

МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Основы проектирования машин»

ПАКЕТЫ ПРИКЛАДНЫХ ПРОГРАММ ДЛЯ МАТЕМАТИЧЕСКОГО АНАЛИЗА

*Методические рекомендации к лабораторным работам
для студентов направления подготовки
15.03.03 «Прикладная механика»
очной формы обучения*



Могилев 2024

УДК 004.4
ББК 32.973
П74

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой «Основы проектирования машин» «15» декабря 2023 г.,
протокол № 5

Составитель канд. техн. наук, доц. О. В. Пузанова

Рецензент канд. техн. наук, доц. В. В. Кутузов

Методические рекомендации разработаны на основании рабочей программы по дисциплине «Пакеты прикладных программ для математического анализа» для студентов направления подготовки 15.03.03 «Прикладная механика» очной формы обучения и предназначены для использования при выполнении лабораторных работ.

Учебное издание

ПАКЕТЫ ПРИКЛАДНЫХ ПРОГРАММ
ДЛЯ МАТЕМАТИЧЕСКОГО АНАЛИЗА

Ответственный за выпуск	А. П. Прудников
Корректор	И. В. Голубцова
Компьютерная верстка	Н. П. Полевничая

Подписано в печать . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.
Печать трафаретная. Усл. печ. л. . Уч.-изд. л. . Тираж 26 экз. Заказ №

Издатель и полиграфическое исполнение:
Межгосударственное образовательное учреждение высшего образования
«Белорусско-Российский университет».

Свидетельство о государственной регистрации издателя,
изготовителя, распространителя печатных изданий
№ 1/156 от 07.03.2019.

Пр-т Мира, 43, 212022, г. Могилев.

© Белорусско-Российский
университет, 2024

Содержание

Введение.....	4
1 Ознакомление с интегрированной средой разработки MathCAD.....	5
2 Работа с символьным процессором в MathCAD.....	7
3 Построение и форматирование графиков в MathCAD.....	11
4 Решение задач линейного программирования в MathCAD.	13
5 Программный комплекс MATLAB. Интерфейс, основные настройки. Рабочая область, история команд, командная строка. Работа в командном режиме.	15
6 Запись выражений на языке MATLAB. Приоритет операций. Стандартные функции языка.....	18
7 Формирование векторов и матриц с требуемыми свойствами. Операции с элементами и блоками элементов матриц.....	19
8 Создание простейших программ. Реализация линейного алгоритма на языке MATLAB. Организация диалога с пользователем. Отладка программы.....	21
9 Программирование разветвляющегося алгоритма на языке MATLAB ...	22
10 Программирование циклических алгоритмов на языке MATLAB.....	24
11 Построение графиков в декартовых и полярных координатах	27
12 Организация вычислений с использованием функций пользователя	30
13 Обзор среды визуального моделирования Simulink.....	32
14 Источники и приемники сигналов Simulink.....	36
15 Математические операции с сигналами в Simulink.....	37
16 Операции отношения и логические операторы в Simulink.....	39
17 Блоки маршрутизации сигналов в Simulink	40
18 Решение алгебраических уравнений и систем в Simulink	41
19 Решение обыкновенных дифференциальных уравнений и их систем в Simulink.....	44
Список литературы	46

Введение

Целью изучения дисциплины «Математический анализ» является формирование у обучающихся знаний, умений и навыков постановки, алгоритмизации и решения инженерных задач с помощью современных средств вычислительной техники.

В результате освоения учебной дисциплины обучающийся

познает:

- основы алгоритмизации инженерных задач;
- основные принципы разработки документов в пакете MathCAD;
- принципы организации численных и символьных вычислений в пакете MathCAD;
- программирование на алгоритмическом языке MATLAB;
- принципы визуального программирования;
- технологии применения программ для компьютерного моделирования технических задач;

научится:

- ставить прикладные задачи, строить их математические модели, разрабатывать алгоритмы решения;
- реализовывать построенный алгоритм в виде собственной программы на алгоритмическом языке или с использованием стандартных программ;
- использовать разработанные программные комплексы в профессиональной деятельности;

овладеет:

- навыками самостоятельной постановки прикладных задач;
- методами компьютерного моделирования технических систем и технологических процессов;
- методами программирования, использования стандартных программ для решения задач профессиональной деятельности.

Одним из важных видов учебной деятельности обучающихся, предусмотренных учебным планом дисциплины, является выполнение лабораторных работ. Цель методических рекомендаций – оказание помощи обучающимся при их выполнении. В методических рекомендациях изложено содержание каждой лабораторной работы, даны краткие теоретические сведения по решаемой задаче и порядок выполнения работы.

Отчет содержит код программы с пояснениями в комментариях. Защита представляет собой ответы на вопросы по коду программы и ответы на контрольные вопросы к лабораторной работе.

1 Ознакомление с интегрированной средой разработки MathCAD

Цель работы: изучить структуру окна среды MathCAD; получить представление об основных возможностях среды MathCAD; приобрести навыки формирования индикаторов.

1.1 Общие сведения

В состав интегрированной среды MathCAD входят несколько связанных между собой компонентов:

- *редактор документов* для ввода и редактирования текста, формул с возможностью вставки шаблонов графиков и других объектов;
- *вычислительный процессор* для проведения расчетов согласно введенным формулам;
- *символьный процессор*, являющийся, по сути, системой искусственного интеллекта;
- *центр ресурсов* – интегратор ресурсов системы, содержащий *электронные книги* с описанием типовых расчетов в различных областях науки и техники, *справочную систему* для получения информации по тематическому и индексному каталогам, а также для поиска по ключевому слову или фразе, «*быстрые шапргалки*» – короткие примеры с минимальными комментариями, описывающие применение всех встроенных операторов и функций системы, *браузер интернета* – собственное средство для выхода в сеть Интернет.

При запуске MathCAD открывается окно приложения.

В строке заголовка за названием приложения (MathCAD) следует имя рабочего документа (**Untitled:1**). Под заголовком располагается панель главного меню из девяти пунктов. Несмотря на то, что заголовки меню почти полностью совпадают с их аналогами в прочих популярных Windows-приложениях, команды, собранные в них, во многом специфичны для MathCAD.

Под меню располагается панель инструментов, на которой собраны кнопки, дублирующие наиболее часто используемые команды главного меню. Под инструментальной панелью располагается панель форматирования, которая полезна в процессе оформления данных (в основном текстовых) в документе.

Под панелью форматирования расположены в ряд еще три панели. Щелчок по одной из кнопок математической панели приводит к тому, что отображается соответствующая палитра. Каждая палитра имеет свое назначение, а их использование существенно упрощает процесс взаимодействия пользователя с системой. В основном панели полезны при вычислениях и преобразованиях выражений.

Справа от математической панели располагается панель ресурсов, с помощью которой можно активизировать дополнительные ресурсы MathCAD. Рядом располагается панель элементов управления, позволяющая вставлять в MathCAD-документы элементы графического интерфейса (окна ввода, списки, переключатели и т. д.).

Большая белая область внизу под описанными панелями является непо-

средственно рабочей областью, где пользователем вводятся команды. Там же отображается результат их выполнения. В нижней части окна под рабочей областью видна строка состояния.

1.2 Порядок выполнения работы

1 Изучить структуру и интерфейс окна MathCAD.

2 Включить и разместить в правой части рабочего листа панели инструментов Calculator и Greek (остальные палитры панели Math, если они уже выведены на экран, отключить). Разместить данные панели в области, где размещены основные панели инструментов.

3 Ввести заголовок документа в «первой строке» рабочего листа: Практическая работа № 1. Подготовил: Иванов А. Ф., группа ПИТТ-201.

4 Сформировать идентификаторы по аналогии с примером на рисунке 1.1.

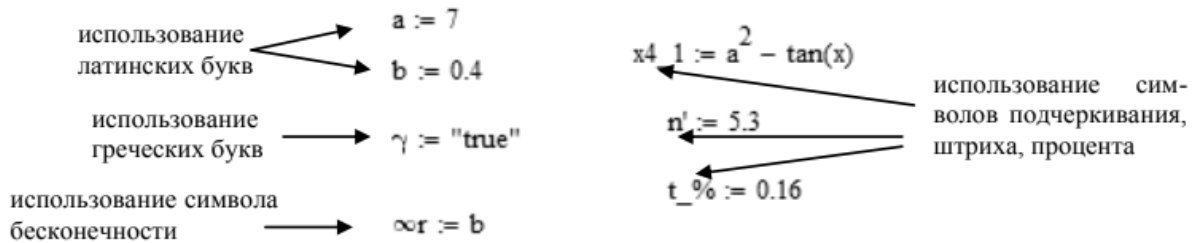


Рисунок 1.1 – Примеры формирования идентификаторов

5 Определить значение выражений:

$$1) a = \frac{\sqrt{|x-1|} - \sqrt[3]{|y|}}{1 + x^2/2 + y^2/4}, \quad b = x \left(\operatorname{arctg} z + e^{-(x+3)} \right);$$

$$2) a = \frac{3 + e^{y-1}}{1 + x^2 |y - \operatorname{tg} z|}, \quad b = 1 + |y - x| + \frac{(y-x)^2}{2} + \frac{|y-x|^3}{3};$$

$$3) a = (1+y) \frac{x+y/(x^2+4)}{e^{-(x+2)} + 1/(x^2+4)}, \quad b = 1 + \operatorname{tg}^2 \frac{z}{2};$$

$$4) a = y + \frac{x}{y^2 + \left| \frac{x^2}{y + x^3/3} \right|}, \quad b = \frac{1 + \cos(y-2)}{x^4/2 + \sin^2 z};$$

$$5) a = \frac{2 \cos(x - \pi/6)}{1/2 + \sin^2 y}, \quad b = 1 + \frac{z^2}{3 + 0,2z^2};$$

$$6) a = \frac{1 + \sin^2(x+y)}{2 + \left| x - 2x/(1 + x^2 y^2) \right|}, \quad b = \cos^2 \left(\operatorname{arctg} \frac{1}{z} \right) + x;$$

$$7) a = \ln \left| \left(y - \sqrt{|x|} \right) \left(x - \frac{y}{z + x^2/4} \right) \right|, \quad b = x - \frac{x^2}{3y} + \frac{x^5}{5y^2};$$

$$8) a = \frac{\sqrt{2-x} - \sqrt{y+6}}{x^2 + 2x + 6y} \ln \left| \sqrt{x^2 + 1} \right|, \quad b = \cos(4x^2 - y^2) + 4z^2 + 16 \ln x^2 y;$$

$$9) a = e^{-x^2} + \sqrt{\left| \sin \left(y^3 + \frac{\pi}{3} z \right) \right|}, \quad b = \frac{(2x+1)(\ln|x+3z| - \ln y)}{\sqrt{x^2 - 4yz}};$$

$$10) a = \operatorname{tg} \frac{\sqrt{1+3x^2z-1}}{x^2 + y^2 + z^2} - \sqrt{x^2 + y^2}, \quad b = \cos^2 \frac{\ln(1 - \cos(x^2))}{e^{x+2y}};$$

$$11) a = 8^{\frac{3}{x-2}} + \operatorname{lg} \frac{\sqrt{1 - \cos z}}{y}, \quad b = \frac{xyz}{\sqrt{|x| + \sqrt{|y| + \sqrt{|z|}}}};$$

$$12) a = \operatorname{lg} \frac{10x^3 + 3y - 1}{z^2 - 4x + 2y^2}, \quad b = \sin \left(\frac{x+1}{y-2} \right)^z + \sqrt{x + \sqrt{yz}}.$$

6 Ввести два любых комплексных числа и две ранжированные переменные.

7 Сохранить результаты работы в файл.

Контрольные вопросы

- 1 Какие компоненты входят в состав интегрированной среды MathCAD?
- 2 Как можно называть идентификаторы в MathCAD?
- 3 Какие символы равенства использовались в задании 5 при определении значений выражений? Что они означают?
- 4 Чем отличается текстовый регион от математического?
- 5 В какой последовательности выполняется расчет документа MathCAD?

2 Работа с символьным процессором в MathCAD

Цель работы: ознакомиться с возможностями и особенностями символьной обработки формул в пакете MathCAD в командном режиме и освоить способы создания и обработки векторов и матриц.

2.1 Общие сведения

Символьные вычисления в MathCAD можно осуществлять в двух различных вариантах: с помощью команд меню; с помощью оператора символьного вывода "→", ключевых слов символьного процессора и обычных формул (в справочной системе MathCAD этот способ называется «символьными вычис-

лениями в реальном времени»).

Первый способ более удобен, когда требуется быстро получить какой-либо аналитический результат для однократного использования, не сохраняя сам ход вычислений. Второй способ более нагляден, т. к. позволяет записывать выражения в традиционной математической форме и сохранять символьные вычисления в документах MathCAD. Кроме того, аналитические преобразования, проводимые через меню, касаются только одного, выделенного в данный момент, выражения. Соответственно, на них не влияют формулы, находящиеся в документе MathCAD выше этого выделенного выражения (например, операторы присваивания значений каким-либо переменным). Оператор символьного вывода, напротив, учитывает все предыдущее содержимое документа и выдает результат с его учетом.

Для символьных вычислений при помощи команд предназначено главное меню **Символика**, объединяющее математические операции, которые MathCAD умеет выполнять аналитически. Для реализации второго способа применяются все средства MathCAD, пригодные для численных вычислений (например, панели **Калькулятор**, **Оценка** и т. д.), и специальная математическая панель инструментов **Символика**. На ней находятся кнопки, соответствующие специфическим командам символьных преобразований, например, таким как разложение выражения на множители, расчет преобразования Лапласа и другим операциям, которые в MathCAD нельзя проводить численно и для которых, соответственно, не предусмотрены встроенные функции.

2.2 Порядок выполнения работы

- 1 Изучить возможности символьных операций MathCAD.
- 2 Выполнить символьное упрощение (рисунок 2.1).

Исходное выражение	Результат операции
$\sin(x)^2 + \cos(x)^2$	1
$\frac{a^2 - b^2}{(a + b)^2}$	$\frac{(a - b)}{(a + b)}$
$\frac{d}{dx} \sin(x)$	$\cos(x)$
$\int \frac{1}{\cos(x)^2} dx$	$\frac{1}{\cos(x)} \cdot \sin(x)$
$\sum_x (1 + x)^2$	$\frac{1}{6} \cdot x + \frac{1}{2} \cdot x^2 + \frac{1}{3} \cdot x^3$

Рисунок 2.1 – Примеры символьных упрощений

3 Выполнить операцию разложения на простые множители (факторизация) (рисунок 2.2).

Исходное выражение	Результат операции
$x^2 - y^2$	$(x - y) \cdot (x + y)$
$x^3 - 6x^2 + 11x - 6$	$(x - 1) \cdot (x - 2) \cdot (x - 3)$
$x^4 - x$	$x \cdot (x - 1) \cdot (x^2 + x + 1)$
$x^3 - 1$	$(x - 1) \cdot (x^2 + x + 1)$
123456	$2^6 \cdot 3 \cdot 643$
12345678	$2 \cdot 3^2 \cdot 47 \cdot 14593$

Рисунок 2.2 – Примеры разложения на простые множители

4 Выполнить дифференцирование (рисунок 2.3).

Исходное выражение	Результат операции
$\sin(x)$	$\cos(x)$
$\sin(x)^2 \cdot \cos(x)$	$2 \cdot \sin(x) \cdot \cos(x)^2 - \sin(x)^3$
$\log(a \cdot x^b)$	$\frac{b}{x \cdot \ln(10)}$
$(x - 1)^2 \cdot \cos(x)$	$2 \cdot (x - 1) \cdot \cos(x) - (x - 1)^2 \cdot \sin(x)$
$x \cdot e^x$	$\exp(x) + x \cdot \exp(x)$
$\operatorname{erf}(x)$	$\frac{2}{\sqrt{\pi}} \cdot \exp(-x^2)$

Рисунок 2.3 – Примеры символического дифференцирования

5 Выполнить интегрирование (рисунок 2.4).

Исходное выражение	Результат операции
$a \cdot x^n$	$\frac{1}{2} \cdot a^2 \cdot x^n$ относительно выделенной переменной a
$a \cdot x^n$	$\frac{x^{(n+1)}}{(n+1)} \cdot a$ относительно выделенной переменной x
$a \cdot x^n$	$\frac{1}{\ln(x)} \cdot x^n \cdot a$ относительно выделенной переменной n
$\ln(x)^3$	$\ln(x)^3 \cdot x - 3 \cdot x \cdot \ln(x)^2 + 6 \cdot x \cdot \ln(x) - 6 \cdot x$
$x^2 \cdot \sinh(x)$	$x^2 \cdot \cosh(x) - 2 \cdot x \cdot \sinh(x) + 2 \cdot \cosh(x)$
$\frac{1}{2 - x^2}$	$\frac{1}{2} \cdot \sqrt{2} \cdot \operatorname{atanh}\left(\frac{1}{2} \cdot x \cdot \sqrt{2}\right)$
$2 \cdot x \cdot \operatorname{erf}(x)$	$x^2 \cdot \operatorname{erf}(x) - 2 \cdot \left(\frac{-1}{2} \cdot \frac{x}{\exp\left(\frac{x^2}{2}\right)} + \frac{1}{4} \cdot \pi^{\frac{1}{2}} \cdot \operatorname{erf}(x) \right)$

Рисунок 2.4 – Примеры символического интегрирования

6 Разложить в ряд (рисунок 2.5).

Исходное выражение Ряд Тейлора

$$\frac{\sin(x)}{x} \qquad 1 - \frac{1}{6} \cdot x^2 + \frac{1}{120} \cdot x^4 + O(x^5)$$

Построение графиков функции $\sin(x)/x$ и ряда Тейлора

$$F1(x) := \frac{\sin(x)}{x} \qquad F2(x) := 1 - \frac{1}{6} \cdot x^2 + \frac{1}{120} \cdot x^4 \qquad x := -4, -3.9..4$$

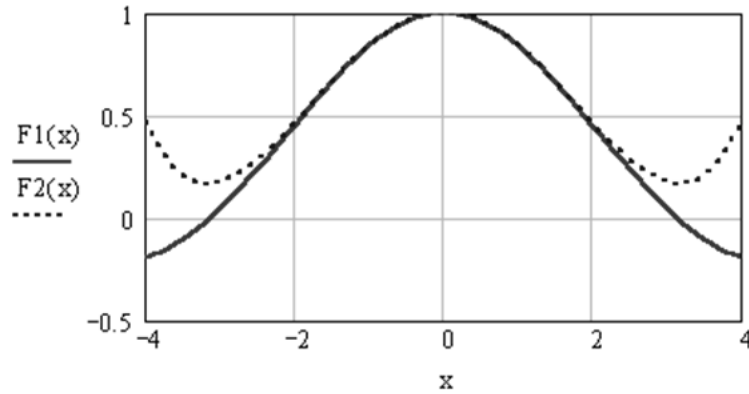


Рисунок 2.5 – Пример разложения в ряд

7 Выполнить операции с матрицами (рисунок 2.6).

Исходное выражение Результат операции

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \qquad \begin{pmatrix} a & c \\ b & d \end{pmatrix} \qquad \text{Транспонирование матрицы}$$

Последовательность команд "Обратная матрица" и "Упрощение"

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \qquad \frac{1}{(a \cdot d - b \cdot c)} \cdot \begin{pmatrix} d & -b \\ -c & a \end{pmatrix} \qquad \left[\begin{array}{cc} \frac{d}{(a \cdot d - b \cdot c)} & \frac{-b}{(a \cdot d - b \cdot c)} \\ \frac{-c}{(a \cdot d - b \cdot c)} & \frac{a}{(a \cdot d - b \cdot c)} \end{array} \right]$$

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \qquad a \cdot d - b \cdot c \qquad \text{Определитель матрицы}$$

Рисунок 2.6 – Примеры операций с матрицами

Контрольные вопросы

- 1 Назовите варианты символьных вычислений в MathCAD.
- 2 Как выполнить операцию разложения на простые множители в MathCAD?
- 3 Как выполнить дифференцирование?
- 4 Как выполнить интегрирование?
- 5 Как выполнить разложение в ряд Тейлора?

3 Построение и форматирование графиков в MathCAD

Цель работы: ознакомиться с возможностями и особенностями построения и форматирования графиков в пакете MathCAD.

3.1 Общие сведения

Для создания графиков в системе MathCAD имеется программный графический процессор. Основное внимание при его разработке было уделено обеспечению простоты задания графиков и их модификации с помощью соответствующих параметров. Процессор позволяет строить самые разные графики, например, в декартовой и полярной системе координат, графики поверхностей, трехмерные фигуры, графики уровней и т. д.

Для вывода шаблона двумерной графики в декартовой системе координат служит команда **X-Y Plot** (Декартов график) или клавиша **@**. Она выводит в текущем положении курсора шаблон двумерного графика.

Чтобы упростить создание трехмерных графиков, используется специальный мастер, разбивающий процедуру построения и форматирования графика на несколько вполне очевидных этапов. Для создания трехмерного графика с помощью мастера вначале надо ввести функцию или матрицу поверхности, например: $f(x,y) := \sin(0.2 \cdot x \cdot y)$. Далее требуется выбрать команду **Insert** (Вставка) | **Graph** (Графики) | **3D Plot Wizard** (Мастер трехмерной графики). В окне документа появится первое окно мастера – **Plot Type** (Тип графика).

3.2 Порядок выполнения работы

1 Изучить возможности построения и форматирования графиков в пакете MathCAD.

2 Построить двумерные графики функций (рисунки 3.1 и 3.2).

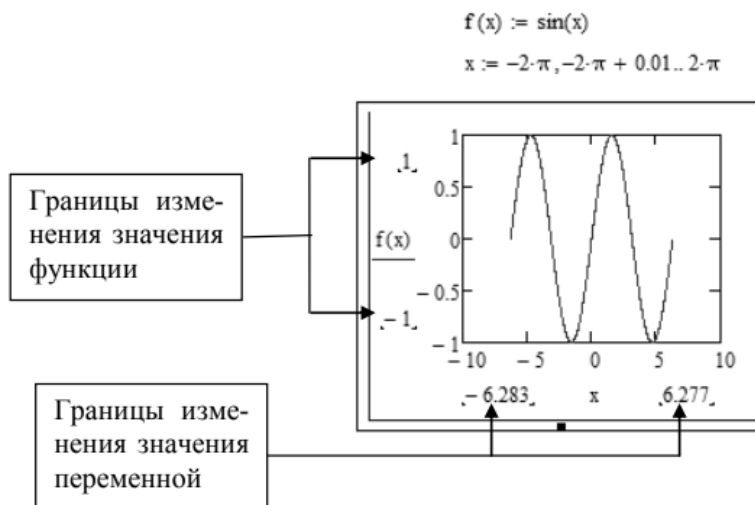


Рисунок 3.1 – Пример построения двумерного графика

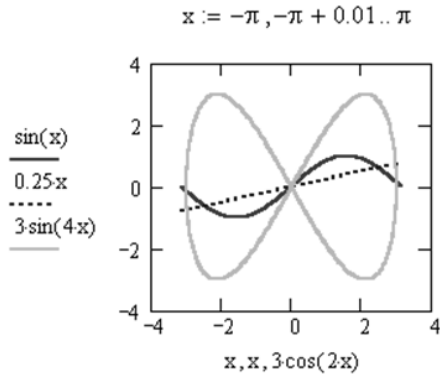


Рисунок 3.2 – Пример построения нескольких графиков на одном поле

2 Построить трехмерные графики функций (рисунок 3.3).

$$z(x, y) := \cos(x \cdot y)$$

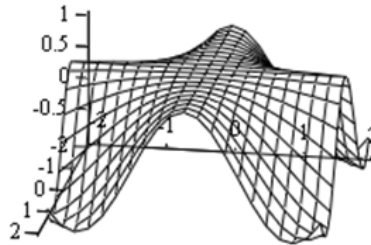


Рисунок 3.3 – Пример построения трехмерной поверхности

3 Выполнить построение двух пространств (рисунок 3.4).

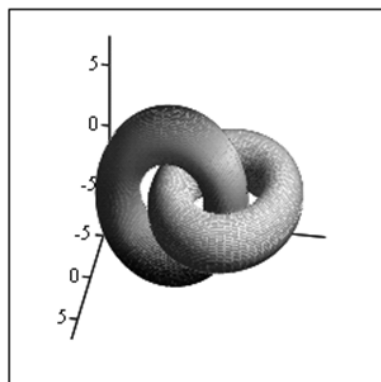
$$N := 60 \quad i := 0..N \quad \phi_i := i \cdot 2 \cdot \frac{\pi}{N}$$

$$j := 0..N \quad \theta_j := j \cdot 2 \cdot \frac{\pi}{N}$$

$$x_{i,j} := (5 + 2 \cdot \cos(\phi_i)) \cdot \cos(\theta_j)$$

$$y_{i,j} := (5 + 2 \cdot \cos(\phi_i)) \cdot \sin(\theta_j)$$

$$z_{i,j} := 2 \cdot \sin(\phi_i)$$



$$(x, y + 6, z), (z, y + 1, x)$$

Рисунок 3.4 – Пример построения пространств

Контрольные вопросы

- 1 Назовите варианты графиков, которые можно построить в MathCAD.
- 2 Какие данные надо предоставить, чтобы построить двумерные графики?
- 3 Какие данные надо предоставить, чтобы построить трехмерные графики?
- 4 Как изменить или убрать сетку на графике?
- 5 Как изменить цвет и стиль линии графика?
- 6 Как изменить масштаб графика?
- 7 Где указываются диапазоны визуализации графика?

4 Решение задач линейного программирования в MathCAD

Цель работы: приобрести практические навыки решения задач программирования в пакете MathCAD.

4.1 Общие сведения

На рисунке 4.1 приведен пример решения системы линейных уравнений.

$$A := \begin{pmatrix} 2 & 1 & 2 \\ 12 & 2 & 5 \\ 3 & 2 & 3 \end{pmatrix} \quad b := \begin{pmatrix} 2 \\ 5 \\ 1 \end{pmatrix}$$

Найдем значение определителя:
 $|A| = 7$ - отличен от нуля.

Согласно методу Крамера, вводим вспомогательные матрицы A1, A2, A3 и заменяем в них соответствующий столбец на вектор b:

$$\begin{array}{lll} A1 := A & A2 := A & A3 := A \\ A1^{(1)} := b & A2^{(2)} := b & A3^{(3)} := b \end{array}$$

Находим значения неизвестных:

$$x1 := \frac{|A1|}{|A|} = -0.286 \quad x2 := \frac{|A2|}{|A|} = -4 \quad x3 := \frac{|A3|}{|A|} = 3.286$$

Проверим решение подстановкой в исходную систему:

$$x := \begin{pmatrix} x1 \\ x2 \\ x3 \end{pmatrix} \quad A \cdot x = \begin{pmatrix} 2 \\ 5 \\ 1 \end{pmatrix}$$

Аналогичный результат можно получить, используя операцию Обращение панели Матрица:

$$x := A^{-1} \cdot b = \begin{pmatrix} -0.286 \\ -4 \\ 3.286 \end{pmatrix}$$

Рисунок 4.1 – Пример решения системы линейных уравнений

На рисунке 4.2 приведен пример решения задачи оптимизации, используя блок решения и вычисления целевой функции в оптимальной точке.

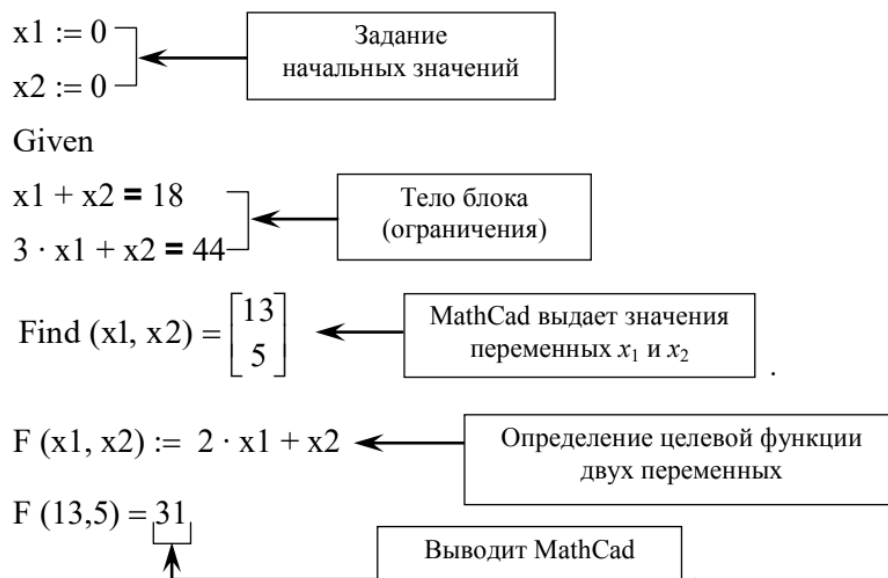


Рисунок 4.2 – Пример решения задачи оптимизации

4.2 Порядок выполнения работы

1 Решить задачу линейного программирования (рисунок 4.3).

№ п/п	Целевая функция	Ограничения	Ответ ($x_1; x_2; F_{\max(\min)}$)
1	$F = x_1 + 5x_2 \rightarrow \max$	$3x_1 - x_2 \geq 9;$ $2x_1 + 3x_2 \leq 50;$ $-x_1 + 4x_2 \geq 19$	(7; 12; 67)
2	$F = 5x_1 + 3x_2 \rightarrow \max$	$6x_1 - 5x_2 \geq 17;$ $x_1 + 2x_2 \leq 34;$ $-4x_1 + 9x_2 \geq 17$	(16; 9; 107)
3	$F = 9x_1 + 2x_2 \rightarrow \max$	$11x_1 - 3x_2 \geq 24;$ $9x_1 + 4x_2 \leq 110;$ $-2x_1 + 7x_2 \geq 15$	(10; 5; 100)
4	$F = 2x_1 + 3x_2 \rightarrow \max$	$-3x_1 + 4x_2 \leq 30;$ $3x_1 - x_2 \leq 15;$ $4x_1 + 2x_2 \geq 28$	(10; 15; 65)
5	$F = x_1 + x_2 \rightarrow \max$	$x_1 + 2x_2 \leq 14;$ $-5x_1 + 3x_2 \leq 15;$ $4x_1 + 6x_2 \geq 24;$ $x_1, x_2 \geq 0$	(14; 0; 14)
6	$F = 3x_1 + 2x_2 \rightarrow \max$	$4x_1 - 3x_2 \geq 24;$ $x_1 + 2x_2 \leq 30;$ $-2x_1 + 10x_2 \geq 20$	(18,57; 5,71; 67,14)

Рисунок 4.3 – Исходные данные для решения задачи оптимизации

2 Проанализировать и сохранить результат.

Контрольные вопросы

- 1 Какая задача линейного программирования может быть решена графическим методом?
- 2 Что такое область допустимых решений задачи линейного программирования?
- 3 Что такое линия уровня целевой функции?
- 4 Что такое вектор-градиент целевой функции? Каковы его свойства?
- 5 Поясните принцип отыскания оптимального решения линейной задачи графическим методом.
- 6 Может ли задача линейного программирования не иметь решения? Поясните этот случай графически.
- 7 Может ли задача линейного программирования иметь бесконечное множество решений? Поясните на графическом примере.
- 8 Какие виды знака равенства существуют в MathCAD? Поясните их отличия.
- 9 Как в MathCAD задать переменную, имеющую более одного значения (изменяющуюся при каждом обращении к ней)? Для какой цели в работе используется такая переменная?
- 10 Как в MathCAD получить результат вычислений по некоторой формуле?
- 11 Как в MathCAD из относительного равенства выразить одну переменную через другую?
- 12 Поясните порядок ввода сложного выражения в MathCAD.
- 13 Как решить систему уравнений с помощью MathCAD?
- 14 Расскажите о принципах работы с графиком в MathCAD: вставка графика; настройка графика; вывод графиков нескольких функций одновременно.

5 Программный комплекс MATLAB. Интерфейс, основные настройки. Рабочая область, история команд, командная строка. Работа в командном режиме

Цель работы: изучить структуру окна системы MATLAB; получить представление об основных возможностях системы MATLAB; приобрести навыки выполнения вычислений в командном режиме работы MATLAB.

5.1 Общие сведения

В состав MATLAB входят интерпретатор команд, графическая оболочка, редактор-отладчик, библиотеки команд, компилятор, символьное ядро для проведения аналитических вычислений, математические библиотеки MATLAB, генератор отчетов и инструментарий.

Интерфейс (рисунок 5.1) многооконный и имеет ряд средств прямого доступа к различным компонентам системы. Вся работа организуется через командное окно (*Command Window*), которое появляется при запуске

программы. В процессе работы данные располагаются в памяти (*Workspace*) в виде матриц.

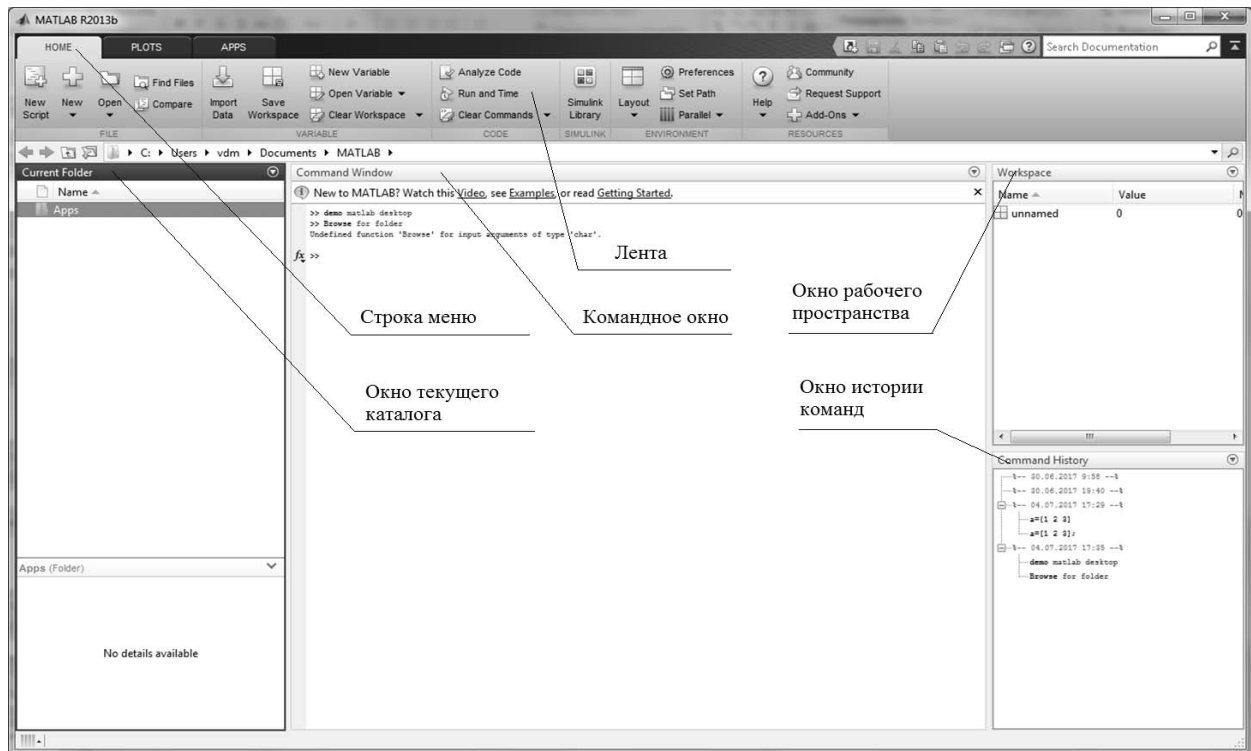


Рисунок 5.1 – Интерфейс программы MATLAB

В начале работы с системой MATLAB следует выбрать активную папку, которая будет использоваться при выполнении работы с файлами. С этой целью необходимо мышью нажать на кнопку *Browse for folder*.

Командное окно разделено на две зоны: зону просмотра и зону редактирования. Зона редактирования находится в строке командного окна, отмеченной знаком «>>», после которого записывается вычисляемое выражение.

Нажатие клавиши *Enter* заставляет систему вычислить выражение и показать результат. Если запись оператора не заканчивается символом «;», то результат выводится в командное окно, в противном случае – не выводится.

Если оператор не содержит знака присваивания «=», то значение результата присваивается системной переменной *ans*.

Клавиши-стрелки курсора *Вверх* и *Вниз* используются для подстановки после маркера строки ввода «>>» ранее введенных строк из стека команд для их исправления, дублирования или дополнения.

Стек команд – это область оперативной памяти ПК, отведенная для хранения выполненных в данном сеансе работы команд.

Все значения переменных, вычисленные в течение текущего сеанса работы, сохраняются в специально зарезервированной области памяти компьютера, называемой рабочим пространством системы MATLAB (*Workspace*).

Для просмотра значения любой переменной из текущего рабочего пространства системы достаточно набрать ее имя и нажать клавишу *Enter*.

Для сохранения рабочего пространства в файле с расширением **.mat* нуж-

но выполнить команду *Save Workspace*.

Команды *clc* и *clear* отвечают за очистку командного окна и буфера команд с рабочим пространством соответственно.

При работе в среде MATLAB для упрощения записи выражений целесообразно применять стандартные функции, являющиеся частью языка, их назначение и принципы использования определены разработчиком языка. Имена этих функций недопустимо указывать в качестве имен переменных.

Для представления чисел на экране имеются разные форматы. Нужный формат может быть определен в меню (*File/Preferences*) либо при помощи команды *Format*.

Алфавит MATLAB включает:

- прописные и строчные буквы латинского алфавита;
- арабские цифры от 0 до 9;
- специальные символы.

Переменная – именованная область памяти, в которой хранятся данные определенного типа, которые можно изменить в любой момент работы программы. Имя переменной должно начинаться с буквы, за ней могут идти буквы, цифры и символ подчеркивания. Допустимы имена любой длины, но MATLAB идентифицирует их по первым 31 символу и различает прописные и строчные буквы.

Арифметические выражения представляют собой последовательности чисел, констант, переменных и функций, объединенных знаками арифметических операций. Операции одинакового приоритета выполняются в порядке слева направо. Умножение и деление имеют более высокий приоритет перед вычитанием и сложением, но круглые скобки могут изменить порядок.

1.2 Порядок выполнения работы

1 Изучить структуру и интерфейс окна MATLAB.

2 Командами *Command Window*, *Current Directory*, *Workspace* из группы *Environment* отработать отображение основных окон MATLAB.

3 Вычислить значения выражений по заданию преподавателя, повторить расчет, изменив исходные данные в истории команд.

4 Сохранить результаты из рабочего пространства в файл.

5 Изменить формат представления результатов.

Контрольные вопросы

1 Как обозначена строка ввода, как повторить ранее введенную команду?

2 Каковы основные системные переменные, стандартные функции и специальные символы MATLAB и их назначение?

3 Каковы основные элементы интерфейса MATLAB?

4 Что такое переменная?

5 Как MATLAB представляет введенные значения и переменные?

6 Какие существуют правила именования переменных в MATLAB?

7 Каков состав алфавита языка MATLAB?

8 Для чего предназначены команды *clc* и *clear*?

6 Запись выражений на языке MATLAB. Приоритет операций. Стандартные функции языка

Цель работы: ознакомиться с языком программирования системы MATLAB; изучить структуру интерфейса редактора/отладчика m-файлов; научиться записывать выражения с использованием стандартных функций MATLAB; изучить программный режим работы; научиться создавать файлы-сценарии.

6.1 Общие сведения

Работать в MATLAB можно в командном и программном режимах.

В командном режиме исправить ошибки без повторного введения последовательности команд после нажатия клавиши *Enter* уже нельзя.

Работа в программном режиме осуществляется путем вызова имени программы, написанной пользователем на языке программирования MATLAB.

Достоинство программного способа в том, что он позволяет сохранять разработанные вычислительные алгоритмы и благодаря этому повторять вычисления при других исходных данных.

Записывать программы, их редактировать и отлаживать необходимо в m-файлах, которые делятся на два класса:

- 1) файлы-сценарии, не имеющие входных параметров;
- 2) файлы-функции, имеющие входные параметры.

Для создания файла сценария выполняется команда *New script* вкладки *File*, получаем окно редактора m-файла. Редактор позволяет создать m-файл.

Редактор дает возможность устанавливать в тексте файла специальные метки, именуемые точками прерывания (*breakpoints*), для оценки промежуточных результатов. Строки программы в нем нумеруются в последовательном порядке. Редактор является многооконным. Окно каждой программы оформляется как вкладка.

Редактор/отладчик m-файлов выполняет синтаксический контроль программного кода по мере ввода текста. При этом используются следующие цветовые выделения: ключевые слова – синий цвет; операторы, константы и переменные – черный цвет; комментарии – зеленый цвет; символьные переменные – фиолетовый цвет; синтаксические ошибки – красный цвет.

По умолчанию m-файл имеет название *Untitled* (Безымянный). Чтобы дать ему имя, необходимо в меню *File* выполнить команду *Save as* и в диалоговом окне указать папку и имя этого файла.

Запуск файла сценария производится кнопкой *Выполнить* или командой, содержащей имя m-файла. Результат выполнения появится в окне команд.

Файл-сценарий имеет следующую структуру:

% Основной комментарий
 % Дополнительный комментарий
 Тело программы

Для использования m-файла необходимо его вызвать в командном окне по имени. Для блокировки вывода на экран промежуточных результатов в теле программы все операторы завершаются символом «;».

Переменные, использованные в m-файле, для вычисления выражения должны быть определены перед его вызовом в командном окне, при работе программы с различными исходными данными, либо в тексте m-файла, что ведет к выполнению m-файла только с этими исходными данными.

6.2 Порядок выполнения работы

- 1 Создать программу на языке MATLAB для вычисления выражений при заданных исходных данных по заданию преподавателя.
- 2 Сохранить результат в m-файл.
- 3 Установить точку прерывания перед выполнением расчета по формуле.

Контрольные вопросы

- 1 Каковы преимущества программного режима работы в MATLAB?
- 2 Как создать, сохранить, открыть, выполнить m-файл?
- 3 Как использовать m-файл для расчета с разными исходными данными?
- 4 Для чего используют комментарии? Как записать комментарий?
- 5 Какие переменные, выражения и стандартные функции MATLAB использованы в программе?
- 6 Как идентифицируются строки программы в окне редактора?

7 Формирование векторов и матриц с требуемыми свойствами. Операции с элементами и блоками элементов матриц

Цель работы: получить навыки работы с массивами, векторами и матрицами в программном режиме.

7.1 Общие сведения

Все данные MATLAB представляет в виде массивов. Массив – упорядоченная, пронумерованная совокупность однородных данных.

MATLAB не требует от пользователя предварительного задания размерности массива. Пользователь может вводить элементы массива последовательно, при этом MATLAB будет динамически перестраивать его структуру.

Массивы в MATLAB состоят из элементов типа *double*.

Под вектором в MATLAB понимается одномерный массив чисел, а под

матрицей – двумерный массив.

Для создания одномерных массивов используют:

– операцию конкатенации: $z = [1\ 2\ 3]$ – строка, $b = [1;2;3]$ – столбец;

– операцию индексации для экономии ресурсов присваивание значений элементам массива начинают с последних: $z(3) = 3$; $z(2) = 2$; $z(1) = 1$;

– вызов специальных функций:

zeros(m,n) – создает матрицу размером $m \times n$ с нулевыми элементами;

ones(m,n) – создает матрицу размером $m \times n$ с единичными элементами;

eye(m,n) – создает единичную матрицу размером $m \times n$;

rand(m,n) – создает матрицу $m \times n$ из случайных чисел, равномерно распределенных в диапазоне от 0 до 1;

– специальную операцию, обозначаемую двоеточием: $z = 1:1:3$.

При необходимости изменить элемент сформированного одномерного массива используют операцию индексации и операцию присваивания: $z(3) = 78$.

Запись несуществующего элемента массива означает добавление нового элемента к уже существующему массиву: $z(4) = 7$.

Удлинение массива z можно выполнить и с помощью операции конкатенации: $z = [z\ 7]$. Можно подвергнуть конкатенации и несколько массивов: $y = [z\ z\ 9\ z]$.

Для создания двумерного массива (матрицы) в MATLAB используют операции конкатенации, индексации и вызов специальных функций.

Значения элементов матрицы вводятся в квадратных скобках по строкам, при этом элементы строки матрицы разделяются пробелом или запятой, а строки отделяются одна от другой точкой с запятой: $X = [1\ 2; 3\ 4; 5\ 6]$.

Двумерные массивы можно создать с помощью операции индексации, прописывая по отдельности элементы необходимыми числовыми значениями: $X(3, 3) = 1$; $X(3,2) = 2$ и т. д.

Выделение блоков матриц осуществляется индексацией при помощи двоеточия: $X1 = X(2:3, 2:3)$.

Для чтения из массивов используют операцию индексации справа от знака присваивания: $y = z(2)$.

Для выделения из матрицы столбца или строки следует в качестве одного из индексов использовать номер столбца или строки матрицы, а другой индекс заменить двоеточием: $x = X(2, :)$.

При выделении блока до конца матрицы можно не указывать его размеры, а использовать слово **end**: $x = X(2, 2:end)$.

Для удаления строк или столбцов присваивают им пустой массив: $[]$.

7.2 Порядок выполнения работы

1 Создать по заданию преподавателя массивы в командном режиме, произвести их обработку с помощью программы-скрипта на языке MATLAB.

2 Сохранить файл на диске. Запустить его на выполнение. Просмотреть результаты в Workspace.

Контрольные вопросы

- 1 Какие способы создания матриц Вам известны?
- 2 Как заполнить массив различными способами?
- 3 Как создать дополнительные элементы вектора или матрицы?
- 4 Как удаляются строки и столбцы матриц?
- 5 Как создать пустой массив, выделить из матрицы строку или столбец?
- 6 Какие стандартные функции по работе с массивами Вам известны?
- 7 Назовите основные поэлементные действия над матрицами.

8 Создание простейших программ. Реализация линейного алгоритма на языке MATLAB. Организация диалога с пользователем. Отладка программы

Цель работы: получить навыки работы в режиме диалога с пользователем и решения задач на основе линейного алгоритма в программном режиме работы.

8.1 Общие сведения

При составлении программ с линейным алгоритмом одним из основных операторов является оператор присваивания:

$$\text{Имя_переменной} = \text{Выражение},$$

где Имя_переменной – имя простой переменной, вектора, матрицы, функции;

Выражение – арифметическое, логическое или строковое выражение.

Тип имени переменной определяется системой автоматически по типу выражения.

Работа оператора: вычисляется выражение справа от «=» и результат присваивается переменной слева от «=».

В MATLAB в качестве оператора ввода используется функция `input`, которую принято называть оператором ввода:

$$\text{Имя_переменной} = \mathbf{input}(\text{Символьная константа}),$$

где Символьная константа – набор символов, заключенный в апострофы.

Пока пользователь не ввел значение в ответ на запрос программы, блокируется дальнейшее выполнение программы.

Если необходимо вывести данные на экран дисплея, применяется функция `disp`, которую принято называть оператором вывода:

$$\mathbf{disp}(\text{Выражение}),$$

где Выражение – это арифметическое, логическое или символьное выражение.

Символы и текстовые строки в MATLAB всегда вводятся при помощи простых кавычек: `s = 'Привет'`.

Над строками и символьными данными выполняется операция конкатенации: `h = [s+'от MATLAB']`, `s` – символьная переменная, `'от MATLAB'` – строковое значение.

Если в строковом выражении, создаваемом конкатенацией, необходимо использовать число, его преобразуют в строку, используя функции преобразования.

8.2 Порядок выполнения работы

1 Создать `m`-файл для решения задач, реализующих линейный алгоритм, по вариантам, используя режим диалога с пользователем.

2 Организовать вызов `m`-файла для расчета с другими исходными данными.

Контрольные вопросы

1 Какой оператор (операция) реализует линейный алгоритм?

2 Что происходит при операции присвоения?

3 Какой тип данных используют в аргументе функций `input` и `disp`?

4 Как объединить строковые значения в одно выражение?

5 Для чего используют функции преобразования?

9 Программирование разветвляющегося алгоритма на языке MATLAB

Цель работы: изучить операции отношения, логические операции и условные операторы; приобрести навыки их использования при реализации разветвляющихся алгоритмов.

9.1 Общие сведения

Разветвляющийся алгоритм – алгоритм, содержащий хотя бы одно условие, обеспечивающий ветвление основного потока программы.

Для организации ветвлений в MATLAB используются условные операторы `if` и оператор переключения `switch`.

В качестве условий в условных операторах используют логические выражения, которые строятся из арифметических выражений, операндов (значений, констант, переменных, функций) логических операций и отношений, результатом вычисления логических выражений являются логические значения «ложь» (`false`) и «истина» (`true`).

Полная форма 1:

```

if Условие 1
    Инструкции_1
elseif Условие 2
    Инструкции_2
else
    Инструкции_3
End

```

Полная форма 2:

```

if Условие
    Инструкции_1
else
    Инструкции_2
end

```

Сокращенная форма:

```

if Условие
    Инструкции_1
end

```

Инструкции_1, Инструкции_2, Инструкции_3 – операторы и функции для обработки данных

Работа оператора **if**:

- если условие возвращает логическое значение «истина», выполняются инструкции_1 и управление передается оператору, следующему за оператором **if**;
- если условие возвращает логическое значение «ложь», проверяется условие 2 (если есть), выполняются инструкции_2 и управление передается оператору, следующему за оператором **if** и т. д.;
- в сокращенной форме, если условие возвращает значение «ложь», то инструкции не выполняются, а управление передается оператору, следующему за оператором **if**.

Для осуществления множественного выбора (или ветвления) используется конструкция с переключателем типа **switch**.

Полная форма:

```

switch Выражение селектор
    case список констант 1, Список инструкций_1
    case список констант 2, Список инструкций_2
        ...
    case список констант n, Список инструкций_n
otherwise Список инструкций_N+1

end

```

Сокращенная форма:

```

switch Выражение селектор
    case список констант 1, Список инструкций_1
    case список констант 2, Список инструкций_2

```

...
case список констант n, Список инструкций_n
end

Работа оператора switch:

- вычисляется выражение селектор;
- последовательно проверяется совпадение значения селектора со значениями списка констант, если совпадение с каким-либо списком констант есть, то выполняются инструкции, соответствующие этому списку констант, при этом следующие далее списки констант не проверяются и управление передается оператору следующему за switch, если совпадений ни с одним списком констант нет, то выполняются инструкции, следующие за словом otherwise, и управление передается оператору, следующему за switch;
- в сокращенной форме, если совпадений ни с одним списком констант нет, то управление передается оператору, следующему за switch.

9.2 Порядок выполнения работы

1 Создать m-файл для решения задач с использованием условного оператора по вариантам, ввод исходных данных выполнить в режиме диалога с пользователем.

2 Создать m-файл для решения задач с использованием оператора выбора по вариантам, ввод и вывод данных выполнить в режиме диалога с пользователем.

Контрольные вопросы

- 1 Для чего используются условные операторы?
- 2 В чем отличие операторов if и switch?
- 3 Какой тип выражений используют в качестве условий?
- 4 Каковы основные логические операции и отношения?
- 5 Какие значения может принимать результат логического выражения?
- 6 Каковы синтаксис и работа использованных операторов?

10 Программирование циклических алгоритмов на языке MATLAB

Цель работы: приобрести навыки использования операторов повтора при организации вычислений в циклических алгоритмах.

10.1 Общие сведения

Циклический алгоритм – алгоритм, предусматривающий многократное повторение одного и того же действия над новыми исходными данными. К цик-

лическим алгоритмам сводится большинство методов вычислений, перебора вариантов. Многократно повторяющиеся участки называются циклами или телом цикла.

В MATLAB используют:

- оператор цикла с параметрами **for**;
- оператор цикла с предусловием **while**.

Оператор цикла **for** предназначен для программирования циклических алгоритмов с заданным числом повторений.

Синтаксис при явно заданном шаге изменения переменной цикла:

```
for Имя ПЦ= НПЦ : ШПЦ : КПЦ
    Инструкции
End
```

Синтаксис при единичном шаге изменения переменной цикла:

```
for Имя ПЦ = НПЦ : КПЦ
    Инструкции
End
```

где Имя ПЦ – идентификатор переменной цикла;

НПЦ – начальное значение переменной цикла;

КПЦ – конечное значение переменной цикла;

ШПЦ – шаг изменения переменной цикла при повторении цикла;

Инструкции – оператор или группа операторов, составляющих тело цикла.

Работа оператора **for**:

- 1) проверяется условие Начальное значение ПЦ \leq Конечное значение ПЦ;
- 2) если условие не выполняется, то оператор цикла прекращает свою работу и выполняется следующий за ним оператор программы;
- 3) если условие выполняется, то переменной цикла присваивается ее начальное значение (или принимается следующее с заданным шагом);
- 4) выполняется тело цикла (инструкции);
- 5) переменная цикла увеличивается на шаг и выполняется шаг 4;
- 6) шаг 5 повторяется, пока значение ПЦ \leq КПЦ, в противном случае оператор цикла прекращает свою работу и выполняется следующий за ним оператор программы.

Корректное задание параметров цикла определяется выражением $\text{НПЦ} \leq \text{КПЦ}$ при $\text{ШПЦ} > 0$, $\text{НПЦ} \geq \text{КПЦ}$ при $\text{ШПЦ} < 0$.

Количество повторений цикла определяется из зависимости $(\text{КПЦ} - \text{НПЦ}) / \text{ШПЦ} + 1$.

Оператор **while** предназначен для программирования любых циклов, где проверка условия повторения цикла выполняется перед выполнением рабочей части цикла и действий, число которых заранее не известно, но известно условие продолжения цикла:

while ЛВ
 Инструкции
end

где ЛВ – логическое выражение.

Работа оператора **while**:

- 1) проверяется (вычисляется) значение логического выражения;
- 2) пока оно истинно, выполняется тело цикла (инструкции);
- 3) повторяются шаги 1 и 2;
- 4) если логическое выражение принимает ложное значение, то работа оператора прекращается и выполняется следующий за оператором **while** оператор программы.

Особенности использования оператора **while**:

- если переменная цикла явно выражена, нужно присвоить ей начальное значение до оператора цикла;
- необходимо предусмотреть изменение переменной цикла в теле цикла;
- переменная цикла должна использоваться в логическом выражении условия повторения цикла;
- если логическое выражение в заголовке цикла **while** заведомо ложно, то цикл может не выполниться ни разу.

В управляющих структурах циклов используются операторы, влияющие на их выполнение. Для этого в теле цикла предусматривают дополнительное условие, при котором выполняют оператор **break** для досрочного прерывания выполнения цикла или оператор **continue** для передачи управления в следующую итерацию цикла, пропуская операторы, которые записаны за ним, причем во вложенном цикле он передает управление на следующую итерацию основного цикла.

В теле любого оператора цикла могут находиться другие операторы цикла. При этом цикл, содержащий в себе другой, называют внешним, а цикл, находящийся в теле первого, – внутренним (вложенным). Таких вложенных циклов во внешнем цикле может быть несколько. Все операторы внутреннего цикла должны полностью располагаться в теле внешнего цикла.

При вложении циклов **for** внутренний цикл выполняется полностью от начального до конечного значения его параметра цикла при неизменном значении параметра внешнего цикла. Затем значение параметра внешнего цикла изменяется на единицу, и опять от начала и до конца выполняется вложенный цикл. И так до тех пор, пока значение параметра внешнего цикла не станет больше конечного значения параметра внешнего цикла.

Аналогично осуществляется работа циклов **while** за исключением того, что проверка условия выполнения внешнего цикла производится только после всех итераций внутреннего цикла.

Возможна комбинация разных операторов цикла при использовании вложенности, условия их работы определяются комбинированием вышеуказанных правил.

10.2 Порядок выполнения работы

1 Создать программу для вычисления значения суммы ряда по вариантам с использованием операторов `for` и `while` и сравнить результаты вычислений.

2 Изменить созданную программу для досрочного выхода из цикла, когда номер суммируемого элемента ряда станет равен номеру варианта.

3 Создать программу для решения задачи по вариантам с использованием вложенных циклов.

Контрольные вопросы

1 Для чего используются операторы повтора?

2 Каковы синтаксис и работа операторов `for` и `while`?

3 Назовите основные требования к организации циклов `for` и `while`.

4 Какие способы управления работой цикла Вы знаете?

5 Что такое вложенные циклы?

6 Правила работы вложенных циклов.

11 Построение графиков в декартовых и полярных координатах

Цель работы: приобрести практические навыки использования средств MATLAB для представления данных в графической форме.

11.1 Общие сведения

Основной функцией, обеспечивающей построение графиков на экране дисплея, является функция `plot(x1, y1, s1, x2, y2, s2,...)`, где `x1, y1, x2, y2` – векторы значений аргументов и функций, `s1, s2, ...` – символьные переменные, определяющие тип линии, тип точки, цвет линии.

Наиболее простыми частными случаями являются команды:

– `plot(X, Y)` – для построения графика функций одной переменной;

– `plot(X, Y, S)` – тип линии графика можно задавать с помощью строковой константы `S`;

– для построения нескольких графиков в одних и тех же координатных осях, функция `plot` записывается следующим образом: `plot(x1,y1,x2,y2)`.

Если переменные `s` не указаны, то тип линии – сплошной отрезок прямой, тип точки – пиксел.

Цвет устанавливается в очередности построения графиков: синий, зеленый, красный, голубой, фиолетовый, желтый, черный и белый.

При построении графиков толщина линии задается параметром ('LineWidth', толщина линии в пикселях), который указывается после основных параметров.

Графики в MATLAB выводятся в отдельное графическое окно, которое называют фигурой. Для создания нового графического окна используют функцию **figure**.

Для того чтобы на экране всегда отображалось два окна с нужными графиками, при вызове функции **figure** необходимо в качестве аргумента указывать номер графического окна, которое следует создать или сделать активным, если оно уже создано (**figure(1)**, **figure(2)** и т. д.).

Отображение двух графиков в одном графическом окне обеспечивает функция: **subplot**(число строк, число столбцов, номер подокна).

Для построения графика в трехмерных координатных осях, используется функция **plot3(X,Y,Z)**, где в качестве первых двух аргументов принимаются матрицы с координатами точек по осям Ox и Oy соответственно, а в качестве третьего передается матрица значений точек по оси Oz .

Для получения матрицы координат x и y получающейся прямоугольной сетки применяют функцию **meshgrid**: $[X , Y] = \text{meshgrid}(x , y)$.

Для отображения графика в виде сетки используют функцию **mesh(X,Y,Z)**, при этом получается график, образованный интерполяцией точек массивов X , Y и Z линиями по осям Ox и Oy .

Цветом указывается уровень точки по оси Oz от самого малого значения (синего) до самого большого (красного) и производится удаление «невидимых» линий.

Для управления прозрачностью графика используют команды:

- **hidden off** – прозрачность включена;
- **hidden on** – прозрачность выключена.

В системе MATLAB предусмотрена функция визуализации непрерывной поверхности в трехмерных осях **surf(X,Y,Z)**, которая строит каркасную поверхность графика функции и заливает каждую клетку поверхности цветом, зависящим от значения функции в точках, соответствующих углам клетки.

Команда **shading flat** позволяет убрать каркасные линии.

Для получения поверхности, плавно залитой цветом, зависящим от значения функции, предназначена команда **shading interp**.

Вернуться к начальному виду поверхности можно с помощью команды **shading faceted**.

Для изменения цветовой карты используется функция **colormap**(имя карты).

Для управления созданным окном графика применяется команда **set(object name, 'Property name1', Property value1, ...)**, где **object name** – имя управляемого объекта, **'Property name1'** – строковая константа, содержащая название изменяемого свойства, **'Property value1'** – строковая константа, содержащая значение изменяемого свойства.

Для установки титульной надписи предназначена команда **title(string)**. Для подписей осей x , y и z применяются команды **xlabel(string)**, **ylabel(string)**, **zlabel(string)**, содержание определяется строковой переменной **string**.

Пояснение в виде отрезков линий со справочными надписями, размещаемое внутри графика или около него, называется легендой и создается командой:

– **legend**(string1,string2,string3,...) – добавляет к текущему графику легенду в виде строк, указанных в списке параметров;

– **legend**(string1,string2,string3,...,Pos) – помещает легенду в точно определенное место, специфицированное параметром Pos: Pos = 0 – автоматически; Pos = 1 – верхний правый угол; Pos = 2 – верхний левый угол; Pos = 3 – нижний левый угол; Pos = 4 – нижний правый угол; Pos = -1 – справа от графика.

Изменить масштабирование графиков позволяет команда:

– **axis**([xmin xmax ymin ymax]) – установка диапазонов координат по осям x и y для текущего двумерного графика;

– **axis auto** – установка параметров осей по умолчанию.

Для улучшения чтения графиков среда MATLAB позволяет создавать масштабную сетку:

– **grid on** – добавляет сетку к текущему графику;

– **grid off** – отключает сетку.

Для построения наложенных друг на друга графиков в одном и том же окне служит команда управления графическими построениями:

– **hold on** – обеспечивает продолжение вывода графиков в текущее окно, что позволяет добавлять последующие графики к уже существующим;

– **hold off** – отменяет режим продолжения графических построений.

Для вывода текста используется команда **text**(x,y, 'текст'), которая создает текстовую надпись в координатах (x,y).

Изменить свойства линии, вывести на график обозначения осей, легенду, включить сетку можно с помощью свойств графика в вызываемой в меню графического окна команде *Edit Plot*.

Для построения графиков функций в полярной системе координат используется функция **polar**(Theta,R,S), которая строит график в полярной системе координат, представляющий собой положение конца радиус-вектора с длиной R и углом Theta, строковая константа S задает стиль построения.

11.2 Порядок выполнения работы

1 Вычислить вектор y значений функции y(x).

2 Построить график функций в декартовых координатах с использованием различных типов и толщины линий, цвета и маркера.

3 Вычислить функцию с пределами изменения x от -2π до 2π и y от -2π до 2π и построить трехмерный график.

4 Вывести в одно графическое окно функцию в виде трехмерных графиков разного типа (используя три разные команды вывода трехмерных графиков).

Контрольные вопросы

1 Какие команды используются для построения графиков?

2 Каков порядок построения графиков?

3 Как создать новое окно для построения графика?

4 Как создать дополнительное подокно для вывода графика?

- 5 Как создать в одном окне наложенные друг на друга графики?
- 6 Как редактируется внешний вид графиков?
- 7 Как использовать греческие буквы и специальные символы?

12 Организация вычислений с использованием функций пользователя

Цель работы: изучить возможности MATLAB для организации вычислений повторяющихся алгоритмов; приобрести практические навыки создания и использования функций пользователя.

12.1 Общие сведения

В программировании встречаются случаи, когда приходится выполнять одни и те же вычисления, но при различных исходных данных. Для исключения одинаковых фрагментов программы повторяющиеся вычисления выделяют в самостоятельную часть – подпрограмму.

Подпрограммой называется автономная часть программы, реализующая определенный алгоритм и допускающая обращение к ней из различных частей общей программы неограниченное число раз по мере необходимости.

Преимущества использования подпрограмм:

- отсутствие дублирования, что сокращает трудоемкость создания программы;
- более удобный процесс отладки и внесения изменений;
- повышение надежности программы.

В языке MATLAB все подпрограммы можно разделить на встроенные и пользовательские.

Встроенные (стандартные) подпрограммы являются частью языка и могут вызываться по имени без предварительного описания.

Пользовательские (нестандартные) подпрограммы пользователя пишутся самим программистом в соответствии с синтаксисом языка.

Передача данных в подпрограмму и возврат результата выполнения осуществляются с помощью параметров. Различают формальные и фактические параметры. Параметры, которые указываются при объявлении (описании) подпрограммы, называются формальными. Параметры, которые указываются при вызове подпрограммы, называются фактическими.

Количество фактических параметров должно соответствовать числу формальных параметров.

В языке MATLAB выделяют два вида функций пользователя – встроенные и внешние.

Для создания функции вручную используют окно редактора m-файлов либо команду *New function* вкладки *File*.

Структура описания внешней функции:

```
function[выходные параметры]=Имя функции(входные параметры)
%Тело функции – любые выражения и операторы MATLAB
end
```

Слово **end** допускается не указывать за исключением случая, когда используется вложенность функций.

Вызов функции в основной программе осуществляется по ее имени с указанием фактических входных параметров.

После завершения работы функции фактические значения присваиваются выходным параметрам.

В простейшем случае у функции есть один входной и один выходной параметры. У функции может быть несколько входных и несколько выходных параметров. Список как входных, так и выходных параметров может быть пустым, и получение исходных данных для проведения вычислений и вывод полученных результатов могут осуществляться исключительно с помощью операций ввода-вывода.

Встраиваемая функция создается с помощью процедуры `inline` или анонимной командой `@` непосредственно в командном окне и часто называется `inline`-функцией:

```
Имя функции=inline (выражение, список аргументов);
Имя функции=@(аргументы)выражение.
```

Определяющее функцию выражение и имена аргументов передаются в `inline`-функцию только в виде строковых выражений. У `inline`-функций может быть только один выходной параметр, совпадающий с именем самой функции.

MATLAB допускает два способа описания нескольких функций внутри одного файла:

- 1) основная функция и ее подфункции;
- 2) основная функция и вложенные функции.

Такой `m`-файл с программой-функцией может содержать описание не одной, а нескольких функций. Имя главной функции должно совпадать с именем файла. При вызове функции можно будет обратиться только к той функции, имя которой совпадает с именем `m`-файла. Все остальные функции называются внутренними (подфункциями) и вызываются только из главной функции.

Чтобы создать вложенную функцию, нужно разместить ее тело внутри тела другой функции, каждая из функций должна заканчиваться ключевым словом `end`. Количество уровней вложенности не ограничено.

Область видимости – это та часть программного кода, в которой может быть использована данная переменная. Видимость зависит от того, на каком уровне объявлена переменная: на внутреннем (локально), т. е. внутри некоторого блока, или на внешнем (глобально), т. е. вне всех блоков.

Для того чтобы переменная была видна за пределами функции (являлась глобальной), в которой она объявлена, используют обращение с помощью ключевого слова **global**, за которым следует имя глобальной переменной.

12.2 Порядок выполнения работы

- 1 Создать функцию для решения задачи по вариантам.
- 2 Составить программу для ввода исходных данных в режиме диалога, организовать их обработку, вызвав созданную на шаге 1 функцию.
- 3 Решить задачи по вариантам с использованием вложенных функций пользователя; использовать внешнюю и встроенную (inline) функции.

Контрольные вопросы

- 1 Что называют подпрограммой?
- 2 Какие виды функций используют в MATLAB?
- 3 Каково определение формальных и фактических параметров?
- 4 Какова структура описания разных типов функций в MATLAB?
- 5 Как вызывают разные типы функции в основной программе?
- 6 Что называют локальными и глобальными переменными?

13 Обзор среды визуального моделирования Simulink

Цель работы: ознакомиться с составом среды визуального моделирования Simulink, библиотекой блоков, способами работы с ними, основными настройками среды; получить практические навыки работы с библиотекой блоков Simulink, размещения и связывания блоков в окне модели.

13.1 Общие сведения

Модуль Simulink является приложением к пакету MATLAB. При моделировании с использованием Simulink реализуется принцип визуального программирования, в соответствии с которым пользователь на экране из библиотеки стандартных блоков создает модель устройства и осуществляет расчеты. При этом, в отличие от классических способов моделирования, пользователю не нужно досконально изучать язык программирования и численные методы математики, а достаточно общих знаний, требующихся при работе на компьютере и знаний в той предметной области, в которой он работает.

При работе с Simulink пользователь имеет возможность модернизировать библиотечные блоки, создавать свои собственные, а также составлять новые библиотеки блоков. В ходе моделирования можно следить за процессами, происходящими в системе. Для этого используются специальные устройства наблюдения, входящие в состав библиотеки Simulink. Результаты моделирования могут быть представлены в виде графиков или таблиц.

Для запуска Simulink необходимо предварительно запустить пакет MATLAB. После открытия основного окна MATLAB нужно запустить программу Simulink. Это можно сделать одним из трех способов.

1 Нажать кнопку Simulink (Simulink Library) на панели инструментов командного окна MATLAB.

2 В командной строке главного окна MATLAB напечатать `simulink` и нажать клавишу `Enter` на клавиатуре.

3 Выполнить команду `Open ...` в меню `File` и открыть файл модели.

Последний вариант удобно использовать для запуска уже готовой и отлаженной модели, когда требуется лишь провести расчеты и не нужно добавлять новые блоки в модель. Использование первого и второго способов приводит к открытию окна обозревателя разделов библиотеки Simulink (рисунок 13.1).

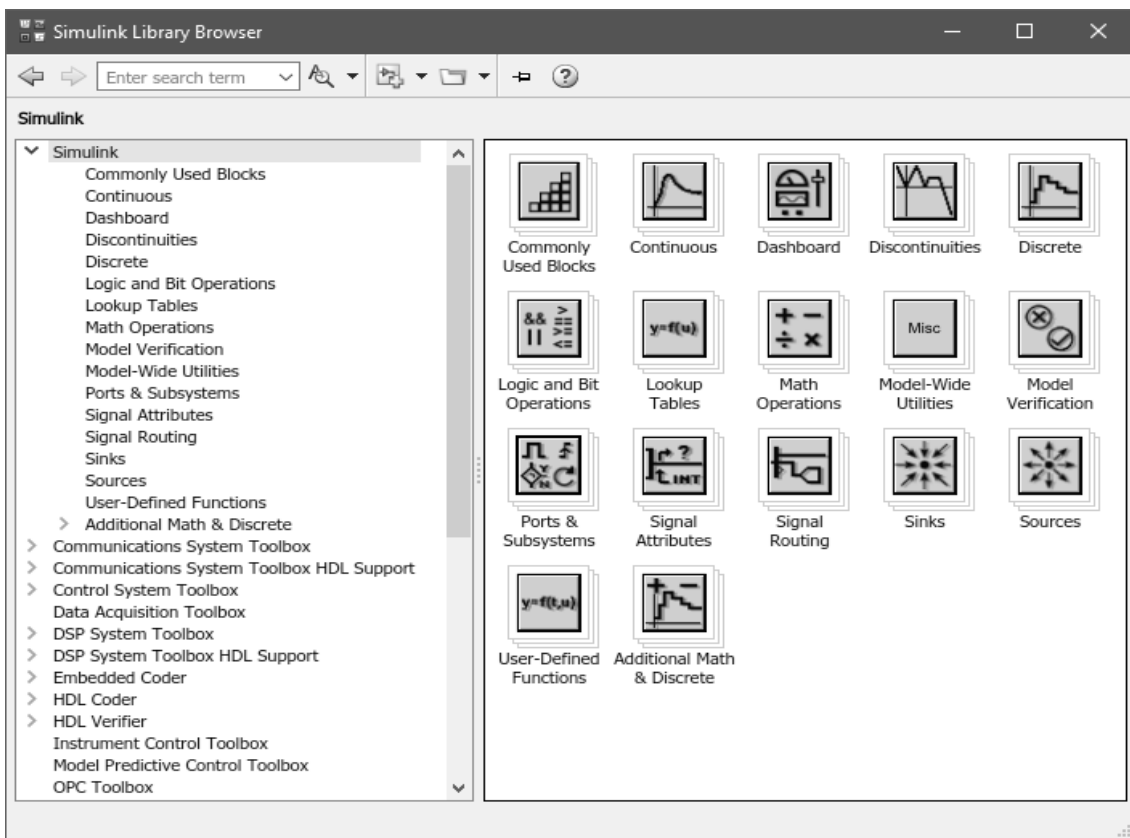


Рисунок 13.1 – Библиотека блоков Simulink

Окно обозревателя библиотеки блоков содержит следующие элементы:

- заголовок с названием окна – `Simulink Library Browser`;
- меню с командами `File`, `Edit`, `View`, `Help`;
- панель инструментов с ярлыками наиболее часто используемых команд;
- список разделов библиотеки, реализованный в виде дерева в левой части окна;
- окно содержимого выделенного раздела библиотеки (список вложенных разделов библиотеки или блоков);
- строку состояния, содержащую подсказку по выполняемому действию.

При выборе соответствующего раздела библиотеки в правой части окна отображается его содержимое (рисунок 13.2).

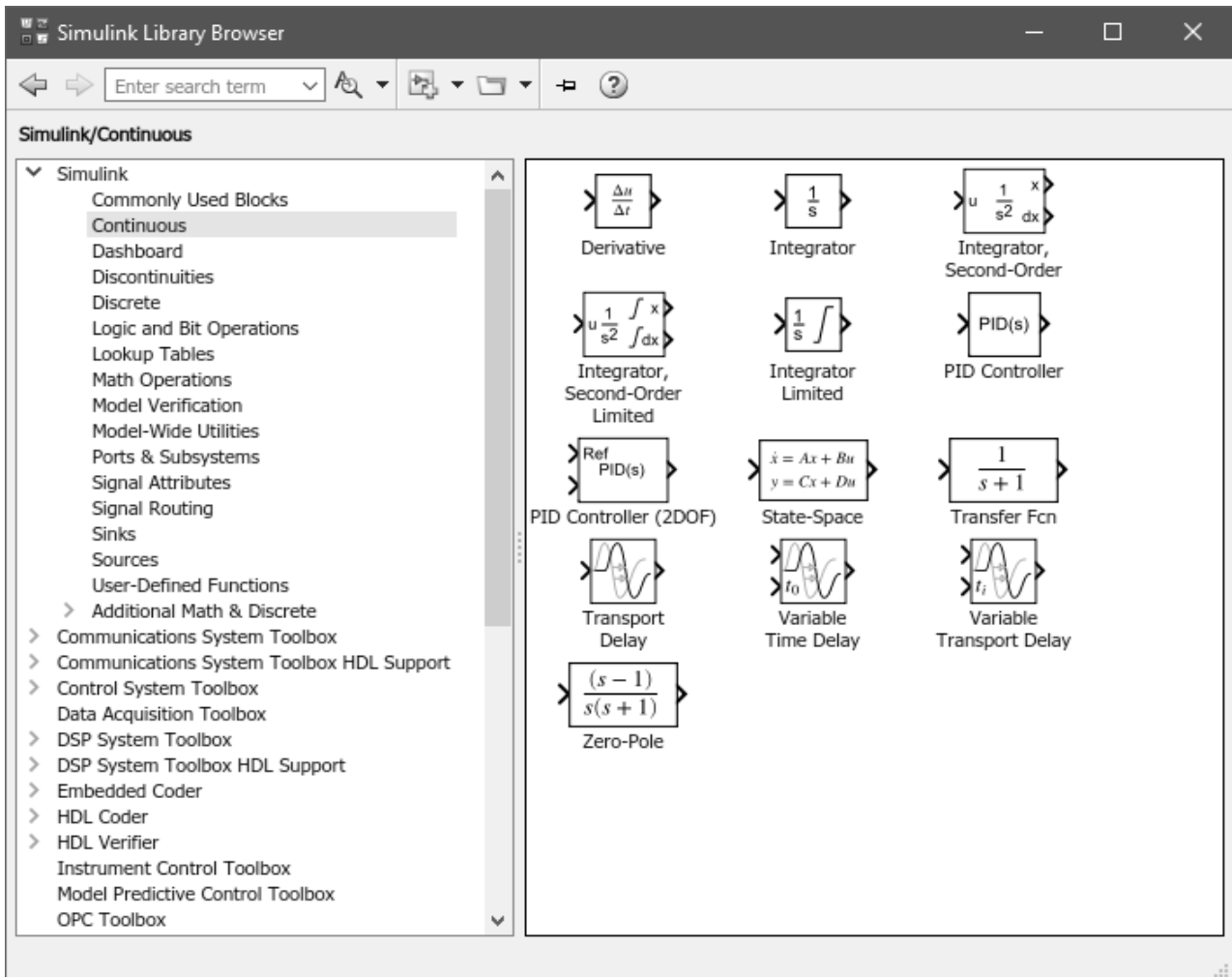


Рисунок 13.2 – Содержимое раздела Continuous библиотеки Simulink

Для создания модели в среде Simulink необходимо последовательно выполнить ряд действий.

1 Создать новый файл модели с помощью команды File/New/Model или используя соответствующую кнопку на панели инструментов. Окно новой модели представлено на рисунке 13.3.

2 Расположить требуемые блоки в окне модели. Для этого необходимо открыть соответствующий раздел библиотеки и, указав курсором на требуемый блок и нажав левую клавишу мыши, перетащить блок в окно модели. **Клавишу мыши нужно держать нажатой.** Для удаления блока следует выбрать его мышью и нажать клавишу Delete на клавиатуре.

Изменить размеры блока можно, выделив его и потянув мышью с нажатой левой клавишей за один из угловых маркеров. Курсор при этом превратится в двухстороннюю стрелку.

3 Далее, если это требуется, изменить параметры блоков, установленные программой по умолчанию. Для этого необходимо дважды щелкнуть левой клавишей мыши, указав курсором на изображение блока. Откроется окно редактирования параметров данного блока. При задании численных параметров следует иметь в виду, что в качестве десятичного разделителя должна использоваться точка, а не запятая. После внесения изменений нужно закрыть окно кнопкой ОК.

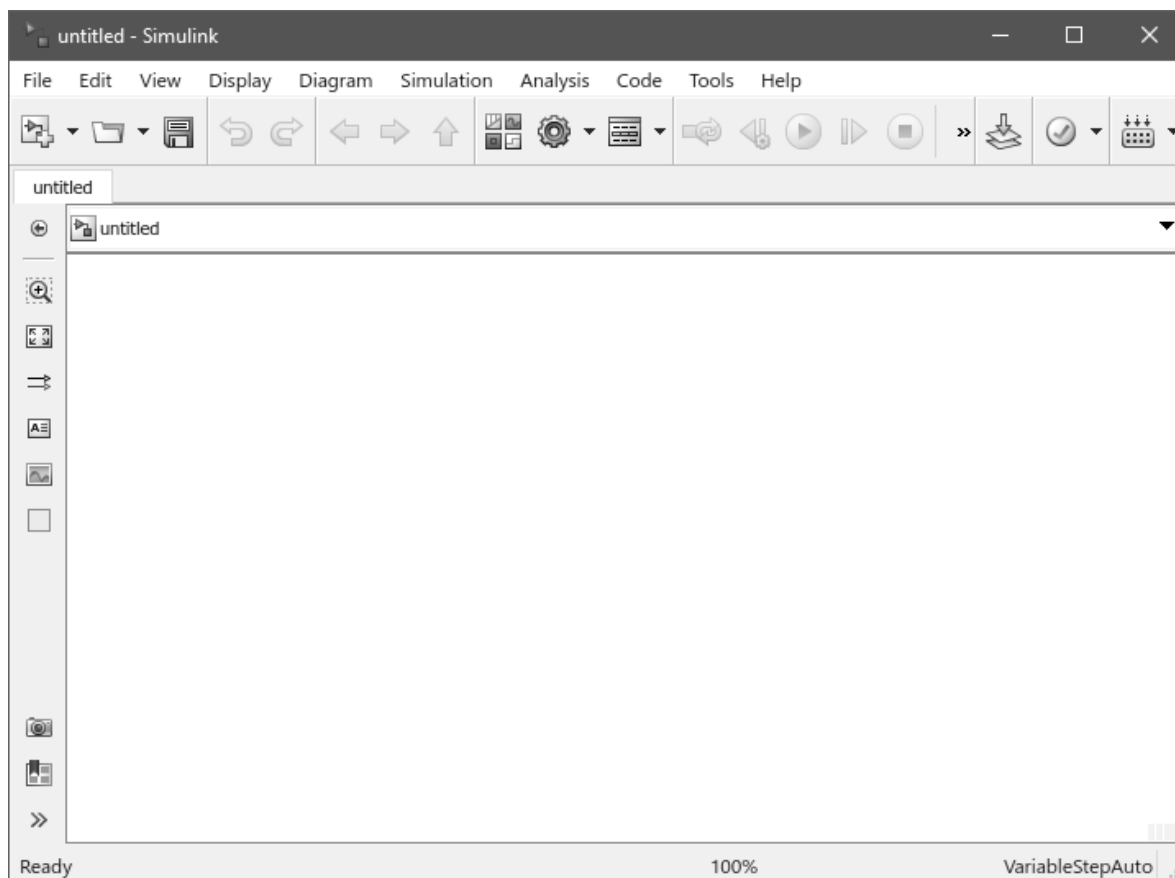


Рисунок 13.3 – Окно новой модели Simulink

4 После установки на схеме всех блоков из требуемых библиотек нужно выполнить соединение элементов схемы. Для соединения блоков необходимо указать курсором на выходной порт блока, а затем нажать и, не отпуская левую клавишу мыши, провести линию к входному порту другого блока.

5 После составления расчетной схемы необходимо сохранить ее в виде файла на диске, выбрав пункт меню File/Save As ... в окне схемы и указав папку и имя файла. По умолчанию файл модели сохраняется в текущей директории MATLAB. Следует иметь в виду, что имя файла не должно превышать 32 символов, должно начинаться с буквы и не может содержать символы кириллицы и спецсимволы. Это же требование относится и к пути файла модели (к тем папкам, в которых сохраняется файл).

13.2 Порядок выполнения работы

- 1 Запустить среду визуального моделирования Simulink.
- 2 Открыть библиотеку блоков и ознакомиться с ее структурой, панелями инструментов и меню.
- 3 Создать новую модель Simulink. Ознакомиться с элементами панелей управления, составом главного меню.
- 4 Получить навыки размещения блоков в окне модели, масштабирования, поворота, зеркального отражения. Отработать приемы построения соединительных линий, создания разветвлений, меток на линиях связи.

5 Ознакомиться с основными настройками Simulink: выбором метода решения, настройкой его параметров, запуска и остановки решения.

6 Оформить отчет.

Контрольные вопросы

1 Перечислите основные способы запуска среды Simulink.

2 Перечислите способы соединения блоков модели, построения ответвлений сигнальных линий.

3 Продемонстрируйте основные операции с блоками: поворот, зеркальное отражение, изменение размера, выравнивание, настройка формата отображения.

4 Укажите требования Simulink к имени файла модели и пути к рабочей директории.

14 Источники и приемники сигналов Simulink

Цель работы: получить практические навыки по использованию блоков из библиотек Sources (источники) и Sinks (приемники), настройке их параметров.

14.1 Общие сведения

Источники сигналов (библиотека Sources) служат для формирования сигналов, обеспечивающих управление работой Simulink-модели в целом или отдельных ее частей. Блоки-источники имеют по одному выходу и не имеют входов.

Приемники сигналов (Sinks) являются средством отображения результатов моделирования в удобной для восприятия форме. Блоки-приемники существенно отличаются по функциональному назначению. Их можно условно разделить на три вида:

1) блоки, используемые в качестве «смотровых окон», – Scope, XY Graph, Display;

2) блоки, обеспечивающие сохранение результатов моделирования, – To Workspace, To File;

3) блок управления моделированием Stop Simulation, который останавливает процесс моделирования при возникновении на его входе ненулевого сигнала.

14.2 Порядок выполнения работы

1 Создать новую модель Simulink.

2 Из библиотеки Sources поместить в рабочее окно модели блоки Constant, Clock, Sine Wave, Pulse Generator. Ознакомиться с набором и назначением параметров каждого блока.

3 Поместить в окно модели блоки Scope, Display, To Workspace из библиотеки Sinks. Ознакомиться с набором и назначением параметров каждого блока. Связать выход каждого блока-источника со входом соответствующего приемника.

4 Сохранить модель и запустить ее на выполнение. Просмотреть результа-

ты моделирования в каждом блоке-приемнике. При необходимости изменить параметры блоков.

5 Повторить пп. 2–4 для других блоков из библиотек Sources и Sinks. Пример блок-схемы модели приведен на рисунке 14.1.

6 Проанализировать полученные результаты, оформить отчет.

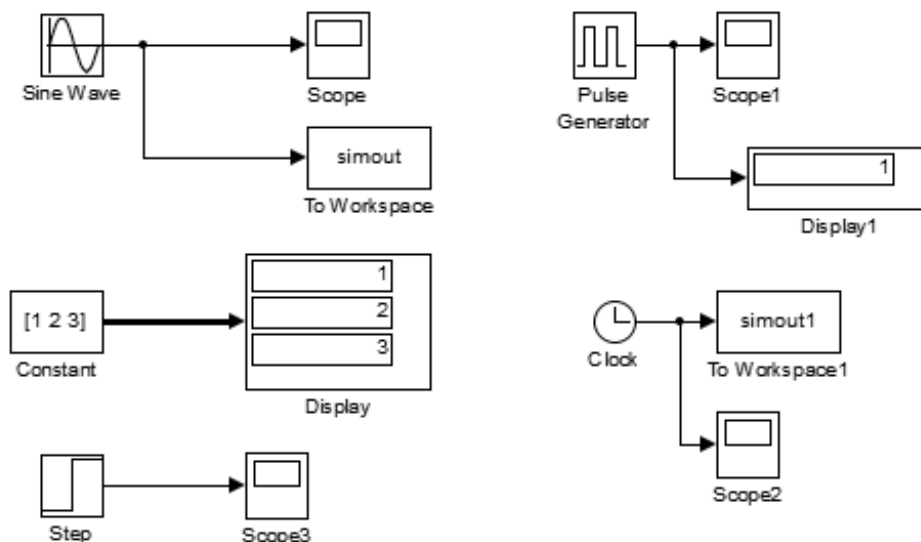


Рисунок 14.1 – Пример выполнения работы

Контрольные вопросы

- 1 Назначение и параметры блоков Clock, Digital Clock.
- 2 Назначение и параметры блоков Constant, Ramp.
- 3 Назначение и параметры блоков Ground, Pulse Generator.
- 4 Назначение и параметры блоков Random Number, Uniform Random Number.
- 5 Назначение и параметры блоков Step, Sine Wave.
- 6 Назначение и принципы работы с блоком Signal Generator.
- 7 Назначение и параметры блоков Display, Scope.
- 8 Назначение и параметры блоков Stop Simulation, XY Graph.

15 Математические операции с сигналами в Simulink

Цель работы: получить практические навыки по использованию блоков из библиотеки Math Operations (математические операции), настройке их параметров.

15.1 Общие сведения

Блоки из библиотеки Math Operations реализуют элементарные (алгебраические и тригонометрические) функции над входными сигналами и могут быть использованы в любом месте модели.

15.2 Порядок выполнения работы

Даны действительные числа x, y, z . Составить блок-схему Simulink для вычисления значений выражений согласно варианту задания:

$$1) a = \frac{\sqrt{|x-1|} - \sqrt[3]{|y|}}{1 + x^2/2 + y^2/4}, \quad b = x \left(\operatorname{arctg} z + e^{-(x+3)} \right);$$

$$2) a = \frac{3 + e^{y-1}}{1 + x^2 |y - \operatorname{tg} z|}, \quad b = 1 + |y - x| + \frac{(y-x)^2}{2} + \frac{|y-x|^3}{3};$$

$$3) a = (1+y) \frac{x+y/(x^2+4)}{e^{-(x+2)} + 1/(x^2+4)}, \quad b = 1 + \operatorname{tg}^2 \frac{z}{2};$$

$$4) a = y + \frac{x}{y^2 + \left| \frac{x^2}{y + x^3/3} \right|}, \quad b = \frac{1 + \cos(y-2)}{x^4/2 + \sin^2 z};$$

$$5) a = \frac{2 \cos(x - \pi/6)}{1/2 + \sin^2 y}, \quad b = 1 + \frac{z^2}{3 + 0,2z^2};$$

$$6) a = \frac{1 + \sin^2(x+y)}{2 + \left| x - 2x/(1+x^2y^2) \right|}, \quad b = \cos^2 \left(\operatorname{arctg} \frac{1}{z} \right) + x;$$

$$7) a = \ln \left| \left(y - \sqrt{|x|} \right) \left(x - \frac{y}{z + x^2/4} \right) \right|, \quad b = x - \frac{x^2}{3y} + \frac{x^5}{5y^2};$$

$$8) a = \frac{\sqrt{2-x} - \sqrt{y+6}}{x^2 + 2x + 6y} \ln \left| \sqrt{x^2 + 1} \right|, \quad b = \cos(4x^2 - y^2) + 4z^2 + 16 \ln x^2 y;$$

$$9) a = e^{-x^2} + \sqrt{\left| \sin \left(y^3 + \frac{\pi}{3} z \right) \right|}, \quad b = \frac{(2x+1)(\ln|x+3z| - \ln y)}{\sqrt{x^2 - 4yz}};$$

$$10) a = \operatorname{tg} \frac{\sqrt{1+3x^2z-1}}{x^2 + y^2 + z^2} - \sqrt{x^2 + y^2}, \quad b = \cos^2 \frac{\ln(1 - \cos(x^2))}{e^{x+2y}};$$

$$11) a = 8^{\frac{3}{x-2}} + \operatorname{lg} \frac{\sqrt{1 - \cos z}}{y}, \quad b = \frac{xyz}{\sqrt{|x| + \sqrt{|y| + \sqrt{|z|}}}};$$

$$12) a = \operatorname{lg} \frac{10x^3 + 3y - 1}{z^2 - 4x + 2y^2}, \quad b = \sin \left(\frac{x+1}{y-2} \right)^z + \sqrt{x + \sqrt{yz}}.$$

1 Создать новую модель Simulink. Поместить необходимые блоки-источники для задания входных сигналов.

2 Из библиотеки Math Operations поместить в окно модели требуемые блоки для выполнения математических операций согласно заданию.

3 Соединить размещенные блоки в необходимом порядке. При необходимости дублирования входного сигнала следует использовать ответвление сигнальной линии, а не дублировать блок-источник.

4 Вывести результат работы модели в подходящий блок-приемник. Сохранить модель и запустить ее на выполнение. Просмотреть консольное окно MATLAB на наличие предупреждений, в случае необходимости скорректировать параметры решателя.

5 Проанализировать полученные результаты, оформить отчет.

Контрольные вопросы

1 Блоки для сложения и вычитания сигналов. Особенности применения, настройка параметров.

2 Блоки умножения и деления сигналов. Особенности применения, настройка параметров.

3 Блоки масштабирования и округления значения сигналов. Особенности применения, настройка параметров.

4 Блоки для вычисления математических функций от сигналов. Особенности применения, настройка параметров.

16 Операции отношения и логические операторы в Simulink

Цель работы: получить практические навыки по использованию блоков из библиотеки Logic And Bit Operations (логические и битовые операции), настройке их параметров.

16.1 Порядок выполнения работы

Составить блок-схему Simulink для вычисления значений функции по заданному действительному значению x согласно приведенным вариантам:

$$\begin{array}{ll}
 1) f(x) = \begin{cases} x^2 + 2, & \text{если } x \leq 1; \\ 2x, & \text{если } 1 < x \leq 3; \\ x - 3, & \text{если } x > 3; \end{cases} & 3) f(x) = \begin{cases} 1/x, & \text{если } x < 0; \\ x, & \text{если } 0 \leq x < 2; \\ 5, & \text{если } x \geq 2; \end{cases} \\
 2) f(x) = \begin{cases} 2x^2, & \text{если } x \leq 0; \\ \sin x, & \text{если } 0 < x \leq \pi/2; \\ x, & \text{если } x > \pi/2; \end{cases} & 4) f(x) = \begin{cases} \operatorname{tg} x, & \text{если } x < 0; \\ 2 - x, & \text{если } 1 \leq x < 2; \\ x, & \text{если } x \geq 2; \end{cases}
 \end{array}$$

$$5) f(x) = \begin{cases} \cos x^2, & \text{если } x < 2; \\ 3 - x, & \text{если } 2 \leq x < 5; \\ x - 3, & \text{если } x \geq 5; \end{cases} \quad 9) f(x) = \begin{cases} \cos^2 x, & \text{если } x \leq 2; \\ 0, & \text{если } 2 < x < 4; \\ -x, & \text{если } x \geq 4; \end{cases}$$

$$6) f(x) = \begin{cases} x, & \text{если } x \leq 0; \\ 1 - x, & \text{если } 0 < x \leq 1; \\ 1/3, & \text{если } x > 1; \end{cases} \quad 10) f(x) = \begin{cases} x + 4, & \text{если } x \leq -2; \\ x^2 + 2, & \text{если } -2 < x < 4; \\ -2x, & \text{если } x \geq 4; \end{cases}$$

$$7) f(x) = \begin{cases} x^2 + 2, & \text{если } x < 0; \\ -x^2 + 1, & \text{если } 0 \leq x < 3; \\ 1, & \text{если } x \geq 3; \end{cases} \quad 11) f(x) = \begin{cases} 2x - 1, & \text{если } x \leq -2; \\ |\sin 2x|, & \text{если } -2 < x < 2; \\ 0, & \text{если } x \geq 2; \end{cases}$$

$$8) f(x) = \begin{cases} \sqrt{x^2 + 2}, & \text{если } x < -3; \\ 2 - x^2, & \text{если } -3 \leq x < 3; \\ 5 + x, & \text{если } x \geq 3; \end{cases} \quad 12) f(x) = \begin{cases} \sqrt{|x| + 2}, & \text{если } x \leq -1; \\ x^3 + \pi, & \text{если } -1 < x \leq 2,5; \\ \sqrt{2}, & \text{если } x > 2,5. \end{cases}$$

Контрольные вопросы

- 1 Назначение и параметры блоков Compare To Constant, Compare To Zero.
- 2 Назначение и параметры блоков Logical Operator, Relational Operator.
- 3 Особенности использования блоков If, If Action Subsystem.
- 4 Постройте таблицу истинности оператора AND (И).
- 5 Постройте таблицу истинности оператора OR (ИЛИ).
- 6 Постройте таблицу истинности оператора NAND (НЕ-И).

17 Блоки маршрутизации сигналов в Simulink

Цель работы: получить практические навыки по использованию блоков из библиотеки Signal Routing (маршрутизация сигналов), настройке их параметров.

17.1 Порядок выполнения работы

- 1 Создать новую модель Simulink.
- 2 Поместить в окно модели несколько блоков-источников из библиотеки Sources.
- 3 Поместить в окно модели блоки Bus Creator, Bus Selector, Mux, Demux. Изучить их параметры и настройки. Связать блоки-источники с блоками маршрутизации по своему усмотрению.

4 Поместить необходимое количество блоков-приемников из библиотеки Sinks. Создать линии связи (рисунок 17.1).

5 Запустить модель на выполнение. Просмотреть результаты.

6 Ознакомиться с остальными блоками библиотеки Signal Routing.

7 Оформить отчет.

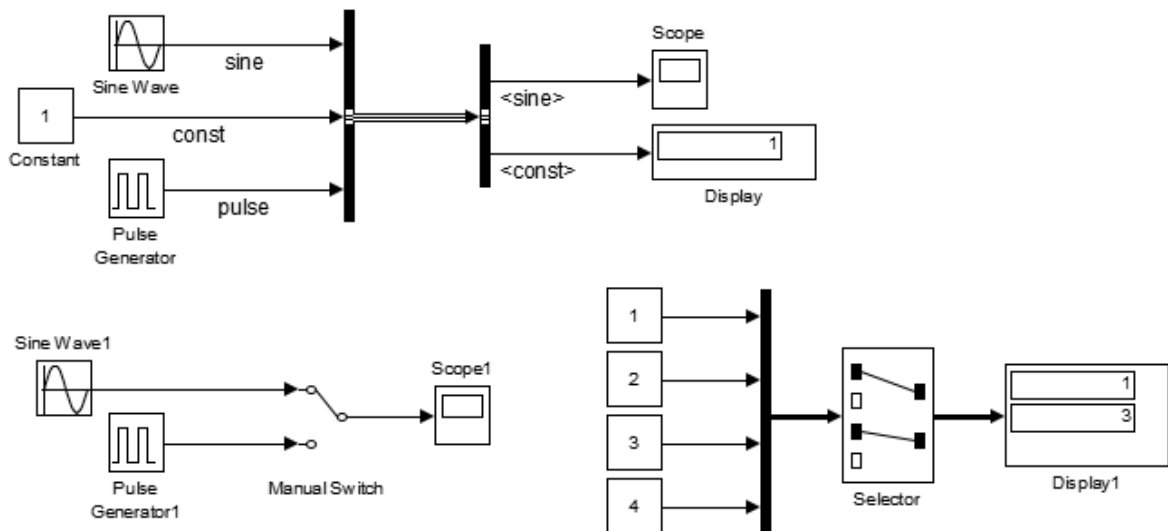


Рисунок 17.1 – Пример выполнения работы

Контрольные вопросы

- 1 Назначение и параметры блоков Mux и Demux.
- 2 Назначение и параметры блоков Bus Creator и Bus Selector.
- 3 Назначение и параметры блоков Manual Switch и Switch.
- 4 Назначение и параметры блока Multiport Switch.
- 5 Назначение и параметры блоков Goto и From.

18 Решение алгебраических уравнений и систем в Simulink

Цель работы: освоить методику решения алгебраических уравнений с помощью Simulink.

18.1 Общие сведения

Для решения алгебраических уравнений вида $F(x) = 0$ в Simulink используется блок Algebraic Constraint из раздела Math Operations библиотеки блоков.

Блок Algebraic Constraint находит такое значение выходного сигнала, при котором значение входного сигнала становится равным нулю. При этом входной сигнал должен быть прямо или опосредованно связан с выходным сигналом.

Блок имеет один параметр Initial Guess (начальное значение), которое может быть скалярным или векторным, если система уравнений имеет несколько

решений и необходимо найти их все (рисунок 18.1).

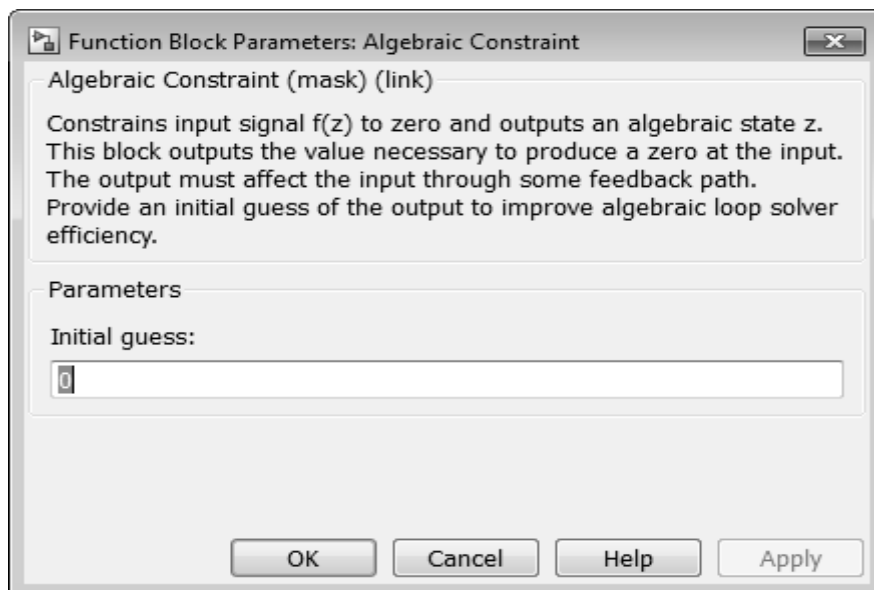


Рисунок 18.1 – Окно параметров блока Algebraic Constraint

18.2 Порядок выполнения работы

1 Создать новую модель Simulink и поместить в нее два блока Algebraic Constraint.

2 Добавить другие необходимые блоки, задающие входные сигналы, связанные с выходными.

3 Для отображения результатов решения системы уравнений использовать блоки Display.

4 Настроить параметры решателя, задать начальные значения для неизвестных и запустить модель на выполнение.

5 Выполнить проверку правильности решения вручную или с использованием MATLAB. Оформить отчет.

Решить следующие системы алгебраических уравнений:

$$1) \begin{cases} (x + 0,2)^2 + (y + 0,3)^2 = 1; \\ x + y = 0,9; \end{cases}$$

$$2) \begin{cases} y^2 - xy = -12; \\ x^2 - xy = 28; \end{cases}$$

$$3) \begin{cases} x^2y + xy^2 = 6; \\ xy + x + y = 5; \end{cases}$$

$$4) \begin{cases} x^3 + y^3 = 7; \\ x^3y^3 = -8; \end{cases}$$

$$5) \begin{cases} x + y + \frac{x}{y} = 9; \\ \frac{(x + y)x}{y} = 20; \end{cases}$$

$$6) \begin{cases} x^3 + y^3 = 65; \\ x^2y + xy^2 = 20; \end{cases}$$

$$7) \begin{cases} \frac{x^2 + y^2}{x + y} = \frac{10}{3}; \\ \frac{1}{x} + \frac{1}{y} = \frac{3}{4}; \end{cases}$$

$$9) \begin{cases} \frac{x + y}{x - y} + \frac{x - y}{x + y} = \frac{13}{6}; \\ xy = 5; \end{cases}$$

$$8) \begin{cases} x^3 + y^3 = 7; \\ xy(x + y) = -2; \end{cases}$$

$$10) \begin{cases} \sqrt{\frac{y}{x}} - 2\sqrt{\frac{x}{y}} = 1; \\ \sqrt{5x + y} + \sqrt{5x - y} = 4. \end{cases}$$

В качестве примера найдем решение системы алгебраических уравнений

$$\begin{cases} x^2 + y^2 = 6; \\ x + y = 2. \end{cases}$$

Для решения задачи используем два блока Algebraic Constraint, сигналы на входах которых вычисляются по заданным уравнениям на основании выходных сигналов. Блок-схема решения приведена на рисунке 18.2.

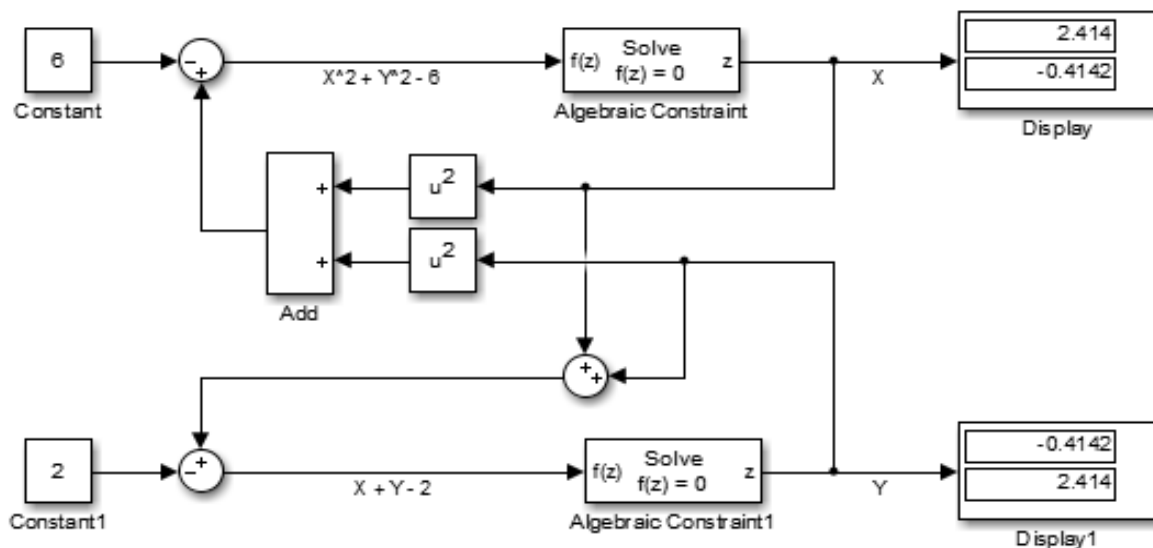


Рисунок 18.2 – Схема решения системы алгебраических уравнений

Поскольку данная система уравнений имеет два решения, то начальные значения блоков Algebraic Constraint заданы в виде векторов. Для первого (верхнего) блока начальное значение задано вектором $[1, -1]$, а для второго (нижнего) блока – вектором $[-1, 1]$.

Контрольные вопросы

- 1 Назначение и принцип работы блока Algebraic Constraint.
- 2 Настройка параметров блока Algebraic Constraint для решения алгебраических уравнений и систем.

19 Решение обыкновенных дифференциальных уравнений и их систем в Simulink

Цель работы: ознакомиться с методикой и получить практические навыки решения обыкновенных дифференциальных уравнений в среде Simulink.

19.1 Общие сведения

Для решения ОДУ и их систем в Simulink используется блок Integrator, который выполняет интегрирование входного сигнала по времени с заданными начальными условиями.

Количество применяемых блоков Integrator зависит от порядка дифференциального уравнения.

Рассмотрим одномассовую колебательную систему, динамическая модель которой приведена на рисунке 19.1.

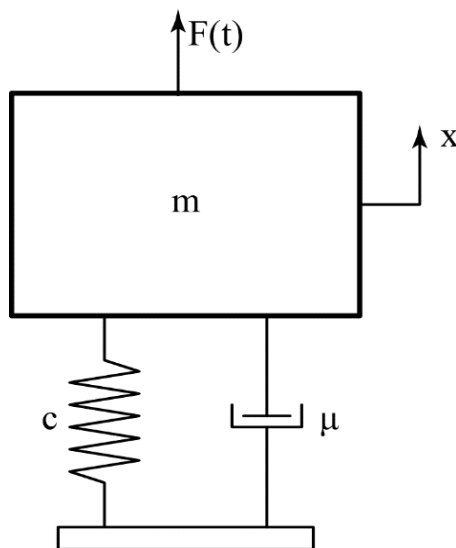


Рисунок 19.1 – Динамическая модель одномассовой колебательной системы

Как известно, движение груза описывается следующим дифференциальным уравнением второго порядка:

$$\frac{d^2x}{dt^2} = \frac{1}{m} \left(F(t) - cx - \mu \frac{dx}{dt} \right),$$

где m – масса тела;

$F(t)$ – внешнее силовое воздействие;

c – коэффициент жесткости пружины;

μ – коэффициент сопротивления диссипативного элемента.

Блок-схема решения ОДУ приведена на рисунке 19.2. При моделировании были приняты следующие параметры: масса груза – 10 кг; коэффициент жесткости пружины – 1000 Н/м; коэффициент сопротивления диссипативного элемен-

та – 25 Н·с/м. Внешним воздействием являлся единичный ступенчатый сигнал.

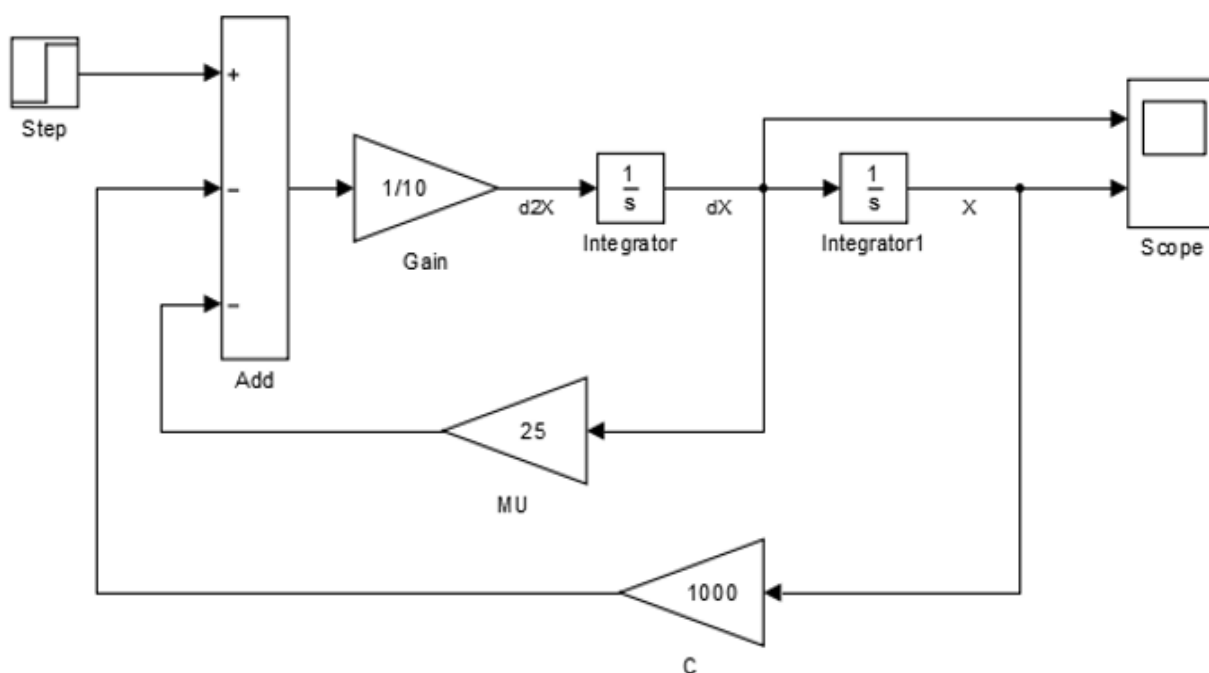


Рисунок 19.2 – Блок-схема решения обыкновенного ДУ движения груза

19.2 Порядок выполнения работы

1 Составить уравнения движения двухмассовой колебательной системы автомобиля, приведенной на рисунке 19.3, при ступенчатом единичном воздействии $q(t)$. Параметры модели взять из таблицы 19.1.

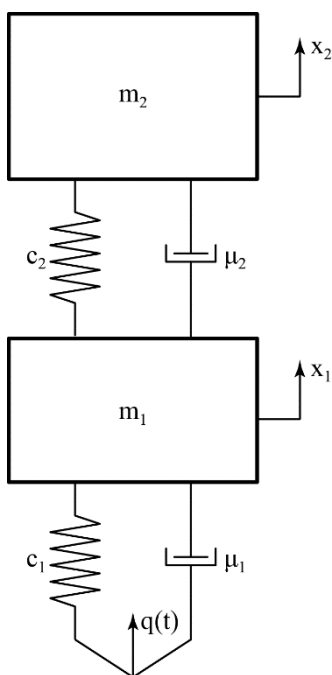


Рисунок 19.3 – Динамическая модель двухмассовой колебательной системы

Таблица 19.1 – Параметры двухмассовой колебательной системы

Вариант	1	2	3	4	5	6	7	8	9	10	11	12
m_1 , кг	65	80	83	80	100	140	110	150	155	164	150	155
m_2 , кг	615	740	775	670	870	950	920	1120	1685	1925	1800	2060

2 Определить коэффициенты жесткости упругих элементов из условия обеспечения собственной парциальной частоты колебаний в диапазоне 6,28...12,56 рад/с.

3 Определить параметры диссипативных элементов с учетом, что коэффициенты относительного затухания $\gamma_1 = 0,1$, $\gamma_2 = 0,3...0,4$.

4 Создать новый файл модели Simulink.

5 Поместить в окно модели блоки Step, Sum, Gain, Integrator, Scope.

6 Соединить блоки необходимым образом для решения составленной системы ОДУ, настройте параметры блоков. В блоки-приемники завести сигналы перемещений и скоростей сосредоточенных масс m_1 и m_2 , а также усилий в упругих элементах c_1 и c_2 .

7 Выбрать конечное время моделирования с расчетом, чтобы в системе установилось равновесное состояние. Проанализировать полученные графики переходных процессов. Оформить отчет.

Контрольные вопросы

1 Порядок решения ОДУ с помощью Simulink.

2 Порядок настройки блоков Integrator для решения систем ОДУ с помощью Simulink. Способы задания начальных условий.

Список литературы

1 **Черняк, А. А.** Математические расчеты в среде MathCAD: учебное пособие для вузов / А. А. Черняк, Ж. А. Черняк. – 3-е изд., испр. и доп. – Москва: ЮРАЙТ, 2021. – 163 с.

2 **Сызранцева, К. В.** Основы инженерных расчётов в математическом процессоре MathCAD с примерами применения / К. В. Сызранцева, Л. Н. Бакановская. – Тюмень: Тюмен. индустриал. ун-т, 2021. – 94 с.

3 **Фомин, В. Г.** Математическое моделирование в системе MATHCAD / В. Г. Фомин. – Саратов: Саратов. гос. техн. ун-т им. Ю. А. Гагарина, 2020. – 80 с.

4 **Рябикова, Т. В.** Решение задач математического анализа с использованием Matlab: учебно-методическое пособие / Т. В. Рябикова, Л. Ю. Уразаева. – Москва: ФЛИНТА, 2021. – 108 с.

5 **Герман-Галкин, С. Г.** Matlab & Simulink. Проектирование мехатронных систем на ПК / С. Г. Герман-Галкин. – Санкт-Петербург: КОРОНА-Век, 2008. – 368 с.

6 **Гулятьев, А.** Визуальное моделирование в среде MATLAB: учебный курс / А. Гулятьев. – Санкт-Петербург: Питер, 2002. – 432 с.

7 **Дьяконов, В. П.** MATLAB 6.5 SP1/ 7/ 7SP1/ 7SP2+Simulink 5/6. Инструменты искусственного интеллекта и биоинформатики / В. П. Дьяконов, В. В. Круглов. – Москва: СОЛОН-ПРЕСС, 2006. – 456 с.

8 **Кетков, Ю. Л.** MATLAB 7. Программирование, численные методы / Ю. Л. Кетков. – Санкт-Петербург: БХВ-Петербург, 2005. – 734 с.