

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Санкт-Петербургский государственный университет
промышленных технологий и дизайна»

СИСТЕМЫ КОМПЬЮТЕРНОЙ МАТЕМАТИКИ. Приемы работы в среде MATLAB

Утверждено редакционно-издательским советом университета
в качестве учебного пособия для обучающихся направления
15.03.02 – Технологические машины и оборудование
всех форм обучения

Санкт-Петербург
2019

УДК 004.421:681.3 (075.8)

ББК 32.973я73

С40

Р е ц е н з е н т ы :

профессор кафедры автоматизации производственных процессов
Санкт-Петербургского государственного университета промышленных
технологий и дизайна *А. Б. Кикин*;
генеральный директор ООО «Радел» *Р. А. Мамлеев*

С40 Системы компьютерной математики: приемы работы в среде
MATLAB: учеб. пособие / И. М. Беспалова, К. И. Мартынчик, А. В. Мар-
ковец, А. Г. Усов. – СПб.: ФГБОУВО «СПбГУПТД», 2019. – 107 с.

ISBN 978-5-7937-1757-1

Учебное пособие предназначено для оказания помощи студентам при выполнении лабораторных работ по курсу «Системы компьютерной математики». Рассмотрены особенности практической работы в MATLAB – интерактивной компьютерной системе для выполнения инженерных и научных расчетов, ориентированной на работу с массивами данных. Приводятся задания для самостоятельного выполнения и вопросы для контроля полученных знаний.

УДК 004.421:681.3 (075.8)

ББК 32.973я73

ISBN 978-5-7937-1757-1

© ФГБОУВО «СПбГУПТД», 2019

© И. М. Беспалова, 2019

© К. И. Мартынчик, 2019

© А. В. Марковец, 2019

© А. Г. Усов, 2019

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	4
ОБЩИЕ СВЕДЕНИЯ О СИСТЕМАХ КОМПЬЮТЕРНОЙ МАТЕМАТИКИ	5
ЧАСТЬ 1. ВВЕДЕНИЕ В MATLAB	7
1.1. Рабочая среда системы MATLAB	7
1.2. Массивы в среде MATLAB	23
1.3. Построение графиков в среде MATLAB (краткие сведения).....	39
1.4. Построение и оформление графиков функций двух переменных.....	49
1.5. КОНТРОЛЬНЫЕ ЗАДАНИЯ.....	57
ЧАСТЬ 2. ПРОГРАММИРОВАНИЕ В MATLAB	67
2.1. Основные понятия программирования	67
2.2. Основы подготовки, редактирования и отладки М-файлов	69
2.3. Основы программирования в MATLAB.....	75
2.4. Вычисление функции по заданному имени	81
2.5. Использование подфункций и вложенных функций.....	82
2.6. Локальные и глобальные переменные	87
2.7. Операторы языка программирования	89
2.8. Условный оператор.....	94
2.9. Оператор альтернативного выбора	98
2.10. Операторы цикла.....	99
2.11. Создание паузы в вычислениях	104
2.12. КОНТРОЛЬНЫЕ ЗАДАНИЯ.....	105

ВВЕДЕНИЕ

Целью дисциплины «Системы компьютерной математики» является формирование компетенций в области применения современных ЭВМ для проведения математических вычислений с использованием современного программного обеспечения.

При изучении данной дисциплины ставятся следующие задачи:

- рассмотреть возможности и области применения современных программных систем для выполнения математических расчетов на ЭВМ;
- раскрыть принципы работы в системах компьютерной математики при проведении вычислений, подготовке и оформлению результатов расчета;
- показать особенности выполнения вычислений со скалярными величинами и матрицами (массивами) в системах компьютерной математики;
- предоставить обучающимся возможности для формирования умений и навыков работы в системах компьютерной математики на примере системы инженерных и научных расчетов MATLAB.

ОБЩИЕ СВЕДЕНИЯ О СИСТЕМАХ КОМПЬЮТЕРНОЙ МАТЕМАТИКИ

В практике инженерной деятельности часто возникают задачи, связанные с выполнением математических расчетов. Сложность производимых вычислений, как правило, определяется сложностью решаемой задачи. В ряде простых случаев требуется выполнение расчетов по заданным точным аналитическим зависимостям (прямые вычисления). В более сложных случаях возникает необходимость выполнения многократно повторяющихся вычислений, расчетов по приближенным зависимостям с оценкой точности. Применение ЭВМ позволяет автоматизировать выполнение математических операций, ускорить расчеты и увеличить точность вычислений.

Как правило, выполнение математических вычислений с помощью ЭВМ предполагает этапы разработки и отладки программы. Указанные этапы, особенно в случае решения сложных задач, являются достаточно трудоемкими, и требуют от инженера серьезных знаний, навыков и опыта в области программирования.

Другим инструментом для решения вычислительных задач на ЭВМ являются так называемые системы компьютерной математики (СКМ). Эти системы позволяют облегчить процесс решения вычислительных задач, оформлять результаты в виде, удобном для дальнейшего анализа (построение графиков, диаграмм, таблиц и т.п.). Как правило, СКМ имеют интерфейс, позволяющий вводить операторы, математические выражения, команды в виде более привычном и известном нам по курсу математики.

На рынке современных математических систем в настоящее время присутствует целый ряд крупных фирм: *Macsyma, Inc.*, *Waterloo Maple Software, Inc.*, *Wolfram Research, Inc.*, *MathWorks, Inc.*, *MathSoft, Inc.*, *SciFace GmbH* и др. К разработке каждой такой математической системы привлекаются сотни профессионалов из известных университетов и крупных научных центров, а также высококвалифицированные программисты и эксперты в области проектирования сложных программных систем. Ими разработан ряд программных продуктов, включающих существенные математические понятия и обладающих богатым набором методов для решения общих и специальных математических и научно-технических задач.

Характерными представителями СКМ являются: **MATLAB**, **MATHCAD**, **Maple**, **Mathematica**, **Macsyma**, **MuPAD**, **S-PLUS** и др. По способу взаимодействия с пользователем эти системы условно можно разделить на две группы. К первой группе можно отнести СКМ, в которых вычисления выполняются подобно тому, как это делается и в языках программирования высокого уровня, но с тем отличием, что эти системы имеют значительное количество встроенных и хорошо документированных математических библиотек, а также графический

интерфейс для построения различных графиков. Примером такой СКМ является MATLAB. К другой группе СКМ следует отнести программы, в которых запись математических выражений, операторов и функций представлена в виде, привычном для математиков. В таких системах за счет использования панелей инструментов имеется возможность вводить греческие символы, символы математических операций (интегралов, производных, деления, умножения, возведения в степень и др.). Примером такой СКМ является MATHCAD.

Отличительной чертой некоторых СКМ является возможность проводить вычисления в символьном виде. Это позволяет получать результат не в числовой, а в аналитической форме, упрощать математические выражения. Характерным представителем такой СКМ является программа Maple.

В дальнейшем основное внимание уделено изучению СКМ MATLAB и изучению на ее примере основных вычислительных алгоритмов, используемых в практике инженерных расчетов. Система MATLAB (сокращение от MATrix LABoratory – матричная лаборатория) представляет собой интерактивную компьютерную систему для выполнения инженерных и научных расчетов, ориентированную на работу с массивами данных. MATLAB – это чрезвычайно развитая система, которая содержит большое количество процедур и функций, необходимых инженеру и научному работнику для осуществления сложных численных расчетов, моделирования поведения технических и физических систем, оформления результатов этих расчетов в наглядном виде.

ЧАСТЬ I. ВВЕДЕНИЕ В MATLAB

1.1. Рабочая среда системы MATLAB

При запуске MATLAB на экране открывается рабочая среда MATLAB, изображенная на *рис. 1.1*.

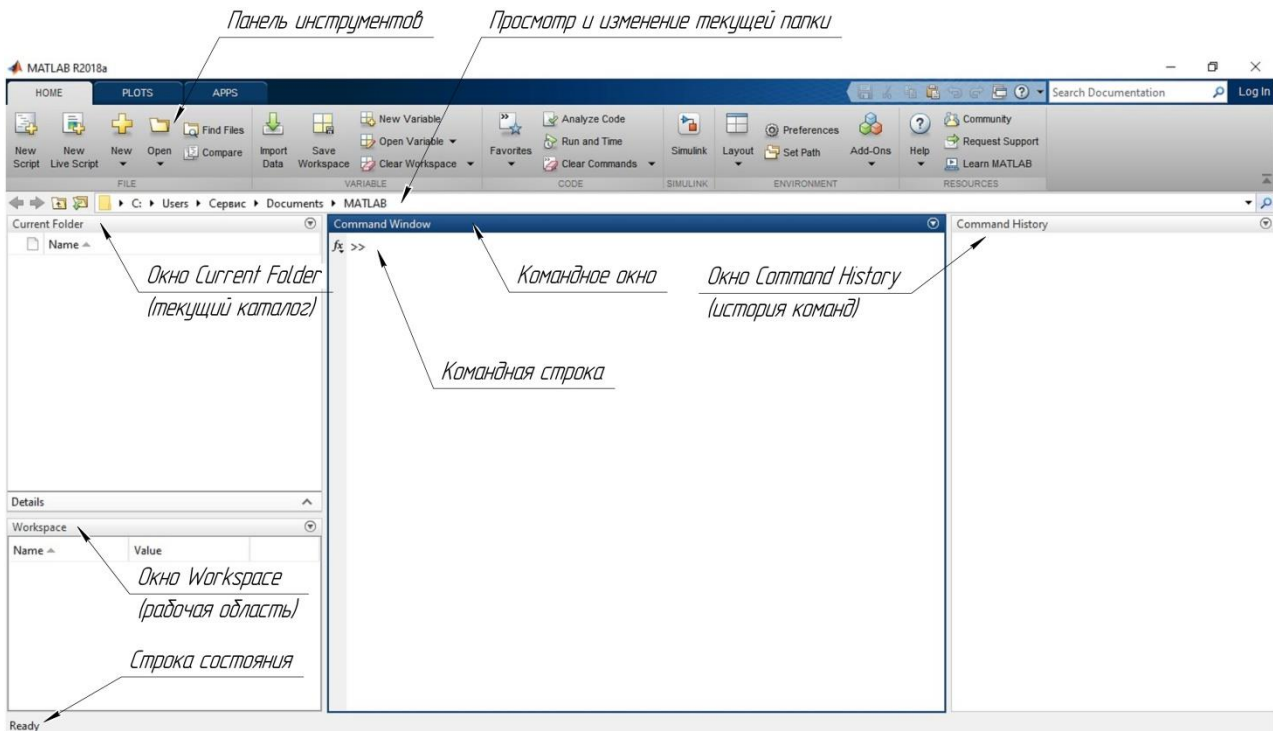


Рис. 1.1. Рабочая среда MATLAB

Основными элементами рабочей среды являются:

- панель инструментов с кнопками и раскрывающимся списком;
- окна **Workspace** и **Current Folder** для просмотра переменных и содержимого текущего каталога (текущей папки);
- окно **Command Window**, служащее для ввода команд и вывода результата;
- окно **Command History**, предназначенное для просмотра и повторного выполнения ранее введенных команд (окно **Command History** может быть не пустым, если до этого пакет **MATLAB** использовался);
- строка состояния.

Окно **Command Window** состоит из следующих элементов:

- заголовка с названием окна и кнопкой справа, предназначенной для вызова контекстного меню;
- рабочей области с командной строкой, в которой находится мигающий вертикальный курсор;
- полос скроллинга.

Все команды следует набирать в командной строке после символа “ >> ”. Набор команды или выражения должен заканчиваться нажатием на клавишу <Enter> для того, чтобы программа MATLAB выполнила эту команду или вычислила выражение. В командной строке можно выполнять простейшие вычисления. Например, набрав 1+2 и нажав на <Enter>, в командном окне MATLAB можно получить следующий результат:

```
>> 1+2
ans =
     3
>>
```

Таким образом, программа MATLAB вычислила сумму. Так как мы не указали системе, куда следует сохранить результат вычисления, то MATLAB записал результат в специальную переменную ans и вывел ее значение, равное 3, в командное окно. Переменная ans, если она не была создана ранее, всегда создается автоматически, когда пользователь не указал явно имя переменной, куда следует записать результат вычисляемого выражения или выполняемой команды.

В окне Command Window ниже ответа расположена командная строка с мигающим курсором, обозначающая, что среда MATLAB готова к дальнейшим вычислениям.

Если требуется продолжить работу с предыдущим выражением, например, вычислить (1+2)/4.5, то проще всего воспользоваться уже имеющимся результатом, который хранится в переменной ans, например:

```
>>ans/4.5
ans =
    0.6667
```

Как видно из представленного примера, значение переменной ans изменилось.

Рассмотрим следующий пример. Пусть требуется вычислить значение следующего математического выражения:

$$\frac{\left(15\frac{1}{7} - 3\frac{1}{12} + 1,45\right) \cdot 11,73 - 6,051}{3,02 \cdot 8,47}.$$

При вычислении данного выражения необходимо учитывать, что запись $15\frac{1}{7}$ следует читать как “пятнадцать и одна седьмая”, что не соответствует операции умножения числа 15 на дробь $\frac{1}{7}$, т. е. $15\frac{1}{7} \neq 15 \cdot \frac{1}{7}$. В данном случае выражение $15\frac{1}{7}$ эквивалентно следующему математическому выражению

$$15\frac{1}{7} = 15 + \frac{1}{7}.$$

С учетом данного замечания, рассматриваемое математическое выражение в командном окне среды MATLAB можно записать в следующем виде:

```
>>( ( 15+1/7 - ( 3+1/12 ) + 1.45 ) * 11.73 - 6.051 ) / ( 3.02 * 8.47 )
```

По окончании ввода этого выражения следует нажать на клавишу <Enter>. Результат вычислений представлен на *рис. 1.2*.

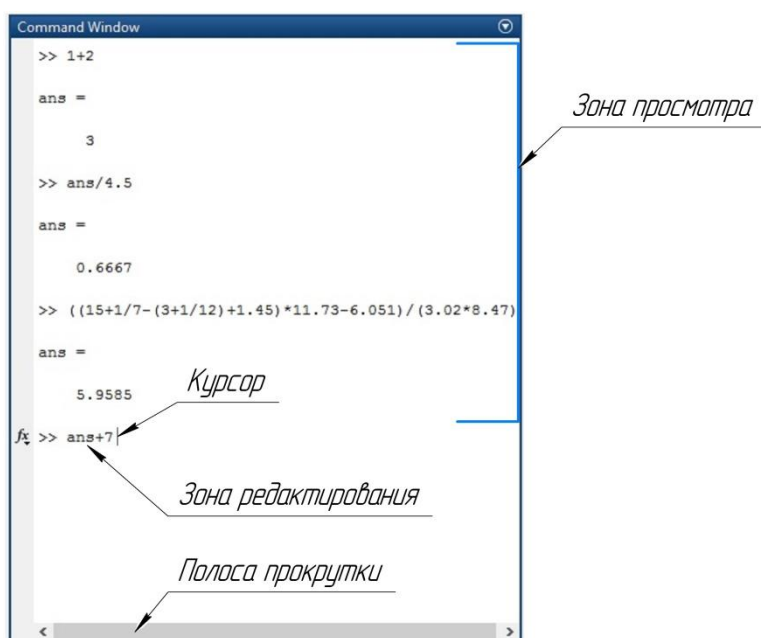


Рис. 1.2. Окно Command Window в процессе выполнения вычислений

Как видно из *рис. 1.2*, вся информация, размещенная в командном окне, располагается в двух зонах: *зоне просмотра* и *зоне редактирования*. Информацию (команды, выражения), размещенную в зоне просмотра, исправить (отредактировать) *нельзя*. В любую строку зоны просмотра с помощью мыши может быть помещен курсор, однако при попытке ввода данных с клавиатуры курсор автоматически будет перемещен в зону редактирования. В зоне просмотра с помощью мыши можно лишь выделять любую информацию и копировать ее в буфер обмена операционной системы Windows. После этого содержимое буфера обмена может быть вставлено в строку редактирования, в документ Mi-

Microsoft Word или другую программу, допускающую работу с буфером обмена Windows.

Расположенные на клавиатуре клавиши $\langle \uparrow \rangle$ (стрелка вверх) и $\langle \downarrow \rangle$ (стрелка вниз), используемые в других программах, как правило, для управления движением курсора вверх-вниз, в системе MATLAB работают иначе. Эти клавиши позволяют вернуть в строку ввода (зону редактирования) введенные ранее с клавиатуры команды и другую информацию. Введенные при этом команды и выражения могут быть отредактированы и запущены на выполнение (после нажатия на клавишу $\langle \text{Enter} \rangle$).

Вводимая в командную строку информация запоминается системой MATLAB в специальной области памяти, которую называют *стеком команд*. Содержимое этой области памяти отображается также в окне Command History. Перемещение по вертикали в зоне просмотра возможно с использованием полосы вертикальной прокрутки (полосы скроллинга), а также с помощью клавиш $\langle \text{PageUp} \rangle$ и $\langle \text{PageDown} \rangle$.

Основным типом данных, с которым производятся вычисления в среде MATLAB, являются *вещественные числа*. Вещественные числа задаются в системе MATLAB с использованием *экспоненциальной формы записи*. Эта форма записи представляет действительные числа в виде *мантиссы* и *порядка*. Она удобна при представлении очень больших и очень маленьких чисел, а также, для унификации их написания. Экспоненциальный формат записи вещественных чисел имеет вид

$$N = M \cdot n^p,$$

где N – записываемое число, M – мантисса, $n = 10$ – основание степени, p – порядок числа (целое).

Например, запись

$$-2.851038547e + 012$$

обозначает следующее:

$$\begin{array}{ccccccc}
 - & \underbrace{2.851038547} & e & + & 012 \\
 \text{знак числа} & \text{мантисса} & \text{обозначение основания} & \text{знак} & \text{порядок} \\
 & & \text{степени, равное 10} & \text{порядка} &
 \end{array}$$

Таким образом, записано число: $-2.851038547 \cdot 10^{12}$.

В среде MATLAB вещественные числа представлены типом данных, который называется *double*. Под мантиссу и показатель степени (на машинном уровне используется двоичная система записи) отводится 8 байт памяти. В результате для десятичных чисел достигается точность порядка 15 значащих цифр.

Следует иметь в виду, что при записи вещественных чисел в MATLAB для отделения целой части числа от дробной используется не запятая (“,”), как это принято в математике, а десятичная точка (“.”).

При выводе значений вычисленных выражений в командном окне по умолчанию используется формат `short`, при котором показываются только четыре десятичные цифры после запятой. Например:

```
>> 100/3
ans =
    33.3333
```

Если при отображении слишком большого или малого числа результат не укладывается в формат `short`, тогда результат выводится в экспоненциальной форме:

```
>> 100000/3
ans =
    3.3333e+004
```

Первоначальная установка формата сохраняется и при дальнейших вычислениях – для небольших чисел вывод результата снова будет происходить в формате `short`.

Короткий формат `short` вывода результатов вычислений, установленный в **MATLAB** по умолчанию, может быть изменен пользователем. Задавать формат вывода можно непосредственно из командной строки при помощи команды `format`. Например, для установки длинного с плавающей точкой формата вывода результатов вычислений следует ввести команду `format long e` в командной строке:

```
>> format long e
>> 1.25/3.11
ans =
    4.019292604501608e-001
```

Независимо от того, какой формат вывода установлен, вычисления **MATLAB** производит с *двойной точностью*.

1.1.1. Использование переменных в среде MATLAB

Для организации вычислений в среде **MATLAB** допускается использование переменных. Объявлять используемые переменные заранее, как это требуется в ряде других языков программирования, не нужно. По умолчанию (если пользователь не указал явно) все переменные относятся к вещественным типа `double`. Имена переменных должны начинаться с буквы и состоять из букв, цифр и знаков подчеркивания. Например:

```
x, y, x1, y1, x_1, y_1, Rab, Lab, R_ab, L_ab.
```

Недопустимым является использование имен переменных, содержащих буквы кириллицы, использование в именах переменных пробелов, знаков математических операций, недопустима ситуация когда имя переменной начинается с цифры.

Имя переменной может содержать до 30 символов и не должно совпадать с именами функций, команд среды **MATLAB** и системных переменных. При

этом **различается регистр символов** в именах переменных (большие и маленькие буквы). Так, например, следующие имена

amenu, Amenu, aMenu

в среде MATLAB обозначают разные переменные.

MATLAB различает прописные и строчные буквы. Попытка набора команды прописными буквами приведет к ошибке.

Знак равенства (“=”) закреплен за операцией присваивания, т. е. выражение вида

```
>> x = 10
```

означает “присвоить переменной с именем x значение, равное 10”.

При задании значений переменным допустимо использовать экспоненциальную форму записи. В этом случае важно помнить, что символ e обозначает не число e (основание натурального логарифма), а основание степени, равное 10. В следующем примере создается переменная z со значением $-7.5 \cdot 10^2$:

```
>> z = -7.5e+2
```

Ввод команд и математических выражений в командном окне должен завершаться нажатием на клавишу <Enter>. Символ (“;”) служит для отделения команд и выражений друг от друга, когда они вводятся в одной строке, а также для предотвращения вывода результата вычислений на экран в командное окно.

Рассмотрим следующий пример:

```
>> x = 10; y = 25.8;
```

```
>> R = 0.5*(x^2 + y^2)
```

```
R =
```

```
382.8200
```

```
>>
```

В данном случае в первой строке выполняются две операции – переменным x и y присваиваются соответствующие значения. Так как две операции записаны в одной строке, то они отделяются друг от друга символом (“;”). Этот же символ предотвращает вывод результата операции в командное окно после нажатия на клавишу <Enter>. Во второй строке вычисляется следующее математическое выражение:

$$R = \frac{1}{2}(x^2 + y^2).$$

Символ (“^”) используется для записи операции возведения числа в степень. В конце данного выражения отсутствует точка с запятой, и именно поэтому результат вычислений отображен в командном окне (рис. 1.3, а). После вычислений появится опять символ ввода “>>”.

В случае, если в последнем выражении поставить в конце точку с запятой, то результат вычислений будет присвоен переменной R, но не будет выведен в командное окно (рис. 1.3, б).

```

>> x = 10; y = 25.8;
>> R = 0.5*(x^2 + y^2)

R =

    382.8200

>>

```

a

б

Рис. 1.3. Использование точки с запятой в арифметических выражениях

Все значения переменных, вычисленные в течение текущего сеанса работы, сохраняются в специально зарезервированной области памяти, называемой рабочим пространством системы MATLAB (Workspace).

Содержимое рабочего пространства отображается в окне Workspace (см. рис. 1.1). Узнать о содержимом рабочего пространства можно также с помощью команд `who` или `whos`. Разница в применении указанных команд хорошо видна из примеров, представленных на рис. 1.4.

```

>> who
Your variables are:
R  x  y

>> whos
      Name      Size      Bytes  Class  Attribute
      ----      -
      R          1x1         8  double
      x          1x1         8  double
      y          1x1         8  double

>>

```

a

б

Рис. 1.4. Использование команд `who` и `whos`

В первом случае (рис. 1.4, *a*) отображаются только имена переменных, хранящихся в данный момент в рабочей области, а во втором (рис. 1.4, *б*) – показывается дополнительная информация о размере переменной, количестве байтов, и типе переменной.

Командой

```
>> clc
```

можно стереть видимое содержимое командного окна (очистить окно), однако это не затронет содержимого рабочего пространства. Если после этого набрать имя ранее созданной переменной, например,

```
>> R
```

и нажать на клавишу `<Enter>`, то на экране будет показано значение этой переменной.

Удаление всех созданных в текущем сеансе работы системы переменных из рабочей области памяти осуществляется с помощью команды

```
>> clear
```

Восстановить удаленные значения нельзя!

Если требуется удалить лишь некоторые переменные из рабочей области, то необходимо указать имена этих переменных в качестве параметров команды `clear`:

`>> clear x y`

В этой команде имена стираемых переменных отделяются друг от друга пробелом (не запятой).

Система **MATLAB** имеет несколько имен переменных, которые используются самой системой и входят в состав зарезервированных:

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

`pi` – число π (сохраняется в виде 3.141592653589793);

`inf` – обозначение машинной бесконечности;

`NaN` – обозначение неопределенного результата (например, типа $0/0$ или `inf/inf`);

`eps` – погрешность операций над числами с плавающей запятой;

`ans` – результат последней операции без знака присваивания;

`realmax` и `realmin` – максимально и минимально возможные величины числа, которое может быть использованы.

Эти переменные можно использовать в математических выражениях.

Задания для самостоятельного выполнения

Вычислите указанное арифметическое выражение.

Арифметические выражения	Арифметические выражения
1. $\frac{\left(12\frac{1}{6} - 6\frac{1}{27} - 5,25\right)13,5 - 0,111}{0,02}$	5. $18\frac{1}{6} - 1\frac{11}{14} - \frac{2}{15} \cdot 2\frac{6}{7}$
2. $\frac{\left(1\frac{1}{12} + 2\frac{5}{32} + \frac{1}{24}\right) : 9,6 + 2,13}{0,0004}$	6. $\frac{\left(6,6 - 3\frac{3}{14}\right)5\frac{5}{6}}{(21 - 1,25) : 2,5}$
3. $\frac{\left(58\frac{4}{15} - 56\frac{7}{24}\right) : 0,8 + 2\frac{1}{9} \cdot 0,225}{8,75 \cdot 0,6}$	7. $\frac{\left(\frac{0,216}{0,15} + 0,56\right) : 0,5}{\left(7,7 : 24,75 + \frac{2}{15}\right)4,5}$
4. $\frac{2,625 - \frac{2}{3} \cdot 2\frac{5}{14}}{\left(3\frac{1}{12} + 4,375\right) : 19\frac{8}{9}}$	8. $\frac{1\frac{4}{11} \cdot 0,22 : 0,3 - 0,96}{\left(0,2 - \frac{3}{40}\right)1,6}$

Арифметические выражения	Арифметические выражения
9. $\frac{\left(\frac{3}{5} + 0,425 - 0,005\right) : 0,12}{30,5 + \frac{1}{6} + 3\frac{1}{3}}$	13. $\frac{\left(5\frac{4}{45} - 4\frac{1}{6}\right) : 5\frac{8}{15}}{\left(4\frac{2}{3} + 0,75\right) \cdot 3\frac{9}{13}} \cdot 34\frac{2}{7}$
10. $\frac{3\frac{1}{3} + 2,5}{2,5 - 1\frac{1}{3}} \cdot \frac{4,6 - 2\frac{1}{3}}{4,6 + 2\frac{1}{3}} : \left(\frac{0,05}{\frac{1}{7} - 0,125} + 5,7\right)$	14. $\frac{1\frac{4}{11} \cdot 0,22 : 0,3 - 0,96}{\left(0,2 - \frac{3}{40}\right) \cdot 1,68}$
11. $\frac{0,725 + 0,426}{0,128 - 6,25 - (0,0345 : 0,12)} \cdot 0,25$	15. $\frac{\left(4,5 \cdot 1\frac{2}{3} - 6,75\right) \cdot 0,6}{\left(3,333 \cdot 0,3 + 0,222 \cdot \frac{4}{9}\right) \cdot 2\frac{2}{3}}$
12. $\frac{\left(40\frac{7}{30} - 38\frac{5}{12}\right) : 10,9 + \left(0,875 - \frac{7}{30}\right) \cdot \frac{20}{11}}{0,008}$	

1.1.2. Использование математических функций в среде MATLAB

В языке MATLAB предусмотрены следующие элементарные арифметические функции (функций на самом деле намного больше, ниже приводятся лишь наиболее употребляемые).

Тригонометрические функции

$\sin(Z)$ – синус числа Z	$\text{asin}(Z)$ – арксинус (в радианах, в диапазоне от $-\pi/2$ к $+\pi/2$)
$\cos(Z)$ – косинус	$\text{acos}(Z)$ – арккосинус (в диапазоне от 0 к π)
$\tan(Z)$ – тангенс	$\text{atan}(Z)$ – арктангенс (в диапазоне от $-\pi/2$ к $+\pi/2$)
$\cot(Z)$ – котангенс	$\text{acot}(Z)$ – арккотангенс

Экспоненциальные функции:

$\exp(Z)$ – экспонента числа Z ;
 $\log(Z)$ – натуральный логарифм;
 $\log_{10}(Z)$ – десятичный логарифм;
 $\text{sqrt}(Z)$ – квадратный корень из числа Z ;
 $\text{abs}(Z)$ – модуль числа Z .

Целочисленные функции:

$\text{fix}(Z)$ – округление к ближайшему целому в сторону нуля;

$\text{floor}(Z)$ – округление к ближайшему целому в сторону отрицательной бесконечности;

$\text{ceil}(Z)$ – округление к ближайшему целому в сторону положительной бесконечности;

$\text{round}(Z)$ – обычное округление числа Z к ближайшему целому;

$\text{mod}(X, Y)$ – целочисленное деление X на Y ;

$\text{rem}(X, Y)$ – вычисление остатка от деления X на Y ;

$\text{sign}(Z)$ – вычисление знака числа Z (0 при $Z = 0$, -1 при $Z < 0$, 1 при $Z > 0$).

Для изменения порядка действий при вычислениях используются круглые скобки, например:

```
>> 4.5^2*(7.23 - 3.14*10.4)
```

```
ans =
```

```
-514.8765
```

```
>> 4.5^2*7.23 - 3.14*10.4
```

```
ans =
```

```
113.7515
```

В первом случае вычислено выражение $4.5^2 \cdot (7.23 - 3.14 \cdot 10.4)$, а во втором $(4.5^2 \cdot 7.23) - (3.14 \cdot 10.4)$.

При записи выражений необходимо учитывать порядок вычислений. Особенно в случаях, когда имеется операция деления. Например, для выражения вида

$$\frac{a + b}{a - b} \cdot z$$

правильной будет следующая запись:

```
>> (a+b)/(a-b)*z.
```

В тех случаях, когда есть сомнения в порядке вычислений записываемого выражения, лучше проставить требуемый порядок с помощью круглых скобок. Использование скобок другого вида (квадратных, фигурных) недопустимо.

Если требуется вычислить значение выражения, похожего на предыдущее, то необязательно снова набирать его в командной строке. Можно воспользоваться тем, что **MATLAB** запоминает все вводимые команды. Для повторного занесения их в командную строку служат клавиши $\langle \uparrow \rangle$, $\langle \downarrow \rangle$.

При нажатии на клавишу $\langle \uparrow \rangle$ в командной строке появится введенное ранее выражение. Можно ввести в него необходимые изменения (для перемещения по строке с выражением служат клавиши $\langle \leftarrow \rangle$, $\langle \rightarrow \rangle$, $\langle \text{Home} \rangle$, $\langle \text{End} \rangle$) и вычислить измененное выражение, нажав на $\langle \text{Enter} \rangle$.

При вычислениях возможны некоторые исключительные ситуации. При делении положительного числа на ноль в **MATLAB** получается inf (бесконечность), а при делении отрицательного числа на ноль получается $-\text{inf}$ (минус бесконечность) и выдается предупреждение:


```
>> 1/0
Warning: Divide by zero
ans =
     Inf
```

При делении нуля на ноль получается NaN (не число) и также выдается предупреждение:

```
>> 0/0
Warning: Divide by zero
ans =
     NaN
```

При вычислении, например, $\sqrt{-1}$ MATLAB автоматически переходит в область комплексных чисел:

```
>> sqrt(-1.0)
ans =
     0 + 1.0000i
```

При наборе комплексных чисел в командной строке MATLAB можно использовать либо *i*, либо *j*, а сами числа при умножении, делении и возведении в степень необходимо заключать в круглые скобки:

```
>> (2.1 + 3.2i)*2+(4.2 + 1.7i)^2
ans =
    18.9500 + 20.6800i
```

Если не использовать скобки, то умножаться или возводиться в степень будет только мнимая часть, и получится неверный результат.

Для вычисления комплексно-сопряженного числа применяется апостроф, который следует набирать сразу за числом, без пробела:

```
>> 2 - 3i'
ans =
    2.0000 + 3.0000i
```

Если необходимо найти комплексно-сопряженное выражение, то исходное выражение должно быть заключено в круглые скобки:

```
>> ((3.2 + 1.5i)*2 + 4.2 + 7.9i)'
ans =
    10.6000 - 10.9000i
```

MATLAB позволяет использовать комплексные числа в качестве аргументов встроенных элементарных функций:

```
>> sin(2 + 3i)
ans =
    9.1545 - 4.1689i
```

Рассмотрим функции для работы с комплексными числами. К ним относятся следующие:

• $\text{abs}()$, $\text{angle}()$ — модуль r и фаза φ (в радианах от $-\pi$ до π) комплексного числа $z = a + i \cdot b = R \cdot (\cos \varphi + i \cdot \sin \varphi) = R e^{-i\varphi}$ (если число не комплексное, то функция $\text{abs}(x)$ соответствует $|x|$);

• $\text{conj}()$ — возвращает комплексно-сопряженное число;

• $\text{imag}()$, $\text{real}()$ — мнимая и действительная части комплексного числа.

Более подробные сведения можно получить из обширной интерактивной справочной системы, выбрав в меню рабочей среды пункт **Help**.

Задания для самостоятельного выполнения

1. Проведите вычисления по заданной формуле при заданных значениях параметров с использованием встроенных математических функций системы MATLAB.

Выражение	Значение параметров	
	а)	б)
1. $3m^2 + \frac{\sqrt[3]{2n^2}}{m}$	$m = -\frac{14}{5}, n = \text{tg} \frac{\pi}{8}$	$m = 2,2 \cdot 10^{-2}, n = \frac{1}{3,1}$
2. $\frac{4}{3} l^3 \sin^2 \frac{\alpha}{2} \sqrt{\cos \alpha}$	$l = 1,7 \cdot 10^3, \alpha = 18^\circ$	$l = \frac{16}{21}, \alpha = \frac{\pi}{5}$
3. $\sqrt{\frac{a\sqrt{b}}{\sqrt[3]{\text{tg} \alpha}}}$	$a = 1,5, b = 0,8, \alpha = 61^\circ$	$a = 3 \cdot 10^{-2}, b = 0,71,$ $\alpha = \frac{3}{7} \pi$
4. $\frac{3a^2 \sqrt{6,8 \cdot (a-b)}}{4(a+b)^3}$	$a = 4,13 \cdot 10^{-1}, b = \frac{1}{261}$	$a = \sin \frac{5\pi}{8},$ $b = -\text{tg} 12^\circ$
5. $\frac{c^3}{6} \cos \frac{\alpha}{2} \sqrt{\sin \alpha}$	$c = \lg 2,38, \alpha = \frac{\pi}{5}$	$c = e^{-0,3}, \alpha = 65^\circ$
6. $\sqrt{\frac{n^3}{16,3 \sin \alpha \sin 2\alpha}}$	$n = 3,1516 \cdot 10^{-2}, \alpha = 5^\circ$	$n = e^{3,5}, \alpha = \frac{2\pi}{13}$
7. $5 \sin 35^\circ \sqrt{\frac{S^3 \cos 36^\circ}{\pi^3 \text{tg} \alpha}}$	$S = \ln 3, \alpha = 44^\circ$	$S = \frac{18}{25}, \alpha = \frac{7}{12} \pi$
8. $ \lg(1 + \sin \alpha) + \ln(1 - \sin \beta) $	$\alpha = \frac{3\pi}{7}, \beta = 83^\circ$	$\alpha = \frac{2}{3} \pi, \beta = 16^\circ$
9. $\sqrt[3]{\sin^2(\alpha + \beta) - \sin^2(\alpha - \beta)}$	$\alpha = \frac{5}{7} \pi, \beta = 0,3\pi$	$\alpha = 12^\circ, \beta = 220^\circ$

Выражение	Значение параметров	
	а)	б)
10. $(\log_a (b+1,4))^{-3/4}$ <i>примечание: использовать преобразование</i> $\log_a b = \frac{\log_c b}{\log_c a}$	$a = 3,56, b = e^{0,316}$	$a = 2,$ $b = 2,1649 \cdot 10^{-2}$
11. $3 \cdot (p^{-2/3} + q^{-1/2}) \sqrt[3]{p \cdot q}$	$p = \ln 3, q = \lg 3$	$p = 0,013,$ $q = 1,4 \cdot 10^2$
12. $\frac{2}{3} m \sqrt{m \cdot \sqrt[3]{m \cdot \sqrt[4]{m}}}$	$m = 3,6485 \cdot 10^2$	$m = \frac{24}{37}$
13. $\frac{8}{3} S \sqrt{\frac{S}{\pi}} \sin^6 \frac{\alpha}{2}$	$S = e^{1,11}, \alpha = \frac{7}{11} \pi$	$S = 5,403, \alpha = 28^\circ$
14. $2 \sqrt{\frac{F}{\pi}} \operatorname{tg} \alpha \sin^2 \frac{\alpha}{2}$	$F = \frac{1}{0,03}, \alpha = \frac{5}{7} \pi$	$F = \ln 7, \alpha = 1,34^\circ$
15. $\frac{1}{12} \cdot \frac{m^3 \cos \alpha}{(\sin \alpha + \cos \alpha)^3}$	$m = -20,1, \alpha = 20^\circ$	$m = \lg 13,6, \alpha = 1,48$

2. Найти значение алгебраических выражений с использованием встроенных математических функций системы MATLAB.

Алгебраические выражения	
1.	$f_1 = \cos x + 2 \operatorname{tg} \frac{3\pi}{4} - 0,3 \lg x_1 + 2^{x_1}; f_2 = 4 \arcsin \frac{1}{2} - \sqrt{2x_1} + \operatorname{ctg} \alpha;$ $f_3 = \operatorname{arctg} \sqrt{3} + e^2 + 2 \ln x_1 - \frac{\sin 3x}{\sqrt{3}}; \text{ при } x = \frac{2\pi}{3}, x_1 = 8, \alpha = \frac{\pi}{2}$
2.	$s_1 = \sin b - \cos b_1 + 0,2 \sqrt{a_1} + 3e^{a_1}; s_2 = 3 \operatorname{ctg} \varphi - 6 \lg 100 + \lg 16 + 3^{a_1};$ $s_3 = \arccos \frac{1}{2} + 4 \operatorname{arctg}(-1) - \frac{\operatorname{arctg} \sqrt{3}}{2}; \text{ при } b = \frac{5\pi}{6}, b_1 = \frac{7\pi}{4}, a_1 = 5, \varphi = \frac{3\pi}{4}$
3.	$p_1 = \sin 2\beta + \cos \frac{4\pi}{3} + 0,5 \operatorname{tg} \beta_1 + \frac{\operatorname{ctg} \frac{3\pi}{4}}{3}; p_2 = -\ln 16 + \lg k + e^3 + 2\sqrt{2k};$ $p_3 = \arcsin \frac{\sqrt{3}}{2} - \operatorname{arctg} 1 + \operatorname{arctg}(-\sqrt{3}) + k^2; \text{ при } \beta = \frac{\pi}{3}, \beta_1 = \frac{\pi}{4}, k = 32$

Алгебраические выражения

4. $x_0 = \sqrt{a+2} + a^{0,5} - \frac{1}{e^{-2}} + \lg a_1 - \lg a + 1,8 \ln 6;$
 $x_1 = \arccos \frac{\sqrt{2}}{2} - 1,3 \arcsin \frac{\sqrt{3}}{2} + \operatorname{arctg}(-\sqrt{3}) + \frac{\operatorname{arcctg}(-1)}{(a-5)};$
 $x_2 = \sin \frac{3\pi}{4} + 2 \cos \alpha - \operatorname{tg} \frac{5\pi}{6} + 3,5 \operatorname{ctg} \frac{7\pi}{4};$ при $a=14, a_1=1000, \alpha = \frac{7\pi}{6}$

5. $y = 1,5 \operatorname{arctg} x_1 - \arcsin \frac{x_1}{2} + 10 \arccos(-1);$
 $y_1 = 3 \operatorname{tg} \alpha + 1,8 \operatorname{ctg} \frac{\alpha}{3} - \sin \frac{5\pi}{6} - \cos \frac{4\pi}{3};$
 $y_2 = \sqrt{x_2+4} + e^3 - \lg 32 + \ln x_2 + 4^5;$ при $x_1 = \sqrt{3}, \alpha = \frac{3\pi}{4}, x_2 = 60$

6. $a = \arcsin \frac{\sqrt{2}}{2} + \operatorname{arctg} 2b - \operatorname{arcctg} \frac{1}{\sqrt{3}} + 2 \arccos \frac{1}{2};$
 $a_1 = m^{\frac{2}{3}} - e^{-3} - \sqrt{2+b} + 1,4 \lg 100 - 4,7 \ln 12 - \ln 64;$
 $a_2 = -2 \operatorname{tg} \frac{\pi}{6} + 12 \cos \beta - 3 \sin \beta + \operatorname{ctg} \frac{4\pi}{3};$ при $b = \frac{1}{2}, m = \frac{1}{8}, \beta = \frac{7\pi}{6}$

7. $q = \sqrt{2k_1} + (k_1-3)^{\frac{3}{4}} - e^{-2} + \cos \varphi - 3,4 \sin \frac{5\pi}{6};$
 $n = \lg 128 - 10 \ln 15 - \lg k_2 + \frac{\operatorname{tg} \varphi}{4} - \operatorname{ctg} \varphi;$
 $m = \arcsin \frac{\sqrt{3}}{2} + \arccos k_3 - \operatorname{arctg} \sqrt{3};$ при $k_1=16, \varphi = \frac{3\pi}{4}, k_2=500, k_3 = -\frac{\sqrt{2}}{2}$

8. $z = -2(\sqrt{m} + e^2) - m^{\frac{3}{8}} + \lg(6m+10) - \lg 32;$
 $z_1 = \operatorname{arctg}(-\sqrt{3}) - \arcsin\left(-\frac{1}{2}\right) + 3 \arccos\left(-\frac{\sqrt{3}}{2}\right);$
 $z_2 = \cos \alpha_1 - 0,4 \sin \frac{\pi}{3} + \operatorname{tg} \frac{2\pi}{3} + \operatorname{ctg} \alpha_2;$ при $m=15, \alpha_1 = \frac{\pi}{6}, \alpha_2 = \frac{5\pi}{6}$

Алгебраические выражения

9. $g = \frac{3\sqrt{x} + 4}{e^x} + \lg 50x - x^{\frac{4}{5}} + 1,8 \lg x_1 - \ln x_2;$

$$g_1 = \arcsin \frac{\sqrt{2}}{2} - \arccos \left(-\frac{\sqrt{2}}{2} \right) + \operatorname{arctg} \frac{1}{\sqrt{3}} + 2 \operatorname{arccotg} \sqrt{3};$$

$$g_2 = \sin \frac{3\pi}{4} + \frac{\cos \beta}{2x} - \operatorname{tg} \frac{\pi}{3} + 2 \operatorname{ctg} \frac{3\pi}{4}; \text{ при } x = 2, x_1 = 8x, x_2 = 5x, \beta = \frac{7\pi}{6}$$

10. $m = 0,8 \sin 3x_1 + \frac{\operatorname{tg} \frac{\pi}{4}}{\operatorname{ctg} x_2} + \cos x_2 - \left(3\sqrt{x_3} + \lg x_3 \right)^{\frac{1}{5}};$

$$m_1 = \operatorname{arctg} \sqrt{3} - \arcsin \left(-\frac{1}{2} \right) + \frac{\arccos 0,5}{\operatorname{arccotg}(-1)};$$

$$m_2 = -e^{-4} + \lg x_3 - 1,5 \lg(10x_3); \text{ при } x_1 = \frac{\pi}{2}, x_2 = \frac{7\pi}{6}, x_3 = 8$$

11. $a = \operatorname{arctg} \sqrt{3} + e^2 + 2 \ln x_1 - \frac{\sin 3x}{\sqrt{3}}; a_1 = \cos x + 2 \operatorname{tg} \frac{3\pi}{4} - 0,3 \lg x_1 + 2^{x_1};$

$$a_2 = 4 \arcsin \frac{1}{2} - \sqrt{2x_1} + \operatorname{ctg} \alpha; \text{ при } x = \frac{2\pi}{3}, x_1 = 8, \alpha = \frac{\pi}{2}$$

12. $f = \arccos \frac{1}{2} + 2 \operatorname{arccotg}(-1) - \frac{\operatorname{arctg} \sqrt{3}}{2}; f_1 = 3 \operatorname{ctg} \varphi - 6 \lg 100 + \lg 16 + 3^{a_1};$

$$f_2 = \sin b - \cos b_1 + 0,5 \sqrt{a_1} + 6e^{a_1}; \text{ при } b = \frac{5\pi}{6}, b_1 = \frac{7\pi}{4}, a_1 = 5, \varphi = \frac{3\pi}{4}$$

13. $d = \lg k - \ln 16 + e^3 + 2\sqrt{2k}; d_1 = \sin 2\beta + \cos \frac{4\pi}{3} + 0,4 \operatorname{tg} \beta_1 + \frac{\operatorname{ctg} \frac{3\pi}{4}}{5};$

$$d_2 = \arcsin \frac{\sqrt{3}}{2} - \operatorname{arccotg} 1 + \operatorname{arctg}(-\sqrt{3}) + k^2; \text{ при } \beta = \frac{\pi}{3}, \beta_1 = \frac{\pi}{4}, k = 32$$

14. $x_1 = \frac{\operatorname{arccotg}(-1)}{(a-5)} + \arccos \frac{\sqrt{2}}{2} - 1,3 \arcsin \frac{\sqrt{3}}{2} + \operatorname{arctg}(-\sqrt{3});$

$$x_2 = \lg a_1 - \lg a + 1,8 \ln 6 + \sqrt{a+2} + a^{0,5} - \frac{1}{e^{-2}};$$

$$x_3 = \sin \frac{3\pi}{4} + 2 \cos \alpha - \operatorname{tg} \frac{5\pi}{6} + 3 \operatorname{ctg} \frac{7\pi}{4}; \text{ при } a = 14, a_1 = 1000, \alpha = \frac{7\pi}{6}$$

Алгебраические выражения

15. $y_1 = 3 \operatorname{tg} \alpha + 1,8 \operatorname{ctg} \frac{\alpha}{3} - \sin \frac{5\pi}{6} - \cos \frac{4\pi}{3};$

$$y_2 = 1,5 \operatorname{arctg} x_1 - \arcsin \frac{x_1}{2} + 10 \operatorname{arccos}(-1);$$

$$y_3 = \sqrt{x_2 + 4} + e^3 - \lg 32 + \ln x_2 + 4^5; \text{ при } x_1 = \sqrt{3}, \alpha = \frac{3\pi}{4}, x_2 = 60$$

Вычислить значение выражения. Результат представить в алгебраической форме (в виде $z = a + b \cdot i$) и в экспоненциальной форме (в виде $z = Ae^{j \cdot \varphi}$).

Выражение	Значение параметров			
	z_1	z_2	z_3	z_4
1. $z = z_1^2 \cdot z_2 : z_3 + z_4$	$4 + 3i$	$2,71e^{i \frac{\pi}{12}}$	$1,82e^{-1,2i}$	$\sqrt{3} - 2i$
2. $z = z_1^2 : z_2 + z_3 - z_4$	$0,8 - 2i$	$3,08e^{i \frac{7\pi}{12}}$	$8,01e^{2i}$	$-5 + \sqrt{2}i$
3. $z = \sqrt{z_1 : z_2} \cdot z_3 + z_4$	$0,7 + 4i$	$1,74e^{i \cdot 0,3\pi}$	$3 + 4i$	$2,1e^{-2,3i}$
4. $z = \sqrt{z_1 \cdot z_2} : z_3 + z_4$	$-3 - 2i$	$3,21e^{15^\circ i}$	$1,2 + 3i$	$2,71e^{-78^\circ i}$
5. $z = \sqrt{z_1 : z_2} \cdot z_3 - z_4$	$2,71e^{i \cdot \frac{\pi}{12}}$	$-0,7 + 4i$	$1,31e^{-i \cdot \frac{5\pi}{12}}$	$-8 - 3i$
6. $z = (z_1 + z_2)^4 \cdot z_3 : z_4$	$3,08e^{i \cdot \frac{7\pi}{12}}$	$-3 - 2i$	$2,03e^{i \cdot \frac{14}{13}}$	$\sqrt{3} + \sqrt{2}i$
7. $z = (\sqrt{z_1} + z_2) \cdot z_3 : z_4$	$1,74e^{0,3\pi i}$	$0,8 - 2i$	$3,28e^{-1,2i}$	$\sqrt{3} - \sqrt{2}i$
8. $z = (z_1 - z_2)^3 \cdot z_3 + z_4$	$3,21e^{15i}$	$4 + 3i$	$\sqrt{3} - 4i$	$1,23e^{111^\circ i}$
9. $z = (z_1 : z_2 + z_3)^3 \cdot z_4$	$1 + 0,5\pi i$	$1,2e^{107^\circ i}$	$0,8 - 2i$	$2,5e^{14^\circ i}$
10. $z = (z_1 : z_2 + z_3)^2 - z_4$	$\sqrt{5} - i$	$0,7e^{1,7i}$	$1,2e^{0,9i}$	$-3 - 2i$
11. $z = \sqrt[3]{z_1 + z_2 + z_3} \cdot z_4$	$0,187 - 3,94i$	$0,3e^{-107^\circ i}$	$-0,7 + 4i$	$1,5e^{23^\circ i}$
12. $z = \sqrt[3]{z_1 \cdot z_2} : z_3 + z_4$	$-1 + \sqrt{5}i$	$2,1e^{211^\circ i}$	$0,4e^{32^\circ i}$	$4 + 3i$
13. $z = (\sqrt[3]{z_1 \cdot z_2} + z_3) : z_4$	$-\sqrt{3} + 4i$	$1,25e^{-0,8i}$	$-3 - 2i$	$0,75e^{0,7i}$
14. $z = (\sqrt[3]{z_1 : z_2} + z_3) \cdot z_4$	$1,2e^{1,7^\circ i}$	$0,18 - 3,9i$	$0,71e^{4i}$	$0,8 - 2i$
15. $z = (\sqrt{z_1 \cdot z_2} - z_3) : z_4$	$0,3e^{-97^\circ i}$	$-1 + \sqrt{5}i$	$-0,7 + 4i$	$5,2e^{71^\circ i}$

1.2. Массивы в среде MATLAB

1.2.1. Формирование одномерных и двумерных массивов в среде MATLAB

Все данные MATLAB представляет в виде массивов. *Массив* — упорядоченная, пронумерованная совокупность однородных данных. У массива должно быть *имя*. Массивы различаются по *размерности*: одномерные, двумерные, многомерные. Одномерный массив принято называть *вектором*, а двумерный массив — *матрицей*. Вектор может быть записан в столбик (*вектор-столбец*) и в строку (*вектор-строка*). *Размером* массива называют число элементов вдоль каждого из измерений. Доступ к элементам осуществляется при помощи *индекса*. В MATLAB нумерация элементов массивов начинается с *единицы*. Это значит, что индексы должны быть больше или равны единице.

Ввод вектор-столбца в командной строке осуществляется в квадратных скобках, элементы вектор-столбца разделяются точкой с запятой:

```
>> a = [1.3; 5.4; 6.9]
a =
    1.3000
    5.4000
    6.9000
```

Для подавления вывода вектора на экран необходимо завершить выражение точкой с запятой:

```
>> b = [7.1; 3.5; 8.2];
```

Ввод вектор-строки осуществляется в квадратных скобках, однако, в отличие от вектор-столбца, элементы следует разделять пробелами или запятыми:

```
>> s1 = [3 4 9 2]
s1 =
    3 4 9 2
```

Из нескольких вектор-столбцов можно составить один, используя квадратные скобки и разделяя исходные вектор-столбцы точкой с запятой:

```
>> v1 = [1; 2];
>> v2 = [3; 4; 5];
>> v = [v1; v2]
v =
     1
     2
     3
     4
     5
```

Для сцепления (соединения) вектор-строк также применяются квадратные скобки, но сцепляемые вектор-строки отделяются пробелами или запятыми:

```
>> g1 = [pi pi/2] ;  
>> g2 = [pi/3 pi/4 pi/5];  
>> g = [g1 g2]  
g =  
    3.1416  1.5708  1.0472  0.7854  0.6283
```

Возможен ввод элементов массива в виде арифметических выражений:

```
>> m = [(12+2)/(3+4) sqrt(16) exp(5)];  
m =  
    2.0000  4.0000  148.4132
```

Возможно задание массивов с комплексными элементами:

```
>> f = [1+2i 7+3i 2-5i];
```

Для формирования упорядоченной числовой последовательности, которая идет от начального значения с заданным шагом и завершается конечным значением, в MATLAB используется оператор “:” (двоеточие) в виде:

Переменная = Начальное значение : Шаг : Конечное значение

Операция формирования диапазона работает следующим образом. Сначала она включает в формируемый массив нижнюю границу диапазона (начальное значение). Затем она к числовому значению прибавляет шаг. Если сумма не превосходит верхней границы диапазона (конечного значения), то она включается в качестве элемента в формируемый массив. Это все повторяется до тех пор, пока очередное значение не превысит верхнюю границу диапазона. Пример:

```
>> x = 0.6:-0.2:0  
x =  
    0.6000  0.4000  0.2000  0.0000  
>> y = 1:2:10  
y =  
    1  3  5  7  9
```

Отсюда видно, что конечное значение в массиве не всегда совпадает с заданной в операции “двоеточие” верхней границей диапазона.

Если шаг равен единице, его можно не указывать.

```
>> x = 1:5;  
x =  
    1  2  3  4  5
```

Доступ к элементам вектора-столбца или вектор-строки осуществляется при помощи индекса, заключаемого в круглые скобки после имени массива, в котором хранится вектор. Если среди переменных рабочей среды есть массив V, определенный вектор-строкой


```
>> v = [1.3 3.6 7.4 8.2 0.9];
```

то для обращения, например, к четвертому элементу используется *индексация*:

```
>> h = v(4)
```

```
h =
```

```
8.2000
```

Указание элемента массива в левой части оператора присваивания приводит к изменению в массиве

```
>> v(2) = 555
```

```
v =
```

```
1.3000 555.0000 7.4000 8.2000 0.9000
```

Из элементов массива можно формировать новые массивы, например:

```
>> u = [v(3); v(2); v(1)]
```

```
u =
```

```
7.4000
```

```
555.0000
```

```
1.3000
```

В массив можно добавить элементы, например:

```
>> u(4) = 15
```

```
v =
```

```
7.4000
```

```
555.0000
```

```
1.3000
```

```
15.0000
```

Для помещения определенных элементов вектора в другой вектор в заданном порядке служит *индексация при помощи вектора*. Например, запись в вектор-строку *w* четвертого, второго и пятого элементов *v* производится следующим образом:

```
>> ind = [4 2 5];
```

```
>> w = v(ind)
```

```
w =
```

```
8.2000 555.0000 0.9000
```

Элементы массива могут входить в выражения:

```
>> p = w(1)*5+w(4)
```

```
p =
```

```
5.6
```

Ввод матрицы из командной строки осуществляется в квадратных скобках, элементы матриц в каждой строке разделяются пробелом или запятой, а строки разделяются точкой с запятой:

```
>> A=[1 3 5; 7 9 0]
```

```
A =
```

```
1 3 5
```

```
7 9 0
```

Другой способ ввода матриц заключается в том, что матрицу можно трактовать как вектор-строку, каждый из элементов которой является вектором столбцом:

```
>> B=[[3; 4; 6] [-1; 2; 8] [7; 0; 9]]
B =
3 -1 7
4 2 0
6 8 9
```

или как вектор-столбец, каждый элемент которого является вектор-строкой:

```
>> B=[[3 -1 7]; [4 2 0]; [6 8 9]]
```

Доступ к элементам матриц осуществляется при помощи двух индексов – номеров строки и столбца, заключенных в круглые скобки:

```
>> B(2,3)
ans =
0
```

Элементы матриц могут входить в состав выражений:

```
>> B(1,1)+B(3,1)
ans =
9
```

В качестве индексов могут выступать векторы, содержащие номера нужных строк и столбцов. Например, для выделения элементов первой и второй строк второго и третьего столбцов матрицы **B** достаточно ввести команды:

```
>> i=[1 2]; j=[2 3]; B1=B(i, j)
B1 =
-1 7
2 0
```

Используя индексы элементов, можно создавать матрицы, задавая элементы по отдельности в любом порядке. При задании элементов возможно использование арифметических операций и различных функций, например:

```
>> D(1,1)=2; D(1,2)=7.5; D(2,1)=D(1,1)+2*D(1,2); D(2,2)=sin(pi/2);
```

Существуют специальные команды для формирования массивов различных видов.

Формирование массивов, состоящих из единиц или нулей, выполняется соответственно с помощью функций `ones()` и `zeros()`, аргументами которых являются число строк и число столбцов массива:

```
>> M1 = ones(2,3)
M1 =
1 1 1
1 1 1
>> M2 = zeros(1,4)
M2 =
0 0 0 0
```

Использование одного аргумента в функциях `ones()` и `zeros()` приводит к созданию квадратных матриц.

Для создания единичной матрицы используется функция `eye()`:

```
>> M3 = eye(3)
```

```
M3 =  
    1  0  0  
    0  1  0  
    0  0  1
```

Для заполнения массивов элементами, случайно распределенными между нулем и единицей, используется функция `rand()`, аргументами которой являются число строк и число столбцов массива. При наличии одного входного аргумента формируется квадратная матрица.

Формирование диагональной матрицы, у которой все внедиагональные элементы равны нулю, применяется функция `diag()`. В качестве аргумента этой функции используется вектор-строка или вектор-столбец, элементы которого располагаются по диагонали создаваемой матрицы.

```
>> d = [1; 2; 3];  
>> M4 = diag(d)
```

```
M4 =  
    1  0  0  
    0  2  0  
    0  0  3
```

MATLAB предоставляет удобный способ обращения к блокам последовательно расположенных элементов массива. Для этого служит *индексация при помощи знака “:” (двоеточия)*. Предположим, что в заданной вектор-строке `w` из семи элементов требуется заменить нулями элементы со второго по шестой. Индексация при помощи двоеточия позволяет просто и наглядно решить поставленную задачу:

```
>> w = [0.1 2.9 3.3 5.1 2.6 7.1 9.8];
```

```
>> w(2:6) = 0;
```

```
w =  
    0.1000  0  0  0  0  0  9.8000
```

Индексация при помощи двоеточия оказывается удобной при выделении части из большого объема данных в новый массив:

```
>> w = [0.1 2.9 3.3 5.1 2.6 7.1 9.8];
```

```
>> w1 = w(3 : 5)
```

```
w1 =  
    3.3000  5.1000  2.6000
```

```
>> m = [1 2 0 2; 4 10 20 5; 2 30 40 6];
```

```
m =  
    1  2  0  2  
    4 10 20  5
```

```

    2 30 40 6
>> m1 = m(2 : 3, 2 : 3)
m1 =
    10 20
    30 40

```

Запись `w(5:end)` означает, что берутся элементы вектора `w`, начиная с пятого и заканчивая последним. Если же ввести просто `w(end)`, то получим последний элемент массива:

```

>> w(end)
ans =
    9.8000

```

Для выделения строки или столбца массива следует в качестве одного из индексов использовать номер строки или столбца, а другой индекс заменить на двоеточие без указания пределов:

```

>> m2 = m(2,:)
m2 =
    4 10 20 5

```

Для удаления строки, столбца матрицы или элемента вектора следует присвоить удаляемому элементу пустой массив `[]`. Например, для удаления первой строки матрицы `m` нужно записать:

```

>> m(1,:) = []
m =
    4 10 20 5
    2 30 40 6

```

Для удаления нескольких столбцов следует присвоить им пустой массив:

```

>> m(:,2:3)=[ ]
m =
    4 5
    2 6

```

Логическое индексирование позволяет выбрать из массивов элементы, удовлетворяющие определенным условиям, которые заданы логическим выражением. Для выделения из матрицы `m` элементов, которые больше 10, нужно записать:

```

>> f = m(m>10)
f =
    30
    20
    40

```

При этом индексы элементов остаются неизвестными. Для поиска индексов элементов, удовлетворяющих условию, используется функция `find()`:

```

>> [i,j]=find(m>10)

```

$i =$
 3
 2
 3
 $j =$
 2
 3
 3

Задания для самостоятельного выполнения

Сформировать матрицу \mathbf{a} размера $m \times n$, вектор-строку \mathbf{b} из k элементов и вектор-столбец \mathbf{c} из k_1 элементов, а также упорядоченную числовую последовательность \mathbf{d} с начальным значением d_0 , конечным значением d_k и шагом h . Добавить во все массивы по x элементов. Выделить из вектор-строки все элементы, кроме элемента номер x_1 , используя двоеточие и сцепление строк. Выделить из вектор-столбца элементы с номера x_2 по x_3 . Извлечь из матрицы первую строку и удалить последний столбец. Определить индексы положительных элементов массивов. Определить размеры и размерности полученных массивов.

Номер задания	m	n	k	k_1	d_0	d_k	h	x	x_1	x_2	x_3
1	4	2	5	10	0	20	0,5	3	3	4	10
2	3	5	8	6	-2	2	0,1	2	4	2	5
3	2	3	6	8	0,5	5,5	0,5	2	3	3	7
4	3	4	4	9	0	100	1	3	2	2	8
5	3	3	6	7	1	3	0,25	2	4	1	5
6	2	4	7	9	-5	0	0,5	1	4	3	9
7	4	3	5	6	0,5	2	0,1	2	3	2	4
8	6	6	6	7	40	30	-5	3	4	3	5
9	3	4	4	8	-8	8	0,2	1	2	2	6
10	5	2	8	9	30	10	-1	2	6	3	7
11	4	2	7	5	0	100	5	2	3	2	5
12	2	2	5	6	-1,5	1,5	0,5	1	3	3	6
13	3	2	4	8	7	0	-1	3	2	4	7
14	3	3	6	10	6	16	2	2	4	5	9
15	2	5	7	7	-5	5	2,5	1	4	3	7

1.2.2. Вычисления с применением массивов

В системе MATLAB допускаются групповые вычисления над всем массивом сразу. При этом нет необходимости выполнять каждую отдельную операцию над отдельным элементом массива. Это позволяет компактно создавать выражения, при вычислении которых реально выполняется значительный объем работы.

Начнем рассмотрение с арифметических операций. Над массивами одинакового размера допускаются операции сложения и вычитания, обозначаемые стандартными знаками “+” и “-”. Например:

```
>> A = [1 1 1; 2 2 2; 3 3 3];
>> B = [0 0 0; 7 7 7; 1 2 3];
>> C = A+B
C =
     1     1     1
     9     9     9
     4     5     6
```

Эти операции выполняются поэлементно и порождают массивы тех же размеров, что и исходные операнды. Если используются массивы разных размеров, то выдается сообщение об ошибке. Исключение составляют случаи, когда одним из операндов является скаляр (число или переменная размером 1×1), например:

```
>> C+1
ans =
     2     2     2
    10    10    10
     5     6     7
```

Для поэлементного перемножения и поэлементного деления массивов одинаковых размеров применяются операции, обозначаемые комбинациями двух символов: “.”*” (точка умножить) и “./” (точка поделить). Использование комбинации символов объясняется тем, что символами “*” и “/” в системе MATLAB обозначены специальные операции линейной алгебры над векторами и матрицами. Кроме операции “./”, называемой операцией правого поэлементного деления, есть еще операция левого поэлементного деления “.\”. Выражение A./B приводит к получению матрицы с элементами $A(k,m)/B(k,m)$, а выражение A.\B приводит к матрице с элементами $B(k,m)/A(k,m)$.

При помощи “.^” осуществляется поэлементное возведение в степень:

```
>> v1 = [2 -3 4 1];
>> p = v1.^2
p =
     4     9    16     1
```

Показателем степени может быть вектор той же длины, что и возводимый в степень. При этом каждый элемент первого вектора возводится в степень, равную соответствующему элементу второго вектора:

```
>> v2 = [3 2 2 9];
>> p1 = v1.^v2
p1 =
    8.0000  9.0000 16.0000  1.0000
```

Кроме указанных операций над массивами, можно производить вычисления с использованием математических функций. Например:

```
>> x = 0:0.01:2*pi;
>> y = sin(x);
```

В данном случае вычисляется массив y значений функции $\sin(x)$ для значений x , лежащих в пределах от 0 до 2π с шагом 0,01. При этом выполняется большой объем вычислений, так как значений x много, и для каждого x необходимо вычислить y .

Знак “*” закреплен за перемножением матриц и векторов в смысле линейной алгебры. Эта операция выполнима лишь только тогда, когда число столбцов в левом операнде равно числу строк в правом операнде:

```
>> A1 = [1 2 3; 4 5 6; 7 8 9]
```

```
A1 =
     1     2     3
     4     5     6
     7     8     9
```

```
>> v = [1; 2; 3]
```

```
v =
     1
     2
     3
```

```
>> B1 = A1*v
```

```
B1 =
    14
    32
    50
```

С помощью операции “\” (левого деления) можно решать системы линейных уравнений. Например, пусть задана система линейных уравнений

$$\begin{cases} 2x_1 + 3x_2 = 18; \\ 5x_1 + 6x_2 = 39. \end{cases}$$

Эту систему можно переписать в матричном виде, т. е.

$$A x = b,$$

где

$$\mathbf{A} = \begin{bmatrix} 2 & 3 \\ 5 & 6 \end{bmatrix}, \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} 18 \\ 39 \end{bmatrix}.$$

Для нахождения \mathbf{x} в системе MATLAB достаточно написать:

```
>> A = [2 3; 5 6]; b = [18;39];
>> x = A \ b
x =
```

```
3
4
```

Для замены строк матрицы столбцами используется операция транспонирования, например:

```
>> D = A1'
```

```
D =
    1    4    7
    2    5    8
    3    6    9
```

Перемножение элементов векторов, а также элементов столбцов матриц осуществляется при помощи функции `prod()`:

```
>> z = [3; 2; 1; 4; 6; 5];
```

```
>> p = prod(z)
```

```
p =
    720
```

Функция `sum()` предназначена для суммирования элементов вектора или элементов столбцов матриц.

```
>> z1 = [3 2 1; 4 6 5];
```

```
>> s = sum(z)
```

```
s =
    7    8    6
```

Для вычисления среднего арифметического элементов вектора используется функция `mean()`.

```
>> mean(z)
```

```
ans =
    3.5000
```

Для нахождения минимума и максимума из элементов вектора или элементов столбцов матриц служат встроенные функции `min()` и `max()`:

```
>> m = min(z)
```

```
m =
    1
```

```
>> M = max(z)
```

```
M =
    6
```

Чтобы определить максимальный или минимальный элемент матрицы, нужно использовать соответствующие функции дважды:


```
>> M1 = max(max(D))
M1 =
     9
```

```
>> m1 = min(min(D))
m1 =
     1
```

или

```
>> M1 = max(D(:))
M1 =
     9
```

```
>> m1 = min(D(:))
m1 =
     1
```

При обращении к функции `min()` с двумя векторами в качестве входных аргументов получится вектор, каждый элемент которого есть минимум из двух элементов исходных векторов с одинаковыми номерами.

```
>> p = [3 12 8];
>> s = [4 10 7];
>> min(p, s)
ans =
     3 10 7
```

Часто необходимо знать не только значение минимального или максимального элемента в массиве, но и его индекс (порядковый номер). Для этого необходимо вызвать функцию `min()` с двумя выходными аргументами:

```
>> [m, k] = min(z)
m =
     1
k =
     3
```

В результате переменной `m` будет присвоено значение минимального элемента массива `z`, а номер минимального элемента занесен в переменную `k`.

В число основных функций для работы с массивами входит функция упорядочения элементов вектора или элементов столбцов матрицы по возрастанию `sort()`.

```
>> r = [9.4 -2.3 -5.2 7.1 0.8 1.3];
>> R = sort(r)
R =
    -5.2000    -2.3000     0.8000     1.3000     7.1000     9.4000
```

Можно упорядочить элементы по убыванию, используя функцию `sort()`:

```
>> R1 = -sort(-r)
R1 =
     9.4000     7.1000     1.3000     0.8000    -2.3000    -5.2000
```

Упорядочение элементов в порядке возрастания их модулей производится с привлечением функции `abs()`:

```
>> R2 = sort(abs(r))
R2 =
     0.8000     1.3000     2.3000     5.2000     7.1000     9.4000
```

Вызов `sort()` с двумя выходными аргументами приводит к образованию массива индексов соответствия элементов упорядоченного и исходного массивов:

```
>> [rs, ind] = sort(r)
rs =
    -5.2000  -2.3000  0.8000  1.3000  7.1000  9.4000
ind = 3  2  5  6  4  1
```

Задания для самостоятельного выполнения

1. Ввести вектор-строку **a**, вектор-столбец **b** и матрицу **M**. Вычислить сумму и произведение элементов каждого массива. Найти минимальные и максимальные элементы массивов. Упорядочить вектор-строку по возрастанию элементов, а вектор-столбец – по убыванию.

2. Ввести вектор-строки **v₁** и **v₂**, для которых нужно выполнить поэлементное сложение, вычитание, умножение и деление. Ввести матрицы **M₁** и **M₂** для выполнения их перемножения.

3. Определить значения функции *y* для значений аргумента *x*, распределенных в пределах от *x₀* до *x_k* с шагом *h*.

Номер задания	<i>y</i>	<i>x₀</i>	<i>x_k</i>	<i>h</i>	Номер задания	<i>y</i>	<i>x₀</i>	<i>x_k</i>	<i>h</i>
1	$x \sin x$	-2π	2π	0,01	9	$x e^x$	0	3	0,5
2	$2x\sqrt{x}$	0	20	1	10	$\frac{\lg x}{x^2}$	10	20	1
3	$\frac{\operatorname{tg} x}{2x}$	π	2π	0,01	11	$\frac{2 \ln x}{x}$	2	8	0,2
4	$\frac{\lg x}{x}$	10	100	10	12	$\frac{\operatorname{ctg} x}{x}$	π	2π	0,05
5	$\frac{2x+3}{x}$	0	10	0,5	13	$\frac{x+3}{x^2}$	-2	2	0,02
6	$\frac{\sin x}{x}$	0	$\pi/2$	0,01	14	$3x^3\sqrt{x}$	0	3	0,1
7	$3x \cos x$	$\pi/2$	π	0,01	15	$5x^2$	2	19	1
8	$2x \cos x$	0	π	0,02					

4. Решить систему линейных алгебраических уравнений.

Система уравнений	Система уравнений
1. $\begin{cases} 5x + z = 11 \\ x + 3y - z = 4 \\ -3x + 2y + 10z = 6 \end{cases}$	9. $\begin{cases} 4x + y - z = 7 \\ 2x + 3y = 7 \\ x - y + 5z = 11 \end{cases}$
2. $\begin{cases} 2x - z = -3 \\ -x + 3y + z = 2 \\ x - y + 4z = 3 \end{cases}$	10. $\begin{cases} 2x - z = 1 \\ x - 4y + 2z = -5 \\ x + y + 3z = 6 \end{cases}$
3. $\begin{cases} 2x - z = 1 \\ x - 3y + z = 2 \\ x + y + 3z = 4 \end{cases}$	11. $\begin{cases} 3x - 2y + z = 0 \\ 2y - z = 0 \\ -x + y + 5z = 23 \end{cases}$
4. $\begin{cases} 5x + y - z = -5 \\ -x + 3y + z = 5 \\ x - 2y + 4z = 1 \end{cases}$	12. $\begin{cases} 5x + z = 16 \\ x + 3y - z = 8 \\ -3x + 2y + 10z = 5 \end{cases}$
5. $\begin{cases} 3x + y - z = -1 \\ -2x + 4y + z = 5 \\ x + y + 3z = -3 \end{cases}$	13. $\begin{cases} 2x - z = 0 \\ x - 3y + z = -3 \\ x + y + 3z = 41 \end{cases}$
6. $\begin{cases} 3x + y - z = 6 \\ 2x + 4y + z = 9 \\ x - y + 3z = 4 \end{cases}$	14. $\begin{cases} 2x - z = 10 \\ -x + 3y + z = 15 \\ x - y + 4z = 17 \end{cases}$
7. $\begin{cases} 2x - y = -2 \\ 2x + 5y - 2z = -4 \\ x - y + 3z = 2 \end{cases}$	15. $\begin{cases} 4x + y - z = 16 \\ 2x + 3y = 11 \\ x - y + 5z = 8 \end{cases}$
8. $\begin{cases} 3x - y + z = 1 \\ 2y - z = 3 \\ -x + y + 5z = -5 \end{cases}$	

5. Вычислить значение функции $f(x)$ на отрезке $x \in [a; b]$ с шагом h .

$f(x)$	a	b	h
1. $\frac{x^2}{1+0,25\sqrt{x}}$	1,1	3,1	0,2
2. $\frac{x^3 - 0,3x}{\sqrt{1+2x}}$	2,05	3,05	0,1
3. $\frac{2e^{-x}}{2\pi + x^3}$	0	1,6	0,16
4. $\frac{\cos \pi x^2}{\sqrt{1-3x}}$	-1	0	0,1
5. $\sqrt{1+4x} \sin \pi x$	0,1	0,8	0,07
6. $\frac{e^{\frac{x}{3}}}{1+x^2}$	1,4	2,4	0,1
7. $e^{-2x} + x^2 - 1$	0,25	2,25	0,2
8. $(e+x) \sin(\pi\sqrt{x-1})$	1,8	2,8	0,1
9. $\sqrt{3+2x} \cdot \operatorname{tg} \frac{\pi x^3}{2}$	0,1	0,9	0,08
10. $\sqrt{2+3x} \cdot \ln(1+3x^2)$	-0,1	0,9	0,1
11. $\sqrt[3]{x^2+3} \cdot \cos \frac{\pi x}{2}$	1	2,5	0,15
12. $(4+7x) \sin(\pi\sqrt[3]{1+x})$	0	7	0,7
13. $e^{-x^2} (1+3x-x^2)$	0	2	0,2
14. $x^3 - 3x + \frac{8}{\sqrt{1+x^2}}$	0	1,7	0,17
15. $\sqrt{\operatorname{sh} \sqrt{2\pi x}}, \left(\operatorname{sh} x = \frac{e^x - e^{-x}}{2} \right)$	0	1,2	0,12

6. Индивидуальное задание

Матрица	Матрица
<p>1.</p> $A = \begin{vmatrix} -4 & -11 & 3 & -5 & 10 & 2 \\ -16 & 11 & 1 & 21 & 0 & -13 \\ 1 & 11 & -1 & -1 & 0 & 7 \\ 2 & 0 & 7 & 1 & -8 & 16 \end{vmatrix}$ <p>Найдите сумму наибольших элементов ее строк.</p>	<p>5.</p> $A = \begin{vmatrix} 13 & -12 & 14 & -8 & -5 & -6 \\ 2 & 0 & -5 & -12 & -1 & -13 \\ 14 & -3 & -8 & -1 & -4 & 4 \\ 11 & -8 & -2 & 0 & 0 & -9 \\ -6 & 4 & 6 & 4 & 15 & 0 \end{vmatrix}$ <p>Найдите вектор, элементами которого являются наибольшие элементы строк матрицы.</p>
<p>2.</p> $A = \begin{vmatrix} -6 & 5 & 12 & -1 & -8 \\ 8 & -3 & 6 & -16 & 5 \\ 12 & 6 & 11 & 2 & 2 \\ -15 & 8 & -12 & -10 & -9 \\ -14 & 7 & 0 & 14 & 21 \end{vmatrix}$ <p>Найдите сумму элементов строки, в которой размещен элемент с наименьшим значением.</p>	<p>6.</p> $A = \begin{vmatrix} -6 & 1 & -6 & 0 & 21 & -13 \\ 5 & 15 & -10 & -3 & -13 & 3 \\ 5 & 10 & -12 & -4 & -10 & 15 \\ -2 & -15 & 2 & 3 & 10 & 7 \\ -20 & 0 & -4 & 7 & -3 & 19 \end{vmatrix}$ <p>Постройте вектор, элементами которого являются суммы наибольшего и наименьшего элементов каждой строки матрицы.</p>
<p>3.</p> $A = \begin{vmatrix} 0 & -6 & -3 & 5 & -2 & -3 & 2 & 0 & 5 \\ -10 & 18 & 10 & 0 & -14 & 6 & -10 & -6 & -8 \\ 6 & -10 & -18 & 6 & -2 & 7 & -7 & -5 & -2 \\ 5 & 4 & 5 & 1 & 9 & 10 & 4 & -11 & 0 \\ 16 & 0 & 8 & -2 & 3 & -9 & -1 & -9 & -22 \\ 5 & 0 & 7 & -3 & 14 & 2 & 3 & 7 & 9 \end{vmatrix}$ <p>Поменяйте местами строку с наибольшим элементом и строку с наименьшим элементом.</p>	<p>7.</p> $A = \begin{vmatrix} 50 & 119 & -109 & -32 & 105 & -64 \\ 186 & -11 & 8 & 123 & -11 & -70 \\ -33 & 63 & -200 & -63 & 37 & -101 \\ -113 & -60 & -49 & -232 & 94 & -18 \\ -21 & 55 & 46 & -123 & -212 & 152 \end{vmatrix}$ <p>Постройте вектор, элементами которого являются средние значения элементов соответствующей строки матрицы.</p>
<p>4.</p> $A = \begin{vmatrix} -51 & -94 & 5 & -27 & -46 & 1 \\ 32 & -37 & -121 & 95 & -26 & -64 \\ 28 & -118 & -4 & 12 & -121 & 80 \\ 2 & -105 & -112 & 65 & -131 & 23 \\ -100 & 147 & -134 & -116 & 93 & -98 \end{vmatrix}$ <p>В каждой строке выберите элемент с наименьшим значением, из полученных чисел выберите наибольшее. Найдите индексы полученных элементов.</p>	<p>8.</p> $A = \begin{vmatrix} 0 & -54 & -25 & 62 & -116 & -65 \\ 7 & 26 & -140 & 126 & 118 & 31 \\ 35 & -1 & 177 & -89 & -1 & 10 \\ 49 & -58 & 32 & 13 & 53 & 184 \\ 127 & 213 & -111 & -13 & -71 & -27 \end{vmatrix}$ <p>Постройте вектор, элементами которого являются средние арифметические наибольшего и наименьшего элементов соответствующей строки матрицы.</p>

Матрица						Матрица										
9.	$A =$	221	0	27	63	47	13.	$A =$	-42	74	-7					
		150	83	105	131	11			22	-89	-35					
		-194	-72	62	32	-59			-95	13	-207					
		-168	-72	-175	-67	-65			-14	-23	-14					
		-57	-20	69	-14	-108										
		-18	-2	81	-244	-4										
		Постройте вектор, элементами которого являются суммы квадратов элементов соответствующего столбца матрицы.								Найдите сумму всех элементов матрицы.						
10.	$A =$	37	-95	-11	-200	-90	14.	$A =$	-3	58	88	5	127	68		
		-33	129	80	108	-41			122	-25	230	-110	186	-236		
		-49	44	4	-98	-50			-69	48	52	48	-52	99		
		-3	128	-75	-68	161			0	66	-1	0	10	21		
		-17	-49	-8	133	8			-78	-7	91	-27	-80	26		
		Постройте векторы, элементами которых являются суммы элементов столбцов матрицы, произведения элементов столбцов и наименьшие значения элементов столбцов.								Удалите в матрице строку, содержащую наибольший элемент.						
11.	$A =$	-108	-125	-97	-26	-122	15.	$A =$	-230	-40	-83	-134				
		-112	-21	-44	70	-66			178	-153	-20	148				
		173	-19	108	-48	134			39	22	75	3				
		193	30	237	186	38			2	-137	37	187				
		163	-57	22	110	39										
		Найдите среднее арифметическое наибольших и наименьших ее элементов.														
12.	$A =$	-18	17	67	-104	127	$B =$	-120	-97	25	16					
		-105	-54	-50	153	63			-78	-96	-18	-50				
		-7	163	85	43	138			-76	-237	-16	-70				
		27	82	26	-191	131			-10	-83	-11	50				
		137	23	62	46	-90										
		Постройте вектор, элементами которого являются элементы главной диагонали матрицы.								Даны две матрицы. Постройте новую матрицу размером (4×8) , включая в первые 4 столбца строки первой матрицы, а в другие – столбцы второй матрицы.						

1.3. Построение графиков в среде MATLAB (краткие сведения)

В результате вычислений в системе MATLAB обычно получается большой массив данных, который трудно анализировать без наглядной визуализации. Графические возможности системы MATLAB являются мощными и разнообразными. В первую очередь целесообразно изучить наиболее простые в использовании возможности. Их часто называют *высокоуровневой графикой*. Это название отражает тот факт, что пользователю нет необходимости вникать во все детали работы с графикой.

Например, чтобы построить график функции одной вещественной переменной $y = \sin(x)$, $x \in [0, 4\pi]$, необходимо:

- 1) задать вектор значений аргумента x ;
- 2) вычислить вектор y значений функции $\sin(x)$;
- 3) использовать команду `plot()` для построения графика.

Таким образом, для построения графика функции $y = \sin(x)$ необходимо ввести в командном окне следующие выражения:

```
>> x=0:0.01:4*pi;  
>> y=sin(x);  
>> plot(x,y)
```

Команды для задания вектора x и вычисления функции y лучше завершать точкой с запятой для подавления вывода в командное окно значений x и y . После команды `plot()` точку с запятой ставить не нужно, так как эта команда ничего не выводит в командное окно.

MATLAB показывает графические объекты в специальных *графических окнах*, имеющих в заголовке слово **Figure** (фигура, изображение). После выполнения команды на экране появится графическое окно с заголовком **Figure 1**, содержащее график функции (рис. 1.5).

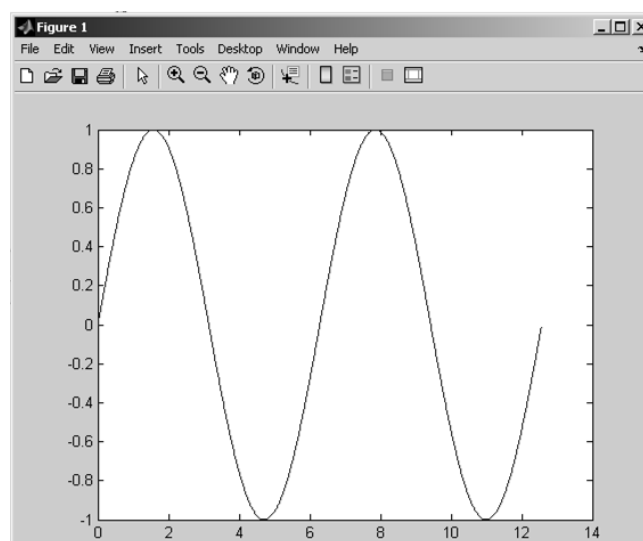


Рис. 1.5. График функции $y = \sin(x)$

При построении графиков функций очень большую часть работы MATLAB берет на себя. После введения в командной строке одной функции `plot()` система сама создает графическое окно, строит оси координат, вычисляет диапазоны изменения переменных x и y , ставит на осях метки и соответствующие им числовые значения, проводит через опорные точки график функции некоторым выбранным по умолчанию цветом, в заголовке графического окна надписывает номер графика в текущем сеансе работы.

В общем случае функция

`plot(x1, y1, 's1', x2, y2, 's2', ...)`

позволяет объединить в одном графическом окне несколько графиков функций $y_1(x_1)$, $y_2(x_2)$,... проведя их со стилями s_1 , s_2 ,... и т. д.

Стили s_1 , s_2 , ... задаются в виде набора *символов*, заключенных в одиночные кавычки (апострофы). Один из этих символов задает цвет, а другой — тип линии или маркера. В *табл. 1.1* приведены возможные цвета и типы линий и маркеров с указанием результата.

Т а б л и ц а 1.1. Свойства линий и маркеров

Цвет		Тип маркера		Тип линии	
y	Желтый	.	Точка	-	Сплошная
m	Розовый	o	Кружок	:	Пунктирная
c	Голубой	x	Крестик	-.	Штрих-пунктирная
r	Красный	+	Знак «плюс»	- -	Штриховая
g	Зеленый	*	Звездочка		
b	Синий	s	Квадрат		
w	Белый	d	Ромб		
k	Черный	v	Треугольник вершиной вниз		
		^	Треугольник вершиной вверх		
		<	Треугольник вершиной влево		
		>	Треугольник вершиной вправо		
		p	Пятиконечная звезда		
		h	Шестиконечная звезда		

Например, для построения графика $y = \sin(x)$ красной штриховой линией, а графика $z = \cos(x)$ — маркерами в виде звездочек синего цвета на месте вычисляемых точек (*рис. 1.6*) следует использовать команды:

```
>> x=0:0.1:4*pi;
>> y=sin(x);z=cos(x);
>> plot(x,y,'r- ',x,z,'b*')
```

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

Сетка наносится командой `grid on` (или просто `grid`), а удаляется командой `grid off` (или повторением команды `grid`).

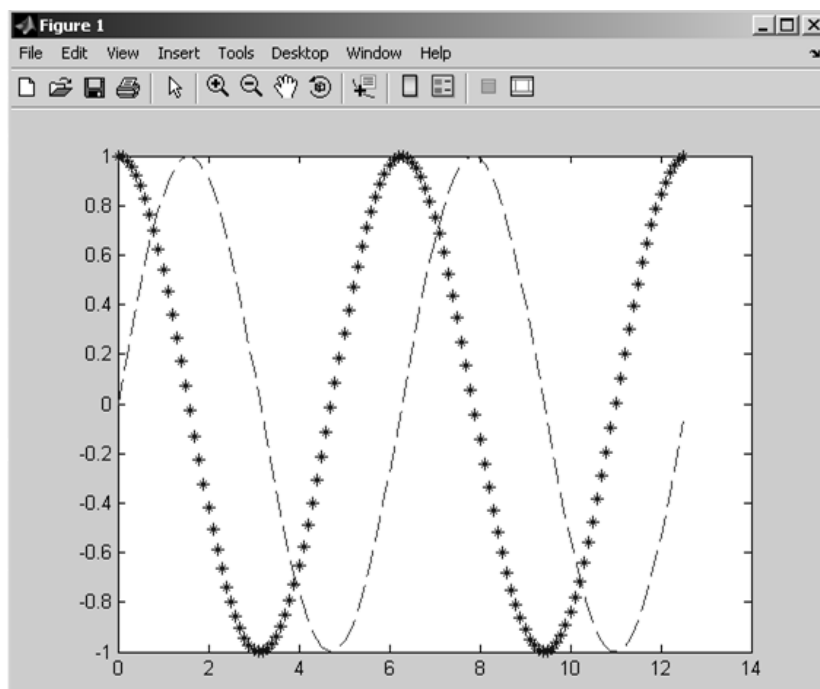


Рис. 1.6. Изменение параметров линий

Для размещения подписей к осям служат функции `xlabel()` и `ylabel()`. Функция `xlabel('название оси')` предназначена для проставления названия горизонтальной оси, функция `ylabel('название оси')` — то же для вертикальной оси (причем эти надписи ориентированы вдоль осей координат).

Общий заголовок для графика проставляется функцией `title('заголовок')`.

Если требуется разместить какую-либо надпись в произвольном месте рисунка, применяем функцию `text()`:

```
text(a,b,'надпись')
```

Надпись, задаваемая функцией `text()`, помещается начиная от точки с координатами, указанными первыми двумя аргументами `a` и `b`. По умолчанию координаты задаются в тех же единицах измерения, что и координаты, указанные на горизонтальной и вертикальной осях.

Для размещения легенды на графике используется команда:

```
legend('график 1','график2',...)
```

При размещении легенды следует учесть, что порядок и количество аргументов команды `legend()` должны соответствовать линиям на графике.

Чтобы применить все эти средства для оформления графиков, представленных на *рис. 1.6*, запишем:

```
>> title('Графики функций y=sin x и z=cos x');  
>> xlabel('x')  
>> ylabel('sin x, cos x')
```

```
>> legend('sin x','cos x')
```

```
>> grid on;
```

Результаты оформления графиков представлены на *рис. 1.7*.

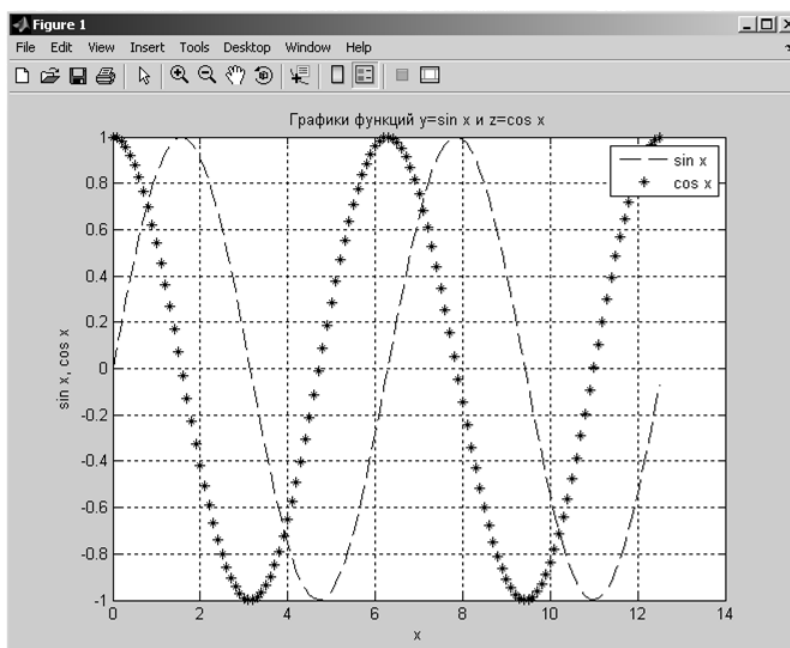


Рис. 1.7. Оформление графического окна

Система **MATLAB** устанавливает пределы на горизонтальной оси равными тем значениям, что указаны пользователем для независимой переменной. Для зависимой переменной по вертикальной оси **MATLAB** самостоятельно вычисляет диапазон изменения значений функции. В результате график функции оказывается как бы вписанным в прямоугольник.

Если необходимо отказаться от этой особенности масштабирования при построении графиков, то можно явным образом задать свои пределы изменения переменных по осям координат. Это делается с помощью функции `axis([xmin, xmax, ymin, ymax])`,

причем команду на выполнение этой функции можно вводить с клавиатуры сколько угодно раз уже после построения графика функции, чтобы, глядя на получающиеся визуальные изображения, добиться наилучшего восприятия. Такое масштабирование позволяет получить подробные изображения тех частей графика, которые вызывают наибольший интерес в конкретном исследовании.

1.3.1. Вывод нескольких графиков

Для работы с несколькими графиками **MATLAB** предоставляет следующие возможности:

- 1) вывод нескольких графиков в одном графическом окне и в одних координатных осях;
- 2) вывод каждого графика в отдельное окно;

3) отображение в пределах одного окна нескольких графиков, каждого – на своих координатных осях.

1.3.2. Вывод нескольких графиков в одних координатных осях

Если нужно второй график провести «поверх первого графика», то перед вторичным вызовом графической функции `plot()` нужно выполнить команду `hold on`, которая предназначена для удержания текущего графического окна. В результате будет получено следующее изображение (рис. 1.8).

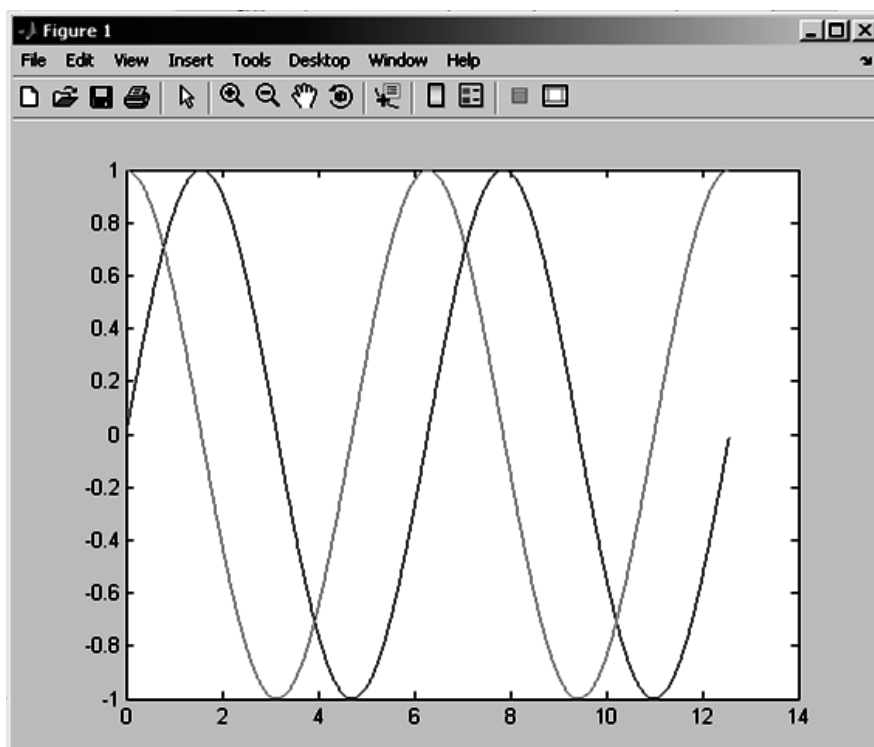


Рис. 1.8. Графики функций $y = \sin(x)$ и $z = \cos(x)$ в одном графическом окне

Того же результата можно добиться, потребовав от функции `plot()` построить сразу несколько графиков в рамках одних и тех же осей координат:

```
>> x=0:0.01:4*pi;  
>> y=sin(x);z=cos(x);  
>> plot(x,y,x,z)
```

При этом разные графики автоматически строятся разным цветом.

Можно сократить запись команд следующим образом:

```
>> x=0:0.01:4*pi; plot(x, sin(x),x, cos(x))
```

К недостаткам указанных способов построения нескольких графиков в пределах одних и тех же осей координат относится использование одного и того же диапазона изменения координат, что при несопоставимых значениях двух функций приведет к плохому изображению графика одной из них.

1.3.3. Вывод графиков в отдельные окна

Для создания пустого графического окна и отображения его на экране служит команда `figure`. Окно становится текущим, то есть все последующие графические функции будут осуществлять построение графиков в этом окне. Для получения нового графического окна следует снова использовать команду `figure`. Например, построив графики функций $\sin(\)$ и $\cos(\)$ в пределах одного графического окна (показано выше), вычисляем массив значений w для функции x^2 :

```
>> w=x.^2;
```

После этого выполняем команды:

```
>> figure; plot(x,w)
```

которые строят график функции x^2 в новом графическом окне, так как команда `figure` создает это новое графическое окно и заставляет все последующие за ней функции построения графиков выводить их туда (*рис. 1.9*). В результате в первом графическом окне (**Figure 1**) по вертикальной оси переменные изменяются в диапазоне от -1 до 1 , а во втором графическом окне (**Figure 2**) — от 0 до 160 .

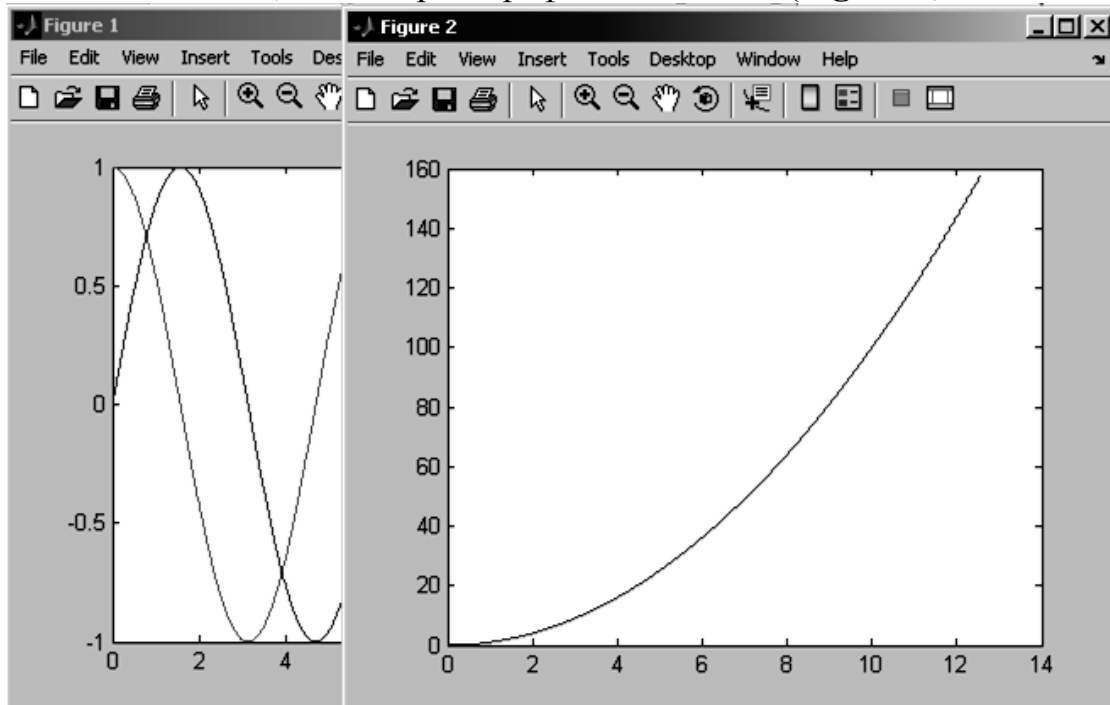


Рис. 1.9. Графики функций $y = \sin(x)$, $z = \cos(x)$ и $w = x^2$ в разных графических окнах

К графическому окну можно обращаться по его номеру n с указанием этого номера в команде `figure(n)`. Такая возможность используется, если, например, необходимо добавить какой-либо график в одно из ранее созданных графических окон:

```
>> figure(1);hold on; plot(x,2*x)
```

1.3.4 Вывод графиков в разных координатных осях одного графического окна

MATLAB позволяет разместить в графическом окне несколько осей и вывести на них различные графики. Для этого окно разбивается на несколько областей построения графиков по вертикали и горизонтали с использованием функции `subplot(m,n,p)`, которая располагает области построения в виде матриц и используется с тремя параметрами. Первый параметр m равен числу горизонтальных строк областей построения, второй n равен числу вертикальных столбцов областей построения, а третий аргумент p обозначает номер области построения, в которой надо строить текущий график (номер отсчитывается от левого верхнего угла построчно). Эта функция позволяет разбить графическое окно на несколько подобластей, в каждую из которых можно вывести графики различных функций.

Например, для ранее выполненных вычислений с функциями `sin()`, `cos()` и x^2 строим графики первых двух функций в первой подобласти, а график третьей функции – во второй подобласти одного и того же графического окна:

```
>> subplot(1,2,1); plot(x,y,x,z);  
>> subplot(1,2,2); plot(x,w)
```

В результате этого получаем графическое окно следующего вида (рис. 1.10).

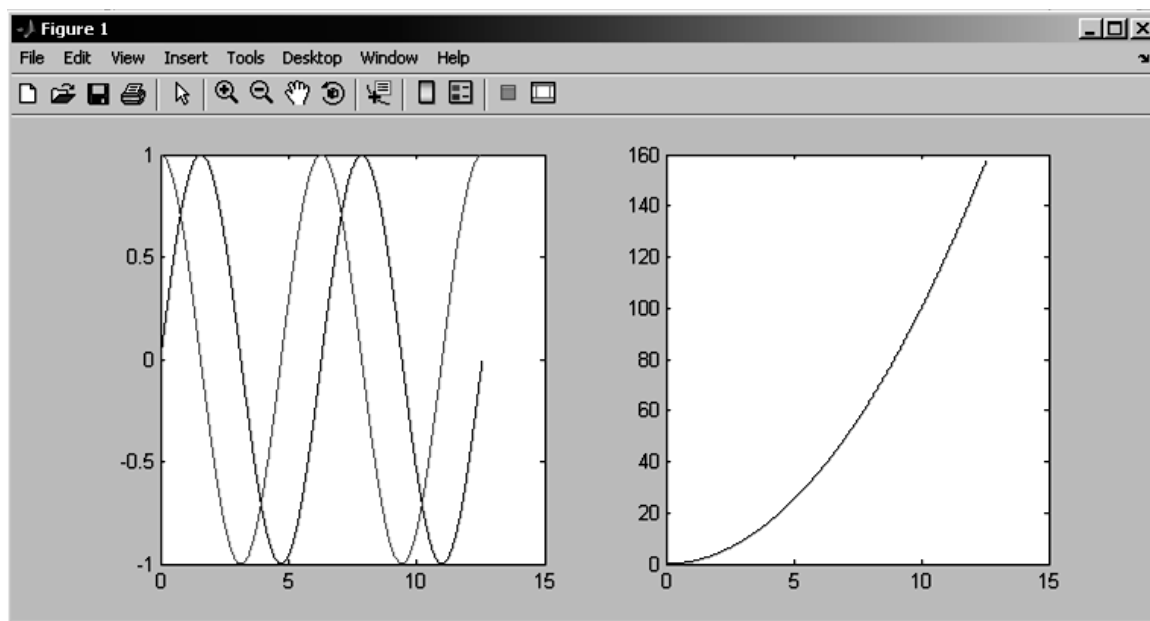


Рис. 1.10. Графики функций $y = \sin(x)$, $z = \cos(x)$ и $w = x^2$ в разных подобластях одного графического окна

Диапазоны изменения переменных на осях координат этих подобластей независимы друг от друга.

Задания для самостоятельного выполнения

1. Построить графики функций на заданном интервале изменения значений аргумента, расположив их в одном окне на одних осях. Установить для графиков разные цвета и типы линий или маркеров, подписи к осям, заголовок, нанести сетку и легенду.

Номер задания	Функции
1	$f = \sin^3 x, f_1 = \cos^2 x, x \in [-2\pi; 2\pi]$
2	$f = \cos x + \sin x , f_1 = 2 \sin x , x \in [-2\pi; 2\pi]$
3	$f = \sin 3x, f_1 = \cos(x+5), x \in [-2\pi; 2\pi]$
4	$f = x^3, f_1 = \cos 5x, x \in [-2\pi; 2\pi]$
5	$f = \sin^2 x, f_1 = \cos^3 x, x \in [-2\pi; 2\pi]$
6	$f = \cos 2x, f_1 = \cos x , x \in [-2\pi; 2\pi]$
7	$f = 2\sin 1,5x, f_1 = \cos(x-2), x \in [-2\pi; 2\pi]$
8	$f = 3x^2, f_1 = \cos 4x, x \in [-2\pi; 2\pi]$
9	$f = 2\sin^3 x, f_1 = \cos^2 2x, x \in [-2\pi; 2\pi]$
10	$f = 2\sin 2x, f_1 = \cos 2x , x \in [-2\pi; 2\pi]$
11	$f = \frac{\sin x}{x}, f_1 = e^{-x} \cos x, x \in [0,01; 2\pi]$
12	$f = e^{-0,1x} \sin^2 x, f_1 = e^{-0,2x} \cos^2 x, x \in [-2\pi; 2\pi]$
13	$f = \sin(2\pi x^2), f_1 = \cos(2\pi x^2), x \in [-1; 1]$
14	$f = 1 - 0,5\sin 2x, f_1 = 1 - 2\sin 2x, x \in [0; \pi]$
15	$f = \sin x + \sin x , f_1 = \cos 2x , x \in [-2\pi; 2\pi]$

2. Построить графики функций на заданном интервале изменения значений аргумента, расположив их в одном окне на отдельных осях. Установить для графиков цвет и тип линий или маркеров, подписи к осям, заголовки, нанести сетку.

Номер задания	Функции
1	$y = \sin x , y_1 = \operatorname{tg}(\cos x), x \in [-2\pi; 2\pi]$
2	$y = x^2 + 3, x \in [-2\pi; 2\pi]; y_1 = \frac{\ln(x_1 + \sqrt{x_1^2 + 1})}{\sqrt{x_1^2 + 1}}, x_1 \in [0; 1]$
3	$y = \cos 3x, y_1 = x^4 + 8, y_2 = x - 5, y_3 = \sin 3x, x \in [-2\pi; 2\pi]$
4	$y = x^2 + 3, y_1 = \sin 2x, x \in [-2\pi; 2\pi]$
5	$y_1 = \sin x , x \in [-2\pi; 2\pi]; y_2 = \sqrt{2x_1 + 1}, x_1 \in [0; 1]$
6	$y = x^2 + 3, x \in [-2\pi; 2\pi]; y_1 = \frac{x_1}{1 - \ln x_1}, x_1 \in [1; 2]$
7	$y = \sin 3x, y_1 = \cos 3x, y_2 = x^4 + 8, y_3 = x - 5, x \in [-2\pi; 2\pi]$
8	$y = (x - \pi/2)\sin x, x \in [\pi/2; \pi/2 + 1]; y_1 = \sin 3x_1, x_1 \in [-2\pi; 2\pi]$
9	$y = \cos x , x \in [-2\pi; 2\pi]; y_1 = \sin x_1 + e^{-\sin x_1} - 1, x_1 \in [0; 1]$
10	$y = x^2 + 3x, x \in [-2\pi; 2\pi]; y_1 = \frac{2x_1}{1 - \ln x_1}, x_1 \in [1; 2]$
11	$y = 2 + \cos\left(x + \frac{\pi}{3}\right), x \in [-\pi; \pi]; y_1 = \operatorname{tg} x_1 , x_1 \in [-2\pi; 2\pi]$
12	$y = x^3 - 3x^2, x \in [-4; 4]; y_1 = \frac{x_1^4}{4} - \frac{x_1^3}{3} - x_1^2, x_1 \in [-3; 3]$
13	$y = 3x^2 - x^3, x \in [-4; 4]; y_1 = \sin x_1 - 0,5\sin 2x_1, x_1 \in [-3; 3]$
14	$y = \frac{2x}{1 - x^2}, x \in [-3; 3]; y_1 = \lg 3x_1, x_1 \in [0; 5]; y_2 = 3^{ x_2 - 1 }, x_2 \in [-2; 3]$
15	$y = \frac{6(x - 1)}{x^2 + 3}, x \in [-5; 5]; y_1 = \lg(-3x_1), x_1 \in [-5; 0]; y_2 = 2^{ x_2 }, x_2 \in [-4; 4]$

3. Построить графики функций на заданном интервале изменения значений аргумента, расположив их в разных графических окнах. Установить для графиков цвет и тип линий или маркеров, подписи к осям, заголовок, нанести сетку.

Номер задания	Функции
1	$y = \operatorname{arctg}\left(\operatorname{tg}\frac{x}{2}\right), y_1 = \cos 3x, x \in [-2\pi; 2\pi]$
2	$y = 0,85\operatorname{sign}(\sin x), y_1 = x^3, x \in [-2\pi; 2\pi]$
3	$y = \cos^5 x, y_1 = \sin^4 x, x \in [-2\pi; 2\pi]$
4	$y = 0,85\operatorname{sign}(\sin 2x), y_1 = x^3, x \in [-2\pi; 2\pi]$
5	$y = \operatorname{tg}\frac{x}{2}, x \in [-2\pi; 2\pi]; y_1 = e^{\frac{x_1^2}{2}}, x_1 \in [0; 1]$
6	$y = 0,85\operatorname{sign}(\sin 3x), x \in [-2\pi; 2\pi]; y_1 = \frac{1}{2}x_1^2 e^{-x_1^2}, x_1 \in [0; 1]$
7	$y = \cos^5 2x, y_1 = \sin^4 2x, x \in [-2\pi; 2\pi]$
8	$y = 0,85\operatorname{sign}\left(\sin\frac{x}{2}\right), x \in [-2\pi; 2\pi]; y_1 = 2x_1 \operatorname{arctg} x_1, x_1 \in [1; 2]$
9	$y = \operatorname{arctg}\left(\operatorname{tg}\frac{x}{2}\right), x \in [-2\pi; 2\pi]; y_1 = (1 - 2\cos x_1)\operatorname{tg} x_1, x_1 \in [0; 1]$
10	$y = 2(\sqrt{2})^x, x \in [-3; 3]; y_1 = \left \frac{2x_1 - 3}{x_1 - 2}\right , x_1 \in [-2; 5];$
11	$y = x^2 + 3x, x \in [-2\pi; 2\pi]; y_1 = \frac{2x_1}{1 - \ln x_1}, x_1 \in [1; 2]$
12	$y = 2^{\sqrt{-\sin^2 x}}, x \in [-3; 3]; y_1 = \frac{4x_1 + 1}{2x_1 - 3}, x_1 \in [-5; 5];$
13	$y = (\sin^2 x + \cos^2 x)^x, x \in [-2; 2]; y_1 = \frac{1}{ x_1 - 2 }, x_1 \in [-3; 5]$
14	$y = \frac{5x + 4}{x}, x \in [-5; 5]; y_1 = \sin x_1 , x_1 \in [-2\pi; 2\pi]$
15	$y = x^2 + 3, x \in [-4; 4]; y_1 = \sin 3x_1, x_1 \in [-3\pi; 3\pi]$

1.4. Построение и оформление графиков функций двух переменных

1.4.1. Построение трехмерных графиков функций

Построение графика функции двух переменных в MATLAB на прямоугольной области определения переменных включает следующие этапы:

1) разбиение области определения прямоугольной сеткой с использованием функции `meshgrid()`;

2) вычисление функции в узлах сетки и запись полученных значений в матрицу;

3) использование одной из графических функций системы MATLAB (`plot3()`, `mesh()`, `meshc()`, `meshz()`, `surf()`, `surfl()`, `surfc()`, `contour3()` или `waterfall()`);

Например, чтобы построить график функции двух переменных $z = e^{-x^2-y^2}$ на прямоугольной области определения $x \in [-2; 2]$, $y \in [-1; 1]$, необходимо подготовить матрицы **X** и **Y** с координатами узлов сетки, а также матрицу со значениями функции **Z**:

```
>> [x, y]=meshgrid (-2:0.1:2, -1:0.1:1);
```

```
>> z=exp(-x.^2-y.^2);
```

Затем, используя, например, функцию `plot3()`, построить график:

```
>> plot3(x,y,z)
```

Получаем изображение трехмерного графика функции $z(x,y)$, представленное на *рис. 1.11*, из которого видно, что функция `plot3()` строит график в виде набора линий в пространстве, каждая из которых является сечением трехмерной поверхности плоскостями, параллельными плоскости YOZ.

Кроме простейшей функции `plot3()`, система MATLAB располагает еще рядом функций, позволяющих добиваться большей наглядности при изображении трехмерных графиков. Это функции `mesh()`, `meshc()`, `meshz()`, `surf()`, `surfl()`, `surfc()`, `waterfall()`, `contour3()`, которые создают графические объекты типа `surface`.

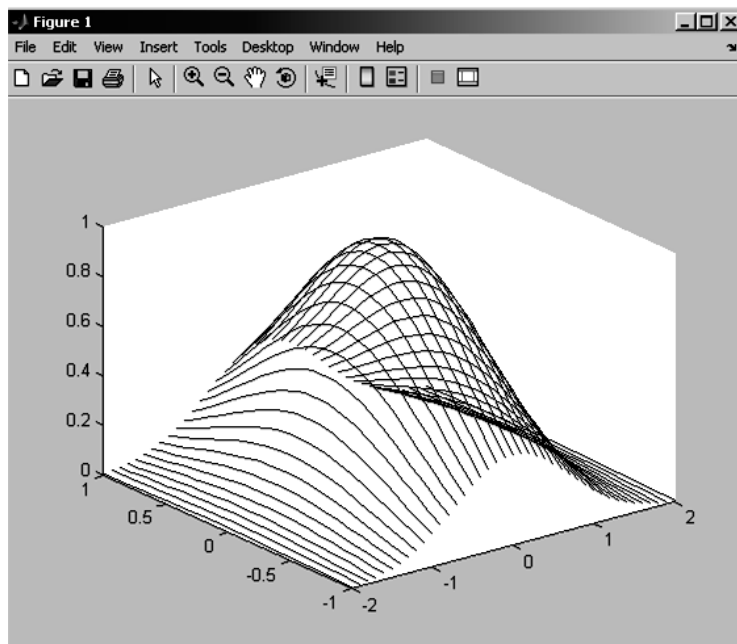


Рис. 1.11. Поверхность, построенная линиями (`plot3()`)

Функция `mesh(x,y,z)` строит каркасную поверхность (рис. 1.12). По умолчанию рисуется только видимая часть поверхности. При помощи команды `hidden off` можно сделать каркасную поверхность прозрачной, добавив невидимую часть. Команда `hidden on` убирает невидимую часть поверхности, возвращая графику прежний вид.

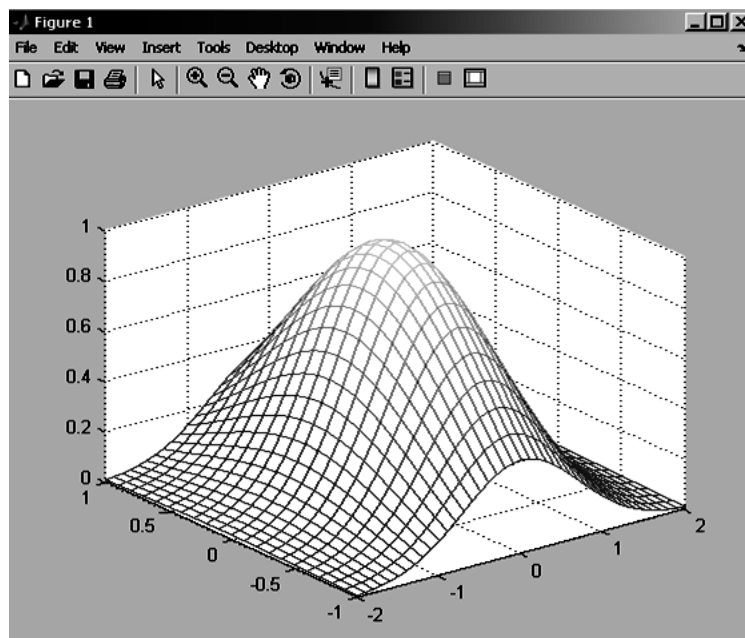


Рис. 1.12. Каркасная поверхность (`mesh()`)

Для получения непрозрачной сетчатой поверхности, отдельные ячейки которой автоматически окрашиваются в разные цвета, используется команда `surf(x,y,z)`. Результат работы функции `surf()` показан на *рис. 1.13*.

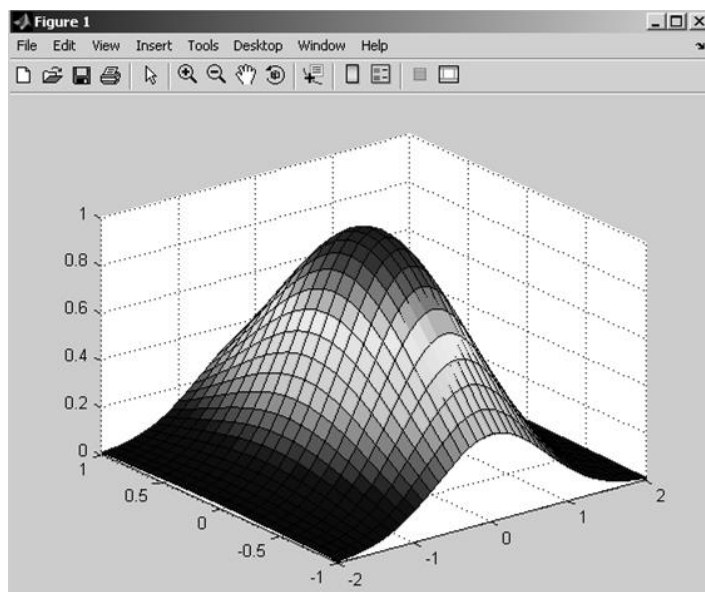


Рис. 1.13. Каркасная поверхность, залитая цветом (`surf()`)

Команда `shading flat` позволяет убрать каркасные линии. Для получения поверхности, плавно залитой цветом, используется команда `shading interp`. Команда `shading faceted` позволяет вернуться к виду поверхности, представленному на *рис. 1.13*. Со всеми функциями, строящими трехмерные объекты, можно использовать команду `colorbar`, которая выводит рядом с графиком цветовую шкалу, устанавливающую связь между цветом и значением функции (*рис. 1.14*).

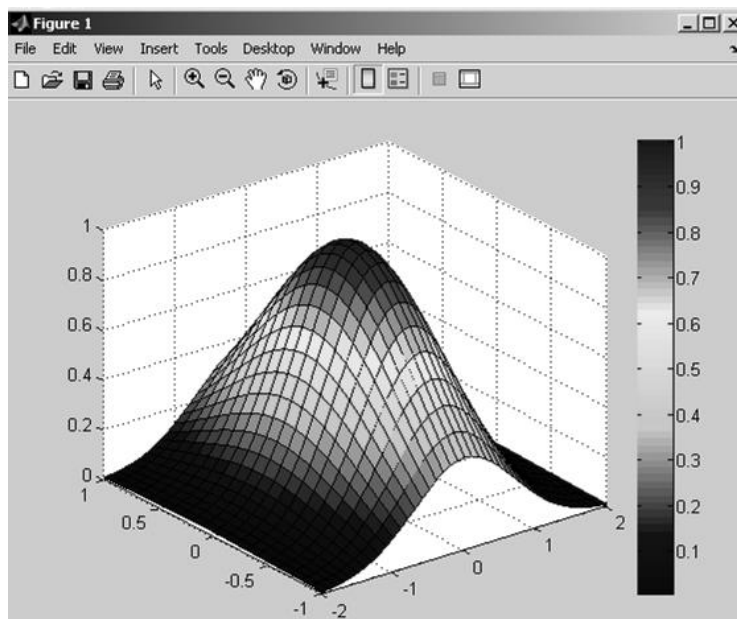


Рис. 1.14. График с цветовой шкалой (`colorbar`)

Для того, чтобы добиться более естественного окрашивания поверхности, следует использовать функцию `surf()`, которая использует встроенную засветку графика со стороны некоторой геометрической точки пространства. В результате точки поверхности графика, обращенные к источнику света, изображаются красным цветом, а точки, находящиеся в тени, – синим.

Более информативными являются графики, содержащие линии уровня, т. е. линии постоянства значений функции, на плоскости OXY . Для получения таких графиков следует использовать функции `meshc()` или `surfsc()`. Например, команды

```
>> [x, y]=meshgrid(-2:0.1:2);  
>> z=x.*exp(-x.^2-y.^2);  
>> meshc(x,y,z)
```

позволяют получить следующее изображение (рис. 1.15).

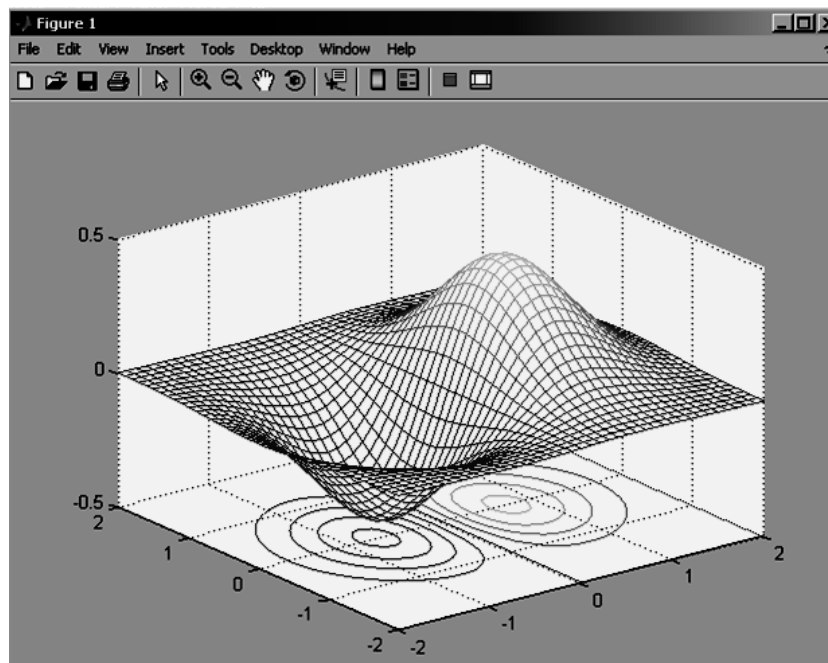


Рис. 1.15. График с линиями уровня (`meshc()`)

MATLAB позволяет построить поверхность, состоящую из линий уровня, при помощи функции `contour3()`, которую можно использовать так же, как и описанные выше функции с тремя аргументами. Имеется возможность задать четвертым аргументом в `contour3()` либо число линий уровня, либо вектор, элементы которого равны значениям функции, отображаемым в виде линий уровня. Например (рис. 1.16):

```
>> contour3(x,y,z,-0.5:0.02:0.5)
```

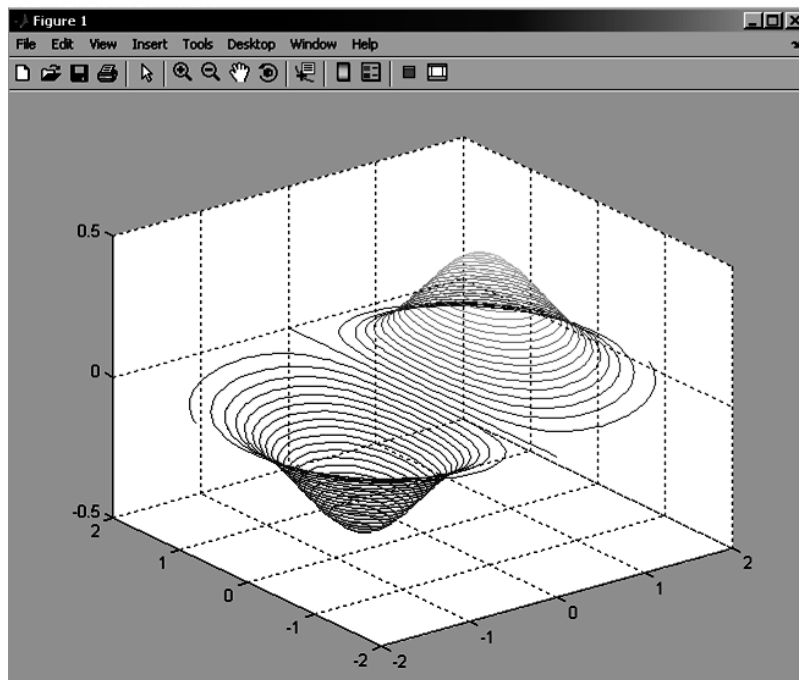


Рис. 1.16. График, состоящий из линий уровня (`contour3()`)

Использование команды `meshz()` приводит к построению графика с пьедесталом (рис. 1.17).

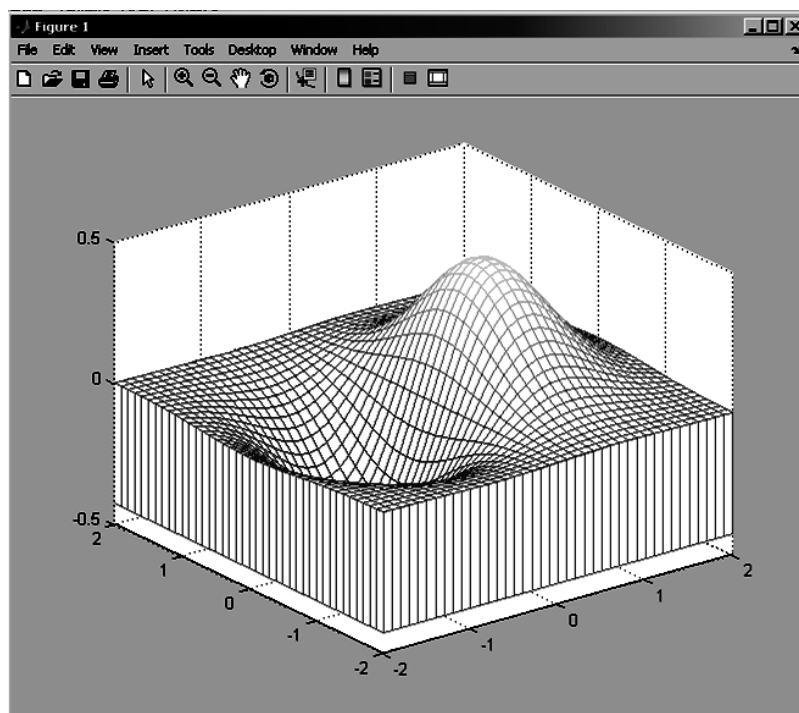


Рис. 1.17. График с пьедесталом (`meshz()`)

При помощи команды `waterfall()` можно построить поверхность, состоящую из тонких слоев (рис. 1.18).

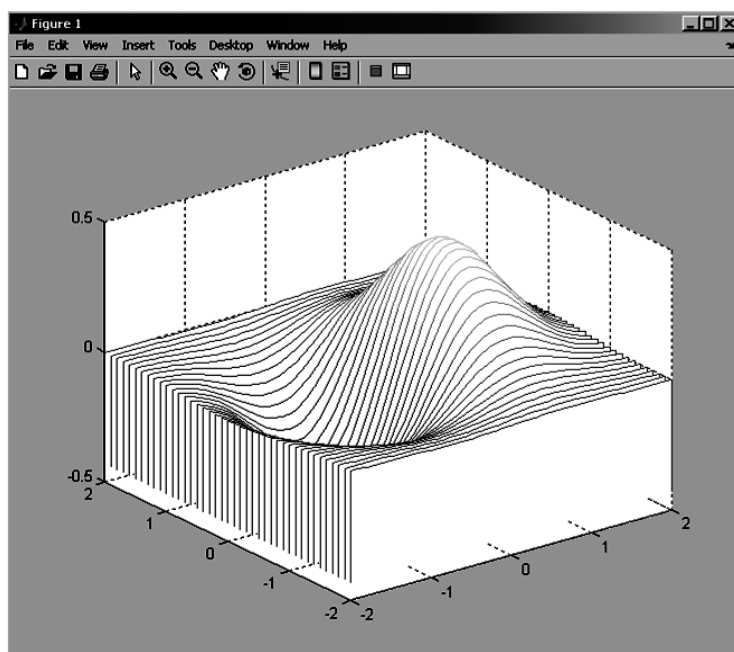


Рис. 1.18. Поверхность, состоящая из слоев (waterfall())

1.4.2. Оформление графиков

При необходимости изменения цветовой палитры можно использовать функцию `colormap()` с указанием типа палитры. Цветовые палитры, доступные в MATLAB, приведены в табл. 1.2.

Т а б л и ц а 1.2. Палитры цвета

Палитра	Изменение цвета
autumn	Красный-оранжевый-желтый
bone	Оттенки серого и синего
colorcube	Изменение цветов от темного к яркому
cool	Оттенки голубого и пурпурного
copper	Оттенки медного цвета
flag	Красный-белый-синий-черный
gray	Оттенки серого
hot	Черный-красный-оранжевый-желтый-белый
hsv	Цвета радуги
jet	Синий-голубой-зеленый-желтый-красный
pink	Оттенки коричневого
prism	Красный-оранжевый-желтый-зеленый-синий-фиолетовый
spring	Оттенки пурпурного и желтого
summer	Оттенки зеленого и желтого
vga	Палитра из шестнадцати цветов
white	Белый
winter	Оттенки синего и зеленого

Для восстановления первоначального цвета палитры следует применить команду `colormap('default')`.

Добавление заголовка графика и подписей к осям осуществляются теми же командами `title()`, `xlabel()`, `ylabel()`, что применялись при визуализации функции одной переменной. Для вертикальной оси предназначена команда `zlabel()`.

Пример оформления графика (рис. 1.19):

```
>> surfc(x,y,z)
>> colorbar
>> colormap(copper)
>> title('ГРАФИК ФУНКЦИИ z(x,y)')
>> xlabel('x')
>> ylabel('y')
>> zlabel('z')
```

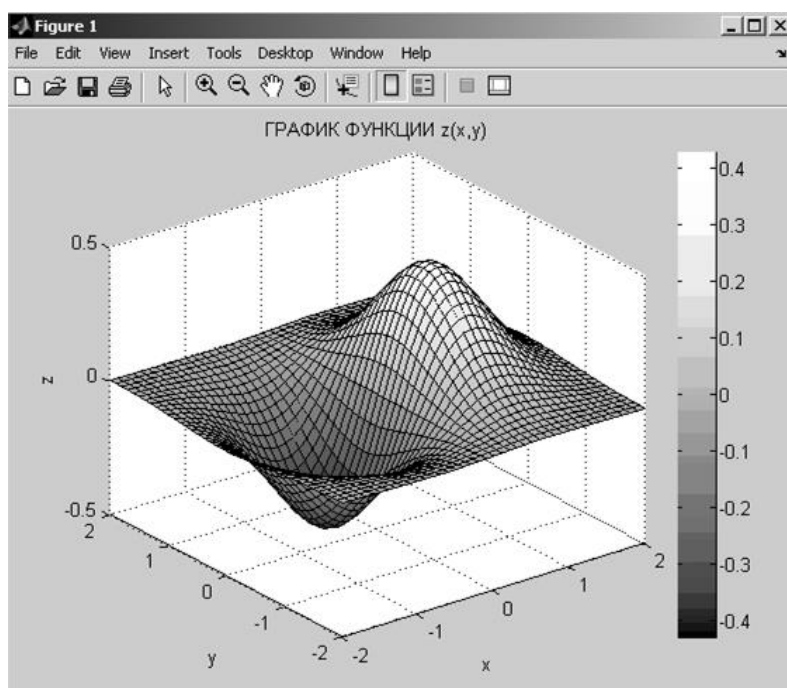


Рис. 1.19. График с нанесенными обозначениями (палитра `copper`)

Построение нескольких поверхностей можно осуществить в одних координатных осях, отдельных графических окнах, нескольких областях одного графического окна. При этом используются те же функции, что и для построения нескольких графиков функций одной переменной.

Для изменения масштаба поверхности можно воспользоваться кнопками `Zoom In` и `Zoom Out` графического окна. Для вращения поверхности с использованием мыши необходимо нажать на кнопку `Rotate 3D`. Нажатие кнопки `Show Plot Tools` приводит к появлению редактора графиков (рис. 1.20), ис-

пользование которого позволяет легко изменять свойства поверхностей. Работа с редактором графиков поверхностей осуществляется аналогично работе с редактором графиков функций одной переменной.

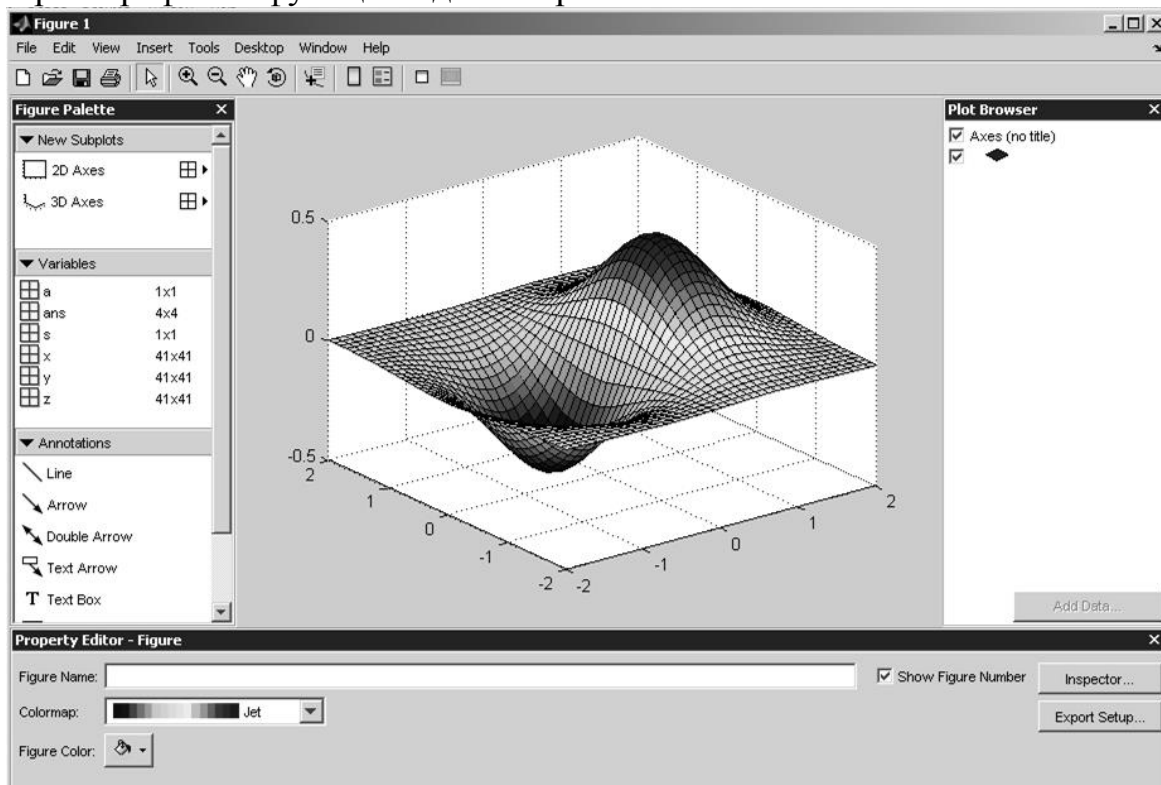


Рис. 1.20. Среда редактора графиков

Построение поверхностей можно выполнять с использованием окна *Workspace*, по аналогии с построением функций одной переменной.

Задания для самостоятельного выполнения

Построить графики функций двух переменных на заданном интервале изменения значений аргументов. Использовать разные функции для построения поверхностей. Установить подписи к осям, заголовков, цветовую палитру, вывести цветовую шкалу.

Номер задания	Функции
1	$z = \frac{\sin(\sqrt{x^2 + y^2})}{\sqrt{x^2 + y^2}}, x, y \in [-8; 8]; z_1 = x_1 \sin(x_1 + y_1), x_1, y_1 \in [-5; 5]$
2	$z = \frac{\sin x}{x^2 + y^2 + 0.3}, x, y \in [-3; 3]; z_1 = \sin^3 x_1 + 4 \cos^2 y_1, x_1, y_1 \in [-5; 5]$
3	$z = (1 - x^2)y(1 - y), x \in [-1; 1], y \in [0; 1]; z_1 = x_1^2 + y_1^2, x_1, y_1 \in [-3; 3]$

Номер задания	Функции
4	$z = e^{2x+y}$, $x, y \in [-3;3]$; $z_1 = (x_1^2 - y_1^2) \sin 3x_1$, $x_1, y_1 \in [-4;4]$
5	$z = \cos x \sin y$, $x, y \in [-5;5]$; $z_1 = x_1^2 - y_1^2$, $x_1, y_1 \in [-3;3]$
6	$z = \cos^2 x \sin^2 y$, $x, y \in [0;4]$; $z_1 = \sqrt{\cos x_1 \sin y_1}$, $x_1, y_1 \in [-4;4]$
7	$z = (1 - x^3)y$, $x, y \in [-4;4]$; $z_1 = (1 - x_1^2) \sin y_1$, $x_1, y_1 \in [-5;5]$
8	$z = x / \sin 2y$, $x, y \in [-2;2]$; $z_1 = \cos(x_1 - y_1) / x_1^2$, $x_1, y_1 \in [-2;2]$
9	$f = 2e^{x+y}$, $x, y \in [-3;3]$; $f_1 = (x_1^2 - 2y_1^2) \sin y_1$, $x_1, y_1 \in [-4;4]$
10	$z = \frac{\cos(\sqrt{x^2 + y^2})}{\sqrt{x^2 + y^2}}$, $x, y \in [-7;7]$; $z_1 = y_1 \cos(x_1 + y_1)$, $x_1, y_1 \in [-5;5]$
11	$f = (1 - x^2)x(1 - 4y)$, $x \in [-1;1]$, $y \in [0;1]$; $f_1 = 2x_1^2 + y_1^2$, $x_1, y_1 \in [-3;3]$
12	$f = \cos y \sin x$, $x, y \in [-5;5]$; $f_1 = x_1^2 - 2y_1^2$, $x_1, y_1 \in [-3;3]$
13	$f = (1 - 3x^3)5y$, $x, y \in [-4;4]$; $f_1 = (1 - x_1^2) \sin 2y_1$, $x_1, y_1 \in [-5;5]$
14	$f = 3x / \sin y$, $x, y \in [-2;2]$; $f_1 = \sin(x_1 - y_1) / x_1^2$, $x_1, y_1 \in [-2;2]$
15	$f = \frac{\cos x}{x^2 + y^2 + 0.3}$, $x, y \in [-3;3]$; $f_1 = \sin^3 x_1 + 4 \cos^2 y_1$, $x_1, y_1 \in [-5;5]$

1.5. КОНТРОЛЬНЫЕ ЗАДАНИЯ

1.5.1. Вычисление арифметических выражений и функций

Задание 1				
1	Вычислите указанное арифметическое выражение. $\frac{(68,023 - 66,028) : 6\frac{1}{9} + \frac{7}{40} \cdot 4,5}{0,042 + 0,086}$			
2	Проведите вычисления по заданной формуле при заданных значениях параметров. $\frac{\sqrt{3h^3}}{\cos^2 \alpha} \sin(\alpha + 30^\circ) \sin(\alpha - 30^\circ)$			
	<table border="0"> <tr> <td>а)</td> <td>б)</td> </tr> <tr> <td>$h = 0,28$; $\alpha = 41^\circ$</td> <td>$h = e^{0,415}$; $\alpha = 237^\circ$</td> </tr> </table>	а)	б)	$h = 0,28$; $\alpha = 41^\circ$
а)	б)			
$h = 0,28$; $\alpha = 41^\circ$	$h = e^{0,415}$; $\alpha = 237^\circ$			

Задание 1				
3	Вычислить значение выражения. Результат представить в алгебраической форме (в виде $z = a + b \cdot i$) и экспоненциальной форме (в виде $z = Ae^{j\varphi}$)			
	$z = (\sqrt{z_1 : z_2 - z_3}) \cdot z_4$			
	z_1	z_2	z_3	z_4
	$1,25e^{0,6i}$	$-\sqrt{3} - 4i$	$4 + 3i$	$2,5e^{3,8i}$

Задание 2				
1	Вычислите указанное арифметическое выражение. $\frac{(2,1 - 1,965) : (1,2 \cdot 0,045)}{0,00325 : 0,013} - \frac{4}{0,2 \cdot 0,73}$			
2	Проведите вычисления по заданной формуле при заданных значениях параметров. $\frac{\alpha}{3} (\lg(d + 2) - \operatorname{tg} \alpha)^2$			
	а)		б)	
	$d = 6,178; \alpha = 20^\circ$		$d = -2,2461 \cdot 10^{-2}; \alpha = 1,146$	
3	Вычислить значение выражения. Результат представить в алгебраической форме (в виде $z = a + b \cdot i$) и экспоненциальной форме (в виде $z = Ae^{j\varphi}$)			
	$z = \sqrt{(z_1 : z_2 + z_3)} \cdot z_4$			
	z_1	z_2	z_3	z_4
	$1,05e^{-0,4i}$	$\sqrt{5} - i$	$2,7e^{0,8i}$	$-0,7 + 4i$

Задание 3			
1	Вычислите указанное арифметическое выражение. $\frac{(1,88 + 2,127) \cdot 0,01875}{0,625 - \frac{13}{18} : 3,13} + 8,29$		
2	Проведите вычисления по заданной формуле при заданных значениях параметров. $d^3 \operatorname{ctg} \alpha \cdot \sqrt{\sin^4 \alpha - \cos^4 \alpha}$		
	а)		б)
	$d = 10,6; \alpha = 50^\circ$		$d = e^{2,3}; \alpha = 1$

Задание 3				
3	Вычислить значение выражения. Результат представить в алгебраической форме (в виде $z = a + b \cdot i$) и экспоненциальной форме (в виде $z = Ae^{j \cdot \varphi}$)			
	$z = \sqrt{(z_1 \cdot z_2 - z_3)} : z_4$			
	z_1	z_2	z_3	z_4
	$2,71e^{73^\circ i}$	$1 + i \frac{\pi}{2}$	$\sqrt{2} + \sqrt{3}i$	$1,93e^{192^\circ i}$

Задание 4				
1	Вычислите указанное арифметическое выражение. $\frac{3 : 0,4 - 0,009 : (0,15 : 2,5)}{0,32 \cdot 6 + 0,033 - (5,3 - 3,88)}$			
2	Проведите вычисления по заданной формуле при заданных значениях параметров. $\frac{a^2 \sqrt{3}}{4} (\sec \alpha + \operatorname{cosec} \alpha)^4$			
	а) $a = 5,08; \alpha = 25^\circ$	б) $a = \ln 1,37; \alpha = \frac{12}{25} \pi$		
3	Вычислить значение выражения. Результат представить в алгебраической форме (в виде $z = a + b \cdot i$) и экспоненциальной форме (в виде $z = Ae^{j \cdot \varphi}$)			
	$z = \sqrt{(z_1 + z_2)} : z_3 \cdot z_4$			
	z_1	z_2	z_3	z_4
	$2,7 + 0,8i$	$2e^{-\sqrt{3}i}$	$0,81e^{i \cdot \frac{\pi}{7}}$	$-\sqrt{2} - \sqrt{3}i$

Задание 5	
1	Вычислите указанное арифметическое выражение. $\frac{(34,06 - 33,81) \cdot 4}{6,84 : (28,57 - 25,15)} + 1,33 : \frac{4}{21}$
2	Проведите вычисления по заданной формуле при заданных значениях параметров. $\frac{\sqrt{\pi}}{3} \cdot \frac{1}{(\operatorname{ctg} A + \operatorname{ctg} B)^2}$
	а) $A = 51^\circ; B = 39^\circ$

Задание 5				
3	Вычислить значение выражения. Результат представить в алгебраической форме (в виде $z = a + b \cdot i$) и экспоненциальной форме (в виде $z = Ae^{j \cdot \varphi}$)			
	$z = \sqrt{(z_1 + z_2) \cdot z_3 : z_4}$			
	z_1	z_2	z_3	z_4
	$-0,8 + 2,7i$	$-2e^{\sqrt{3}i}$	$0,9e^{i \cdot \frac{\pi}{7}}$	$3,1 - 2,1i$

Задание 6				
1	Вычислите указанное арифметическое выражение.			
	$\frac{8,8077}{20 - (28,2 : (13,333 \cdot 0,3 + 0,0125)) \cdot 2,004}$			
2	Проведите вычисления по заданной формуле при заданных значениях параметров.			
	$\lg\left(3^{x^2 - x - 9} + \frac{8}{27}\right)$			
	a)	б)		
	$x = e^{1,648}$	$x = \operatorname{tg} 1,21$		
3	Вычислить значение выражения. Результат представить в алгебраической форме (в виде $z = a + b \cdot i$) и экспоненциальной форме (в виде $z = Ae^{j \cdot \varphi}$)			
	$z = \sqrt{z_1 - z_2 \cdot z_3 : z_4}$			
	z_1	z_2	z_3	z_4
	$-1,1 - 3,2i$	$0,33e^{-1,9i}$	$2e^{\sqrt{2}i}$	$2,08 + i$

Задание 7				
1	Вычислите указанное арифметическое выражение.			
	$\frac{\left(1,75 : \frac{2}{3} - 1,75 \cdot 1,125\right) : \frac{7}{12}}{(0,2012 - 0,0325) : 400}$			
2	Проведите вычисления по заданной формуле при заданных значениях параметров.			
	$\frac{\sqrt[5]{5e^{4a}} (a + 12,36)^2}{\ln(a + 7)}$			
	a)	б)		
	$a = 2,1754$	$a = \cos 17^\circ$		

Задание 7			
3	Вычислить значение выражения. Результат представить в алгебраической форме (в виде $z = a + b \cdot i$) и экспоненциальной форме (в виде $z = Ae^{j \cdot \varphi}$) $z = \sqrt{z_1 - z_2} : z_3 \cdot z_4$		
	z_1	z_2	z_3
	$2,1 - 3,2i$	$0,68e^{148^\circ i}$	$\sqrt{5} + \sqrt{2}i$

Задание 8			
1	Вычислите указанное арифметическое выражение. $\frac{\left(26\frac{1}{3} - 18,02 \cdot 0,75\right) \cdot 2,4 : 0,88}{1,37 - 23\frac{2}{3} : 1,82}$		
2	Проведите вычисления по заданной формуле при заданных значениях параметров. $\lg^2 x - \left(\frac{27}{8}\right)^{x-1} \sin \sqrt{x}$		
	а) $x = e^{2,145}$		
	б) $x = 2,468 \cdot 10^{-1}$		
3	Вычислить значение выражения. Результат представить в алгебраической форме (в виде $z = a + b \cdot i$) и экспоненциальной форме (в виде $z = Ae^{j \cdot \varphi}$) $z = \left((z_1 + z_2)^2 - z_3\right) : z_4$		
	z_1	z_2	z_3
	$1,1e^{-0,8i}$	$\sqrt{5} - \sqrt{2}i$	$-1,7 + i$

Задание 9	
1	Вычислите указанное арифметическое выражение. $26 : \frac{3 : (0,48 - 0,27)}{2,52(1,38 + 2,45)} + 1,27$
2	Проведите вычисления по заданной формуле при заданных значениях параметров. $\sqrt[5]{(x - y)^2} \cdot \sqrt[3]{\frac{1}{y - x}}$

Задание 9			
	а)		б)
	$x = e^{-0,37}; y = \ln 2,1517$		$x = 37^\circ; y = \cos \frac{7}{24} \pi$
3	Вычислить значение выражения. Результат представить в алгебраической форме (в виде $z = a + b \cdot i$) и экспоненциальной форме (в виде $z = Ae^{j \cdot \varphi}$)		
	$z = \left((z_1 - z_2)^2 - z_3 \right) : z_4$		
	z_1	z_2	z_3
	$2,1e^{0,8i}$	$-\sqrt{5} + 2i$	$1,7e^{\sqrt{3}i}$
			z_4
			$0,8e^{2,5i}$

Задание 10			
1	Вычислите указанное арифметическое выражение. $\left(16,5 - 13 \frac{7}{9} \right) \frac{6}{11} + 2,2 : (0,241 - 0,91)$		
2	Проведите вычисления по заданной формуле при заданных значениях параметров. $\frac{\sin A + \operatorname{tg} B}{\sqrt[5]{(A - 3B)^2}}$		
	а)		б)
	$A = 5,6; B = \lg 25$		$A = \frac{8}{9} \pi; B = \frac{\pi}{10}$
3	Вычислить значение выражения. Результат представить в алгебраической форме (в виде $z = a + b \cdot i$) и экспоненциальной форме (в виде $z = Ae^{j \cdot \varphi}$)		
	$z = (z_1 - z_2)^2 : z_3 + z_4$		
	z_1	z_2	z_3
	$1,1e^{-2,1i}$	$\frac{\pi}{8} - 2,1i$	$2,71 + 0,4i$
			z_4
			$1,71e^{-\sqrt{3}i}$

1.5.2. Работа в системе MATLAB с векторами и матрицами

1. Вычислить значение функции $f(x)$ на отрезке $x \in [a; b]$ с шагом h .

$f(x)$	a	b	h
1. $\sqrt{\operatorname{ch} \frac{x}{\sqrt{2\pi}}}, \left(\operatorname{ch} x = \frac{e^x + e^{-x}}{2} \right)$	0,5	1,5	0,1

	$f(x)$	a	b	h
2.	$\frac{x^3 + 2x}{\sqrt{1 + e^x}}$	-0,2	0,8	0,1
3.	$\sqrt{1 + 2x^2} \cdot \sin \frac{3x}{2}$	2	4	0,2
4.	$\sqrt{3x^2 + 5} \cdot \cos \frac{\pi x}{2}$	0,5	1,5	0,1
5.	$\arccos e^{-\sqrt[3]{3x}}$	0,2	0,5	0,03
6.	$\arcsin e^{\frac{x^2}{5}}$	8	13	0,5
7.	$x + \ln(x + \sqrt{1 + x^2})$	-0,5	0,5	0,1
8.	$\frac{1 + e^{-\frac{x}{2}}}{\sqrt{3x^2 + 1}}$	3	5	0,2
9.	$3x^3 + \frac{1}{x} + e^{-2x^2}$	1,2	2,2	0,1
10.	$x^{2x+1} + x^3 - 2x$	1	5	0,4

2. Матрицы и операции с матрицами.

1) Дана матрица:

$$A = \begin{vmatrix} -43 & 119 & -18 & 11 & 29 & 85 \\ -166 & 118 & 72 & 106 & -133 & 125 \\ 12 & -3 & -58 & 5 & 71 & -159 \\ 28 & 32 & 218 & -9 & 162 & -144 \\ -114 & 17 & -13 & -83 & -69 & 57 \end{vmatrix}$$

Определить в матрице индексы наибольшего и наименьшего элементов. Переставить местами строки, содержащие наибольший и наименьший элементы.

2) Дана матрица:

$$A = \begin{vmatrix} -43 & 119 & -18 & 11 & 29 & 85 \\ -166 & 118 & 72 & 106 & -133 & 125 \\ 12 & -3 & -58 & 5 & 71 & -159 \\ 28 & 32 & 218 & -9 & 162 & -144 \\ -114 & 17 & -13 & -83 & -69 & 57 \end{vmatrix}$$

Определить в матрице индексы наибольшего и наименьшего элементов. Переставить местами столбцы, содержащие наибольший и наименьший элементы.

3) Дана матрица:

$$A = \begin{vmatrix} -43 & 119 & -18 & 11 & 29 & 85 \\ -166 & 118 & 72 & 106 & -133 & 125 \\ 12 & -3 & -58 & 5 & 71 & -159 \\ 28 & 32 & 218 & -9 & 162 & -144 \\ -114 & 17 & -13 & -83 & -69 & 57 \end{vmatrix}$$

Удалить в матрице строку, содержащую наибольший элемент.

4) Дана матрица:

$$A = \begin{vmatrix} -43 & 119 & -18 & 11 & 29 & 85 \\ -166 & 118 & 72 & 106 & -133 & 125 \\ 12 & -3 & -58 & 5 & 71 & -159 \\ 28 & 32 & 218 & -9 & 162 & -144 \\ -114 & 17 & -13 & -83 & -69 & 57 \end{vmatrix}$$

Удалить в матрице столбец, содержащий наибольший элемент.

5) Дана матрица:

$$A = \begin{vmatrix} -43 & 119 & -18 & 11 & 29 & 85 \\ -166 & 118 & 72 & 106 & -133 & 125 \\ 12 & -3 & -58 & 5 & 71 & -159 \\ 28 & 32 & 218 & -9 & 162 & -144 \\ -114 & 17 & -13 & -83 & -69 & 57 \end{vmatrix}$$

Удалить в матрице строку, содержащую наименьший элемент.

6) Дана матрица:

$$A = \begin{vmatrix} -43 & 119 & -18 & 11 & 29 & 85 \\ -166 & 118 & 72 & 106 & -133 & 125 \\ 12 & -3 & -58 & 5 & 71 & -159 \\ 28 & 32 & 218 & -9 & 162 & -144 \\ -114 & 17 & -13 & -83 & -69 & 57 \end{vmatrix}$$

Удалить в матрице столбец, содержащий наименьший элемент.

7) Дана матрица:

$$A = \begin{vmatrix} -43 & 119 & -18 & 11 & 29 & 85 \\ -166 & 118 & 72 & 106 & -133 & 125 \\ 12 & -3 & -58 & 5 & 71 & -159 \\ 28 & 32 & 218 & -9 & 162 & -144 \\ -114 & 17 & -13 & -83 & -69 & 57 \end{vmatrix}$$

Найти в матрицы индексы положительных элементов. Изменить знак положительных элементов матрицы.

8) Дана матрица:

$$A = \begin{vmatrix} -43 & 119 & -18 & 11 & 29 & 85 \\ -166 & 118 & 72 & 106 & -133 & 125 \\ 12 & -3 & -58 & 5 & 71 & -159 \\ 28 & 32 & 218 & -9 & 162 & -144 \\ -114 & 17 & -13 & -83 & -69 & 57 \end{vmatrix}$$

Найти в матрицы индексы отрицательных элементов. Изменить знак отрицательных элементов матрицы.

9) Дана матрица:

$$A = \begin{vmatrix} -43 & 119 & -18 & 11 & 29 & 85 \\ -166 & 118 & 72 & 106 & -133 & 125 \\ 12 & -3 & -58 & 5 & 71 & -159 \\ 28 & 32 & 218 & -9 & 162 & -144 \\ -114 & 17 & -13 & -83 & -69 & 57 \end{vmatrix}$$

Найти среднее арифметическое значение элементов матрицы. Элементом матрицы, значение которых больше среднего арифметического, присвоить значения, равные нулю.

10) Дана матрица:

$$A = \begin{vmatrix} -43 & 119 & -18 & 11 & 29 & 85 \\ -166 & 118 & 72 & 106 & -133 & 125 \\ 12 & -3 & -58 & 5 & 71 & -159 \\ 28 & 32 & 218 & -9 & 162 & -144 \\ -114 & 17 & -13 & -83 & -69 & 57 \end{vmatrix}$$

Найти среднее арифметическое значение элементов матрицы. Элементом матрицы, значение которых меньше среднего арифметического, присвоить значения, равные нулю.

1.5.3. Построение и оформление графиков в системе MATLAB

Вычислить значение функции $f(x)$ на отрезке $x \in [a; b]$ с шагом h . Построить график функции. Нанести на график название, сетку и обозначение осей.

	$f(x)$	a	b	h
1.	$\sqrt{\operatorname{ch} \frac{x}{\sqrt{2\pi}}}, \left(\operatorname{ch} x = \frac{e^x + e^{-x}}{2} \right)$	0,5	1,5	0,1
2.	$\frac{x^3 + 2x}{\sqrt{1 + e^x}}$	-0,2	0,8	0,1
3.	$\sqrt{1 + 2x^2} \cdot \sin \frac{3x}{2}$	2	4	0,2
4.	$\sqrt{3x^2 + 5} \cdot \cos \frac{\pi x}{2}$	0,5	1,5	0,1
5.	$\arccos(e^{-\sqrt[3]{3x}})$	0,2	0,5	0,03
6.	$\arcsin\left(e^{-\frac{x^2}{5}}\right)$	8	13	0,5
7.	$x + \ln(x + \sqrt{1 + x^2})$	-0,5	0,5	0,1
8.	$\frac{1 + e^{-\frac{x}{2}}}{\sqrt{3x^2 + 1}}$	3	5	0,2
9.	$3x^3 + \frac{1}{x} + e^{-2x^2}$	1,2	2,2	0,1
10.	$x^{2x+1} + x^3 - 2x$	1	5	0,4

ЧАСТЬ 2. ПРОГРАММИРОВАНИЕ В MATLAB

2.1. Основные понятия программирования

При решении множества математических задач с применением системы MATLAB достаточно использовать режим прямых вычислений в рабочей области – *командный режим*. Однако часто возникает необходимость сохранить разработанный алгоритм вычислений в виде файла для многократного повторения его с различными исходными данными в разных сессиях работы. В таких случаях необходимо использовать возможности языка программирования системы – *программный режим работы*.

Система MATLAB имеет собственный язык программирования высокого уровня. Программы на языке программирования MATLAB сохраняются в виде файлов с расширением *.m* и называются *M-файлы*. При этом могут сохраняться как целые программы в виде файлов-сценариев, так и отдельные *программные модули* – функции.

2.1.1. Средства языка программирования

Язык программирования системы MATLAB имеет следующие средства:

- данные;
- константы и переменные;
- операторы;
- встроенные функции и команды;
- функции, определенные пользователем;
- управляющие структуры;
- системные операторы и функции;
- средства расширения языка.

2.1.2. Типы программирования

В системе MATLAB реализуются различные типы программирования.

Если в основе программирования лежат процедуры, операторы или функции, используемые как основные объекты языка, такое программирование относится к *процедурному*, *операторному* или *функциональному* типам.

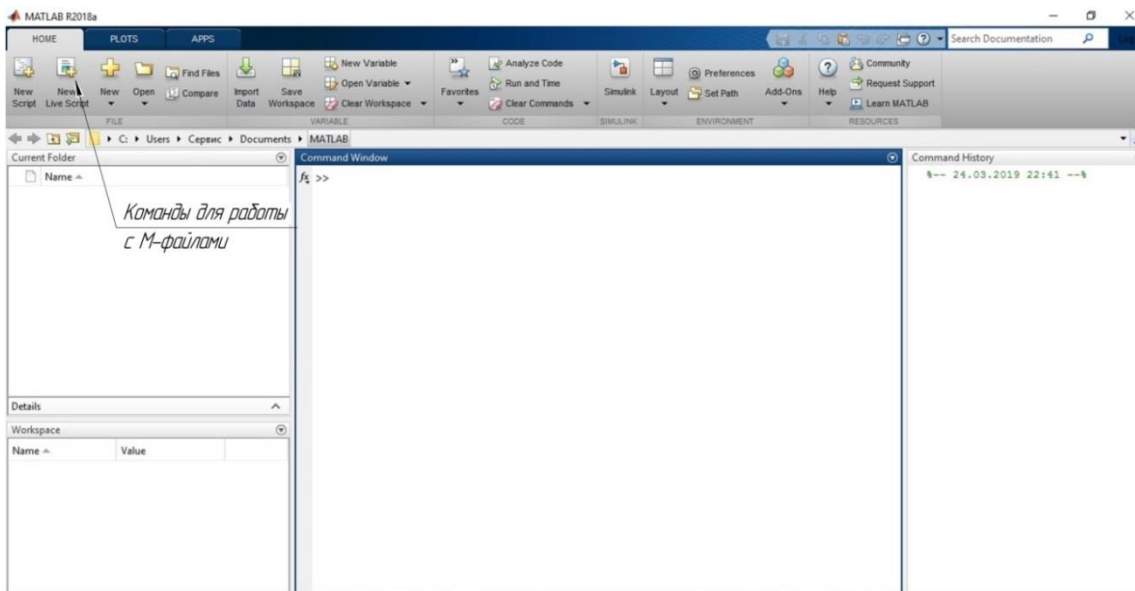
Логические операторы и функции MATLAB используются при программировании *логического* типа.

Структурное программирование предусматривает составление программ из отдельных модулей в виде *M-файлов*, которые включают функции и команды языка. Между указанными модулями в процессе работы программы происходит обмен информацией.

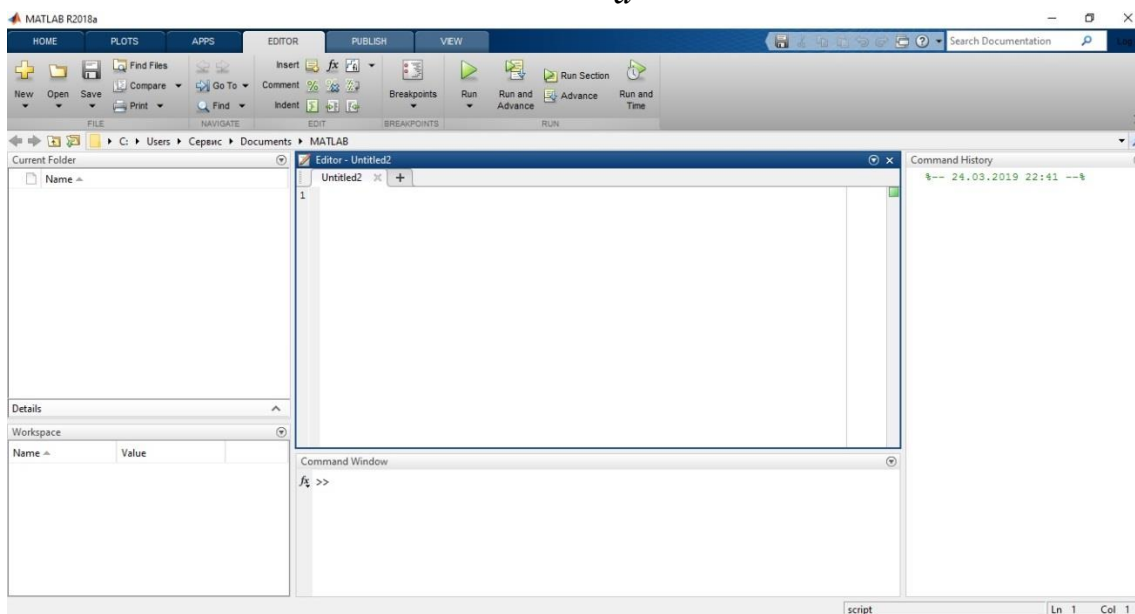
MATLAB также предоставляет возможности *объектно-ориентированного* и *визуально-ориентированного* программирования.

2.1.3. Работа с файлами

Операции и команды для работы с файлами содержатся на панели быстрого доступа (вкладки **Home** и **Editor**). При этом вкладка **Editor** появляется только после создания (открытия уже существующего) М-файла (рис. 2.1, а, б).



а



б

Рис. 2.1. Команды для работы с файлами

Рассмотрим назначение некоторых операций и команд для работы с файлами:

- **New** – открывает подменю для создания следующих типов документов: М-файла-сценария (**Script**); М-файла-функции (**Function**); интерактивного документа, который объединяет код **MATLAB** с форматированным текстом, уравнениями и изображениями (**Live script** или **Live function**); новый собственный класс для вычислительных объектов (**Class**); системный объект для работы с потоковыми данными (**System object**), используется, например, для обработки звуковых сигналов, оптического потока и т. д.;
- **Open...** – открывает окно для загрузки ранее созданного М-файла, рисунка или модели;
- **Import Data** – открывает окно для загрузки файлов с расширением **.mat**, содержащих определенные в предыдущих сессиях переменные и функции;
- **Save Workspace** – сохранение всех определенных в данной сессии переменных и функций в виде файла с расширением **.mat**, то есть сохранение рабочего пространства **Workspace** системы;
- **Set Path...** - выводит окно настройки путей файловой системы;
- **Preferences...** – выводит окно настройки параметров системы;
- **Page Setup...** – выводит окно настройки параметров страницы.

2.2. Основы подготовки, редактирования и отладки М-файлов

Создание М-файлов в системе **MATLAB** требует выполнения следующих действий:

- 1) вызов специального *редактора* (**Editor**) для написания текста, редактирования и отладки файла;
- 2) ввод текста файла в окно редактора;
- 3) сохранение введенной информации;
- 4) запуск файла для исполнения;
- 5) просмотр результатов выполнения М-файла в рабочей области.

После просмотра результатов выполнения файла при необходимости можно вернуться в окно редактора для внесения изменений в текст файла, а затем, сохранив изменения, снова запустить файл для исполнения.

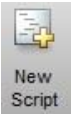
Для работы с *ранее созданным* файлом нужно:

- 1) открыть файл в окне редактора;
- 2) при необходимости внести изменения в текст файла;
- 3) сохранить информацию;
- 4) запустить файл для исполнения;
- 5) просмотреть результаты счета в рабочей области.

Рассмотрим этапы разработки М-файлов подробно.

2.2.1. Редактор М-файлов

Для разработки М-файлов служит специальный редактор, который можно вызвать одним из следующих способов:

- выполнить команду **New – script** или **New – Function** главного меню основного окна приложения (рис. 2.2);
- воспользоваться кнопкой  панели инструментов основного окна;
- набрать команду **edit** в командной строке и нажать на клавишу **<Enter>**;
- воспользоваться комбинацией клавиш **<Ctrl+N>**.

В появившемся окне редактора можно создавать новый файл, производить его отладку и запуск. Окно редактора М-файлов имеет заголовок, меню, панели инструментов и строку состояния.

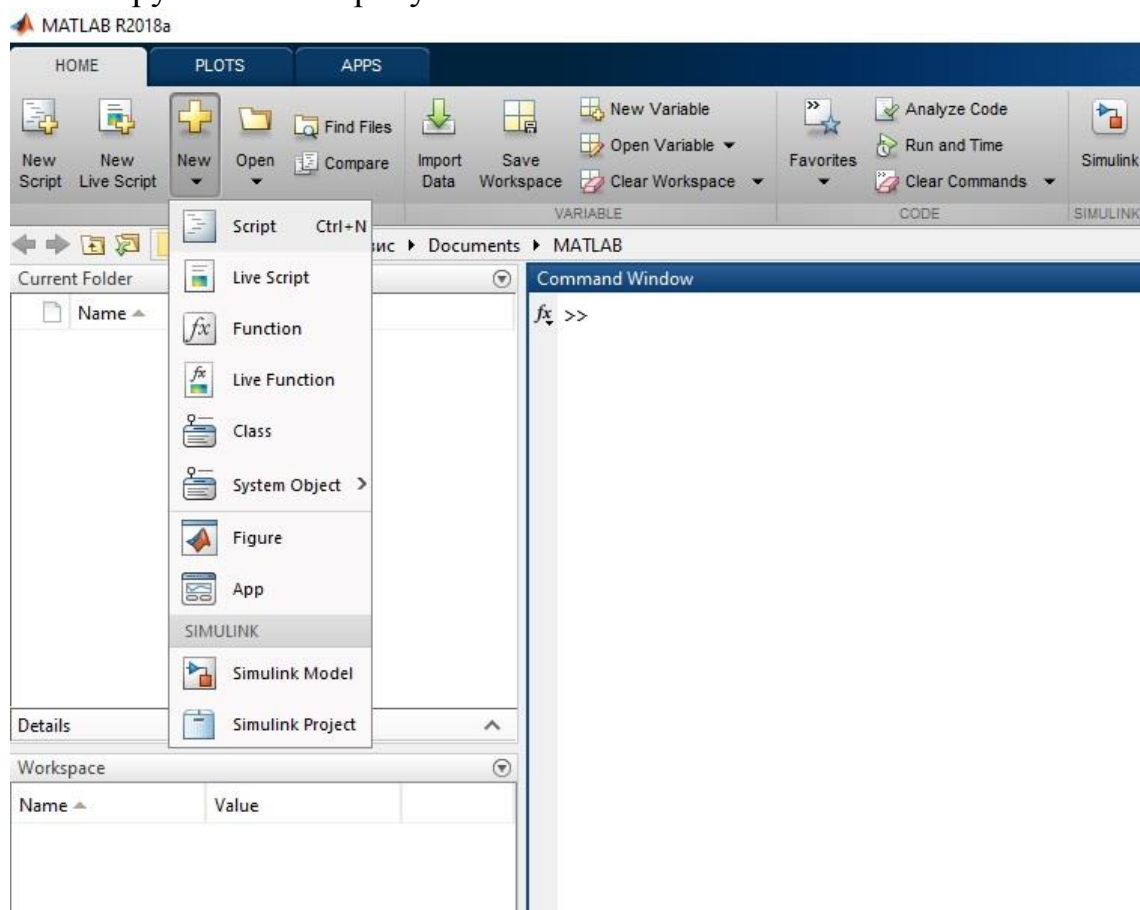




Рис. 2.2. Вызов редактора М-файлов с помощью меню

Окно редактора может располагаться отдельно от основного окна приложения или может стать частью среды системы MATLAB (рис. 2.3). Изменить

расположение окна редактора позволяют кнопки  контекстного меню, вызываемого кнопкой  (рис. 2.3).

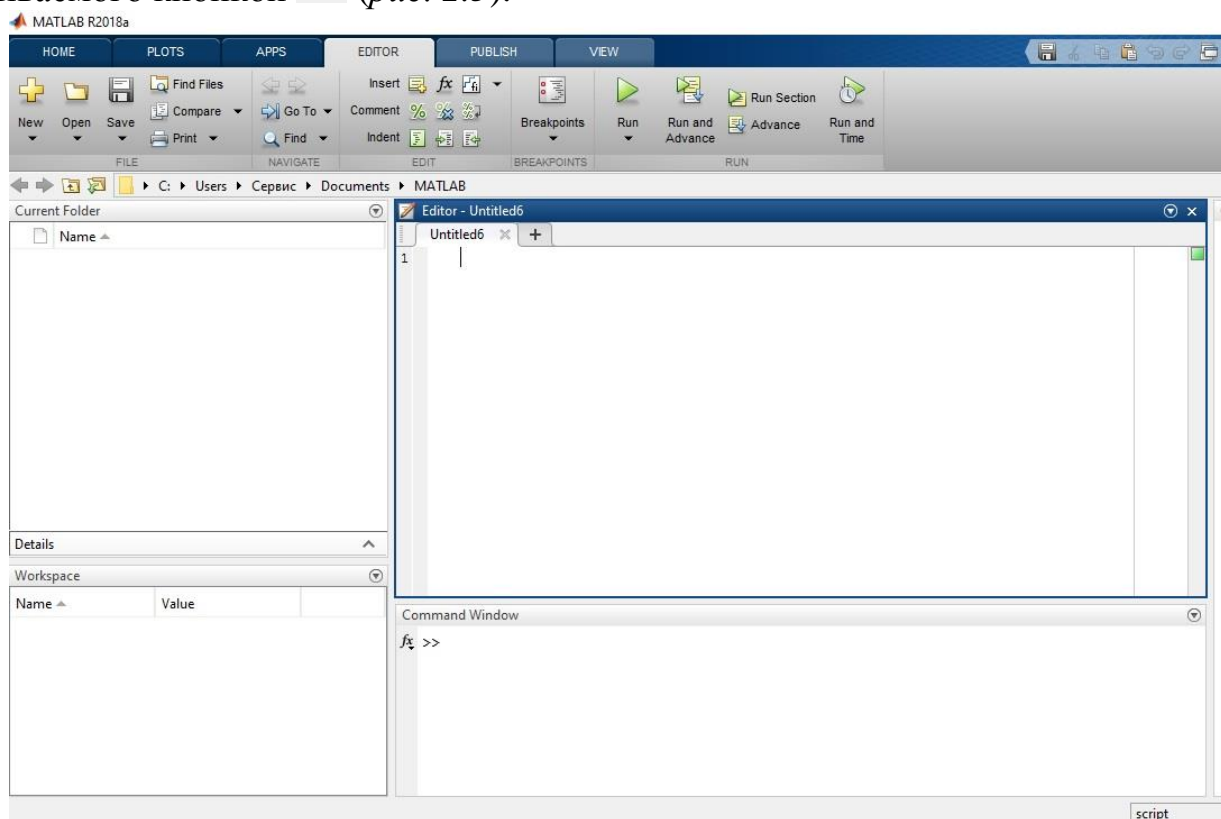


Рис. 2.3. Окно редактора М-файлов как часть среды системы MATLAB

Если необходимо работать с *ранее созданным* файлом, его можно *открыть* любым из следующих способов:

- через главное меню основного окна приложения (рис. 2.1, а) командой **Open**;
- с помощью комбинации клавиш **<Ctrl> + <O>**.

2.2.2. Ввод текста М-файлов

При вводе текста в окно редактора М-файлов необходимо соблюдать правила, принятые для команд MATLAB. На рис. 2.4 приведен пример ввода текста М-файла, который позволяет вычислить площадь основания и объем цилиндра по известным значениям радиуса основания и высоты.

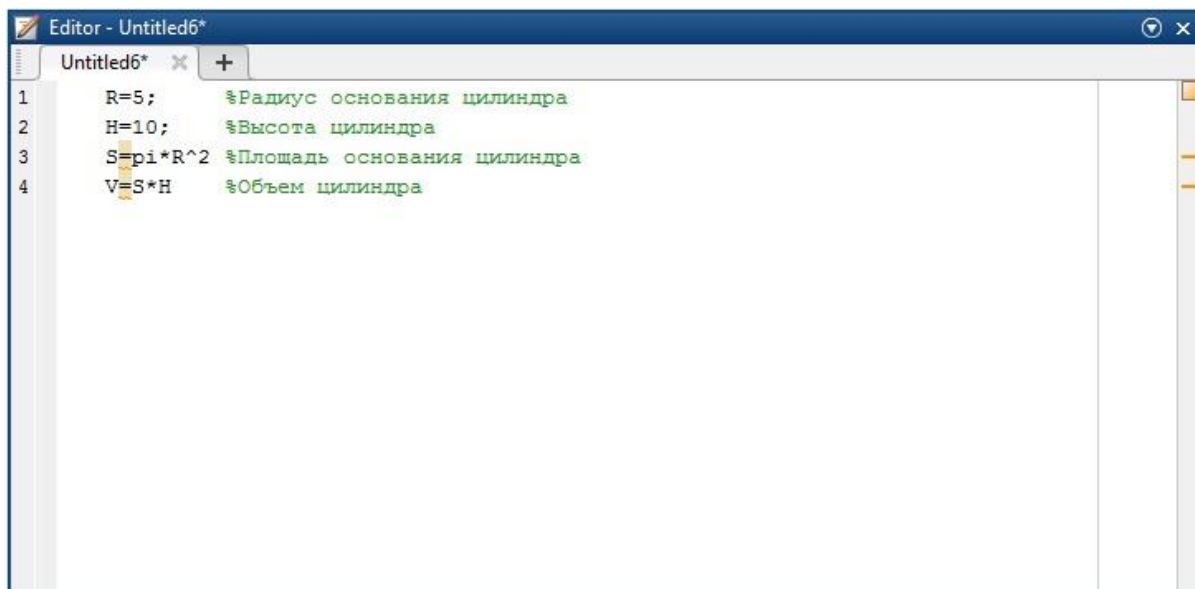


Рис. 2.4. Ввод текста М-файла

При вводе текста редактор использует цветовые выделения, перечисленные ниже:

- константы, переменные, функции MATLAB – черный цвет;
- операторы языка программирования – синий цвет;
- текстовые комментарии после знака % – зеленый цвет;
- символы, вводимые в апострофах, – сиреневый цвет;
- синтаксические ошибки – красный цвет.

2.2.3. Сохранение М-файлов

После ввода текста М-файла его необходимо сохранить. Для сохранения можно использовать:

- команду меню редактора (рис. 2.1, б) **Save**;
- комбинацию клавиш **<Ctrl> + <S>**;
- набрав в командной строке команду **edit <имя файла>** и нажав на **<Enter>**.

После появления окна **Save file as** следует ввести имя файла и нажать на кнопку **Сохранить**.

Имя файла должно начинаться с буквы, не может содержать спецсимволы и символы кириллицы, не может состоять из одних лишь цифр и не может повторять имена встроенных функций и операторов системы MATLAB.

Например, файл, представленный на рис. 2.4, можно сохранить с именем **Programma1** (рис. 2.5).

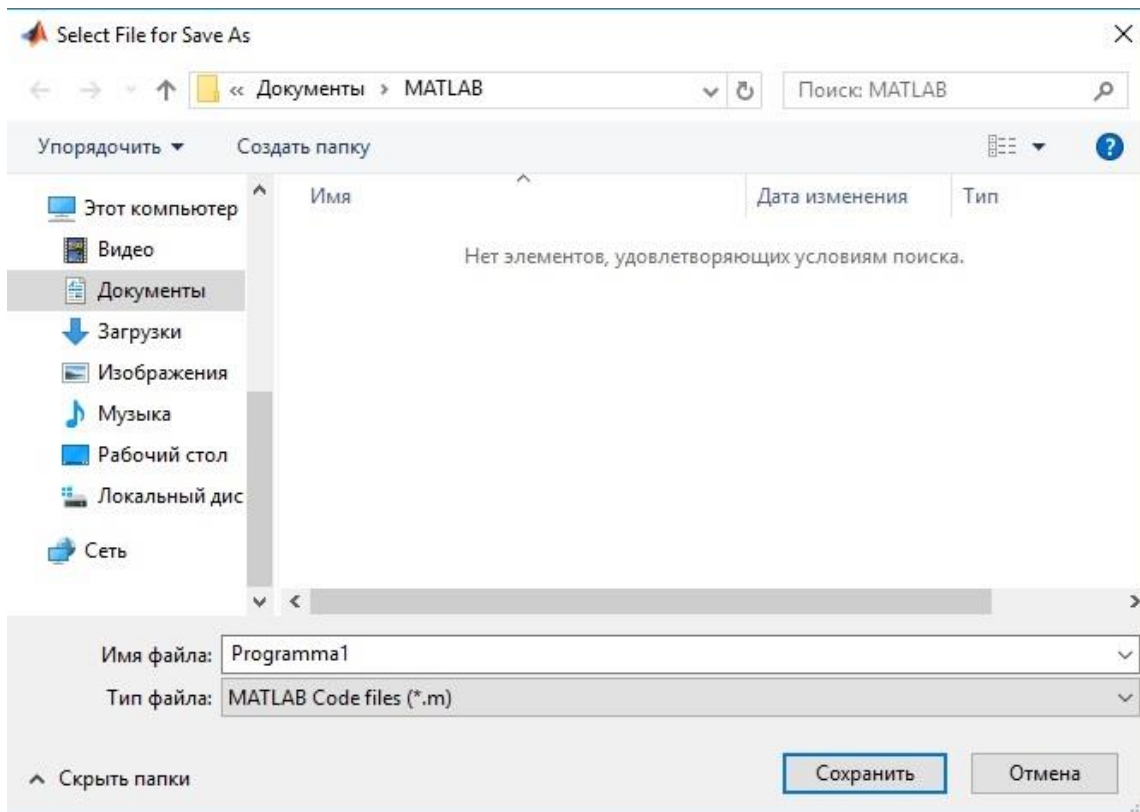


Рис. 2.5. Окно сохранения М-файла

2.2.4. Запуск М-файлов для исполнения

Для исполнения файла можно выполнить одно из следующих действий:

- вызвать команду **Run** из меню редактора;
- набрать имя М-файла в командной строке и нажать на клавишу **<Enter>**;
- нажать на клавишу **<F5>** на клавиатуре.

Результаты выполнения М-файла или сообщения об ошибках появятся в рабочей области. В результате выполнения файла **Programma1** в рабочей области (рис. 2.6) появились значения площади основания и объема цилиндра.

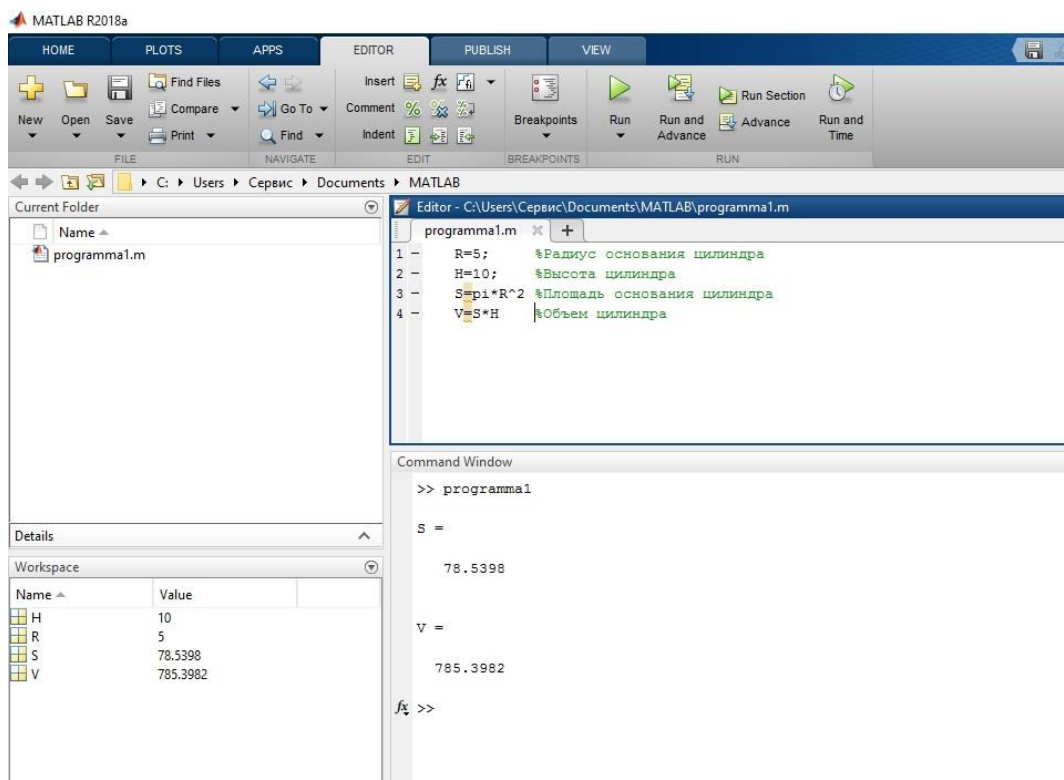


Рис. 2.6. Результаты выполнения М-файла

Задания для самостоятельного выполнения

1. Создать М-файл сценарий `Programma1`, записать в нем последовательность действий для вычисления площади основания и объема цилиндра. Посмотреть, какие переменные сохранились в окне `Workspace`, где можно увидеть результат выполнения программы.

2. Разработать М-файл сценарий вычисления значения выражения в заданном диапазоне $x \in [x_1; x_2]$ в m точках и построения графика.

Номер задания	x_1	x_2	m	$f(x)$
1	0,2	5	20	$\sin(x)$
2	1	10	30	$\cos(x)$
3	0,3	3	40	$\exp(x)$
4	0,4	4	50	$\ln(1+x)$
5	0,5	5	30	$\ln(x)$
6	0,6	6	40	$\ln(x)$
7	0,7	7	50	$\ln(x)$
8	0,8	8	45	$\ln(x+a)$
9	1,1	11	40	$\text{ctg}(x)$

Номер задания	x_1	x_2	m	$f(x)$
10	1,2	12	50	$\operatorname{cosec}(x)$
11	1,3	13	50	$\operatorname{cosec}(x)$
12	1,4	14	60	$\operatorname{arctg}(x)$
13	1,5	15	45	$\operatorname{arctg}(x)$
14	1,6	16	40	$\ln(x)$
15	0,9	9	50	$\sin(x)$

2.3. Основы программирования в MATLAB

Все М-файлы можно разделить на два типа: файлы-сценарии и файлы-функции. Каждый тип файла имеет свои характеристики.

Файл-сценарий (М-сценарий) – это внешний файл, который представляет собой зафиксированную последовательность команд, операторов и функций без входных и выходных аргументов. Такой файл позволяет автоматизировать выполнение последовательностей команд, которые в ином случае должны были бы многократно вводиться из командной строки. Файл-сценарий оперирует данными из рабочей области, а также может создавать новые данные для последующей обработки в самом файле. Файл-сценарий не возвращает выходных аргументов, однако все переменные, которые используются в файле, являются глобальными, они сохраняются в рабочей области и могут быть использованы после выполнения сценария для дальнейших вычислений. Имена файлов-сценариев не разрешается использовать в качестве аргументов функций или операндов в выражениях.

Файл-сценарий включает следующие компоненты:

- основной комментарий (первая строка комментария);
- дополнительный комментарий;
- тело файла (команды, операторы, функции).

Структуру файла-сценария можно представить в виде:

```
% Основной комментарий
% Дополнительный комментарий
Тело файла
```

Основной комментарий определяет назначение файла-сценария. Если ввести ключевое слово из основного комментария после команды `lookfor` в командной строке, то система выведет имя файла, в котором находится данный комментарий.

Дополнительный комментарий выводится на экран вместе с основным при использовании команды `help<имя файла>`.

Команда `type <имя файла>` выводит на экран полный текст программы.

Если использовать команду `help <имя каталога>`, то в командном окне будут выведены имена файлов каталога и первые строки комментариев файлов.

В качестве примера рассмотрим файл-сценарий с именем `программа2.m` для построения графика функции $y = x^2$ в интервале $[x_n, x_k]$, причем x_n и x_k программа получает из рабочей области.

```
%Парабола
%График на интервале [xn, xk]
%с нанесением масштабной сетки
x=xn:0.1:xk;
y=x.^2;
plot(x,y)
grid
```

Чтобы запустить указанный файл, необходимо сначала задать в командной строке переменные `xn` и `xk`, например:

```
>> xn=- 4; xk=4;
```

Тогда в результате работы программы получим график, представленный на *рис. 2.7*.

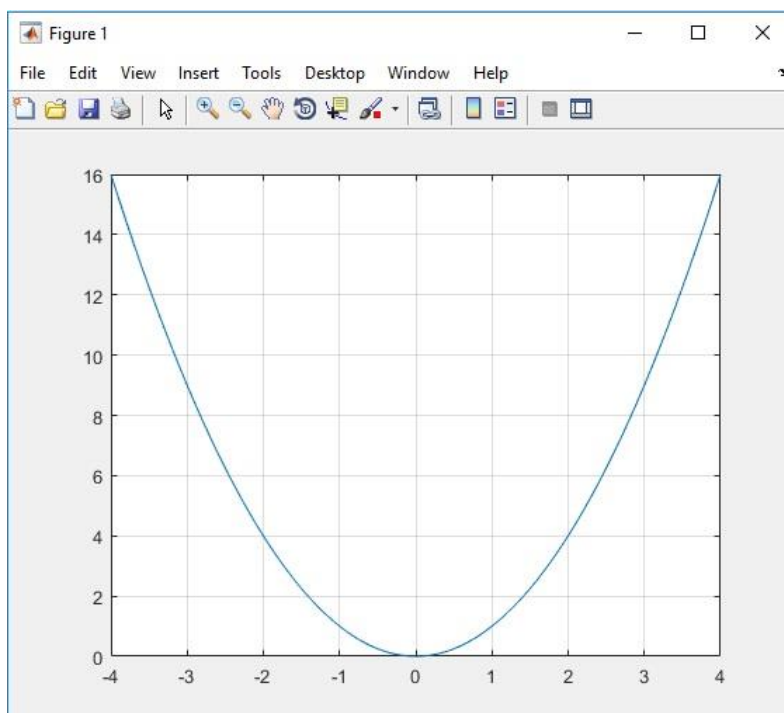
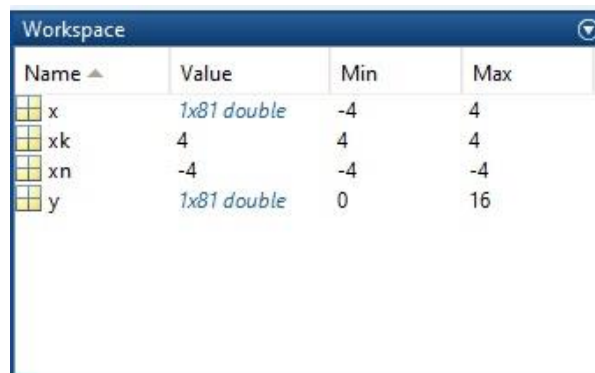


Рис. 2.7. Результаты выполнения М-файла

Переменные, которые используются в файле, сохраняются в рабочей области и отображаются в окне *Workspace* (рис. 2.8).



Name	Value	Min	Max
x	1x81 double	-4	4
xk	4	4	4
xn	-4	-4	-4
y	1x81 double	0	16

Рис. 2.8. Содержание окна *Workspace*

Эти переменные могут быть использованы после выполнения программы для дальнейших вычислений, например, для определения максимального и минимального значений функции *y*:

```
>> m=max(y)
m =
    16
>> m1=min(y)
m1 =
     0
>>
```

Результаты использования команд *lookfor*, *help* и *type* для файла *программа2.m* представлены ниже:

```
>> lookfor Парабола
программа2                - Парабола
>> help программа2
Парабола
График на интервале [xn, xk]
с нанесением масштабной сетки
>> type программа2
%Парабола
%График на интервале [xn, xk]
%с нанесением масштабной сетки
x=xn:0.1:xk;
y=x.^2;
plot(x,y)
grid
>>
```

M-файлы можно оформлять как отдельные функции. В этом случае имя функции должно совпадать с именем файла, в котором она хранится.

Файл-функция (M-функция) является самостоятельным программным модулем, который общается с другими модулями через входные и выходные аргументы. Самостоятельное написание пользователем файлов-функций позволяет расширить возможности системы MATLAB. Файлы-функции используются так же, как обычные встроенные функции системы. Имена файлов-функций можно применять при записи арифметических выражений.

Возможность добавления новых функций обеспечивает системе MATLAB важное свойство расширяемости. Новые функции получают расширение .m и становятся доступными наряду со встроенными функциями.

Файлы-функции допускают наличие входных и выходных аргументов. Они работают с переменными в пределах собственной рабочей области, отличной от рабочей области системы MATLAB. По умолчанию внутренние переменные являются локальными по отношению к функции.

Файл-функция включает следующие компоненты:

- строка определения функции;
- основной комментарий (первая строка комментария);
- дополнительный комментарий;
- тело функции.

Структура файла-функции выглядит следующим образом:

```
function[<выходные переменные>]=<имя функции>( <входные переменные>)  
% Основной комментарий  
% Дополнительный комментарий  
Тело функции
```

Строка определения функции является первой строкой файла-функции. В ней задается имя функции, входные и выходные аргументы.

Основной и дополнительный комментарии используются аналогично соответствующим компонентам файла-сценария с командами `lookfor` и `help`.

Тело функции реализует вычисления и присваивает значения выходным аргументам.

Если функция имеет *один* выходной параметр, то ее структура может быть представлена следующим образом:

```
function P=name(Входные параметры)  
% Основной комментарий  
% Дополнительный комментарий  
Тело функции
```

P=Выражение

Здесь P – имя переменной – выходного параметра; name – имя самой функции; P=Выражение – конструкция, которая вводится, если требуется, чтобы функция возвращала результат вычислений.

Примером функции с одним выходным параметром может служить функция `summa` для определения суммы чисел от 1 до n:

```
function S=summa(n)
%Суммирование чисел
%Вычисление суммы чисел
%от 1 до n
S=sum(1:n);
```

Результат использования функции:

```
>> summa(5)
ans =
    15
>>
```

Функция возвращает свое значение и может использоваться в виде `name(Входные параметры)` в математических выражениях, например:

```
>> n=5*summa(4)+10
n =
    60
>>
```

Если выходных параметров несколько (P1,P2,...), то структура функции имеет вид:

```
function [P1,P2,...]=name(Входные параметры)
% Основной комментарий
% Дополнительный комментарий
Тело функции
P1=Выражение
P2=Выражение
....
```

Такую функцию нельзя использовать непосредственно в математических выражениях, поскольку она возвращает не единственный результат, а несколь-

ко результатов – по числу выходных параметров. Данная функция используется как отдельный элемент в виде:

```
[P1,P2,...]=name(Входные параметры)
```

При этом выходные переменные становятся определенными и их можно использовать в дальнейших расчетах. Если функция используется в виде `name(Входные параметры)`, то возвращается значение только первого выходного параметра `P1`.

Примером функции с несколькими выходными параметрами может служить функция, определяющая площадь основания и площадь боковой поверхности цилиндра. Входными параметрами при этом являются радиус основания и высота цилиндра:

```
function [so,sb]=cilindr(r,h)
%Расчет параметров цилиндра
%Вычисление площади основания
%и площади боковой поверхности
so=pi*r^2;
sb=2*pi*r*h;
```

При обращении к функции из командной строки получим:

```
>> [so,sb]=cilindr(2,3)
so =
    12.5664
sb =
    37.6991
>>
```

Как отмечалось выше, выходные переменные данной функции можно использовать для дальнейших расчетов, например для определения полной площади поверхности цилиндра:

```
>> sp=so+sb
sp =
    50.2655
>>
```


Задания для самостоятельного выполнения

1. Создать файл-сценарий для построения графиков функции $y = \frac{x^2}{1 + 0,25\sqrt{x}}$ на интервале $x \in [x_n; x_k]$. Начальное $x_n = 1$ и конечное $x_k = 10$

значения для аргумента функции задать в Command Window. На график нанести сетку, название, подписи к осям. Посмотреть, какие переменные сохранились в окне Workspace. В командном окне определить максимальное и минимальные значения функции y .

2. Воспользоваться командами lookfor, help и type для файла программа2. Результат записать в тетрадь.

3. С использованием файла-функции summa вычислить в Command Window:

$$\left(\sum_{k=1}^{10} k - 13\frac{7}{9} \right) \frac{6}{11} + 2,2 : (0,241 - 0,91),$$

$$\lg^2 x - \left(\frac{27}{8} \right)^{x-1} \sin \sqrt{\sum_{n=1}^6 n}.$$

4. Записать файл-функцию для определения площади основания и площади боковой поверхности цилиндра. Входными параметрами считать радиус основания и высота цилиндра. В Command Window вычислить полную площадь поверхности цилиндра.

5. Разработать М-файл функцию для вычисления заданного выражения (см. таблицу с заданиями на с. 74). С использованием разработанной функции вычислить значения в диапазоне $x \in [x_1; x_2]$ в m точках и построить график.

2.4. Вычисление функции по заданному имени

Имя функции можно передавать на вычисление в качестве строки. Для таких вычислений используется функция feval, которая может быть представлена в двух вариантах.

Первый вариант записи функции feval позволяет передавать вызываемой функции входные аргументы x_1, \dots, x_n и возвращает один выходной аргумент:

```
feval(<Имя функции>, x1, ..., xn)
```

Подобная запись может быть использована при вычислении ранее рассмотренной функции summa:

```
>> s=feval('summa',5)
s =
    15
```

```
>>
```

или при вычислении других функций системы MATLAB, например:

```
>> feval('sin', 3*pi/2)
ans =
    -1
>>
```

Второй вариант записи функции `feval` возвращает множество выходных переменных y_1, \dots, y_n и имеет вид:

```
[y1,...,yn]=feval(<имя функции>, x1,...,xn)
```

Такая запись может быть использована, например, для передачи на вычисление функции `cilindr`, приведенной выше:

```
>> [so, sb]=feval('cilindr', 2, 3)
so =
    12.5664
sb =
    37.6991
>>
```

2.5. Использование подфункций и вложенных функций

В файле с расширением `.m` кроме основной функции, имя которой совпадает с именем файла, могут содержаться *подфункции*. Они являются видимыми только для основной функции и других подфункций этого файла. То есть заданные в некотором файле подфункции нельзя использовать ни в командном режиме работы, ни в других М-файлах. Благодаря локальному действию подфункций их имена могут совпадать с именами основных функций системы.

Подфункции записываются друг за другом после основной функции и имеют структуру, идентичную основной функции. Каждая подфункция имеет собственный заголовок. Основная функция выполняется первой. Подфункции могут вызываться в любом порядке.

Общую структуру функции можно записать следующим образом:

```
function [P1,P2,...]=name(Входные параметры)
%Начало основной функции
% Основной комментарий
% Дополнительный комментарий
```

```

Тело основной функции
end
% Конец основной функции

function[x1,x2,...]=name1(Входные параметры)
%Начало 1-й подфункции
Тело первой подфункции
end
%Конец 1-й подфункции

.....

function[xn1,xn2,...]=nameN(Входные параметры)
%Начало n-й подфункции
Тело n-й подфункции
end
%Конец n-й подфункции

```

Можно завершать любую функцию командой `end`, однако, в большинстве случаев это не обязательно. Команда `end` требуется только в **M-файлах**, которые имеют одну или более *вложенных* функций (см. ниже). В таких файлах каждая функция должна завершаться командой `end`. Если функция содержит условный оператор, который завершается командой `end`, то аналогичной командой нужно завершить и саму функцию во избежание ошибок.

Пример. Требуется вычислить значение выражения $f = x \cdot \sin x - 1$ для значения $x = 3\pi / 2$, воспользовавшись встроенной функцией `feval`. Для вычисления создать основную функцию, само выражение записать в теле вложенной подфункции. Текст программы выглядит следующим образом:

```

function F = primer(x)
% использование вложенных подфункций
% вычисление значений выражения с применением встроенной
функции feval
F=feval('fun',x);
end

function f = fun(x)
f=x*sin(x)-1;
end

```

Запуск программы производится в окне **Command Window**:

>> F = primer(3*pi/2)

В результате выполнения программы получим

F =

-5.7124

Задание для самостоятельного выполнения

Создать файл-функцию для вычисления следующих выражений. Параметры, заданные буквенно, использовать в качестве входных аргументов файла-функции:

Алгебраические выражения	
1.	$f_1 = \cos x + 2 \operatorname{tg} \frac{3\pi}{4} - 0,3 \lg x_1 + 2^{x_1}; f_2 = 4 \arcsin \frac{1}{2} - \sqrt{2x_1} + \operatorname{ctg} \alpha;$ $f_3 = \operatorname{arctg} \sqrt{3} + e^2 + 2 \ln x_1 - \frac{\sin 3x}{\sqrt{3}}; \text{при } x = \frac{2\pi}{3}, x_1 = 8, \alpha = \frac{\pi}{2}$
2.	$s_1 = \sin b - \cos b_1 + 0,2\sqrt{a_1} + 3e^{a_1}; s_2 = 3 \operatorname{ctg} \varphi - 6 \lg 100 + \lg 16 + 3^{a_1};$ $s_3 = \arccos \frac{1}{2} + 4 \operatorname{arctg}(-1) - \frac{\operatorname{arctg} \sqrt{3}}{2}; \text{при } b = \frac{5\pi}{6}, b_1 = \frac{7\pi}{4}, a_1 = 5, \varphi = \frac{3\pi}{4}$
3.	$p_1 = \sin 2\beta + \cos \frac{4\pi}{3} + 0,5 \operatorname{tg} \beta_1 + \frac{\operatorname{ctg} \frac{3\pi}{4}}{3}; p_2 = -\ln 16 + \lg k + e^3 + 2\sqrt{2k};$ $p_3 = \arcsin \frac{\sqrt{3}}{2} - \operatorname{arctg} 1 + \operatorname{arctg}(-\sqrt{3}) + k^2; \text{при } \beta = \frac{\pi}{3}, \beta_1 = \frac{\pi}{4}, k = 32$
4.	$x_0 = \sqrt{a+2} + a^{0,5} - \frac{1}{e^{-2}} + \lg a_1 - \lg a + 1,8 \ln 6;$ $x_1 = \arccos \frac{\sqrt{2}}{2} - 1,3 \arcsin \frac{\sqrt{3}}{2} + \operatorname{arctg}(-\sqrt{3}) + \frac{\operatorname{arctg}(-1)}{(a-5)};$ $x_2 = \sin \frac{3\pi}{4} + 2 \cos \alpha - \operatorname{tg} \frac{5\pi}{6} + 3,5 \operatorname{ctg} \frac{7\pi}{4}; \text{при } a = 14, a_1 = 1000, \alpha = \frac{7\pi}{6}$

Алгебраические выражения

5. $y = 1,5 \operatorname{arctg} x_1 - \arcsin \frac{x_1}{2} + 10 \arccos(-1);$

$$y_1 = 3 \operatorname{tg} \alpha + 1,8 \operatorname{ctg} \frac{\alpha}{3} - \sin \frac{5\pi}{6} - \cos \frac{4\pi}{3};$$

$$y_2 = \sqrt{x_2 + 4} + e^3 - \lg 32 + \ln x_2 + 4^5; \text{ при } x_1 = \sqrt{3}, \alpha = \frac{3\pi}{4}, x_2 = 60$$

6. $a = \arcsin \frac{\sqrt{2}}{2} + \operatorname{arctg} 2b - \operatorname{arcctg} \frac{1}{\sqrt{3}} + 2 \arccos \frac{1}{2};$

$$a_1 = m^{\frac{2}{3}} - e^{-3} - \sqrt{2+b} + 1,4 \lg 100 - 4,7 \ln 12 - \ln 64;$$

$$a_2 = -2 \operatorname{tg} \frac{\pi}{6} + 12 \cos \beta - 3 \sin \beta + \operatorname{ctg} \frac{4\pi}{3}; \text{ при } b = \frac{1}{2}, m = \frac{1}{8}, \beta = \frac{7\pi}{6}$$

7. $q = \sqrt{2k_1} + (k_1 - 3)^{\frac{3}{4}} - e^{-2} + \cos \varphi - 3,4 \sin \frac{5\pi}{6};$

$$n = \lg 128 - 10 \ln 15 - \lg k_2 + \frac{\operatorname{tg} \varphi}{4} - \operatorname{ctg} \varphi;$$

$$m = \arcsin \frac{\sqrt{3}}{2} + \arccos k_3 - \operatorname{arctg} \sqrt{3}; \text{ при } k_1 = 16, \varphi = \frac{3\pi}{4}, k_2 = 500, k_3 = -\frac{\sqrt{2}}{2}$$

8. $z = -2(\sqrt{m} + e^2) - m^{\frac{3}{8}} + \lg(6m + 10) - \lg 32;$

$$z_1 = \operatorname{arctg}(-\sqrt{3}) - \arcsin\left(-\frac{1}{2}\right) + 3 \arccos\left(-\frac{\sqrt{3}}{2}\right);$$

$$z_2 = \cos \alpha_1 - 0,4 \sin \frac{\pi}{3} + \operatorname{tg} \frac{2\pi}{3} + \operatorname{ctg} \alpha_2; \text{ при } m = 15, \alpha_1 = \frac{\pi}{6}, \alpha_2 = \frac{5\pi}{6}$$

9. $g = \frac{3\sqrt{x} + 4}{e^x} + \lg 50x - x^{\frac{4}{5}} + 1,8 \lg x_1 - \ln x_2;$

$$g_1 = \arcsin \frac{\sqrt{2}}{2} - \arccos\left(-\frac{\sqrt{2}}{2}\right) + \operatorname{arctg} \frac{1}{\sqrt{3}} + 2 \operatorname{arcctg} \sqrt{3};$$

$$g_2 = \sin \frac{3\pi}{4} + \frac{\cos \beta}{2x} - \operatorname{tg} \frac{\pi}{3} + 2 \operatorname{ctg} \frac{3\pi}{4}; \text{ при } x = 2, x_1 = 8x, x_2 = 5x, \beta = \frac{7\pi}{6}$$

Алгебраические выражения

10. $m = 0,8 \sin 3x_1 + \frac{\operatorname{tg} \frac{\pi}{4}}{\operatorname{ctg} x_2} + \cos x_2 - (3\sqrt{x_3} + \lg x_3)^{\frac{1}{5}};$

$$m_1 = \operatorname{arctg} \sqrt{3} - \arcsin \left(-\frac{1}{2} \right) + \frac{\arccos 0,5}{\operatorname{arctg}(-1)};$$

$$m_2 = -e^{-4} + \lg x_3 - 1,5 \lg(10x_3); \text{ при } x_1 = \frac{\pi}{2}, x_2 = \frac{7\pi}{6}, x_3 = 8$$

11. $a = \operatorname{arctg} \sqrt{3} + e^2 + 2 \ln x_1 - \frac{\sin 3x}{\sqrt{3}}; a_1 = \cos x + 2 \operatorname{tg} \frac{3\pi}{4} - 0,3 \lg x_1 + 2^{x_1};$

$$a_2 = 4 \arcsin \frac{1}{2} - \sqrt{2x_1} + \operatorname{ctg} \alpha; \text{ при } x = \frac{2\pi}{3}, x_1 = 8, \alpha = \frac{\pi}{2}$$

12. $f = \arccos \frac{1}{2} + 2 \operatorname{arctg}(-1) - \frac{\operatorname{arctg} \sqrt{3}}{2}; f_1 = 3 \operatorname{ctg} \varphi - 6 \lg 100 + \lg 16 + 3^{a_1};$

$$f_2 = \sin b - \cos b_1 + 0,5 \sqrt{a_1} + 6e^{a_1}; \text{ при } b = \frac{5\pi}{6}, b_1 = \frac{7\pi}{4}, a_1 = 5, \varphi = \frac{3\pi}{4}$$

13. $d = \lg k - \ln 16 + e^3 + 2\sqrt{2k}; d_1 = \sin 2\beta + \cos \frac{4\pi}{3} + 0,4 \operatorname{tg} \beta_1 + \frac{\operatorname{ctg} \frac{3\pi}{4}}{5};$

$$d_2 = \arcsin \frac{\sqrt{3}}{2} - \operatorname{arctg} 1 + \operatorname{arctg}(-\sqrt{3}) + k^2; \text{ при } \beta = \frac{\pi}{3}, \beta_1 = \frac{\pi}{4}, k = 32$$

14. $x_1 = \frac{\operatorname{arctg}(-1)}{(a-5)} + \arccos \frac{\sqrt{2}}{2} - 1,3 \arcsin \frac{\sqrt{3}}{2} + \operatorname{arctg}(-\sqrt{3});$

$$x_2 = \lg a_1 - \lg a + 1,8 \ln 6 + \sqrt{a+2} + a^{0,5} - \frac{1}{e^{-2}};$$

$$x_3 = \sin \frac{3\pi}{4} + 2 \cos \alpha - \operatorname{tg} \frac{5\pi}{6} + 3 \operatorname{ctg} \frac{7\pi}{4}; \text{ при } a = 14, a_1 = 1000, \alpha = \frac{7\pi}{6}$$

Алгебраические выражения

15. $y_1 = 3 \operatorname{tg} \alpha + 1,8 \operatorname{ctg} \frac{\alpha}{3} - \sin \frac{5\pi}{6} - \cos \frac{4\pi}{3};$

$$y_2 = 1,5 \operatorname{arctg} x_1 - \arcsin \frac{x_1}{2} + 10 \operatorname{arccos}(-1);$$

$$y_3 = \sqrt{x_2 + 4} + e^3 - \lg 32 + \ln x_2 + 4^5; \text{ при } x_1 = \sqrt{3}, \alpha = \frac{3\pi}{4}, x_2 = 60$$

2.6. Локальные и глобальные переменные

Переменные, используемые в *файлах-сценариях*, являются *глобальными*, то есть действуют одинаково внутри *файла-сценария* и в рабочей области. Поэтому заданные в рабочей области переменные могут использоваться в теле файла, а переменные, заданные в файле, доступны из рабочей области.

Переменные, указанные в списке параметров *функции*, являются *локальными*. Они используются для переноса значений, которые подставляются на их место при вызовах *функций*. Такие переменные изолированы от рабочей области и от переменных других *функций*.

В MATLAB существует возможность объявить необходимые переменные глобальными для нескольких функций и сценариев, чтобы все модули могли использовать эти переменные.

Чтобы объявить переменные файла глобальными, используется команда

```
global имя1 имя2....
```

Чтобы несколько файлов могли совместно использовать переменную, ее нужно объявить глобальной в каждом из этих файлов.

В приведенную ранее функцию для вычисления площади основания и площади боковой поверхности цилиндра можно внести изменения, которые продемонстрируют использование глобальных переменных. Например, можно объявить высоту цилиндра h глобальной переменной. Функция изменит вид:

```
function [so,sb]=cilindr(r)
%Расчет параметров цилиндра
%Вычисление площади основания
%и площади боковой поверхности
global h
so=pi*r^2;
sb=2*pi*r*h;
```

Теперь переменную h можно задать в рабочей области, и она станет доступной для использования внутри функции:

```
>> global h
>> h=3;
>> [so, sb]=cilindr(2)
so =
    12.5664
sb =
    37.6991
>>
```

Еще один пример. Создаем файл-функцию:

```
function v=cilindr(h)
global s
%Вычисление объема цилиндра
v=s*h;
```

Далее в **Command Window** набираем:

```
>> global s
>> r=2;s=pi*r^2
s =
    12.5664
>> v=cilindr(3)
v =
    37.6991
>>
```

Задания для самостоятельного выполнения

1. Создать файл-функцию для построения графика функции $y = \sin \alpha \cdot \cos \beta + \frac{10x^2}{1 + 0,25\sqrt{x}}$ на интервале $x \in [0;10]$, при $\alpha = 30^\circ$, $\beta = 60^\circ$. Значения для углов α и β задать в **Command Window** и объявить их глобальными.

2. Написать программу для нахождения корней квадратного уравнения $ax^2 + bx + c = 0$, используя файл-функцию. Выражение для вычисления дискриминанта записать во вложенной подфункции. Обращение к подфункции выполнять с помощью встроенной функции `feval`.

2.7. Операторы языка программирования

В этом разделе рассмотрим основные операторы языка программирования и примеры их использования.

2.7.1. Организация диалогового ввода

При разработке программ можно организовать ввод исходных данных в диалоговом режиме. Для этого следует использовать функцию `input`.

В общем виде функцию можно записать

```
a = input('Текст запроса')
```

Рассмотрим использование функции `input` на примере программы построения графика $y = \sin(x)/x$. Пусть начальное значение аргумента, шаг изменения и конечное значение аргумента задаются пользователем в диалоговом режиме. Тогда текст программы будет иметь вид:

```
%Построение графика функции y(x)
x1=input('Начальное значение аргумента x1=');
x2=input('Конечное значение аргумента x2=');
h=input('Шаг изменения аргумента h=');
x=x1:h:x2;
y=sin(x) ./x;
plot(x,y)
```

После запуска программы в рабочей области появляется запрос в виде строки: Начальное значение аргумента $x1=$. При этом происходит остановка работы программы и ожидается ввод численного значения параметра. После ввода необходимо нажать на клавишу `<Enter>`, чтобы введенное число было присвоено переменной $x1$. Затем появляются аналогичные запросы для значений $x2$ и h . То есть работа с данной программой будет выглядеть следующим образом:

```
Начальное значение аргумента x1=-20
Конечное значение аргумента x2=20
Шаг изменения аргумента h=0.1
>>
```

После ввода требуемых значений в графическом окне (рис. 2.10) появится график функции $y(x)$.

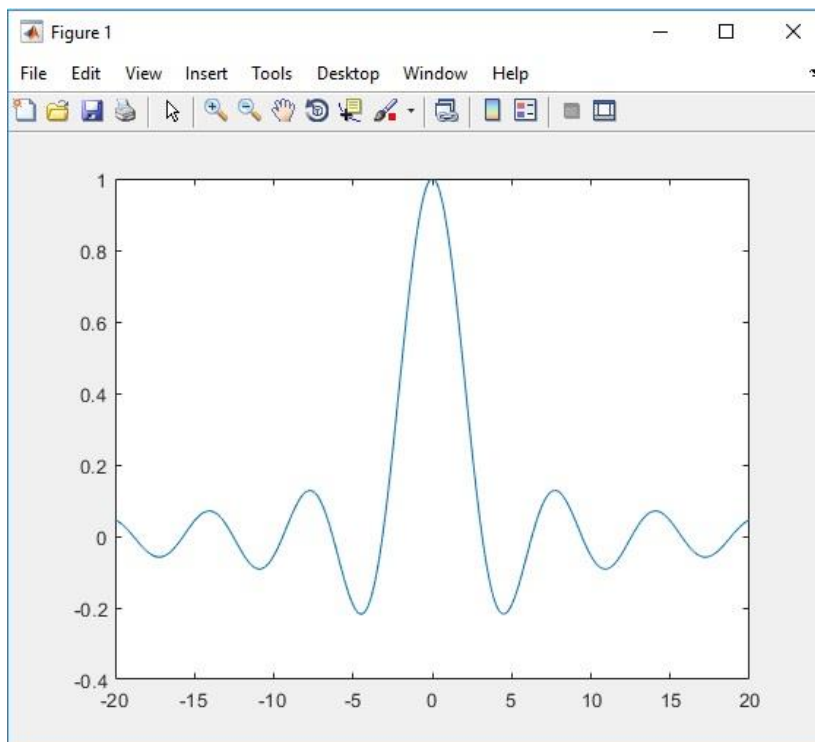


Рис. 2.10. График функции $y = \sin(x)/x$

Функция `input` может использоваться и для ввода выражений в символьном виде. При этом запись функции меняется:

```
a = input('Текст запроса','a')
```

При выполнении функции необходимо ввести строковое выражение, которое функция возвращает.

```
>> v = input('Введите выражение','v')
Введите выражение cos(2)/3
v =
cos(2)/3
>>
```

Существует функция `eval`, которая позволяет вычислить выражение, полученное от функции `input` в символьном виде:

```
>> p = eval(v)
p =
-0.1387
>>
```

2.7.2. Организация вывода результатов счета

Одним из наиболее простых способов вывода в рабочую область результатов работы программы является отсутствие знака «;» после расчетного выражения. Другим способом вывода значения переменной является использование функции `disp`, которая имеет вид

```
disp(a)
```

Здесь `a` – имя переменной, значение которой необходимо вывести.

Функция `disp` может использоваться для вывода текста, при этом она записывается следующим образом:

```
disp('Текст для вывода').
```

В качестве примера использования функции `disp` рассмотрим программу, которая определяет максимальный и минимальный элементы, а также сумму и произведение элементов вектора-строки, вводимого пользователем.

```
V=input('Введите вектор-строку V=');  
disp('Минимальный элемент');m1=min(V)  
disp('Максимальный элемент');m2=max(V)  
S=sum(V);  
disp('Сумма элементов');disp(S)  
P=prod(V);  
disp(sprintf('Произведение элементов P=%d',P))
```

В данном примере функция `disp` применяется тремя способами, что позволяет по-разному представить результаты отдельных операций:

```
Введите вектор-строку V=[-1 2 0.5 4 10]  
Минимальный элемент  
m1 =  
    -1  
Максимальный элемент  
m2 =  
    10  
Сумма элементов  
    15.5000  
Произведение элементов P=-40  
>>
```

Для вывода произведения элементов дополнительно использована функция `sprintf`, которая преобразовывает данные в строки символов в соответствии

с указанным форматом (см. ниже *табл. 2.1*) и возвращает их в виде строковой переменной.

Наряду с функцией `sprintf` для вывода форматированных данных используется функция `fprintf`. Эта функция преобразовывает данные в строки символов в соответствии с указанным форматом и выводит их на экран или в файл.

Чтобы вывести данные *на экран* при помощи функции `fprintf`, функцию нужно записать в виде

```
fprintf('Выходная строка с форматом', A)
```

Здесь *A* – выводимые данные.

Выходная строка должна содержать формат, который определяет способы записи, выравнивания, количество значащих цифр, ширину поля для вывода данных, символы перевода строки и другие характеристики. Образец записи формата представлен в *табл. 2.1*.

Т а б л и ц а 2.1. Запись формата вывода

%	+	10.5	g	\n
Начальный символ	Флаг	Ширина поля и точность	Способ записи	Перевод строки

Запись формата начинается с символа `%`. Затем может следовать один из символов, представленных в *табл. 2.2*. Ширина поля представляет собой минимальное количество выводимых цифр числа, а точность – количество выводимых цифр после десятичной точки. Цифры, обозначающие ширину поля и точность, разделяются точкой. Возможные способы записи выводимых значений и символы перевода строки представлены в *табл. 2.2 – 2.4*.

Т а б л и ц а 2.2. Символы форматирования

Символ	Назначение	Пример
-	Выравнивание числа по полю	<code>%-4.2d</code>
+	Обязательный вывод знака числа	<code>%+4.2d</code>
Пробел	Добавление пробелов перед числом	<code>% 4.2d</code>
0	Дополнение нулями, а не пробелами	<code>%04.2d</code>

Т а б л и ц а 2.3. Способы записи выводимых значений

Запись	Описание
<code>%c</code>	Одиночный символ
<code>%d</code>	Десятичный формат со знаком
<code>%e</code>	Экспоненциальный формат вида 3.1415e+00
<code>%E</code>	Экспоненциальный формат вида 3.1415E+00
<code>%f</code>	Формат с фиксированной точкой

Запись	Описание
%g	Наиболее компактный из форматов %e и %f
%G	Наиболее компактный из форматов %E и %F
%o	Восьмеричная запись без знака
%s	Строка символов
%u	Десятичная запись без знака
%x	Шестнадцатеричная запись с символами a-f
%X	Шестнадцатеричная запись с символами A-F

Т а б л и ц а 2.4. Символы перевода строки

Символ	Описание
\n	Перевод на новую строку
\t	Горизонтальная табуляция
\b	Возврат на один символ
\r	Возврат каретки
\f	Перевод страницы
\\	Наклонная черта влево
\” или \”	Апостроф
%%	Символ процента

Ниже представлены четыре варианта вывода длины окружности, рассчитанной по заданному значению радиуса, при различных способах форматирования.

```
>> r = 8;
>> fprintf('Длина окружности радиуса %d равна
%d\n', r, 2*pi*r)
Длина окружности радиуса 8 равна 5.026548e+001

>> fprintf('Длина окружности радиуса %e равна
%e\n', r, 2*pi*r)
Длина окружности радиуса 8.000000e+000 равна
5.026548e+001

>> fprintf('Длина окружности радиуса %10.4f равна
%10.4f\n', r, 2*pi*r)
Длина окружности радиуса      8.0000 равна      50.2655

>> fprintf('Длина окружности радиуса %g равна
%g\t', r, 2*pi*r)
Длина окружности радиуса 8 равна 50.2655 >>
```

Функция `fprintf` может получать данные из массива. Тогда строка формата циклически применяется к элементам столбца, пока не будут исчерпаны все элементы. Например, задав массив **D**, который содержит радиусы окружностей и соответствующие им длины, можно получить отформатированные строки:

```
>> D=[1 2 3; 2*pi 2*pi*2 2*pi*3]
D =
    1.0000    2.0000    3.0000
    6.2832   12.5664   18.8496
>> fprintf('Длина окружности радиуса %f равна %f\n',
D)
Длина окружности радиуса 1.000000 равна 6.283185
Длина окружности радиуса 2.000000 равна 12.566371
Длина окружности радиуса 3.000000 равна 18.849556
>>
```

Функция `fprintf` может использоваться для преобразования данных в строки символов и записи их в файл. Для этого функция должна быть записана следующим образом

```
fprintf(F,'Выходная строка с форматом', A)
```

Здесь **F** – идентификатор файла – целое число, которое может быть получено с помощью функции `foren`. Оно имеет значение 1 для вывода на экран и 2 в случае стандартной ошибки. По умолчанию идентификатор файла равен 1.

При записи

```
V= fprintf(F,'Выходная строка с форматом', A)
```

в качестве выходного аргумента **V** возвращается количество записанных байт.

2.8. Условный оператор

Условный оператор в **MATLAB** может использоваться в нескольких формах. Простейшая форма оператора имеет вид

```
if Условие
    Действия
end
```

Такая запись оператора означает, что до тех пор, пока выполняется **Условие**, программа будет совершать **Действия**, составляющие тело условного оператора. **Условие** представляет собой логическое выражение вида

Операнд 1 Оператор отношения Операнд 2.

Операндами могут быть переменные, численные значения и выражения. В качестве операторов отношения используются: > (больше), < (меньше), == (равно), != (не равно), >= (больше или равно), <= (меньше или равно), а также (при наличии нескольких условий) & (И), | (ИЛИ).

Другая конструкция условного оператора:

```
if Условие
    Действия 1
else
    Действия 2
end
```

В данном случае в программе выполняются **Действия 1**, если справедливо **Условие**, иначе выполняются **Действия 2**.

Если при программировании недостаточно одного условия, можно воспользоваться расширенной формой условного оператора в виде

```
if Условие 1
    Действия 1
elseif Условие 2
    Действия 2
elseif Условие 3
    Действия 3
...
elseif Условие N
    Действия N
else
    Действия
end
```

Если **Условие 1** выполняется, то программа осуществляет **Действия 1**; иначе проверяется **Условие 2**, если оно выполняется, то осуществляются **Действия 2**; иначе проверяется **Условие 3** и т. д. Если ни одно из условий не является истинным, то программа выполняет действия, записанные после служебного слова **else**.

В качестве примера использования расширенной формы условного оператора рассмотрим программу, которая обеспечивает построение двух графиков функций в удобном для пользователя виде: если пользователь вводит вариант построения 1, то графики строятся в одних координатных осях, если вариант 2 – в разных окнах, если вариант 3 – в разных областях одного окна.

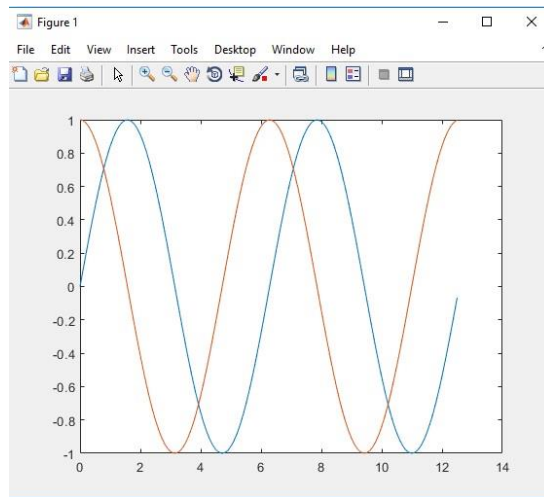
```
%Программа построения графиков функций
disp('Выберите вариант построения графиков:')
disp('1 - в одних координатных осях;')
disp('2 - в разных окнах;')
disp('3 - в разных областях одного окна.')
v=input('Введите вариант ');
x=0:0.1:4*pi; y=sin(x); z=cos(x);
if v==1
    plot(x,y,x,z)
elseif v==2
    plot(x,y);figure;plot(x,z)
elseif v==3
    subplot(1,2,1);plot(x,y)
    subplot(1,2,2);plot(x,z)
else
    disp('Неправильный номер варианта')
end
```

Результаты работы программы при выборе пользователем различных вариантов построения графиков представлены на *рис. 2.11*.

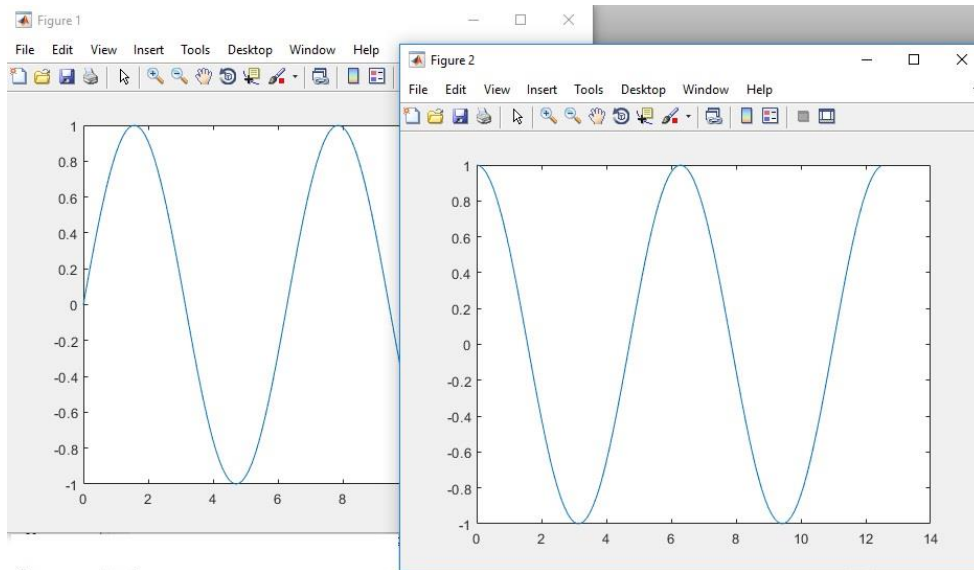
Задания для самостоятельного выполнения

1. Написать программу для нахождения корней квадратного уравнения. Если уравнение имеет два вещественных корня, вывести их пользователю на экран при помощи команды `disp`. Если уравнение имеет один вещественный корень, вывести его пользователю с сообщением о том, что уравнение имеет всего один корень. Если уравнение не имеет вещественных корней, вывести пользователю сообщение об ошибке.

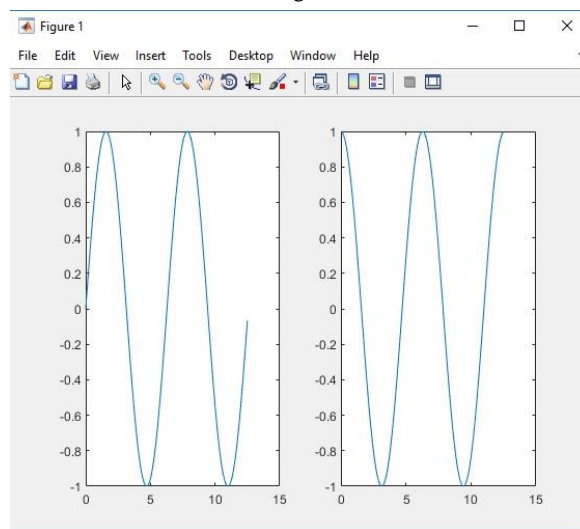
2. Проверить, пройдет ли кирпич со сторонами a , b , c в отверстие со сторонами n и k .



a



б



в

Рис. 2.11. Результаты работы программы:
a – вариант 1; *б* – вариант 2; *в* – вариант 3

2.9. Оператор альтернативного выбора

Для организации множественного выбора (разветвления) используется оператор `switch`, структура которого приведена ниже:

```
switch Параметр
  case Значение 1
    Действия 1
  case Значение 2
    Действия 2
  case Значение 3
    Действия 3
  ...
  otherwise
    Действия
end
```

Если **Параметр** равен одному из **Значений**, то выполняются соответствующие **Действия**, в противном случае выполняются **Действия**, записанные после служебного слова **otherwise**. Значение может представлять собой число, константу, переменную, вектор, строчную переменную.

Приведенную выше программу построения графиков функций можно написать с использованием оператора альтернативного выбора в виде

```
%Программа построения графиков функций
disp('Выберите вариант построения графиков:')
disp('1 - в одних координатных осях;')
disp('2 - в разных окнах;')
disp('3 - в разных областях одного окна.')
v=input('Введите вариант ');
x=0:0.1:4*pi; y=sin(x); z=cos(x);
switch v
  case 1
    plot(x,y,x,z)
  case 2
    plot(x,y);figure;plot(x,z)
  case 3
    subplot(1,2,1);plot(x,y)
    subplot(1,2,2);plot(x,z)
  otherwise
    disp('Неправильный номер варианта')
end
```

Задания для самостоятельного выполнения

1. Написать программу, которая по требованию пользователя вычисляла бы перемещение предмета при равномерном, равноускоренном и равнозамедленном движениях. Для каждого случая построить график зависимости перемещения от времени.

Равномерное движение	$s = s_0 + Vt$
Равноускоренное движение	$s = V_0t + \frac{at^2}{2}$
Равнозамедленное движение	$s = V_0t - \frac{at^2}{2}$

2. Если при равнозамедленном движении перемещение становится отрицательным, считать его равным нулю.

2.10. Операторы цикла

В языке программирования MATLAB используются два вида оператора цикла: оператор `for...end` с заданным числом повторений и оператор `while...end` с предусловием. Рассмотрим указанные операторы последовательно.

2.10.1. Оператор `for...end`

Данный оператор используется для организации вычислений с заданным числом повторений определенных действий. Конструкция оператора имеет вид:

```
for p=rp:h:pk
    Действия
end
```

Здесь `p` – имя переменной цикла, `rp` – начальное значение переменной цикла; `h` – шаг цикла (приращение переменной); `pk` – конечное значение переменной цикла, при достижении которого цикл завершается.

Если шаг цикла равен 1, то его можно не указывать:

```
for p=rp:pk
    Действия
end
```

Выполнение цикла начинается с присвоения переменной цикла начального значения ($p = pn$). Затем производится сравнение текущего значения переменной цикла p с ее конечным значением pk . Если $p \leq pk$, то выполняются действия в теле цикла, после чего переменная цикла меняет свое значение ($p = p+h$). Далее снова производится сравнение p и pk . Если $p > pk$, то цикл завершается, и выполняются следующие за ним действия.

Для досрочного прерывания выполнения цикла используется оператор `break`.

Примером использования оператора `for` может служить программа для определения количества положительных элементов вектора, вводимого пользователем.

```
%Определение количества положительных элементов вектора
v=input('Введите вектор ');
k=length(v); %Количество элементов вектора
s=0; %Счетчик положительных элементов
for i=1:k
    if v(i)>0
        s=s+1;
    end
end
disp('Количество положительных элементов: ');
disp(s)
```

Оператор `for` удобно использовать для организации циклов, в которых действия должны быть выполнены строго заданное число раз.

В качестве переменной цикла можно использовать массив. В этом случае число повторений цикла будет соответствовать числу столбцов массива. Например:

```
v=input('Введите массив ');
for i=v
    i
end
```

Здесь переменная цикла i на каждом шаге представляет собой вектор, соответствующий текущему столбцу массива \mathbf{v} :

```
Введите массив [4 5 9; -7 -8 0]
i =
```

```

    4
   -7
i =
    5
   -8
i =
    9
    0
>>

```

В качестве еще одного примера можно рассмотреть программу для определения экспериментальной вероятности выпадения количества орлов и решек. В программе используется встроенный оператор `rand(m,n)`, предназначенный для формирования массива размера $m \times n$, элементами которого являются случайные величины, распределенные по равномерному закону в интервале (0, 1). Текст программы:

```

% создаем серию экспериментов:
A=rand(1,1000);
% с помощью цикла и условного оператора определяем,
какой стороной падает монетка при каждом подбрасывании
for i=1:length(A)
if A(i)>0.5
    disp('решка')
else
    disp ('орел')
end
end
% общее количество выпавших "решек":
X=length(find(A>0.5))
% общее количество выпавших "орлов":
Y=length(find(A<0.5))

```

2.10.2. Оператор `while...end`

Данный оператор используется в виде:

```

while Условие
    Действия
end

```

До тех пор, пока выполняется заданное **Условие**, будут повторяться **Действия**, составляющие тело цикла. Прерывание выполнения цикла реализуется с помощью оператора **break**.

Приведенная выше программа для определения количества положительных элементов вектора, вводимого пользователем, может быть записана с использованием оператора **while** следующим образом:

```
%Определение количества положительных элементов вектора
v=input('Введите вектор');
k=length(v); %Количество элементов вектора
s=0; %Счетчик положительных элементов
i=1; %Начальное значение переменной цикла
while i<=k
    if v(i)>0
        s=s+1;
    end
    i=i+1;
end
disp('Количество положительных элементов:');
disp(s)
```

Задания для самостоятельного выполнения

1. Написать программу, имитирующую бросание кубика, для определения экспериментальной вероятности выпадения одной из шести его граней.

2. В соответствии с приведенной ниже таблицей выполнить:

— вычисление точных (используя стандартные функции **MATLAB**) значений соответствующей функции в диапазоне изменения аргумента от x_1 до x_2 в m равноотстоящих точках этого диапазона, включая его границы;

— вычисление по указанным в таблице степенным рядам приближенных значений функции в тех же точках, ограничиваясь l первыми членами ряда;

— расчет погрешности приближенного определения функции в каждой точке, сравнивая приближенное значение с точным, и построение графика зависимости погрешности от аргумента;

— вычисление приближенных значений функции в тех же точках с относительной погрешностью не более $\varepsilon = 0,001$; построение графика полученных относительных погрешностей.

Указанные действия записать в виде программы (**M**-файла).

Номер задания	x1	x2	m	r	$f(x)$	Приближенная формула
1	0,2	5	20	4	$\sin(x)$	$\sum (-1)^k \frac{x^{2k-1}}{(2k-1)!}$
2	1	10	30	5	$\cos(x)$	$1 - \sum (-1)^k \frac{x^{2k}}{(2k)!}$
3	0,3	3	40	5	$\exp(x)$	$1 + \sum \frac{x^k}{k!}$
4	0,4	4	50	4	$\ln(1+x)$	$\sum (-1)^{k-1} \frac{x^k}{k}$
5	0,5	5	30	3	$\ln(x)$	$\sum \frac{a^k}{k}, a = \frac{x-1}{x}$
6	0,6	6	40	4	$\ln(x)$	$\sum (-1)^k \frac{a^k}{k}, a = x-1$
7	0,7	7	50	5	$\ln(x)$	$2 \sum \frac{a^{2k-1}}{2k-1}, a = \frac{x-1}{x+1}$
8	0,8	8	45	6	$\ln(x+a)$	$\ln(a) + 2 \sum \frac{b^{2k-1}}{2k-1}, b = \frac{x}{2a+x}$
9	1,1	11	40	3	$\text{ctg}(x)$	$\frac{1}{x} + 2x \sum \frac{1}{x^2 - k^2 \pi^2}$
10	1,2	12	50	4	$\text{cosec}(x)$	$\sum \frac{1}{(x - k\pi)^2}$
11	1,3	13	50	5	$\text{cosec}(x)$	$\frac{1}{x} + 2x \sum \frac{(-1)^k}{x^2 + k^2 \pi^2}$
12	1,4	14	60	6	$\text{arctg}(x)$	$\sum (-1)^{k-1} \frac{x^{2k-1}}{2k-1}$
13	1,5	15	45	5	$\text{arctg}(x)$	$\frac{\pi}{2} + \sum (-1)^k \frac{1}{(2k-1)x^{2k-1}}$
14	1,6	16	40	4	$\ln(x)$	$\sum (-1)^k \frac{a^k}{k}, a = x-1$
15	0,9	9	50	6	$\sin(x)$	$x \prod \left(1 - \frac{x^2}{(k\pi)^2} \right)$

2.11. Создание паузы в вычислениях

Чтобы остановить работу программы, можно использовать оператор `pause`, который имеет следующие формы записи:

- `pause` – останавливает вычисления до нажатия любой клавиши;
- `pause(n)` – останавливает вычисления на n секунд;
- `pause off` – выключает режим пауз;
- `pause on` – включает режим пауз.

Например, добавление оператора `pause` в программу так, как показано ниже, позволит останавливать вычисления после обнаружения каждого положительного элемента до нажатия любой клавиши.

```
%Определение количества положительных элементов век-
тора
v=input('Введите вектор ');
k=length(v); %Количество элементов вектора
s=0; %Счетчик положительных элементов
i=1; %Начальное значение переменной цикла
while i<=k
    if v(i)>0
        i
        s=s+1
        pause
    end
    i=i+1;
end
disp('Количество положительных элементов:'); disp(s)
```

В процессе работы программы в командном окне будет отображаться номер положительного элемента i и текущее количество положительных элементов s :

```
Введите вектор [4 -5 6 -7]
i =
    1
s =
    1
i =
    3
s =
    2
Количество положительных элементов:
    2
```


2.12. КОНТРОЛЬНЫЕ ЗАДАНИЯ

2.12.1. Разработка программ в среде MATLAB; m-сценарии и m-функции

1. Разработать m-файл сценарий вычисления значения выражения в заданном диапазоне $x \in [x_1; x_2]$ в m точках и построения графика.

2. Разработать m-файл функцию для вычисления заданного выражения. С использованием разработанной функции вычислить значения в диапазоне $x \in [x_1; x_2]$ в m точках и построить график.

Номер задания	x_1	x_2	m	$f(x)$
1	1	10	50	$\cos(x)$
2	0,6	5	50	$\ln(x)$
3	-0,9	0,9	45	$\text{arcctg}(x)$
4	1	20	50	$\text{sh}(x)$
5	1	20	50	$\text{ch}(x)$
6	-0,9	0,9	50	$\text{arcth}(x)$
7	1	20	50	$\text{arcth}(x)$
8	-0,8	0,8	50	$\text{arcsin}(x)$
9	-0,8	0,8	50	$\text{arccos}(x)$
10	-5	5	50	$\exp(x)$

2.12.2. Разработка программ с использованием условных операторов и операторов цикла

В соответствии с приведенной ниже таблицей выполнить:

— вычисление точных (используя стандартные функции MATLAB) значений соответствующей функции в диапазоне изменения аргумента от x_1 до x_2 в m равноотстоящих точках этого диапазона, включая его границы;

— вычисление по указанным в таблице степенным рядам приближенных значений функции в тех же точках, ограничиваясь l первыми членами ряда;

— расчет погрешности приближенного определения функции в каждой точке, сравнивая приближенное значение с точным, и построение графика зависимости погрешности от аргумента;

— вычисление приближенных значений функции в тех же точках с относительной погрешностью не более $\varepsilon = 0,001$; построение графика полученных относительных погрешностей.

Указанные действия записать в виде программы (M-файла).

Номер задания	x1	x2	m	r	$f(x)$	Приближенная формула
1	1	10	50	4	$\cos(x)$	$\prod \left(1 - \frac{x^2}{(2k-1)^2 \pi^2} \right)$
2	0,6	5	50	3	$\ln(x)$	$\sum \frac{a^k}{k}, a = \frac{x-1}{x}$
3	-0,9	0,9	45	4	$\text{arcctg}(x)$	$\frac{\pi}{2} - \sum (-1)^{k-1} \frac{x^{2k-1}}{2k-1}$
4	1	20	50	5	$\text{sh}(x)$	$\sum \frac{x^{2k-1}}{(2k-1)!}$
5	1	20	50	5	$\text{ch}(x)$	$1 + \sum \frac{x^{2k}}{(2k)!}$
6	-0,9	0,9	50	5	$\text{arth}(x)$	$\sum \frac{x^{2k-1}}{(2k-1)!}$
7	1	20	50	5	$\text{arth}(x)$	$\sum \frac{1}{(2k-1)x^{2k-1}}$
8	-0,8	0,8	50	4	$\arcsin(x)$	$x + \sum \frac{1 \cdot 3 \cdot 5 \dots (2k-1) \cdot x^{2k+1}}{2 \cdot 4 \cdot 6 \dots (2k) \cdot (2k-1)}$
9	-0,8	0,8	50	4	$\arccos(x)$	$\frac{\pi}{2} - \left\{ x + \sum \frac{1 \cdot 3 \cdot 5 \dots (2k-1) \cdot x^{2k+1}}{2 \cdot 4 \cdot 6 \dots (2k) \cdot (2k-1)} \right\}$
10	-5	5	50	6	$\exp(x)$	$1 + \sum \frac{x^k}{k!}$

Задачи

1. Найти произведение положительных, сумму и количество отрицательных из 10 введенных целых значений. Ввод исходных данных организовать в диалоговом режиме.

2. Из N целых чисел найти минимальное среди положительных и максимальное среди отрицательных значений.

3. В ЭВМ вводятся результаты соревнований по прыжкам в высоту. Число участников произвольно. Напечатать сообщение о победителе.

4. Даны два целых числа, каждое число записано в отдельной строке. Выведите наибольшее из данных чисел.

5. Требуется определить, является ли данный год високосным. Напомним, что год является високосным, если его номер кратен 4, но не кратен 100, а также если он кратен 400.

6. Решить в целых числах уравнение $ax + b = 0$. Вводятся 2 числа: a и b . Необходимо вывести все решения, если их число конечно, сообщение «нет однозначного решения» (без кавычек), если решений нет, и сообщение об ошибке, если присутствует деление на ноль.

7. Даны координаты двух точек на плоскости, требуется определить, в каких координатных четвертях они лежат. Вывести сообщение на экран, если точки лежат они в одной координатной четверти или нет (все координаты отличны от нуля). Вводятся 4 числа: координаты первой точки (x_1, y_1) и координаты второй точки (x_2, y_2).

8. Даны три натуральных числа a, b, c . Требуется определить, существует ли треугольник с такими сторонами. Сообщение о существовании вывести на экран.

9. Даны три целых числа. Определите, сколько среди них совпадающих. Сообщение о результате вывести на экран.

10. Дано три числа. Упорядочите их в порядке неубывания.

11. Вычислить значение функции $y = f(x)$ и построить ее график:

$$\begin{cases} y = 2x - 10, & \text{если } x > 0, \\ y = 0, & \text{если } x = 0, \\ y = 2|x| - 1, & \text{если } x < 0. \end{cases}$$

12. Напишите программу, доказывающую или проверяющую, что для множества натуральных чисел выполняется равенство $1+2+\dots+n = n(n+1)/2$, где n – любое натуральное число.

13. Вычислить: $1+2+4+8+\dots+210$.

14. Вычислить: $(1+2)*(1+2+3)*\dots*(1+2+\dots+10)$.

15. Для заданного числа N составьте программу вычисления суммы $S = 1+1/2+1/3+1/4+\dots+1/N$, где N – натуральное число.

16. Составьте программу вывода на экран всех простых чисел, не превосходящих заданного N . Простым называется натуральное число больше единицы, имеющее только два делителя: единицу и само это число.

17. Найти сумму n -го количества элементов ряда $1, -0.5, 0.25, -0.125, \dots$

Учебное издание

Беспалова Ирина Марковна
Мартынчик Ксения Игоревна
Марковец Алексей Владимирович
Усов Алексей Георгиевич

**СИСТЕМЫ КОМПЬЮТЕРНОЙ МАТЕМАТИКИ.
Приемы работы в среде MATLAB**

Учебное пособие

Издательский редактор Т. Н. Козлова

Учебное электронное издание сетевого распространения

Системные требования:
электронное устройство с программным обеспечением
для воспроизведения файлов формата PDF

Режим доступа: http://publish.sutd.ru/tp_get_file.php?id=2019256, по паролю.
– Загл. с экрана.

Дата подписания к использованию 11.07.2019 г. Рег. № 256/19

ФГБОУВО «СПбГУПТД»
Юридический и почтовый адрес:
191186, Санкт-Петербург, ул. Большая Морская, 18.
<http://sutd.ru/>