***Министерство Образования, культуры и исследований Республики Молдова***

***Технический Университет Молдовы***

***Департамент Физики***

**Отчёт**

По лабораторной работе № 1

Тема: ”Элементы MATLAB. Графики.”

Вариант 12

Выполнил студент гр. TI-202 F/R Дронов Дмитрий

Проверил Балмуш И.

Кишинёв 2020

**Цель работы**: изучить инструменты языка программирования MATLAB. Изучить основные инструкции MATLAB. Научиться проводить вычисления. Научиться строить графики функций.

**Ход работы**

**I. Опишите основные команды программы MATLAB в режиме командной строки**

Элементарные математические функции.

*Тригонометрические функции.*  
**sin** - Синус.  
**sind** - Синус аргумента в градусах.  
**sinh** - Гиперболический синус.  
**asin** - Арксинус.  
**asind** - Арксинус, результат в градусах.  
**asinh** - Гиперболический арксинус.  
**cos** - Косинус.  
**cosd** - Косинус аргумента в градусах.  
**cosh** – Гиперболический косинус.  
**acos** - Арккосинус.  
**acosd** - Арккосинус, результат в градусах.  
**acosh** - Гиперболический арккосинус.  
**tan** - Тангенс.  
**tand** - Тангенс аргумента в градусах.  
**tanh** - Гиперболический тангенс.  
**atan** - Арктангенс.  
**atand** - Арктангенс, результат в градусах.  
**atan2** - Арктангенс для четырех квадрантов.  
**atan2d** - Арктангенс для четырех квадрантов, результат в градусах.  
**atanh** - Гиперболический арктангенс.  
**sec** - Секанс.  
**secd** - Секанс аргумента в градусах.  
**sech** - Гиперболический секанс.  
**asec** - Арксеканс.  
**asecd** - Арксеканс, результат в градусах.  
**asech** - Гиперболический арксеканс.  
**csc** - Косеканс.  
**cscd** - Косеканс аргумента в градусах.  
**csch** - Гиперболический косеканс.  
**acsc** - Арккосеканс.  
**acscd** - Арккосеканс, результат в градусах.  
**acsch** - Обратный гиперболический косеканс.  
**cot** - Котангенс.  
**cotd** - Котангенс аргумента в градусах.  
**coth** - Гиперболический котангенс.  
**acot** - Арккотангенс.  
**acotd** - Арккотангенс, результат в градусах.  
**acoth** - Гиперболический арккотангенс.  
**hypot** - Корень квадратный из суммы квадратов.  
**deg2rad** - Конвертировать углы из градусов в радианы.  
**rad2deg** - Конвертирует углы из радианов в градусы.  
  
*Экспоненциальные функции.*  
**exp** - Экспонента.  
**expm1** - Точное вычисление exp (x) -1.  
log - Натуральный логарифм.  
**log1p** - Точное вычисление log (1 + x).  
**log10** - Общий (с основанием 10) логарифм.  
**log2** - Логарифм с основанием 2 и число с плавающей запятой.  
**pow2** – Возведение в степень 2 и число с плавающей запятой.  
**realpow** – Возведение в степень, которая выдаст ошибку при комплексном результате.  
**reallog** - Натуральный логарифм действительного числа.  
**realsqrt** - Квадратный корень из числа, большего или равного нулю.  
**sqrt** - Квадратный корень.  
**nthroot** - Действительный корень n-й степени действительных чисел.  
**nextpow2** - Ближайшая степень по основанию 2

*Комплексные функции*.  
**abs** - Абсолютное значение(Модуль).  
**angle** – Аргумент комплексного числа.  
**complex** - Построение комплексного числа из реальных и мнимых частей.  
**conj** -Сопряженное комплексному числу.  
**imag** - Мнимая часть.  
**real** - Действительная часть.  
**unwrap** - Развернуть фазовый угол.  
**isreal** - Истинно для реального массива.  
  
*Округление и остаток.*  
**fix** - Округлить до нуля.  
**floor** - Круглый в сторону минус бесконечности.  
**ceil** - округление в сторону плюс бесконечности.  
**round** - округление до ближайшего целого числа.  
**mod** - Модуль (остаток со знаком после деления).  
**rem** - Остаток после разделения.  
**sign**- Вычисление знака функции.

*Краткая справка:*

1. Если в конце строки ставить ; (точка запятая) , то результат выполнения команды не выводится на экран.
2. Если в начале строки ставить %(знак процент) , то дальше в этой строке следует комментарий.
3. Если в начале строки набрать команду *clc* и нажать *enter* , то происходит очищение экрана.
4. Если не указана переменная которой присваивается результат некоторых вычислений, то он присваивается встроенной переменной *ans.*
5. Если набрать команду *whos* и нажать *enter,* тона экранпоявляется список всех переменных рабочего пространства (*Workspace*).
6. Команда *clear* очищает все ячейки памяти всех переменных*;* команда *clear x –* только переменной *х;* команда *clear a, b, c* –переменных *a, b, c.*
7. В *MATLAB* все функции пишутся малыми буквами а аргументы заключаются в круглых скобках. В сложных выражениях используются только круглые скобки. Ниже приведены наиболее часто встречаемые функции:

|  |  |
| --- | --- |
| В математике | *MATLAB* |
| *sinx, cosx, tgx, ctgx* | *sin(x),cos(x), tan(x),cot(x)* |
| *arcsinx,arccosx,arctgx,arcctgx* | *asin(x),acos(x),atan(x),acot(x)* |
| *shx,chx,thx cthx* | *sinh(x),cosh(x),tanh(x),coth(x)* |
| *ex,lnx,logx* | *exp(x),log(x),log10(x)* |
|  | *sqrt(x),abs(x),a^(x), x^(1/3)* |
|  | *pi* |
|  | *i или j, Inf* |

Таблица 1.1

1. Полный список элементарных функции можно получить набирая в командную строку *help elfun.* Для получения полной информации о синтаксисе некоторой функции следует набирать в командную строку *help* и нажать *enter.*
2. Команда *x=[a:h: b]* генерирует векторстроку с равноудалёнными элементами от *а* до *b* с шагом *,* где *n* количество точек на участке [*a;b*].
3. Команда *x= [a b c d …e]* вводит вектор строку с элементами *a, b ,c ,d, …,e.*
4. Команда *x= [a; b;c;d;…;e]* вводит вектор столбец с элементами *a ,b ,c ,d, …,e.*
5. Команда *x=[a b;c d]* вводит матрицу .
6. В MATLABможно выполнять матричные вычисления по правилам математики.
7. В MATLABможно выполнять поэлементные вычисления с элементами матриц (векторов). Для этого необходимо писать операции умножения (. \*), деления (. /) и возведения в степень (. ^) с точкой спереди.

Для вывода данных в той или иной форме используются форматы: *format short; format long; format short e; format long e; format short g; format bank; format rat;* и другие, например (*format short; x),* выводит переменную *x* с четырмя цифрами после запятой.

**II. Вычислить заданные выражения при x *= -1,75·10* и *y = 3,1·.* Представить результаты в разных форматах. Изучить информацию о переменных рабочего пространства используя команду whos.**

|  |  |
| --- | --- |
| 12 |  |

Таблица 2.1

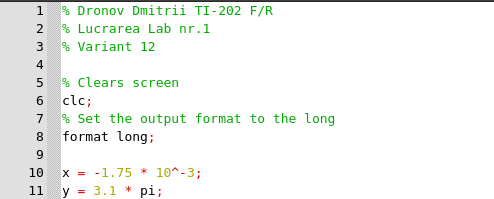
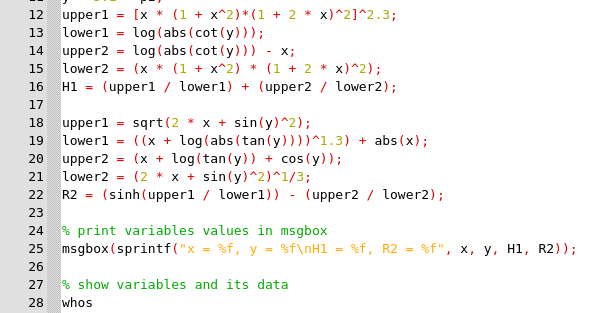


Рисунок 2.1

Сперва скрипт очищает экран командой clc. Далее выполняется 8 строка, устанавливая формат вывода long. После этого на строках 10 и 11 мы задаем значения x и y.

Рисунок 2.2

Упрощаем чтение формул, декларированием переменных upper1, lower1 и upper2, lower2. Задавая им значения, вычисляем H1 и R2.

Выводим значения переменных x, y, H1, R2 в message box.

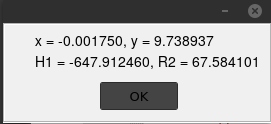


Рисунок 2.3

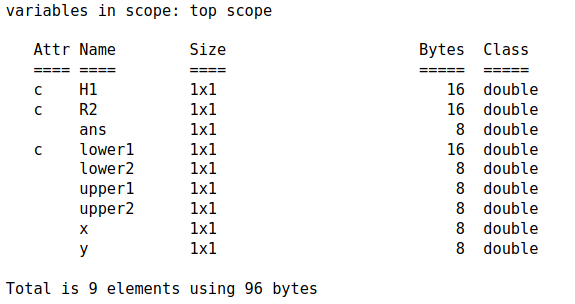


Рисунок 2.4

Командой whos выводим в консоль переменные и информацию о них.

**III. Опишите основные команды программы MATLAB для построения графиков.**

**График в линейном масштабе, функция plot**

*Синтаксис:*

            plot(y)  
            plot(x, y)  
            plot(x, y, s)  
            plot(x1, y1, s1, x2, y2, s2, ...)

*Описание:*

Команда plot(y) строит график элементов одномерного массива y в зависимости от номера элемента; если элементы массива y комплексные, то строится график plot(real(y), imag(y)). Если Y - двумерный действительный массив, то строятся графики для столбцов; в случае комплексных элементов их мнимые части игнорируются.

Команда plot (x, y) соответствует построению обычной функции, когда одномерный массив x соответствует значениям аргумента, а одномерный массив y - значениям функции. Когда один из массивов X или Y либо оба двумерные, реализуются следующие построения:

* если массив Y двумерный, а массив x одномерный, то строятся графики для столбцов массива Y в зависимости от элементов вектора x;
* если двумерным является массив X, а массив y одномерный, то строятся графики столбцов массива X в зависимости от элементов вектора y;
* если оба массива X и Y двумерные, то строятся зависимости столбцов массива Y от столбцов массива X.

Команда plot(x, y, s) позволяет выделить график функции, указав способ отображения линии, способ отображения точек, цвет линий и точек с помощью строковой переменной s, которая может включать до трех символов из следующей таблицы:

|  |  |  |
| --- | --- | --- |
| *Тип линии* | *Тип точки* | *Цвет* |
| |  |  | | --- | --- | | Непрерывная | - | | Штриховая | -- | | Двойной пунктир | : | | Штрих-пунктирная | -. | | |  |  | | --- | --- | | Точка | . | | Плюс | + | | Звездочка | \* | | Кружок | o | | Крестик | х | | |  |  | | --- | --- | | Желтый | y | | Фиолетовый | m | | Голубой | c | | Красный | r | | Зеленый | g | | Синий | b | | Белый | w | | Черный | k | |

Таблица 3.1

Если цвет линии не указан, он выбирается по умолчанию из шести первых цветов, с желтого до синего, повторяясь циклически.

Команда plot (x1, y1, s1, x2, y2, s2, ...) позволяет объединить на одном графике несколько функций y1(x1), y2(x2), ..., определив для каждой из них свой способ отображения.

Обращение к командам plot вида plot(x, y, s1, x, y, s2) позволяет для графика y(x) определить дополнительные свойства, для указания которых применения одной строковой переменной s1 недостаточно, например при задании разных цветов для линии и для точек на ней.

**Примеры:**

Построим график функции y = sin(x) на отрезке [-p p ] с шагом p /500:

           x = -pi:pi/500:pi;  
           y = sin(x);  
           plot(y) % рис. а  
           plot(x, y) % рис. б

График на рис. а отображает значения одномерного массива y, состоящего из 1001 элемента, как функцию от номера элемента; график на рис. б отображает значения того же массива как функцию элементов массива x.

|  |  |
| --- | --- |
| а) |  |
| б) |  |

Рисунок 3.1

**subplot - Разбиение графического окна**

Синтаксис:

subplot(m, n, p)  
subplot(h)  
subplot(mnp)

Описание:

Данная команда выполняется перед обращением к функциям построения графиков для одновременной выдачи нескольких графиков в различных частях графического окна.

Команды subplot(mnp) или subplot(m, n, p), где mnp - 3 цифры, производит разбивку графического окна на несколько подокон, создавая при этом новые объекты axes; значение m указывает, на сколько частей разбивается окно по горизонтали, n - по вертикали, а p - номер подокна, куда будет выводиться очередной график. Эти же команды могут использоваться для перехода от одного подокна к другому.

Команда subplot(h), где h - дескриптор для объекта axes соответствующего подокна, - другой способ выбора подокна для размещения графика.

Команды clf, subplot(111), subplot(1, 1, 1) выполняют одну и ту же функцию - удаляют все подокна и возвращают графическое окно в штатное состояние.

**mesh, meshc, meshz - Трехмерная сетчатая поверхность**

Синтаксис:

|  |  |  |
| --- | --- | --- |
| mesh(X, Y, Z, C) | meshc(X, Y, Z, C) | meshz(X, Y, Z, C) |
| mesh(x, y, Z, C ) | meshc(x, y, Z, C ) | meshz(x, y, Z, C ) |
| mesh(Z, C) | meshc(Z, C) | meshz(Z, C) |
| mesh(X, Y, Z) | meshc(X, Y, Z) | meshz(X, Y, Z) |
| mesh(x, y, Z) | meshc (x, y, Z) | meshz(x, y, Z) |
| mesh(Z) | meshc(Z) | meshz(Z) |

Таблица 3.2

Описание:

Команда mesh(X, Y, Z, C) выводит на экран сетчатую поверхность для значений массива Z, определенных на множестве значений массивов X и Y. Цвета узлов поверхности задаются массивом C. Цвета ребер определяются свойством EdgeColor объекта surface. Можно задать одинаковый цвет для всех ребер, определив его в виде вектора [r g b] интенсивности трех цветов - красного, зеленого, синего. Если определить спецификацию none, то ребра не будут прорисовываться. Если определить спецификацию flat, то цвет ребер ячейки определяется цветом того узла, который был первым при обходе этой ячейки. Поскольку одни и те же ребра обходятся несколько раз, то цвета будут замещаться. Если определить спецификацию interp, то будет реализована линейная интерполяция цвета между вершинами ребра.

Применение функции shading после обращения к функции mesh изменяет спецификации свойств EdgeColor и FaceColor согласно следующей таблице.

|  |  |  |  |
| --- | --- | --- | --- |
| Свойство | Применяемая функция | | |
| mesh | shading flat | shading interp |
| EdgeColor | flat | flat | interp |
| FaceColor | Цвет фона | Цвет фона | Цвет фона |

Таблица 3.3

Команда mesh(x, y, Z, C) выполняет ту же функцию, но вместо двумерных массивов X, Y использует их одномерные проекции, так что если length(x) = n, а length(y) = m, то [m, n] = size(Z). В этом случае узлы сетчатой поверхности определяются тройками {x(j), y(i), Z(i, j)}, где вектор x определяет столбцы массива Z, а y - строки.

Команда mesh(Z, C) использует сетку, которая определяется одномерными массивами x = 1 : n и y = 1 : m.

Команды mesh(X, Y, Z), mesh(x, y, Z), mesh(Z) используют в качестве массива цвета C = Z, то есть цвет в этом случае пропорционален высоте поверхности.

Группа команд meshc(...) в дополнение к трехмерным поверхностям строит проекцию линий постоянного уровня.

Группа команд meshz(...) в дополнение к трехмерным поверхностям строит плоскость отсчета на нулевом уровне, закрывая поверхность, лежащую ниже этого уровня.

Функция h = mesh(...) возвращает дескриптор h для графического объекта surface.

**surf, surfc - Затененная сетчатая поверхность**

Синтаксис:

|  |  |  |
| --- | --- | --- |
|  | surf(X, Y, Z, C) | surfc(X, Y, Z, C) |
|  | surf(x, y, Z, C ) | surfc(x, y, Z, C ) |
|  | surf(Z, C) | surfc(Z, C) |
|  | surf(X, Y, Z) | surfc(X, Y, Z) |
|  | surf(x, y, Z) | surfc (x, y, Z) |
|  | surf(Z) | surfc(Z) |

Таблица 3.4

Описание:

Команда surf(X, Y, Z, C) выводит на экран сетчатую поверхность для значений массива Z, определенных на множестве значений массивов X и Y. Цвет ячейки определяется массивом C. Цвет ребер - черный, определяется свойством EdgeColor, специфицированным как [0 0 0]. Можно задать одинаковый цвет для всех ребер, определив его в виде вектора [r g b] интенсивности трех цветов - красного, зеленого, синего. Если определить спецификацию none, то ребра не будут прорисовываться.

Применение функции shading после обращения к функции surf изменяет спецификации свойств EdgeColor и FaceColor графического объекта surface согласно следующей таблице.

|  |  |  |  |
| --- | --- | --- | --- |
| Свойство | Применяемая функция | | |
| surf | shading flat | shading interp |
| EdgeColor | [0 0 0] | none | none |
| FaceColor | flat | flat | interp |

Таблица 3.5

Команда surf(x, y, Z, C) выполняет ту же функцию, но вместо двумерных массивов X, Y использует их одномерные проекции, так что если length(x) = = n, а length(y) = m, то [m, n] = size(Z). В этом случае узлы сетчатой поверхности определяются тройками {x(j), y(i), Z(i, j)}, где вектор x определяет столбцы массива Z, а y - строки.

Команда surf(Z, C) использует сетку, которая определяется одномерными массивами x = 1 : n и y = 1 : m.

Команды surf(X, Y, Z), surf(x, y, Z), surf(Z) используют в качестве массива цвета C = Z, то есть цвет в этом случае пропорционален высоте поверхности.

Группа команд surfc(...) в дополнение к трехмерным затененным поверхностям строит проекцию линий постоянного уровня.

Функция h = surf(...) возвращает дескриптор h для графического объекта surface.

**contour - Изображение линий уровня для трехмерной поверхности**

Синтаксис:

|  |  |  |
| --- | --- | --- |
|  | contour(Z) | contour(x, y, Z) |
|  | contour(Z, n) | contour(x, y, Z, n) |
|  | contour(Z, v) | contour(x, y, Z, v) |
|  | contour(...,’тип\_линии’) |  |
|  | C = contour(...) |  |
|  | [C, h] = contour(...) |  |

Таблица 3.6

Описание:

Команда contour(Z) рисует двумерные линии уровня для массива данных Z, определяющих поверхность в трехмерном пространстве без учета диапазона изменения координат x и y.

Команда contour(x, y, Z), где x и y - векторы, рисует линии уровня для массива данных Z с учетом диапазона изменения координат x и y.

Команды contour(Z, n), contour(x, y, Z, n) рисует n линий уровня для массива данных Z; по умолчанию, n равно 10.

Команды contour(Z, v), contour(x, y, Z, v) рисуют линии уровня для заданных значений, которые указаны в векторе v.

Команда contour(...,’<тип линии>’) рисует линии уровня, тип и цвет которых определяются параметром <тип линии> команды plot.

Функция C = contour(...) возвращает массив C описания линий уровней по аналогии с функцией contourc для последующего использования командой clabel.

Функция [C, h] = contour(...) возвращает массив C и вектор-столбец дескрипторов h графических объектов line для каждой линии уровня.

**contour3 - Изображение трехмерных линий уровня**

Синтаксис:

|  |  |  |
| --- | --- | --- |
|  | contour3(Z) | contour3(X, Y, Z) |
|  | contour3(Z, n) | contour3(X, Y, Z, n) |
|  | C = contour3(...) |  |
|  | [C, h] = contour3(...) |  |

Таблица 3.7

Описание:

Команда contour3(Z) рисует трехмерные линии уровня для массива данных Z, определяющих поверхность в трехмерном пространстве без учета диапазона изменения координат x и y.

Команда contour3(X, Y, Z), где X и Y - двумерные массивы, вычисленные с помощью функции meshgrid, рисует линии уровня для массива данных Z с учетом диапазона изменения координат x и y.

Команды contour3(Z, n), contour(X, Y, Z, n) рисуют n линий уровня для массива данных Z; по умолчанию n равно 10.

Функция C = contour3(...) возвращает массив C описания линий уровней по аналогии с функцией contourc для последующего использования командой clabel.

Функция [C, h] = contour3(...) возвращает массив C и вектор-столбец дескрипторов h графических объектов line для каждой линии уровня.

**contourf-** **Закрашенный двухмерный контурный график**

*Cинтаксис:*

contourf (Z)

contourf (Z, n)

contourf (Z, v)

contourf (X, Y, Z)

contourf (X, Y, Z, n)

contourf (X, Y, Z, v)

contourf (axes\_handle, ...)

[C, h, CF] = contourf (...)

*Описание:*

Контур с заливкой отображает изолинии, рассчитанные по матрице Z, и заполняет области между изолиниями постоянными цветами. Цвет залитых областей зависит от цветовой карты текущей фигуры.

contourf (Z) рисует контурный график матрицы Z, где Z интерпретируется как высота относительно плоскости. Z должна быть как минимум матрицей 2 на 2. Количество контурных линий и значения контурных линий выбираются автоматически.

contourf (Z, n) рисует контурный график матрицы Z с n уровнями контура.

contourf (Z, v) рисует контурный график матрицы Z с уровнями контуров при значениях, указанных в векторе v.

contourf (X, Y, Z), contourf (X, Y, Z, n) и contourf (X, Y, Z, v) создают контурные графики Z с использованием X и Y для определения пределов осей x и y . Когда X и Y являются матрицами, они должны быть того же размера, что и Z, и в этом случае они определяют поверхность, как это делает surf.

contourf (axes\_handle, ...) отображает оси с ручкой axes\_handle вместо текущих осей (gca).

[C, h, CF] = contourf (...) возвращает контурную матрицу C, вычисленную функцией contourc и используемую clabel, вектор ручек h для исправления графических объектов и контурную матрицу CF для заполненных областей.

**Оформление графиков.**

На построенный график нужно нанести информационные записи следующими командами:

**grid on** – включает сетку;

**xlabel**(‘запись’), **ylabel**(‘запись’)-наносит записи на декартовые оси координат;

**title**(‘запись’)-наносит поясняющую запись для названия построенного графика;

**legend(‘запись’)-наносит соответствующую математическую запись построенной функции.**

**IV. Построить графики функции. Вставить информационные записи. Использовать цвета, типы линий и маркеры.**

**Представить графики разными способами:**

**a) в разных окнах  
b) в одном окне с общими осями**

**c) используя команду subplot**

**c1) в одном окне с разными осями**

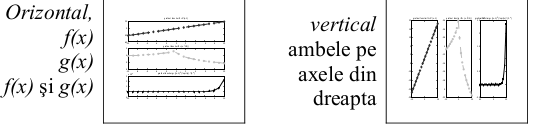
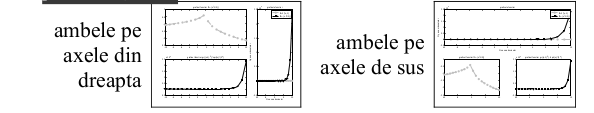


Рисунок 4.1

**c2) в одном окне – каждая на разных осях**

Рисунок 4.2

|  |  |
| --- | --- |
| 12 |  |

Таблица 4.1

a)

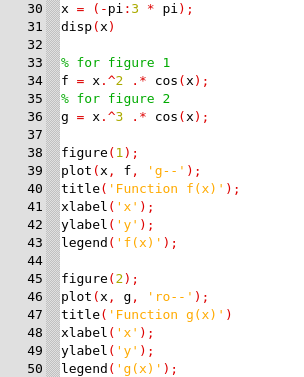


Рисунок 4.3

Первый шаг – декларирование переменной x и инициализация матрицы в этой переменной от -pi до 3 \* pi без шага. Выводим x на экран для удобства. Далее на строках 34 и 36 декларируем и инициализируем переменные f и g. Следующим этапом будет создание графиков функции f, g, f и g в разных окнах соответсвенно.

Для каждого графика обозначаем его индекс функцией figure, передавая как аргумент индекс графика. В нашем случае это будет 1, 2 и 3 соответсвенно. Функция plot строит график фунции в соответсвии с аргументами, которые мы передаем в данную функцию. Далее следует название графика и обозначение осей координат. В нашем случае последним этапом построения графика будет добавление легенды графика.

b) По похожему принципу строим третий график, код которого указан на рисунке ниже:

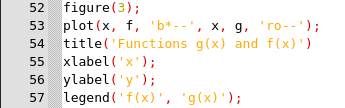


Рисунок 4.4

Три указаных выше графика функций будут выглядеть следующим образом:

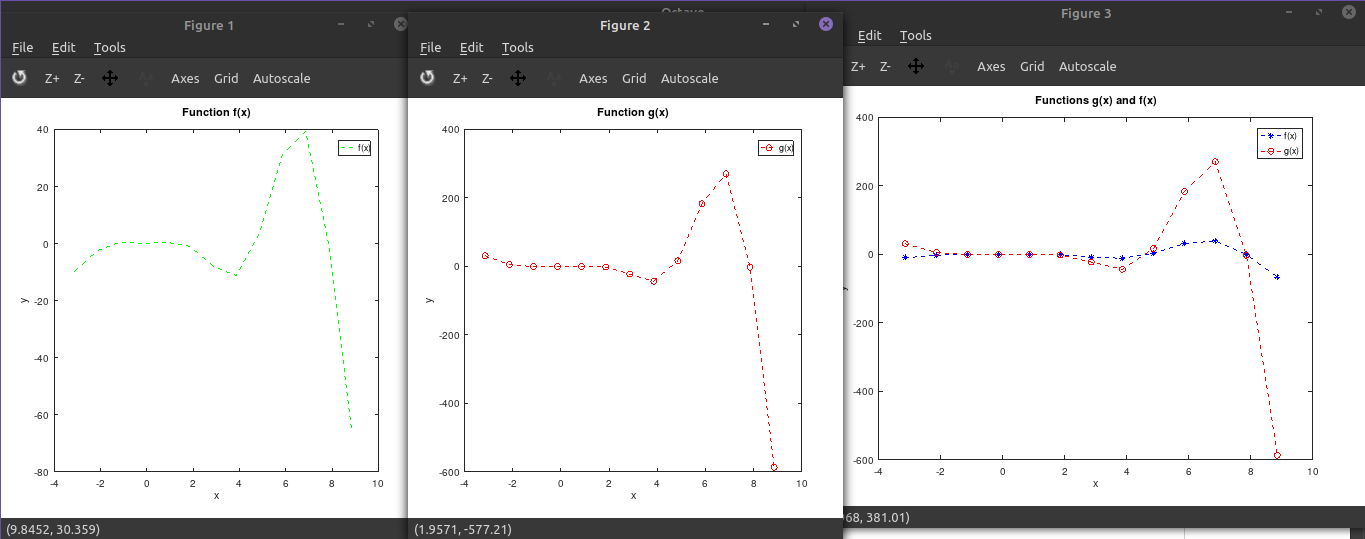


Рисунок 4.5

c)

с1)

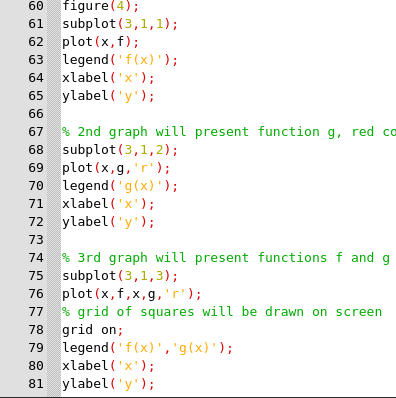


Рисунок 4.6

Обозначаем индекс графика как в предыдущем задании и далее используем функцию subplot для того, чтобы разбить графическое окно на несколько частей. Первый агрумент указывает на сколько горизонтальных частей мы хотим разбить окно. Второй аргумент указывает на количество вертикальных частей окна и третий аргумент указывает на номер графика. Далее рисуем графики как в предыдущем задании. Команда grid on отвечает за рисование сетки.

Результат данных команд будет выглядеть следующим образом:

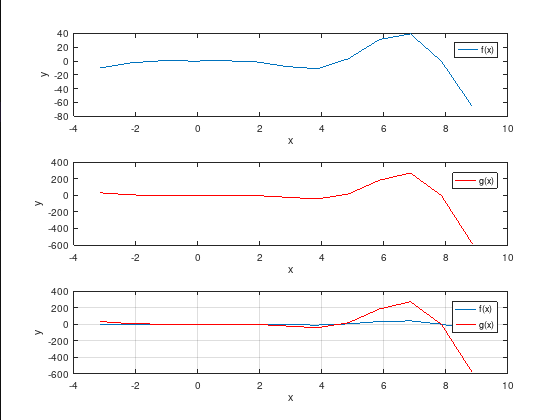


Рисунок 4.7

Для того, чтобы сделать графики вертикальными мы просто меняем аргументы фунции subplot.

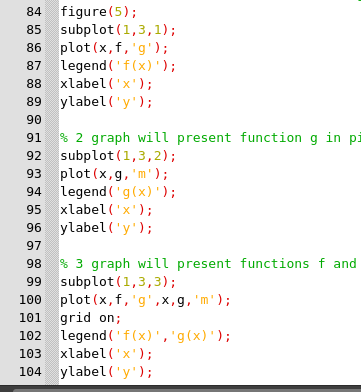


Рисунок 4.8

В данном случае мы будем менять только второй и третий аргумент. Вывод данных команд будет выглядеть следующим образом:

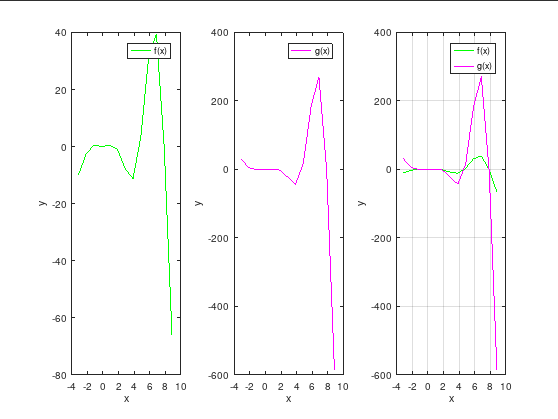


Рисунок 4.9

с2)

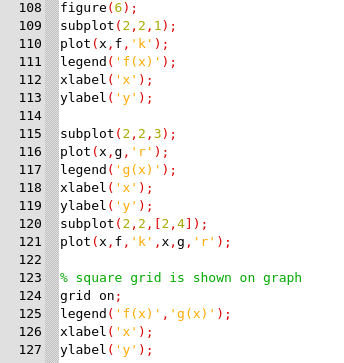


Рисунок 4.10

В данном случае нам необходимо поменять аргументы функции subplot. Указываем, что у нас 2 графа по горизонтали и 2 по вертикали. Выглядят они следующим образом:

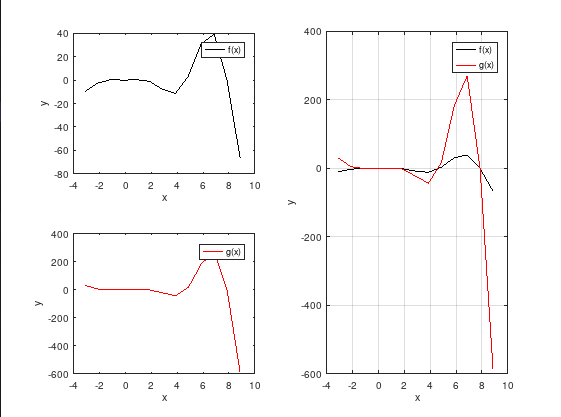


Рисунок 4.11

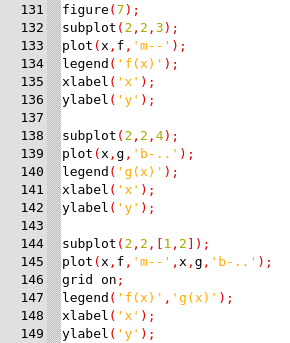


Рисунок 4.12

Указываем аргументы функции subplot по похожему принципу и получаем следующий результат:

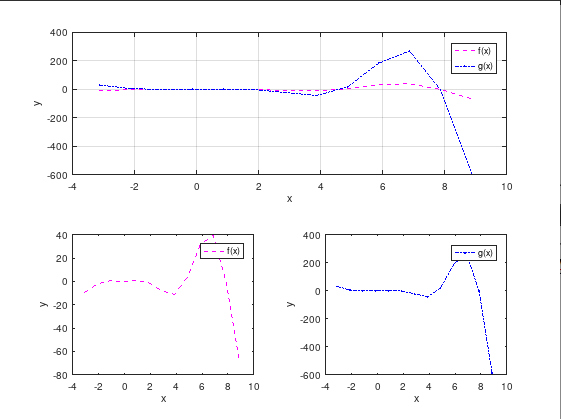


Рисунок 4.13

**Вывод:** В ходе данной лабораторной работы был изучен функционал языка программирования MATLAB, а так же были изучены основные функции MATLAB. Удалось провести вычислительные процессы с данными функциями и представить их в различных форматах, а так же вычислил значения функции на заданном интервале и на производных точках. Был изучен проценн создания графиков в разных форматах, например: в разных окнах, в одном окне; в одном окне, используя функцию subplot. Был понятен принцип использования данного языка программирования. Были изучены разные среды разработки такие как: MATLAB, GNU Octave, Octave Online.