ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

НИГМЕДЗЯНОВА А.М., МИНСАФИНА Э.И., МАМЕШИНА А.Н.

«СРЕДА ПРОГРАММИРОВАНИЯ OCTAVE. ЗАДАЧИ ЛИНЕЙНОЙ АЛГЕБРЫ. ГРАФИКА»

Учебно-методическое пособие

Казань 2016

УДК 51-7, 372.8
ББК 22.1, 74.2

Печатается по решению учебно-методической комиссии
Института математики и механики им. Н.И. Лобачевского
Протокол № 00 от 00 ноября 2016 года

Рецензенты:
доктор физ.–мат. наук, доцент
А.А. Попов;
кандидат физ.–мат. наук, доцент
Е.К. Липачев

Нигмедзянова А.М.,Минсафина Э.И., Мамешина А.Н.
Среда программирования OCTAVE. Задачи линейной алгебры. Графика: учеб.-методич. пособие / Нигмедзянова А.М.,Минсафина Э.И., Мамешина А.Н.. – Казань: Казан. ун-т, 2016. – 78 с.

Учебно-методическое пособие включает в себя теоретическую справку по работе с OCTAVE, а так же по решению задач линейной алгебры и построению графиков в данной среде программирования. Пособие содержит индивидуальные задания по каждой из этих тем с методическими указаниями к их выполнению. Предназначено для студентов математических факультетов, а также учеников, изучающих информатику на элективных курсах.

Введение

В период реализации Концепции профильного образования на старшей ступени особо актуальным является внедрение в процесс обучения информатике и информационным технологиям таких систем и программ, которые дают возможность учащимся раскрыть свои умственные и творческие способности, получить основы профессионального навыка и определить курс своей будущей карьеры.

Учащимся необходимо дать умения и навыки компьютерного моделирования, что является одним из приоритетных направлений в прикладных науках. Рассматриваемая среда программирования Octave позволяет легко проводить сложные вычисления и наглядно представить изучаемый объект в графической форме. В последние несколько лет в математике очень быстро развилось новое направление – компьютерная или символьная математика, представленная в настоящее время пакетами программ, как Mathematica, Maple, MathCad, MatLab и др. Данные пакеты программ позволяют проводить вычисления на очень серьезном математическом уровне в различных областях как самой математики, так и ее приложений, обладают понятным интерфейсом и мощными графическими возможностями. Существующие системы компьютерной математики делятся на два типа: коммерческие и бесплатные,то есть со свободным доступом. Сегодня можно найти такие бесплатные СКМ, которые могут стать аналогами коммерческим проектам. Но чаще всего такие системы не русифицированы. К таковым средам программирования можно отнести язык программирования Octave.

В связи с этим и было разработано данное методическое пособие, которое включает в себя теоретическую справку по работе с Octave, а так же по решению задач линейной алгебры и построению графиков в данной среде программирования, и индивидуальные задания по каждой из этих тем с методическими указаниями к их выполнению. Данное методическое пособие предназначено для студентов математических факультетов, а также учеников, изучающих информатику на элективных курсах.

В этой работе приводятся основные понятия, теоремы, которые используются при исследовании и решении задач линейной алгебры и построении графиков.

Всего представлено 17 заданий: 9 задач линейной алгебры и 8 задач на построение графиков. Каждый тип задач, предлагаемых для самостоятельной работы, представлен в 10 вариантах.

1. Первоначальные сведения о работе с пакетом Octave

1.1 Начало работы с Octave

Первоначально Octave была разработана для студентов факультета инженерной химии, в качестве вспомогательной программы для учебника по проектированию химического реактора, в 1993 году. Разработчиками данной программы являются James B. Rawlings, из Висконского университета в Мэдисоне и John G.Eckerd из Техасского университета. Впоследствии область применения данного продукта расширилась, его стали использовать в университете при обучении дифференциальным уравнениям и линейной алгебре. Практически каждый думает, что название «октава» имеет какое-то отношение к музыке, на самом деле так звали профессора, написавшего известный учебник по моделированию химических реакций, который славился хорошими умениями в математических подсчетах.

Очевидно, что Octave сейчас больше, чем просто ещё один программный пакет для учебного курса с ограниченной областью применения за пределами учебного класса. Сегодня Octave- высокоуровневый интерпретируемый язык программирования, предназначенный для решения задач вычислительной математики. Программа представляет удобный интерактивный командный интерфейс для решения линейных и нелинейных математических задач, и для выполнения других численных экспериментов.

Язык Octave оперирует арифметикой вещественных и комплексных скаляров и матриц, имеет расширения для решения линейных алгебраических задач, нахождения корней систем нелинейных алгебраических уравнений, работы с полиномами, решения различных дифференциальных уравнений, интегрирования систем дифференциальных и дифференциально-алгебраических уравнений первого порядка, интегрирования функций на конечных и бесконечных интервалах. Этот список можно легко расширить, используя язык Octave.

Свойства Octave:

- Octave написан на C++ с использованием библиотеки STL.
- Для запуска скриптов Octave использует интерпретатор.
- Octave можно дополнять динамически подгружаемыми модулями.
- Для создания графиков используется gnuplot.
- Octave является кроссплатформенной.
- Язык, совместимый с MatLab.
- В Интернете существуют онлайн-сервисы, позволяющие выполнять скрипты Octave и отображать полученные результаты в браузере: www.tutorialspoint.com, octave-online.net и др.

Синтаксис языка Octave:

- Название переменной может содержать буквы, цифры и символы подчеркивания "_", но не должно начинаться с цифры;
- язык различает прописные и строчные буквы, то есть a и A-это разные переменные;
- создавать переменные можно в любом месте программы, они не описываются и не обновляются;
- переменные не имеют постоянства типа данных, то есть сначала можно присвоить $a = 10$, а затем $a = Hello$;

Преимущества Octave:

- Язык Octave отличается от других языков тем, что студенты осваивают его куда быстрее.
- Octave не является коммерческой проектом, скачать ее может любой пользователь, так же как и внести свои коррективы.
- Данную программу можно скачать на любое ПК устройство с ОС Android или Windows с помощью Google Play. Окно для скачивания данного приложения представлено на Рисунке 1.1 .

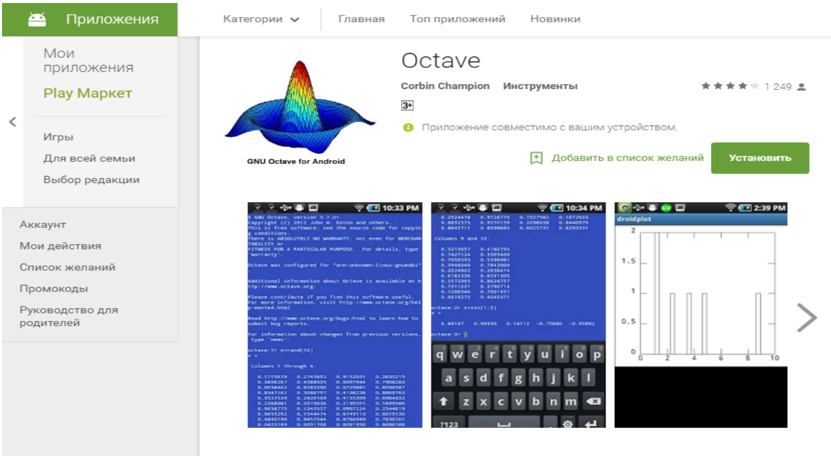

Рисунок 1.1. Окно для скачивания Octave в Google Play

1.2 Установка программы для ОС Windows

Данная программа может быть установлена как для ОС Linux , так и для ОС Windows. Мы рассмотрим, как работает данная программа в ОС Windows. Скачать последнюю версию Octave можно с официального сайта ftp://ftp.gnu.org. На сегодняшний день доступна версия Octave 4.0.1, на основе которой будет построена наша работа. Инсталляция программы происходит так же, как и для любых других программ для ОС Windows. После инсталляции Octave 4.0.1 на системном диске образуется новая папка «Octave». В ней можно обнаружить несколько папок и ярлыков, из которых наиболее важные следующие:

- Octave – открывает окно Octave
- Notepad++ – текстовый редактор
- Uninstall–удаление программы и ее компонентов

В то же время на рабочем столе появляются 2 ярлыка:

1. Octave(CLI)– интерфейс командной строки;
2. Octave(GUI)– графический пользовательский интерфейс.

1.3 Запуск программы Octave-4.0.1

Запуск программы можно осуществить через меню Windows, открываемой кнопкой «Пуск». Найдя папку Octave-4.0.1, необходимо открыть подменю и щелкнуть на команде Octave(GUI). Также возможен запуск программы с помощью ярлыка .

После запуска программы пользователь видит окно интерпретатора (Рисунок 1.2) с сообщением, в котором указано название программы, ее версия, разработчики, краткие характеристики пакета и ссылки на официальный сайт, где можно узнать более подробную информацию о программе.

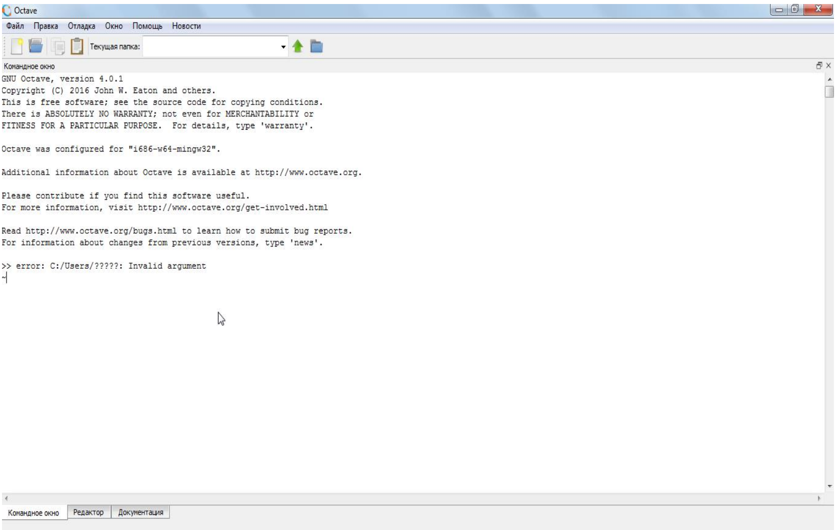

Рисунок 1.2. Окно интерпретатора

Перед началом работы в программе пользователь должен определить, в каком режиме он будет работать, так как в Octave существуют 2 режима работы:

1. **Терминальный режим.** В данном режиме пользователь остается во вкладке, которая открылась при запуске программы (Рисунок 1.2). Работа с системой происходит в диалоговом режиме

– пользователь вводит команды, а система выполняет их и выводит результаты на экран.

2. **Программный режим.** Для работы в этом режиме, следует открыть вкладку «Редактор» , расположенную в нижней части окна, где пользователь создает текстовый файл с расширением .m, в котором хранятся последовательно выполняемые команды Octave. Затем этот текстовый файл (программа на языке Octave) запускается на выполнение в среде Octave. Окно редактора продемонстрировано на Рисунке 1.3

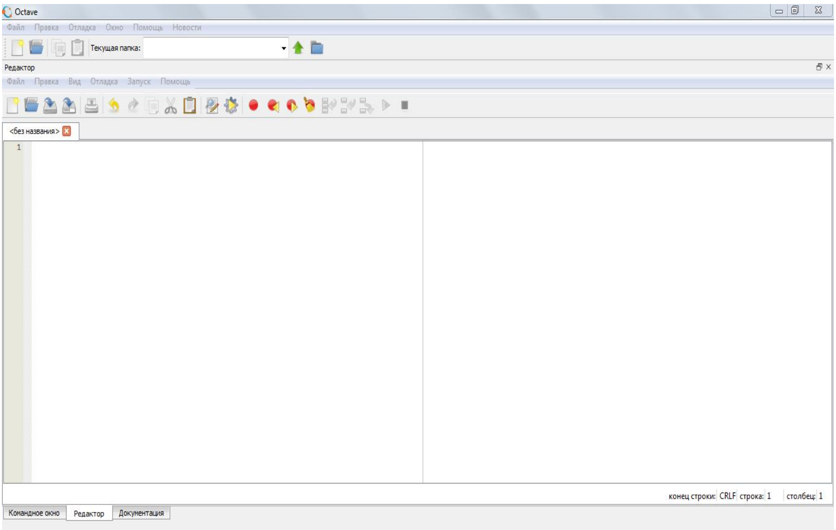

Рисунок 1.3. Окно редактора

1.4 Интерфейс пользователя

После запуска программы, перед нами появляется окно интерпретатора, представленное на Рисунке 1.3. Структура данного окна включает в себя перечень следующих позиций:

- строка заголовка (в левом верхнем углу);
- строка главного меню;
- главная панель инструментов;
- окно ввода и редактирования документа;
- панель вкладок.

Язык главного меню программы определяется автоматически согласно тому, какой язык используется в операционной системе. К сожалению, отсутствует контекстная панель инструментов и панель специальных математических символов.

Пользовательский интерфейс Octave-4.0.1 позволяет создавать документы, содержащие одновременно текстовые комментарии, команды входного языка, результаты вычислений в виде математических формул и графические данные.

Пользователь Octave-4.0.1 работает с документами, которые являются описанием алгоритмов решения задач, программами и результатами их исполнения. Графические построения выполняются в отдельных окнах и имеют свое меню для оперативного управления параметрами.

Рассмотрим подробнее меню системы, которое располагается под строкой заголовка. Меню системы включает:

- Файл – работа с файлами и областью переменных
- Правка – команды редактирования документа, операции с буфером обмена, настройки программы
- Отладка – инструменты отладки
- Окно – управление видом пользовательского интерфейса

- Помощь – работа со справочной системой
- Новости – примечания к выпуску.

При работе в программном режиме, во вкладке «Редактор», основная структура окна остается той же, добавляется лишь контекстная панель инструментов (Рисунок 1.4) и строка состояния в нижней части окна (Рисунок 1.5) .

Рисунок 1.4. Панель инструментов

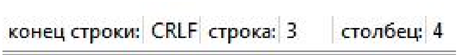

Рисунок 1.5. Строка состояния

Полезно обратить внимание на то, что в настройках можно изменить параметры интерфейса, как командной строки, так и редактора, а так же устанавливать комбинации клавиш.

Еще один важный элемент интерфейса - это всплывающие подсказки, которые появляются при наведении курсора мыши на тот или иной элемент интерфейса. На Рисунке 1.6 Показана одна из всплывающих подсказок у кнопки 

Рисунок 1.6. Всплывающая подсказка

1.5 Помощь и документация

Octave имеет достаточно обширную справочную базу, но на английском языке. Документация доступна как из командной строки Octave в онлайн-режиме, так и в печатном варианте. Для того, чтобы вызвать справку, достаточно ввести в командную строку слово help и название команды, которую нужно описать. Например, если ввести команду "help plot"программа отобразит вам справочную информацию по команде построения функций. Иногда справочная информация имеет большой объем, такой, что справка не способна поместиться на экран. Тогда, для того, чтобы пролистать справку на следующую страницу, следует нажать клавишу <f> (forward), <b>(back), чтобы вернуться назад, и <q> quit, чтобы выйти из справки. Но для удобства лучше использовать руководство в разделе Документация, просто выбрав нужную вкладку в нижней области окна или через опцию главного меню Помощь-Документация-На диске.

1.6 Выход из системы

Для выхода из программы достаточно нажать комбинацию клавиш <Ctrl>+ <D> , либо набрать в командной строке Octave одну из команд exit или quit. Если вы работаете в окне Редактора, перед выходом из системы следует сохранить файл под англоязычным названием в формате .m в папку, которая добавлена в путь загрузки.

1.7 Начальные навыки работы

Признаком того, что система готова к работе, является наличие знака приглашения » в командном окне. При работе в командном окне нажатие клавиши Enter заставляет систему выполнить команду и вывести результат:

```
Командное окно
>> 1+3
ans =  4
```

Рисунок 1.7. Результат выполнения команды

В переменной ans (от англ. Answer-ответ) хранится результат последней операции, если команда не содержит знака присваивания. Следует помнить, что значение переменной ans изменяется после каждого вызова команды без операции присваивания.

Работая в командном окне, очевидно, что все выполняемые команды не могут одновременно находиться в поле зрения пользователя. Поэтому просмотреть информацию, которая покинула видимую часть окна, можно с помощью полос прокрутки и клавиш Page Up и Page Down.

В ходе выполнения более сложных вычислений, как правило, существуют промежуточные операции, результат выполнения которых выводить нет необходимости. Чтобы отменить вывод результата, следует после соответствующей команды поставить символ «;» . В случае отсутствия данного символа в конце строки, результаты выводятся на экран.

```
Командное окно
>> A=[1 2;3 4]
A =

   1   2
   3   4

>> B=[1 -1 3 5;2 2 0 10; 4 6 7 8;1 10 34 5];
>> |
```

Рисунок 1.8. Вывод результата на экран

Код программы может сопровождаться комментариями. Текст в строке, который идет после символа % или #, игнорируется программой и интерпретатором не обрабатывается. К сожалению, русскоязычные комментарии можно оставлять только в режиме редактора. В командном окне русская раскладка не воспринимается.

Важно отметить, что при работе в командном окне, Octave сохраняет набранные команды во внутреннем буфере, так что их можно вызвать заново и редактировать. Клавиши $\Uparrow$ и $\Downarrow$ позволяют вернуть в командную строку ранее введённые команды или другую входную информацию, так как вся эта информация сохраняется в специальной области памяти. Так, если в пустой активной командной строке нажать клавишу $\Uparrow$, в ней появится последняя вводимая команда. Повторное нажатие вызовет предпоследнюю команду, и так далее. Клавиша $\Downarrow$ выводит команды в обратном порядке.

1.8 Сообщения об ошибках

При написании программы, Octave может выдавать сообщение об ошибке, которое отображается в строке сразу же после команды с ошибкой. Octave выделяет два вида ошибок для поврежденных программ. *Синтаксическая ошибка* возникает, если вы допустили ошибку при написании команды и Octave отказывается понимать то, что вы набрали. Например, если вы ошибетесь, набирая ключевое слово, октава немедленно выдаст сообщение об ошибке.

```
>> y=f(x)  f(x)=2***x
parse error:

  syntax error

>>> y=f(x)  f(x)=2***x
            ^
```

Рисунок 1.9. Ошибка синтаксиса

Для большинства ошибок синтаксического анализа, Octave использует каретку '$\wedge$', чтобы отметить тот участок в команде, который не имеет смысла. В случае на Рисунке 1.9 , Octave сгенерировала сообщение об ошибке, потому что команда для возведения в степень была написана неправильно.

Другой класс сообщения об ошибке происходит во время выполнения команды. Они именуются как (*run-time errors*) , а иногда, как ошибки оценки (*evaluation errors*), потому что они происходят, когда ваша программа выполняет скрипт, или оценивает. Пример данной ошибки показан на Рисунке 1.10.

```
>> y=f(x) f(x)=2***x
parse error:

  syntax error

>>> y=f(x) f(x)=2***x
           ^

>> f(x)
error: 'x' undefined near line 1 column 3
error: evaluating argument list element number 1
```

Рисунок 1.10. Ошибка при выполнении команды

Это сообщение об ошибке состоит из нескольких частей, и дает довольно много информации, чтобы найти источник ошибки. В приведенном выше примере (Рисунок 1.10) первая строка указывает на то, что переменная «x», стоящая в строке 1, графе 3 была не определена. Для ошибок, возникающих в функциях, строки отсчитываются от начала файла, содержащего определение функции. Для ошибки, возникающие за пределами функции включения, номер строки указывает на номер строки ввода, которая обычно отображается в основной строке приглашения.

Сообщения об ошибке помогают пользователю проследить, на каком этапе он допустил ошибку, а затем исправить ее и попробовать снова.

2. Задачи линейной алгебры

2.1 Ввод и формирование векторов и матриц

Познакомимся с инструментами Octave, предназначенными для работы с векторами и матрицами, а также с возможностями, которые предоставляет пакет при непосредственном решении задач линейной алгебры.

Векторы и матрицы в Octave задаются путём ввода их элементов. Элементы вектора-строки отделяют пробелами или запятыми, а всю конструкцию заключают в квадратные скобки:

```
>> a=[2 3 -1 4]
a =
   2   3  -1   4
>> b=[3,5,7]
b =
   3   5   7
```

Вектор-столбец можно задать, если элементы отделять друг от друга точкой с запятой «;» :

```
>> c=[pi/2;pi/3;pi/4]
c =
   1.57080
   1.04720
   0.78540
```

Обратиться к элементу вектора можно указав имя вектора, а в круглых скобках - номер элемента, под которым он хранится в этом векторе:

```
>> a(1)
ans =  2
>> b(3)
ans =  7
>> c(2)
ans =  1.0472
```

Ввод элементов матрицы осуществляется тем же способом, при этом элементы строк отделяются между собой пробелом или запятой, строки же между собой отделяются точкой с запятой «;».

```
 >> A=[1,2;3,4]
A =
   1   2
   3   4
>> B=[1 6 5 -9;3 4 7 10;3 5 9 8;11 13 14 0]
B =
    1    6    5   -9
    3    4    7   10
    3    5    9    8
   11   13   14    0
```

Обратиться к элементу матрицы можно указав после имени матрицы, в круглых скобках, через запятую, номер строки и номер столбца, на пересечении которых элемент расположен.

```
>> B(2,4)
ans =  10
```

Таким же образом можно изменить любой элемент матрицы на другой.

```
>> B(2,4)=6
B =
    1    6    5   -9
    3    4    7    6
    3    5    9    8
   11   13   14    0
>> B(2,4)
ans =  6
```

Матрицы и векторы можно формировать из векторов, заданных ранее.

```
>> a=[1 2 3]; b=[4 5 9]; c=[6 10 7];
>> Matr=[a b c]
Matr =
    1     2     3     4     5     9     6    10     7
>> Matr=[a;b;c]
Matr =
    1     2     3
    4     5     9
    6    10     7
```

Так же матрицы можно формировать из ранее заданных матриц.

```
>>>> a=[1 2 3]; b=[4 5 9]; c=[6 10 7];
>> Matr=[a; b; c]
Matr =
    1     2     3
    4     5     9
    6    10     7
 M=[Matr Matr Matr]
M =
    1     2     3     1     2     3     1     2     3
    4     5     9     4     5     9     4     5     9
    6    10     7     6    10     7     6    10     7
>> M=[Matr;Matr;Matr]
M =
    1     2     3
    4     5     9
    6    10     7
    1     2     3
    4     5     9
    6    10     7
    1     2     3
    4     5     9
    6    10     7
```

Вообще говоря, элементами матрицы могут быть произвольные выражения, но главное, чтобы их размерность совпадала при слиянии. Например, в предыдущем примере мы совместили три одинаковые матрицы Matr, размерность которых совпадает и равна 3x3. Очевидно, что нам не удастся выполнить горизонтальную конкатенацию матриц, если количество строк будет различным, и вертикальную, если различным будет количество столбцов, а так же, если размерность матриц полностью не совпадает.

```
>> A=[1,2,3;1,2,5]; B=[1 0 5 9; 1 2 3 4];
>> M=[A B]
M =
   1   2   3   1   0   5   9
   1   2   5   1   2   3   4
>> M=[A;B]
error: vertical dimensions mismatch (2x3 vs 2x4)
```

Octave выдает ошибку, т.к. размерность матрицы A (2×3) не совпадает с размерностью матрицы B (2×4). Аналогично выражение [A,1] выдает ошибку горизонтального несоответствия размеров:

```
>> M=[A,1]
error: horizontal dimensions mismatch (2x3 vs 1x1)
```

Важную роль при работе с матрицами играет знак двоеточия «:»

```
>> A=[1,2,5,6;3,5,7,9]
A =
   1   2   5   6
   3   5   7   9
>> A(:,3) #выделение из матрицы 3 столбца
ans =
   5
   7
>>  A(1,:) #выделение из матрицы 1 строки
ans =
   1   2   5   6
>> B= A(1:2,2:3) #выделение из матрицы подматрицу, включающую
```

```
элементы 1,2 строки и 2,3 столбца
B =
   2   5
   5   7
```

2.2 Функции для работы с матрицами и векторами

В Octave существуют специальные функции, предназначенные для работы с матрицами и векторами. Эти функции можно разделить на следующие группы:

1. функции для работы с векторами;
2. функции для работы с матрицами;
3. функции, реализующие численные алгоритмы решения задач линейной алгебры.

Функции для работы с векторами

Рассмотрим наиболее часто используемые функции для работы с векторами, которые приведены в Таблице 2.1

Таблица 2.1 Функции для работы с векторами

Функция	**Значение**
rand(1,n) rand(n,1)	Генерирует вектор-строку случайных элементов Генерирует вектор-столбец случайных элементов
length(v)	Определяет длину вектора $\vec{v}$
prod(v)	Вычисляет произведение элементов вектора $\vec{v}$
sum(v)	Вычисляет сумму элементов вектора $\vec{v}$
mean(v)	Определяет среднее арифметическое массива $\vec{v}$
dot(v_1, v_2)	Вычисляет скалярное произведение векторов $\vec{v}_1$ и $\vec{v}_2$
cross(v_1, v_2)	Вычисляет векторное произведение векторов $\vec{v}_1$ и $\vec{v}_2$

В Листинге приведены примеры применения данных функций.

```
>> rand(1,n) %вектор-строка случайных элементов
ans =
   0.09   0.44   0.08
>> rand(n,1) %вектор-столбец случайных элементов
ans =
   0.94
   0.13
   0.87
>> v=[1 6 8]
v =
   1.00   6.00   8.00
>> u=[2 7 4]
u =
   2.00   7.00   4.00
>> length(v) %длина вектора
ans =  3.00
>> prod(v) %произведение элементов
ans =  48.00
>> sum(v) %сумма элементов
ans =  15.00
>> mean(v) %среднее арифметическое элементов
ans =  5.00
>> dot(v,u) %скалярное произведение
ans =  76.00
>> cross(v,u) %векторное произведение
ans =
  -32.00   12.00   -5.00
```

2.3 Функции для работы с матрицами

В теории линейной алгебры выделяют специальные виды матриц, например, единичную, диагональную, блочную или нулевую. Создать такие матрицы в Octave позволяют функции, прописанные в таблице 2.2.

Таблица 2.2 Функции для работы с матрицами

Функция	**Значение**
rand(m) rand(m,n)	Генерирует квадратную матрицу случайных элементов размерности $(m \times m)$ Генерирует матрицу случайных элементов размерности $(m \times n)$
eye(m) eye(m,n)	Генерирует единичную квадратную матрицу размерности $(m \times m)$ Генерирует единичную матрицу размерности $(m \times n)$
ones(m) ones(m,n)	Генерирует матрицу $(m \times m)$ с элементами 1 Генерирует матрицу $(m \times n)$ с элементами 1
zeros(m) zeros(m,n)	Генерирует нулевую матрицу $(m \times m)$ Генерирует нулевую матрицу $(m \times n)$
diag(v, k)	Генерирует диагональную матрицу с элементами вектора $\vec{v}$ на диагонали k. при k=0 – главная диагональ (по умолчанию) при k>0 элементы $\vec{v}$ ставятся на k-ую диагональ выше главной диагонали при k<0 на k-ую диагональ ниже главной диагонали
[ ]	Генерирует пустую матрицу

rot90(M,[k])	Осуществляет поворот матрицы M на 90° или $90 \cdot k, k \in Z$ против часовой стрелки
tril (M,[k])	Формирует из матрицы M нижнюю треугольную матрицу, начиная с главной или с k–й диагонали
sort(M)	Выдаёт матрицу того же размера, что и M, каждый столбец которой упорядочен по возрастанию

Ниже в Листинге приведен разбор рассматриваемых функций.

```
>> A=rand(3)
A =
   0.58   0.26   0.78
   0.93   0.51   0.17
   0.40   0.78   0.44
>> A=rand(2,3)
A =
   0.01   0.81   0.46
   0.14   0.45   0.87
>> M=eye(5) #единичная квадратная матрица размерности 5*5
M =
Diagonal Matrix
   1.00      0      0      0      0
      0   1.00      0      0      0
      0      0   1.00      0      0
      0      0      0   1.00      0
      0      0      0      0   1.00
>> M=eye(2,3) #единичная матрица размерности 2*3
M =
Diagonal Matrix
   1.00      0      0
      0   1.00      0
>> N=ones(3)
N =
```

```
   1.00   1.00   1.00
   1.00   1.00   1.00
   1.00   1.00   1.00
>> N=ones(3,5)
N =
   1.00   1.00   1.00   1.00   1.00
   1.00   1.00   1.00   1.00   1.00
   1.00   1.00   1.00   1.00   1.00
>> Z=zeros(2)
Z =
   0.00   0.00
   0.00   0.00
>> Z=zeros(2,3)
Z =
   0.00   0.00   0.00
   0.00   0.00   0.00
>> D=diag([1 3 6], 0) #диагональная матрица с элементами вектора
 в главной диагонали
D =
Diagonal Matrix
   1.00      0      0
      0   3.00      0
      0      0   6.00
>> D=diag([1 3 6], 2) #диагональная матрица с элементами вектора
во 2 диагонали
D =
   0.00   0.00   1.00   0.00   0.00
   0.00   0.00   0.00   3.00   0.00
   0.00   0.00   0.00   0.00   6.00
   0.00   0.00   0.00   0.00   0.00
   0.00   0.00   0.00   0.00   0.00
>> D=diag([1 3 6],-2) #диагональная матрица с элементами вектора
на 2 диагонали ниже главной диагонали
D =
```

```
   0.00   0.00   0.00   0.00   0.00
   0.00   0.00   0.00   0.00   0.00
   1.00   0.00   0.00   0.00   0.00
   0.00   3.00   0.00   0.00   0.00
   0.00   0.00   6.00   0.00   0.00
>> P=[] #пустая матрица
P = [](0x0)
>> M=[1 2 3 4;5 6 7 8;9 10 11 12]
M =
    1.00    2.00    3.00    4.00
    5.00    6.00    7.00    8.00
    9.00   10.00   11.00   12.00
>> rot90(M) #поворот матрицы на 90 градусов против часовой стрелки
ans =
    4.00    8.00   12.00
    3.00    7.00   11.00
    2.00    6.00   10.00
    1.00    5.00    9.00
>> rot90(M,[2]) #поворот матрицы на 180 градусов против
часовой стрелки
ans =
   12.00   11.00   10.00    9.00
    8.00    7.00    6.00    5.00
    4.00    3.00    2.00    1.00
>> tril(M) # нижняя треугольная матрица,начиная с главной диагонали
ans =
    1.00    0.00    0.00    0.00
    5.00    6.00    0.00    0.00
    9.00   10.00   11.00    0.00
>> tril(M,[2]) #нижняя треугольная матрица,начиная со
 2 диагонали
ans =
   1.00    2.00    3.00    0.00
    5.00    6.00    7.00    8.00
```

```
    9.00   10.00   11.00   12.00
>> tril(M,[-1]) #нижняя треугольная матрица,начиная с
 -1 диагонали
ans =
    0.00    0.00    0.00    0.00
    5.00    0.00    0.00    0.00
    9.00   10.00    0.00    0.00
>> N=[4 4 9 ;25 64 49; 36 4 1]
N =
    4.00    4.00    9.00
   25.00   64.00   49.00
   36.00    4.00    1.00
>> sort(M) #матрица, каждый столбец упорядочен по возрастанию
ans =
    1.00    2.00    3.00    4.00
    5.00    6.00    7.00    8.00
    9.00   10.00   11.00   12.00
>> sort(N)
ans =
    4.00    4.00    1.00
   25.00    4.00    9.00
   36.00   64.00   49.00
```

Пустые матрицы удобно использовать, когда необходимо удалить из заданной матрицы некоторые строки или столбцы.

```
>> A=[1,2,5,6;3,5,7,9;3 6 8 9; 5,7,8,10]
A =
    1    2    5    6
    3    5    7    9
    3    6    8    9
    5    7    8   10
>> A(:,2)=[]  #удалить 2 столбец
A =
    1    5    6
    3    7    9
```

```
    3    8    9
    5    8   10
>> A(1,:)=[] # удалить 1 строку
A =
    3    7    9
    3    8    9
    5    8   10
>> A(1:2,:)=[] #удаление 1,2 строки
A =
    5    8   10
```

Одновременно удалить столбцы и строки Octave не умеет и выдает нам ошибку:

```
>> A(1:2,1:1)=[]
error: a null assignment can only have one non-colon index
```

2.4 Функции, реализующие численные алгоритмы решения задач линейной алгебры

Таблица 2.3 Функции для решения задач линейной алгебры

Функция	**Значение**
det(M)	Вычисляет определитель квадратной матрицы M
rank(M)	Вычисляет ранг матрицы M
trace(M)	Вычисляет след матрицы M, то есть сумму элементов главной диагонали
norm(M[,p])	Формирует различные виды норм матрицы M в зависимости от p, если аргумент p=1,2,inf, не задан, то вычисляется вторая норма матрицы M
inv(M)	Формирует матрицу обратную к M
eig(M)	Формирует вектор собственных значений матрицы M
eig(A,B)	Формирует вектор обобщённых собственных значений,если A и B квадратные матрицы
poly(M)	Возвращает вектор–строку коэффициентов характеристического полинома матрицы M
rref(M)	Осуществляет приведение матрицы M к треугольной форме, используя метод исключения Гаусса

В Листинге ниже показано, как работают данные функции.

```
>> A=[1 3 4; 2 6 7; 3 8 9]
A =
   1.00   3.00   4.00
   2.00   6.00   7.00
   3.00   8.00   9.00
>> B=[2 8 4;6 3 8;10 21 5]
B =
    2.00    8.00    4.00
    6.00    3.00    8.00
   10.00   21.00    5.00
```

```
>> det(A) %определитель матрицы
ans = -1.00
>> rank(A) %ранг матрицы
ans =  3.00
>> trace(A)% след матрицы
ans =  16.00
>> norm(A) %вторая норма матрицы
ans =  16.39
>> inv(A) %обратная матрица
ans =
   2.00  -5.00   3.00
  -3.00   3.00  -1.00
   2.00  -1.00   0.00
>> eig(A) %вектор собственных значений матрицы
ans =
   16.30
   -0.44
    0.14
>> eig(A,B) %вектор обобщённых собственных значений
ans =
   0.74
   0.12
  -0.02
>> poly(A) % вектор коэффициентов характеристического полинома
 матрицы
ans =
    1.00  -16.00   -5.00    1.00
>> rref(A) %метод Гаусса
ans =
   1.00   0.00   0.00
   0.00   1.00   0.00
   0.00   0.00   1.00
```

2.5 Действия над матрицами

Для того чтобы рассмотреть операции, которые применимы к матрицам с математической точки зрения, необходимо вспомнить, что многие из этих операций требуют определенной согласованности между размерностями матриц. Следовательно, первым делом нужно научиться определять размерность той или иной матрицы.

Узнать число строк и столбцов данной матрицы позволяет функция *size*. Аргументом функции является матрица или вектор, первый возвращаемый параметр определяет число строк, второй - число столбцов.

Если интересует только количество строк матрицы, то вместо *size* можно использовать функцию *rows*. Если необходимо определить только количество столбцов, то используется функция *columns*.

```
>> A=rand(5,15); %задаем произвольную матрицу
>> size(A) %вычисляем число строк и столбцов
ans =
    5   15
>> rows(A)
ans =  5
>> columns(A)
ans =  15
```

Операция транспонирования «'» меняет в заданной матрице строки на столбцы и также применима к матрицам любой размерности.

```
>> B=[1 6 5 -9;3 4 7 10;3 5 9 8;11 13 14 0]
B =
    1    6    5   -9
    3    4    7   10
    3    5    9    8
   11   13   14    0
>> B' #транспонированная матрица
ans =
    1    3    3   11
```

```
    6     4     5    13
    5     7     9    14
   -9    10     8     0
```

Сложение (+), вычитание (-), умножение (*) матриц в Octave основано на тех же правилах, что и в линейной алгебре. Складывать и вычитать можно только матрицы одинаковой размерности.

При умножении матриц «*» важно помнить, что число столбцов первой перемножаемой матрицы должно быть равно числу строк второй.

```
>> A=[1,2,10;4,5,29;12,34,25];
>> B=[1,1,1;2,4,6;88,12,13];
>> C=A+B
C =
      2      3     11
      6      9     35
    100     46     38
>> D=A-B
D =
     0     1     9
     2     1    23
   -76    22    12
>> F=A*B
F =
     885     129     143
    2566     372     411
    2280     448     541
```

Не вызовет ошибки, если применить бинарные операции сложения, вычитания или деления к матрице и скалярной величине, при этом каждая операция будет выполняться поэлементно:

```
>> A=[1,2,10;4,5,29;12,34,25];
>> B=A+2
B =
```

```
     3     4    12
     6     7    31
    14    36    27
>> C=A-2
C =
    -1     0     8
     2     3    27
    10    32    23
>> D=A*2
D =
     2     4    20
     8    10    58
    24    68    50
```

Возведение матрицы в степень $\wedge$ эквивалентно её умножению на себя указанное число раз. Заметим, что возводить в степень можно только квадратные матрицы. При этом целочисленный показатель степени может быть как положительным, так и отрицательным. Матрица в степени -1 называется обратной к данной. При возведении матрицы в положительную степень выполняется алгоритм умножения матрицы на себя указанное число раз. Возведение в отрицательную степень означает, что умножается на себя матрица, обратная к данной.

```
>> A=[1,2,10;4,5,29;12,34,25];
>> A^2
ans =
     129     352     318
     372    1019     910
     448    1044    1731
>> A^(-1)
ans =
  -2.1797468   0.7341772   0.0202532
   0.6278481  -0.2405063   0.0278481
   0.1924051  -0.0253165  -0.0075949
```

3. Индивидуальные задания по теме: Линейная алгебра

3.1 Определитель матрицы

Пусть

$$A = \begin{pmatrix} a_{11} & a_{12} & ... & a_{1n} \\ a_{21} & a_{22} & ... & a_{2n} \\ ... & ... & ... & ... \\ a_{n1} & a_{n2} & ... & a_{nn} \end{pmatrix},$$

квадратная матрица n-ого порядка. Выпишем произведение n-элементов данной матрицы, взятых по одному из каждой строки и каждого столбца. Оно будет иметь вид

$$a_{1\alpha_1} \cdot a_{2\alpha_2} \cdot a_{n\alpha_n},$$

здесь первые индексы обозначают номера строк, откуда взяты сомножители, а вторые- номера столбцов. Индексы $a_1 \cdot a_2 \cdot a_n$, составляют некоторую перестановку из чисел 1,2, ...n (номера столбцов). Из n элементов можно составить n! перестановок, поэтому элементов вида

$$a_{1\alpha_1} \cdot a_{2\alpha_2} \cdot a_{n\alpha_n}$$

будет также n!. Каждому такому элементу можно поставить в соответствие подстановку

$$\sigma = \begin{pmatrix} 1 & 2 & ... & n \\ a_1 & a_2 & ... & a_n \end{pmatrix}.$$

Известно, что $\frac{n!}{2}$ подстановок- четные и столько же подстановок- нечетные. Если подстановка четная, то соответствующий элемент берется со знаком "плюс". Если же подстановка нечетная, то соответствующий элемент берется со знаком "минус".

Полученная алгебраическая сумма $n!$ слагаемых вида

$$a_{1\alpha_1} \cdot a_{2\alpha_2} \cdot a_{n\alpha_n}$$

называется *определителем квадратной матрицы* A и обозначается $|A|$ (или $\det A$, или D_n). Таким образом,

$$\det A = \Sigma \ sgn\sigma \ a_{1\alpha_1} \cdot a_{2\alpha_2} \cdot a_{n\alpha_n},$$

$$sgn \ \sigma = \begin{cases} +1, \text{ если } \sigma\text{-четная} \\ -1, \text{ если } \sigma\text{-нечетная} \end{cases}$$

Если в матрице A=a_{ij} размерностью $m \times n$ заменить строки соответствующими столбцами, то получится транспонированная матрица: $A^T = a_{ji}$.

> **Задание 1.** Вычислите определитель, ранг,след матрицы A. Найдите транспонированную матрицу матрицы A.

Вариант 0.

$$A = \begin{pmatrix} 3 & 1 & 1 & 1 \\ 1 & 3 & 1 & 1 \\ 1 & 1 & 3 & 1 \\ 1 & 1 & 1 & 3 \end{pmatrix}$$

Решение. Для вычисления определителя, воспользуемся функцией det(A), для вычисления ранга – rank(A), следа – trace(A).

Решение задачи в Octave.

```
>>A=[3 1 1 1;1 3 1 1 ; 1 1 3 1; 1 1 1 3]
A =
   3   1   1   1
   1   3   1   1
   1   1   3   1
   1   1   1   3
>> det(A) % Определитель матрицы
ans =  48.000
>> rank(A) % Ранг матрицы
ans =  4
>> trace(A) % След матрицы
```

```
ans =  12
>> A' % Транспонированная матрица
ans =
   3   1   1   1
   1   3   1   1
   1   1   3   1
   1   1   1   3
```

Задача 1 для самостоятельного решения.

Вариант 1.

$$A = \begin{pmatrix} 1 & -1 & 2 & 1 \\ 2 & -1 & 4 & 3 \\ -2 & 2 & -5 & -3 \\ 5 & -5 & 10 & 4 \end{pmatrix}$$

Вариант 2.

$$A = \begin{pmatrix} 1 & 6 & 4 & 1 \\ 3 & 19 & 12 & 3 \\ -2 & -12 & -9 & -2 \\ 4 & 24 & 16 & 5 \end{pmatrix}$$

Вариант 3.

$$A = \begin{pmatrix} 1 & -3 & 2 & -1 \\ -4 & 13 & -8 & 4 \\ 2 & -6 & 4 & -2 \\ 3 & -9 & 6 & -2 \end{pmatrix}$$

Вариант 4.

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

Вариант 5.

$$A = \begin{pmatrix} -1 & 2 & -3 & -1 \\ 2 & -5 & 6 & 2 \\ 5 & -10 & 16 & 5 \\ -3 & 6 & -9 & -2 \end{pmatrix}$$

Вариант 6 .

$$A = \begin{pmatrix} -1 & 4 & -1 & 3 \\ 2 & -7 & -2 & 6 \\ 3 & -12 & -2 & 9 \\ 4 & 16 & -4 & 13 \end{pmatrix}$$

Вариант 7.

$$A = \begin{pmatrix} 1 & 2 & 1 & 4 \\ 2 & 3 & 2 & 8 \\ -3 & -6 & -2 & -12 \\ 4 & 8 & 4 & 15 \end{pmatrix}$$

Вариант 8 .

$$A = \begin{pmatrix} -1 & 0 & 0 & -1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{pmatrix}$$

Вариант 9.

$$A = \begin{pmatrix} 1 & -1 & 2 & 0 \\ 8 & -7 & 16 & 0 \\ -3 & 3 & -5 & 0 \\ 4 & 4 & 8 & 1 \end{pmatrix}$$

Вариант 10.

$$A = \begin{pmatrix} 7 & 14 & 0 & -7 \\ -1 & -3 & 0 & 1 \\ 1 & 2 & 1 & -1 \\ 3 & 6 & 0 & 2 \end{pmatrix}$$

3.2 Матрицы и действия над ними

Числовой *матрицей* называется таблица вида:

$$A = \begin{pmatrix} a_{11} & a_{12} & ... & a_{1n} \\ a_{21} & a_{22} & ... & a_{2n} \\ ... & ... & ... & ... \\ a_{m1} & a_{m2} & ... & a_{mn} \end{pmatrix},$$

где a_{ij}- некоторые числа, $i = 1, 2, ..., m, j = 1, 2, ..., n$. Данная матрица содержит m-строк, n-столбцов. Каждая строка представляет собой арифметический n-мерный вектор, а каждый столбец-арифметический m-мерный вектор. Число $m \times n$ называется *размерностью* матрицы.

Если $A = (a_{ij})_{m \times n}$ и $B = (b_{ij})_{m \times n}$ – матрицы одинаковой размерности $m \times n$, то их *суммой* называется матрица той же размерности.

$$A + B = (a_{ij} + b_{ij})_{m \times n}$$

Произведением матрицы $A = (a_{ij})_{m \times n}$ на число λ называется матрица $\lambda A = (\lambda a_{ij})_{m \times n}$ той же размерности.

Произведением матрицы $A = (a_{ij})_{m \times n}$ *на матрицу* $B = (b_{st})_{n \times k}$ называется матрица

$$AB = (c_{pq})_{m \times k},$$

где $c_{pq} = a_{p1}b_{1q} + a_{p2}b_{2q} + a_{pn}b_{nq}, p = 1, ..., m, q = 1, ..., k.$

Задание 2. Найдите произведение матриц A и B, если A,B данные матрицы. Если существует BA, то найдите и ее.

$$A = \begin{pmatrix} 1 & 2 & -1 \\ 0 & 1 & 2 \end{pmatrix}, B = \begin{pmatrix} 3 & -1 \\ -1 & 1 \\ 0 & -2 \end{pmatrix}$$

Решение. Матрицу A размерности 2×3 и матрицу B размерности 3×2 можно перемножить. Так как количество столбцов матрицы B совпадает с количеством строк матрицы A, то мы можем найти произведение BA .

Решение задачи в Octave.

```
>>A=[1 2 -1;0 1 2]
A =
 1  2  -1
 0  1   1
>> B= [3 -1;-1 1; 0 -2]
B=
   3  -1
  -1   1
   0  -2
>> A*B
ans =
   1   3
  -1  -3
>> B*A
ans =

   3  5 -5
  -1 -1  3
   0 -2 -4
```

Задача 2 для самостоятельного решения.

Вариант 1.

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

Вариант 2.

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

Вариант 3.

$$A = \begin{pmatrix} 3 & 5 \\ 7 & 9 \end{pmatrix}, B = \begin{pmatrix} 1 & 1 & 3 \\ 9 & 1 & 2 \end{pmatrix}$$

Вариант 4.

$$A = \begin{pmatrix} -2 & 3 \\ -6 & -5 \end{pmatrix}, B = \begin{pmatrix} 5 & 2 \\ 4 & -7 \end{pmatrix}$$

Вариант 5.

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

Вариант 6.

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

Вариант 7.

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 5 & 8 & 3 \\ 1 & 2 & 4 \end{pmatrix}, B = \begin{pmatrix} -1 \\ 3 \\ 1 \end{pmatrix}$$

Вариант 8.

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

Вариант 9.

$$A = \begin{pmatrix} 2 \\ 4 \\ 5 \end{pmatrix}, B = \begin{pmatrix} 4 & 0 & 2 \end{pmatrix}$$

Вариант 10.

$$A = \begin{pmatrix} -1 & 2 & 5 \end{pmatrix},\ B = \begin{pmatrix} 0 & 1 & 3 \\ 5 & 2 & 4 \\ -1 & 2 & 4 \end{pmatrix}$$

3.3 Обратная матрица. Критерий обратимости матрицы

Пусть A, B – квадратные матрицы n-ого порядка,

$$E = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & ... & 0 \\ ... & ... & ... & ... \\ 0 & 0 & ... & 1 \end{pmatrix}$$

– единичная матрица этого же порядка.

Если $AB = BA = E$, то матрица B называется *обратной матрицей* для матрицы A и обозначается A^{-1}.

Если существует обратная матрица A^{-1}, то матрица A называется *обратимой матрицей.*

Матрица A *обратима* тогда и только тогда, когда A – *невырожденная матрица*, т.е., когда ранг матрицы A равняется ее порядку.

Задание 3. Докажите,что матрица A обратима и найдите обратную матрицу A^{-1}.

Вариант 0.

$$A = \begin{pmatrix} 1 & 1 & -1 \\ 1 & -1 & 0 \\ 2 & -1 & -1 \end{pmatrix}$$

Решение. Для того,чтобы доказать,что матрица A обратима, следует выполнить следующие действия:

1. Найти обратную матрицу A^{-1} .
2. Найти обратную матрицу с помощью функции inv(A).
3. Выполнить проверку $AA^{-1} = A^{-1}A = E$, где E – единичная матрица.

Решение задачи в Octave.

```
>> A=[1 1 -1;1 -1 0;2 -1 -1]
A =
   1   1  -1
   1  -1   0
   2  -1  -1
>> B=A^(-1)
B =
   1.00000   2.00000  -1.00000
   1.00000   1.00000  -1.00000
   1.00000   3.00000  -2.00000
>> inv(A)
ans =
   1.00000   2.00000  -1.00000
   1.00000   1.00000  -1.00000
   1.00000   3.00000  -2.00000
>> A*B
ans =
   1.00000  -0.00000   0.00000
  -0.00000   1.00000   0.00000
  -0.00000  -0.00000   1.00000
>> B*A
ans =
   1.00000  -0.00000  -0.00000
   0.00000   1.00000  -0.00000
   0.00000  -0.00000   1.00000
```

Задача 3 для самостоятельного решения.

Вариант 1.

$$A=\begin{pmatrix} 2 & -1 & 2 \\ -1 & 2 & -2 \\ -1 & 1 & -2 \end{pmatrix}$$

Вариант 2.

$$A=\begin{pmatrix} 1 & 1 & 2 \\ 1 & 2 & 3 \\ 0 & 1 & 0 \end{pmatrix}$$

Вариант 3.

$$A=\begin{pmatrix} 1 & 1 & 0 \\ 1 & -1 & 1 \\ -1 & -4 & 1 \end{pmatrix}$$

Вариант 4.

$$A=\begin{pmatrix} 1 & 1 & 0 \\ 1 & 2 & 0 \\ -1 & 0 & 1 \end{pmatrix}$$

Вариант 5.

$$A=\begin{pmatrix} 1 & 1 & -1 \\ 1 & -1 & 0 \\ 2 & -1 & -1 \end{pmatrix}$$

Вариант 6.

$$A=\begin{pmatrix} -2 & 1 & 0 \\ 0 & 0 & -1 \\ -2 & -1 & -5 \end{pmatrix}$$

Вариант 7.

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

Вариант 8.

$$A=\begin{pmatrix} 1 & -1 & 0 \\ 0 & 0 & -1 \\ -2 & 3 & -2 \end{pmatrix}$$

Вариант 9.

$$A=\begin{pmatrix} 1 & 1 & -1 \\ 1 & 2 & 0 \\ -3 & 1 & 6 \end{pmatrix}$$

Вариант 10.

$$A=\begin{pmatrix} 2 & 1 & -2 \\ 1 & 2 & -2 \\ -1 & -1 & 2 \end{pmatrix}$$

Задание 4. Решите матричное уравнение $AX = B$(или $XA = B$),

где A-матрица из задания 3, $B=\begin{pmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ -1 & 1 & 1 \end{pmatrix}$,

затем найдите матрицу $X + 2B$.

Вариант 0.

$$A=\begin{pmatrix} 1 & 1 & -1 \\ 1 & -1 & 0 \\ 2 & -1 & -1 \end{pmatrix}$$

Решение.

А)$A \cdot X = B$

Если матрица А обратима, умножив обе части этого уравнения слева на матрицу обратную к А, оно примет вид:

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

Имеем

$$(A^{-1} \cdot A) \cdot X = A^{-1}.$$

Понятно, что матричное уравнение имеет единственное решение, если А и В - квадратные матрицы n- го порядка, и определитель матрицы А не равен нулю. Отсюда $X = A^{-1} \cdot B$

Б) $X \cdot A = B$

Аналогично, умножим обе части данного уравнения на A^{-1} справа, получим, что $X = B \cdot A^{-1}$

Данные матричные уравнения можно решить двумя способами в Octave.

Решение задачи в Octave.

```
>> % A*X=B
>> A=[1 1 -1;1 -1 0;2 -1 -1]
A =
   1    1   -1
   1   -1    0
   2   -1   -1
>> B=[1 -1 1; -1 1 -1; -1 1 1]
B =
   1   -1    1
  -1    1    -1
  -1    1    1
>> X=A\B
X =
   0    0    -2
   1   -1   -1
   0    0   -4
```

```
>> A*X % Проверка AX=B
ans =
   1  -1   1
  -1   1  -1
  -1   1   1
>> % 2 способ
>> A=[1 1 -1;1 -1 0;2 -1 -1];
>> B=[1 -1 1; -1 1 -1; -1 1 1];
>> C=inv(A)
C =
   1.00000   2.00000  -1.00000
   1.00000    1.00000  -1.00000
   1.00000    3.00000   -2.00000
>> X=C*B;
>> A*X
ans =
   1.00000 -1.00000   1.00000
  -1.00000  1.00000  -1.00000
  -1.00000  1.00000   1.00000
>> X+2*B
ans =
   2  -2   0
  -1   1  -3
  -2   2  -2
```

Дополнительные упражнения для самостоятельной работы.

Задание 5. Для произвольных невырожденных матриц А и В, докажите равносильность следующих равенств.

$$\begin{array}{ll} AB = BA, & AB^{-1} = B^{-1}A, \\ A^{-1}B = BA^{-1}, & A^{-1}B^{-1} = B^{-1}A^{-1}. \end{array}$$

Задание 6. Пусть А- матрица из задания 1. Через A^T обозначим матрицу, полученную из матрицы А путем транспонирования. Докажите равносильность следующих равенств.

$$\begin{array}{ll} (A+B)^T = A^T + B^T, & (\lambda A)^T = \lambda \cdot A^T (\lambda \in \Re), \\ (A \cdot B)^T = B^T \cdot A^T & (A^{-1})^T = (A^T)^{-1} \end{array}$$

3.4 Основные понятия и метод Гаусса

Система m уравнений с n неизвестными вида

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n = b_2 \\ ... ... \\ a_{m1}x_1 + a_{m2}x_2 + ... + a_{mn}x_n = b_m \end{cases}$$

называется *системой линейных алгебраических уравнений* (СЛАУ), причём x_j - неизвестные, a_{ij} - коэффициенты при неизвестных, b_i - свободные коэффициенты $(i = 1...m, j = 1...n)$.

Решением системы линейных уравнений называется арифметический n-мерный вектор $\alpha = (\alpha_1, \alpha_2, ..., \alpha_n)$, координаты которого удовлетворяют каждому уравнению системы.

Матрица

$$A = \begin{pmatrix} a_{11} & a_{12} & ... & a_{1n} \\ a_{21} & a_{22} & ... & a_{2n} \\ ... & ... & ... & ... \\ a_{m1} & a_{m2} & ... & a_{mn} \end{pmatrix},$$

составленная из коэффициентов при неизвестных, называется *основной матрицей* данной системы линейных уравнений.

Матрица,которая получается из основной добавлением столбца из свободных членов, называется *расширенной матрицей*. Таким образом, расширенная матрица данной системы линейных уравнений имеет вид

$$\bar{A} = \begin{pmatrix} a_{11} & a_{12} & ... & a_{1n} & b_1 \\ a_{21} & a_{22} & ... & a_{2n} & b_2 \\ ... & ... & ... & ... & ... \\ a_{m1} & a_{m2} & ... & a_{mn} & b_m \end{pmatrix}$$

Система линейных уравнений, имеющая хотя бы одно решение называется *совместной*.

Система линейных уравнений, не имеющая ни одного решения называется *несовместной*.

Система линейных уравнений, юмеющая единственное решение, называется *определенной.*

Система линейных уравнений, имеющая бесконечно много решений, называется *неопределенной.*

Две системы линейных уравнений называются *равносильными*, если множества их решений совпадают.

При определении совместности систем уравнений важную роль играет понятие ранга матрицы. Пусть дана матрица A размером $m \times n$. Вычеркиванием некоторых строк или столбцов из неё можно получить квадратные матрицы k-го порядка, определители которых называются минорами порядка k матрицы A. Наивысший порядок не равных нулю миноров матрицы A называют рангом матрицы и обозначают $r(A)$.

Из определения вытекает, что $r(A) = min(m, n)$, а $r(A) = 0$, только если матрица нулевая и $r(A) = n$ для невырожденной матрицы n-го порядка.

При элементарных преобразованиях (перестановка строк матрицы, умножение строк на число отличное от нуля и сложение строк) ранг матрицы не изменяется.

Итак, если речь идёт об исследовании системы на совместность, следует помнить, что система m линейных уравнений с n неизвестными:

- *несовместна*, если $r(\bar{A}) > r(A)$;
- *совместна*, если $r(\bar{A}) = r(A)$, причём при $r(\bar{A}) = r(A) = m$ имеет единственное решение;
- при $r(\bar{A}) = r(A) < m$ имеет бесконечно много решений.

Решение системы линейных уравнений при помощи метода Гаусса основывается на том, что с помощью элементарных преобразований данная система линейных уравнений приводится к ступенчатому виду, затем находится решение полученной системы.

Задание 7. Исследовать данную систему уравнений на совместность. В случае совместности решить ее методом Гаусса.

Вариант 0.

$$\begin{cases} 2x_1 + x_2 - 5x_3 + x_4 = 8 \\ x_1 - 3x_2 - 6x_4 = 9 \\ 2x_2 - x_3 + 2x_4 = -5 \\ x_1 + 4x_2 - 7x_3 + 6x_4 = 0 \end{cases}$$

Решение.

Порядок решения задачи в Octave следующий:

1. Сформировать матрицу коэффициентов A и вектор свободных членов b заданной системы;
2. Сформировать расширенную матрицу системы, объединив A и b;
3. Вычислить ранг основной и расширенной матрицы с помощью функции $rank$.
4. Вычислить размерность основной матрицы. Сделать вывод о совместности системы.
5. Используя функцию $rref$ привести расширенную матрицу к ступенчатому виду;
6. Найти решение системы, выделив последний столбец матрицы, полученной в предыдущем пункте;
7. Выполнить вычисление $Ax - b$. Если в результате получился нулевой вектор, задача решена верно.

Решение задачи в Octave.

```
>>A=[2 1 -5 1;1 -3 0 -6;0 2 -1 2;1 4 -7 6]; %Матрица системы
>> b=[ 8 ; 9 ; -5 ; 0 ];  %Вектор свободных коэффициентов
```

```
>> C=rref([A b]) %Расширенная матрица системы, приведенная к
 ступенчатому виду
C =
   1.00000   0.00000   0.00000   0.00000   3.00000
   0.00000   1.00000   0.00000   0.00000  -4.00000
   0.00000   0.00000   1.00000   0.00000  -1.00000
   0.00000   0.00000   0.00000   1.00000   1.00000
>> n=size(C) %Размерность матрицы C
n =
   4   5
>> x=C(:,n(2)) %Вектор решений СЛАУ Ax=b
x =
   3.00000
  -4.00000
  -1.00000
   1.00000
>> A*x-b  %Проверка Ax-b
ans =
  0.0000e+000
  1.7764e-015
  0.0000e+000
  -2.2204e-015
```

Задание 7 для самостоятельного решения.

Вариант 1.

$$\begin{cases} 3x_1 + 4x_2 - x_3 - 3x_4 - 3x_5 = 3 \\ 2x_1 + 3x_2 - 2x_3 + 3x_4 + x_5 = 4 \\ 5x_1 + 7x_2 - 3x_3 - 2x_5 = 7 \\ x_1 + x_2 + x_3 - 6x_4 - 4x_5 = -1 \end{cases}$$

Вариант 2.

$$\begin{cases} 2x_1 - x_3 + x_4 - 2x_5 = -3 \\ -x_1 + 2x_2 + 3x_3 - 3x_4 + 3x_5 = 1 \\ x_1 + 2x_2 + 2x_3 - 2x_4 + x_5 = -2 \\ 3x_1 - 2x_2 - 4x_3 + 4x_4 - 5x_5 = -4 \end{cases}$$

Вариант 3.

$$\begin{cases} 2x_1 + 3x_3 + 3x_4 + 3x_5 = -1 \\ x_1 - x_2 + 3x_3 - 2x_5 = 1 \\ 3x_1 - x_2 + 6x_3 + 3x_4 = 0 \\ x_1 + x_2 + 3x_4 = -2 \end{cases}$$

Вариант 4.

$$\begin{cases} 2x_1 + x_2 + x_3 - 3x_4 = -1 \\ x_1 + 3x_2 - x_4 - 2x_5 = 2 \\ 3x_1 + 4x_2 + x_3 - 4x_4 - 2x_5 = 1 \\ x_1 - 2x_2 + x_3 - 2x_4 + 2x_5 = -3 \end{cases}$$

Вариант 5.

$$\begin{cases} x_1 - 2x_2 - 35x_3 + 3x_4 - x_5 = 3 \\ 2x_1 - 3x_2 + x_3 + 2x_4 - 2x_5 = 3 \\ 3x_1 - 5x_2 - 2x_3 + 5x_4 - 3x_5 = 6 \\ -x_1 + x_2 - 4x_3 + x_4 + x_5 = 0 \end{cases}$$

Вариант 6.

$$\begin{cases} 4x_1 + 3x_2 + 3x_3 + 3x_4 - x_5 = 3 \\ 3x_1 + 2x_2 + x_3 + 4x_4 - 2x_5 = 4 \\ 7x_1 + 5x_2 + 4x_3 + 7x_4 - 3x_5 = 7 \\ x_1 + x_2 + 2x_3 - x_4 + x_5 = -1 \end{cases}$$

Вариант 7.

$$\begin{cases} 7x_1 - 4x_2 - 3x_3 - 3x_4 + 2x_5 = -3 \\ x_1 - 2x_2 + 2x_3 - x_4 - 7x_5 = 6 \\ 8x_1 - 6x_2 - x_3 - 4x_4 - 5x_5 = 3 \\ 6x_1 - 2x_2 - 5x_3 - 2x_4 + 9x_5 = -9 \end{cases}$$

Вариант 8.

$$\begin{cases} 4x_1 - 3x_2 + x_3 - 3x_4 + 3x_5 = 3 \\ -5x_1 - 4x_2 + 4x_3 + x_4 + 2x_5 = 2 \\ -x_1 7x_2 + 5x_3 - 2x_4 + 5x_5 = 5 \\ 9x_1 + x_2 - 3x_3 - 4x_4 + x_5 = 1 \end{cases}$$

Вариант 9.

$$\begin{cases} 2x_1 - 3x_2 - 2x_3 - 3x_4 + x_5 = 3 \\ -5x_1 - 4x_2 - 3x_3 + 4x_4 + 3x_5 = 4 \\ -3x_1 - 7x_2 - 5x_3 + x_4 + 4x_5 = 7 \\ 7x_1 + x_2 + x_3 - 7x_4 - 2x_5 = -1 \end{cases}$$

Вариант 10.

$$\begin{cases} x_1 - 3x_2 - 2x_3 + x_4 + 3x_5 = -3 \\ -2x_1 - 4x_2 + x_3 + 2x_4 + 4x_5 = 2 \\ x_1 + 7x_2 + x_3 - 3x_4 - x_5 = 1 \\ 3x_1 + x_2 - 3x_3 - x_4 - 7x_5 = -5 \end{cases}$$

3.5 Правило Крамера

Рассмотрим еще один способ решения систем линейных уравнений, который применяется, если

1. число уравнений m равняется числу неизвестных n,

2. основная матрица A невырожденная, т.е. $\det A \neq 0$.

Пусть дана система уравнений, удовлетворяющая данным условиям:

$$\bar{A} = \begin{pmatrix} a_{11} & a_{12} & ... & a_{1n} & b_1 \\ a_{21} & a_{22} & ... & a_{2n} & b_2 \\ ... & ... & ... & ... & ... \\ a_{n1} & a_{n2} & ... & a_{nn} & b_n \end{pmatrix}$$

Обозначим $D = \det A$, где

$$A = \begin{pmatrix} a_{11} & a_{12} & ... & a_{1n} \\ a_{21} & a_{22} & ... & a_{2n} \\ ... & ... & ... & ... \\ a_{n1} & a_{n2} & ... & a_{nn} \end{pmatrix}$$

,

$$D_1 = \det A_1, \text{ где } A_1 = \begin{pmatrix} b_1 & a_{12} & ... & a_{1n} \\ b_2 & a_{22} & ... & a_{2n} \\ ... & ... & ... & ... \\ b_n & a_{n2} & ... & a_{nn} \end{pmatrix},$$

$$D_2 = \det A_2, \text{ где } A_2 = \begin{pmatrix} a_{11} & b_1 & ... & a_{1n} \\ a_{21} & b_2 & ... & a_{2n} \\ ... & ... & ... & ... \\ a_{n1} & b_n & ... & a_{nn} \end{pmatrix},$$

$$D_n = \det A_n, \text{ где } A_n = \begin{pmatrix} a_{11} & a_{12} & ... & b_1 \\ a_{21} & a_{22} & ... & b_2 \\ ... & ... & ... & ... \\ a_{n1} & a_{n2} & ... & b_n \end{pmatrix}.$$

Тогда $x_1 = \frac{D_1}{D} = \alpha_1, x_2 = \frac{D_2}{D} = \alpha_2, ..., x_n = \frac{D_n}{D} = \alpha_n$ и единственным решением данной системы линейных уравнений будет арифметический n-мерный вектор $(\alpha_1, \alpha_1, ..., \alpha_n)$.

Задание 8. Данную систему уравнений решите по правилу Крамера.

Вариант 0.

$$\begin{cases} x_1 + x_2 + 2x_3 = -1 \\ 2x_1 - x_2 + 5x_3 = -4 \\ 4x_1 + x_2 + 4x_3 = -2 \end{cases}$$

Решение.

Итак, для решения поставленной задачи необходимо выполнить следующие действия:

1. Представить систему в матричном виде, то есть сформировать матрицу системы A и вектор правых частей b.
2. Вычислить главный определитель $\det A$.
3. Сформировать вспомогательные матрицы для вычисления определителей $\det A_n$.
4. Вычислить определители $\det A_n$.
5. Найти решение системы по формуле $x_i = \frac{\det A_n}{\det A}$.

Решение задачи в Octave.

```
>> A=[1 1 2; 2 -1 2; 4 1 4];  #задаем матрицу системы
>> b=[-1 ;-4;-2]; # вектор свободных коэффициентов системы
>> D=det(A)  # главный определитель
D =  6
>> A1=A; A1(:,1)=b %задаем вспомогательные матрицы
A1 =
  -1   1   2
  -4  -1   2
  -2   1   4
>> A2=A; A2(:,2)=b
A2 =
```

```
   1  -1   2
   2  -4   2
   4  -2   4
>> A3=A; A3(:,3)=b
A3 =
   1   1  -1
   2  -1  -4
   4   1  -2
>> d(1)=det(A1) %вычисляем вспомогательные определители
d =  6.0000
>> d(2)=det(A2)
d =
    6.0000   12.0000
>> d(3)=det(A3)
d =
    6.0000   12.0000  -12.0000
>> X=d/D
X =
   1.00000   2.00000  -2.00000
```

Задание 8 для самостоятельного решения.

Вариант 1.

$$\begin{cases} 2x + 3y + 5z = -3 \\ 5x + 5y + 11z = 5 \\ 2x + y + 4z = 0 \end{cases}$$

Вариант 2.

$$\begin{cases} x - y + z = -5 \\ 5x + z = 0 \\ x + 2y + 2z = -3 \end{cases}$$

Вариант 3.

$$\begin{cases} 5x + 4y + 11z = 5 \\ x + y + 3z = 3 \\ 5x + 4y + 10z = -3 \end{cases}$$

Вариант 4.

$$\begin{cases} 4x + 5y + 9z = 5 \\ x + y + z = -1 \\ 6x + 7y + 12z = -4 \end{cases}$$

Вариант 5.

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

Вариант 6.

$$\begin{cases} -2x - 3y - 3z = 3 \\ -4x - 4y - 9z = 3 \\ 5x + 6y + 10z = 1 \end{cases}$$

Вариант 7.

$$\begin{cases} 2x + y + 3z = 3 \\ 4x + 4y + 9z = 1 \\ -5x - 4y - 10z = 5 \end{cases}$$

Вариант 8.

$$\begin{cases} 5x + 6y + 11z = -3 \\ x + y + 3z = 4 \\ 5x + 6y + 10z = 4 \end{cases}$$

Вариант 9.

$$\begin{cases} 5x + 6y + 9z = 0 \\ -4x - 4y - 9z = 3 \\ 8x + 9y + 16z = 2 \end{cases}$$

Вариант 10.

$$\begin{cases} 3x + 4y + 5z = 1 \\ 2x + 2y + 5z = 4 \\ 4x + 5y + 8z = 2 \end{cases}$$

3.6 Решение систем линейных уравнений в матричной форме

Система n-линейных уравнений с n-неизвестными может быть записана в матричной форме, т.е. в виде $AX = B$, где A-основная матрица данной системы уравнений, $X = \begin{pmatrix} x_1 \\ x_2 \\ ... \\ x_n \end{pmatrix}$- столбец из неизвестных, $B = \begin{pmatrix} b_1 \\ b_2 \\ ... \\ b_n \end{pmatrix}$-столбец из свободных членов. Если A-невырожденная матрица (т.е. $\det A \neq 0$ или $r(A) = n$), то существует обратная матрица A^{-1}. Тогда, умножив обе части матричного уравнения слева на матрицу A^{-1}, получим $A^{-1} \cdot A \cdot X = A^{-1} \cdot B$, т.е. $X = A^{-1} \cdot B$. Таким образом, будет найдено единственное решение исходной системы линейных уравнений.

Задание 9. Решить в матричной форме систему уравнений, приведенную в Задании 8.

Вариант 0.

$$\begin{cases} x_1 + x_2 + 2x_3 = -1 \\ 2x_1 - x_2 + 5x_3 = -4 \\ 4x_1 + x_2 + 4x_3 = -2 \end{cases}$$

Решение.

Чтобы решить данную систему линейных уравнений в матричной форме необходимо выполнить следующие действия:

1. Сформировать матрицу коэффициентов и вектор свободных членов заданной системы.

2. Решить систему, представив вектор неизвестных как произведение матрицы обратной к матрице системы и вектора свободных членов.

Решение задачи в Octave.

```
>> A=[1 1 2;2 -1 2;4 1 4]; %Матрица системы:
>> b=[ -1 ; -4 ; -2 ]; %Вектор свободных коэффициентов
>> x=A^(-1)*b %Вектор решений СЛАУ Ax=b
x =
   1
   2
  -2
>> x=inv(A)*b %Вектор решений СЛАУ Ax=b с помощью функции
inv(A)
x =
   1
   2
  -2
>> A*x %Проверка Ax=b
ans =
  -1
  -4
  -2
```

4. Построение двумерных графиков

Двумерная графика подразумевает график, точки которого определяются двумя величинами. Для построения графиков чаще всего используются декартова и полярная системы координат.

Для построения графика в среде программирования Octave для начала открываем окно редактора, в котором необходимо написать саму программу. Когда программа написана, нажимаем кнопку сохранить файл и запустить. После запуска программы перед нами появится вспомогательное окно с заданным графиком.

Для наглядности мы можем менять тип, маркер, размер линии, а так же при необходимости выводить легенду.

Таблица 2.2

Символ маркера	**Изображение маркера**
·	· (точка)
*	⋇
x	×
+	+
o	⊙
s	■
d	⧫
v	▼
^	▲
<	▽
>	△
p	□
h	◊

Цвет линии определяется буквой латинского алфавита, что соответствует первой букве цветов в английском языке для легкого запоминания.

Таблица 2.3

Символ	**Цвет линии**
y	желтый
m	розовый
c	голубой
r	красный
g	зелёный
b	синий
w	белый

За размер маркера отвечает команда $''markersize''$, размер которого пишется через запятую, после команды, например $''markersize'', 5$. Так же в Octave есть функция $pause(n)$, которая приостанавливает построение графика на n секунд. Так же в Octave существуют другие функции, такие как:

- $grid\ on/off$ – снимает/наносит сетку
- $axis[xmin, xmax, ymin, ymax]$выводит только ту часть графика, которую мы задаем
- команда $title('$Заголовок$')$ выводит заголовок графика
- $xlabel('$Подпись под осью $x')$, $ylabel('$Подпись под осью $y')$служат для подписей осей x и y
- функция $text(x, y, '$текст$')$ выводит текст левее точки с координатами (x, y)
- функция $legend('$легенда1$', '$легенда2$', \ldots, '$легенда$n', m)$выводит легенды для каждого из графиков, параметр m определяет месторасположение легенды в графическом окне: 1 – в правом верхнем углу графика (значение по умолчанию); 2 – в левом верхнем углу графика; 3 – в левом нижнем углу графика; 4 – в правом нижнем углу графика.

При выводе текста с помощью функций $xlabel$, $ylabel$,$title$, $text$ можно выводить греческие буквы, использовать символы верхнего и нижнего индекса. Для вывода текста в верхнем индексе используется символ << ^ >>, в нижнем – символ << _ >>. При работе с текстом можно также использовать синтаксис TEX.

4.1 Построение графиков в прямоугольной системе координат

Для того, чтобы построить график в прямоугольной системе координат для начала необходимо сформировать массив x, после чего массив y. Затем для построения графика используем функцию *plot*. Когда программа написана, нажимаем кнопку сохранить файл и запустить.

Задание 10. Вывести на экран график функции двумя способами.

Вариант 0. $7\sin(x) + 9$

Решение задачи в Octave.

Способ первый: задаем в окне редактора функцию, сформировав сперва массив x, массив y и после этого используем функцию *plot*

$$x = -20 : 0.1 : 20;$$
$$y = 7 * \sin(x) + 9;$$
$$\text{plot(x,y)}$$

Способ второй (полная форма функции *plot*): $plot(x_1, y_1)$, где x_1 - массив абсциссы графика, y_1 - массив ординаты графика

$$plot(x = -20 : 0.1 : 20, y = 7 * \sin(x) + 9)$$

В результате получаем:

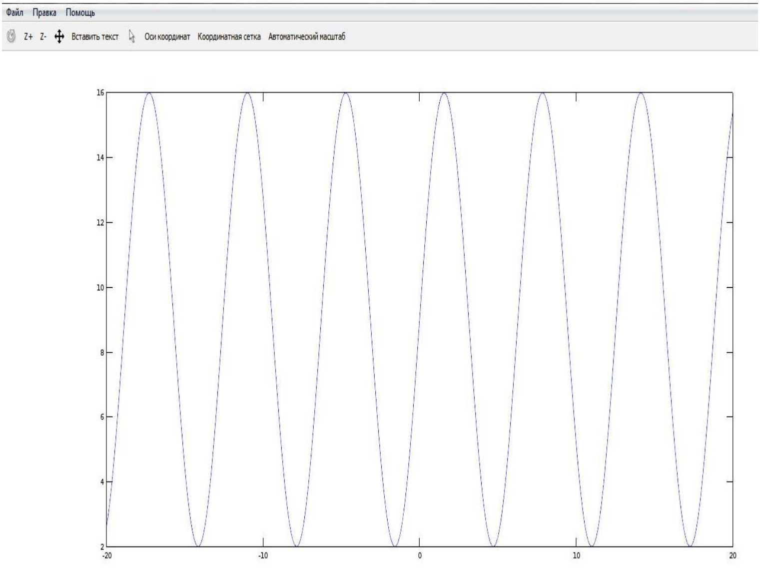

Рисунок 1. График функции $7\sin(x) + 9$

Вариант 1. $y = -x^2 + 4$
Вариант 2. $y = \cos(x)$
Вариант 3. $y = 2x^2 - 8x + 5$
Вариант 4. $y = 9\cos 3x + 2$
Вариант 5. $y = 1 + \frac{3}{x}$
Вариант 6. $y = 2\operatorname{tg}(x)$
Вариант 7. $y = -2x^2 + 7$
Вариант 8. $y = 4x^2 + 2$
Вариант 9. $y = \sin(x)$
Вариант 10. $y = -2x$

Задание 11. Последовательно вывести графики различных цветов с заголовками графиков и задержкой в 5 секунд.

Вариант 0. $y = \sin(x)$, $y = \cos(x)$
Решение задачи в Octave.

Чтобы вывести графики функций последовательно, необходимо для начала создать графическое окно с дескриптором $okno1$, далее последовательно выполняем шаги, как показано в предыдущем примере. После команды $plot$ выводим линии сетки, используя команду $grid\ on$ и затем выводим заголовок командой $title$. Чтоб приостановить построение графика на n секунд необходимо использовать функцию $pause(n)$, где n - количество секунд. Эта процедура проводится для каждой заданной функции. Для того, чтобы в конце процедуры командное окно было закрыто, надо ввести команду $delete(okno1)$

$$\begin{aligned}
&okno1 = figure();\\
&x = -6*\mathrm{pi}():\mathrm{pi}()/50:\mathrm{pi}();\\
&y = \sin(x);\\
&plot(x, y,' k');\\
&grid\ on;\\
&title('Plot\ y = \sin(x)');\\
&pause(5);\\
&y = \cos(x);\\
&plot(x, y,' b');\\
&grid\ on;\\
&title('Plot\ y = \cos(x)');\\
&pause(5);\\
&delete(okno1);
\end{aligned}$$

В результате получаем:

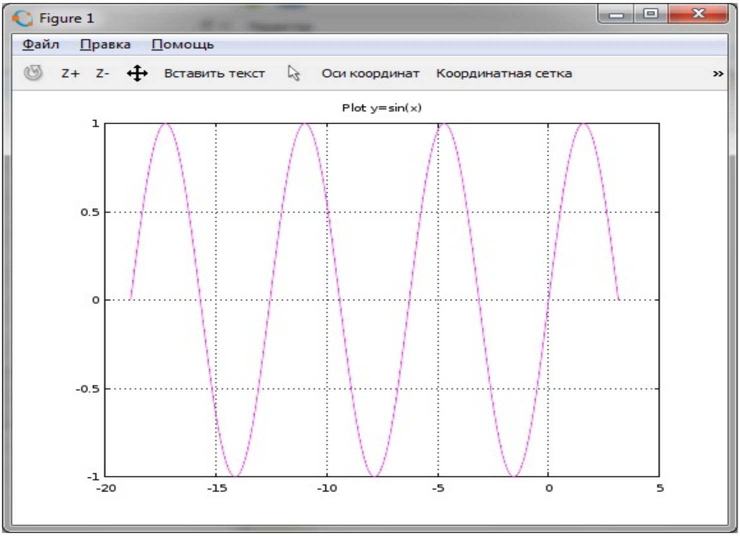

Рисунок 2. График функции $y = \sin(x)$

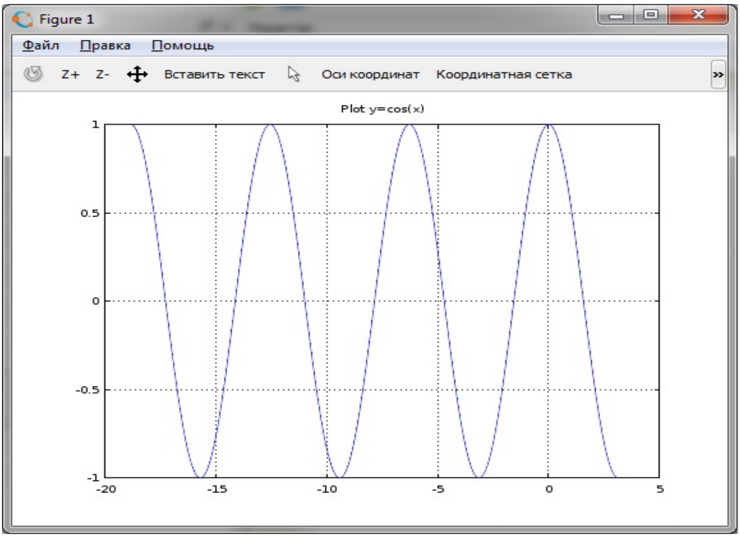

Рисунок 3. График функции $y = \cos(x)$

Вариант 1. $y = \operatorname{tg}(x)$, $y = 2\operatorname{tg}(x)$, $y = -x^2 + 4$
Вариант 2. $y = -\frac{x}{2} - 1$, $y = 2x + 3$, $y = 4x - 3$
Вариант 3. $y = 2x^2 - 8x + 5$, $y = \cos(x)$, $y = 4x^2 - 4x - 5$
Вариант 4. $y = 1 + \frac{3}{x}$, $y = 1 - \frac{2}{x-1}$, $y = 5 + \frac{3}{4x}$
Вариант 5. $y = \sin(x)$, $y = \cos(x)$, $y = \tan(x)$
Вариант 6. $y = \arcsin(x)$, $y = \arccos(x)$, $y = \arctan(x)$
Вариант 7. $y = \sinh(x)$, $y = \cosh(x)$, $y = \tanh(x)$
Вариант 8. $y = \sin(x)$, $y = \cos(3x)$, $y = \tanh(x)$
Вариант 9. $y = \sinh(x)$, $y = -2x^2 + 7$, $y = -2x$
Вариант 10. $y = 3x$, $y = 3\cos(x)$, $y = 4x^2 + 2$

4.2 Построение графиков в полярной системе координат

Полярная система координат состоит из заданной фиксированной точки O, называемой полюсом, концентрических окружностей с центром в полюсе и лучей, выходящих из точки O, один из которых, OM, называют полярной осью. Положение любой точки M в полярных координатах можно задать положительным числом $\rho = |OM|$ (полярный радиус) и числом ϕ, равным величине угла $\angle XOM$(полярный угол). Числа ρ и ϕназывают полярными координатами точки M и обозначают $M(\rho, \phi)$

Для формирования графика в полярной системе координат необходимо сформировать массивы значений полярного угла и полярного радиуса и обратиться к функции $polar$: $polar(\phi, r_0, s)$, где ϕ - массив полярных углов; r_0 - массив полярных радиусов; s - строка, состоящая из трёх символов, которые определяют цвет линии, тип маркера и тип линии.

Задание 12. Построить график функции в полярной системе координат.

Вариант 0. Построить график лемнискаты $\rho = 3\sqrt{2\cos(2\phi)}$, функция ρ определена при $-\frac{\pi}{2} \leq \phi \leq \frac{\pi}{2}$ $(\cos 2\phi \geq 0)$

Решение задачи в Octave.

$$fi = -\text{pi}/4 :\text{pi}/200 :\text{pi}/4;$$
$$ro = 3{*}\text{sqrt}(2 * \cos(2{*}fi));$$
$$polar(fi, ro,' r');$$
$$hold\ on;$$
$$polar(fi, -ro,' r');$$
$$grid\ on;$$

Полученный график:

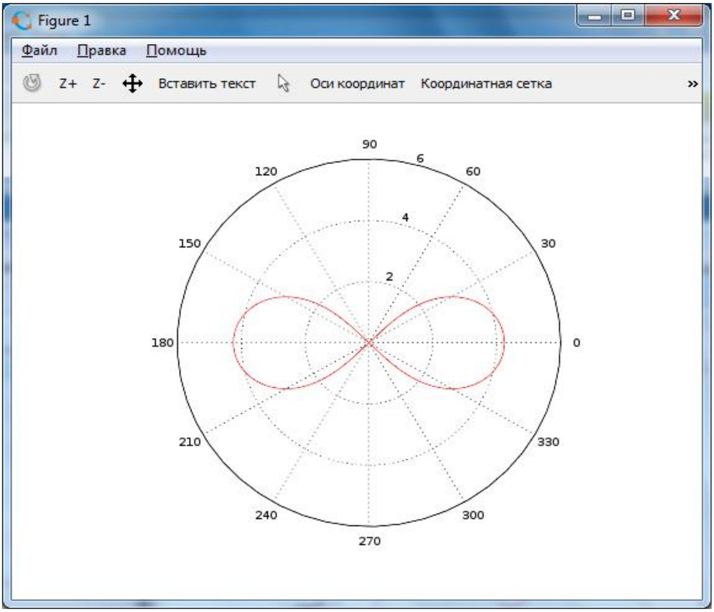

Рисунок 4. График лемнискаты $\rho = 3\sqrt{2\cos(2\phi)}$

Вариант 1. $r(t) = \sin(\frac{3}{4}t)$, $t \in [0; 8\pi]$

Вариант 2. $r(t) = 2t$, $t \in [0; 8\pi]$

Вариант 3. $r(t) = 1 - \sin(t)$, $t \in [0; 8\pi]$

Вариант 4. $r(t) = 2 - 4\sin(t)$, $t \in [0; 2\pi]$

Вариант 5. $r(t) = \frac{1}{1-\cos(t)}$, $t \in [0; 2\pi]$

Вариант 6. $r(t) = \sin(6t)$, $t \in [0; 2\pi]$

Вариант 7. $r(t) = \sin(\frac{7}{4}t)$, $t \in [0; 8\pi]$
Вариант 8. $r(t) = 1$, $t \in [0; 2\pi]$
Вариант 9. $r(t) = 3$, $t \in [0; 2\pi]$
Вариант 10. $r(t) = \sin(8t)$, $t \in [0; 2\pi]$

4.3 Построение графиков, заданных параметрически

Задание функции $y(x)$ с помощью равенств $x = f(t)$ и $y = g(t)$ называют параметрическим, а вспомогательную величину t - параметром. Построение графика функции, заданной параметрически, можно осуществлять следующим образом:

1. Определить массив t;

2. Определить массивы $x = f(t)$ и $y = g(t)$;

3. Построить график функции $y(x)$ с помощью функции $plot(x, y)$.

Задание 13. Построить график функции, заданный параметрически.

Вариант 0. Построить график эпициклоиды $x = 4\cos(t) - \cos(4t)$, $y = 4\sin(t) - \sin(4t)$, $t \in [0; 2\pi]$.
Решение задачи в Octave.

$t = 0 : \mathrm{pi}/50 : 2{*}\mathrm{pi};$
$x = 4 * \cos(t) - \cos(4 * t);$
$x = 4 * \sin(t) - \sin(4 * t);$
$plot(x, y);$
$grid\ on;$

Полученный график:

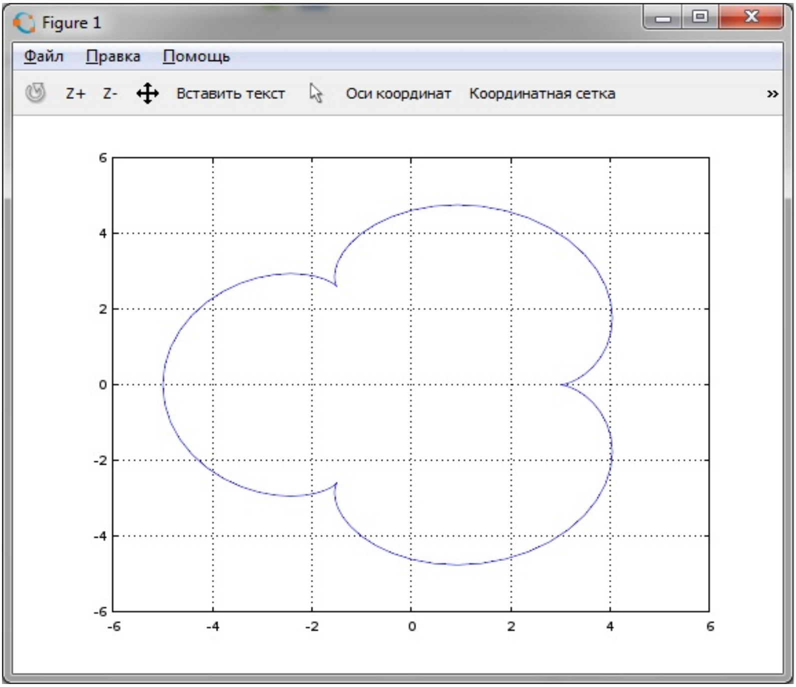

Рисунок 5. График эпициклоиды $x = 4\cos(t) - \cos(4t)$, $y = 4\sin(t) - \sin(4t)$

Вариант 1.

$$\begin{cases} x = 13(\cos(t) + \frac{\cos(6{,}5t)}{6{,}5}) \\ y = 13(\sin(t) + \frac{\sin(6{,}5t)}{6{,}5}), \ \ t \in [0; 4\pi]; \end{cases}$$

Вариант 2.

$$\begin{cases} x = t\sin(t) \\ y = t\cos(t), \ \ t \in [0; 5\pi]; \end{cases}$$

Вариант 3.

$$\begin{cases} x = 2\cos(t) + \cos(2t) \\ y = 2\sin(t) - \sin(2t), \ \ t \in [0; 2\pi]; \end{cases}$$

Вариант 4.

$$\begin{cases} x = 2\sin^3(t) \\ y = 2\cos^3(t),\ t \in [0; 2\pi]; \end{cases}$$

Вариант 5.

$$\begin{cases} x = 20(\cos(t) + \frac{\cos(5t)}{5}) \\ y = 20(\sin(t) + \frac{\sin(5t)}{5}),\ t \in [0; 2\pi]; \end{cases}$$

Вариант 6.

$$\begin{cases} x = 4,4(\cos(t) + \frac{\cos(1{,}1t)}{1{,}1}) \\ y = 4,4(\sin(t) + \frac{\sin(1{,}1t)}{1{,}1}),\ t \in [0; 20\pi]; \end{cases}$$

Вариант 7.

$$\begin{cases} x = 24,8(\cos(t) + \frac{\cos(6{,}2t)}{6{,}2}) \\ y = 24,8(\sin(t) + \frac{\sin(6{,}2t)}{6{,}2}),\ t \in [0; 10\pi]; \end{cases}$$

Вариант 8.

$$\begin{cases} x = (1 + cos(t))cos(t) \\ y = (1 + cos(t))sin(t),\ t \in [0; 2\pi]; \end{cases}$$

Вариант 9.

$$\begin{cases} x = 6\cos(t) - 4\cos^3(t) \\ y = 4\sin^3(t),\ t \in [0; 2\pi]; \end{cases}$$

Вариант 10.

$$\begin{cases} x = 6,2(\cos(t) + \frac{\cos(3{,}1t)}{3{,}1}) \\ y = 6,2(\sin(t) + \frac{\sin(3{,}1t)}{3{,}1}),\ t \in [0; 20\pi]; \end{cases}$$

4.4 Построение гистограммы

Так же в Octave существует еще одна функция bar, которая предназначена для построение гистограммы. Функция $bar(y)$ выводит элементы массива y в виде гистограммы, а качестве массива x выступает массив номеров элементов массива y. Функция $bar(x, y)$ выводит гистограмму элементов массива y в виде столбцов в позициях, определяемых массивом x элементы которого должны быть упорядочены в порядке возрастания.

Задание 15. Построить гистограмму прогноза погоды.

Вариант 0. Температура в октябре была соответственно 5, 6, 7, 8, 9, 8, 7, 6, 5, 4, 3 градуса

Решение задачи в Octave.

$$y = [5; 6; 7; 8; 9; 8; 7; 6; 5; 4; 3];$$
$$bar(y);$$

Полученная гистограмма:

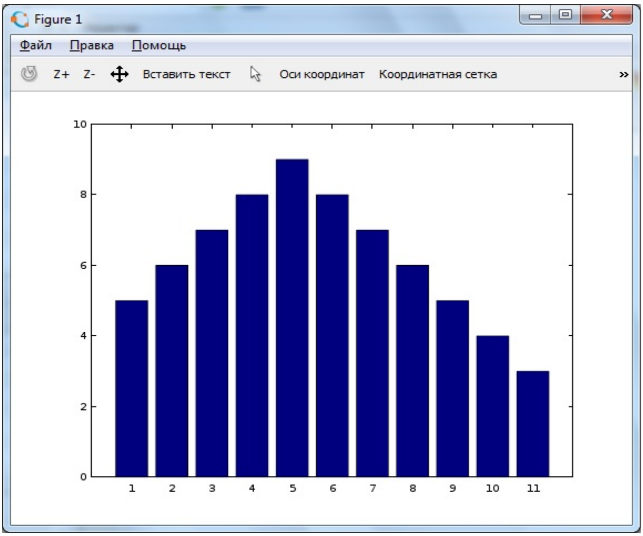

Рисунок 6. Гистограмма прогноза погоды

4.5 Анимация

При изучении движения тела на плоскости Octave позволяет построить график движения и проследить за движением. Построить анимационный ролик можно с помощью функции $comet(x, y)$, которая позволяет увидеть движение вдоль кривой $y(x)$ на плоскости.

Задание 16. Построить движение точки вдоль графика, который представлен в задаче 10.

Вариант 0. $y = \sin(x)$

Решение задачи в Octave.

$$x = 0 : pi/30 : 6 * pi$$
$$y = \sin(x)$$
$$comet(x, y)$$

Полученная анимация:

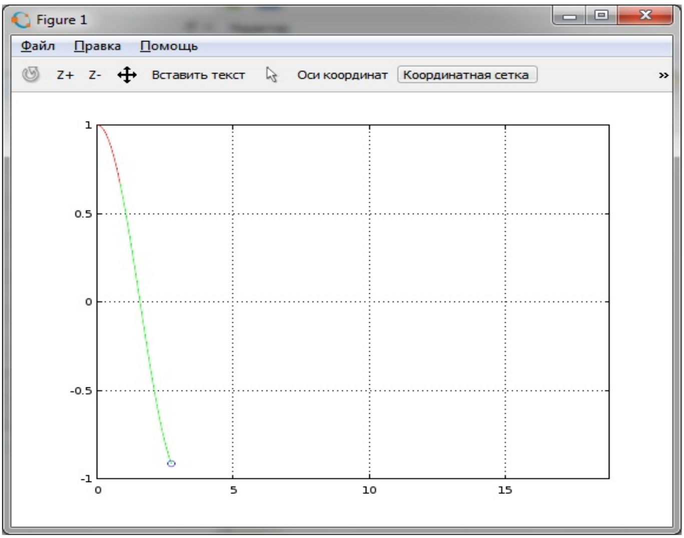

Рисунок 7. Начало движения точки

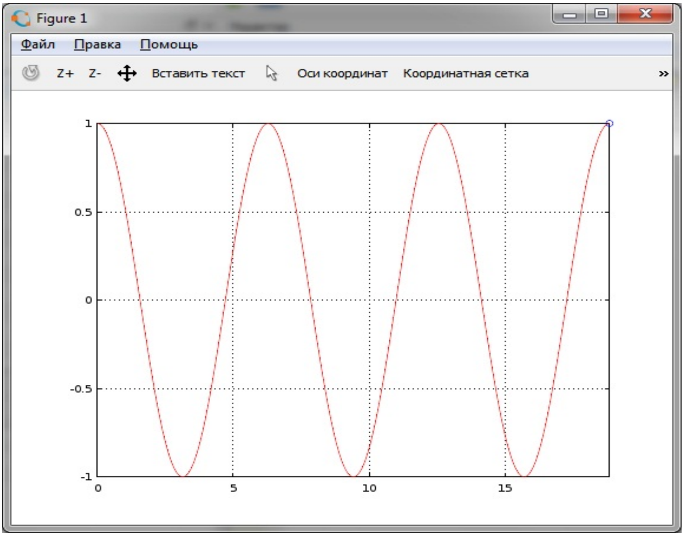

Рисунок 8. Конечное положение точки

5. Построение трёхмерных графиков

График поверхности (трёхмерный или 3D-график) - это график, положение точки в котором определяется значениями трёх координат.

5.1 Построение графиков поверхностей

Дадим определение прямоугольной (или декартовой) системе координат в пространстве. Прямоугольная система координат в пространстве состоит из заданной фиксированной точки O пространства, называемой началом координат, и трёх перпендикулярных прямых пространства OX, OY, OZ, не лежащих в одной плоскости и пересекающихся в начале координат, - их называют координатными осями (OX - ось абсцисс, OY - ось ординат,OZ - ось аппликат). Положение точки M в пространственной системе координат определяется значением трёх координат и обозначается $M(x, y, z)$. Три плоскости, содержащие пары координатных осей, называются координатными плоскостями XY, XZ, YZ.

Величина z называется функцией двух величин x и y, если каждой паре чисел, которые могут быть значениями переменных x и y, соответствует одно или несколько определенных значений величины z. При этом переменные x и y называют аргументами функции $z(x, y)$. Пары тех чисел, которые могут быть значениями аргументов x, y функции $z(x, y)$, в совокупности составляют область определения этой функции.

Для построение графика двух переменных $z = f(x, y)$ необходимо выполнить следующие действия.

1. Сформировать в области построения графика прямоугольную сетку, проводя прямые, параллельные осям $y = y_j$ и $x = x_i$, где $x_i = x_0 + ih, h = \frac{x_n - x_0}{n}, i = 0, 1, 2, ..., n,$

   $y_j = y_0 + jb, b = \frac{y_k - y_0}{k}, j = 0, 1, 2, ..., k.$

2. Вычислить значения $z_{i,j} = f(x_i, y_j)$ во всех узлах сетки.

3. Обратиться к функции построения поверхности, передавая ей в качестве параметров сетку и матрицу $Z = z_{i,j}$ значений в узлах сетки.

Для формирования прямоугольной сетки в Octave есть функция *meshgrid*. Рассмотрим построение трёхмерного графика на следующем примере.

Задание 17. Построить график функции.

Вариант 0. $z(x,y) = 3x^2 - 2\sin^2 y, x \in [-2,2], y \in [-3,3]$
Решение задачи в Octave. Для построения графика функции $z(x,y) = 3x^2 - 2\sin^2 y, x \in [-2,2], y \in [-3,3]$ сформируем сетку. Для этого воспользуемся функцией *meshgrid*. Здесь же вычислим значение функции во всех узловых точках. Результат формирования сетки и вычисления значения функции мы можем увидеть в командном окне.

$$[xy] = meshgrid(-2:2,-3,3);\%\text{формирование сетки}$$
$$z = 3 * x. * x - 2\sin(y).^2$$

```
x =ug>

   -2   -1    0    1    2
   -2   -1    0    1    2
   -2   -1    0    1    2
   -2   -1    0    1    2
   -2   -1    0    1    2
   -2   -1    0    1    2
   -2   -1    0    1    2

y =

   -3   -3   -3   -3   -3
   -2   -2   -2   -2   -2
   -1   -1   -1   -1   -1
    0    0    0    0    0
    1    1    1    1    1
    2    2    2    2    2
    3    3    3    3    3

z =

    11.96017    2.96017   -0.03983    2.96017   11.96017
    10.34636    1.34636   -1.65364    1.34636   10.34636
    10.58385    1.58385   -1.41615    1.58385   10.58385
    12.00000    3.00000    0.00000    3.00000   12.00000
    10.58385    1.58385   -1.41615    1.58385   10.58385
    10.34636    1.34636   -1.65364    1.34636   10.34636
    11.96017    2.96017   -0.03983    2.96017   11.96017

debug>
```

Рисунок 8. Вычисление значения функции во всех узловых точках

Для построения каркасного графика следует обратиться к функции $mesh : mesh(x, y, z)$;

$$
\begin{aligned}
&[xy] = meshgrid(-2:2, -3, 3);\\
&z = 3 * x. * x - 2\sin(y).^2\\
&mesh : mesh(x, y, z);
\end{aligned}
$$

В результате получаем трёхмерный график, который представлен ниже

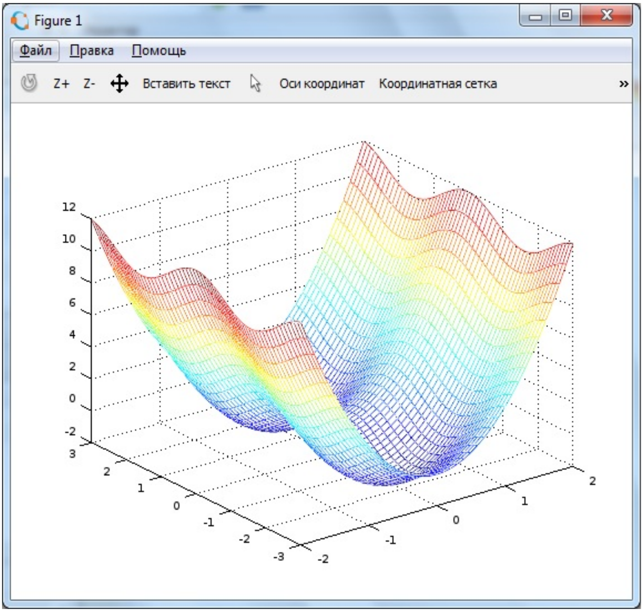

Рисунок 9. График функции $z(x, y) = 3x^2 - 2\sin^2 y$

Любой трёхмерный график можно вращать, используя мышку.

Для построения поверхностей, кроме функции $mesh$ построения каркасного графика, есть функция $surf$, которая строит каркасную поверхность, заливая её каждую клетку цветом, который зависит от значения функции в узлах сетки.

Вариант 1. $z = xy$

Вариант 2. $z = \sin(x)$;

Вариант 3. $z = \sin(x)\cos(y)$;

Вариант 4. $z = x^2 + y^2 + 2$;

Вариант 5. $z = x^2 + y^2$;
Вариант 6. $z = 5y^2 - x^2$;
Вариант 7. $z = (3x^2 + 4y^2) - 1$;
Вариант 8. $z = \cos(x)$;
Вариант 9. $z = x^2 + y^2 + 5$;
Вариант 10. $x = 5y^2 - x^2$;

Литература

1. AlfAlfio Quarteroni, Fausto Saleri, Paola Gervasio. Scientific Computing with MATLAB and Octave (Texts in Computational Science and Engineering). Springer, 2014, – 442 p.

2. G Dr Shoichiro Nakamura. GNU OCTAVE Primer for Begginers: EZ Guide to the Commands and Graphics. Paperback, 2015, – 82 p.

3. Jesper Jesper Schmidt Hansen.GNU Octave Beginner's Guide. Packt Publishing, 2011, – 280 p.

4. John John W. Eaton,David Bateman,Soren Hauberg. GNU Octave. Boston, 2009, – 604 p.

5. Akulich Акулич И.Л. Математическое программирование в примерах и задачах.– М.: Высшая школа, 1986, – 319с.

6. Alekseev Алексеев Е. Р., Чеснокова О. В. Введение в Octave для инженеров и математиков.–М.:Альт Линукс,2012, – 368с.

7. Alekseev Кормилицина Т.В. Лабораторный практикум по информационным технологиям в математике.–С.: МГПИ им М.Е.Евсевьева,2009, – 46с.

8. Kost Кострикин А.И. Ведение в алгебру.Часть 1.Основы алгебры. – М.: Физ мат лит., 2001, – 271с.

9. freesoft Интернет-ресурс http://grafikus.ru/examples/parametric-functions-2d

10. freesoft Интернет-ресурс http://grafikus.ru/examples/polar-functions

Оглавление

Введение 4
1. Первоначальные сведения о работе с пакетом Octave . . . 6
1.1 Начало работы с Octave 6
1.2 Установка программы для ОС Windows 8
1.3 Запуск программы Octave-4.0.1 9
1.4 Интерфейс пользователя 11
1.5 Помощь и документация 13
1.6 Выход из системы 13
1.7 Начальные навыки работы 14
1.8 Сообщения об ошибках 15
2. Задачи линейной алгебры 17
2.1 Ввод и формирование векторов и матриц 17
2.2 Функции для работы с матрицами и векторами . . . 21
2.3 Функции для работы с матрицами 23
2.4 Функции, реализующие численные алгоритмы решения задач линейной алгебры 29
2.5 Действия над матрицами 31
3. Индивидуальные задания по теме: Линейная алгебра 34
3.1 Определитель матрицы 34
3.2 Матрицы и действия над ними 37
3.3 Обратная матрица. Критерий обратимости матрицы 41
3.4 Основные понятия и метод Гаусса 47
3.5 Правило Крамера 53
3.6 Решение систем линейных уравнений в матричной форме 57

4. Построение двумерных графиков 59
4.1 Построение графиков в прямоугольной системе координат 61
4.2 Построение графиков в полярной системе координат 65
4.3 Построение графиков, заданных параметрически . . 67
4.4 Построение гистограммы 69
4.5 Анимация 70
5. Построение трёхмерных графиков 73
5.1 Построение графиков поверхностей 73
Литература 77