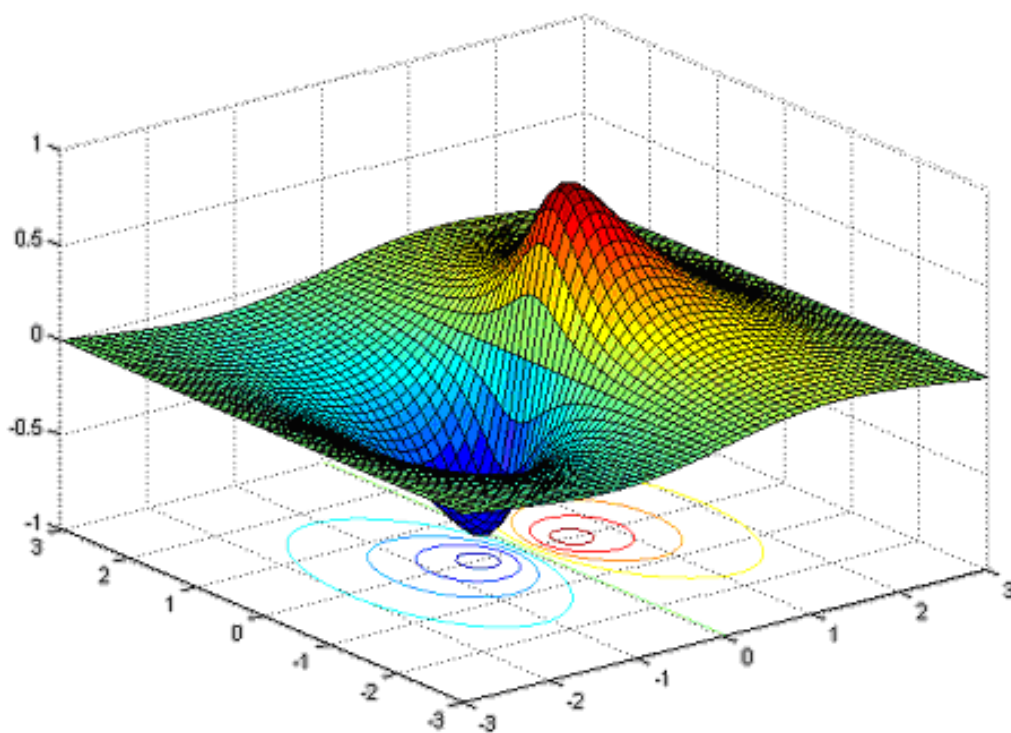


Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Сибирский государственный автомобильно-дорожный университет (СибАДИ)»

С.А.Зырянова, Т.А. Юрина

ПАКЕТЫ ПРИКЛАДНЫХ ПРОГРАММ

Учебно-методическое пособие



Омск • 2018

УДК 004.9
ББК 73.6
397

Согласно 436-ФЗ от 29.12.2010 «О защите детей от информации, причиняющей вред их здоровью и развитию» данная продукция маркировке не подлежит.

Рецензент

д-р техн. наук, проф. В.С. Щербаков (СибАДИ)

Работа утверждена редакционно-издательским советом СибАДИ в качестве учебно-методического пособия.

Зырянова, Светлана Анатольевна.

397 Пакеты прикладных программ [Электронный ресурс] : учебно-методическое пособие / С.А. Зырянова, Т.А. Юрина. – Электрон. дан. – Омск : СибАДИ, 2018. – Режим доступа: <http://bek.sibadi.org/fulltext/esd518.pdf>, свободный после авторизации. – Загл. с экрана.

Включает теоретический материал и задания для самостоятельной работы студентов, позволяющие освоить основные принципы работы с пакетом MATLAB: арифметические вычисления, работа с массивами, построение графиков и поверхностей, решение уравнений и систем уравнений, вычисление интегралов, пределов, производных функций, решение дифференциальных уравнений, работа с М-файлами.

Имеет интерактивное оглавление в виде закладок.

Предназначено для студентов всех форм обучения и для всех направлений подготовки бакалавриата, специалитета, магистратуры и аспирантуры, изучающих дисциплину «Пакеты прикладных программ».

Подготовлено на кафедре «Информационные технологии».

Текстовое (символьное) издание (1,2 МБ)

Системные требования: Intel, 3,4 GHz; 150 Мб; Windows XP/Vista/7; DVD-ROM; 1 Гб свободного места на жестком диске; программа для чтения pdf-файлов: Adobe Acrobat Reader; Foxit Reader

Редактор Н.В. Павлова

Техническая подготовка Н.В. Кенжалинова

Издание первое. Дата подписания к использованию 26.03.2018

Издательско-полиграфический комплекс СибАДИ. 644080, г. Омск, пр. Мира, 5
РИО ИПК СибАДИ, 644080, г. Омск, ул. 2-я Поселковая, 1

© ФГБОУ «СибАДИ», 2018

ВВЕДЕНИЕ

Дисциплина «Пакеты прикладных программ» призвана способствовать развитию и укреплению практических навыков применения ЭВМ, информационных технологий, возможностей современных пакетов прикладных программ в исследовательской деятельности студентов.

MATLAB – одна из старейших, тщательно проработанных и проверенных временем систем автоматизации математических расчетов, построенная на расширенном представлении и применении матричных операций. Однако синтаксис языка программирования системы настолько продуман, что эта ориентация почти не ощущается теми пользователями, которых не интересуют непосредственно матричные вычисления. Матрицы широко применяются в сложных математических расчетах, например при решении задач линейной алгебры и математического моделирования статических и динамических систем и объектов. Они являются основой автоматического составления и решения уравнений состояния динамических объектов и систем.

В MATLAB реализованы современные численные методы компьютерной математики, используются мощные средства графической визуализации и анимационной графики. Возможности системы весьма обширны, а по скорости выполнения задач она нередко превосходит своих конкурентов и применима для расчетов практически в любой области науки и техники.

В данном пособии приведены необходимый теоретический материал и задания для лабораторных работ по использованию пакета MATLAB в инженерных расчетах.

1. ИНТЕРФЕЙС ПОЛЬЗОВАТЕЛЯ

MATLAB (сокращение от англ. «Matrix Laboratory») – это пакет прикладных программ для решения задач технических вычислений и одноименный язык программирования, используемый в этом пакете.

Среда MATLAB включает в себя интерпретатор команд на языке высокого уровня, графическую систему, пакеты расширений и реализована на языке C. Вся работа организуется через командное окно (Command Window), появляющееся при запуске программы, в котором расположена строка ввода, начинающаяся специальными символами «>>» (рис. 1). В этой строке записываются команды для

выполнения системой. В процессе работы данные располагаются в памяти (Workspace); для изображения кривых, поверхностей и других графиков создаются графические окна.

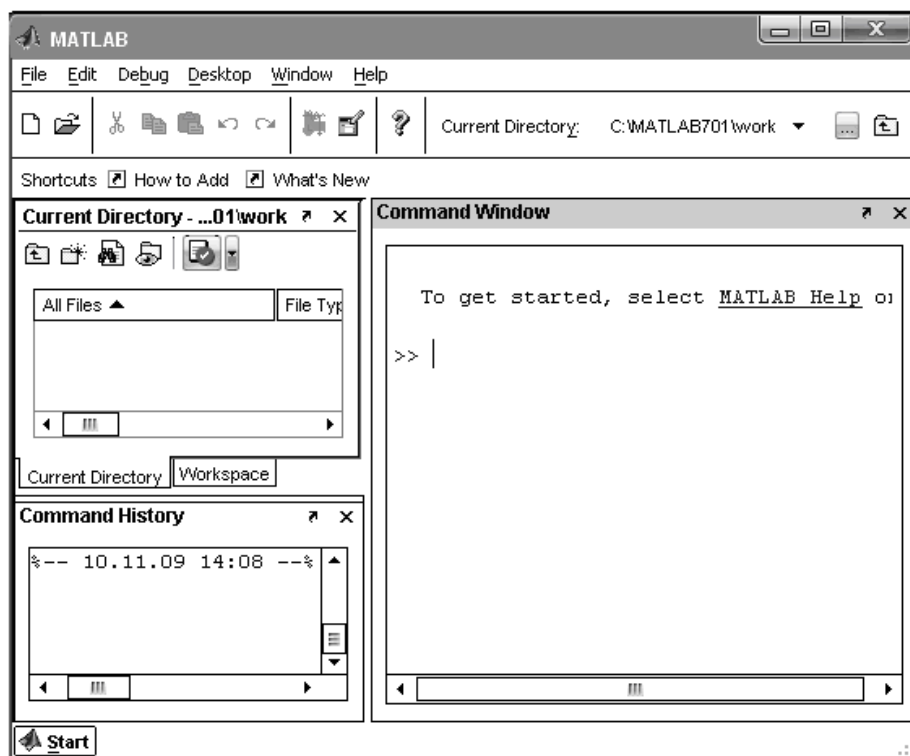


Рис. 1. Интерфейс системы MATLAB

При наборе команд пользователь может использовать клавиши [←], [→], [Home], [End], [Delete], [BackSpace] для перемещения по строке ввода или удаления символов. Для отмены ввода (очистки строки ввода) используется клавиша [Esc]. Следует обратить внимание на применение клавиш [↑] и [↓]. Они используются для подстановки после маркера строки ввода ранее введенных команд, например для их исправления, дублирования или дополнения. При этом указанные клавиши обеспечивают перелистывание ранее введенных строк снизу вверх или сверху вниз.

Ввод команды завершается нажатием клавиши [Enter], при этом MATLAB сразу же выполняет команду и выводит в следующей строке результат. Для того чтобы результат не выводился на экран, в конце команды ставится символ «;».

Если команда слишком длинная, то можно перенести часть ее на новую строку, поставив пробел и троеточие (...), а затем с новой

строки продолжить запись команды. Для очистки командного окна используется команда **clc**.

2. ОСНОВНЫЕ ОБЪЕКТЫ MATLAB

Предупреждение (обычно начинается со слова Warning) – не останавливает вычисления и лишь предупреждает пользователя о том, что диагностируемая ошибка способна повлиять на ход вычислений.

Сообщение об ошибке (после знаков ???) останавливает вычисления.

Математические выражения. Строятся на основе чисел, констант, переменных, операторов, функций и спецзнаков.

Текстовые комментарии. Вводятся с помощью символа «%», указанного перед текстом комментария.

Число – простейший объект языка MATLAB, представляющий количественные данные. Числа могут быть целыми, вещественными с фиксированной и плавающей запятой, комплексные. Примеры представления чисел в MATLAB приведены в табл. 1.

Таблица 1

Примеры представления чисел

| Обычное представление | В системе MATLAB |
|-----------------------|------------------|
| 2 | 2 |
| -3,4 | -3.4 |
| -123,241 | -1.23241e2 |
| 0,000412 | 4.12e-4 |
| 2+3,7i | 2+3.7i |
| 0,54 | .54 |
| 0,2+3i | .2+3i |

Константа – это предварительно определенное числовое или символьное значение, представленное уникальным именем. В MATLAB существуют некоторые стандартные константы, например число «пи» (**pi**), значение мнимой единицы (**i**). Символьной константой считается любая последовательность символов, заключенных в апострофы, например 'Hello!'.

Переменная – это имеющий имя объект, способный хранить некоторые данные. В зависимости от этих данных переменные могут быть числовыми или символьными, векторными или матричными.

Имя переменной должно начинаться с буквы, может содержать буквы, цифры и символ подчеркивания. Имя не должно совпадать с именами других переменных, функций и процедур системы. Прописные и строчные буквы в MATLAB различаются.

Для уничтожения определения переменной используется специальная команда:

- **clear** – уничтожает все переменные;
- **clear x** – уничтожает переменную *x*;
- **clear x,y** – уничтожает переменные *x* и *y*.

Для просмотра значения переменной достаточно набрать ее имя в командной строке (без точки с запятой в конце).

Оператор – это специальное обозначение для определенной операции над данными – *операндами*. Например, арифметическими операторами являются знаки сложения (+), вычитания (–), умножения (*), деления (/), возведения в степень (^).

Функции – это имеющие уникальные имена объекты, выполняющие определенные преобразования своих аргументов и при этом возвращающие результаты этих преобразований. Все имена функций в MATLAB пишутся строчными буквами. Перечень основных функций приведен в табл. 2.

2.1. Присвоение и вывод значений переменных и функций

Для задания переменной значения используется оператор присваивания:

Имя_переменной = Значение;

Для вывода значения переменной нужно в командной строке ввести ее имя и нажать клавишу [Enter].

При вычислении значений арифметических выражений нужно набрать в командной строке это выражение и нажать клавишу [Enter].

Перед тем как вычислять значение математического выражения, необходимо определить значение каждой входящей в него переменной. Вычисляемое выражение может содержать любое количество переменных, операторов и функций.

При арифметических вычислениях в MATLAB соблюдается следующий порядок:

- 1) значения функций;
- 2) возведение в степень;

- 3) умножение и деление в порядке их следования;
 4) сложение и вычитание в порядке их следования.
 Для изменения порядка действий используются круглые скобки.

Таблица 2

Основные функции

| Функция | Описание |
|---|---|
| <i>Алгебраические и арифметические функции</i> | |
| abs(x) | Вычисление модуля числа x |
| exp(x) | Возведение константы e в степень x (e^x) |
| factor(n) | Возвращает вектор-строку, содержащую простые множители числа n (f=factor(221), f=13 17) |
| log(x) | Вычисление натурального логарифма числа x |
| log2(x) | Вычисление логарифма числа x по основанию 2 |
| log10(x) | Вычисление десятичного логарифма числа x |
| mod(x,y) | Вычисление остатка от деления x на y |
| pow2(y) | Возведение числа 2 в степень x |
| sqrt(x) | Вычисление квадратного корня из числа x |
| <i>Тригонометрические и обратные им функции</i> | |
| sin(x) | Вычисление синуса числа x |
| cos(x) | Вычисление косинуса числа x |
| tan(x) | Вычисление тангенса числа x |
| asin(x) | Вычисление арксинуса числа x |
| acos(x) | Вычисление арккосинуса числа x |
| atan(x) | Вычисление арктангенса числа x |
| <i>Функции округления и знака</i> | |
| fix(x) | Округление к ближайшему целому в сторону нуля |
| floor(x) | Округление к ближайшему целому в сторону отрицательной бесконечности |
| ceil(x) | Округление к ближайшему целому в сторону положительной бесконечности |
| round(x) | Округление до ближайшего целого |
| sign(x) | Возвращает -1 при отрицательном x , 1 – при положительном x , 0 – при $x=0$ |

Пример 1

Вычислить значение выражения $z = \frac{x^2 + y}{3 - |\sin x|} + 2$ при $x=25$,
 $y=3,6$.

Порядок ввода:

>> x=25;

```
>> y=3.6;
>> z=(x^2+y)/(3-abs(sin(x)))+2
```

В результате получим 221.204.

Пример 2

Вычислить значение выражения $\cos^2\left(\frac{2}{3}\pi\right)$.

Порядок ввода:

```
>> cos(2/3*pi)^2
```

В результате получим число 0,25.

2.2. Форматированный вывод

Поскольку все вычисления в MATLAB выполняются с двойной точностью, формат вывода может управляться с помощью команд, приведенных в табл. 3.

Таблица 3

Команды форматированного вывода

| Команда | Формат вывода |
|----------------|---|
| format short | Вывод с фиксированной точкой и 4 знаками после десятичной точки (по умолчанию) |
| format long | Вывод с фиксированной точкой и 16 знаками после десятичной точки (по умолчанию) |
| format short e | Вывод в экспоненциальной форме с мантиссой из 5 цифр и показателем из 3 цифр |
| format long e | Вывод в экспоненциальной форме с мантиссой из 16 цифр и показателем из 3 цифр |
| bank | Вывод числа с любым количеством цифр до десятичной точки и 2 цифрами после |

После вызова одного из приведенных выше форматов он сохраняется до вызова другого.

Команда **format compact** подавляет большинство пустых строк, позволяя вывести большее количество информации на экран или страницу. Она не зависит от других команд формата.

Команда **eval** чрезвычайно полезна, позволяя обмениваться между процедурами (особенно внешними) строкой-параметром, которая прину-

ждает систему к выполнению заданной этой строкой команды (или серии команд).

Кроме этого, для вывода значений с нужной точностью используются функции:

1) **digits (D)** – устанавливает заданное число цифр D для чисел арифметики произвольной точности (D – целое число, строка или переменная типа `sym`).

2) **digits** – возвращает число значащих цифр в числах арифметики произвольной точности, заданное ранее (по умолчанию 32);

3) **vpa(S)** – возвращает результат вычислений каждого элемента символьного массива S , используя арифметику произвольной точности с текущим числом цифр D , установленным функцией *digits*. Результат R имеет тип `sym`;

Пример

```
>>digits (7)
```

```
>>vpa(pi)
```

Получим `ans = 3.141593`

4) **vpa(S, D)** – возвращает результат вычислений каждого элемента массива S , используя арифметику произвольной точности с количеством знаков чисел D .

Пример

```
>>vpa(pi,4)
```

Получим `ans = 3.142`

2.3. Табулирование функции

Табулирование функции – это вычисление всех ее значений при каждом значении аргумента в указанном диапазоне. Для задания диапазона чисел необходимо ввести:

Имя_переменной=Начальное_значение:Шаг:Конечное_значение;

Если шаг равен 1, то его можно не указывать, а задать только начальное и конечное значения.

После ввода диапазона значений аргумента функции задается сама функция. Так как аргумент имеет несколько значений, то выполнение операций умножения, деления и возведения в степень должно выполняться поэлементно над каждым из значений. Для этого используются операции `.*`, `./` и `.^` соответственно.

Для умножения на число, сложения и вычитания используются обычные операции `*`, `+` и `-` соответственно.

Пример

Вычислить значения функций $y = 2x \sin x$ и $z = 3x^2 + \cos x$ при $x \in [-1,5; 1,5]$ с шагом 0,5.

Порядок ввода:

```
>> x=-1.5:0.5:1.5
```

```
>> y=2*x.*sin(x)
```

```
>> z=3*x.^2+cos(x)
```

В результате получим

x =

```
-1.5000 -1.0000 -0.5000    0  0.5000  1.0000  1.5000
```

y =

```
 2.9925  1.6829  0.4794    0  0.4794  1.6829  2.9925
```

z =

```
 6.8207  3.5403  1.6276  1.0000  1.6276  3.5403  6.8207
```

3. РАБОТА С МАТРИЦАМИ И ВЕКТОРАМИ

Элементы матрицы задаются в квадратных скобках, причем элементы одной строки отделяются друг от друга пробелом или запятой, а строки отделяются друг от друга точкой с запятой.

Пример

Задать матрицу $M_{2 \times 2} = \begin{pmatrix} 3 & -1 \\ 5,5 & 1 \end{pmatrix}$ и вектор $U = \begin{pmatrix} 5 \\ 2,1 \\ -4 \end{pmatrix}$.

Порядок ввода:

```
>> M = [3 -1; 5.5 1]
```

```
>> U=[5; 2.1; -4]
```

Для обращения к отдельному элементу матрицы нужно указать имя матрицы и индексы элемента в круглых скобках.

Пример

Присвоить элементу второй строки первого столбца матрицы M значение 2.

Порядок ввода:

```
>> M(2, 1)=2
```

Иногда требуется организовать доступ к какому-либо блоку матрицы, что реализуется с помощью знака «:» (например, использование $A(3:5,:)$ дает нам строки с 3-й по 5-ю матрицы A .

Пример

Сформировать вектор с целочисленными значениями от 3 до 7.

Порядок ввода:

```
>> M = [3:7]
```

В результате получим

$M =$

3 4 5 6 7

Для объединения матриц используется оператор объединения – это имена объединяемых матриц, записанные в квадратных скобках (например, $V=[M R]$).

Операции над матрицами и векторами

Для работы с матрицами и векторами в MATLAB существуют специальные команды и функции. Основные операции с матрицами приведены в табл. 4.

Операции с матрицами

| Команда | Описание |
|----------------|---|
| A+B | Сложение матриц A и B одинаковой размерности |
| A-B | Вычитание матриц A и B одинаковой размерности |
| A*B | Матричное произведение массивов ($A_{n \times m} \cdot B_{m \times k} = C_{n \times k}$) |
| A.*B | Поэлементное умножение матриц A и B одинаковой размерности |
| A./B | Поэлементное деление матриц A и B одинаковой размерности |
| A.^k | Поэлементное возведение массива в степень k |
| A' | Транспонирование матрицы A (A^T) |
| A^-1 inv(A) | Вычисление обратной матрицы (A^{-1}) |
| det(A) | Вычисление определителя матрицы ($ A $) |
| size(A) | Определение размерности матрицы A |
| length(B) | Длина вектора |
| trace(A) | След матрицы A (сумма элементов на главной диагонали) |
| sum(A) | Сумма элементов в каждом столбце матрицы A |
| prod(A) | Произведение элементов в каждом столбце матрицы A |
| diag(A) | Вектор-столбец элементов главной диагонали матрицы A |
| sort(A) | Сортировка каждого столбца матрицы A |
| max(A) | Вычисление максимума в каждом столбце матрицы A |
| min(A) | Вычисление минимума в каждом столбце матрицы A |
| mean(A) | Вычисление среднего значения в каждом столбце матрицы A |
| rot90(A) | Поворот матрицы A влево на 90° |
| rot90(A,-1) | Поворот матрицы A вправо на 90° |
| rot90(A,k) | Поворот матрицы A на кратное 90° значение ($k = \pm 1, \pm 2, \dots$ – множитель, на который умножается угол 90°) |
| fliplr(A) | Отразить матрицу A слева направо |
| flipud(A) | Отразить матрицу A сверху вниз |
| tril(A) | Создание нижней треугольной матрицы A |
| triu(A) | Создание верхней треугольной матрицы A |
| eye(n) | Формирование единичной матрицы заданного размера n |
| eye(size(A)) | Формирование единичной матрицы по размеру данной квадратной матрицы A |
| ones(n,m) | Создание матрицы из единиц данного размера $n \times m$ |
| ones(n) | Создание квадратной матрицы из единиц размера $n \times n$ |
| zeros(n,m) | Создание матрицы из нулей данного размера $n \times m$ |
| rand(n,m) | Создание матрицы из случайных чисел со значениями из $[0; 1]$ |
| poly(A) | Формирование характеристического полинома заданной числовой матрицы A |
| poly(eig(A)) | Формирование коэффициентов характеристического полинома матрицы A по ее заданным собственным числам |
| A(n,:) = [] | Удаление из матрицы A строки с номером n |
| A(:,n) = [] | Удаление из матрицы A столбца с номером n |

Пример

Даны матрицы

$$A = \begin{pmatrix} 2 & -1 \\ 1 & 3 \end{pmatrix}, \quad B = \begin{pmatrix} 4 & 6 \\ -5 & 3 \end{pmatrix}.$$

Вычислить $A \cdot B^{-1}$.

Порядок ввода:

```
>> A=[2 -1;1 3]
```

```
>> B=[4 6;-5 3]
```

```
>> A*B^-1
```

В результате получим матрицу $\begin{pmatrix} 0,0238 & -0,381 \\ 0,4286 & 0,1429 \end{pmatrix}$.

САМОСТОЯТЕЛЬНАЯ РАБОТА № 1

I. Вычислить значения выражений при $a=3$, $b=2,6$:

1) $\cos \frac{3\pi}{7}$;

7) $\sin^2 \frac{3\pi}{8} - \sqrt{963 - 27^2}$;

2) $e^{-2} + \sin^2 3\pi$;

8) $\ln 3 + \lg 4$;

3) $\sqrt[3]{\operatorname{tg}\left(\frac{7}{9}\pi\right)}$;

9) $\frac{\sin b + \cos(a + 2b)}{2}$;

4) $\ln \frac{2b}{3} + e$;

10) $\sqrt{\sin 5\pi} + \sqrt{|\cos 5\pi|}$;

11) $\arcsin 0,73$;

5) $\log_2 256 + 2^a$;

12) $\frac{|a-3| - 5\sqrt{|b-2|}}{a+2b}$.

6) $\sqrt[3]{12+a} - \frac{\ln 7}{b+3}$;

II. Вычислить значения выражений при $a=5$, $b=5,7$:

$$\frac{a^2 + \sqrt[5]{b^3 + e^4}}{|b^2 - a^2|} + \sin^3 5\pi; \quad \sin^3 5\pi + \cos|b-a| + \frac{\ln 10}{\sqrt{\operatorname{tg}\left(\frac{7}{9}\right)\pi}}.$$

III. Протабулировать функции f , g , z :

1) $x \in [-2; 2]$, $h=0,2$, $f=|x-1|^2$, $g=\cos^2(3x)$, $z=2x^3-3x^2+1$;

$$2) x \in [-4; 3], h=0,6, f=\ln|x+5|, g=\sin x + \cos(2x), z=\sqrt{x^2+2};$$

$$3) x \in [2; 12], h=1, f=\lg x + 1, g=3|x-3|, z=\frac{2x^2}{5+x}.$$

IV. Создать вектор-строки:

- вектор p_1 с целочисленными элементами от 2 до 8;
- вектор p_2 с целочисленными значениями от 12 до 28 с шагом 2;
- вектор p_3 со значениями от 0 до 4 с шагом 0,8.

V. Создать матрицы:

- единичную матрицу C размерности 4;
- нулевую матрицу D размерности 2×3 ;
- V – из случайных чисел размерности 4×3 .

САМОСТОЯТЕЛЬНАЯ РАБОТА № 2

I. Даны матрицы

$$A = \begin{pmatrix} 2 & 1 & -1 \\ 3 & 1 & 2 \\ 2 & 1 & -1 \end{pmatrix}; \quad B = \begin{pmatrix} 3 & 2 & 3 \\ -2 & 7 & 2 \\ -4 & 1 & 5 \end{pmatrix}.$$

1. Вычислить:

- сумму матриц A и B ;
 - разность матрицы B и A ;
 - поэлементное произведение матриц;
 - матричное произведение матриц;
 - квадрат матрицы A (умножить матрицу A саму на себя);
 - поэлементное возведение матрицы A в квадрат;
 - определитель матрицы A ;
 - обратную матрицу к матрице B ;
 - след матрицы B ;
 - сумму элементов каждого столбца матрицы A ;
 - произведение элементов каждой строки матрицы B ;
 - сумму элементов на главной диагонали матрицы B ;
 - произведение минимальных элементов каждого столбца матрицы A ;
 - среднее значение всех элементов матрицы B .
2. Отобразить матрицу A сверху вниз.
3. Отобразить матрицу B слева направо.

4. Повернуть матрицу B влево на 90° .
5. Повернуть матрицу B вправо на 90° .
6. Умножить матрицу B на число 2.
7. Транспонировать матрицу A .
8. Присвоить последнему элементу матрицы B значение 7.
9. Увеличить второй элемент первой строки матрицы A на 2.
10. Удалить из матрицы A последнюю строку.
11. Удалить из матрицы B первый столбец.
12. Отсортировать элементы в каждом столбце матрицы B .
13. Отсортировать элементы в каждой строке матрицы A .

II. Даны массивы

$$A = \begin{pmatrix} 1 & 2 & -1 \\ 2 & 3 & 1 \\ -1 & -2 & 2 \end{pmatrix}; \quad B = (-3 \quad 1 \quad 2); \quad C = \begin{pmatrix} 2 & 4 & 3 \\ -2 & 1 & 2 \\ 4 & 5 & 1 \end{pmatrix}; \quad D = \begin{pmatrix} 2 \\ -1 \\ 3 \end{pmatrix}.$$

Вычислить:

- | | | |
|-------------------------|--------------------------|---------------------------|
| 1. B^T . | 5. $A \cdot C$. | 9. $ A + C $. |
| 2. $A + C^{-1}$. | 6. $C \cdot A$. | 10. $(A^T + C)^2$. |
| 3. $A^{-1} \cdot B^T$. | 7. $A \cdot A \cdot A$. | 11. $A \cdot D \cdot B$. |
| 4. $B \cdot D$. | 8. $2 \cdot D + B^T$. | 12. $3 \cdot A / C$. |

III. Даны массивы

$$A = \begin{pmatrix} 2 & 3 & 1 \\ 5 & 3 & 6 \\ 3 & 1 & 4 \end{pmatrix}; \quad B = \begin{pmatrix} 5 & 1 & 3 \\ 5 & 7 & 1 \\ 3 & 1 & 7 \end{pmatrix}; \quad C = (-1 \quad 2 \quad 4)$$

Вычислить:

- a) $A + B^{-1}$;
- b) $(A^T + B^T)^2$;
- c) $A \cdot C^T$.

4. ПОСТРОЕНИЕ ДВУМЕРНЫХ ГРАФИКОВ

Функции одной переменной $y(x)$ находят широкое применение в практике математических и других расчетов, а также в технике компьютерного математического моделирования. Для отображения таких функций используются графики в декартовой (прямоугольной) сис-

теме координат. При этом обычно строятся две оси – горизонтальная X и вертикальная Y – и задаются координаты x и y , определяющие узловые точки функции $y(x)$. Эти точки соединяются друг с другом отрезками прямых, т.е. при построении графика осуществляется линейная интерполяция для промежуточных точек. Поскольку MATLAB – матричная система, совокупность точек $y(x)$ задается векторами X и Y одинакового размера.

При построении графиков появляется графическое окно. Иногда оно бывает скрыто ранее имеющимися окнами как системы MATLAB, так и других приложений. Если вы не увидели окна графика, то поищите его в списке открытых окон (приложений) на панели задач или с помощью клавиш [Alt]+[Tab].

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.

4.1. Графики в декартовой системе координат

Для построения графиков функций в MATLAB служит команда **plot**, имеющая несколько вариантов записи:

- **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 и т. д.) и менять тип линии графика, x – аргумент функции, y – функция. Значения строковой константы представлены в табл. 5–7.

При отсутствии указания на цвет линий и точек он выбирается автоматически из таблицы цветов.

Таблица 5

Цвет линии

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

Таблица 6

Тип точки

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

Таблица 7

Тип линии

| Код | Описание |
|-----|--------------|
| - | Сплошная |
| -. | Штрихпунктир |
| -- | Штриховая |

Пример 1

Построить график функции $y=1/x$ на отрезке $[1; 8]$, шаг 0,1.

Порядок ввода:

```
>> x = 1: 0.1: 8;
```

```
>> y = 1./x;
```

```
>> plot(x,y)
```

В результате получится график, представленный на рис. 2.

Пример 2

Построить графики функций: $y=\sin(x)$ на отрезке $[0; 2\pi]$ с шагом $\pi/100$ синим цветом двойным пунктиром и $y=\cos(x)$ на отрезке $[0; 2\pi]$ с шагом $\pi/10$ красным и маркером в виде окружности.

Порядок ввода:

```
>> x1 = 0: pi/100: 2*pi;
```

```
>> x2 = 0: pi/10: 2*pi;
```

```
>> plot(x1, sin(x1), 'b:', x2, cos(x2), 'r0')
```

В результате получится график, представленный на рис. 3.

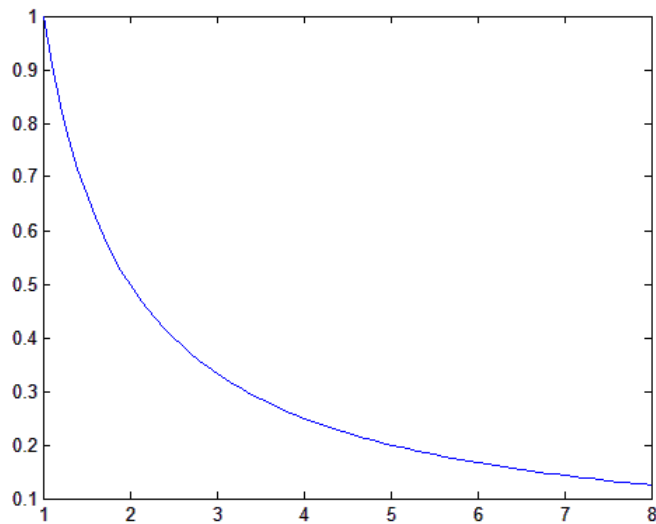


Рис. 2. Стандартный график функции $y=1/x$

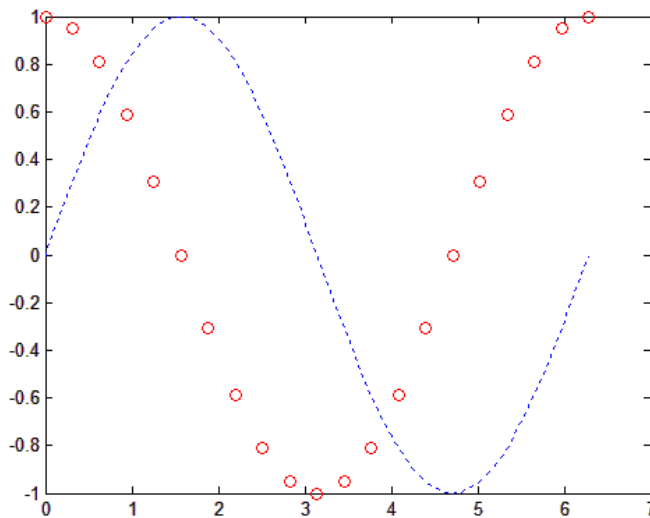


Рис. 3. Графики функций $y=\sin x$ и $y=\cos x$

4.2. Графики в полярной системе координат

Для построения графика в полярной системе координат используется команда **polar**.

polar(phi, r) – строит график с углом phi и радиусом r .

Пример

Построить график функции $y = \sin(2t) \cdot \cos(2t)$ на отрезке $[0; 2\pi]$, шаг 0,01.

Порядок ввода:

```
>> t=0: .01: 2*pi;
```

```
>> polar (t, sin(2*t).*cos(2*t), '--r')
```

В результате этого построения получится график, представленный на рис. 4.

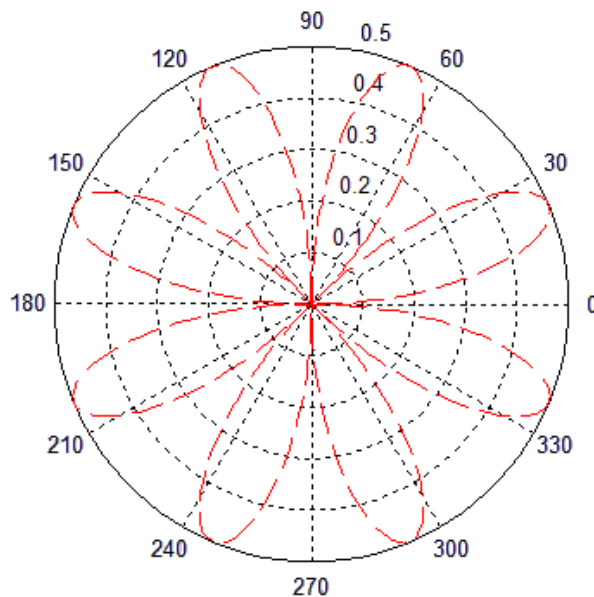


Рис. 4. Полярная система координат

4.3. Разбиение графического окна

Команда **subplot(m, n, p)** разбивает графическое окно на $m \times n$ подокон, при этом m – число подокон по вертикали, n – число подокон по горизонтали, p – номер подокна, в котором будет строиться текущий график (нумерация идет по строкам).

Пример

```
>> t= 0: pi/50: 2*pi;
```

```
>> subplot(2, 1, 1); plot(t, sin(t))
>> subplot(2, 1, 2); plot(t, cos(t))
```

В результате этого построения получится график, представленный на рис. 5.

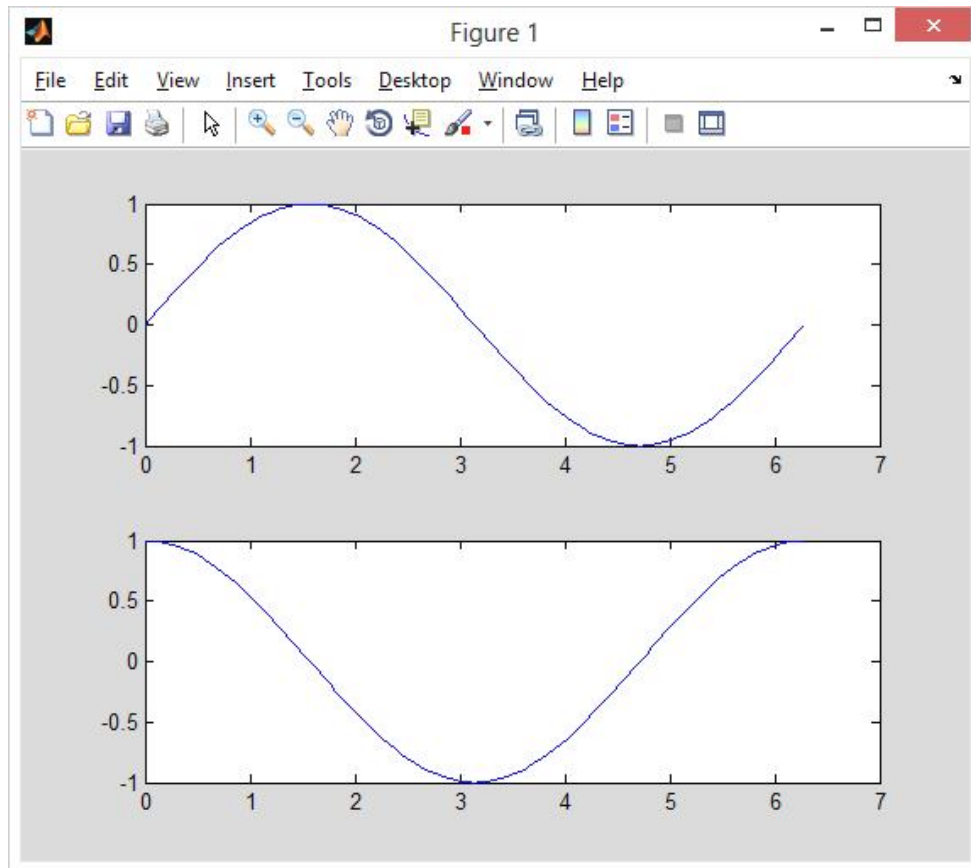
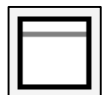


Рис. 5. Разбиение графического окна

4.4. Форматирование графиков



Для включения и выключения режима редактирования графика используются кнопки **Show Plot Tools (Показать окно свойств графика)** и **Hide Plot Tools (Скрыть окно свойств графика)** на панели инструментов в окне графика.

В нижней части окна редактирования находится панель для форматирования графика (*Property Editor*), которая имеет различный вид в зависимости от того, какой элемент графика выделен (рис. 6). Щелкнув по нужному элементу, можно изменить параметры форма-

тирования данного элемента (толщина и цвет линии, тип и размеры маркеров, подписи и т.д.). Основные команды форматирования для разных элементов графика приведены в табл. 8.

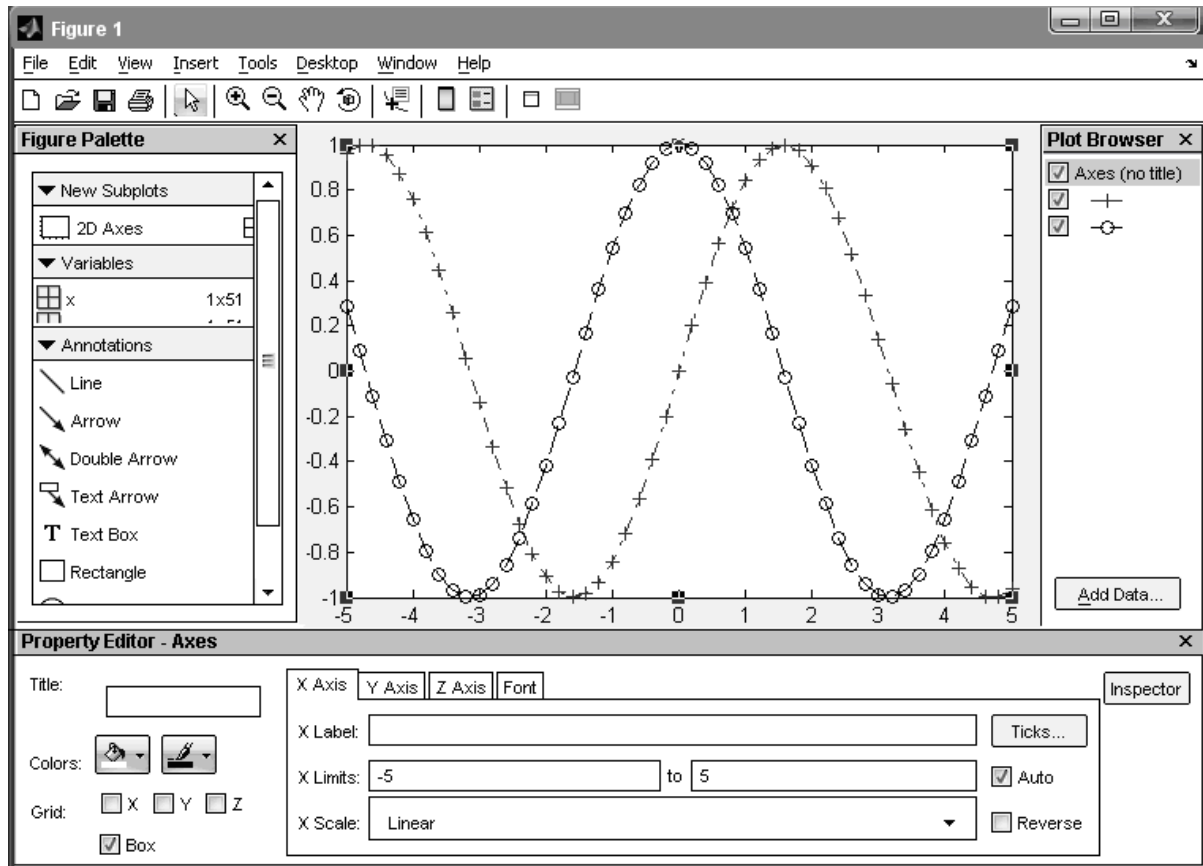


Рис. 6. Окно графика в режиме редактирования

Также можно отформатировать график, используя специальные команды в командном окне:

- **grid** – добавление сетки;
- **xlabel('текст')** – добавляет подпись по оси X ;
- **ylabel('текст')** – добавляет подпись по оси Y ;
- **title('текст')** – добавляет заголовок графика;
- **legend('текст1', ...)** – добавляет легенду;
- **text(X, Y, 'текст')** – добавляет текст в точку с координатами (X, Y) ;
- **axis([xmin xmax ymin ymax])** – задает шкалу по осям X и Y ;
- **line([X1 X2],[Y1 Y2])** – строит линию от точки с координатами $(X1, Y1)$ до точки с координатами $(X2, Y2)$;

- **set(график,'параметр1',значение,...)** – задает параметры форматирования графика (color – цвет линии, linewidth – толщина линии, linestyle – тип линии).

Таблица 8

Команды форматирования графика

| Выделенный элемент графика | Команда | Описание |
|-----------------------------|-------------------|------------------------------|
| Окно графика | Figure Name | Заголовок окна графика |
| | Figure Color | Цвет фона окна графика |
| График (область построения) | Title | Заголовок графика |
| | Colors | Цвет фона области построения |
| | Grid | Линии сетки |
| | X Axis: | Ось X |
| | X Label | Подпись оси X |
| | X Limits | Интервал значений на оси X |
| | X Scale | Тип шкалы оси X |
| Ticks | Метки по осям | |
| Font | Шрифт | |
| Линия графика | Display Name | Имя ряда |
| | Plot Type | Тип графика |
| | Line Style | Тип линии |
| | Line Width | Толщина линии |
| | Color | Цвет линии |
| | Marker | Тип маркера |
| | Marker Size | Размер маркера |
| | Marker Edge Color | Цвет маркера |
| Подпись | Line Style | Тип линии обрамления |
| | Line Width | Толщина линии обрамления |
| | Edge Color | Цвет линии |
| | Background | Цвет фона |

Пример

Построить в одной системе координат графики функций $y=\sin x$ и $z=\sin^3 x$ на отрезке $[-6; 6]$ с шагом 0,1.

Порядок ввода:

```
>> x=-6:0.1:6;
>> y=sin(x);
>> z=sin(x).^3;
>> plot(x,y,x,z)
```

Полученный график отформатируем с помощью окна свойств графика (рис. 7).

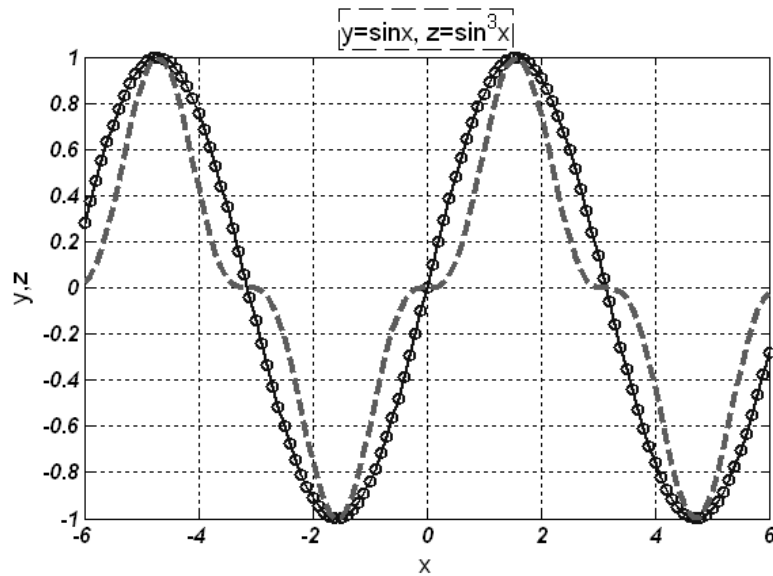


Рис. 7. Отформатированные графики функций

Пример

Построить и отформатировать графики функций $y=2x^3+3x^2$ и $z=3|3x-5|$ на отрезке $[-5; 5]$ с шагом 0,1.

Порядок ввода:

```
>> x=-5:0.1:5;
>> y=2*x.^3+3*x.^2;
>> z=3*abs(3*x-5);
>> g=plot(x,y,x,z)
>> set(g,'linewidth',3,'linestyle','--')
>> title('Plots')
>> xlabel('x')
>> ylabel('y, z')
>> grid
>> legend('y','z')
>> text(-2.5,80,'z')
>> line([-2.4 -2.2],[70 50])
>> text(-1.5,-50,'y')
>> line([-1.4 -1.2],[-40 5])
```

Полученный график показан на рис. 8.

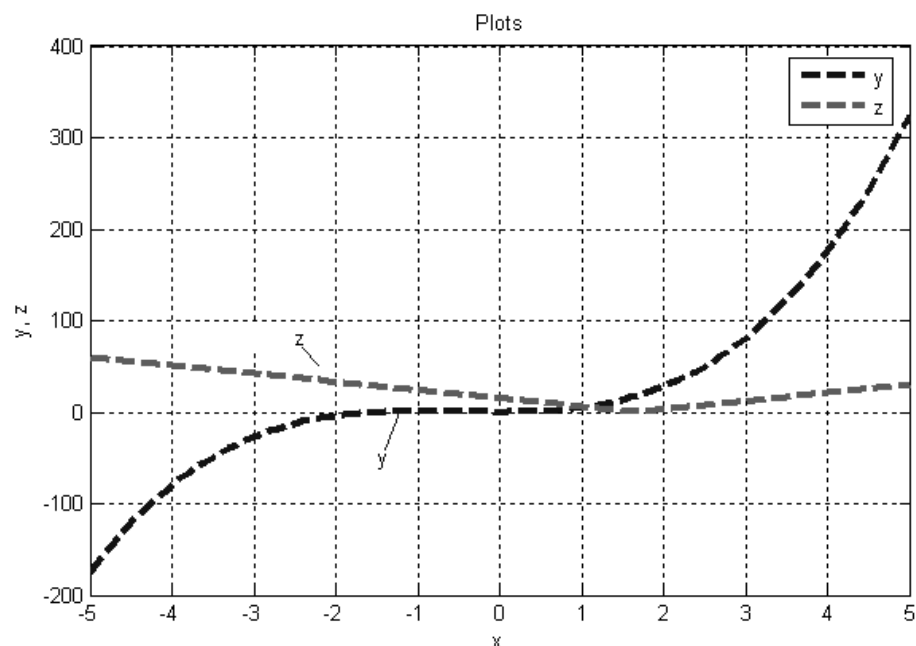


Рис. 8. Форматирование графиков

4.5. Добавление объектов на график

Дополнительно на график можно нанести различные объекты с помощью соответствующих команд пункта меню **Insert**:

- X Label** – подпись оси X ;
- Y Label** – подпись оси Y ;
- Title** – заголовок графика;
- Legend** – легенда;
- Line** – линия;
- Arrow** – стрелка;
- Text Arrow** – стрелка с текстом;
- Double Arrow** – двойная стрелка;
- TextBox** – текстовое поле;
- Rectangle** – прямоугольник;
- Ellipse** – эллипс;
- Axes** – оси.

4.6. Применение графической «лупы»

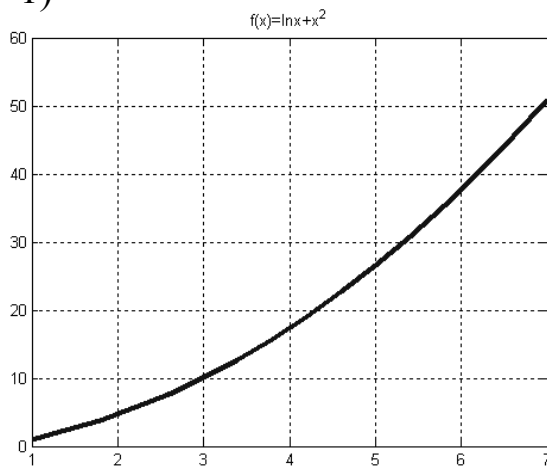
На панели инструментов есть кнопки с изображением лупы и знаками «+» и «-». С их помощью выполняются команды **Zoom In (Увеличить)** и **Zoom Out (Уменьшить)**. Это позволяет увеличивать или уменьшать масштаб просмотра изображения.

САМОСТОЯТЕЛЬНАЯ РАБОТА № 3

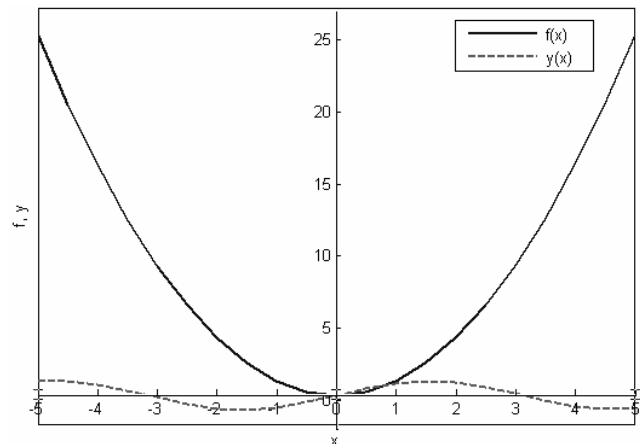
I. Построить графики функций в одной системе координат, отформатировав их с помощью окна свойств графика по образцу:

- 1) $f = \ln x + x^2$, $x \in [1; 7]$, шаг 0,4;
- 2) $f = x^2$, $y = \sin x$, $x \in [-5; 5]$, шаг 0,5;
- 3) $f = \sin x^2 - \cos x$, $y = x^2 - 3$, $x \in [-4; 4]$, шаг 0,3;
- 4) $f = \sin x^2 + \cos x$, $y = x^2 - 4$, $z = \sqrt{|x|} - 0,8$, $x \in [-4; 4]$, шаг 0,4.

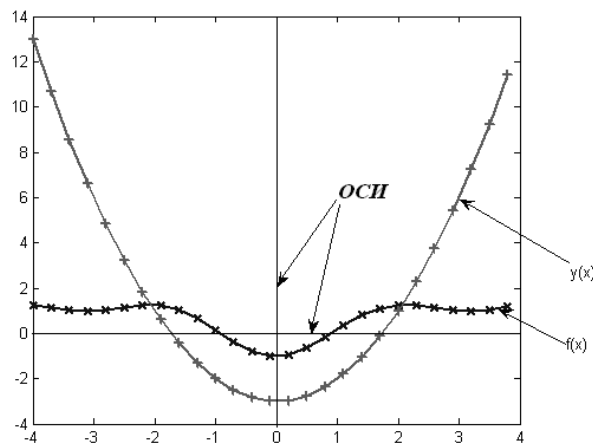
1)



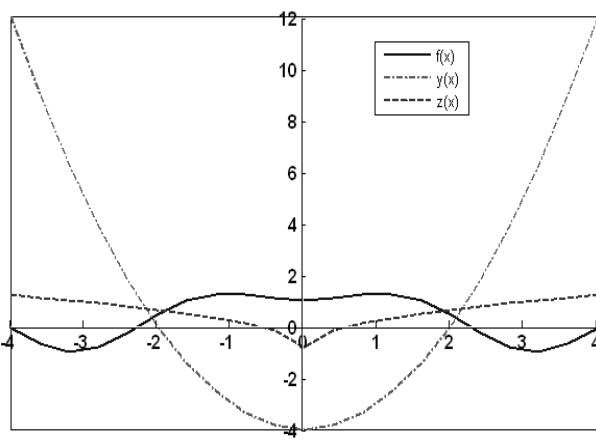
2)



3)



4)



II. Построить в одном окне в разных подокнах графики для $t \in [0, 2\pi]$: $y_1 = 3(1 + \cos t)$; $y_2 = 3(1 - \cos t)$; $y_3 = 3(1 + \sin t)$; $y_4 = 3(1 - \sin t)$. Нанести названия графиков.

III. Построить графики функций, заданных параметрически (по одной оси – $x(t)$, по другой – $y(t)$):

- 1) $x(t) = t \cdot \cos t$, $y(t) = t \cdot \sin t$, $t \in [0; 10\pi]$, шаг $\pi/10$;
- 2) $b = 3$, $x(t) = b \cdot \cos^3 t$, $y(t) = b \cdot \sin^3 t$, $t \in [0; 2\pi]$, шаг $\pi/12$;

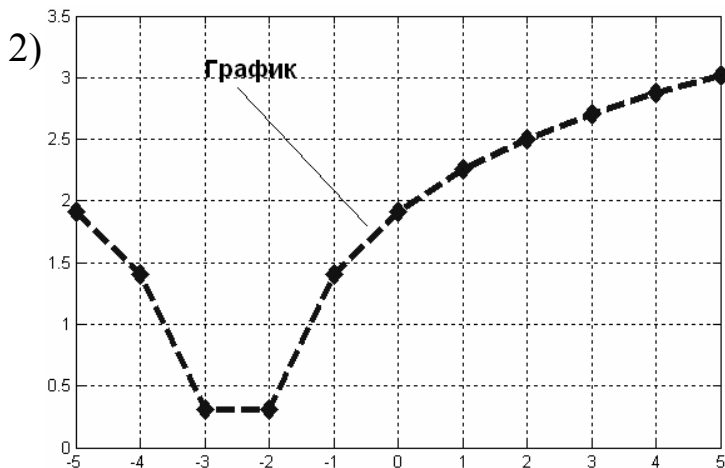
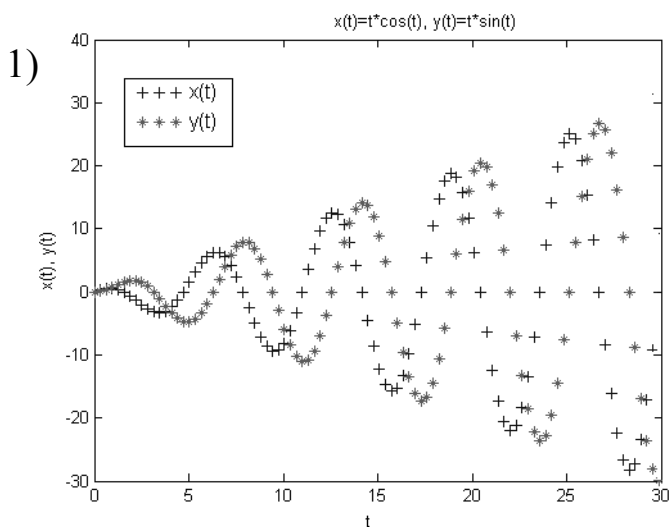
3) $a=4$, $x(t)=a \cdot (t^2 - 1)/(t^2 + 1)$, $y(t)=a \cdot t \cdot (t^2 - 1)/(t^2 + 1)$, $t \in [-10; 10\pi]$, шаг 0,5.

IV. Построить в одном окне в разных подокнах графики для $t \in [0, 10\pi]$: $y_1=1+\cos t$; $y_2=1-\cos t$; $y_3=3\cos t+3\sin t$. Нанести названия графиков.

V. Построить графики функций в одной системе координат и отформатировать их с помощью команд форматирования:

1) $x(t)=t \cdot \cos t$, $y(t)=t \cdot \sin t$, $t \in [0; 10\pi]$, шаг $\pi/10$;

2) $f=\ln|x+2,5|+1$, $x \in [-5; 5]$, шаг 1;



VI. Построить график в полярной системе координат функции $r=\cos(k\varphi)$ для $\varphi \in [0; 2\pi]$ при $k=3$ и при $k=5$.

5. ПОСТРОЕНИЕ ТРЕХМЕРНЫХ ГРАФИКОВ

Построение трехмерных графиков (поверхностей) во многом похоже на построение двумерных графиков. Команды построения трехмерных графиков можно разделить на несколько групп:

1. Поверхности:

- **plot3(x, y, z)** – строит поверхность по точкам, координаты которых берутся из матриц X , Y , Z ;

- **plot3(x1,y1,z1,x2,y2,z2,...)** – строит несколько поверхностей z_1 , z_2 и т. д.;

- **plot3(x,y,z,s)** – строит поверхность с заданным типом и цветом линии и точек (s – строковая константа, задающая тип и цвет линии и точек);

- **plot3(x1,y1,z1,s1,x2,y2,z2,s2,x3,y3,z3,s3,...)** – строит несколько поверхностей z_1 , z_2 и т.д. заданным типом и цветом линии и точек (s_1 , s_2 и т.д.).

Значения строковой константы S приведены ранее (см. табл. 5–7).

2. Сплошные цветные поверхности:

- **surf(x,y,z)** – строит цветную параметрическую поверхность по данным матриц X , Y и Z ;

- **surfc(x,y,z)** – строит цветную параметрическую поверхность по данным матриц X , Y , Z и проекцию фигуры на опорную плоскость.

3. Графики трехмерных слоеных поверхностей, как бы состоящие из тонких пластинок – слоев:

- **waterfall(X,Y,Z)** – строит поверхность, состоящую из тонких пластинок – слоев.

4. Каркасные поверхности:

- **mesh (x, y, z)** – строит каркасную поверхность.

5. Контурные графики в виде линий равного уровня:

- **contour (x, y, z)** – строит линии уровня по данным матрицы z с учетом диапазона изменения координат x и y .

- **contour(z)** – строит линии уровня по данным матрицы z с автоматическим заданием диапазона изменения координат x и y .

- **contour(x, y, z, n)** – строит контурные графики с явным заданием n линий уровня.

- **contour(x, y, z, v)** – строит линии уровня для высот, указанных значениями вектора V .

Порядок построения трехмерных графиков:

1. Задать матрицы X и Y на основе диапазонов значений переменных x и y с помощью команды преобразования диапазонов значений переменных в соответствующие матрицы:

[X,Y]=meshgrid(диапазон1, диапазон2)

Если диапазоны одинаковые:

[X,Y]=meshgrid(диапазон)

2. Задать функцию $z(X,Y)$.

3. Построить поверхность нужного вида с помощью соответствующей команды.

4. Отформатировать график.

Отформатировать трехмерный график можно с помощью окна свойств графика или специальных команд форматирования. Кроме того, для форматирования цветных поверхностей есть дополнительные команды:

- **colormap(gray)** – задает окраску тонами серого цвета;
- **shading interp** – устраняет изображения линий и задает интерполяцию для оттенков цвета поверхности;
- **colorbar** – выводит на экран цветовую шкалу.

Пример

Построить каркасную и цветную параметрическую поверхности параболоида $z = x^2 + y^2$ на отрезке $[-3; 3]$ с шагом 0,5.

Порядок ввода:

```
>> [X,Y]=meshgrid([-3:0.5:3]);  
>> Z=X.^2+Y.^2;  
>> plot3(X,Y,Z)  
>> surf(X,Y,Z)
```

В результате каждого построения получим поверхности, показанные на рис. 9.

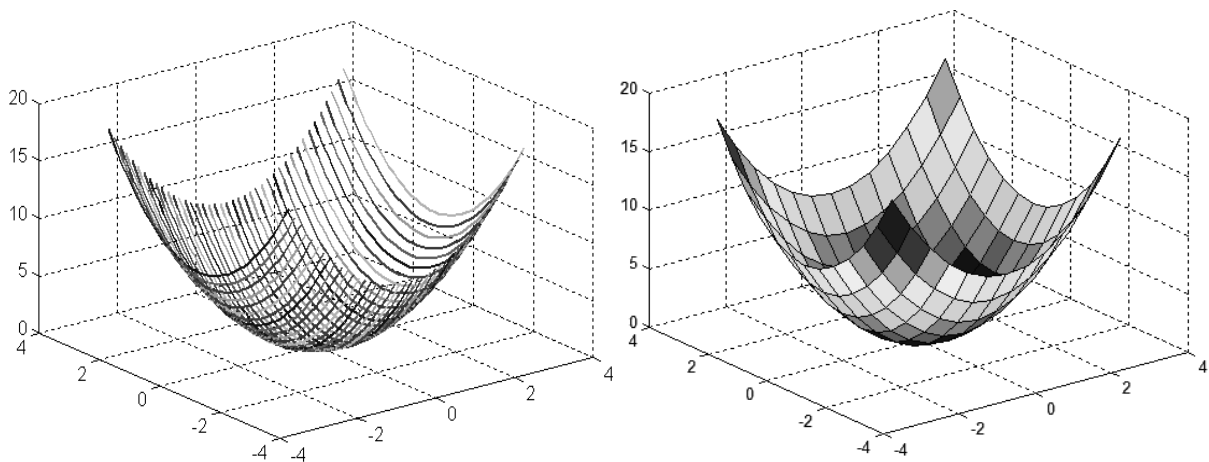
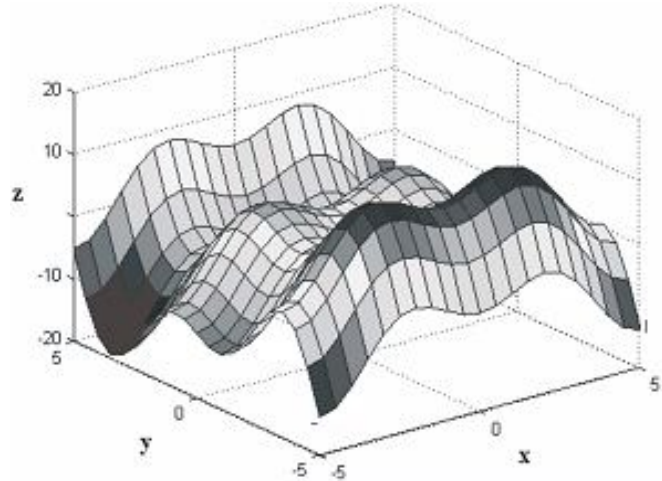
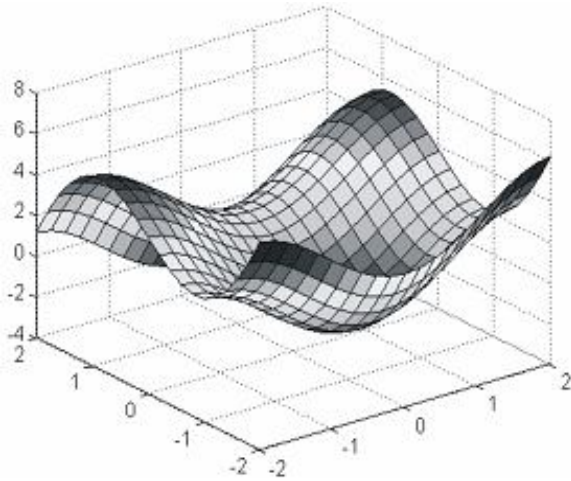


Рис. 9. Поверхности параболоида $z = x^2 + y^2$

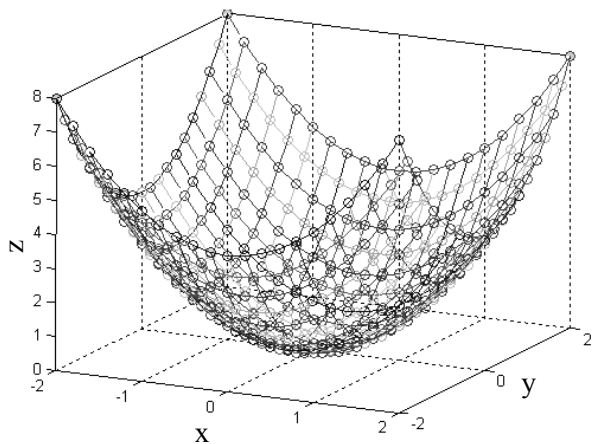
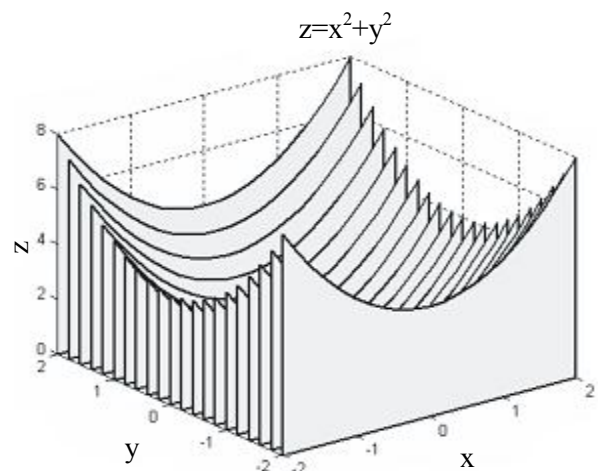
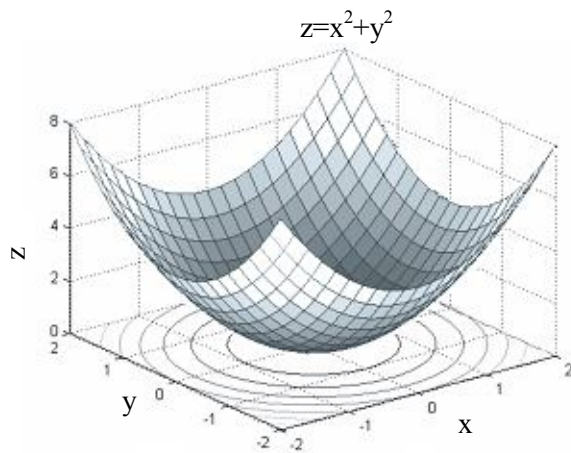
САМОСТОЯТЕЛЬНАЯ РАБОТА № 4

I. Построить цветные поверхности функции $z=2x\sin x+3y\cos y$ на заданных отрезках и отформатировать их по образцу:

- 1) на отрезке $[-2; 2]$, шаг 0,2; 2) на отрезке $[-5; 5]$, шаг 0,5.



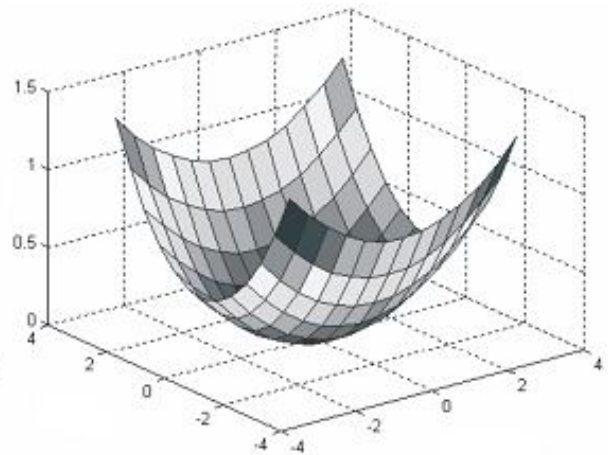
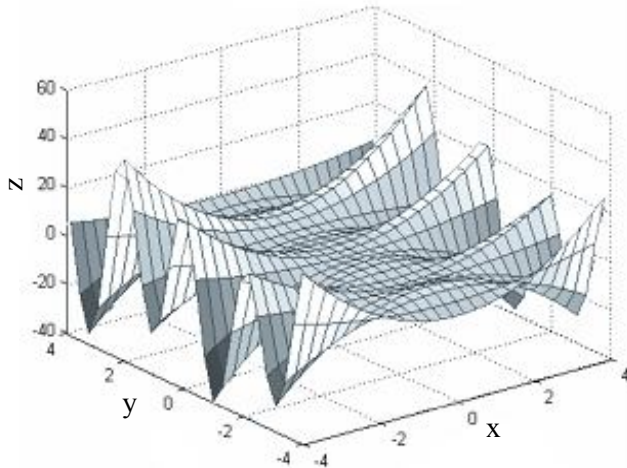
II. Построить с помощью соответствующих команд графики функции $z=x^2+y^2$ на отрезке $[-2; 2]$ с шагом 0,2. Отформатировать графики по образцу:



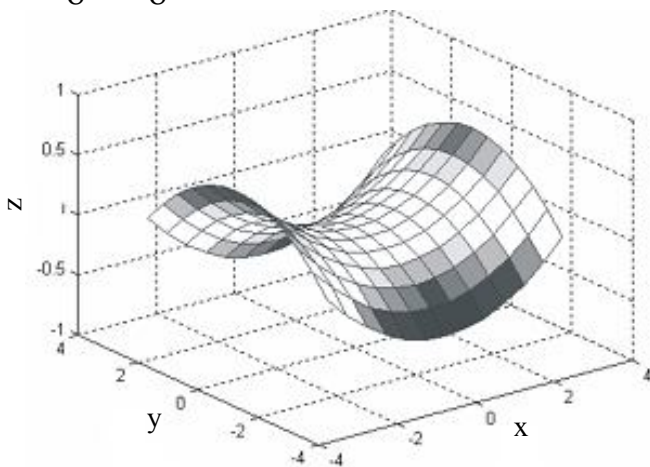
III. Построить цветные поверхности функций на отрезке $[-4; 4]$ с шагом 0,2 и отформатировать их по образцу:

$$1) g = \frac{2x^2 + y}{3} \cdot (3 \cos 3y + \sin y); \quad 2) \frac{x^2}{10} + \frac{y^2}{5} = 2z;$$

$$(x^2 + y)/3 * (\cos 3y + \sin y)$$



$$3) \frac{x^2}{8} - \frac{y^2}{8} = 2z.$$



IV. Построить и отформатировать поверхность $\frac{x^2}{50} + \frac{y^2}{5} + \frac{z^2}{50} = 1$ на отрезке $[-5; 5]$ с шагом 0,5.

6. СИСТЕМЫ ЛИНЕЙНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ (СЛАУ)

Решение СЛАУ относится к самой массовой области применения матричных методов.

6.2. Решение системы линейных уравнений методом Крамера

Для решения системы линейных уравнений методом Крамера необходимо:

1. Задать главную матрицу из коэффициентов при неизвестных.
2. Задать вспомогательные матрицы (в главной матрице заменить поочередно один столбец значениями свободных членов).
3. Вычислить неизвестные системы уравнений, разделив определитель соответствующей вспомогательной матрицы на определитель главной матрицы.

Пример

Решить систему уравнений

$$\begin{cases} 2 \cdot x_1 - x_2 = 4; \\ 5 \cdot x_1 + 2 \cdot x_2 = 3. \end{cases}$$

Порядок ввода:

```
>> A=[2 -1;5 2]
>> A1=[4 -1;3 2]
>> A2=[2 4;5 3]
>> x1=det(A1)/det(A)
>> x2=det(A2)/det(A)
```

В результате получим $x_1=1,222$; $x_2=-1,556$.

6.3. Решение систем уравнений графическим способом

Если система (линейных или нелинейных) уравнений содержит два уравнения и они представляют собой несложные выражения, то можно решить систему графическим способом. Для этого нужно:

1. Выразить одно из неизвестных системы уравнений через другое в каждом из уравнений.
2. Задать диапазон значений аргумента полученных функций.
3. Задать функции.
4. Построить графики функций в одной системе координат.
5. Добавить линии сетки.
6. Найти точку пересечения графиков функций. Её координаты – решение системы уравнений (абсцисса – значение неизвестного, через

которое было выражено другое неизвестное, ордината – значение выраженного неизвестного). При необходимости масштабировать график).

Пример

Решить систему уравнений

$$\begin{cases} 2 \cdot x_1 - x_2 = 4; \\ 5 \cdot x_1 + 2 \cdot x_2 = 3. \end{cases}$$

Выразим второе неизвестное системы уравнений через первое, получим

$$\begin{cases} x_2 = 2 \cdot x_1 - 4; \\ x_2 = 1,5 - 2,5 \cdot x_1. \end{cases}$$

Пусть первая функция $Y1=2X-4$, вторая функция $Y2=1,5-2,5X$.

Порядок ввода:

```
>> X=-2:0.5:2;
>> Y1=2*X-4;
>> Y2=1.5-2.5*X;
>> plot(X,Y1,X,Y2)
>> grid
```

В результате получим график, представленный на рис. 10, масштабируя который можно увидеть, что $x_1=1,222$; $x_2=-1,556$.

6.4. Решение систем уравнений с помощью функции solve

Для решения систем линейных или нелинейных уравнений в MATLAB существует специальная функция **solve**. Чтобы решить систему уравнений с помощью этой функции, нужно:

1. Определить символьные переменные (неизвестные системы уравнений).
2. Вычислить неизвестные по формуле $[x1,x2,\dots]=\text{solve}(\text{'уравнение1'},\text{'уравнение2'},\dots)$.
3. Вывести найденное решение с заданной точностью, используя функцию **вра(переменная, число знаков)**.

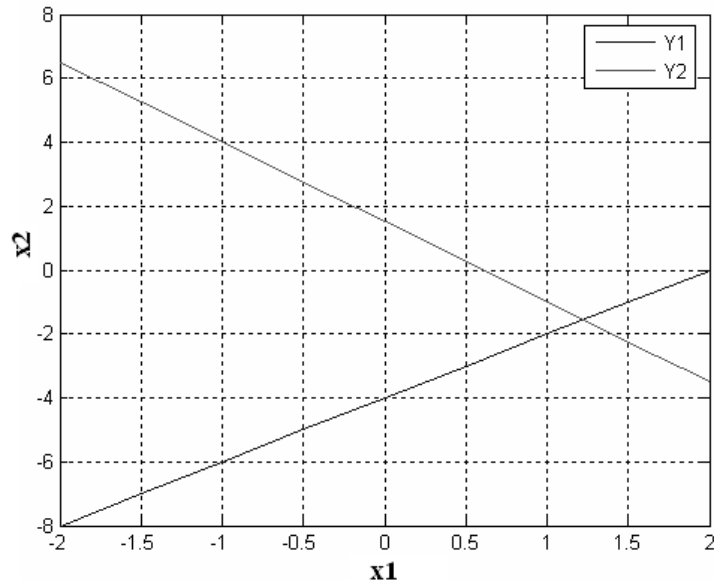


Рис. 10. Графики функций Y_1 и Y_2

Пример

Решить систему уравнений

$$\begin{cases} 2 \cdot x_1 - x_2 = 4; \\ 5 \cdot x_1 + 2 \cdot x_2 = 3. \end{cases}$$

Порядок ввода:

```
>> syms x1 x2
>> [x1,x2]=solve('2*x1-x2=4','5*x1+2*x2=3');
>> vpa(x1,4)
>> vpa(x2,4)
```

В результате получим $x_1=1,222$; $x_2=-1,556$.

САМОСТОЯТЕЛЬНАЯ РАБОТА № 5

I. Решить системы линейных уравнений матричным способом, методом Крамера и с помощью функции solve:

$$1. \begin{cases} 2x_1 + x_2 + x_4 = 8; \\ x_1 - 3x_2 + 2x_3 + 4x_4 = 9; \\ -5x_1 - x_3 - 7x_4 = -5; \\ x_1 - 6x_2 + 2x_3 + 6x_4 = 0. \end{cases} \quad 2. \begin{cases} 2x_1 + 3x_2 - 4x_3 + x_4 = 3,1; \\ 0,1x_1 - 2x_2 - 5x_3 + x_4 = 2; \\ 0,15x_1 - 3x_2 + x_3 - 4x_4 = 1; \\ 10x_1 + 2x_2 - x_3 + 2,1x_4 = -4,7. \end{cases}$$

$$3. \begin{cases} 4x_1 + 0,24x_2 - 0,08x_3 = 8; \\ 0,09x_1 + 3x_2 - 0,15x_3 = 9; \\ 0,04x_1 - 0,08x_2 + 4x_3 = 20. \end{cases} \quad 4. \begin{cases} 3x_1 - x_2 = 5; \\ -2x_1 + x_2 - x_3 = 0; \\ 2x_1 - x_2 + 4x_3 = 15. \end{cases}$$

II. Решить системы линейных уравнений матричным способом, методом Крамера, графическим способом и с помощью функции solve:

$$1. \begin{cases} 2x - y = 4; \\ 3x + 2y = 3. \end{cases} \quad 3. \begin{cases} 2x + 3y = 1; \\ x + 2y = 4. \end{cases}$$

$$2. \begin{cases} 5x + 2y = 10; \\ 2x - y = 7. \end{cases} \quad 4. \begin{cases} 7x + 2y = 2; \\ x + 4y = 1. \end{cases}$$

III. Решить системы нелинейных уравнений графическим способом и с помощью функции solve:

$$1. \begin{cases} 2xy = 16; \\ x^2 + y = 8. \end{cases} \quad 3. \begin{cases} 2x^2 - y = 3; \\ x + 2y = 4. \end{cases}$$

$$2. \begin{cases} x^3 - 2y = 1; \\ x^2 + 3y = 2. \end{cases} \quad 4. \begin{cases} 3x^3 + 2x^2 + y = 5; \\ 2x^2 - 3x + y = 4. \end{cases}$$

IV. Решить системы нелинейных уравнений с помощью функции solve:

$$1. \begin{cases} x^3 - 2y = 1; \\ 3x + 3y^2 = 2. \end{cases} \quad 3. \begin{cases} \sqrt{x+y} + \sqrt[4]{x-y} = 8; \\ \sqrt[4]{x^3 + x^2y - xy^2 - y^3} = 12. \end{cases}$$

$$2. \begin{cases} uvx^2 = 8; \\ vx^2w = 24; \\ x^2wu = 12; \\ u + v + w = x + 4. \end{cases} \quad 4. \begin{cases} x^3 - 2x^2 + 3y + z = 2; \\ 2x^2 - x + y^2 - z^2 = 1; \\ y^3 - 2z^2 + 3z = 1. \end{cases}$$

7. РЕШЕНИЕ УРАВНЕНИЙ

Для решения уравнений вида $f(x)=0$ в MATLAB существует несколько способов.

7.1. Графический способ решения уравнений

Для решения уравнений графическим способом нужно:

1. Объявить символьные переменные (аргумент функции и саму функцию).
2. Задать функцию.
3. Построить график функции.
4. Добавить на график линии сетки.
5. С помощью графической «лупы» масштабировать график, чтобы достичь требуемой точности в определении корней уравнения.
6. Абсциссы точек пересечения графика функции с осью Ox – корни уравнения.

Пример

Решить уравнение $2x^2 - 3x - 1 = 0$.

Порядок ввода:

```
>> syms x f
>> f=2*x^2-3*x-1;
>> ezplot(f)
>> grid
```

В результате получим следующий график (рис. 11).

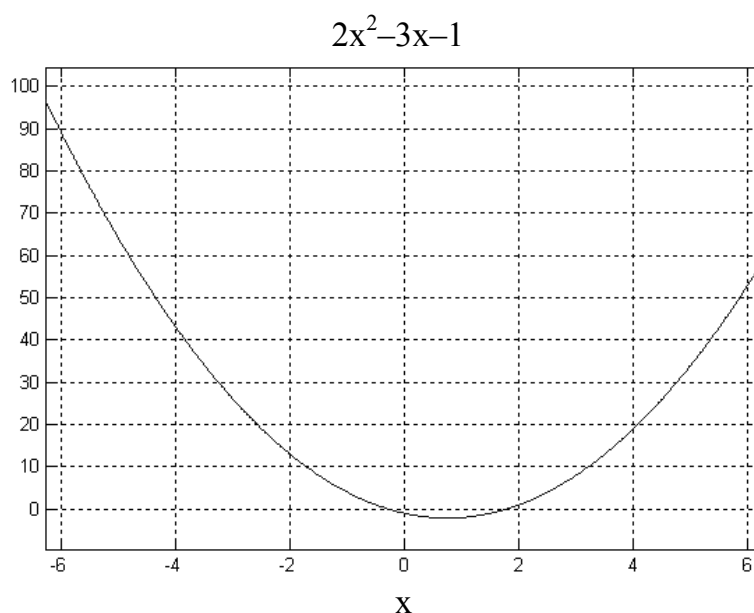


Рис. 11. График функции $f(x)$

Используя кнопку **Zoom In** на панели инструментов и масштабируя график, можно достичь требуемой точности в определении корней уравнения.

В результате получим приближенные значения корней уравнения: $x_1 = -0,28$; $x_2 = 1,78$.

7.2. Решение уравнений с помощью функции solve

Для решения уравнений, заданных символьными переменными, используется встроенная функция **solve**, позволяющая найти решение в аналитической форме, и функция **vpa** для численной оценки с контролируемой точностью найденных решений.

Для решения уравнения вида $f(x)=0$ с помощью функции **solve** нужно:

1. Объявить символьные переменные (аргумент функции и саму функцию).
2. Задать функцию.
3. Найти решение в аналитической форме с помощью функции **solve(функция, аргумент)**.
4. Вывести результат с заданной точностью с помощью функции **vpa(переменная, число знаков)**.

Пример

Решить уравнение $2x^2 - 3x - 1 = 0$.

Порядок ввода:

```
>> syms x f
>> f=2*x^2-3*x-1;
>> r=solve(f,x)
>> vpa(r,5)
```

В результате получим $x_1 = -0,2808$; $x_2 = 1,7808$.

7.3. Нахождение корней полинома

Если функция $f(x)$ является полиномом (многочленом n -й степени), то найти корни уравнения $f(x)=0$ можно с помощью функции **roots**. Для этого нужно:

1. Задать вектор коэффициентов полинома (начиная со старшего).
2. Вычислить корни с помощью функции **roots(вектор)**.

Пример

Найти корни полинома $2x^2-3x-1=0$.

Порядок ввода:

```
>> a=[2 -3 -1];
```

```
>> x=roots(a)
```

В результате получим $x_1=-0,2808$; $x_2=1,7808$.

7.4. Нахождение минимума функции

1. Нахождение минимума функции одной переменной на отрезке $[a; b]$:

x=fminbnd('f ', a, b)

Пример

Найти минимум функции на отрезке $[0,2; 1]$:

$$f(x) = \frac{1}{(x-0,3)^2 + 0,01} + \frac{1}{(x-0,9)^2 + 0,04} - 6.$$

Порядок ввода:

```
>> x=fminbnd('1/((x-0.3)^2+0.01)+1/((x-0.9)^2+0.04)-6',0.2,1)
```

В результате получим $x=0,6370$.

2. Нахождение минимума функции нескольких переменных:

x=fminsearch('f ',x0),

где x_0 – начальное приближение (может быть скаляром, вектором или матрицей).

Пример

Найти минимум функции $f(x) = 100 \cdot (x_2 - x_1^2)^2 + (1 - x_1)^2$.

Порядок ввода:

```
>> x=fminsearch('100*(x(2)-x(1)^2)^2+(1-x(1))^2',[-1.2,1])
```

В результате получим

```
x =  
    1.0000    1.0000
```

3. Нахождение нулей функции одной переменной:

x=fzero('f',x0)

где x0 – начальное приближение или интервал поиска.

Пример

Решить уравнение $x \cdot 2^x = 1$.

Порядок ввода:

```
>> x=fzero('x*2^x-1',[0,3])
```

В результате получим $x=0,6412$.

САМОСТОЯТЕЛЬНАЯ РАБОТА № 6

I. Решить уравнения графически и с помощью функции **solve**:

1) $|x + 5| - |2 \cdot x - 1| = 3$;

2) $\sqrt{|2x + 1|} - 1 = 0$;

3) $\ln(x + 2) = 2$;

4) $|x + 1| + |x + 2| - 2 = 0$;

5) $x^3 + 2 \cdot |x - 1|$.

II. Решить уравнения графически, с помощью функции **solve** и с помощью функции **roots**:

1) $2x^2 - 3x = 1$;

4) $10x^3 - 3x^2 - 2x + 0,5 = 0$;

2) $3x^3 - 8x^2 + 2x + 2 = 0$;

5) $2x^2 + x - 2 = 0$.

3) $x^4 + 5x^2 + 6x - 20 = 0$;

КОНТРОЛЬНАЯ РАБОТА № 1

Вариант 1

I. Вычислить значения выражений при $x=3$, $b=2,3$:

1) $a = \frac{b^2 \cdot e^{|2x|} + \ln|x|}{x\sqrt{x+b}}$;

2) $f = x + \cos^2 5\pi - 7 \sin 3\pi$.

II. Вычислить значения функций на соответствующем отрезке:

1) $x \in [-2; 2]$, шаг 0,5, $f = |x+1| \cdot \sin x$;

2) $x \in [-10; 8]$, шаг 2, $y = 3x^2 + 2x - 1$.

III. Выполнить операции над матрицами:

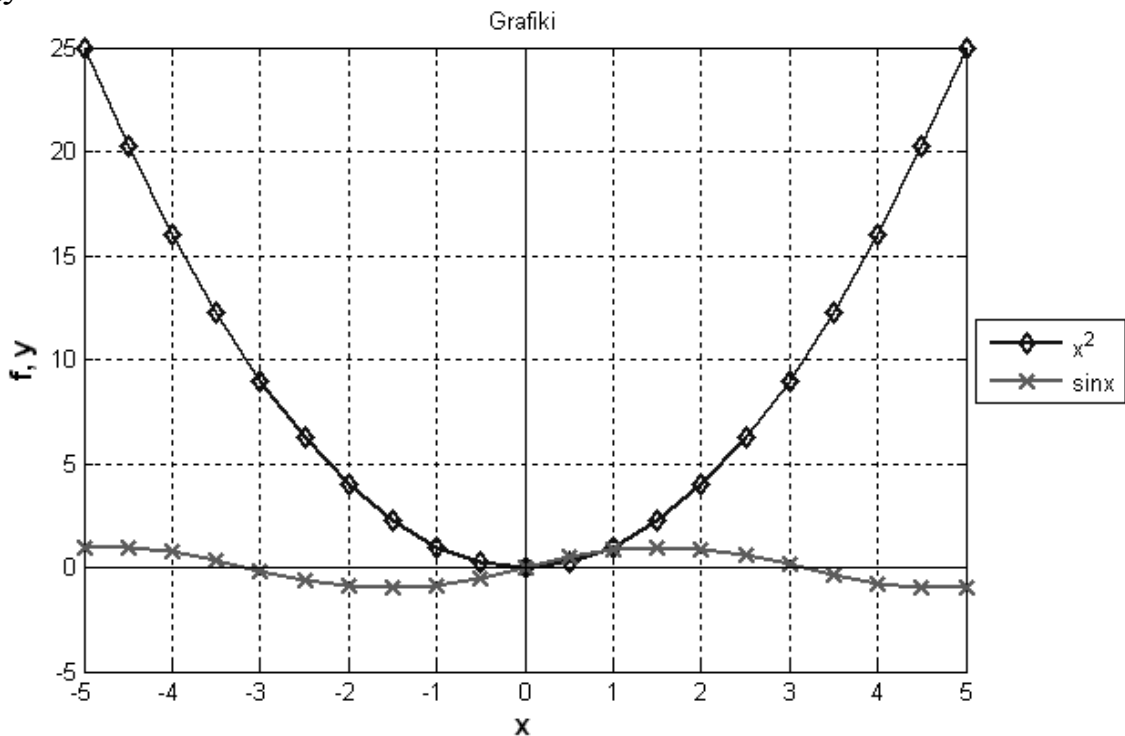
$$A = \begin{pmatrix} 2 & -1 & 1 \\ 1 & 4 & 0 \\ -2 & 3 & 2 \end{pmatrix}; \quad B = \begin{pmatrix} 3 & 0 & 1 \\ 2 & 1 & 2 \\ 3 & 2 & 3 \end{pmatrix}.$$

1) $2A+B^{-1}$;

2) $A \cdot B^T$;

3) $|A|$.

IV. Построить в одной системе координат графики функций $f=x^2$ и $y=\sin x$ при $x \in [-5; 5]$ с шагом 0,5. Отформатировать графики по образцу.



V. Решить уравнение $\sqrt{|2x+1|} - 1 = 0$.

VI. Найти корни полинома $10x^3 - 3x^2 - 2x + 0,5 = 0$.

VII. Решить систему линейных уравнений

$$\begin{cases} 10x_1 + x_2 + x_3 = 12; \\ -2x_1 + 10x_2 + x_3 = 13; \\ -2x_1 + 2x_2 + 10x_3 = 14. \end{cases}$$

VIII. Решить систему нелинейных уравнений

$$\begin{cases} 2x^2 - y = 3; \\ x + 2y = 4. \end{cases}$$

Вариант 2

I. Вычислить значения выражений при $x=-3$, $b=2$:

1) $d = b(\cos^3 x + e^{(b+3)})$;

2) $s = \frac{b\sqrt{x} - x^2 \sin 2\pi}{\ln|x| + 2}$.

II. Вычислить значения функций на соответствующем отрезке:

1) $x \in [-3;3]$, шаг 0,7, $f = 3|x-1|^2$;

2) $x \in [-8;10]$, шаг 3, $y = 3x \cdot \cos(x+2)$.

III. Выполнить операции над матрицами:

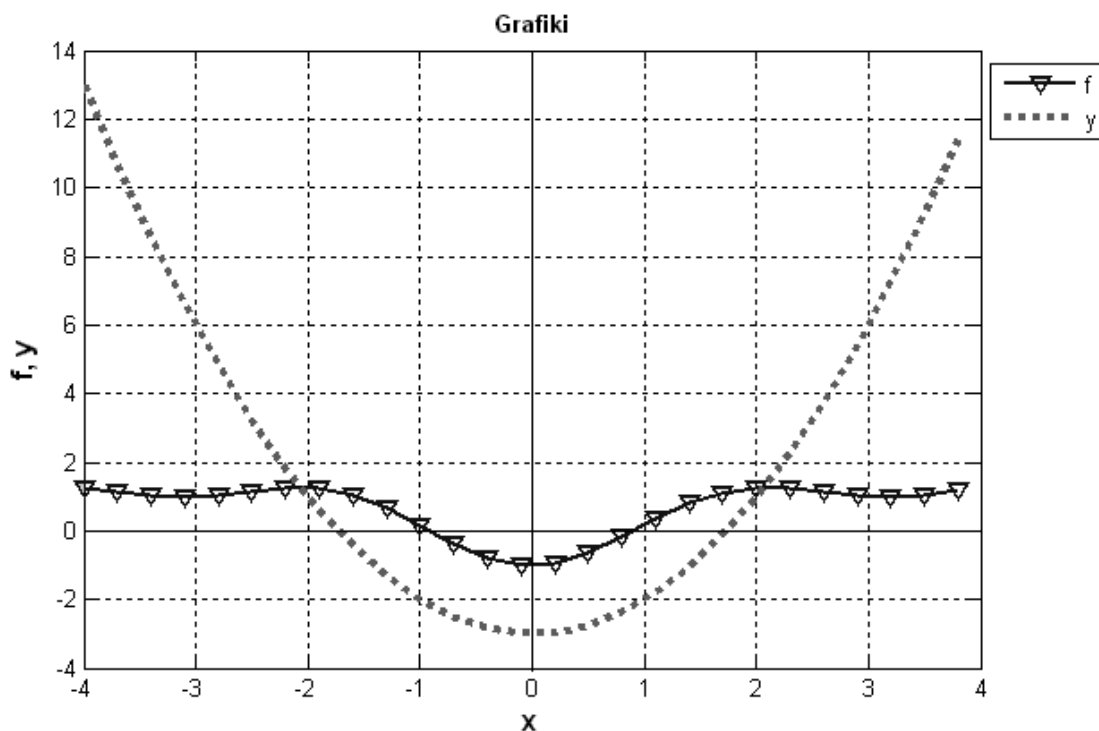
$$A = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 3 & 2 \\ -2 & 4 & -1 \end{pmatrix}; \quad B = \begin{pmatrix} 1 & 2 & -2 \\ 2 & -1 & 0 \\ 4 & 2 & 3 \end{pmatrix}.$$

1) $A^{-1} + 3B$;

2) $A^T \cdot B$;

3) $|B|$.

IV. Построить в одной системе координат графики функций $f = \sin x^2 - \cos x$ и $y = x^2 - 3$ при $x \in [-4;4]$ с шагом 0,3. Отформатировать графики по образцу.



V. Решить уравнение $|x + 1| + |x + 2| - 2 = 0$.

VI. Найти корни полинома $3x^3 - 8x^2 + 2x + 2 = 0$.

VII. Решить систему линейных уравнений

$$\begin{cases} 6x_1 - x_2 - x_3 = 11,33; \\ -x_1 + 6x_2 - x_3 = 32; \\ -x_1 - x_2 + 6x_3 = 42. \end{cases}$$

VIII. Решить систему нелинейных уравнений

$$\begin{cases} x^3 - 2y = 1; \\ 3x + 3y^2 = 2. \end{cases}$$

8. ВЫЧИСЛЕНИЕ ИНТЕГРАЛОВ

Численное интегрирование (историческое название: *квадратура*) – вычисление значения определённого интеграла (как правило, приближённое), основанное на том, что величина интеграла численно равна площади криволинейной трапеции, ограниченной осью абсцисс, графиком интегрируемой функции и отрезками прямых $x = a$ и $x = b$, где a и b – пределы интегрирования (рис. 12).

Для вычисления интегралов в MATLAB можно использовать функции:

- **int(f,x)** – вычисляет неопределённый интеграл;

• **int(f,x,a,b)** – вычисляет определенный интеграл, где f – подынтегральная функция, x – переменная интегрирования, a и b – пределы интегрирования.

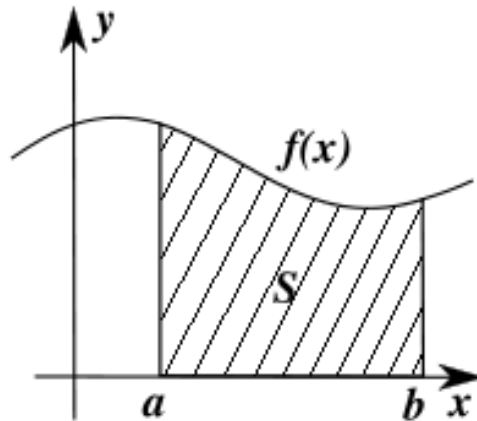


Рис. 12. Определённый интеграл как площадь фигуры

Пример 1

Вычислить неопределенный интеграл $\int x^3 dx$.

Порядок ввода:

```
>> syms x  
>> f=x^3;  
>> int(f,x)
```

В результате получим выражение $1/4*x^4$.

Пример 2

Вычислить определенный интеграл $\int_1^3 x^3 dx$.

Порядок ввода:

```
>> syms x  
>> f=x^3;  
>> int(f,x,1,3)
```

В результате получим значение определенного интеграла 20. Если MATLAB не выводит сразу численное значение определенного интеграла, то используйте функцию **vpa**, например,

```
>> vpa(int(f,x,1,3),3).
```

Для вычисления двойных, тройных и т.д. интегралов необходимо использовать функцию **int** несколько раз.

Пример 1

Вычислить двойной интеграл $\int \int 2x^3 y dx dy$.

Порядок ввода:

```
>> syms x y
>> f=2*x^3*y;
>> int(int(f,x),y)
```

В результате получим выражение $1/4*x^4*y^2$.

Пример 2

Вычислить двойной интеграл $\int_{-1}^2 \int_{-1}^3 2x^3 y dx dy$.

Порядок ввода:

```
>> syms x y
>> f=2*x^3*y;
>> int(int(f,x,1,3),y, -1,2)
```

В результате получим 60.

САМОСТОЯТЕЛЬНАЯ РАБОТА № 7

I. Вычислить неопределенные интегралы:

1) $\int \cos x dx$;

4) $\int (3x + \ln x) dx$;

7) $\int \sqrt{x^2 + 1} dx$;

2) $\int x^2 dx$;

5) $\int (x^3 + 3x^2 + 1) dx$;

8) $\int \frac{\sqrt{x+1}}{x^2 + 2x} dx$.

3) $\int (e^x - x) dx$;

6) $\int \frac{1}{x} dx$;

II. Вычислить определенные интегралы:

$$\begin{array}{ll} 1) \int_1^3 \sin x dx; & 4) \int_0^5 (0,5^x \cdot \cos x) dx; \\ 2) \int_1^3 (x^2 - \cos 2x) dx; & 5) \int_0^4 (\sin(2x + 3) - 2 \cos 5x) dx; \\ 3) \int_1^2 (\sin x + \frac{x^2}{2}) dx; & 6) \int_{-5}^0 (e^x \sin(x + 3)) dx. \end{array}$$

III. Вычислить двойные интегралы:

$$\begin{array}{ll} 1) \iint (2x + \frac{y^2}{2}) dx dy; & 3) \int_{-1}^2 \int_{-2}^1 (4x^3 + 3y^2) dx dy; \\ 2) \iint (18x^2 y^2 + 32x^3 y) dx dy; & 4) \iint (18x^2 y^2 + 32x^3 y) dx dy. \end{array}$$

IV. Вычислить тройные интегралы:

$$\begin{array}{ll} 1) \iiint_{-11}^1 \int_0^2 \int_3^3 ((x^2 - 2)y + 3z) dx dy dz; & 2) \iiint_{12}^3 \int_{-1}^3 \int_1^1 (\sin x - 2y + e^z) dx dy dz. \end{array}$$

9. ВЫЧИСЛЕНИЕ ПРЕДЕЛОВ

Вычисление пределов от символьных выражений производится с помощью встроенной функции **limit**:

- **limit(f)** – вычисление предела функции f при стремлении аргумента функции к нулю;
- **limit(f,a)** – вычисление предела функции f при стремлении аргумента функции к числу a ;
- **limit(f,x,a,'left')** – вычисление предела функции f при стремлении переменной x к числу a слева;
- **limit(f,x,a,'right')** – вычисление предела функции f при стремлении переменной x к числу a справа;
- **limit(f,y,a)** – вычисление предела функции нескольких переменных f при стремлении переменной y к числу a .

Примечание. Символ бесконечность (∞) в MATLAB записывается как **inf**. Неопределенное значение в MATLAB записывается как **NaN**.

Пример 1

Вычислить $\lim_{x \rightarrow 0} \frac{\sin x}{x}$.

Порядок ввода:

```
>> syms x
>> y=sin(x)/x;
>> limit(y)
```

В результате получим 1.

Пример 2

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

Порядок ввода:

```
>> syms x
>> f=(2*x^3+3*x^2+1)/(x^2-2*x+3);
>> vpa(limit(f,2),3)
```

В результате получим 9,67.

Пример 3

Вычислить $\lim_{x \rightarrow \infty} \left(1 + \frac{1}{x}\right)^x$.

Порядок ввода:

```
>> y=(1+1/x)^x;
>> limit(y,inf)
```

В результате получим $\exp(1)$, т.е. число e .

Пример 4

Вычислить $\lim_{x \rightarrow 0^-} \frac{1}{x}$.

Порядок ввода:

```
>> y=1/x;
>> limit(y,x,0,'left')
```

В результате получим **-inf**, т.е. минус бесконечность.

Пример 5

Вычислить $\lim_{x \rightarrow 0^+} \frac{1}{x}$.

Порядок ввода:

```
>> y=1/x;  
>> limit(y,x,0,'right')
```

В результате получим **inf**, т.е. бесконечность.

Пример 6

Вычислить $\lim_{h \rightarrow 0} \left(\frac{\sin(x+h) - \sin x}{h} \right)$.

Порядок ввода:

```
>> syms x h  
>> y=(sin(x+h)-sin(x))/h;  
>> limit(y,h,0) % Вычисление предела по переменной h
```

В результате получим $\cos x$.

10. ДИФФЕРЕНЦИРОВАНИЕ ФУНКЦИИ

Дифференцирование функций в MATLAB осуществляется с помощью функции **diff**.

Для функций одной переменной:

- **diff(f)** – вычисляет первую производную функции f ;
- **diff(f,k)** – вычисляет производную k -го порядка функции f .

Для функций нескольких переменных:

- **diff(f,x)** – вычисляет первую производную функции f по переменной x ;
- **diff(f,x,k)** – вычисляет производную k -го порядка функции f по переменной x .

Пример 1

Вычислить производную функции $y=2x^3-3x^2+3$.

Порядок ввода:

```
>> syms x
>> y=2*x^3-3*x^2+3;
>> diff(y)
```

В результате получим $6x^2 - 6x$.

Пример 2

Найти производную функции $y=\sin(x+h)$ по переменной x .

Порядок ввода:

```
>> syms x h
>> y=sin(x+h);
>> diff(y,x)
```

В результате получим $\cos(x+h)$.

Пример 3

Найти производную функции $y=\frac{\sin(x+h)}{x}$ по переменной h .

Порядок ввода:

```
>> syms x h
>> y=sin(x+h)/x;
>> diff(y,h)
```

В результате получим $\cos(x+h)/x$.

Пример 4

Найти вторую производную функции $y=5/x$.

Порядок ввода:

```
>> syms x
>> diff(5/x,2)
```

В результате получим $10/x^3$.

Пример 5

Найти вторую производную функции $y=3x^3h-2h^2x^2+3$ по переменной x .

Порядок ввода:

```
>> syms x h
>> y=3*x^3*h-2*h^2*x^2+3
>> diff(y,x,2)
```

В результате получим $18xh-4h^2$.

Пример 6

Найти третью производную функции $y=3h^2 \cdot \ln(x) + 3e^h$ по переменной h .

Порядок ввода:

```
>> syms x h
>> diff(3*h^2*log(x)+3*exp(h),h,3)
```

В результате получим $3e^h$.

САМОСТОЯТЕЛЬНАЯ РАБОТА № 8

I. Вычислить пределы функций:

- | | |
|---|---|
| 1) $\lim_{x \rightarrow 3} \frac{x-3}{x^2-4x+3}$; | 6) $\lim_{x \rightarrow 4^+} \frac{2-\sqrt{x}}{\sqrt{5x+1}-5}$; |
| 2) $\lim_{x \rightarrow 0} \frac{\sqrt{7+x}-\sqrt{7-x}}{5x}$; | 7) $\lim_{x \rightarrow 3^-} \frac{\sqrt{x^2-3}-1}{\sqrt{x^2-5}-2}$; |
| 3) $\lim_{x \rightarrow 0} \left(\frac{1}{\sin x} - \frac{1}{\operatorname{tg} x} \right)$; | 8) $\lim_{x \rightarrow \pi} \frac{\sin 5x}{\sin 6x}$; |
| 4) $\lim_{x \rightarrow \infty} \frac{x+2}{x^2-x-6}$; | 9) $\lim_{x \rightarrow 3} \frac{3y-\sqrt{x+9}}{x+y}$; |
| 5) $\lim_{x \rightarrow \frac{\pi}{2}} (\sin x)^{\operatorname{tg} x}$; | 10) $\lim_{y \rightarrow -1^+} \frac{3x+2y}{5x-3y}$. |

II. Вычислить производные функций:

1) $y(x) = x^2 + 3x + 1$;

5) $y(x) = x - \cos x$;

2) $y(x) = \frac{2x^2 + 1}{\sqrt{x+2}}$;

6) $y(x) = \sin x^2 + x^2$;

3) $y(x) = \arcsin 2x$;

7) $y(x) = 0,5^x \cdot \cos x$;

4) $y(x) = \sqrt{x-3} + \frac{\operatorname{tg} 2x}{4}$;

8) $y(x, z) = 5(\sin x - \cos 5z)$;

9) $y(x, z) = \sin(z+3) + 2^x$.

III. Вычислить производные старших порядков:

1) $y(x) = x^2 - \cos x$, второго порядка;

2) $y(x) = e^{-2x} + x^3$, третьего порядка;

3) $y(x) = e^x + x^4 / 3$, шестого порядка.

11. РЕШЕНИЕ ОБЫКНОВЕННЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ

Дифференциальные уравнения часто встречаются при построении моделей, описывающих динамику объектов исследования.

Для решения дифференциальных уравнений в MATLAB существует функция **dsolve**, которая имеет следующие форматы обращения:

1. $y = \text{dsolve}('Dy(x)'),$

где $Dy(x)$ – уравнение; y – возвращаемые функцией **dsolve** решения.

2. $y = \text{dsolve}('Dy(x)', 'HY'),$

где $Dy(x)$ – уравнение; HY – начальные условия.

Первая производная функции обозначается Dy , вторая производная – D^2y и так далее.

Функция **dsolve** предназначена также для решения системы дифференциальных уравнений. В этом случае она имеет следующий формат обращения:

$[f,g] = \text{dsolve}('Df(x), Dg(x)', 'HY'),$

где $Df(x), Dg(x)$ – система уравнений; HY – начальные условия.

Пример 1

Решить дифференциальное уравнение $\frac{dx}{dt} = -0,5x$ с начальным условием $x(0)=10$. Построить график решения в интервале $[-0,5; 7]$.

Порядок ввода:

```
>> x=dsolve('Dx=-0.5*x','x(0)=10')  
>> ezplot(x,[-0.5,7]);  
>> grid
```

В результате получим функцию $x=10e^{-1/2t}$ и график (рис. 13).

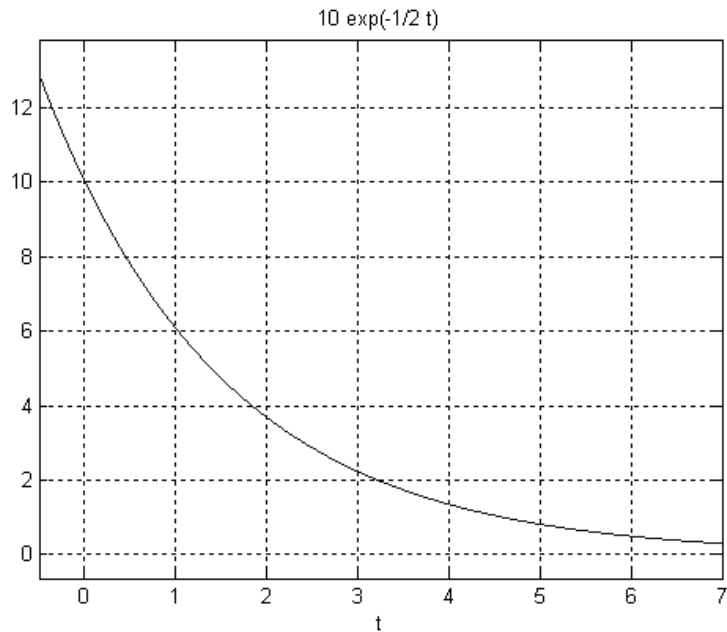


Рис. 13. График функции-решения уравнения

Пример 2

Решить систему однородных дифференциальных уравнений

$$\begin{cases} \frac{dx_1}{dt} = -0,5x_2; \\ \frac{dx_2}{dt} = 3x_1. \end{cases}$$

с начальными условиями $x_1(0)=0$, $x_2(0)=1$. Построить график решения в интервале $[-0,5; 13]$.

Порядок ввода:

```
>> [x1,x2]=dsolve('Dx1=-0.5*x2','Dx2=3*x1','x1(0)=0','x2(0)=1')  
>> ezplot(x1,0,13)  
>> grid
```

```
>> hold on
>> ezplot(x2,[0,13])
```

В результате получим функции $x_1 = -\sqrt{6}/6 \cdot \sin(\sqrt{6} \cdot t/2)$ и $x_2 = \cos(\sqrt{6} \cdot t/2)$. Графики функций показаны на рис. 14.

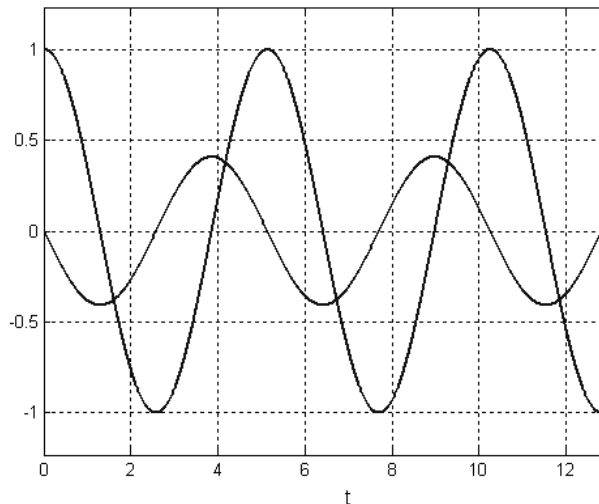


Рис. 14. Графики функций x_1 и x_2

Пример 3

Решить систему неоднородных дифференциальных уравнений

$$\begin{cases} \frac{dx_1}{dt} = -3x_1 + 12; \\ \frac{dx_2}{dt} = 2,5x_1 - 1,25x_2. \end{cases}$$

с нулевыми начальными условиями и построить график решения в интервале $[0; 5]$ для первой x_1 координаты и в интервале $[0; 9]$ для второй x_2 координаты.

Порядок ввода:

```
>> [x1,x2]=dsolve('Dx1=-3*x1+12','Dx2=2.5*x1-1.25*x2', ...
'x1(0)=0','x2(0)=0')
>> ezplot(x1,[0,5])
>> grid
>> hold on
>> ezplot(x2,[0,9])
```

В результате решения получим функции $x_1 = 4 - 4e^{-3t}$, $x_2 = 8 + 40/7 \cdot e^{-3t} - 96/7 \cdot e^{-5/4t}$ и график (рис. 15).

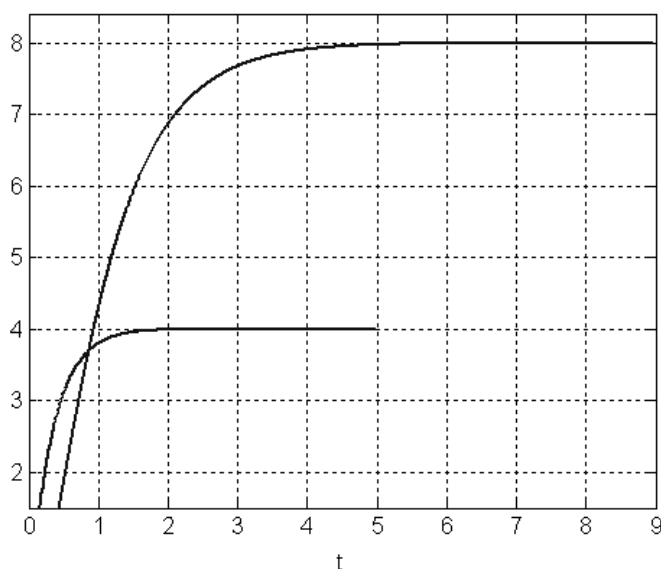


Рис. 15. Графики функций x_1 и x_2

САМОСТОЯТЕЛЬНАЯ РАБОТА № 9

I. Решить дифференциальные уравнения при заданном начальном условии и построить графики решения любых трех уравнений:

1) $\frac{dx}{dt} = t \cos t + \frac{x}{t}$, $x(1) = 0$;

5) $\frac{dx}{dt} - x \cdot \operatorname{tg} t = \frac{1}{\cos t}$, $x(0) = 0$;

2) $\frac{dx}{dt} = \frac{1}{e^{t-x}}$, $x(1) = 1$;

6) $(x+t) \frac{dx}{dt} = 1$, $x(-1) = 0$;

3) $\frac{dx}{dt} = \frac{1+x^2}{1+t^2}$, $x(0) = 1$;

7) $(1+e^t)x \frac{dx}{dt} = e^t$, $x(0) = 1$;

4) $\frac{dx}{dt} + x = \cos t$, $x(0) = \frac{1}{2}$;

8) $\frac{dx}{dt} = t^2 x^4 - \frac{x}{t}$, $x(1) = \frac{1}{\sqrt[3]{3}}$.

II. Решить дифференциальные уравнения старшего порядка при заданных начальных условиях:

1) $\frac{d^2x}{dt^2} + 4x = 0$; $x(0) = 0$; $\frac{dx}{dt}(0) = 2$;

- 2) $\frac{d^2x}{dt^2} + 3\frac{dx}{dt} + 2x = 0; x(0) = 1; \frac{dx}{dt}(0) = -1;$
- 3) $\frac{d^2x}{dt^2} + 2\frac{dx}{dt} = 0; x(0) = 1; \frac{dx}{dt}(0) = 0;$
- 4) $t\frac{d^2x}{dt^2} = \frac{dx}{dt}; x(0) = 0; \frac{dx}{dt}(0) = 0;$
- 5) $(1+t^2)\frac{d^2x}{dt^2} - 2t\frac{dx}{dt} = 0; x(0) = 0; \frac{dx}{dt}(0) = 3;$
- 6) $\frac{d^2x}{dt^2}(1+\ln t) + \frac{1}{t}\frac{dx}{dt} = 2 + \ln t; x(1) = \frac{1}{2}; \frac{dx}{dt}(1) = 1.$

III. Решить системы дифференциальных уравнений при заданных начальных условиях и построить графики решения:

$$1) \begin{cases} \frac{dx_1}{dt} = 2x_1 - 1; \\ \frac{dx_2}{dt} = 3x_1 + 2x_2 + 1; \\ x_1(0) = 1, x_2(0) = 1, \end{cases} \quad 2) \begin{cases} \frac{dx_1}{dt} = x_1 + x_2; \\ \frac{dx_2}{dt} = 2x_1 + 2; \\ x_1(1) = 0, x_2(1) = 1. \end{cases}$$

КОНТРОЛЬНАЯ РАБОТА № 2

I. Решить уравнение $|x-1| + 2 \cdot |x+3| - 7 = 0.$

II. Найти корни полинома $x^3 - 2x^2 - 5,5x + 1 = 0.$

III. Решить систему линейных уравнений

$$\begin{cases} 6x_1 - x_2 - x_3 = 11,33; \\ -x_1 + 6x_2 - x_3 = 32; \\ -x_1 - x_2 + 6x_3 = 42. \end{cases}$$

IV. Решить систему нелинейных уравнений

$$\begin{cases} x^3 - 2y = 1; \\ 3x + 3y^2 = 2. \end{cases}$$

V. Вычислить интегралы:

$$1) \int 3x^3 dx; \quad 2) \int_1^3 (e^{2x+1} + 1) dx;$$

$$3) \iint (x^2 + 2y \sin x) dx dy; \quad 4) \int_{-1}^1 \int_{-1}^1 (2y + 3x^2 y) dx dy.$$

VI. Вычислить пределы:

$$1) \lim_{x \rightarrow 1} \frac{2x^2 + 3x + 4}{x^3 + 2x^2 + 3}; \quad 3) \lim_{x \rightarrow \infty} \frac{2^x + 1}{x^2};$$

$$2) \lim_{x \rightarrow 0} \frac{\sqrt{3x^2 + 1}}{2x^2 + 5}; \quad 4) \lim_{x \rightarrow 2^-} \frac{\sin x + 1}{\operatorname{tg} x}.$$

VII. Вычислить производные функций:

- 1) $y(x) = 2x^3 + 1$, первого порядка;
- 2) $y(x) = \frac{2x}{3x^2 + 2}$, второго порядка;
- 3) $y(x) = e^{3x} + x\sqrt{x}$, первого и третьего порядка;
- 4) $z(x, y) = 3 \sin x^2 + x \cos y$, по переменной x и по переменной y .

VIII. Решить дифференциальные уравнения при заданном начальном условии и построить графики решения:

- 1) $\frac{dx}{dt} = 2x + 3t^2$, $x(0) = 1$;
- 2) $\frac{dx}{dt} = 2 \sin x + 2$, $x(0) = 0$;
- 3) $\frac{d^2x}{dt^2} + 2 \frac{dx}{dt} + 2x + 2 = 0$, $\frac{dx}{dt}(1) = 1$, $x(0) = 0$.

IX. Решить следующую систему обыкновенных дифференциальных уравнений с заданными начальными условиями:

$$\begin{cases} \frac{dx_1}{dt} = -x_1 + 2, & x_1(0) = 10; \\ \frac{dx_2}{dt} = 2x_1^2 - 0,5x_2, & x_2(0) = 5. \end{cases}$$

12. ПРОГРАММИРОВАНИЕ В СРЕДЕ MATLAB

12.1. М-файлы

Самым удобным способом выполнения команд MATLAB является использование *М-файлов*, в которых можно набирать команды, выполнять их все сразу или частями, сохранять в файле и использовать в дальнейшем. Для работы с М-файлами предназначен редактор М-файлов. При помощи этого редактора можно создавать собственные функции и вызывать их, в том числе и из командной строки.

Для подготовки, редактирования и отладки М-файлов служит специальный многооконный редактор. Он выполнен как типичное приложение Windows. Редактор можно вызвать командой **edit** из командной строки или командой **New – Script (Function)** из меню **File**.

Выполнение М-файла: меню **Debug – Save and Run (F5)**.

Открытие существующего М-файла:

- 1) меню *File - Open* рабочей среды либо редактора М-файлов.
- 2) команда MATLAB *edit* из командной строки, указав в качестве аргумента имя файла, например:

```
>> edit mydemo
```

Типы М-файлов:

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

12.1.1. Файл-сценарий

Файл-сценарий называется также *Script-файлом* или просто *скриптом*.

Файл-сценарий имеет весьма простую структуру:

1. % основной комментарий, если необходимо.
2. % дополнительный комментарий, если необходимо.
3. Тело программы с любыми допустимыми выражениями.

Все переменные, объявленные в файл-программе, становятся доступными в рабочей среде после ее выполнения. Переменные, определенные в одной файл-программе, можно использовать в других файл-программах и в командах, выполняемых из командной строки.

Выполнение команд, содержащихся в файл-сценарии:

1. Из редактора М-файлов так, как описано выше.

2. Из командной строки или другого файл-сценария, при этом в качестве команды используется имя М-файла.

12.1.2. Файл-функция

Файл-функция является аналогом подпрограммы типа *function* языков программирования.

Файл-функция имеет входные параметры, список которых указывается в круглых скобках. Используемые в файл-функции переменные являются локальными переменными, изменение значений которых в теле функции никоим образом не влияет на значения, которые те же самые переменные могут иметь за пределами функции. Т.е. локальные переменные могут иметь те же имена (идентификаторы), что и глобальные переменные.

Структура М-файла функции с одним выходным параметром имеет вид:

```
function var = f_name (Список параметров)
% основной комментарий, если необходимо.
% дополнительный комментарий, если необходимо.
Тело программы с любыми выражениями.
var = выражение
```

Пример

Предположим, что в вычислениях часто необходимо использовать функцию $e^{-x} \cdot \sqrt{\frac{x^2 + 1}{x^4 + 0,1}}$.

```
function f = myfun(x)
f = exp(-x)*sqrt((x^2+1)/(x^4+0.1));
```

Теперь созданную функцию можно использовать так же, как и встроенные *sin*, *cos* и другие, например из командной строки:

```
>> y = myfun(1.3)
y =
    0.2600
```

Вызов собственных функций может осуществляться из файл-сценария и из другой файл-функции.

Каталог, в котором содержатся файл-функции, должен быть текущим, или путь к нему должен быть добавлен в пути поиска, иначе MATLAB просто не найдет функцию или вызовет вместо нее другую с тем же именем (если она находится в каталогах, доступных для поиска).

Файл-функция, приведенная в листинге, имеет один существенный недостаток. Попытка вычисления значений функции от массива приводит к ошибке, а не к массиву значений, как это происходит при вычислении встроенных функций.

Устраним этот недостаток:

```
function f = myfun(x)
f = exp(-x) .* sqrt((x.^2+1)./(x.^4+0.1));
```

Теперь аргументом функции *myfun* может быть как число, так и вектор или матрица значений, например:

```
>> x = [1.3 7.2];
>> y = myfun(x)
y =
0.2600 0.0001
```

Переменная *y*, в которую записывается результат вызова функции *myfun*, автоматически становится вектором нужного размера.

MATLAB предоставляет еще одну возможность работы с файл-функциями – использование их в качестве аргументов некоторых команд. Например, для построения графика служит специальная функция *fplot*, заменяющая последовательность команд, приведенную выше. При вызове *fplot* имя функции, график которой требуется построить, заключается в апострофы, пределы построения указываются в вектор-строке из двух элементов.

12.1.2.1. Файл-функции с несколькими входными аргументами

Все входные аргументы размещаются в списке через запятую.

Пример

Написать файл-функцию, вычисляющую длину радиуса-вектора точки трехмерного пространства.

```
function r = radius3(x, y, z)
r = sqrt(x.^2 + y.^2 + z.^2);
```

Для вычисления длины радиуса-вектора теперь можно использовать функцию *radius3*, например:

```
>> R = radius3(1, 1, 1)
R =
    1.732
```

12.1.2.2. Файл-функции с несколькими выходными аргументами

Файл-функции с несколькими выходными аргументами удобны при вычислении функций, возвращающих несколько значений (в математике они называются *вектор-функциями*). Выходные аргументы добавляются через запятую в список выходных аргументов, а сам список заключается в квадратные скобки.

Пример

Функция, переводящая время, заданное в секундах, в часы, минуты и секунды.

```
function [hour, minute, second] = hms(sec)
hour = floor(sec/3600);
minute = floor((sec-hour*3600)/60);
second = sec-hour*3600-minute*60;
```

При вызове файл-функций с несколькими выходными аргументами результат следует записывать в вектор соответствующей длины:

```
>> [H, M, S] = hms(10000)
```

```
H =
    2
M =
   46
S =
   40
```

12.2. Основные операторы программирования в MATLAB

Рассмотрим основные операторы, задающие последовательности выполнения команд MATLAB. Операторы можно использовать как в файл-процедурах, так и в функциях, что позволяет создавать программы со сложной разветвленной структурой.

12.2.1. Оператор присваивания

Имя _ *переменной* = *Выражение*;

Типы переменных заранее не объявляются. Они определяются выражением, значение которого присваивается переменной.

Имя переменной может содержать сколько угодно символов, но идентифицируется только 31 начальный символ. Имя любой переменной должно быть уникальным. Имя должно начинаться с буквы, может содержать буквы, цифры.

12.2.2. Условный оператор

Полная форма условного оператора:

```
if <Логическое выражение>  
    <операторы1>  
else  
    <операторы2>  
end
```

Сокращенная форма условного оператора:

```
if <Логическое выражение>  
    <операторы>  
end
```

Логическое выражение может содержать операции отношения и логические операции (табл. 9).

Допустима еще одна конструкция условного оператора:

```
if <Логическое выражение1>  
    <операторы1>  
elseif <Логическое выражение2>  
    <операторы2>
```

```

elseif <Логическое выражение3>
    <операторы3>
    ...
else
    <операторы>
end

```

Таблица 9

Операции отношения и логические операции

| Операции отношения | |
|---------------------|-----------------------------------|
| < | меньше |
| > | больше |
| <= | меньше или равно |
| >= | больше или равно |
| = = | равно |
| ~ = | не равно |
| Логические операции | |
| & | логическая операция “И” (“AND”) |
| | логическая операция “ИЛИ” (“OR”) |
| ~ | логическая операция “НЕТ” (“NOT”) |

Пример

Пусть требуется вычислить выражение $\sqrt{x^2 - 1}$. Предположим, что вычисления выполняются в области действительных чисел и требуется вывести предупреждение о том, что результат является комплексным числом.

Команда **warning** служит для вывода предупреждения в командное окно.

Создаем М-файл:

```

function f = Rfun(x)
% вычисляет sqrt(x^2-1)
% выводит предупреждение, если результат комплексный
% использование y = Rfun(x)
% проверка аргумента
if abs(x)<1
    warning('Результат комплексный')
end
% вычисление функции
f = sqrt(x^2-1);

```

Теперь вызов *Rfun* от аргумента, меньшего единицы, приведет к выводу в командное окно предупреждения:

```
>> y = Rfun(0.2)
```

Результат комплексный

```
y =  
0 + 0.97979589711327i
```

Файл-функция *Rfun* только предупреждает о том, что ее значение комплексное, а все вычисления с ней продолжаются. Если же комплексный результат означает ошибку вычислений, то следует прекратить выполнение функции, используя команду *error* вместо *warning*.

12.2.3. Оператор варианта

```
switch <выражение, скаляр или строка символов>  
  case <значение1>  
    <операторы1>  
  case <значение2>  
    <операторы2>  
  ...  
  otherwise  
    <операторы>  
end
```

12.2.4. Операторы циклов

12.2.4.1. Цикл с предусловием

```
while <Логическое выражение>  
  <операторы>  
end
```

Пример

Вычислить все значения функции $y=\sin x$ для значений аргумента от 0,2 до 4 с шагом 0,2.

```

>> x=0.2;
>> while x<2
y=sin(x);
disp([x,y])
x=x+0.2;
end

```

В результате получим следующие значения:

```

0.2000  0.1987
0.4000  0.3894
0.6000  0.5646
0.8000  0.7174
1.0000  0.8415
1.2000  0.9320
1.4000  0.9854
1.6000  0.9996
1.8000  0.9738
2.0000  0.9093

```

12.2.4.2. Цикл с параметром

```

for<имя>=<Начальное_значение>:<Шаг>:<Конечное_значение>
    операторы
end

```

Пример

Написать программу формирования массива, состоящего из элементов 2^i , $i=0,5,7$.

```

>> k=1;
>> for i=[0 5 7]
x(k)=2^i;
k=k+1;
end
>> x

```

В результате получим

```

x =
    1    32   128

```

12.3. Численное решение задачи Коши для системы обыкновенных дифференциальных уравнений

В системе MATLAB предусмотрены специальные средства решения задачи Коши для систем обыкновенных дифференциальных уравнений, заданных в форме $\frac{dx}{dt} = F(t, x)$, где F – вектор правых частей системы, x – вектор неизвестных функций.

В простейшем варианте для решения задачи Коши для систем обыкновенных дифференциальных уравнений достаточно воспользоваться командой

$[T, X] = \text{solver}('F', [DT], X0, \dots)$,

где DT – диапазон интегрирования, $X0$ – вектор начальных значений, F – имя функции вычисления правых частей системы (и соответственно имя m-файла), solver – имя используемой функции (решателя). Например:

- *ode45* – метод Рунге-Кутты 4 и 5-го порядков,
- *ode23* – тот же метод 2 и 3-го порядков,
- *ode113* – метод Адамса для нежестких систем,
- *ode23s*, *ode15s* – метод Адамса для жестких систем.

Версии решателя различаются используемыми методами (по умолчанию относительная погрешность 10^{-3} и абсолютная 10^{-6}) и соответственно временем и успешностью решения. Под жесткостью здесь понимается повышенное требование к точности – использование минимального шага во всей области интегрирования. При отсутствии информации о жесткости рекомендуется попытаться получить решение посредством *ode45* и затем *ode15s*.

Если диапазон DT задан начальным и конечным значениями $[t_0, t_n]$, то количество элементов в массиве T (и в массиве решений X) определяется необходимым для обеспечения точности шагом; при задании DT в виде $[t_0, t_1, t_2, \dots, t_n]$ или $[t_0:\Delta t:t_n]$ – указанными значениями.

Пример

Решить систему линейных дифференциальных уравнений с заданными начальными условиями:

$$\begin{cases} \frac{dx_1}{dt} = -3x_1 + 10, & x_1(0) = 0; \\ \frac{dx_2}{dt} = x_1 - 2x_2, & x_2(0) = 0; \\ \frac{dx_3}{dt} = 4x_2 - x_3, & x_3(0) = 0. \end{cases}$$

```
% создаем M-функцию с именем sdu.m  
function f=sdu(t,x); % t, x – входные переменные для m-функции  
f=[-3*x(1)+10;x(1)-2*x(2);4*x(2)-x(3)];
```

```
% создаем M-сценарий  
% сценарий решения примера  
T=[0,12];  
x0=[0;0;0]; % x0 – вектор начальных условий  
[t,x]=ode45('sdu',T,x0);  
plot(t,x),grid,title('dif32'),legend('X1','X2','X3')
```

САМОСТОЯТЕЛЬНАЯ РАБОТА № 10

1. Вычислить значение функции для $a=2,5$:

$$T = \begin{cases} x^3 \cdot \sqrt{x-a} & \text{при } x > a; \\ x \cdot \sin ax & \text{при } x = a; \\ e^{-ax} \cdot \cos ax & \text{при } x < a. \end{cases}$$

2. Написать программу для вычисления вектора Y , состоящего из положительных элементов матрицы $A(3,2)$.

3. Найти все значения функции $y = 2x^3 - 4$ для $x \in [-3;3]$ с шагом $h=0,3$.

4. Вычислить значение $\sum_{k=1}^{\infty} \frac{1}{(k+1)(k+2)}$ с точностью $\varepsilon=0,001$.

САМОСТОЯТЕЛЬНАЯ РАБОТА № 11

Решить задачу Коши для следующих систем дифференциальных уравнений при заданных начальных условиях с использованием М-сценария для методов *ode23* и *ode45*:

$$1) \begin{cases} \frac{dx_1}{dt} = 2x_1 - 1; \\ \frac{dx_2}{dt} = 3x_1 + 2x_2 + 1 \end{cases}$$

при $x_1(1) = 0, x_2(1) = 1$.

$$2) \begin{cases} \frac{dx_1}{dt} = x_1 + x_2; \\ \frac{dx_2}{dt} = 2x_1 + 2 \end{cases}$$

при $x_1(1) = 0, x_2(1) = 1$.

$$3) \begin{cases} \frac{dx_1}{dt} = -x_1 + 2; \\ \frac{dx_2}{dt} = 2x_1^2 - 0,5x_2 \end{cases}$$

при $x_1(0) = 10, x_2(0) = 5$.

Библиографический список

1. Лазарева, Т. И. Решение задач в системе Matlab: практическое пособие / Т. И. Лазарева, И. В. Мартынова, И. К. Ракова; Балт. гос. техн. ун-т. – СПб., 2013.
2. Краков, М.С. Компьютерные технологии решения инженерных задач в MATLAB [Электронный ресурс] : учебно-методическое пособие / М.С. Краков, С.Г. Погирницкая. – Электрон. дан. – Минск: БНТУ, 2012. – Режим доступа: <http://rep.bntu.by/handle/data/3866>, свободный. – Загл. с экрана (дата обращения к ресурсу: 15.01.18).
3. Ощепков, А. Ю. Системы автоматического управления: теория, применение, моделирование в MATLAB : учебное пособие / А. Ю. Ощепков и др. – 2-е изд., испр. и доп. – СПб. [и др.]. – 2013.
4. Дьяконов, В. П. MATLAB. Полный самоучитель / В.П. Дьяконов. – М. : ДМК-Пресс. – 2017.