

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Методические указания
по выполнению лабораторных работ по дисциплине

«Информационные системы и технологии в научных исследованиях»

для студентов направления подготовки
09.04.02 Информационные системы и технологии
направленность (профиль)
«Искусственный интеллект, математическое моделирование и суперкомпьютерные
технологии в разработке информационных систем»

СОДЕРЖАНИЕ

Введение

Часть 1. Основные сведения о системе MATLAB. Обзор расширений MATLAB

1. Пакеты математических вычислений
2. Пакеты для обработки сигналов и изображений
3. Пакеты анализа и синтеза систем управления
4. Описание пакета программ MATLAB

Лабораторные работы

Лабораторная работа 1. Режим прямых вычислений. Программирование, работа с файлами

Лабораторная работа 2. Операции с рабочей областью. Переменные и функции системы MATLAB

Лабораторная работа 3. Формирование векторов и матриц

Лабораторная работа 4. Создание и отладка программных файлов. Работа с файлами, импорт и экспорт данных

Лабораторная работа 5. Средства создания баз данных. Графика, анимация

Лабораторная работа 6. Графический интерфейс пользователя (GUI)

Часть 2. Решение задач в пакете MATHCAD

Общие сведения о системе MATHCAD

Лабораторные работы

Лабораторная работа 7. Основы работы в MathCAD

Лабораторная работа 8. Построение графиков

Лабораторная работа 9. Векторы и матрицы

Лабораторная работа 10. Решение уравнений

Лабораторная работа 11. Символьные вычисления

Часть 3. Свободная программная среда R

Общие сведения о свободной программной среде R

Лабораторные работы

Лабораторная работа 12. Основы работы с R. Обработка статистических данных

Лабораторная работа 13. Линейная регрессия

Лабораторная работа 14. Временные ряды

Лабораторная работа 15-16. Генетические алгоритмы

Приложение. Список основных функций MATLAB

Литература

Введение

Данные учебно-методические указания по выполнению лабораторных работ разработаны на основе программы дисциплины «Информационные системы и технологии в научных исследованиях», предназначенной для студентов, обучающихся в магистратуре по направлению 09.04.02 «Информационные системы и технологии» очной и заочной формы обучения.

Целью учебной дисциплины «Информационные системы и технологии в научных исследованиях» является формирование и развитие у студентов профессиональных компетенций, предназначенных для автоматизации научных экспериментов, а также для осуществления моделирования исследуемых объектов, явлений и процессов, изучение которых традиционными средствами затруднено или невозможно.

Задачи освоения дисциплины: изучение и применение информационных систем и технологий в научных исследованиях.

В результате усвоения тем и разделов дисциплины «Информационные системы и технологии в научных исследованиях», работы на лабораторных работах у студентов должны сформироваться следующие компетенции:

Код, формулировка компетенции	Код, формулировка индикатора	Планируемые результаты обучения по дисциплине (модулю), характеризующие этапы формирования компетенций, индикаторов
ОПК-2: Способен разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач	ОПК-2.1: Способен разрабатывать оригинальные алгоритмы и программные средства для решения профессиональных задач	Грамотно анализирует современные алгоритмы, программные средства и современные интеллектуальные технологии для решения профессиональных задач Правильно использует современные интеллектуальные технологии при модификации существующих алгоритмов и программных средств, а также разработке оригинальных алгоритмов и программных средств для решения профессиональных задач Грамотно анализирует и правильно использует современные интеллектуальные технологии для разработки оригинальных программных средств при решении профессиональных задач
	ОПК-2.2: Способен использовать современные интеллектуальные технологии для решения профессиональных задач	
ОПК-3: Способен анализировать профессиональную информацию, выделять в ней главное, структурировать, оформлять и представлять в виде аналитических обзоров с обоснованными выводами и рекомендациями	ОПК-3.1: Выполняет обобщение, структурирование и критический анализ профессиональной информации	Правильно анализирует и использует методики поиска анализа необходимой профессиональной информации для решения поставленной задачи и критерии оценки достоверности найденной профессиональной информации Правильно использует правила оформления и представления профессиональной информации в виде аналитических обзоров, современное программное обеспечение для презентации результатов анализа профессиональной информации и современные способы презентации этих результатов в виде аналитических обзоров с обоснованными выводами и рекомендациями.
	ОПК-3.2: Способен оформлять и представлять профессиональную информацию в виде аналитических обзоров с обоснованными выводами и рекомендациями	
ОПК-4: Способен применять на практике новые научные принципы и методы исследований.	ОПК-4.1: Способен осуществлять выбор методов исследования задач в ИТ-области	Грамотно анализирует базовые понятия методологии, системы методов и методики научного исследования Правильно может осуществлять выбор методов исследования задач в ИТ-области. Грамотно анализирует методологические принципы, структуру, функции научного знания, особенности организации и проведения научного исследования, современные информационные технологии для проведения научного исследования задач в ИТ-области. Правильно выстраивает логику научного исследования, определяет методологический аппарат исследования, использует необходимый инструментарий
	ОПК-4.2: Способен применять научные принципы и методы исследований задачи в ИТ-области	

ОПК-6: Способен использовать методы и средства системной инженерии в области получения, передачи, хранения, переработки и представления информации посредством информационных технологий	ОПК-6.1 Анализирует, выбирает методы и средства в области получения, передачи, хранения, переработки и представления информации посредством информационных технологий	Анализирует, выбирает методы и средства с учетом специфики системной инженерии в области получения, передачи, хранения, переработки и представления информации с помощью соответствующих инструментальных средств информационных технологий
	ОПК-6.2 Способен применять и методы и средства системной инженерии в профессиональной деятельности	Анализирует и правильно применяет в профессиональной деятельности методы и средства системной инженерии с учетом специфики инструментов и средств. Грамотно может развивать методы и средства системной инженерии в профессиональной деятельности и использовать их в ИТ области
ОПКД-2 Способен разрабатывать оригинальные алгоритмы и программные средства с использованием современных интеллектуальных компьютерные технологии, для решения задач в области создания и применения искусственного интеллекта	ОПКД-2.1: Применяет современные информационно-коммуникационные и интеллектуальные компьютерные технологии, инструментальные среды, программно-технические платформы для решения задач в области создания и применения искусственного интеллекта	Грамотно выбирает и применяет современные информационно-коммуникационные и интеллектуальные компьютерные технологии, инструментальные среды, программно-технические платформы для решения профессиональных задач с учетом особенностей, специфики и существующих проблем при решении задач в области создания и применения искусственного интеллекта. Анализирует и правильно выбирает и обосновывает современные информационно-коммуникационные и интеллектуальные компьютерные технологии для решения задач в области создания и применения искусственного интеллекта Разрабатывает оригинальные программные средства с помощью современных информационно коммуникационных и интеллектуальных компьютерных технологий для решения задач в области создания и применения искусственного интеллекта.
	ОПКД-2.2. Обосновывает выбор современных информационно-коммуникационных и интеллектуальных компьютерных технологий	
	ОПКД-2.3. Разрабатывает оригинальные программные средства, в том числе с использованием современных информационно коммуникационных и интеллектуальных компьютерных технологий, для решения задач в области создания и применения искусственного интеллекта	
ОПКД-6 Способен использовать методы научных исследований и математического моделирования в области проектирования и управления системами искусственного интеллекта, в том числе универсального искусственного интеллекта.	ОПКД-6.1: Применяет логические методы и приемы научного исследования, методологические принципы современной науки, направления, концепции, источники знания и приемы работы с ними, основные особенности научного метода познания, программно-целевые методы решения научных проблем в профессиональной деятельности	Анализирует и правильно применяет логические методы и приемы научного исследования; методологические принципы современной науки, концепции, источники знания и приемы работы с ними; основные методы научного познания, программно-целевые методы решения научных проблем, основы моделирования управленческих решений; динамические оптимизационные модели; математические модели оптимального управления для непрерывных и дискретных процессов и проводит их сравнительный анализ, многокритериальные методы принятия решений в профессиональной деятельности Осуществляет грамотное методологическое обоснование научного исследования. Грамотно использует полученные результаты при создании и применении библиотек искусственного интеллекта
	ОПКД-6.2. Осуществляет методологическое обоснование научного исследования, создание и применение библиотек искусственного интеллекта	

Часть 1. Основные сведения о системе MATLAB. Обзор расширений MATLAB.

Система MATLAB поставляется совместно с главным расширением Simulink. Это самое большое приложение системы, которое входит в блок расширений Blockset. Simulink, представляет собой систему программирования в геометрической форме, т.е. в виде блок-диаграмм, пакет предназначен для визуально-ориентированной подготовки имитационных моделей и анализа широкого класса динамических систем. Поддерживаются линейные, нелинейные, непрерывные, дискретные и гибридные системы. Модели могут образовывать иерархии, упрощая тем самым анализ подсистем. Моделирование может быть интерактивным или выполняться из командной строки MATLABa. В сочетании с такими пакетами, как Real-Time Workshop и Stateflow, Simulink образует семейство программ для решения задач цифровой обработки сигналов, разработки коммуникационных систем, систем управления и энергетических систем. Большие возможности по увеличению мощности среды моделирования предоставляют специализированные приложения SP Blockset, Fixed-Point Blockset, Power System Blockset и др. О пакете Simulink и особенностях его использования подробно написано в [1].

Другие приложения (программные дополнения к ядру системы) входят в состав большого «инструментального ящика» Toolboxes. Состав и версии этих пакетов постоянно обновляются, краткие аннотации к последним версиям можно найти в интернете по адресу: www.softline.ru. Для установки приложения его следует отметить в общем списке при инсталляции системы (иногда требуется задать некоторую дополнительную информацию). Приложения можно условно разбить на классы:

1. Пакеты математических вычислений.

а) Symbolic Math Toolbox – пакет прикладных программ, представляющий возможность решения задач в символьном (аналитическом) виде. Пакет базируется на применении ядра символьной математики одной из самых мощных систем компьютерной алгебры Maple. Обеспечивает выполнение символьного дифференцирования и интегрирования, вычисление сумм и произведений, разложение в ряды Тейлора и Маклорена, операции с полиномами, вычисление корней полиномов и др. Имеет команды прямого доступа к ядру системы.

б) NAG Foundation Toolbox – одна из мощнейших библиотек математических функций, созданная совместно с группой The Numerical Algorithms Group, Ltd. Пакет содержит более 240 функций, обеспечивающих решение самых разнообразных задач, включая оптимизацию, решение ОДУ и уравнений в частных производных, интерполяцию, вычисление собственных значений и векторов, сингулярных чисел, аппроксимацию кривых и поверхностей полиномами, кубическими сплайнами и полиномами Чебышева, минимизацию и максимизацию функций, статистические расчеты, корреляционный и регрессионный анализ и пр.

в) Spline Toolbox – пакет прикладных программ для работы со сплайнами. Включает функции для создания, отображения, интерполяции, аппроксимации и обработки сплайнов в В-форме и в кусочно-полиномиальной форме.

г) Statistics Toolbox – пакет прикладных программ по статистике, расширяющий возможности системы MATLAB в области реализации статистических вычислений и статистической обработки данных. К возможностям пакета относятся описательная статистика, распределения вероятностей (20 различных распределений), оценка параметров и аппроксимация, проверка гипотез, множественная регрессия, интерактивная пошаговая регрессия, моделирование Монте-Карло, статистическое управление процессами, планирование эксперимента, статистические графики и пр. Предусмотрено множество интерактивных инструментов для динамической визуализации и анализа данных.

д) Optimization Toolbox – пакет программ, реализующих широко известные методы минимизации и максимизации линейных и нелинейных функций: безусловную оптимизацию нелинейных функций, метод наименьших квадратов и нелинейную интерполяцию, решение нелинейных уравнений, линейное программирование, квадратичное программирование, условную минимизацию нелинейных функций, метод минимакса, многокритериальную оптимизацию и пр. В пакет включены версии традиционных и новейших алгоритмов оптимизации.

е) Partial Differential Equations Toolbox – пакет программ, содержащий множество функций для решения систем дифференциальных уравнений в частных производных. Используется метод конечных элементов. Команды и графический интерфейс пакета могут быть использованы для решения широкого класса инженерных задач, включая задачи сопротивления материалов, задачи тепломассопереноса, расчеты электромагнитных устройств.

ж) Fuzzy Logic Toolbox – пакет программ, относящихся к теории нечетких (размытых) множеств. Пакетом обеспечивается поддержка современных методов нечеткой кластеризации и адаптивных нечетких нейронных сетей. Графические средства пакета позволяют интерактивно отслеживать особенности поведения системы.

з) Neural Networks Toolbox – пакет программ, обеспечивающих всестороннюю поддержку проектирования, обучения и моделирования множества известных сетевых парадигм. Пакет может быть использован для исследования и применения нейронных сетей к таким задачам, как обработка сигналов, нелинейное управление и финансовое моделирование. В пакет включено более 15 типов сетей и обучающих правил.

и) Financial Toolbox – пакет прикладных программ по финансово-экономическим расчетам. Пакет содержит множество функций по расчету сложных процентов, операций по банковским вкладам, вычислению прибыли и др. К основным возможностям пакета относятся обработка данных, дисперсионный анализ эффективности портфеля инвестиций, анализ временных рядов, расчет доходности ценных бумаг и оценка курсов, статистический анализ, анализ чувствительности рынка, калькуляция ежегодного дохода и расчет денежных потоков, начисление износа и амортизационных отчислений. В пакет включены функции для манипулирования датами и временем в различных форматах, а также средства для презентации данных и результатов в виде графиков и диаграмм. До установки пакета на компьютер следует установить Excel, к которому должен быть указан путь при инсталляции системного приложения Excel Link.

2. Пакеты для обработки сигналов и изображений.

а) Signal Processing Toolbox – пакет, обеспечивающий широкие возможности создания программ обработки сигналов. Используется разнообразная техника фильтрации и алгоритмы спектрального анализа. Пакет можно использовать в таких областях, как обработка аудио и видеоинформации, телекоммуникации, геофизика, задачи управления в реальном режиме времени, экономика, финансы и медицина.

б) Image Processing Toolbox – пакет программ для цифровой обработки и анализа изображений. К возможностям пакета относятся восстановление и выделение деталей изображения, фильтрация сигналов изображения, работа с выделенным участком, анализ изображения, увеличение контрастности деталей, цветовые и геометрические преобразования, преобразование типов изображений. Пакет обеспечивает гибкий интерфейс для визуализации набора данных и обработки графических объектов (в том числе и трехмерных).

в) Wavelet Toolbox – пакет программ для исследования многомерных нестационарных явлений с помощью вейвлетов. Пакет позволяет анализировать такие особенности, которые упускают другие методы анализа сигналов: тренды, выбросы, разрывы в производных высоких порядков. Он позволяет сжимать и фильтровать сигналы без явных потерь. Программы сжатия выделяют минимальное число коэффициентов, представляющих исходную информацию наиболее точно. Новейшие методы пакета расширяют возможности в тех областях, где применяется техника Фурье-разложения.

г) Mapping Toolbox – пакет программ для работы с научными географическими данными. Пакет предоставляет графический и командный интерфейс для анализа данных, отображения карт и доступа к внешним источникам данных. Пакет включает более 60 проекций карт (прямые и инверсные), позволяет проектировать и отображать векторные, матричные и составные карты, трехмерные карты со встроенными средствами подсветки и затенения, глобальные и региональные атласы данных. Содержатся функции графической статистики и навигации, полезные при решении задач перемещения, таких как позиционирование и планирование маршрутов. В состав пакета входят широко известные атласы мира, США, астрономические атласы. Географическая структура данных упрощает извлечение данных из карт и их обработку. Используются конверторы для популярных форматов географических данных: DCW, TIGER, ETOPO5.

д) Virtual Reality Toolbox – пакет виртуальной реальности (доступен начиная с версии MATLAB 6.1). Позволяет осуществлять трехмерную анимацию и мультипликацию, в том числе моделей Simulink.

е) Curve Fitting Toolbox – новый математический пакет для подгонки (аппроксимации) кривых в графическом окне. Для одного и того же набора данных возможно проведение нескольких видов подгонки (полиномами от первой до девятой степени). Пакет полезен при решении типовых задач приближения данных.

ж) Dials & Gauges Blockset – новый пакет для разработки сложных моделей устройств и систем с повышенной степенью визуализации, таких, как приборные доски автомобилей и самолетов, диспетчерские пульта электростанций и аэропортов, пульта управления оборудованием и пр. Отличительная черта пакета – повышенная реалистичность отображения блоков (например, вольтметр изображается не простым прямоугольником, а прибором с четкой шкалой и движущейся стрелкой).

з) Higher-Order Spectral Analysis – пакет, содержащий специальные алгоритмы самых передовых методов для анализа сигналов с использованием моментов высшего порядка. Пакет позволяет анализировать сигналы, поврежденные негауссовым шумом, и процессы, происходящие в нелинейных системах.

и) Frequency Domain System Identification – пакет, предоставляющий специализированные средства для идентификации линейных динамических систем по их временному или частотному отклику. Частотные методы направлены на идентификацию непрерывных систем, что является мощным дополнением к традиционной дискретной методике. Методы пакета применимы к таким задачам, как моделирование электрических, механических и акустических систем. Функции для тестирования модели включают вычисление невязок, передаточных функций и нулей/полюсов, прогонку модели с использованием тестовых данных.

3. Пакеты анализа и синтеза систем управления.

а) Control System Toolbox – пакет программ для моделирования, анализа и проектирования систем автоматического управления, как непрерывных, так и дискретных. Пакет реализует традиционные методы передаточных функций и современные методы пространства состояний. Содержит полный набор средств для анализа МИМО-систем, временные характеристики, частотные характеристики (диаграммы Боде, Николса, Найквиста и др.), характеристики моделей (управляемость, наблюдаемость, понижение порядка), реализует проектирование LQR/LQE-регуляторов, разработку обратных связей, поддержку систем с запаздыванием. Пакет обладает настраиваемым окружением и позволяет создавать собственные m-файлы.

б) Robust Control Toolbox – пакет программ для проектирования и анализа многопараметрических устойчивых систем управления. К функциям пакета относятся синтез LQG-регуляторов на основе минимизации равномерной и интегральной нормы, многопараметрический частотный отклик, построение модели пространства состояний, преобразование моделей на основе сингулярных чисел, понижение порядка модели, спектральная факторизация.

в) Model Predictive Control Toolbox – пакет программ для реализации стратегии предиктивного (упреждающего) управления. Пакет обеспечивает моделирование, идентификацию и диагностику систем, поддержку MISO, MIMO, переходных характеристик, моделей пространства состояний, системный анализ, конвертирование моделей в различные формы представления (пространство состояний, передаточные функции), предоставление учебников и демонстрационных примеров. Пакет включает две функции Simulink, позволяющие тестировать нелинейные модели.

г) Communications Toolbox – пакет прикладных программ для построения и реализации телекоммуникационных устройств. Пакет содержит более 100 функций MATLAB-типа и 150 блоков SIMULINK-типа, обеспечивающих кодирование и оцифровку, контроль ошибок при кодировании, модуляцию и демодуляцию, фильтрацию при передаче и приеме, синхронизацию, коллективный доступ, вычисление в полях Галуа, генерацию сигналов, анализ и построение графиков.

д) μ -Analysis and Synthesis – пакет программ для проектирования устойчивых систем управления. Пакет использует оптимизацию в равномерной норме и сингулярный параметр μ . Полноценный графический интерфейс упрощает операции с блоками при проектировании оптимальных ресурсов, предоставляя возможность устанавливать диапазон вводимых данных и

редактировать D-K-итерации для приближенного μ -синтеза. Содержатся средства понижения порядка модели. Пакет поддерживает работу с непрерывными и дискретными моделями.

е) Quantitative Feedback Theory Toolbox – пакет функций и программ для создания устойчивых систем с обратной связью. Пакет базируется на методах количественной теории обратной связи (QFT), использующей частотное представление моделей для удовлетворения различных требований к качеству при наличии неопределенных характеристик объекта. Графический интерфейс пользователя оптимизирует процесс нахождения параметров обратной связи. Пакет позволяет вычислять различные параметры обратных связей и фильтров, проводить тестирование регуляторов как в непрерывном, так и в дискретном пространстве.

ж) LMI Control Toolbox – пакет, предоставляющий интегрированную среду для постановки и решения задач линейного программирования, таких, как задачи совместности ограничений, минимизация линейных целей при наличии линейных ограничений, минимизация собственных значений. К возможностям пакета относятся задание ограничений в символьном виде, графическое редактирование задач линейного программирования, многокритериальное проектирование регуляторов, проверка устойчивости (квадратичная устойчивость линейных систем, устойчивость по Ляпунову, проверка критерия Попова для нелинейных систем). Пакет является мощным средством для решения выпуклых задач оптимизации в различных областях (управление, идентификация, фильтрация, структурное проектирование, теория графов и пр.).

з) System Identification Toolbox – пакет, содержащий средства для создания математических моделей динамических систем на основе наблюдаемых входных и выходных данных. Методы идентификации, применяемые в пакете, пригодны для решения широкого класса задач, от проектирования систем управления и обработки сигналов до анализа временных рядов и вибрации. Пакет предусматривает предварительную обработку данных (фильтрацию, удаление трендов и смещений), выбор диапазона данных для анализа, анализ отклика во временной и частотной области, отображение нулей и полюсов передаточной функции системы, анализ невязок при тестировании модели, построение сложных диаграмм (диаграмма Найквиста и др.). Пакет поддерживает все традиционные структуры моделей (включая авторегрессию и структуру Бокса-Дженкинса), линейные модели пространства состояний, которые могут быть определены как в дискретном, так и в непрерывном пространствах. Число входов и выходов произвольно.

4. Описание пакета программ MATLAB

4.1. Рабочая область MATLAB

Пакет программ MATLAB предназначен для решения большого круга задач, в том числе для моделирования процессов в объектах управления, а также процессов в системах управления объектами, которые можно описать дифференциальными уравнениями. Работа пакета базируется на интегрировании дифференциальных уравнений, вводимых в пакет с помощью графических блоков структурных схем с соответствующими значениями параметров. Работа пакета программ MATLAB описана в [1]. Данный пакет предоставляет инженеру-разработчику или исследователю богатую библиотеку различного рода блоков структурных схем и вспомогательных блоков для наиболее полного анализа динамических систем.

После запуска MATLAB на экране появится окно, представленное на рис. 1.

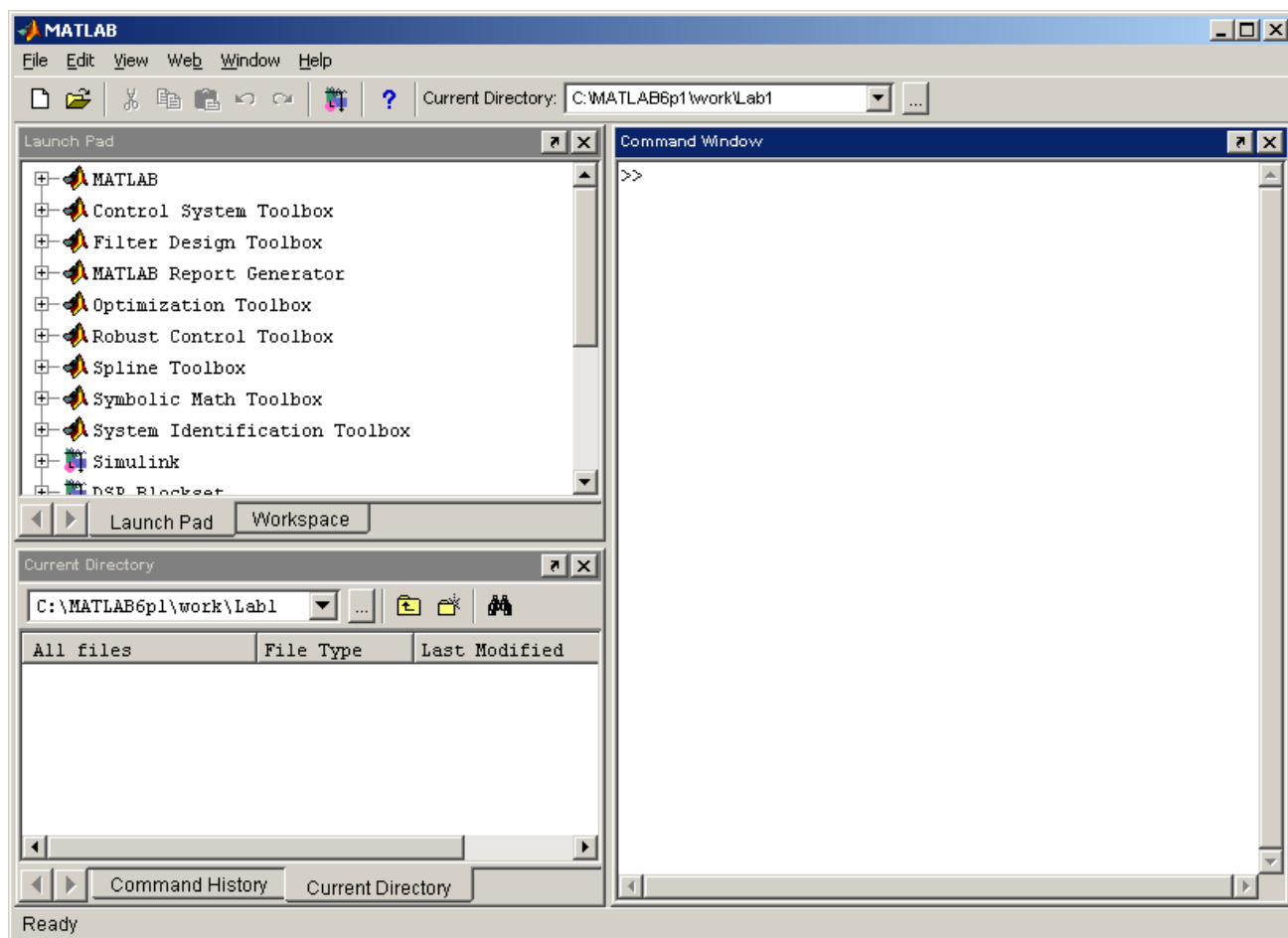


Рисунок 1 – Окно запуска MATLAB

Для создания новой модели необходимо в данном окне щелкнуть левой кнопкой мыши на иконке (Simulink) и в появившемся окне, представленном на рис. 2, выбрать в меню File\New\Model (для быстрого создания можно использовать сочетание клавиш Ctrl+N). В результате вместе с окном Simulink Library Browser на экране отобразится окно, которое является рабочей областью создания модели (рис. 3).

Выбор составных блоков модели осуществляется с помощью браузера библиотеки Simulink (Simulink Library Browser) (рис. 2), состоящего из девяти разделов: Continuous, Discrete, Functions & Tables, Math, Nonlinear, Signals & Systems, Sinks, Sources, Subsystems.

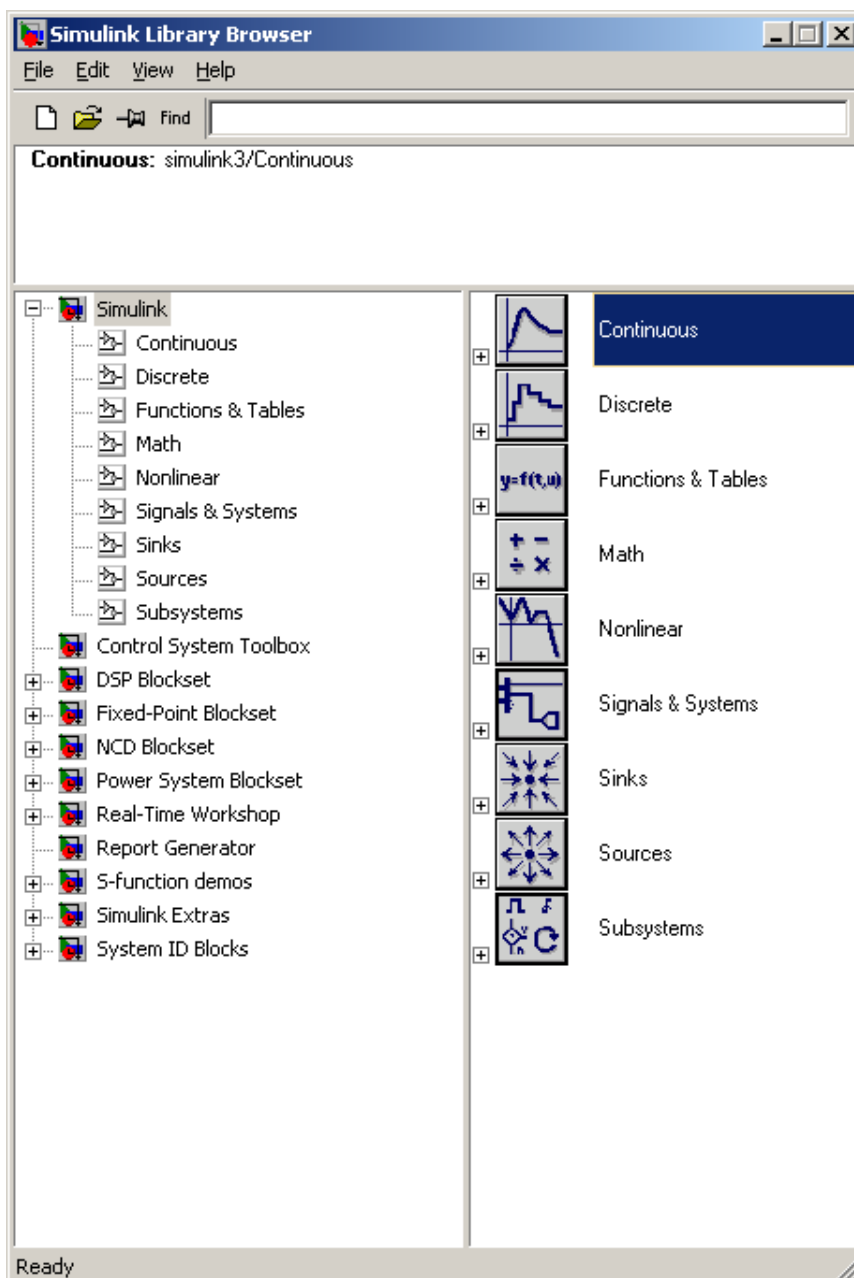


Рисунок 2 – Окно браузера библиотеки Simulink

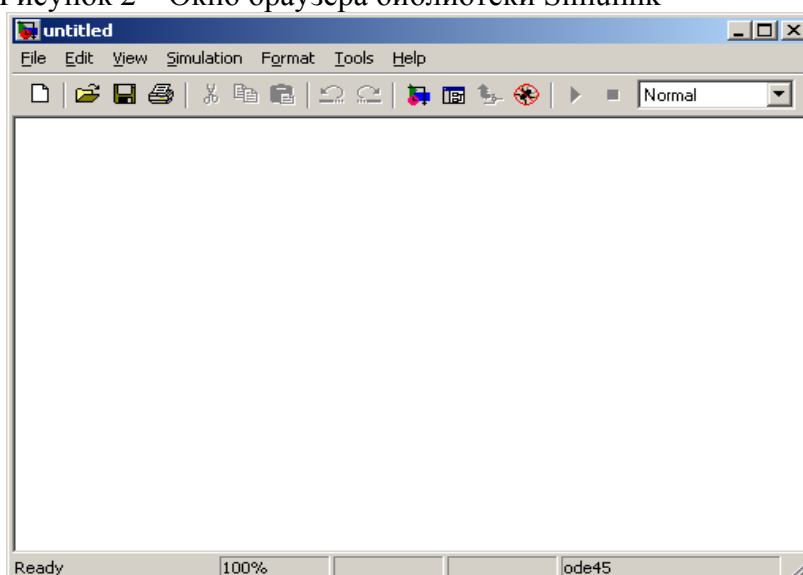


Рисунок 3 – Окно рабочей области MATLAB

4.2. Раздел CONTINUOUS

Раздел Continuous (непрерывные системы) содержит блоки, которые можно условно разделить на две группы: блоки, непосредственно предназначенные для описания непрерывных систем, и блоки общего назначения, которые могут быть использованы в модели любой системы.

- 1). Derivative (производная) – блок вычисления производной входного сигнала по времени;
- 2). Integrator (интегратор) – сумматор непрерывного действия;
- 3). Memory (регистр) – блок задержки входного сигнала на один шаг моделирования;
- 4). State-Space (пространство состояний) – блок описания системы в векторно-матричной форме;
- 5). Transfer Fcn (передаточная функция) – блок задания передаточных функций;
- 6). Transport Delay (транспортное запаздывание) – блок, реализующий произвольную задержку входного сигнала;
- 7). Variable Transport Delay (переменное транспортное запаздывание) – блок, реализующий динамически изменяемую задержку входного сигнала;
- 8). Zero-Pole (область перехода) – блок описания непрерывной системы в виде нулей и полюсов.

4.3. Раздел DISCRETE

В раздел Discrete (дискретные системы) входят блоки, с помощью которых в модели может быть описано поведение дискретных систем.

- 1). Discrete Transfer Fcn (дискретная передаточная функция) – блок задания дискретной передаточной функции;
- 2). Discrete Zero-Pole (дискретная область перехода) – блок описания дискретной системы в виде нулей и полюсов;
- 3). Discrete Filter (дискретный фильтр) – блок создания дискретного фильтра, порядок и свойства которого задаются полиномом от частного $1/z$ в числителе передаточной функции;
- 4). Discrete State-Space (дискретное пространство состояний) – блок формирования дискретного пространства состояний;
- 5). Discrete-Time Integrator (дискретный интегратор времени) – блок дискретного интегрирования времени. Обычно он служит для управления логической работой модели, например для остановки процесса моделирования по заданному значению интеграла времени;
- 6). First-Order Hold (экстраполятор первого порядка) – блок задержки выходного сигнала на заданный промежуток времени и задания линейного изменения выходного сигнала на каждом такте дискретизации, в соответствии с крутизной входного сигнала в данный момент времени;
- 7). Unit Delay (дискретная единичная задержка) – блок, обеспечивающий задержку входного сигнала на один шаг эталонного времени;
- 8). Zero-Order Hold (экстраполятор первого порядка) – блок задержки выходного сигнала на заданный промежуток времени. Данный блок оставляет неизменными значения выходного сигнала на каждом такте дискретизации.

4.4. Раздел FUNCTIONS & TABLES

Раздел Functions & Tables (функции и таблицы) содержит компоненты функций и таблиц. Блоки функций позволяют вводить в модели практически любые функции, что имеет большое значение в математическом моделировании различных объектов и систем. Блоки таблиц обеспечивают задание табличных данных с различной размерностью, а также позволяют использовать линейную или сплайновую интерполяцию и экстраполяцию данных.

- 1). Direct Look-Up Table (n-D) (многомерная таблица с прямым доступом) – блок, позволяющий задавать многомерные таблицы с прямым доступом к их элементам;
- 2). Fcn (функция) – блок задания функций одной переменной или ряда переменных $u(i)$;
- 3). Interpolation (n-D) using PreLook-Up (многомерная интерполяционная таблица) – блок, позволяющий создать многомерную интерполяционную таблицу для представления в табличном виде функций ряда переменных;
- 4). Look-Up Table (одномерная таблица) – блок задания в табличной форме некоторых данных, представляющих ряд значений функции одной переменной;
- 5). Look-Up Table (2-D) (двумерная таблица) – блок задания таблиц, представляющих значения функций двух переменных;

- 6). Look-Up Table (n-D) (многомерная таблица) – блок табличного представления данных, имеющий расширенные возможности в части задания размерности таблиц и интерполяции и экстраполяции их данных;
- 7). MATLAB Fcn (блок задания функции MATLAB) – блок задания одной переменной или ряда переменных $u(i)$ по правилам, принятым для языка программирования базовой системы MATLAB. Это означает, что в теле функции могут встречаться как встроенные функции системы MATLAB, так и любые процедуры и функции, реализованные в виде m-файлов;
- 8). Polynomial (полином) – блок задания степенного многочлена $P(u)$ в виде вектора его коэффициентов;
- 9). PreLook-Up Index Search (работа с индексами) – блок вычисления принадлежности одномерных данных к ближайшим узлам при приближении к ним снизу, а также контроль разности (дистанции) в относительных единицах между данными и значениями этих узловых точек;
- 10). S-Function (S-функция) – блок задания S-функций.

4.5. Раздел MATH

Раздел Math (математические блоки) содержит блоки, которые реализуют элементарные (алгебраические и тригонометрические) функции, а также операции математической логики и могут быть использованы в модели любой системы.

- 1). Abs (абсолютное значение) – блок формирования абсолютного значения входного сигнала;
- 2). Algebraic Constraint (алгебраический корень) – блок, предназначенный для отыскания корней алгебраических уравнений;
- 3). Bitwise Logical Operator (побитовый логический оператор) – блок, содержащий набор основных логических операторов: AND, OR, NAND, NOR, XOR, NOT. Этот блок выполняет вышеперечисленные операции побитово;
- 4). Combinatorial Logic (комбинаторная логика) – блок, обеспечивающий преобразование входного сигнала в соответствии с правилами, задаваемыми так называемой таблицей истинности;
- 5). Complex to Magnitude-Angle (выделение из комплексного числа амплитуды и фазы) – блок, позволяющий получить из входной комплексной величины амплитудно-фазовые характеристики сигнала;
- 6). Complex to Real-Imag (выделение из комплексного числа действительной и мнимой части) – блок, служащий для выделения из комплексного числа действительной и мнимой части;
- 7). Dot Product (свертка) – блок, обеспечивающий скалярное умножение (свертку) двух векторов, подаваемых на его входы;
- 8). Gain (умножитель) – блок, выполняющий роль умножителя сигнала, поступающего на его вход;
- 9). Logical Operator (логический оператор) – блок, содержащий набор основных логических операторов: AND, OR, NAND, NOR, XOR, NOT;
- 10). Magnitude-Angle to Complex (преобразование амплитуды и фазы в комплексное число) – блок, позволяющий получить из амплитудно-фазовой характеристики сигнала комплексную величину;
- 11). Math Function (математическая функция) – блок, позволяющий использовать для преобразования входного сигнала элементарные нетригонометрические функции (вычисление экспоненты, натурального и десятичного логарифмов, возведение в степень, извлечение квадратного корня и т.д.);
- 12). Matrix Gain (матричный множитель) – блок, предназначенный для умножения входного сигнала на матрицу (или вектор);
- 13). MinMax (минимум или максимум) – блок, обеспечивающий поиск минимального или максимального элемента входного вектора;
- 14). Product (умножение) – блок, позволяющий выполнять умножение или деление нескольких входных сигналов (величин);
- 15). Real-Imag to Complex (преобразование действительной и мнимой части в комплексное число) – блок, позволяющий получить из действительной и мнимой части комплексное число;
- 16). Relational Operator (оператор отношения) – блок, реализующий операции отношения ($>$, $<$, $?$, $==$ (тождественно), $!=$ (не равно) над двумя входными сигналами;

- 17). Rounding Function (функция округления) – блок, содержащий различные функции округления значения амплитуды выходного сигнала;
- 18). Sign (знак) – блок, выполняющий проверку знака выходного сигнала;
- 19). Slider Gain (настраиваемый множитель) – блок, позволяющий динамически изменять коэффициент умножения в процессе моделирования;
- 20). Sum (сложение) – блок, выполняющий суммирование входных сигналов;
- 21). Trigonometric Function (тригонометрическая функция) – блок, обеспечивающий преобразование входного сигнала с помощью одной из тригонометрических функций.

4.6. Раздел NONLINEAR

Раздел Nonlinear (нелинейные системы) содержит блоки, которые используются для описания нелинейных систем.

- 1). Backlash (люфт) – блок, имитирующий эффект возникновения люфта, при котором создается передаточная характеристика гистерезисного типа;
- 2). Coulomb & Viscous Friction (блок фрикционных эффектов) – блок, служащий для моделирования фрикционных эффектов;
- 3). Dead Zone (зона нечувствительности) – блок, заменяющий значение входного сигнала, лежащее в заданном диапазоне, нулем;
- 4). Manual Switch (ручной переключатель) – блок, позволяющий вручную выбирать один из двух входных портов, сигнал с которого будет передаваться на выход блока;
- 5). Multipoint Switch (многоходовой переключатель) – блок, позволяющий задавать требуемое количество входов (портов) и подключать тот или иной вход к выходу в зависимости от уровня сигнала на управляющем входе;
- 6). Quantizer (квантователь) – блок, служащий для квантования меняющихся сигналов с одинаковым шагом по уровню;
- 7). Rate Limiter (ограничитель скорости) – блок, предназначенный для контроля и ограничения скорости изменения (то есть первой производной) входного сигнала;
- 8). Relay (реле) – релейный блок, имеющий разрывную передаточную функцию с гистерезисом (или без него);
- 9). Saturation (насыщение) – блок, представляющий собой нелинейное устройство, сигнал на выходе которого равен входному сигналу до тех пор, пока не достигает порогов ограничения: верхнего Upper limit или нижнего Lower limit;
- 10). Switch (ключ) – блок управляемого переключателя с тремя входами: двумя (крайними) для сигналов данных и одним (средним) для сигналов управления.

4.7. Раздел SIGNALS & SYSTEMS

Раздел Signals & Systems (сигналы и системы) – один из самых больших разделов, содержит блоки, которые могут быть использованы для описания широкого класса систем: непрерывных, дискретных и др.

- 1). Bus Selector (селектор шины) – блок, выделяющий из группы сигналов один заданный сигнал;
- 2). Data Store Memory (память для данных) – блок, обеспечивающий хранение данных на интервале моделирования;
- 3). Data Store Read (чтение данных) – блок, обеспечивающий чтение данных на интервале моделирования;
- 4). Data Store Write (запись данных) – блок, обеспечивающий запись данных на интервале моделирования. Блоки 2, 3, 4 используются совместно;
- 5). Data Type Conversion (приведение типа данных) – блок, обеспечивающий приведение типа сигнала к требуемому;
- 6). Demux (разделитель) – блок, разделяющий векторный входной сигнал на несколько сигналов;
- 7). Function-Call Generator (генератор вызова функций) – блок, запускающий подключенные к нему подсистемы с заданной периодичностью;
- 8). From (принять) – блок, предназначенный для обмена (принятия) данными между различными компонентами S-модели с учетом доступности (видимости) этих данных;

- 9). Goto (передать) – блок, предназначенный для обмена (передачи) данными между различными компонентами S-модели с учетом доступности (видимости) этих данных;
- 10). Goto Tag Visibility (видимость блока передачи) – блок, предназначенный для обмена данными между различными компонентами S-модели с учетом доступности (видимости) этих данных. Блоки 8, 9, 10 используются совместно;
- 11). Hit Crossing (индикатор пересечения) – блок, позволяющий идентифицировать момент времени, когда входной сигнал «пересекает» некоторое значение;
- 12). IC (начальное состояние) – блок, позволяющий устанавливать произвольное начальное состояние входного сигнала;
- 13). Matrix Concatenation (объединение сигналов в матрицу) – блок объединения сигналов в матрицу образует из ряда векторов матрицу, обеспечивая их объединение (конкатенацию) по горизонтали и по вертикали;
- 14). Merge (поглотитель) – блок, отбирающий из входных сигналов один, значение которого было вычислено последним;
- 15). Model Info (информация о модели) – блок, позволяющий контролировать информацию, относящуюся к разработке модели (версия, имя разработчика и т.п.);
- 16). Mux (смеситель) – блок, объединяющий входные сигналы в один векторный сигнал;
- 17). Probe (проверка) – блок, предназначенный для просмотра основных характеристик сигнала (длительность, тип), а также величины шага моделирования;
- 18). Reshape (восстановление) – блок, позволяющий восстановить входной сигнал;
- 19). Selector (селектор) – блок, выделяющий из входного вектора заданные элементы;
- 20). Signal Specification (спецификация сигналов) – блок, служащий для распознавания сигнала на его входе и для выдачи спецификации;
- 21). Width (длительность) – блок вычисления длительности сигнала, поступающего на его вход; вычисленное значение выводится непосредственно внутри блока.

4.8. Раздел SINKS

В разделе Sinks (получатели) размещены блоки-приемники сигналов.

- 1). Display (экран) – блок, предназначенный для отображения численных значений величин;
- 2). Scope (осциллограф) – блок, позволяющий в процессе моделирования наблюдать динамику интересующих исследователя характеристик системы;
- 3). Stop Simulation (остановка моделирования) – блок, позволяющий прервать моделирование при выполнении тех или иных условий;
- 4). Terminator (ограничитель) – блок, использующийся в качестве «заглушки» для выходных портов;
- 5). To File (запись в файл) – блок, обеспечивающий сохранение промежуточных и/или выходных результатов моделирования в файле;
- 6). To Workspace (запись в рабочую область) – блок, обеспечивающий сохранение промежуточных и/или выходных результатов моделирования в рабочей области MATLAB;
- 7). XY Graph (двумерный график) – блок, обеспечивающий создание двумерных графиков в прямоугольной системе координат.

4.9. Раздел SOURCES

Блоки, входящие в раздел Sources (источники), предназначены для формирования сигналов, которые обеспечивают управление работой модели в целом или отдельных ее частей.

- 1). Band-Limited White Noise («белый шум» с ограниченной полосой) – блок-генератор «белого шума» с ограниченной полосой;
- 2). Chirp Signal (гармонический сигнал) – блок-источник гармонических колебаний переменной частоты;
- 3). Clock (часы) – блок-генератор непрерывного временного сигнала;
- 4). Constant (константа) – блок, обеспечивающий формирование постоянного скалярного или векторного сигнала;
- 5). Digital Clock (цифровые часы) – блок-источник дискретного временного сигнала;
- 6). From Workspace (ввод из рабочей области) – блок, реализующий ввод в модель данных непосредственно из рабочей области MATLAB;

- 7). From File (ввод из файла) – блок, реализующий ввод в модель данных, хранящихся в MAT-файле;
- 8). Ground (земля) – блок, использующийся в качестве «заглушки» для входных портов;
- 9). Pulse Generator (импульсный генератор) – блок-генератор импульсных сигналов;
- 10). Ramp (возбудитель) – блок-генератор линейно возрастающего (убывающего) сигнала вида $F(t)=k \cdot t$;
- 11). Random Number (случайное число) – блок-источник случайного сигнала, служащий для создания случайного сигнала с равномерным распределением уровня;
- 12). Repeating Sequence (периодический сигнал) – блок-генератор периодического дискретного сигнала произвольной формы;
- 13). Signal Generator (генератор сигнала) – блок-генератор непрерывного сигнала произвольной формы;
- 14). Sine Wave (генератор гармонических колебаний) – блок-генератор гармонических колебаний;
- 15). Step («ступенька») – блок-источник воздействия в виде одиночного перепада;
- 16). Uniform Random Number (равномерное случайное число) – блок-источник дискретного сигнала, амплитуда которого является равномерно распределенной случайной величиной.

4.10. Раздел SUBSYSTEMS

Раздел Subsystems (подсистемы) содержит блоки, предназначенные для создания подсистем (субмоделей). Внутри подсистем первого уровня могут располагаться подсистемы второго уровня и т.д. Это напоминает ситуацию, когда сложная система набирается из отдельных систем – модулей, каждый из которых, в свою очередь, является системой или устройством.

Лабораторные работы

Лабораторная работа 1. Режим прямых вычислений

Цель лабораторной работы:

формирование у студентов навыков синтеза системы и анализа чувствительности параметров системы с помощью пакета MATLAB.

Организационная форма занятий: решение проблемных задач, разбор конкретных ситуаций

Вопросы для обсуждения на лабораторном занятии: построение простейших моделей в пакете Simulink системы MATLAB.

Формируемые компетенции: ОПК-2, ОПК-3, ОПК-4, ОПКД-2, ОПКД-3, ОПКД-4.

Основные сведения

Систему MATLAB можно использовать, прежде всего, как мощный калькулятор, способный производить, помимо обычных вычислений, операции с векторами и матрицами, комплексными числами, рядами и полиномами. Можно выводить графики различных функций – от простой синусоиды до сложной трехмерной фигуры.

Работа в режиме прямых вычислений (командном режиме) носит диалоговый характер. Пользователь набирает выражение, редактирует его и после нажатия клавиши ENTER получает ответ. Например, команда

```
>> v= [1 3 4];
```

задает трехэлементный вектор v (возможна также запись не через пробел, а через запятую $v=[1,3,4]$). Знак ; (точка с запятой) в конце выражения блокирует немедленный вывод результата вычислений. Далее можно подать команду

```
>> sin(v)
```

и получить ответ

```
ans= 0.8415 0.1411 -0.7568
```

Как видим, система вычисляет функцию **sin** от векторного аргумента, при этом встроенные функции (см. Приложение) записываются строчными буквами, их аргументы указываются в круглых скобках. Если пользователем не указана переменная для значения результата, то MATLAB назначает переменную с именем **ans**.

Выражения записываются в соответствии с правилами языка MATLAB (п.3). Если выражение оказывается длинным, то его часть можно перенести на новую строку с помощью знака ... (многоточие). Текстовые комментарии вводятся с помощью знака % перед текстом строки.

Полезно усвоить команды управления окном сессии:

cls – очистка экрана, **home** – возвращение курсора в левый верхний угол окна. Размеры и положение командного окна можно менять. Размер окна оптимален, если оно занимает нижнюю четверть экрана, а верхняя часть занята графическим окном, тогда вычисления выполняются одновременно с просмотром графиков.

С помощью клавиш $\uparrow$ и $\downarrow$ можно вернуться к выполнению любой из предыдущих команд.

Начало работы в Simulink

Для начала работ в Simulink необходимо зарегистрироваться на сайте mathworks.com как показано на рисунке 1.

Рисунок 1 – Окно регистрации на сайте mathworks.com

После подтверждения через адрес электронной почты аккаунта на данном сайте, можно авторизоваться и ввести информацию для учетной записи в MathWorks (рисунок 2).

Рисунок 2 – Учетная запись в MathWorks

Далее необходимо получить номер лицензии для бесплатного 30-дневного доступа (рисунок 3).

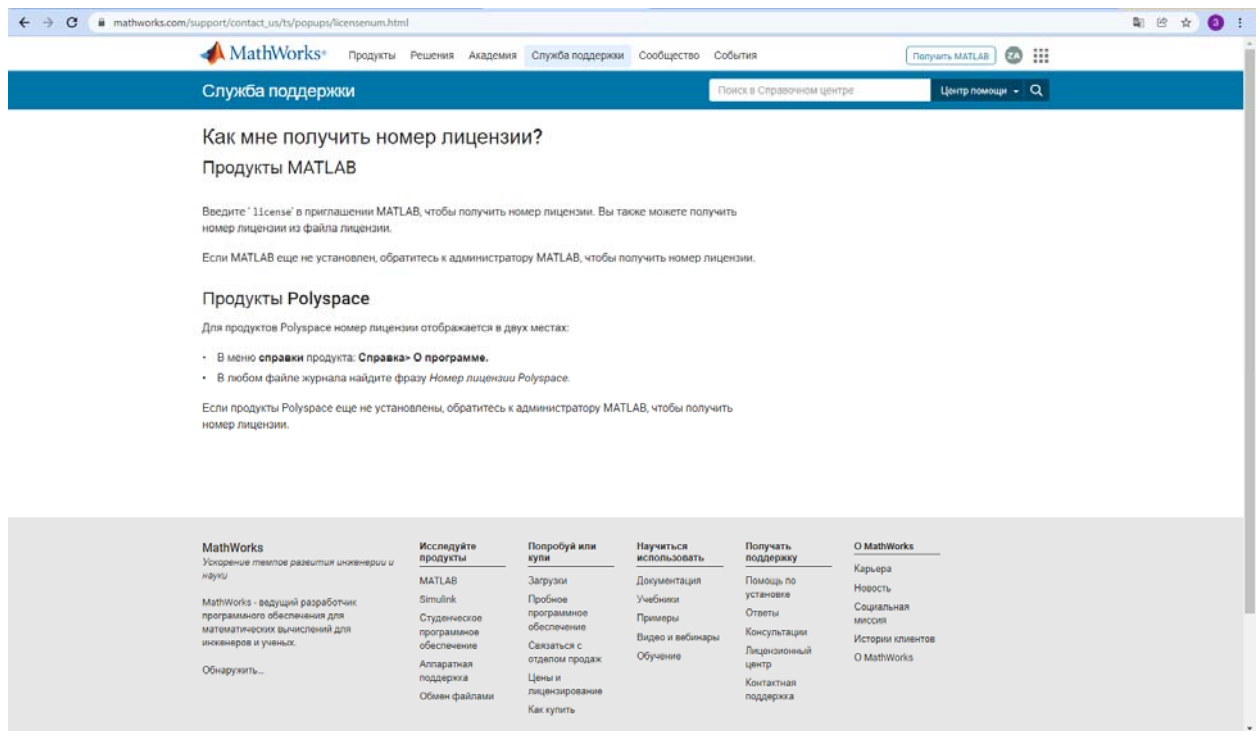


Рисунок 3 – Инструкция по получению номера лицензии

Для получения номера лицензии на сайте MathWorks необходимо будет заполнить данные пользователя: рабочий или университетский адрес электронной почты, университет, телефон, город, почтовый индекс и др. Далее сайт предложит скачать несколько версий MathLab для разных платформ. Выберите нужный вам вариант, скачайте установочный файл и установите на своем ПК.

Помимо локального использования MathLab на персональном компьютере, можно выполнять задания лабораторных работ и онлайн с теми же параметрами доступа на сайте по ссылке mathlab.mathworks.com, как показано на рисунке 4.

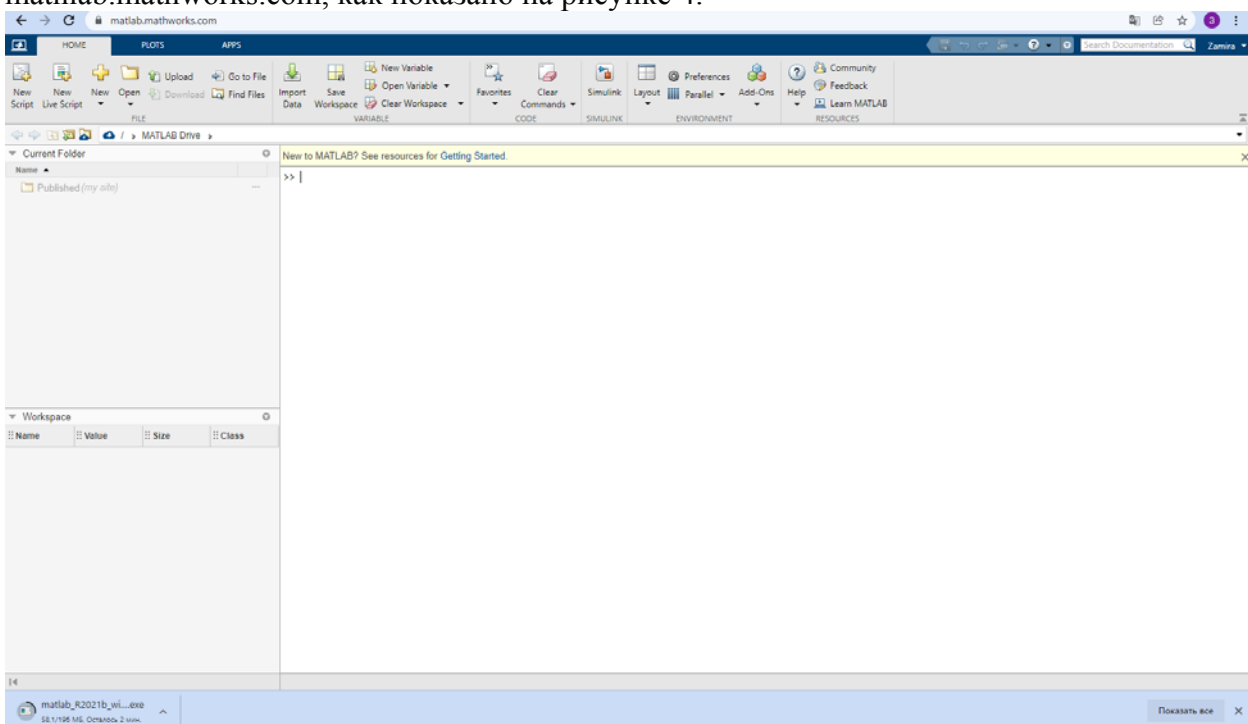


Рисунок 4 – Окно браузера для работы онлайн в MathLab

Выполнить задание

1.1. Вычислить произведение полиномов $P(x) = x^5 + x^3 + 1$ и $Q(x) = x^2 + 2x + 3$.

```
>> P=[1 0 1 0 0 1]; %вектор коэффициентов полинома P(x)
>> Q=[1 2 3]; %вектор коэффициентов полинома Q(x)
>> pr=conv(P,Q) %произведение
pr = 1 2 4 2 3 1 2 3
```

1.2. Найти минимум функции $y = x^2 - 3x + 2$ на отрезке $[-5, 5]$.

```
>> y=inline('x.^2-3*x+2'); %функция пользователя
>> x=-5:0.1:5; %вектор значений x
>> x1=fminbnd(y,-5,5); %определение абсциссы x1 точки минимума
>> %форматированный вывод на экран значений x1 и y(x1)
>> fprintf('ymin = %3.1f',y(x1)); fprintf(' при x = %2.1f',x1);
ymin = -0.3 при x = 1.5
```

1.3. Найти частичную сумму S_n ряда $\sum (-1)^{k+1}/k^2$. Сравнить ответ с точным значением суммы $S = \pi^2/12$.

```
>> n=input(' n='); %интерактивный ввод параметра n
n=100
>> for k=1:n a(k)= (-1)^(k+1)*k^(-2); end %вычисление слагаемых ряда
>> s=sum(a) %суммирование
s = 0.8224
>> pi^2/12
ans = 0.8225
```

1.4. Определить, находится ли точка $M(1.8; 1.8)$ внутри области D , заданной неравенствами: $x^2 + y^2 \geq 4$, $|x| \leq 2$, $|y| \leq 2$.

```
>> x=1.8; y=1.8;
>> z=and(x^2+y^2>=4,and(abs(x)<=2,abs(y)<=2));
>> if z ==1 disp('да'); else disp('нет'); end
да
```

1.5 Вычислить гамма-функцию $\Gamma(x)$ для $x = 0.1, 0.2, \dots, 10$.

1.6 Вычислить все значения корня n -й степени из комплексного числа $z = -64$, используя формулу: $z^{1/n} = r^{1/n} [\cos((\varphi + 2k\pi)/n) + i \sin((\varphi + 2k\pi)/n)]$, где $k = 0, \dots, n-1$; r, φ - модуль и аргумент числа z . Сравнить полученный результат с результатом выполнения команды $x = z^{1/n}$.

1.7 Вычислить среднее значение и дисперсию для элементов вектора x .

Первый шаг – создание функции stat:

```
function [mx,dx]=stat(x) %объявление имени функции
n=length(x); %вычисление длины вектора x
mx=sr(x,n); %вычисление среднего значения с помощью подфункции sr
dx=sum((x-sr(x,n)).^2)/n; %вычисление дисперсии
function s=sr(x,n) %подфункция в теле основной функции
s=sum(x)/n;
```

Второй шаг – сохранение функции в виде m-файла под именем stat.m.

Третий шаг - использование функции stat в командной строке:

```
>> x=[-1,2.2,3.1,-4,5];
>> [mx,dx]=stat(x)
mx = 1.060
dx = 10.167
```

1.8 Построить таблицу значений функции y на отрезке $[x1; x2]$ с шагом dx , записать ее в файл tab.txt и вывести таблицу на экран. Использовать программу для вычисления значений функции $y = \ln x$ на отрезке $[1; 2]$ с шагом 0.1.

1 шаг - создание файла – функции:

```

function []=table(func,x1,x2,dx)
f=inline(func);           %функция пользователя, строка func задается в командном окне
x=x1:dx:x2;                %вектор значений аргумента
y=f(x);                    %вектор значений функции
fid=fopen('tab.txt','wt'); %открытие файла tab.txt для записи
if fid ~ = -1
    fprintf(fid,'-----\n') %линия (можно также начертить линию с...
                                %помощью функций blanks и strep)
    fprintf(fid,'| %9s | %9s |\n','x','y'); %вывод названий столбцов
    fprintf(fid,'-----\n'); %линия
    fprintf(fid,'| %1.5f | %1.5f |\n',[x y]); % вывод чисел
    fprintf(fid,'-----\n'); %линия
    fclose(fid); %заккрытие файла
    fid= fopen('tab.txt','rt'); % открытие файла tab.txt для чтения
    while~feof(fid) % до конца файла
        stroka=fgetl(fid); %считывание строк таблицы
        disp(stroka) %вывод на экран
    end
    fclose(fid); %заккрытие файла
else fprintf('Ошибка при открытии файла!')
end

```

2 шаг – сохранение файла под именем table.m.

3 шаг - построение таблицы для функции $y=\ln x$:

```

>> x1=1; x2=2; dx=.1;
>> func='log(x)';
>> table(func,x1,x2,dx)

```

x	y
1.00000	0.00000
1.10000	0.09531
1.20000	0.18232
1.30000	0.26236
1.40000	0.33647
1.50000	0.40547
1.60000	0.47000
1.70000	0.53063
1.80000	0.58779
1.90000	0.64185
2.00000	0.69315

1.9 Подсчитать количество положительных чисел в произвольном числовом массиве, вывести положительные числа на экран.

1 шаг - создание файла – сценария:

```

x=input('Исходный массив чисел:'); %ввод исходного числового массива
y=x(:)>0; % логический массив
if any(y)
    s=sum(y);
    string=['В массиве содержится ', num2str(s), ' положительных чисел'];
disp(string)
    x(y')
else
    string=['В массиве нет положительных чисел'];
    disp(string)
end

```

2 шаг – сохранение файла под именем prim2_3.m.

3 шаг - проверка работы программы:

```

>> prim2_3

```

Исходный массив чисел: [-1,2.2,3.1,-4,5,8,9.3]
В массиве содержится 5 положительных чисел
ans = 2.2 3.1 5 8 9.3

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. Для чего используется система MATLAB?
2. Как осуществляется программирование в среде MATLAB?
3. Для чего используется пакет Simulink?
4. Как с помощью Simulink можно моделировать поведение сложных систем?

Список литературы, рекомендуемый к использованию по данной теме

Основная литература:

1. Голубева, Н.В. Математическое моделирование систем и процессов. - Санкт-Петербург: Лань, 2021;
2. Казиев, В.М. Введение в анализ, синтез и моделирование систем: Учебное пособие. - Москва, Саратов: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020;
3. Пospelов, Е.А., Попов, И.С. Пакеты прикладных программ в научных исследованиях: учебно-методическое пособие. - Омск: Омский государственный университет им. Ф.М. Достоевского, 2019.

Дополнительная литература:

1. Агалаков, С.А. Анализ данных в среде R: практикум. - Омск: Омский государственный университет им. Ф.М. Достоевского, 2020;
2. Воскобойников, Ю.Е. Регрессионный анализ данных в пакете MATHCAD. - Санкт-Петербург: Лань, 2021.

Интернет-ресурсы

1. <http://biblioclub.ru> – ЭБС «Университетская библиотека on-line», <http://old.exponenta.ru/> – Образовательный математический сайт по математике и программированию
2. <http://MATLAB.exponenta.ru/Simulink/default.php> – сайт, посвященный интерактивному инструменту для моделирования Simulink,
3. <https://www.intuit.ru/> – Национальный Открытый Университет «ИНТУИТ»
4. <http://window.edu.ru/> – Информационная система «Единое окно доступа к образовательным ресурсам»
5. <http://artspb.com/> – Образовательный портал: математика, кибернетика и программирование дополнительного образования
6. <http://www.arenasimulation.com> – Официальный сайт разработчика среды имитационного моделирования ARENA
7. <https://www.anylogic.ru> – Официальный сайт разработчика AnyLogic

Лабораторная работа 2. Операции с рабочей областью. Переменные и функции системы MATLAB

Цель лабораторной работы:

формирование у студентов навыков выполнения операций с рабочей областью, изучение переменных и функций системы MATLAB.

Организационная форма занятий: решение проблемных задач, разбор конкретных ситуаций

Вопросы для обсуждения на лабораторном занятии: операции с рабочей областью. Переменные и функции системы MATLAB.

Формируемые компетенции: ОПК-2, ОПК-3, ОПК-4, ОПКД-2, ОПКД-3, ОПКД-4.

Основные сведения

Переменные и определения новых функций хранятся в рабочей области, которая при длительной работе в командном режиме перестает быть непрерывной и засоряется. Команда **save** позволяет сохранять значения переменных в бинарных файлах с расширением **.mat**. Команда может быть записана в одной из форм:

save fname – все переменные записываются в файл с именем **fname.mat**

save fname X – переменная **X** записывается в файл с именем **fname.mat**

save fname X Y Z – переменные **X, Y, Z** записываются в файл с именем **fname.mat**

После параметров команды могут быть указаны ключи, уточняющие формат записи:

mat (двоичный по умолчанию), **ascii** (8 цифр), **ascii-double** (16 цифр), **append** (добавление в уже существующий файл).

Для загрузки рабочей области служит команда **load fname** в аналогичных формах.

В системе MATLAB есть возможность сохранить текст уже набранных команд с помощью дневника сессии. Для этого служат следующие команды:

diary file_name – записывает весь текст сессии в файл с расширением **.txt**, **diary off** – приостанавливает запись, **diary on** – вновь начинает запись, **type file_name** – позволяет просмотреть записанный текстовый файл.

Простейший объект языка MATLAB – число. Примеры представления чисел: 15, -4, 3.205, 12.35e-24 (пробелы между символами в записи числа не допускаются!), 2+3.00i (мнимая часть комплексного числа имеет множитель **i** или **j**). Все операции над числами производятся в формате с двойной точностью. Выдача результата производится в нормализованной форме с четырьмя цифрами после десятичной точки и одной до нее. Для смены формата вывода чисел служит команда

>> format name

где **name**: 1) **short** (5 знаков), 2) **short e** (5 знаков мантиссы и 3 знака порядка), 3) **long** (15 знаков), 4) **long e** (15 и 3), 5) **hex** (в 16-тиричном формате), 6) **bank** (для денежных единиц).

Система обладает большим набором встроенных функций (математических и др.), позволяющих производить над числами, помимо обычных арифметических операций, самые разнообразные действия. Так, например, функции **real(z)**, **imag(z)**, **abs(z)**, **angle(z)** возвращают соответственно действительную и мнимую части комплексного числа, его модуль и аргумент.

В MATLAB имеются системные константы, которые могут переопределяться, например: **i** или **j** (мнимая единица), **pi** (число π), **eps** (2^{-52}), **realmin** ((2^{-1022})), **realmax** (2^{1023}), **inf** (∞), **ans**, **NaN** (Not-a-Number – нечисловой характер данных или неопределенность), **computer**.

Еще один объект языка - символьная константа, т.е. цепочка символов, заключенных в апострофы, например: 'MATLAB – наш помощник'.

Переменные в MATLAB могут иметь имя, состоящее из любого количества символов (идентифицируются первые 31), имя должно начинаться с буквы, но может содержать цифры и символ подчеркивания. Строчные и прописные буквы различаются! Значения переменных хранятся в рабочей области, для очистки которой используется команда **clear** в разных формах: **clear X** (стирается одна переменная **X**), **clear a b c** (несколько переменных), **clear all** (все переменные). Команда **who** перечисляет все переменные, хранящиеся в рабочей области на данный момент. Более подробную информацию выдает команда **whos**.

Все переменные в MATLAB рассматриваются как матрицы, вектора или массивы. Тип переменной явно не указывается. Размер самых простых переменных (чисел) 1x1, они допускают основные арифметические действия **+**, **-**, *****, **/** с числовой переменной, такие действия выполняются

поэлементно. Для матриц и векторов поэлементные операции (они обозначаются добавлением знака `.` (точка) перед соответствующей операцией, например, `.*` и `./`) возможны лишь для операндов с одинаковыми размерами, например:

```
>> v1=[3 6 9 12];
>> v2=[1,2,3,4];
>> v1/v2
ans=3
>> v1.*v2
ans=3 12 27 48
>> v1./v2
ans=3 3 3 3
```

Операция обычного умножения (без знака `.`) выполняется по правилам линейной алгебры. Используются также операции `\` и `.\` (левое матричное деление), `^` и `.^` (возведение в степень и поэлементное возведение в степень).

Специфическая операция `:` (двоеточие) применяется для формирования упорядоченных последовательностей (начальное значение : шаг : конечное значение, по умолчанию шаг=1), например:

```
>> 2:7
ans =
    2    3    4    5    6    7
```

Операция `:` широко используется при работе со средствами построения графиков, а также при формировании матриц и векторов. Выражения с операцией `:` могут использоваться в качестве аргументов функций, например:

```
>> %построение графика функции cos, аргумент меняется от 0 до 5 с шагом 0.2
>> plot(cos(0:0.2:5))
>> % вычисление функции Бесселя порядка от 0 до 5 с аргументом 0.5
>> bessel (0:5,1/2)
ans =
    0.9385    0.2423    0.0306    0.0026    0.0002    0.0000
```

Помимо арифметических, в языке MATLAB существуют операции отношения (`=`, `~=`, `<`, `>`, `<=`, `>=`), логические операции (`&`, `and`, `or`, `not`, `~`, `xor`, `any`, `all`), а также операции `=` (присваивание), `.'` (простое транспонирование), `'` (комплексно-сопряженное транспонирование), `[,]` (горизонтальное соединение), `;` (вертикальное соединение).

MATLAB содержит большой набор (более тысячи) встроенных функций, элементарных и специальных, от одного или нескольких аргументов. Наиболее значимые функции перечислены в Приложении (для краткости приведены лишь названия функций без указания аргументов). Более подробную информацию о конкретной функции можно получить, подав команду: `help имя функции`. Представление о полном наборе операторов, конструкций языка и системных функций можно получить с помощью команды: `help имя`, где `имя` – название соответствующей подпапки в папке MATLAB/TOOLBOX/MATLAB, а именно:

- `general` – команды общего назначения;
- `ops` - операторы и специальные символы;
- `lang` – конструкции языка;
- `strfun` – строковые функции;
- `iofun` – функции ввода-вывода;
- `timefun` – функции времени и дат;
- `datatypes` – типы и структуры данных;
- `elmat` – операции с элементарными матрицами;
- `elfun` – элементарные математические функции;
- `specfun` – специальные математические функции;
- `matfun` – матричные функции линейной алгебры;
- `datafun` – анализ данных и преобразования Фурье;
- `polyfun` – полиномиальные функции и функции интерполяции;
- `funfun` – функции функций и функции решения ОДУ;
- `soarfun` – функции разреженных матриц;
- `graph2d` – команды двумерной графики;

graph3d – команды трехмерной графики;
spesgraph – команды специальной графики;
graphics – команды дескрипторной графики;
uitools – графика пользовательского интерфейса.

Язык MATLAB предоставляет возможность задать функцию пользователя. Для этого используется функция inline, аргументом которой является символьная строка, задающая функцию одной или нескольких переменных, например:

```
>> f=inline('1+sin(x)^2');  
>> y=f(pi/2)  
y=2
```

Ту же задачу можно решить с помощью оператора eval(s), который вычисляет значение выражения, заданного в виде символьной переменной s:

```
>> x=pi/2;  
>> y=eval('1+sin(x)^2')  
y=2
```

Функцию можно задать также в виде m-файла (см. п. 5) или с помощью оператора @ (так называемая дескрипторная или handle-функция).

4.

Выполнить задание

2.1. Вычислить все корни полинома $P(x) = x^7 + 3.2x^5 - 5.2x^4 + 0.5x^2 + x - 3$.

Для одного из корней сделать проверку.

```
>> p=[1 0 3.2 -5.2 0 0.5 1 -3];           % коэффициенты полинома  
>> r=roots(p)                             % вычисление корней  
r=  
    -0.5668 + 2.0698i  
    -0.5668 - 2.0698i  
    -0.6305 + 0.5534i  
    -0.6305 - 0.5534i  
     1.2149  
     0.5898 + 0.6435i  
     0.5898 - 0.6435i  
>> polyval(p,1.2149)                     % вычисление значения полинома при x=1.2149  
ans =  
     0.0000
```

2.2 Найти нули функции $y = 0.25x + \sin x - 1$ на отрезке $[0; 10]$.

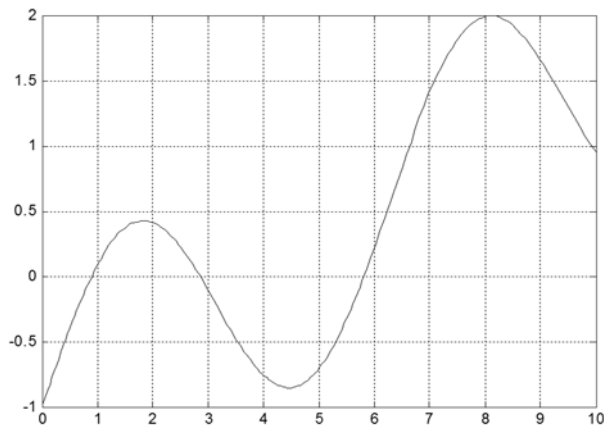
1 шаг – создание функции:

```
function f=fun1(x)  
f=0.25*x+sin(x)-1;
```

2 шаг – сохранение файла fun1.m

3 шаг – работа в командном окне:

```
>> fplot(@fun1,[0,10]); %построение графика функции fun1 на отрезке [0; 10]...  
                        для приближенного определения нулей функции  
>> grid on;           % координатная сетка
```

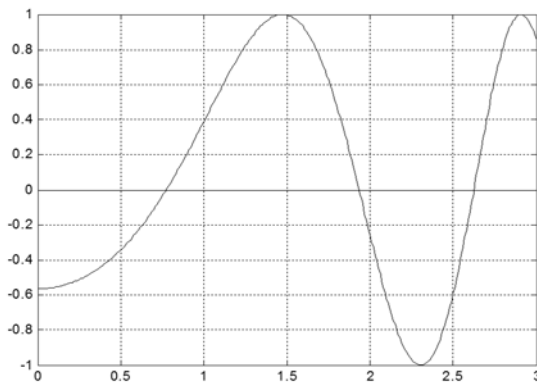


```
>> x1=fzero(@fun1,[0 1])           %вычисление нуля функции на отрезке [0; 1]
x1 =
    0.8905
>> x2=fzero(@fun1,[2 3])           %вычисление нуля функции на отрезке [2; 3]
x2 =
    2.8500
>> x3=fzero(@fun1,5,0.001)         %вычисление нуля функции при x ≥ 5,...
                                     точность=0.0001
x3 =
    5.8128
```

2.3 Решить уравнение: $\sin(x^2 - 0.6) = 0$ на отрезке $[0; 3]$.

1. Графический способ:

```
>> x=0:.01:3;                     %значения аргумента
>> f=sin(x.^2-0.6);               %значения функции
>> plot(x,[f;0*f])                % графики функций y=f и y=0
>> grid on;                       %координатная сетка
```



```
>> x1=ginput                       %интерактивный вывод на экран координат точки (щелчок...
                                     мыши по нужной точке и нажатие клавиши enter)
```

```
x1 =
    0.7746   -0.0012                %второе число соответствует координате y1≈0
>> g2=ginput
x2 =
    1.9343    0.0023
>> x3= fzero(f,[2 3])
x3 =
    2.6326
```

2. Аналитический способ:

```
>> x=0:.01:3;                     %значения аргумента
>> n=length(x);                   %вычисление длины вектора x
>> ind=1:n-1;                     %вектор индексов
>> f=sin(x.^2-0.6);               % значения функции
```

```
>> korni=x(f(ind).*f(ind+1)<=0)      %сравниваются знаки функции на концах...
                                     отрезка, вектор korni попадают только те...
                                     значения  $x_i$ , для которых  $f(x_i)$  и  $f(x_{i+1})$ ...
                                     имеют разные знаки
```

korni =
0.7746 1.9300 2.6200

2.4 Составить программу для решения алгебраического уравнения $f(x) = 0$ методом Ньютона (итерационная процедура по формуле: $x_{k+1} = x_k - f(x_k) / f'(x_k)$, начальное значение x_0 , функция f и ее производная f' задаются в командном окне). Применить программу к решению уравнения из примера 3.3.

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. Назовите возможные операции с рабочей областью MATLAB.
2. Перечислите типы переменных MATLAB.
3. Перечислите функции системы MATLAB.

Список литературы, рекомендуемый к использованию по данной теме

Основная литература:

1. Голубева, Н.В. Математическое моделирование систем и процессов. - Санкт-Петербург: Лань, 2021;
2. Казиев, В.М. Введение в анализ, синтез и моделирование систем: Учебное пособие. - Москва, Саратов: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020;
3. Поспелов, Е.А., Попов, И.С. Пакеты прикладных программ в научных исследованиях: учебно-методическое пособие. - Омск: Омский государственный университет им. Ф.М. Достоевского, 2019.

Дополнительная литература:

1. Агалаков, С.А. Анализ данных в среде R: практикум. - Омск: Омский государственный университет им. Ф.М. Достоевского, 2020;
2. Воскобойников, Ю.Е. Регрессионный анализ данных в пакете MATHCAD. - Санкт-Петербург: Лань, 2021.

Интернет-ресурсы

1. <http://biblioclub.ru> – ЭБС «Университетская библиотека on-line», <http://old.exponenta.ru/> – Образовательный математический сайт по математике и программированию
2. <http://MATLAB.exponenta.ru/Simulink/default.php> – сайт, посвященный интерактивному инструменту для моделирования Simulink,
3. <https://www.intuit.ru/> – Национальный Открытый Университет «ИНТУИТ»
4. <http://window.edu.ru/> – Информационная система «Единое окно доступа к образовательным ресурсам»
5. <http://artspb.com/> – Образовательный портал: математика, кибернетика и программирование дополнительного образования
6. <http://www.arenasimulation.com> – Официальный сайт разработчика среды имитационного моделирования ARENA
7. <https://www.anylogic.ru> – Официальный сайт разработчика AnyLogic

Лабораторная работа 3. Формирование векторов и матриц

Цель лабораторной работы:

формирование у студентов способностей создания векторов и матриц в системе MATLAB.

Организационная форма занятий: решение проблемных задач, разбор конкретных ситуаций

Вопросы для обсуждения на лабораторном занятии: Формирование векторов и матриц в MATLAB.

Формируемые компетенции: ОПК-2, ОПК-3, ОПК-4, ОПКД-2, ОПКД-3, ОПКД-4.

Основные сведения

Если надо задать вектор из n элементов, то их значения следует перечислить в квадратных скобках через пробел или через запятую, например:

```
>> B=[2 -3 4 6];
```

При задании матрицы используется символ ; (точка с запятой) для разграничения строк:

```
>> A=[1 3; 0 -3];
```

Можно вводить элементы векторов и матриц в виде арифметических выражений, содержащих доступные системе функции. Элементы матриц могут быть комплексными. Для указания отдельного элемента используется выражение вида $B(k)$ или $A(m,n)$, например:

```
>> A(1,2)
```

```
ans=3
```

Выражение $A(k)$ дает доступ к элементам матрицы, развернутым в один столбец:

```
>> A(4)
```

```
ans=-0.3
```

Индекс k может быть логическим выражением:

```
>> A(A>0)
```

```
ans= 1 3
```

В качестве индекса может быть вектор. Например, запись $A([1,2],2)$ обозначает первый и второй элементы второго столбца. Для указания на m -ю строку или n -й столбец матрицы используются выражения вида $A(m,:)$ или $A(:,n)$. Запись $A(k:m;n)$ обозначает вектор-столбец, сформированный из нескольких элементов n -го столбца матрицы A (с k по m -й).

Имеется ряд функций для задания особых векторов и матриц, например $\text{magic}(n)$ (матрица размером $n \times n$, у которой сумма строк, сумма столбцов и сумма диагоналей одинаковы), $\text{hadamard}(n)$ (матрица Адамара), $\text{hilb}(n)$ (матрица Гильберта), $\text{zeros}(m,...,q)$ (многомерный массив из нулей), $\text{ones}(m,...,q)$ (массив из единиц), $\text{rand}(m,...,q)$ (массив из случайных чисел), $\text{eye}(n)$ (единичная матрица), $\text{repmat}(A,m,n)$ ($m \times n$ дублей матрицы A , m вниз, n вправо), $\text{linespace}(a,b,n)$ (вектор-строка из n чисел, равномерно расположенных на отрезке $[a,b]$, по умолчанию $n=100$), $\text{logspace}(a,b,n)$ (узлы логарифмической сетки по десятичному логарифму, по умолчанию $n=50$) и др.

Элементами матрицы могут быть вектора и матрицы, что позволяет формировать большие матрицы, объединяя малые. Для удаления строк или столбцов удобно использовать пустые квадратные скобки $[]$:

```
>> A(1,:)=[]
```

```
A= 0 -0.3
```

Основную информацию о векторах и матрицах можно получить с помощью функций $\text{size}(A)$ (вектор-строка с размерами матрицы A), $\text{length}(B)$ (число компонентов вектора B), $\text{ndims}(X)$ (размерность массива X), $\text{disp}(X)$ (массив X без его имени), $\text{isempty}(X)$ ($=1$, если $X=[]$), $\text{isequal}(X,Y)$ ($=1$, если $X=Y$), $\text{isnumeric}(X)$ ($=1$, если X – числовая переменная), $\text{islogical}(X)$ ($=1$, если X – логическая переменная).

При формировании матриц можно использовать следующие функции:

$\text{reshape}(X,m,n)$ – формирует матрицу $m \times n$ из элементов массива X ;

$\text{diag}(A,k)$ – формирует вектор-столбец из k -й диагонали A , по умолчанию – из главной;

$\text{tril}(A)$ – формирует нижнюю треугольную матрицу для A ;

$\text{triu}(A)$ – формирует верхнюю треугольную матрицу для A ;

$\text{fliplr}(A)$ – меняет на обратный порядок столбцов;

$\text{flipud}(A)$ – меняет на обратный порядок строк;

$\text{rot90}(A,k)$ – поворачивает A на 90 градусов k раз;

end – на месте индекса задает его последнее значение.

Определитель квадратной матрицы вычисляется с помощью функции $\det(A)$. Для вычисления обратной матрицы методом Гаусса-Жордана используется функция $\text{inv}(A)$.

Выполнить задание

3.1. Вычислить алгебраические дополнения к элементам матрицы A и обратную к этой матрице (A – квадратная матрица третьего порядка). Результат сравнить с функциями $\text{inv}(A)$ и $A^{(-1)}$.

```
function [C,C1,C2]=obr(A)    % объявление функции, сохраняемой по именем obr.m
d=det(A)                    %вычисление определителя матрицы
if d==0 disp('матрица вырожденная');
else
    A11=det(A([2,3],[2,3]));
    A12=-det(A([2,3],[1,3]));
    A13=det(A([2,3],[1,2]));
    A21=-det(A([1,3],[2,3]));
    A22=det(A([1,3],[1,3]));
    A23=-det(A([1,3],[1,2]));
    A31=det(A([1,2],[2,3]));
    A32=-det(A([1,2],[1,3]));
    A33=det(A([1,2],[1,2]));
    B=[A11,A12,A13;A21,A22,A23;A31,A32,A33];
    C=B'/d;
    C1=inv(A);
    C2=A^(-1);
end                          %конец оператора if, далее сохраняем функцию в файле obr.m
```

Работа в командном окне:

```
>> A=[1 2 1;2 11 1;2 1 2] %исходная матрица
>> [C,C1,C2]=obr(A) %вызов функции
C =
    -7.0000    1.0000    3.0000
     0.6667    0.0000   -0.3333
     6.6667   -1.0000   -2.3333
C1 =
    -7.0000    1.0000    3.0000
     0.6667    0.0000   -0.3333
     6.6667   -1.0000   -2.3333
C2 = C1
```

3.2. Решить систему уравнений:

$$\begin{aligned} 2x + y - 5z + t &= 8, \\ x - 3y - 6t &= 9, \\ 2y - z + 2t &= -5, \\ x + 4y - 7z + 6t &= 0. \end{aligned}$$

```
>> A=[2 1 -5 1;1 -3 0 -6;0 2 -1 2;1 4 -7 6]; %матрица системы
>> B=[8;9;-5;0]; %вектор – столбец правых частей (может быть матрицей)
>> A1=[A,B]; %расширенная матрица системы
>> if and(rank(A)=rank(A1),rank(A)=4) %проверка ранга матрицы
    disp('система имеет единственное решение:');
    x=A\B; % обратный слэш или деление слева – решение линейной...
            системы методом Гаусса
    x1=x';
end
x1 =
    3.0000   -4.0000   -1.0000    1.0000
>> x=A^(-1)*B; %второй вариант записи A\B
x2 =
```

```

x2 =
    3.0000 -4.0000 -1.0000  1.0000
>> x=inv(A)*B; x3=x' %третий вариант записи A\B
x3 =
    3.0000 -4.0000 -1.0000  1.0000

```

3.3. Решить систему из примера 4.2 методом наименьших квадратов.

```

>> A=[2 1 -5 1;1 -3 0 -6;0 2 -1 2;1 4 -7 6]; %матрица системы
>> B=[8;9;-5;0]; % вектор – столбец правых частей
>> x=lsqr(A,B) % встроенная функция решения линейной системы...
                (метод наименьших квадратов)

x =
    3.0000 -4.0000 -1.0000  1.0000

```

3.4 Решить систему из примера 4.2 с помощью формул Крамера.

3.5 Решить систему нелинейных уравнений:

$$\begin{aligned} 2x - y^2 + z &= 1, \\ y^2 - z^2 &= 0, \\ 2y - 3z &= 1, \end{aligned}$$

для решения применить метод последовательных приближений
(итерационная процедура $x_{k+1} = B \setminus A(x_k)$).

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. Опишите формирование векторов MATLAB
2. Опишите формирование матриц MATLAB

Список литературы, рекомендуемый к использованию по данной теме

Основная литература:

1. Голубева, Н.В. Математическое моделирование систем и процессов. - Санкт-Петербург: Лань, 2021;
2. Казиев, В.М. Введение в анализ, синтез и моделирование систем: Учебное пособие. - Москва, Саратов: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020;
3. Поспелов, Е.А., Попов, И.С. Пакеты прикладных программ в научных исследованиях: учебно-методическое пособие. - Омск: Омский государственный университет им. Ф.М. Достоевского, 2019.

Дополнительная литература:

1. Агалаков, С.А. Анализ данных в среде R: практикум. - Омск: Омский государственный университет им. Ф.М. Достоевского, 2020;
2. Воскобойников, Ю.Е. Регрессионный анализ данных в пакете MATHCAD. - Санкт-Петербург: Лань, 2021.

Интернет-ресурсы

8. <http://biblioclub.ru> – ЭБС «Университетская библиотека on-line», <http://old.exponenta.ru/> – Образовательный математический сайт по математике и программированию
9. <http://MATLAB.exponenta.ru/Simulink/default.php> – сайт, посвященный интерактивному инструменту для моделирования Simulink,
10. <https://www.intuit.ru/> – Национальный Открытый Университет «ИНТУИТ»

11. <http://window.edu.ru/> – Информационная система «Единое окно доступа к образовательным ресурсам»
12. <http://artspb.com/> – Образовательный портал: математика, кибернетика и программирование дополнительного образования
13. <http://www.arenasimulation.com> – Официальный сайт разработчика среды имитационного моделирования ARENA
<https://www.anylogic.ru> – Официальный сайт разработчика AnyLogic

Лабораторная работа 4. Создание и отладка программных файлов. Работа с файлами, импорт и экспорт данных

Цель лабораторной работы:

формирование у студентов способностей создания и отладки программных файлов, формирование навыков работы с файлами, импорт и экспорт данных.

Организационная форма занятий: решение проблемных задач, разбор конкретных ситуаций

Вопросы для обсуждения на лабораторном занятии: Создание и отладка программных файлов. Работа с файлами, импорт и экспорт данных в MATLAB.

Формируемые компетенции: ОПК-2, ОПК-3, ОПК-4, ОПКД-2, ОПКД-3, ОПКД-4.

Основные сведения

Программами в системе MATLAB являются m-файлы текстового формата, содержащие запись программ в виде программных кодов. Основными средствами программирования являются:

- данные различных типов,
- операторы,
- функции пользователя,
- управляющие структуры,
- системные операторы и функции,
- средства расширения языка.

В MATLAB определены следующие основные типы данных: single, double, char, logical, sparse, cell, struct, function_handle, Java classes и др. Кроме того, предусмотрен тип данных UserObject (тип данных, определяемый пользователем). Каждому типу данных можно соотнести характерные для него операции.

Язык программирования системы MATLAB ориентирован на структурное программирование, в нем нет номеров строк и оператора безусловного перехода GO TO. Имеются управляющие структуры следующих типов:

- условный оператор **if...elseif...else...end**, простейший вариант: **if** условие инструкции **end**;
- цикл типа **for...end**, конструкция цикла имеет вид: **for** var=b:s:f, инструкции, **end**, где b – начальное значение переменной цикла var, s – шаг (по умолчанию s=1), f – конечное значение переменной var;
- цикл типа **while...end**, конструкция цикла имеет вид: **while** условие инструкции **end**, досрочный выход из цикла может быть реализован с помощью операторов **break** и **continue**;
- переключатель **switch...case...otherwise...end**;
- конструкция **try...catch...end**.

Имеются также функции диалогового ввода и вывода:

- **r=input** ('String') или **input** ('String','s');
- **disp(X)**.

Функция **pause** (или **pause(N)**) позволяет приостановить выполнение программы, N – количество секунд. Функции **tic** и **toc** позволяют включить и выключить таймер.

Программы вызываются по имени, под которым записаны в виде файла с расширением .m. Имя программы начинается с латинской буквы и состоит из таких букв и цифр. При первом обращении программа ищется на диске. Рекомендуется записать программу в текущую директорию. Поиск m-файла по ключевому слову осуществляется командой **lookfor**. Путь к программе можно указать командой **path**.

Программы бывают двух видов: сценарии (scripts) и функции (functions). Программа-скрипт представляет собой набор команд без входных и выходных параметров. Она может использовать все переменные рабочей области, а не только те, что создаются в ней. Собственные переменные скрипта располагаются в рабочей области без защиты от внешнего доступа, что позволяет передать управление из скрипта в командную строку с помощью команды **keyboard** (без параметров), при этом работа в командной строке заканчивается командой **return**. Таким образом,

скрипт удобен для работы в интерактивном режиме, но требует повышенного внимания при использовании в нем своих переменных.

Пример программы-скрипта (определение локальных максимумов вещественного вектора):

```
m=length(x); % вычисление длины вектора x
v=2:m-1; % формирование вектора индексов
i=find(x(v-1)<x(v)&x(v)>x(v+1))+1; % поиск индексов локальных максимумов
y=x(i); % значения локальных максимумов
```

После набора программы ее следует сохранить, например, под именем lokmax.m. Теперь возможен ее вызов из командной строки:

```
>> x=[1,2,-3,3,-2,4]; lokmax, [y;i]
y = 2 3
i = 2 4
```

Комментарий, включенный в программу, выводится командой:

```
>> help имя файла
```

Первая строка программы-функции имеет вид: **function** [y1,y2,...yj]=ff(x1,x2,...xi), где ff – имя функции, x1,x2,...xi – входные формальные параметры, y1,y2,...yj – выходные формальные параметры. Обращение к функции записывается так: [v1,v2,...vj]=ff(u1,u2,...ui), где u1,u2,...ui,v1,v2,...vj – фактические параметры. Входных аргументов у функции может и не быть. Все переменные программы-функции имеют локальный характер и недоступны для их использования извне. Если при создании файла-функции желательно применение глобальных переменных, то следует подать команду: **global** var1 var2... . Таким образом, программа-функция неудобна для использования в интерактивном режиме, но любая ее переменная может быть открыта для использования в командной строке.

Пример программы-функции (та же задача):

```
function [y,i]=lokmax(x)
m=length(x);
v=2:m-1;
i=find(x(v-1)<x(v)&x(v)>x(v+1))+1;
y=x(i);
```

После сохранения функции к ней можно обратиться из командной строки:

```
>> x=[1,2,-3,3,-2,4]; [y,i]= lokmax(x)
```

и получить тот же ответ.

Файлы-функции могут вызываться так же из других m-файлов. Начиная с версии 5.0 появилась возможность включать в тело основной функции подфункции. Они имеют идентичную конструкцию.

При создании программ-функций часто бывают полезны две функции:

- nargin – возвращает число входных параметров данной функции;
- nargout - возвращает число выходных параметров данной функции;

Иногда в программе-функции бывает необходимо выдать сообщение об ошибке, для этого служит команда **error** ('сообщение').

Важным этапом программирования является отладка программы. MATLAB располагает большими возможностями, облегчающими процесс отладки. В частности, полный листинг программного файла можно вывести с помощью команды:

```
>> type имя файла (или для вывода листинга с пронумерованными строками dbtype)
```

Для установки в тестируемый m-файл точек прерывания используются команды:

```
>> dbstop in M-file at lineno – установить точку прерывания в заданной строке;
```

```
>> dbstop in M-file at subfun – установить точку прерывания в подфункции;
```

```
>> dbstop if error – установить точку прерывания при сообщении об ошибке.
```

Команда **dbstatus** (без параметров) выводит список установленных в данной сессии точек прерывания. Для удаления точек прерывания используется команда **dbclear** с тем же синтаксисом, что и **dbstop**, например

```
>> dbclear M-file at lineno.
```

Для пошагового выполнения программы используется команда **dbstep** (или **dbstep nlines** для выполнения заданного числа строк).

Для редактирования и отладки m-файлов в MATLAB существует специальный многооконный редактор, выполненный как типичное приложение Windows. Он позволяет

нумеровать строки, устанавливать контрольные точки, вычислять переменные по месту нахождения, получать справку и прочее (вход в редактор осуществляется из главного меню: (File – M-file).

Работа с файлами

Открыть файл в системе MATLAB можно с помощью специального Мастера импорта Import Wizard (Import Data в меню File). Возможности Мастера достаточно очевидны, они зависят от того, какие данные импортируются. Можно также использовать обычные файловые операции чтения и записи файлов в тех или иных форматах. Операции импорта и экспорта файлов открывают обширные возможности по обмену данными между системой MATLAB и другими программами. Вот некоторые из них:

open имя – открывает файл в зависимости от анализа параметра имя и его расширения;

fid=fopen(filename,permission)-открывает указанный файл под управлением permission (этот параметр принимает значение 'r', если файл открывается для чтения, 'w' – для записи, 'a' – для присоединения), при открытии текстового файла к содержимому permission добавляется t (например, 'rt' или 'wt'), fid содержит идентификатор файла;

status=fclose(fid) – закрывает файл с указанным идентификатором, status=0, если операция прошла успешно и -1 при ошибке;

[A,count]=fread(fid,size,precision) – читает двоичные данные из файла с указанным идентификатором и записывает их в матрицу A, параметр count содержит число успешно прочитанных элементов (не обязателен), если параметр size не задан, то читается весь файл;

count=fwrite(fid,A,precision) – записывает двоичные данные из матрицы A в файл;

fscanf - читает форматируемые данные из файла;

fprintf(fid,format,A,...) – записывает форматируемые данные в файл, fid=1 для стандартного вывода (экран по умолчанию), fid=2 для стандартной ошибки, format - строка, одерживающая спецификаторы %, *, \, символы преобразования f, d, i, o, u, x и пр. (см. Language Reference Guide), оператор подобен одноименному оператору языка C;

g=input('запрос') - вводит выражение с клавиатуры, результат заносится в g;

imread - читает образ из графического файла;

imwrite – записывает образ в графический файл;

iminfo – возвращает информацию о графическом файле;

auread (или wavread)– считывает заданный аудиофайл;

auwrite (или wavwrite)- записывает заданную информацию в виде аудиофайла;

matlabroot– возвращает имя директории, в которой установлено программное обеспечение MATLAB.

Для позиционирования файлов можно использовать следующие команды:

frewind(fid)- устанавливает начальное значение указателя положения в файле с указанным идентификатором;

ftell(fid) – возвращает значение указателя положения;

fseek(fid,offset,origin) – устанавливает положение указателя на байт с заданным смещением offset относительно положения origin (=-1 для начала файла, =0 для текущей позиции указателя, =1 для конца файла);

feof(fid) – указывает, является ли считываемый символ признаком конца файла;

ferror(fid) – возвращает сообщение об ошибке в файле.

Для записи файлов на диск и считывания файлов с диска служат команды **load** и **save**, используемые в очевидных формах. В этих командах имя файла указывается по правилам, принятым в операционных системах класса MS-DOS.

Для запуска Мастера импорта можно использовать также команды:

uiimport (fname) – открывает файл и запускает Мастер импорта;

uiimport ('-file') – вначале выводит диалог выбора файла;

uisave – управляемое пользователем сохранение;

Команда **delete('имя файла')** удаляет файл из текущей папки.

Выполнить задание

4.1. Вычислить производную полинома $P(x) = x^5 + x^3 + 1$.

```
>> P=[1 0 1 0 0 1]; %вектор коэффициентов исходного полинома
>> P1=polyder(P) %вектор коэффициентов производной P'(x)
P1 =
5 0 3 0 0
```

4.2. Вычислить приближенное значение производной функции $y = \sin x$ и убедиться, что $y'(a) \approx \cos a$.

```
>> x=0:.05:10; %вектор аргументов
>> y=sin(x); %вектор значений функции
>> d=diff(y); %вектор разностей соседних элементов: d=[y(2)-y(1),...,y(n)-y(n-1)]
>> pr=d/0.05; %вектор значений производной
>> pr(5) %значение производной при x=x(5)=0.2
0.9747
>> cos(x(5)) %сравнение с точным результатом
0.9801
```

4.3. Найти приближенное значение производной функции $y = \ln(1-x)$ в виде полинома 5-й степени. Вычислить $y'(0.6)$.

```
>> h=.01; x=0:h:0.8; %значения аргумента
>> y=log(1-x); %значения функции
>> y1=diff(y)/h; %значения производной
>> p=polyfit(x(1:length(n)-1),y1,5) %коэффициенты аппроксимирующего полинома
p =
-52.0782 75.5962 -43.1952 8.8005 -1.8620 -0.9894
>> c=polyval(p,0.6) %значение полинома p при x=0.6
c =
-2.5209 %точное значение c= - 2.5
```

4.4. Вычислить интеграл двумя способами: $\int_0^1 (e^{2x} - 1) dx$.

1 способ – метод трапеций:

```
>> h=0.001; x=0:h:1;
>> y=exp(2*x)-1;
>> int=trapz(y)*h
int =
2.1945
```

2 способ – метод Симпсона:

```
>> int=quad('exp(2*x)-1',0,1,1.0e-5)
int = 2.1945
```

4.5. Решить задачу Коши: $y'' + 2y' + 10y = \sin x$; $y(a) = c$, $y'(a) = d$, $x \in [a, b]$,
к решению применить метод Эйлера, построить график функции $y(x)$.
Указание: записать уравнение в виде системы: $y' = z$,
 $z' = -2z - 10y + \sin x$,
разбить отрезок $[a, b]$ на n частей $[x_k, x_{k+1}]$ и применить формулу
 $V_{k+1} = V_k + h \cdot F(x_k, y_k)$, где V - вектор неизвестных, F - вектор правых
частей, h - длина интервала, $V_0 = [c; d]$.

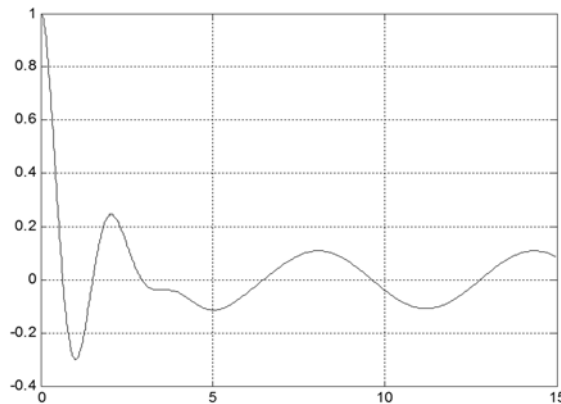
```
function y=solvejler(a,b,y0,h) % начало файла solvejler.m
xt=a; yt=y0; k=1; %начальные значения текущих переменных xt, yt
while xt<b;
    x(k)=xt;
    y(k)=yt(1);
    k=k+1;
    c=[yt(2);-2*yt(2)-10*yt(1)+sin(xt)]; %вектор правых частей
    yt=yt+h*c;
    xt=xt+h;
```

```
end
plot(x,y); grid %график решения, сетка; конец m-файла
```

Работа в командном окне:

```
>> a=0; b=15; y0=[1;0]; h=0.01;
>> solvejler(a,b,y0,h)
```

%вызов функции



4.6. Решить задачу примера 6.1 методом Адамса с помощью стандартной функции ode113(). Построить графики функций $y(x)$ и $y'(x)$.

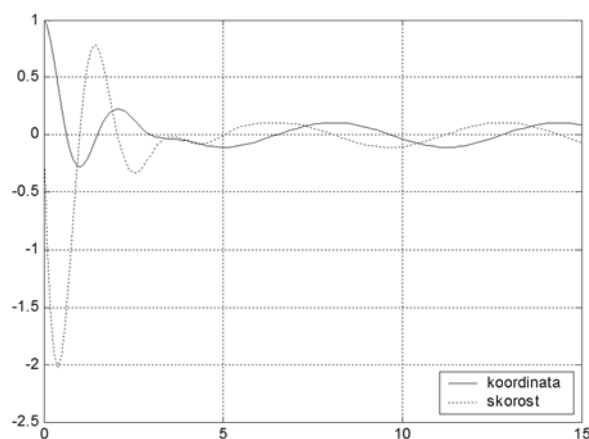
```
function F=primer(x,y)
F=[y(2);-2*y(2)-10*y(1)+sin(x)];
```

%m-файл, задающий вектор правых частей
%конец m-файла

Работа в командном окне:

```
>> y0=[1;0];
>> [X,Y]=ode113(@primer,[0 15],y0);
>> plot(X,Y(:,1),'-')
>> hold on
>> plot(X,Y(:,2),'-')
>> grid
>> legend('koordinata','skorost',4)
```

%решение задачи Коши
%график функции $y(x)$
%режим добавления графиков в текущее окно
%график функции $y'(x)$
%координатная сетка
%легенда



4.7. Решить краевую задачу: $y'' - y = x$; $y(0) = 0$, $y(1) = 1$.

Указание: разбить отрезок $[0; 1]$ на n равных частей. Производные $y''(x_k)$ заменить конечными разностями $y''_k = (y_{k-1} - 2y_k + y_{k+1}) / h^2$, $h = x_{k+1} - x_k$, $k \geq 2$, решить полученную линейную систему $Ay=B$. Построить график решения и сравнить его с графиком точного решения $y = c_1 \exp(x) + c_2 \exp(-x) - x$, где $c_1 = 2e/(e^2-1)$, $c_2 = -c_1$.

```
>> n=11; h=1/(n-1); x=0:h:1;
>> B=[0,h*h*x(2:n-1),1]';
>> k=-2-h^2;
>> A=[1 0 0 0 0 0 0 0 0 0;1 k 1 0 0 0 0 0 0 0;...
0 1 k 1 0 0 0 0 0 0;0 0 1 k 1 0 0 0 0 0;...
0 0 0 1 k 1 0 0 0 0;0 0 0 0 1 k 1 0 0 0 0;...]
```

%разбиение отрезка
%вектор правых частей системы
%матрица линейной системы

```
0 0 0 0 0 1 k 1 0 0 0; 0 0 0 0 0 0 1 k 1 0 0; ...
0 0 0 0 0 0 0 1 k 1 0; 0 0 0 0 0 0 0 0 1 k 1; ...
0 0 0 0 0 0 0 0 0 1];
```

```
>> y1=A\B;
```

%решение линейной системы

```
>> plot(x,y1); grid
```

%график приближенного решения

```
>> hold on
```

%режим добавления графиков

```
>> c1=2*exp(1)/(exp(1)^2-1); c2=-c1;
```

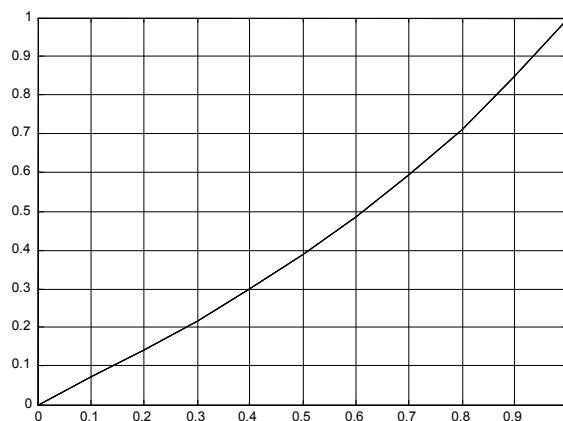
%коэффициенты точного решения

```
>> y2=c1*exp(x)+c2*exp(-x)-x;
```

%точное решение

```
>> plot(x,y2)
```

%график точного решения (совпадает с графиком y1)



Другой вариант формирования матрицы A:

```
>> v=[k,1,zeros(1,n-2)];
```

```
>> A=toeplitz(v); %матрица Тёплица с одним аргументом (см. [2])
```

```
>> A(1,1)=1; A(1,2)=0;
```

```
>> A(n,n-1)=0; A(n,n)=1;
```

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. Опишите процесс создания и отладки программных файлов в среде MATLAB.
2. Опишите процесс работы с файлами, импорт и экспорт данных в среде MATLAB

Список литературы, рекомендуемый к использованию по данной теме

Основная литература:

1. Голубева, Н.В. Математическое моделирование систем и процессов. - Санкт-Петербург: Лань, 2021;
2. Казиев, В.М. Введение в анализ, синтез и моделирование систем: Учебное пособие. - Москва, Саратов: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020;
3. Поспелов, Е.А., Попов, И.С. Пакеты прикладных программ в научных исследованиях: учебно-методическое пособие. - Омск: Омский государственный университет им. Ф.М. Достоевского, 2019.

Дополнительная литература:

1. Агалаков, С.А. Анализ данных в среде R: практикум. - Омск: Омский государственный университет им. Ф.М. Достоевского, 2020;

2. Воскобойников, Ю.Е. Регрессионный анализ данных в пакете MATHCAD. - Санкт-Петербург: Лань, 2021.

Интернет-ресурсы

1. <http://biblioclub.ru> – ЭБС «Университетская библиотека on-line», <http://old.exponenta.ru/> – Образовательный математический сайт по математике и программированию
2. <http://MATLAB.exponenta.ru/Simulink/default.php> – сайт, посвященный интерактивному инструменту для моделирования Simulink,
3. <https://www.intuit.ru/> – Национальный Открытый Университет «ИНТУИТ»
4. <http://window.edu.ru/> – Информационная система «Единое окно доступа к образовательным ресурсам»
5. <http://artspb.com/> – Образовательный портал: математика, кибернетика и программирование дополнительного образования
6. <http://www.arenasimulation.com> – Официальный сайт разработчика среды имитационного моделирования ARENA
<https://www.anylogic.ru> – Официальный сайт разработчика AnyLogic

Лабораторная работа 5. Средства создания баз данных. Графика, анимация

Цель лабораторной работы:

формирование у студентов способностей создания баз данных, работы с графикой и анимацией.

Организационная форма занятий: решение проблемных задач, разбор конкретных ситуаций

Вопросы для обсуждения на лабораторном занятии: Средства создания баз данных. Графика, анимация в среде MATLAB.

Формируемые компетенции: ОПК-2, ОПК-3, ОПК-4, ОПКД-2, ОПКД-3, ОПКД-4.

Основные сведения

Одним из типов данных, используемых системой MATLAB, являются структуры. Каждая структура состоит из ряда полей, имеющих имена. Поля могут содержать данные любого типа - от пустого поля [] до массивов. MATLAB поддерживает и массивы структур, что позволяет создавать мощные базы данных.

Для задания структуры используется оператор присваивания, например:

```
>> man.name='Петр';  
>> man.fam='Иванов';  
>> man.dr=1976;  
>> man.mr='Казань';
```

Чтобы создать массив структур, надо ввести индексацию имени структуры, например:

```
>> man(1).fam='Иванов';  
>> man(3).dr=1972;
```

Количество структур в массиве можно определить с помощью функции length(имя):

```
>> length(man)  
ans=3
```

Для просмотра структуры или отдельного поля следует указать соответствующее имя:

```
>> man(3).dr  
ans=1972
```

Структуру можно создать также с помощью функции **struct**('field1',value1,...), например:

```
>> A=struct('gorod','Казань','ulica','Московская','dom',12,'kvart',52)  
A=  
gorod: 'Казань'  
ulica: 'Московская'  
dom: 12  
kvart: 52
```

С помощью функции deal возможно множественное присваивание входных данных:

```
>> [A,B,C,...]=deal(X,Y,Z,...).
```

Проверка имен полей и структур осуществляется с помощью функций isfield(S,'field') и isstruct(S).

Для работы с полями структуры можно использовать функции:

```
>> fieldname(S) – возвращает имена полей заданной структуры  
>> getfield(S,'field') – возвращает содержимое заданного поля  
>> setfield(S,'field',V) – присваивает заданному полю значение переменной V  
>> rmfield(S,'field') – удаляет заданное поле.
```

Наиболее сложный тип данных в системе MATLAB – массив ячеек. Это массив, элементами которого являются ячейки, содержащие любые типы массивов, включая массивы ячеек. Содержимое ячеек задается в фигурных скобках {}.

Существует два способа присваивания данных отдельным ячейкам – индексация ячеек или индексация содержимого. Для создания массива ячеек можно использовать также функцию:

```
>> cell(N) - создание массива ячеек из NxN пустых матриц  
>> cell(size(A)) - создание массива ячеек из пустых матриц того же размера, что A.
```

Для отображения массива ячеек C служит команда celldisp(C).

Для более наглядного графического представления можно использовать команду:

```
>> cellplot(C,'legend').
```

Для создания строкового массива ячеек может использоваться функция `cellstr(S)`, например:

```
>> S={'Я''живу''в Казани'}  
>> C=cellstr(S)  
C='Я' 'живу' 'в Казани'.
```

Графика, анимация

Система MATLAB обладает большими возможностями для графической визуализации вычислений – от построения простых графиков функций до трехмерных комбинированных и презентационных графиков с элементами анимации, а также средствами проектирования графического интерфейса (GUI).

Для **работы с графическим окном** можно воспользоваться командами:

`figure(i)` – делает активным i -е окно при $i \leq k$ (если открыто k окон) или открывает окно с номером i при $i > k$

`clf` – очищает текущее графическое окно (по умолчанию окно очищается при построении очередного графика)

`hold on` – запрещает очистку окна до подачи команды `hold off`

`zoom on` – включает режим лупы (`zoom(N)` изменяет масштаб в N раз)

`whitebg` – открывает графическое окно, меняя цвет фона на противоположный

`subplot(m,n,p)` – разбивает окно на $m \times n$ подокон, p – номер текущего подокна.

Выдачу содержимого графического окна на печать можно осуществить с помощью опции `Print` в меню `File`. Копирование содержимого текущего графического окна осуществляется опцией `Copy Figure` в меню `Edit`.

Для **построения графика функции одной переменной в декартовых координатах** служит команда

`plot(X,Y)` – строит график функции $Y(X)$, где Y и X – вектора одинаковой длины.

Если X и Y – матрицы, то строится семейство графиков по данным, содержащимся в колонках матриц. Команда `plot(Y)` строит график $Y(i)$, где i – индекс элемента в векторе Y , команда `plot(X,Y,S)` позволяет задать тип и цвет линии графика с помощью строковой константы S (Y – желтый, M – фиолетовый, C – голубой, R – красный, G – зеленый, B – синий, W – белый, K – черный, – сплошная линия, `-.` – штрихпунктир, `:` – двойной пунктир, `_` – штриховая линия, `.` – линия строится точками, `*` – звездочками, `+` – плюсами, `O` – окружностями, `S` – квадратами, `X` – крестами, `V` – треугольниками вниз, `^` – треугольниками вверх, `D` – ромбами, `<` – треугольниками влево, `>` – треугольниками вправо, `P` – пятиугольниками, `H` – шестиугольниками). Для построения нескольких линий на одном графике возможна подача команды `plot(X1,Y1,S1,X2,Y2,S2,...)`, например:

```
>> x=-2*pi:0.1*pi:2*pi;  
>> y1=cos(x);  
>> y2=cos(x).^2;  
>> y3=cos(x).^3;  
>> plot(x,y1,'-Y',x,y2,'-+R',x,y3,'_OK')
```

Команда `plot(Y,X)` позволяет построить график обратной функции, а `plot(X(t),Y(t))` – график функции, заданной параметрически.

Разметкой осей и линиями сетки можно управлять с помощью команд:

`axis off` – убирает с осей их обозначения и маркеры

`axis on` – восстанавливает введенные ранее обозначения

`axis([xmin xmax ymin ymax])` – устанавливает диапазон координат по осям

`axis auto` – устанавливает параметры осей по умолчанию

`axis equal` – устанавливает параметры осей с одинаковым расстоянием между метками

`axis manual` – «замораживает» масштаб осей в текущем состоянии

`grid on` – добавляет сетку к текущему графику

`grid off` – отключает сетку.

Для **текстовых сообщений** на графике служат команды:

`title('string')` – титульная надпись

`xlabel('string')` или `ylabel('string')` – названия осей

`text(X,Y,'string')` – текст, начало которого находится в точке (X,Y)

`gtext('string')` – позиционирование текста с помощью мышки

legend(string1,string2,...,pos) – легенда, размещенная в зависимости от параметра pos (0 – лучшее место, 1 – верхний правый угол, 2 – верхний левый угол, 3 – нижний левый угол, 4 – нижний правый угол, -1 – справа от графика).

Для **построения графиков в полярной системе координат** служит команда polar(theta,ro) или polar(theta,ro,S) с синтаксисом, аналогичным plot, например:

```
>> x=1:.01:2;  
>> nx=length(x);  
>> ro=x.^3;  
>> fi=linespace(0,3*pi,nx);  
>> polar(fi,ro)
```

Команды loglog(...), semilogx(...) и semilogy(...) позволяют построить графики функций **в логарифмическом и полулогарифмическом масштабах**.

Для **построения диаграмм** служат команды:

bar(Y) – столбцовый график элементов вектора Y (или bar(X,Y), где положение столбцов задается вектором X, или bar(X,Y,width,S) с заданной шириной столбцов, параметр S аналогичен рассмотренному ранее параметру для команды plot);

hist(...) – гистограмма, параметры те же, что у команды bar;

stairs(...) – лестничный график, параметры те же, что у команды bar;

rose(theta,N) – угловая гистограмма, N – число интервалов от 0 до 2π (по умолчанию 20).

MATLAB обладает большим набором средств **трехмерной графики**. Прежде всего, это различные средства **построения поверхностей**, описываемых функцией двух переменных

например:

meshgrid(X,Y) – задание опорной области для построения трехмерной поверхности (команда meshgrid(X,X) эквивалентна команде meshgrid(X));

plot3(X,Y,Z) – трехмерный аналог команды plot, где X, Y, Z – матрицы одинакового размера (возможен вариант с дополнительным параметром S или plot3(X1,Y1,Z1,...)), команда строит трехмерные точки и соединяет их отрезками прямых.

Пример:

% параболоид, изображенный методом сеток

```
>> [X,Y]=meshgrid([-2:1:2]);
```

```
>> Z=X.^2+Y.^2;
```

```
>> plot3(X,Y,Z,'r',Y,X,Z,'-r')
```

Более наглядное изображение обеспечивают команды:

mesh(X,Y,Z,C) – сетчатое изображение поверхности (C – цвет узловых точек, по умолчанию C=Z, т.е. цвет линий зависит от их высоты);

meshc(...) – изображение поверхности и ее проекции в виде линий равного уровня;

meshz(...) – построение поверхности столбцами;

surf(X,Y,Z,C) – цветная параметрическая поверхность, цвет задается массивом C;

surfc(...) – построение поверхности и ее проекции;

surf1(...) – построение поверхности с подсветкой от источника света, возможен параметр S – вектор координат источника (S=[Sx,Sy,Sz] в декартовых координатах и S=[AZ,EL] в сферических координатах), можно также задать цвет подсветки дополнительным параметром 'light'.

Окрашку поверхности и различные эффекты можно задать с помощью функций:

colormap(S) – задает окраску тонами указанного цвета, например colormap('gray');

shading interp – устраняет изображение сетки и задает интерполяцию для оттенков цвета;

colorbar – выдает на экран шкалу оттенков с соответствующими координатами;

diffuse – задает эффект диффузионного рассеяния;

lighting – управляет подсветкой;

material – имитирует свойства рассеивания света различными материалами;

specular – задает эффект зеркального отражения;

view – задает положение точки просмотра;

rotate3d – задает поворот трехмерной фигуры.

Существуют специальные команды для построения цилиндра и сферы, а также для построения объемных фигур с помощью плоских треугольников:

[X,Y,Z]=cylinder(R,N) – создает массивы для дальнейшего построения с помощью функции surf цилиндрической поверхности с радиусом R и числом узловых точек N;

[X,Y,Z]=sphere(N) – генерирует матрицы размера (N+1)*(N+1) для последующего построения с помощью команды surf;

trimesh(TRI,X,Y,Z,C) – строит объемную каркасную фигуру с треугольниками, заданными матрицей поверхности TRI, каждая строка которой содержит три элемента и определяет треугольную грань путем указания индексов, по которым выбираются координаты из векторов X,Y,Z;

trisurf(...) – аналогична предыдущей команде, но строит закрашенные треугольники.

MATLAB обладает большим набором функций, обеспечивающих **анимацию**, например:

comet(X,Y) – отображает движение точки по заданной векторами X,Y траектории;

comet3(X,Y,Z) – движение трехмерной точки;

capture – захват с экрана текущего графического окна;

getframe – формирование вектор-столбца, определяющего один кадр для анимации;

m=moviein(N) – формирование матрицы, рассчитанной на N анимационных кадров;

movie(m) – отображение кадров, содержащихся в матрице m;

rotate – вращение фигуры вокруг заданной точки в заданном направлении;

frame2im – преобразование кадра в графический образ;

im2frame – преобразование графического образа в кадр.

Центральным понятием **дескрипторной** (handle) графики, на которой базируются графические средства MATLAB, является графический объект. Используются следующие типы объектов: root (первичный – экран компьютера), figure (объект создания графического окна), uicontrol (создание пользовательского интерфейса), axis (объект, задающий область расположения графика), uimenu (объект создания меню), uicontextmenu (создание контекстного меню), image (создание растровой графики), line (создание линии), path (создание закрашенных фигур), rectangle (закрашенный прямоугольник), surface (поверхность), text (текст), light (свет). Пример создания объекта:

```
% построение отрезка прямой
```

```
>> h=line([0 2 5],[1 4 -1], 'color', 'blue');
```

К объектам применяются операции:

get(h) – вывод свойств графического объекта;

set(h,'Name',Value,...) – установка свойств объекта, например: set(h,'color','red');

reset(h) – восстановление свойства объекта;

delete(h) – удаление объекта;

drawnow – выполнение очереди задержанных графических команд;

findobj – поиск объекта с заданными свойствами;

copyobj – копирование объекта с заданными свойствами и др.

К свойствам, задающим вид изображения, относятся, например:

Clipping – отсечение по границам осей;

SelectionHighlight – подсветка с выводом дескрипторов (описателей);

Visible – видимость (значение on (по умолчанию) или off);

EraseMode – свойство для создания анимации (normal (по умолчанию), none (не стирает изображение при перемещении или изменении), xor (стирание изображения с использованием исключающего «или»), background (стирание изображения путем прорисовывания цветом фона).

Получение изображения из графического объекта и вывод объекта на экран можно осуществить с помощью функций getimage(h) и image(C), где C – матрица, элементы которой точно указывают цвет и прямоугольный сегмент изображения.

Основные средства по обработке изображений входят в пакет Image Processing Toolbox.

Выполнить задание

5.1. Создать структуру **exam**, состоящую из пяти полей: fi (фамилия имя),
ng (номер группы), phys (оценка по физике), chem (оценка по химии),
math (оценка по математике).

```

>> exam.fi='Петров Петр';
>> exam.ng='921';
>> exam.phys= 5;
>> exam.chem= 3;
>> exam.math= 2;

```

5.2 К созданной в примере 8.1. структуре добавить еще две записи.

```

>> exam(2).fi='Иванов Иван';
>> exam(2).ng='922';
>> exam(2).phys= 4;
>> exam(2).chem= 3;
>> exam(2).math= 3;
>> exam(3).fi='Сидоров Сергей';
>> exam(3).ng='921';
>> exam(3).phys= 3;
>> exam(3).chem= 4;
>> exam(3).math= 5;

```

5.3 Подсчитать среднюю оценку одного из студентов.

```

>> k=input('Номер студента в базе: ');
Номер студента в базе: 3
>> s1=exam(k).phys; s2=exam(k).chem; s3=exam(k).math;
>> sred=num2str((s1+s2+s3)/3); %вычисление среднего и перевод в...
                                символный вид
>> f=exam(k).fi; %значение поля fi для записи с номером k
>> g=getfield(exam(k),'n' %значение поля ng для записи с номером k
>> string=[f,' ',sred];
>> disp(string)
Сидоров Сергей: 4.0000

```

5.4. Упорядочить поле fi структуры **exam** по алфавиту.

```

>> s=exam.fi;
>> s1=orderfields(s);
>> exam.fi=s1
ans = Иванов Иван
      Петров Петр
      Сидоров Сергей

```

5.5. Разбить текущее графическое окно на четыре подокна и в каждом из них построить график: 1 – два графика функции одной переменной в одном графическом пространстве, 2 – график кусочно - заданной функции, 3 – график поверхности, 4 – график с эффектом анимации.

1. >> subplot(2,2,1) %открытие первого подокна (всего подокон...
четыре - две строки и два столбца)

```

>> x=-pi:0.01:2*pi;
>> f=exp(-0.1*x).*sin(x).^2;
>> y=-2*pi:0.01:pi;
>> g=exp(-0.2*y).*sin(y).^2;
>> plot(x,f,'r',x,g,'k-')

```

```

2. >> subplot(2,2,2);
>> x1=-2*pi:pi/30:-pi;
>> y1=pi*sin(x1);
>> x2=-pi:pi/30:pi;
>> y2=pi-abs(x2);
>> x3=pi:pi/30:2*pi;
>> y3=pi*sin(x1).^3;
>> x=[x1 x2 x3];

```

```

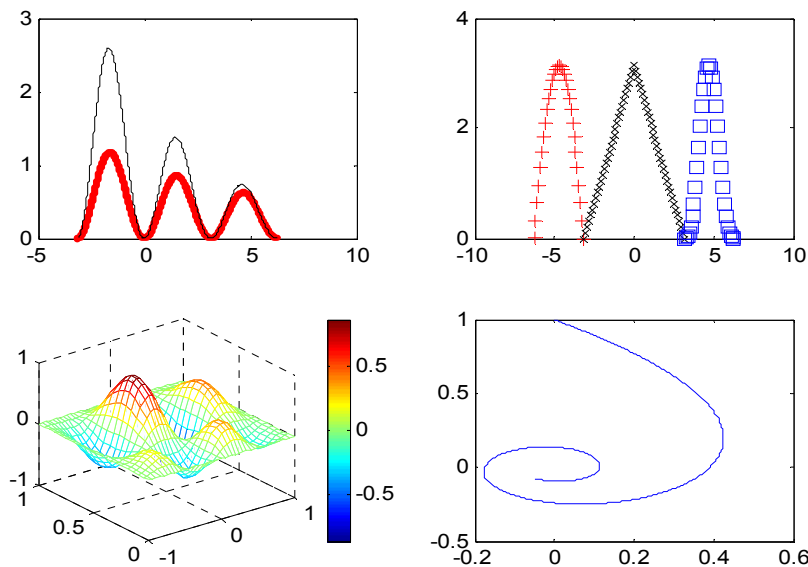
>> y=[y1 y2 y3];
>> plot(x1,y1,'r+',x2,y2,'kx',x3,y3,'bs')

3. >> subplot(2,2,3);
>> [X,Y]=meshgrid(-1:0.05:1,0:0.05:1);           %проекции узловых точек...
                                                    поверхности

>> Z=4*sin(2*pi*X).*cos(1.5*pi*Y).*(1-X.^2).*Y.*(1-Y);
>> mesh(X,Y,Z);
>> colorbar                                         %шкала цветов

4. >> subplot(2,2,4);
>> t=[0:0.001:10];
>> x=sin(t)./(t+1);
>> y=cos(t)./(t+1);
>> comet(x,y);                                     %эффект кометы

```

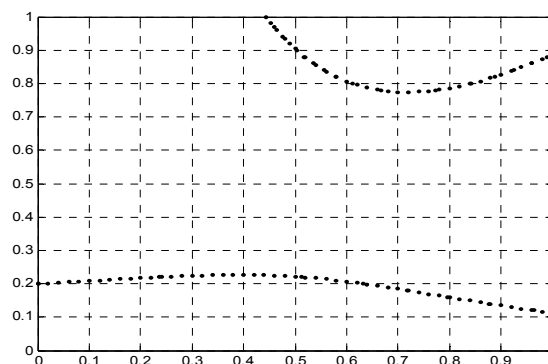


5.6. Построить график функции, заданной неявно: $x^3y - 2xy^2 + y - 0.2 = 0$ в квадрате $|x| \leq 1, |y| \leq 1$ (см. [2]).

```

>> h=0.02; x=0:h:1;
>> [X,Y]=meshgrid(x);
>> f=X.^3.*Y-2*X.*Y.^2.+Y-.2;
>> v=[0 0]; contour(x,x,f,v); grid               %линия уровня поверхности...
                                                    z=f(x,y) при z=0

```



5.7. Построить график поверхности $z = f(x,y)$ ($f(x,y)$ – левая часть уравнения из примера 9.2) с нанесенным на нем рисунком из файла uzog.jpg. В соседнем подокне изобразить исходный (заранее подготовленный) рисунок.

```

>> h=0.02; x=0:h:1;
>> [X,Y]=meshgrid(x);
>> Z=X.^3.*Y-2*X.*Y.^2.+Y-.2;

```

```

>> A=imread('uzor','jpg');    %считывание рисунка из файла, занесение...
                                данных в графический объект A
>> subplot(1,2,1);
>> image(A)                    %вывод рисунка
>> axes off
>> subplot(1,2,2);
>> surf(X,Y,Z,'CData',A,'FaceColor','texture')    %график поверхности
>> axes off

```

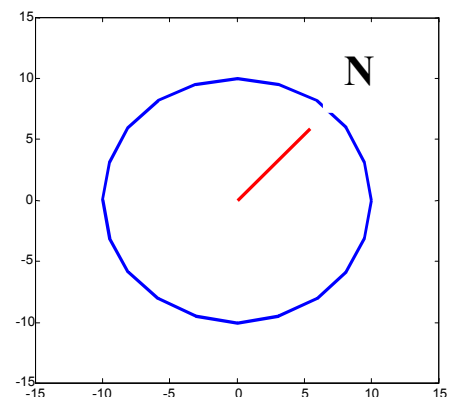
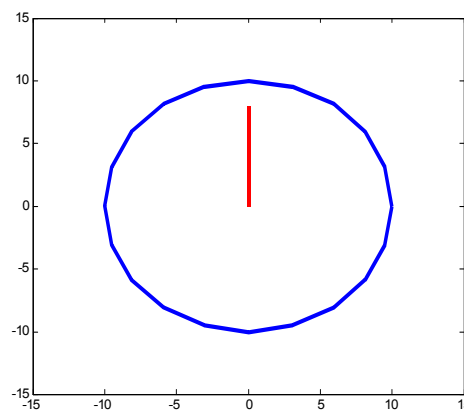


5.8. Изобразить движение стрелки компаса при изменении положения точки N (север). Точку N вывести в диалоговом режиме с помощью функции `gtext()`.

```

>> f=figure('Name','Kompas');    %открытие графического окна
>> %рисование окружности
>> plot(10*cos(0:pi/10:2*pi),10*sin(0:pi/10:2*pi),'LineWidth',3);
>> hold on                        %режим добавления графиков
>> axis([-15,15,-15,15]);        %границы осей
>> x1=[0 0]; y1=[0 8];          %координаты стрелки
>> h=plot(x1,y1,'LineWidth',3,'Color','r','EraseMode','xor');    %стрелка
>> a=gtext('N');                %ввести точку N
>> b=get(a);                    %вывод графических характеристик объекта a
>> v=b.Position;
>> xn=v(1); yn=v(2);            %координаты точки N
>> R=sqrt(xn^2+yn^2);            %вычисление расстояния до точки N
>> alfa=asin(xn/R);              %угол поворота
>> xh=x1*cos(alfa)-y1*sin(alfa); %преобразование координат
>> yh=x1*sin(alfa)+y1*cos(alfa);
>> set(h,'XData',xh,'YData',yh) %новая стрелка

```



Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.
Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. Опишите средства создания баз данных в среде MATLAB.
2. Поясните о возможностях среды MATLAB для работы с графикой.
3. Поясните о возможностях среды MATLAB для работы с анимацией.

Список литературы, рекомендуемый к использованию по данной теме

Основная литература:

1. Голубева, Н.В. Математическое моделирование систем и процессов. - Санкт-Петербург: Лань, 2021;
2. Казиев, В.М. Введение в анализ, синтез и моделирование систем: Учебное пособие. - Москва, Саратов: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020;
3. Пospelов, Е.А., Попов, И.С. Пакеты прикладных программ в научных исследованиях: учебно-методическое пособие. - Омск: Омский государственный университет им. Ф.М. Достоевского, 2019.

Дополнительная литература:

1. Агалаков, С.А. Анализ данных в среде R: практикум. - Омск: Омский государственный университет им. Ф.М. Достоевского, 2020;
2. Воскобойников, Ю.Е. Регрессионный анализ данных в пакете MATHCAD. - Санкт-Петербург: Лань, 2021.

Интернет-ресурсы

1. <http://biblioclub.ru> – ЭБС «Университетская библиотека on-line», <http://old.exponenta.ru/> – Образовательный математический сайт по математике и программированию
2. <http://MATLAB.exponenta.ru/Simulink/default.php> – сайт, посвященный интерактивному инструменту для моделирования Simulink,
3. <https://www.intuit.ru/> – Национальный Открытый Университет «ИНТУИТ»
4. <http://window.edu.ru/> – Информационная система «Единое окно доступа к образовательным ресурсам»
5. <http://artspb.com/> – Образовательный портал: математика, кибернетика и программирование дополнительного образования
6. <http://www.arenasimulation.com> – Официальный сайт разработчика среды имитационного моделирования ARENA
<https://www.anylogic.ru> – Официальный сайт разработчика AnyLogic

Лабораторная работа 6. Графический интерфейс пользователя (GUI)

Цель лабораторной работы:

формирование у студентов способностей работы с графическим пользователем (GUI).

Организационная форма занятий: решение проблемных задач, разбор конкретных ситуаций

Вопросы для обсуждения на лабораторном занятии: Графический интерфейс пользователя (GUI) системы MATLAB.

Формируемые компетенции: ОПК-2, ОПК-3, ОПК-4, ОПКД-2, ОПКД-3, ОПКД-4.

Основные сведения

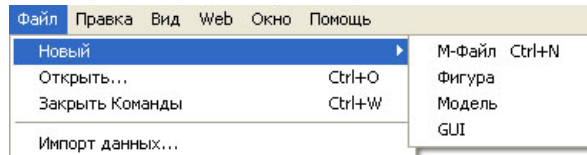
Дескрипторная графика MATLAB позволяет конструировать детали графического пользовательского интерфейса (Graphic User Interface) для создания собственных приложений. При этом различные функции и m-файлы вызываются из графического окна стандартного вида, но элементы интерфейса (кнопки, меню, слайдеры, надписи и пр.) задаются пользователем. Для этой цели используются функции:

- `uicontrol` – управление пользовательским интерфейсом;
- `uimenu` – создание меню пользователя;
- `ginput` – графический ввод с помощью мыши (работа команды заканчивается нажатием клавиши Enter или устанавливается число обращений `ginput(n)`);
- `dragrect` – создание и перетаскивание прямоугольника с помощью мыши;
- `rbbox` – растягивание прямоугольника мышью;
- `selectmoveresize` – выделение, копирование, перемещение и изменение размеров объекта;
- `waitforbuttonpress` – задержка выполнения программы до нажатия кнопки мыши;
- `waitfor` – блокировка исполнения и ожидание события;
- `uiwait` – прекращение выполнения программы до команды `uiresume` или закрытия окна;
- `uiresume` – возобновление работы программы;
- `uisuspend` – запрет интерактивного состояния графического окна;
- `uirestore` – восстановление интерактивного состояния;
- `guide` – создание интерфейса в интерактивном режиме;
- `align` – выравнивание объектов интерфейса;
- `cbedit` – редактирование объектов интерфейса в интерактивном режиме;
- `menuedit` – изменение меню;
- `propedit` – изменение свойств объекта;
- `dialog` – создание диалогового окна;
- `axlimdlg` – вызов окна для изменения координатных осей графического окна;
- `errordlg` – создание окна с сообщением об ошибке;
- `helpdlg` – создание справочного окна;
- `inputdlg` – создание окна для ввода;
- `listdlg` – создание окна для выбора вариантов значений параметра из списка;
- `menu` – создание меню диалогового ввода (`choice=menu(header, item1, item2, ...)`);
- `msgbox` – создание окна сообщений;
- `questdlg` – создание окна запроса;
- `warnldg` – создание окна предупреждения;
- `uigetfile` – создание окна открытия файла;
- `uiputfile` – создание окна сохранения файла;
- `uisetcolor` – создание окна выбора цвета;
- `uisetfont` – создание окна выбора шрифта;
- `pagedlg` – создание окна параметров страницы;
- `printdlg` – создание окна для вывода на печать;
- `waitbar` – создание “панели ожидания”;
- `makemenu` – создание структуры меню;
- `menubar` – установка типовых свойств для объекта `MenuBar`;

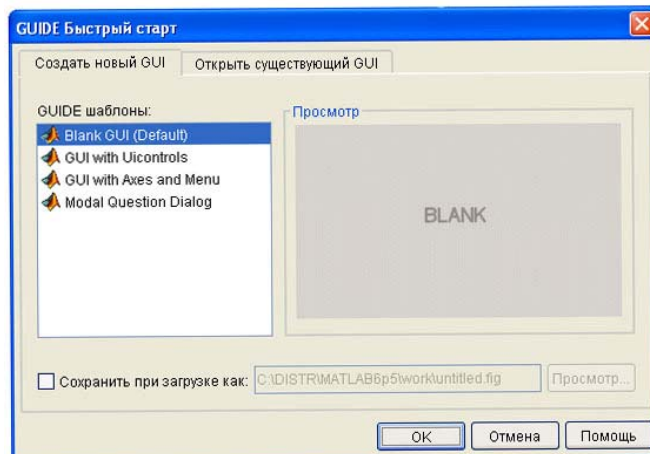
- btngroup - создание инструментальной панели с группой кнопок;
- btnstate – запрос статуса кнопки;
- btnpress – управление кнопкой;
- btndown – нажатие кнопки;
- btnup – “поднятие” кнопки.

Полный список функций GUI можно изучить с помощью команды `help uitools`.

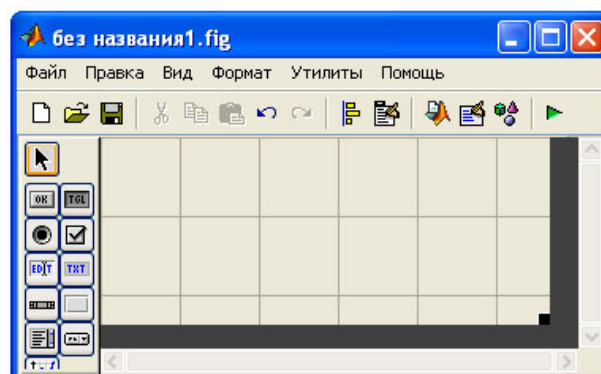
При создании GUI удобно пользоваться встроенным редактором, который вызывается из основного меню:



После выбора пункта GUI появляется окно:



Кнопка **ok** открывает конструктор GUI:



Далее можно выбрать объекты интерфейса: Static Text – статический текст, Edit Text – редактируемый текст, Popupmenu – всплывающее меню, Checkbox – флажок, Axes – график, Push Button – командная кнопка и др.

Свойства выбранных объектов можно изменить с помощью Инспектора, который появляется после двойного щелчка мышью по объекту.

Созданный интерфейс следует сохранить в файле с расширением `.fig`. При этом автоматически создается `m`-файл (программа-функция), который можно отредактировать с

помощью кнопки **f** из основного меню. В частности, указав курсором в открывшемся меню на имя нужного объекта, можно изменить его программный код.

Сообщения об ошибках, справочная система MATLAB

При ошибочном написании математических выражений или команд MATLAB выдает соответствующие сообщения об ошибке или предупреждения, например:

```
??? Undefined function or variable 'sqr'  
Warning: Divide by Zero
```

Предупреждения не останавливают вычисления, а лишь предупреждают о том, что ошибка способна повлиять на ход вычислений, при этом результат иногда выдается в виде сокращения NaN, которое означает неопределенность (например, вида 0/0 или Inf/Inf). Сообщение об ошибке (после ???) останавливает вычисления.

Для исправления ошибки можно нажать клавишу Tab, предварительно поставив курсор в конец исправляемого выражения. Система введет подсказку, анализируя введенные символы. Можно также воспользоваться справочной системой MATLAB.

Основной доступ к справочной информации обеспечивает меню Help, окно которого имеет 5 вкладок:

Contents – поиск информации по оглавлению;

Index – поиск информации по алфавитному каталогу;

Search – поиск информации по заданной справке или отдельному слову;

Demos – доступ к демонстрационным примерам;

Favorites – доступ к специальным возможностям справочной системы (например, печать справочной документации).

Недостатком справочной системы является ее громоздкость (система поставляется на трех CD-ROM). Во многом она дублируется другими справочными подсистемами, например, справками, вызываемыми из командной строки и имеющимися в виде PDF-файлов.

Команда help (без параметров) вызывает список разделов интерактивной справки (список папок, содержащих m-файлы с определениями операторов, функций и др. объектов). Для получения справки по конкретному объекту служат команды:

help имя (в качестве имени может быть константа, функция, оператор и пр.)

doc имя (для вывода более полной информации)

Можно получить справку по группе объектов, например, команда:

help timefun

выдаст информацию об имеющихся в MATLAB функциях времени и дат.

Важное значение имеет поиск m-функций по ключевым словам. Для этого служит команда:

lookfor 'ключевые слова'

она осуществляет поиск всех файлов, в заголовках которых встречаются данные ключевые слова (или слово).

Система MATLAB содержит множество демонстрационных примеров (практически на каждый оператор или функцию), изучение которых позволяет ближе познакомиться с системой. Список примеров можно получить, исполнив команду:

help demos

Просмотреть демонстрационные примеры можно также, подав команду:

demo

Важно, что демонстрационные примеры сопровождаются соответствующими листингами программ-файлов, реализующих поставленную задачу.

Полезными также могут оказаться команды:

helpwin (открывает окно справок Matlab Help Window), helpdesk (открывает в отдельном окне доступ к документации о командах системы), ver (выводит справку о версиях

MATLAB и приложений), whatsnew (выводит информацию о новом в указанном разделе) и readme.

Интерфейс системы MATLAB

Наиболее простой и удобный способ работы с системой MATLAB – работа с помощью панели инструментов, при этом основные команды вводятся нажатием левой клавиши мыши на нужную кнопку:

- New M-file – выводит окно редактора m-файлов;
- Open file – открывает окно для загрузки m-файла;
- Cut – вырезает в буфер выделенный фрагмент;
- Copy – копирует в буфер выделенный фрагмент;
- Paste – переносит фрагмент из буфера в текущую строку;
- Undo – отменяет последнюю операцию;
- Redo – восстанавливает последнюю операцию;
- Simulink – открывает браузер библиотек Simulink;
- Help – открывает окно справки.

Можно также использовать контекстное меню, появляющееся при нажатии правой кнопки мыши на выделенный фрагмент и отображающее доступные в данный момент команды.

В левой части окна системы имеется браузер рабочей области - Workspace Browser, который дает наглядную визуализацию рабочей области, позволяет редактировать содержимое находящихся в памяти объектов и удалять их. Для работы с конкретным объектом необходимо открыть его двойным щелчком по имени в списке.

Просмотр файловой структуры осуществляется специальным браузером (Path Browser), для запуска которого используется окно Current Directory (Текущая папка).

Основное меню последних версий системы MATLAB 6.* содержит шесть пунктов:

- File
- Edit
- View
- Web
- Window
- Help.

Меню **File** содержит команды для работы с файлами: New, M-file (открывает окно редактора/отладчика m-файлов, где используются цветовые выделения – синий цвет для ключевых слов, черный для операторов, констант и переменных, зеленый для комментариев (после знака %) и символьных переменных, красный для синтаксических ошибок), Figure (открывает пустое окно графики), Model (открывает окно для создания Simulink-модели), GUI (открывает окно для создания элементов графического интерфейса), Open, Close Command Windows, Import Data, Save Workspace As..., Set Path, Preferences..., Print..., Print Selection..., Exit.

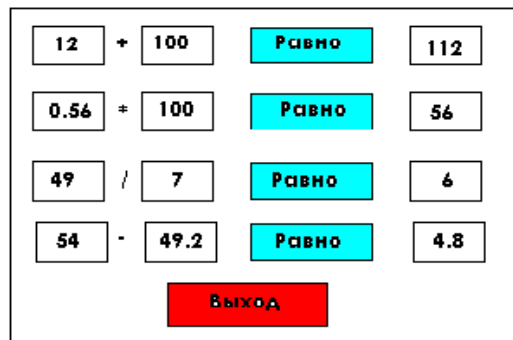
Меню **Edit** содержит команды редактирования, типичные для приложений Windows: Undo, Redo, Cut, Copy, Paste, Clear, Select All, Delete, Clear Command Windows, Clear Command History, Clear Workspace.

Меню **View** позволяет управлять окнами интерфейса. Кроме командного, можно вывести окно command history с дневником сессии, окно рабочей области и др.

В основном меню новейшей версии системы MATLAB 7 появился пункт **Graphics**, содержащий команды New Figure, Plot Tools и More Plots..., удобные для построения графиков. Например, исполнив команду More Plots..., можно, указав имя переменной-массива, задать категорию и тип графика из предложенного каталога, в котором представлены все типы графиков, включая трехмерные и особую handle-графику. Команда Plot Tools открывает окно редактора графики, что позволяет выводить данные по месту установки курсора, вставлять в рисунок панель цветов и легенду, выбирать средства украшения графиков (стрелки, текстовые надписи, окружности, эллипсы, прямоугольники и пр.), задавать титульную надпись, установку для осей, а также выбирать из списка переменных тот массив, по данным которого строится очередной график.

Выполнить задание

6.1. Разработать графическое окно, реализующее функции калькулятора:



Разработка: 1 шаг – создание бланка (File-GUI-Blank)

для ввода цифр – объекты типа Edit (свойство Tag – a, b или rez);

для надписей + - * / – объекты типа Text ;

для кнопок равно и выход – объекты типа Pushbutton (Tag - равно и Close).

2 шаг – сохранение бланка в файле под именем calcullyator.fig.

3 шаг – изменение программы (файл calculyator.m):

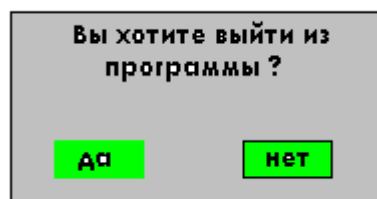
а) изменение кода кнопки равно для + (клавиша f – равно_Callback):

```
function равно_Callback(hObject,evendata,handles)
a=get(handles.a,'String');           %считывание первого слагаемого
a=str2double(a);                     %перевод из символьного вида в числовой
b=get(handles.b,'String');           %считывание второго слагаемого
b=str2double(b);                     %перевод из символьного вида в числовой
rez=a+b;                             %суммирование
set(handles.rez,'String',rez);       %запись результата в объект rez
```

б) изменение кода кнопок равно для * / - :

по аналогии с пунктом а), но вместо a, b и rez соответственно a1, b1, rez1, a2, b2, rez2, a3,b3, rez3.

4 шаг – создание вспомогательного диалогового окна выход (File-GUI-Blank)



для надписи – объект типа Text ;

для кнопок да и нет – объекты типа Push Button (Tag - da и net).

5 шаг – сохранение вспомогательного окна в файле под именем danet.fig.

6 шаг – изменение программы диалогового окна (файл danet.m) :

а) изменение кода кнопки да (клавиша f – da_Callback):

```
function da_Callback(hObject,evendata,handles)
close;                               %закрытие диалогового окна выход
Close_Callback=close;                %выход из программы
```

б) изменение кода кнопки нет (клавиша f – net_Callback):

```
function net_Callback(hObject,evendata,handles)
close;                               %закрытие диалогового окна выход
```

7 шаг – изменение кода кнопки выход: (клавиша f – Close_Callback):

```
function Close_Callback(hObject,evendata,handles)
danet; %вызов функции danet
```

10.2. Разработать диалоговое окно для работы с функцией `payadv()` (пример 8.1).

Разработка: 1 шаг – создание бланка (File-GUI-Blank)

для ввода цифр–объекты типа Edit (свойство Tag – rate, nper, pv, fv, adv);
 для надписей – объекты типа Text ;
 для кнопок вычислить и выход–объекты типа Push Button (Tag-pusk и close)
 для вывода результата – объект типа Text (Tag – pmt).

2 шаг – сохранение бланка в файле под именем primer2.fig.

3 шаг – изменение программы (файл primer2.m):

изменение кода кнопки вычислить (клавиша f – pusk_Callback):

```
function pusk_Callback(hObject,evendata,handles)
rate=get(handles.rate,'String'); %считывание первого аргумента
rate=eval(rate); %вычисление выражения
nper=get(handles.nper,'String'); %считывание второго аргумента
nper=str2double(nper); %перевод из символьного вида в числовой
pv=get(handles.pv,'String'); %считывание третьего аргумента
pv=str2double(pv); %перевод из символьного вида в числовой
fv=get(handles.fv,'String'); %считывание четвертого аргумента
fv=str2double(fv); %перевод из символьного вида в числовой
adv=get(handles.adv,'String'); %считывание пятого аргумента
adv=str2double(adv); %перевод из символьного вида в числовой
pmt=payadv(rate,nper,pv,fv,adv) %вызов функции
set(handles.pmt,'String',pmt); %запись результата в объект pmt
```

4 шаг – работа с диалоговым окном выход (см. пример 10.1).

6.2 Разработать GUI для просмотра нескольких графических окон.

Указание: создать объект `PopupMenu`, перечислить несколько характерных примеров (график функции одной переменной, эффект comet, график поверхности и пр.). Для вывода графического окна создать объект типа `axes`. В программе воспользоваться оператором выбора `switch`.

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;

5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. Как конструировать детали графического пользовательского интерфейса (Graphic User Interface) для создания собственных приложений?
2. Какие используются функции графического пользовательского интерфейса (Graphic User Interface) для создания собственных приложений?

Список литературы, рекомендуемый к использованию по данной теме

Основная литература:

1. Голубева, Н.В. Математическое моделирование систем и процессов. - Санкт-Петербург: Лань, 2021;
2. Казиев, В.М. Введение в анализ, синтез и моделирование систем: Учебное пособие. - Москва, Саратов: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020;
3. Пospelов, Е.А., Попов, И.С. Пакеты прикладных программ в научных исследованиях: учебно-методическое пособие. - Омск: Омский государственный университет им. Ф.М. Достоевского, 2019.

Дополнительная литература:

1. Агалаков, С.А. Анализ данных в среде R: практикум. - Омск: Омский государственный университет им. Ф.М. Достоевского, 2020;
2. Воскобойников, Ю.Е. Регрессионный анализ данных в пакете MATHCAD. - Санкт-Петербург: Лань, 2021.

Интернет-ресурсы

1. <http://biblioclub.ru> – ЭБС «Университетская библиотека on-line», <http://old.exponenta.ru/> – Образовательный математический сайт по математике и программированию
2. <http://MATLAB.exponenta.ru/Simulink/default.php> – сайт, посвященный интерактивному инструменту для моделирования Simulink,
3. <https://www.intuit.ru/> – Национальный Открытый Университет «ИНТУИТ»
4. <http://window.edu.ru/> – Информационная система «Единое окно доступа к образовательным ресурсам»
5. <http://artspb.com/> – Образовательный портал: математика, кибернетика и программирование дополнительного образования
6. <http://www.arenasimulation.com> – Официальный сайт разработчика среды имитационного моделирования ARENA
<https://www.anylogic.ru> – Официальный сайт разработчика AnyLogic

Часть 2. Решение задач в пакете MathCAD

Общие сведения о системе MathCAD

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.

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, при этом точность, с которой отображается результат, задается пользователем, имеет чрезвычайно удобный математико-ориентированный интерфейс и прекрасные средства научной графики.

В состав MathCAD входят несколько интегрированных между собой компонентов:

- мощный текстовый редактор, позволяющий вводить, редактировать и форматировать как текст, так и математические выражения;
- вычислительный процессор, умеющий проводить расчеты по введенным формулам, используя встроенные численные методы;
- символьный процессор, позволяющий проводить аналитические вычисления и являющийся, фактически, системой искусственного интеллекта;
- набор шаблонов, который ускоряет ввод исходных данных и обеспечивает удобную и эффективную систему ввода;
- огромное хранилище справочной информации, как математической, так и инженерной, оформленной в качестве интерактивной электронной книги.

Для освоения основных приемов работы в MathCAD выполняются 5 лабораторных работ:

1. Основы работы с MathCAD;
2. Построение графиков;
3. Вектора и матрицы;
4. Решение уравнений;
5. Символьные вычисления.

Лабораторная работа 7. Основы работы в MathCAD

Цель лабораторной работы:

сформировать у студентов навыки работы с системой MathCAD

Организационная форма занятий: решение проблемных задач, разбор конкретных ситуаций

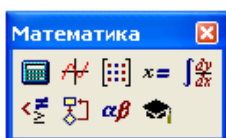
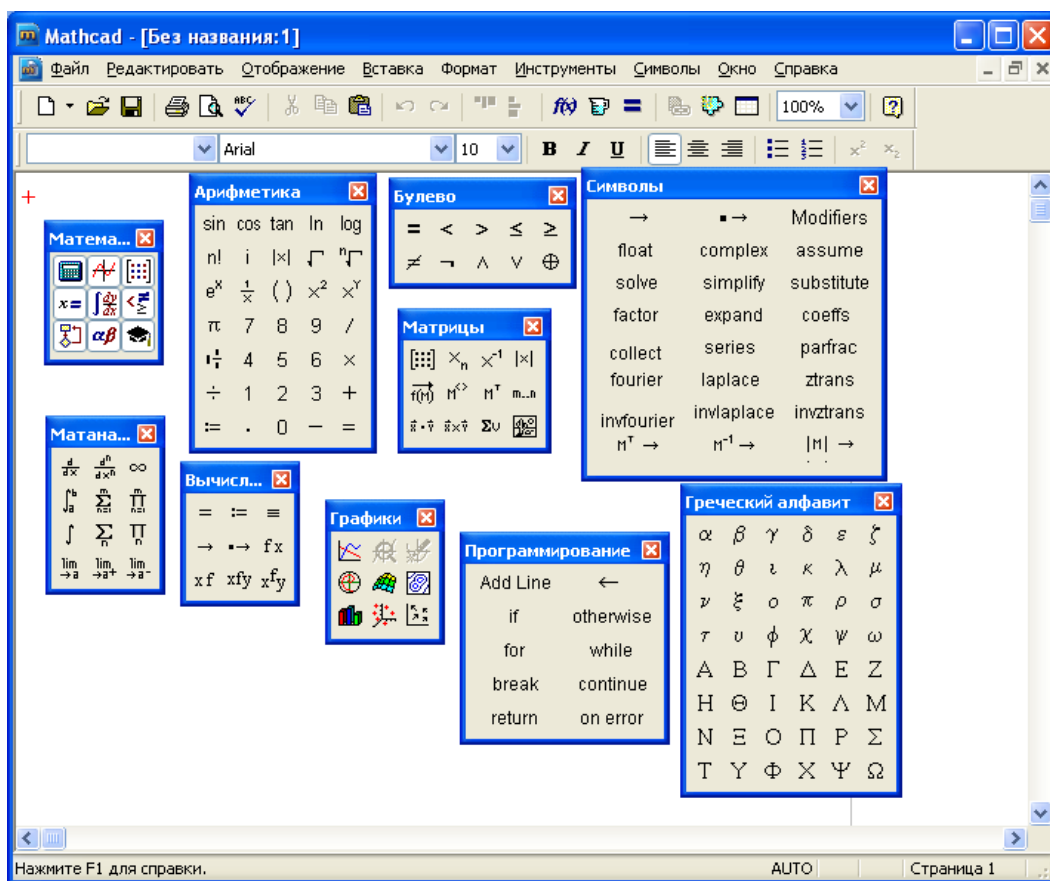
Вопросы для обсуждения на лабораторном занятии: Основы работы в MathCAD.

Формируемые компетенции: ОПК-2, ОПК-3, ОПК-4, ОПКД-2, ОПКД-3, ОПКД-4.

Общие сведения

Основное окно приложения имеет ту же структуру, что и большинство приложений Windows. Сверху вниз располагаются заголовок окна, строка меню, панели инструментов (стандартная и форматирования) и рабочий лист, или рабочая область, документа. Новый документ создается автоматически при запуске MathCAD. Файлы документов в MathCAD имеют расширение **.mcd**.

Большинство команд можно выполнить как с помощью меню (верхнего или контекстного), так и панелей инструментов или клавиатуры.



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

В окне редактирования формируется документ MathCAD. Новый документ получает имя Untitled (Без названия) и порядковый номер. Одновременно открыто может быть до восьми документов.

Документ состоит из трех видов областей: формульных, текстовых и графических. Расположение нетекстовых блоков в документе имеет принципиальное значение. Области просматриваются системой, интерпретируются и исполняются. *Просмотр идет слева направо и сверху вниз.*

Для ввода текстового комментария нужно выполнить команду Text Region (Текстовая область) из пункта меню Insert или нажать клавишу с двойной кавычкой (“”), или нажать на кнопку текста на панели инструментов. Текстовая область служит для размещения текста между формулами и графиками.

При этом в месте ввода появляется курсор в виде вертикального штриха, на место которого вводятся символы текста. Внутри текста курсор перемещается клавишами перемещения курсора. Переход на новую строку производится нажатием на клавишу Enter. Для окончания ввода нужно щелкнуть мышью вне текстовой области.

Для ввода формулы нужно установить указатель мыши в свободном месте окна редактирования и щелкнуть левой кнопкой мыши. Появится визир в виде красного крестика. Он указывает место, с которого начинается набор формулы.

Константы и переменные

Константами называются поименованные объекты, хранящие некоторые значения, которые не могут быть изменены.

В MathCAD применяются десятичные, восьмеричные и шестнадцатеричные числовые константы. Десятичные константы могут быть целочисленными, вещественными, заданными с фиксированной точкой, и вещественными, заданными в виде мантиссы и порядка.

В MathCAD содержится особый вид констант - размерные. Помимо своего числового значения они характеризуются еще и указанием на то, к какой физической величине они относятся. Для этого указания используется символ умножения. В системе MathCAD заданы следующие основные типы физических величин: time (время), length (длина), mass (масса) и charge (заряд). При необходимости их можно изменить на другие.

Переменные являются поименованными объектами, которым присвоено 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.

Имя переменной называется идентификатором. MathCAD различает в идентификаторах символы верхнего и нижнего регистров. Например: ABC и AbC имена разных переменных.

Идентификаторы MathCAD должны начинаться с буквы и могут содержать следующие символы:

- латинские буквы любого регистра;
- арабские цифры от 0 до 9;
- символ подчеркивания (_), символ процент (%) и символ (.);
- буквы греческого алфавита (набираются с использованием клавиши Ctrl или применяется палитра греческих букв).

Определение переменных

Переменные 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. Попытка использовать неопределенную переменную ведет к выводу сообщения об ошибке.

В MathCAD различают: локальные и глобальные переменные.

Локальные переменные вводятся:

Имя_переменной : выражение

На экране:

Имя_переменной := выражение

Глобальные переменные вводятся:

Имя_переменной ~ выражение

На экране:

Имя_переменной $\equiv$ выражение

Если переменной присваивается начальное значение с помощью оператора $:=$, такое присваивание называется локальным. До этого присваивания переменная не определена и ее нельзя использовать. MathCAD читает рабочий документ слева направо и сверху вниз, поэтому определив переменную, ее можно использовать в вычислениях везде правее и ниже равенства, в котором она определена. Однако с помощью знака $\equiv$ (три горизонтальные черточки) можно обеспечить глобальное присваивание, т. е. оно может производиться в любом месте документа. К примеру, если переменной присвоено таким образом значение в самом конце документа, то она будет иметь это же значение и в начале документа.

Например:

Ввод с клавиатуры	Вид на экране
local:137	local := 137 локальное определение переменной local;
global~987.23	global $\equiv$ 987.23 глобальное определение переменной global.

Переменные могут использоваться в математических выражениях, быть аргументами функций или операндом операторов.

Переменные могут быть и размерными, т. е. характеризоваться не только своим значением, но и указанием физической величины, значение которой они хранят. Проведение расчетов с размерными величинами и переменными особенно удобно при решении различных физических задач.

Предопределенные переменные

Предопределенные (системные) переменные – особые переменные, которым изначально системой присвоены начальные значения.

Переменная	Ввод	Назначение	Значение по умолчанию
π	Ctrl + Shift + p	Число π	3.14159
e	e	Основание натурального логарифма	2.718
∞	Ctrl + Shift + z	Системная бесконечность	10^{307}
i или j	1i или 1j	Мнимая единица	
%		Процент	0.01
TOL		Погрешность численных методов	0.001
ORIGIN		Нижняя граница индексации массивов	0

Операторы

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

Операторы, обозначающие основные арифметические действия, вводятся с панели Calculator (Калькулятор, Арифметика).

Вычислительные операторы вставляются в документы при помощи панели инструментов Calculus (Матанализ). При нажатии любой из кнопок в документе появляется символ

соответствующего математического действия, снабженный несколькими местозаполнителями. Количество и расположение местозаполнителей определяется типом оператора и в точности соответствует их общепринятой математической записи.

Результатом действия логических, или булевых, операторов являются только числа 1 (если логическое выражение, записанное с их помощью, истинно) или 0 (если логическое выражение ложно).

Вычислительные операторы сгруппированы на панели Evaluation (Вычисления):

- Численный вывод (Evaluate Numerically) =
- Символьный (аналитический) вывод (Evaluate Symbolically) →
- Присваивание (Definition) :=
- Глобальное присваивание (Global Definition) ≡.

Оператор	Клавиша	Назначение оператора
$X := Y$	$X : Y$	Локальное присваивание X значения Y
$X \equiv Y$	$X \sim Y$	Глобальное присваивание X значения Y
$X =$	$X =$	Вывод значения X
$X + Y$	$X + Y$	Сложение X с Y
$X - Y$	$X - Y$	Вычитание из X значения Y
$X \cdot Y$	$X * Y$	Умножение X на Y
$\frac{X}{Z}$	X / Z	Деление X на Z
$X \div Y$	$Ctrl + /$	Линейное деление
$\frac{b}{a - \frac{c}{d}}$	$Ctrl + Shift + +$	Дробь (смешанный номер)
z^w	$z \wedge w$	Возведение z в степень w
$\sqrt{z}$	$z \sqrt{}$	Вычисление квадратного корня из z
$n!$	$n !$	Вычисление факториала
B_n	$B \lfloor n$	Ввод нижнего индекса n
$A_{n,m}$	$A \lfloor n, m$	Ввод двойного нижнего индекса
$A^{<n>}$	$A Ctrl + 6 n$	Ввод верхнего индекса (для векторов)

Ранжированные (дискретные) переменные

Ранжированная переменная – переменная, которая принимает ряд значений при каждом ее использовании.

Для определения ранжированной переменной общего вида используется выражение:

Имя_переменной := начальное_значение, начальное_значение + шаг .. конечное_значение.

Если шаг равен 1, тогда ранжированную переменную можно задавать следующим образом:

Имя_переменной := начальное_значение.. конечное_значение.

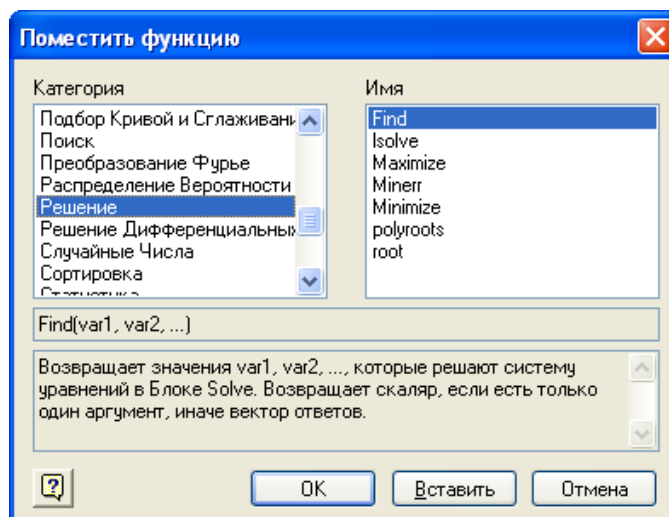
Любое выражение с ранжированными переменными после знака равенства (=) создает таблицу вывода.

Определение функций

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

Функции в пакете MathCAD могут быть встроенные и определенные пользователем.

В MathCAD имеется множество встроенных функций. Для их ввода используется команда меню Вставка → Функция или кнопка на панели инструментов . В диалоговом окне нужно выбрать Категорию и соответствующую функцию.



Функция пользователя вначале должна быть определена, а затем к ней может быть произведено обращение. Функция пользователя определяется следующим образом:

Имя_функции(Переменная1, Переменная2, ...) := Выражение

Задается имя функции, в скобках указывается список аргументов функции - это перечень используемых в выражении переменных, разделяемых запятыми. Затем записывается знак присваивания, справа от которого записывается выражение. Выражение - это любое арифметическое выражение, содержащее доступные системе операторы и функции с операндами и аргументами, указанными в списке аргументов.

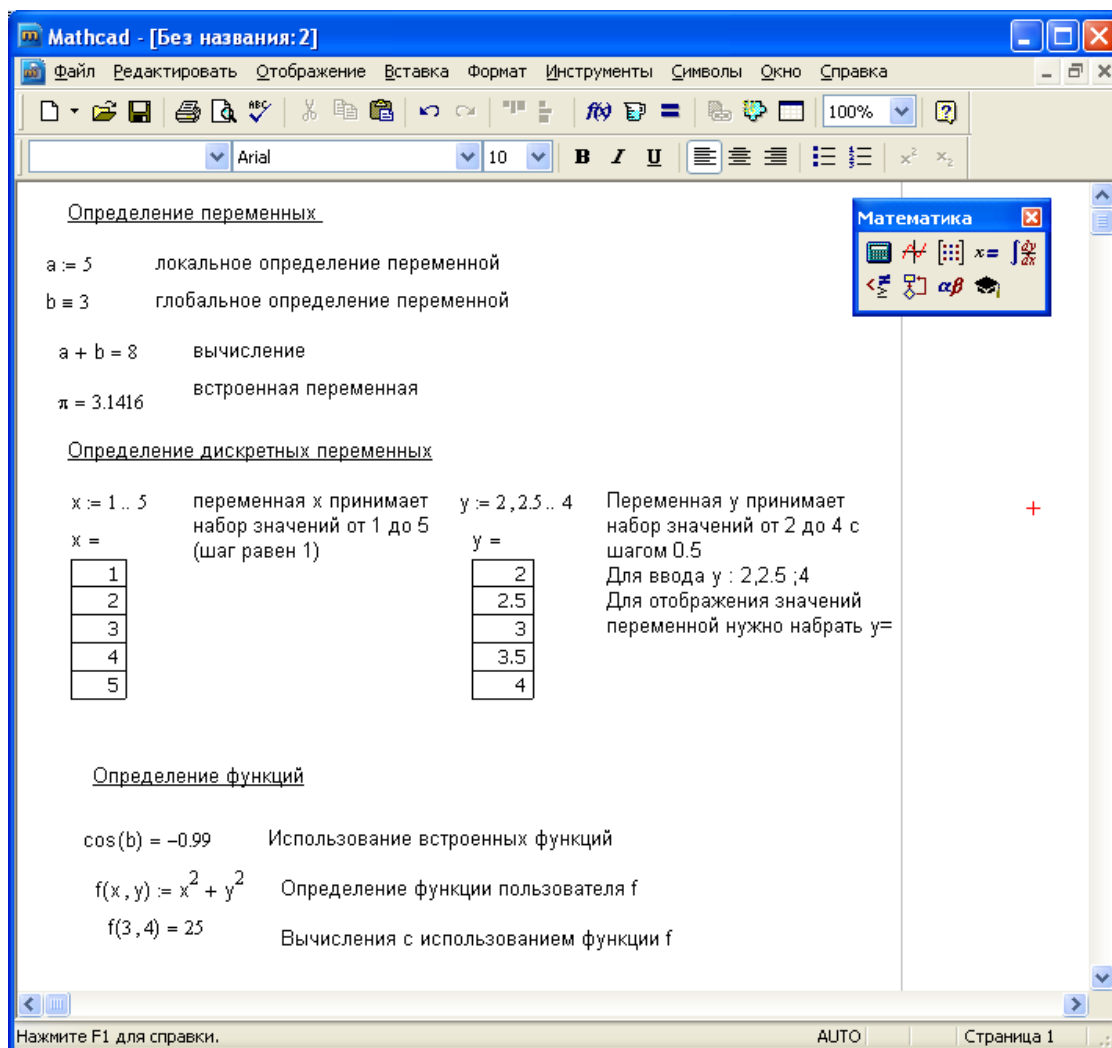
Примеры задания функций одной и двух переменных:

$f(x) := 10 - \exp(x)$

$\text{mult}(x, y) := x * y$

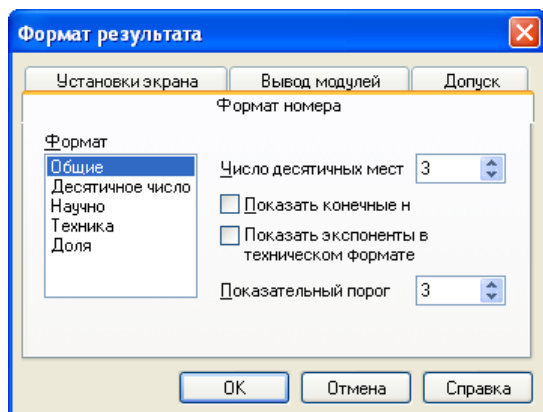
Обращение к функции осуществляется по ее имени с подстановкой на место аргументов констант, переменных, определенных до обращения к функции, и выражений. Например:

$f(3), \sin(1), \text{mult}(2,3).$



Форматирование результатов

Способ, которым MathCAD выводит числа, называется форматом результата. Формат результата может быть установлен для всего документа (глобальный формат) или для отдельного результата (локальный формат).



Глобальный формат устанавливается командой меню Формат→Результат. В диалоговом окне, появляющемся после выбора этой команды, устанавливается выводимая точность числа, диапазон показателя степени (если вывод чисел нужен в форме с плавающей запятой) и точность нуля. После внесения требуемых изменений нужно нажать кнопку ОК.

Для установки формата отдельного числа нужно: щелкнуть мышью на выражении, результат которого нужно переформатировать; вызвать команду форматирования и проделать вышеописанные действия.

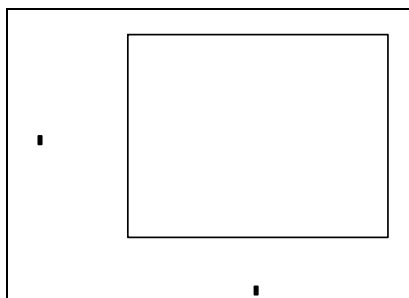
Построение графиков

Для построения графика используется команда меню Вставка→Графики.

Для создания декартового графика:

1. Установить визир в пустом месте рабочего документа;
2. Выбрать команду Вставка ⇒ График ⇒ X-Y график, или нажать комбинацию клавиш Shift

+ @, или щелкнуть кнопку  панели Графики. Появится шаблон декартового графика;



3. Введите в средней метке под осью X первую независимую переменную, через запятую – вторую и так до 10, например: x1, x2, ...;

4. Введите в средней метке слева от вертикальной оси Y первую независимую переменную, через запятую – вторую и т. д., например: y1(x1), y2(x2), ..., или соответствующие выражения;

5. Щелкните за пределами области графика, чтобы начать его построение.

Можно построить несколько зависимостей на одном графике. Для этого нужно ввести соответствующие функции у вертикальной оси (оси ординат). Чтобы разделить описания функций, вводить их нужно через запятую.

Выполнить задание

7.1 Вычислить значение арифметического выражения:

Вариант	Выражение	Вариант	Выражение
1	$1\frac{1}{4} + \frac{1}{9}$	2	$1\frac{1}{7} + 2\frac{1}{5}$
3	$3\frac{3}{4} - \frac{4}{5}$	4	$\frac{5}{7} \div \frac{4}{21}$
5	$\frac{1}{3} \div \frac{5}{12}$	6	$\frac{5}{6} \cdot 2.4$
7	$\frac{4}{5} - 2.5$	8	$3\frac{1}{11} + \frac{1}{3}$
9	$\frac{1}{5} + 2\frac{1}{9}$	10	$5\frac{2}{3} \cdot \frac{9}{17}$
11	$8\frac{1}{2} \cdot \frac{7}{14}$	12	$\frac{1}{5} + 4\frac{1}{2}$
13	$\frac{1}{35} \div \frac{4}{5}$	14	$\frac{5}{9} \cdot 1.8$
15	$\frac{2}{9} \cdot 1.8$	16	$\frac{2}{3} \div \frac{8}{9}$

7.2 Вычислить значение арифметического выражения:

Вариант	Выражение	Вариант	Выражение
---------	-----------	---------	-----------

1	$\frac{\left(13.75+9\frac{1}{6}\right)\cdot 1.2}{\left(10.3-8\frac{1}{2}\right)\cdot \frac{5}{9}}+\frac{\left(6.8-3\frac{3}{5}\right)\cdot 5\frac{5}{6}}{\left(3\frac{2}{3}-3\frac{1}{6}\right)\cdot 56}-27\frac{1}{6}$	2	$\frac{\left(\frac{1}{6}+0.1+\frac{1}{15}\right)\div\left(\frac{1}{6}+0.1-\frac{1}{15}\right)\cdot 2.52}{\left(0.5-\frac{1}{3}+0.25-\frac{1}{5}\right)\div\left(0.25-\frac{1}{6}\right)\cdot \frac{7}{13}}$
3	$\left(\frac{3\frac{1}{3}+2.5}{2.5-\frac{1}{3}}\cdot\frac{4.6-2\frac{1}{3}}{4.6+2\frac{1}{3}}\cdot 5.2\right)\div\left(\frac{0.05}{\frac{1}{7}-0.125}+5.7\right)$	4	$\frac{0.4+8\cdot\left(5-0.8\cdot\frac{5}{8}\right)-5\div 2\frac{1}{2}}{\left(1\frac{7}{8}\cdot 8-\left(8.9-2.6\div\frac{2}{3}\right)\right)\cdot 34\frac{2}{5}}\cdot 90$
5	$\frac{\left(\frac{3}{5}+0.425-0.005\right)\div 0.1}{30.5+\frac{1}{6}+3\frac{1}{3}}+\frac{6\frac{3}{4}+5\frac{1}{2}}{26\div 3\frac{5}{7}}-0.05$	6	$\frac{3\frac{1}{3}\cdot 1.9+19.5\div 4\frac{1}{2}}{\frac{62}{75}-0.16}\div\frac{3.5+4\frac{2}{3}+2\frac{2}{15}}{0.5\cdot\left(1\frac{1}{20}+4.1\right)}$
7	$\frac{\left(1\frac{1}{5}\div\left(\frac{17}{40}+0.6-0.005\right)\right)\cdot 1.7}{\frac{5}{6}+1\frac{1}{3}-1\frac{23}{30}}+\frac{4.75+7\frac{1}{2}}{33\div 4\frac{5}{7}}\div 0.25$	8	$\frac{\left(4.5\cdot 1\frac{2}{3}-6.75\right)\cdot \frac{2}{3}}{\left(3\frac{1}{3}\cdot 0.3+5\frac{1}{3}\cdot\frac{1}{8}\right)\div 2\frac{2}{3}}+\frac{1\frac{4}{11}\cdot 0.22\div 0.3-0.96}{\left(0.2-\frac{3}{40}\right)\cdot 1.6}$
9	$\frac{\left(1.88+2\frac{3}{25}\right)\cdot \frac{3}{16}}{0.625-\frac{13}{18}\div\frac{26}{9}}+\frac{\left(\frac{0.216}{0.15}+0.56\right)\div 0.5}{\left(7.7\div 24\frac{3}{4}+\frac{2}{15}\right)\cdot 4.5}$	10	$\frac{0.128\div 3.2+0.86}{\frac{5}{6}\cdot 1.2+0.8}\cdot\frac{\left(1\frac{32}{63}-\frac{13}{21}\right)\cdot 3.6}{0.505\cdot\frac{2}{5}-0.002}$
11	$\frac{0.125\div 0.25+1\frac{9}{16}\div 2.5}{(10-22\div 2.3)\cdot 0.46+1.6}+\left(\frac{17}{20}+1.9\right)\cdot 0.5$	12	$\frac{(3.4-1.275)\cdot\frac{16}{17}}{\frac{5}{18}\cdot\left(1\frac{7}{85}+6\frac{2}{17}\right)}+0.5\cdot\left(2+\frac{12.5}{5.75+\frac{1}{2}}\right)$
3	$\frac{\left(0.3275-\left(2\frac{15}{88}+\frac{4}{33}\right)\div 12\frac{2}{9}\right)\div 0.07}{(13-0.416)\div 6.05+1.92}$	4	$\frac{3.75\div 1\frac{1}{2}+\left(1.5\div 3\frac{3}{4}\right)\cdot 2\frac{1}{2}+\left(1\frac{1}{7}-\frac{23}{49}\right)\div\frac{2}{1}}{2\div 3\frac{1}{5}+\left(3\frac{1}{4}\div 13\right)\div\frac{2}{3}-\left(2\frac{5}{18}-\frac{17}{36}\right)\cdot\frac{18}{65}}$
5	$\frac{\left(\left(4.625-\frac{13}{18}\cdot\frac{9}{26}\right)\div\frac{9}{4}+2.5\div 1.25\div 6.75\right)}{\left(\frac{1}{2}-0.375\right)\div 0.125+\left(\frac{5}{6}-\frac{7}{12}\right)\div(0.358-1.4)}$	6	$\frac{\left(\left(3\frac{7}{12}-2\frac{11}{18}+2\frac{1}{24}\right)\cdot 1\frac{5}{31}-\frac{3}{52}\cdot\left(3\frac{1}{2}+\frac{5}{6}\right)\right)}{\frac{19}{84}\div\left(5\frac{13}{42}-2\frac{13}{28}+\frac{5}{24}\right)+1\frac{2}{27}-\frac{1}{3}\cdot\frac{4}{9}}$

7.3 Вычислить значение арифметического выражения. Результат выведите с 6 знаками после запятой.

Вариант	Значения переменных	Выражение
---------	---------------------	-----------

1	$x = 3.981$ $y = 1.625$ $c = 0.512$	$h = \frac{\sqrt{c + x^2 \cdot (\cos^5(x) - c)} + \sqrt[5]{\sin x + \ln y}}{c + y}$
2	$x = -6.251$ $a = 0.827$ $z = 25.001$	$b = \frac{x^3 + z}{\cos^2 x + 1} + \operatorname{tg} x^2 - \sqrt{\sin x - a} + \frac{e^x}{3x^2}$
3	$x = 3.251$ $y = 3.325$ $z = 0.466$	$h = \frac{\sin z + \cos 2x}{2x^5 + \operatorname{tg} x} + \sqrt[3]{3x + 2y}$
4	$x = 0.622$ $y = 3.325$ $z = 5.541$	$\varphi = \frac{(\cos x - \sin y)^3}{\sqrt{\operatorname{tg}(z)}} + \ln^2(x \cdot y \cdot z)$
5	$x = 17.421$ $b = 10.365$ $z = 0.828$	$k = \frac{1 + \sin^3 x}{z^2} + \cos^2 x + \frac{\ln^2 x + b}{x^4}$
6	$x = 2.444$ $y = 0.869$ $z = -0.166$	$g = \left x^2 - \frac{1}{e^a + 3} \right - \frac{1 + \sin^3 x}{a^2}$
7	$x = 0.335$ $y = 0.025$ $z = 32.005$	$t = y^{x+1} + \sqrt{ x + e^y} - \frac{z^{3x} - \sin^2(y)}{y + z^2/(e^x)}$
8	$x = 3.258$ $r = 4.005$ $z = -0.666$	$p = \frac{e^x - 2}{z + 3} + \sqrt{\sin^2 x^5} - \frac{r^3 + 1}{\cos^2(r - 2) + 1}$
9	$x = 0.100$ $y = -8.750$ $z = 0.765$	$y = \left((1 + y) \cdot \sqrt{\sin^2(z)} - \frac{ y - x }{5} \right)^3$
10	$x = 1.542$ $a = 3.261$ $z = 8.005$	$r = \frac{x^2}{e^a} + \frac{1}{3} \cdot \sin^2 z - \ln \sqrt{2x}$
11	$a5 = 1.426$ $a = -1.220$ $p = 3.500$	$w = p^{0.8} + \frac{a}{a - p} - \sin^2 \frac{a^5}{a^5 - 1}$
12	$x = -4.500$ $y = 0.750$ $z = 0.845$	$z = \sqrt{\left \frac{-3 \cdot \operatorname{tg} y \cdot \ln(x^4 + z)}{e^{-x} + 1} \right }$
13	$a = 3.741$ $x = 0.825$ $z = 5.160$	$v = \operatorname{tg} \frac{\sqrt[3]{a}}{5 + a^3} + \frac{\sin z - \operatorname{tg} 2x}{e^x}$
14	$x = 0.400$ $a = 2.875$ $f = -0.475$	$d = \frac{\sin^2 x + 1}{x^4} + \cos^3 x + e^{a-1} - \frac{\sqrt{\sin f^2 + f}}{\cos^2 a}$
15	$t = 0.750$ $a = 0.845$ $m = 2.5$	$f = \frac{e^{ma+t}}{\sqrt[5]{\frac{ma}{t} + ma^2}} + \frac{(am) - e^t}{\sqrt{2 + a^2} - m^3 - \ln t }$

16	$x = -15.24$ $a = 5.642$ $b = 20.001$	$u = (1 + x^2) \cdot x + \frac{e^x}{2x^2 + 5} - \frac{a}{b \cdot \cos x^3}$
----	---	---

7.4 Определить ранжированные переменные x, y, и z, показать их значения в таблицах вывода.

Определить по заданному выражению функцию пользователя, вычислить значения функции для переменных x, y, и z и показать их в таблице вывода.

Вариант	Ранжированная переменная	Выражение
1	$x = 3, 3.9..5$ $y = 5, 4.6..1$ $z = 5..10$	$2x^3 - 9x^2 + 1$
2	$x = 4, 4.9..6$ $y = 3, 1.4..-2$ $z = 6..11$	$5x^3 - x^2 + 3$
3	$x = 5, 5.9..7$ $y = -7, -8.6..-13$ $z = 7..12$	$x^2 - 10x + 2$
4	$x = 6, 6.9..8$ $y = 8, 7.4..4$ $z = -8..-4$	$x^2 - 4 \sin(x)$
5	$x = 7, 7.9..9$ $y = 8, 7.3..5$ $z = 9..15$	$\cos 2x - 0.4x^3 + 1$
6	$x = 8, 8.9..10$ $y = 9, 7.8..5$ $z = 10..15$	$(x + 1)^3 + x - 2$
7	$x = 3, 3.5..6$ $y = 6, 5.3..2$ $z = 6..10$	$\ln\left(\frac{1}{x}\right) - 4$
8	$x = 5, 5.6..8$ $y = 4, 2.3..0$ $z = 8..13$	$2 \cdot x - \sin x$
9	$x = 3, 3.5..8$ $y = 9, 7.5..-4$ $z = -9..4$	$\frac{1}{x^2} - 3$
10	$x = 2, 2.5..7$ $y = 8, 6.5..1$ $z = 6..12$	$\cos\left(-\frac{x}{100}\right) \cdot \sin\left(\frac{x}{10}\right)$
11	$x = 8, 8.3..10$ $y = 10, 7.5..3$ $z = 1..5$	$\sin\left(\frac{2}{x + 25}\right) - 4$
12	$x = 7, 7.5..9.5$ $y = 9, 8.8..6$ $z = 22..27$	$x^2 - 3x + 10$
13	$x = 8, 8.9..10$ $y = 2, 1.8..-0.2$ $z = -1..5$	$x^3 + x^2 - 12$
14	$x = 2, 2.5..6$ $y = 10, 6.8..1$ $z = -3..5$	$x^2 - 2 \cos(x)$
15	$x = 8, 8.5..10$ $y = 6, 4.3..1$	$(x - 0.5)^3 + 1.5x$

	$z = 10..25$	
16	$x = 1,1.5..5$ $y = 8,6.8..1$ $z = -2..4$	$x^2 - 2\sin(x)$

7.5

Определить функцию $f(x)$, вычислить ее значение при $x = 2,9$ и построить таблицу значений функции для $x [2; 12]$ с шагом 1. Построить график функции.

Вариант	Функция	Вариант	Функция
1	$\frac{8(x-1)}{(x+2)^2}$	2	$(2x+4)e^{2(x+2)}$
3	$\frac{5x}{x^2+3}$	4	$2 - \frac{3x}{x^2+3}$
5	$\frac{x^3-27x+54}{x^3}$	6	$-(x+4)e^{-x-3}$
7	$-\frac{5x}{x^2+2}$	8	$\left(2 + \frac{1}{x}\right)^2$
9	$\left(\frac{x+2}{x-2}\right)^2$	10	$\frac{2x^2+1}{x^2+3}$
11	$(x+1)e^{x+2}$	12	$(2x-1)e^{2(x-1)}$
13	$\frac{x^2-6x+9}{(x-1)^2}$	14	$\frac{5x^2}{x^2+3}$
15	$(x+4)e^{-x-3}$	16	$\frac{3x-2}{(x+1)^3}$

7.6

На одном графике постройте графики функций:

1. $\sin x$
2. $\sin 2x$
3. $2 \sin x$
4. $\sin x^2$

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. Назовите основные элементы интерфейса программы MathCAD.
2. С помощью какого оператора можно вычислить выражение?
3. Назовите правила записи имен переменных.
4. Как вставить текстовую область в документ MathCAD?
5. Чем отличается глобальное и локальное определение переменных? С помощью каких операторов определяются?
6. Как изменить формат результата для всего документа?
7. Как изменить формат результата для отдельного выражения?
8. Какие системные (предопределенные) переменные Вам известны? Как узнать их значение? Как изменить их значение?
9. Какие виды функций в MathCAD Вам известны?
10. Как вставить встроенную функцию в документ MathCAD?
11. Как создать функцию пользователя?
12. Как определить дискретные переменные с произвольным шагом? Какой шаг по умолчанию?
13. Как вывести значения дискретной переменной?
14. Как построить график?

Список литературы, рекомендуемый к использованию по данной теме

Основная литература:

1. Голубева, Н.В. Математическое моделирование систем и процессов. - Санкт-Петербург: Лань, 2021;
2. Казиев, В.М. Введение в анализ, синтез и моделирование систем: Учебное пособие. - Москва, Саратов: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020;
3. Поспелов, Е.А., Попов, И.С. Пакеты прикладных программ в научных исследованиях: учебно-методическое пособие. - Омск: Омский государственный университет им. Ф.М. Достоевского, 2019.

Дополнительная литература:

1. Агалаков, С.А. Анализ данных в среде R: практикум. - Омск: Омский государственный университет им. Ф.М. Достоевского, 2020;
2. Воскобойников, Ю.Е. Регрессионный анализ данных в пакете MATHCAD. - Санкт-Петербург: Лань, 2021.

Интернет-ресурсы

1. <http://biblioclub.ru> – ЭБС «Университетская библиотека on-line», <http://old.exponenta.ru/> – Образовательный математический сайт по математике и программированию
2. <http://MATLAB.exponenta.ru/Simulink/default.php> – сайт, посвященный интерактивному инструменту для моделирования Simulink,
3. <https://www.intuit.ru/> – Национальный Открытый Университет «ИНТУИТ»
4. <http://window.edu.ru/> – Информационная система «Единое окно доступа к образовательным ресурсам»
5. <http://artspsb.com/> – Образовательный портал: математика, кибернетика и программирование дополнительного образования
6. <http://www.arenasimulation.com> – Официальный сайт разработчика среды имитационного моделирования ARENA
7. <https://www.anylogic.ru> – Официальный сайт разработчика AnyLogic

Лабораторная работа 8. Построение графиков

Цель лабораторной работы:

сформировать у студентов навыки построения графиков в системе MathCAD

Организационная форма занятий: решение проблемных задач, разбор конкретных ситуаций

Вопросы для обсуждения на лабораторном занятии: построение графиков

Формируемые компетенции: ОПК-2, ОПК-3, ОПК-4, ОПКД-2, ОПКД-3, ОПКД-4.

Общие сведения

В MathCAD встроено несколько различных типов графиков, которые можно разбить на две большие группы.

Двумерные графики:

- X-Y (декартовый) график (X-Y Plot);
- полярный график (Polar Plot).

Трёхмерные графики:

- график трёхмерной поверхности (Surface Plot);
- график линий уровня (Contour Plot);
- трёхмерная гистограмма (3D Bar Plot);
- трёхмерное множество точек (3D Scatter Plot);
- векторное поле (Vector Field Plot).

Деление графиков на типы несколько условно, т. к., управляя установками многочисленных параметров, можно создавать комбинации типов графиков, а также новые типы (например, двумерная гистограмма распределения является разновидностью простого X-Y графика).

Для построения графиков используются шаблоны. Их перечень содержится в команде меню Вставка→Графики. Большинство параметров графического процессора, необходимых для построения графиков, по умолчанию задается автоматически. Поэтому для начального построения того или иного вида достаточно задать тип графика. В подменю Graph содержится список из семи основных типов графиков.

X-Y Plot	График в декартовой системе координат
Polar Plot	График в полярных координатах
Surface Plot	Трёхмерный график
Contour Plot	Контурный график трёхмерной поверхности
3D Scatter Plot	График в виде точек (фигур) в трёхмерном пространстве
3D Bar Chart	График для изображения в виде совокупности столбиков в трёхмерном пространстве (гистограмма)
Vector Field Plot	График векторного поля на плоскости

MathCAD представляет пользователю разнообразные средства форматирования графика - изменение толщины и цвета линий, вида осей координат, координатные сетки, текстовые комментарии и др. Для того чтобы изменить вид изображения, нужно щелкнуть дважды по полю графика и установить требуемые параметры в окнах настройки.

Графики любого вида, как любые объекты документа, можно выделять, заносить в буфер обмена, вызывать их оттуда и переносить в любое новое место документа. Их можно и просто перетаскивать с места на место курсором мыши, а также растягивать по горизонтали, по вертикали и по диагонали, цепляясь за специальные маркеры выделенных графиков курсором мыши.

Порядок действий при построении всех графиков одинаков. После выбора шаблона построения графика в рабочем документе открывается поле построения графика с помеченными для ввода позициями, которые нужно заполнить для определения графика.

Когда график определен (заполнены все помеченные позиции), то для построения графика при автоматическом режиме вычислений достаточно щелкнуть мышью вне поля графика.

Заполнение шаблона для разных типов графиков имеет свои особенности.

Можно начертить *несколько кривых на одном* и том же чертеже. Чтобы представить графически несколько выражений по оси ординат относительно одного выражения по оси абсцисс, введите первое выражение по оси ординат, сопровождаемое запятой. Непосредственно под первым выражением появится пустое поле. Введите туда второе выражение, сопровождаемое другой запятой, чтобы получить пустое поле, и т. д.

Чтобы построить несколько независимых кривых на одном чертеже, введите два или более выражения, отделяемых запятыми по оси абсцисс, и то же самое выражение по оси ординат. MathCAD согласует выражения попарно - первое выражение по оси абсцисс с первым выражением по оси ординат, второе со вторым и т. д. Затем рисуется график каждой пары.

Можно построить до 16 функций по оси ординат в зависимости от одного аргумента по оси абсцисс. Однако если для каждой кривой используется свой аргумент, то можно отобразить только до 10 графиков.

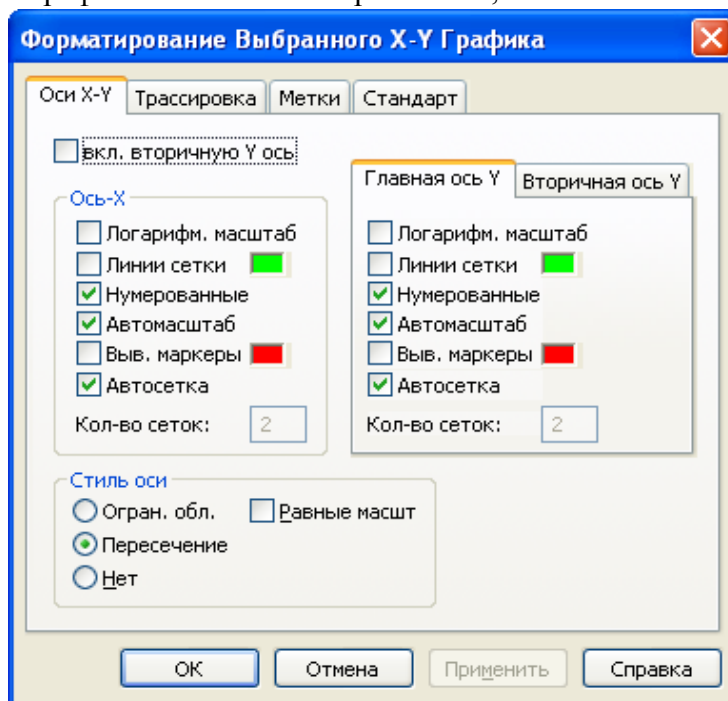
Точно так же можно построить несколько графиков на одном и том же чертеже в полярных координатах, используя эту же технологию заполнения шаблона графика.

Форматирование графиков

Чтобы изменить *формат графика*, необходимо дважды щелкнуть мышью в области графика.

Если строим график в декартовой системе координат, то появится следующее диалоговое окно для форматирования графика (разные типы графиков имеют разный вид диалоговых окон, но аналогичную технологию форматирования).

Форматирование оси графика можно также произвести, выполнив на ней двойной щелчок.



В MathCAD можно делать следующие надписи на чертеже:

- заголовок выше или ниже графика;
- названия осей, чтобы описать, что отложено на каждой оси;
- имена кривых, идентифицирующих отдельные графики;
- переменные - выражения, определяющие координаты.

Можно использовать эти надписи все вместе или в любой комбинации.

Для того чтобы *добавить заголовок* к графику в диалоговом окне для форматирования графика, следует щелкнуть по закладке Labels (Метки) и напечатать заголовок графика в поле Title (Название). Пометить место размещения заголовка: кнопка Above (Вверху) или Below (Внизу) и удостовериться, что флажок Show Title (Выводить) отмечен.

Чтобы надписать одну или обе оси графика, необходимо указать название осей в поле Метки осей.

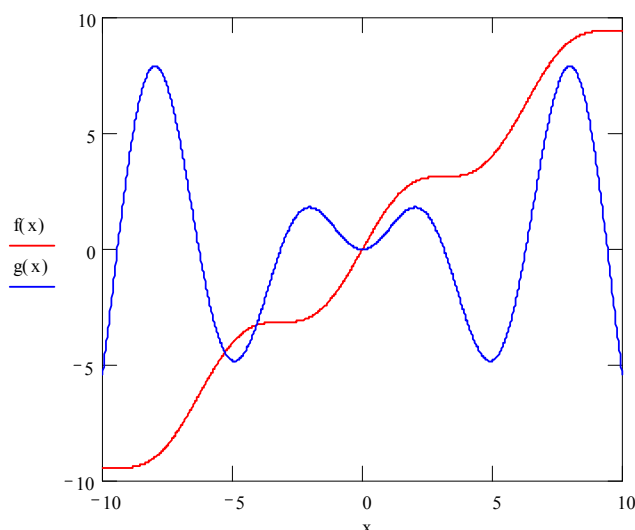
Можно построить до 16 разных графиков. Каждому графику соответствует строка в прокручиваемом списке, который откроется, если в диалоговом окне для форматирования графика щелкнуть по вкладке Traces (Трассировка). На этой вкладке можно изменить параметры: тип, цвет, толщину линии.

По мере появления новых графиков MathCAD ставит в соответствие каждому одну из этих строк.

Построение графика функции $y = f(x)$

Пример 1

$$f(x) := x + \sin(x) \quad g(x) := x \cdot \sin(x)$$

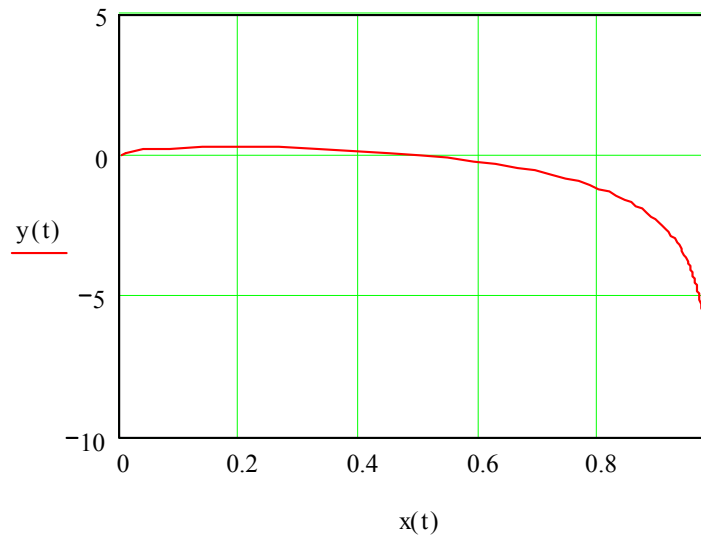


Построение кривой, заданной параметрически

Построение кривой, заданной параметрически, осуществляется аналогично. Отличие состоит в том, что в позиции аргумента и функции вводятся выражения или имена соответствующих функций.

Пример 2

$$x(t) := \frac{t^2}{1+t^2} \quad y(t) := \frac{t \cdot (1-t^2)}{1+t^2} \quad t := 0, 0.1..10$$



Графики в полярной системе координат

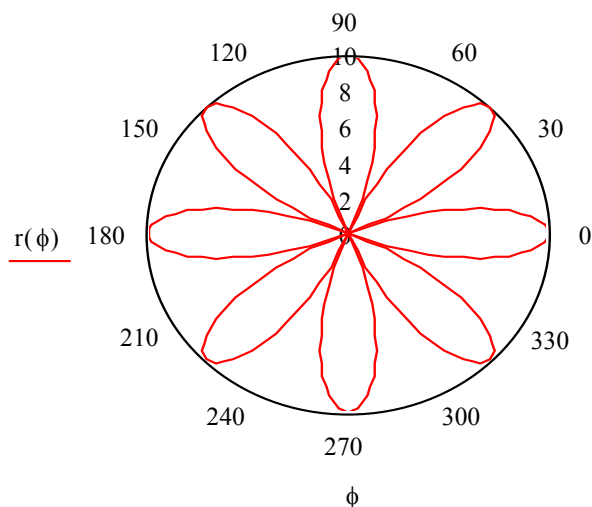
В полярной системе координат каждая точка задается углом ϕ и модулем радиуса-вектора $r(\phi)$. График функции обычно строится в виде линии, которую описывает конец радиуса-вектора при изменении угла ϕ в определенных пределах, чаще всего от 0 до 2π . Опция Полярные координаты (Polar Plot) выводит шаблон таких графиков в форме окружности с шаблонами данных.

Перед построением таких графиков надо задать значения переменной ϕ и функцию $r(\phi)$.

Пример 3

$$a := 10 \quad m := 4$$

$$\phi := -\pi, -\pi + \frac{\pi}{120} .. \pi \quad r(\phi) := a \cdot \cos(m \cdot \phi)$$



Графики поверхностей

Трехмерные, или 3D-графики, отображают функции двух переменных вида $Z(X, Y)$.

При построении трехмерных графиков в ранних версиях MathCAD поверхность нужно было определить математически.

Теперь применяют функцию MathCAD *CreateMesh*.

Функция **CreateMesh**(F (или G, или f1, f2, f3), x0, x1, y0, y1, xgrid, ygrid, fmap) - создает сетку на поверхности, определенной функцией F.

x0, x1, y0, y1 – диапазон изменения переменных; xgrid, ygrid – размеры сетки переменных; fmap – функция отображения.

Функция CreateMesh по умолчанию создает сетку на поверхности с диапазоном изменения переменных от –5 до 5 и с сеткой 20×20 точек.

Пример 4. (Построение графиков поверхности двумя способами)

1 способ

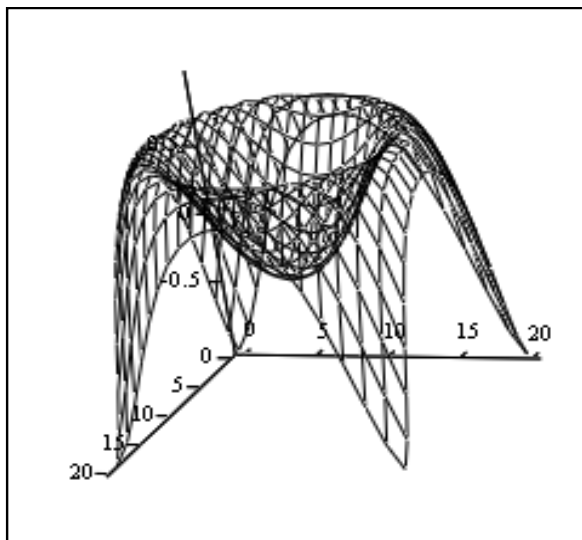
$$f(x, y) := \sin(x^2 + y^2)$$

$$i := 0..20 \quad j := 0..20$$

$$x_i := -1.5 + i \cdot 0.15$$

$$y_j := -1.5 + j \cdot 0.15$$

$$MT_{i,j} := f(x_i, y_j)$$

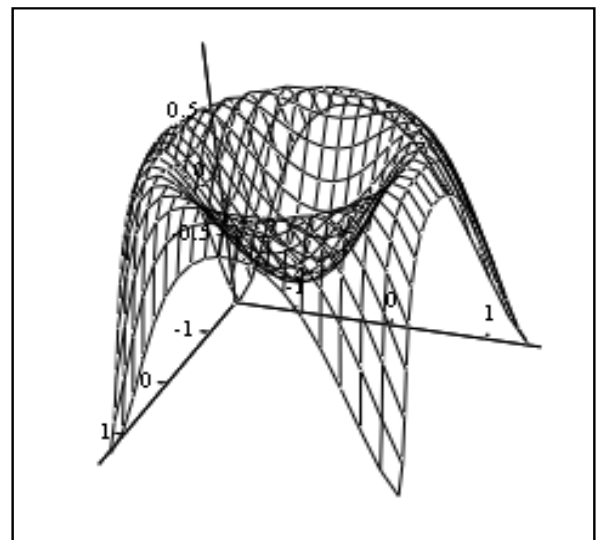


MT

2 способ

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

$$MM := \text{CreateMesh}(f, -1.5, 1.5, -1.5, 1.5, 20, 20)$$



MM

Нередко поверхности и пространственные кривые представляют в виде точек, кружочков или иных фигур. Такой график создается операцией Вставка → График → 3D Точечный, причем поверхность задается параметрически – с помощью трех матриц (X, Y, Z).

Для определения исходных данных для такого вида графиков используется функция *CreateSpace*.

Функция CreateSpace (F, t0, t1, tgrid, fmap) - возвращает вложенный массив трех векторов, представляющих x, y, и z - координаты пространственной кривой, определенной функцией F. t0 и t1 – диапазон изменения переменной, tgrid – размер сетки переменной, fmap – функция отображения.

Еще один вид представления поверхности - *векторное представление*. Оно задается построением коротких стрелочек - векторов. Стрелки обращены острием в сторону нарастания высоты поверхности, а плотность расположения стрелок зависит от скорости этого нарастания.

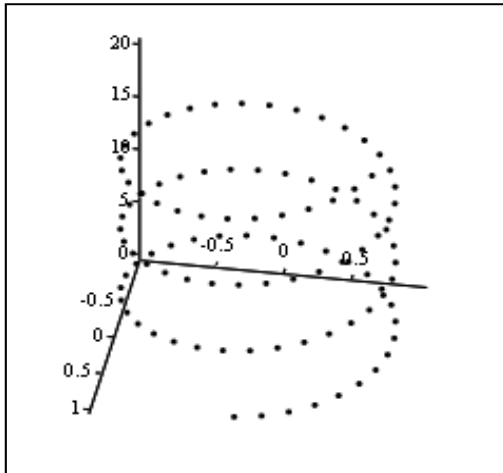
Для его построения используется шаблон Vector Field Plot (график векторного поля на плоскости). В шаблон необходимо внести имя матрицы M.

Пример 5. Построение точечного графика двумя способами

1 способ

$t := 0..100$

$$x_t := \cos\left(\frac{t}{5}\right) \quad y_t := \sin\left(\frac{t}{5}\right) \quad z_t := \frac{t}{5}$$

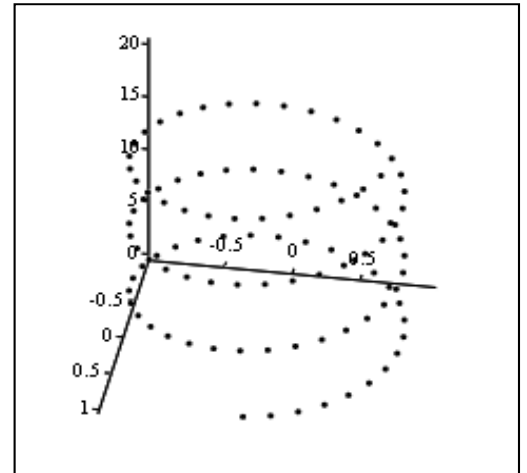


(x, y, z)

2 способ

$$F(t) := \begin{pmatrix} \cos(t) \\ \sin(t) \\ t \end{pmatrix}$$

$M := \text{CreateSpace}(F, 0, 20, 100)$



M

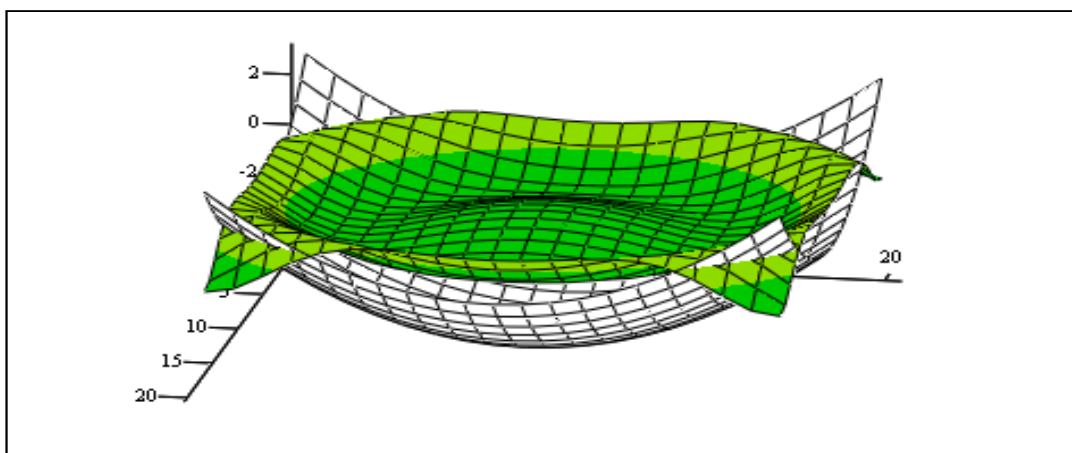
Построение пересекающихся фигур

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

$x := 0..20 \quad y := 0..20$

$$f1(x, y) := -\sin(x^2 + y^2) \quad f2(x, y) := x^2 + y^2 - 5$$

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



M1, M2

Выполнить задание

8.1

Постройте графики функций.

Вариант	Функция одной переменной	Полярная система координат	Функция двух переменных
1	$y = \frac{3}{x^3} + \frac{2}{x^2} + \frac{1}{x}$	$x = t^3 - 3\pi$ $y = t^3 - 6 \cdot \arctg(t)$	$z = \sin\left(\frac{x}{y}\right) \cos\left(\frac{y}{x}\right)$
2	$y = \sqrt{x} - \sqrt[3]{x^2} + \sqrt[4]{x^3}$	$x = 4\cos^2(t)$ $y = 4\sin^2(t)$	$z = \frac{1}{\arctg\left(\frac{y}{x}\right)}$
3	$y = \ln(3x) + \frac{\exp(-3x)}{\sqrt{x}}$	$x = sh(t) - t$ $y = ch(t) - 1$	$z = x^3 y - xy^3$
4	$y = \frac{x^2 - \sqrt{x}}{1 - x}$	$x = t$ $y = t + 2\arctg(t)$	$z = \exp\left(-\frac{x}{y}\right)$
5	$y = \frac{x^2}{x^3 + 1}$	$x = 2 \cdot (3\cos(t) + \cos(3t))$ $y = 2 \cdot (3\sin(t) + \sin(3t))$	$z = 4.25x \cdot \exp(-t) + 6t$
6	$y = \sin(x) - 4\cos(x)$	$x = t^3 + 3t + 1$ $y = t^3 - 3t + 1$	$z = \frac{x^3 + y^3}{x^2 + y^2}$
7	$y = x^2 \cdot tg(x)$	$x = \frac{3t}{1 + t^3}$ $y = \frac{3t^2}{1 + t^3}$	$z = \ln\left(x + \sqrt{x^2 + y^2}\right)$
8	$y = \frac{\sqrt[3]{x}}{\cos(x)}$	$x = t \cdot \exp(t)$ $y = t \cdot \exp(-t)$	$z = \ln\left(tg\left(\frac{x}{y}\right)\right)$
9	$y = \frac{\cos(x) - \sin(x)}{\cos(x) + \sin(x)}$	$x = 3t + 1$ $y = t^3 + 2t$	$z = \ln(x^2 + y^2)$
10	$y = (1 + x^2) \arccos(x)$	$x = t + \exp(-t)$ $y = 2t + \exp(-2t)$	$z = x^{x \cdot y}$
11	$y = \sqrt{x^3} \arctg(x)$	$x = \frac{t}{t + 3}$ $y = \frac{2t^2 + 3t}{(t + 3)^2}$	$z = (1 + \lg(x))^y$
12	$y = \sin(x) \cdot \arcsin(x)$	$x = \ln\left(t + \sqrt{t^2}\right)$ $y = t\sqrt{t^2 + 1}$	$z = \frac{x + y}{x - y}$
13	$y = \frac{x^2 - 1}{\lg(x)}$	$x = 6\cos(t) - 3\cos(2t)$ $y = 6\sin(t) - 3\sin(2t)$	$z = \frac{1}{2} \ln(x^2 + y^2)$
14	$y = x \cdot \cos(x) \cdot \ln(x)$	$x = t\sqrt{t^2 + 1}$ $y = \ln\left(\frac{1 + \sqrt{1 + t^2}}{t}\right)$	$z = \frac{x^2 + y^2}{x^2 - y^2}$

15	$y = \ln(\sqrt{\exp(x)})$	$x = 2 \cos(t)$ $y = 3 \sin(t)$	$z = x^2 y^4 - x^3 y^3 + x^4 y^2$
16	$y = \exp(x) \cdot (tg(x) - x)$	$y = a(1 + \cos(\varphi))$	$z = 2 \sqrt{\frac{1 - \sqrt{xy}}{1 + \sqrt{xy}}}$

8.2

Отобразить графически пересечение поверхностей $f1(x, y) := \frac{(x + y)^2}{10}$ и

$f2(x, y) := 5 \cos\left(\frac{x - y}{3}\right)$. Матрицы для построения поверхностей задать с помощью функции *CreateMesh*.

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. Как построить график?
2. Как построить несколько графиков в одной системе координат?
3. Как построить декартовый график?
4. Как отформатировать построенный график?
5. Как построить график кривой, заданной параметрически?
6. Как построить график в полярной системе координат?
7. Как построить график поверхности?
8. Для чего используются функции *CreateMesh*, *CreateSpace*?

Список литературы, рекомендуемый к использованию по данной теме

Основная литература:

1. Голубева, Н.В. Математическое моделирование систем и процессов. - Санкт-Петербург: Лань, 2021;
2. Казиев, В.М. Введение в анализ, синтез и моделирование систем: Учебное пособие. - Москва, Саратов: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020;
3. Поспелов, Е.А., Попов, И.С. Пакеты прикладных программ в научных исследованиях: учебно-методическое пособие. - Омск: Омский государственный университет им. Ф.М. Достоевского, 2019.

Дополнительная литература:

1. Агалаков, С.А. Анализ данных в среде R: практикум. - Омск: Омский государственный университет им. Ф.М. Достоевского, 2020;
2. Воскобойников, Ю.Е. Регрессионный анализ данных в пакете MATHCAD. - Санкт-Петербург: Лань, 2021.

Интернет-ресурсы

1. <http://biblioclub.ru> – ЭБС «Университетская библиотека on-line», <http://old.exponenta.ru/> – Образовательный математический сайт по математике и программированию
2. <http://MATLAB.exponenta.ru/Simulink/default.php> – сайт, посвященный интерактивному инструменту для моделирования Simulink,
3. <https://www.intuit.ru/> – Национальный Открытый Университет «ИНТУИТ»
4. <http://window.edu.ru/> – Информационная система «Единое окно доступа к образовательным ресурсам»
5. <http://artspb.com/> – Образовательный портал: математика, кибернетика и программирование дополнительного образования
6. <http://www.arenasimulation.com> – Официальный сайт разработчика среды имитационного моделирования ARENA
<https://www.anylogic.ru> – Официальный сайт разработчика AnyLogic

Лабораторная работа 9. Векторы и матрицы

Цель лабораторной работы:

Сформировать у студентов способности работы с векторами и матрицами в системе MathCAD.

Организационная форма занятий: решение проблемных задач, разбор конкретных ситуаций

Вопросы для обсуждения на лабораторном занятии: векторы и матрицы в системе MathCAD

Формируемые компетенции: ОПК-2, ОПК-3, ОПК-4, ОПКД-2, ОПКД-3, ОПКД-4.

Общие сведения

Задачи линейной алгебры, решаемые в MathCAD, можно условно разделить на два класса. Первый - это простейшие матричные операции, которые сводятся к определенным арифметическим действиям над элементами матрицы. Они реализованы в виде операторов и нескольких специфических функций, предназначенных для создания, объединения, сортировки, получения основных свойств матриц и т. д. Второй класс - это более сложные действия, которые реализуют алгоритмы вычислительной линейной алгебры, такие как вычисление определителей и обращение матриц, вычисление собственных векторов и собственных значений, решение систем линейных алгебраических уравнений и различные матричные разложения.

Простейшие операции матричной алгебры реализованы в MathCAD в виде операторов, причем их запись максимально приближена к математическому значению. Каждый оператор выражается соответствующим символом. Некоторые операции применимы только к квадратным матрицам $N \times N$, некоторые допускаются только для векторов (например, скалярное произведение), а другие, несмотря на одинаковое написание, по-разному действуют на векторы и матрицы.

Создание матриц

Имеется два способа создать матрицу.

1-й способ. Использование команды создания массивов:

- Воспользоваться командой Вставка → Матрица;
- нажатие клавиш Ctrl+M;
- выбор пиктограммы с изображением шаблона матрицы на панели инструментов

Матрицы.

В диалоговом окне указать размерность матрицы, т. е. количество ее строк m (Rows) и столбцов n (Columns).

Для векторов один из этих параметров должен быть равен 1. При $m = 1$ получим вектор-столбец, а при $n = 1$ - вектор-строку.

Далее на экране появится шаблон $\begin{pmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{pmatrix}$, в который нужно ввести значения элементов массива.

Обращаться к отдельным элементам вектора или матрицы можно используя нижний индекс. Для элемента матрицы указываются два индекса, один - для номера строки, другой - для номера столбца.

Чтобы ввести нижний индекс, нужно нажать клавишу [после имени вектора или матрицы

или выбрать команду  на панели Матрицы.

Команды панели инструментов Матрицы

Кнопка	Назначение
	Создание матрицы
	Обратная матрица
	Определитель матрицы
	Транспонирование матрицы
	Выделение столбца матрицы

Операторы для работы с массивами

Обозначения: для векторов - V , для матриц - M и для скалярных величин - z .

Оператор	Ввод	Назначение оператора
$V1+V2$	$V1+V2$	Сложение двух векторов $V1$ и $V2$
$V1-V2$	$V1-V2$	Вычитание двух векторов $V1$ и $V2$
$-M$	$-M$	Смена знака у элементов матрицы M
$V-z$	$V-z$	Вычитание из вектора V скаляра z
$z*V, V*z$	$z*V, V*z$	Умножение вектора V на скаляр z
$z*M, M*z$	$z*M, M*z$	Умножение матрицы M на скаляр z
$V1*V2$	$V1*V2$	Умножение двух векторов $V1$ и $V2$
$M*V$	$M*V$	Умножение матрицы M на вектор V
$M1*M2$	$M1*M2$	Умножение двух матриц $M1$ и $M2$
$\frac{V}{z}$	V/z	Деление вектора V на скаляр z
$\frac{M}{z}$	M/z	Деление матрицы M на скаляр z
M^n	M^n	Возведение матрицы M в степень n

Функции для работы с векторами и матрицами.

Некоторые из них (V должен быть вектором, A может быть вектором либо матрицей):

$\text{length}(V)$ -возвращает число элементов в векторе v ;

$\text{last}(V)$ - возвращает индекс последнего элемента;

$\text{max}(A)$ - возвращает максимальный по значению элемент;

$\text{min}(A)$ - возвращает минимальный по значению элемент.

Матричные функции

Для работы с матрицами также существует ряд встроенных функций:

$\text{augment}(M1, M2)$ - объединяет в одну матрицы $M1$ и $M2$, имеющие одинаковое число строк;

$\text{identity}(n)$ - создает единичную квадратную матрицу размером $n \times n$, (n – размер матрицы(число));

stack(M1, M2) - объединяет две матрицы M1 и M2, имеющие одинаковое число столбцов, сажая M1 над M2;
 diag(V) - создает диагональную матрицу, элемент главной диагонали которой - вектор V;
 cols(M) - возвращает число столбцов матрицы M;
 rows(M) - возвращает число строк матрицы M;
 rank(M) - возвращает ранг матрицы M;
 tr(M) - возвращает след (сумму диагональных элементов) квадратной матрицы M;
 mean(M) - возвращает среднее значение элементов массива M;
 median(M) - возвращает медиану элементов массива M;
 eigenvals(M) - возвращает вектор, элементами которого являются собственные значения матрицы M (M должна быть квадратной матрицей.);
 submatrix(M,ir,jr,ic,jc) - возвращает подмассив, состоящий из всех элементов, которые содержатся в строках с ir по jr и столбцах с ic по jc массива M.

Символьные вычисления

Все матричные и векторные операторы допустимо использовать как в численных, так и в символьных расчетах. Мощь символьных операций заключается в возможности проводить их не только над конкретными числами, но и над переменными.

Фрагмент документа MathCAD:

$$\begin{pmatrix} a & b & c \\ d & f & g \end{pmatrix} + \begin{pmatrix} u & v & w \\ x & y & z \end{pmatrix} \rightarrow \begin{pmatrix} a+u & b+v & c+w \\ d+x & f+y & g+z \end{pmatrix}$$

$$\begin{pmatrix} a & b & c \\ d & f & g \end{pmatrix} - \begin{pmatrix} u & v & w \\ x & y & z \end{pmatrix} \rightarrow \begin{pmatrix} a-u & b-v & c-w \\ d-x & f-y & g-z \end{pmatrix}$$

$$\begin{pmatrix} a & b & c \\ d & f & g \end{pmatrix} + k \rightarrow \begin{pmatrix} a+k & b+k & c+k \\ d+k & f+k & g+k \end{pmatrix}$$

Выполнить задание

1. Ввести в документ название лабораторной работы, вариант задания и фамилию студента
2. Создать квадратные матрицы A, B, D, размером (5,5,4 соответственно) первым способом
3. Исследовать следующие свойства матриц на примере преобразования заданных массивов:
 - транспонированная матрица суммы двух матриц равна сумме транспонированных матриц $(A+B)^T = A^T + B^T$;
 - транспонированная матрица произведения двух матриц равна сумме произведению транспонированных матриц, взятых в обратном порядке: $(A*B)^T = B^T * A^T$;
 - при транспонировании квадратной матрицы определитель не меняется : $|D| = |D^T|$;
 - произведение квадратной матрицы на соответствующую ей квадратную дает единичную матрицу (элементы главной диагонали единичной матрицы равны 1, а все остальные – 0) $D * D^{-1} = E$.

4. Для матриц A,B найти обратные матрицы.
5. Найти определители матриц A,B.
6. Для матрицы A увеличить значения элементов в № раз, где № - номер варианта.
7. Для матрицы B увеличить значения элементов на №.
8. Создать вектор C вторым способом, количество элементов которого равно 6.
9. Применить к матрицам A, B, D встроенные матричные функции (всевозможные) из приведенных в пункте “Функции для работы.....”
10. Применить к вектору C встроенные векторные функции.
11. Применить ко всем матрицам и вектору общие встроенные функции.
12. Сохранить документ.

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. Как создать матрицу, вектор - строку, вектор - столбец?
2. Какие операторы есть для работы с матрицами?
3. Перечислите команды панели инструментов Матрицы.
4. Как вставить матричные функции?
5. Как выполнять вычисления, если матрица задана в символьном виде?

Список литературы, рекомендуемый к использованию по данной теме

Основная литература:

1. Голубева, Н.В. Математическое моделирование систем и процессов. - Санкт-Петербург: Лань, 2021;
2. Казиев, В.М. Введение в анализ, синтез и моделирование систем: Учебное пособие. - Москва, Саратов: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020;
3. Поспелов, Е.А., Попов, И.С. Пакеты прикладных программ в научных исследованиях: учебно-методическое пособие. - Омск: Омский государственный университет им. Ф.М. Достоевского, 2019.

Дополнительная литература:

1. Агалаков, С.А. Анализ данных в среде R: практикум. - Омск: Омский государственный университет им. Ф.М. Достоевского, 2020;
2. Воскобойников, Ю.Е. Регрессионный анализ данных в пакете MATHCAD. - Санкт-Петербург: Лань, 2021.

Интернет-ресурсы

14. <http://biblioclub.ru> – ЭБС «Университетская библиотека on-line», <http://old.exponenta.ru/> – Образовательный математический сайт по математике и программированию
15. <http://MATLAB.exponenta.ru/Simulink/default.php> – сайт, посвященный интерактивному инструменту для моделирования Simulink,
16. <https://www.intuit.ru/> – Национальный Открытый Университет «ИНТУИТ»
17. <http://window.edu.ru/> – Информационная система «Единое окно доступа к образовательным ресурсам»

18. <http://artspb.com/> – Образовательный портал: математика, кибернетика и программирование дополнительного образования
19. <http://www.arenasimulation.com> – Официальный сайт разработчика среды имитационного моделирования ARENA
- <https://www.anylogic.ru> – Официальный сайт разработчика AnyLogic

Лабораторная работа 10. Решение уравнений

Цель лабораторной работы:

Сформировать у студентов способности решать уравнения в системе MathCAD

Организационная форма занятий: решение проблемных задач, разбор конкретных ситуаций

Вопросы для обсуждения на лабораторном занятии: решение уравнений в системе MathCAD.

Формируемые компетенции: ОПК-2, ОПК-3, ОПК-4, ОПКД-2, ОПКД-3, ОПКД-4.

Общие сведения

Огромное количество задач вычислительной математики связано с решением нелинейных алгебраических уравнений, а также систем таких уравнений. При этом необходимость решения нелинейных уравнений возникает зачастую на промежуточных шагах, при реализации фрагментов более сложных алгоритмов (к примеру, при расчетах дифференциальных уравнений при помощи разностных схем и т. п.).

Численное решение нелинейного уравнения

Алгоритм приближенного решения уравнения $f(x)=0$ состоит из двух этапов:

1. нахождения промежутка, содержащего корень уравнения (или начальных приближений для корня);
2. получения приближенного решения с заданной точностью с помощью функции root.

Если после многих итераций MathCAD не находит подходящего приближения, то появится сообщение **Can't converge to a solution.** (отсутствует сходимость).

Нахождение корней полинома

Для нахождения корней выражения, имеющего вид $v_0 + v_1x + \dots + v_{n-1}x^{n-1} + v_nx^n$, лучше использовать функцию polyroots, нежели root. В отличие от функции root, функция polyroots не требует начального приближения и возвращает сразу все корни, как вещественные, так и комплексные.

Функция **Polyroots(v)** - возвращает корни полинома степени n. Коэффициенты полинома находятся в векторе v длины n + 1. Возвращает вектор длины n, состоящий из корней полинома.

Задание 2. Нахождение корней полинома

$$0.75 \cdot x^3 - 8 \cdot x + 5$$

$$v := \begin{pmatrix} 5 \\ -8 \\ 0 \\ .75 \end{pmatrix}$$

$$\text{polyroots}(v) = \begin{pmatrix} -3.542 \\ 0.651 \\ 2.892 \end{pmatrix}$$

Для создания вектора v:

1. Поставьте курсор на переменную x в выражении

$$0.75 \cdot x^3 - 8 \cdot x + 5$$

2. Выполните команду **Символы → Коэффициенты полинома**
3. Выполните команду **Правка → Вырезать**
4. Напечатайте v:= и выполните команду **Правка → Вставить**

Решение систем уравнений

Решение систем уравнений матричным методом

Рассмотрим систему n линейных алгебраических уравнений относительно n неизвестных $x_1, x_2, \dots, x_n$:

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

Система линейных уравнений может быть записана в матричном виде:

$$Ax = b,$$

где:

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

Если $\det A \neq 0$ то система или эквивалентное ей матричное уравнение имеет единственное решение.

Решение систем уравнений с помощью функции `Lsolve`

Системы линейных уравнений удобно решать с помощью функции `lsolve`.

Функция `lsolve(A, b)` - возвращает вектор решения x такой, что $Ax = b$.

Пример 3. Решение системы уравнений

$$\begin{cases} x_1 + 2x_2 + 3x_3 + 4x_4 = 30 \\ -x_1 + 2x_2 - 3x_3 + 4x_4 = 10 \\ x_2 - x_3 + x_4 = 3 \\ x_1 + x_2 + x_3 + x_4 = 10 \end{cases}$$

Запишем в матричном виде:

$$\underline{\underline{A}} := \begin{pmatrix} 1 & 2 & 3 & 4 \\ -1 & 2 & -3 & 4 \\ 0 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \quad \underline{\underline{b}} := \begin{pmatrix} 30 \\ 10 \\ 3 \\ 10 \end{pmatrix}$$

$$|A| = -4$$

$$\underline{\underline{x}} := A^{-1} \cdot \underline{\underline{b}} \quad \underline{\underline{x}} = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix}$$

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

$$\underline{\underline{x}} = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix}$$

Решение системы уравнений методом Гаусса

Метод Гаусса, его еще называют методом Гауссовых исключений, состоит в том, что систему уравнений приводят последовательным исключением неизвестных к эквивалентной системе с треугольной матрицей.

В матричной записи это означает, что сначала (прямой ход метода Гаусса) элементарными операциями над строками приводят расширенную матрицу системы к ступенчатому виду, а затем (обратный ход метода Гаусса) эту ступенчатую матрицу преобразуют так, чтобы в первых n столбцах получилась единичная матрица. Последний, $(n + 1)$ столбец этой матрицы содержит решение системы.

В MathCAD прямой и обратный ходы метода Гаусса выполняет функция $\text{rref}(A)$.

Задача 4. Решение системы уравнений методом Гаусса

$$\begin{cases} x_1 + 2x_2 + 3x_3 + 4x_4 = 30 \\ -x_1 + 2x_2 - 3x_3 + 4x_4 = 10 \\ x_2 - x_3 + x_4 = 3 \\ x_1 + x_2 + x_3 + x_4 = 10 \end{cases} \quad \underline{A} := \begin{pmatrix} 1 & 2 & 3 & 4 \\ -1 & 2 & -3 & 4 \\ 0 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \quad \underline{b} := \begin{pmatrix} 30 \\ 10 \\ 3 \\ 10 \end{pmatrix}$$

$$\underline{\text{ORIGIN}} := 1$$

Формирование расширенной матрицы системы:

$$A1 := \text{augment}(A, b) \quad A1 = \begin{pmatrix} 1 & 2 & 3 & 4 & 30 \\ -1 & 2 & -3 & 4 & 10 \\ 0 & 1 & -1 & 1 & 3 \\ 1 & 1 & 1 & 1 & 10 \end{pmatrix}$$

Приведение расширенной матрицы к ступенчатому виду (прямой и обратный ходы метода Гаусса)

$$A2 := \text{rref}(A1) \quad A2 = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 2 \\ 0 & 0 & 1 & 0 & 3 \\ 0 & 0 & 0 & 1 & 4 \end{pmatrix}$$

$$\underline{x} := \text{submatrix}(A2, 1, 4, 5, 5) \quad x = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix} \quad \text{Проверка:} \quad A \cdot x - b = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Решение систем уравнений с помощью функций Find или Minner

Для решения системы уравнений с помощью функции Find необходимо выполнить следующее:

1. Задать начальное приближение для всех неизвестных, входящих в систему уравнений. MathCAD решает систему с помощью итерационных методов;
2. Напечатать ключевое слово Given. Оно указывает MathCAD, что далее следует система уравнений;
3. Введите уравнения и неравенства в любом порядке. Используйте [Ctrl]= для печати символа =. Между левыми и правыми частями неравенств может стоять любой из символов <, >, ≥ и ≤;
4. Введите любое выражение, которое включает функцию Find, например: x:= Find(x, y).

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

Задание 5. Решение системы уравнений с помощью функции Find

$x_1 := 0$ $x_2 := 0$ $x_3 := 0$ $x_4 := 0$ Начальные приближения

Given

$$x_1 + 2x_2 + 3x_3 + 4x_4 = 30$$

$$-x_1 + 2x_2 - 3x_3 + 4x_4 = 10$$

$$x_2 - x_3 + x_4 = 3$$

$$x_1 + x_2 + x_3 + x_4 = 10$$

$$\text{Find}(x_1, x_2, x_3, x_4) = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix}$$

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

Функция **Minerr(x1, x2, ...)** - возвращает приближенное решение системы уравнений.

Число аргументов должно быть равно числу неизвестных.

Символьное решение уравнений

Имеются некоторые задачи, для которых возможности MathCAD позволяют находить решения в символьном (аналитическом) виде.

Решение уравнений в символьном виде позволяет найти точные или приближенные корни уравнения:

- если решаемое уравнение имеет параметр, то решение в символьном виде может выразить искомый корень непосредственно через параметр. Поэтому вместо того чтобы решать уравнение для каждого нового значения параметра, можно просто заменять его значение в найденном символьном решении;
- если нужно найти все комплексные корни полинома со степенью меньше или равной 4, символьное решение даст их точные значения в одном векторе или в аналитическом или цифровом виде.

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

Чтобы решить уравнение символьно, необходимо:

1. Напечатать выражение (для ввода знака равенства используйте комбинацию клавиш Ctrl + =);
2. Выделить переменную, относительно которой нужно решить уравнение, щелкнув на ней мышью;
3. Выбрать пункт меню Символы → Переменные → Вычислить.

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

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

1. Напечатать ключевое слово Given;

2. Напечатать уравнения в любом порядке ниже слова Given. Удостоверьтесь, что для ввода знака = используется Ctrl + =;
3. Напечатать функцию Find, соответствующую системе уравнений;
4. Нажать Ctrl + . (клавиша CTRL, сопровождаемая точкой). MathCAD отобразит символьный знак равенства $\rightarrow$;
5. Щелкнуть мышью на функции Find.

Задание 6. Решение системы уравнений в символьном виде

Given

$$x + 2 \cdot \pi \cdot y = a$$

$$4 \cdot x + y = b$$

$$\text{Find}(x, y) \rightarrow \begin{bmatrix} \frac{2 \cdot \pi \cdot b - a}{-1 + 8 \cdot \pi} \\ \frac{-(-4 \cdot a + b)}{-1 + 8 \cdot \pi} \end{bmatrix}$$

Выполнить задание

Задание № 1

Построить график функции $f(x)$ и приблизительно определить один из корней уравнения. Решить уравнение $f(x) = 0$ с помощью встроенной функции MathCAD root.

Вариант	$f(x)$	Вариант	$f(x)$
1	$e^{x-1} - x^3 - x$ $x \in [0, 1]$	2	$0.25x^3 + x - 2$ $x \in [0, 2]$
3	$x - \frac{1}{3 + \sin(3.6x)}$ $x \in [0, 1]$	4	$\arccos \frac{1-x^2}{1+x^2} - x$ $x \in [2, 3]$
5	$\arccos x - \sqrt{1 - 0.3x^3}$ $x \in [0, 1]$	6	$3x - 4 \ln x - 5$ $x \in [2, 4]$
7	$\sqrt{1 - 0.4x^2} - \arcsin x$ $x \in [0, 1]$	8	$e^x - e^{-x} - 2$ $x \in [0, 1]$
9	$3x - 14 + e^x - e^{-x}$ $x \in [1, 3]$	10	$\sqrt{1-x} - \operatorname{tg} x$ $x \in [0, 1]$

11	$\sqrt{2x^2 + 1.2} - \cos x - 1$ $x \in [0, 1]$	12	$1 - x + \sin x - \ln(1 + x)$ $x \in [0, 2]$
13	$\cos\left(\frac{2}{x}\right) - 2\sin\left(\frac{1}{x}\right) + \frac{1}{x}$ $x \in [1, 2]$	14	$x^5 - x - 0.2$ $x \in [1, 2]$
15	$0.1x^2 - x \ln x$ $x \in [1, 2]$	16	$\sin\left(\frac{\pi x}{5.2}\right) + 0.84e^x$ $x \in [-2, 1]$

Задание № 2

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

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

Вариант	$g(x)$	Вариант	$g(x)$
1	$x^4 - 2x^3 + x^2 - 12x + 20$	2	$x^4 + x^3 - 17x^2 - 45x - 100$
3	$x^4 + 6x^3 + x^2 - 4x - 60$	4	$x^4 - 5x^3 + x^2 - 15x + 50$
5	$x^4 - 14x^2 - 40x - 75$	6	$x^4 - 4x^3 - 2x^2 - 20x + 25$
7	$x^4 - x^3 + x^2 - 11x + 10$	8	$x^4 + 5x^3 + 7x^2 + 7x - 20$
9	$x^4 - x^3 - 29x^2 - 71x - 140$	10	$x^4 - 7x^3 + 7x^2 - 5x + 100$
11	$x^4 + 7x^3 + 9x^2 + 13x - 30$	12	$x^4 + 10x^3 + 36x^2 + 70x + 75$
13	$x^4 + 3x^3 - 23x^2 - 55x - 150$	14	$x^4 + 9x^3 + 31x^2 + 59x + 60$
15	$x^4 - 6x^3 + 4x^2 + 10x + 75$	16	$15x^4 - 6x^3 + 4x^2 - 12x - 10$

Задание № 3

Решить систему линейных уравнений:

1. матричным способом и используя функцию `lsolve`;
2. методом Гаусса;
3. используя функцию `Find`.

Вариант	Система линейных уравнений	Вариант	Система линейных уравнений
1	$\begin{cases} 2x_1 + x_2 + 2x_3 + 3x_4 = 8 \\ 3x_1 + 3x_3 = 6 \\ 2x_1 - x_2 + 3x_4 = 4 \\ x_1 + 2x_2 - x_3 + 2x_4 = 4 \end{cases}$	2	$\begin{cases} 2x_1 + x_2 - 5x_3 + x_4 = -4 \\ x_1 - 3x_2 - 6x_4 = -7 \\ 2x_2 - x_3 + 2x_4 = 2 \\ x_1 + 4x_2 - 7x_3 + 6x_4 = -2 \end{cases}$
3	$\begin{cases} x_1 + 2x_2 + 3x_3 + 4x_4 = 22 \\ 2x_1 + 3x_2 + x_3 + 2x_4 = 17 \\ x_1 + x_2 + x_3 - x_4 = 8 \\ x_1 - 2x_3 - 3x_4 = -7 \end{cases}$	4	$\begin{cases} x_1 + 2x_2 + 3x_3 + 4x_4 = 26 \\ 2x_1 + 3x_2 + 4x_3 + x_4 = 34 \\ 3x_1 + 4x_2 + x_3 + 2x_4 = 26 \\ 4x_1 + x_2 + 2x_3 + 3x_4 = 26 \end{cases}$
5	$\begin{cases} 9x_1 + 10x_2 - 7x_3 - x_4 = 23 \\ 7x_1 - x_3 - 5x_4 = 37 \\ 5x_1 - 2x_3 + x_4 = 22 \\ 4x_1 + x_2 + 2x_3 + 3x_4 = 26 \end{cases}$	6	$\begin{cases} 2x_1 - 8x_2 - 3x_3 - 2x_4 = -18 \\ x_1 - 2x_2 + 3x_3 - 2x_4 = 28 \\ x_2 + x_3 + x_4 = 10 \\ 11x_2 + x_3 + 2x_4 = 21 \end{cases}$
7	$\begin{cases} 6x_1 - x_2 + 10x_3 - x_4 = 158 \\ 2x_1 + x_2 + 10x_3 + 7x_4 = 128 \\ 3x_1 - 2x_2 - 2x_3 - x_4 = 7 \\ x_1 - 12x_2 + 2x_3 - x_4 = 17 \end{cases}$	8	$\begin{cases} 2x_1 - x_2 + 4x_3 + x_4 = 66 \\ 2x_2 - 6x_3 + x_4 = -63 \\ 8x_1 - 3x_2 + 6x_3 - 5x_4 = 146 \\ 2x_1 - 7x_2 + 6x_3 - x_4 = 80 \end{cases}$
9	$\begin{cases} x_1 - 2x_2 + 6x_3 + x_4 = 88 \\ 5x_1 + 2x_3 - 3x_4 = 88 \\ 7x_1 - 3x_2 + 7x_3 + 2x_4 = 181 \\ 3x_1 - 7x_2 + 5x_3 + 2x_4 = 99 \end{cases}$	10	$\begin{cases} 2x_1 - 3x_3 - 2x_4 = -16 \\ 2x_1 - x_2 + 13x_3 + 4x_4 = 213 \\ 3x_1 + x_2 + 2x_3 + x_4 = 72 \\ x_1 - 12x_3 - 5x_4 = -159 \end{cases}$
11	$\begin{cases} x_1 - 2x_2 - 8x_4 = -7 \\ x_1 + 4x_2 - 7x_3 + 6x_4 = -8 \\ x_1 + x_2 - 5x_3 + x_4 = -10 \\ 2x_1 - x_2 + 2x_4 = 7 \end{cases}$	12	$\begin{cases} 7x_1 + 7x_2 - 7x_3 - 2x_4 = 5 \\ 3x_1 + 4x_2 + 5x_3 + 8x_4 = 60 \\ 2x_1 + 2x_2 + 2x_3 + x_4 = 27 \\ 2x_1 - 2x_3 - x_4 = -1 \end{cases}$
13	$\begin{cases} 2x_1 + 2x_2 + 6x_3 + x_4 = 15 \\ -x_2 + 2x_3 + x_4 = 18 \\ 4x_1 - 3x_2 + x_3 - 5x_4 = 37 \\ 3x_1 - 5x_2 + x_3 - x_4 = 30 \end{cases}$	14	$\begin{cases} 6x_1 - 9x_2 + 5x_3 + x_4 = 124 \\ 7x_2 - 5x_3 - x_4 = -54 \\ 5x_1 - 5x_2 + 2x_3 + 4x_4 = 83 \\ 3x_1 - 9x_2 + x_3 + 6x_4 = 45 \end{cases}$
15	$\begin{cases} 4x_1 - 5x_2 + 7x_3 + 5x_4 = 165 \\ 2x_1 + x_2 - 3x_3 - x_4 = -15 \\ 9x_1 + 4x_3 - x_4 = 194 \\ x_1 - x_2 - 2x_3 - 3x_4 = -19 \end{cases}$	16	$\begin{cases} x_1 + 2x_2 + 2x_3 + 4x_4 = 30 \\ -x_1 + 2x_2 - 3x_3 + 4x_4 = 10 \\ x_2 - x_3 + x_4 = 3 \\ x_1 + 5x_2 + x_3 + x_4 = 10 \end{cases}$

Задание № 4

Преобразовать нелинейные уравнения системы к виду $f_1(x) = y$ и $f_2(y) = x$. Построить их графики и определить начальное приближение решения.

Решить систему нелинейных уравнений с помощью функции *Minerr*.

Вариант	Система нелинейных уравнений	Вариант	Система нелинейных уравнений
1	$\begin{cases} \sin x + 2y = 2 \\ \cos(y - 1) + x = 0,7 \end{cases}$	2	$\begin{cases} \sin y + x = -0,4 \\ 2y - \cos(x + 1) = 0 \end{cases}$
3	$\begin{cases} \sin(x + 0,5) - y = 1 \\ \cos(y - 2) + x = 0 \end{cases}$	4	$\begin{cases} \sin(x + 2) - y = 1,5 \\ \cos(y - 2) + x = 0,5 \end{cases}$
5	$\begin{cases} \cos x + y = 1,5 \\ 2x - \sin(y - 0,5) = 1 \end{cases}$	6	$\begin{cases} \cos(x + 0,5) - y = 2 \\ \sin y - 2x = 1 \end{cases}$
7	$\begin{cases} \cos(x + 0,5) + y = 0,8 \\ \sin y - 2x = 1,6 \end{cases}$	8	$\begin{cases} \cos(x - 2) + y = 0 \\ \sin(y + 0,5) - x = 1 \end{cases}$
9	$\begin{cases} \sin(x - 1) = 1,3 - y \\ x - \sin(y + 1) = 0,8 \end{cases}$	10	$\begin{cases} \cos(x + 0,5) + y = 1 \\ \sin(y + 0,5) - x = 1 \end{cases}$
11	$\begin{cases} \cos(x + 0,5) + y = 1 \\ \sin y - 2x = 2 \end{cases}$	12	$\begin{cases} \sin(x) - 2y = 1 \\ \cos(y + 0,5) - x = 2 \end{cases}$
13	$\begin{cases} -\sin(x + 1) + y = 0,8 \\ \sin(y - 1) + x = 1,3 \end{cases}$	14	$\begin{cases} 2y - \sin(x - 0,5) = 1 \\ \cos(y) + x = 1,5 \end{cases}$
15	$\begin{cases} \sin(x) - 2y = 1 \\ \sin(y - 1) + x = 1,3 \end{cases}$	16	$\begin{cases} x^2 + y^2 = 25 \\ x^2 - 1 = 0 \end{cases}$

Задание № 5

Символьно решить системы уравнений.

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

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. Как можно решить нелинейное уравнение в MathCAD?
2. Как найти начальное приближение корня уравнения?
3. Для чего используется функция *polyroots*?
4. Как можно решить систему линейных уравнений?
5. Как можно решить систему нелинейных уравнений?

Список литературы, рекомендуемый к использованию по данной теме

Основная литература:

1. Голубева, Н.В. Математическое моделирование систем и процессов. - Санкт-Петербург: Лань, 2021;
2. Казиев, В.М. Введение в анализ, синтез и моделирование систем: Учебное пособие. - Москва, Саратов: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020;
3. Поспелов, Е.А., Попов, И.С. Пакеты прикладных программ в научных исследованиях: учебно-методическое пособие. - Омск: Омский государственный университет им. Ф.М. Достоевского, 2019.

Дополнительная литература:

1. Агалаков, С.А. Анализ данных в среде R: практикум. - Омск: Омский государственный университет им. Ф.М. Достоевского, 2020;
2. Воскобойников, Ю.Е. Регрессионный анализ данных в пакете MATHCAD. - Санкт-Петербург: Лань, 2021.

Интернет-ресурсы

1. <http://biblioclub.ru> – ЭБС «Университетская библиотека on-line», <http://old.exponenta.ru/> – Образовательный математический сайт по математике и программированию
2. <http://MATLAB.exponenta.ru/Simulink/default.php> – сайт, посвященный интерактивному инструменту для моделирования Simulink,
3. <https://www.intuit.ru/> – Национальный Открытый Университет «ИНТУИТ»
4. <http://window.edu.ru/> – Информационная система «Единое окно доступа к образовательным ресурсам»
5. <http://artspb.com/> – Образовательный портал: математика, кибернетика и программирование дополнительного образования
6. <http://www.arenasimulation.com> – Официальный сайт разработчика среды имитационного моделирования ARENA

<https://www.anylogic.ru> – Официальный сайт разработчика AnyLogic

Лабораторная работа 11. Символьные вычисления

Цель лабораторной работы:

сформировать способности осуществлять символьные вычисления в системе MathCAD

Организационная форма занятий: решение проблемных задач, разбор конкретных ситуаций

Вопросы для обсуждения на лабораторном занятии: символьные вычисления в системе MathCAD

Формируемые компетенции: ОПК-2, ОПК-3, ОПК-4, ОПКД-2, ОПКД-3, ОПКД-4.

Общие сведения

Символьные операции можно выполнять двумя способами:

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

Символы



Выделение выражений для символьных вычислений

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

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

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

Символьные операции

Операции с выделенными выражениями

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

Расчеты - преобразовать выражение с выбором вида преобразований из подменю:

- Символические [Shift] + F9 - выполнить символьное преобразование выделенного выражения;
- С плавающей запятой... - вычислить выделенное выражение в вещественных числах;
- Комплексные – выполнить вычисления в комплексном виде;

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

Развернуть - раскрыть выражение (например, для $(X + Y)(X - Y)$ получаем $X^2 - Y^2$);

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

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

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

Операции с выделенными переменными

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

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

Замена - заменить указанную переменную содержимым буфера обмена;

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

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

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

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

Операции с выделенными матрицами

Операции с выделенными матрицами представлены командой меню **Символы → Матрицы**, которая имеет свое подменю со следующими операциями:

Транспонирование - получить транспонированную матрицу;

Инвертирование - создать обратную матрицу;

Определитель - вычислить детерминант (определитель) матрицы.

Операции преобразования

В позиции **Конвертировать** содержится раздел операций преобразования со следующими возможностями:

Фурье - выполнить прямое преобразование Фурье относительно выделенной переменной;

Фурье Обратное - выполнить обратное преобразование Фурье относительно выделенной переменной;

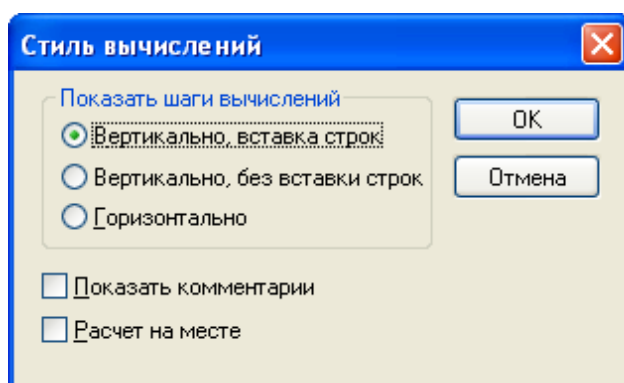
Лапласа - выполнить прямое преобразование Лапласа относительно выделенной переменной (результат - функция переменной s);

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

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

Обратное Z - выполнить обратное Z - преобразование относительно выделенной переменной (результат - функция переменной n) .

Стиль представления результатов вычислений



На наглядность вычислений влияет стиль представления их результатов. Следующая команда позволяет задать тот или иной стиль: **Символы → Стиль Вычислений...**

Символьные вычисления

$$\pi \quad \sqrt{-3}$$

$$3.14159265359 \quad \frac{1}{i \cdot 3^2}$$

$$\frac{d}{dx} \cos(x)^2 \rightarrow -2 \cdot \cos(x) \cdot \sin(x) \quad \sum_{i=1}^{10} i \rightarrow 55$$

$$\int \sin(2 \cdot x) \, dx \rightarrow \frac{-1}{2} \cdot \cos(2 \cdot x)$$

$$\cos(4 \cdot x) \text{ expand } , x \rightarrow 8 \cdot \cos(x)^4 - 8 \cdot \cos(x)^2 + 1$$

$$\cos(4 \cdot x)$$

$$8 \cdot \cos(x)^4 - 8 \cdot \cos(x)^2 + 1$$

$$\sqrt{3.01} \text{ simplify } \rightarrow 1.7349351572897472412$$

Выполнить задание

Задание № 1. Используя операцию Символы → Расчеты → С плавающей запятой..., представьте:

- 1) число π в семи позициях;
- 2) число 12,345667 в трех позициях.

Задание № 2. Выведите следующие числа в комплексной форме, используя операцию Символы → Расчеты → Комплексные:

$$1) \sqrt{-7};$$

$$2) e^{1 + \frac{\pi}{4}i};$$

3) для выражения 2) последовательно выполните операции Расчеты → Комплексные и Символы → Упростить.

Задание № 3. Для полинома $g(x)$ выполнить следующие действия:

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

Вариант	$g(x)$	Вариант	$g(x)$
---------	--------	---------	--------

1.	$x^4 - 2x^3 + x^2 - 12x + 20$	2.	$x^4 + x^3 - 17x^2 - 45x - 100$
3.	$x^4 + 6x^3 + x^2 - 4x - 60$	4.	$x^4 - 5x^3 + x^2 - 15x + 50$
5.	$x^4 - 14x^2 - 40x - 75$	6.	$x^4 - 4x^3 - 2x^2 - 20x + 25$
7.	$x^4 - x^3 + x^2 - 11x + 10$	8.	$x^4 + 5x^3 + 7x^2 + 7x - 20$
9.	$x^4 - x^3 - 29x^2 - 71x - 140$	10.	$x^4 - 7x^3 + 7x^2 - 5x + 100$
11.	$x^4 + 7x^3 + 9x^2 + 13x - 30$	12.	$x^4 + 10x^3 + 36x^2 + 70x + 75$
13.	$x^4 + 3x^3 - 23x^2 - 55x - 150$	14.	$x^4 + 9x^3 + 31x^2 + 59x + 60$
15.	$x^4 - 6x^3 + 4x^2 + 10x + 75$	16.	$15x^4 - 6x^3 + 4x^2 - 12x - 10$

Задание № 4. Разложите выражения на элементарные дроби используя операцию Символы → Переменные → Конвертировать в частичные доли:

$$\begin{array}{ll}
 1) \quad \frac{6x^2 - x + 1}{x^3 - x}; & 2) \quad \frac{3x^2 - 2}{(x^2 + x + 1)(x + 1)}; \\
 3) \quad \frac{x + 1}{x(x - 1)^3}; & 4) \quad \frac{5x^2 - 4x + 16}{(x^2 - x + 1)^2(x - 3)}
 \end{array}$$

Задание № 5. Разложите выражения в ряд с заданной точностью, используя операцию Символы → Переменные → Разложить...:

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

Задание № 6. Найти первообразную аналитически заданной функции $f(x)$ используя команду Символы → Переменные → Интеграция.

Задание № 7. Определить символьное значение первой и второй производных $f(x)$, используя команду Символы → Переменные → Дифференциалы.

Варианты заданий № 6 и № 7

Вариант	$f(x)$	Вариант	$f(x)$	Вариант	$f(x)$
1	$\frac{1}{\lg 2x + 1}$	6	$x^2 \arctg\left(\frac{x}{3}\right)$	11	$(2x + 3) \sin x$
2	$\frac{\cos x}{2x + 5}$	7	$e^{2x} \sin 3x$	12	$\frac{\cos 3x}{(1 - \cos 3x)^2}$

3	$\frac{1}{x\sqrt{x^3+4}}$	8	$\frac{\operatorname{ctg} 2x}{\sin^2 2x}$	13	$\frac{1}{1+x+x^2}$
4	$\frac{\sin x}{1+\sin x}$	9	$(x+1)\sin x$	14	$\frac{1+x}{2+x}$
5	$x^2 \lg(x+2)$	10	$5x+x\lg x$	15	$\sqrt{1+e^{-x}}$

Задание № 8.

1. Транспонируйте матрицу M с помощью операции Символы $\rightarrow$ Матрицы $\rightarrow$ Транспонирование.

$$\begin{pmatrix} 1 & a & b \\ x & 2 & c \\ x^2 & 3 & d \end{pmatrix}$$

2. Инвертируйте матрицу $\begin{pmatrix} 1 & y \\ x & 2 \end{pmatrix}$ с помощью операции Символы $\rightarrow$ Матрицы $\rightarrow$ Инвертирование.

3. Вычислите определитель матрицы $\begin{pmatrix} 1 & a & b \\ x & 2 & c \\ x^2 & 3 & d \end{pmatrix}$ с помощью операции Символы $\rightarrow$ Матрицы $\rightarrow$ Определитель.

Задание № 9. Вычислите пределы.

1. $\lim_{x \rightarrow 4} \frac{\sqrt{1+2x}-3}{\sqrt{x}-2}.$

2. $\lim_{x \rightarrow -8} \frac{\sqrt{1-x}-3}{2+\sqrt[3]{x}}.$

3. $\lim_{x \rightarrow 1} \frac{x^2-1}{\ln x}.$

4. $\lim_{x \rightarrow 1} \frac{\sqrt{x^2-x+1}-1}{\ln x}.$

Задание № 10. Найдите сумму ряда.

1. $\sum_{n=1}^{\infty} \frac{6}{9n^2+12n-5}.$

2. $\sum_{n=2}^{\infty} \frac{24}{9n^2-12n-5}$

Задание № 11. Найдите производную и упростите выражение.

$$1. y = \frac{4x+1}{16x^2+8x+3} + \frac{1}{\sqrt{2}} \operatorname{arctg} \frac{4x+1}{\sqrt{2}}.$$

$$2. y = \frac{2}{x-1} \sqrt{2x-x^2} + \ln \frac{1+\sqrt{2x-x^2}}{x-1}.$$

Задание № 12. Вычислите неопределенные интегралы.

$$1. \int (3x+4)e^{3x} dx.$$

$$2. \int (4x-2)\cos 2x dx.$$

Задание № 13. Вычислите определенные интегралы.

$$1. \int_{-2}^0 (x^2 + 5x + 6)\cos 2x dx.$$

$$2. \int_{-2}^0 (x^2 - 4)\cos 3x dx.$$

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. Опишите способы выполнения символьных операций
2. Перечислите символьные операции.

Список литературы, рекомендуемый к использованию по данной теме

Основная литература:

1. Голубева, Н.В. Математическое моделирование систем и процессов. - Санкт-Петербург: Лань, 2021;
2. Казиев, В.М. Введение в анализ, синтез и моделирование систем: Учебное пособие. - Москва, Саратов: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020;
3. Поспелов, Е.А., Попов, И.С. Пакеты прикладных программ в научных исследованиях: учебно-методическое пособие. - Омск: Омский государственный университет им. Ф.М. Достоевского, 2019.

Дополнительная литература:

1. Агалаков, С.А. Анализ данных в среде R: практикум. - Омск: Омский государственный университет им. Ф.М. Достоевского, 2020;
2. Воскобойников, Ю.Е. Регрессионный анализ данных в пакете MATHCAD. - Санкт-Петербург: Лань, 2021.

Интернет-ресурсы

1. <http://biblioclub.ru> – ЭБС «Университетская библиотека on-line», <http://old.exponenta.ru/> – Образовательный математический сайт по математике и программированию
2. <http://MATLAB.exponenta.ru/Simulink/default.php> – сайт, посвященный интерактивному инструменту для моделирования Simulink,
3. <https://www.intuit.ru/> – Национальный Открытый Университет «ИНТУИТ»
4. <http://window.edu.ru/> – Информационная система «Единое окно доступа к образовательным ресурсам»

5. <http://artspb.com/> – Образовательный портал: математика, кибернетика и программирование дополнительного образования
6. <http://www.arenasimulation.com> – Официальный сайт разработчика среды имитационного моделирования ARENA
<https://www.anylogic.ru> – Официальный сайт разработчика AnyLogic

Часть 3. Свободная программная среда R

Общие сведения о свободной программной среде R

Данный практикум призван познакомить читателя с некоторыми особенностями и возможностями популярного и активно развивающегося программного средства – языка R.

Классическая статистика в основном основывается на проверке статистических гипотез - априорных предположений о свойствах исследуемых данных.

Появление компьютеров позволило существенно повысить эффективность вычислений. Но в первое время, пока компьютеры были еще очень слабы в области отображения информации, особенно графической, практика статистики, как и подготовка статистиков, ориентировалась на построение моделей и на проверку гипотез.

Подобный подход реализован в программных средствах, подобных SPSS, которые базируются на электронных таблицах и управляются с помощью меню.

Фактически первые версии программных продуктов SPSS и SAS Analytics состояли из подпрограмм, которые можно было вызвать из основной программы (на Fortran или на другом языке) с целью подгонки и проверки модели из имеющегося набора моделей.

Альтернативой этому подходу стала концепция разведочного анализа данных (Exploratory Data Analysis, EDA), предложенная Джоном Тьюки (John Tukey). Разведочный анализ данных применяется для нахождения связей между переменными в ситуациях, когда отсутствуют (или не достаточны) априорные представления о природе этих связей. Как правило, при разведочном анализе учитывается и сравнивается большое число переменных, а для поиска закономерностей используются самые разные методы.

R соединяет в себе эти концепции, позволяя использовать как основные статистические методы, так и более сложные, специально разработанные методы многомерного анализа, предназначенные для отыскания закономерностей в многомерных данных.

R, в отличие от SAS и SPSS, это универсальный язык программирования, разработанный для применения в таких областях, как разведочный анализ данных, классические статистические тесты и высокоуровневая графика.

R обладает рядом существенных достоинств.

R – свободная программная среда вычислений с открытым исходным кодом. Это реализация языка S с дополнительными моделями, разработанными в языке S-Plus.

В некоторых случаях моделями в обоих языках занимались одни и те же люди. R доступен в соответствии с лицензией GNU.

На этом фундаменте R продолжает развиваться, в значительной степени посредством добавления пакетов.

R – пакет представляет собой коллекцию наборов данных, функций языка R, документации и динамически загружаемых элементов на языке C или Fortran. Посредством этих пакетов исследователи могут с легкостью обмениваться вычислительными методами со своими коллегами.

R уже имеет собственную обширную и непрерывно расширяющуюся библиотеку пакетов, содержащую большое количество готовых решений различных задач. Некоторые пакеты имеют ограниченную область применения, другие представляют целые области статистики, а некоторые отражают новейшие разработки. Многие новые разработки в области статистики, эконометрики, биологии, химии, медицины, географии и других прикладных областей сначала появляются как R-пакеты, и только потом реализуются в коммерческих программных продуктах.

R – это мощный скриптовый язык. Скриптом называется программный сценарий, любая исполняемая процедура, которая запускается автоматически или же с помощью команды пользователя. Скрипты используются не только в программировании, область их применения шире. Скрипт может быть использован для повторяемых и сложных для запоминания пользователю операций.

Для обработки неупорядоченных данных требуются возможности языка программирования: продукты SAS и SPSS имеют скриптовые языки для решения отдельных задач, однако R был создан именно как язык программирования и поэтому является более подходящим средством для этой цели.

Таким образом, R является полезным инструментом в области анализа больших массивов данных. Он уже интегрирован в ряд коммерческих пакетов, таких как IBM SPSS и InfoSphere, а также Mathematica. R - язык, ориентированный на статистику, который можно рассматривать в качестве конкурента для таких аналитических систем, как SAS Analytics, не говоря уже о таких более простых пакетах, как StatSoft STATISTICA или Minitab.

R органично интегрируется с системами публикации документов, что позволяет встраивать статистические результаты и графику из среды R в документы публикационного качества.

Эта возможность нужна не всем, однако если вы хотите написать книгу о своем анализе данных или просто не любите копировать свои результаты в документы текстового процессора, то самый короткий и самый элегантный маршрут состоит в использовании R. В качестве языка программирования R подобен многим другим языкам. Любый человек, который когда-либо писал программный код, найдет в R множество знакомых моментов. Отличительные особенности R лежат в статистической философии, которую он исповедует. Язык R имеет легкий синтаксис - это универсальный инструмент, разработанный специально для работы с данными. R также включает в себя чрезвычайно мощные графические возможности.

В данном практикуме рассматривается работа с R на операционной системе Windows. Для выполнения практикума читателю необходимо установить:

1. Язык R – классический <https://cran.rstudio.com/bin/windows/base/>; или PRO <https://mran.revolutionanalytics.com/download/>.

2.

Графический интерфейс пользователя RStudio: <https://www.rstudio.com/products/rstudio/download/> – набор интегрированных инструментов, разработанных, чтобы наиболее продуктивно использовать R.

RStudio включает в себя консоль, редактор с подсветкой синтаксиса, поддерживающий как прямое выполнение кода, так и инструменты для построения графиков, сохранение истории команд, отладку и управление рабочим пространством

Для работы в R потребуется стабильное Интернет-соединение, т. к. может потребоваться онлайн-загрузка дополнительных пакетов для обработки данных.

Англоязычных источников по R чрезвычайно много. При возникновении вопросов, ответов на которые нет в справке R, можно использовать следующие ресурсы:

– <http://rseek.org/> - Google, модифицированный под поиск по источникам, связанным с R;

– <http://stats.stackexchange.com/> - обращаться с вопросами по поводу статистических методов и их реализации в R;

– <http://stackoverflow.com/> - обращаться с вопросами по поводу программирования в R;

– <http://vk.com/rstatistics> - группа в социальной сети «ВКонтакте», участники которой могут ответить на возникшие вопросы;

– <http://r-analytics.blogspot.ru/> - один из самых масштабных русскоязычных проектов по R.

Лабораторные работы

Лабораторная работа 12. Основы работы с R. Обработка статистических данных

Цель лабораторной

работы: ознакомиться с интерфейсом RStudio, научиться работать в режиме консоли и путем написания скриптов, а также подключать внешние пакеты, изучить основные методы обработки статистических данных.

Организационная форма занятий: решение проблемных задач, разбор конкретных ситуаций

Вопросы для обсуждения на лабораторном занятии: Основы работы с R. Обработка статистических данных.

Формируемые компетенции: ОПК-2, ОПК-3, ОПК-4, ОПКД-2, ОПКД-3, ОПКД-4.

Задание 1. Загрузить данные для своего варианта в переменную – вектор.

2. Получить справочную информацию по своим данным, просмотреть их содер-жимое.

3. Проверить, есть ли среди данных пропуски.

4. Создать новую переменную – вектор, в которой будут 1, если значение в исходном векторе больше среднего, и –1, если значение переменной меньше среднего, и 0, если значение равно среднему.

5. Вывести описательную статистику.

6. Построить графики абсолютных частот и плотности распределения.

Указания к выполнению работы

Консольный режим

После запуска RStudio пользователь попадает в консольный режим работы (рис. 1). Любая команда, написанная пользователем, будет сразу выполнена R по нажатию Enter.

Рассмотрим основные выражения в R: числа, строки и логические переменные.

Можно использовать R как калькулятор, например:

```
> 1 + 1
```

```
[1] 2
```

```
> 6 * 7
```

```
[1] 42
```

```
> sqrt(16)
```

```
[1] 4
```

Результат сразу появится в консоли.

Строки печатаются в кавычках: двойных или одинарных:

```
> "Hello world!"
```

```
[1] "Hello world!"
```

```
> 'Hello world!'
```

```
[1] 'Hello world!'
```

Логические выражения возвращают TRUE или FALSE:

```
> 3 < 4
```

```
[1] TRUE
```

Чтобы сравнить два выражения, используется двойной знак равенства:

```
> 2 + 2 == 5
```

```
[1] FALSE
```

Как и в других языках программирования, можно сохранять значения в переменную. Сохраним 42 в переменную x:

```
> x <- 42
```

И в обратную сторону:

```
> 5 -> x
```

Можно распечатать значение переменной в любое время, просто набрав ее имя в консоли. Попробуем напечатать текущее значение x:

```
> x
```

[1] 42

Можно так же повторно назначить любое значение переменной в любое время.

R чувствителен к регистру: переменные x и X – это разные переменные:

```
> X
```

[1] 5

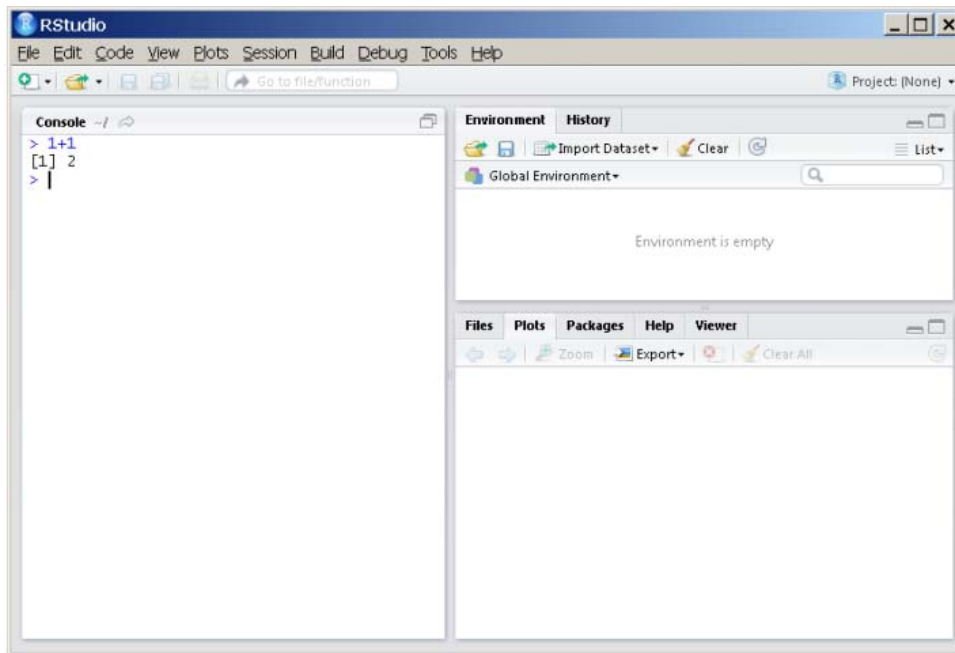


Рисунок 1 – Консольный режим в RStudio

Чтобы вызвать функцию, нужно обратиться к ней по имени, указав в скобках нужные аргументы. Например, функция суммы:

```
> sum(1, 3, 5)
```

[1] 9

Получить помощь по функции можно командой `help(functionname)` или `?functionname`.

В правом нижнем углу на вкладке Help появится справка (рис. 2):

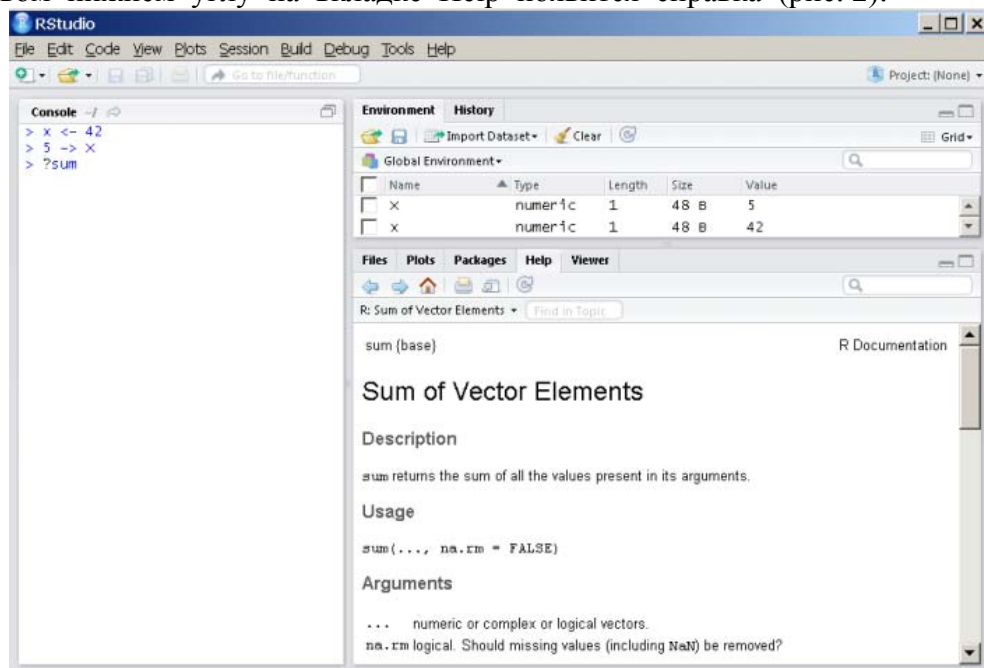


Рисунок 2 – Справка по функции sum

Зададим вектор с помощью функции `c` (сокр. от англ. Combine):

```
> y <- c(-3, 2, NA, 5)
```

```
> y
```

[1] -3 2 NA 5

NA – это пропущенное наблюдение (от англ. Not Available). Его не следует путать с NaN (Not a Number — «не число», неопределенность):

```
> 0/0
```

```
[1] NaN
```

Попробуем просуммировать элементы вектора y:

```
> sum(y)
```

```
[1] NA
```

Необязательным аргументом функции sum является na.rm (сокр. от англ. Remove NA), по умолчанию равный FALSE.

Если указать для него значение «истина», то функция суммы будет складывать все элементы вектора, исключая пропущенные:

```
> sum(y, na.rm = TRUE)
```

```
[1] 4
```

Последовательность чисел можно задать двумя способами: start:end либо функцией seq():

```
> 5:9
```

```
[1] 5 6 7 8 9
```

```
> seq(5,9)
```

```
[1] 5 6 7 8 9
```

```
> seq(10,50, by = 10)
```

```
[1] 10 20 30 40 50
```

Обращаться к элементам вектора можно, используя квадратные скобки:

```
> sentence <- c('mack', 'the', 'knife')
```

```
> sentence[3]
```

```
[1] "knife"
```

```
> sentence[c(1,3)]
```

```
[1] "mack" "knife"
```

Либо можно задать элементам вектора имена:

```
> ranks <- 1:3
```

```
> names(ranks) <- c("first", "second", "third")
```

```
> ranks
```

```
first second third
```

```
1      2      3
```

```
> ranks["first"]
```

```
first
```

```
1
```

Написание скриптов

В R удобнее писать не по одной команде, а сразу целый набор команд и потом запускать их все на выполнение. Для этого нужны скрипты. Чтобы создать скрипт, следует выбрать File –> New File –> R Script. Откроется новая область, в которой можно писать команды. Комментарии, которые не будет выполнять R, пишутся со знака # (рис. 3).

По нажатию Enter команды в скрипте выполняться не будут, а будет лишь осуществлен переход на новую строку.

Чтобы выполнить команду в режиме скрипта, следует поставить курсор на нужную строку и нажать Ctrl+Enter. Если команда занимает более одной строки, то необходимо ставить знак + в конце каждой строки. Команда выполняется построчно, в каждой строке требуется нажимать Ctrl+Enter, либо выделить всю команду целиком и нажать Ctrl+Enter один раз.

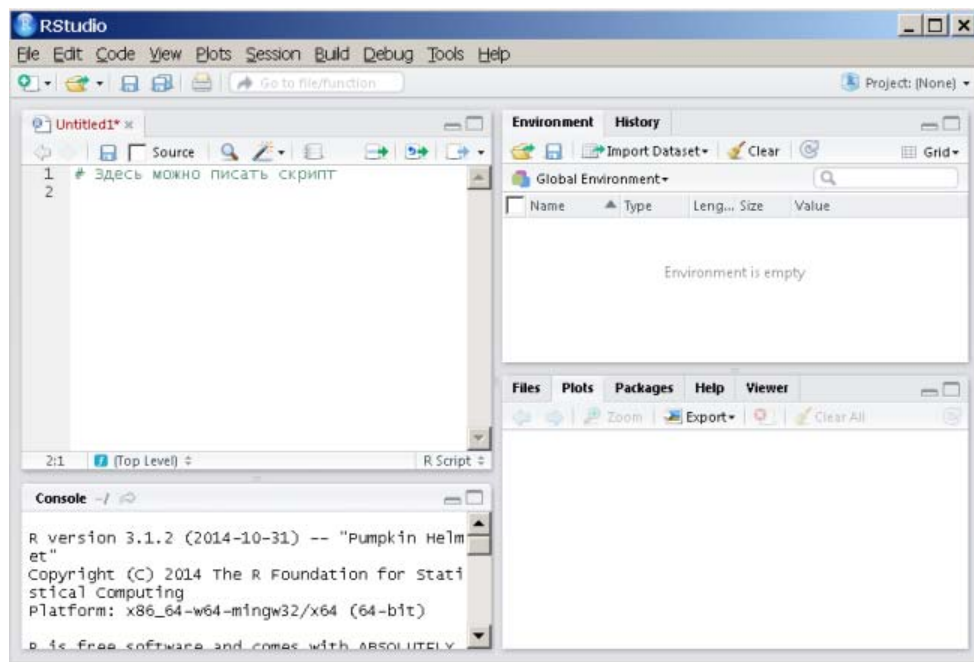


Рисунок 3 – создание нового скрипта

R может подсказывать, какие команды доступны, если начать вводить первые символы и нажать либо Tab, либо Ctrl+Enter.

В основном в R работают с наборами данных. Такая структура носит в R название data.frame и представляет собой таблицу, в которой каждый столбец – это переменная, а каждая строка – это одно наблюдение.

Создадим в режиме скрипта data.frame. Пусть имеются наблюдения за ростом и весом некоторых людей. Зададим два вектора:

```
rost <- c(160, 175, 155, 190, NA)
ves <- c(NA, 70, 48, 85, 60)
```

И объединим их в набор данных, который поместим в переменную df, а затем выведем на экран:

```
df <- data.frame(rost, ves)
```

В консоли получим следующую таблицу:

```
df
  rost ves
1 160 NA
2 175 70
3 155 48
4 190 85
5 NA 60
```

Обращаться к конкретным наблюдениям df можно, используя квадратные скобки:

```
> df[3,1] [1] 155
```

Обращаться к переменным можно, используя знак \$ или указывая столбец с пропуском номера строки:

```
> df$rost
[1] 160 175 155 190 NA
или
```

```
> df[,1]
[1] 160 175 155 190 NA
```

Обращаться к наблюдениям можно, указывая конкретную строку и пропуская номер столбца:

```
> df[4,]
  rost ves
4 190 85
```

Основные описательные статистики (среднее, стандартное отклонение и медиану) можно получить с помощью функций mean, sd и median:

```
mean(df$rost, na.rm = T)
```

```
[1] 170
sd(df$rost, na.rm = T)
[1] 15.81139
median(df$rost, na.rm = T)
[1] 167.5
```

Можно сохранить скрипт, нажав File → Save. При первом сохранении R предложит выбрать кодировку. Рекомендуется указать UTF-8 (рис. 4), чтобы русские буквы (например, в комментариях) отображались корректно. Затем необходимо выбрать директорию и задать имя файла, который будет сохранен с расширением *.R.

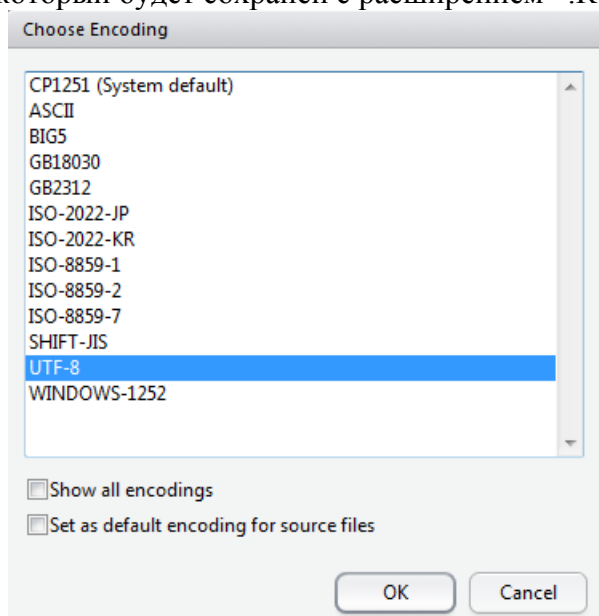


Рисунок 4 – Выбор кодировки при сохранении

Установка пакетов

В R существует базовый набор пакетов (библиотек), содержащих самые необходимые функции. Для реализации задач эконометрики требуются дополнительные пакеты.

Для работы с внешними пакетами необходимо выполнить два действия – установку и подключение нужного пакета. Установка необходимо выполнить один раз, а подключать – в каждой рабочей сессии.

Для установки пакетов существует функция `install.packages`. Также автоматически будут доустановлены связанные пакеты.

Для подключения установленного пакета следует воспользоваться функцией `library`.

Например, установим следующие пакеты:

`psych` – содержит функции для расчета описательных статистик;

`dplyr` – содержит функции для работы с `data.frame`;

`ggplot2` – самый мощный пакет для построения красивых графиков, диаграмм, карт и т. д.

`install.packages(c("psych", "dplyr", "ggplot2"))`

Прямым сообщением об ошибке установки является только слово `Error`, появляющееся в консоли. Все остальные сообщения `Warning` являются просто предупреждениями о чем-либо.

Для того чтобы определить, какие пакеты нужны для работы, можно воспользоваться поиском в сети Интернет, задав вопрос на английском языке. Например, чтобы найти, в каком пакете находится алгоритм Левенберга-Марквардта для расчета нелинейного МНК, можно набрать в поиске «`levenberg-marquardt algorithm in r`» и первой же ссылкой будет пакет `minpack.lm` в R.

Для выполнения данной работы понадобится подключить следующие пакеты:

<code>library("psych")</code> # описательные статистики	<code>library("lmtest")</code> # тестирование гипотез в
линейных	моделях
<code>library("ggplot2")</code> #	графики
<code>library("dplyr")</code> #	манипуляции с
<code>library("MASS")</code> # подгонка распределений	данными

Получим, например, описание набора данных по автомобилям `cars` командой:

`help(cars)`

Результат выполнения команды (в правом нижнем углу на вкладке Help) показан на рисунке 5.

В этом наборе данных 50 наблюдений и две переменных (скорость, миль/час и длина тормозного пути в футах).

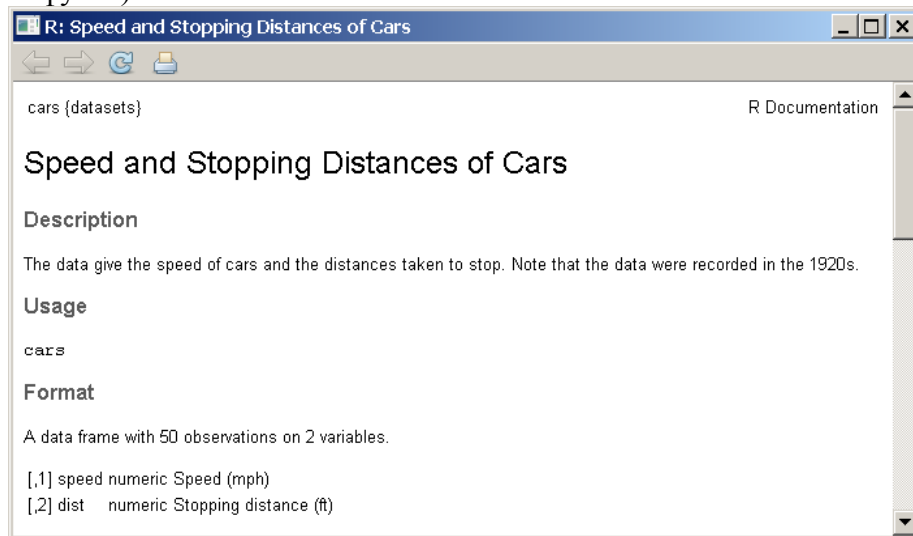


Рисунок 5 – Справка по набору данных cars

Поместим в переменную d встроенный в R набор данных по автомобилям1:

```
d <- cars # этот набор данных находится в базовом пакете datasets
```

Теперь d имеет тип данных data.frame (набор данных), в чем можно удостовериться, посмотрев в правом верхнем углу окна табулицу среды Environment (рисунок 6).

Для этого должен быть выбран режим Grid. С помощью кнопки  можно просмотреть содержимое набора данных в виде таблицы.

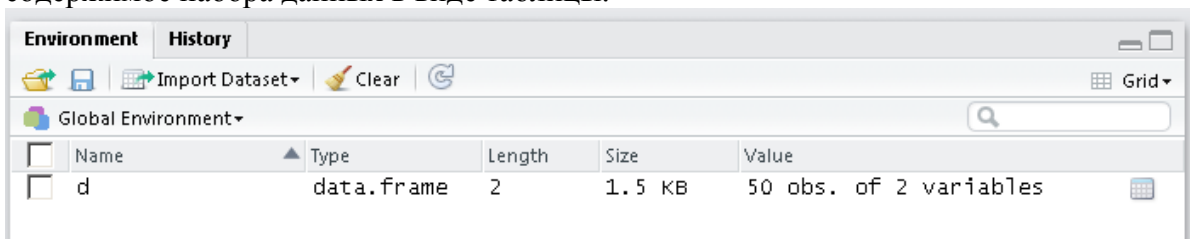


Рисунок 6 – Описание набора данных d в таблице среды Environment

Следующей командой можно посмотреть на этот набор данных, в результате чего будут перечислены все переменные и типы данных:

```
glimpse(d) # функция из пакета dplyr
```

Результат выполнения команды появится в консоли:

```
>d<-cars
> glimpse(d)
Observations: 50
Variables: 2
 $ speed (dbl) 4, 4, 7, 7, 8, 9, 10, 10, 10, 11, 11, 12, 12, 12, 13,...
 $dist (dbl) 2,10,4,22,16,10,18,26,34,17,28,14,20,24,28,...
```

Переменные speed и dist имеют тип данных dbl (double) и содержат по 50 наблюдений. Для других типов данных используются следующие сокращения: chr (character/string), int (integer), fctr (factor), tims (time), lgl (logical).

Посмотрим на первые шесть наблюдений набора данных d:

```
> head(d) # функция из базового пакета utils
  speed dist
1     4    2
2     4   10
3     7    7
4     7   22
5     8   16
```

6 9 10

и последние шесть наблюдений:

```
> tail(d) # функция из базового пакета utils speed dist
```

```
45 23 54
```

```
46 24 70
```

```
47 24 92
```

```
48 24 93
```

```
49 24 120
```

```
50 25 85
```

Получим таблицу с описательными статистиками: среднее, мода, медиана, стандартное отклонение, минимум/максимум, асимметрия, эксцесс и т. д.:

```
> describe(d) # функция из пакета psych
```

```
vars n mean sd median trimmed mad min max range skew kurtosis
```

```
speed 1 50 15.40 5.29 15 15.47 5.93 4 25 21 -0.11 -0.67
```

```
dist 2 50 42.98 25.77 36 40.88 23.72 2 120 118 0.76 0.12
```

```
se
```

```
speed 0.75
```

```
dist 3.64
```

Построим гистограмму абсолютных частот для переменной dist (длины тормозного пути).

Воспользуемся функцией qplot, задав источник данных d (аргумент data), переменную для построения графика (dist), подпишем оси (параметры функции xlab и ylab) и название графика (параметр main):

```
# функция из пакета ggplot2
```

```
qplot(data=d, dist, xlab="Длина тормозного пути (футы)", ylab="Число автомобилей", main="Данные по автомобилям 1920х")
```

Результат выполнения данной функции показан на рисунке 7.

Можно построить также гистограмму плотности распределения (рисунок 8):

```
# функция из базового пакета graphics h
```

```
hist(d$dist, probability = TRUE, col="grey")
```

Подгоним плотность распределения Вейбулла, поместив результат (оценки параметров распределения) в переменную fit: fit <- fitdistr(d\$dist, "weibull") # функция из пакета MASS

Переменная fit теперь представляет собой список (List) из 5 элементов, что можно увидеть в Environment (рисунок 9).

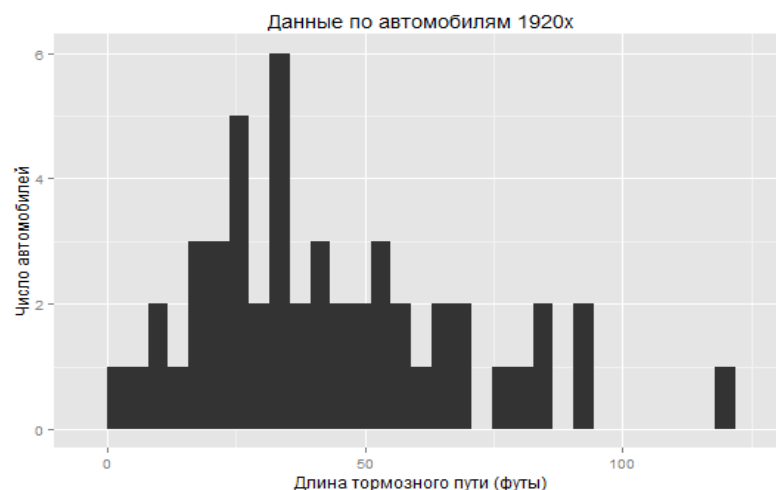


Рисунок 7 – Гистограмма абсолютных частот для переменной dist

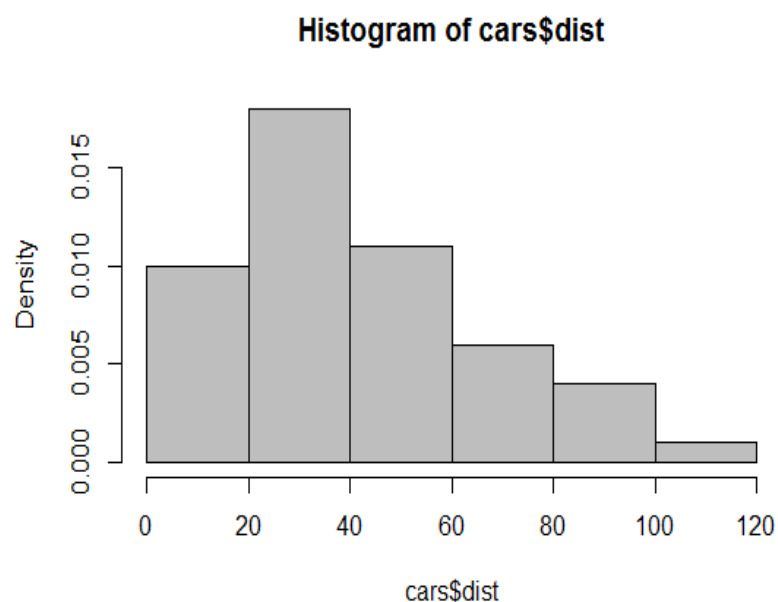


Рисунок 8 – Гистограмма плотности распределения для переменной dist

Environment History					
Global Environment					
Name	Type	Length	Size	Value	
d	data.frame	2	1.5 KB	50 obs. of 2 variables	
fit	fitdistr	5	2.1 KB	List of 5	

Рисунок 9 – Описание переменной fit в таблице среды Environment

Доступ к элементам списка можно получить через значок доллара \$.

Оценки двух параметров распределения Вейбулла были рассчитаны методом максимального правдоподобия.

Посмотрим их, обратившись к элементу списка fit:

```
> fit$estimate shape scale 1.72371 48.15234
```

Покажем на том же графике теоретическую плотность распределения Вейбулла (рисунок 10):

```
xvals <- seq(0, 120, .20) # значения по оси абсцисс от 0 до 120 с шагом 0,2
```

```
lines(xvals, dweibull(xvals, shape=fit$estimate[1],
scale=fit$estimate[2]))
```

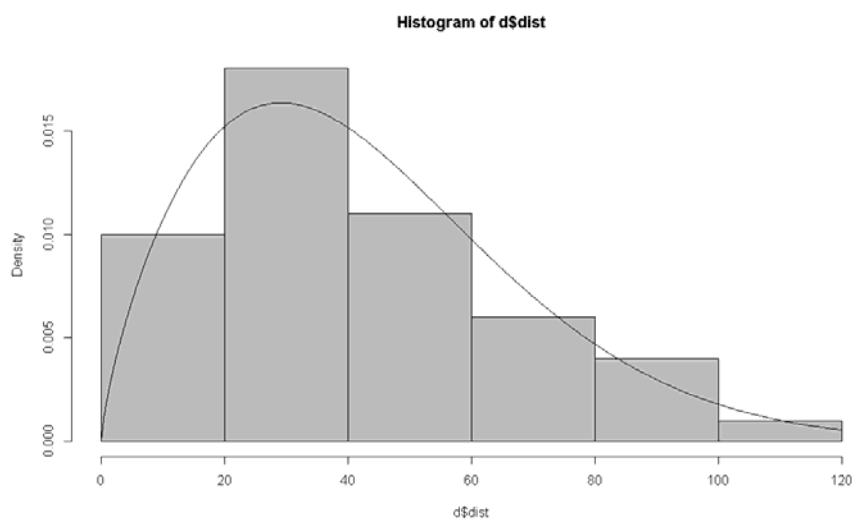


Рисунок 10 – Гистограмма распределения переменной dist

Первый аргумент функции lines – это значения по оси абсцисс, на основе которых будет построен график.

Далее указывается функция плотности `dweibull`. Для нее нужно указать значения аргумента для расчета и значения двух параметров распределения: коэффициент формы (`shape`) и масштаба (`scale`).

Варианты заданий

№ варианта	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.
Набор данных	CO2		ChickWeight		Orange		airquality		faithful	
Имя переменной (вектора)	conc	uptake	weight	Time	age	circumference	Wind	Temp	eruptions	waiting

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. Как найти среднее выборочное значение и стандартное отклонение?
2. Как проверить наличие пропусков в исходных данных?
3. Каким образом функция `qplot` разбивает выборку на интервалы?

Список литературы, рекомендуемый к использованию по данной теме

Основная литература:

1. Голубева, Н.В. Математическое моделирование систем и процессов. - Санкт-Петербург: Лань, 2021;
2. Казиев, В.М. Введение в анализ, синтез и моделирование систем: Учебное пособие. - Москва, Саратов: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020;
3. Поспелов, Е.А., Попов, И.С. Пакеты прикладных программ в научных исследованиях: учебно-методическое пособие. - Омск: Омский государственный университет им. Ф.М. Достоевского, 2019.

Дополнительная литература:

1. Агалаков, С.А. Анализ данных в среде R: практикум. - Омск: Омский государственный университет им. Ф.М. Достоевского, 2020;
2. Воскобойников, Ю.Е. Регрессионный анализ данных в пакете MATHCAD. - Санкт-Петербург: Лань, 2021.

Интернет-ресурсы

1. <http://biblioclub.ru> – ЭБС «Университетская библиотека on-line», <http://old.exponenta.ru/> – Образовательный математический сайт по математике и программированию
2. <http://MATLAB.exponenta.ru/Simulink/default.php> – сайт, посвященный интерактивному инструменту для моделирования Simulink,
3. <https://www.intuit.ru/> – Национальный Открытый Университет «ИНТУИТ»
4. <http://window.edu.ru/> – Информационная система «Единое окно доступа к образовательным ресурсам»

5. <http://artspb.com/> – Образовательный портал: математика, кибернетика и программирование дополнительного образования
6. <http://www.arenasimulation.com> – Официальный сайт разработчика среды имитационного моделирования ARENA
- <https://www.anylogic.ru> – Официальный сайт разработчика AnyLogic

Лабораторная работа 13. Линейная регрессия

Цель лабораторной работы:

Изучить приемы исследования корреляционной зависимости, построения парной и множественной линейной регрессии.

Организационная форма занятий: решение проблемных задач, разбор конкретных ситуаций

Вопросы для обсуждения на лабораторном занятии: Линейная регрессия.

Формируемые компетенции: ОПК-2, ОПК-3, ОПК-4, ОПКД-2, ОПКД-3, ОПКД-4.

Задание

1. Загрузить набор данных для своего варианта, ознакомиться с его содержимым.
2. Построить график корреляционного поля для каждого фактора.
3. Построить уравнение парной линейной регрессии для каждого фактора.
4. Проверить значимость каждого из полученных уравнений регрессии. Показать уравнения регрессии с заданным в варианте доверительным интервалом на графиках.
5. Построить прогнозы по каждому из уравнений парной регрессии для заданных в варианте значений факторов.
6. Построить уравнение множественной линейной регрессии и получить корреляционную матрицу.
7. Построить прогноз по уравнению множественной регрессии для заданных в варианте значений факторов.

Указания к выполнению работы

Парная линейная регрессия

Рассмотрим построение парной линейной регрессии на встроенном наборе данных cars.

```
d <- cars
```

Будем рассматривать зависимость длины тормозного пути (переменная dist) от скорости (переменная speed).

Построим график зависимости длины тормозного пути от скорости автомобиля:

```
qplot(data=d, speed, dist)
```

Чтобы настроить внешний вид графика, необходимо использовать функцию ggplot (рисунок 11).

```
ggplot() +  
geom_point(aes(x=d$speed, y=d$dist), size = 2) + theme_bw(base_size = 18) +  
xlab("Скорость, миль/ч") + ylab("Длина тормозного пути, футы") +  
labs(title = "Корреляционное поле")
```

Данная функция имеет множество других настроек, с которыми можно ознакомиться в справке. Оценим модель линейной регрессии длины тормозного пути на скорость автомобиля.

Для этого командой lm поместим в переменную model модель линейной регрессии, указав dist в качестве зависимой переменной, и через значок ~ переменную speed в качестве регрессора:

```
model <- lm(data=d, dist~speed) # базовый пакет stats
```

Посмотрим на коэффициенты уравнения линейной регрессии:

```
model$coefficients
```

Результат в консоли:

```
> model$coefficients (Intercept) speed -17.579095 3.932409
```

(Intercept) – это константа в уравнении регрессии, speed – коэффициент регрессии.

Таким образом, уравнение регрессии имеет вид:

$$dist^m = -17.579 + 3.9324 \cdot speed$$

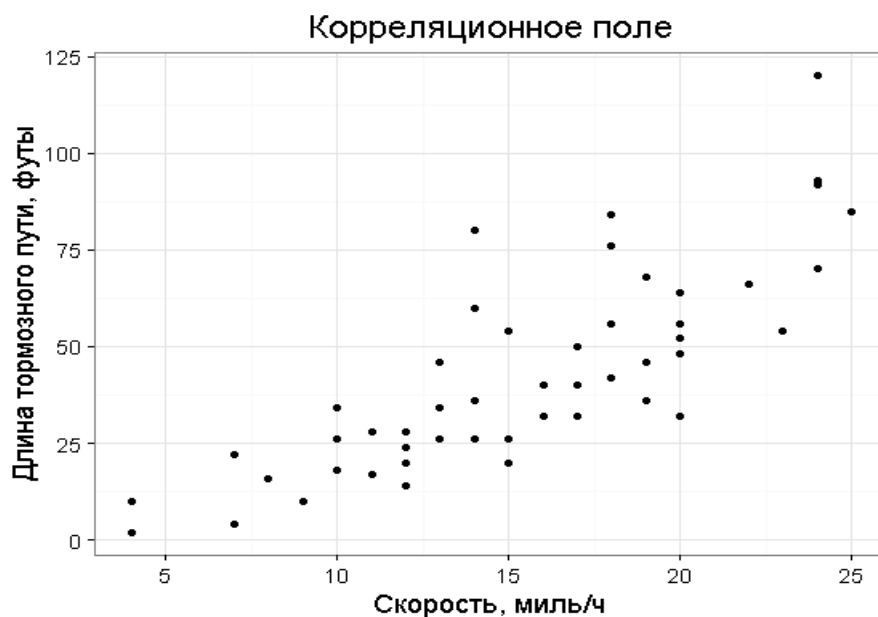


Рисунок 11 – График зависимости длины тормозного пути `dist` от скорости автомобиля `speed`

Тип `lm` представляет собой список из 12 элементов (рисунок 12).

Name	Type	Length	Size	Value
d	data.frame	2	1.5 KB	50 obs. of 2 variables
model	lm	12	24.1 KB	List of 12

Рисунок 12 – Переменная-список `lm` в таблице среды `Environment`

Также можно посмотреть значения вектора ошибок модели – разницу между реальной длиной тормозного пути `dist` и полученной по модели $dist_i^m$. Выведем первые 10 значений этого вектора с точностью две цифры после запятой:

```
model$residuals[1:10]
options(digits = 3)
1      2      3      4      5      6      7      8      9     10
3.85 11.85 -5.95 12.05 2.12 -7.81 -3.74 4.26 12.26 -8.68
```

Более полный набор расчетов по модели можно получить командой `summary`:

```
summary(model) # базовый пакет
```

Call:

```
lm(formula = dist ~ speed, data base = d)
```

Residuals:

```
Min      1Q  Median      3Q      Max
-29.069  -9.525   -2.272   9.215  43.201
```

Coefficients:

Estimate Std. Error

```
(Intercept) -17.5791 6.7584 -2.601  0.0123 *
```

```
speed 3.9324  0.4155 9.464 1.49e-12 ***
```

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 15.38 on 48 degrees of freedom

Multiple R-squared: 0.6511, Adjusted R-squared: 0.6438

F-statistic: 89.57 on 1 and 48 DF, p-value: 1.49e-12

Помимо коэффициентов регрессии, R выводит:

– стандартные ошибки коэффициентов (Std. Error);

- наблюдаемые значения t-критерия при проверке значимости коэффициентов регрессии (t-value);
- P-значения для коэффициентов регрессии (P-value). Звездочками или точками в столбце справа R показывает значимость или незначимость коэффициентов: *** — значимы на уровне значимости менее 0.001; ** — значимы на уровне значимости 0.001; * — значимы на уровне значимости 0.01; . — значимы на уровне значимости 0.05 и т. д. Эти обозначения приведены в разделе Signif. codes.

Коэффициент детерминации (Multiple R-squared) равен 0.6511; скорректированный коэффициент детерминации (Adjusted R-squared) равен 0.6438. Наблюдаемое значения F-критерия проверки значимости уравнения в целом и P-значение:

F-statistic: 89.57 on 1 and 48 DF, p-value: 1.49e-12

Таким образом, уравнение регрессии получилось значимым.

Проведем на графике полученную линию регрессии с 95% доверительными интервалами (рисунок 13):

```
qplot(data = d, speed, dist) + stat_smooth(method="lm", level = 0.95) + theme_bw(base_size = 18)
```

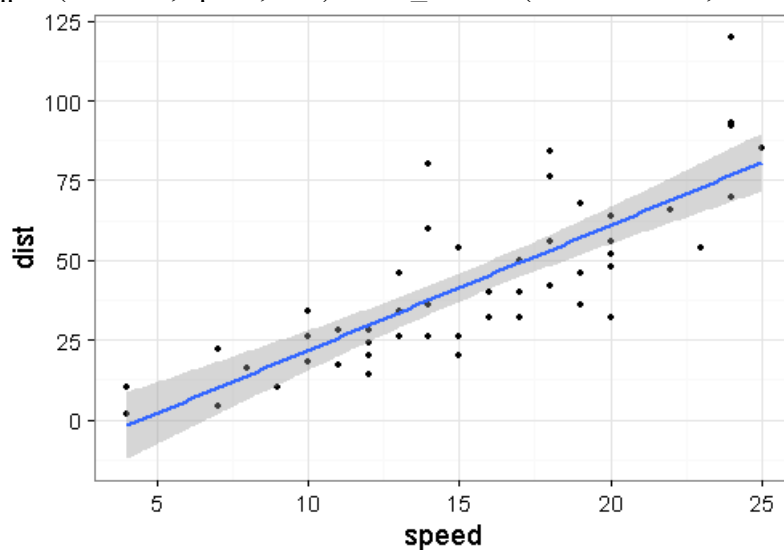


Рисунок 13 – График линейной регрессии с доверительными интервалами

Рассчитаем также 95% доверительные интервалы для параметров линейной регрессии:

```
> confint(model, level = 0.95) # базовый пакет stats
```

```
2.5 % 97.5 %
```

```
(Intercept) -31.167850 -3.990340
```

```
speed 3.096964 4.767853
```

Рассчитанные значения по модели можно получить командой f

```
itted(model) # базовый пакет stats
```

Необъясненная сумма квадратов отклонений:

```
RSS <- deviance(model) # базовый пакет stats
```

Можно также рассчитать полную сумму квадратов, воспользовавшись уже известными функциями sum и mean:

```
TSS <- sum((y-mean(y))^2)
```

Для того чтобы построить прогноз по полученной модели, нужно задать значения регрессора и поместить их в новый data.frame.

```
# создаем новый набор данных
```

```
nd <- data.frame(speed=c(40,60))
```

Строим прогноз функцией predict:

```
> predict(model,nd)
```

```
1          2
139.7173  218.3654
```

Множественная линейная регрессия

Рассмотрим встроенный набор данных по социально-экономическим показателям в 47 провинциях Швейцарии в 1888 г.

```
t <- swiss # встроенный набор данных по Швейцарии
```

Этот набор данных содержит 6 переменных по 47 наблюдений, каждая из которых измеряется в процентах (help(swiss)):

Fertility — рождаемость;

Agriculture — % мужчин, занятых в сельском хозяйстве;

Examination — % призывников, получивших высшую оценку на экзамене в армии;

Education — % призывников, имеющих образование помимо начального;

Catholic — % католиков среди населения;

Infant.Mortality — % детей, умерших до года.

Посмотрим на этот набор данных:

```
> glimpse(t) Observations: 47
```

```
Variables: 6
```

```
$ Fertility (dbl) 80.2, 83.1, 92.5, 85.8, 76.9, 76.1, 83.8, 92.4...
```

```
$ Agriculture (dbl) 17.0, 45.1, 39.7, 36.5, 43.5, 35.3, 70.2, 67.8...
```

```
$ Examination (int) 15, 6, 5, 12, 17, 9, 16, 14, 12, 16, 14, 21, 1...
```

```
$ Education (int) 12, 9, 5, 7, 15, 7, 7, 8, 7, 13, 6, 12, 7, 12, ...
```

```
$ Catholic (dbl) 9.96, 84.84, 93.40, 33.77, 5.16, 90.57, 92.85, ...
```

```
$ Infant.Mortality (dbl) 22.2, 22.2, 20.2, 20.3, 20.6, 26.6, 23.6, 24.9...
```

Встроенный пакет graphics содержит функцию pairs, позволяющую получить все возможные диаграммы рассеяния на одном графике, а также выполнить их сглаживание с помощью опции panel.smooth:

```
pairs(swiss, panel = panel.smooth)
```

Результатом будет график, показанный на рисунке 14.

Функция cor позволяет как вычислить корреляцию между двумя выборками, так и получить корреляционную матрицу для всех переменных из набора данных:

```
> cor(swiss)
```

```
Fertility
```

```
Fertility 1.000
```

```
Agriculture 0.353
```

```
Examination -0.646
```

```
Education -0.664
```

```
Catholic 0.464
```

```
Infant.Mortality 0.417
```

```
Agriculture 0.3531 1.0000 -0.6865 -0.6395 0.4011 -0.0609
```

```
Examination -0.646 -0.687 1.000 0.698 -0.573 -0.114
```

```
Education -0.6638 -0.6395 0.6984 1.0000 -0.1539 -0.0993
```

```
Catholic 0.464 0.401 -0.573 -0.154 1.000 0.175
```

```
Infant.Mortality
```

```
Fertility 0.4166
```

```
Agriculture -0.0609
```

```
Examination -0.1140
```

```
Education -0.0993
```

```
Catholic 0.1755
```

```
Infant.Mortality 1.0000
```

Корреляционную матрицу можно получить и в других видах, например, с помощью функции sjp.corr из пакета sjPlot (рисунок 15):

```
library("sjPlot")
```

```
sjp.corr(t)
```

Существует еще одна функция, позволяющая получить корреляционную матрицу, диаграммы рассеяния и сглаженные распределения одновременно (рисунок 16):

```
library("GGally")
```

```
ggpairs(t) # функция из пакета GGally
```

Чтобы оценить регрессию рождаемости на остальные переменные, можно воспользоваться уже знакомой функцией `lm`, а регрессоры перечислить через знак «плюс»:

```
model2 <- lm(data=t, Fertility~Agriculture+Education+Catholic)
```

В данном случае регрессорами стали % занятых в с/х; % католического населения и % имеющих образование выше начального.

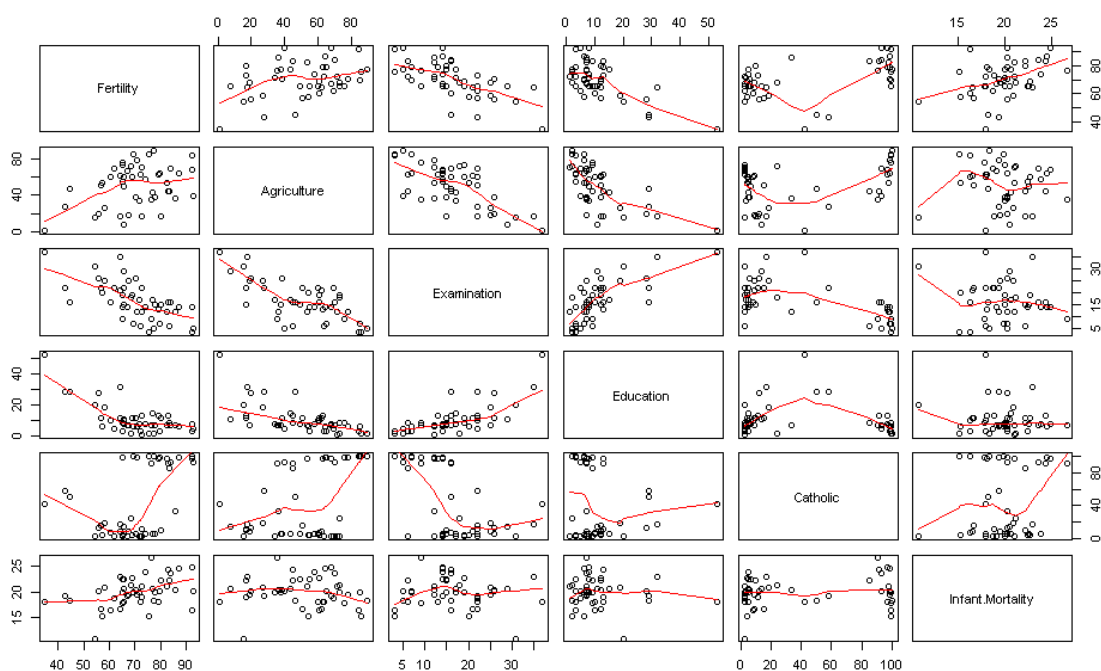


Рисунок 14 – Диаграммы рассеяния, полученные с помощью функции `pairs`

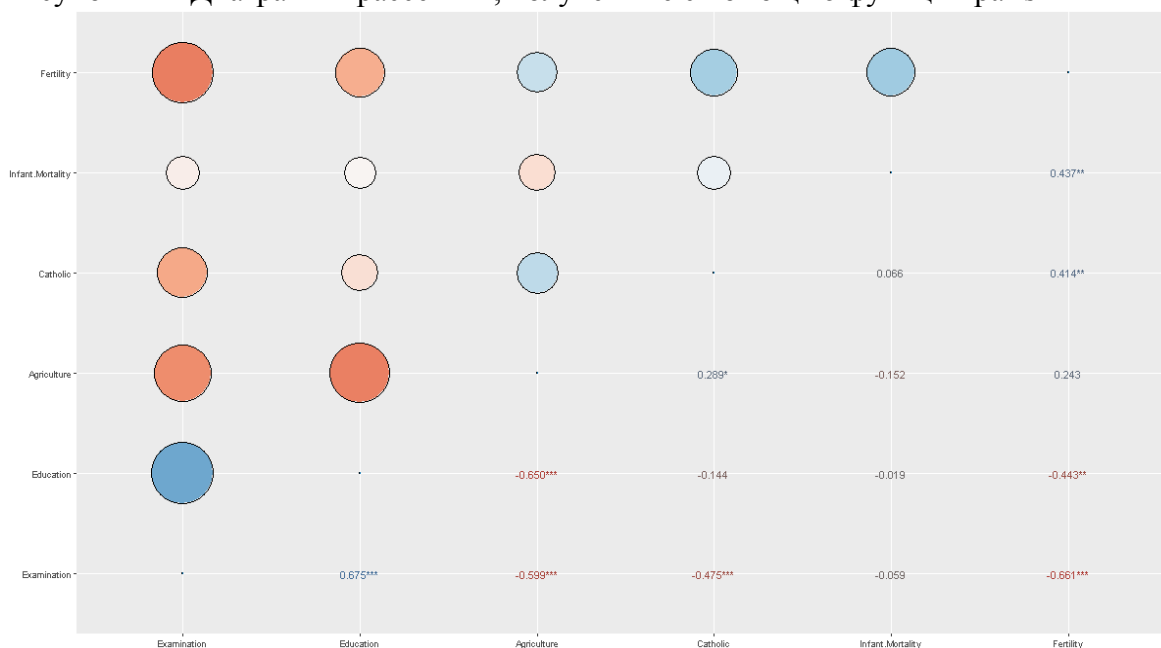


Рисунок 15 – Корреляционная матрица, полученная с помощью функции `sjr corr`

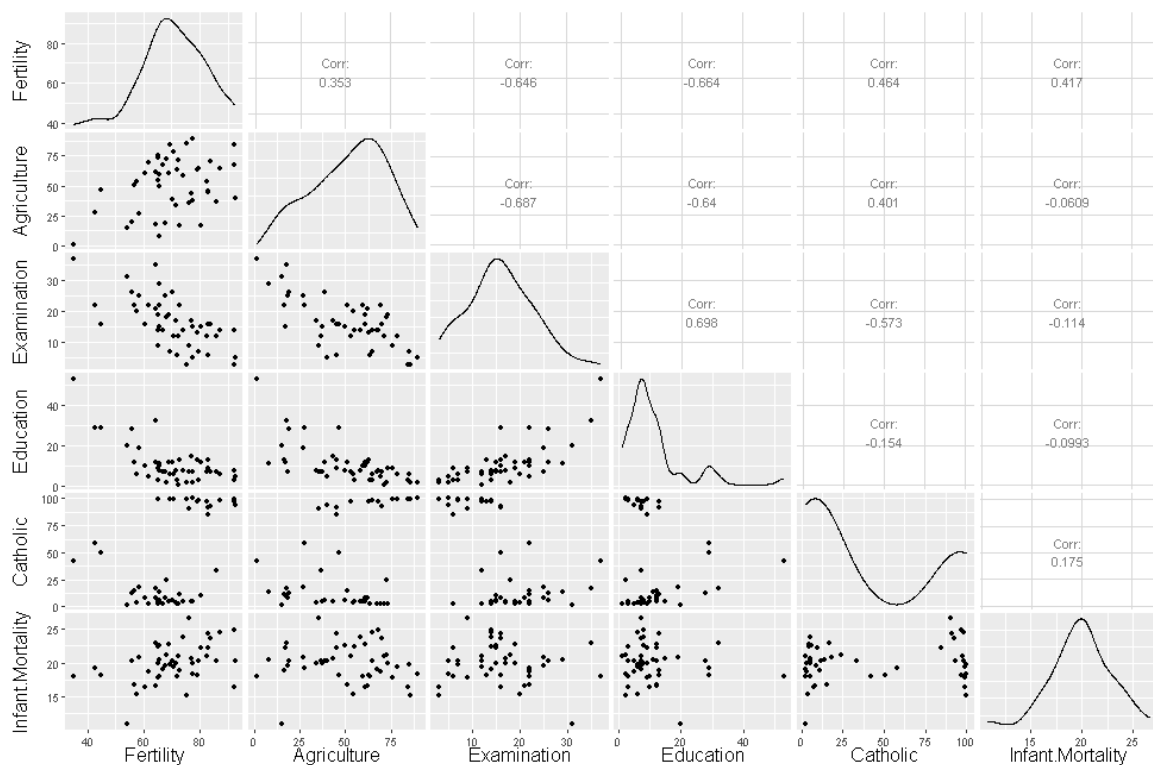


Рисунок 16 – Корреляционная матрица, диаграммы рассеяния и сглаженные распределения, полученные с помощью функции `ggpairs`

Получить оценки коэффициентов уравнения регрессии, а также проверить основные гипотезы поможет функция `summary`:

```
> summary(model2)
```

Call:

```
lm(formula = Fertility ~ Agriculture + Education + Catholic,
    data = t)
```

Residuals:

```
Min 1Q Median 3Q Max
-15.178 -6.548 1.379 5.822
```

Coefficients: Estimate Std. Error t value Pr(>|t|)

```
(Intercept) 86.22502 4.73472 18.211 < 2e-16 ***
```

```
Agriculture -0.20304 0.07115 -2.854 0.00662 **
```

```
Education -1.07215 0.15580 -6.881 1.91e-08 ***
```

```
Catholic 0.14520 0.03015 4.817 1.84e-05 ***
```

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.728 on 43 degrees of freedom

Multiple R-squared: 0.6423, Adjusted R-squared: 0.6173

F-statistic: 25.73 on 3 and 43 DF, p-value: 1.089e-09

Построим прогноз по аналогии с парной линейной регрессией. Отличие заключается лишь в том, что в наборе данных необходимо указать значения каждого фактора:

```
# создаем новый набор данных
```

```
nd2 <- data.frame(Agriculture=0.5, Catholic=0.5, Education=20)
```

```
> predict(model2, nd2)
```

```
1
```

```
64.75316
```

Построение прогноза по нескольким точкам выполняется с помощью векторов значений:

```
# создаем новый набор данных
```

```
nd2 <- data.frame(Agriculture=c(0.5, 0.8), Catholic=c(0.5, 0.65),
                  Education=c(20, 25))
```

```
# прогнозируем predict(model2,nd2)
1      2
64.75316 59.35330
```

Варианты заданий

№ варианта	1.	2.	3.	4.	5.
Набор данных	airquality	state.x77	Cars93*	Cars93*	Cars93*
y	Ozone	Life Exp***	Price	Min.Price	Max.Price
Факторы и их значения для прогноза	Solar.R = 350 Wind = 8,3 Temp = 80	Illiteracy = 1,0 Murder = 12 Income = 4000	Horsepower = 200 RPM = 5200 Passengers = 4	Horsepower = 210 RPM = 5500 Passengers = 4	Horsepower = 220 RPM = 6000 Passengers = 6
№ варианта	6.	7.	8.	9.	10.
Набор данных	stackloss	longley	longley	LifeCycleSavings	Anscombe**
y	stackloss	GNP	Employed	sr	education
Факторы и их значения для прогноза	Air.Flow = 55 Water.Temp = 20 Acid.Conc = 89	Unemployed = 221 Armed.Forces = 180 Population = 125	GNP.deflator = 102 Armed.Forces = 170 Population = 110	pop15 = 35,5 pop75 = 1,5 dpi = 2500 ddpi = 2,15	income = 3200 young = 347,8 urban = 425

* необходимо подключить библиотеку MASS

** необходимо подключить библиотеку car

*** поскольку в названии переменной нельзя использовать символ пробела, данную переменную необходимо переименовать командой colnames, например, в Life_Exp, либо обращаться к ней по номеру столбца [, 4]

```
d <- state.x77
```

```
colnames(d)[4] <- "Life_Exp"
```

```
#или
```

```
d$Life_Exp <- d[,4]
```

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. Что такое Intercept в парной линейной регрессии?
2. Как проверить значимость уравнения регрессии в целом?
3. Как узнать разброс значений коэффициентов регрессии?

Список литературы, рекомендуемый к использованию по данной теме

Основная литература:

1. Голубева, Н.В. Математическое моделирование систем и процессов. - Санкт-Петербург: Лань, 2021;
2. Казиев, В.М. Введение в анализ, синтез и моделирование систем: Учебное пособие. - Москва, Саратов: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020;
3. Поспелов, Е.А., Попов, И.С. Пакеты прикладных программ в научных исследованиях: учебно-методическое пособие. - Омск: Омский государственный университет им. Ф.М. Достоевского, 2019.

Дополнительная литература:

1. Агалаков, С.А. Анализ данных в среде R: практикум. - Омск: Омский государственный университет им. Ф.М. Достоевского, 2020;

2. Воскобойников, Ю.Е. Регрессионный анализ данных в пакете MATHCAD. - Санкт-Петербург: Лань, 2021.

Интернет-ресурсы

1. <http://biblioclub.ru> – ЭБС «Университетская библиотека on-line», <http://old.exponenta.ru/> – Образовательный математический сайт по математике и программированию
2. <http://MATLAB.exponenta.ru/Simulink/default.php> – сайт, посвященный интерактивному инструменту для моделирования Simulink,
3. <https://www.intuit.ru/> – Национальный Открытый Университет «ИНТУИТ»
4. <http://window.edu.ru/> – Информационная система «Единое окно доступа к образовательным ресурсам»
5. <http://artspb.com/> – Образовательный портал: математика, кибернетика и программирование дополнительного образования
6. <http://www.arenasimulation.com> – Официальный сайт разработчика среды имитационного моделирования ARENA
<https://www.anylogic.ru> – Официальный сайт разработчика AnyLogic

Лабораторная работа 14. Временные ряды

Цель лабораторной работы:

Изучить базовые способы анализа, моделирования и прогно- зирования временных рядов.

Организационная форма занятий: решение проблемных задач, разбор конкретных ситуаций

Вопросы для обсуждения на лабораторном занятии: Временные ряды.

Формируемые компетенции: ОПК-2, ОПК-3, ОПК-4, ОПКД-2, ОПКД-3, ОПКД-4.

Задание

1. Загрузить данные из временного ряда для своего варианта за последние полгода.
2. Построить график временного ряда с графиками автокорреляционной и частной автокорреляционной функций.
3. Подобрать вручную порядок ARMA-модели. Для подбора использовать визуальный анализ графика.
4. Подобрать ARMA-модель автоматически.
5. Построить прогноз на указанное в варианте число шагов. Показать прогноз на графике.

Указания к выполнению работы

Анализ временных рядов

Для работы подключим следующие пакеты:

```
library("lubridate") # работа с датами
library("zoo") # работа с временными рядами
library("xts") # дополнительные функции для работы с временными рядами
library("dplyr") # работа с наборами данных
library("ggplot2") # графики
library("forecast") # прогнозы
library("lmtest") # тестирование гипотез в линейных моделях
library("quantmod") # загрузка данных с различных источников
```

Для начала посмотрим, как сгенерировать тестовые выборки. Для этого воспользуемся функцией `arima.sim` из базового пакета `stats`.

Выполним симуляцию AR(1)-процесса в 100 наблюдений по модели: $y_t = 0.6 y_{t-1} + \varepsilon_t$ и поместим в переменную `y`.

```
y <- arima.sim(n=100, list(ar=0.6))
```

В этой функции необходимо указать объем выборки, а также в параметре `list` коэффициенты модели.

Можно построить полученный ряд на графике (рисунок 17):

```
plot(y)
```

Графики автокорреляционной и частной автокорреляционной функций для процесса `y` вызываются командами соответственно:

```
Acf(y) Pacf(y)
```

Все три графика одновременно могут быть вызваны командой:

```
tsdisplay(y)
```

Результат выполнения команды показан на рисунке 18.

Посмотрим еще несколько примеров симуляции AR, MA, ARMA, ARIMA процессов.

Процесс MA(1) $y_t = \varepsilon_t - 0.8\varepsilon_{t-1}$ `arima.sim(n=100, list(ma=-0.8))`

Процесс ARMA(1,1) $y_t = 0.5y_{t-1} + \varepsilon_t - 0.8\varepsilon_{t-1}$ `arima.sim(n=100, list(ma=-0.8, ar=0.5))`

Процесс ARMA(2,2) $y_t = 0.9y_{t-1} - 0.5y_{t-2} + \varepsilon_t - 0.2\varepsilon_{t-1} + 0.3\varepsilon_{t-2}$ `arima.sim(n = 100, list(ar = c(0.9, -0.5), ma = c(-0.2, 0.3))`

Процесс случайного блуждания $y_t = y_{t-1} + \varepsilon_t$ `arima.sim(n=100, list(order=c(0,1,0)))`

Белый шум $y_t = \varepsilon_t$: `arima.sim(n=100, list(order=c(0,0,0)))`

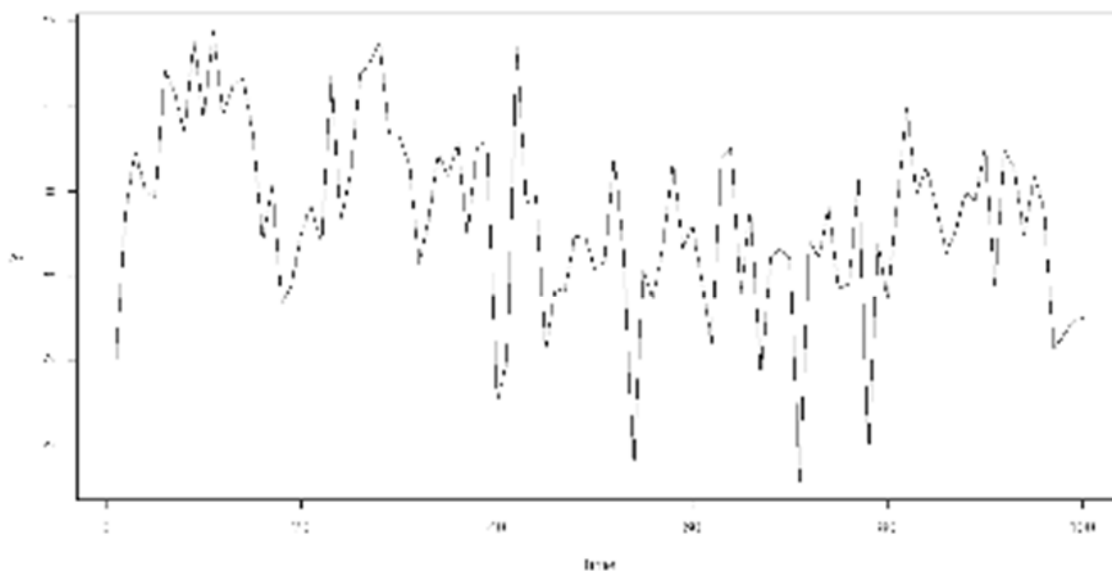


Рисунок 17 – График временного ряда

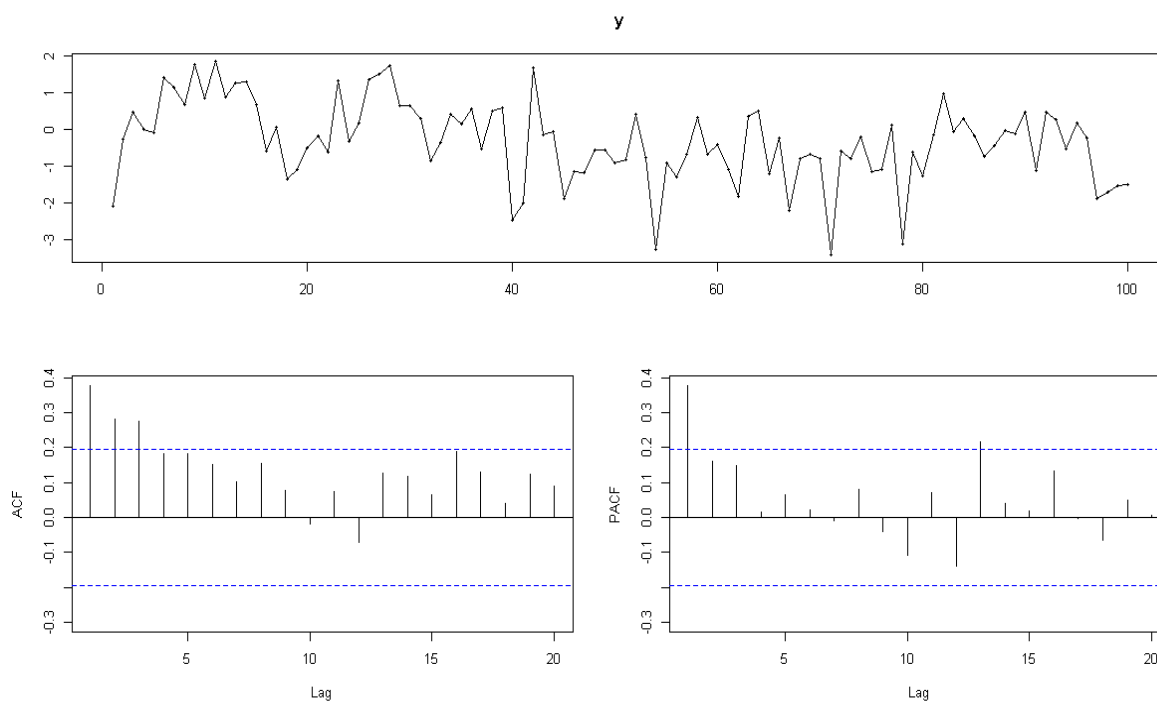


Рисунок18 – График временного ряда с графиками автокорреляционной и частной автокорреляционной функций

Попробуем подобрать различные ARIMA-модели под реальные данные. Для примера выберем временной ряд, содержащий 98 ежегодных наблюдений за уровнями воды в озере Гурон с 1875 по 1972 гг. (рисунок 19).

`help(LakeHuron)`

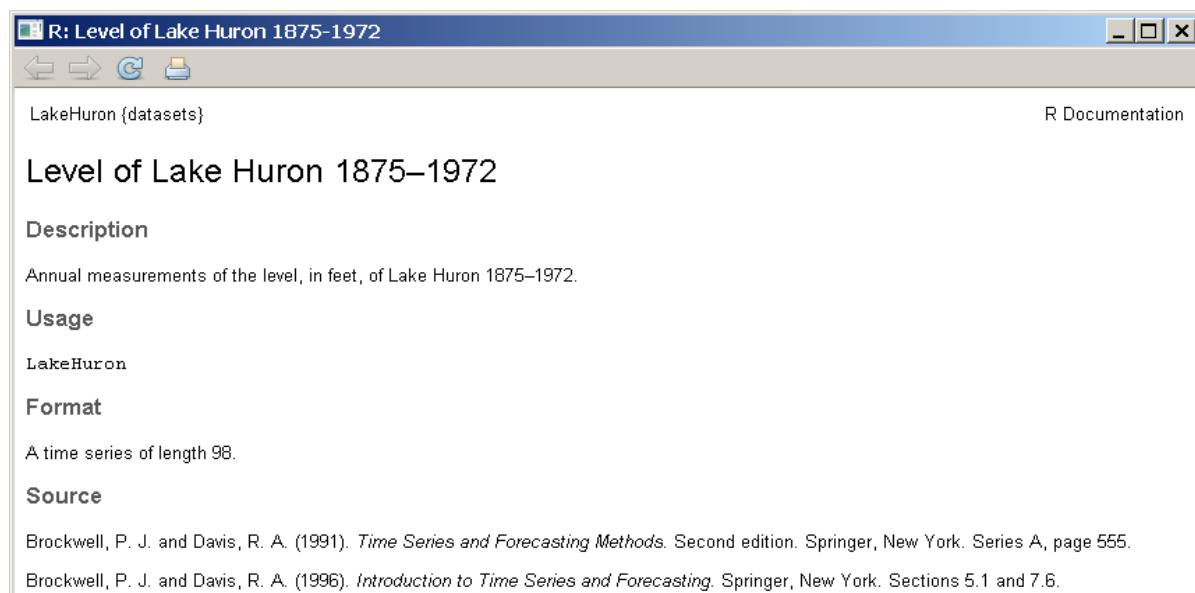


Рисунок 19 – Справка по набору данных LakeHuron

Оценим ARMA(2,1) модель с помощью функции `Arima` из пакета `forecast`:

```
mod<- Arima(y, order=c(2,0,1))
```

Посмотрим на результат оценивания:

```
summary(mod)
```

Series: y

ARIMA(2,0,1) with non-zero mean Coefficients:

```
ar1 ar2 ma1 intercept
```

```
0.7829 -0.0342 0.2857 579.0533
```

```
s.e. 0.3262 0.2845 0.3144 0.3467
```

```
sigma^2 estimated as 0.4749: log likelihood=-103.24
```

```
AIC=216.48 AICc=217.13 BIC=229.4
```

Training set error measures:

```
ME RMSE MAE MPE MAPE MASE
```

```
Training set -0.008638553 0.6891058 0.5492008 -0.001635858 0.09486097 0.9378958
```

```
ACF1
```

```
Training set 0.002260998
```

Получили модель $y_t = 579.0533 + 0.7829y_{t-1} - 0.0342y_{t-2} + \epsilon_t + 0.2857\epsilon_{t-1}$. В строке `s.e.` перечислены стандартные ошибки коэффициентов.

Оценка дисперсии $\sigma^2 = 0.4749$. Строкой ниже перечислены значения критерия Акаике ($AIC=216.48$), скорректированного критерия Акаике ($AICc=217.13$) и критерия Шварца ($BIC=229.4$).

Можно отдельно вывести значение критерия Акаике командой:

```
AIC(mod)
```

Командой `fitted(mod)` можно получить модельные значения временного ряда. Построим реальные и модельные значения на одном графике (рисунок 20):

```
matplot(cbind(y,fitted(mod)),type='l')
```

Функция `matplot` строит значения столбцов матрицы, поэтому ее аргументом нужно задать столбцы. Команда `cbind` соединяет в матрицу столбец `y` и столбец значений, рассчитанных по модели. `type='l'` указывает, что тип графика – линия.

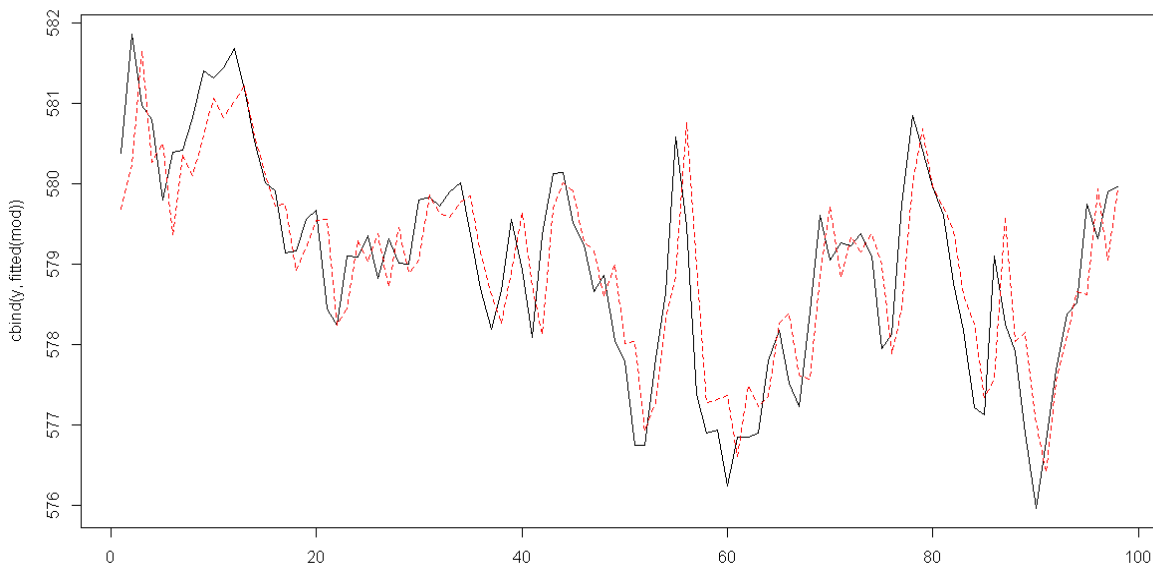


Рисунок 20 – График модельных и реальных значений временного ряда

Построим прогноз на 5 шагов вперед:

```
prognos<- forecast(mod, h=5)
```

Выведем полученные значения. Результат включает 80% и 95% доверительный интервал для прогноза:

```
prognos
```

Point	Forecast	Lo 80	Hi 80	Lo 95	Hi 95
1973	579.7407	578.8576	580.6238	578.3901	581.0913
1974	579.5605	578.2680	580.8530	577.5838	581.5372
1975	579.4269	577.9528	580.9009	577.1725	581.6813
1976	579.3284	577.7645	580.8924	576.9366	581.7203
1977	579.2559	577.6453	580.8666	576.7927	581.7192

Строим график прогноза (рисунок 21):

```
plot(prognos)
```

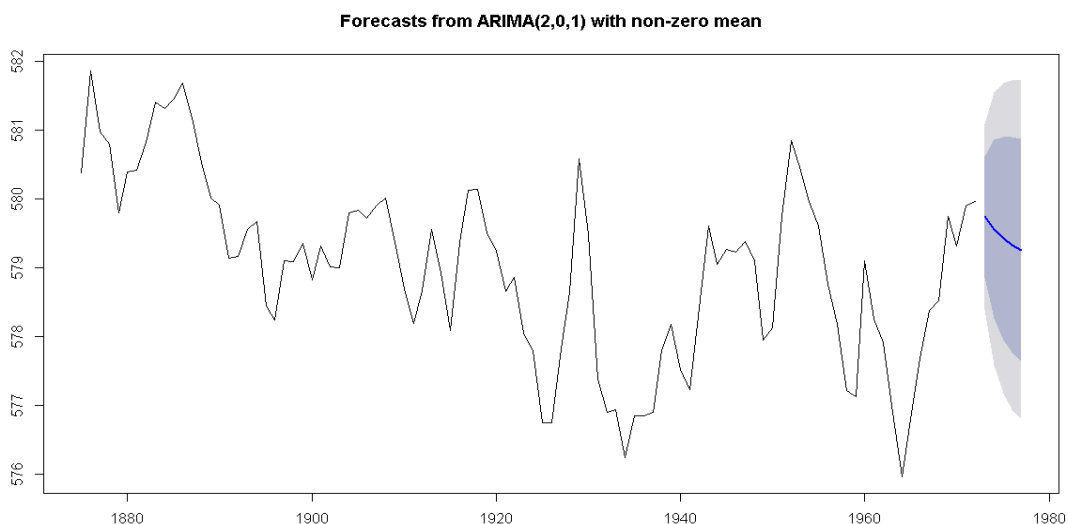


Рисунок 21 – График прогноза по модели ARIMA

R также может автоматически подбирать ARIMA-модель по штрафному критерию (минимальное значение критерия Акаике):

```
mod_a<- auto.arima(y)
```

```
summary(mod_a)
```

Series: y

ARIMA(0,1,4) with drift

Coefficients:

ma1 ma2 ma3 ma4 drift

```
0.0584 -0.3158 -0.3035 -0.2349 -0.0205
s.e. 0.1031 0.1058 0.1100 0.1299 0.0167
sigma^2 estimated as 0.4806: log likelihood=-101.09
AIC=214.18 AICc=215.12 BIC=229.63
```

Training set error measures:

ME RMSE MAE MPE MAPE

MASE

Training set 0.002898915 0.6886758 0.5441565 0.0003830464 0.093983 0.9292814

ACF1 Training set 0.01081339

«ARIMA(0,1,4) with drift» означает, что была подобрана ARIMA(0,1,4) с линейным трендом (drift), но без константы (отсутствует параметр intercept).

Таким образом, получим модель

$y_t = y_{t-1} + \epsilon_t + 0.0584\epsilon_{t-1} - 0.3158\epsilon_{t-2} - 0.3035\epsilon_{t-3} - 0.2349\epsilon_{t-4} - 0.0205\epsilon_{t-5}$.

Загрузка данных из различных источников в R

Для корректного перевода дат на английский язык в русскоязычной Windows, необходимо сначала выполнить следующую команду:

```
Sys.setlocale("LC_TIME", "C")
```

Пакет quantmod позволяет загружать данные из нескольких источников, а именно:

– Yahoo! Finance (OHLC data) – <http://finance.yahoo.com/>;

– Federal Reserve Bank of St. Louis FRED® (11,000 экономических временных рядов) – <http://research.stlouisfed.org/fred2/>;

– Google Finance (OHLC data) – <http://finance.google.com/>;

– Oanda, The Currency Site (FX and Metals) – <http://www.oanda.com/>.

Для загрузки данных используется одна и та же функция getSymbols, например:

#данные о стоимости акций Гугл с finance.google.com за период с 1 января 2015г. по 1 декабря 2015г.

```
getSymbols("GOOG", src="google", from="2015-01-01", to="2015-12-01")
```

GOOG – это краткое название акций компании на бирже (тíкер). Теперь эти данные загружены в переменную с таким же названием GOOG.

Если не указывать начальную или конечную дату, то будут загружены все доступные данные. Можно посмотреть шесть первых значений и шесть последних значений командами head и tail соответственно.

```
head(GOOG)
```

	GOOG.Open	GOOG.High	GOOG.Low	GOOG.Close	GOOG.Volume	GOOG.Adjusted
2015-01-02	529.0124	531.2724	524.1023	524.8124	1447600	524.8124
2015-01-05	523.2624	524.3324	513.0623	513.8723	2059800	513.8723
2015-01-06	515.0024	516.1773	501.0523	501.9623	2899900	501.9623
2015-01-07	507.0023	507.2463	499.6522	501.1023	2065100	501.1023
2015-01-08	497.9922	503.4823	491.0022	502.6823	3353600	502.6823
2015-01-09	504.7623	504.9223	494.7922	496.1723	2069400	496.1723

```
tail(GOOG)
```

	GOOG.Open	GOOG.High	GOOG.Low	GOOG.Close	GOOG.Volume	GOOG.Adjusted
2015-11-23	757.45	762.708	751.82	755.98	1414500	755.98
2015-11-24	752.00	755.279	737.63	748.28	2333100	748.28
2015-11-25	748.14	752.000	746.06	748.15	1122100	748.15
2015-11-27	748.46	753.410	747.49	750.26	838500	750.26
2015-11-30	748.81	754.930	741.27	742.60	2035300	742.60
2015-12-01	747.11	768.950	746.70	767.04	2129900	767.04

Посмотрим на график (рисунок 22):

```
barChart(GOOG, theme = "white")
```



Рисунок 22 – График стоимости акций Гугл GOOG

Также можно посмотреть на цену закрытия с помощью функции `tsdisplay` (рисунок 23):
`tsdisplay(GOOG$GOOG.Close)`

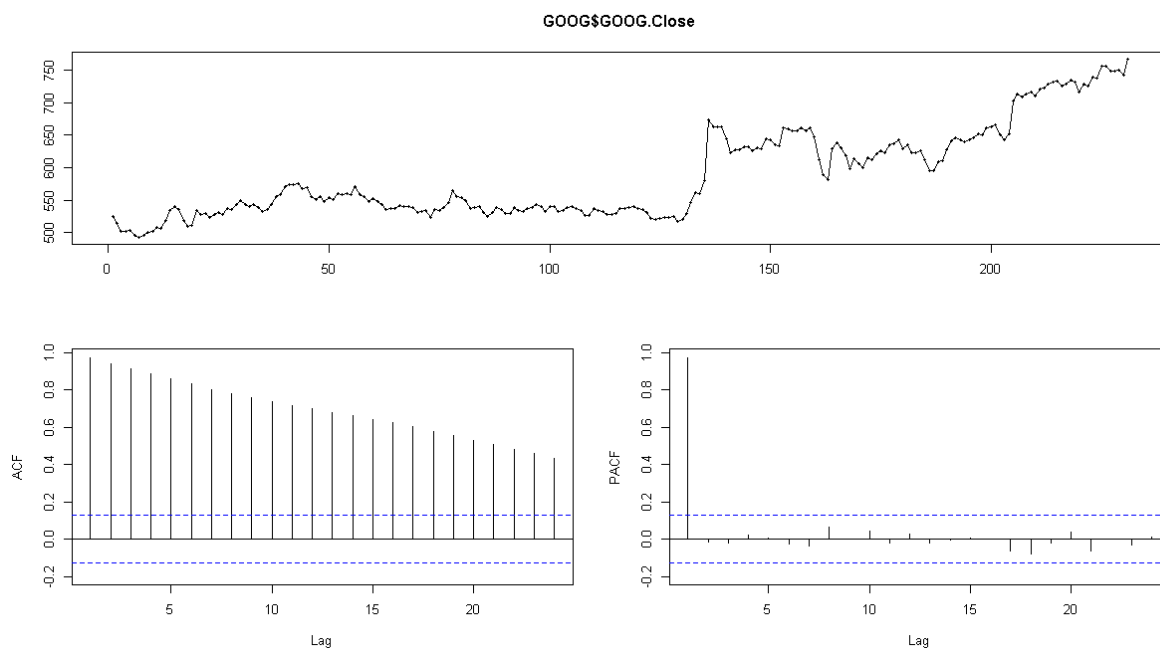


Рисунок 23 – График цены закрытия акций Гугл GOOG

Приведем еще несколько примеров:

#данные о стоимости акций Yahoo с `finance.google.com` `getSymbols("YHOO",src="google")`

#данные о стоимости акций Apple с `finance.yahoo.com` `getSymbols("AAPL",src="yahoo",
 from="2015-01-01", to="2015-12-01")`

Японские свечи (рисунок 24):

`candleChart(AAPL, multi.col=TRUE, theme="white")`



Рисунок – 24. График стоимости акций Apple AAPL

Для более детальной информации можно обратиться на сайт разработчиков пакета.

Варианты заданий

№ варианта	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.
Компания	Yandex	Twitter	Facebook	ВКонтакте	Amazon	EBay	Microsoft	Сбербанк	Вымпелком	МТС
Тикер	YNDX	TWTR	FB	VK	AMZN	EBAY	MSFT	SBNC	VIP	MBT
Глубина прогноза	10	15	20	25	30	25	20	15	10	20

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. В чем отличие автокорреляционной и частной автокорреляционной функций?
2. Что такое случайное блуждание и белый шум?
3. Как можно сравнить модели временного ряда по точности?

Список литературы, рекомендуемый к использованию по данной теме

Основная литература:

1. Голубева, Н.В. Математическое моделирование систем и процессов. - Санкт-Петербург: Лань, 2021;
2. Казиев, В.М. Введение в анализ, синтез и моделирование систем: Учебное пособие. - Москва, Саратов: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020;
3. Пospelов, Е.А., Попов, И.С. Пакеты прикладных программ в научных исследованиях: учебно-методическое пособие. - Омск: Омский государственный университет им. Ф.М. Достоевского, 2019.

Дополнительная литература:

1. Агалаков, С.А. Анализ данных в среде R: практикум. - Омск: Омский государственный университет им. Ф.М. Достоевского, 2020;

2. Воскобойников, Ю.Е. Регрессионный анализ данных в пакете MATHCAD. - Санкт-Петербург: Лань, 2021.

Интернет-ресурсы

1. <http://biblioclub.ru> – ЭБС «Университетская библиотека on-line», <http://old.exponenta.ru/> – Образовательный математический сайт по математике и программированию

2. <http://MATLAB.exponenta.ru/Simulink/default.php> – сайт, посвященный

интерактивному инструменту для моделирования Simulink,

3. <https://www.intuit.ru/> – Национальный Открытый Университет «ИНТУИТ»

4. <http://window.edu.ru/> – Информационная система «Единое окно доступа к образовательным ресурсам»

5. <http://artspb.com/> – Образовательный портал: математика, кибернетика и программирование дополнительного образования

6. <http://www.arenasimulation.com> – Официальный сайт разработчика среды имитационного моделирования ARENA

<https://www.anylogic.ru> – Официальный сайт разработчика AnyLogic

Лабораторная работа 15-16. Генетические алгоритмы

Цель лабораторной работы:

научиться применять генетические алгоритмы (ГА) для поиска оценок параметров нелинейных моделей.

Организационная форма занятий: решение проблемных задач, разбор конкретных ситуаций

Вопросы для обсуждения на лабораторном занятии: Генетические алгоритмы.

Формируемые компетенции: ОПК-2, ОПК-3, ОПК-4, ОПКД-2, ОПКД-3, ОПКД-4.

Задание

1. Выполнить генерацию значений x и y .
2. Построить график зависимости y от x .
3. Задать функцию зависимости y от x в R и соответствующую ей функцию МНК.
4. Определить минимальные и максимальные значения параметров.
5. Запустить поиск решения с помощью ГА. Размер популяции, предельное число итераций и условие останова алгоритма заданы в варианте.
6. Построить график значений fitness-функции и график с исходными данными и полученной моделью.
7. Повторить пункты 5 и 6 еще 2 раза и сравнить полученные результаты.
8. Выбрать наилучший из трех результатов по значению коэффициента детерминации.

Указания к выполнению работы

Генетические алгоритмы

Рассмотрим решение задачи нелинейного МНК с помощью генетического алгоритма на встроенном наборе данных `trees`.

Для работы подключим следующие пакеты:

```
library("GA") # генетические алгоритмы
library("dplyr") # работа с наборами данных
library("ggplot2") # графики
library("spuRs") # содержит набор данных trees
```

Получим описание набора данных по деревьям `trees` из пакета `spuRs` командой (рисунок 25):

```
help("trees", package = "spuRs")
```

В этом наборе данных содержится 1200 наблюдений и три переменных (ID дерева, который включает номер местоположения, площадки и номер дерева; возраст дерева; объем в куб. дм). Будем рассматривать зависимость объема дерева (переменная `Vol`) от возраста (переменная `Age`).

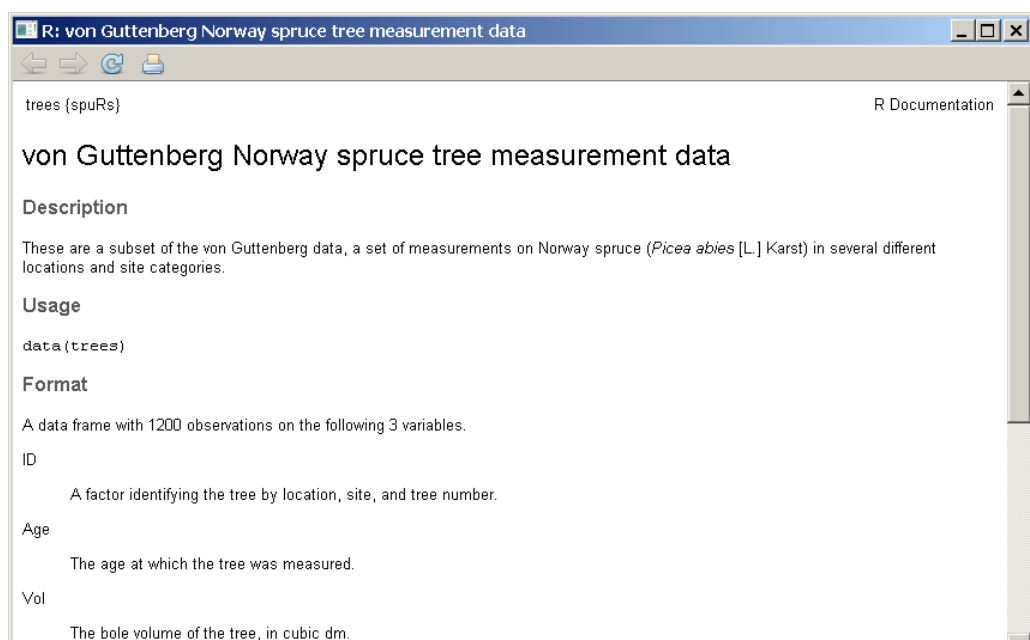


Рисунок 25 – Справка по набору данных `trees` из пакета `spuRs`

Активируем набор данных командой:

```
data("trees", package = "spuRs")
```

Посмотрим на этот набор данных:

```
> glimpse(trees)
```

Observations: 1200

Variables:

```
$ ID (fctr) 1.1.1, 1.1.1, 1.1.1, 1.1.1, 1.1.1, 1.1.1, 1.1.1, 1.1.1, 1.1.1, 1.1.1, 1....
```

```
$ Age (dbl) 9.67, 19.67, 29.67, 39.67, 49.67, 59.67, 69.67, 79.67, 89.67, 99.67, 109....
```

```
$ Vol (dbl) 5.0, 38.0, 123.0, 263.0, 400.0, 555.0, 688.0, 820.0, 928.0, 1023.0, 1104....
```

Выберем для дальнейшей работы только данные для деревьев с определенным местоположением, например, ID = 1.3.111. Поместим в переменную tree выбранную часть исходной выборки:

```
tree<- trees[trees$ID == "1.3.11", 2:3]
```

Командой trees[] можно выбрать нужные столбцы из набора данных. Таким образом, предыдущая команда выбирает из набора данных только второй и третий столбцы, но так, чтобы в первом столбце ID был при этом равен 1.3.11.

Посмотрим теперь на tree:

```
> glimpse(tree)
```

Observations: 12

Variables:

```
$ Age (dbl) 2.44, 12.44, 22.44, 32.44, 42.44, 52.44, 62.44, 72.44, 82.44, 92.44, 102....
```

```
$ Vol (dbl) 2.2, 20.0, 93.0, 262.0, 476.0, 705.0, 967.0, 1203.0, 1409.0, 1659.0, 1898...
```

Получен набор данных из 12 наблюдений с двумя переменными.

Попробуем описать зависимость объема дерева (y) от его возраста (x) с помощью логистической функции Ричардса:

$$y = a(1 - e^{-bx})^c.$$

причем параметры входят в модель нелинейно:

$$a, b, c = \arg \min_{a, b, c} \sum_{i=1}^{12} \left(Vol_i - a(1 - e^{-b \cdot Age_i})^c \right)^2.$$

Затем нужно будет применить метод наименьших квадратов,

Минимизировать сумму квадратов ошибок будем численно – с помощью генетического алгоритма. Генетический алгоритм реализован в R функцией ga из пакета GA.

Функция ga максимизирует функцию fitness, которую следует задать. Поскольку стоит задача минимизации, то воспользуемся следующим фактом:

$$\arg \max_{\Theta} (\Theta) = \arg \min_{\Theta} (-\Theta).$$

Для этого необходимо задать в R функцию Ричардса – она будет использована для функции fitness, указанной в аргументах функции ga. Пусть вектор параметров функции обозначен theta. Тогда a = theta[1], b = theta[2], c = theta[3]. Задаем функцию richards:

```
richards<- function(x, theta)
```

```
theta[1] * (1 - exp(-theta[2] * x))^theta[3]
```

В первой строке указан аргумент функции x и вектор параметров theta. Через пробел в следующей строке описывается сама функция Ричардса, порядок параметров a, b, c в векторе theta должен сохраняться.

тавим перед функцией суммы квадратов ошибок знак минус:

```
fit<- function(theta, x, y) -sum((y - richards(x, theta))^2)
```

В функции ga задаются следующие параметры:

fit – нужная нам функция для аргумента fitness, которую максимизирует генетический алгоритм;

type выберем real-valued, поскольку возраст деревьев и объем являются действительными числами;

x и y – это соответственно возраст дерева (tree\$Age) и объем дерева (tree\$Vol) – аргументы функции fitness;

min – это вектор минимальных значений параметров a, b, c функции Ричардса;

max – вектор максимальных значений параметров a, b, c;

crossover – функция в R, выполняющая кроссовер, т. е. функция, которая образует потомков, объединив часть генетической информации от родителей;

popsize – размер популяции;

maxiter – максимальное число итераций, после которого работа генетического алгоритма прекращается;

run – число последовательных поколений без какого-либо улучшения в значении fitness-функции перед остановом алгоритма и т. д.

Запишем результат выполнения функции ga в переменную myGA, задав значения описанным аргументам:

```
myGA<- ga(type = "real-valued", fitness = fit,x = tree$Age, y = tree$Vol, min = c(3000, 0, 2), max = c(4000, 1, 4),
```

```
popSize = 500, crossover = gareal_blxCrossover, maxiter = 5000,
```

```
run = 200, names = c("a", "b", "c"))
```

Функция при работе будет выводить в консоль значения fit на каждой итерации алгоритма:

```
Iter = 1 | Mean = -74580573 | Best = -6766.369
```

```
Iter = 2 | Mean = -64840448 | Best = -6766.369
```

```
Iter = 3 | Mean = -56793683 | Best = -6766.369
```

```
Iter = 4 | Mean = -48988112 | Best = -6766.369
```

```
Iter = 5 | Mean = -38083470 | Best = -6766.369
```

```
Iter = 6 | Mean = -31671411 | Best = -4661.229
```

```
Iter = 7 | Mean = -21836845 | Best = -4661.229 ...
```

Посмотрим на краткие результаты подбора функции Ричардса на исходные данные:

```
>summary(myGA)
```

```
| Genetic Algorithm |
```

```
GA settings:
```

```
Type = real-valued
```

```
Population size = 500
```

```
Number of generations = 5000
```

```
Elitism = 25
```

```
Crossover probability = 0.8
```

```
Mutation probability = 0.1
```

```
Search domain
```

```
abc
```

```
Min 3000 0 2
```

```
Max 4000 1 4
```

```
GA results:
```

```
Iterations = 792
```

```
Fitness function value = -2774.113
```

```
Solution =
```

```
abc
```

```
[1,] 3589.788 0.01546055 2.786145
```

Таким образом, получено следующее уравнение регрессии:

$$\hat{y} = 3589.788 \left(1 - e^{-0.015x}\right)^{2.786}.$$

Командой plot(myGA) можно посмотреть, как изменялись значения fitness-функции на протяжении всех 792 итераций алгоритма (рисунок 26).

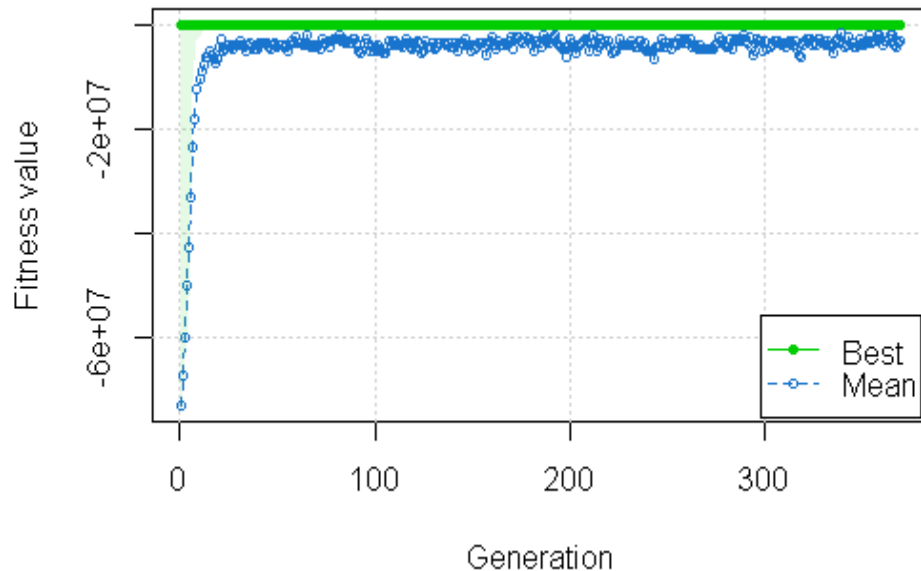


Рисунок 26 – Изменение значений fitness-функции в зависимости от числа итераций
 Модельные значения у можно получить командой: `richards(tree$Age, myGA@solution)`
 Построим на графике исходную выборку и результат подбора функции Ричардса (рис. 27):

```
ggplot() +  
  geom_point(aes(x=tree$Age, y=tree$Vol)) +  
  geom_line(aes(x=tree$Age, y=richards(tree$Age, myGA@solution)))
```

В данной команде указано:

`ggplot()` активирует построение графика с помощью функций пакета `ggplot2`;

`geom_point(aes(x=tree$Age, y=tree$Vol))` – строит точками зависимость объема дерева от возраста по исходной выборке;

`geom_line(aes(x=tree$Age, y=richards(tree$Age, myGA@solution)))` – строит линией модельные значения объема деревьев в зависимости от возраста, рассчитанные по функции Ричардса с параметрами, найденными генетическим алгоритмом.

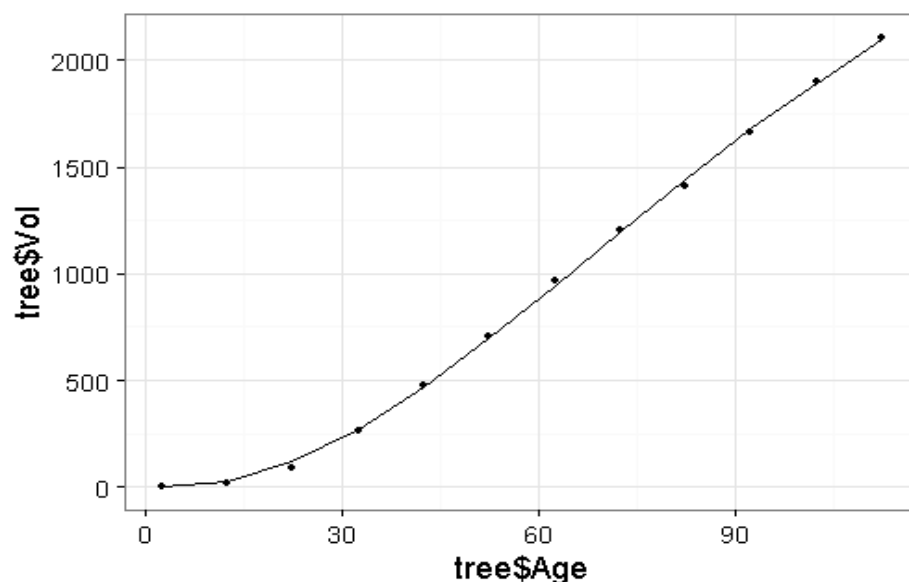


Рисунок 27 – График исходных данных и полученной функции Ричардса

Для оценки точности модели рассчитаем ее коэффициент детерминации:

```
RSS <- sum((tree$Vol-richards(tree$Age, myGA@solution))^2)
```

```
TSS <- sum((tree$Vol-mean(tree$Vol))^2)
```

```
R2 <- 1-RSS/TSS
```

```
>R2
```

```
[1] 0.9995553
```

Построим по полученной модели прогноз объема дерева для возраста 150 и 200 лет:

```
Age_forecast <- c(150,200)
```

```
richards(Age_forecast, myGA@solution)
```

```
[1] 2691.205 3156.228
```

Для построения графика прогноза в виде гладкой линии по- требуется большое число точек, которые удобнее задать в виде по- следовательности от 115 до 200 с шагом 5:

```
Age_forecast <- seq(115,200, by = 5)
```

```
ggplot() +
```

```
geom_point(aes(x=tree$Age, y=tree$Vol)) +
```

```
geom_line(aes(x=tree$Age, y=richards(tree$Age, myGA@solution))) +
```

```
geom_line(aes(x=Age_forecast, y=richards(Age_forecast, myGA@solution),  
color = "maroon")) + theme_bw(base_size = 18)
```

Результат показан на рисунке 28.

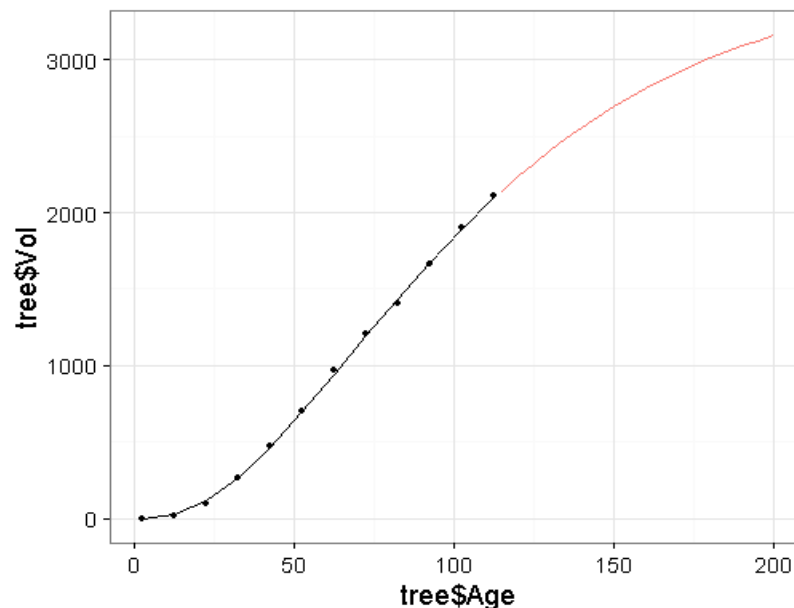


Рисунок 28 – График исходных данных и полученной функции Ричардса с прогнозом до 200 лет

Генерация данных

Выполним генерацию значений x с нормальным законом распределения, математическим ожиданием 45 и среднеквадратическим отклонением (СКО) 7. Объем выборки зададим равным 100:

```
x<-rnorm(n=100,mean=45,sd=7)
```

Для генерации y необходимо задать формулу зависимости, на- пример, линейную, и наложить на нее случайную помеху ε :

```
a<-20 b<-5
```

```
y<-a+b*x
```

Генерацию ε также выполним по нормальному закону распределения, с нулевым математическим ожиданием и единичной дисперсией. Однако, поскольку генераторы случайных чисел неидеальны, необходимо дополнительно выполнить нормировку и центрирование значений:

```
er<-rnorm(n=100,mean=0,sd=1)
```

```
er <- (er- mean(er))/sd(er)
```

После этого можно наложить помеху на формулу зависимости с заданным СКО, равным 2:

```
Ser<-5
```

```
y<-y+er*Ser
```

В результате получим зависимость, показанную на рисунке 29.

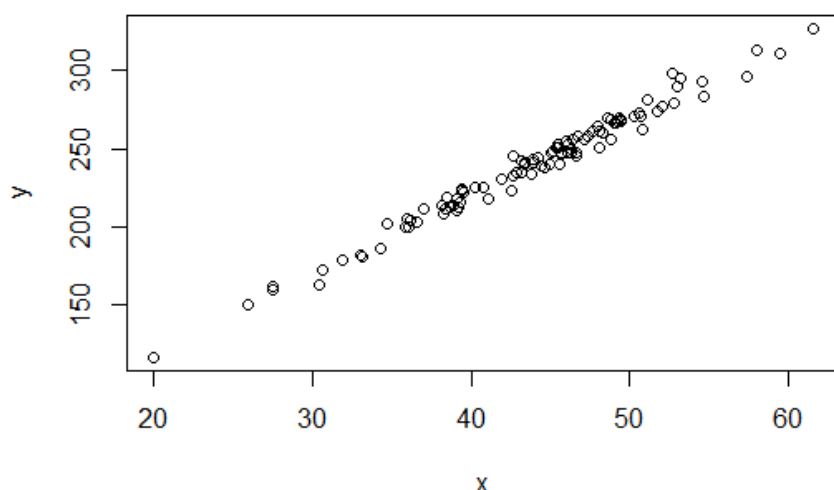


Рисунок 29 – Сгенерированная зависимость y от x

Варианты заданий

№ варианта	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.
Среднее x	7	30	100	20	50	10	25	40	60	33
Ст. отклон. x	2	3	15	4	6	1	4	7	10	8
Объем выборки	50	100	150	70	50	30	60	90	110	80
Формула y	$y = a + b(x - c)^2 + \varepsilon$			$y = a + \frac{b}{x - c} + \varepsilon$		$y = a + e^{bx+c} + \varepsilon$		$y = \frac{a}{1 + e^{-b(x-c)}} + \varepsilon$		
a	10	100	-5	10	150	20	1000	50	80	20
b	-2	5	-0.1	3	-10	0.8	-0.25	0.25	0.3	0.22
c	5	32	90	10	30	0.5	5	35	52	30
Ст. отклон. помехи	4	28	14	0.05	0.1	3000	0.25	3	6	1.5

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. Максимизирует или минимизирует генетический алгоритм в R fitness-функцию?
2. Как задать в R функцию $y = a(x - b)^3 + c$?
3. Каким образом в R можно сгенерировать выборку со стандартным нормальным законом распределения?

Список литературы, рекомендуемый к использованию по данной теме

Основная литература:

1. Голубева, Н.В. Математическое моделирование систем и процессов. - Санкт-Петербург: Лань, 2021;

2. Казиев, В.М. Введение в анализ, синтез и моделирование систем: Учебное пособие. - Москва, Саратов: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020;

3. Пospelов, Е.А., Попов, И.С. Пакеты прикладных программ в научных исследованиях: учебно-методическое пособие. - Омск: Омский государственный университет им. Ф.М. Достоевского, 2019.

Дополнительная литература:

1. Агалаков, С.А. Анализ данных в среде R: практикум. - Омск: Омский государственный университет им. Ф.М. Достоевского, 2020;

2. Воскобойников, Ю.Е. Регрессионный анализ данных в пакете MATHCAD. - Санкт-Петербург: Лань, 2021.

Интернет-ресурсы

1. <http://biblioclub.ru> – ЭБС «Университетская библиотека on-line», <http://old.exponenta.ru/> – Образовательный математический сайт по математике и программированию
2. <http://MATLAB.exponenta.ru/Simulink/default.php> – сайт, посвященный интерактивному инструменту для моделирования Simulink,
3. <https://www.intuit.ru/> – Национальный Открытый Университет «ИНТУИТ»
4. <http://window.edu.ru/> – Информационная система «Единое окно доступа к образовательным ресурсам»
5. <http://artspb.com/> – Образовательный портал: математика, кибернетика и программирование дополнительного образования
6. <http://www.arenasimulation.com> – Официальный сайт разработчика среды имитационного моделирования ARENA
7. <https://www.anylogic.ru> – Официальный сайт разработчика AnyLogic

Приложение. Основные команды ядра системы MATLAB

Команды общего назначения:

- 1) Общая информация: help, helpwin, helpdesk, demo, ver, whatsnew, readme.
- 2) Управление рабочим пространством: who, whos, clear, pack, load, save, quit, exit.
- 3) Управление командами и функциями: what, type, edit, lookfor, which, pcode, inmem, mex.
- 4) Управление путями поиска: path, addpath, rmpath, editpath.
- 5) Управление командным окном: echo, more, diary, format, clc, home.
- 6) Команды операционной системы: cd, copyfile, pwd, dir, delete, getenv, mkdir, !, dos, web, computer.
- 7) Отладка М-файлов: debug, dbstop, dbclear, dbcont, dbdown, dbstack, dbstatus, dbstep, dbtype, dbup, dbquit.
- 8) Профилирование М-файлов: profile.

Конструкции языка программирования:

- 1) Арифметические операторы: +, -, *, ./, ^, .^, \, /, .\, ./.
- 2) Операторы отношения: ==, ~=, <, >, <=, >=.
- 3) Логические операторы: &, and, or, ~, not, xor, any, all.
- 4) Специальные символы: : () [] { } , ; ! = ' . ' [.] [;]
- 5) Форматы чисел: format name.
- 6) Поразрядные (битовые) операции: bitand, bitcmp, bitor, bitmax, bitxor, bitset, bitget, bitshift.
- 7) Специальные константы и переменные: ans, eps, realmax, realmin, pi, i, j, inf, NaN, isnan, isinf, isfinite, flops, why.
- 8) Операторы управления программой: if, else, elseif, end, for, while, break, switch, case, otherwise, try, catch, return.
- 9) Операторы вычисления и выполнения: eval, feval, evalin, builtin, assignin, run.
- 10) Операторы работы с аргументами функций: nargchk, nargin, nargsout, varargin, varargout, inputname.
- 11) Вывод сообщений: error, warning, lasterr, lastwarn, errortrap, disp, fprintf, sprintf.
- 12) Интерактивный ввод: input, keyboard, pause, uimenu, uicontrol, disp.
- 13) Скрипты, функции и переменные: script, function, global, persistent, mfilename, exist, isglobal, mlock, munlock, mislocked, clear.
- 14) Функции пользователя: inline, argnames, formula, char.
- 15) Комментарии: %.

Операции над матрицами и множествами:

- 1) Множества: union, unique, intersect, setdiff, setxor, ismember.
- 2) Элементарные матрицы: zeros, ones, eye, repmat, rand, randn, linspace, logspace, meshgrid.
- 3) Информация о матрицах и массивах: size, length, ndims, disp, isempty, isequal, isnumeric, islogical, logical.
- 4) Формирование матриц: reshape, diag, tril, triu, fliplr, flipud, flipdim, rot90, :, find, end, sub2ind, ind2sub.
- 5) Специальные матрицы: compan, gallery, hadamard, hankel, hilb, invhilb, magic, pascal, rosser, toeplitz, vander, wilkinson.
- 6) Матричные функции: norm, normest, rank, det, trace, null, orth, rref, subspace, inv, chol, cholinc, lu, luinc, qr, nnls, pinv, lsconv, expm, logm, sqrtm, funm.
- 7) Собственные значения и сингулярные числа: eig, svd, gsvd, eigs, svds, polyeig, condeig, hess, qz, schur.
- 8) Переупорядочение: colmmd, symmmd, symrcm, colperm, randperm, dmperm.
- 9) Разреженные матрицы: speye, sprand, sprandn, sprandsym, spdiags, sparse, full, find, spconvert, nnz, nonzeros, nzmax, spones, spalloc, issparse, spfun, spy.

Математические функции:

- 1) Тригонометрические и гиперболические: sin, sinh, asin, asinh, cos, cosh, acos, acosh, tan, tanh, atan, atan2, atanh, sec, sech, asec, asech, csc, csch, acsc, acsch, cot, coth, acot, acoth.
- 2) Экспоненциальные, степенные и логарифмические: exp, log, log10, log2, pow2, sqrt.
- 3) Комплексные: abs, angle, conj, imag, real, unwrap, isreal, cplxpair.
- 4) Округления, знаки, модули и остатки: fix, floor, ceil, round, mod, rem, sign.
- 5) Специальные функции: airy, besseli, bessely, besselh, besseli, bessellk, beta, betainc, betaln, ellipj, ellipke, erf, erfc, erfcx, erfinv, expint, gamma, gammainc, gammaln, legendre, cross.
- 6) Теоретико-числовые: factor, isprime, primes, gsd, lcm, rat, rats, perms, nchoosek, kron.

Анализ данных и их математическая обработка:

- 1) Основные операции: max, min, mean, median, std, sort, sortrows, sum, prod, hist, trapz, cumsum, cumprod, cumtrapz.
- 2) Конечные разности: diff, gradient, del2.
- 3) Корреляция: corrcoef, cov, subspace.
- 4) Фильтрация и свертка: filter, filter2, conv, conv2, convn, deconv.
- 5) Интерполяция и аппроксимация: interp1, interp1q, interpft, interp2, interp3, interpn, griddata, spline, ppval.
- 6) Операции с полиномами: poly, roots, polyval, polyvalm, residue, polyfit, polyder, conv, deconv, polyeig.
- 7) Преобразования координат: cart2sph, cart2pol, pol2cart, sph2cart.
- 8) Преобразование Фурье: fft, fft2, fftn, ifft, ifft2, ifftn, fftshift, ifftshift.

Численные методы:

- 1) Минимизация и нахождение нулей функции: fmin, fmins, fzero.
- 2) Численное интегрирование: quad, quad8, dblquad.
- 3) Решение обыкновенных дифференциальных уравнений: ode45, ode23, ode113, ode23t, ode15s, ode23s, ode23tb, odeset, odeget, odeplot, odeprint, odephas2, odephas3.
- 4) Решение линейных алгебраических уравнений: pcg, bicg, bicgstab, cgs, gmres, qmr.

Графика:

- 1) Двумерные графики: plot, loglog, semilogx, semilogy, polar, plotyy.
- 2) Трехмерные графики: plot3, mesh, meshc, meshz, surf, surfc, fill, fill3, surface.
- 3) Управление осями координат: axis, zoom, grid, box, hold, axes, subplot, daspect, pbaspect, xlim, ylim, zlim, gca, cla, caxis, ishold.
- 4) Надписи на графиках: legend, title, xlabel, ylabel, zlabel, text, gtext, colorbar, plottedit (on-off), edtext.
- 5) Вывод графической страницы на печать: print, printopt, orient, vml.
- 6) Управление цветом: colormap, shading, hidden, brighten, contrast, spinmap, rgbplot, colstyle.
- 7) Освещение: surf1, lighting, material, specular, diffuse, surfnorm, camlight, lightangle.
- 8) Палитры: hsv, hot, gray, bone, copper, pink, white, flag, lines, colorcube, jet, prism, cool, autumn, spring, winter, summer.
- 9) Выбор точки наблюдения: view, viewmtx, rotate3d.
- 10) Управление фотокамерой: campos, camtarget, camva, camup, camproj, camorbit, campan, camdolly, camzoom, camroll, camlookat.
- 11) Специальная графика: area, bar, barh, bar3, bar3h, comet, errorbar, feather, fill, hist, pareto, pie, pie3, plotmatrix, ribbon, scatter, stem, staris, ezplot, fplot, comet3, quiver3, scatter3, slice, trisurf, trimesh, waterfall, cylinder, sphere, path, line.
- 12) Линии уровня и поле направлений: contour, contoure, contourf, contour3, clabel, quiver, voronoi, pcolor.
- 13) Отображение образов: image, imagesc, imread, imwrite, iminfo.
- 14) Движение и анимация: capture, moviein, getframe, movie, rotate, frame2im, im2frame.
- 15) Управление графическими окнами и объектами: figure, clf, shg, close, refresh, closerec, newplot, ishandle, set, get, reset, delete, gco, gcbo, gcbf, drawnow, findobj, copyobj, allchild, findall, hidegui.

16) Графический интерфейс пользователя (GUI): `uicontrol`, `uimenu`, `uiconextmenu`, `ginput`, `dragrect`, `rbbox`, `selectmoveresize`, `waitforbuttonpress`, `waitfor`, `uiwait`, `uiresume`, `uisuspend`, `uirestore`, `guide`, `align`, `cbedit`, `menuedit`, `propedit`, `dialog`, `axlimdlg`, `errordlg`, `helpdlg`, `warndlg`, `inputdlg`, `listdlg`, `menu`, `msgbox`, `questdlg`, `uigetfile`, `uiputfile`, `uisetcolor`, `uisetfont`, `pagedlg`, `printdlg`, `waitbar`, `makemenu`, `menubar`, `inttoggle`, `winmenu`, `btngroup`, `btnstate`, `btnpress`, `btndown`, `btnup`, `clruprop`, `getuprop`, `setupprop`.

Звук и аудио:

- 1) Общие команды: `sound`, `soundsc`, `speak`, `recordsound`, `soundcap`, `mu2lin`, `lin2mun`.
- 2) Прием и передача: `auwrite`, `auread`, `wavwrite`, `wavread`.

Символьные строки:

- 1) Общие команды: `char`, `double`, `cellstr`, `blanks`, `deblank`, `eval`.
- 2) Проверка строк: `ischar`, `iscellstr`, `isletter`, `isspace`.
- 3) Операции со строками: `strcat`, `strvcat`, `strcmp`, `strncmp`, `strcmpi`, `strncmpi`, `findstr`, `strjust`, `strmath`, `strrep`, `strtok`, `upper`, `lower`.
- 4) Преобразование строк в числа и чисел в строки: `num2str`, `int2str`, `mat2str`, `str2num`, `sprintf`, `sscanf`, `hex2num`, `hex2dec`, `dec2hex`, `bin2dec`, `dec2bin`, `base2dec`, `dec2base`.

Работа с файлами:

- 1) Открытие и закрытие: `fopen`, `fclose`.
- 2) Бинарный ввод-вывод: `fread`, `fwrite`.
- 3) Форматированный ввод-вывод: `fscanf`, `fprintf`, `fgetl`, `fgets`, `input`.
- 4) Позиционирование: `ferror`, `feof`, `fseek`, `ftell`, `frewind`.
- 5) Обработка имен: `fullfile`, `fileparts`, `partialpath`, `tempdir`, `tempname`.
- 6) Импорт – экспорт: `load`, `save`, `dlmread`, `dlmwrite`, `wklread`, `wklwrite`, `hdf`.

Время и даты:

- 1) Текущие дата и время: `now`, `date`, `clock`.
- 2) Операции с датами: `calendar`, `weekday`, `eomday`, `datetick`, `datenum`, `datestr`, `datevec`.
- 3) Операции со временем: `cputime`, `tic`, `toc`, `etime`, `pause`.

Типы данных и структуры:

- 1) Типы данных (классы): `double`, `sparse`, `char`, `cell`, `unit8`, `inline`.
- 2) Операции с многомерными массивами: `cat`, `ndims`, `ngrid`, `permute`, `ipermute`, `shiftdim`, `squeeze`, `cell`, `celldisp`, `cellplot`, `num2cell`, `deal`, `cell2struct`, `struct2cell`.
- 3) Операции со структурами: `struct`, `fieldnames`, `getfield`, `setfield`, `rmfield`, `isfield`, `isstruct`.

Все функции приведены без аргументов. Напомним, что более подробную информацию о любой из перечисленных выше функций и примеры их применения можно получить из командного окна:

```
>> help имя функции  
или  
>> demo имя функции
```

ЛИТЕРАТУРА

1. Агалаков, С.А. Анализ данных в среде R: практикум. - Омск: Омский государственный университет им. Ф.М. Достоевского, 2020.
2. Анализ данных в R [Электронный ресурс] // Stepic.org. – Режим доступа: <https://stepic.org/course/Анализ-данных-в-R-129>.
3. Белова Н.Д. Решение задач в пакете MathCAD: методические указания по выполнению лабораторных работ № 1 – 5 по информатике для студентов дневной формы обучения / сост. Н. Д. Белова. – Хабаровск : Изд-во Тихоокеан. гос. ун-та, 2007. – 51 с.
4. Воскобойников, Ю.Е. Регрессионный анализ данных в пакете MATNCAD. - Санкт-Петербург: Лань, 2021.
5. Голубева, Н.В. Математическое моделирование систем и процессов. - Санкт-Петербург: Лань, 2021.
6. Дьяконов В.П. MATLAB 6.5 SP1/7 + Simulink 5/6. Основы применения. Серия «Библиотека профессионала». – М.: СОЛОН-Пресс, 2005, 800 стр.
7. Зорин А. В., Федоткин М. А. Введение в прикладной статистический анализ в пакете R: учебно-методическое пособие. – Нижний Новгород: ННГУ, 2010. – 50 с.
8. Казиев, В.М. Введение в анализ, синтез и моделирование систем: Учебное пособие. - Москва, Саратов: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020.
9. Кривилев А. В. Основы компьютерной математики с использованием системы MATLAB. - М.: Лекс-Книга, 2005, 496 стр.
10. Кондрашов В. Е., Королев С.Б. MATLAB как система программирования научно-технических расчетов. - М.: Мир, Институт стратегической стабильности Минатома РФ, 2002, 350 стр.
11. Мاستицкий С. Э., Шитиков В. К. Статистический анализ и визуализация данных с помощью R. – М.: ДМК Пресс, 2015. – 496 с.
12. Поспелов, Е.А., Попов, И.С. Пакеты прикладных программ в научных исследованиях: учебно-методическое пособие. - Омск: Омский государственный университет им. Ф.М. Достоевского, 2019.
13. Семенычев В. К., Коробецкая А. А., Кожухова В. Н. Предложения эконометрического инструментария моделирования и прогнозирования эволюционных процессов: монография. – Самара: САГМУ, 2015. – 384 с.
14. Фазылов В.Р., Шульгина О.Н., Щербакова Н.К. Применение MATLAB для обработки экспериментальных данных. Методическое пособие. – Казань: Казанский государственный университет, 2005, 56 стр.
15. Эконометрика [Электронный ресурс] // Coursera. – Режим доступа: <https://www.coursera.org/course/econometrics>.
16. Code School – Try R [Электронный ресурс]. – Режим доступа: <http://tryr.codeschool.com/>.
17. Coghlan A. A Little Book of R for Time Series [Электронный ресурс]. – 2015. – Режим доступа: <https://media.readthedocs.org/pdf/a-little-book-of-r-for-time-series/latest/a-little-book-of-r-for-time-series.pdf>.
18. ggplot2 Help [Электронный ресурс]. – Режим доступа: <http://docs.ggplot2.org/current/index.html>.
19. Package ‘GA’ [Электронный ресурс] // The Comprehensive R Archive Network. – Режим доступа: <https://cran.r-project.org/web/packages/GA/GA.pdf>.
20. Peng R. D. R Programming for Data Science [Электронный ресурс]. – Режим доступа: <https://leanpub.com/rprogramming/>.
21. quantmod: Quantitative Financial Modelling & Trading Framework for R [Электронный ресурс]. – Режим доступа: <http://www.quantmod.com/>.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

по организации самостоятельной работы по дисциплине
«Информационные системы и технологии в научных
исследованиях»

для студентов направления подготовки

09.04.02 «Информационные системы и технологии»

Направленность (профиль)

«Искусственный интеллект, математическое
моделирование и суперкомпьютерные технологии
в разработке информационных систем»

Ставрополь
2022

СОДЕРЖАНИЕ

Введение

1. Цель и задачи освоения дисциплины
2. Технологическая карта самостоятельной работы студента
3. Организационно-методические рекомендации по освоению дисциплины
4. Методические указания по выполнению самостоятельной работы студентов
5. Содержание самостоятельной работы
6. Примерная тематика заданий для самостоятельной работы студентов
7. План-график выполнения СРС
8. Организация контроля знаний студентов
9. Рекомендации по работе с литературой и источниками
10. Критерии оценивания компетенций
11. Учебно-методическое и информационное обеспечение дисциплины

Введение

1. Цель и задачи освоения дисциплины

Целью учебной дисциплины «Информационные системы и технологии в научных исследованиях» является формирование и развитие у студентов профессиональных компетенций, предназначенных для автоматизации научных экспериментов, а также для осуществления моделирования исследуемых объектов, явлений и процессов, изучение которых традиционными средствами затруднено или невозможно.

Дисциплина «Информационные системы и технологии в научных исследованиях» должна обеспечивать формирование фундамента подготовки будущих магистров в области телекоммуникаций и интеллектуальных систем, а также, создавать необходимую базу для успешного овладения последующими специальными дисциплинами учебного плана. Она должна способствовать развитию творческих способностей студентов, умению формулировать и решать задачи изучаемой специальности, умению творчески применять и самостоятельно повышать свои знания.

Эти цели достигаются на основе фундаментализации, интенсификации и индивидуализации процесса обучения путём внедрения и эффективного использования современных компьютерных технологий. В результате изучения дисциплины у студентов должны сформироваться знания, умения и навыки, позволяющие разрабатывать оригинальные алгоритмы и программные средства с использованием современных интеллектуальных компьютерных технологий, для решения задач в области создания и применения искусственного интеллекта.

Задачи освоения дисциплины: изучение и применение информационных систем и технологий в научных исследованиях.

2. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесённых с планируемыми результатами освоения образовательной программы

Код, формулировка компетенции	Код, формулировка индикатора	Планируемые результаты обучения по дисциплине (модулю), характеризующие этапы формирования компетенций, индикаторов
ОПК-2 Способен разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач	ОПК-2.1: Разрабатывает оригинальные алгоритмы и программные средства для решения профессиональных задач	Анализирует алгоритмы и программные средства для решения профессиональных задач Разрабатывает оригинальные алгоритмы и программные средства для решения профессиональных задач
	ОПК-2.2: Использует современные интеллектуальные технологии для решения профессиональных задач	Использует современные интеллектуальные технологии для решения профессиональных задач. Оценивает эффективность применения современных интеллектуальных технологий для решения профессиональных задач в выбранной предметной области. Использует практические навыки по разработке

		оригинальных, а также модификации существующих алгоритмов и программных средств с использованием интеллектуальных технологий для решения профессиональных задач.
ОПК-3: Способен анализировать профессиональную информацию, выделять в ней главное, структурировать, оформлять и представлять в виде аналитических обзоров с обоснованными выводами и рекомендациями	ОПК-3.1: Выполняет обобщение, структурирование и критический анализ профессиональной информации	Обобщает, структурирует и анализирует профессиональную информацию. Использует методы структурирования необходимой профессиональной информации для решения поставленной задачи. Использует методы критического анализа необходимой профессиональной информации для решения поставленной задачи. Применяет различные методики поиска, обобщения, структурирования необходимой профессиональной информации.
	ОПК-3.2: Оформляет и представляет профессиональную информацию в виде аналитических обзоров с обоснованными выводами и рекомендациями	
ОПК-4: Способен применять на практике новые научные принципы и методы исследований	ОПК-4.1: Осуществляет выбор методов исследования задач в ИТ-области	Учитывает современные информационные технологии для проведения научного исследования. Определяет методологический аппарат научного исследования в ИТ-области. Использует необходимый инструментарий научного исследования в ИТ-области. Анализирует структуру и логику научного исследования.
	ОПК-4.2: Применяет научные принципы и методы исследований задачи в ИТ-области	
ОПКД-2. Способен разрабатывать оригинальные алгоритмы и программные средства с	ОПКД-2.1. Применяет современные информационно-коммуникационные и	Использует современные информационно-коммуникационные и интеллектуальные

использованием современных интеллектуальных компьютерных технологий, для решения задач в области создания и применения искусственного интеллекта	интеллектуальные компьютерные технологии, инструментальные среды, программно-технические платформы для решения задач в области создания и применения искусственного интеллекта	компьютерные технологии, инструментальные среды, программно-технические платформы для решения задач в области создания и применения искусственного интеллекта. Выбирает современные информационно-коммуникационные и интеллектуальные компьютерные технологии. Разрабатывает оригинальные программные средства, в том числе с использованием современных информационно-коммуникационных и интеллектуальных компьютерных технологий, для решения задач в области создания и применения искусственного интеллекта.
	ОПКД-2.2. Обосновывает выбор современных информационно-коммуникационных и интеллектуальных компьютерных технологий	
	ОПКД-2.3. Разрабатывает оригинальные программные средства, в том числе с использованием современных информационно-коммуникационных и интеллектуальных компьютерных технологий, для решения задач в области создания и применения искусственного интеллекта	
ОПКД-3. Способен анализировать профессиональную информацию для решения задач в области создания и применения технологий и систем искусственного интеллекта, выделять в ней главное, структурировать, оформлять и представлять в виде аналитических обзоров и презентаций с обоснованными выводами и рекомендациями	ОПКД-3.1. Применяет принципы, методы и средства анализа и структурирования профессиональной информации для решения задач области создания и применения технологий и систем искусственного интеллекта	Анализирует и применяет принципы, методы и средства анализа и структурирования профессиональной информации для решения задач области создания и применения технологий и систем искусственного интеллекта. Анализирует профессиональную информацию, выделяет в ней главное, структурирует, оформляет и представляет в виде аналитических обзоров. Подготавливает научные доклады, публикации и аналитические обзоры с обоснованными выводами и рекомендациями,
	ОПКД-3.2. Анализирует профессиональную информацию, выделяет в ней главное, структурирует, оформляет и представляет в виде аналитических обзоров	
	ОПКД-3.3. Подготавливает научные доклады, публикации и аналитические обзоры с обоснованными выводами	

	и рекомендациями, участвует в российских и международных конференциях в области искусственного интеллекта и соревнованиях в этой области	участвует в российских и международных конференциях в области искусственного интеллекта и соревнованиях в этой области
ОПКД-4. Способен адаптировать и применять на практике для решения задач в области создания и применения технологий и систем искусственного интеллекта классические и новые научные принципы, и методы исследований	ОПКД-4.1. Решает профессиональные задачи на основе применения новых научных принципов и методов исследования ОПКД-4.2. Адаптирует и применяет на практике классические и новые научные принципы, и методы исследований для решения задач в области создания и применения технологий и систем искусственного интеллекта	Анализирует и решает профессиональные задачи на основе применения новых научных принципов и методов исследования. Применяет на практике классические и новые научные принципы, и методы исследований для решения задач в области создания и применения технологий и систем искусственного интеллекта

3. Технологическая карта самостоятельной работы студента

Коды реализуемых компетенций, индикатора(ов)	Вид деятельности студентов	Средства и технологии оценки	Объём часов, в том числе		
			СРС	Контактная работа с преподавателем	Всего
2 семестр					
ОПК-2, ОПК-2.1, ОПК-2.2,ОПК-3, ОПК-3.1, ОПК-3.2,ОПК-4, ОПК-4.1, ОПК-4.2,ОПКД-2, ОПКД-2.1, ОПКД-2.2, ОПКД-2.3, ОПКД-3, ОПКД-3.1, ОПКД-3.2, ОПКД-3.3,ОПКД-4, ОПКД-4.1, ОПКД-4.2	Подготовка к лабораторной работе	ЛР, контрольные вопросы к защите отчета, собеседование	2.50	1.00	3.50
ОПК-2, ОПК-2.1, ОПК-2.2,ОПК-3, ОПК-3.1, ОПК-3.2,ОПК-4, ОПК-4.1, ОПК-4.2,ОПКД-2, ОПКД-2.1, ОПКД-2.2, ОПКД-2.3, ОПКД-3, ОПКД-3.1, ОПКД-3.2, ОПКД-3.3,ОПКД-4, ОПКД-4.1, ОПКД-4.2	Самостоятельное изучение литературы	Вопросы по теоретическому курсу, собеседование	9.00	1.00	10.00
ОПК-2, ОПК-2.1, ОПК-2.2,ОПК-3, ОПК-3.1, ОПК-3.2,ОПК-4, ОПК-4.1, ОПК-4.2,ОПКД-2,	Подготовка к экзамену	Экзамен, вопросы к экзамену, фонд тестовых заданий Тестирование,	36.00	4.50	40.50

ОПКД-2.1, ОПКД-2.2, ОПКД-2.3, ОПКД-3, ОПКД-3.1, ОПКД-3.2, ОПКД-3.3, ОПКД-4, ОПКД-4.1, ОПКД-4.2					
Итого за 1 семестр			47.50	6.50	54.00
Итого			47.50	6.50	54.00

4. Организационно-методические рекомендации по освоению дисциплины

Самостоятельная работа студентов является важнейшим условием формирования научного способа познания. Она проводится накануне каждого лабораторного занятия и включает изучение необходимого для выполнения лабораторной работы теоретического материала, а также подготовку отчета по выполненным на лабораторном занятии заданиям.

Подготовленный материал оформляется в виде отчета.

Самостоятельные занятия (СЗ) являются одной из активных форм обучения.

Самостоятельные занятия по дисциплине «Информационные системы и технологии в научных исследованиях» имеют целью:

- закрепить и углубить знания, полученные студентами на лекциях и в процессе лабораторных занятий;
- привить практические навыки при разработке, эксплуатации и исследовании информационных процессов, систем и технологий.

Самостоятельные занятия проводятся в специализированных аудиториях, оборудованных СБТ и возможностью пользования Интернет-ресурсами, в библиотеке, а также дома.

Предлагаемые методические рекомендации содержат информацию для студентов, необходимую при подготовке и проведении лабораторных занятий по дисциплине «Информационные системы и технологии в научных исследованиях».

5. Методические указания по выполнению самостоятельной работы студентов

Методические указания должны включать следующие разделы:

- цель работы;
- задание, которое должно быть выполнено студентом в результате проведения самостоятельной работы;
- варианты индивидуальных заданий;
- основные теоретические положения, необходимые для выполнения задания, они должны быть краткими и содержать ссылки на литературу, в которой эти положения изложены в объеме, достаточном для выполнения самостоятельной работы;
- этапы выполнения задания с указанием конкретных сроков выполнения каждого из этапов и всего задания в целом;
- требования к оформлению графической и текстовой части самостоятельной работы;
- пример выполнения одного из вариантов задания и

- оформления отчета;
- библиографический список использованных источников.

6. Содержание самостоятельной работы

Наименование разделов и тем дисциплины, их краткое содержание; вид самостоятельной работы представлено в таблице ниже.

№	Наименование разделов и тем дисциплины, их краткое содержание; вид самостоятельной работы	Форма контроля
	1 семестр	
1	Подготовка к лабораторным работам	собеседование
2	Самостоятельное изучение тем: – Основы моделирования с помощью инструментальных средств – Теоретические и экспериментальные исследования – Пакеты и программное обеспечение статистической обработки результатов экспериментов – Среда обработки статистических данных R – Пакеты для научных и технических расчетов. Пакеты MATLAB, MATCAD	собеседование

Наименование тем дисциплины, цель, форма контроля, задания, требования к оформлению, перечень литературных источников представлен в таблице 3.

Таблица 3 – Наименование тем дисциплины, цель, форма контроля, задания и требования к оформлению самостоятельной работы

Название темы	Цель	Форма контроля	Задания для СРС	Требования
Подготовка к лекциям и лабораторным работам	изучить информационные системы и технологии, применяемые в научных исследованиях, а также тенденции их развития	индивидуальное собеседование	Сформулировать ответы на контрольные вопросы к лабораторным работам	Устно ответить на контрольные вопросы к лабораторным работам
Основы моделирования с помощью инструментальных средств	Изучить основы моделирования с помощью инструментальных средств			

Теоретические и экспериментальные исследования	Изучить основы теоретического и экспериментального исследований	индивидуальное собеседование	Сформулировать и письменно ответить на вопросы для контроля владения компетенциями данного раздела программы	Письменно ответить на вопросы для контроля владения компетенциями и данного раздела программы
Пакеты и программное обеспечение статистической обработки результатов экспериментов	Изучить пакеты и программное обеспечение статистической обработки результатов экспериментов			
Среда обработки статистических данных R	Изучить среду обработки статистических данных R			
Пакеты для научных и технических расчетов. Пакеты MATLAB, MATCAD	Изучить пакеты для научных и технических расчетов. Пакеты MATLAB, MATCAD			

7. Примерная тематика заданий для самостоятельной работы студентов

Учебным планом предусмотрено выполнение курсовой работы (сроки выполнения 2-й семестр, форма контроля — защита)

Примерные темы курсовых работ по дисциплине:

1. Компьютерное моделирование информационных процессов. Сравнение различных алгоритмов генерации случайного процесса .
2. Построение алгоритмов, основанных на классических методах моделирования случайных процессов, с учетом современных возможностей среды Matlab.
3. Нейросетевые алгоритмы моделирования цветных изображений пространственно распределенных и локализованных объектов.
4. Компьютерное моделирование помех в задачах компьютерной обработки изображений.
5. Моделирование каналов передачи информации в Matlab.
6. Разработка компьютерных имитационных моделей систем массового обслуживания средствами Matlab.
7. Моделирование систем массового обслуживания использованием гибридных автоматов средствами Matlab. с
8. Разработка системы компьютерного моделирования динамики намагничивающихся капель в среде Matlab.
9. Моделирование систем с адаптивной структурой.
- 10 Автоматизированные технологии создания моделей в среде Matlab .

Примерные темы для самостоятельного изучения по дисциплине «Информационные системы и технологии в научных исследованиях»:

1. Методы анализа качества информационных процессов и систем.
2. Современные средства вычислительной техники, коммуникаций и связи.
3. Применение методологии системного анализа и информационных технологий при проектировании информационных систем.
4. Перспективы и тенденции развития информационных технологий;
5. Порядок, методы и средства защиты интеллектуальной собственности.
6. Основные требования к организации труда при исследовании и разработке информационных процессов и систем.

7. План-график выполнения срс

№ п/п	Виды самостоятельной работы	Рекомендуемые источники информации (№ источника)			
		Основная	Дополнительная	Методическая	Интернет-ресурсы
1	Подготовка к лабораторной работе	1 – 8	1 – 2	1 – 2	1 – 2
2	Подготовка к лекции	1 – 8	1 – 2	1 – 2	1 – 2
3	Подготовка к контрольным вопросам	1 – 8	1 – 2	1 – 2	1 – 2
4	Самостоятельное изучение литературы	1 – 8	1 – 2	1 – 2	1 – 2

8. Организация контроля знаний студентов

8.1. Формы контроля знаний студентов

Контроль и оценка знаний, умений и навыков студентов осуществляется на лабораторных занятиях, консультациях, при сдаче экзамена. В ходе контроля знаний преподаватель оценивает понимание студентом содержания дисциплины «Информационные системы и технологии в научных исследованиях», его способность анализировать развитие информационных систем и технологий.

Контроль знаний студентов может осуществляться в следующих формах:

- текущий контроль знаний;
- итоговый контроль знаний.

Текущий контроль знаний студентов имеет целью: дать оценку работы каждого студента по усвоению им учебного материала, выявить недостатки в его подготовке и оказать практическую помощь в их устранении;

Основными формами текущего контроля знаний студентов являются:

- устный контрольный опрос;
- защита лабораторной работы;
- проверка конспектов лекций.

Устный контрольный опрос студентов проводится на лекциях (и лабораторных занятиях). По его результатам преподаватель оценивает качество подготовки студента к занятию.

На лабораторных занятиях знания и практические навыки студентов оцениваются по 4-балльной системе. Полученные оценки выставляются в журнале.

При проверке конспектов лекций дается анализ качества их ведения. Отмечаются допущенные ошибки, в рецензии преподавателя оценивается качество конспектирования учебного материала, даются рекомендации по улучшению качества конспектирования лекционного материала.

8.2. Рекомендации по подготовке к экзамену

Подготовка к зачету начинается с начала изучения дисциплины.

Необходимо посещать все лекции и лабораторные занятия. Экзамен, как итоговый контроль знаний студентов имеет целью проверить и оценить учебную работу студентов, уровень полученных знаний и практических навыков. Экзамен проводится в 1 семестре после защиты всех лабораторных работ в объеме учебной программы.

9. Рекомендации по работе с литературой и источниками

Изучение литературы и источников необходимо начинать с прочтения соответствующих глав учебных изданий, учебных пособий или литературы, рекомендованной в качестве основной или дополнительной по дисциплине «Информационные системы и технологии в научных исследованиях», которые прямо или косвенно относятся к изучаемой теме.

При изучении литературы и источников студенту рекомендуется вести краткий конспект. Однако не следует переписывать все содержание изучаемой темы, нужно выписывать лишь основные идеи и главные на ваш взгляд мысли. В отдельных случаях, когда встречаются важные определения, понятия, необходимый фактический материал и примеры, статистическая информация, имеющие отношение к изучаемой теме, студенту следует выписать их в виде цитат с полным указанием библиографических источников.

Конспектирование рекомендуемой литературы и источников необходимо вести с распределением собранных материалов по отдельным главам и параграфам согласно учебно-тематическому плану. Необходимо выписывать все выходные данные по используемой литературе и источникам.

Основой технологии интенсификации обучения на платформе цифровых образовательных технологий являются учебно-иллюстрационные материалы (опорный конспект) по дисциплине «Информационные системы и технологии в научных исследованиях».

Работа с учебно-иллюстрационными материалами имеет следующие этапы.

1. Изучение теоретических основ учебного материала в аудитории: изложение преподавателем изучаемого материала студентам с объяснением по опорному конспекту;
2. Самостоятельная работа: индивидуальная работа студентов по опорному конспекту; фронтальное закрепление по блокам опорного конспекта.
3. Первое повторение - воспроизведение содержания заданной темы опорного конспекта по памяти.
4. Устное проговаривание материала опорного конспекта – необходимый этап внешнеречевой деятельности при усвоении учебного материала.
5. Второе повторение – взаимопрос и взаимопомощь студентов друг другу.

Необходимо помнить, что главное для студента в самостоятельной работе с рекомендуемой литературой и источниками - это формирование своего индивидуального стиля, который может стать основой в будущей профессиональной деятельности.

10. Критерии оценивания компетенций

Оценка «отлично» выставляется студенту, если глубокие, исчерпывающие знания и творческие способности в понимании, изложении и использовании учебно-программного материала; логически последовательные, содержательные, полные, правильные и конкретные ответы на все поставленные вопросы и дополнительные вопросы преподавателя; свободное владение основной и дополнительной литературой, рекомендованной учебной программой.

Оценка «хорошо» выставляется студенту, если твердые и достаточно полные знания всего программного материала, правильное понимание сущности и взаимосвязи рассматриваемых процессов и явлений; последовательные, правильные, конкретные ответы на поставленные вопросы при свободном устранении замечаний по отдельным вопросам; достаточное владение литературой, рекомендованной учебной программой.

Оценка «удовлетворительно» выставляется студенту, если твердые знания и понимание основного программного материала; правильные, без грубых ошибок ответы на поставленные вопросы при устранении неточностей и несущественных ошибок в освещении отдельных положений при наводящих вопросах преподавателя; недостаточное владение литературой, рекомендованной учебной программой.

Оценка «неудовлетворительно» выставляется студенту, если неправильные ответы на основные вопросы, допущены грубые ошибки в ответах, непонимание сущности излагаемых вопросов; неуверенные и неточные ответы на дополнительные вопросы.

11. Учебно-методическое и информационное обеспечение дисциплины

11.1. Рекомендуемая литература

11.1.1. Основная литература:

1. Голубева, Н.В. Математическое моделирование систем и процессов. - Санкт-Петербург: Лань, 2021
2. Казиев, В.М. Введение в анализ, синтез и моделирование систем: Учебное пособие. - Москва, Саратов: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020
3. Поспелов, Е.А., Попов, И.С. Пакеты прикладных программ в научных исследованиях: учебно-методическое пособие. - Омск: Омский государственный университет им. Ф.М. Достоевского, 2019

11.1.2. Дополнительная литература:

1. Агалаков, С.А. Анализ данных в среде R: практикум. - Омск: Омский государственный университет им. Ф.М. Достоевского, 2020
2. Воскобойников, Ю.Е. Регрессионный анализ данных в пакете MATHCAD. - Санкт-Петербург: Лань, 2021

11.1.3. Интернет-ресурсы:

1. <http://biblioclub.ru> – ЭБС «Университетская библиотека on-line»,
<http://old.exponenta.ru/> – Образовательный математический сайт по математике и программированию
2. <http://MATLAB.exponenta.ru/Simulink/default.php> – сайт, посвященный интерактивному инструменту для моделирования Simulink,
3. <https://www.intuit.ru/> – Национальный Открытый Университет «ИНТУИТ»
4. <http://window.edu.ru/> – Информационная система «Единое окно доступа к образовательным ресурсам»
5. <http://artspb.com/> – Образовательный портал: математика, кибернетика и программирование дополнительного образования

6. <http://www.arenasimulation.com> – Официальный сайт разработчика среды имитационного моделирования ARENA
7. <https://www.anylogic.ru> – Официальный сайт разработчика AnyLogic