Министерство образования Республики Беларусь
БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Кафедра ЮНЕСКО «Энергосбережение и возобновляемые источники энергии»

КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ РЕШЕНИЯ ИНЖЕНЕРНЫХ ЗАДАЧ В MATLAB

Учебно-методическое пособие

для студентов специальностей
1-43 01 06 «Энергоэффективные технологии и энергетический менеджмент»
1-36 20 01 «Низкотемпературная техника»

Электронный учебный материал

Минск 2012

УДК 519.876.5 (076.5)
ББК 22.1 я 7
К 79

Авторы:
М.С. Краков, С.Г. Погирницкая

Рецензент:
И.В. Никифоров, доцент кафедры вычислительной математики БГУ, кандидат физико-математических наук, профессор, доцент

В учебно-методическом пособии изложены основы применения системы инженерных и научных расчетов MATLAB. Пособие содержит постановки инженерных задач, описание технологий расчетов в MATLAB, упражнения и задания к лабораторным работам по дисциплине «Компьютерные технологии решения инженерных задач».

Для студентов технических специальностей, осваивающих систему MATLAB.

Белорусский национальный технический университет
пр-т Независимости, 65, г. Минск, Республика Беларусь
Тел.(017) 292-77-52 факс (017) 292-91-37
Регистрационный № БНТУ/ФТУГ93-95.2012

Содержание

Лабораторная работа № 1. Приемы работы в системе MATLAB ...4
Лабораторная работа № 2. Режим прямых вычислений MATLAB. Определение расхода газа ...13
Лабораторная работа № 3. Векторизация задач. Сравнительное исследование источников освещения ...19
Лабораторная работа № 4. Технологии решения систем линейных уравнений. Определение расходов жидкости в трубопроводе ...27
Лабораторная работа № 5. Технологии построения графиков. Исследование параметров солнечного модуля...32
Лабораторная работа № 6. Трехмерная графика. Распределение температуры на пластине ...42
Лабораторная работа № 7. Программирование в среде MATLAB ...47
Лабораторная работа № 8. Управляющие конструкции языка MATLAB. Определение гидравлического сопротивления ...53
Лабораторная работа № 9. Нахождение минимумов и нулей функций. Оптимальный размер бака...63
Лабораторная работа № 10. Аппроксимация данных. Определение модуля упругости...66
Лабораторная работа № 11. Численное интегрирование. Определение количества теплоты ...73
Лабораторная работа № 12. Решение обыкновенных дифференциальных уравнений. Изменение температуры воды ...76
Лабораторная работа № 13. Решение ОДУ второго порядка. Расчет колебательной системы...80
Лабораторная работа № 14. Решение задач в частных производных. Распределение температуры...83
Лабораторная работа № 15. Решение задач в символьном виде ..89
ПРИЛОЖЕНИЕ 1. Оформление отчета ...93
ПРИЛОЖЕНИЕ 2. Краткая справка-указатель MATLAB...94
ПРИЛОЖЕНИЕ 3. Типичные ошибки ...99
Литература ...103

Лабораторная работа № 1. Приемы работы в системе MATLAB

Цель занятия: ознакомиться с интерфейсом программы MATLAB, овладеть приемами работы в командной строке, рабочей области, научиться пользоваться справочной системой.

Основные положения

MATLAB – это высокопроизводительный пакет программ, предназначенный для выполнения инженерных и научных расчетов. Он объединяет вычисления, визуализацию и программирование в удобной среде, где задачи и решения выражаются в форме, близкой к математической постановке. Система MATLAB ориентирована на работу с массивами данных (матрицами), что повышает эффективность вычислительного процесса и упрощает программирование.

MATLAB – это интерактивная система, которая имеет удобный для пользователя интерфейс, встроенные средства помощи и диагностики ошибок.

Запуск MATLAB

Программа запускается из **Главного меню** Windows или с **Рабочего стола** помощью ярлыка.

Окна системы MATLAB

После запуска открывается окно системы MATLAB (рис.1.1). Общее окно системы (**Desktop**) разделено на несколько областей (окон) и содержит типовые средства Windows: меню и панель инструментов.

Основные операции пользователь выполняет в рабочей области **Command Window** (Командном окне), где расположена командная строка, предназначенная для ввода команд. В этой же области выводятся результаты вычислений и сообщения об ошибках.

В области **Command History (**Окно истории команд) выводится список выполнявшихся ранее команд и время предыдущих загрузок системы.

Кнопка **Start** обеспечивает быстрый доступ к расширениям MATLAB, справочной системе и документации.

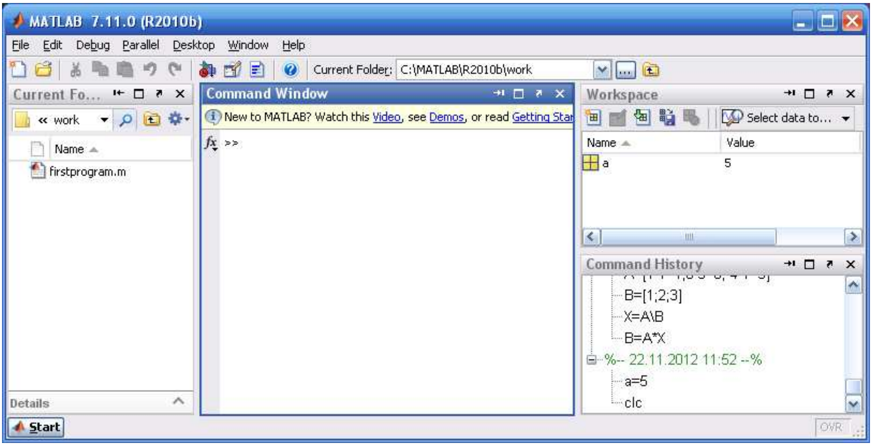


Рис. 1. 1. Окно системы MATLAB

Окно **Current Folder** (Текущая папка) предназначено для просмотра файловой структуры и установки текущей папки, к которой обращается система во время работы.

Окно **Workspace** (Рабочая область) обеспечивает просмотр и внесение изменений в переменные, загруженные в оперативную память системы во время текущей сессии.

Размер и положение окон можно менять, окна можно закрывать. Чтобы вернуться к стандартному расположению окон, достаточно выбрать команду меню **Desktop** > **Desktop Layout** > **Default**.

Работа в командном окне

В командном окне проводятся вычисления в режиме диалога. Пользователь вводит команды или запускает на выполнение файлы с текстами на языке MATLAB. Интерпретатор MATLAB обрабатывает введенные команды и выдает результаты вычислений или сообщения об ошибках.

Командная строка

Командной строкой называется строка, расположенная в командном окне после символа >> и предназначенная для ввода команд. Типичная команда MATLAB выглядит следующим образом: переменной присваивается результат выполнения некоторого выражения. Если имя переменной и знак присваивания опущены, то результат присваивается переменной `ans` (ответ). Обработка команды производится при нажатии клавиши **Enter**.

Пример 1.

```
>> a=sin(2)
a=
   0.9093
```

Пример 2.

```
>> sin(2)
ans=
     0.9093
```

Набранные ранее команды можно вызвать при помощи клавиш ↑ и ↓, а затем отредактировать. Для редактирования командной строки можно использовать следующие клавиши:

← и → – перемещение курсора на символ влево или вправо;
Home – переход на начало строки;
End – переход на конец строки;
Delete – удаление символа за курсором;
Backspace – удаление символа перед курсором;
Esc – очистка строки.

При работе с командным окном можно использовать операции правки из меню **Edit** (а также контекстное меню и кнопки на панели инструментов):

Cut – вырезать,
Copy – копировать,
Paste – вставить,
Undo – отменить ввод,
Redo – повторить ввод (восстановить отмененную операцию).

Длинные выражения удобно разбивать на отдельные строки, используя троеточие и **Enter**.

Пробелы между отдельными операндами, добавленные для лучшего восприятия текста выражения, не влияют на результат.

Для ввода нескольких команд в одной строке следует в конце каждой команды ставить точку с запятой. Чтобы набрать ряд команд, занимающих много строк, и только потом их выполнить, следует каждую строку, кроме последней, оканчивать вводом **Shift+Enter**, а затем нажать **Enter**.

Управление выводом данных на экран

Если набрать в командной строке выражение и нажать клавишу **Enter**, MATLAB выдаст результат на экран. Для отмены вывода данных на экран в конце выражения следует ставить точку с запятой.

Форматом вывода на экран численных значений управляет команда `format`. Синтаксис команды:

```
format формат
```

где `формат` указывает на способ представления чисел:

`short` – число отображается с 4 цифрами после десятичной точки;
`short e` – число в экспоненциальной форме с мантиссой из 5 цифр и показателем из 3 цифр;
`long` – число с 16 десятичными цифрами;
`long e` – число в экспоненциальной форме с мантиссой из 16 цифр и показателем из 3 цифр;
`bank` – число с любым количеством цифр до десятичной точки и двумя цифрами после;
`hex` – число в шестнадцатеричной форме;
`rat` – представление в виде рационального дробного числа.

Пример:

```
>>format long; pi
  ans =
     3.14159265358979
```

Все операции MATLAB проводит с удвоенной точностью и команда `format` не влияет на вычисления. По умолчанию действует формат `short`.

Для очистки командного окна можно использовать команду `clc` в командной строке или **Clear Command Windows** (Очистить командное окно) из меню **Edit**.

Дневник командного окна Command History

Сеанс работы с MATLAB принято именовать *сессией* (session). Просмотреть выполненные ранее команды текущей сессии, а также предшествующих сессий, можно с помощью дневника командного окна **Command History**. Дневник каждой сессии начинается с даты и времени. Используя полосы прокрутки и клавиши перемещения можно просмотреть весь дневник. По мере необходимости записи из дневника можно удалять.

Сделав двойной щелчок на записи дневника, можно выполнить соответствующую команду. Строчки из дневника можно копировать и переносить в командное окно.

Операции с рабочей областью

Переменные в системе MATLAB хранятся в особой области памяти – рабочей области **Workspace**.

Окно **Workspace** (рис. 1.2) позволяет просматривать существующие в памяти объекты, редактировать их содержимое и удалять объекты из памяти.

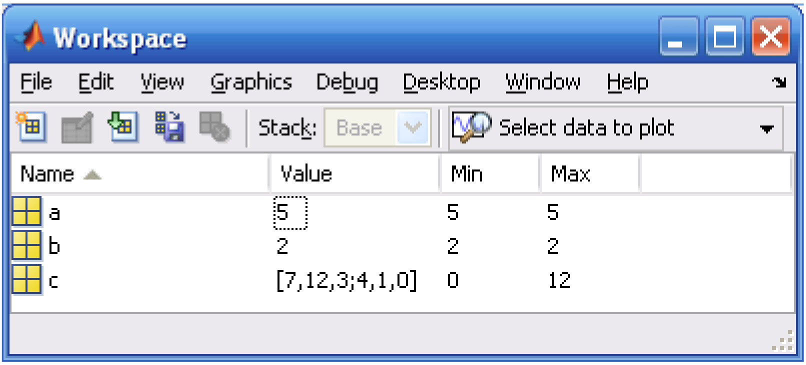


Рис. 1.2. Окно рабочей области Workspace

Для вывода содержимого объекта достаточно выполнить двойной щелчок по имени (рис. 1.2). Окно редактирования массива (рис. 1.3) дает удобный доступ для редактирования любого элемента по правилам, принятым при работе с электронными таблицами.

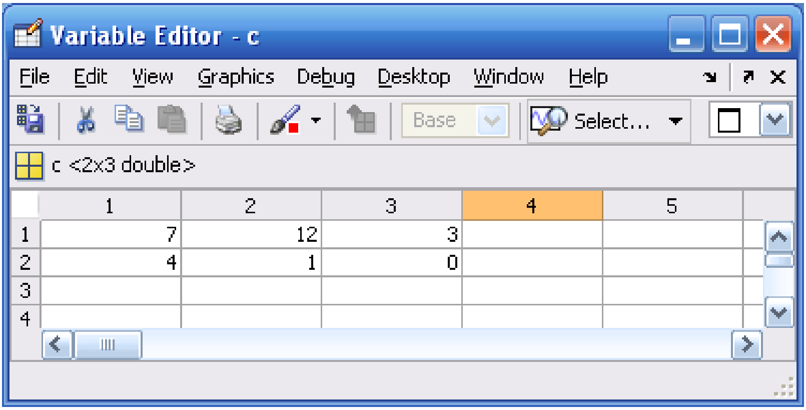


Рис. 1.3. Окно редактирования массива

Просмотр рабочей области возможен и в командном режиме. Команда `who` выводит список переменных, а команда `whos` — список переменных с указанием их размера и объема занимаемой памяти.

MATLAB позволяет сохранять значения переменных в виде двоичных файлов с расширением **.mat** с помощью команды **File** > **Save as Workspace**. Команда **Open** позволяет считать с диска данные рабочей области.

Операции с файлами

Файловые операции используют текущий каталог в качестве отправной точки. Любой файл, который будет использоваться, должен находиться в текущем каталоге или входить в маршрут поиска.

Основным инструментом для работы с файлами является браузер файловой системы **Current Folder**. Браузер позволяет сделать новый каталог текущим, а также добавлять каталоги в перечень путей MATLAB, создавать и открывать файлы.

Быстро сменить текущий каталог позволяет поле со списком **Current Folder**, которое располагается в верхней части окна рабочего стола.

Содержимое рабочей области можно сохранить в двоичном файле формата **.mat**.

Файлы, содержащие тексты программ, написанных на специальном языке программирования MATLAB, имеют расширение **.m**.

Справочная система MATLAB

Набрав в командной строке `help` с параметром, можно получить следующую информацию:

`help`	список разделов справки;
`help раздел`	список команд раздела с указанным названием;
`help имя`	описание команды с указанным именем;
`help ops`	операторы и специальные символы;
`help elfun`	элементарные математические функции;
`help demos`	список примеров;
`type имя_m-файла`	просмотр текста m.файла.

Кнопка панели инструментов **Help** или команда меню **Help** открывают окно с перечнем разделов справочной системы (рис. 1.4).

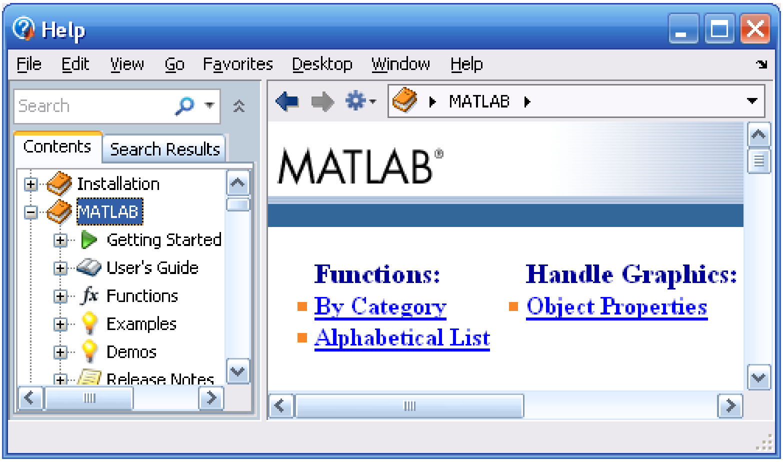


Рис. 1.4 Окно справочной системы

В меню **Help** имеется команда **Demos**, дающая доступ к видео и галерее демонстрационных примеров применения системы MATLAB (рис. 1.5). Это же окно можно вызвать в режиме диалога, набрав в командной строке команду `demo`.

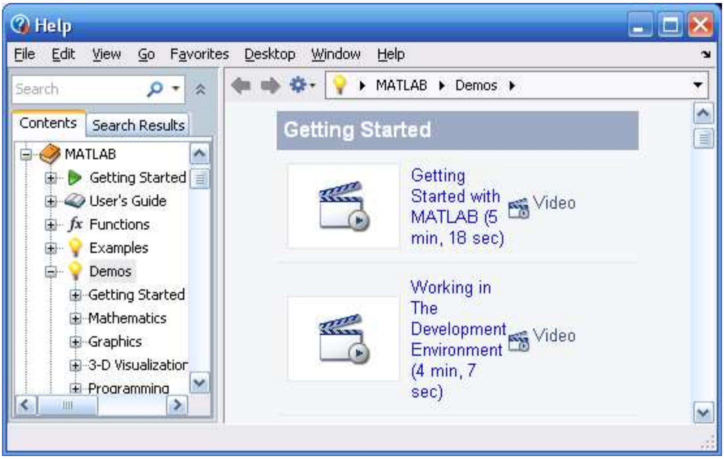


Рис. 1.5. Окно демонстрационных примеров

Контрольные вопросы

1. Назначение системы MATLAB.
2. Перечислите основные элементы интерфейса системы.
3. В каком окне пользователь в диалоговом режиме проводит вычисления?
4. Где выводятся результаты вычислений?
5. Где выводятся сообщения об ошибках?
6. Где можно просмотреть введенные ранее команды?
7. Как узнать имя текущего каталога?
8. Что находится в Рабочей области **Workspace**?
9. Каким образом можно восстановить расположение окон по умолчанию?
10. Как вводятся команды в MATLAB?
11. Что произойдет, если набрать команду и нажать клавишу **Enter**?
12. Что произойдет, если набрать команду и нажать клавишу **Esc**?
13. Как вызвать в командную строку исполненную ранее команду?
14. Каким образом можно отменить ошибочную команду?
15. Как можно очистить Командное окно?
16. Можно ли в одной строке ввести несколько команд?
17. Как расположить длинную команду на нескольких строках?

18.Как отменить вывод результата на экран?

19.Какая команда управляет форматом вывода на экран численных значений?

20.Обязательно ли указывать переменную для значения результата вычислений?

21.Где можно просмотреть и внести изменения в переменные?

22.Как получить перечень элементарных математических функций в системе MATLAB?

23.Как вызвать окно справочной системы MATLAB?

24.Как получить доступ к галерее демонстрационных примеров и видео MATLAB?

Порядок выполнения работы

1. Запустить программу MATLAB.
1. Ознакомиться с окнами системы MATLAB.
2. Вычислить в командной строке:

```
48+13+5
2.5+3*2
```

3. Вывести значение числа π (ввести в командной строке `pi`).
4. Вывести число π в формате:
   – с шестнадцатью цифрами;
   – в экспоненциальной форме.
5. В одной строке присвоить переменной `a` значение `5`, переменной `b` – значение `π/2`.
6. Вычислить `c=a+sin(b)`.
7. Присвоить новые значения переменным `a` и `b` и снова вычислить `c`.
8. Получить справку об элементарных математических функциях MATLAB.
9. Получить справочную информацию о функции `magic`.

10.Ввести команду `M=magic(4)`.

11.Просмотреть рабочую область.

12.Изменить значения элементов матрицы `M` в окне редактора массивов и вывести матрицу в командном окне.

13.Сохранить рабочую область в своей папке.

14.Просмотреть окно истории команд.

Лабораторная работа № 2. Режим прямых вычислений MATLAB. Определение расхода газа

Цель работы: научиться записывать формулы и выполнять вычисления в системе MATLAB в режиме диалога.

Технологии выполнения расчетов

Расчетные формулы записываются в командной строке в виде выражений. *Выражение* – это формульная запись, задающая то, что необходимо вычислить в числовом или символьном виде. Выражение формируется из операторов, функций, имен переменных и констант.

Константы и переменные

Константы (постоянные величины) могут быть *числовыми* и *символьными*.

Для записи чисел MATLAB использует десятичную систему счисления с необязательной десятичной точкой и знаками плюс-минус. При записи числа в экспоненциальной форме используют букву `e` для определения множителя степени десяти. Мнимые числа используют символы `i` или `j`.

Примеры:

```
3            -99            0.0001
9.6397238    1.60210e-20    6.02252e23
2+3i          -3.14159j     3e5i
```

Символьная константа – это цепочка символов, заключенных в апострофы, например: '`текст`'.

Переменные — это имеющие имена объекты, способные хранить данные, обычно разные по значению. MATLAB не требует описания типа переменных или размерности.

Имена переменных состоят из букв, цифр и символов подчеркивания. Первым символом в имени должна быть буква. Число символов в имени не более 31. MATLAB различает заглавные и строчные буквы, поэтому `Abc` и `abc` – это разные переменные.

Чтобы увидеть значение переменной, надо ввести ее имя и нажать **Enter**.

В памяти компьютера переменные занимают определенное место, называемое рабочим пространством (workspace). Все данные хранятся в виде массивов. Числовые массивы состоят из комплексных чисел с двойной точностью. Скаляр – это матрица размера 1×1.

Ряд переменных генерируется системой при ее загрузке (системные переменные):

`pi` – 3.14159265...;

`i, j` – мнимая единица (квадратный корень из –1);

`eps` – относительная точность числа с плавающей точкой, 2^{-52};

`realmin` – наименьшее число с плавающей точкой (2^{-1022});

`realmax` – наибольшее число с плавающей точкой (2^{1023});

`Inf` – бесконечность (появляется при делении на нуль или при выполнении выражения, приводящего к переполнению, т.е. к превышению `realmax`);

`NaN` – не число (генерируется при вычислении выражений типа `0/0`, или `Inf-Inf`, которые не имеют определенного математического смысла).

Информацию об имеющихся константах можно получить, вызвав справку командой `help elmat`.

Значения системных переменных можно изменять. Начальное значение системной переменной восстанавливается командой

```
clear имя_переменной
```

Просматривать, изменять и удалять переменные можно в окне рабочей области **Workspace**. В командной строке для уничтожения всех переменных в рабочем пространстве используется команда `clear` без аргументов. В качестве аргументов можно указать имена переменных, которые надо удалить. Например, команда

```
clear abc,a
```

уничтожает переменные `abc` и `a`.

Сохранять значения переменных в виде двоичных файлов можно с помощью команды **File / Save as Workspace**.

Операторы

Выражения используют обычные арифметические операции и правила старшинства:

1) `( )` определение порядка вычисления;
2) `^` степень, ‘ комплексно-сопряженное транспонирование;
3) `*` умножение, / деление;
4) `+` сложение, – вычитание.

Полный список операторов можно получить по команде `help ops`.

Функции

MATLAB предоставляет большое количество встроенных математических функций. Основные математические функции:

`abs(x)`	абсолютная величина x ($\lvert x \rvert$);
`sign(x)`	знак числа x;
`sqrt(x)`	корень квадратный $\sqrt{x}$;
`exp(x)`	экспонента e^x;
`log(x)`	натуральный логарифм $\ln x$;
`log10(x)`	десятичный логарифм $\lg x$;
`sin(x), cos(x)`	синус ($\sin x$), косинус ($\cos x$);
`asin(x), acos(x)`	арксинус ($\arcsin x$), арккосинус;
`tan(x), cot(x)`	тангенс ($\mathrm{tg}\, x$), котангенс ($\mathrm{ctg}\, x$);
`atan(x), acot(x)`	арктангенс ($\mathrm{arctg}\, x$),арккотангенс($\mathrm{arcctg} x$).

В тригонометрических функциях используется угол в радианах.

Для вывода списка всех элементарных математических функций следует выполнить команду `help elfun`. Перечень более сложных математических и матричных функций можно получить по командам `help specfun` и `help elmat` соответственно.

Имя функции записывается строчными буквами, аргументы указываются в круглых скобках через запятую. В качестве аргументов можно использовать выражения и другие функции.

Примеры.

```
>>rho=(1+sqrt(5))/2
rho=
   1.6180
```

```
>>a=abs(3+4i)
a=
   5
```

Текстовые комментарии

Текстовые комментарии обычно используются для пояснений. Комментарии начинаются с символа **%**, например:

```
% Создание магического квадрата
magic(4)    % создает магический квадрат 4x4
```

Контрольные вопросы

1. Что входит в понятие «выражение» в MATLAB?
2. Перечислите составляющие компоненты выражений в MATLAB.
3. Необходимо ли описывать тип переменной в системе MATLAB?
4. Каким требованиям должно удовлетворять имя переменной в MATLAB?
5. Какой символ является десятичным разделителем в системе MATLAB?
6. Как записываются числа в экспоненциальной форме?
7. Как записывается мнимое число?
8. Как записываются символьные константы?
9. Где можно просмотреть информацию о переменных?
10. Какая команда управляет форматом вывода численных значений на экран?
11. Как сохранить значения переменных в файле?
12. Что такое системные переменные?
13. Как удалить переменную?
14. Перечислите основные операторы MATLAB и их обозначения.
15. Укажите порядок выполнения операций.
16. Как вывести на экран справку об элементарных функциях MATLAB?
17. Как записывается комментарий?

Порядок выполнения работы

1. Вывести на экран справку об операторах и элементарных функциях MATLAB.

2. Выполнить следующие упражнения: записать выражения и вычислить их при заданных значениях параметров:

1) $z = \ln\left(y^{-\sqrt{|x|}}\right)\cdot\left(\sin x + e^{(x+y)}\right)$ при $x = 4$, $y = 0{,}5$ и $x = -1$; $y = 2$.

Ответ: z= 123.7411; . -1.3009.

2) $y = \sqrt{1 + \dfrac{\operatorname{arctg} x}{\ln(2\cdot e^{\sin x})}}$ при $x = \pi$ и $x = 0{,}5$.

Ответ: y=1.6798; 1.1813.

3) $y = \ln\left|\sin^2 x - \sqrt{\cos x}\right|$ при $x = \pi/4$ и $x = 4$;

Ответ: y= -1.0762; -0.0092.

4) $y = \sqrt{\left|\dfrac{e^{2x} + \sqrt{x}}{\sin x - \arcsin x}\right|}$ при $x = 0{,}4$ и $x = -2$;

Ответ: y=5.1375; 0.9712.

5) $f = \sqrt[3]{m\cdot \operatorname{tg} t + |c\cdot \sin t|}$ при $m = 2$, $c = 4$ и $t = 0{,}5$; 1,5.

Ответ: f =1.4439; 3.1812.

6) $y = e^{-bt}\sin(at + b) - \sqrt{|bt + a|}$ при $a = 2$, $b = 4$ и $t = 0{,}7$; 1,8.

Ответ: y= -2.2379; -3.0324.

7) $z = \operatorname{arctg}\sqrt{\left|\dfrac{x - \sin y}{x + \sin y} + \dfrac{x + \sin y}{x - \sin y}\right|}$ при $x = 4$, $y = 0$ и $x = 5{,}5$; $y = \pi/4$.

Ответ: z = 0.9553; 0.9631

8) $f = \sqrt{c\left(\sqrt{y} + x^2\right)}\cdot\left(\cos x - |c - y|\right)$ при $c = 3$, $y = 4$ и $x = 1{,}5$; 2.

Ответ: f = -3.3181; -6.0082

3. Записать символьные константы.

1) abc 2) $1 + x$ 3) $\sin(x)$ 4) $text$

4. Записать как комментарий свою фамилию и название работы

Постановка задачи

Рассматривается обратимое адиабатное истечение газа через сопло (рис 2.1). Газ поступает в канал 1. Скорость газа на входе в сопло w_1. Скорость газа на выходе из сопла – w_2. Давление газа на входе в сопло – P_1. Давление газа на срезе сопла – P_2.

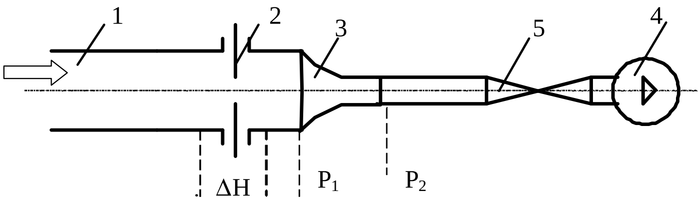


Рис. 2.1. Схема установки.
1- канал,2 - расходомерная диафрагма, 3 -суживающееся сопло,
4 - вакуумный насос, 5 - регулирующий вентиль.

Определить скорость и расход на выходе из сопла для различных газов. Расчетные формулы:

$$w_2 = \sqrt{2\frac{k}{k-1}P_1 v_1\left[1-\left(\frac{P_2}{P_1}\right)^{\frac{k-1}{k}}\right]+w_1^2}\text{, м/с}$$

Здесь v_1 - удельный объем газа, м^3/кг.

Расход газа через сопло

$$G = F\sqrt{2\frac{k}{k-1}\frac{P_1}{v_1}\left[\left(\frac{P_2}{P_1}\right)^{\frac{2}{k}}-\left(\frac{P_2}{P_1}\right)^{\frac{k+1}{k}}\right]}\text{, кг/с.}$$

Здесь $F=\dfrac{\pi d^2}{4}$ – площадь выходного сечения сопла, м^2.

Исходные данные: d=0,01 м; P_1=10^{5}Па; P_2=0,8·10^{5}Па, w_1=100 м/с. Для воздуха: k=1,4; v_1=0,83 м^3/кг; для углекислого газа k=1,29; v_1=0,55 м^3/кг; для гелия k=1,67; v_1=6,05 м^3/кг.

Сохранить рабочую область.

Лабораторная работа № 3. Векторизация задач. Сравнительное исследование источников освещения

Цель занятия: получить навыки ввода и формирования векторов и матриц в MATLAB, научиться записывать и решать задачи в матричном виде.

Технологии расчетов

Большинство функций MATLAB может работать с аргументами в виде векторов и матриц, вычисляя значения для каждого их элемента. Данная операция называется *векторизацией* и обеспечивает упрощение записи операций, производимых одновременно над всеми элементами векторов и матриц, и существенное повышение скорости их выполнения. По умолчанию все числовые переменные в MATLAB считаются матрицами. Скалярная величина есть матрица первого порядка `1x1`, а векторы являются матрицами, состоящими из одного столбца или одной строки.

Ввод векторов и матриц

Непосредственный ввод. В командной строке элементы матрицы записываются в квадратных скобках. Элементы строки отделяются друг от друга пробелами или запятыми, строки разделяются точкой с запятой. В качестве элементов матриц можно использовать арифметические выражения.

Примеры:

```
>>u = [3;1;4] % ввод вектора-столбца
u =
     3
     1
     4
>>v = [2 0 -1] % ввод вектора-строки
v =
     2 0 -1

>>M = [1 2 3; 4 5 6; 7 8 9] % ввод матрицы
M
     1 2 3
     4 5 6
     7 8 9
```

Импорт данных. В контекстном меню окна **Workspace** или в меню **File** выбрать пункт **Import Data** и в появившимся диалоговом окне открыть нужный файл. Программа распознает данные различных типов, в том числе текстовые файлы **.txt** и электронные таблицы **.xls**. Текстовый файл должен представлять собой прямоугольную таблицу чисел, отделенных пробелами, с равным количеством элементов в каждой строке.

Данные можно передать через буфер обмена по команде **Edit** > **Paste Special**.

Генерирование векторов и матриц

Диапазон чисел может быть сформирован в виде вектора при помощи двоеточий, разделяющих начальное значение, шаг и предельное значение. Если величина шага отсутствует, то по умолчанию его значение равно единице.

Примеры:

```
>>x = 1:-2:-6          >>t = 1:10
x =                    t =
   1 -1 -3 -5             1 2 3 4 5 6 7 8 9 10
```

Вектор равноотстоящих точек формируется с помощью функции `linspace(a,b,n)`, где `n` – количество точек, равномерно распределенных на отрезке от `a` до `b`. Если параметр `n` не указан, то формируется массив из 100 точек.

```
>>z = linspace(1,2,5)
z =1  1.2500  1.5000  1.7500  2.0000
```

Матрицы с заданными свойствами формируются с помощью команд, аргументами которых являются размерности создаваемых матриц. Если указано одно число, то формируется квадратная матрица.

`eye`	единичная матрица
`ones`	матрица из единиц
`zeros`	нулевая матрица
`rand`	случайная матрица со значениями из интервала [0,1]
`magic`	матрица магического квадрата
`diag`	диагональ матрицы

Примеры:

```
>>A = eye(3)
A =
    1    0    0
    0    1    0
    0    0    1
```

```
>>M = ones(3,4)
M =
    1    1    1    1
    1    1    1    1
    1    1    1    1
```

```
>>d = diag(A)
d =
     1
     1
     1
```

```
>> R = rand(3,1)
R =
  0.8913
  0.7621
  0.4565
```

Размерность матриц

`>>size(M)` выдает размер матрицы в виде вектора-строки

`ans=3   4`

`>>Q = zeros(size(M))` – возвращает нулевую матрицу такой же размерности, что и `M` (3x4).

```
Q =
    0    0    0    0
    0    0    0    0
    0    0    0    0
```

`>>length(d)` выдает длину вектора

`ans =5`

Транспонирование матриц

Матрицы можно транспонировать с помощью оператора «′» (апостроф), например `A′`. Строки транспонированной матрицы соответствуют столбцам исходной матрицы.

Обращение к элементам матрицы

Индексы. Для указания отдельного элемента матрицы используются индексы. Элемент в строке `I` и столбце `J` матрицы `A` обозначается `A(I,J)`.

`>>p=M(1,2)` переменной присвоить значение элемента матрицы

`>>A(2,3)=10` элементу матрицы присвоить значение.

Выражение `A(I)` с одним индексом дает доступ к элементам матрицы, развернутым в один столбец. В этом случае массив рассматривается как длинный вектор, сформированный из *столбцов* исходной матрицы.

Если попытаться использовать значение элемента вне матрицы, MATLAB выдаст ошибку. Однако если элементу матрицы с индексами, превышающими ее размерность, присвоить некоторое значение, то размер матрицы увеличится.

Выделение частей матрицы. Оператор «двоеточие» используется для формирования подвекторов и подматриц из векторов, матриц и многомерных массивов. Двоеточие само по себе означает строку или столбец целиком:

`M(:,J)` – это `J`-й столбец из `M`;
`M(I,:)` – это `I`-я строка из `M`;
`M(:,:)` – это матрица `M`;
`M(J:K)` – это `M(J), M(J+1)`,…, `M(K)`;
`M(:,:,K)` – это `K`-я страница трехмерного массива `M`;
`M(:)` записывает все элементы массива `M` в виде столбца.

Удаление строк и столбцов. Элементы, строки и столбцы матрицы можно удалить, используя пустые квадратные скобки [].

Пример.

```
>>M=magic(4)
```

Удалить 2-й столбец	четвертую строку
`>>M(:,2)=[]`	`>>M(4,:)=[]`

Объединение матриц – это процесс соединения нескольких матриц в одну большую. Оператор объединения – это пара квадратных скобок.

Пример:

```
>>M=[1 2; 3 4];
>>R =[8; 6];
>>B=[M R]
B =
    1   2   8
    3   4   6
```

Матричные вычисления

При нахождении максимума и минимума, вычислении сумм, произведений, средних значений и др. для матриц по умолчанию действует следующее правило: вычисляется соответствующая операция для элементов столбцов и результат помещается в вектор-строку. Чтобы проделать вычисления построчно, можно выполнить операцию над транспонированной матрицей. Кроме того, для многомерных массивов можно явно указать размерность, по которой будет действовать операция. Команды для работы с массивами:

`sum`	Суммирование элементов массива
`prod`	Произведение элементов массива
`max`	Определение максимальных элементов массива
`min`	Определение минимальных элементов массива
`mean`	Определение средних элементов массива
`median`	Определение медианы (срединных значений)
`sort`	Сортировка элементов массива по возрастанию

Примеры:

```
>>M=[1 2; 3 4]
M =
     1   2
     3   4
```

```
>>sum(M)          >> sum(M,1)         >> sum(M, 2)
ans =             ans =               ans =
   4  6                4  6                 3
                                            7
```

```
>> sum(sum(M))
ans = 10
```

```
>>max(M)                >>[C,I]=max(M)
ans =                   C= 3    4 (максимальные элементы)
    3   4               I= 2    2 (индексы строк)
```

Контрольные вопросы

1. Как ввести данные в виде вектора-строки, вектора-столбца, матрицы?
2. Как задать диапазон значений?

3. Как создать вектор равноотстоящих точек?
4. Как создать единичную матрицу?
5. По какой команде выводится размерность матрицы?
6. Как построить нулевую матрицу такой же размерности, как имеющаяся матрица?
7. Как обратиться к элементу матрицы?
8. В каком порядке располагаются элементы матрицы?
9. Как выделить столбец матрицы?
10. Как удалить строку или столбец матрицы?
11. Каким образом можно объединить матрицы?
12. Как транспонировать матрицу?
13. Как вычислить сумму элементов матрицы по столбцам?
14. Как вычислить сумму элементов матрицы по строкам?
15. Как вычислить сумму всех элементов матрицы?
16. Как вычислить произведение элементов матрицы?

Порядок выполнения работы

1. Ввести вектор-строку `u`, вектор-столбец `v`, матрицу `A`.

$$u = [5 \quad 8 \quad 10], \quad v = \begin{pmatrix} 1{,}5 \\ 4 \\ 2 \end{pmatrix}, \quad A = \begin{pmatrix} -1 & 2 & 4 \\ 3 & 8 & -2 \\ 7 & 3 & 1 \end{pmatrix}.$$

2. Присвоить элементу матрицы `A` из 3–й строки, 2-го столбца. значение, равное 4.
3. Используя оператор «двоеточие», создать:

– вектор-строку `t1` с целочисленными элементами от 1 до 10;
– вектор-строку `t2` с компонентами от 0 до 10 с шагом 2;
– вектор-строку `t3` с компонентами от 10 до 0.

4. Сформировать вектор `t4` из 11 равноотстоящих точек на отрезке [0; 50].

5. Создать матрицы с заданными свойствами:

– единичную матрицу `D` размерности 5;
– матрицу `N` размерности 3x4, все элементы которой 1;
– нулевую матрицу `Z` размерности 2x4;

– вектор `SL` с пятью случайными величинами;

– магический квадрат `MAG` размерности 4;

– вектор `DM`, равный диагонали магического квадрата `MAG`.

6. Проверить свойство магической матрицы `MAG`: суммы элементов по строкам, столбцам и диагоналям совпадают.

7. В текстовом редакторе **Блокнот** создать файл `data.txt` с данными и загрузить их в MATLAB.

```
46 12 34 1.8
3.1 9 15 1.22
```

8. Импортировать исходные данные из электронных таблиц MS Excel. Предварительно в MS Excel создать книгу с одним листом, ввести значения матрицы и сохранить.

Постановка задачи

При исследовании источников света (рис. 3.1) были получены следующие данные: значения освещенности E в пяти точках поверхности включенных светильников и потребляемые мощности N (табл. 3.1).

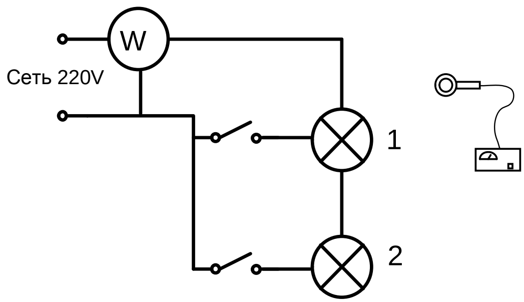


Рис. 3.1. Схема установки.
1- лампа накаливания, 2 – люминесцентная лампа

Таблица 3.1.

№ светильника	E_1, лк	E_2, лк	E_3, лк	E_4, лк	E_5, лк	$E_{ср}$, лк	N, Вт	F, Вт	K
1.	1500	1700	1600	1300	1500		25		
2.	1800	1600	1700	1800	1900		5		

Найти среднее значение освещенности $E_{ср}$ для каждого светильника.

Найти поток излучения для каждого светильника по формуле:

$$F = \left(E_{cp} \cdot S\right)/\,217 \text{ , Вт,}$$

где S площадь поверхности светильников, имеющих форму цилиндра с диаметром d=0,095 м и высотой h=0,145 м.

Найти к.п.д. каждого светильника $K = \dfrac{F}{N} \cdot 100\%$.

Найти максимальное значение к.п.д. K.

Методика расчета

1. Ввести размеры светильников и вычислить площадь поверхности по формуле $S = \dfrac{\pi d^2}{4} + \pi dh$.
2. Ввести таблицу значений освещенности в виде матрицы `E`, строки которой соответствуют данным для каждого из светильников. Ввести значения потребляемых мощностей в виде вектора-столбца `N`.
3. Найти средние значения освещенности для каждого светильника по команде `ES=mean(E,2)`.
4. Вычислить поток излучения `F=ES*S/217`.
5. Вычислить к.п.д. каждого светильника `K=F./N*100`. Пояснения об операции «деление с точкой» даны в работе № 4.
6. Найти максимальное значение к.п.д. K и номер светильника по команде `[Kmax,nomer]=max(K)`.
7. Определить размер матрицы `E`, длину векторов `N`.
8. Из матрицы `E` выделить строку с данными для первого светильника, из вектора `N` – значение его мощности.
9. Составить матрицу `LAMP` из матрицы значений освещенности `E`, вектора средних значений освещенности `ES`, вектора мощностей `N` и вектора `K`. В результирующей матрице строки представляют собой наборы данных для каждого из светильников.
10. Удалить из матрицы `LAMP` экспериментальные данные для первого светильника.

Лабораторная работа № 4. Технологии решения систем линейных уравнений. Определение расходов жидкости в трубопроводе

Цель работы: научиться решать задачи, сводящиеся к системе линейных алгебраических уравнений, выполнять матричные вычисления.

Матричные и поэлементные операции

В MATLAB реализовано два типа арифметических операций: операции над матрицами в соответствии с правилами линейной алгебры и поэлементные операции. Чтобы их различить, при записи поэлементной операции применяется точка.

`+` `–` Сложение и вычитание. Операнды должны быть одинакового размера. Если один из операндов является скаляром, то скаляр применяется к каждому элементу другого операнда.

`*` Матричное умножение. Число столбцов первого сомножителя должно быть равно числу строк второго. На скаляр умножаются все элементы сомножителя.

`.*` Поэлементное умножение двух массивов одинакового размера. На скаляр умножаются все элементы массива.

`^` Степень матрицы. Если показатель степени целое положительное число, то матрица перемножается сама на себя.

`.^` Поэлементное возведение в степень.

`\` Левое деление. Решение систем линейных уравнений AX=B. X=A\B

`.\` Деление элементов второго массива на соответствующие элементы первого. Массивы должны быть одинаковых размеров, за исключением случая, когда один из них скаляр.

`/` Правое деление. Операция равносильна умножению первой матрицы на обратную вторую или B/A = (A'\B')'.

`./` Поэлементное деление. Массивы должны быть одинаковых размеров, за исключением случая, когда один из них скаляр.

`‘` Транспонирование матриц (с комплексным сопряжением).

`.‘` Транспонирование массивов. Строки заменяются столбцами.

Решение систем линейных уравнений

Систему линейных уравнений

$$\begin{aligned}
&a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \\
&a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2 \\
&\ldots \\
&a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n = b_n
\end{aligned}$$

обычно кратко записывают в матричном виде

$$\mathrm{Ax} = \mathrm{b},$$

где `A` – квадратная матрица с размерностью `nxn`, а `B` – вектор-столбец с `n` компонентами.

Решение системы линейных уравнений в MATLAB находят с помощью левого деления

```
x = A\b
```

Определитель и обратная матрица

`det(A)` – возвращает определитель квадратной матрицы `A`.

`inv(A)` – возвращает матрицу, обратную квадратной матрице `A`.

Контрольные вопросы

1. В чем различие арифметических операций над матрицами и поэлементными операциями?
2. Как выполняется умножение матриц?
3. Что означает символ «точка» перед символом оператора?
4. Что такое левое и правое деление?
5. Можно ли использовать операцию сложения матрицы и числа?
6. Какая функция MATLAB позволяет узнать размер матрицы?
7. Как решить систему линейных уравнений в MATLAB?
8. Как найти обратную матрицу?
9. Как вычислить определитель матрицы?

Порядок выполнения работы

1. Даны два вектора *x* и *y*. Выполнить операции. Для вывода результатов на экран использовать формат `format rat`. Объяснить случаи, когда MATLAB выдает сообщение об ошибке.

$$x = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \qquad y = \begin{pmatrix} 4 \\ 5 \\ 6 \end{pmatrix}$$

x'	*y'*	*x+2*	*2–y*	*x*2*	*x.*2*
*x*y*	*x.*y*	*x'*y*	*x'.*y*	*x*y'*	*x.*y'*
x/y	*x./y*	*x\y*	*x.\y*	*2\x*	*2. \x*
x^y	*x.^y*	*x^2*	*x.^2*	*2^x*	*2.^x*

2. Решить системы линейных уравнений (по вариантам). Выполнить проверку.

1) $\begin{cases} x_1 + 3x_2 - 3x_3 = -1 \\ 2x_1 + 4x_2 + 3x_3 = 3 \\ 3x_1 + 2x_2 - 5x_3 = 13 \end{cases}$ $\quad \begin{cases} x_1 + 0.47x_2 - 0.11x_3 + 0.55x_4 = 1.33 \\ 0.42x_1 + x_2 + 0.35x_3 + 0.17x_4 = 1.29 \\ -0.25x_1 + 0.67x_2 + 5x_3 + 0.36x_4 = 2.11 \\ 0.54x_1 - 0{,}32x_2 - 0.74x_3 + x_4 = 0.1 \end{cases}$

2) $\begin{cases} x_1 - x_2 + x_3 = 3 \\ 2x_1 + x_2 + x_3 = 11 \\ x_1 + x_2 - 2x_3 = 8 \end{cases}$ $\quad \begin{cases} 2x_1 - 3x_2 + x_3 - 2 = 0 \\ x_1 + 5x_2 - 4x_3 + 5 = 0 \\ 4x_1 + x_2 - 3x_3 + 4 = 0 \end{cases}$

3) $\begin{cases} 4x_1 = 7x_2 - 3x_3 = 10 \\ 2x_1 + 9x_2 - x_3 = 8 \\ x_1 - 6x_2 + 3x_3 = -3 \end{cases}$ $\quad \begin{cases} x + y - z = 1 \\ 8x + 3y - 6z = 2 \\ 4x + y - 3z = 3 \end{cases}$

4) $\begin{cases} 3x_1 + 2x_2 + x_3 = 5 \\ 2x_1 + 3x_2 + x_3 = 1 \\ 2x_1 + x_2 + 34x_3 = 11 \end{cases}$ $\begin{cases} x_1 + 2x_2 + 3x_3 - 2x_4 = 6 \\ 2x_1 + 4x_2 - 2x_3 - 3x_4 = 18 \\ 3x_1 + 2x_2 - x_3 + 2x_4 = 4 \\ 2x_1 - 3x_2 + 2x_3 + x_4 = -8 \end{cases}$

3. Найти определители и обратные матрицы.

$$C = \begin{pmatrix} 2 & 7 & -8 & 1 \\ 3 & 15 & 10 & 2 \\ 1 & 0 & 0 & 0 \\ -2 & -3 & 5 & 4 \end{pmatrix}; \qquad D = \begin{pmatrix} 0.79 & -1.12 & 0.34 & 0.16 \\ -0.34 & 1.18 & -0.17 & 0.18 \\ -0.16 & -0.34 & 0.85 & 0.31 \\ -0.12 & -0.26 & 0.08 & 0.75 \end{pmatrix}.$$

Постановка задачи

Имеется трубопровод (рис. 4.1), состоящий из двух параллельно соединенных трубопроводов с круглыми сечениями с диаметрами d_1, d_2. Длины ветвей трубопровода l_1, l_2. Известен общий расход жидкости Q. Жидкость однородна, несжимаема, плотность жидкости ρ постоянная. Движение жидкости установившееся. Исследовать, как будут меняться потери давления и распределение расходов в ветвях трубопровода при изменении их параметров.

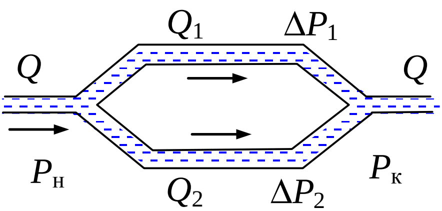


Рис. 4.1. Схема трубопровода

Методика расчета

Как следует из гидравлики, потери давления в каждом ветви трубопровода определяются выражением

$$\Delta P_1 = c_1 Q_1^{\ 2}; \ \ \Delta P_2 = c_2 Q_2^{\ 2},$$

где

$$c_1 = \left(\lambda_1 \frac{l_1}{d_1} + \xi_1 \right) \frac{8\rho}{\pi^2 d_1^{\,4}}; \quad c_2 = \left(\lambda_2 \frac{l_2}{d_2} + \xi_2 \right) \frac{8\rho}{\pi^2 d_2^4};$$

λ_1 , λ_2 – коэффициенты гидравлического трения, зависящие от режима течения жидкости и шероховатости труб;
ξ_1,, ξ_2 – коэффициенты местных сопротивлений (участков изменения конфигурации трубопровода, вентили, задвижки и т.п.).

Поскольку на основании законов сохранения энергии $\Delta P_1 = \Delta P_2$, $Q_1 + Q_2 = Q$, то получается система уравнений:

$$\begin{cases} c_1 Q_1^{\;2} = c_2 Q_2^{\;2} \\ Q_1 + Q_2 = Q \end{cases}$$

После извлечения квадратного корня из первого уравнения, данная система сводится к системе линейных уравнений.

$$\begin{cases} \sqrt{c_1} Q_1 - \sqrt{c_2} Q_2 = 0 \\ Q_1 + Q_2 = Q \end{cases}$$

Исходные данные:

Q = 0,2 *м*^{3}/*с*, ρ = 998 *кг*/*м*^{3},
l_1 = 100 *м*, l_2 = 150 *м*, d_1 = 0,05 *м*, d_2 = 0,15 *м*,
ξ_1 = 20, ξ_2 = 35, $\lambda_1 = \lambda_2$ = 0,02.

Лабораторная работа № 5. Технологии построения графиков. Исследование параметров солнечного модуля

Цель занятия: научиться строить двухмерные графики, управлять выводом графиков.

Технологии построения двумерных графиков

Практически любое решение инженерной задачи сопровождается построением графиков.

Построение графиков

Графики в декартовой системе координат строят с помощью команд `plot` и `fplot`. Результат выводится в специальном окне.

`plot(y)` – создает кусочно–линейный график зависимости компонент вектора `y` от их индексов. Ось абсцисс отображает номер индекса в векторе `y` и начинается с 1.

```
y = [0 1 4 9 16 25]; plot(y)
```

`plot(x,y)` – создает график зависимости `y` от `x`. Обратите внимание, что значения `y` вычисляются поэлементно, поэтому перед знаками операторов умножения, деления, возведения в степень необходимо ставить точку.

```
x=1:0.1:8; y=1./x;
plot(x,y)
```

`plot(x1,y1,x2,y2, …)` – строит несколько кривых на одном графике.

```
x=0:pi/100:8;
y=sin(x); y2=sin(x-0.25);
plot(x,y,x,y2)
```

`plot(x,y,'цвет_стиль_маркер')` – позволяет задавать цвет, стиль линий и маркеры. Аргумент `'цвет_стиль_маркер'` – это строковая константа, составленная из символов цвета, стиля линий и маркеров, записывается слитно, без разделителей (табл.5.1).

Если определить только тип маркеров, но не указать стиль линий, MATLAB выведет на график только маркеры (рис. 5.1).

Таблица 5.1. Параметры линий.

Цвет линий		Стиль линий		Тип маркера	
y	Желтый	-	Сплошная	o	Окружность
m	Фиолетовый	--	Штриховая	x	Крест
c	Голубой	-.	Штрих-пунктир	+	Плюс
r	Красный	:	Двойной пунктир	*	Звездочка
g	Зеленый			s	Квадрат
в	Синий			d	Ромб
w	Белый			V	Треугольник
к	Черный				

```
x1=0:pi/100:2*pi;
x2=0:pi/10:2*pi;
plot(x1,sin(x1),' k:',x2,sin(x2),'ro')
```

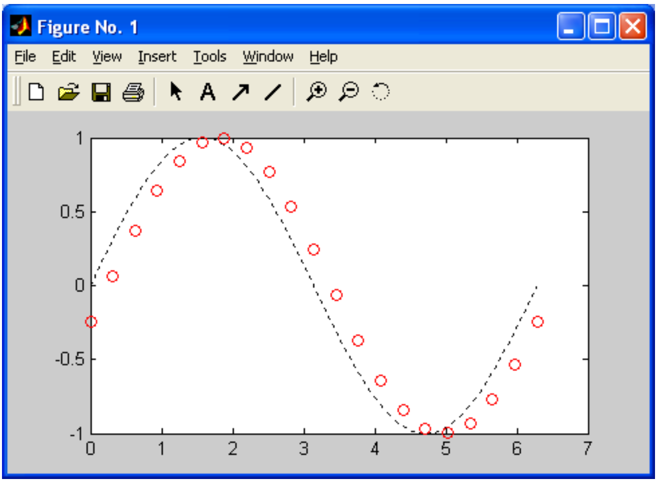


Рис. 5.1. Построение графиков с заданными свойствами линий

`fplot('fun',[x1 x2])` – строит график функции, заданной в символьном виде, в интервале изменения аргумента от `x1` до `x2`. Аргумент `'fun'` – это выражение для вычисления функции.

```
fplot('sin(x)+0.5',[0 2*pi])
```

Графики в полярной системе координат. В полярной системе координат точка представляется как конец радиус–вектора, исходящего из начала системы координат, имеющего угол φ и длину *r*.

`polar(phi,r)` – реализует построение графиков в полярной системе координат, задаваемых углом `phi` и радиусом `r` (рис. 5.2).

```
t = 0:.01:2*pi;
polar(t, sin(2*t).*cos(2*t),'--r')
```

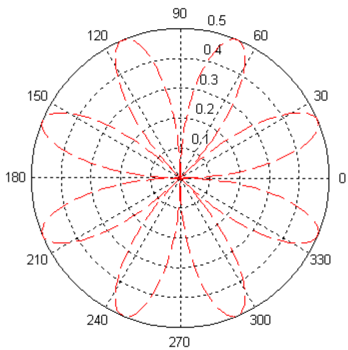


Рис. 5.2. График в полярной системе координат

Управление выводом графика

Нанесение надписей и сетки на графике

`title('TEXT')` – вывод заголовка TEXT.

`xlabel('TEXT')` – маркировка оси x.

`ylabel('TEXT')` – маркировка оси y.

`text(x,y,'TEXT')` – вывод текста в заданное место графика.

`legend('TEXT1','TEXT2',..)` –идентификация кривых.

Греческие буквы и символы записываются в текстовой строке в виде команд типа `\alpha`, `\beta`, `\phi`, `\infty`, `\rightarrow`, `_1` (α, β, ϕ, ∞, →, нижний индекс 1 соответственно).

`grid` – нанесение координатной сетки. По умолчанию на двумерных графиках сетка не наносится. Для задания сетки нужно выполнить команду `grid on`, а убрать по команде `grid off`. Команда `grid` без параметров действует как переключатель.

```
x =0:pi/100:8;y=sin(x);y2=sin(x-0.25);
plot(x,y,x,y2,'m--');
title('sin \alpha')
legend('sin(x)','sin(x–0.25)')
xlabel('ОСЬ АБСЦИСС')
ylabel('ОСЬ ОРДИНАТ')
text(3,0.8,'ГРАФИК')
grid on
```

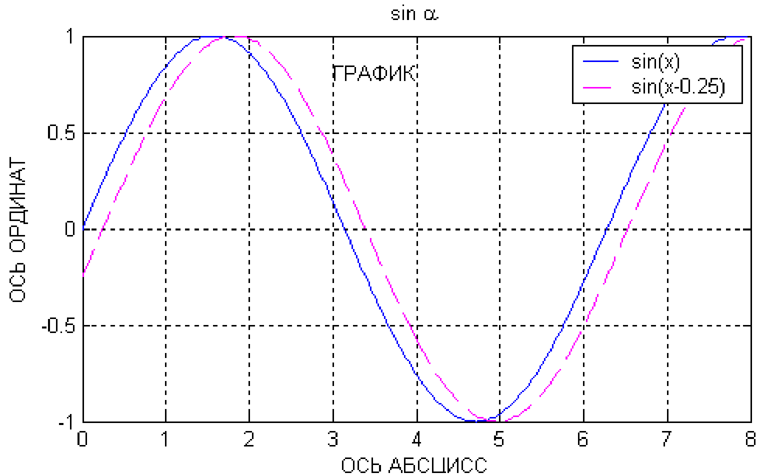


Рис. 5.3. Нанесение надписей и сетки на график

Добавление кривых на существующий график

`hold on` – обеспечивает продолжение вывода графиков в текущее окно;

`hold off`–отменяет режим продолжения;

`hold` – работает как переключатель с одного режима на другой.

Разбиение графического окна

Команда `subplot` используется для разбиения графического окна на подокна:

`subplot(m,n,p)` разбивает графическое окно на `mxn` подоокон, при этом `m` – число подокон по горизонтали, `n` – число подокон по вертикали, `p`- номер подокна, в котором будет строиться текущий график (нумерация идет по строкам) (рис. 5.4).

```
t = 0:pi/50:2*pi;
subplot(2,1,1); plot(t,sin(t))
subplot(2,1,2); plot(t,cos(t))
```

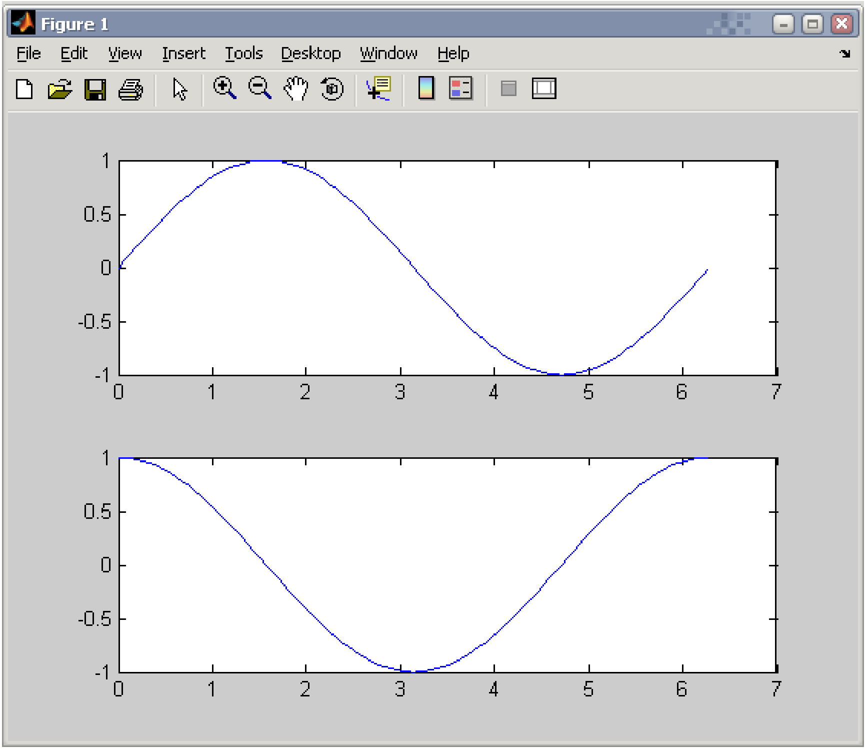


Рис. 5.4. Разбиение графического окна

Окна изображений

Функция `plot` автоматически открывает новое окно изображения, если до этого его не было на экране. Если же окно существует, то `plot` использует его по умолчанию. Новое окно открывается по команде `figure`. Если окон изображения несколько, то для выбора одного из них в качестве текущего окна следует использовать команду `figure(n)`, где `n` – это номер в заголовке окна. Результаты всех команд будут выводиться в текущее окно.

Редактирование изображений в интерактивном режиме

Графики строятся в специальном графическом окне. Это окно имеет свое меню и панели инструментов, при помощи которых легко производить настройку вида выводимого графика (рис. 5.5).

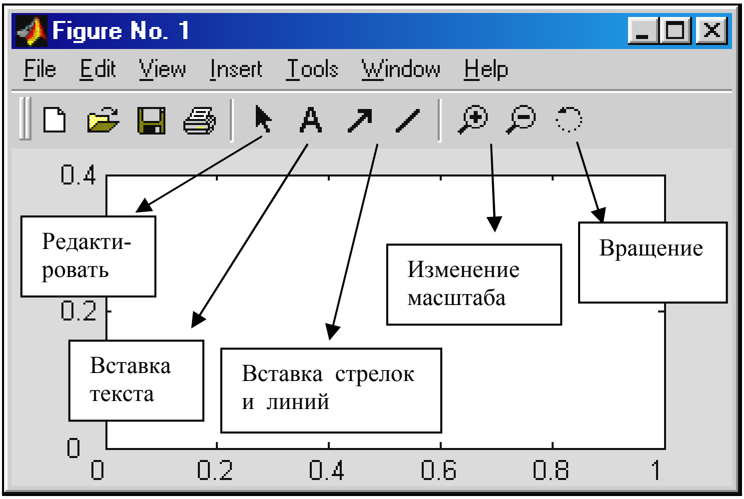


Рис. 5.5. Графическое окно

Для перехода в режим редактирования достаточно щелкнуть на кнопке **Edit Plot** (Редактировать график) на панели инструментов или выбрать пункт **Enable Plot Editing** в меню **Tools**. После этого двойной щелчок мыши по объекту редактирования вызывает окно свойств соответствующего объекта. В этом режиме графиком можно также управлять с помощью контекстного меню.

В интерактивном режиме можно задавать параметры осей, сетки, надписей, линий, угол зрения на рисунок и прочие параметры.

С помощью кнопок панели инструментов можно дополнить рисунок текстом, стрелками и линиями, а также изменять масштаб и вращать изображение.

Команда **Edit > Copy Figure** позволяет через буфер обмена скопировать график в другие приложения.

Контрольные вопросы

1. Какая команда в MATLAB строит графики в декартовой системе координат?
2. Где выводится график?
3. Какие входные параметры имеет функция `plot`?
4. Как построить несколько кривых на одном графике?
5. Как задать цвет и стиль кривых на графике?
6. Как построить график функции, заданной выражением?
7. Какая функция MATLAB реализует построение графиков в полярной системе координат?
8. Какие команды MATLAB позволяют наносить надписи на графике?
9. Как задать режим нанесения координатной сетки?
10. Как добавить кривую на существующий график?
11. Как разбить графическое окно на подокна для вывода нескольких графиков?
12. Как создать новое графическое окно?
13. Как редактировать изображение графика в интерактивном режиме?

Порядок выполнения работы

1. Построить график распределения температур по длине медного стержня по показаниям термопар, расположенным через равные промежутки: 88°C, 65°C, 49°C, 35°C, 21°C.

2. Построить график функции $y = \dfrac{2x}{x^2+1}$ на отрезке [–3; 3]. Использовать функции `plot` и `fplot`.

3. На построенный в п. 2 график добавить кривую $y = \dfrac{x+1}{x^2+1}$.

4. Построить графики следующих функций. Варианты заданий:

1) $y = \sin(x)$ (линия сплошная, зеленая) и $y = \sin^2 x - 2\sin x$ (линия штриховая, красная) на отрезке x [0;2π];

2) $y = 3e^{-x}\sin(x)$ (линия сплошная, красная) и $y = e^{-x}\sin(x + \pi/8)$ (линия штриховая, синяя) на отрезке [0;5π];

3) $y = e^{-x}\sin(x)$ (линия штрих-пунктирная, фиолетовая) и $y = e^{-2x}\sin(x)$ (линия двойной пунктир, зеленая) на отрезке [0;3π];

4) $y = x^4 + 2x^2 + 3$ (линия сплошная, черная) и $y = 2x^2 + 3$ (маркеры звездочки желтого цвета) на отрезке [–2; 2].

5. В одном графическом окне в двух подокнах построить в полярных системах координат графики двух выбранных функций. Нанести названия графиков. Угол φ меняется от 0 до 2 π, если не указано иначе.

1) Окружность с центром в точке (2,0) и радиусом 2: r=4cosφ;
2) Окружность с центром в точке (0,1) и радиусом 1: $r = 2\sin\varphi$;
3) k-лепестковая роза $r = \cos k\varphi \quad k = 5$;
4) $2k$-лепестковая роза $r = |\cos k\varphi|, \quad k = 3$;
5) k-лепестковая роза $r = \sin k\varphi, \quad k = 4$;
6) Трилистник $r = 1 + \cos 3\varphi + \sin^2 3\varphi$;
7) Лемниската Бернулли a=2; $r^2 = 2a^2 \cos 2\varphi$;
8) Кардиоида $r = a(\cos\varphi - 1)$;
9) Улитка Паскаля $r = \cos 2\varphi - 1$;
10) Спираль Архимеда $r = k\varphi, \quad \varphi$ *от* 0 *до* 4π;
11) Логарифмическая спираль $r = k\exp(a\varphi), \quad \varphi$ *от* 0 *до* 4π;
12) Гиперболическая спираль $r = k/\varphi, \quad \varphi$ *от* 0 *до* 6π;

13) Спираль Галилея $r^2 = k^2\varphi^2$, φ *от* 0 *до* 4π;

14) Спираль Ферма $r^2 = k^2\varphi$, φ *от* 0 *до* 4π;

15) Луч $\varphi=\pi/4$; r от 0 до 5;

16) Окружность с центром в полюсе 0 $r=2$, φ от 0 до 2π.

Постановка задачи

При исследовании солнечного модуля (рис.5.6) были получены следующие данные (табл. 5.1): значения освещенности в пяти точках поверхности солнечного модуля и величины ЭДС, вырабатываемых солнечным модулем при четырех положениях источника света (без нагрузки).

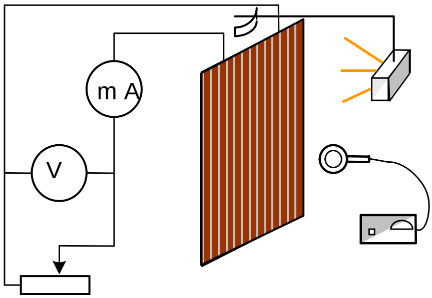


Рис. 5.6. Схема установки с солнечным модулем

Таблица 5.1.

E_1, лк	E_2, лк	E_3, лк	E_4, лк	E_5, лк	$E_{ср}$, лк	W, Вт/м^2	*ЭДС*, В
3000	2300	2400	2000	2000			17
2800	2200	2300	1900	2000			16,5
2700	2100	2000	1700	1800			16
2500	2000	1800	1300	1500			15

Определить среднее значение освещенности $E_{ср}$ в каждом опыте и плотность потока излучения по формуле $W=4{,}6\cdot10^{-3}\,E_{ср}$.

Построить график зависимости ЭДС солнечного модуля от плотности потока излучения W, падающего на его поверхность.

При изменении сопротивления нагрузки в цепи была определена вольт-амперная характеристика солнечного модуля (табл. 5.2):

Таблица 5.2.

U, В	0	2,5	5	7,5	10	12,5	14,5
I, мА	195	190	185	175	140	120	75
$\boldsymbol{N}$, Вт							

Для каждого измерения вычислить электрическую мощность $N = UI$, вырабатываемую солнечным модулем, и найти ее наибольшее значение.

Построить графики зависимостей силы тока I и мощности N от напряжения U.

Методика расчетов

1. Ввести опытные данные в виде матрицы значений освещенности `E`, векторов `EDS`, `U` и `I`.
2. Выполнить вычисления.
3. Построить график зависимости ЭДС солнечного модуля (вектор `EDS`) от плотности потока излучения W (вектор `W`). Нанести маркеры для опытных данных. Нанести надписи и сетку.
4. В одном графическом окне в двух подокнах построить графики зависимостей силы тока `I` и мощности `N` от напряжения U. Подписать оси.
5. Отредактировать графики в графическом окне MATLAB.

Лабораторная работа № 6. Трехмерная графика. Распределение температуры на пластине

Цель занятия: научиться строить трехмерные графики.

Технологии построения графиков

Формирование сетки

`[X,Y]=meshgrid(x,y)` преобразует область определения, заданную векторами `x` и `y`, в матрицы `X` и `Y`, используемые при вычислении функции двух переменных и построении трехмерных графиков. Строки матрицы `X` дублируют вектор `x`, а столбцы `Y` – вектор `y`;

`[X,Y]=meshgrid(x)` – аналогична `[X,Y]=meshgrid(x,x)`.

Построение трехмерных линий

Команда `plot3` является трехмерным аналогом команды `plot`.

`plot3(x,y,z)` строит массив точек, представленных векторами `x,y,z` одинаковой размерности, и соединяет их отрезками прямых.

`plot3(X,Y,Z)` строит линии по точкам, координаты которых берутся из столбцов матриц `X, Y, Z`.

Пример. Построение трехмерной спирали (рис.6.1).

```
t=0:pi/50:8*pi;
plot3(sin(t),cos(t),t)
```

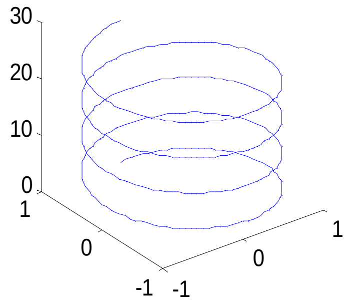


Рис. 6.1. Трехмерная спираль

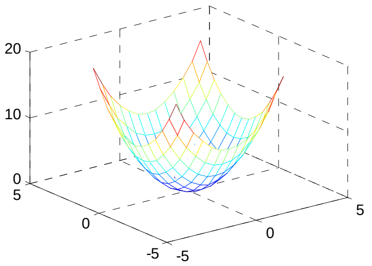
Построение поверхностей

`mesh(X,Y,Z)` строит каркасную поверхность Z(X,Y), где цвет линий, соединяющих заданные точки, определяется высотой поверхности; `mesh(Z)` – строит каркасную поверхность, используя соотношения: `X=1:n` и `Y=1:m`, где `[m,n]=size(Z)` (рис.6.2 а);

`surf(X,Y,Z)` – строит сплошную цветную поверхность по данным матриц `X`, `Y` и `Z`, цвет поверхности задается высотой (рис.6.2 б).

Пример. Построение поверхности параболоида $z = x^2 + y^2$.

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



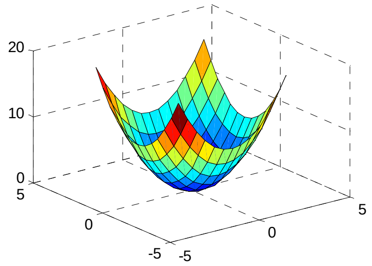


Рис. 6.2. Трехмерные поверхности

Контурные графики

Контурные графики служат для представления на плоскости функции двух переменных вида $z(x,y)$ с помощью линий равного уровня.

`contour(X,Y,Z)` строит линии уровня по данным матрицы `Z` с учетом диапазона изменения координат `x` и `y` с указанием спецификаций (рис. 6.3 а);

`contour(Z)` строит график по данным матрицы `Z` с автоматическим заданием диапазонов изменения `x` и `y`;

`contour(X,Y,Z,N)` строит контурные графики с явным заданием `N` линий равного уровня;

`contour(X,Y,Z,V)` строит линии равного уровня для высот, указанных значениями элементов вектора `V`;

Графики полей градиентов

`quiver(x,y,dx,dy)` формирует и выводит на экран поле градиентов функции в виде стрелок для каждой пары элементов массивов `x` и `y`, а пары элементов `dx, dy` используются для указания направления и размера стрелки (рис. 6.3 б);

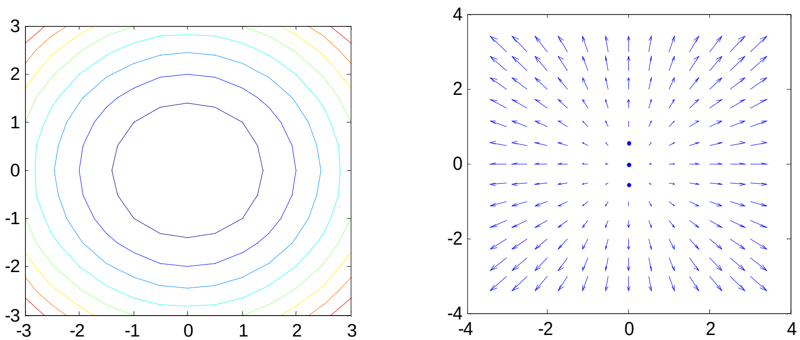
`quiver(dx,dy)` строит векторы в равнорасположенных точках на плоскости (x, y);

`quiver(...,scale)` автоматически масштабирует векторы, а затем вытягивает их в соответствии с коэффициентом `scale`.

Функция `[dx,dy]=gradient(F,h1,h2)` вычисляет градиент функции $F(x,y)$, заданной на двумерной сетке и представляющей собой массив чисел. Шаг разбиения сетки `h1,h2` может быть опущен.

Пример. Построение контурного графика и поля градиентов для функции $z = x^2 + y^2$:

```
[X,Y]=meshgrid([-3:0.5:3]);
Z=X.^2+Y.^2;
contour(X,Y,Z)
figure
[dx, dy]=gradient(Z);
quiver(X,Y,dx,dy)
```



а) б)

Рис. 6.3. Контурный график и поле градиентов

Интерактивный режим

Для форматирования трехмерных графиков в графическом окне имеется специальная панель инструментов **Camera** (Камера).

Несмотря на множество кнопок, пользование панелью инструментов **3D**-графики достаточно просто, если представить себе, что вы смотрите на предмет через объектив фотокамеры. Наглядные рисунки на кнопках поясняют смысл их действия — это перемещение и вращение 3D-рисунков относительно тех или иных координатных осей, включение отображения перспективы, изменение цветовой схемы и др.

Контрольные вопросы

1. Как построить график линии в трехмерном пространстве?
2. Как задать область определения для трехмерного графика?
3. Какие команды предназначены для построения поверхностей?
4. В чем различие функций `plot3, mesh, surf`?
5. Как построить график в виде линий равного уровня?
6. Как построить поле градиентов функции?
7. Какие возможности форматирования трехмерных графиков предоставляются в графическом окне MATLAB?

Порядок выполнения работы

1. Построить график линии, заданной параметрически:

$$x = 0.5 \cdot t \cdot \cos t,\ y = 0.5 \cdot t \cdot \sin t,\ z = 0.2t, \quad t \in [0;50]$$

2. Построить графики поверхностей, контурные графики и поля градиентов (по вариантам). Значения параметров a, b и c выбрать самостоятельно, $x, y \in [-2;2]$.

1) поверхность $z = 0.5 \cdot e^{-(x^2+y^2)}$;

2) эллиптический параболоид $z = \dfrac{x^2}{a^2} + \dfrac{y^2}{b^2}$;

3) гиперболический параболоид $z = \dfrac{x^2}{a^2} - \dfrac{y^2}{b^2}$;

4) верхнюю часть эллипсоида $\dfrac{x^2}{a^2} + \dfrac{y^2}{b^2} + \dfrac{z^2}{c^2} = 1$;

5) верхнюю часть конуса $\frac{x^2}{a^2}+\frac{y^2}{b^2}-\frac{z^2}{c^2}=0$;

6) верхнюю часть двуполостного гиперболоида

$$\frac{x^2}{a^2}+\frac{y^2}{b^2}-\frac{z^2}{c^2}=-1;$$

7) верхнюю часть однополостного гиперболоида

$$\frac{x^2}{a^2}+\frac{y^2}{b^2}-\frac{z^2}{c^2}=1;$$

8) поверхность $z=\sin x\cdot\cos(y+\pi/2), \quad x,y\in[0;\pi]$.

Постановка задачи

С помощью инфракрасного термометра измерена температура в узловых точках равномерной сетки на поверхности пластины (рис.6.4, табл. 6.1). Построить трехмерный график распределения температуры, изотермы и поле градиентов температуры.

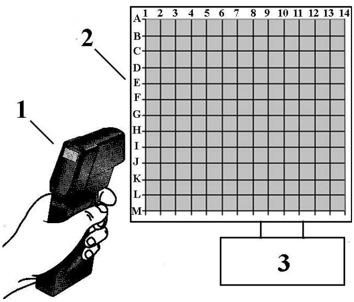


Рис. 6.4. Схема лабораторной установки.
1 – инфракрасный термометр;
2 – металлическая пластина;
3 – источник питания нагревателей на пластине

Таблица 6.1.

	1	2	3	4	5	6
A	22,2	22,2	22,1	22	22	21,8
B	22,5	22,6	22,6	22,4	22,3	22
C	23	23,4	23,3	23	22,9	22,6
D	23,1	23,6	24	23,5	23,1	22,7
E	23,2	24,2	25,9	24,4	23,4	22,8
F	23,4	23,9	24,5	23,9	23,2	22,6
G	23,1	23,2	23,4	23,1	22,9	22,6

Лабораторная работа № 7. Программирование в среде MATLAB

Цель занятия: ознакомиться со структурой программ MATLAB, научиться составлять программы для расчетов в среде MATLAB.

Основные положения

Программы на языке программирования MATLAB сохраняются в виде текстовых файлов, имеющих расширение *.m (*m-файлов*). Различают два типа m-файлов – *файлы-сценарии* и *файлы-функции*.

Язык программирования MATLAB является интерпретатором. Это означает, что MATLAB не создает исполняемых конечных программ. Для выполнения программ необходима среда MATLAB.

Программы создаются в окне Редактора, вызываемого щелчком по кнопке **New M-File** на панели инструментов (рис.7.1).

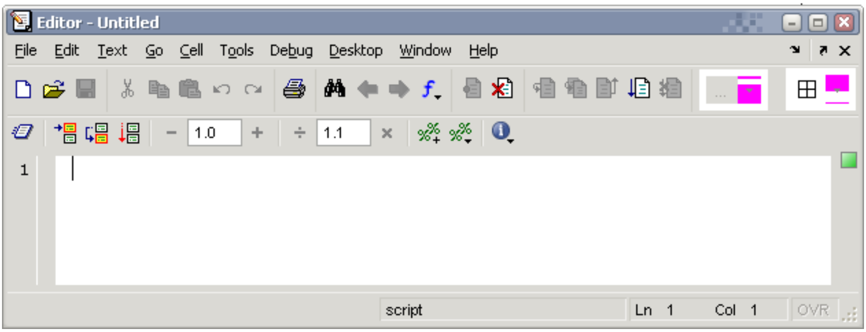


Рис. 7.1. Окно Редактора MATLAB

Структура файлов-сценариев

Файл-сценарий (script) – это m-файл, в котором записана серия команд без входных и выходных параметров. Файлы-сценарии работают с данными из рабочей области. Переменные, используемые в файлах-сценариях, являются *глобальными*, т. е. они действуют одинаково в командах сессии и внутри файла-сценария. Каждый сценарий имеет следующую структуру:

```
%Комментарий
Операторы
```

Записанные в файле-сценарии команды будут исполнены, если в командной строке ввести его имя и нажать **Enter**.

Пример. В окне Редактора введем текст сценария:

```
%Построение графика
%Строит график функции  y= 2*sin(5*x)^2
x = -pi:0.01:pi;
polar(x, 2*sin(5*x).^2)
```

После сохранения под именем `pl` в текущем каталоге будет создан файл `pl.m`.

По команде `pl`, записанной в командном окне, программа запускается. Команда `type pl` выводит листинг файла.

Структура m-файлов функций

М-файл функция содержит входные и выходные параметры и использует аппарат *локальных* переменных. Локальные переменные доступны только в пределах данной функции. Структура функции:

```
function [var1,var2,…]=f_name(Список параметров)
%Комментарий
Команды
var1=…
var2=…
…
```

Связь с другими модулями проходит через входные и выходные параметры. При вызове функции происходит замена формальных параметров фактическими параметрами.

Пример. Построить функцию, вычисляющую факториал числа n.

```
function f = fact(n)
%FACT(N) факториал числа N
f=prod(1:n);
```

Используя созданную функцию, вычислить значение 3!.

```
>>fact(3)
```

Подфункции объявляются и записываются в теле основных функций и имеют аналогичную структуру. Подфункции доступны только в пределах m-файла, определяющего основную функцию.

Операторная функция

Операторная функция определяет функцию с помощью одного оператора.

```
g=inline('выражение')
g=inline('выражение','аргумент1','аргумент2', ...)
```

Здесь `'выражение'` – строковая константа, определяющая математическое выражение, `'аргумент1'`,`'аргумент2'`, ... – список входных параметров (аргументов) этого выражения.

Определения операторных функций должны предшествовать их вызову. К операторной функции можно обращаться только в том программном модуле, в котором она определена.

Примеры:

```
>>g=inline('t.^2')
>>a=2+g(3)
>>fplot(g,[0 2])

>>g1=inline('sin(alpha*x)','x','alpha')
>>g1(3.149, 0.5)
```

Передача данных через глобальные переменные

Поскольку переменные в функциях являются *локальными*, то передача данных из модуля в модуль происходит только через входные и выходные параметры. Команда `global X Y Z` объявляет переменные `X, Y, Z` модуля-функции *глобальными*. Чтобы несколько программных модулей могли совместно использовать глобальную переменную, ее идентификатор должен быть объявлен как `global` во всех этих модулях.

Пример.

```
function m = massa(volume)
global DENSITY
m=DENSITY*volume
```

В командной строке наберем

```
>>global DENSITY, DENSITY=1;
>>massa(3)
ans=3
```

Параметры функционального типа

Часто во многих функциях возникает необходимость передачи им в качестве параметров имени других функций. В таких случаях в MATLAB используют параметры функционального типа. `@fname`.

Здесь `fname` – имя функции, `@fname` –указатель функции. Для определения значения функции с помощью ее функционального указателя можно использовать функцию `feval`.

Пример:

```
s=@sin;
feval(s, 1)
```

Вычислить сумму, когда вид функции *F* заранее неизвестен.

```
function f=sm(fh)
f=0;
for i=1:10
    f=f+feval(fh,i);
end;
```

Тогда для вычисления суммы, например, функции sin(x) достаточно выполнить оператор: `sm(@sin)`

Контрольные вопросы

1. Что собой представляют программы в MATLAB?
2. В каком окне создаются программы MATLAB?
3. Какие типы программ используются в MATLAB?
4. Что такое файл-сценарий, какова его структура?
5. Что такое m-файл функция, какова его структура?
6. Чем отличается m-файл функция от файла-сценария?
7. Для чего используются комментарии в программах?
8. Какой командой можно вывести листинг файла?
9. Как запустить на выполнение файл-сценарий?
10. Как обратиться к файлу-функции?
11. Где записываются входные параметры программы-функции?
12. Может ли в функции быть несколько выходных параметров?
13. Что такое локальные и глобальные переменные?
14. Как осуществляется передача данных из модуля?
15. Что такое операторная функция?
16. Как передать в качестве параметра имя другой функции?
17. Как описать переменную функционального типа?

Порядок выполнения работы

1. Написать программу сценарий (по вариантам):
   1) Нахождение корней квадратного уравнения $ax^2+bx+c=0$
   $$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$
   2) Нахождение площади S треугольника по сторонам a, b, c.
   $S = \sqrt{p(p-a)(p-b)(p-c)}$, где $p=(a+b+c)/2$.
   3) Нахождение определителя и обратной матрицы A.
   4) Нахождение максимального и среднего значения матрицы A.

2. Написать функцию для преобразования величины, выраженной в старых единицах измерения, в новые единицы измерения (табл. 7.1):

Таблица 7.1.

Тип	старые ед. изм.	Новые ед. изм.
время	час	секунда
скорость	км/час	м/с
давление	атм техн.	Паскаль (1 атм=$9{,}80665\cdot10^4$ Па)
давление	мм рт.ст.	Паскаль (1 мм рт.ст.=133,32 Па)
температура	град.Фаренгейта	°С (t°С= (t_f -32°F)·5/9)
объем	баррель	$м^3$ (1 баррель нефт. США = 0,15899 $м^3$)
мощность	лошад. силы	Ватт (1 л. с. =735,49875 Вт)
энергия	калория	Джоуль (1 кал=4,19 Дж)

3. Написать программу-функцию, которая вычисляет:
   1) площадь круга и длину окружности по заданному диаметру;
   2) площадь и длину дуги сектора по заданному радиусу и углу;
   3) площадь поверхности и объем кругового цилиндра по заданному радиусу и высоте.
   4) объем и массу шара заданного радиуса и плотности.

4. Написать операторную функцию и построить с помощью команды `fplot` график этой функции на интервале от x_1 до x_2.

1) $f = \ln^2 x / x, \quad x_1 = 1, x_2 = 4$;

2) $g = \dfrac{1}{x^2} \cdot \sin\dfrac{1}{x}; \quad x_1 = 1.5; x_2 = 4.5$;

3) $q = x \cdot arctgx; \quad x_1 = 0; x_2 = 2$;

4) $p = \dfrac{\sqrt{x}}{x^2 + 1}; \quad x_1 = 0; x_2 = 3$.

5. Написать функцию для перевода валют. Курс валюты передавать через глобальную переменную.

Лабораторная работа № 8. Управляющие конструкции языка MATLAB. Определение гидравлического сопротивления

Цель занятия: ознакомиться с логическими операторами, овладеть навыками алгоритмизации и программирования вычислительных процессов нелинейной структуры.

Основные положения

Помимо программ с *линейной структурой*, инструкции которых исполняются строго по порядку, существуют программы, структура которых *нелинейна*. При этом ветви программ могут выполняться в зависимости от определенных условий, либо часть программы может повторяться конечное число циклов, либо повторы могут завершаться при выполнении заданного условия.

В MATLAB имеются специальные структуры управления последовательностью выполнения операторов, входящих в программу.

Логические операции

Операторы отношения (табл. 8.1) выполняют поэлементное сравнение двух величин, векторов или матриц одинаковой размерности и возвращают значение `1 (True)`, если элементы идентичны, и значение `0 (False)` в противном случае.

Таблица 8.1

Функция	Оператор	Описание	Пример
`lt`	<	Меньше чем	X<Y
`le`	<=	Меньше или равно	X<=Y
`gt`	>	Больше чем	X>Y
`ge`	>=	Больше или равно	X>=Y
`eq`	==	Равно	X==Y
`ne`	~=	Не равно	X~=Y

Логические операторы и соответствующие им функции реализуют логические операции над массивами одинаковой размерности (табл.8.2). Число ноль воспринимается как логический ноль, а любое отличное от нуля значение элемента массива – как логическая единица.

Таблица 8.2

Оператор	Функция	Название
&	and	Логическое И
|	or	Логическое ИЛИ
~	not	Логическое НЕ
	xor	Исключающее ИЛИ
	any	Истина, если все элементы массива нулевые
	all	Истина, если все элементы массива ненулевые

Приоритет выполнения операторов

Операторы в порядке убывания их приоритета:

1) выполняются операторы, заключенные в скобки ()
2) транспонирование (.'), возведение в степень (.^), комплексно-сопряженное транспонирование ('), матричное возведение в степень (^)
3) унарное сложение (+), унарное вычитание (–), логическое отрицание (~)
4) умножение (.*), правое деление (./), левое деление (.\), матричное умножение (*), матричное правое деление (/), матричное левое деление (\)
5) сложение (+), вычитание (–)
6) оператор двоеточие (:)
7) меньше чем (<), меньше или равно (<=), больше чем (>), больше или равно (>=), равно (==), не равно (~=)
8) логическое И (&)
9) логическое ИЛИ (|)

Можно изменить принятый по умолчанию порядок выполнения операторов с помощью круглых скобок.

Циклы с определенным числом повторений

```
for V=A:H:B
    команды
end
```

```
for V=A:B
    команды
end
```

где `V` -переменная/параметр цикла, `A,B` – начальное и конечное значения; `H` – приращение; при его отсутствии по умолчанию принимает значение `1`. В заголовке цикла можно использовать одномерный массив. Допускаются и вложенные циклы.

Пример. Цикл формирует массив, состоящий из элементов 2^i, (i=0; 5; 7).

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

Условный оператор if

```
if условие
   команды
end
```

```
if  условие
    команды
else
    команды
end
```

```
if  условие
    команды
elseif условие
    команды
else
    команды
end
```

Оператор `if` вычисляет логическое выражение `условие` и выполняет группу операторов `команды`, если выражение истинно. Необязательные ключевые слова `elseif` и `else` служат для выполнения альтернативных групп операторов. Ключевое слово `end`, которое согласуется с `if`, завершает последнюю группу операторов.

Пример. Построить квадратную трехдиагональную матрицу, у которой значения всех элементов на главной диагонали равны двум, значения элементов на выше- и нижележащих диагоналях равны минус единице.

```
for i=1:5
 for j=1:5
  if i==j
     a(i,j)=2;
   elseif abs(i-j)==1
     a(i,j)=-1;
    else
     a(i,j)=0;
  end
 end
end
```

Оператор переключения switch

```
switch  выражение
   case  значение 1
         команды
   case  значение 2
         команды
         . . . . .
   otherwise        % может отсутствовать
         команды
end
```

Оператор `switch` выполняет группу операторов, базируясь на значении переменной или выражения «`выражение`». Ключевые слова `case` и `otherwise` разделяют эти группы. Выполняется только первый соответствующий случай. Необходимо использовать `end` для согласования со `switch`.

Пример. Предыдущий пример можно переписать с использованием оператора `switch`:

```
for i=1:5
 for j=1:5
  switch i-j
   case 0
    a(i,j)=2;
   case {1,-1}
    a(i,j)=-1;
```

```
    otherwise
      a(i,j)=0;
    end
  end
end
```

Оператор цикла с неопределенным числом повторений

```
while условие
      команды
end
```

Оператор обеспечивает выполнение команд тела цикла, пока истинно проверяемое условие. Ключевое слово `end` ограничивает тело цикла.

Пример. Программа иллюстрирует работу операторов `while` и `if` для нахождения одного из корней многочлена $x^3–2x–5$. В программе реализован метод деления пополам.

```
a=0; fa=-Inf;
b=3; fb=Inf;
while b-a > eps
  x=(a+b)/2;
  fx=x^3-2*x-5;
  if sign(fx)==sign(fa)
    a=x; fa=fx;
   else
    b=x; fb=fx;
  end
end
```

Для вывода результата работы программы необходимо в командной строке набрать `x`:

```
>>x
x =
    2.0946
```

Прерывание циклов и паузы

Оператор `break` позволяет досрочно выходить из циклов `for` или `while`. Во вложенных циклах `break` осуществляет выход только из того цикла, в который он вложен.

Для организации паузы в ходе выполнения программы служит команда `pause`. Выполнение программы будет продолжаться после нажатия любой клавиши. Для организации паузы в `n` секунд, следует использовать команду `pause(n)`.

Диалоговый ввод и вывод текста

```
переменная=input('текст')
```

На экран выводится запрос `'текст'` в виде строки, затем происходит остановка работы и ожидание ввода с клавиатуры.

При использовании команды `input` в виде:

```
переменная=input('текст','s')
```

вводимые с клавиатуры символы воспринимаются системой как строковая константа.

Пример:

```
i = input('Введите значение i=');
k = input('Do you want more? Y/N [Y]: ', 's');
```

Текст можно выводить по команде:

```
disp('текст')
```

Контрольные вопросы

1. Перечислите операторы отношения.
2. Как выполняются операторы отношения?
3. Перечислите логические операции. Что является результатом выполнения логических операций.
4. Перечислите операторы в порядке убывания их приоритета исполнения.
5. Как можно изменить принятый по умолчанию порядок выполнения операторов?

6. Какие операторы предназначены для организации циклов?
7. Опишите структуру операторов `for, while`.
8. Какой оператор позволяет досрочно выйти из цикла?
9. Какие операторы предназначены для организации разветвлений?
10. Опишите структуру операторов `if, switch`.
11. Как организовать паузу в ходе выполнения программы?
12. Каким образом можно ввести данные в диалоговом режиме?

Выполнение работы

1. Выполнить упражнения (по вариантам). Вычислить значение выражения:

$$1)\quad Q=\begin{cases}\pi x^2-\dfrac{7}{x^2}\pi & x>1{,}4\\ x^3+7\sqrt{x} & x=1{,}4\\ \ln\left(x+7\sqrt{|x+\pi|}\right) & x<1{,}4\end{cases}$$

$$2)\quad R=\begin{cases}(x+y)^2-\sqrt{x\cdot y}, & x\cdot y>0\\ (x+y)^2-\sqrt{|x\cdot y|}, & x\cdot y<0\\ (x+y)^2+1, & x\cdot y=0\end{cases}$$

$$3)\quad P=\begin{cases}x^3\cdot\sqrt{x-a} & x>a\\ x\sin ax & x=a \qquad a=2{,}5\\ e^{-ax}\cos ax & x<a\end{cases}$$

$$4)\quad N=\begin{cases}\left(\ln^3 x+x^2\right)/\sqrt{x+2} & x<0.5\\ \sqrt{x+2}+1/x & x=0{,}5\\ \cos x+\sin^2 x & x>0{,}5\end{cases}$$

5) $F = \begin{cases} \sin x \lg x & x > 3{,}5 \\ \cos^2 x & x \le 3{,}5 \end{cases}$

6) $W = \begin{cases} \lg(x+1) & x > 1 \\ \sin^2 x\sqrt{|x|} & x \le 1 \end{cases}$

7) $V = \begin{cases} \sqrt{x^2 + \sin x + 1} & x < 0{,}1 \\ x+1 & x = 0{,}1 \\ \sqrt{x^2 + \cos x + 1} & x > 0{,}5 \end{cases}$

8) $H = \begin{cases} 2x^2 + x + 1 & x < 1{,}2 \\ 1/x + \sqrt{x^2+1} & x = 1{,}2 \\ (1+x)/\sqrt{x^2+1} & x > 1{,}2 \end{cases}$

2. Написать программу для выполнения действий:
   1) вычисление суммы положительных значений матрицы;
   2) вычисление суммы отрицательных значений матрицы;
   3) определение количества нулевых элементов матрицы;
   4) формирование вектора из наибольших элементов строк матрицы;
   5) формирование вектора из положительных элементов матрицы;
   6) замена отрицательных элементов матрицы нулевыми;
   7) нахождение максимального элемента матрицы и его номера;
   8) нахождение минимального элемента матрицы и его номера

3. Заданы вектора координат точек X и Y. Подсчитать количество точек, находящихся внутри:

1) круга радиуса $r = 3$, с координатами центра (1; 2);

2) прямоугольника, у которого верхний левый и нижний правый углы имеют координаты (1; 4) и (6; 1,5) соответственно.

4. Вычислить выражение, число n ввести с клавиатуры.

$$q = \left(1 + \frac{1}{2} + \frac{1}{2^2} + \frac{1}{2^3} + \ldots \frac{1}{2^n}\right)^n$$

Постановка задачи

Определить коэффициент гидравлического сопротивления λ для разных режимов течения и труб из различных материалов.

Режим течения жидкости определяется числом Рейнольдса Re. Законы сопротивления представлены в таблице 8.3.

Таблица 8.3.

Зона сопротивления	Закон сопротивления
Ламинарное течение Re<2300	$\lambda = 64/\mathrm{Re}$
Зона гидравлически гладких труб при турбулентном режиме течения $2300 < \mathrm{Re} < 20\dfrac{d}{K_э}$	$\lambda = \dfrac{0{,}3164}{\mathrm{Re}^{0,25}}$, или $\lambda = \dfrac{1}{(1{,}8 \cdot \lg \mathrm{Re} - 1{,}64)^2}$
Переходная зона $20\dfrac{d}{K_э} < \mathrm{Re} < 500\dfrac{d}{K_э}$	$\lambda = 0{,}11\left(\dfrac{K_э}{d} + \dfrac{68}{\mathrm{Re}}\right)^{0,25}$
Зона квадратичного сопротивления $500\dfrac{d}{K_э} < \mathrm{Re}$	$\lambda = \dfrac{1}{\left(1{,}74 + 2\lg\dfrac{d}{K_э}\right)^2}$ или $\lambda = 0{,}11\left(\dfrac{K_э}{d}\right)^{0,25}$

В таблице 8.4 представлены значения эквивалентной шероховатости $K_э$ для новых труб из различных материалов.

Таблица 8.4.

Тип	Материал труб, состояние труб	$К_э$ мм
1	Тянутые из стекла и цветных металлов	0,005
2	Стальные бесшовные	0,03
3	Стальные сварные	0,5
4	Стальные оцинкованные	0,15
5	Чугунные	0,3
6	Бетонные с грубой поверхностью	6
7	Шланги резиновые	0,03

Методика расчетов

Ввести тип трубы. Задать диаметр труб (от 32 мм до 2 м, стальные – 50 -100 мм).

Задать значения эквивалентной шероховатости К с помощью оператора `switch` для различных типов труб.

Определить коэффициент гидравлического сопротивления для разных режимов течения с помощью оператора `if`.

Указание: Поскольку проверок больше трех, то необходимо использовать вложенные условные операторы. В таблице 8.3 для некоторых значений Re приведены формулы разных авторов. Для расчетов использовать одну из формул.

Лабораторная работа № 9. Нахождение минимумов и нулей функций. Оптимальный размер бака

Цель занятия: научиться использовать функции MATLAB для решения задач численного анализа.

Основные положения

Нахождение минимума функции

Для нахождения минимума функции одной переменной на заданном интервале [a, b] следует использовать функцию:

```
fminbnd('fun',a,b)
```

Пример. Найти на интервале [0,2; 1] минимум функции

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

```
f=inline('1./((x-0.3).^2+0.01)+1./((x-0.9).^2+ 0.04)-6');
x=fminbnd(f,0.2,1)
x =
  0.6370
```

Для минимизации функций нескольких переменных можно использовать функцию `fminsearch`. Одна их форм обращения к этой функции выглядит следующим образом:

```
X=fminsearch('fun', X0)
```

где `X0` – начальное приближение, а `X` – найденное значение локального минимума функции `'fun'`. `X0` может быть скаляром, вектором или матрицей.

Пример. Поиск минимума тестовой функции Розенброка:

$$f(x)=100(x_2-x_1^2)^2+(1-x_1)^2$$

```
>>f=inline('100*(x(2)-x(1)^2)^2+(1-x(1))^2;');
>>x=fminsearch(f,[-1.2, 1])
x =
   1.0000   1.0000
```

Нахождения нулей функции

```
fzero('fun',x)
```

Функция находит нуль действительной функции от одной переменной, x – начальное приближение или интервал поиска.

Начальное значение можно найти графическим способом: построить график функции и найти пересечение его с осью X.

Контрольные вопросы

1. Какая функция MATLAB позволяет найти минимум функции?
2. Как найти нуль функции?
3. Как найти координаты пересечения графиков двух функций?

Выполнение работы

1. Найти минимум функции

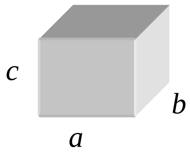
1) $y=\dfrac{2x}{x^2+1}$;	$y=-\dfrac{\sqrt{x}}{(x^2+1)}$;
2) $y=3\cdot e^x-5x-2$;	$y=-e^{-x}\cdot x$;
3) $y=\sqrt{x^2-6x+25}$;	$y=\dfrac{1}{2}e^{x^2}$
4) $y=e^x-2x$	$y=\sqrt{x^2-2x+5}$

2. Решить уравнение:
   1) $e^x\cos x=0$;
   2) $x\cdot 2^x=1$;
   3) $e^x-2=x$
   4) $\sqrt{x+1}=x-1$;
   5) $x-2\sqrt{x}-15=0$;
   6) $2\lg x=\lg(5x-4)$;
   7) $\sqrt{x+25}=x-5$;
   8) $3\cos x=\cos^2 x$;

Постановка задачи

Определить размеры бака заданного объема, чтобы на его изготовление пошло как можно меньше материала, т. е. минимизировать площадь поверхности.

Бак имеет форму прямоугольного параллелепипеда.



$V = 0{,}9$ м^3

Рис. 9.1. Геометрия задачи

Методика расчета

Составить уравнение площади поверхности бака S относительно сторон основания.

$$S = 2 \cdot \left(a \cdot b + (a + b) \cdot c\right)$$

Третью сторону выразить через заданный объем V.

$$V = a \cdot b \cdot c; \quad \text{тогда} \quad c = \frac{V}{a \cdot b}$$

$$S = 2 \cdot \left(a \cdot b + (a + b) \cdot \frac{V}{a \cdot b} \right)$$

Необходимо найти минимум функции S от двух переменных.

По умолчанию в MATLAB переменные являются элементами вектора `x`, поэтому при записи выражения в функции `fminsearch` вместо *a* и *b* используйте обозначения `x(1)` и `x(2)`, вместо буквенного обозначения объема V подставьте численное значение.

Начальные размеры сторон выберите из практических соображений.

Лабораторная работа № 10. Аппроксимация данных. Определение модуля упругости

Цель занятия: научиться находить функциональные зависимости, описывающие экспериментальные данные, получить навыки работы с полиномами в MATLAB.

Основные положения

Представление полиномов в среде MATLAB

В системе MATLAB полином (многочлен) представляется с помощью вектора-строки, содержащего упорядоченные по убыванию степени коэффициенты полинома.

Пример. Чтобы представить многочлен $p(x) = x^3 - 2x - 5$, следует создать вектор

```
p = [1 0 -2 -5].
```

Для вычисления значений многочлена `p` в точках `x` следует использовать функцию `polyval(p,x)`.

```
>>p = [1 0 -2 -5];
>>polyval(p,5)
ans =
     110
```

`conv(a,b)` – умножение полиномов *a*, *b*.

`[q,r]=deconv(a,b)` – деление полиномов (получение частного и остатка).

`roots(p)` – вычисление корней многочлена *p*.

`poly(r)` – возвращает коэффициенты многочлена по его корням `r`. Корни `r` задаются в виде вектора.

```
>>q = poly([2 3])
q =
   1   -5   6
```

`polyder(p)` вычисление производной многочлена *p*.

Интерполяция

Задача *интерполяции* состоит в следующем: найти функцию, которая проходит через заданные точки и получить значения в промежуточных точках.

При интерполяции сплайнами на каждом отдельном интервале строится кусочно-непрерывная функция (сплайн).

```
yy = spline(x,y,xx)
```

Функция использует интерполяцию кубическими сплайнами (полиномами третьего порядка) для нахождения `yy`, вектора значений искомой функции в точках, заданных вектором `xx`. Вектор `x` содержит точки, в которых заданы исходные данные `y`. Если `y` матрица, то интерполяция выполняется для каждой строки матрицы `y`.

```
pp = spline(x,y)
```

Функция возвращает кусочно-полиномиальную форму кубических сплайнов, которая в дальнейшем может использоваться функцией `ppval(pp,XX)` для вычисления значений этой формы в точках `XX`)

Пример.

```
>> x=0:10; y=3*cos(x);
>> x1=0:0.1:11; y1=spline(x,y,x1);
>> plot(x,y,'o',x1,y1,'--')
```

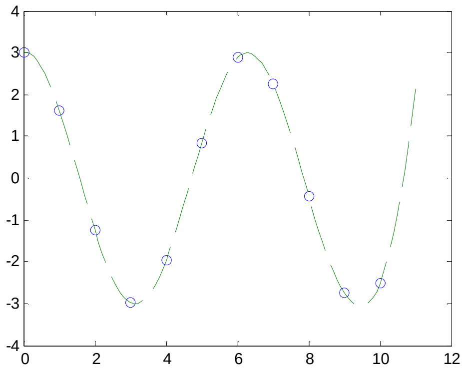


Рис. 10.1. Пример применения функции spline

Аппроксимация

Функцию, приближенно описывающую экспериментальные данные и сглаживающую случайные отклонения, называют *аппроксимирующей*, а сам процесс построения функции – *аппроксимацией*.

Для построения аналитической зависимости используют метод наименьших квадратов. Метод базируется на применении в качестве критерия близости сумму квадратов отклонений заданных и расчетных значений.

Метод наименьших квадратов реализуется в системе MATLAB при решении переопределенных систем уравнений (т.е. систем, в которых число уравнений больше числа неизвестных).

Пример. Заданы значения функции y в моменты времени t:

t	0.0	0.3	0.8	1.1	1.6	2.3
y	0.82	0.72	0.63	0.6	0.55	0.5

Будем искать приближение табличной функции в виде $y(t) \approx c_1 + c_2 e^{-t}$, где c_1 и c_2 – неизвестные коэффициенты. Вектор `y` разыскивается в виде линейной комбинации вектора с единичными компонентами и вектора с компонентами `exp(-t)`.

Введем экспериментальные данные в виде векторов-столбцов. Построим матрицу системы `A`, содержащую коэффициенты при неизвестных c_1 и c_2, найдем вектор решения системы уравнений `A·C=y`:

```
>>t=[0 0.3 00.8 1.1 1.6 2.3]';
>>y=[0.82 0.72 0.63 0.60 0.55 0.50]';
>>A=[ones(size(t)) exp(-t)]
>>C=A\y
C=
   0.4760
   0.3413
```

Таким образом, аппроксимирующая функция имеет вид: $y(t) \approx 0.4760 + 0.3413 e^{-t}$.

Построим график аппроксимирующей функции вместе с исходными табличными данными (рис. 10.2):

```
>>T=(0:0.1:2.5)';
>>Y=[ones(size(T)) exp(-T)]*C;
>>plot(T,Y,'-',t,y,'o')
```

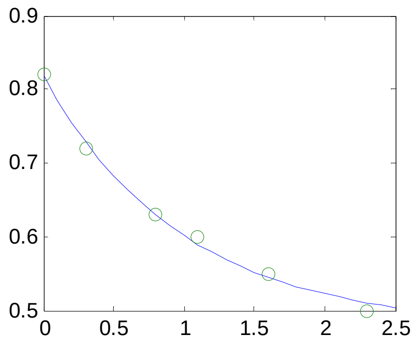


Рис. 10.2. Построение аппроксимирующей функции

Аппроксимация полиномом n-й степени

```
p=polyfit(x,y,N),
```

где `x,y` – исходные данные, `N` – порядок полинома, `p` – вектор его коэффициентов.

Пример.

```
>>x=[0 1 2 3 4 5];
>>y=[-3  2  6  11  16  21];
>> p=polyfit(x,y,1)
p =
  4.7714 -3.0952
```

Полученная функция $p(x) = 4{,}7714x - 3{,}0952$.

Контрольные вопросы

1. Как задается полином в MATLAB?
2. Как вычислить значение полинома в заданных точках?

3. Как найти произведение, частное и остаток от деления двух полиномов?

4. Как найти корни полинома?

5. Как построить полином по его корням?

6. Как вычислить производную от многочлена?

7. Что такое интерполяция и аппроксимация?

8. Какой метод используется для нахождения коэффициентов функции, аппроксимирующей экспериментальную зависимость?

9. Какая функция MATLAB используется для аппроксимации данных полиномом?

Порядок выполнения работы

1. Представить в MATLAB полином $p(x)$ и выполнить следующие операции: (по вариантам):

1) $3x^3 - 4x^2 + 5x - 6$
2) $x^3 - 10x^2 + 35x - 27$
3) $x^3 - 11x^2 + 31x - 21$
4) $3x^4 + 4x^3 - 12x^2 - 5$
5) $x^4 - x - 1$
6) $3x^4 + 8x^3 + 6x^2 - 10$
7) $x^4 - 18x^2 + 5x - 8$
8) $x^4+4x^3-12x^2+1$

– Вычислить значение полинома $p(x)$ в точке $x = 3$.
– Найти корни этого полинома.
– Построить график полинома $p(x)$.
– Умножить полином $p(x)$ на полином $x^2 - x - 1$.
– Разделить полином $p(x)$ на полином $x + 1$.
– Найти производную от полинома $p(x)$.
– Известны корни полинома $x_1 = 2{,}4$ и $x_2 = 5$. Построить полином $q(x)$ по его корням.

2. В таблицах представлены экспериментальные данные. Имеется предположение, какой зависимостью они могут быть аппроксимированы. С помощью решения переопределенных систем уравнений найти значения коэффициентов, построить график аппроксимирующей функции вместе с исходными табличными данными:

1) $Y = C_1 + C_2 x$

x	-26	-22	-16	-11	-5	3	10	25	42
y	66.7	71	76.3	80.6	85.7	92.9	99.4	113.6	125.1

2) $Y = C_1 + C_2\sqrt{x}$

x	0.1	0.3	0.7	1.0	1.4	1.9	2.3	2.9	3.5	4.1	7.0
y	0.85	1.23	1.65	1.9	2.25	2.41	2.61	2.96	3.23	3.49	4.43

3) $Y = C_1\sin(x) + C_2\cos(2x)$

x	0.1	0.3	0.7	1.0	1.4	1.9	2.3	2.9	3.5	4.1	7.0
y	0.53	0.50	0.22	-0.04	-0.28	-0.22	0.10	0.53	0.31	-0.34	0.21

4) $Y = C_1\ln(x) + C_2 / x$

x	0.1	0.3	0.7	1.0	1.4	1.9	2.3	2.9	3.5	4.1	7.0
y	2.73	0.93	0.47	0.53	0.70	0.97	1.15	1.29	1.51	1.67	2.14

5) $Y = C_1\sin(x) + C_2\sin(x)\cos(x)$

x	0.20	0.80	1.34	1.95	2.40	2.98	3.20	3.70	4.10	4.80	5.30	6.10
y	0.31	0.85	0.72	0.16	-0.13	-0.07	0.03	0.17	0.03	-0.61	-0.93	-0.27

6) $Y = C_1 x \cdot \exp(-x) + C_2\cos(x)$

x	0.21	0.80	1.38	2.02	2.43	2.83	3.33	3.80	4.14	5.00	5.32	6.38
y	0.33	0.47	0.38	0.22	0.10	0.026	-0.028	-0.036	-0.016	0.076	0.124	0.17

7) $Y = C_1 x^2 \exp(-0{,}4x^2) + C_2 x$

x	0.21	0.8	1.38	2.02	2.43	2.83	3.33	3.8	4.14	5	5.32	6.38
y	0.31	2.28	3.90	4.03	3.63	3.46	3.11	3.30	3.51	4.08	4.67	5.35

8) $Y = C_1 + C_2 x + C_3 x^2$

x	0	1	2	3	4	5	6	7	8	9
y	298	299	301	304	306	309	312	316	319	322

3. По табличным данным пункта 2 построить кубические сплайны, найти значение в промежуточной точке и нанести полученную точку на график, построенный при выполнении пункта 2.

Постановка задачи

Тестирование материалов в предельных режимах часто включает в себя испытание на разрыв, котором образец материала зажимается в двух оправках и подвергается воздействию силы. На рис. 10.3 приведена типичная кривая зависимости напряжения от относительной деформации образца (отношения его абсолютной деформации к первоначальной длине). Сначала имеет место обратимая деформация (участок AB), затем необратимая деформация (участок BD). Наконец образец рвется (точка D). Точка C называется точкой предельного напряжения или пределом прочности.

Обратимая деформация представлена линейной частью кривой (отрезок AB). Коэффициент пропорциональности, равный тангенсу угла наклона прямой, на которой лежит этот отрезок, к оси относительных деформаций, называется модулем Юнга или модулем упругости.

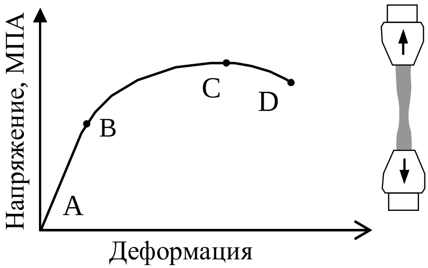


Рис. 10.3. Типичная кривая зависимости напряжения от деформации

В таблице 10.1 приведены данные по испытанию материала. Описать данные линейной функцией и найти модуль упругости.

Таблица 10.1

Деформация	0,0000	0,0015	0,0030	0,0045	0,0060
Напряжение, МПа	0	168	336	504	672

Лабораторная работа № 11. Численное интегрирование. Определение количества теплоты

Цель занятия: научиться использовать функции MATLAB для решения задач численного интегрирования.

Технологии решения

Физический смысл интегрирования – определение площади, совершенной работы, подведенной энергии и т. п.

Функция `trapz(x,y)` вычисляет интеграл от функции `y` по переменной `x`, используя метод трапеций.

Пример.

```
>>x=0:0.001:1; y=x.^3;
>>z=trapz(x, y)
```

Функция `quad` вычисляет определенный интеграл от заданной функции `'fun'` на отрезке `[a,b]` с относительной погрешностью `tol`. Подынтегральная функция может быть строковой константой, указателем функции или операторной функцией. При определении функции следует использовать операторы с точкой. Параметр, указывающий относительную погрешность вычисления интеграла, может быть опущен, по умолчанию он равен `1e-6` (10^{-6}).

```
quad('fun',a,b,tol)
```

Пример. Найти значение определенного интеграла для функции $f(x) = x^3$ на отрезке [0; 1]:

```
>>quad('x.^3', 0, 1)
>>quad('x.^3', 0, 1, 1e-8)   % задана погрешность
```

Контрольные вопросы

1. Какие функции MATLAB вычисляют значение интеграла? Перечислите их параметры.
2. Как найти интеграл от таблично заданной функции?
3. С какой погрешностью вычисляется интеграл?

Порядок выполнения работы

1. Вычислить значения определенных интегралов.

1) $\int_{0}^{0,6} x \cdot tg(x^2+1)dx$	$\int_{0}^{1} \frac{x+1}{\sqrt{x^2+4}} \sin(\pi x)dx$;
2) $\int_{0}^{1} \frac{x^x \cdot \ln\left(1+x^x\right)}{1+x^x}dx$	$\int_{0}^{1} x \cdot (2+x)e^x dx$
3) $\int_{0}^{0.75} \frac{\sin^2(x)}{\sqrt{1-x^2}} e^{-(1+x)}dx$	$\int_{1}^{\pi} \frac{dx}{x\left(1+\ln(x)\right)}$;
4) $\int_{0}^{0.5} \frac{\arcsin(x)}{\sqrt{1-x^2}}dx$	$\int_{0}^{0.5} x \cdot tg\left(x^2+1\right)\ln(x+1)dx$;
5) $\int_{0.5}^{1} \frac{\cos(x)}{x^2+1} e^{-x}dx$	$\int_{0}^{1} \frac{x+1}{\sqrt{x^2+4}} arctg(x)dx$;
6) $\int_{0}^{0.5} \frac{\sin(x)}{\sqrt{1-x^2}}dx$	$\int_{0}^{1} x^3 e^{-x^2} dx$;
7) $\int_{0}^{0.5} \frac{x-\sqrt{1-x^2}}{x+\sqrt{1-x^2}}dx$	$\int_{0}^{1} \sin x \cdot (5+x)e^x dx$;
8) $\int_{1}^{\pi} \frac{dx}{x \cdot (1+\ln(x))}$	$\int_{0}^{1} \frac{x+1}{\sqrt{x^2+4}} \cos(\pi x)dx$.

2. Найти площадь фигуры, ограниченной кривыми. Изобразить фигуру графически, найти координаты узлов, ее ограничивающих, вычислить площадь как определенный интеграл.
   1) $y = \sqrt[3]{x}$ и $y = 1 + 0.07x$;
   2) $y = \ln x$ и $y = 1/x$, а также прямой $y = 3$;
   3) $y = \sin x$ и $y = (\pi - x)/5$ в диапазоне от 0 до π;
   4) $y = \sqrt[4]{x}\ln x$ и $y = 1/x - 1$ и прямой $x = 5$;

5) $y = -(x-2.5)^2 + 4 - 1.5x^2 e^{-0.035x^3}$ и $y = -2$;

6) $y = \sqrt{x}$ и $y = 0.75x$;

7) $y = e^x$ и $y = -1.44x^2 + 1.2x + 1.75$;

8) $y = x^2$ и $y = \cos x$.

Постановка задачи

Имеется табличная зависимость изобарных массовых теплоемкостей газов от температуры (табл. 11.1).

Таблица 11.1

Температура °C	Массовая теплоемкость c_p, кДж/(кг·К)							
	кислород	азот	окись углерода	углекислый газ	водород	воздух	водяной пар	сернистый газ
600	0,9927	1,076	1,0861	1,0396	14,542	1,0496	2,0092	0,737
700	1,0048	1,0869	1,0978	1,0639	14,587	1,0605	2,0419	0,754
800	1,0157	1,0974	1,1091	1,0852	14,641	1,071	2,0754	0,764
900	1,0258	1,1078	1,12	1,1045	14,706	1,0815	2,1097	0,775
1000	1,035	1,1179	1,1304	1,1225	14,776	1,0907	2,1436	0,783
1100	1,0434	1,1271	1,1401	1,1384	14,853	1,0999	2,1771	0,791
1200	1,0509	1,1359	1,1493	1,153	14,934	1,1082	2,2106	0,795
1300	1,058	1,1447	1,1577	1,166	15,023	1,1166	2,2429	
1400	1,0647	1,1526	1,1656	1,1782	15,113	1,1242	2,2743	
1500	1,0714	1,1602	1,1731	1,1895	15,202	1,1313	2,3048	

Количество теплоты, необходимой для нагревания газа от температуры T_1 до температуры T_2, определяется по формуле:

$$q = \int_{T_1}^{T_2} c_p dT$$

Найти количество подведенной теплоты к выбранному газу в данном интервале изменения температур:

Описать зависимость теплоемкости от температуры квадратичным полиномом. Найти значение интеграла. Построить график. Аппроксимировать зависимость в окне графика.

Лабораторная работа № 12. Решение обыкновенных дифференциальных уравнений. Изменение температуры воды

Цель работы: научиться численно решать инженерные задачи, сводящиеся к обыкновенным дифференциальным уравнениям.

Технологии решения

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

В системе MATLAB реализованы различные численные методы для решения обыкновенных дифференциальных уравнений (ОДУ) с заданными начальными условиями (задача Коши) в виде решателей (solvers). Одним из них является функция `ode45`:

```
[t,y]=ode45('ODEFUN',[t0 tf],y0)
```

или `[t,y]=ode45(@ODEFUN,[t0 tf],y0)`

Функция интегрирует систему ОДУ $y' = F(t, y)$ на интервале `[t0 tf]` с начальными условиями $y(t_0)$=`y0`. Функция вычисления правых частей $F(t,y)$ определяется именем `'ODEFUN'`, она должна быть заранее подготовлена. Каждая строка выходного массива `y` есть вектор решения, отвечающий значению независимой переменной из вектора-столбца `t`.

Пример. Найти решение ОДУ $y' = 2t^2 + 2y$ на интервале [0; 1] с начальным условием $y(0) = 2$.

1. Создадим m-файл для вычисления правой части ОДУ. Сохраним его под именем функции `uravnenie`.

```
function dydt=uravnenie(t,y);
dydt=2*t^2+2*y;
```

2. В командном окне введем функцию `ode45` с указанием интервала интегрирования [0; 1] и начального условия y_0=2.

```
[t,y]=ode45(@uravnenie,[0 1],2);
```

3. Построим график полученной таблично функции.

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

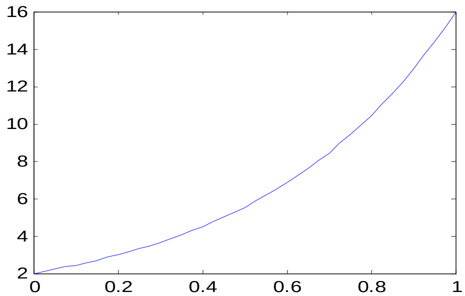


Рис. 12.1. График решения ОДУ

Контрольные вопросы

4. Каким уравнением описывается изменение температуры в бассейне?
5. Какая функция MATLAB предназначена для решения ОДУ?
6. Как задается функция правых частей ОДУ?
7. Как задать интервал интегрирования?
8. Как задать начальные условия?

Порядок выполнения работы

1. Выполнить следующие упражнения: найти решение ОДУ на интервале $[t_0; t_f]$ с заданными начальными условиями $y(t_0) = y_0$.

1) $y' = t^3 \cos(y/\sqrt{5})$, $t_0=0$; $t_f=1$; $y_0=3$;

2) $y' = \dfrac{y}{1-t^2}$, $t_0=0$; $t_f=0,5$; $y_0=1$;

3) $y' = \ln t \cos(y/3)$, $t_0=1$; $t_f=2$; $y_0=0$;

4) $y' = \dfrac{ty}{\sqrt{t^2-4}}$, $t_0=2,5$; $t_f=5$; $y_0=1$;

5) $y' = y\sqrt{\dfrac{1+t}{1-t}}$, $t_0= -1$; $t_f=0,99$; $y_0=1$;

6) $y' = \dfrac{t}{2y} \cdot \dfrac{2-t}{(1-t)^2}$, t_0= 2; t_f=3,615; y_0=1;

7) $y' = \dfrac{y^2 \ln t}{t}$, t_0=0,01; t_f=3; y_0=1;

8) $y' = \dfrac{\operatorname{ctg} t}{y^2}$, t_0=0,04; t_f=3,1; y_0=1.

Постановка задачи

Бассейн был наполнен холодной водой. Чтобы повысить температуру в бассейне, открыли кран с горячей водой. Зависимость температуры воды в бассейне T от температуры горячей воды, поступающей в ванну $T_{вх}$, объема ванны V и объемной скорости потока горячей воды $Q = dV/dt$ описывается дифференциальным уравнением (предполагается идеальное смешивание, потери тепла не учитываются.):

$$\frac{dT}{dt} = \frac{Q}{V}\left(T_{вх} - T\right)$$

Температура воды в бассейне в начальный момент времени T_0=10,2°C. Температура горячей воды $T_{вх}$=54,4°C. Бассейн наполняется со скоростью 30 л/мин и в него вмещается 3000 л воды.

Необходимо найти зависимость температуры воды в бассейне от времени. Найти время, в течение которого температура воды поднимется до 26°C.

Методика расчета

1. Создать m-файл для вычисления правой части ОДУ. Сохранить его под именем `temperature`. Необходимые параметры будут передаваться через глобальные переменные. Поэтому их необходимо описать в программе-функции и в командном окне.

```
function dT=temperature(t,T);
global V Q Tinp;
dT=Q/V*(Tinp-T);
```

2. В командном окне ввести функцию `ode45` с указанием интервала интегрирования (продолжительности времени, в течение которого будет наблюдаться процесс) и начальных условий (начального значения температуры воды).

интервал интегрирования (60 мин) – `[0 60]`
начальное условие – T_0=10,2°С

```
global V Q Tinp
V=3000; Q=30; Tinp=54.4;
[t,T]=ode45(@temperature,[0 60],10.2);
```

3. Построить график зависимости температуры воды от времени На график нанести координатную сетку, подписать оси.

```
plot(t,T)
grid on
xlabel('time t'); ylabel('Temperature T');
```

По графику визуально определить время, в течение которого температура воды поднимется до 26°С.

В лабораторной работе №15 будет показана возможность аналитического решения дифференциальных уравнений.

Лабораторная работа № 13. Решение ОДУ второго порядка. Расчет колебательной системы

Цель занятия: получить навыки численного решения задач, описываемых обыкновенными дифференциальными уравнениями второго порядка.

Технологии решения ОДУ высших порядков в MATLAB

Дифференциальные уравнения высших порядков $y^{(n)} = f(x, y, y', y'', ..., y^{(n-1)})$ решаются сведением к системе уравнений первого порядка путем замены переменных:

$$y_1 = y',\ y_2 = y'' \text{ и т.д.}$$

При этом дифференциальное уравнение n-го порядка заменяется системой из n уравнений:

$$\begin{aligned} &y' = y_1, \\ &y_1' = y_2 \\ &....... \\ &y_{n-1}' = f(x, y, y_1, y_2, ..., y_{n-1}) \end{aligned}$$

Для интегрирования системы уравнений в MATLAB создается m-файл со столбцом правых частей уравнений и затем применяется функция `ode45` с указанием интервала интегрирования и вектора начальных условий.

Постановка задачи

Колебательная система (рис. 13.1), состоит из груза, упругого элемента (пружины) и гасителя колебаний (демпфера). Груз подвергается действию силы. Уравнение движения имеет вид:

$$m\frac{d^2x}{dt^2} = -kx - \alpha\frac{dx}{dt} + f_0\sin(\omega_f t)$$

где m – масса тела, x – координата тела, dx/dt – скорость, с которой оно движется, d^2x/dt^2 – ускорение, k – коэффициент жесткости пружины, α - коэффициент демпфирования, ω_f – частота вынуждающих колебаний, а f_0 – амплитуда вынуждающей силы.

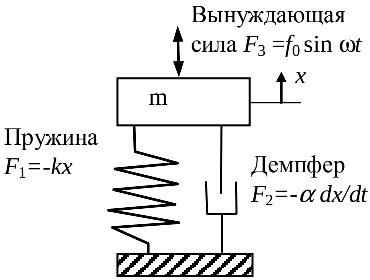


Рис. 13.1. Колебательная система

Построить график зависимости смещения и скорости от времени.

Исследовать поведение колебательной системы при различных значениях параметров. Рассмотреть случаи, когда отсутствует

- вынуждающая сила;
- демпфирование.

Технологии решения задачи

Для того чтобы воспользоваться решателем `ode45`, необходимо исходное уравнение второго порядка представить в виде системы двух уравнений первого порядка.

1. Исходное уравнение может быть переписано в виде

$$\frac{d^2x}{dt^2} = -\frac{k}{m}x - \frac{\alpha}{m}\frac{dx}{dt} + \frac{f_0}{m}\sin\omega_f t$$

или, введя обозначение $\beta = \alpha / m$, $\omega^2 = k / m$

$$\frac{d^2x}{dt^2} = -\omega^2 x - \beta\frac{dx}{dt} + a_f \sin\omega_f t$$

2. Ввести новые переменные:

$$y_1 = x, y_2 = dx / dt \quad ,$$

3. Записать исходное уравнение в виде системы двух уравнений

$$\begin{cases} \dfrac{dy_1}{dt} = y_2 \\ \dfrac{dy_2}{dt} = -\omega^2 y_1 - \beta y_2 + a_f \sin \omega_f t \end{cases}.$$

4. Создать m-файл функцию `oscil`, описывающую правые части этой системы уравнений. Параметры передаются через глобальные переменные.

```
function dydt = oscil(t,y)
global w2 wf af b;
dydt=[y(2);-w2*y(1)-b*y(2)+af*sin(wf*t)];
```

4. Найти решение с помощью функции `ode45` и построить график.

```
global w2 wf af b; w2=4; wf=3.5; af=4;b=0.25;
[t,y]=ode45(@oscil,[0 30],[2;0]);
plot(t,y(:, 1),'-',t,y(:, 2),'--')
title('Solution of oscillations equation');
xlabel('time t');
ylabel('solution x');
legend('coordinate','velocity')
```

Контрольные вопросы

1. Каким образом можно свести ОДУ высших порядков к системе уравнений первого порядка?
2. Как задается функция правых частей ОДУ?
3. Как задать интервал интегрирования?
4. Как задать начальные условия?
5. Каким уравнением описывается колебательная система?

Лабораторная работа № 14. Решение задач в частных производных. Распределение температуры

Цель занятия: научиться решать задачи, сводящиеся к уравнениям в частных производных, с использованием пакета PDE.

Основные положения

Для решения дифференциальных уравнений в частных производных предназначен пакет PDE (Partial Differential Equations Toolbox). Команды и графический интерфейс этого пакета используются для математического моделирования уравнений в частных производных применительно к широкому классу инженерных и научных приложений, включая задачи сопротивления материалов, расчеты электромагнитных устройств, задачи тепломассопереноса и диффузии. В пакете реализован метод конечных элементов. Пакет позволяет пойти все шаги решения задачи в визуальном режиме.

Постановка задачи

Найти распределение температуры в области заданной формы (рис. 14.1). Процесс стационарный

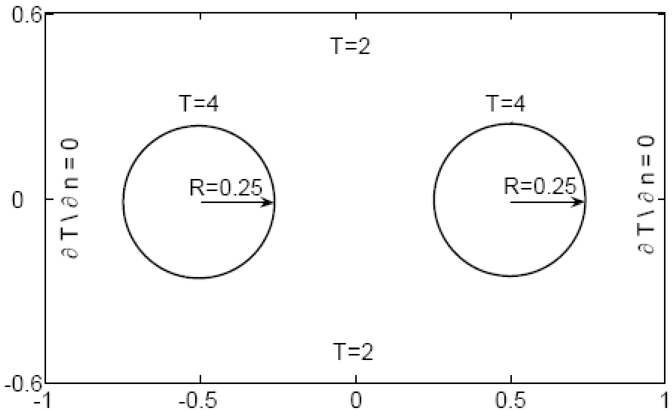


Рис. 14.1. Область и граничные условия

Левая и правая границы области теплоизолированы, т.е. тепловой поток на них равен нулю, на верхней и нижней границах температура равна 2, а на внутренних границах круговых вырезов температура равна 4.

Внутри области нет источников тепла. Стационарный процесс распространения тепла описывается формулой:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$$

Технологии решения

1. Запустить пакет PDETool командой `pdetool` из командного окна системы MATLAB или из нижнего меню Рабочего стола **Start / Toolboxes / Partial Differential Equations / PDETool GUI** (рис. 14.2).

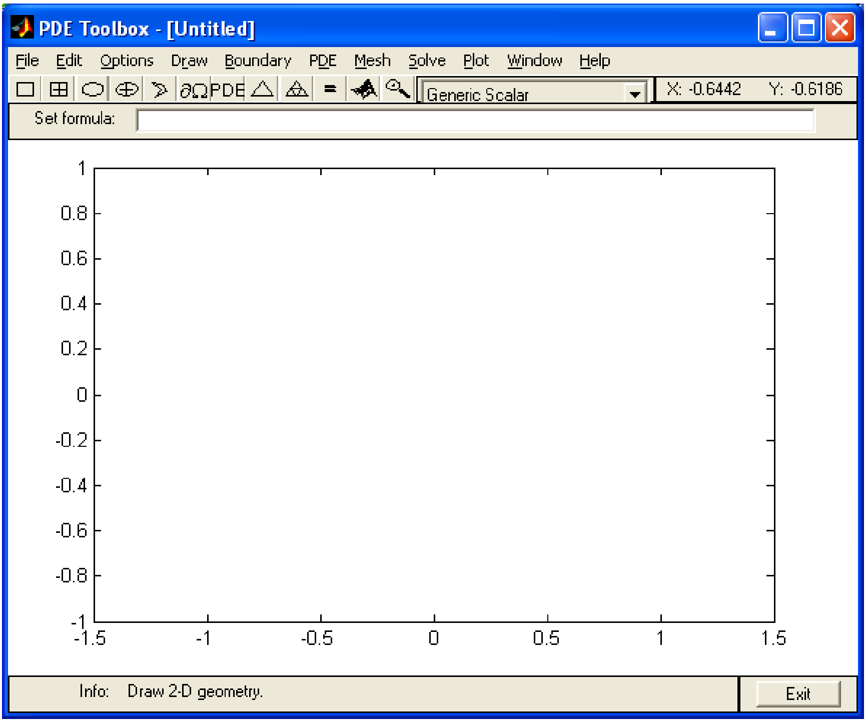


Рис. 14.2. Среда PDETool

2. В появившемся графическом окне (рис. 14.2) определить геометрию области решения задачи, используя кнопки на панели инструментов. Первые две () позволяют создать прямоугольник (первая от одного угла к другому по диагонали, а вторая – от центра к углу), а третья и четвертая () – эллипс. С помощью пятой () можно сконструировать область, имеющую форму многоугольника.

Для конструирования прямоугольника размером 2×1,2 воспользуемся первой кнопкой, задав с помощью мыши приблизительный размер. Затем двойным щелчком мыши по заданной области можно перейти в окно **Object Dialog** и установить точные размеры.

Аналогично с помощью мыши, предварительно нажав кнопку , следует нарисовать фигуру, близкую к кругу с центром вблизи точки пересечения диагоналей левой половины прямоугольника, а затем, после двойного щелчка, в окне диалога **Object Dialog** установить точные параметры круга: координаты центра и длины полуосей. Аналогично создается круг с центром вблизи точки пересечения диагоналей правой половины прямоугольника.

Когда геометрические примитивы для задания области созданы, следует задать комбинацию, определяющую расчетную область: круги должен быть удалены из прямоугольника. Связь между примитивами задается в строке **Set Formula**. Знак плюс означает объединение объектов, а минус – вычитание.

Расчетной области, изображенной на рис. 14.1, соответствует формула **R1–E1–E2**.

3. Определение уравнения.

Из раскрывающегося списка выбрать тип решаемой задачи. Пункт **Heat Transfer** соответствует задаче о распределении тепла.

Далее следует задать коэффициенты уравнения. Для этого нужно перейти в режим дифференциального уравнения, нажав кнопку 

. Выбор эллиптического типа уравнения соответствует стационарному, а параболического – нестационарному уравнению теплопроводности. Задание значений коэффициентов в окне диалога позволяет определить уравнение, соответствующее поставленной задаче. В нашем случае это значения Q = 0, h = 0, k = 1.

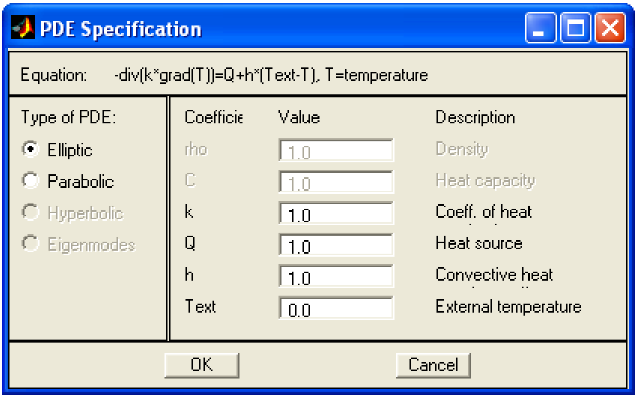


Рис. 14.3. Окно для установки коэффициентов уравнения

4. Задание граничных условий. Нажав кнопку «Boundary Conditions» , перейти в режим установки граничных условий. В окне отображаются только границы области. При этом прямоугольник имеет четыре границы, по числу сторон, а окружность также составлена из четырех дуг. Щелчком мыши выделяем текущую границу (например, верхнюю границу прямоугольника) и выбираем в меню **Boundary** пункт **Specify Boundary Conditions...**, или нажимаем кнопку  Появляется диалоговое окно **Boundary Condition**, предназначенное для выбора типа граничного условия и коэффициентов (рис. 14.4).

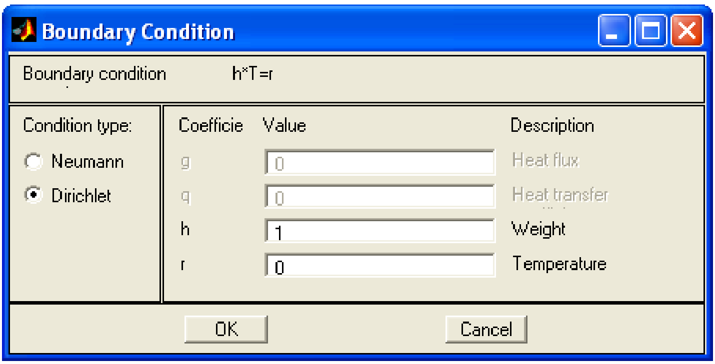


Рис. 14.3. Окно для установки граничных условий

На верхней границе прямоугольника задано значение температуры, равное 2 (см. рис. 14.1). Это граничное условие Дирихле. Установим при помощи строк ввода значение **h**, равное единице, а **r** приравняем 2. Сохраним значения, нажав ОК, и проделаем аналогичную операцию для нижней стороны прямоугольника.

Можно определить одинаковые граничные условия сразу для нескольких частей границы. Добавление части границы в группу производится щелчком мыши с одновременным удержанием клавиши **Shift**. Двойной щелчок мышью по части границы или по группе частей дает быстрый доступ к диалоговому окну **Boundary Condition**. Установим температуру 4 на границах отверстий, сгруппировав предварительно четыре части каждой окружности.

На правой и левой стороне прямоугольной области поток тепла равен нулю, поскольку границы теплоизолированы. Для них выберем условие Неймана **n*k*grad(T) + q*T = g**. Для получения теплоизолированных границ требуется задать коэффициенты **q** и **g** равными нулю.

В режиме установки граничных условий границы с условиями Дирихле отображаются красным цветом, а синий цвет выделяет границы, на которых задано условие Неймана.

5. Триангуляция области. После нажатия кнопки  область покрывается сеткой, состоящей из треугольников. Для уменьшения размеров сетки следует нажать на кнопку Refine Mesh **.**

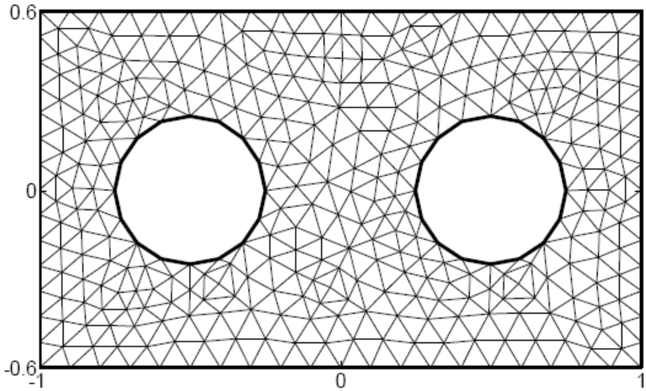


Рис. 14.4. Расчетная триангуляция

6. Решение. Решение задачи производится выбором пункта **Solve PDE** меню **Solve** или нажатием на кнопку . Найденное распределение температуры отображается в окне PDE Toolbox**.**

7. Параметры графика устанавливаются в окне Plot Selection.

8. Изменение геометрии области, граничных условий, типа уравнения и его коэффициентов может быть выполнено, даже если решение уже найдено. Следует перевести среду PDETool в соответствующий режим и произвести требуемые действия.

Изменить значение температуры на внешних стенках; на внутренних границах круговых вырезов.

9. Сохранение работы производится в m-файле из пункта **Save** меню **File.**

Контрольные вопросы

1. Для решения каких задач предназначен модуль PDE?
2. Как задать геометрию задачи?
3. Как задать граничные условия?
4. Как задать коэффициенты уравнения?
5. Как построить сетку на области
6. Как задать начальные условия?
7. Как запустить задачу на выполнение?

Лабораторная работа № 15. Решение задач в символьном виде

Цель занятия: научиться решать задачи в символьном виде.

Основные положения

Пакет прикладных программ **Symbolic Math Toolbox** предоставляет возможность решения задач в символьном (аналитическом) виде. Пакет обеспечивает выполнение символьного дифференцирования и интегрирования, вычисление сумм и произведений, разложение в ряды Тейлора, операции с полиномами, решение в аналитическом виде нелинейных уравнений, всевозможные символьные преобразования.

Для проведения символьных операций необходимо, чтобы переменные были предварительно объявлены.

Пример:

```
>>x=sym('x'); y=sym('y');
```

или

```
>>syms x y
```

Функция `simplify` осуществляет упрощение символьных выражений:

```
>>simplify((x^2-2*x*y+y^2)/(x-y))
ans=
   x-y
```

Функция `expand` раскрывает алгебраические и функциональные выражения:

```
>>expand(sin(x+y))
ans=sin(x)*cos(y)+cos(x)*sin(y)
```

Функция `diff(y,x)` вычисляет производные от *y* по *x*:

```
>>y=x^2-4*x-7;
>>diff(y,x)
ans=
2*x-4
```

Функция `int` осуществляет интегрирование функций:

```
>>syms x y k
>>y=x^k;
>>int(y,x)
ans=
x^(k+1)/(k+1)
```

Для аналитического решения алгебраических уравнений предназначена функция `solve`. По умолчанию предполагается, что выражение, являющееся аргументом функции, приравнивается нулю.

```
>>syms x a b c; solve(a*x^2+b*x+c)
ans=
 1/2/a*(-b+(b^2-4*a*c)^(1/2))
 1/2/a*(-b-(b^2-4*a*c)^(1/2))
```

Решение системы нелинейных уравнений также находится при помощи `solve`. Входными аргументами `solve` являются в данном случае левые части уравнений и переменные, по которым требуется разрешить систему. Например, для системы из двух уравнений с правыми частями, которые определены в символьных функциях `f1` и `f2`, зависящих от `x` и `y`, вызов `solve` может выглядеть так:

```
>>[x,y]=solve(fl,f2,x,y);
```

Входными аргументами `solve` могут быть строки с уравнениями, заключенные в апострофы, причем не обязательно переносить все слагаемые в левую часть. Независимые переменные можно не указывать:

```
>>[x,y]=solve('x*(2-y)=cos(x)*exp(y)','2+x-
y=cos(x)+exp(y)')
```

Контрольные вопросы

1. Как в системе MATLAB описывают символьные переменные?
2. Какая функция позволяет упростить выражение?
3. Как найти производную в символьном виде?
4. Как вычислить интеграл?
5. Какая функция предназначена для аналитического решения алгебраических уравнений?

Порядок выполнения работы

1. Упростить выражение (по вариантам):

1) $\dfrac{1+\cos 4a}{\operatorname{ctg} a - \operatorname{tg} a}$;

2) $\dfrac{\operatorname{tg} a - \cos^{-1} a}{\cos a - \operatorname{ctg} a}$;

3) $\dfrac{x+1}{x^2+x+1} : \dfrac{1}{x^3-1}$;

4) $\dfrac{a^2-b^2}{a-b} - \dfrac{a^3-b^3}{a^2-b^2}$;

5) $\dfrac{4ab}{b^2-a^2}\left(\dfrac{1}{b^2-a^2}+\dfrac{1}{a^2+2ab+b^2}\right)$;

6) $\dfrac{4x}{3y^2+2xy} - \dfrac{9y}{3xy+2x^2}$;

7) $\dfrac{x}{2a^2-ax} - \dfrac{4a}{2ax-x^2}$;

8) $\sqrt{\left(5-\sqrt{3}\right)^2}+\sqrt{\left(1-\sqrt{3}\right)^2}$.

2. Найти производную:

1) $y=\dfrac{2x}{1-x^2}$;

2) $y=\dfrac{x}{\sqrt{a^2-x^2}}$;

3) $y=\cos 2x - 2\sin x$;

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

5) $y=\dfrac{\sin^2 x}{\sin x^2}$;

6) $y=e^x\left(1+\operatorname{ctg}\dfrac{x}{2}\right)$;

7) $y=2^{\operatorname{tg} x}$;

8) $y=\ln^3 x^2$.

3. Найти интеграл:

1) $\displaystyle\int \frac{dx}{2+3x^2}$;

2) $\displaystyle\int \frac{dx}{2-3x^2}$;

3) $\displaystyle\int \frac{dx}{\sqrt{2-3x^2}}$;

4) $\displaystyle\int \sqrt{1-\sin^2 x}dx$;

5) $\displaystyle\int \frac{dx}{1+\cos x}$;

6) $\displaystyle\int \frac{1}{x^2}\sin\frac{1}{x}dx$;

7) $\displaystyle\int \sin^2 xdx$;

8) $\displaystyle\int x \operatorname{arctg} x\, dx$.

4. Решить систему уравнений, используя методы символьной математики:

1) $\sin(x+1) - y = 1.2$
   $2x + \cos(y) = 2$

2) $\mathrm{tg}(xy + 0.4) = x^2$
   $0.6x^2 + 2y^2 = 1$

3) $\cos(x - 1) + y = 0.5$
   $x - \cos x = 3$

4) $\sin x + 2y = 2$
   $\cos(y - 1) + x = 0.7$

5) $x^4 + y^4 = 5$
   $y = e^{-x}$

6) $x^2 + y^2 = 1$
   $x - y^3 = 0.5$

7) $x^2 + y^2 = 1$
   $\sin(x + y) = 1.2x$

8) $x + \cos(y - 1) = 0.8$
   $y - \cos x = 2.$

Постановка задачи

См. постановку задачи об изменении температуры воды в бассейне в лабораторной работе № 12

Математическая постановка задачи

$$\frac{dT}{dt} = \frac{Q}{V}\left(T_{вх} - T\right), \quad \text{начальное условие} \quad T(0) = T_0$$

Технологии решения

В MATLAB для решения дифференциальных уравнений используется функция `dsolve`. По умолчанию независимая переменная обозначается буквой `t`. Для обозначения первой производной используется символ `D`, второй – `D2`, третьей –`D3` и т.д.

```
>> syms T t Q V Tinp T0
>> T=dsolve('DT=Q/V*(Tinp-T)','T(0)=T0')
T =
Tinp+exp(-Q/V*t)*(-Tinp+T0)
```

Подстановка в аналитическое решение численных значений.

```
>> V=3000; Q=30; Tinp=54.4;T0=10.2;
>> subs(T)
ans =
272/5-221/5*exp(-1/100*t)
```

```
>> t=50;
>> subs(T)
ans =
   27.5913
```

ПРИЛОЖЕНИЕ 1. Оформление отчета

1. Титульный лист.

Белорусский национальный технический университет
Кафедра ЮНЕСКО «Энергосбережение
и возобновляемые источники энергии»

ОТЧЕТ
по лабораторным работам
по дисциплине «Компьютерные технологии решения инженерных задач»

Выполнил: студ. гр. № …
ФИО
Проверил
преподаватель

Минск 20__

2. Название работы.
3. Цель работы.
4. Ответы на контрольные вопросы.
5. Ход выполнения работы:

Математическая постановка задачи.
Команды MATLAB с пояснениями.

ПРИЛОЖЕНИЕ 2. Краткая справка-указатель MATLAB

Справочная система MATLAB

`help`	список разделов справки;
`help раздел`	список команд раздела;
`help имя_команды`	описание команды с указанным именем;
`help ops`	операторы и специальные символы;
`help elfun`	элементарные математические функции;
`help demos`	список примеров;
`type имя_m-файла`	просмотр текста m.файла с примером.

Константы, переменные

`0.9093`	дробное число
`1.6021e-20`	число $1{,}6021\cdot10^{-20}$ в экспоненциальной форме
`3i` или `3j`	мнимое число
`2+3i`	комплексное число
`pi`	число π=3,14159265...
`'текст'`	символьная константа
`%текст`	текстовый комментарий

Имена переменных состоят из прописных и строчных букв английского алфавита, цифр и символа подчеркивания. Первым символом в имени должна быть буква.

`abc=3`	присвоение переменной значения
`Abc=5`	`Abc` и `abc` – это разные переменные.

Вектора и матрицы

По умолчанию все числовые переменные в MATLAB считаются матрицами. Скалярная величина есть матрица порядка `1×1`. Вектором является одномерный массив размера `1×n` (вектор-строка) или `m×1` (вектор-столбец).

`v=[2 0 1]` или `v=[2,0,1]` ввод вектора-строки.

`u=[3;1;4]` ввод вектора-столбца.

`M=[1 2;3 8]` ввод матрицы $M = \begin{pmatrix} 1 & 2 \\ 3 & 8 \end{pmatrix}$

`M(1,2)` элемент матрицы M из 1–й строки 2-го столбца
`r=M(:,1)` вектор равен 1-му столбцу матрицы M
`M(2,:)` 2-я строка матрицы M
`M(:,2)=[]` удаление 2-й столбца матрицы M
`A=[M r]` объединение матриц в одну матрицу.
`x=0:0.2:6` формирование вектора как диапазона чисел от 0 до 6 с шагом 0.2
`t=1:10` формирование вектора с элементами от 1 до 10 с шагом 1
`z=linspace(1,10,5)` формирование вектора из 5 элементов, равномерно распределенных на отрезке от 1 до 10
`E=eye(3)` задание единичной матрицы размера 3x3
`N=ones(3,4)` задание матрицы из единиц размера 3x4
`Z=zeros(3)` задание нулевой матрицы размера 3x3
`size(N)` размер матрицы в виде вектора-строки [3 4]
`zeros(size(M))` нулевая матрица такого же размера как M
`length(z)` длина вектора `z`

Функции

Основные встроенные математические функции:

`abs(x)` абсолютная величина *x*;
`sqrt(x)` корень квадратный из *x*.
`exp(x)` экспонента e^x;
`log(x)` натуральный логарифм ln *x*;
`sin(x), cos(x)` синус *x*, косинус *x*.

Имя функции записывается строчными буквами, аргументы указываются в круглых скобках через запятую. В тригонометрических функциях используется угол в радианах.

Арифметические операции

В MATLAB реализовано два типа арифметических операций: в соответствии с правилами линейной алгебры и поэлементные операции. Оператор поэлементной операции записывается с точкой.

`v*u` Матричное умножение. Число столбцов первого сомножителя должно быть равно числу строк второго.

`A.*E` Поэлементное умножение двух массивов одинакового размера

`A^2` Степень матрицы. Если показатель степени целое положительное число, то матрица перемножается сама на себя

`A.^2` Возведение элементов матрицы в степень

`\` Левое деление. Решение систем линейных уравнений AX=B. X=A\B

`./` Поэлементное деление

Матричные вычисления

`det(A)` определитель квадратной матрицы `A`.

`inv(A)` матрица, обратная квадратной матрице `A`.

`sum(A)` возвращает вектор-строку, содержащую сумму элементов каждого столбца.

`prod` произведение элементов массива

`max` определение максимальных элементов массива

`min` определение минимальных элементов массива

`mean` определение средних элементов массива

`sort` сортировка элементов массива по возрастанию

Полиномы в MATLAB

`p=[1 0 -2 -5]` представление полинома $p(x) = x^3 - 2x - 5$ с помощью вектора-строки, содержащего упорядоченные по убыванию степени коэффициенты полинома

`polyval(p,x)` вычисляет значения многочлена `p` в точках `x`.

Построение графиков

`plot(x,y)` создает кусочно–линейный график зависимости компонент вектора y от x.

`plot(x1,y1,x2,y2, …)` строит несколько кривых.

`polar(phi,r)` строит график в полярной системе координат.

`fplot('fun',[a b])` график функции, заданной в символьном виде, в интервале изменения аргумента от `a` до `b`.

Трехмерная графика

`plot3(x,y,z)` трехмерный аналог команды `plot`

`[X,Y]=meshgrid(x,y)` формирует прямоугольную сетку, заданную векторами `x` и `y`

```
>> [X,Y]=meshgrid([-3:0.15:3]);
>>Z=X.^2+Y.^2;
```

`mesh(X,Y,Z)` строит каркасную поверхность Z(X,Y)

`surf(X,Y,Z)` строит сплошную цветную поверхность

`contour(X,Y,Z)` строит контурный график

Управление выводом графика

`title('TEXT')` – вывод заголовка TEXT

`xlabel('TEXT')`, `ylabel('TEXT')` – маркировка осей x, y

`legend('TEXT1','TEXT2',..)` – идентификация кривых

`grid on` – нанесение координатной сетки

`hold on` –продолжение вывода графиков в текущее окно

`figure` – открывается новое графическое окно

`subplot(m,n,p)` разбивает графическое окно на `mxn` подокон, , `p`- номер текущего подокна, нумерация идет по строкам.

Программирование в среде MATLAB

Программы на языке программирования MATLAB сохраняются в виде текстовых файлов, имеющих расширение .m (*m-файлов*).

`%Комментарий` ` операторы`	m-файл сценарий. Работают с данными из рабочей области

`function [v1,v2,…]=fname(p1,p2..)`
`%Комментарий`
`операторы`
`var1=…`
`var2=…`
	m-файл функция `fname` содержит входные `(p1,p2,…)`и выходные параметры `(v1,v2,…)` и использует локальные переменные, доступные только в пределах данной функции.

`global X Y Z` `g=inline('выражение')`	глобальные переменные операторная функция

Управляющие конструкции языка MATLAB

```
for V=A:H:B
команды
end
```
циклы с определенным числом повторений

```
while условие
    команды
end
```
цикл с неопределенным числом повторений

```
if условие
   команды
else
   команды
end
```
условный оператор

Численный анализ в MATLAB

`quad('x.^3',0,1)` вычисляет значение определенного интеграла для функции $f(x) = x^3$ на отрезке [0, 1].

`trapz(x,y)` вычисляет интеграл по значениям точек (x, y).

`fminbnd('fun',a,b)` минимум функции на интервале [a, b].

`fzero('fun',x)` находит нуль действительной функции, `x` – начальное приближение или интервал поиска.

Аппроксимация функции полиномом

`p=polyfit(x,y,N)` `x,y` –данные, `N` – порядок полинома

Решение дифференциальных уравнений

`[t,y]=ode45('F',[t0 tf],y0)` интегрирует систему ОДУ $y'=F(t,y)$ на интервале `[t0 tf]` с начальными условиями $y(t_0)$=`y0`. Функция вычисления правых F должна быть заранее подготовлена в m-файле `F`

Решение уравнений частных производных

Модуль PDE

Символьные вычисления

`syms x y` объявление переменных символьными

ПРИЛОЖЕНИЕ 3. Типичные ошибки

1. Ошибки при записи дробных чисел.

Правильно	*Неправильно*
`1.458`	`1,458`

2. Ошибочное использование скобок при записи матриц.

Правильно	*Неправильно*
`a=[0,2,8]`	`a=(0,2,8)`
`a(1)`	`a[1]`

3. Ошибки при записи выражений.

3.1. Забывают ставить знак умножения.

Правильно	*Неправильно*
`2*sin(x)`	`2sin(x)`

3.2. Непарность скобок.

Правильно	*Неправильно*
`sqrt(sin(x))`	`sqrt(sin(x)`

3.3. Невнимание к порядку старшинства арифметических действий. Ошибки при записи дробных выражений.

Математическое выражение $\frac{1+x}{2a}$

Правильно	*Неправильно*
`(1+x)/(2*a)`	`1+x/(2*a)`
`(1+x)/2/a`	`(1+x)/2*a`

4. Ошибки при записи функций.

4.1. Неправильный синтаксис (без скобок, запись экспоненты).

Правильно	*Неправильно*
`sin(x)`	`sin x`
`exp(x)`	`exp^(x)`
`sin(x)^2`	`sin^2(x)`

4.2. Неправильное имя функции (математическое обозначение функции и запись в MATLAB могут отличаться; орфографические ошибки).

Правильно	*Неправильно*
`log(x), tan(x)`	`ln(x), tg(x)`
`abs(x), sqrt(x)`	`abc(x), sgrt(x)`

4.3. Использование прописных букв в командах MATLAB.

Правильно	*Неправильно*
`sin(x)`	`Sin(x), SIN(x)`

5. Неправильные имена переменных и m-файлов.

5.1. Имя начинается с цифры.

Правильно	*Неправильно*
`p1`	`1p`
`p1.m`	`1p.m; 123.m`

5.2. Имена переменных и m-файлов совпадают с именами встроенных функций MATLAB.

```
>> max=1;  a=[2 3;4 5];
>> max(a)
??? Index exceeds matrix dimensions.
```

После удаления переменной `max`

```
>> max(a)
ans =
     4   5
```

Студенту Максиму не следует называть свой файл `max.m`.

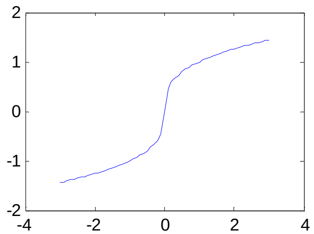
5.3. Иногда студенты вообще забывают указать имя функции.

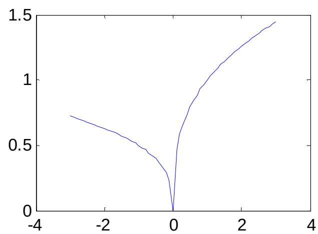
Правильно	*Неправильно*
`s= quad('sin(x+2)',a,b)`	`s=('sin(x+2)',a,b)`

6. Незнание особенностей вычисления некоторых функций в MATLAB.

Вычисление кубического корня из отрицательного числа

Правильно	*Неправильно*
`x=-3:0.1:3;`	
`plot(x,sign(x).*abs(x).^(1/3))`	`plot(x,x.^(1/3))`





7. Ошибки в алгоритмах

7.1. Невыполнение очистки переменных. Неприсвоение переменным начальных значений.

Для накопления суммы элементов перед циклом переменной следует присвоить значение, равное нулю, произведения – 1.

Для вычисления факториала `n`*! пользователь написал программу, предполагая, что будет присвоено начальное значение* `f=1` *вне программы при выборе* `n`*.*

```
for k=1:n
f=f*k; end
```

Если не выполнить присвоения начального значения `f` *во второй раз перед выполнением этой программы, то вычисления будут ошибочны.*

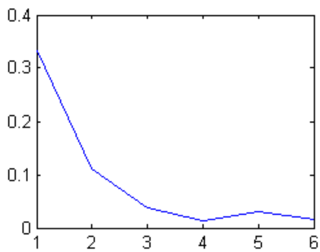
7.2. Невыполнение очистки переменных. Ошибки при формировании матриц и векторов.

При неоднократном поэлементном формировании вектора с помощью программ возможна ситуация, когда новый вектор меньшего размера, чем предыдущий, в «хвосте» остаются элементы предыдущего вектора. Это может привести к ошибкам при выполнении последующих операций с вектором.

```
for k=1:6
    v(k)=1/(2^k);
end
```

Затем программа пересчитывается после внесения изменений.

```
for k=1:4
    v(k)=1/(3^k);
end
plot(v)
```



Строится ошибочный график

7.3. Ошибки в начале и конце цикла.

Необходимо внимательно выбирать начальную и конечную переменную цикла, например, начинать цикл с нуля или 1, заканчивать переменной `n` или `n-1`.

8. Иногда студенты переписывают промежуточные результаты вычислений и вставляют числа в формулы вместо того, чтобы присвоить эти значения переменным и затем использовать их в расчетах.

Правильно	*Плохо*
`a= fzero('sin(x+2)',1);`	`fzero('sin(x+2)',1)`
`quad('sin(x+2)',0,a)`	`ans=1.1416`
	`quad('sin(x+2)',0,1.1416)`

9. Рекомендуется подавлять вывод на экран больших массивов, результатов промежуточных вычислений и вычислений в циклах. Вывод промежуточных результатов полезен на этапе отладки программы.

Литература

1. Ануфриев, И.Е. MATLAB 7/ И.Е. Ануфриев, А.Б. Смирнов, Е.Н. Смирнова. – СПб.: БХВ-Петербург, 2005 – 1104 с.

2. Бондаренко, В.Ф. Matlab. Основы работы и программирования, компьютерная математика. Учебный курс / В.Ф. Бондаренко, В.Д.Дубовец. – Минск: Харвест, 2010. – 256 с.

3. Дьяконов, В.П. MATLAB 6.5 SP1/7 + Simulink 5/6(r). Основы применения/ В.П. Дьяконов. – М.: СОЛОН-Пресс, 2005. – 800 с.

4. Кетков, Ю.Л., MATLAB 7: программирование, численные методы / Ю.Л. Кетков, А.Ю. Кетков, М.М. Шульц. – СПб.: БХВ-Петербург, 2005. – 752 с.

5. Краков, М. MATLab для инженеров: современные технологии расчетов / М.Краков. И.Никифоров. - Palmarium Academic Publishing. Saarbrucken, Germany, 2012. - 152 p.

6. Половко, А.М. MATLAB для студента/ А.М. Половко, П.Н. Бутусов. – СПБ.: БХВ-Петербург, 2005. – 320 с.

7. Сидорик, В.В. Математическое моделирование в среде Matlab: учебно-методическое пособие / В.В. Сидорик, С.Г. Погирницкая. – Минск: БНТУ, 2008. – 112 с.

8. Современные технологии решения инженерных задач: Практикум для студентов специальностей 1-43 01 06 «Энергоэффективные технологии и энергетический менеджмент» и 1-36 20 02 «Упаковочное производство» /Сост. Краков М.С., Погирницкая С.Г. . – Минск: БНТУ, 2007. – 76 с.

9. Hunt, Brian R. Matlab: официальный учеб.курс Кембриджского университета: [пер. с англ.] / Brian R. Hunt [и др.]. / - М.: Изд-во ТРИУМФ, 2008. – 352 с.