся в n-ом столбце является решением задачи $M \cdot x = v_n \cdot N \cdot x$, где v_n есть соответствующий элемент вектора v, возвращаемого *genvals*
62	*hist(int,A)*	Возвращает вектор, представляющий частоты, с которыми величины, содержащиеся в векторе A, попадают в интервалы, представляемые вектором *int*
63	*I0(x)*	Функция Бесселя I_1. Аргумент должен быть вещественным
64	*I1(x)*	Функция Бесселя I_2. Аргумент должен быть вещественным
65	*In(m,x)*	Функция Бесселя I_m. x должен быть вещественным; $0 \le m \le 100$
66	*icfft(A)*	Обратное преобразование Фурье, соответствующее *cfft*. Возвращает массив того же размера, что и аргумент
67	*ICFFT(A)*	Обратное преобразование Фурье, соответствующее *CFFT*. Возвращает массив того же размера, что и аргумент
68	*identity(n)*	Единичная матрица с числом строк n. n должно быть натуральным числом
69	*if(cond,x,y)*	Возвращает значение x, если *cond* отличнен от 0 (истина). Возвращает значение y, если *cond* равен 0 (ложь)
70	*ifft(v)*	Обратное преобразование Фурье, соответствующее *fft*. Принимает вектор размера $2^{n-1}+1$, где n - целое. Возвращает вещественный вектор размера 2^n
71	*IFFT(v)*	Обратное преобразование Фурье, соответствующее *FFT*. Принимает вектор размера $2^{n-1}+1$, где n - целое. Возвращает вещественный вектор размера 2^n

72	*Im(z)*	Мнимая часть *z*
73	*intercept(vx,vy)*	Возвращает скаляр: смещение по оси *y* линии регрессии в смысле наименьших квадратов для данных из *vx* и *vy*
74	*interp(vs,vx,vy,x)*	Возвращает интерполируемое значение *y*, соответствующее аргументу *x*. Вектор *vs* вычисляется на основе векторов данных *vx* и *vy* одной из функций *lspline*, *pspline* или *cspline*
75	*interp(vs,Mxy,Mz,v)*	Возвращает интерполируемое значение *z*, соответствующее точкам $x=v_0$ и $x=v_1$. Вектор *vs* вычисляется *lspline*, *pspline* или *cspline* на основе данных из *Mxy* и *Mz*
76	*iwave(v)*	Обратное волновое преобразование, соответствующее *wave*. Берет вещественный вектор размера 2^n, где *n* есть целое число
77	*J0(x)*	Функция Бесселя J_1. Аргумент должен быть вещественным
78	*J1(x)*	Функция Бесселя J_2. Аргумент должен быть вещественным
79	*Jn(m,x)*	Функция Бесселя J_m. *x* должен быть вещественным; $0 \leq m \leq 100$
80	*K0(x)*	Функция Бесселя K_1. Аргумент должен быть вещественным
81	*K1(x)*	Функция Бесселя K_2. Аргумент должен быть вещественным
82	*Kn(m,x)*	Функция Бесселя K_m. *x* должен быть вещественным; $0 \leq m \leq 100$
83	*ksmooth(vx,vy,b)*	Возвращает *n*-мерный вектор, созданный сглаживанием при помощи гауссова ядра данных из *vy*. *vy* и *vx* есть *n*-мерные векторы вещественных чисел. Параметр *b* управляет окном сглаживания и должен быть установлен в несколько раз больше величины интервала между точками *x*
84	*last(v)*	Индекс последнего элемента в векторе *v*
85	*length(v)*	Число элементов в векторе *v*
86	*linfit(vx,vy,F)*	Возвращает вектор, содержащий коэффициенты, используемые, 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
87	*linterp(vx,vy,x)*	Использует векторы данных *vx* и *vy*, чтобы возвратить линейно интерполируемое значение *y*, соответствующее третьему аргументу *x*
88	*ln(z)*	Натуральный логарифм *z*
89	E *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*>0) определяет, насколько большие окрестности *loess* будет использовать при выполнении локального приближения
90	E *loess(Mxy,vz,span)*	Возвращает вектор, требуемый *interp*, чтобы найти набор полиномов второго порядка, которые наилучшим образом приближают определённые окрестности выборочных точек, определенных в массивах *Mxy* и *vz*. *Mxy* есть $m \times 2$ мерная матрица, содержащая координаты *x* и *y*. *vz* есть *m*-мерный вектор, содержащий координаты *z*, соответствующие *m* точкам, определенным в *Mxy*. Аргумент *span* > 0) определяет, насколько большие окрестности *loess* будет использовать при выполнении локального приближения
91	*log(z)*	Логарифм *z* по основанию 10
92	*lsolve(M,v)*	Вектор *x* такой, что $M \cdot x = v$
93	*lspline(vx,vy)*	Коэффициенты кубического сплайна, имеющего на концах равные нулю вторую и третью производные. *vx* и *vy* – вещественные векторы одного размера. Элементы *vx* должны идти в возрастающем порядке
94	*lspline(Mxy,Mz)*	Вектор, используемый функцией *interp* для интерполяции данных из *Mxy* и *Mz*. Интерполирующая поверхность имеет на границе сетки, определяемой *Mxy*, равные нулю производные выше первого порядка
95	*lu(M)*	Матрица, содержащая составленные бок о бок в указанном порядке матрицы *P*, *L* и *U*, имеющие

		одинаковый размер с M и удовлетворяющие уравнению $P \cdot M = L \cdot U$. L и U есть нижняя и верхняя треугольные матрицы соответственно
96	*matrix(m,n,f)*	Создает матрицу, в которой *(i,j)*-й элемент содержит *f(i,j)*, где *i*=0, 1,...*m* и *j*=0, 1,...*n*
97	*max(A)*	Наибольший элемент в *A*. Если массив *A* комплексный, возвращает $\min(Re(A)) + i \cdot \max(Im(A))$
98	*mean(A)*	Среднее значение элементов массива *A*
99	*median(A)*	Медиана элементов массива *A*
100	*medsmooth(vy,n)*	Возвращает *m*-мерный вектор, созданный сглаживанием *vy* с помощью скользящей медианы. *vy* есть *m*-мерный вектор вещественных чисел. *n* – ширина окна, по которому происходит сглаживание. *n* должно быть нечетным числом меньшим, чем число элементов в *vy*
101	*min(A)*	Наименьший элемент в *A*. Если массив *A* комплексный, возвращает $\min(Re(A)) + i \cdot \min(im(A))$
102	*minerr(var1,var2,…)*	Значения *var1,var2, …* , доставляющие минимум функционалу невязки системы уравнений и неравенств. Размерность возвращаемого значения равна числу аргументов
103	*mod(x,modulus)*	Остаток от деления *x* на *modulus*. Аргументы должны быть вещественными. Результат имеет тот же знак, что и *x*
104	*norm1(M)*	L_1 норма матрицы *M*
105	*norm2(M)*	L_2 норма матрицы *M*
106	*norme(M)*	Евклидова норма матрицы *M*
107	*normi(M)*	Равномерная норма матрицы *M*
108	E *pbeta(x,s1,s2)*	Функция *бэта*-распределения
109	*pbinom(k,n,p)*	Функция биноминального распределения для *k* успехов в *n* испытаниях
110	E *pcauchy(x,l,s)*	Функция распределения Коши с параметрами масштаба *l* и *s*
111	*pchisq(x,d)*	Функция *хи*-квадрат распределения, в которой *d* > 0 есть число степеней свободы и *x* > 0
112	E *pexp(x,r)*	Функция экспоненциального распределения, в которой *r* > 0 является параметром и *x* > 0
113	$pF(x,d_1,d_2)$	Функция *F*-распределения, в которой d_1, d_2 > 0 есть числа степеней свободы. *x* > 0

114	E *pgamma(x,s)*	Функция *Гамма*-распределения, в которой $s > 0$ есть параметр формы. $x > 0$
115	E *pgeom(k,p)*	Функция геометрического распределения. p есть вероятность успеха. $k \geq 0$ и $0 < p \leq 1$
116	E *plnorm(x,m,s)*	Функция логнормального распределения, в которой m есть логарифм среднего, $s > 0$ есть логарифм стандартного отклонения и $x > 0$
117	E *plogis(x,l,s)*	Функция логистического распределения. l есть параметр расположения . $s > 0$ есть параметр масштаба
118	E *pnbinom(k,n,p)*	Возвращает функцию отрицательного биномиального распределения, в котором $0 < p \leq 1$. n должно быть положительным целым числом
119	*pnorm(x,m,s)*	Функция нормального распределения со средним m и среднеквадратичным отклонением s
120	*polyroots(v)*	Корни многочлена n-ной степени с коэффициентами, расположенными в порядке возрастания степеней в векторе v длины $n+1$
121	*ppoisk(k,l)*	Функция распределения Пуассона. $l > 0$
122	E *predict(v,m,n)*	Возвращает n предсказанных значений, основанных на m последовательных значениях вектора данных v. Элементы в v должны представлять собой значения, взятые через равные интервалы
123	*pspline(vx,vy)*	Коэффициенты кубического сплайна, имеющего на концах равную нулю третью производную. *vx* и *vy* – вещественные векторы одного размера. Элементы *vx* должны идти в возрастающем порядке
124	*pspline(Mxy,Mz)*	Вектор, используемый функцией *interp* для интерполяции данных из *Mxy* и *Mz*. Интерполирующая поверхность имеет на границе сетки, определяемой *Mxy*, равные нулю производные третьего порядка
125	*pt(x,d)*	Функция распределения t-Стьюдента. d есть число степеней свободы, $x > 0$ и $d > 0$
126	*punif(x,a,b)*	Функция равномерного распределения. b и a есть концы отрезка. $a < b$
127	E *pweibull(x,s)*	Функция распределения Вейбулла. $s > 0$
128	E $qbeta(p,s_1,s_2)$	Обращает бета-распределение с параметрами формы s_1 и s_2. $(0 \leq p \leq 1)$ $(s_1, s_2 > 0)$

129	*qbinom(p,n,q)*	Возвращает число успехов в n испытаниях схемы Бернулли при условии, что вероятность успехов не превышает p и r – вероятность успеха на одиночном испытании. $0 < q < 1$ и $0 \le p \le 1$. n есть натуральное число
130	E *qcauchy(p,l,s)*	Обращает распределение Коши с параметром масштаба s и параметром расположения l. $s > 0$. $0 < p < 1$
131	*qchisq(p,n)*	Обращает хи-квадрат распределение, в котором $d > 0$ является числом степеней свободы. $0 \le p < 1$
132	E *qexp(p,r)*	Обращает экспоненциальное распределение, в котором $r > 0$ является параметром. $0 \le p < 1$
133	*qF(p,d$_1$,d$_2$)*	Обращает F -распределение, в котором $d_1, d_2 > 0$ являются числами степеней свободы. $0 \le p < 1$
134	E *qgamma(p,s)*	Обращает *Гамма*-распределение, в котором $s > 0$ является параметром формы. $0 \le p < 1$
135	E *qgeom(p,q)*	Обращает *Гамма*-распределение, в котором $q > 0$ является параметром формы. $0 < p < 1$ и $0 \le q < 1$
136	E *qlnorm(p,m,s)*	Обращает логнормальное распределение, в котором m является натуральным логарифмом среднего значения, $s > 0$ – натуральный логарифм среднеквадратичного отклонения. $0 \le p < 1$
137	E *qlogis(p,l,s)*	Обращает логистическое распределение. l – параметр расположения, $s > 0$ – параметр масштаба. $0 < p < 1$
138	E *qnbinom(p,n,q)*	Обращает отрицательное биномиальное распределение с числом испытаний n и вероятностью успеха в одиночном испытании q. $0 \le q < 1$ и $0 \le p \le 1$
139	*qnorm(p,m,s)*	Обращает нормальное распределение со средним m и среднеквадратичным отклонением s. $0 < p < 1$ и $s > 0$
140	*qpois(p,l)*	Обращает распределение Пуассона. $l > 0$ и $0 \le p \le 1$
141	*qr(A)*	Матрица, чьи первые n столбцов содержат квадратную ортонормированную матрицу Q, а оставшиеся столбцы содержат верхнюю треугольную матрицу R. Матрицы Q и R удовлетворяют уравнению $A = Q \cdot R$, где A есть

		вещественный массив
142	*qt(p,d)*	Обращает t-распределение Стьюдента. d – число степеней свободы. $d > 0$ и $0 < p < 1$
143	*qunif(p,a,b)*	Обращает равномерное распределение. b и a – граничные точки интервала. $a < b$ и $0 \leq p \leq 1$
144	E *qweibull(p,s)*	Обращает распределение Вейбулла. $s > 0$ и $0 < p < 1$
145	*rank(A)*	Ранг вещественной матрицы *A*
146	E *rbeta(m,s₁,s₂)*	Возвращает вектор m случайных чисел, имеющих *бэта*-распределение. $s_1, s_2 > 0$ есть параметры формы
147	*rbinom(m,n,p)*	Возвращает вектор m случайных чисел, имеющих биномиальное распределение. $0 \leq p \leq 1$. *n* есть натуральное число
148	E *rcauchy(m,l,s)*	Возвращает вектор m случайных чисел, имеющих распределение Коши. $s > 0$ есть параметр масштаба. *l* – параметр расположения
149	*rchisq(m,d)*	Возвращает вектор m случайных чисел, имеющих распределение *хи*-квадрат. $d > 0$ есть число степеней свободы
150	*Re(z)*	Вещественная часть числа *z*
151	*READ(file)*	Считывает одиночное значение из файла данных *file*
152	*READPRN(file)*	Считывает матрицу из структурированного файла данных *file*
153	*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*
154	*regress(Mxy,vz,n)*	Возвращает вектор, требуемый *interp*, чтобы найти полином порядка *n*, который наилучшим образом приближает данные из *Mxy* и *vz*. *Mxy* есть $m \times 2$ – мерная матрица, содержащая координаты *x-y*. *vz* есть *m*-мерный вектор, содержащий координаты *z* соответствующие *m* точкам, определенным в *Mxy*
155	*reverse(v)*	Обращает порядок элементов вектора *v*
156	E *rexp(m,r)*	Возвращает вектор m случайных чисел, имеющих экспоненциальное распределение. $r > 0$ – параметр распределения

157	*rF(m,d₁,d₂)*	Возвращает вектор m случайных чисел, имеющих F-распределение. $d_1, d_2 > 0$ есть числа степеней свободы
158	E *rgamma(m,s)*	Возвращает вектор m случайных чисел, имеющих гамма- распределение, $s > 0$ есть параметр формы
159	E *rgeom(m,p)*	Возвращает вектор m случайных чисел, имеющих геометрическое распределение. $0 < p \leq 1$
160	E *rlnorm(m,n,s)*	Возвращает вектор m случайных чисел, имеющих логнормальное распределение, в котором m является натуральным логарифмом среднего значения, а $s > 0$ есть натуральный логарифм среднеквадратичного отклонения
161	E *rlogis(m,l,s)*	Возвращает вектор m случайных чисел, имеющих логистическое распределение, в котором l является параметром расположения, а $s > 0$ есть параметр масштаба
162	E *rnbinom(m,n,p)*	Возвращает вектор m случайных чисел, имеющих отрицательное биномиальное распределение. $0 < p \leq 1$. n есть натуральное число
163	*rnd(x)*	Возвращает равномерно распределенное случайное число между 0 и *x*
164	*rnorm(m,n,s)*	Возвращает вектор m случайных чисел, имеющих нормальное распределение. $s > 0$
165	*root(f(var),var)*	Значение *var*, доставляющее нулевое значение выражению *expr*
166	*rows(A)*	Число строк в массиве *A*
167	*rpois(m,l)*	Возвращает вектор m случайных чисел, имеющих распределение Пуассона. $l > 0$
168	*rref()*	Ступенчатая форма матрицы *A*
169	*rsort(A,n)*	Сортирует строки в порядке возрастания элементов из столбца *n*
170	*rt(m,d)*	Возвращает вектор m случайных чисел, имеющих t-распределение Стьюдента. $d > 0$
171	*runif(m,a,b)*	Возвращает вектор m случайных чисел, имеющих равномерное распределение, в котором b и a являются граничными точками интервала. $a < b$
172	E *rweibull(m,s)*	Возвращает вектор m случайных чисел, имеющих распределение Вейбулла, в котором

		$s > 0$ является параметром формы
173	*sec(z)*	Секанс
174	*sech(z)*	Гиперболический секанс
175	*sin(z)*	Синус
176	*sinh(z)*	Гиперболический синус
177	*slope(vx,vy)*	Возвращает скаляр: наклон линии регрессии в смысле наименьших квадратов для данных из vx и vy
178	*sort(v)*	Сортирует элементы вектора *v*
179	*stack(A,B)*	Массив, созданный помещением матрицы A над матрицей B. Массивы A и B должны иметь одинаковое количество столбцов
180	*stdev(A)*	Возвращает среднеквадратичное отклонение (квадратный корень из дисперсии) элементов $m \times n$ массива A
181	*submatrix(A,ir,jr,ic,jc)*	Субматрица A, состоящая из всех элементов, находящихся на пересечении строк с ir по jr и столбцов с ic по jc. Чтобы сохранить порядок строк и/или столбцов, удостоверьтесь, что $ir \geq jr$ и $ic \geq jc$, иначе порядок будет обращен
182	Е *supsmooth(vx,vy)*	Возвращает n-мерный вектор, созданный локальным использованием симметричной линейной процедуры сглаживания методом наименьших квадратов по правилу k-ближайших соседей, в которой k выбирается адаптивно. vy и vx есть n-мерные векторы вещественных чисел. Элементы vx должны быть расположены в порядке возрастания
183	*svd(A)*	Матрица, составленная из расположенных друг над другом матриц U и V, где U (расположена сверху) есть m х n матрица и V есть $n \times n$ матрица. Матрицы U и V удовлетворяют равенству $A = U \cdot diag(s) \cdot V^T$, где s есть вектор, который содержит n элементов, возвращаемых функцией $svds(A)$. A есть m х n вещественный массив, где $m \geq n$
184	*svds(A)*	Вектор, содержащий сингулярные значения $m \times n$ вещественного массива A, где $m \geq n$
185	*tan(z)*	Тангенс
186	*tanh(z)*	Гиперболический тангенс
187	*tr(M)*	След квадратной матрицы M
188	*until(x,y)*	Возвращает y до тех пор, пока x не станет

		отрицательным
189	*var(A)*	Возвращает дисперсию элементов массива A размерности $m \times n$
190	*wave(v)*	Дискретное волновое преобразование вещественных данных, использующее четырех коэффициентный волновой фильтр Даубечи. Вектор v должен содержать 2^n вещественных значений, где n есть целое число
191	*WRITE(file)*	Записывает одиночное значение в файл данных *file*
192	*WRITEPRN(file)*	Записывает одиночное значение в файл данных *file*
193	*Y0(x)*	Функция Бесселя Y_1. Аргумент должен быть вещественным
194	*Y1(x)*	Функция Бесселя Y_2. Аргумент должен быть вещественным
195	*Yn(m,x)*	Функция Бесселя Y_m. x должен быть вещественным; $0 \leq m \leq 100$
196	*d(x,y)*	Символ Кронекера. Возвращает 1, если $m = n$; иначе 0. Оба аргумента должны быть целочисленными
197	*e(i,j,k)*	Полностью антисимметричный тензор ранга 3. i, j и k должны быть целыми числами между 0 и 2 включительно (или между ORIGIN и ORIGIN + 2 включительно, если ORIGIN≠0). Результат 0, если любые два аргумента одинаковы, 1 для четных перестановок, -1 для нечетных перестановок
198	*G(z)*	Возвращает значение эйлеровой *гамма*-функции в z
199	*F(x)*	Ступенчатая функция Хэвисайда. Возвращает 1, если $x \geq 0$; иначе 0

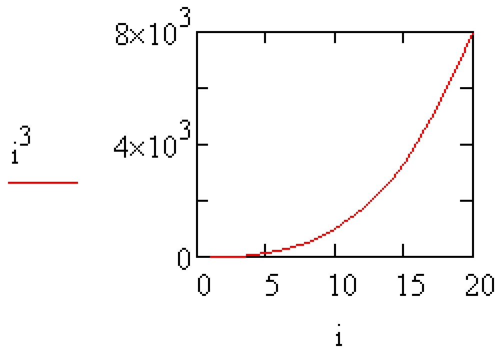
Приложение 3

Галерея графиков

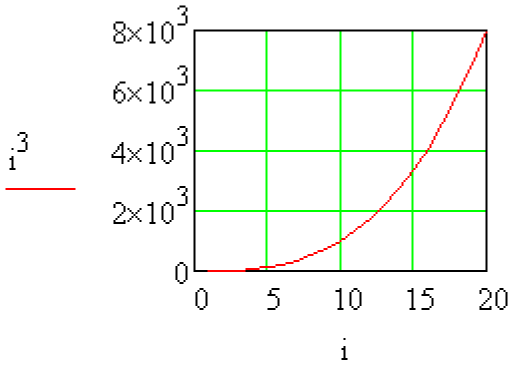
Декартов график

$i := 1 .. 20$

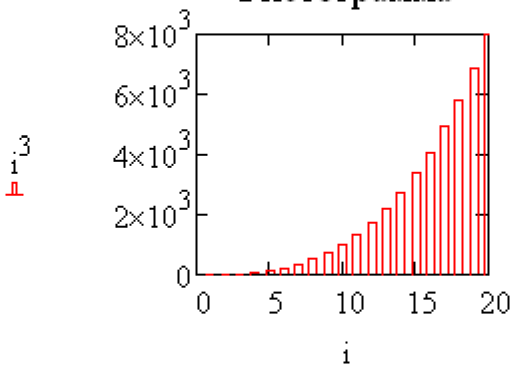
Формат по умолчанию



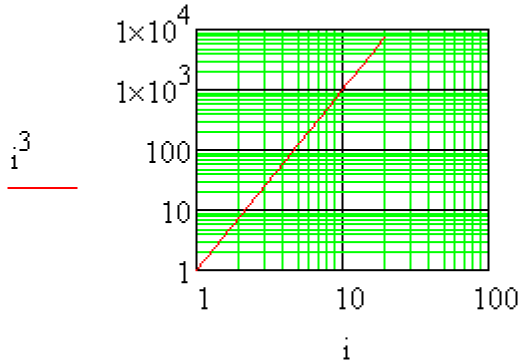
Линии сетки



Гистограмма



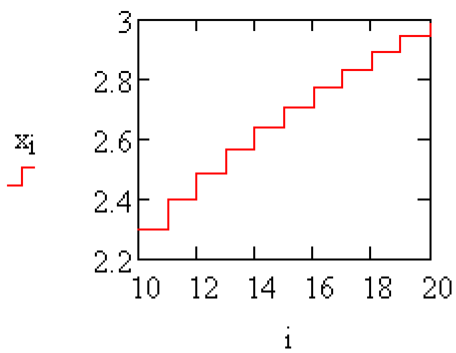
Логарифмический масштаб
и линии сетки



$i := 10 .. 20$

$x_i := \ln(i)$

$\varepsilon_i := \mathrm{rnd}(1) - 5$

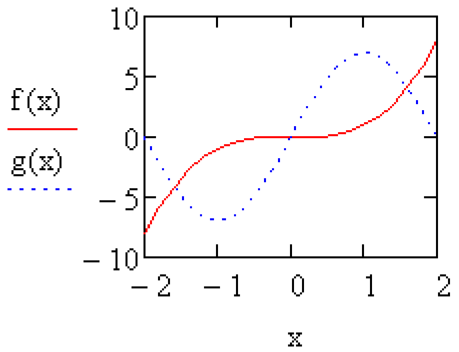


Ступенчатый график

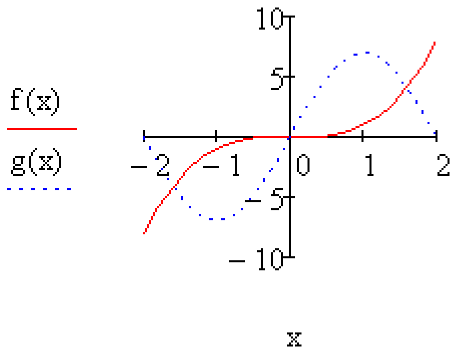
$$x := -2, -1.8 .. 2$$

$$f(x) := x^3$$

$$g(x) := 7 \cdot \sin\left(x \cdot \frac{\pi}{2}\right)$$



Рамка



Репер

Полярные графики

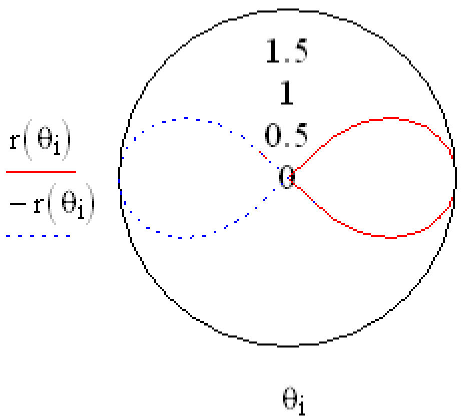
$$n := 2$$

$$r(\theta) := 2 \cdot \sqrt{\cos(n \cdot \theta)}$$

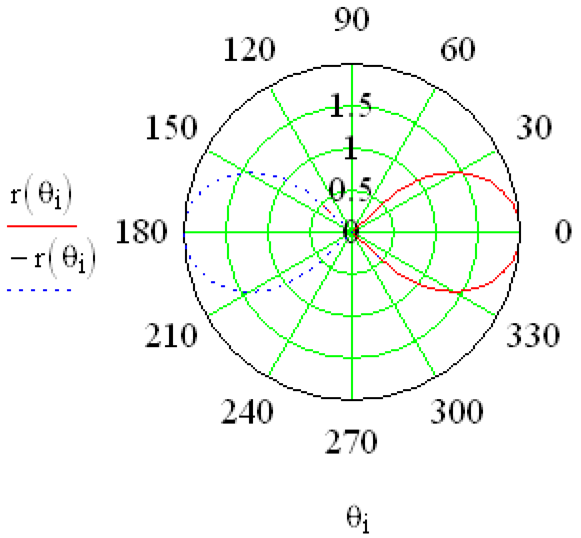
$$N := 200$$

$$i := 0 .. N$$

$$\theta_i := -\frac{\pi}{2 \cdot n} + i \cdot \frac{\pi}{N - n}$$



На графике отсутствует сетка, есть разметка радиальной оси, и нет разметки угловой оси



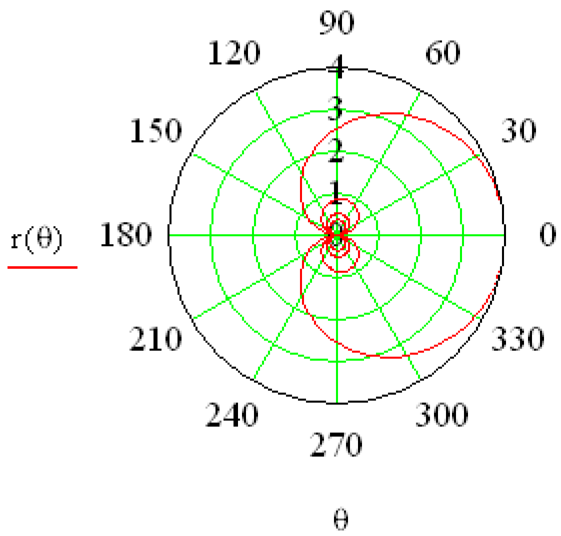
На графике нанесена сетка, есть разметка угловой оси, и нет разметки радиальной оси

$a := 4$

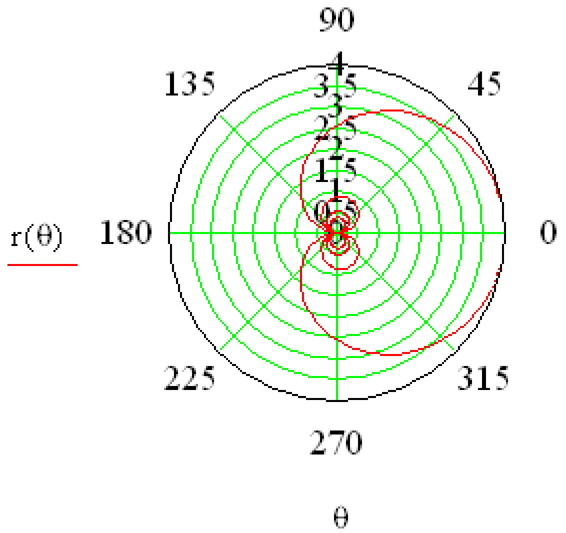
$$r(\theta) := a \cdot \frac{\sin(\theta)}{\theta}$$

$n := 4$

$$\theta := -n \cdot \pi, -n \cdot \pi + \frac{\pi}{60} .. n \cdot \pi$$



Авто сетка - включена
Число интервалов (радиальная ось) = 4
Число интервалов (угловая ось) = 12



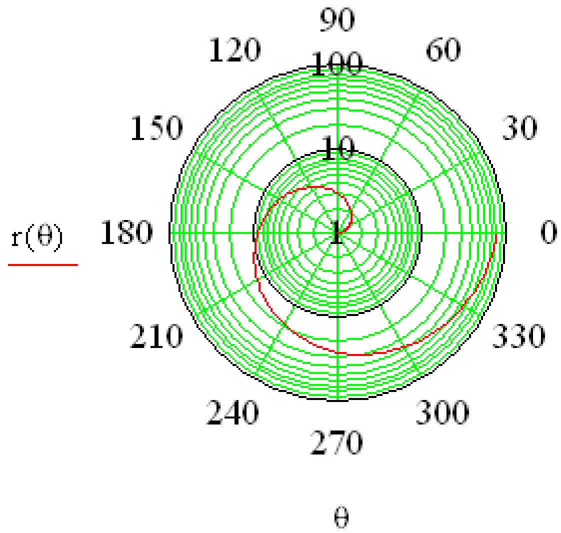
Авто сетка - отключена
Число интервалов (радиальная ось) = 8
Число интервалов (угловая ось) = 8

$a := 2$

$r(\theta) := a^{\theta}$

$n\_spirals := 1$

$\theta := 0, \frac{\pi}{60} .. 2 \cdot n\_spirals \cdot \pi$



Логарифмический масштаб

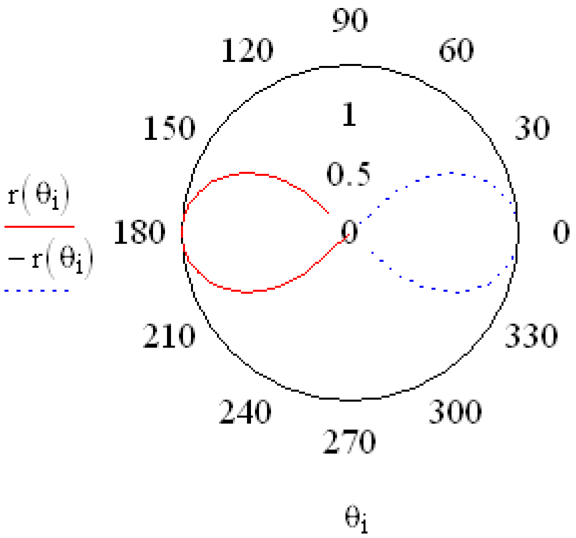
$n := 2$

$r(\theta) := \sqrt{\cos(n \cdot \theta) \cdot 2}$

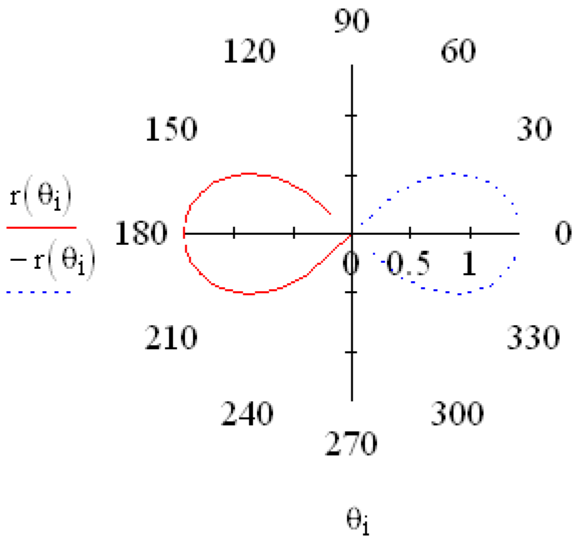
$N := 200$

$i := 0 .. N$

$\theta_i := \frac{\pi}{2 \cdot n} + i \cdot \frac{\pi}{N - n}$



Обрамление (стиль оформления)



Визир (стиль оформления)

Графики поверхностей

$N := 20$

$i := 0..N$

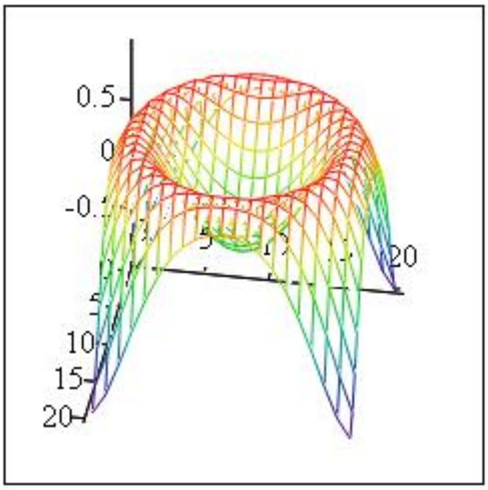
$j := 0..N$

$x_i := -1.5 + 0.15 \cdot i$

$y_j := -1.5 + 0.15 \cdot j$

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

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



M

Параметрическая поверхность

$m := 0..20$

$n := 0..20$

$r := 2$

$R := 6$

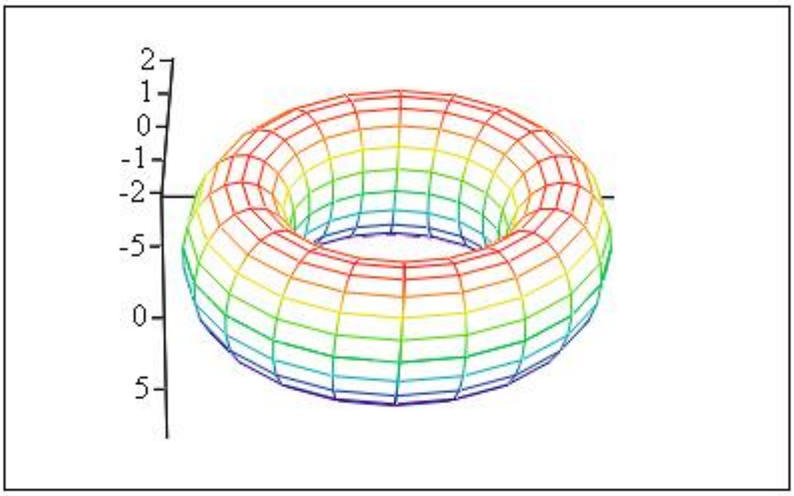
$$\phi_m := \frac{2\cdot\pi\cdot m}{20}$$

$$\theta_n := \frac{2\cdot\pi\cdot n}{20}$$

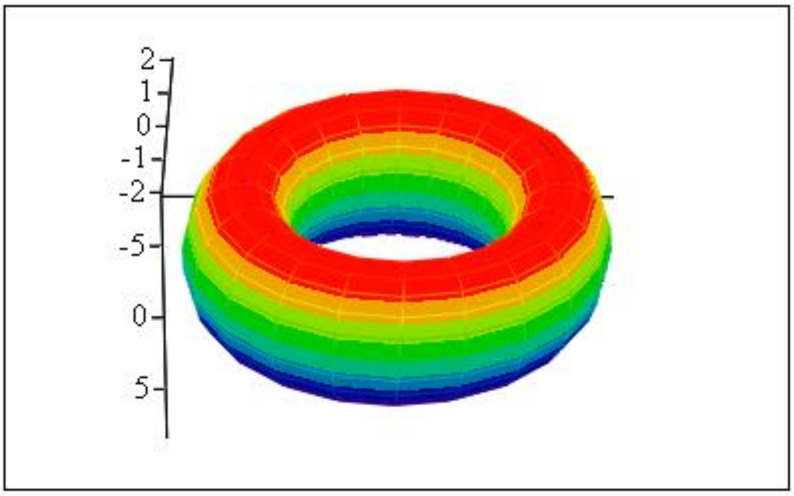
$$X_{m,n} := \left(R + r\cdot\cos\left(\theta_n\right)\right)\cdot\cos\left(\phi_m\right)$$

$$Y_{m,n} := \left(R + r\cdot\cos\left(\theta_n\right)\right)\cdot\sin\left(\phi_m\right)$$

$$Z_{m,n} := r\cdot\sin\left(\theta_n\right)$$



(X,Y,Z)



(X,Y,Z)

Карты линий уровня

$N := 15$

$i := 0 .. N$

$i := 0 .. N$

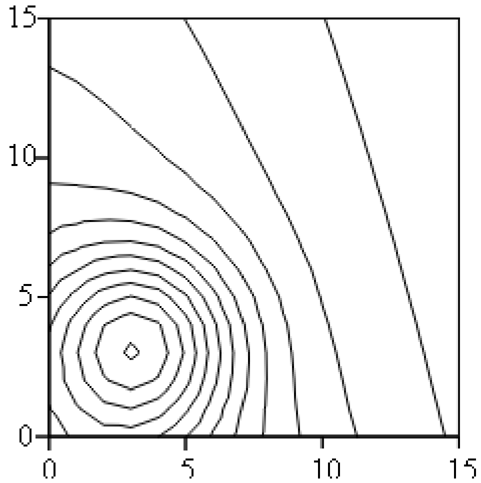
$x_i := \left(\frac{1}{N}\right) \cdot i$

$y_j := \left(\frac{1}{N}\right) \cdot j$

$$f(x,y) := \left[0.75 \cdot e^{-\left[\frac{(9 \cdot x-2)^2+(9 \cdot y-2)^2}{4}\right]}\right] + \left[0.75 \cdot e^{\left[\frac{(9 \cdot x+1)^2}{-49} - \frac{9 \cdot y+1}{10}\right]}\right]$$

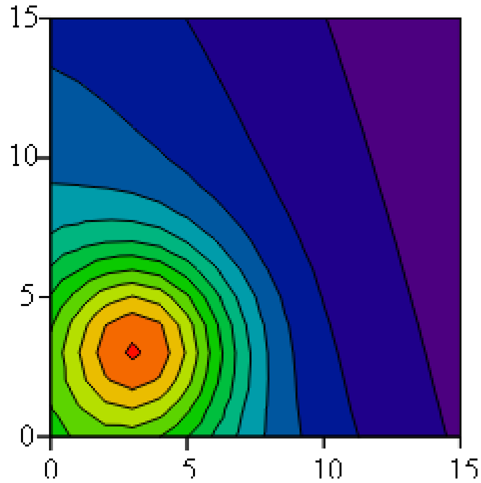
$M_{i,j} := 10 \cdot f(x_i, y_j)$

Окраска отсутствует



M

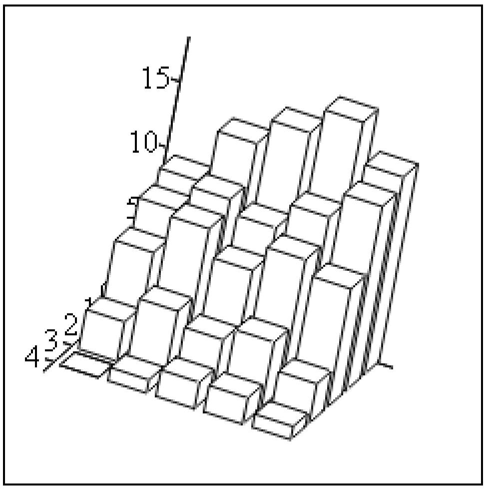
Цветное изображение



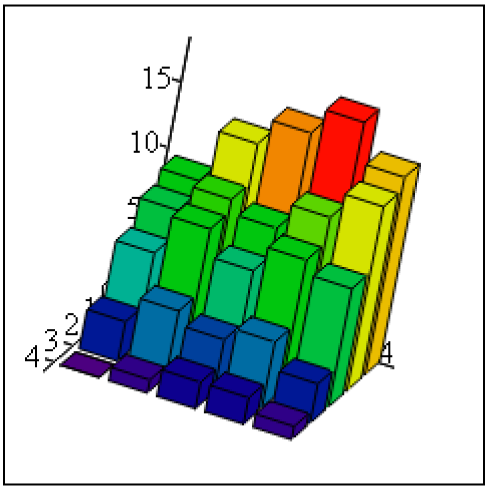
M

Трехмерные гистограммы

$$S := \begin{pmatrix} 10 & 14 & 16 & 18 & 15 \\ 9 & 11 & 10 & 12 & 14 \\ 7 & 10 & 8 & 10 & 9 \\ 3 & 5 & 4 & 5 & 3 \\ 0 & 1 & 2 & 2 & 1 \end{pmatrix}$$



S



S

Точечные графики

$N := 20$

$i := 0..N$

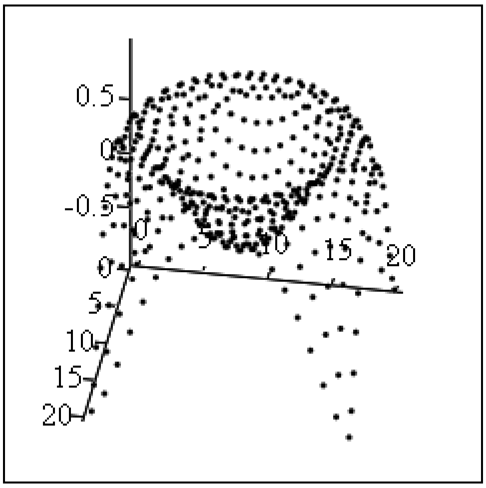
$j := 0..N$

$x_i := -1.5 + 0.15 \cdot i$

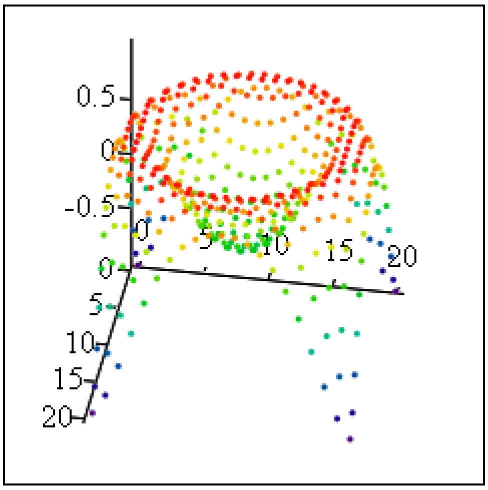
$y_j := -1.5 + 0.15 \cdot j$

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

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



M



M

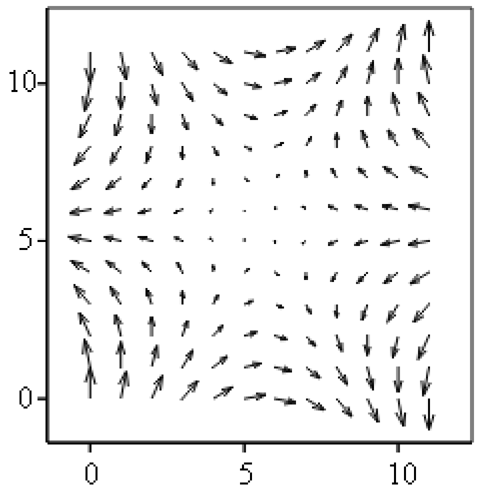
Графики векторных полей

$m := 0..11$

$n := 0..11$

$z_{m,n} := 0.2i\cdot(m-5) + 0.2\cdot(n-5) - 0.1 - 0.1i$

$$f(z) := \frac{z^2}{0.1 + |z|}$$



$\overrightarrow{\mathbf{f}(z)}$