



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»  
(ДГТУ)**

**КАФЕДРА «РОБОТОТЕХНИКА И МЕХАТРОНИКА»**

## **Практикум**

по выполнению лабораторной работы

по дисциплине

**«Моделирование мехатронных систем»**

Ростов-на-Дону

2023

Составители: старший преподаватель Юсупов А.Р.

Практикум по выполнению лабораторной работы по дисциплине «Моделирование мехатронных систем». ДГТУ, г. Ростов-на-Дону, 2023 г.

В практикуме кратко изложены теоретические вопросы, необходимые для успешного выполнения лабораторной работы, рабочее задание и контрольные вопросы для самопроверки.

Предназначено для обучающихся по направлению подготовки (код, название):  
15.03.06 Мехатроника и робототехника

Ответственный за выпуск:

И.о. зав. кафедрой (руководитель структурного подразделения, ответственного за реализацию ОПОП) Изюмов Андрей Игоревич

© Издательский центр ДГТУ, 2023г.

## Содержание

|   |    |
|---|----|
| Введение.....   | 4  |
| Теоретическая часть.....  | 4  |
| Лабораторная работа №1 «Изучение основных возможностей пакета MatLab, а также получение практических навыков работы с матрицами и средствами графической визуализации вычислений.»..... | 25 |
| Лабораторная работа №2 «Имитационное моделирование для решения инженерно-вычислительных задач (методом Монте-Карло)».....   | 29 |
| Лабораторная работа №3 «Моделирование простой модели в Simulink».....   | 31 |
| Лабораторная работа №4 «Моделирование временных характеристик динамической системы с прямыми связями».....  | 32 |
| Лабораторная работа №5 «Моделирование временных характеристик динамической системы с обратной связью».....  | 34 |
| Лабораторная работа №6 «Комплексное моделирование характеристик динамической системы с прямыми связями».....  | 36 |

## Введение

В рамках дисциплины «Моделирование мехатронных систем» студенты получают современные знания по моделированию мехатронных систем в программной среде MATLAB, выполняя соответствующие лабораторные работы.

### Теоретическая часть

#### Краткая характеристика MATLAB

MATLAB - это высокопроизводительный инструмент для выполнения технических расчетов. Он включает в себя вычисления, визуализацию и программирование в удобной среде, где задачи и решения выражаются в форме, близкой к математической. Типичное использование MATLAB - это:

- математические вычисления
- создание алгоритмов
- моделирование
- анализ данных, исследования и визуализация
- научная и инженерная графика
- разработка приложений, включая создание графического интерфейса

В MATLAB важная роль отводится специализированным группам программ, называемым *toolboxes*. *Toolboxes* – это всесторонняя коллекция функций MATLAB, которые позволяют решать частные технические задачи. *Toolboxes* применяются для обработки сигналов, анализа изображений, моделирования систем управления и т.д. Для удобства работы в состав MATLAB входит программа Simulink, которая позволяет выполнять моделирование систем в графическом виде. Simulink содержит библиотеку элементов для построения систем из отдельных блоков и позволяет соединять эти блоки друг с другом с помощью мыши.

Работа с системой в режиме прямых вычислений носит диалоговый характер и происходит по правилу «задал вопрос – получил ответ». Пользователь набирает на клавиатуре вычисляемое выражение, редактирует его (если нужно) в командном окне и завершает ввод нажатием клавиши ENTER.

Примеры:

```
>>1+2
```

```
ans=3
```

```
>>ans/10
```

```
ans=0.08415
```

```
>>4*5; %(для блокировки вывода результата вычислений добавьте символ ";"  
(без кавычек) в конец выражения)
```

```
>>sin(1)
ans=0.8415
```

### Замечание:

- Когда выходная переменная не определена, MATLAB использует переменную ans, коротко от answer - ответ, для хранения результатов вычисления.
- Если вводимое математическое выражение окажется настолько длинным, что на него не хватит одной строки, то часть выражения можно перенести на новую строку с помощью знака многоточия «...» (3 или более точек).
- Текстовый комментарий к выполняемым действиям в MATLAB можно ввести после знака %

Центральным понятием всех математических систем является математическое выражение. Оно задает то, что должно быть вычислено в численном (реже символьном) виде. Но в отличие от других систем, эти выражения в MATLAB включают матрицы. Математические выражения строятся на основе чисел, констант, переменных, операторов, функций и разных спецзнаков.

Вот примеры простых математических выражений:

2.301\*sin(x)

4+exp(3)/5

sqrt(y)/2

sin(pi/2)

В MATLAB нет необходимости в определении типа переменных или размерности. Когда MATLAB встречает новое имя переменной, он автоматически создает переменную и выделяет соответствующий объем памяти. Если переменная уже существует, MATLAB изменяет ее состав и если это необходимо выделяет дополнительную память. Например, если мы назначим:

a = 25,

**система создает матрицу 1x1 с именем a и сохраняет значение 25 в ее единственном элементе.** Имя переменной может содержать сколько угодно символов, но запоминается и идентифицируется только 31 начальный символ. Имя должно начинаться с буквы, может содержать буквы, цифры и символ подчеркивания. Недопустимо включать в имена переменных пробелы и спец. знаки +, - и т.д.

## Операторы:

+ сложение

- вычитание

\* умножение

/, \ деление

^ степень

' комплексно сопряженное транспонирование

() определение порядка вычисления

## Отношения:

< - меньше

> - больше

<= - меньше или равно

>= - больше или равно

== - равно

~= - не равно

MATLAB предоставляет большое количество элементарных математических функций, таких как `abs`, `sqrt`, `exp`, `sin`. Вычисление квадратного корня или логарифма отрицательного числа не является ошибкой: в этом случае результатом является соответствующее комплексное число. Чтобы вывести список всех элементарных математических функций, наберите:

```
help elfun
```

Для вывода более сложных математических и матричных функций, наберите

```
help specfun
```

```
help elmat
```

По умолчанию MATLAB выдает числовые результаты в нормализованной форме с четырьмя цифрами после десятичной точки и одной до нее. Однако это не всегда удобно, поэтому в MATLAB предусмотрена возможность задавать различные форматы представления чисел.

Для установки формата используется команда:

```
>>format name, где name - имя формата
```

*short* - короткое представление в фиксированном формате (5 знаков)

*short e* - короткое представление в экспоненциальном формате (5 знаков мантииссы и 3 знака порядка)

*long* - длинное представление в фиксированном формате (15 знаков)

*long e* - (15 знаков мантииссы и 3 знака порядка)

*hex* - представление чисел в шестнадцатеричной форме

*bank* - представление для денежных единиц

Задание формата сказывается только на форме вывода чисел. Вычисления все

равно происходят в формате двойной точности, а ввод возможен в любом удобном для пользователя виде. Альтернативный вариант задания формата представления числовых данных осуществляется через File -> Preferences -> Command Window -> Numeric Format.

### Задание векторов и матриц

Вы можете вводить матрицы в MATLAB несколькими способами:

а) вводить полный список элементов

>>V=[1 2 3] - задает вектор, имеющий три элемента;

>>M=[1 2 3; 4 5 6; 7 8 9]; - задает матрицу 3x3;

>>V=[2+2/(3+4) exp(5) sqrt(10)]; - задает значение выражения в качестве элементов матрицы.

>> A(end,3:end)=-1

>>A(1:5,1:5)=0

A =

|   |   |    |    |    |
|---|---|----|----|----|
| 0 | 0 | 0  | 0  | 0  |
| 0 | 0 | 0  | 0  | 0  |
| 0 | 0 | 0  | 0  | 0  |
| 0 | 0 | 0  | 0  | 0  |
| 0 | 0 | -1 | -1 | -1 |

A =

|   |   |   |   |   |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |

б) генерировать матрицы, используя встроенные функции

Имеется ряд специальных функций для задания векторов и матриц. Например, функция `magic(n)` задает магическую матрицу размера  $n \times n$ , у которой сумма всех столбцов, всех строк и даже диагоналей равна одному и тому же числу:

>>M=magic(4), а также:

*zeros* все нули

*ones* все единицы

*rand* равномерное распределение случайных элементов

*randn* нормальное распределение случайных элементов

*eye* единичная матрица (в диагонали единицы)

>> Z=zeros (2,4)

>>F=5\*ones(3,3)

>>N=10\*rand(1,10)

в) загружать матрицы из внешних файлов

Команда *load* считывает двоичные файлы, содержащие матрицы, созданные в

MATLAB ранее, или текстовые файлы, содержащие численные данные. Текстовые файлы должны быть сформированы в виде прямоугольной таблицы чисел, отделенных пробелами, с равным количеством элементов в каждой строке.

Например, создадим вне MATLAB текстовый файл, содержащий 4 строки:

|      |      |      |      |
|------|------|------|------|
| 16.0 | 3.0  | 2.0  | 13.0 |
| 5.0  | 10.0 | 11.0 | 8.0  |
| 9.0  | 6.0  | 7.0  | 12.0 |
| 4.0  | 15.0 | 14.0 | 1.0  |

Сохраним этот файл под именем *magik.dat*. Тогда команда *loadmagik.dat* прочитает этот файл и создаст переменную *magik*, содержащую нашу матрицу.

г) загружать матрицы из М-файлов

Вы можете создавать свои собственные матрицы, используя М-файлы, которые представляют собой текстовые файлы, содержащие код MATLAB. Просто создайте файл с выражением, которое вы хотите написать в командной строке MATLAB. Сохраните его в файле с расширением *.m*

Например, создадим файл, содержащий строку:

*A=[16 3 2 13; 5 10 11 8; 9 6 7 12; 4 15 14 1];*

Сохраним его под именем *magik.m*. Тогда выражение *magik* прочитает файл и создаст переменную *A*, содержащую исходную матрицу.

**Замечание:** Очень часто необходимо произвести формирование упорядоченных числовых последовательностей (числовых массивов - векторов). Первоначальное создание таких массивов производится с помощью оператора ":" (без кавычек)

*{Начальное\_значение}:{Шаг}:{Конечное\_значение}*

Пример:

*>>1:5*

*>>i=0:2:10*

*>>x=1:-.2:0*

*>>x=0:5*



## Обращение к элементам матрицы

Если мы ввели какую-либо матрицу, то она автоматически запоминается средой MATLAB и мы можем к ней легко обратиться, указав имя переменной, которой она присвоена.

Для выбора отдельного элемента, расположенного в строке  $i$  и столбце  $j$  матрицы можно использовать индексы  $A(i,j)$

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

```
>>A(2,2)=10
```

Если вы пытаетесь использовать значение элемента вне матрицы, MATLAB выдаст ошибку:

```
>> t=A(4,5)
```

*Index in position 2 exceeds array bounds. Index must not exceed 4.*

Найдем сумму элементов в четвертом столбце матрицы A, набрав:

$A(1,4) + A(2,4) + A(3,4) + A(4,4)$  получим

```
ans = 34
```

То же самое можно сделать, набрав:

```
sum(A(:,4))
```

Суммирование элементов, транспонирование и диагонализация матрицы

Подсчитаем сумму элементов всех столбцов матрицы A:

создавать и сохранять в виде m-файла.

```
sum(A)
```

```
ans =
```

```
34 34 34 34
```

MATLAB предпочитает работать со столбцами матрицы. Для того чтобы получить сумму в строках, необходимо транспонировать матрицу, подсчитать сумму в столбцах, а потом транспонировать результат. Операция транспонирования обозначается апострофом или одинарной кавычкой. Она зеркально отображает матрицу относительно главной диагонали и меняет строки на столбцы.

```
>>sum(A')'
```

Сумму элементов на главной диагонали можно получить с помощью функции `diag`, которая выбирает эту диагональ.

```
diag(A)
```

```
ans = 16 10 7 1
```

И суммированием этих элементов:

```
sum(diag(A))
```

```
ans = 34
```

Замечание: Для доступа к последней строке или столбцу матрицы удобно пользоваться оператором `end`. Так:

```
sum(A(:,end)),
```

**вычисляет сумму элементов в последнем столбце матрицы A.**

Матричные и поэлементные вычисления

При работе с массивами и матрицами для выполнения поэлементных операций перед символом операции вводится точка "." (без кавычек).

Пример:

```
>> a=[1 2 3] % Задаем массив из трех элементов
```

```
>> a*a % Пытаемся умножить массив сам на себя
```

Ответ системы - сообщение об ошибке:

```
>> a*a
```

*Error using \* Incorrect dimensions for matrix multiplication. Check that the number of columns in the first matrix matches the number of rows in the second matrix. To perform elementwise multiplication, use '.\*'.*

Причина ошибки в том, что по умолчанию MATLAB пытается произвести матричное умножение.

Ответом на

```
>> a.*a
```

Будет

```
ans = 1 4 9
```

## Построение графиков функций

MATLAB позволяет строить графики функций в линейном, логарифмическом и полулогарифмическом масштабах. Общий порядок построения графиков функций:

- Задать аргумент в формате  $x = \{\text{нач. значение}\} : \{\text{шаг}\} : \{\text{кон. значение}\}$ .
- Вычислить функцию, например,  $y = f(x)$ .
- Построить график функции при помощи функции `plot(x,y,s)`

Построение графиков функций одной переменной в линейном масштабе осуществляется при помощи функции `plot`. В зависимости от входных аргументов функция `plot` позволяет строить один или несколько графиков, изменять цвет и стиль линий и добавлять маркеры на каждый график.

Функция `plot` имеет различные формы, связанные с входными параметрами, например `plot(y)` создает график зависимости элементов  $y$  от их индексов. Если вы зададите два вектора в качестве аргументов, `plot(x,y)` создаст график зависимости  $y$  от  $x$ . Например, для построения графика значений функции  $\sin$  от нуля до  $2\pi$ , сделаем следующее

```
t = 0:pi/100:2*pi;
```

```
y = sin(t);
```

```
plot(t,y)
```

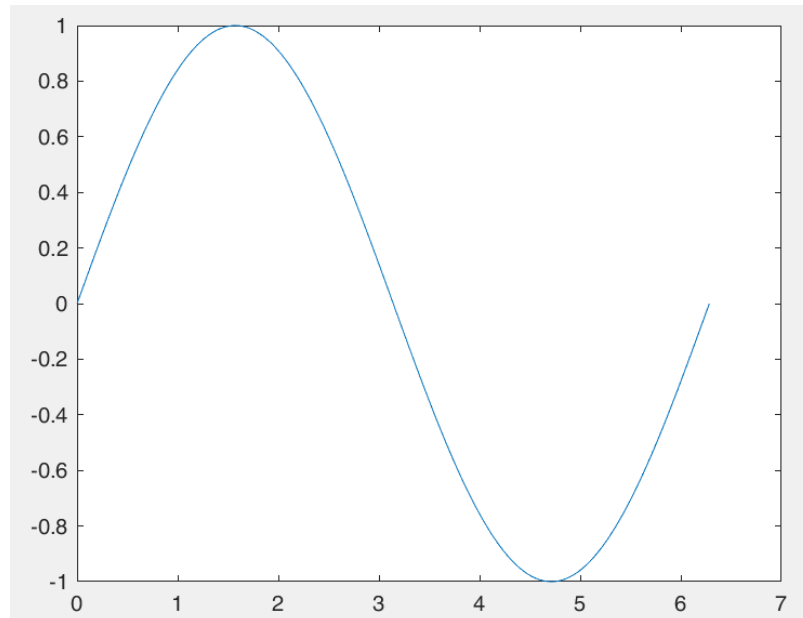


Рис. 1 - График функции  $y=\sin(x)$

Возможно задание цвета, стиля линий и маркеров графиков при создании графика, с помощью параметра команды plot:

plot(x, y, s')), где s - строковая константа, задающая параметры линии графика:

**Таблица 1.1**

| Цвет линии |            | Тип точки |                    | Тип линии |                 |
|------------|------------|-----------|--------------------|-----------|-----------------|
| <i>U</i>   | Желтый     | .         | Точка              | -         | сплошная        |
| <i>m</i>   | Фиолетовый | o         | Кружок             | :         | двойной пунктир |
| <i>C</i>   | Голубой    | x         | Крест              | -.        | штрих пунктир   |
| <i>R</i>   | Красный    | +         | Плюс               | --        | штрих           |
| <i>g</i>   | Зеленый    | *         | звездочка          |           |                 |
| <i>b</i>   | Синий      | s         | Квадрат            |           |                 |
| <i>w</i>   | Белый      | d         | Ромб               |           |                 |
| <i>K</i>   | черный     | v         | треугольник вверх  |           |                 |
|            |            | <         | треугольник влево  |           |                 |
|            |            | >         | треугольник вправо |           |                 |
|            |            | p         | пятиугольник       |           |                 |
|            |            | h         | шестиугольник      |           |                 |

Если на одном графике нужно отобразить несколько функций, например,  $y_1=f(x)$  и  $y_2=f(x)$ , то они вначале вычисляются, а затем выводятся процедурой plot(x,y1,'s1',x,y2,'s2'...), в которой в качестве параметров для каждой функции следуют группы <аргумент, функция, тип линии>.

```

y2= sin(t-.25);
y3 = sin(t-.5);
plot(t, y, 'b', t, y2, 'g', t, y3, 'r')

```

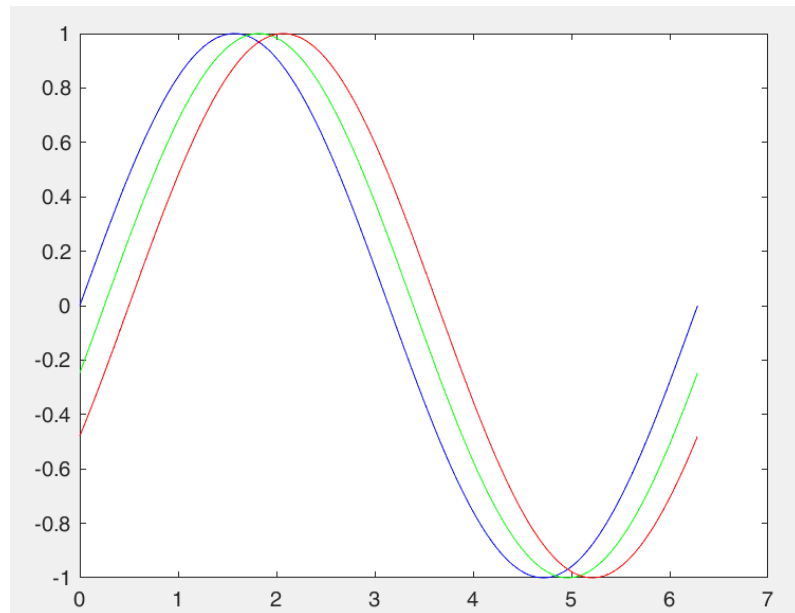


Рис. 2 - Изображение нескольких функций на одном графике  $y=\sin(x)$

Удобство использования графиков во многом зависит от дополнительных элементов оформления: координатной сетки, подписей к осям, заголовка и легенды.

#### Управление осями

Производится с помощью функции `axis`. Она имеет несколько возможностей для настройки масштаба, ориентации и коэффициента сжатия.

```
axis({xmin xmax ymin ymax})
```

Обычно MATLAB находит максимальное и минимальное значение и выбирает соответствующий масштаб осей. Функция `axis` заменяет значения по умолчанию предельными значениями, вводимыми пользователем. Также можно использовать ключевые слова для управления внешним видом осей. Например:

*axis square* - создает x и y оси равной длины,

*axis equal* - создает отдельные отметки приращений для x и y осей одинаковой длины.

Тогда, функция: `plot(exp(i*t))`, следующая либо за `axis square`, либо за `axis equal` превращает овал в правильный круг.

*axis auto* - возвращает значения по умолчанию и переходит в автоматический режим;

*axis on* - включает обозначения осей и метки промежуточных делений;

*axis off* - выключает обозначения осей и метки промежуточных делений;

Замечание: Иногда требуется сравнить поведение двух функций, значения которых сильно отличаются друг от друга. График функции с небольшими

значениями практически сливаются с осью абсцисс, и установить его вид не удастся. В этой ситуации помогает функция *plotyy*, которая выводит графики в окно с двумя вертикальными осями, имеющими подходящий масштаб.

Сетка наносится командой *grid on*, подписи к осям размещаются при помощи *xlabel*, *ylabel*, заголовок задается командой *title*. Наличие нескольких графиков на одних осях требует помещения легенды командой *legend* с информацией о линиях.

Пример: Следующие команды выводят графики изменения суточной температуры.

```
>> time = [0 4 7 9 10 11 12 13 13.5 14 14.5 15 16 17 18 20 22];  
>> temp1 = [14 15 14 16 18 17 20 22 24 28 25 20 16 13 13 14 13];  
>> temp2 = [12 13 13 14 16 18 20 20 23 25 25 20 16 12 12 11 10];  
>> plot(time, temp1, 'ro-', time, temp2, 'go-')  
>> grid on  
>> title('Суточные температуры')  
>> xlabel('Время (час.)')  
>> ylabel('Температура (C)')  
>> legend('10 мая', '11 мая')
```

Использование TEX-представления позволяет применять греческие буквы, математические символы и различные шрифты. Следующий пример демонстрирует эту возможность:

```
t = -pi:pi/100:pi;  
y = sin(t);  
plot (t,y)  
axis([-pi pi -1 1])  
xlabel('-\pi \leq t \leq \pi')  
ylabel('sin(t)')  
title('График функции sin')  
text(-1, -1/3, '\it{Отметьте нечетную симметрию}')
```

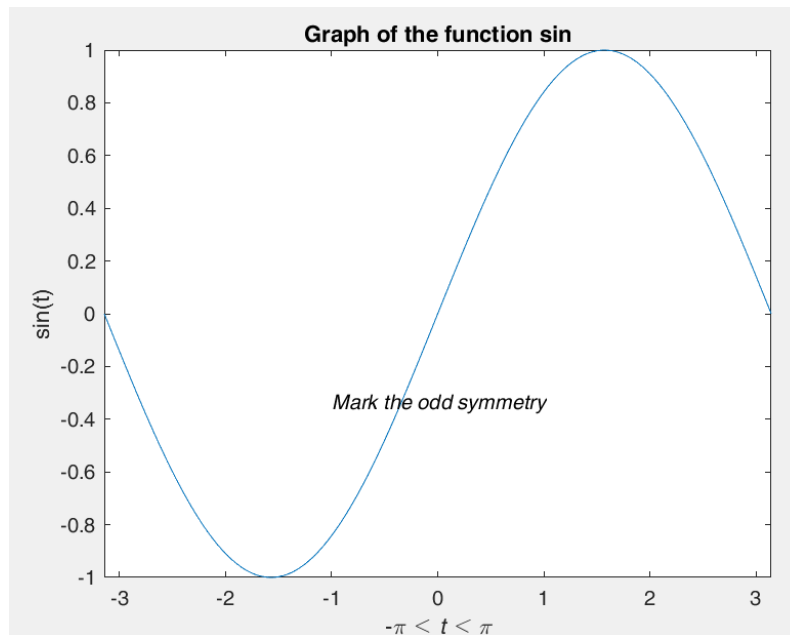


Рис. 3 - График функции  $y=\sin(x)$

### Окна изображений

Функция `plot` автоматически открывает новое окно изображения (далее окно), если до этого его не было на экране. Если же оно существует, то `plot` использует его очищая его содержимое.

Для открытия нового окна и выбора его по умолчанию, наберите:

`figure`

Для того, чтобы сделать существующее окно текущим -

`figure (n)`,

где  $n$  - это номер в заголовке окна. В этом случае результаты всех последующих команд будут выводиться в это окно.

Замечание: Чтобы добавить кривые на существующий график, следует воспользоваться командой `hold on`. Команда `clf` очищает все текущее окно. Команда `cla` - убирает только график, а оси, заголовок и названия осей оставляет

### Подграфики

Функция `subplot` позволяет выводить множество графиков в одном окне или распечатывать их на одном листе бумаги.

`subplot ({ m }, { n }, { p })` -

Разбивает окно изображений на матрицу  $m$  на  $n$  подграфиков и выбирает  $p$ -ый подграфик текущим. Графики нумеруются вдоль первого в верхней строке, потом во второй и т.д. Например, для того, чтобы представить графические данные в четырех разных подобластях окна необходимо выполнить следующее:

`t = -pi:pi/100:pi;`

`x = cos(t);`

`y = sin(t);`

```

z = 1./sin(t);
k = 1./cos(t);
subplot(2,2,1); plot(t,x)
subplot(2,2,2); plot(t,y)
subplot(2,2,3); plot(t,z)
subplot(2,2,4); plot(t,k)

```

### Основы трехмерной графики

MATLAB предоставляет различные способы визуализации функций двух переменных - построение трехмерных графиков и линий уровня, параметрически заданных линий и поверхностей.

Для формирования трехмерных графиков необходимо:

- Сгенерировать матрицы с координатами узлов сетки на прямоугольной области определения функции.
- Вычислить функцию в узлах сетки и записать полученные значения в матрицу.
- Использовать одну из графических функций MATLAB.

#### Генерация сетки

Сетка генерируется при помощи команды *meshgrid*, вызываемой с двумя аргументами, векторами, элементы которых соответствуют сетке прямоугольной области построения функции. Можно использовать один аргумент, если область построения функции - квадрат.

$[X,Y] = \text{meshgrid}(x,y)$  - преобразует область, заданную векторами  $x,y$  в массивы  $X$  и  $Y$ , которые могут быть использованы для вычисления функции двух переменных и построения трехмерных графиков.

#### Трехмерный график с аксонометрией

Команда *plot3(...)* является аналогом команды *plot(...)*, но относится к функциям двух переменных  $z(x,y)$ . Она строит аксонометрическое изображение трехмерных поверхностей.

*plot3(x,y,z)* - строит массив точек, представленных векторами  $x, y$  и  $z$ , соединяя их отрезками прямых.

Задание свойств линий и маркеров для *plot3()* производится также как и для *plot()*

```

plot3(X,Y,Z,S)
>> plot3(X,Y,Z, '-o')
plot3(x1,y1,z1,s1, x2,y2,z2,s2,...)
>> plot3(X,Y,Z, '-k', Y,X,Z, '-k')

```

Наиболее представительными и наглядными являются сетчатые графики поверхностей с заданной или функциональной окраской. В названии их команд

присутствует слово `mesh`. Имеется три группы таких команд:

Сетчатый 3D-график с функциональной окраской

`mesh(X,Y,Z,C)` - выводит в окно сетчатую поверхность  $Z(X,Y)$  с цветами узлов поверхно-сти, заданными массивом `C`.

`mesh(X,Y,Z)` - аналог предшествующей команды при  $C=Z$ .

Пример: Построить график функции двух переменных

$$z(x,y) = 4 \cdot \sin(2\pi x) \cdot \cos(1.5\pi y) \cdot (1-x)^2 \cdot y \cdot (1-y)$$

на прямоугольной области определения  $x \in [-1,1]$ ,  $y \in [0,1]$ .

Цвет линий соответствует значениям функции. MATLAB рисует только видимую часть поверхности. При помощи команды *hidden off* можно сделать каркасную модель «прозрачной», добавив скрытую часть. MATLAB имеет несколько функций, возвращающих матричный образ поверхностей. Например, функция *peaks(N)* возвращает матричный образ поверхности с рядом пиков и впадин. Такие функции удобно использовать для проверки команд трехмерной графики.

Пример:

```
>> z = peaks(25);
```

```
>> mesh(z)
```

Сетчатый 3D-график с функциональной окраской и проекцией

Иногда график поверхности полезно объединить с контурным графиком ее проекции на плоскость, расположенным под поверхностью.

```
meshc(...)
```

График такого типа дает лучшее представление об особенностях поверхности.

Поверхностный 3D-график с функциональной окраской

Наглядное представление о поверхностях дают графики, использующие функциональную закраску ячеек. Например, цвет окраски поверхности  $z(x,y)$  может быть поставлен в соответствие с высотой  $z$  поверхности с выбором для малых высот темных тонов, а для больших - светлых.

Для построения таких поверхностей используются команды класса *surf(...)*.

`surf(X,Y,Z,C)` - строит цветную параметрическую поверхность по данным матриц `X`, `Y` и `Z` с цветом, задаваемым массивом `C`.

`surf(X,Y,Z)` - аналогична предшествующей команде, где  $C=Z$ .

```
>> colormap('') - задает окраску тонами определенного цвета
```

```
>> shading interp - обеспечивает устранение изображения сетки и задает интерполяцию для оттенков цвета объемной поверхности
```

Применение интерполяции для окраски придает поверхностям и фигурам более реалистичный вид, но фигуры каркасного вида дают более точные количественные данные о каждой точке.

Поверхностный 3D-график с функциональной окраской и освещением



Пожалуй, наиболее реалистичный вид имеют графики поверхностей, в которых имитируется освещение от точечного источника света, расположенного в заданном месте координатной системы. Графики имитируют оптические эффекты рассеивания, отражения и зеркального отражения света. Для получения таких эффектов используется команда `surfl`.

`surfl(Z,S)` или `surfl(X,Y,Z,S)` - строит графики поверхности с подсветкой от источника света, положение которого в системе декартовых координат задается вектором  $S=[S_x, S_y, S_z]$ , а в системе сферических координат - вектором  $S=[AZ, EL]$ ;

При использовании `surfl` удобно задавать цветовые палитры: `copper`, `bone`, `gray`, `pink`, в которых интенсивность цвета изменяется линейно.

Пример.

```
>> [X,Y] = meshgrid(-1:0.05:1, 0:0.05:1);  
>> Z = 4*sin(2*pi*X).*cos(1.5*pi*Y).*(1-X.^2).*Y.*(1-Y)  
>> surfl(X,Y,Z)  
>> colormap('copper')  
>> shading interp
```

По умолчанию источник света имеет азимут, больший на  $45^\circ$ , чем наблюдатель, и тот же угол возвышения. Возможно изменение азимута источника. Например, изменим азимут до  $-90^\circ$  по отношению к наблюдателю, а угол возвышения до нуля.

```
>> [Az, El] = view;  
>> surfl(X,Y,Z,[Az-90,0])  
>> shading interp
```

Азимут отсчитывается от оси, противоположной оси  $y$ , а угол возвышения - от плоскости  $xz$ .

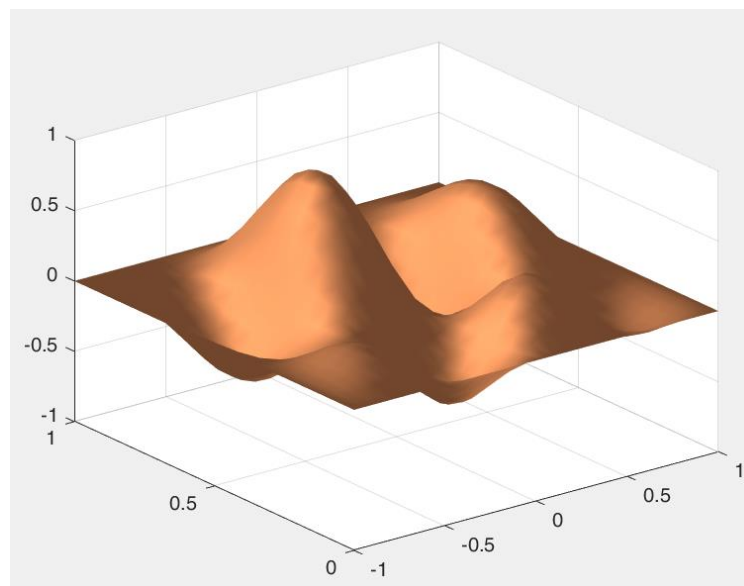


Рис. 8 - Поверхностный 3D-график с функциональной окраской и освещением

## Управление потоками

MATLAB имеет пять видов структур управления потоками:

- оператор if
- оператор switch
- циклы for
- циклы while
- оператор break

В том случае, когда некоторая группа операций должна повторно выполняться несколько раз, используют операторы цикла.

### а) оператор цикла с неизвестным числом повторений (while)

while выражение

<оператор> ...

<оператор>

end

Операторы, указанные в теле цикла, выполняются, пока выражение истинно.

Пример:

*fact=1;*

*k=1;*

*while fact<1e5*

*fact=fact\*k;*

*k=k+1;*

*end*

### б) оператор цикла с известным числом повторений (for)

*for* переменная цикла = начальное значение : <приращение> : конечное значение

<оператор>

...

<оператор>

end

Операторы, указанные в теле цикла, выполняются заданное число раз. При этом переменная цикла последовательно принимает все значения от начального до конечного с приращением после каждого прохода цикла.

*for i=1:1:100*

*s=s+5*

*end*

Замечание: Хорошим стилем являются отступы при использовании циклов для лучшей читаемости, особенно, когда они вложенные.

*for i = 1: m*

```

for j = 1: n
    H(i,j) = 1/( i+j);
end;
end

```

#### в) оператор условного ветвления

|  |  |
|--|--|
| <pre> if &lt;логическое выражение&gt; end </pre> | <pre> if&lt; логическое выражение &gt;     &lt;операторы&gt; elseif &lt; логическое выражение &gt;     &lt;операторы&gt; else     &lt;операторы&gt; end </pre> |
|--|--|

Оператор else не содержит логического условия. Инструкции, связанные с ним, выполняются, если предшествующий оператор if ложен.

Оператор elseif содержит логическое условие, которое вычисляется, если предшествующий оператор if ложен. Оператор elseif может многократно использоваться внутри оператора условия if.

Пример:

```

if n<0
a=2*n;
elseif rem(n,2)==0;
a=n/2;
else a=(n+1)/2; Если n>0 b нечетное, увеличить на 1 и разделить.
end

```

#### г) оператор переключения

```

switch...case ...()ther}vise...end

```

Оператор switch работает, сравнивая значение вычисленного выражения со значениями групп case. Если значение выражения в операторе switch совпадает со значением, указанным в case, то выполняются соответствующая группа операций до следующего оператора case или otherwise, или end. Оператор otherwise выполняется, если <выражение> не совпало ни с одним из значений.

Пример:

```

switch input_num      % выч. выражение
case -1               % значение
disp ('минус один')   % выводит строку text в рабочее окно MATLAB
case 0
disp ('нуль')

```

```
case 1
disp ('плюс один')
otherwise
disp ('другое значение')
end
```

#### д) оператор досрочного выхода из цикла

Оператор break позволяет досрочно выходить из циклов for или while. Во вложенных циклах break осуществляет выход только из самого внутреннего цикла.

```
for i =1:1:100
    s=s+5
    if s>300 then
        break;
    end;
end
```

### Работа с М-файлами

Если требуется выполнять большую последовательность команд или повторять группу команд для различных значений входных переменных, работа в режиме командной строки становится неудобной. Для облегчения работы MATLAB предоставляет возможность организации вычислений в виде так называемых М-файлов. М-файлы являются обычными текстовыми файлами, которые содержат последовательности команд, операторов, функций и т.д., необходимых для вычислений. В состав системы MATLAB входит редактор/отладчик М-файлов.

Для создания m-файла в меню File основного окна MatLab и в пункте New выберите подпункт M-file. Новый файл открывается в редакторе М-файлов. Наберите в редакторе код и сохраните теперь файл с именем mydemo.m в подкаталоге work основного каталога MatLab, выбрав пункт Save as меню File редактора. Для запуска на выполнение всех команд, содержащихся в файле, следует выбрать пункт Run в меню Debug. Созданный М-файл можно как угодно редактировать и повторять все команды заново. Для выполнения только части команд из файла, следует их сначала выделить, а затем выполнить из пункта Evaluate selection меню Text.

Как правило, М-файлы хранятся в каталоге пользователя. Чтобы MatLab мог найти их, следует установить пути, указывающее расположение М-файла.

### Simulink

Для запуска среды Simulink в MATLAB во вкладке HOME нажмите соответствующую иконку

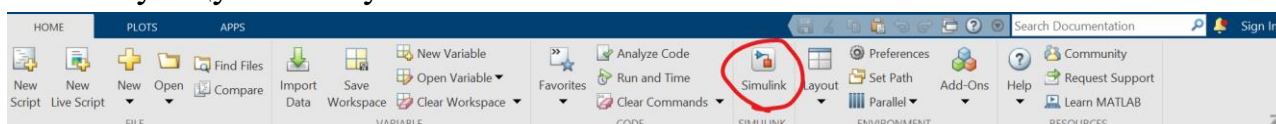


Рис. 9 - Панель управления MATLAB

Откроется окно, в котором необходимо выбрать «Blank Model», создастся пустая модель без элементов.

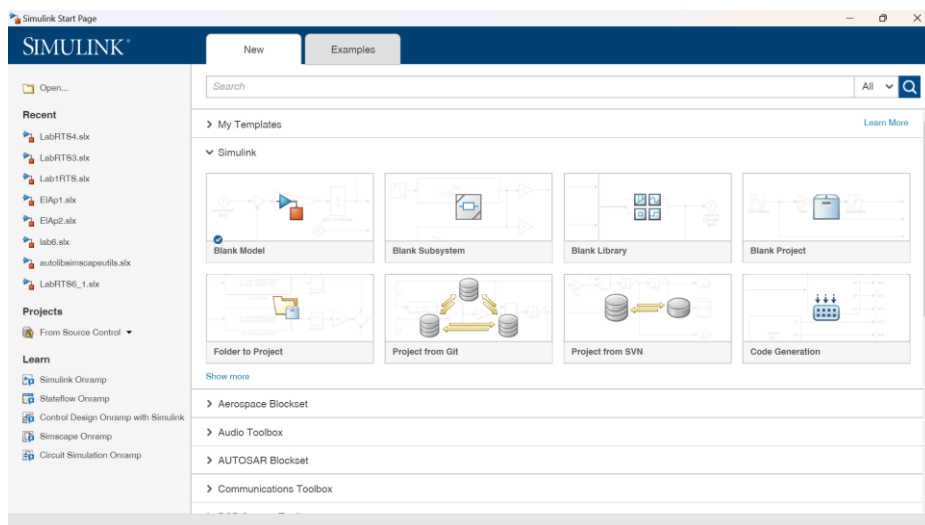


Рис. 10 - Стартовое окно Simulink

Для добавления новых элементов необходимо зайти в «Library Browser» и выбрать элементы из открывшегося окна



Рис. 11 - Панель управления Simulink

Для поиска необходимых элементов можно воспользоваться окном поиска в верхней части интерфейса

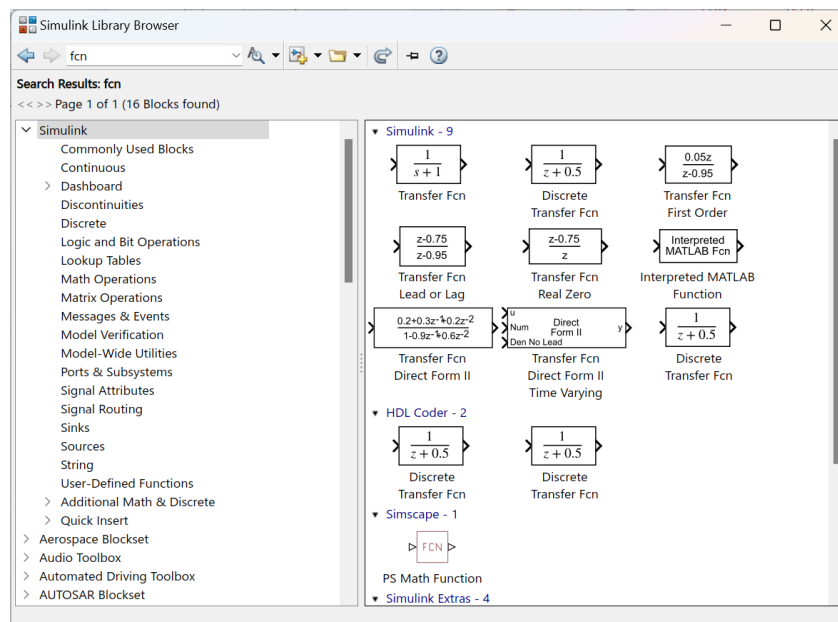


Рис. 12 - Окно библиотеки элементов

## Инструкция по технике безопасности при выполнении лабораторной работы

При выполнении лабораторной работы необходимо соблюдать правила техники безопасности при работе с электрооборудованием, включение электрооборудования, в том числе компьютеров, производить только после разрешения преподавателя или сопровождающего инженера.

**Лабораторная работа №1 «Изучение основных возможностей пакета MatLab, а также получение практических навыков работы с матрицами и средствами графической визуализации вычислений.»**

## Цель работы

Изучить основные возможности пакета MatLab, а также получить практические навыки работы с матрицами и средствами графической визуализации вычислений.

## Рабочее задание

Провести работу над матрицами и графиками в программной среде MATLAB в соответствии с вариантом.

## Материально-техническое обеспечение работы

Персональный компьютер с установленным ПО MATLAB и Word.

## Порядок выполнения работы

## Задание 1

|  |   |   |   |  |  |  |
|--|---|---|---|--|--|--|
| <b>1. Создать матрицу А</b>  |   |   |   |  |  |  |
| Вариант 1  | Вариант 2                                   | Вариант 3                                   | Вариант 4   | Вариант 5                                      | Вариант 6                                      | Вариант 7                                    |
| -1 2 3 5<br>7 8 9 1<br>4 5 6 4<br>11 1 5 9   | 8 4 6 9<br>2 7 4 10<br>3 8 2 6<br>-5 3 7 3  | 8 9 13 2<br>-6 6 4 5<br>8 4 6 4<br>13 1 8 9 | 8 6 7 5<br>7 3 7 15<br>4 8 9 2<br>8 9 8 5           | 1 -2 3 5<br>-7 8 9 1<br>11 3 2 4<br>1 1 4 -9   | 1 5 3 2<br>6 7 3 6<br>3 4 6 7<br>1 5 8 2       | -1 2 3 5<br>3 8 9 1<br>5 3 6 4<br>7 7 5 1    |
| Вариант 8  | Вариант 9                                   | Вариант 10                                  | Вариант 11  | Вариант 12                                     | Вариант 13                                     | Вариант 14                                   |
| 1 5 6 4<br>2 8 9 1<br>4 8 6 4<br>4 5 5 -1  | -1 4 8 5<br>7 7 9 2<br>4 8 6 1<br>1 1 4 0   | -4 4 6 9<br>2 5 4 10<br>3 8 -6 6<br>8 3 7 2 | <b>5 4 3 2</b><br>-4 6 -4 5<br>8 5 6 2<br>3 1 8 9 1 | 8 6 7 5 1<br>0 7 -3 15<br>6 8 11 2<br>8 9 14 5 | 1 2 -3 15<br>5 8 -9 1<br>1 5 2 6<br>1 11 -5 -7 | 1 4 13 2<br>6 7 -4 6<br>3 -2 6 9<br>1 15 4 2 |
| Вариант 15   | Вариант 16                                  | Вариант 17                                  | Вариант 18  | Вариант 19                                     | Вариант 20                                     | Вариант 21                                   |
| -5 2 -5 5<br>3 2 3 1<br>1 3 3 4<br>0 -2 0 1  | 1 0 6 4<br>2 5 9 -11<br>5 9 0 4<br>6 5 5 -1 | 0 9 13 2<br>-6 0 4 3<br>1 4 2 4<br>1 1 0 1  | 1 -3 3 5<br>-7 8 -6 1<br>11 0 0 4<br>0 1 4 -9       | -3 21 5 -2<br>-3 1 3 1<br>-1 3 2 4<br>1 0 1 0  | -4 5 1 9<br>2 5 0 10<br>1 8 -4 6<br>1 3 1 3    | -1 3 6 9<br>2 5 3 10<br>5 12 2 6<br>1 1 1 1  |
| <b>2. Вычислить сумму элементов матрицы А по строкам, по столбцам, а также сумму всех элементов матрицы А.</b> |   |   |   |  |  |  |
| <b>3. Транспонировать матрицу А.</b>   |   |   |   |  |  |  |
| <b>4. Умножить матрицу А на 2.</b>   |   |   |   |  |  |  |
| <b>5. Найти квадратный корень из элементов матрицы А.</b>  |   |   |   |  |  |  |
| <b>6. Возвести в квадрат все элементы матрицы А.</b>   |   |   |   |  |  |  |
| <b>7. Создать массив Х, содержащий 2 строки:</b>   |   |   |   |  |  |  |

[illegible]



|   |  |
|---|--|
|   | 2 строка – четные числа от 1 до 100, в порядке убывания  |
| Вариант 21  | 1 строка - числа от 1 до 100, кратные 3-м, в порядке возрастания<br>2 строка - числа от 1 до 100, кратные 3-м, в порядке возрастания |
| Вариант 22  | 1 строка - числа от 1 до 100, в порядке убывания<br>2 строка - числа от 1 до 100, в порядке убывания                                 |
| <b>8. Создать m-файл, описывающий все произведенные действия.</b> |  |

**Задание 2. Вычислить значения функций 1 и 2 для аргумента в заданном интервале [a b] с шагом h. Вывести графики функций одновременно на одном графике в декартовых координатах. Для разных графиков использовать разный тип линий.**

| №  | Функция 1                        | Функция 2                    | a        | b       | h        |
|----|----------------------------------|------------------------------|----------|---------|----------|
| 1  | $y = \sin(x)$                    | $z = \exp(x+3)/5000 - 1$     | $-2*\pi$ | $2*\pi$ | $\pi/20$ |
| 2  | $y = \cos(x)$                    | $z = 0.00025^3 - x - 0.6$    | $-2*\pi$ | $2*\pi$ | $\pi/20$ |
| 3  | $y = \operatorname{tg}(x) + 0.1$ | $z = (1+x)^6$                | $-2*\pi$ | $2*\pi$ | $\pi/20$ |
| 4  | $y = (x^2-1)/15$                 | $z = 1+\sin(x)$              | $-2*\pi$ | $2*\pi$ | $\pi/20$ |
| 5  | $y = (x^3-2)/15$                 | $z = 5\cos(x)$               | $-2*\pi$ | $2*\pi$ | $\pi/20$ |
| 6  | $y = x^2-10$                     | $z = 0.025\exp(-1.2x)$       | -5       | 5       | 1        |
| 7  | $y = 3\sin(x)$                   | $z=0.015x^3$                 | -5       | 5       | 1        |
| 8  | $y =  \sin(x) \exp(x/2)$         | $z = 5x - x^{1.5} + \sin(x)$ | 0        | 5       | 0.5      |
| 9  | $y = 2-\sin(x)$                  | $z = \cos(x)$                | $-2*\pi$ | $2*\pi$ | $\pi/20$ |
| 10 | $y = \exp(2x+1)$                 | $z = \sin(x)$                | $-2*\pi$ | $2*\pi$ | $\pi/20$ |
| 11 | $y = (x^3-2)/15$                 | $z = x^2-1$                  | $-2*\pi$ | $2*\pi$ | $\pi/20$ |
| 12 | $y = 6\cos(x)$                   | $z = 0.5x^2$                 | $-2*\pi$ | $2*\pi$ | $\pi/20$ |
| 13 | $y = 5x^2$                       | $z = (x^3-3)/11$             | $-2*\pi$ | $2*\pi$ | $\pi/20$ |
| 14 | $y = \exp(-1.7x)$                | $z = \exp(-1.6x)$            | -5       | 5       | 1        |
| 15 | $y = 5\cos(x)$                   | $z = 7\sin(x)$               | -5       | 5       | 1        |
| 16 | $y = 1+\cos(x)$                  | $z = (x^3-6)/13$             | -5       | 5       | 1        |
| 17 | $y = (x^3-3)/10$                 | $z = 7x^3$                   | $-2*\pi$ | $2*\pi$ | $\pi/20$ |
| 18 | $y = 1+\sin(x)$                  | $z = \exp(-1.4x)$            | $-2*\pi$ | $2*\pi$ | $\pi/20$ |
| 19 | $y = 8\cos(x)$                   | $z = (x^2-8)/33$             | $-2*\pi$ | $2*\pi$ | $\pi/20$ |
| 20 | $y = 6x^2+5x^2$                  | $z = (x^3-2)/4$              | -5       | 5       | 1        |
| 21 | $y = 6\cos(x)+\sin(x)$           | $y = x^2$                    | $-2*\pi$ | $2*\pi$ | $\pi/20$ |

**Задание 3. Вычислить значения функции двух аргументов (табл. 1.4) в заданном диапазоне. Вывести функцию в виде 3 трехмерных графиков разного типа. Вывести функцию в виде 2 контурных графиков разного типа.**

| №  | Функция                            | Пределы измерения    |                      |
|----|------------------------------------|----------------------|----------------------|
|    |                                    | x                    | y                    |
| 1  | $z=\sin(x)\cos(y)$                 | от $-2\pi$ до $2\pi$ | От $-2\pi$ до $2\pi$ |
| 2  | $z=\sin(x/2)\cos(y)$               | от $-2\pi$ до $2\pi$ | От $-2\pi$ до $2\pi$ |
| 3  | $z=\sin(2x)\cos(y)$                | от $-2\pi$ до $2\pi$ | От $-2\pi$ до $2\pi$ |
| 4  | $z = \sin(x)\cos(y/2)$             | от $-2\pi$ до $2\pi$ | От $-2\pi$ до $2\pi$ |
| 5  | $z = \sin(x/2)\cos(2y)$            | от $-2\pi$ до $2\pi$ | От $-2\pi$ до $2\pi$ |
| 6  | $z = \sin(2x)\cos(2y)$             | от $-2\pi$ до $2\pi$ | От $-2\pi$ до $2\pi$ |
| 7  | $z = (1+\sin(x)/x)(\sin(y)/y)$     | от $-2\pi$ до $2\pi$ | От $-2\pi$ до $2\pi$ |
| 8  | $z = (\sin(x)/x)\cos(y)$           | от $-2\pi$ до $2\pi$ | От $-2\pi$ до $2\pi$ |
| 9  | $z = \sin(3x)\cos(y/2)$            | от $-2\pi$ до $2\pi$ | От $-2\pi$ до $2\pi$ |
| 10 | $z = (\sin(3x)+3)\cos(2y)$         | от $-2\pi$ до $2\pi$ | От $-2\pi$ до $2\pi$ |
| 11 | $z = \sin(x/3)\cos(4y)$            | от $-2\pi$ до $2\pi$ | От $-2\pi$ до $2\pi$ |
| 12 | $z = (4+\sin(2x)/x)(\sin(4y)/y)$   | от $-2\pi$ до $2\pi$ | От $-2\pi$ до $2\pi$ |
| 13 | $Z = \sin(2x)\cos(3y)$             | от $-2\pi$ до $2\pi$ | От $-2\pi$ до $2\pi$ |
| 14 | $z = (-1+\sin(x)/x)(\sin(y)/3)$    | от $-2\pi$ до $2\pi$ | От $-2\pi$ до $2\pi$ |
| 15 | $z = \sin(x)\cos(4y)$              | от $-2\pi$ до $2\pi$ | От $-2\pi$ до $2\pi$ |
| 16 | $z = (3+\sin(x)/3)(\sin(2y)/y)$    | от $-2\pi$ до $2\pi$ | От $-2\pi$ до $2\pi$ |
| 17 | $z = (1+\sin(3x)/2)(1+\sin(2y)/y)$ | от $-2\pi$ до $2\pi$ | От $-2\pi$ до $2\pi$ |
| 18 | $z = \sin(5x)\cos(1.5y)$           | от $-2\pi$ до $2\pi$ | От $-2\pi$ до $2\pi$ |
| 19 | $z = \sin(5x)\cos(2y)$             | от $-2\pi$ до $2\pi$ | От $-2\pi$ до $2\pi$ |
| 20 | $z = \sin(3x)\cos(3y)$             | от $-2\pi$ до $2\pi$ | От $-2\pi$ до $2\pi$ |
| 21 | $z = \sin(x)\cos(7y)$              | от $-2\pi$ до $2\pi$ | От $-2\pi$ до $2\pi$ |

### Контрольные вопросы

1. Охарактеризуйте моделирование как научный метод и укажите, с какими целями он используется;
2. Какие существуют виды подобия?
3. Из каких этапов состоит процесс моделирования?
4. Какие характеристикам производится оценка моделей?
5. Какие операторы существуют в программном языке MATLAB и как они используются?
6. Приведите краткое описание кода к рабочим заданиям;

## **Лабораторная работа №2 «Имитационное моделирование для решения инженерно-вычислительных задач (методом Монте-Карло)»**

### **Цель работы**

Научится применять метод Монте-Карло при расчете интеграла в программной среде Матлаба.

### **Рабочее задание**

Вычислить интеграл методом Монте-Карло при помощи Матлаба и обработать полученные результаты.

### **Материально-техническое обеспечение работы**

Персональный компьютер с установленным ПО MATLAB и Word.

### **Общие положения**

Метод Монте-Карло — метод компьютерного эксперимента, применяемый для моделирования случайных величин и функций. Сущность МК заключается в многократной реализации состояний моделируемой системы с последующей обработкой информации методами математической статистики..

В частности Метод Монте-Карло - метод численного решения математических задач с помощью моделирования большого количества случайных величин и нахождения их математического ожидания. Вычисление интегралов является основной направленностью этого метода, так как математическое ожидание непрерывной случайной величины выражается через интеграл.

Популярность этому методу принесла его существенная простота. Данный метод основан на теории вероятности, а конкретнее, на Центральной предельной теореме. Таким образом, ошибка вычислений, проводимых на основе этого метода, сильно зависит от количества моделируемых испытаний. Чем больше испытаний мы проводим, тем меньше ошибка, и в то же время больше временные затраты на расчеты. Тем не менее, для некоторых задач, которые требуют высокой точности вычислений, даже достаточно большое количество испытаний, проведенных в рамках метода Монте-Карло, не даст желательной точности.

Есть два важных замечания:

1. Если точки распределены неравномерно, то расчёты выйдут плохими.
2. Расчёты, как правило, неудовлетворительны, если по всей области случайным образом размещено всего несколько точек. В среднем точность улучшается по мере размещения большего количества точек.

Для решения данной задачи в системе Matlab можно воспользоваться следующей М-функцией:

```
PercentUnderCurve = sum(y < fx) / N;
Monte_Integral = PercentUnderCurve * M *(b-a) %Вычисление методом Монте-Карло
Matlab_Integral = integral(f,a,b) %Интеграл матлаба (точное значение)
PercentError = abs(Monta_Integral-Matlab_Integral)/Matlab_Integral*100
%Процент ошибки
```

### Порядок выполнения работы

1. Выбрать функцию в соответствии со своим вариантом и заполнить таблицу.

| Номер варианта | Функция                           | Область значений |
|----------------|-----------------------------------|------------------|
| 1              | $\sin(x)+2$                       | 0; 5             |
| 2              | $\cos(x)+3$                       | 0; 10            |
| 3              | $x^2+1$                           | 1; 10            |
| 4              | $\sqrt{x}+1$                      | 2; 6             |
| 5              | $\sqrt{x}+19$                     | 2; 6             |
| 6              | $(x-4)^2$                         | -1; 1            |
| 7              | $3/(x-1)$                         | 2; 5             |
| 8              | $2x+7$                            | -3; 3            |
| 9              | $\log(2x)$                        | 1; 10            |
| 10             | $\sqrt{(x/7)}+7x$                 | 0; 12            |
| 11             | $2\sin(x)-7$                      | 0.2; 0.4         |
| 12             | $2\operatorname{tg}(x)$           | 0.2; 0.3         |
| 13             | $\exp(2x+1)$                      | -2; 1            |
| 14             | $2\cos(x)+\sin(x)$                | -0.6; 0.8        |
| 15             | $\sin(x)+2\operatorname{tg}(x/2)$ | 0; 0.4           |
| 16             | $\sin(x)\cos(x)$                  | 0; 0.45          |
| 17             | $1+\operatorname{tg}(x)^2$        | -0.4; 0.4        |
| 18             | $\sin(x^2)+\cos(x^2)$             | 0; 0.3           |
| 19             | $2^{(x+4)}$                       | -10; 10          |
| 20             | $3^{(x-3x)}$                      | 10; 20           |
| 21             | $12-2\log(x)$                     | 10; 20           |

| Номер прогона   | Результаты вычислений в соответствии с количеством точек |     |      |      |       |
|-----------------|--|-----|------|------|-------|
|                 | 200  | 500 | 1000 | 5000 | 10000 |
| 1               |  |     |      |      |       |
| 2               |  |     |      |      |       |
| 3               |  |     |      |      |       |
| 4               |  |     |      |      |       |
| 5               |  |     |      |      |       |
| Среднее         |  |     |      |      |       |
| Точное значение |  |     |      |      |       |

### Контрольные вопросы

1. Приведите классификацию методов моделирования по типу модели;
2. Приведите классификацию методов моделирования по типу применяемых математических моделей;
3. Приведите классификацию методов моделирования по этапу построения математической модели;
4. Приведите классификацию методов моделирования по этапу исследования математической модели;
5. В чём заключается суть метода Монте-Карло?
6. В каких областях используется метод Монте-Карло? Какие ограничения существуют для его использования?
7. Укажите преимущества и недостатки метода Монте-Карло;
8. Произведите анализ результатов вычислений, приведите соответствующие выводы.