Федеральное агентство связи

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Сибирский государственный университет телекоммуникаций и

информатики» (СибГУТИ)

Отчет по **Лабораторной работе №4**

по дисциплине «Теория массового обслуживания»

Тема: «Введение в Mathcad. Переменные, функции, графика матричные операции, программирование функций. Марковские цепи. Определение и построение»

**Вариант 10**

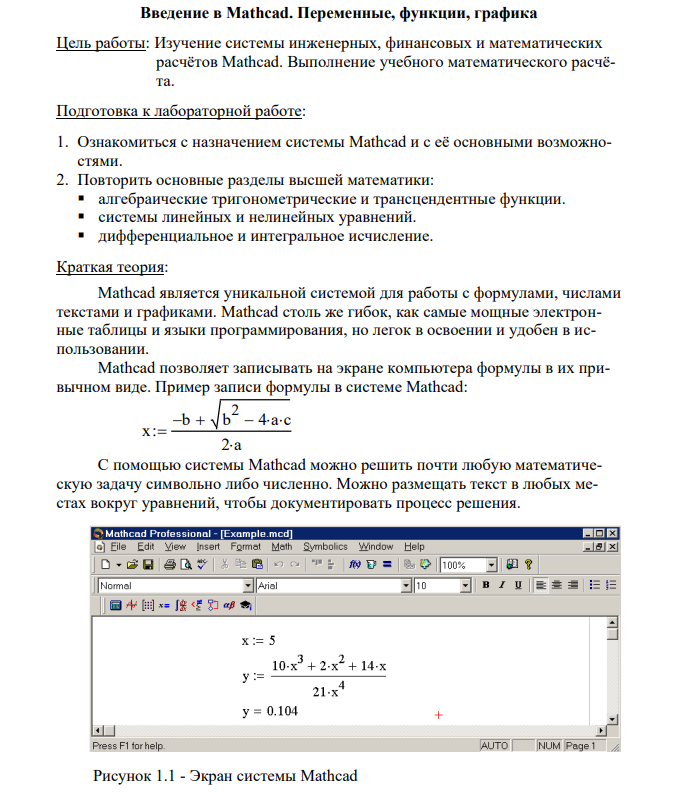
Выполнил:

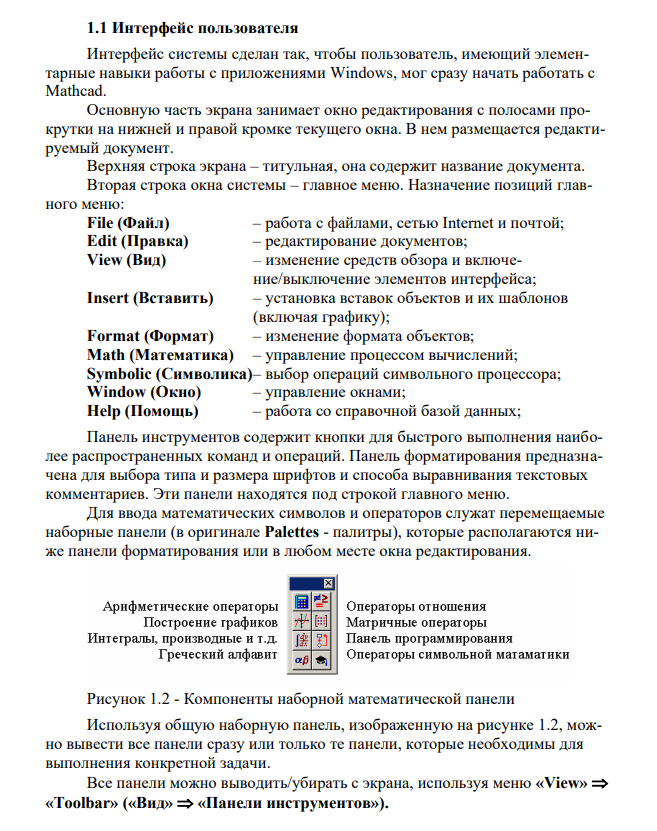
студент гр. ИА-232

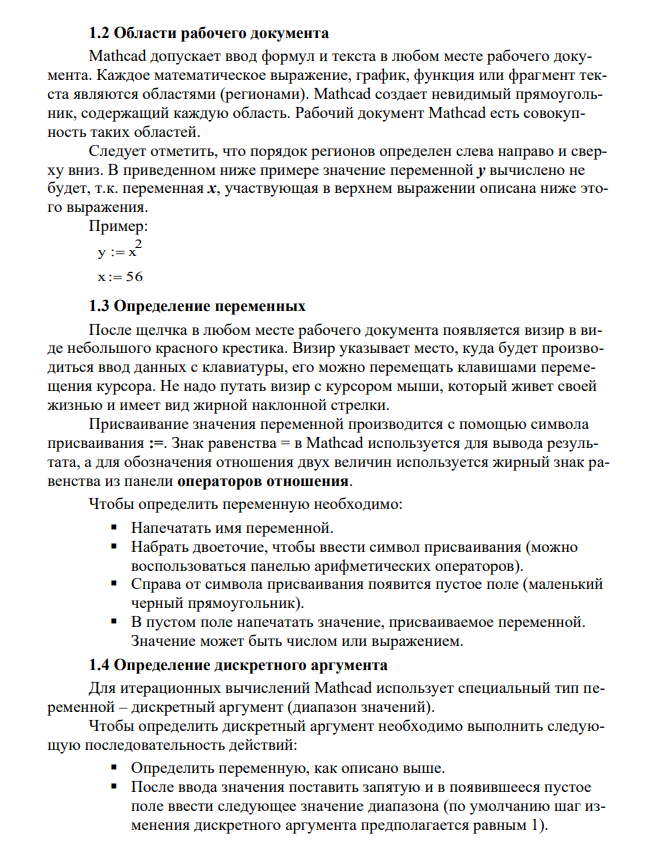
Македон Никита Игоревич

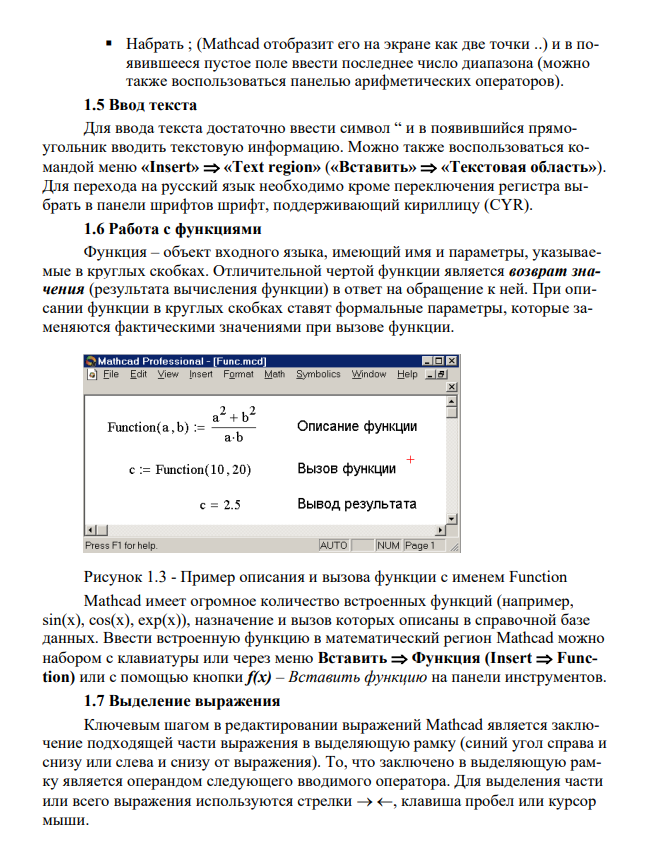
Новосибирск 2024

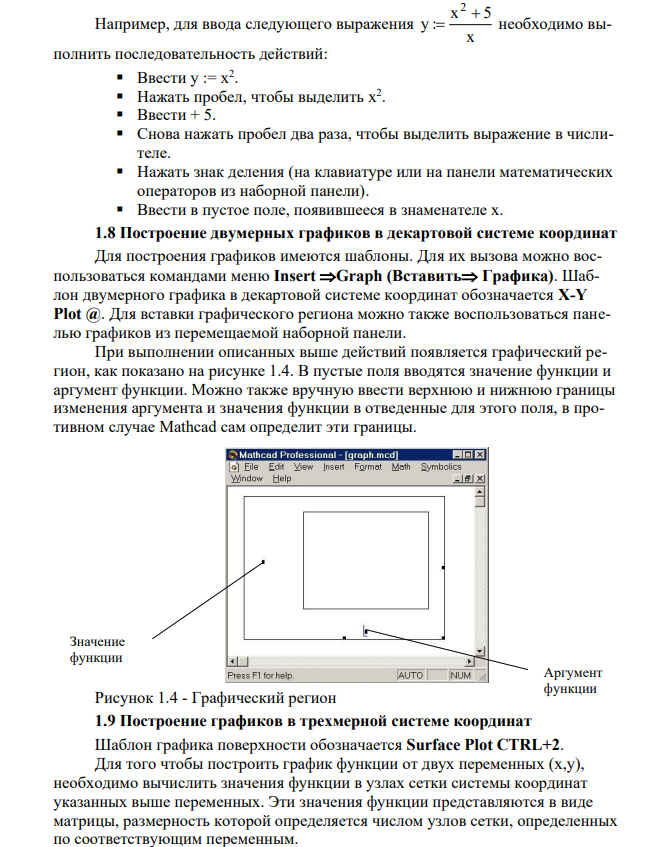
**Часть №1**

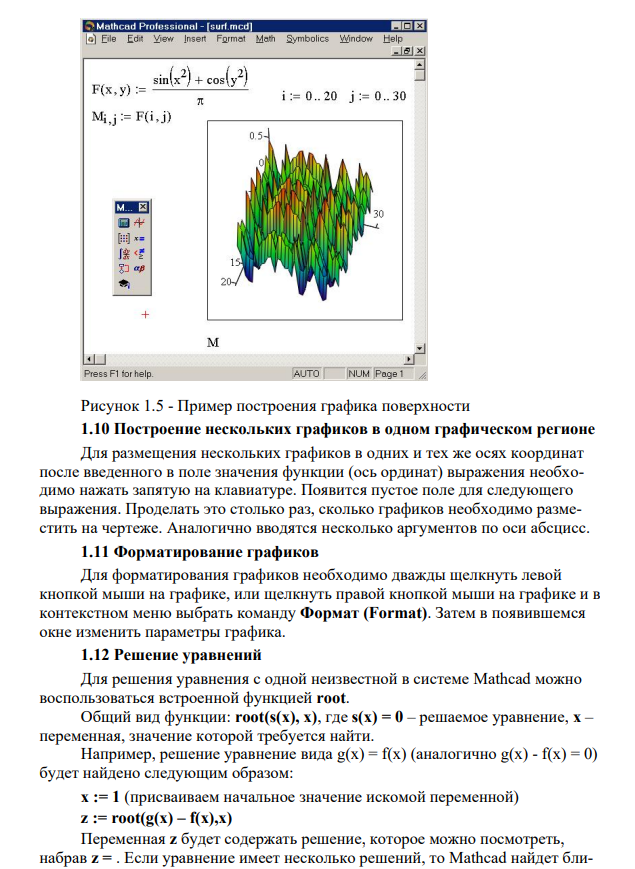
****

****

****

****

****

****

****

**2. Код программы**

**syms x;**

**f = (sin(x^2)^4)/(log(x^5));**

**df = diff(f, x);**

**% Векторизация функции f**

**f\_vec = matlabFunction(f, 'vars', {x});**

**% Интегрирование с численной аппроксимацией**

**F\_numeric = @(x) arrayfun(@(z) integral(@(t) f\_vec(t), 1.1, z), x);**

**x\_vals = linspace(1.1, 10, 1000);**

**figure;**

**subplot(3, 1, 1);**

**fplot(f\_vec, [1.1, 10]);**

**title('Функция f(x)');**

**xlabel('x');**

**ylabel('f(x)');**

**grid on;**

**subplot(3, 1, 2);**

**fplot(matlabFunction(df), [1.1, 10]);**

**title('Производная f(x)');**

**xlabel('x');**

**ylabel('df(x)/dx');**

**grid on;**

**subplot(3, 1, 3);**

**plot(x\_vals, F\_numeric(x\_vals));**

**title('Интеграл f(x)');**

**xlabel('x');**

**ylabel('∫f(x)dx');**

**grid on;**

**hold on;**

**y\_vals = F\_numeric(x\_vals);**

**fill([x\_vals fliplr(x\_vals)], [zeros(1, length(x\_vals)) fliplr(y\_vals)], 'b', 'FaceAlpha', 0.3);**

**hold off;**

**a = 1;**

**b = -0.5;**

**g = a \* x + b;**

**g\_vec = matlabFunction(g, 'vars', {x});**

**figure;**

**fplot(f\_vec, [1.1, 10], 'DisplayName', 'f(x)');**

**hold on;**

**fplot(g\_vec, [1.1, 10], 'DisplayName', 'a\*x + b');**

**legend('show');**

**title('Графическое решение уравнения a\*x + b = f(x)');**

**xlabel('x');**

**ylabel('y');**

**grid on;**

**f\_anon = @(x) f\_vec(x) - (a \* x + b);**

**initial\_guess = 1.1;**

**x\_solution = fsolve(f\_anon, initial\_guess);**

**disp('Аналитическое решение уравнения:');**

**disp(x\_solution);**

**syms y;**

**F = 2\*(x^4) + 3\*(y^4);**

**F\_handle = matlabFunction(F, 'Vars', [x, y]);**

**x\_vals = linspace(-10, 10, 100);**

**y\_vals = linspace(-10, 10, 100);**

**[X, Y] = meshgrid(x\_vals, y\_vals);**

**Z = F\_handle(X, Y);**

**% Избегаем комплексных значений**

**Z(imag(Z) ~= 0) = NaN;**

**figure;**

**surf(X, Y, Z);**

**title('График функции F(x, y)');**

**xlabel('x');**

**ylabel('y');**

**zlabel('F(x, y)');**

**grid on;**

**dim1 = [0.2 0.8 0.1 0.1];**

**annotation('textbox', dim1, 'String', 'График функции F(x, y)', 'FitBoxToText', 'on');**

**dim2 = [0.2 0.7 0.1 0.1];**

**annotation('textbox', dim2, 'String', 'График построен в виде поверхности', 'FitBoxToText', 'on');**

**3. Результат выполнения программы и графики**

****

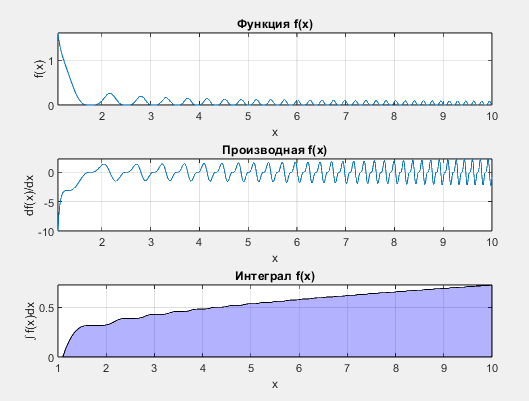
****

График №1. Функция, производная, интеграл

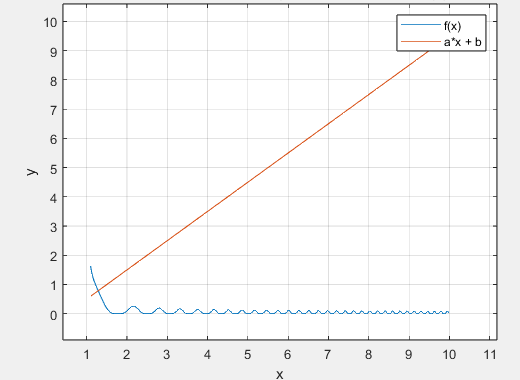


График №2. Графическое решение уравнения a\*x + b = f(x)

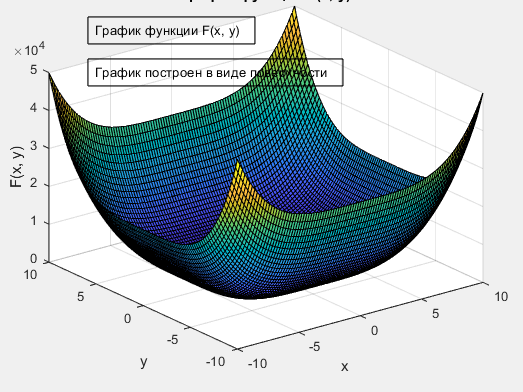


График №3. График функции F(x, y)

4. Контрольные вопросы

1. **Назначение пунктов меню и кнопок панелей инструментов системы Mathcad**

Главное меню:

1. Файл:

• Создание, открытие и сохранение документов.

• Импорт и экспорт данных в различных форматах.

• Печать документа.

2. Правка:

• Операции с текстом: вырезать, копировать, вставить.

• Отмена и повтор действий.

3. Просмотр:

• Настройка отображения документа.

• Масштабирование и навигация по страницам.

4. Вставка:

• Добавление математических выражений, графиков, таблиц и изображений.

• Вставка текстовых блоков и аннотаций.

5. Формулы:

• Доступ к встроенным математическим функциям и константам.

• Настройка параметров вычислений.

6. Справка:

• Доступ к документации и руководствам по использованию Mathcad.

Панели инструментов:

1. Основная панель:

• Кнопки для быстрого доступа к основным функциям: создание нового документа, открытие существующего, сохранение и печать.

2. Математическая панель:

• Кнопки для вставки различных математических операций, функций и символов.

3. Графическая панель:

• Инструменты для построения графиков и диаграмм.

4. Форматирование текста:

• Настройки шрифтов, стилей и размеров текста.

5. Инструменты анализа:

• Функции для выполнения статистического анализа, численного интегрирования и решения уравнений.

1. **Переменная скалярного типа в математическом регионе.**

В Mathcad переменная скалярного типа — это переменная, которая хранит одно числовое значение. Скалярные переменные могут быть использованы для выполнения различных математических операций, таких как сложение, вычитание, умножение и деление.

1. **Переменная типа дискретный аргумент в математическом регионе.**

В Mathcad переменная типа дискретный аргумент (или дискретная переменная) используется для представления конечного набора значений, обычно в контексте последовательностей, массивов или функций, зависящих от целочисленных значений. Дискретные переменные часто применяются в задачах, связанных с анализом данных, интерполяцией и другими вычислениями.

1. **Способы описания и вызова функции в системе Mathcad.**

1. Определение функции

1.1. Явное определение функции

1.2. Определение многоаргументной функции

Для функций с несколькими аргументами используйте запятые

2. Вызов функции

Чтобы вызвать функцию, просто укажите ее имя и передайте аргументы:

3. Векторные и матричные функции

Функции могут принимать векторы или матрицы в качестве аргументов. Mathcad автоматически применяет функцию к каждому элементу:

4. Анонимные функции

Mathcad также поддерживает анонимные функции, которые можно определять "на лету":

5. Использование функций в выражениях

6. Графическое представление функций

Mathcad позволяет строить графики для визуализации функций. Вы можете создать график функции, используя команду "Plot":

1. **Способы построения графиков.**

В Mathcad существуют несколько способов построения графиков:

1. Двумерные графики:

• Построение графиков функций с одним переменным.

• Построение графиков зависимостей между двумя переменными.

2. Трехмерные графики:

• Визуализация функций с двумя переменными.

• Построение поверхностей и контурных графиков.

3. Параметрические графики:

• Построение графиков, основанных на параметрических уравнениях.

4. Графики данных:

• Визуализация экспериментальных данных или точек с помощью точечных диаграмм.

5. Комбинированные графики:

• Наложение нескольких графиков на одном изображении для сравнения.

6. Анимация:

• Создание анимационных графиков для динамических систем или изменений во времени.

7. Настройка графиков:

• Изменение стилей, цветов, меток осей и других параметров для улучшения визуализации.

1. **Текстовые регионы системы Mathcad**

В Mathcad текстовые регионы используются для добавления пояснительных заметок, комментариев и описаний в документ. Они позволяют пользователям структурировать информацию и улучшать читаемость расчетов.

Основные характеристики текстовых регионов:

1. Форматирование: Поддержка различных стилей текста, включая заголовки, списки, жирный и курсивный шрифт.

2. Упрощение навигации: Позволяют организовать документ по разделам, что облегчает поиск информации.

3. Интеграция с расчетами: Текстовые регионы могут быть связаны с математическими выражениями, что позволяет объяснять результаты и шаги расчетов.

4. Поддержка формул: Возможность вставки математических формул и уравнений непосредственно в текст.

5. Комментарии: Используются для добавления пояснений к расчетам или для временного исключения кода.

Текстовые регионы являются важным инструментом для создания понятных и профессионально оформленных документов в Mathcad.

1. Встроенные функции системы Mathcad.

1. Арифметические функции: Основные операции (сложение, вычитание, умножение, деление).

2. Тригонометрические функции: Синус, косинус, тангенс и их обратные функции.

3. Статистические функции: Среднее, медиана, стандартное отклонение, дисперсия и другие.

4. Алгебраические функции: Решение уравнений, работа с полиномами, матрицами и векторами.

5. Логические функции: Условия, логические операции (И, ИЛИ, НЕ).

6. Функции работы с массивами: Операции над массивами и матрицами, включая сортировку и фильтрацию.

7. Интерполяционные функции: Линейная интерполяция, сплайны и другие методы.

8. Калькулятор функций: Функции для численного интегрирования и дифференцирования.

9. Специальные функции: Бета-функция, гамма-функция и другие математические функции.

10. Функции работы с графиками: Построение графиков и визуализация данных.

1. Решение уравнений с помощью встроенных функций системы Mathcad.

1. Функция solve: используется для нахождения корней уравнений. Она позволяет решать как алгебраические, так и трансцендентные уравнения.

2. Функция fsolve: применяется для численного решения нелинейных уравнений. Позволяет находить приближенные значения корней.

3. Функция roots: используется для нахождения корней полиномов.

4. Системы уравнений: Mathcad поддерживает решение систем линейных и нелинейных уравнений с помощью соответствующих функций.

5. Графический метод: можно использовать графическое представление функций для визуального поиска корней.

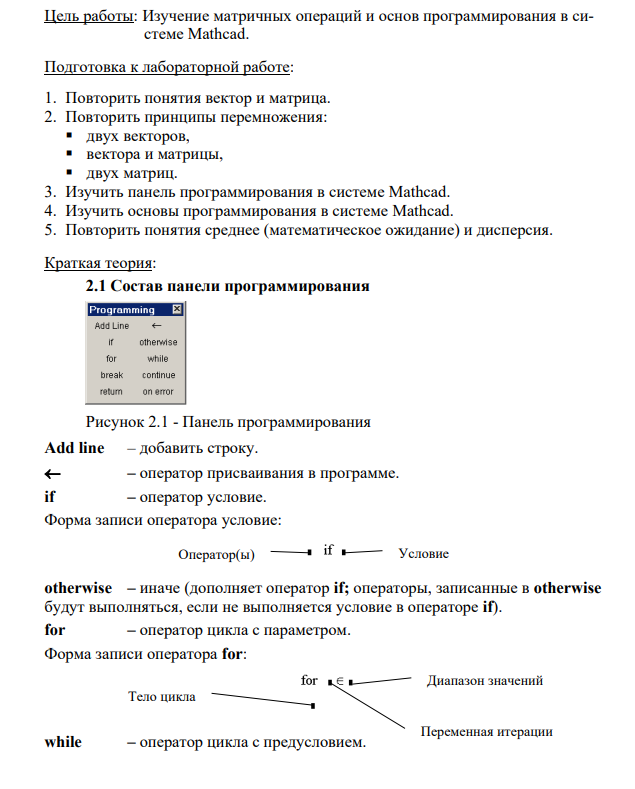
6. Аналитические методы: в некоторых случаях возможно использование символических вычислений для получения точных решений.

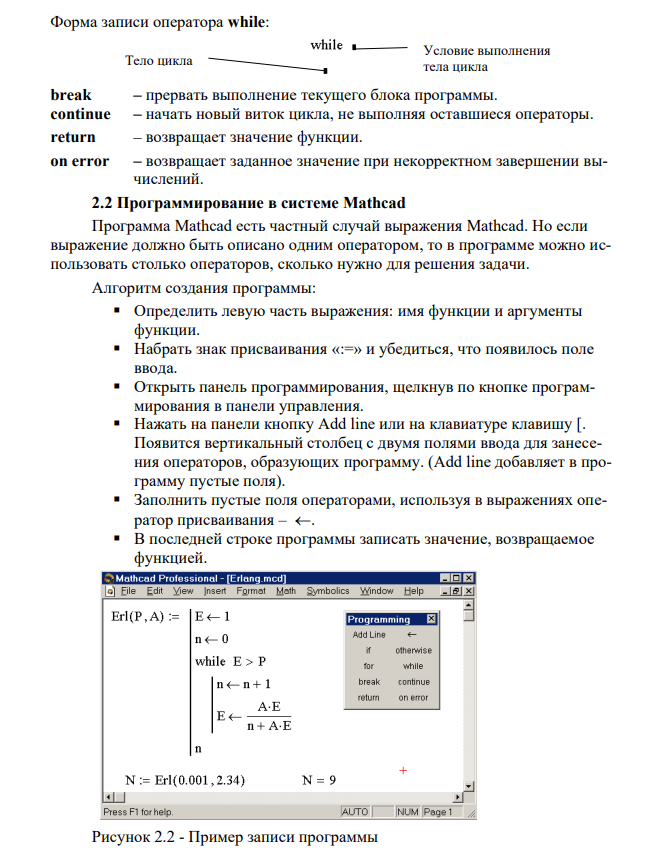
Эти инструменты позволяют пользователям эффективно решать уравнения и анализировать результаты в Mathcad.

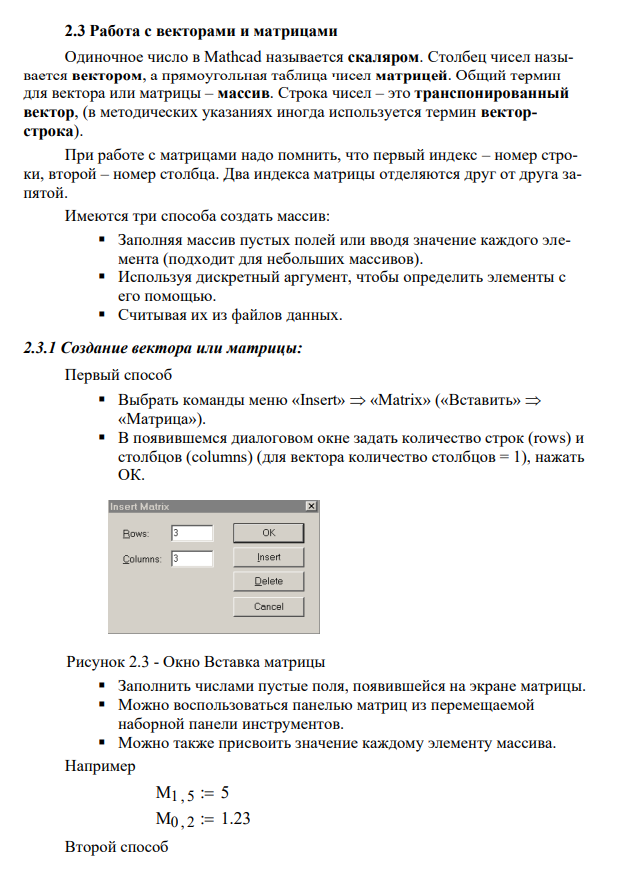
**Вывод**

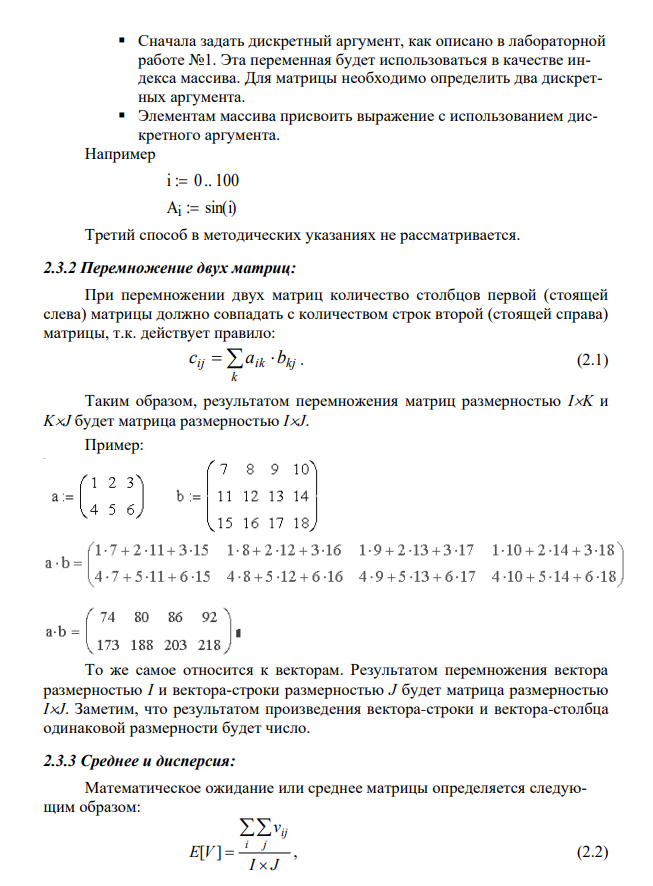
В ходе лабораторной работы были изучены основные пункты меню и кнопки панелей инструментов системы Mathcad, такие как Файл, Правка, Просмотр, Вставка, Формулы и Справка, а также основные панели инструментов, включая математическую, графическую, и панели форматирования текста и анализа. Были рассмотрены переменные скалярного типа и типа дискретного аргумента, способы описания и вызова функций, методы построения графиков, текстовые регионы, встроенные функции и методы решения уравнений. Эти знания позволят эффективно использовать систему Mathcad для выполнения различных математических и аналитических задач.

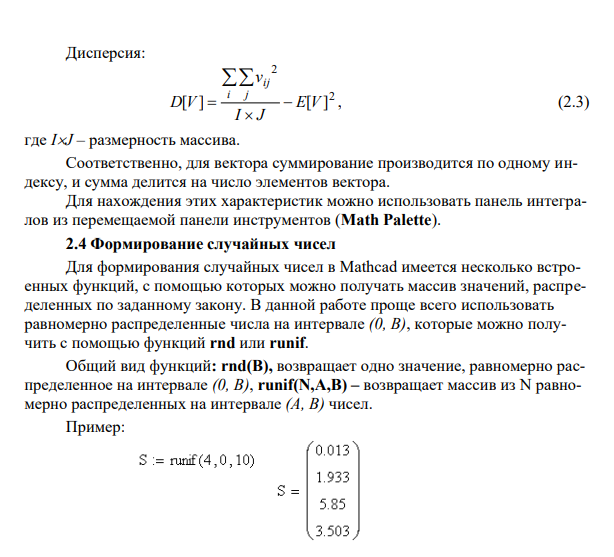
**Часть №2**

****









**3. Код программы**

**% Задание размерностей векторов**

**I = 41;**

**J = 18;**

**% Вызов функции**

**[meanValue, varianceValue] = calculate\_vectors(I, J)**

**function [meanValue, varianceValue] = calculate\_vectors(I, J)**

**% Определение вектора-столбца (случайные числа)**

**vector\_column = rand(I, 1);**

**% Определение вектора-строки (случайные числа)**

**vector\_row = rand(1, J);**

**% Перемножение векторов**

**result\_matrix = vector\_column \* vector\_row;**

**% Вычисление среднего значения**

**meanValue = mean(result\_matrix(:));**

**% Вычисление дисперсии**

**varianceValue = var(result\_matrix(:));**

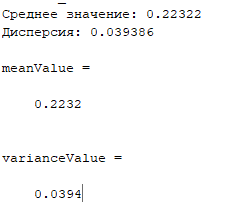
**% Вывод результатов**

**disp(['Среднее значение: ', num2str(meanValue)]);**

**disp(['Дисперсия: ', num2str(varianceValue)]);**

**end**

**4. Результаты выполнения программы**

****

**5. Контрольные вопросы**

1. Переменная скалярного типа в математическом регионе  
   Скалярная переменная в математике — это переменная, принимающая одно числовое значение. В контексте Mathcad переменные могут быть заданы простым присваиванием значения. Например:

a := 5

Здесь a — скалярная переменная, и ей присвоено значение 5.

1. Переменная типа дискретный аргумент в математическом регионе  
   Дискретные переменные используются для описания величин, принимающих конечное или счетное множество значений. В Mathcad можно задать дискретный аргумент с помощью вектора или массива. Например:

x := 0, 1, 2, 3

Здесь переменная x представляет собой набор дискретных значений.

1. Программирование в системе Mathcad  
   Mathcad позволяет использовать программирование для создания сложных вычислительных алгоритмов. Для этого применяются операторы циклов, условий и функций. Пример программы:

if a > 5

b := a \* 2

else

b := a + 1

В этом примере выполняется условная операция: если a больше 5, то b удваивается; иначе прибавляется 1.

1. Разработка функции в системе Mathcad  
   Функции в Mathcad задаются с использованием имен переменных и аргументов. Например, функция, которая вычисляет квадрат числа, может быть записана так:

f(x) := x^2

Затем функция может быть вызвана с любым значением x, например: f(3) вернет 9.

1. Вектор-строка и вектор-столбец; способы их задания  
   В Mathcad векторы можно задать как строку (вектор-строку) или столбец (вектор-столбец). Вектор-строка задается как набор элементов в одной строке:

v\_row := [1, 2, 3]

Вектор-столбец задается как набор элементов в разных строках:

v\_col := [1; 2; 3]

1. Матрицы и способы их задания  
   Матрица в Mathcad может быть задана с помощью квадратных скобок или матричных операторов. Пример:

A := [[1, 2]; [3, 4]]

Это 2x2 матрица с элементами 1, 2, 3 и 4.

1. Перемножение векторов и матриц, а также вектора на матрицу  
   В Mathcad поддерживается стандартное матричное умножение. Для умножения вектора на вектор, матрицы на матрицу или вектора на матрицу можно использовать оператор умножения. Пример умножения матрицы на вектор:

A := [[1, 2]; [3, 4]]

v := [5; 6]

result := A \* v

1. Среднее и дисперсия массивов  
   Mathcad предоставляет встроенные функции для вычисления среднего значения и дисперсии. Пример:

array := [1, 2, 3, 4, 5]

mean := mean(array)

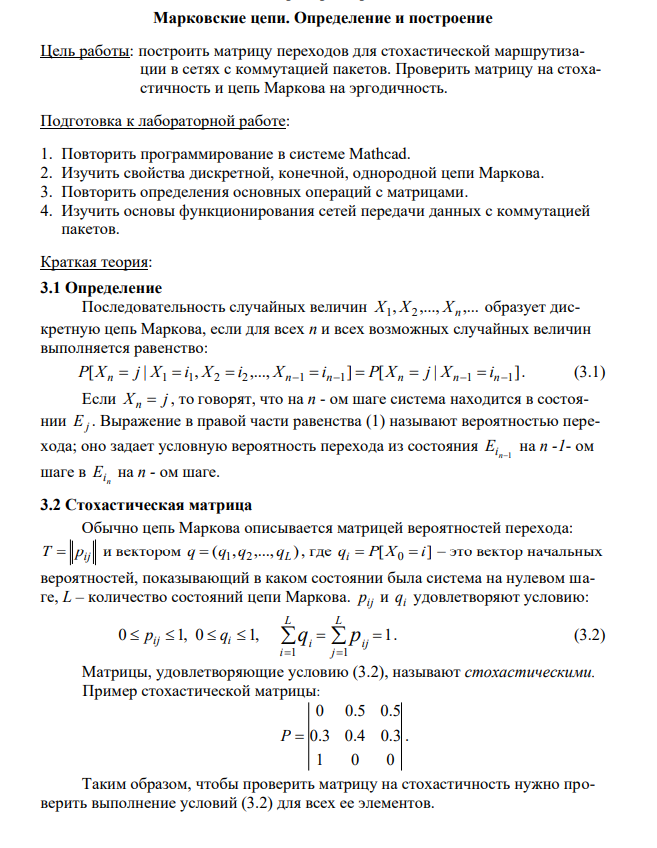
variance := variance(array)

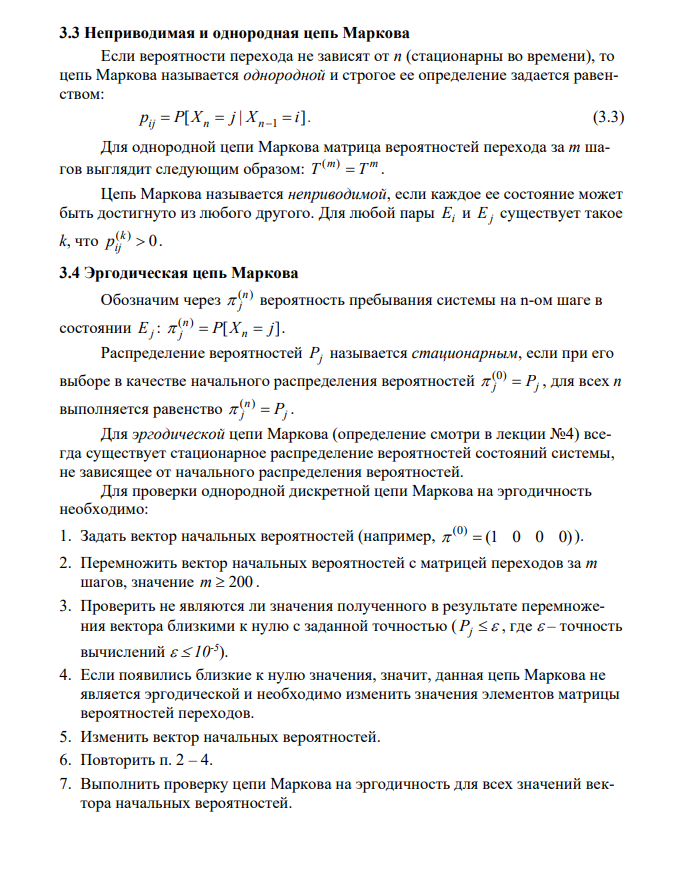
Функция mean() вычисляет среднее арифметическое, а variance() — дисперсию массива.

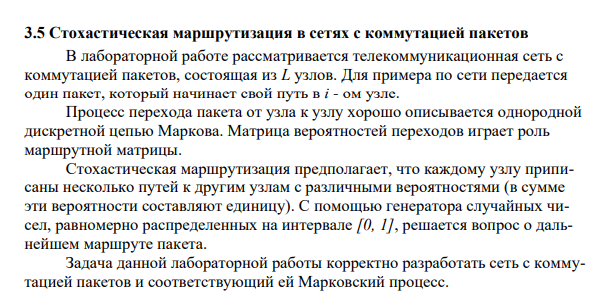
**Вывод:**

В данной лабораторной работе были исследованы переменные скалярного типа и переменные типа дискретный аргумент в математическом регионе Mathcad, возможности программирования в системе, а также разработка и использование функций. Были рассмотрены способы задания векторов и матриц, их перемножение и методы вычисления среднего и дисперсии массивов. Эти знания позволяют эффективно использовать Mathcad для различных математических и вычислительных задач, применяя скалярные и дискретные переменные, программирование, функции, векторы, матрицы и статистический анализ данных.

**Часть №3**

****

****

****

**4. Код программы**

**% Задание размерности матрицы переходов**

**L = 4;**

**T = zeros(L, L); % Обнуление всех элементов матрицы**

**% Задание ненулевых элементов матрицы переходов**

**% Пример структуры**

**% Использование предоставленной матрицы P для начальных элементов матрицы T**

**P = [0.8, 0.1, 0.1, 0;**

**0.2, 0.7, 0.1, 0;**

**0.1, 0.2, 0.5, 0.2;**

**0.2, 0.2, 0.2, 0.4];**

**% Вставка матрицы P в матрицу T**

**T(1:4, 1:4) = P;**

**% (Продолжить для всех узлов, чтобы каждый узел имел не менее трех исходящих маршрутов)**

**% Убедимся, что все строки суммируются до 1 (стоимость каждого перехода)**

**for i = 1:L**

**if sum(T(i, :)) == 0 % Если строка пустая, добавляем равномерные вероятности**

**possible\_destinations = setdiff(1:L, i);**

**chosen\_destinations = possible\_destinations(randperm(numel(possible\_destinations), 3));**

**T(i, chosen\_destinations) = 1/3;**

**end**

**T(i, :) = T(i, :) / sum(T(i, :)); % Нормализация строки**

**end**

**% Проверка стохастичности**

**is\_stochastic = stochastic(T);**

**disp(['Матрица стохастическая: ', num2str(is\_stochastic)]);**

**% Проверка эргодичности**

**is\_ergodic = ergodic(T);**

**disp(['Цепь Маркова эргодическая: ', num2str(is\_ergodic)]);**

**function is\_stochastic = stochastic(matrix)**

**[rows, cols] = size(matrix);**

**is\_stochastic = true;**

**for i = 1:rows**

**if abs(sum(matrix(i, :)) - 1) > eps**

**is\_stochastic = false;**

**break;**

**end**

**end**

**end**

**function is\_ergodic = ergodic(matrix)**

**is\_ergodic = all(eig(matrix) > 0);**

**end**

**5. Результат выполнения программы**

****

**6. Контрольные вопросы**

1. **Определение цепи Маркова**

Цепь Маркова — это случайный процесс с дискретным временем или пространством состояний, в котором будущее состояние системы зависит только от текущего состояния, а не от предыдущих состояний. Этот процесс обладает так называемым "свойством Маркова", согласно которому вероятность перехода в следующее состояние определяется только текущим состоянием.

1. **Классификация цепей Маркова**

Цепи Маркова можно классифицировать по следующим признакам:

По числу состояний: конечные и бесконечные.

По характеру времени: с дискретным и непрерывным временем.

По типу переходов: однородные и неоднородные (зависимость от времени).

По структуре переходов: приводимые и неприводимые, апериодические и периодические.

1. **Свойства цепей Маркова**

Свойство Маркова: будущее состояние зависит только от текущего состояния.

Однородность: вероятность перехода не зависит от времени.

Стационарность: цепь стремится к стационарному распределению при достаточном количестве шагов.

Эргодичность: цепь посещает все состояния с ненулевой вероятностью за достаточно большое время.

1. **Состояния цепи Маркова**

Состояния цепи Маркова представляют собой возможные положения системы в каждый момент времени. Эти состояния могут быть классифицированы как:

Поглощающие состояния: состояние, из которого нет выхода.

Приводимые состояния: состояния, которые можно достичь из любого другого состояния.

Транзиентные состояния: состояния, которые могут быть посещены только временно.

1. **Что такое неприводимая цепь Маркова?**

Неприводимая цепь Маркова — это цепь, в которой существует положительная вероятность достичь любого состояния из любого другого состояния. В такой цепи все состояния связаны друг с другом.

1. **Что такое апериодическая цепь Маркова?**

Апериодическая цепь Маркова — это цепь, у которой период возвращения в любое состояние равен 1, то есть нет фиксированного числа шагов, через которое система обязательно возвращается в состояние.

1. **Что такое однородная цепь Маркова?**

Однородная цепь Маркова — это цепь, в которой вероятности переходов из одного состояния в другое не зависят от времени. Они постоянны на каждом шаге времени.

1. **Свойство эргодичности**

Эргодичность — это свойство цепи Маркова, при котором система со временем посещает все состояния с равной или заранее определенной частотой, вне зависимости от начального состояния. Эргодическая цепь является одновременно неприводимой и апериодической.

1. **Стационарное распределение состояний цепи Маркова**

Стационарное распределение — это распределение вероятностей состояний цепи Маркова, при котором вероятности состояний остаются неизменными со временем. Это состояние достигается, когда вероятность перехода в каждое состояние равна вероятности ухода из него.

1. **Общие понятия о вычислительных сетях с коммутацией пакетов и методах маршрутизации в них**

Вычислительные сети с коммутацией пакетов — это сети, в которых данные разбиваются на небольшие блоки (пакеты), которые передаются независимо друг от друга через сеть. Методы маршрутизации в таких сетях направлены на определение оптимального пути для передачи пакетов от источника к назначению, включая динамическую и статическую маршрутизацию.

**Вывод по лабораторной работе**

В ходе выполнения лабораторной работы были изучены основные функции и возможности системы Mathcad, включая работу с математическими и графическими регионами, встроенными функциями, а также панелями для вычисления производных и интегралов. В процессе работы была определена и построена графически функция одной переменной и её производная, выполнен графический и аналитический метод решения уравнения. Также была успешно построена поверхность для функции двух переменных. Все расчеты и графики были соответствующим образом оформлены и подписаны в текстовых регионах. Работа выполнена согласно требованиям методических указаний, файл сохранен в соответствии с заданными критериями.