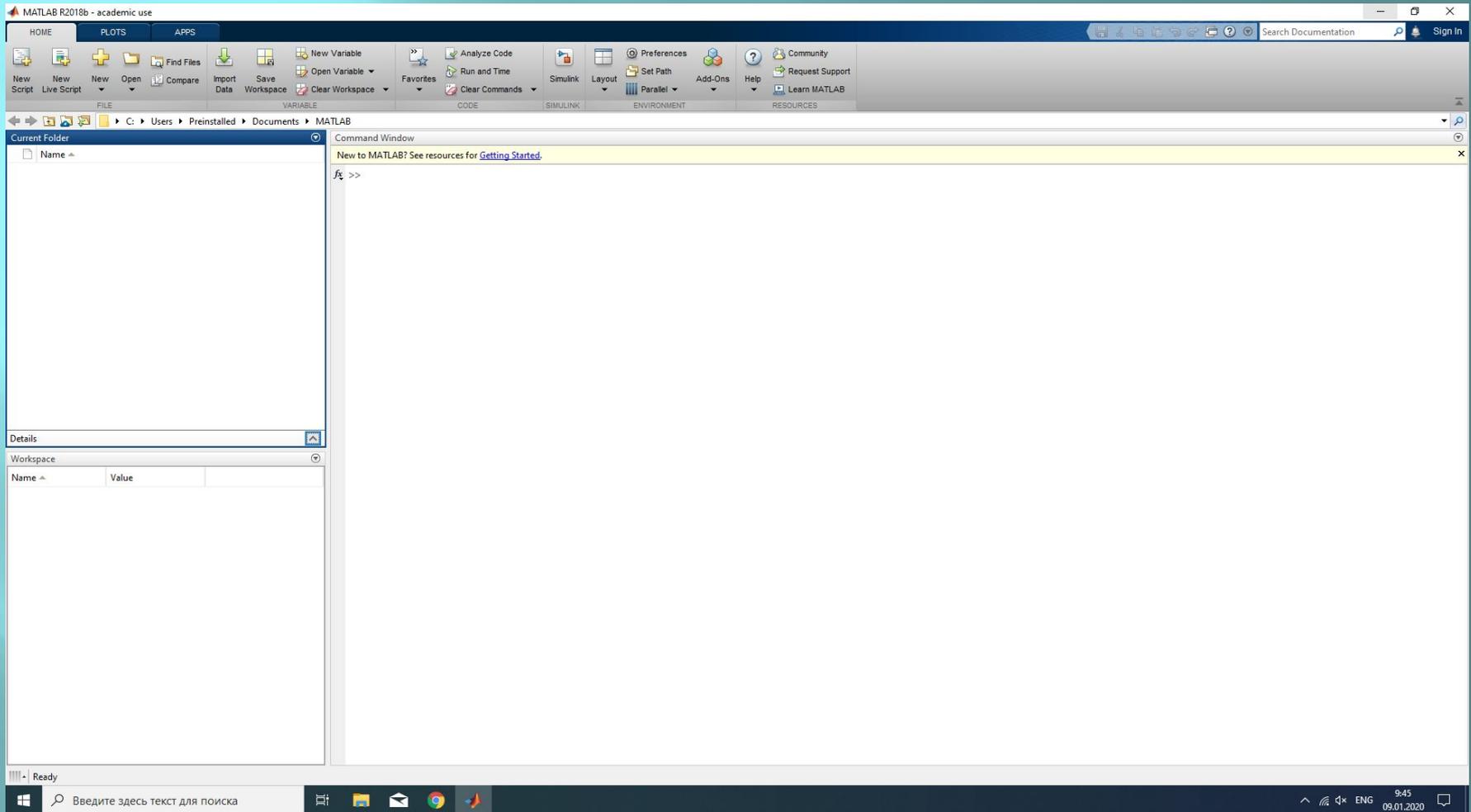
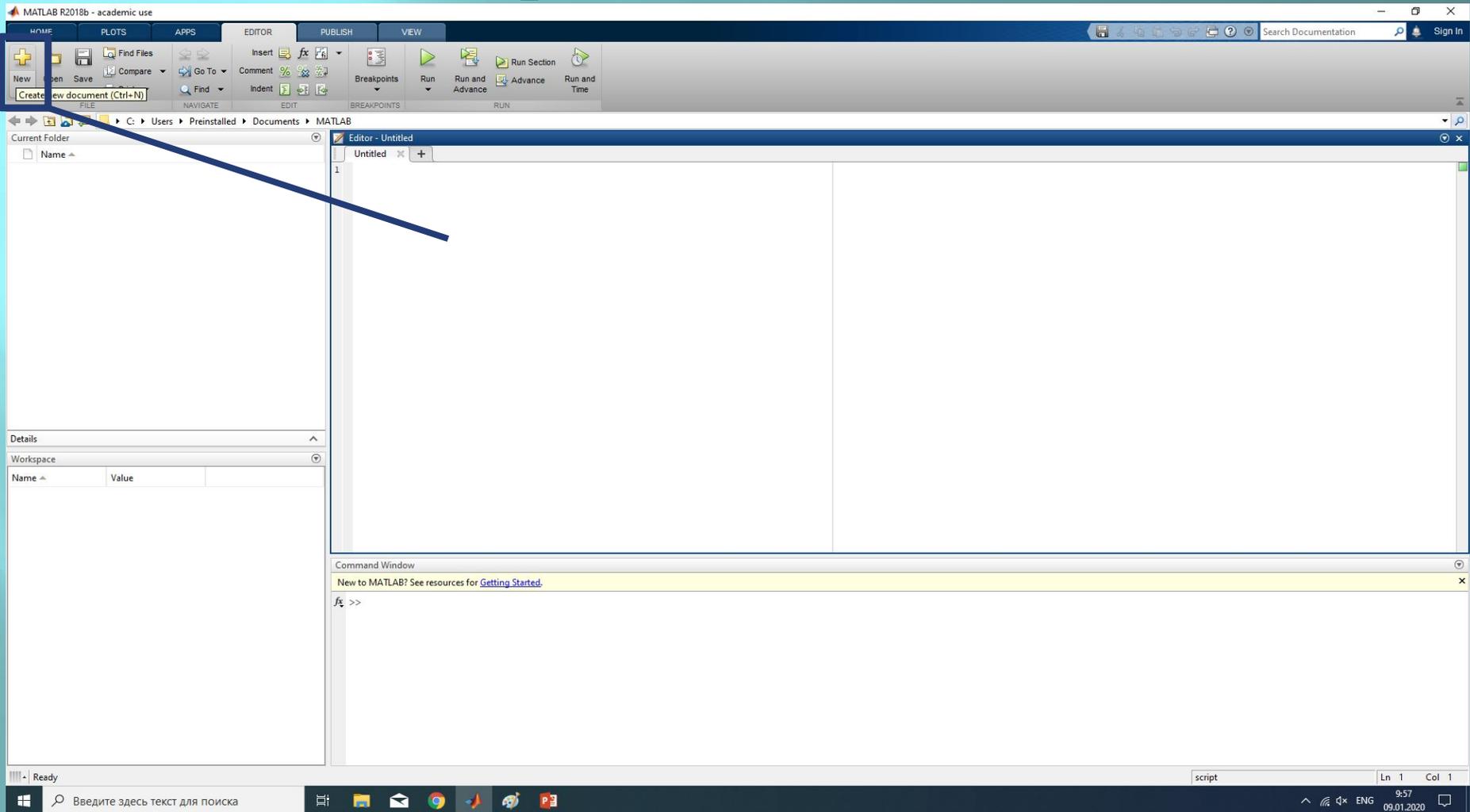


# Общий вид рабочего окна в пакете Matlab (начальный вид)



# Общий вид рабочего окна в пакете Matlab ( рабочий вид)



# Настройка путей

Лента– Set Path – выбрать свою рабочую папку

Имя папки  
должно содержать буквы  
**ТОЛЬКО**  
**английского алфавита!!!!**

Путь до рабочего стола в системе всегда записывается по-английски, вне зависимости от того, что видит пользователь.

# Типы компоновки команд

- Команды в командной строке – однократное выполнение, возможен повторный вызов из истории команд (>>)
- Сценарий – набор команд, записанных в файл, но не объединенных общим именем. Запуск на выполнение осуществляется из файла вручную. Возможно выполнение части сценария
- Функция – набор команд, записанных в файл, объединенных общим именем, может иметь входные/выходные параметры. Запуск на выполнение осуществляется по имени из любого места Matlab (командной строки, сценария, другой функции)

# Написание .m-файлов

Сценарий и функция записываются в файле с расширением «.m».

Это обычный текстовый файл, который может быть набран в любом текстовом редакторе (от блокнота до word'a), но сохранен должен быть как текст в формате txt!!!

Так же возможен набор из редактора M-файлов в самом пакете Matlab.

# Основы написания .m-файлов

Присвоение значения  
осуществляется знаком  
«=».

В редакторе для .m-файлов набрать:

$X=3.25*0.75$

Выделить запись и нажать правую кнопку мыши. Выбрать пункт **Evaluate Section** (первый в списке)

В командном окне появится запись:

$X =$

2.4375

# Основные типы переменных

Переменная обретает тип при присвоении!

Целочисленные: 1,2,3,-5,-8...

Действительные: 1.25, 3.15,-7.234....

Комплексные:  $1+2i$ ,  $3-4.75i$ ....

Вектора: (1 2 3 4)....

Матрицы: (1 2 3  
5 6 7  
15 24 13)

Строки: 'This is string'

# Комментарий - 1

При названии переменных необходимо избегать ситуаций, когда имя переменной совпадает с поименованной константой, например,  $\pi$ .

Для проверки можно использовать команду:  
*which придуманное имя переменной*

```
>> which x  
x is a variable.
```

```
>> which f  
'f' not found.
```

# Основные математические функции - 1

Функция в Matlab	Название функции
$+$ , $-$ , $*$ , $/$	Сложение, вычитание, умножение, деление
$\exp(x)$ , $\text{sqrt}(x)$ , $^{\wedge}$	Экспонента, корень квадратный, степень
$\log(x)$ , $\log_2(x)$ , $\log_{10}(x)$	Натуральный логарифм, логарифм по основанию 2 и 10
$\text{abs}(x)$ , $\text{sign}(x)$	Модуль, знак числа

# Основные математические функции - 2

Функция в Matlab	Название функции
$\sin(x)$ , $\cos(x)$ , $\tan(x)$ , $\cot(x)$ , $\sec(x)$ , $\csc(x)$ , $\pi$	Синус, косинус, тангенс, котангенс, секанс, cosecant, $\pi$
$\arcsin(x)$ , $\arccos(x)$ , $\arctan(x)$ , $\text{acot}(x)$ , $\text{asec}(x)$ , $\text{acsc}(x)$	Арксинус, арккосинус, арктангенс, арккотангенс, арксеканс, арккосеканс
$i$	Мнимая единица (вводится без знака умножения) $1+2i$
$\text{conj}(x)$ , $\text{imag}(x)$ , $\text{real}(x)$	Комплексно-сопряженное число, мнимая и действительная части комплексного числа

## Комментарий - 2

Функций для элементарных математических вычислений намного больше, чем приведено в таблицах. Посмотреть их все можно в Help.

При выборе имени для собственных функций желательно избегать ситуаций, когда имя функции совпадает с поименованной константой или с уже имеющейся функцией.

## Комментарий - 3

При выборе имени для переменной или функции недопустимо совпадение имени переменной или функции с ключевым словом Matlab.

Для проверки можно использовать команду: `iskeyword` ('придуманное имя переменной')

```
>> iskeyword('conj')
```

```
ans =
```

```
logical
```

```
0
```

```
>> iskeyword('if')
```

```
ans =
```

```
logical
```

```
1
```

результат **0**, имя **не является** ключевым словом

результат **1**, имя **является** ключевым словом

## Комментарий - 4

Чтобы проверить корректность выбора имени возможно использовать функцию:

`isvarname(s)`,

возвращающую 0 если имя недопустимо и 1, если выбранное имя допустимо в Matlab

```
>> isvarname('rows_2')
```

```
ans =
```

```
logical
```

```
1
```

```
>> isvarname('2_rows')
```

```
ans =
```

```
logical
```

```
0
```

результат **1**, сочетание символов **допустимо** для имени переменной/функции

результат **0**, сочетание символов **не допустимо** для имени переменной/функции

# Задача для вычисления

Пусть необходимо вычислить выражения

$$\sqrt{\frac{\sin\left(\frac{4}{3}\pi \cdot x\right) + e^{0.1y}}{\cos\left(\frac{4}{3}\pi \cdot x\right) + e^{0.1y}}} + \sqrt[3]{\frac{\sin\left(\frac{4}{3}\pi \cdot x\right) + e^{0.1y}}{\cos\left(\frac{4}{3}\pi \cdot x\right) + e^{0.1y}}}$$

при заданных значениях переменных величин:  $x=0,5$  и  $y=0,2$

# Пример вычисления 1

Ф	$x=0.5$ $y=0.2$ $c=\text{sqrt}((\sin(4*x*\text{pi}/3)+\exp(0.1*y))/(\cos(4*x*\text{pi}/3)+\exp(0.1*y)) + \dots$ $((\sin(4*x*\text{pi}/3)+\exp(0.1*y))/(\cos(4*x*\text{pi}/3)+\exp(0.1*y))^{1/3}$
КС	$x=$ 0.5 $y=$ 0.2 $c=$ 3.44

Продолжение формулы на следующей строке

# Пример вычисления 2

Ф	<pre>x=0.5; y=0.2; a=sin(4*x*pi/3)+exp(0.1*y); b=cos(4*x*pi/3)+exp(0.1*y); c=sqrt(a/b)+(a/b)^1/3</pre>
КС	<pre>c= 3.1128</pre>

«немая строка» - результат вычислений не выводится в командную строку

## Комментарий - 5

Посмотреть значение любой переменной можно с помощью функции:

`disp(имя переменной)`

```
>> disp(c)
    3.1128
```

В качестве результата можно получить как ожидаемое значение, так и ответ в виде:

NaN – не число (Not A Number), например при попытке деления на 0

Inf –  $\infty$  при превышении наибольшего положительного числа (1.798 e+308)

# Вектор-строки и вектор-столбцы

Все в Matlab – матрицы! Индексация начинается с 1 !!!

Matlab	Математика
$a=[3; -1; 7]$ – разделение элементов вектора-столбца идет через «;»	$a = \begin{pmatrix} 3 \\ -1 \\ 7 \end{pmatrix}$
$b=[3 \ -1 \ 7]$ – разделение элементов вектора-столбца идет через «пробел»	$b = (3 \ -1 \ 7)$

# Вектор значений аргумента

Запись :

```
x = -1:0.5:2;
```

создает вектор значений отличающихся друг от друга на постоянный шаг. Разделителем в данном случае является «:».

Первое значение – значение 1-го элемента вектора

Второе значение – шаг

Третье значение – значение последнего элемента вектора

Результат – вектор-строка

```
>>x
```

```
x=
```

```
-1 -0.5 0 0.5 1 1.5 2
```

# Особенности задания вектора значений аргумента

- Шаг можно опустить, если он равен 1 :  $x = -1:5;$

Результат

```
>>x
```

```
x=
```

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

- Шаг может быть отрицательный (следить, чтобы начальное значение было не меньше конечного!):

```
x = 4:-2:-10;
```

Результат

```
>>x
```

```
x=
```

```
4 2 0 -2 -4 -6 -8 -10
```

# Вектор значений функции

Запись :

```
f = sin(x);
```

Создает вектор значений по имеющемуся вектору  $x$ :

```
>>f
```

```
f=
```

```
-0.874 -0.479 0 0.479 0.874 0.997 0.909
```

# Ошибка при вычислении вектора значений функции

Если вычислять более сложные зависимости :

$$f = x * \sin(x) + x^2; \text{ ( запись для функции } f = x \cdot \sin(x) + x^2 \text{)}$$

то, в соответствии с правилами матричного умножения, возникает ситуация умножения *строки*  $x$  на *строку*  $\sin(x)$ , что недопустимо!

В таких случаях необходимо перед операциями  $+$ ,  $-$ ,  $*$ ,  $/$  ставить «.», которая означает покомпонентное выполнение операции.

# Корректный способ вычисления вектора значений функции

При использовании покомпонентного вычисления:

$$f = x.*\sin(x)+x.^2;$$

Создается вектор значений по имеющемуся вектору  $x$ :

```
>>f
```

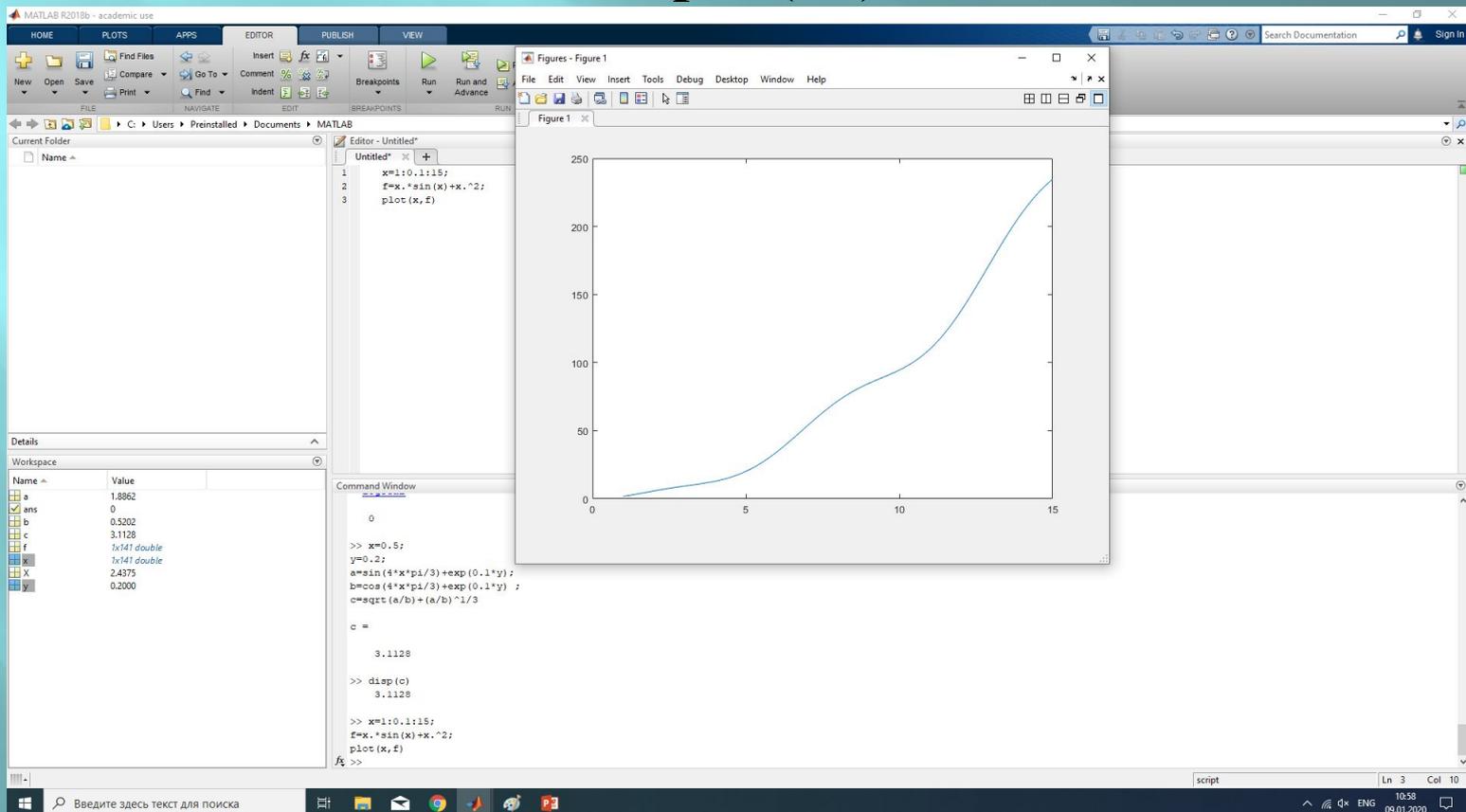
```
f=
```

```
1.841  0.49  0  0.49  1.849  3.746  5.819
```

# Графики

Для вывода графика в новом окне необходимо выполнить команду:

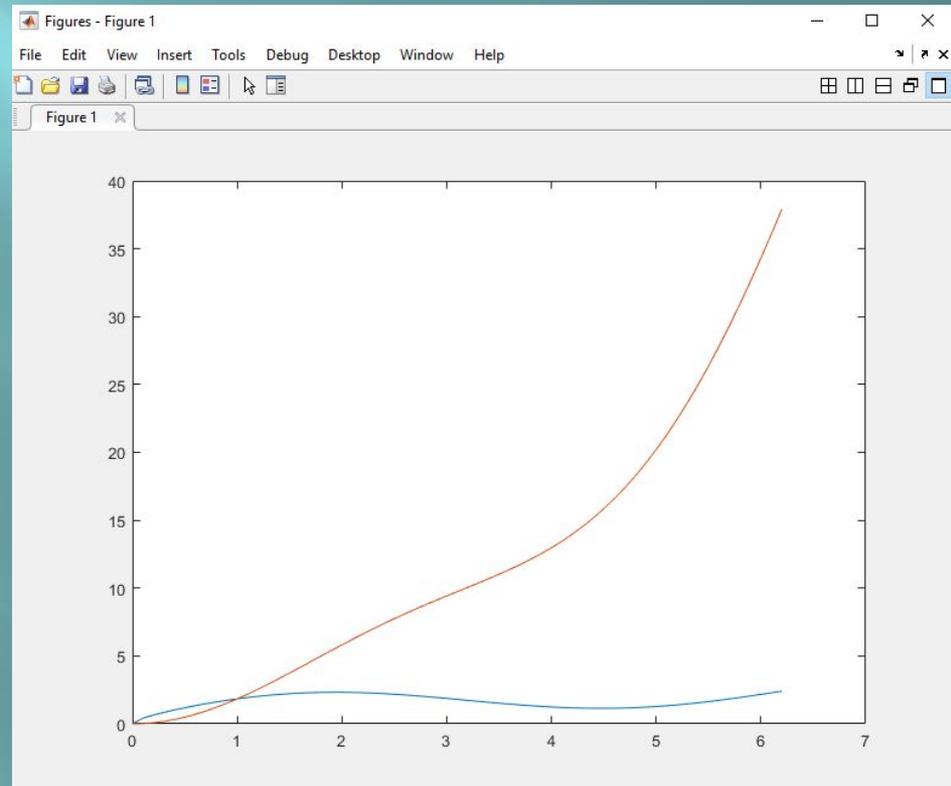
**plot(x,f)**



# Построение более одного графика

Если необходимо на одном графике вывести сразу несколько кривых, то пары (аргумент-функция) указываются через запятую.

```
x1 = 0:0.1:2*pi;  
x2 = 0:0.1:2*pi;  
  
f = sin(x1)+x1.^0.5;  
g = x2.*sin(x2)+x2.^2;  
  
plot(x1,f, x2,g)
```

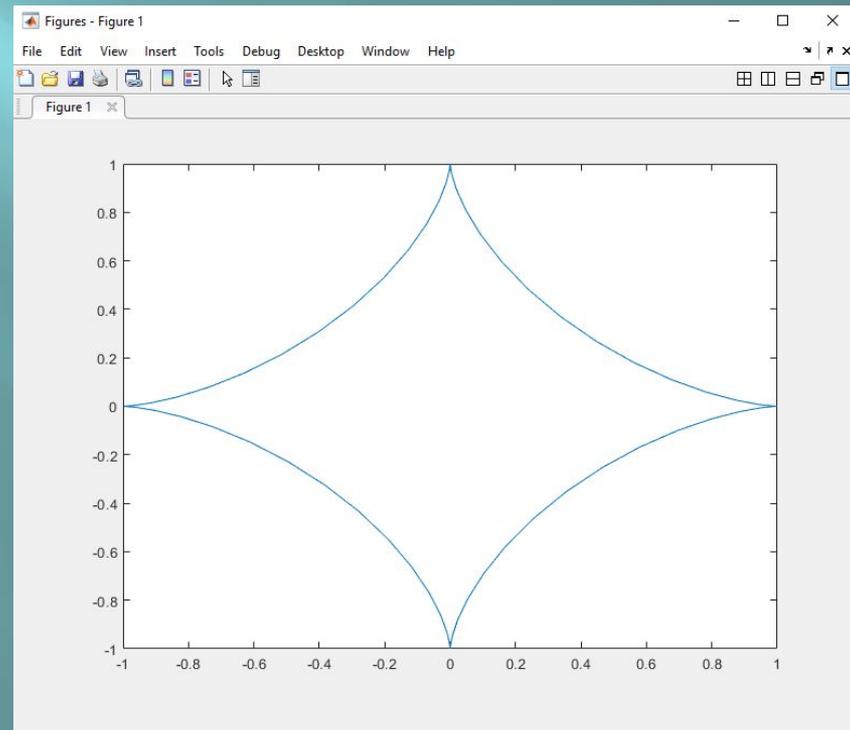


# Построение графиков функций заданных параметрически

Если функция задана параметрически, то строятся обе зависимости от параметра и уже они выводятся на график.

$$\begin{cases} x(t) = \cos^3(t) \\ y(t) = \sin^3(t) \end{cases}$$

```
t = 0:0.1:2*pi;  
x = cos(t).^3;  
y = sin(t).^3;  
plot(x,y)
```



# Задание типа линий для кривой

Тип линии, цвет, маркер задаются опционально в команде `plot` как строковый аргумент. Строковый аргумент – строка, заключенная в апострофы:

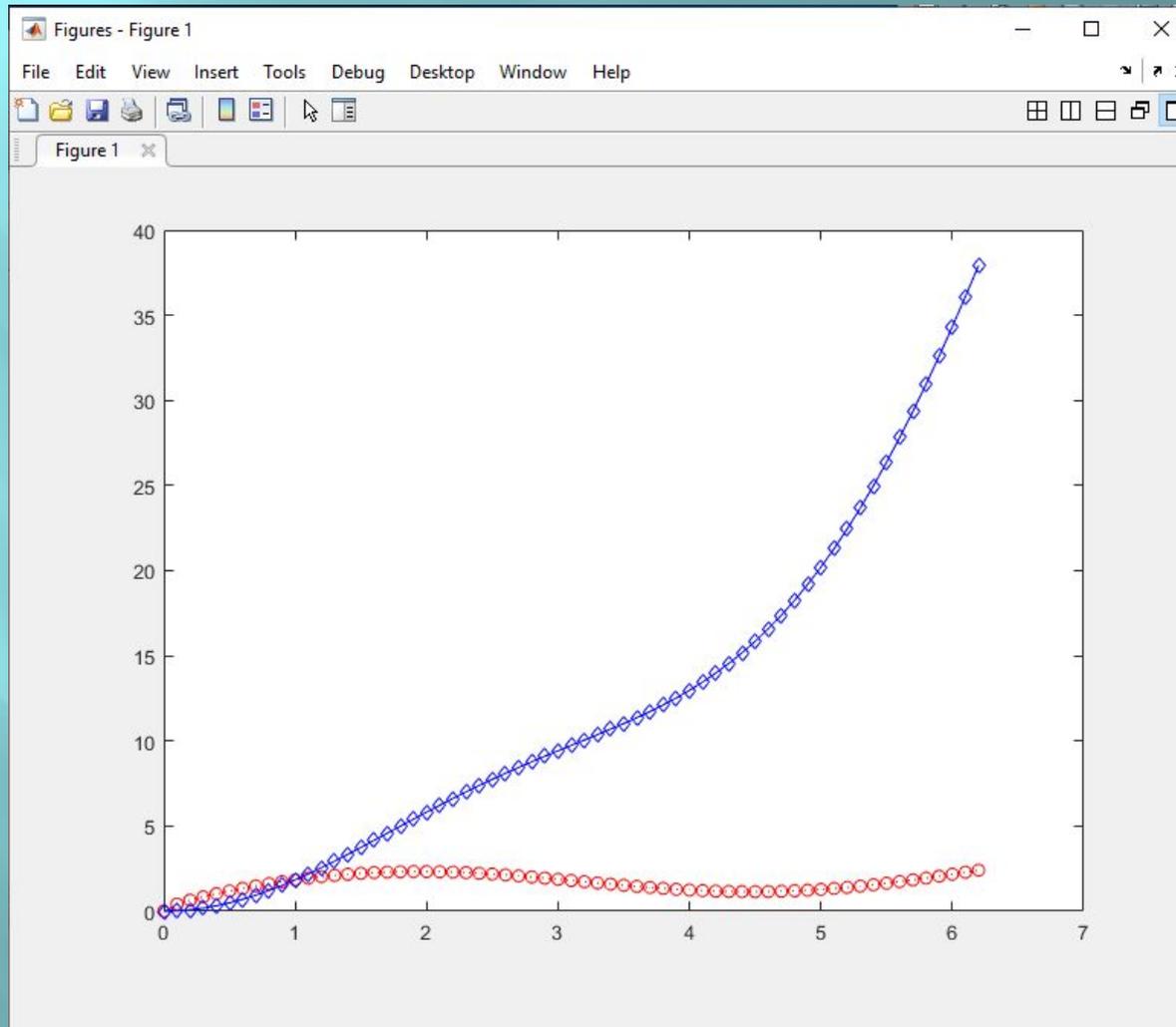
‘это строковый аргумент’

```
x = 0:0.1:2*pi;  
f = sin(x)+x.^0.5;  
g = x.*sin(x)+x.^2;  
plot(x,f,'ro:' , x,g,'bd-')
```

График первой функции будет нарисован красной пунктирной линией с круглыми маркерами.

График второй функции будет нарисован синей сплошной линией с ромбовидными маркерами.

# Задание типа линий для кривой (пример)



# Сокращения для задания цвета ЛИНИИ

Сокращение/ RGB триплет	Цвет линии
y/[1 1 0]	Желтый
m/[1 0 1]	Розовый
c/[0 1 1]	Голубой
r/[1 0 0]	Красный
g/[0 1 0]	Зеленый
b/[0 0 1]	Синий
w/[1 1 1]	Белый
k/[0 0 0]	Черный

# Сокращения для задания цвета линии через свойство 'Color'

RGB триплет	
[0 0.4470 0.7410]	
[0.8500 0.3250 0.0980]	
[0.9290 0.6940 0.1250]	
[0.4940 0.1840 0.5560]	
[0.4660 0.6740 0.1880]	
[0.3010 0.7450 0.9330]	
[0.6350 0.0780 0.1840]	

# Сокращения для задания типа ЛИНИИ

Сокращение	Цвет линии
-	Сплошная
:	Пунктирная
-.	Штрих-пунктирная
--	штриховая

# Сокращения для задания маркера ЛИНИИ

маркер	тип	маркер	тип
.	Точка	v	Треугольник вершиной вниз
o (буква)	Кружок	^	Треугольник вершиной вверх
x	Крестик	<	Треугольник вершиной влево
+	Знак плюс	>	Треугольник вершиной вправо
*	Звездочка	p	Пятиконечная звезда
s	Квадрат	h	Шестиконечная звезда
d	Ромб		

# Опции для оформления надписей

Сокращение	Цвет линии
'FontName'	Шрифт текста
'FontSize'	Размер шрифта
'FontAngle'	Угол наклона шрифта (как правило - 'italic')
'FontWeight'	Толщина линии

# Опции для оформления символов

Сокращение	Цвет линии
'MarkerEdgeColor'	Обводка символа
'MarkerFaceColor'	Заливка символа (по умолч. - отсутствует)
'MarkerSize'	Размер символа

# Функции для оформления графика

Команда	Описание
<code>grid on</code>	Сетка на график наносится
<code>grid off</code>	Сетка на график не наносится (по умолч.)
<code>title('Заголовок графика')</code>	Заголовок графика, расположен над графиком посередине
<code>xlabel('Подпись к оси x')</code>	Подпись по оси абсцисс
<code>ylabel('Подпись к оси y')</code>	Подпись по оси ординат

# Задание легенды

**legend**( {'кривая 1', 'кривая 2', ..., 'кривая N'}, 'Location', lcn)  
Строковых переменных с описанием кривых должно быть столько же сколько кривых на графике!

lcn	Описание
bestoutside	Вне графика, правый верхний угол графического окна
best	автоматически выбирается наилучшее положение в пределах графика
northeast	правый верхний угол графика (по умолч.)
northwest	левый верхний угол графика
southwest	нижний левый угол графика
southeast	нижний правый угол графика

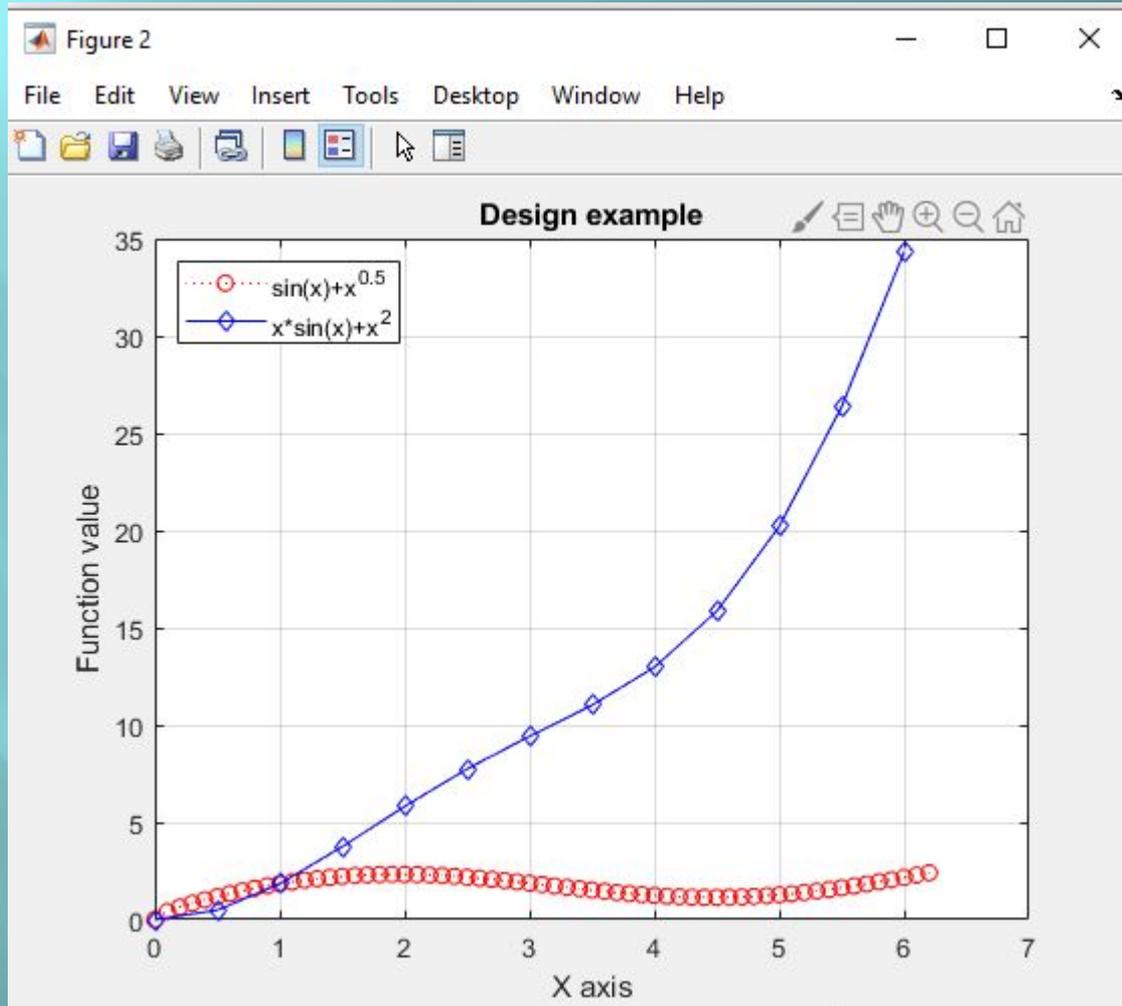
Дополнительные варианты расположения смотрите в Help

# Оформление графика из сценария

```
x = 0:0.1:2*pi;  
x1 = 0:0.5:2*pi;  
f = sin(x)+x.^0.5;  
g = x1.*sin(x1)+x1.^2;  
figure;  
plot(x,f,'ro:' , x1,g,'bd-')  
grid on  
title('Design example')  
xlabel('X axis')  
ylabel('Function value')  
legend({'sin(x)+x^{0.5}','x*sin(x)+x^2'},'Location','northwest');
```

**- на первом месте, потом все настройки!!!**

# Оформление графика из сценария (пример)



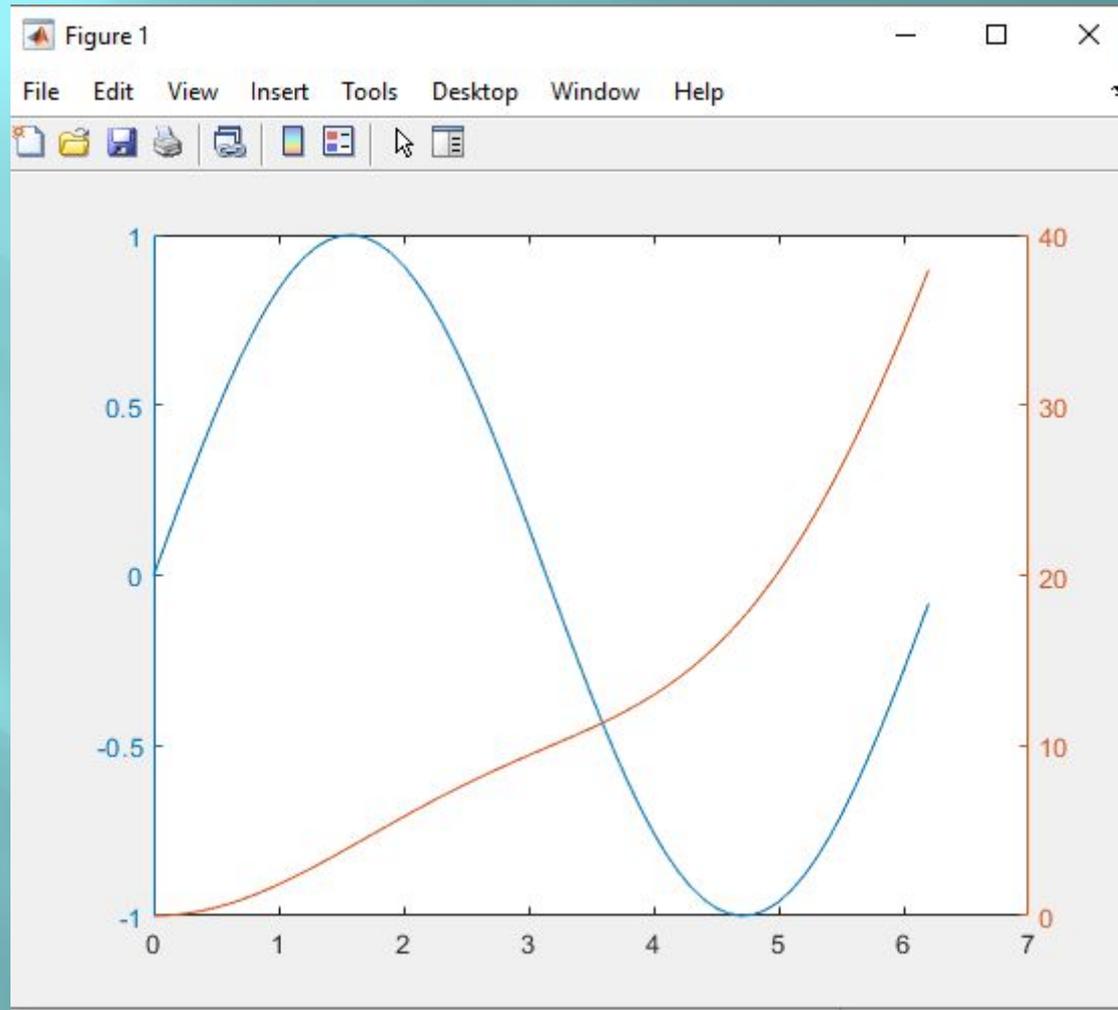
# Несколько графиков с разными осями ординат

Если необходимо сравнить качественное поведение двух графиков, имеющих существенно разные масштабы, то можно создать график с двумя осями.

```
x = 0:0.1:2*pi;  
f = sin(x);  
g = x.*sin(x)+x.^4;  
plotyy(x,f, x,g)
```

Ось абсцисс будет общая, оси ординат – различны! В последних версиях изменить характеристики линий в данной функции нельзя!

# Несколько графиков с разными осями ординат (пример)



# Несколько графиков с разными осями в одном окне

1. Создаем фигуру, в которой будем рисовать **figure;**
2. Дать команду удерживать все графики на фигуре **hold on;**
3. Создать первую зону для графика , она активна в момент создания

**subplot(2,1,1);**

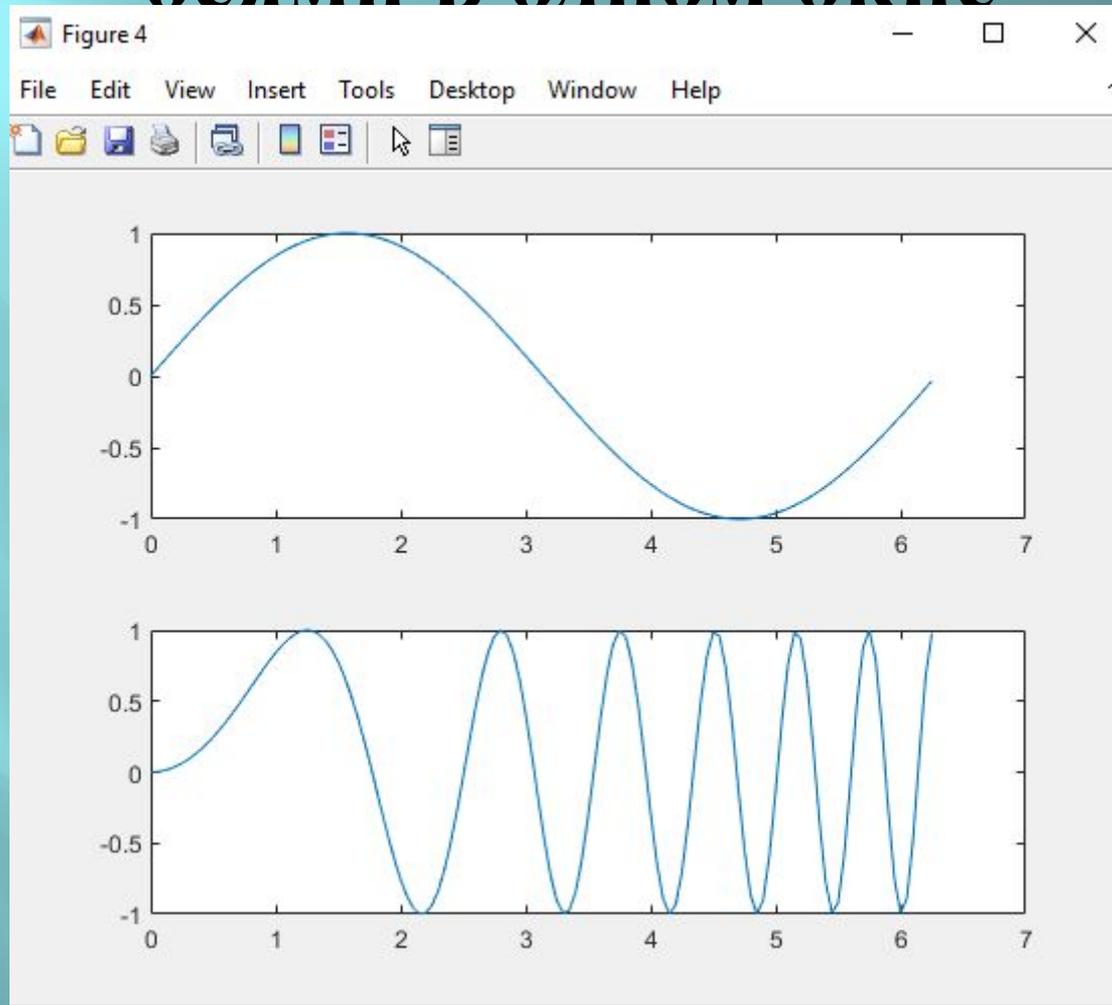
% аргумент 1 – на сколько подокон разбиваем по горизонтали  
% аргумент 2 – на сколько подокон разбиваем по вертикали  
% аргумент 3– номер подокна в которых чертится график

# Несколько графиков с разными осями в одном окне

```
x = 0:0.05:2*pi; % аргумент функций
f = sin(x);      % вычислили вектор значений для функции f
figure; %создали новую фигуру
hold on; % удерживаем все, что в ней будем рисовать
subplot(2,1,1); % создать первую зону для графика, она
                активна в момент создания
plot(x,f)       % построили график в первой паре осей
g = sin(x.^2);  % вычислили вектор значений для функции g
subplot(2,1,2); % создать вторую зону для графика, она
                активна в момент создания
plot(x,g) % построили график во второй паре осей
```

Все команды для построения графиков – применимы, относятся к последнему (активному) подокну!

# Несколько графиков с разными осями в одном окне



# Несколько графиков с разными осями в одном окне

```
x = 0:0.1:2*pi;  
x1 = 0:0.5:2*pi;  
f = sin(x)+x.^0.5;  
g = x1.*sin(x1)+x1.^2;  
figure;  
hold on;  
subplot(2,1,1);  
plot(x,f);  
grid on;  
title('Design example -1st window');
```

# Несколько графиков с разными осями в одном окне

```
xlabel('axis X');  
ylabel('Function f');  
legend({'sin(x)+x^{0.5}'},'Location','northwest');  
subplot(2,1,2);  
plot(x1,g,'r-')  
grid on;  
title('Design example - 2^{nd} window');  
xlabel('axis X1');  
ylabel('Function g');  
legend({'x*sin(x)+x^2'},'Location','best');
```

# Несколько графиков с разными осями в одном окне

