МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ИНСТИТУТ ТЕХНОЛОГИЙ (ФИЛИАЛ) ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ «ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ» В Г. ВОЛГОДОНСКЕ РОСТОВСКОЙ ОБЛАСТИ

(Институт технологий (филиал) ДГТУ в г. Волгодонске)

УТВЕРЖДАЮ
Директор
И.В. Столяр
«26» апреля 2022 г.

Методические указания
по дисциплине
«Компьютерные технологии в машиностроении»
для обучающихся по направлению подготовки
15.03.05 Конструкторско-технологическое обеспечение машиностроительных производств
профиль Технология машиностроения

2022 года набора

Волгодонск
2022

Лист согласования

Методические указания по дисциплине «Компьютерные технологии в машиностроении» составлены в соответствии с требованиями Федерального государственного образовательного стандарта высшего образования по направлению подготовки (специальности)
15.03.05 Конструкторско-технологическое обеспечение машиностроительных производств

Рассмотрены и одобрены на заседании кафедры «*ТСиИТ*» протокол № 9 от «26» апреля 2022 г.

Оглавление

1. Краткое содержание работы

Цель работы: знакомство студентов с компьютерной программой MATLAB, применяемой при математическом моделировании объектов проектирования, получение навыков работы с данным программным средством путем освоения его основных инструментов.

Практическая работа состоит из четырех частей. При выполнении первой части работы студент изучает простейшие вычисления, вычисление функций, получение решения в виде графиков. При выполнении второй части работы студент изучает операции с векторами и матрицами, а также методы решения системы алгебраических уравнений в матричной форме. При выполнении третьей части работы студент изучает методы численного и аналитического решения систем алгебраических уравнений, нелинейных алгебраических уравнений, численного решения обыкновенных дифференциальных уравнений и систем обыкновенных дифференциальных уравнений первого порядка. При выполнении четвертой части работы студент составляет скриптовый файл MATLAB, реализующий расчет конкретной инженерной или научной задачи по заданной методике.

2. Особенности математического моделирования на основе MATLAB

Исторически MATLAB разрабатывался как диалоговая среда для матричных вычислений (MATrix LABoratory). Со временем пакет был оснащен хорошей графической системой, дополнен средствами компьютерной алгебры от Maple и усилен библиотеками команд (или Toolboxes), предназначенными для эффективной работы со специальными классами задач.

В состав MATLAB входят интерпретатор команд, графическая оболочка, редактор-отладчик, библиотеки команд, компилятор, символьное ядро пакета Maple для проведения аналитических вычислений, математические библиотеки MATLAB на C/C++, генератор отчетов и богатый инструментарий (Toolboxes).

Интерфейс MATLAB многооконный и имеет ряд средств прямого доступа к различным компонентам системы.

Главное меню – посредством этого меню осуществляются наиболее общие действия системы MATLAB. Меню имеет стандартный вид и организацию, присущую многим программным продуктам.

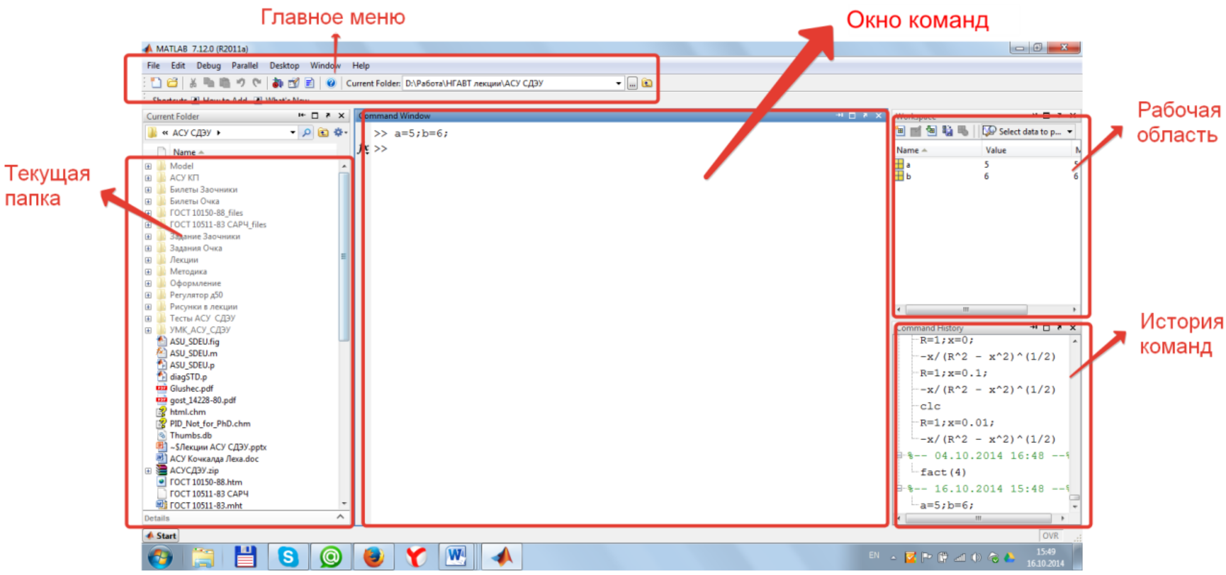

Рис.1. Интерфейс системы MATLAB

Окно команд (Command Window) – наиболее важное окно для пользователя. Посредством этого окна вводятся математические выражения, получают результаты вычислений, а также выдаются сообщения, посылаемые системой. Данное окно становится доступным пользователю сразу же после запуска программы. Математические выражения пишутся в командной строке после знака приглашения >>.

Например:

>>X=2+3

Для получения результата необходимо нажать клавишу «Enter».

Если необходимо изменить одну цифру, то это не получится сделать, установив курсор в уже введенное выражение и провести редактирование. Невозможность редактирования ранее введенной команды простой установкой курсора в нужную строку является одной из особенностей программы MATLAB. Для редактирования ранее введенной команды, необходимо установить курсор в строку ввода и воспользоваться клавишами «↑» и «↓». Эти клавиши позволяют пролистать историю введенных ранее команд и оставить в строке ту команду, которая необходима. Команду можно выполнить сразу (нажав клавишу «Enter») или после редактирования.

Текущая папка (Current Folder) – является аналогом известной программы Проводник, но имеет для MATLAB свое предназначение. Дело в том, что, кроме работы с математическими выражениями из командного окна, пользователь также может работать с файлами. Математические функции, используемые в процессе работы, физически представляют собой файлы, названные по именам функций. В этих файлах записаны программы, реализующие функции. Таким образом, в работе постоянно используются файлы. Например, указывая встроенную функцию, фактически указывается имя файла без расширения, в котором хранится текст программы. Этот файл программа будет искать в текущей папке или в путях, прописываемых отдельно.

Рабочая область (Workspace) – содержит список всех переменных, хранящихся в рабочем пространстве. В процессе работы используются переменные различных типов. Созданные переменные хранятся в специально отведенной области памяти компьютера. Они не исчезают сами по себе, а только при выходе из программы или путем использования специальных команд. При этом значения переменных можно использовать в любом вводимом математическом выражении. С помощью окна «Рабочая область» можно выбрать любую переменную, посмотреть ее содержимое или выполнить другие действия.

История команд (Command History) – содержит список команд, вводимых в командной строке «Окна команд». При необходимости повторить ранее выполненную команду, нужно отыскать в списке «История команд» эту команду и, двойным щелчком левой кнопки мыши, можно выполнить команду. Содержимое этого окна не теряется после выхода из системы и выключения компьютера. Удалить список команд можно только с помощью меню.

Все расчеты в MATLAB выполняются с двойной точностью, а для представления чисел на экране имеются разные форматы. Нужный формат может быть определен в меню (**File/Preferences**) либо при помощи команды **format**. Существуют следующие способы представления чисел (табл. 1).

Переменные в MATLAB не нужно предварительно описывать, указывая их тип. Все данные хранятся в виде массивов: числовые переменные (внутренний тип *numeric*), текстовые строки (*char*), ячейки (*cell*) и структуры (*struct*). Двумерный массив – это матрица, одномерный – вектор, а скаляр – матрица размера 1x1. 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 символам и различает большие и малые буквы.

Таблица 1

Форматы вывода на экран

Формат	Представление
short	Число отображается с 4 цифрами после десятичной точки или в формате short e
short e	Число в экспоненциальной форме с мантиссой из 5 цифр и показателем из 3 цифр
rat	Представление в виде рационального дробного числа
long	Число с 16 десятичными цифрами

long e	Число в экспоненциальной форме с мантиссой из 16 цифр и показателем из 3 цифр
hex	Число в шестнадцатеричной форме

В MATLAB имеется ряд встроенных констант (табл. 2) и специальных символов (табл. 3).

Таблица 2

Зарезервированные имена констант

Имя	Описание
ans	Результат последней операции
i, j	Мнимая единица
pi	Число π
eps	Машинная точность
realmax	Максимальное вещественное число
realmin	Минимальное вещественное число
inf	Бесконечность
NaN	Нечисловая переменная (зарезервировано для результата операций 0/0, 0*inf, inf-inf и т.п.)
end	Наибольшее значение индекса размерности массива

В командном окне в режиме диалога проводятся вычисления. Пользователь вводит команды или запускает на выполнение файлы с текстами на языке MATLAB. Интерпретатор обрабатывает введенное значение и выдает результаты: числовые и строковые данные, предупреждения и сообщения об ошибках.

Таблица 3

Специальные символы

Символ	Назначение
[]	Квадратные скобки используются при задании матриц и векторов
	Пробел служит для разделения элементов матриц
,	Запятая применяется для разделения элементов матриц и оператора в строке ввода
;	Точка с запятой отделяет строки матриц, а точка с запятой в конце оператора (команды) отменяет вывод результата на экран
:	Двоеточие используется для указания диапазона (интервала изменения величины) и в качестве знака групповой операции над элементами матриц
()	Круглые скобки применяются для задания порядка выполнения математических операций, а также для указания аргументов функций и

	индексов матриц
.	Точка отделяет дробную часть числа от целой его части, а также применяется в составе комбинированных знаков (.*, .^, ./, .\)
…	Три точки и более в конце строки отмечают продолжение выражения на следующей строчке
%	Знак процента означает начало комментария
’	Апостроф указывает на символьные строки, а для включения самого апострофа в символьную строку нужно поставить два апострофа подряд

Имена переменных должны начинаться с буквы. Знак «=» соответствует операции присваивания. 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*.

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

После окончания сеанса работы с системой MATLAB все ранее вычисленные переменные теряются. Чтобы сохранить в файле на диске компьютера содержимое рабочего пространства системы MATLAB, нужно выполнить команду меню **File –> Save Workspace As …** .По умолчанию расширение имени файла **mat**, поэтому такие файлы принято называть MAT-файлами.

Система MATLAB работает как с действительными, так и с комплексными числами. Перед использованием операций с комплексными числами необходимо определить переменную i = sqrt(–1) или j = sqrt(–1). В арифметических выражениях применяются следующие знаки операций:

+, - – сложение, вычитание,

*– умножение,

/ – деление слева направо;

\ – деление справа налево;

^ – возведение в степень.

Система MATLAB позволяет вычислять различные математические функции. Следующие элементарные алгебраические функции имеют в качестве аргумента одно или два действительных (x,y) или одно комплексное (z) число (табл. 4).

Таблица 4

Элементарные алгебраические функции

Функция	Описание

abs(z), abs(x),	Вычисление модуля комплексного числа z или абсолютного значения действительного числа x.
angle(z)	Вычисление аргумента z.
sqrt(x)	Вычисление квадратного корня
real(z)	Вычисление действительной части комплексного числа z.
imag(z)	Вычисление мнимой части комплексного числа z.
round(x)	Округление до целого.
fix(x)	Округление до ближайшего целого в сторону нуля.
rem(x, y)	Вычисление остатка от деления x на y.
exp(x)	Вычисление e в степени x.
log(x)	Вычисление натурального логарифма числа x.
log10(x)	Вычисление десятичного логарифма числа x.

Система MATLAB предоставляет возможности для вычисления следующих тригонометрических и обратных тригонометрических функций переменной x (табл.1.5).

Таблица 5

Тригонометрических функций

Функция	Описание
sin(x)	Вычисление синуса
cos(x)	Вычисление косинуса
tan(x)	Вычисление тангенса
asin(x)	Вычисление арксинуса
acos(x)	Вычисление арккосинуса
atan(x)	Вычисление арктангенса
atan2(y, x)	Вычисление арктангенса по координатам точки

Простейшие вычисления, вычисление функций, получение решений в виде графиков

В ходе занятия необходимо выполнить с помощью средств MATLAB ряд заданий, представленных ниже. В скобках даны ответы, которые должны быть выданы программой.

Задание 1. В окне команд задать значения переменных согласно варианту. Необходимо вычислить выражение согласно варианту (табл. 6).

Таблица 6

Варианты выражений для задания 1

№ вар.	Выражение/ Переменные	Ответ

№ вар.	Выражение/ Переменные	Ответ
1	$y = -\dfrac{(x-d)(x^2+b^2)}{\sqrt[3]{x^2+b^2-cd}} + 10^{-3}\operatorname{tg} kn - \dfrac{cos(kx)}{sin(5)}$ k= –1,3; b=0,91; c=0,75;x=2.32; d=1; n=5	-5.6902
2	$y = \operatorname{tg} ik + 10^3 e^{-5} + \sqrt[3]{\dfrac{10^2\lvert xk\rvert}{(a+b)^2} - \dfrac{ax^3-b}{(a+b)^2}}$ a=1,4; b=0.75; i=2,2; k=2;x=–0.25	12.0458
3	$y = \dfrac{ln\lvert kx\rvert}{\sin 7} - \sqrt{\lvert x-a^2\rvert} - \dfrac{10^2 a - b}{\cos kx} + \sqrt[3]{x-a^3} + c^3 x$ a=25; k=–2; x=0.1; c=2,35; b=5	-2.5594e+003 +2.1651e+001i
4	$y = \dfrac{\lvert a^2-b^2\rvert}{\sin kx} + 10^4 \sqrt[5]{\lvert\sin(kx-bc)\rvert} - \dfrac{k^2+\operatorname{tg} 3k}{e^{kx}}$ a=1,7; b=–1.25; k=3; x=2.5;c=–0,3	9.4318e+003
5	$y = \dfrac{\sqrt[3]{\ln x + a^2}}{0.47x^2} - \left\lvert 0.47x^2 - \dfrac{10^4}{7}\cos^2 k\right\rvert - \dfrac{c}{x}$ a=1.3; x=1; k=2; c=–0,83	-243.5631
6	$y = 10^4 \sin^2 i - \dfrac{0.32x^3+4x+b}{\cos ia}\sqrt[6]{0.32x^3-b} + \lvert b\rvert$ i=3; x=2.75; a=–2,5; b=1.35	128.0923
7	$y = -\dfrac{\cos i}{\sin kx} + \dfrac{ax^2+\lvert d\rvert}{(a+b)^2} - 10^4\sqrt[6]{\dfrac{kx}{(a+b)^2}}$ i=2; k=–2; x=–0,8; a=3,5;b=–0.7; d=5	-7.6717e+003
8	$y = \sqrt[5]{\lvert ax^2-b^3\rvert} + \ln kx - \dfrac{e^{kx}+c^2}{\sin kx} - 10^{-3}\sqrt{2157}$ a=1; x=2.5; b=–0,04; k=3; c=5	-1.9508e+003
9	$y = \dfrac{1}{9} - 10^{-4}e^{kx} + \cos\sqrt{x^2+b} + \dfrac{\sqrt{x^2+b}}{0.4x} + \dfrac{\sin 3}{(x^2+b)n}$ k=3; x=0,48; b=–0,31; n=1.72	1.1508 + 1.1786i
10	$y = \cos\big(k(x-a)\big) + 10^{-4}\dfrac{(x+a)^3+x^4 d}{k(x-a)^3} + \dfrac{\sqrt[5]{\lvert x+a\rvert}}{2.4b}$	7.3550e+003

№ вар.	Выражение/ Переменные	Ответ
	k=4; x=8,2; a=–3.25;b=0.05; d=0,95	

Задание 2. Вычислить функцию.

$$y = 4x^2 + 5x + 3 \text{ для } x = 1, 2 \dots 10$$

Ответ: $\begin{pmatrix} 12 \\ 29 \\ 54 \\ 87 \\ 128 \\ 177 \\ 234 \\ 299 \\ 372 \\ 453 \end{pmatrix}$

Задание 3. Для задания 2 получить решение в виде графика.

Порядок работы при построении графика функции следующий.

1. Задать значения аргумента функции.
2. Задать функцию.
3. Построить график.
4. Отформатировать график.
5. Добавить на график дополнительные элементы.

Для построения графиков функций в MATLAB служит команда **plot**, имеющая несколько вариантов записи (*x* – аргумент функции, *y* –функция):

plot(x,y) – строит график одной функции;

plot(x,y,s) – строит график функции с заданным типом и цветом линии и точек (*s* – строковая константа);

plot(x,y1,x,y2,…) – строит графики нескольких функций в одной системе координат;

plot(x,y1,s1,x,y2,s2,…) – строит графики нескольких функций в одной системе координат с заданным типом и цветом линии и точек.

С помощью строковой константы *s* можно изменять цвет линии, представлять узловые точки различными отметками (точка, окружность, крест и т. д.) и менять тип линии графика. Значения строковой константы представлены в табл. 7-9.

Таблица 7

Цвет линии

Код	Описание	Код	Описание
Y	Желтый	G	Зелёный
M	Фиолетовый	B	Синий
C	Голубой	W	Белый
R	Красный	K	Чёрный

Таблица 8

Тип точки

Код	Описание	Код	Описание
.	Точка	D	Ромб
0	Окружность	V	Треугольник
X	Крест	<	Треугольник
+	Плюс	>	Треугольник
*	Звёздочка	P	Пятиугольник
S	Квадрат	H	Шестиугольник

Таблица 9

Тип линии

Код	Описание
-	Сплошная
-.	Штрихпунктир
--	Штриховая

Пример: Построить график функции

$$y = 3x^4 - 7x^3 + 4x^2 - 9x + 2 \text{ для } x = 0, 0.5, 1, 1.5 \ldots 3$$

Команды для построения:

```
>>x=0:0.5:3;
y=3.*x.^4-7.*x.^3+4.*x.^2-9.*x+2;
plot(x,y)
```

Результат выполнения команды показан на рисунке (рис. 2).

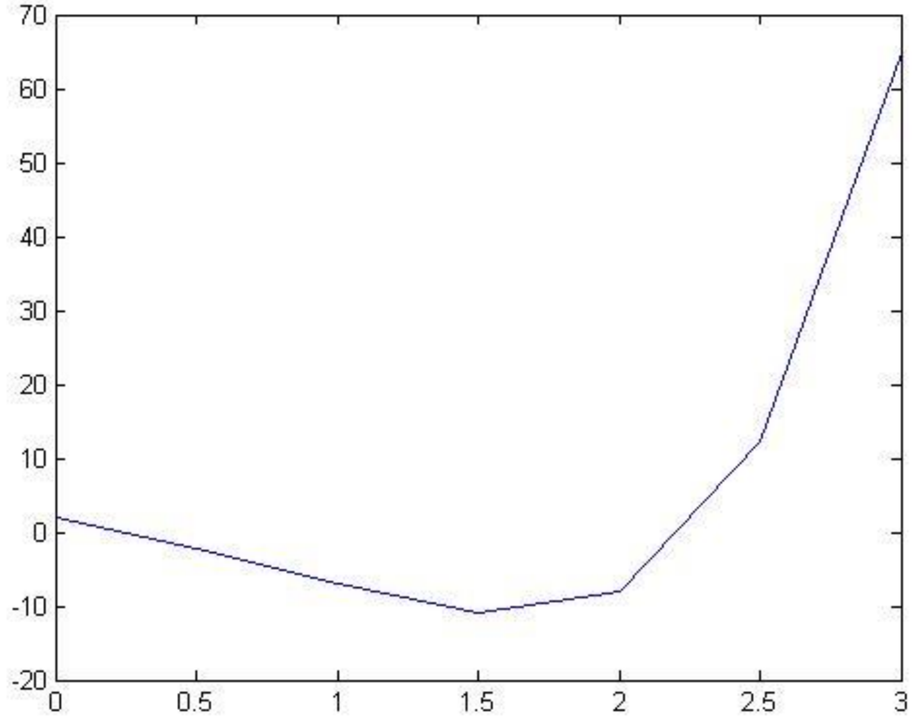

Рис. 2. Результат выполнения команды plot(x,y)

Результат выполнения команды plot(x,y,'d--r') показан на рис. 3.

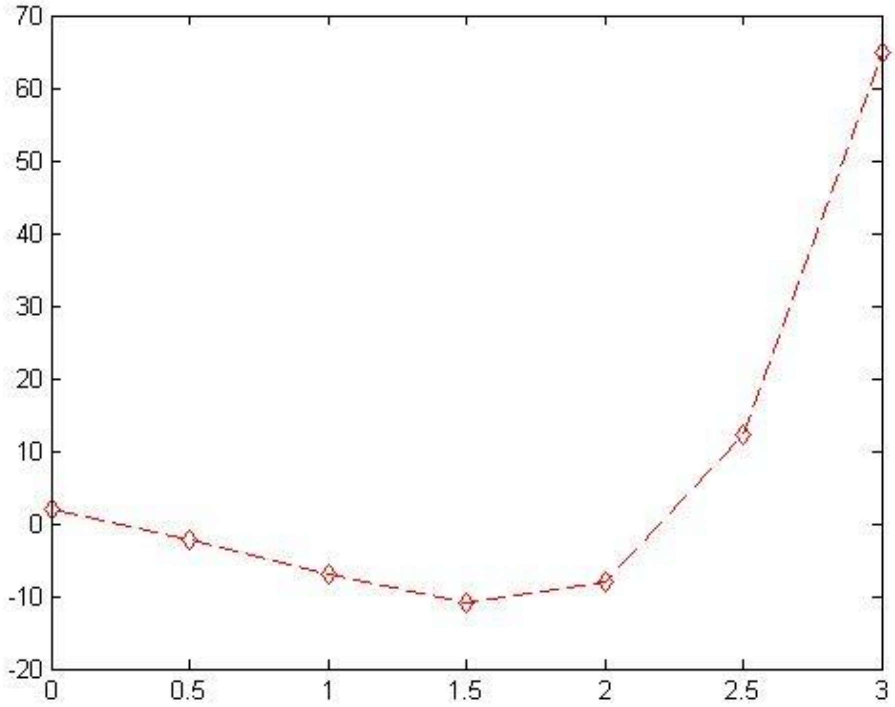

Рис. 3. Результат выполнения команды plot(x,y,'d--r')

Задание 4. Построить на одних координатных осях графики:
$y1 = 4x^2 + 5x + 3$ и $y2 = 3x^4 - 7x^3 + 4x^2 - 9x + 2$ для $x = 1, 2 \dots 10$.
Задание 5. Построить трехмерный график согласно варианту (табл. 10).

Таблица 10

Варианты выражений для задания 5

№ вар.	Значение переменных	Функция	Тип графика
1	от -3 до 3 с шагом 0,2	$z = \dfrac{\sin(x \cdot y)}{x \cdot y}$	mesh
2	от -3 до 3 с шагом 0,2	$z = \dfrac{\cos(x \cdot y)}{x \cdot y}$	surf
3	от -3 до 3 с шагом 0,2	$z = \dfrac{x^2 + y^2}{x \cdot y}$	plot3
4	от -3 до 3 с шагом 0,2	$z = \dfrac{x^3 + y^3}{x \cdot y}$	surf
5	от -3 до 3 с шагом 0,2	$z = x^2 + y^2 + 2$	surf
6	от -3 до 3 с шагом 0,2	$z = x \cdot e^{-x^3 + y^2}$	waterfall
7	от -3 до 3 с шагом 0,2	$z = 5 \cdot x \cdot \sin(y) - 3.5 \cdot y^2$	surf
8	от -3 до 3 с шагом 0,2	$z = x^3 + y^3 + 2$	mesh

9	от -3 до 3 с шагом 0,2	$z = \frac{\cos(x \cdot y)}{x^2 \cdot y^2}$	plot3
10	от -3 до 3 с шагом 0,2	$z = x^2 \cdot e^{-x^3+y^2}$	waterfall

Трехмерные поверхности обычно описываются функцией двух переменных *z(x,y)*. Специфика построения трехмерных графиков требует не просто задания ряда значений x и y, то есть векторов x и y. Она требует определения для X и Y двумерных массивов – матриц.

Для создания таких массивов служит функция **meshgrid**. В основном она используется совместно с функциями построения графиков трехмерных поверхностей. Функция **meshgrid** записывается в следующих формах:

[X,Y,Z] = meshgrid(x,y,z) – возвращает трехмерные массивы, используемые для вычисления функций трех переменных и построения трехмерных графиков;

[X,Y] = meshgrid(x,y) – преобразует область, заданную векторами x и y, в массивы X и Y, которые могут быть использованы для вычисления функции двух переменных и построения трехмерных графиков. Строки выходного массива X являются копиями вектора x, а столбцы Y – копиями вектора y.

Команда **plot3(...)** является аналогом команды **plot(...)**, но относится к функции двух переменных *z(x,y)*. Она строит аксонометрическое изображение трехмерных поверхностей и представлена следующими формами:

plot3(x, y, z) – строит массив точек, представленных векторами x, y и z, соединяя их отрезками прямых. Эта команда имеет ограниченное применение;

plot3(X, Y, Z, S) – обеспечивает построения со спецификацией стиля линий и точек;

plot3(x1, y1, z1, s1, x2, y2, z2, s2,...) – строит на одном рисунке графики нескольких функций *z1(x1,y1), z2(x2,y2)* и т. д. со спецификацией линий и маркеров каждой из них.

Наиболее представительными и наглядными являются сетчатые графики поверхностей с заданной или функциональной окраской. В названии их команд присутствует слово **mesh** (рис. 4). Имеются три группы таких команд:

mesh(X,Y,Z,C) – выводит в графическое окно сетчатую поверхность Z(X,Y) с цветами узлов поверхности, заданными массивом C;

mesh(X, Y, Z) – аналог предшествующей команды при C=Z.

В данном случае используется функциональная окраска, при которой цвет задается высотой поверхности. Функция **mesh** возвращает дескриптор для объекта класса **surface**. Ниже приводится пример применения команды **mesh**:

```
x33=-4:0.2:4;
[X33,Y33]=meshgrid(x33);
z33=(X33.^2+Y33.^2)./(X33.*Y33);
```

mesh(X33,Y33,z33)

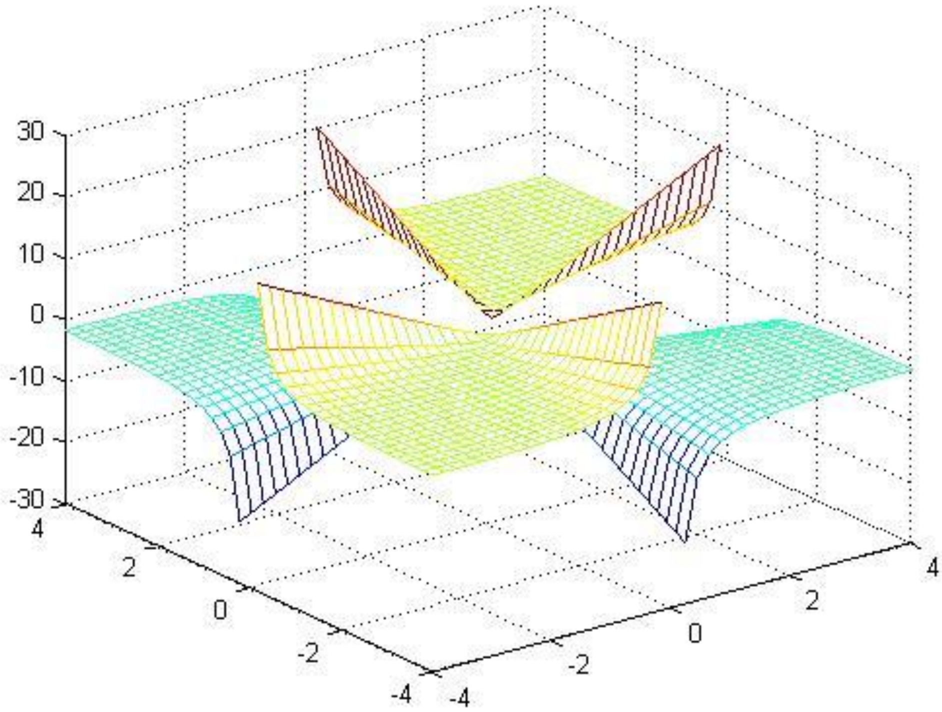

Рис. 4. График поверхности, построенный командой mesh

После того как график уже построен, MATLAB позволяет выполнить его форматирование или оформление в нужном виде. Так, для установки над графиком титульной надписи используется следующая команда **title('string')** – установка на двумерных и трехмерных графиках титульной надписи, заданной строковой константой **'string'**.

Для установки надписей возле осей *x*, *y* и *z* используются следующие команды: **xlabel('String')**, **ylabel('String')**, **zlabel('String')**.

Результат добавления подписей к графику командами: xlabel('Ось X'); ylabel('Ось Y'); zlabel('Ось Z'); title('График') показан на рис. 4.

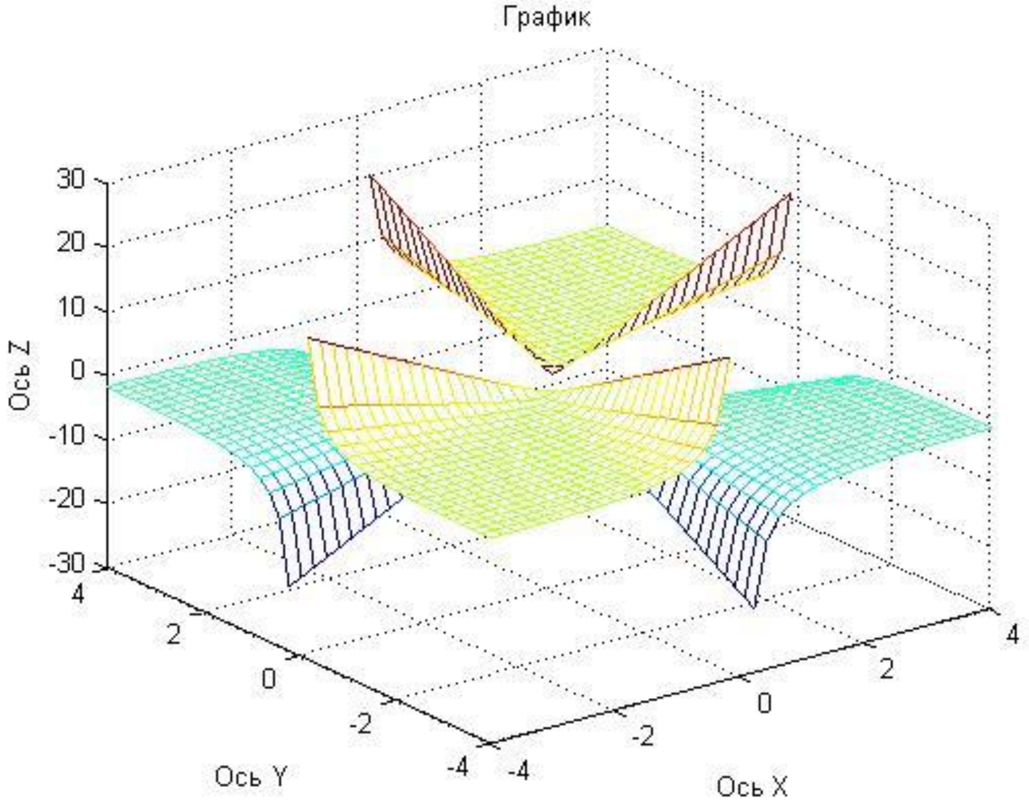

Рис. 5. Добавление надписей на график

Часто возникает необходимость добавления текста в определенное место графика, например для обозначения той, или иной кривой графика. Для этого используется команда **text**:

text(X,Y,'string') – добавляет в двумерный график текст, заданный строковой константой **'string'**, так что начало текста расположено в точке с координатами (X, Y). Если X и Y заданы как одномерные массивы, то надпись помещается во все позиции [x(i), y(i)];

text(X,Y,Z, 'string') – добавляет в трехмерный график текст, заданный строковой константой **'string'**, так что начало текста расположено в позиции, заданной координатами X, Y и Z.

Очень удобный способ ввода текста предоставляет команда **gtext**:

gtext('string') – задает выводимый на график текст в виде строковой константы **'string'** и выводит на график перемещаемый мышью маркер в виде крестика. Установив маркер в нужное место, достаточно щелкнуть любой кнопкой мыши для вывода текста.

Пояснение в виде отрезков линий со справочными надписями, размещаемое внутри графика или около него, называется легендой. Для создания легенды используются различные варианты команды **legend**:

legend(stringl, string2,…, strings) – добавляет к текущему графику легенду в виде строк, указанных в списке параметров;

Пример добавления легенды к графику:

```
>> x=0:0.1:10;y=cos(x);z=sin(x);
>> plot(x,y,x,z)
>> legend('Косинус','Синус')
```

Результат показан на рис. 6.

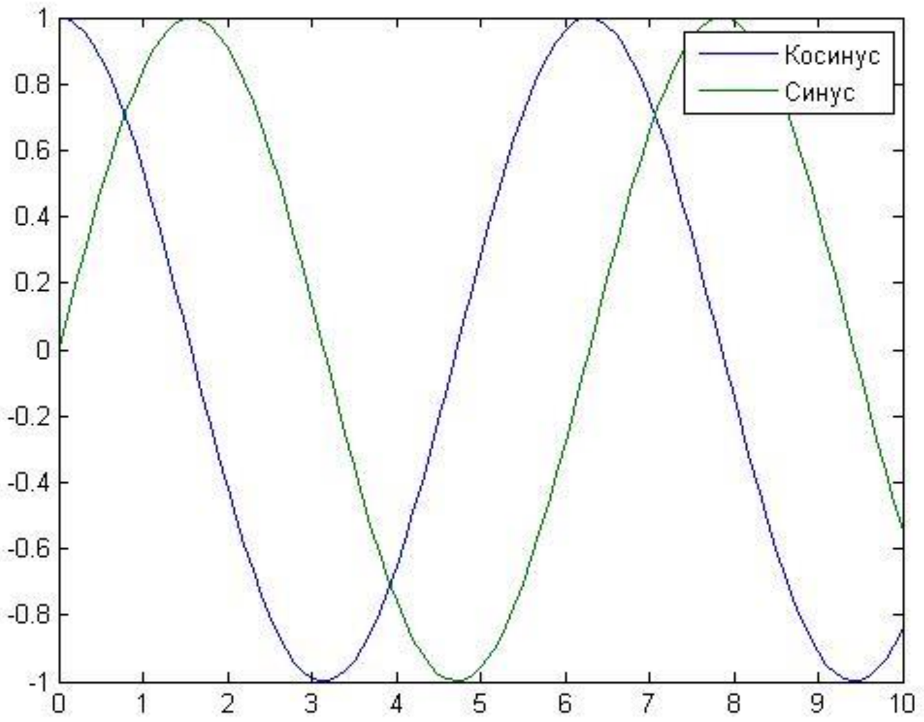

Рис. 6. График с пояснениями

legend (Pos) – помещает легенду в точно определенное место, специфицированное параметром **Pos**:

Pos = 0 – лучшее место, выбираемое автоматически;

Pos = l – верхний правый угол;

Pos = 2 – верхний левый угол;

Pos = 3 – нижний левый угол;

Pos = 4 – нижний правый угол;

Pos = –l – справа от графика.

При добавлении легенды следует учесть, что порядок и количество аргументов команды **legend** должны соответствовать порядку вывода графиков и их количеству

Обычно графики выводятся в режиме автоматического масштабирования. Следующие команды класса **axis** меняют эту ситуацию:

axis([XMIN XMAX YMIN YMAX]) – установка диапазонов координат по осям *x* и *y* для текущего двумерного графика;

axis([XMIN XMAX YMIN YMAX ZMIN ZMAX]) – установка диапазонов координат по осям *x*, *y* и *z* текущего трехмерного графика;

axis auto – установка параметров осей по умолчанию.

В математической, физической и иной литературе при построении графиков в дополнение к разметке осей часто используют масштабную сетку. Команды **grid** позволяют задавать построение сетки или отменять это построение:

grid on – добавляет сетку к текущему графику;

grid off – отключаетт сетку.

Во многих случаях желательно построение многих наложенных друг на друга графиков в одном и том же окне. Для этого служит команда продолжения графических построений **hold**. Она используется в следующих формах:

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

hold off – отменяет режим продолжения графических построений.

Бывает, что в одном окне надо расположить несколько координатных осей с различными графиками без наложения их друг на друга. Для этого используются команды **subplot**, применяемые перед построением графиков:

subplot(m, n, p) – разбивает графическое окно на **m×n** подокон, при этом **m** – число подокон по горизонтали, **n** – число подокон по вертикали, а **p** – номер подокна, в которое будет выводиться текущий график (подокна отсчитываются последовательно по строкам).

Работа функции **subplot** выглядит следующим образом:

```
>> subplot(3,2,1);plot(x,y)
>> subplot(3,2,3);plot(x,z)
>> subplot(3,2,2);plot(x,z,x,y)
>> subplot(3,2,6);mesh(X33,Y33,z33)
```

Результат показан на рис.6. Было сформировано 3 строки и два столбца полей для вывода графиков. Обращение к каждому конкретному полю происходит с указанием его номера. Нумерация происходит слева направо и снизу вверх.

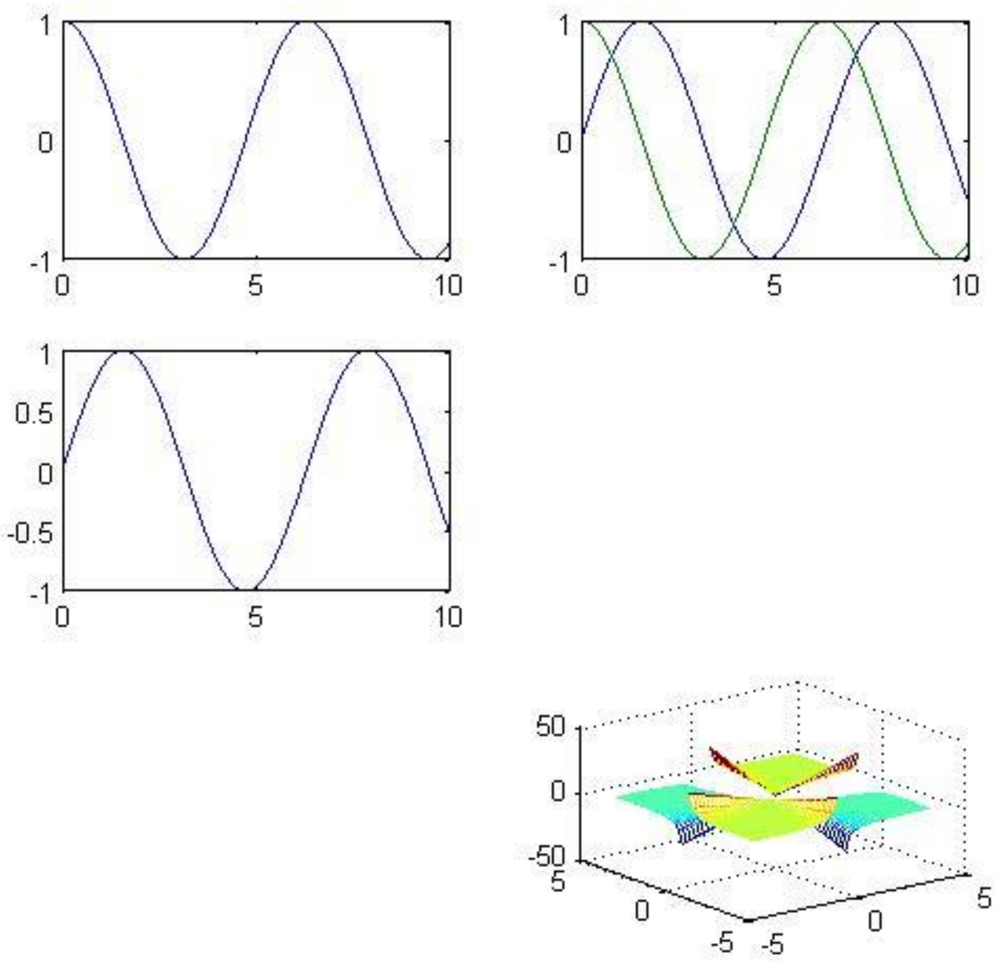

Рис. 7. Работа функции subplot

Задание 6. Используя команду subplot создать фигуру, содержащую три графика.

1. График функции $y = 3x^4 - 7x^3 + 4x^2 - 9x + 2 \text{ для } x = 0, 0.5, 1, 1.5 \ldots 3$. Подписать график «График 1», оси – «Ось X» и «Ось Y» соответственно. Включить отображение сетки графика.

2. График, полученный в задании 4. Подписать график «График 2», оси – «Ось X» и «Ось Y» соответственно. Включить отображение сетки графика.

3. График, полученный в задании 5. Подписать график «График 3», оси – «Ось X», «Ось Y» и «Ось Z» соответственно.

Пример результата выполнения задания 8 показан на рис.8

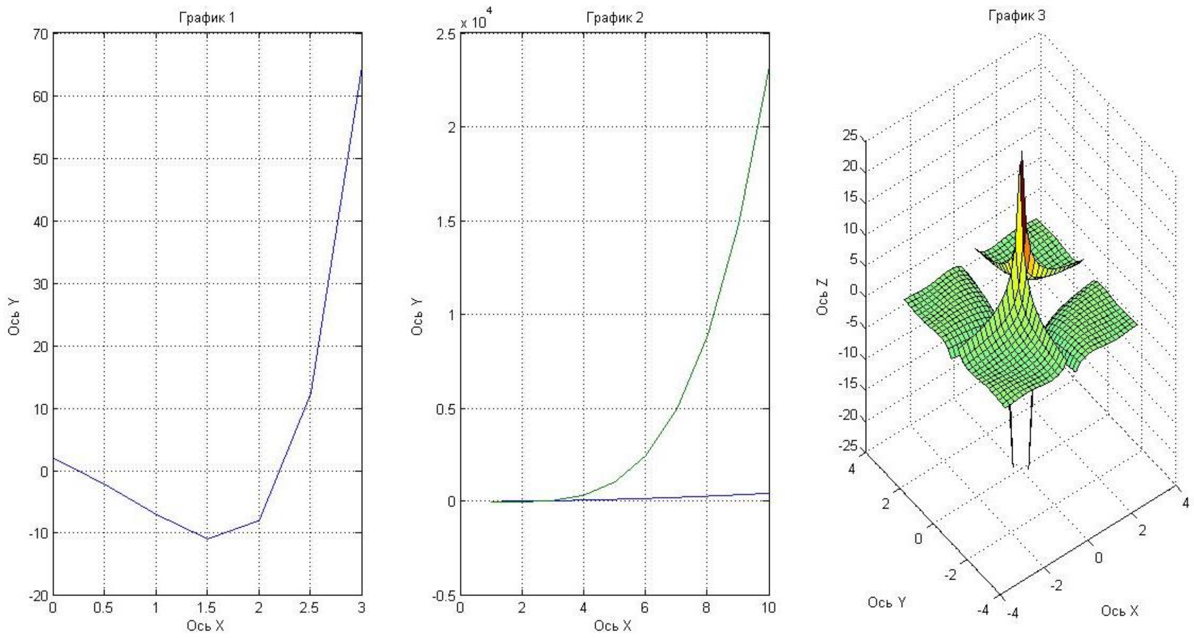

Рис. 8. Пример выполнения задания 8

Операции с векторами и матрицами, методы решения системы алгебраических уравнений в матричной форме

В ходе занятия необходимо выполнить с помощью средств MATLAB ряд заданий, представленных ниже.

По умолчанию все числовые переменные в MATLAB считаются матрицами, так что скалярная величина есть матрица первого порядка, а векторы являются матрицами, состоящими из одного столбца или одной строки. Матрицу можно ввести, задав ее элементы или считав данные из файла, а также в результате обращения к стандартной или написанной пользователем функции.

Матричные данные размещаются в памяти последовательно по столбцам. Элементы матрицы в пределах строки отделяются пробелами или запятыми. Непосредственное задание матрицы можно осуществить несколькими способами. Например, вектор-столбец, то есть матрица, вторая размерность которой равна единице, может быть присвоена переменной A вводом одной строки:

```
>> A=[7+4i; 4; 3.2]          % Ввод вектора-столбца
A =
   7.0000 + 4.0000i
   4.0000
   3.2000
```

или вводом нескольких строк

```
>> A = [                % ввод вектора по строкам
7+4i
4
3.2];
```

Векторы могут быть сформированы как диапазоны – при помощи двоеточий, разделяющих стартовое значение, шаг и предельное значение. Если величина шага отсутствует, то по умолчанию его значение равно единице.

В результате n:m:k будет сформирован вектор, последний элемент которого не больше k для положительного шага m, и не меньше – для отрицательного: [n, n+m,n+m+m,…]

Например:

```
>> a=1:2:5
a =
    1    3    5
```

В табл. 11 представлен некоторый набор функций для создания матриц специального вида.

Таблица 11

Набор функций для создания матриц специального вида

Функция	Описание
eye(m,n)	Единичная матрица размерности $m\times n$
zeros(m,n)	Нулевая матрица размерности $m\times n$
ones(m,n)	Матрица, состоящая из одних единиц размерности $m\times n$
rand(m,n)	Возвращает матрицу случайных чисел равномерно распределенных в диапазоне от 0 до 1, размерность $m\times n$
randn(m, n)	Возвращает матрицу размерности $m\times n$, состоящих из случайных чисел, имеющих гаусовское распределение
tril(A), triu(A)	Выделение нижней треугольной и верхней треугольной частей матрицы *A*
inv(A)	Нахождение обратной матрицы *A*
det(A)	Нахождение определителя (детерминанта) квадратной матрицы *A*

Обращение к элементу матрицы производится по правилу, – в круглых скобках после имени матрицы даются индексы, которые должны быть положительными целыми числами, указывающими номер строки и через запятую, номер столбца. Например, A(2,1) означает элемент из второй строки первого столбца матрицы A.

Для дальнейших примеров введем матрицу 2x2:

```
>> A=[1  2+5*i; 4.6  3]
A =
   1.0000             2.0000 + 5.0000i
```

4.6000 3.0000

Чтобы изменить элемент матрицы, ему нужно присвоить новое значение:

```
>> A(2,2)=10          % Второй элемент второй строки

A =
   1.0000          2.0000 + 5.0000i
   4.6000          10.0000
```

Размер матрицы можно уточнить по команде **size**, а результат команды **size** можно использовать для организации новой матрицы.

Например, нулевая матрица того же порядка, что и матрица **A**, будет сформирована по команде

```
>> A2=zeros(size(A))
A2 =
    0    0
    0    0
```

С помощью двоеточия легко выделить часть матрицы. Например, вектор из первых двух элементов второго столбца матрицы **A** задаётся выражением:

```
>> A(1:2, 2)
ans =
       2.0000 + 5.0000i
       10.0000
```

Двоеточие само по себе означает строку или столбец целиком. Для удаления элемента вектора достаточно присвоить ему пустой массив – пару квадратных скобок []. Чтобы вычеркнуть одну или несколько строк (столбцов) матрицы нужно указать диапазон удаляемых строк (столбцов) для одной размерности и поставить двоеточие для другой размерности. Для нахождения длины вектора можно воспользоваться также командой length.

Набор арифметических операций в MATLAB для работы с матрицами состоит из стандартных операций сложения – вычитания, умножения – деления, операции возведения в степень и дополнены специальными матричными операциями (табл. 2). Если операция применяется к матрицам, размеры которых не согласованы, то будет выведено сообщение об ошибке.

Для поэлементного выполнения операций умножения, деления и возведения в степень применяются комбинированные знаки (точка и знак операции). Например, если за матрицей стоит знак (^), то она возводится в степень, а комбинация (.^) означает возведение в степень каждого элемента матрицы. При умножении (сложении, вычитании, делении) матрицы на число соответствующая операция всегда производится поэлементно.

Таблица 12

Набор арифметических операций для работы с матрицами

Символ	Назначение
+,-	Символы плюс и минус обозначают знак числа или операцию сложения и вычитания матриц, причем матрицы должны быть одной размерности
*	Знак умножения обозначает матричное умножение, для поэлементного умножения матрицы применяется комбинированный знак (.*)
'	Апостроф обозначает операцию транспонирования (вместе с комплексным сопряжением), транспонирование без вычисления сопряжения обозначается при помощи комбинированного знака (.')
/	Левое деление
\	Правое деление
^	Оператор возведения в степень, для поэлементного возведения в степень применяется комбинированный знак (.^)

Проиллюстрируем различие обычного и поэлементного умножений при помощи следующего примера.

Введём матрицу H размера 2x2 и матрицу **D** из единиц той же размерности:

```
>> H=[0 1; 2 3], D=ones(size(H))
H =
    0    1
    2    3
D =
    1    1
    1    1
```

Перемножим матрицы, используя обычное умножение:

```
>> H*D
ans =
    1    1
    5    5
```

Теперь применим поэлементную операцию:

```
>> H.*D
ans =
    0    1
    2    3
```

Система MATLAB имеет ряд функций, предназначенных для обработки данных, заданных в матричной или векторной форме (табл. 13).

Таблица 13

Набор функций, предназначенных для обработки данных, заданных в матричной или векторной форме

Функция	Описание
size(A)	Возвращает массив, состоящий из числа строк и числа столбцов матрицы.
sum(A)	Возвращает сумму всех элементов по столбцу
mean(A)	Возвращает среднее значение столбца матрицы
std(A)	Возвращает среднеквадратическое отклонение столбца матрицы
min(A), max(A)	Возвращает минимум и максимум соответственно, по столбцу матрицы
sort(A)	Сортирует столбец матрицы по возрастанию
prod(A)	Вычисляет произведение всех элементов столбцов

Задание 1. Ввод с клавиатуры векторов и матриц.

Ввести:

– произвольную вектор-строку (v), размерность 2;

– произвольный вектор-столбец (w), размерность 2;

– произвольную матрицу (m), размерности 2×2.

Задание 2. Генерация матриц специального вида.

Создать:

– матрицу с нулевыми элементами ($m0$), размерности 2×2;

– матрицу с единичными элементами ($m1$), размерности 2×2;

– матрицу с элементами, имеющими случайные значения (mr), размерности 2×2;

– матрицу с единичными диагональными элементами (me), размерности 2×2.

Задание 3. Вычислить матрицу М по формуле согласно варианту (табл. 14).

Таблица 14

Варианты выражений для задания 3

№ варианта	Задание
1	$M=v*w+m+mr*me$
2	$M=m+mr*me$
3	$M=(v/m)*(mr+me)$
4	$M=w*v+mr*me$

5	*M=m*mr+me*
6	*M=m.*mr+100*
7	*M=v*w+mr–m*
8	*M=m+mr*me–10*
9	*M=m*w+mr*v'*
10	*M=m'+mr*me*

Задание 4. Изучение функций обработки данных:

– определение числа строк и столбцов матрицы M;

– определение максимального элемента матрицы M;

– определение минимального элемента матрицы M;

– суммирование элементов матрицы M;

– перемножение элементов матрицы M.

Задание 5. Решение системы алгебраических уравнений в матричной форме

Система линейных алгебраических уравнений в матричной форме имеет вид:

$$\mathrm{A} \cdot \mathrm{X} = \mathrm{B}$$

где: A – квадратная матрица коэффициентов; X – вектор-столбец неизвестных; B – вектор-столбец правых частей.

Решение системы в матричной форме имеет вид:

$$\mathrm{X} = \mathrm{A}^{-1} \cdot \mathrm{B}$$

Решим в матричной форме систему:

$$\begin{cases} 11\mathrm{x} + 12\mathrm{y} + 31\mathrm{z} = 9 \\ 4\mathrm{x} + 52\mathrm{y} + 69\mathrm{z} = 8 \\ 7\mathrm{x} + 86\mathrm{y} + 93\mathrm{z} = 7 \end{cases}$$

Задание 6. Написать m-файл, вычисляющую матрицу M, указанную в задании 3.

Методы численного и аналитического решения систем алгебраических уравнений, нелинейных алгебраических уравнений, обыкновенных дифференциальных уравнений и систем обыкновенных дифференциальных уравнений первого порядка

Задание 1. Вычислить все корни полиномов.

$$5x^5 + 6x^3 + 8x^2 + 2x = 0 \quad \begin{pmatrix} 0 \\ 0.4756 + 1.2912\mathrm{i} \\ 0.4756 - 1.2912\mathrm{i} \\ -0.5977 \\ -0.3534 \end{pmatrix};$$

$$5x^4 + 8x^3 + 3x^2 + 9x + 8 = 0 \begin{pmatrix} 0.4021 \ + \ 1.0171\mathrm{i} \\ 0.4021 \ - \ 1.0171\mathrm{i} \\ -1.5301 \\ -0.8742 \end{pmatrix};$$

$$4x^4 + 8x - 3 = 0 \begin{pmatrix} -1.3660 \\ 0.5000 \ + \ 1.1180\mathrm{i} \\ 0.5000 \ - \ 1.1180\mathrm{i} \\ 0.3660 \end{pmatrix}.$$

Пример. Вычисление всех корней полинома.

Полином в MATLAB задается вектором его коэффициентов, например, для определения полинома p = $x^7+3.2x^5–5.2x^4+0.5x^2+x–3$ следует использовать команду

>> p = [1 0 3.2 -5.2 0 0.5 1 -3];

Число элементов вектора, т. е. число коэффициентов полинома, всегда на единицу больше его степени, нулевые коэффициенты должны содержаться в векторе.

Функция poiyval предназначена для вычисления значения полинома от некоторого аргумента:

>> poiyval(p, 1)

ans =

-2.5000

Аргумент может быть матрицей или вектором, в этом случае производится поэлементное вычисление значений полинома и результат представляет матрицу или вектор того же размера, что и аргумент.

Нахождение всех корней полиномов осуществляется при помощи функции roots, в качестве аргумента которой указывается вектор с коэффициентами полинома. Функция roots возвращает вектор корней полинома, в том числе и комплексных:

>> r = roots(p)

r =

-0.5668 + 2.0698i

-0.5668 - 2.0698 i

1.2149

0.5898 + 0,6435 i

0.5898 - 0.6435i

-0.6305 + 0.5534i

-0.6305 - 0.5534i

Известно, что число всех корней полинома совпадает с его степенью. Убедитесь в правильности работы roots, вычислив значение полинома от вектора его корней:

```
» polyval(p,г)
ans =
1.0e-012 *
-0.1008 + 0.0899 i
-0.1008 - 0.0899 i
-0.0666
0.0027 - 0.0018 i
0.0027 + 0.0018i
0.0102 - 0.0053i
0.0102 + 0,0053i
```

В верхней строке результата содержится общий множитель 1.0e-012, на который следует умножить каждую компоненту получившегося вектора.

Задание 2. Решить тремя способами систему уравнений:

$$\begin{cases} 5x + 6y - 9z + 2v - 7w = 90 \\ 3x - 4y + 5z - 3v + 4w = 12 \\ 9x + y + 3z - 2v + 9w = 51 \\ 7x + 2y - 8z + v + 10w = 32 \\ 6x + 5y - 4z + 3v - 2w = 87 \end{cases} \begin{pmatrix} 12.455 \\ -2.963 \\ 0.356 \\ 6.01 \\ -5.242 \end{pmatrix}$$

Пример. Решение небольшой системы, состоящей из трех уравнений с тремя неизвестными

$$\begin{cases} 3x + 8y - 9z = 12 \\ 5x - 9y + 2z = 34. \\ 8x - 6y + 5z = 98 \end{cases}$$

Ввести матрицу системы в массив A, для вектора правой части используйте массив b. Решить систему при помощи символа «\»:

```
>> A=[3 8 -9;5 -9 2;8 -6 5];
>> b=[12;34;98];
>> x=A\b
x =
   11.4572
    3.9126
    5.9636
```

Проверить, правильный ли получился ответ, умножив A на x.

Алгоритм решения систем линейных уравнений при помощи оператора «\» определяется структурой матрицы коэффициентов системы. В частности, MATLAB исследует, является ли матрица треугольной, или может быть приведена перестановками строк и столбцов к треугольному виду, симметричная матрица или нет, квадратная или прямоугольная (MATLAB умеет решать системы с прямоугольными матрицами – переопределенные или неопределённые). Поэтому решать системы при помощи «\» разумно, когда выбор алгоритма решения поручается MATLAB. Если же имеется информация о свойствах матрицы системы, то следует использовать специальные методы.

Решение систем при помощи функции linsolve.

При использовании знака обратной косой черты для решения системы линейных уравнений выбор метода решения остается за MATLAB. Более широкие возможности предоставляет функция linsolve, которая, так же как и знак обратной косой черты, позволяет решать систему или несколько систем с одной и той же необязательно квадратной матрицей AX = B, и при этом допускает выбор метода ее решения в соответствии со свойствами матрицы, указанными пользователем. В простейшем случае обращение к функции linsoive имеет вид:

X = linsoive(A, b)

Применительно к введенной выше системе уравнений, решение будет иметь вид:

```
>> linsolve(A,b)
ans =
  11.4572
   3.9126
   5.9636
```

Решение системы уравнений в аналитическом (символьном) виде.

Пример. Решить аналитически систему уравнений:

$$\begin{cases} a1 \cdot x + b1 \cdot y + c1 \cdot z = d1 \\ a2 \cdot x + b2 \cdot y + c2 \cdot z = d2 \\ a3 \cdot x + b2 \cdot y + c2 \cdot z = d3 \end{cases}$$

Решение:

```
syms x y z
[x,y,z]=solve('a1*x+b1*y+c1*z-d1=0','a2*x+b2*y+c2*z-
d2=0','a3*x+b2*y+c2*z-d3=0');
simplify([x;y;z])
```

Задание 3. Решить аналитически систему уравнений:

$$\begin{cases} 3 \cdot x1 + 4 \cdot x2 + 4 \cdot x3 + 6 \cdot x4 = 21 \\ 5 \cdot x1 + 7 \cdot x2 + 8 \cdot x3 - 7 \cdot x4 = 34 \\ 9 \cdot x1 + 12 \cdot x2 + 9 \cdot x3 + 8 \cdot x4 = 41 \\ 13 \cdot x1 + 3 \cdot x2 + 19 \cdot x3 + 18 \cdot x4 = 141 \end{cases} \begin{pmatrix} 6799/2807 \\ -8606/2807 \\ 16328/2807 \\ 1277/2807 \end{pmatrix}$$

Аналитическое (символьное) решение нелинейных алгебраических уравнений.

Пример. Решить аналитически уравнение $a \cdot x^2 + b \cdot x + c = 0$.

Решение:
```
>> syms x
>> f=sym('a*x^2+b*x+c');
>> solve(f,x)
 ans =
  -(b + (b^2 - 4*a*c)^(1/2))/(2*a)
 -(b - (b^2 - 4*a*c)^(1/2))/(2*a)
 >> pretty(solve(f,x))
```
Задание 4. Решить уравнения в символьном виде:

1. $x^2 + x + 1 = 0 \begin{pmatrix} -\frac{1}{2} + \frac{\sqrt{3}\cdot i}{2} \\ -\frac{1}{2} - \frac{\sqrt{3}\cdot i}{2} \end{pmatrix}$.

2. $a \cdot x^4 + b \cdot x^2 + d = 0$.

Задание 5. Решить нелинейное дифференциальное уравнение второго порядка с нулевыми начальными условиями и построить график решения:

$$100y`` + 10y`^2 + 101y = 50\left(\frac{\sin x}{4}\right).$$

Ответ представлен на рис. 9.

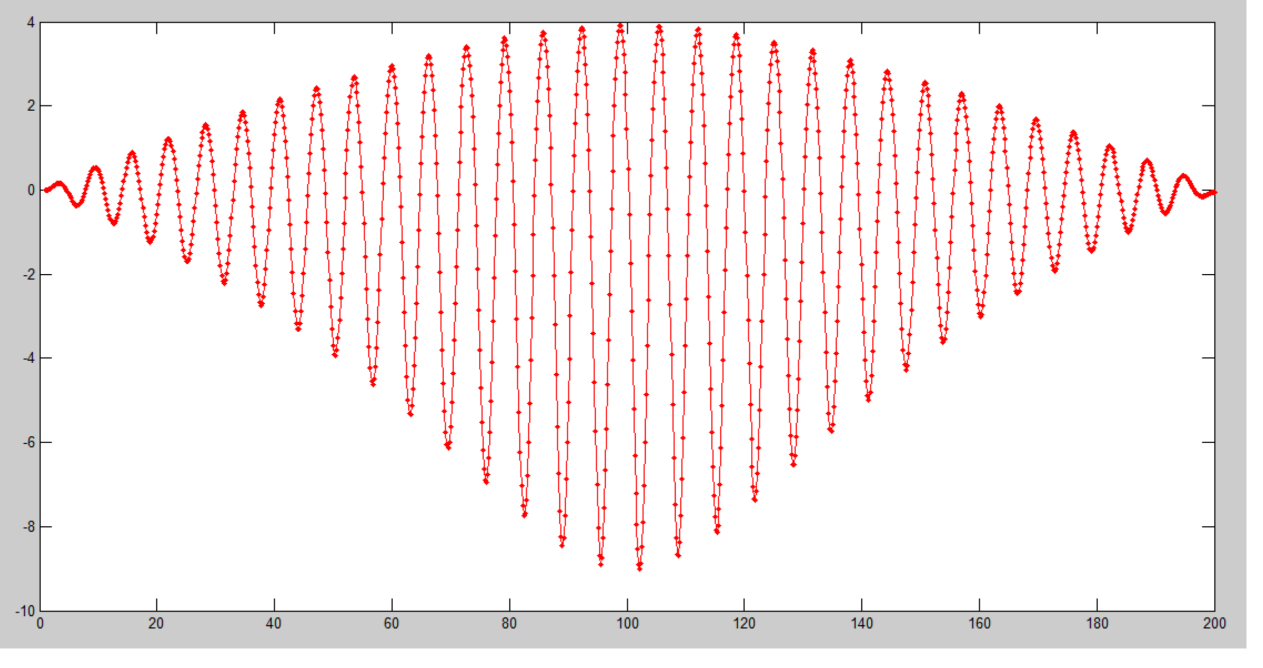

Рис. 9. График решения дифференциального уравнения

Пример. Решение дифференциальных уравнений.

Для решения задачи Коши в MATLAB существует семь солверов: ode45, ode23, ode113, ode15c, ode23s, ode23t и ode23tb. Методика их использования одинакова, включая способы задания входных и выходных аргументов. В общем случае вызов солвера для решения задачи Коши происходит следующим образом (под solver понимается один из перечисленных выше солверов):

$$[T, Y] = solver(odefun, interval, Y0, options)$$

где *odefun* – функция для вычисления вектор-функции правой части системы уравнений, *interval* – массив из двух чисел, задающих промежуток для решения уравнения, *Y0* – заданный вектор начальных значений искомой вектор-функции, *options* – структура для управления параметрами и ходом вычислительного процесса. Солвер возвращает массив *T* с координатами узлов сетки, в которых найдено решение, и матрицу решений *Y*, каждый столбец которой является значением компоненты вектор-функции решения в узлах сетки.

Задача Коши для дифференциального уравнения состоит в нахождении функции, удовлетворяющей дифференциальному уравнению произвольного порядка

$$y^{(n)} = f(t, y, y', \ldots, y^{(n-1)})$$

и начальным условиям при $t = t_0$

$$y(t_0) = u_0, y'(t_0) = u_1, \ldots, y^{(n-1)}(t_0) = u_{(n-1)}$$

Схема решения таких задач в MATLAB состоит из нескольких этапов:

1) приведение дифференциального уравнения к системе дифференциальных уравнений первого порядка (если изначально задана система, то в этом нет необходимости);

2) написание специальной функции для системы уравнений;

3) вызов подходящего солвера;

4) визуализация результатов.

Перемещение точки в среде описывается уравнением:

$$y'' + k^2 y = 0$$

Предположим, что координата точки в начальный момент равна – 0.1 м, а скорость – 0 м/с. Тогда соответствующие начальные условия имеют вид:

$$y(0) = -0.1, \;\; y'(0) = 0$$

Теперь задачу необходимо привести к системе дифференциальных уравнений. Для этого вводят столько вспомогательных функций, каков порядок уравнения. В данном случае необходимо две вспомогательных функции y_1 и y_2, определяемые формулами:

$$y_1 = y, \;\; y_2 = y'$$

Система дифференциальных уравнений с начальными условиями, требуемая для дальнейшей работы, такова:

$$\begin{cases} y_1' = y_2; \\ y_2' = -k^2\, y. \end{cases}$$

$$\begin{cases} y_1' = y_2; \\ y_2' = -10\, y; \end{cases} \quad \begin{bmatrix} y_1(0) \\ y_2(0) \end{bmatrix} = \begin{bmatrix} -0.1 \\ 0 \end{bmatrix}.$$

Второй этап состоит в написании функции для системы дифференциальных уравнений. Функция должна иметь два входных аргумента: переменную *t*, по которой проводится дифференцирование, и вектор, размер которого равен числу неизвестных функций системы. Число и порядок аргументов фиксированы, даже если *t* явно не входит в систему. Выходным аргументом функции является вектор правой части системы. Текст функции *prav* приведен ниже (текст 1).

Текст 1. Текст функции *prav*

```
function solv_01
 %формирование вектора начальных условий
Y0=[-0.1;0];
%вызов солвера от функции prav, начального и конечного момента
%времени и вектора начальных условий
[T,Y]=ode45(@prav,[0 10],Y0);
%вывод графика решения исходного дифференциального уравнения
plot(T,Y(:,1))
%вывод пояснений на график
title('Решение    дифференциального    уравнения    y^{"}=   -10
y','FontName','Arial Unicode MS')
xlabel('Время, сек', 'FontName','Arial Unicode MS')
ylabel('Амплитуда, м', 'FontName','Arial Unicode MS')
grid on
 %подфункция вычисления правых частей уравнения
function F = prav(t,y)
F=[y(2);-10*y(1)+0*t];
```

Выходными аргументами функции являются два вектора: вектор, содержащий значения времени, и матрица значений искомых функций в соответствующие моменты времени. Значения функций расположены по столбцам матрицы, в первом столбце – значения первой функции, во втором – второй и т.д.. В силу проделанных замен $y_1 = y$, $y_2 = y'$, первый столбец матрицы содержит как раз значения неизвестной функции *y(t)*, входящей в исходное дифференциальное уравнение, а второй столбец – значение ее производной. Как правило, размеры матрицы и вектора достаточно велики, поэтому результат лучше сразу отображать на графике.

На рис. 10 представлен график решения полученного функцией solv_01.

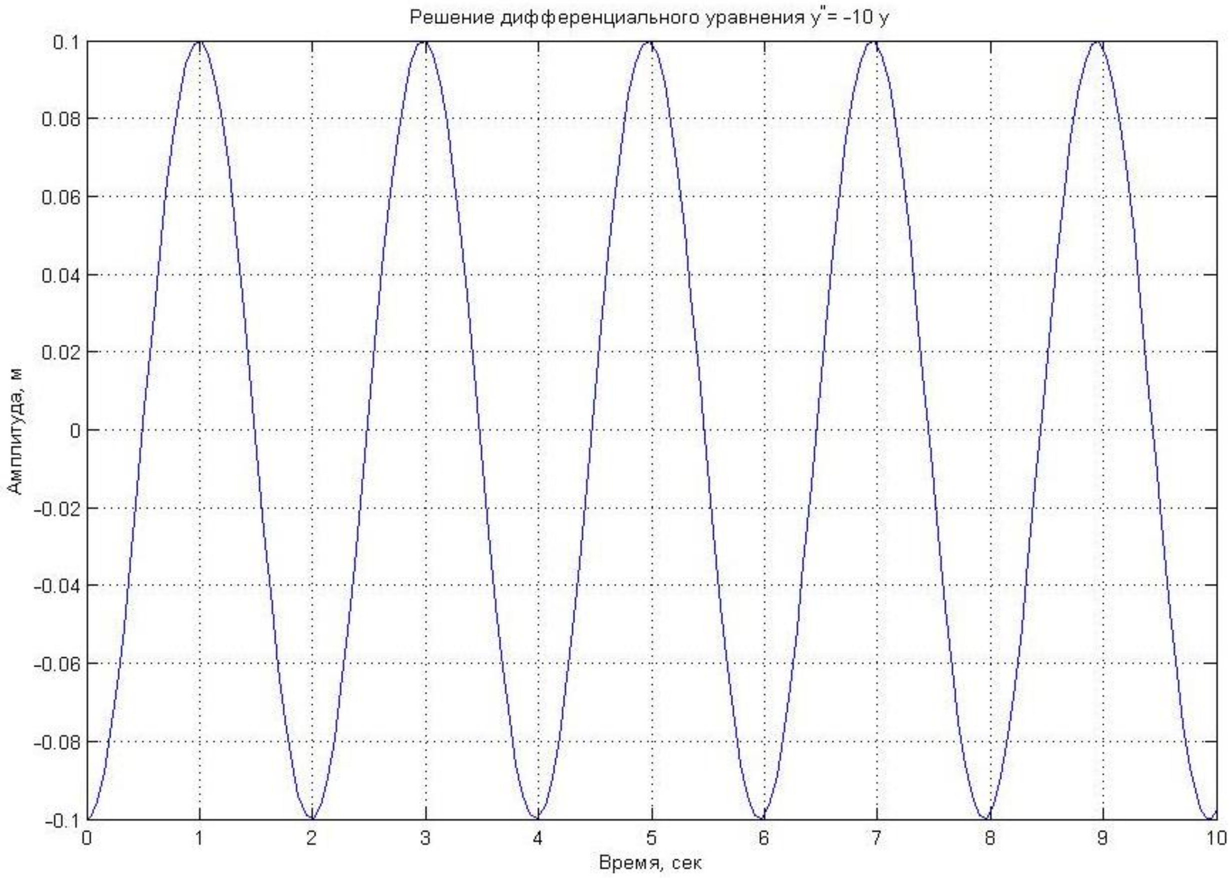

Рис. 10. График решения дифференциального уравнения

Задание 6. Решить дифференциальные уравнения и построить график решения.

$y`` + 5y` + 10y = 5x, y`(0) = 0, y(0) = 0$ (ответ изображен на рис. 3).

$y``` + 50y`` + y`y + 9y = 0, y``(0) = 2, y`(0) = 0, y(0) = 8$ (ответ изображен на рис. 11).

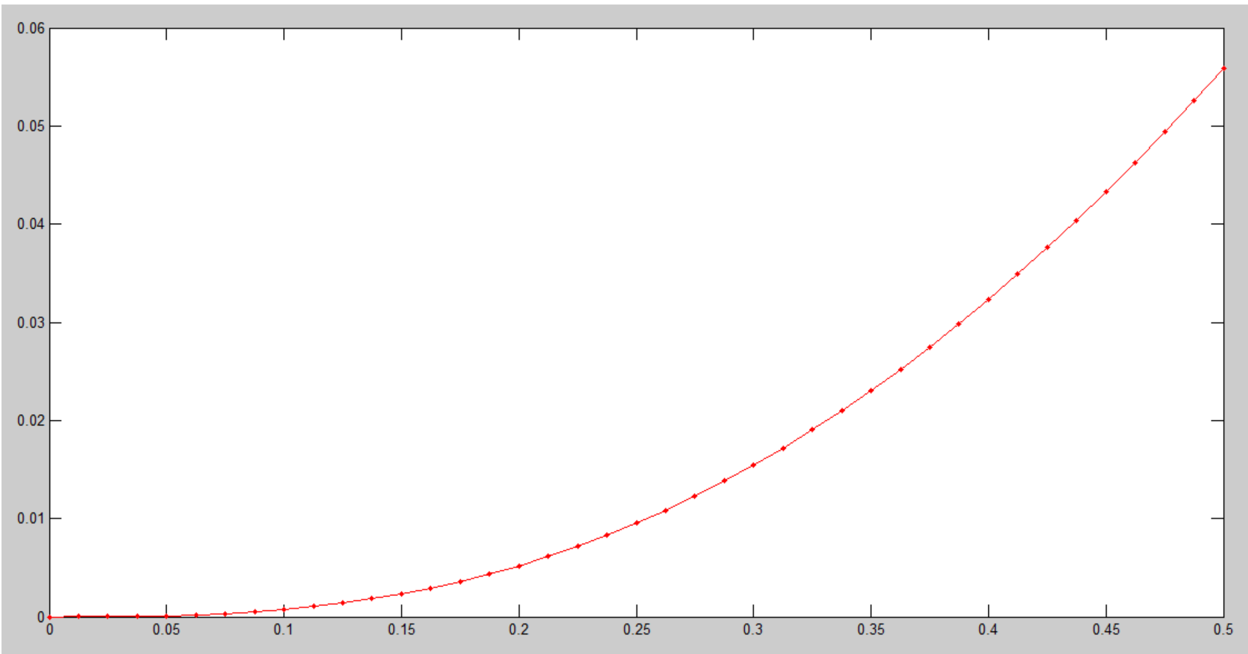

Рис. 3. Результат оформления графика решения

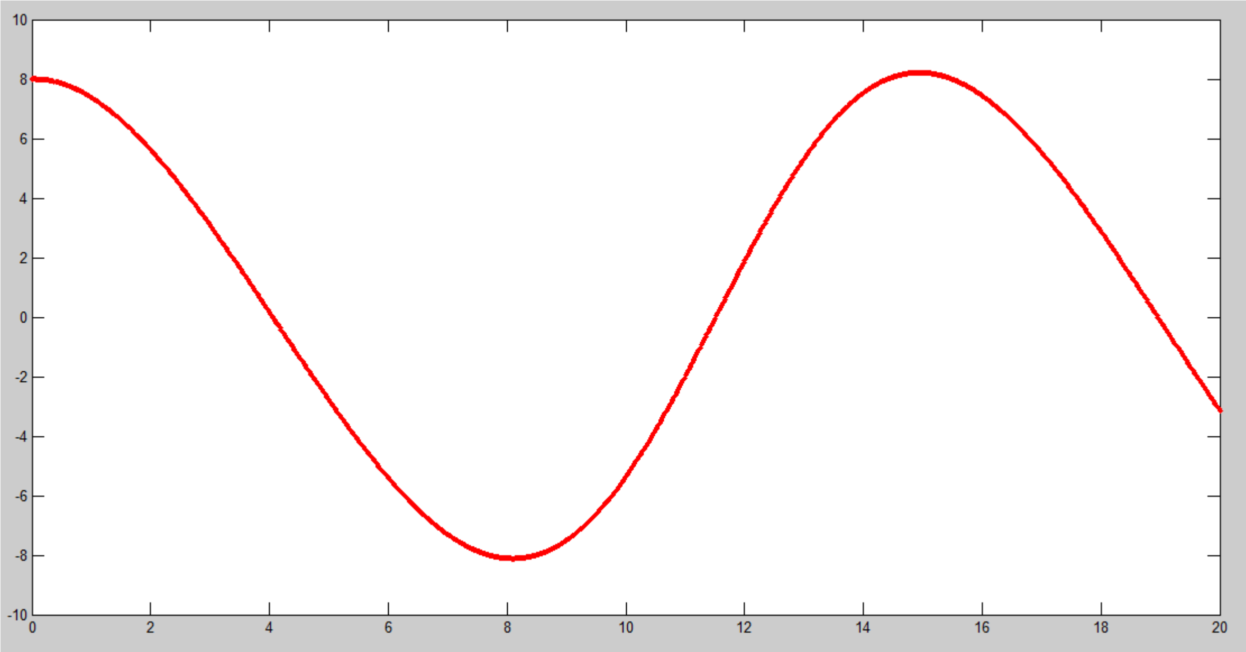

Рис. 11. Результат оформления графика решения

Составление скриптового файла MATLAB, реализующего расчет конкретной инженерной или научной задачи по заданной методике

Задание 1. По приведенным ниже указаниям составить скриптовый файл MATLAB, реализующий расчет по заданной методике.

Скриптовый файл должен содержать все необходимые пояснения.

Таблица 15

Варианты исходных данных

№ варианта	1	2	3	4	5	6	7	8	9	10
N_e, л. с.	150	150	300	100	300	200	75	50	1000	2000
n, об/мин	750	1500	500	2000	700	1000	1500	2000	300	150

Последовательность расчета.

1.1. Построение номинальной характеристики эффективной мощности дизеля.

Для определения динамических параметров дизеля необходимо построить статические совмещенные характеристики комплекса «дизель – гребной винт». Построение начинают с номинальной характеристики эффективной мощности дизеля, которая соответствует номинальному положению рейки топливного насоса hн и неизменной топливоподаче.

Расчет точек характеристики производится по эмпирической формуле

$$N_e(\omega) = N_{ен}\left(0.5\frac{\omega}{\omega_н} + 1.5\left(\frac{\omega}{\omega_н}\right)^2 - \left(\frac{\omega}{\omega_н}\right)^3\right), \quad (1)$$

где $N_{ен}$ – номинальная эффективная мощность дизеля, кВт;

$\omega_н$ – номинальная угловая скорость вращения вала дизеля, рад/с.

При расчетах используем следующие соотношения:

1 кВт = 1.36 л. с.;

$$\omega_н = \frac{\pi \cdot n}{30},$$

где n – частота вращения в об/мин.

Для расчета характеристики изменяем частоту ω в пределах от $0,2 \cdot \omega_н$ до $1,12 \cdot \omega_н$ с шагом около $0,05 \cdot \omega_н$.

Результат расчета представлен на рис. 12.

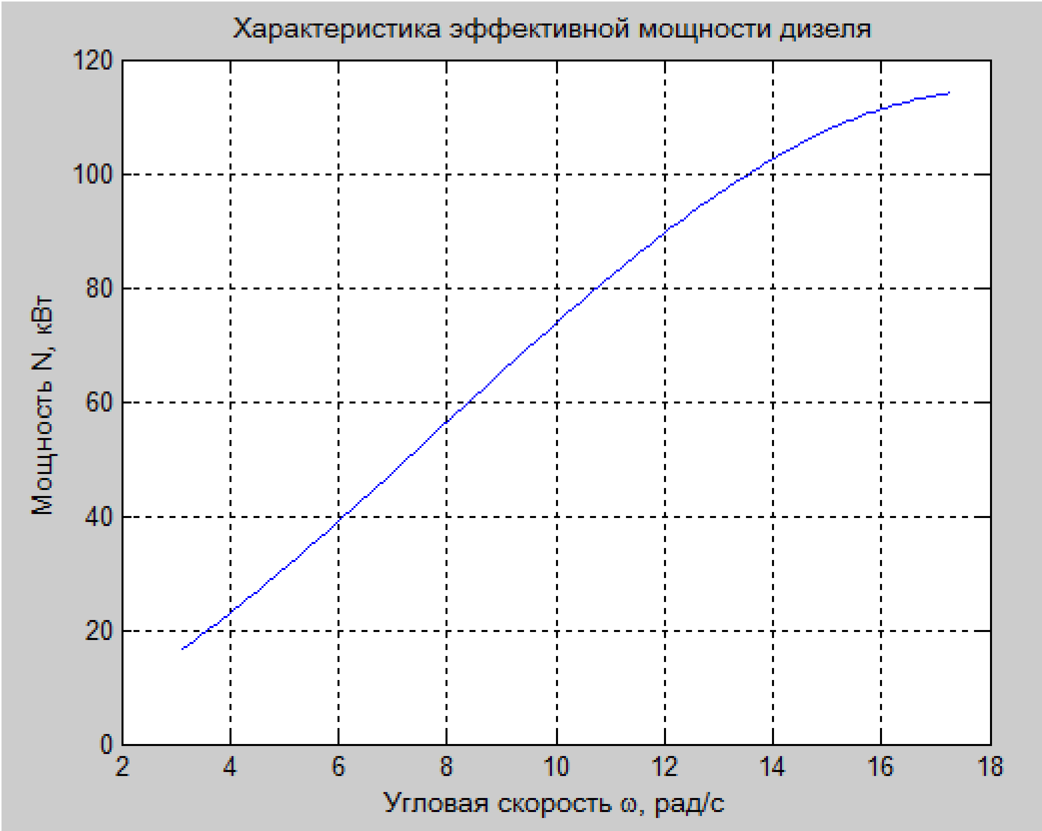

Рис. 12. Результат расчета характеристики эффективной мощности дизеля

1.2. Построение винтовой характеристики.

Зависимость мощности сопротивления винта от его угловой скорости вращения называется винтовой характеристикой.

Винтовая характеристика с достаточной точностью описывается зависимостью

$$N_c(\omega) = K_N \omega^3, \qquad (2)$$

где $N_c(\omega)$ – мощность сопротивления винта вращению, кВт;

K_N – коэффициент нагрузки.

Для построения характеристики, соответствующей нормальному для данного дизеля винту, прежде всего, определяем коэффициент нагрузки нормального винта, приняв $N_с = N_{ен}$, $\omega = \omega_н$.

Затем вычисляем точки винтовой характеристики мощности сопротивления нормального винта по формуле (2).

Строим график мощности сопротивления нормального винта, который совмещается с графиком номинальной характеристики эффективной мощности дизеля (рис. 13).

Точка, в которой пересекаются характеристика эффективной мощности и винтовая характеристика, соответствует номинальному режиму работы дизеля при номинальном положении рейки топливного насоса $h = h_н$.

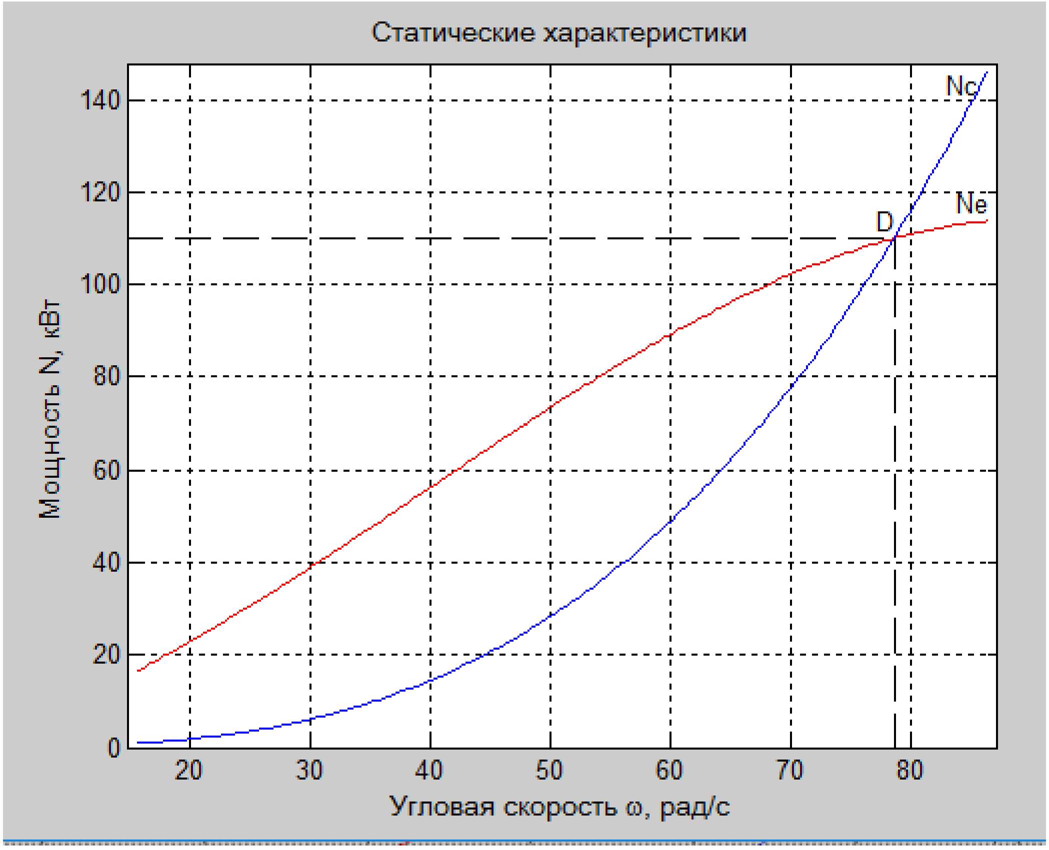

Рис. 13. Результат расчета винтовой характеристики

1.3. Построение характеристик частичной эффективной мощности.

Характеристики частичной эффективной мощности характеризуют работу дизеля при частичной топливоподаче. Для построения характеристик частичной эффективной мощности необходимо определить установившуюся частоту вращения вала дизеля при положении рейки топливного насоса $0{,}75 \cdot h_н$, $0{,}5 \cdot h_н$, $0{,}25 \cdot h_н$.

Для определения установившихся скоростей вращения вала решаем уравнение (2) для условий $N_c(w) = 0{,}75 \cdot N_{ен}$, $N_c(w) = 0{,}5 \cdot N_{ен}$, $N_c(w) = 0{,}25 \cdot N_{ен}$.

Для каждого установившегося режима работы с частичной мощностью дизеля при перемещениях рейки ТНВД $0{,}75 \cdot h_н$, $0{,}5 \cdot h_н$, $0{,}25 \cdot h_н$ строим частичную характеристику эффективной мощности.

Расчет точек характеристики частичной мощности производится по следующей формуле:

$$N_{ei}(\omega) = N_{epi}\left(0{,}5\frac{\omega}{\omega_{pi}} + 1{,}5 \cdot \left(\frac{\omega}{\omega_{pi}}\right)^2 - \left(\frac{\omega}{\omega_{pi}}\right)^3\right), \quad (3)$$

где N_{epi}, ω_{pi} – координаты рабочей точки *i*-го режима работы дизеля.

По результатам расчета строим графики статических характеристик дизеля. По горизонтальной оси графика отложена угловая скорость ω вращения вала дизеля в рад/с, по вертикальной – мощность в кВт. Кривая N_e характеризует зависимость эффективной мощности на валу дизеля от угловой скорости вращения его вала при неизменной топливоподаче, определяемой номинальным положением рейки топливного насоса. Кривая N_c характеризует зависимость мощности сопротивления вращению нормального винта при номинальном значении относительной поступи.

Кривые $0{,}75 \cdot N_{ен}$, $0{,}5 \cdot N_{ен}$, $0{,}25 \cdot N_{ен}$ являются графиками частичных характеристик эффективной мощности при определенных положениях рейки топливного насоса, а именно $0{,}75 \cdot h_н$, $0{,}5 \cdot h_н$, $0{,}25 \cdot h_н$ соответственно (рис. 14).

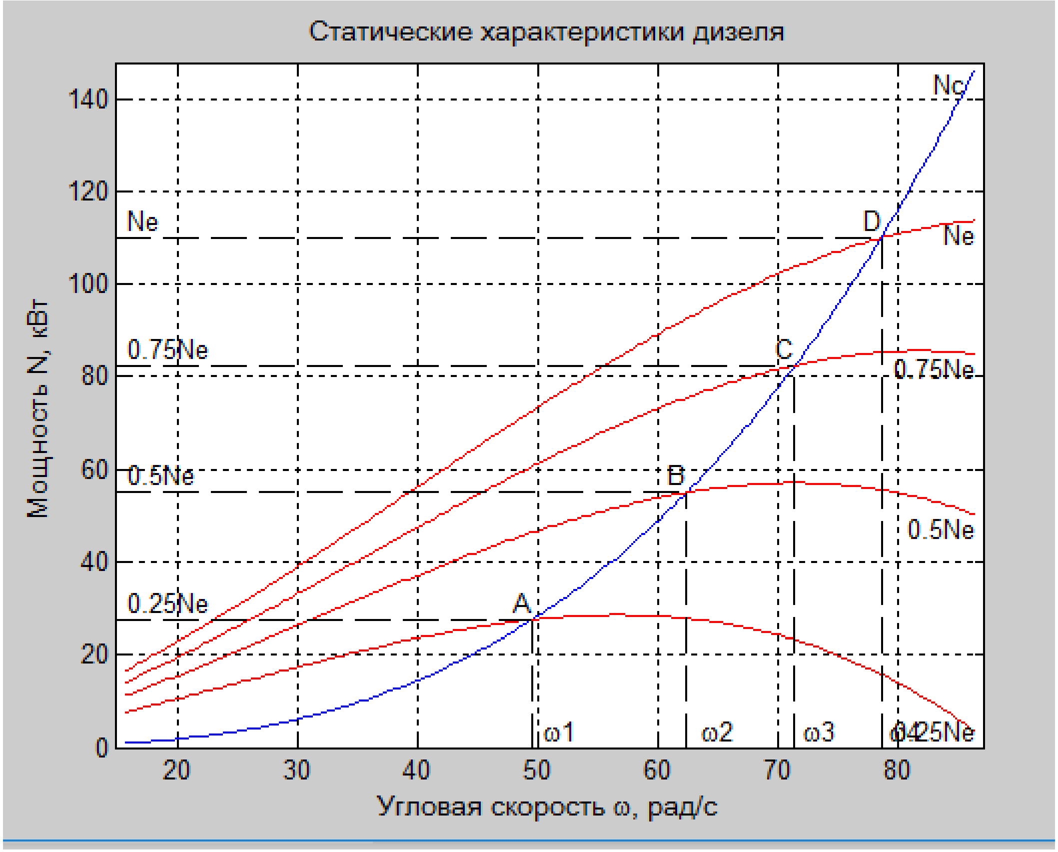

Рис. 14. Результат расчета характеристик частичной эффективной мощности

1.4. Определение фактора устойчивости дизеля.

Параметры дизеля зависят от его режима работы, поэтому их необходимо отдельно определять для рабочих точек. Для всех рабочих точек параметры дизеля определяются аналогично.

Фактор устойчивости дизеля характеризует его способность автоматически компенсировать изменение нагрузки на валу за счет изменения скорости вращения и крутящего момента при неизменной топливоподаче. Для расчета фактора устойчивости используем следующую формулу:

$$F_Д = \frac{1}{\omega_{pi}}\left[\left(\frac{\partial N_c}{\partial \omega}\right)_{\omega_{pi}} - \left(\frac{\partial N_e}{\partial \omega}\right)_{\omega_{pi}}\right],$$

где ω_{pi} – угловая скорость вала дизеля в рабочей точке соответствующего режима частичной топливоподачи.

Определим фактор устойчивости для четырех мощностных режимов работы дизеля.

На графике статических характеристик дизеля (рис. 15) в точках пересечения характеристик эффективной мощности и винтовой характеристики к кривым $N_e(\omega)$, $0{,}75 \cdot N_e(\omega)$, $0{,}5 \cdot N_e(\omega)$, $0{,}25 \cdot N_e(\omega)$ и $N_c(\omega)$ проводятся касательные и для них определяется тангенс угла наклона:

$$\frac{\partial N_c}{\partial \omega} = \frac{\Delta N_c}{\Delta \omega} = \alpha\,;$$

$$\frac{\partial N_e}{\partial \omega} = \frac{\Delta N_e}{\Delta \omega} = \beta\,,$$

где $\Delta\omega$ – небольшое задаваемое изменение частоты в рабочей точке;

$\Delta N_c, \Delta N_e$ – соответствующие изменению частоты $\Delta\omega$ изменения мощности. При вычислении углов наклона мощность следует подставлять в Вт.

Расчет фактора устойчивости дизеля $F_Д$, $Вт \cdot с^2$, определяем по формуле

$$F_Д = \frac{\alpha - \beta}{\omega_{pi}}.$$

Для расчетов принимаем:

для режима $N_e = N_{ен}$: $\Delta\omega_C = 5{,}6$ рад / с, $\Delta\omega_e = 22{,}5$ рад / с, $\Delta N_{CD} = 23{,}5$ КВт, $\Delta N_{eD} = 15{,}8$ КВт;

для режима $N_e = 0{,}75 \cdot N_{ен}$: $\Delta\omega_C = 6{,}54$ рад/с, $\Delta\omega_e = 22{,}3$ рад/с, $\Delta N_{CC} = 22{,}7$ КВт, $\Delta N_{eC} = 13{,}4$ КВт;

для режима $N_e = 0{,}5 \cdot N_{ен}$: $\Delta\omega_C = 9{,}45$ рад/с, $\Delta\omega_e = 22{,}3$ рад/с, $\Delta N_{CB} = 25{,}12$ КВт, $\Delta N_{eB} = 9{,}9$ КВт;

для режима $N_e = 0{,}25 \cdot N_{ен}$: $\Delta\omega_C = 10{,}5$ рад/с, $\Delta\omega_e = 22{,}25$ рад/с, $\Delta N_{CA} = 17{,}56$ КВт, $\Delta N_{eA} = 6{,}27$ КВт.

4. Пример тестового задания для оценки навыков, полученных при выполнении практической работы

Для прохождения теста необходимо решить шесть заданий. Пример варианта тестового задания представлен ниже. Тест выполняется с помощью специальной тестовой программы. После запуска программы необходимо выполнить задание с помощью программы MATLAB и ввести ответ в соответствующее окно тестовой программы. Пример ввода ответа в числовом виде: 10,011.

Задание 1

Вычислить функцию:

$$y(x) = |x|^3 + \frac{1}{|x|}$$

Переменная x лежит в пределах от 0,1 до 5 с шагом 0,5.

Укажите максимальное значение функции.

Задание 2

Вычислите следующее выражение:

$$e^{2+\sin(8)} \cdot \frac{\tan(8)}{4} - \sqrt{56}$$

Задание 3

Вычислите определитель матрицы A:

$$A = \begin{matrix} a & 4 & d+2 & 5 \\ 23 & b & 1+d & c \\ 5 & a+b & 2 & 10 \\ 5 & a^2 & 2 & 2 \end{matrix}$$

Если: a = 1; b = 2; c = 3; d = 4

Задание 4

Решите систему уравнений:

$$\begin{cases} 3x + 4y - 9z = 22 \\ 15x - 9y + 2z = 34 \\ 8x - 16y + 5z = 8 \end{cases}$$

Укажите значение y.