

Графические возможности Matlab

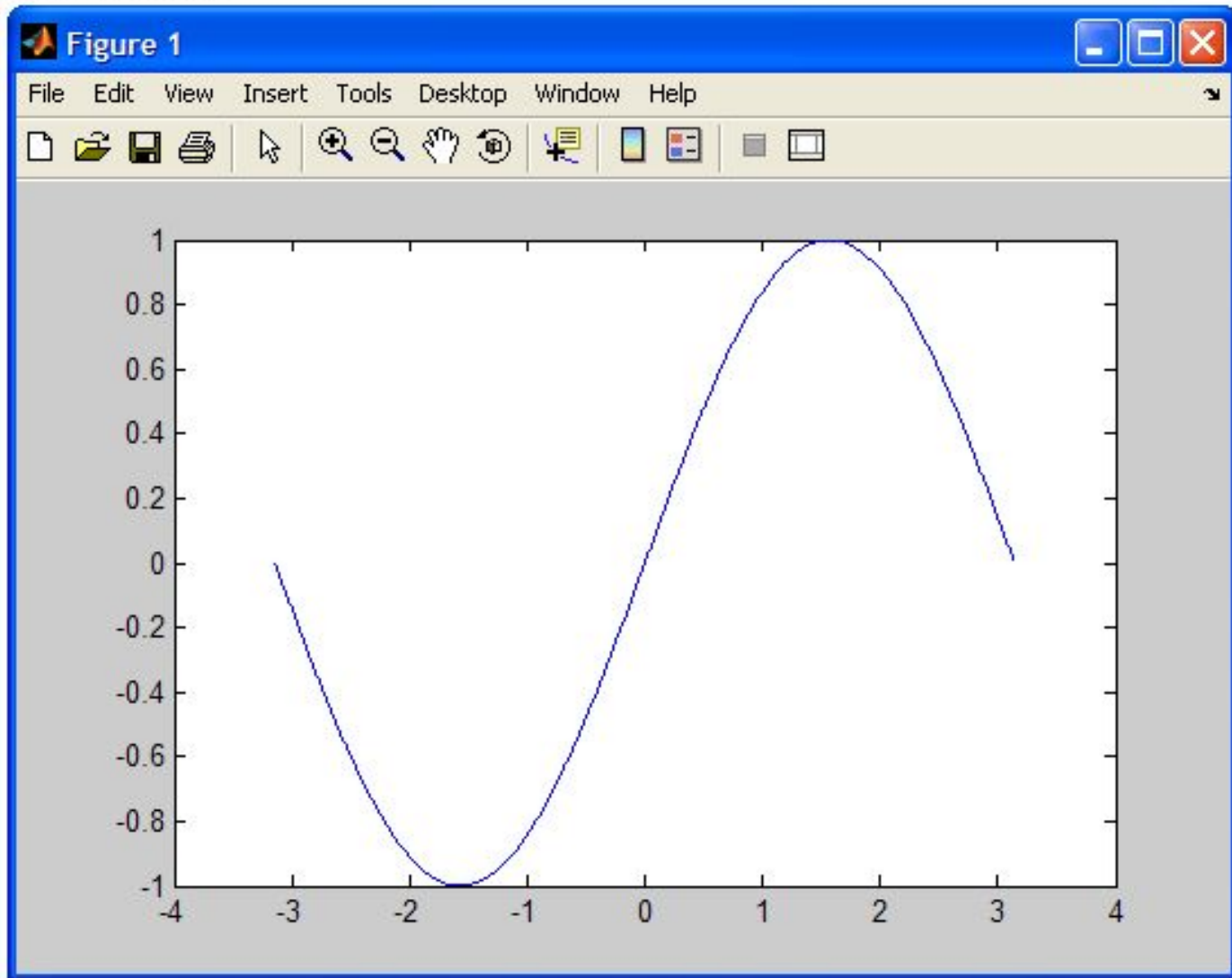
Графика в Matlab

- Высокоуровневая
 - не требует от пользователя детальных знаний о работе графической подсистемы
- Объектная
 - каждый объект на рисунке имеет свойства, которые можно менять
- Управляемая (*handled*)
 - доступ к графическим объектам возможен как через инспектор объектов, так и при помощи встроенных функций (дескрипторная графика)

Двумерные (2D-) графики

- Простейший способ построения 2D-графика:
 1. задать область построения (диапазон);
 2. вычислить значение функции на области построения
 3. построить график при помощи одной из встроенных функций Matlab

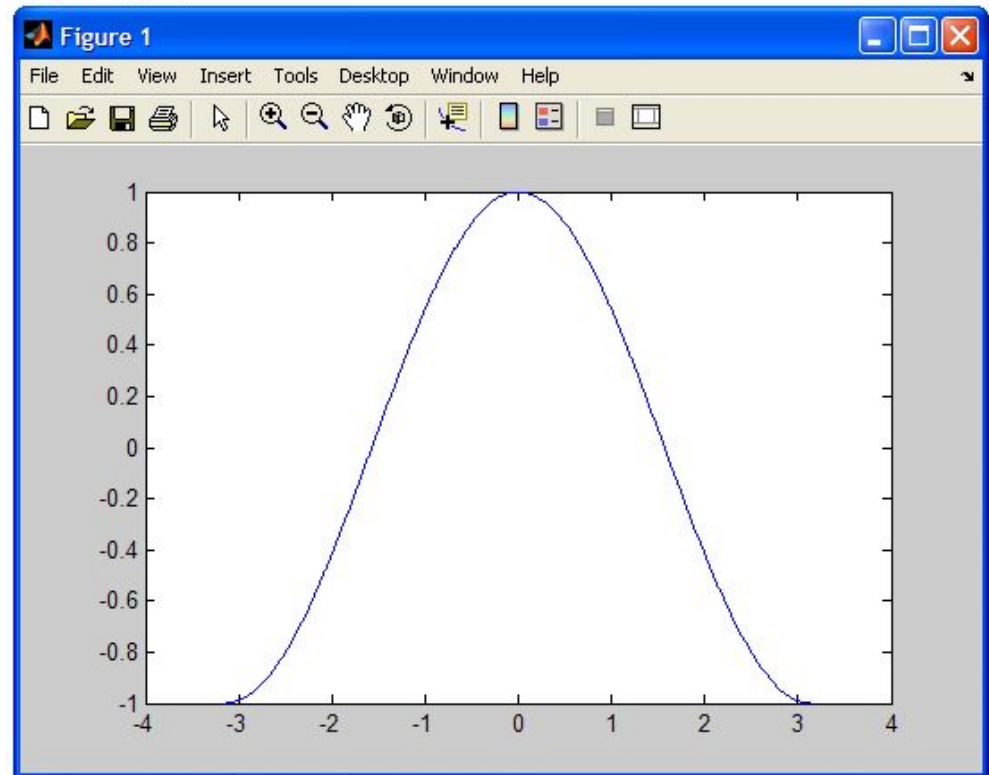
```
>> x = -pi: .01: pi;  
>> y = sin(x);  
>> plot(x, y)
```



Построение второго графика

- Если сразу же построить другой график, то старый график будет удалён из графического окна

```
>> x = -pi: .01: pi;  
>> y = sin(x);  
>> plot(x, y)  
>> z = cos(x);  
>> plot(x, z)
```

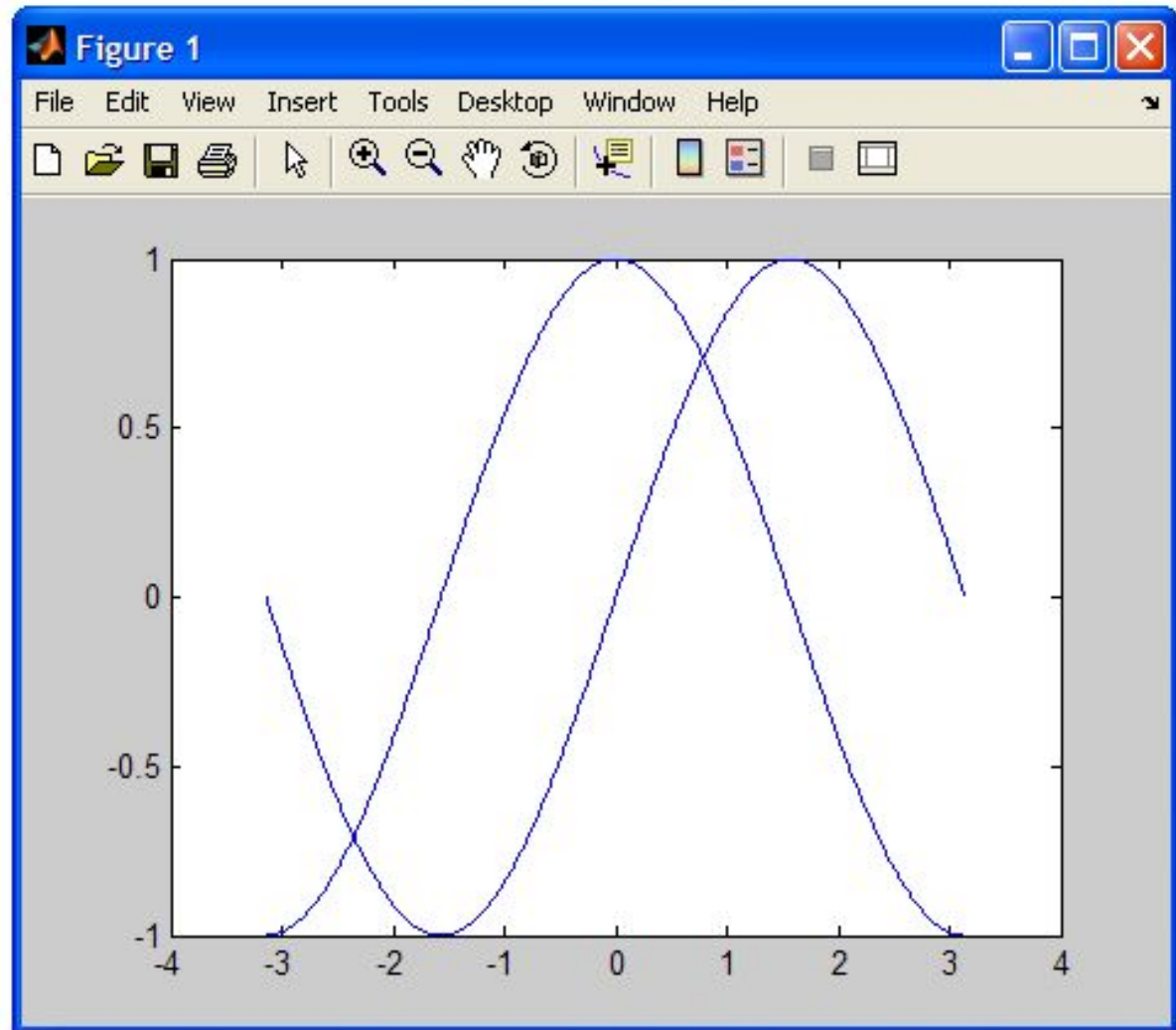


Построение двух графиков в одной системе координат

- Два графика в одной СК можно построить следующими способами:
 1. «закрепить» графическое окно при помощи команды *hold on*
 2. применить одну команду *plot*

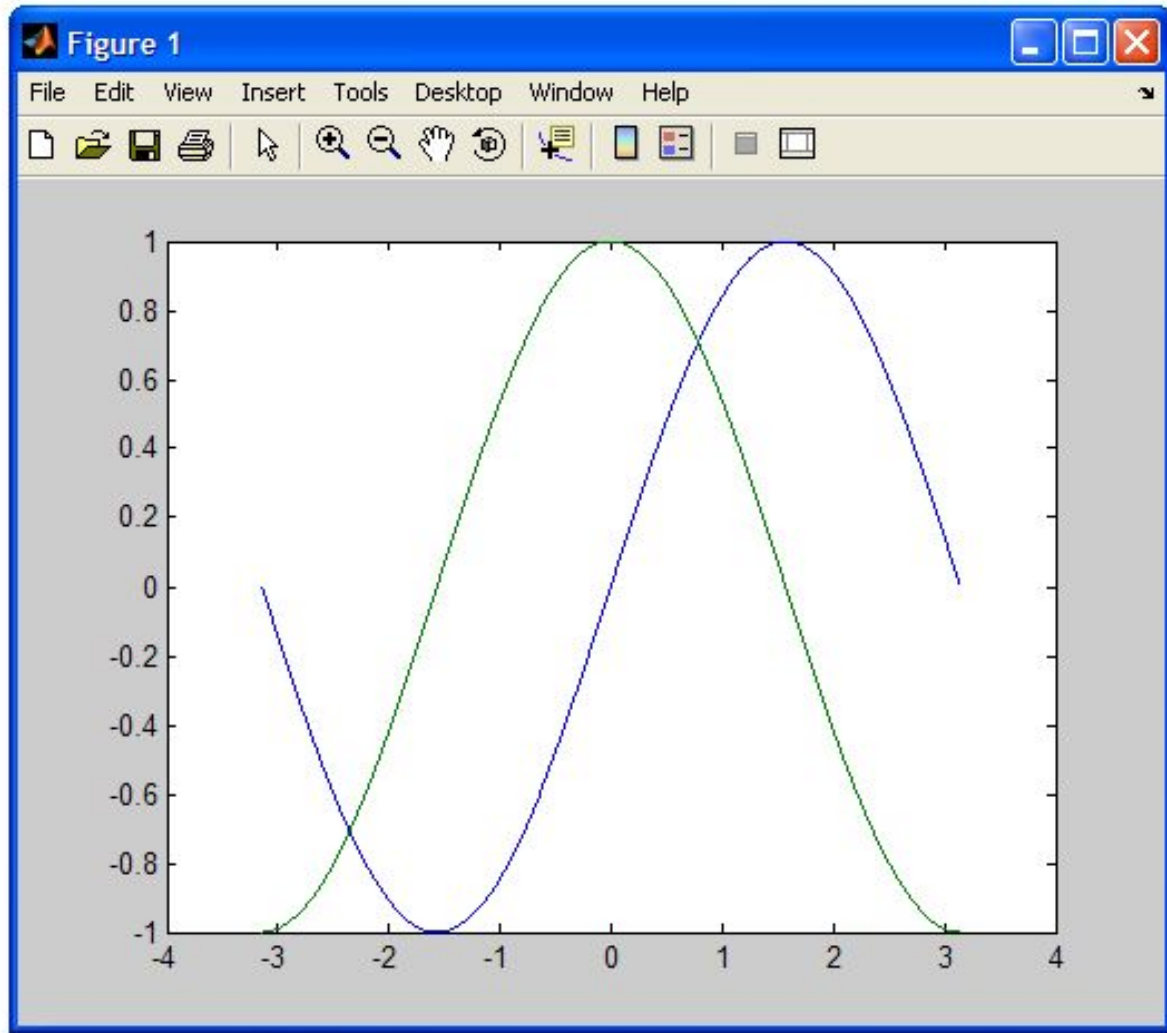
Закрепление графического окна

```
>> x = -pi: .01: pi;  
>> y = sin(x);  
>> plot(x, y)  
>> z = cos(x);  
>> hold on  
>> plot(x, z)  
>> z = cos(x);
```



Дополнительные параметры команды `plot`

```
>> x = -pi: .01: pi;  
>> y = sin(x);  
>> z = cos(x);  
>> plot(x, y, x, z)  
>>
```



Дополнительные параметры команды *plot*

- В команде *plot* можно задать для каждого графика

– цвет линии

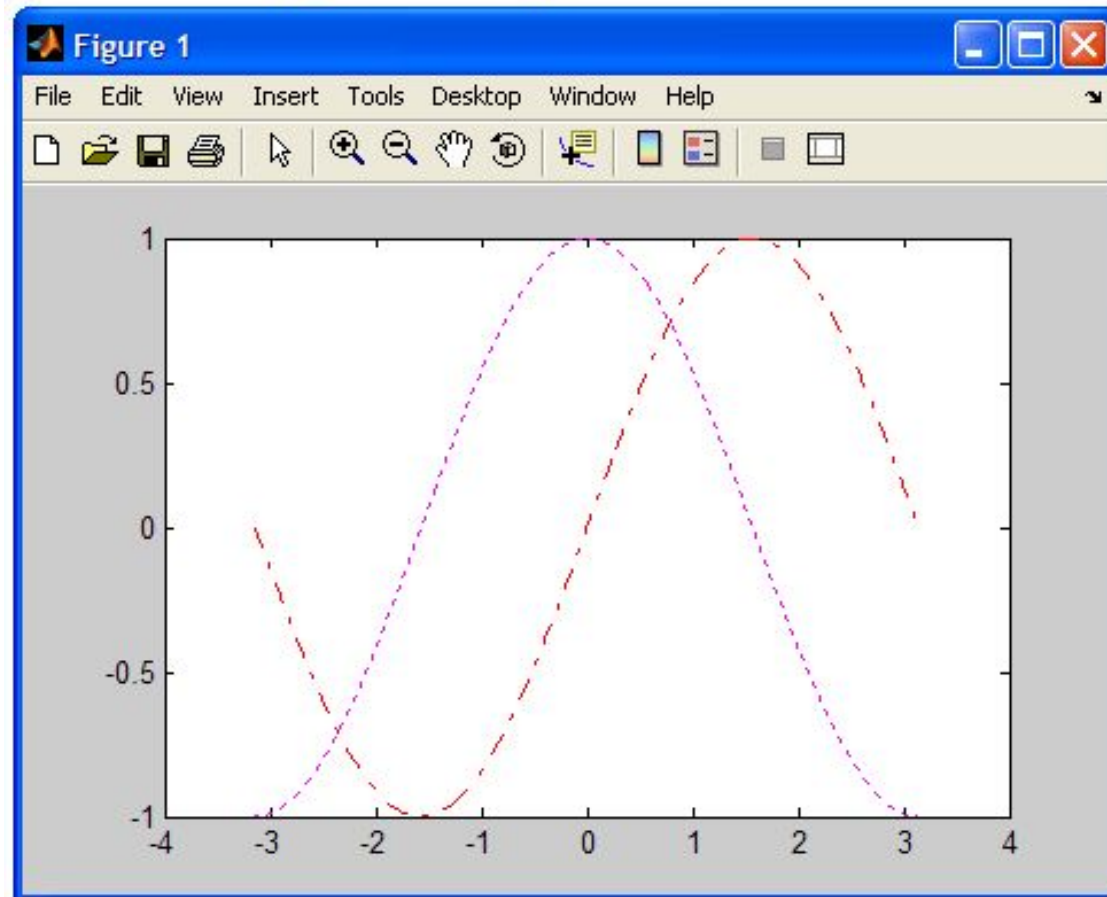
тип маркера

тип линии

b	blue	.	point	-	solid
g	green	o	circle	:	dotted
r	red	x	x-mark	-.	dashdot
c	cyan	+	plus	--	dashed
m	magenta	*	star	(none)	no line
y	yellow	s	square		
k	black	d	diamond		
		v	triangle (down)		
		^	triangle (up)		
		<	triangle (left)		
		>	triangle (right)		
		p	pentagram		
		h	hexagram		

Пример команды `plot`

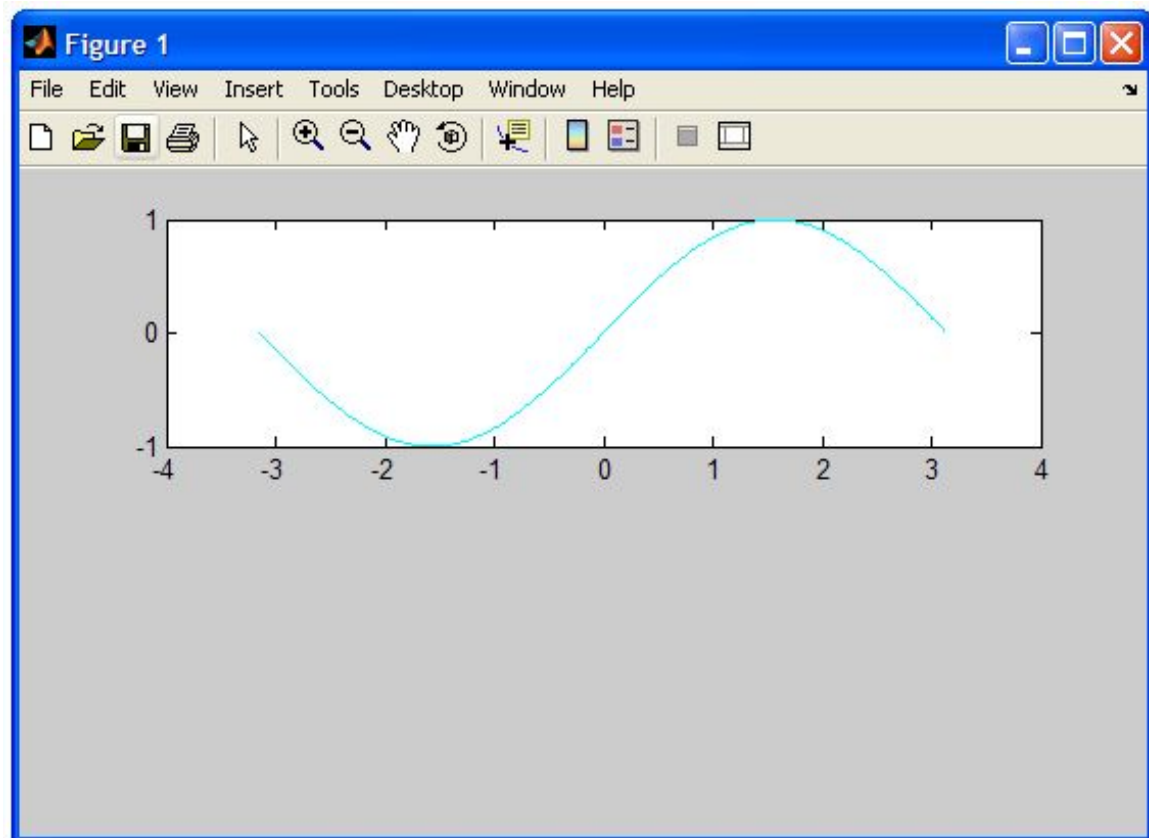
```
>> x = -pi: .01: pi;  
>> y = sin(x);  
>> z = cos(x);  
>> plot(x, y, 'r-.', x, z, 'm:')
```



Построение нескольких графиков в одном окне в разных СК

- Поверхность графического окна можно разделить на зоны, в каждой из которых выводить свой график
- Для этого служит команда *subplot*
- В качестве параметров ей передаётся трёхзначное целое вида *mnk*
- *m* и *n* определяют количество графических «подокон» по горизонтали и вертикали
- *k* задаёт номер графического «подокна»
– порядок нумерации – по строкам

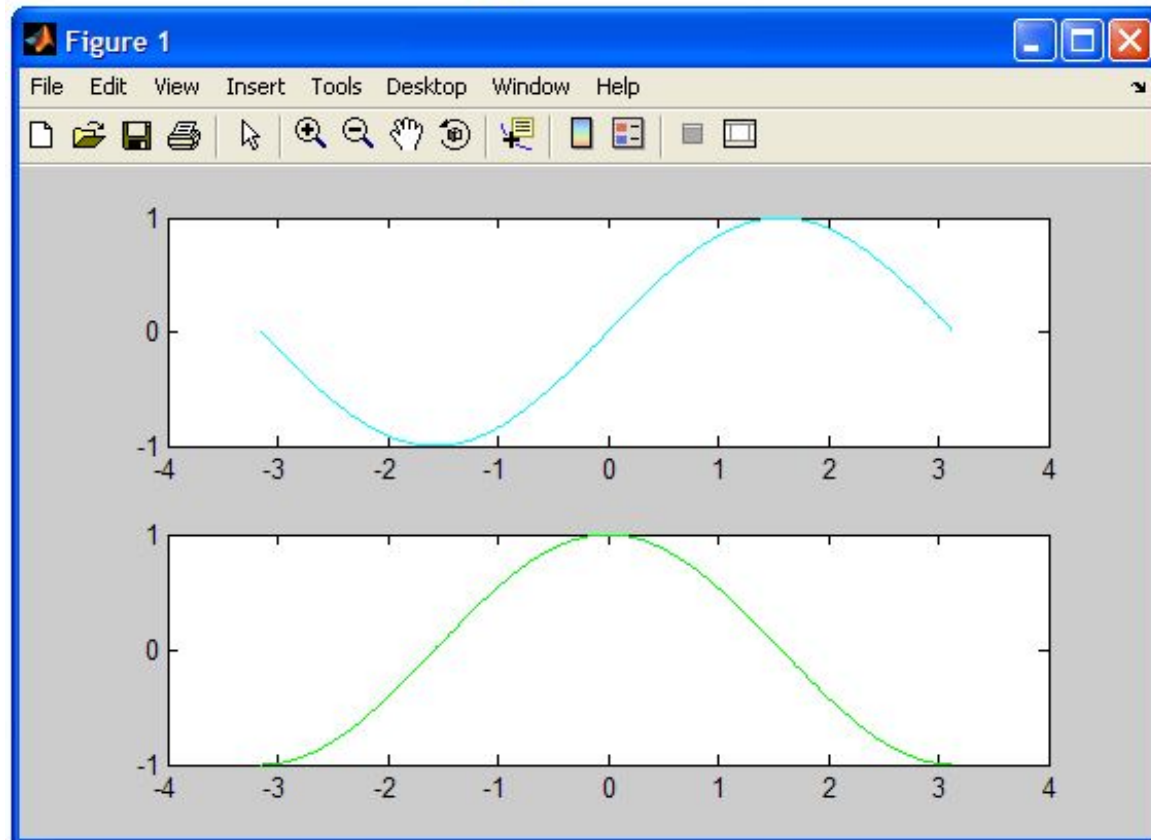
Первый *subplot*



```
>> x = -pi: .01: pi;  
>> y = sin(x);  
>> z = cos(x);  
>> subplot(211); plot(x, y, 'c')
```

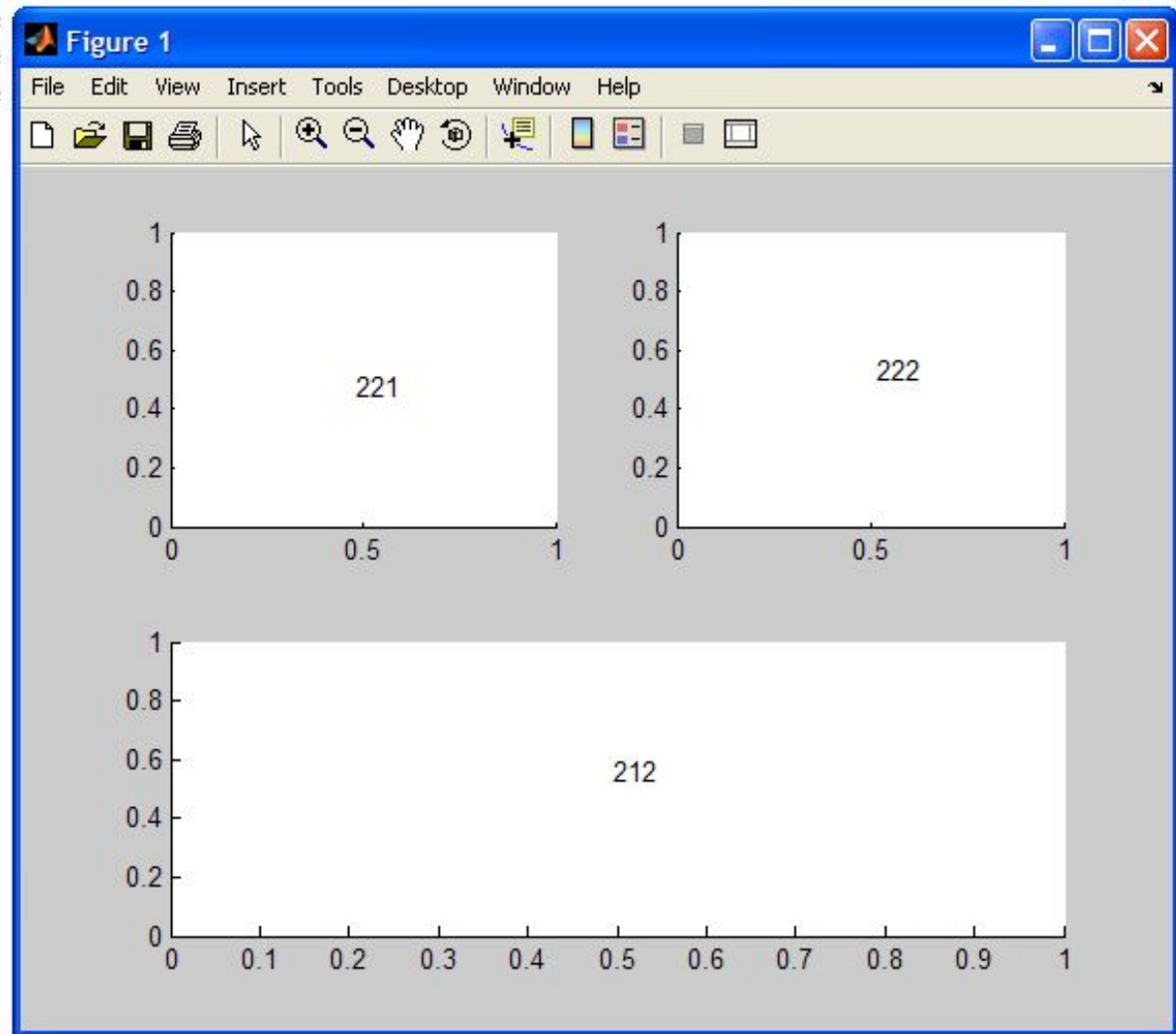
Второй *subplot*

```
>> x = -pi: .01: pi;  
>> y = sin(x);  
>> z = cos(x);  
>> subplot(211); plot(x, y, 'c')  
>> subplot(212); plot(x, z, 'g')
```



Более хитрый пример *subplot*

```
>> subplot(221);  
>> subplot(222);  
>> subplot(212);  
>>  
>>
```



Построение графиков в разных графических окнах

- Создать новое графическое окно можно командой *figure*
- Команда *figure* создаёт графическое окно и возвращает указатель на него:
$$h = figure$$
- Активизировать ранее созданное окно можно командой *figure(h)*

figure : пример использования 1

```
>> x = -pi: .01: pi;  
>> y = sin(x);  
>> z = cos(x);  
>> plot(x, y)  
>> figure  
>> plot(x, z, 'r')
```

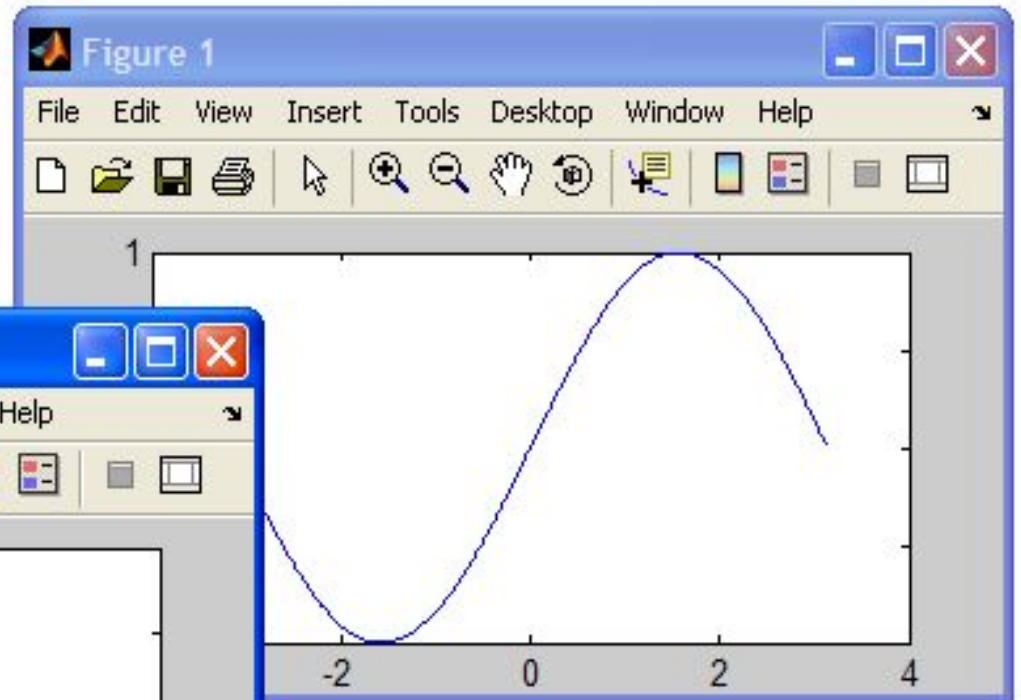
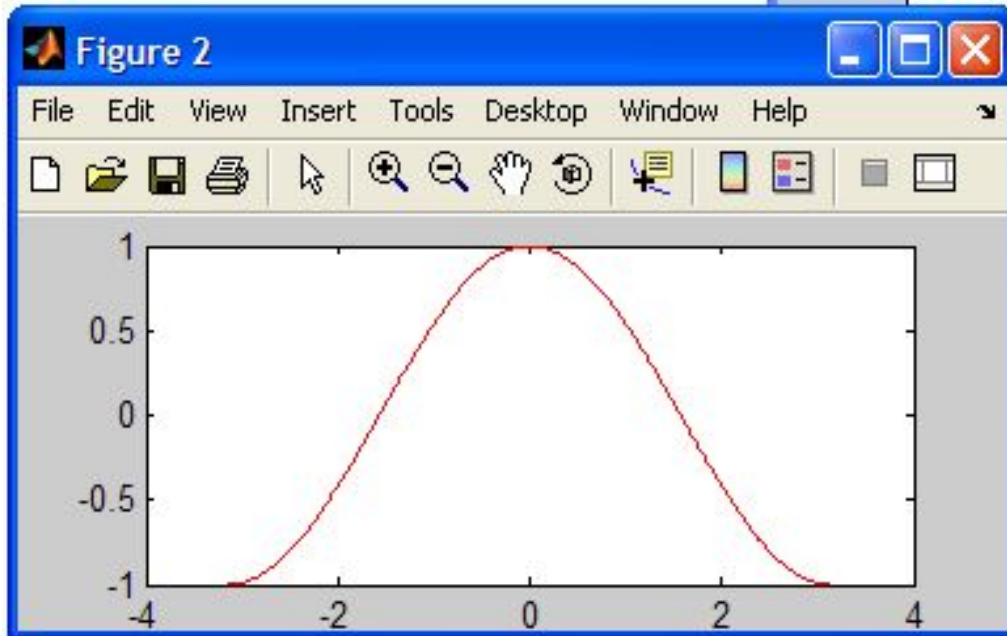


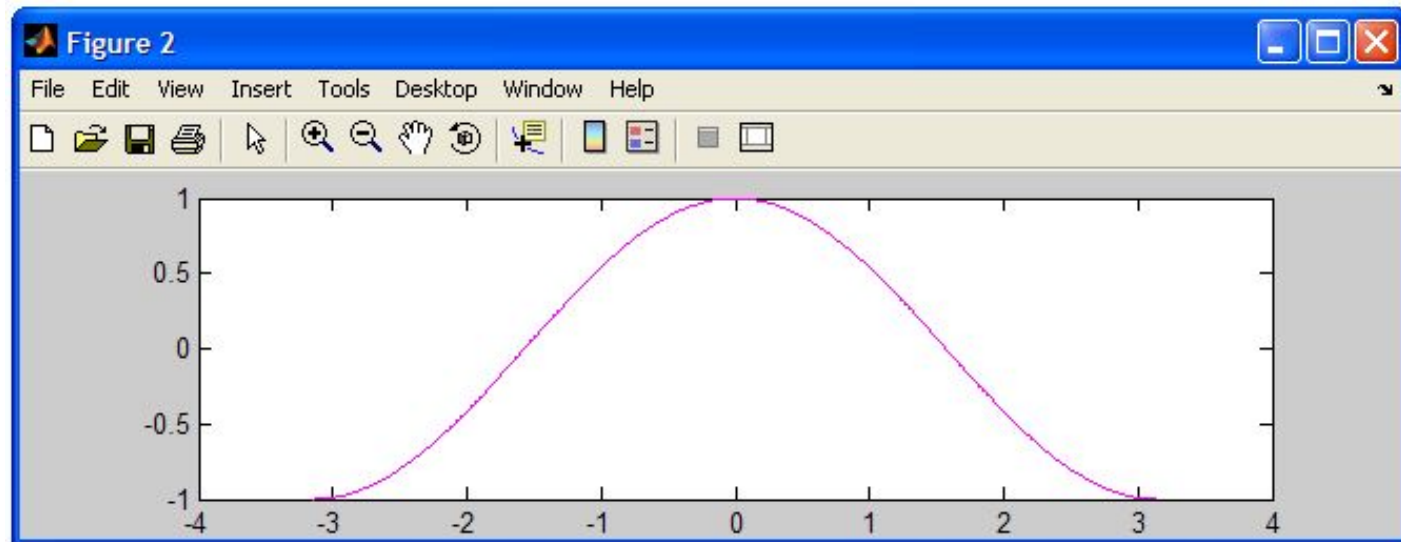
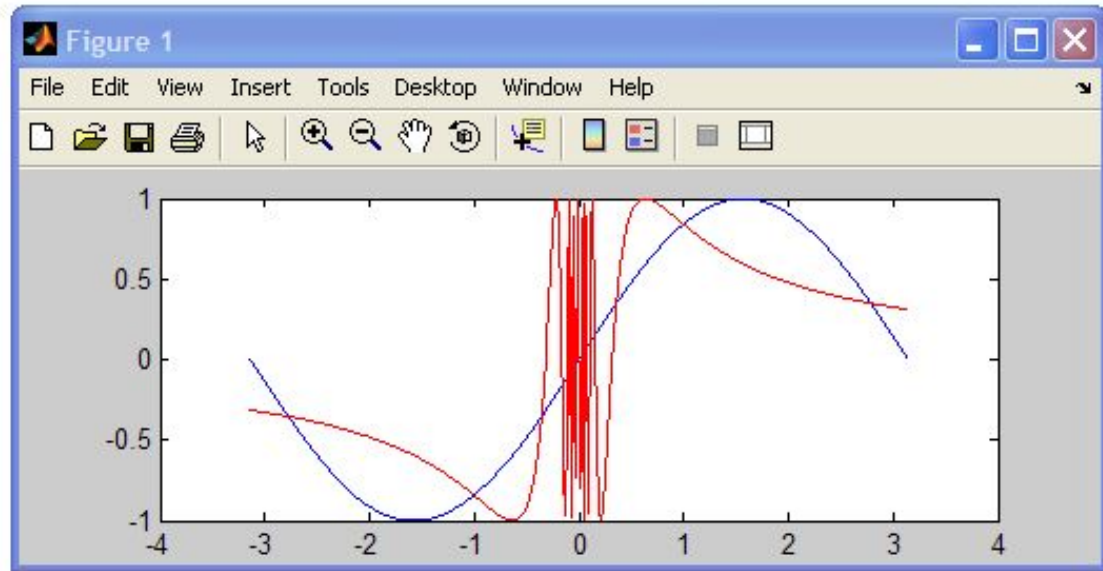
figure : пример использования 2

```
>> x = -pi: .01: pi;  
>> y = sin(x);  
>> z = cos(x);  
>> v = sin(1./x);  
>> f = figure
```

```
f =
```

```
1
```

```
>> plot(x, y)  
>> figure  
>> plot(x, z, 'm')  
>> figure(f)  
>> hold on  
>> plot(x, v, 'r')
```



Axis: управление масштабом

- Команда

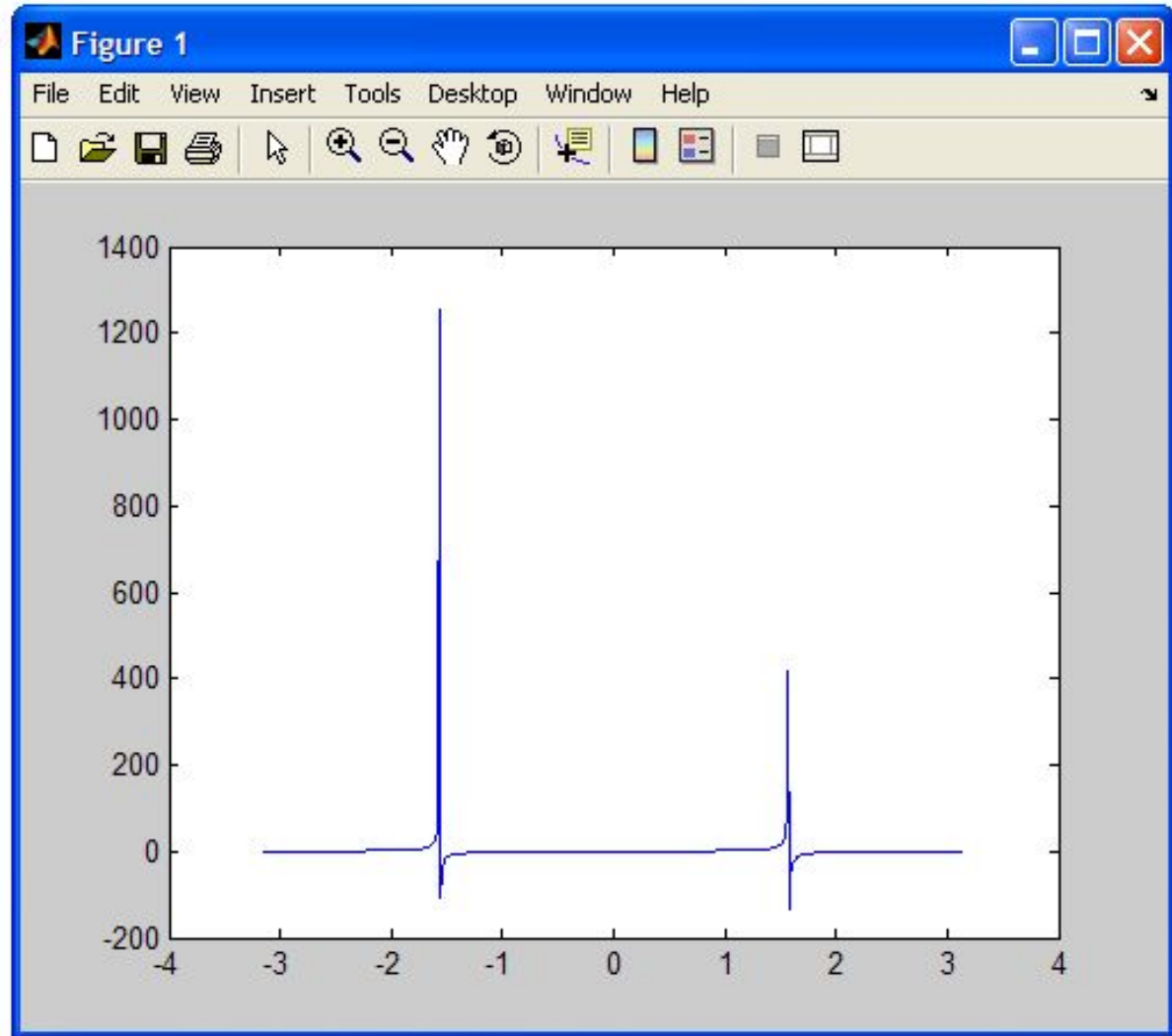
`axis([Xmin Xmax Ymin Ymax])`

задаёт область построения графиков по осям X и Y

- Используется, если результат автомасштабирования неудовлетворителен

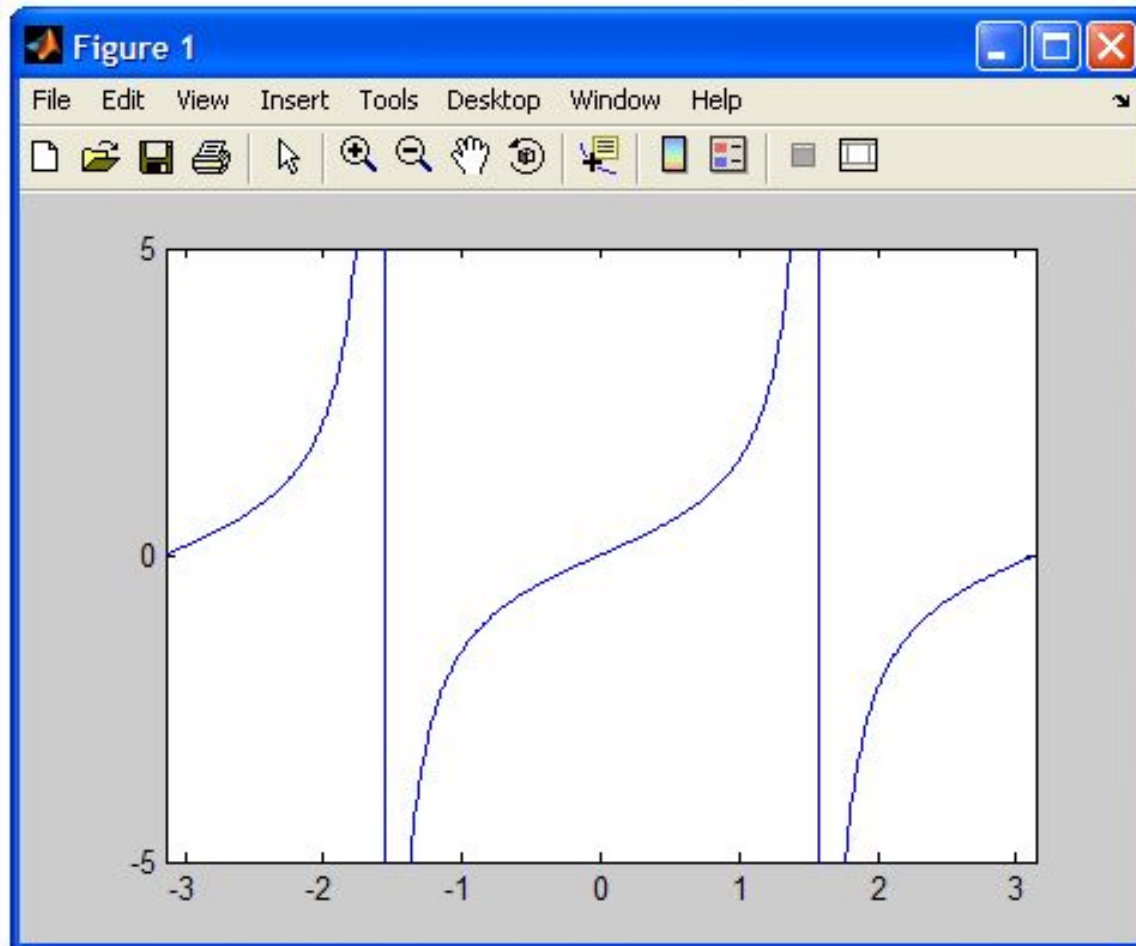
Axis не используется

```
>> x = -pi: .01: pi;  
>> y = tan(x);  
>> plot(x, y)  
>>
```



Axis используется

```
>> x = -pi: .01: pi;  
>> y = tan(x);  
>> plot(x, y), axis([-pi pi -5 5])
```

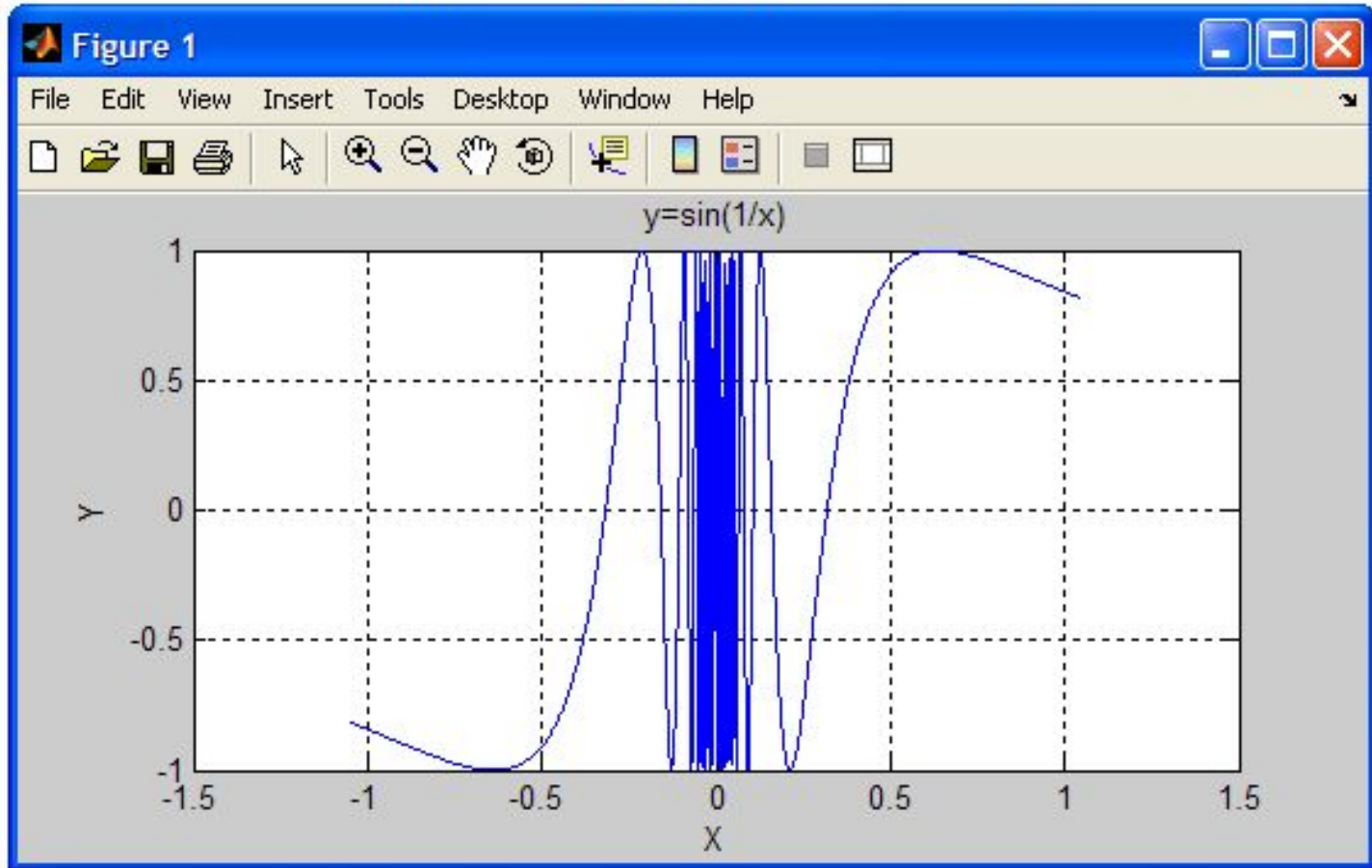


Оформление графиков

- Для графиков можно задать
 - масштабную сетку: `grid on`
 - заголовок: `title('заголовок')`
 - подписи осей: `xlabel('текст')` и `ylabel('текст')`
- В заголовках и подписях можно использовать нотацию системы TeX

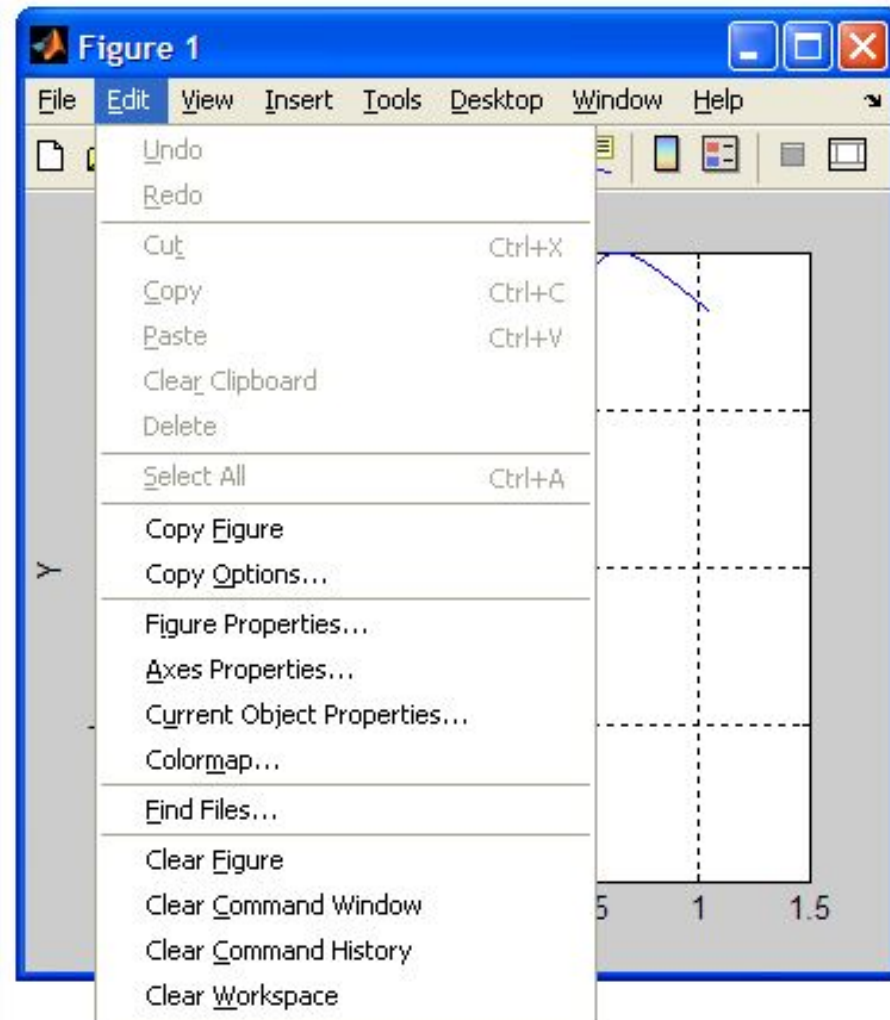
Пример оформления графика

```
>> x = -pi/3: .001: pi/3;  
>> v = sin(1./x);  
>> plot(x, v), grid on, title('y=sin(1/x)'), xlabel('X'), ylabel('Y')
```



Форматирование графиков

- Доступно из меню Edit:
Edit:

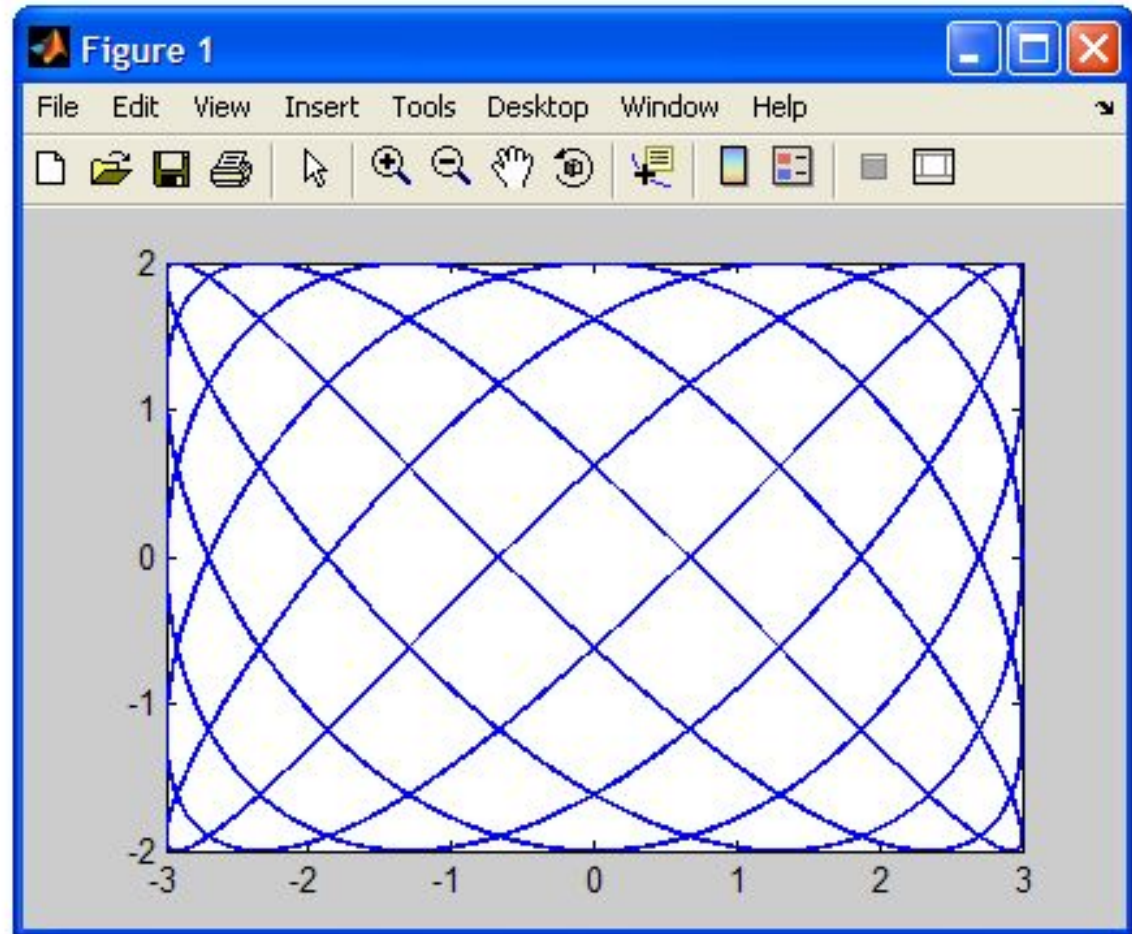


Графики функций, заданных параметрически

- Строятся при помощи оператора `plot`
- Вначале задаётся диапазон построения t
- Затем вычисляются $x(t)$ и $y(t)$
- И строится график

Графики функций, заданных параметрически

```
>> t = 0: .01: 100;  
>> x = 3*cos(5*t);  
>> y = 2*sin(7*t);  
>> plot(x, y)  
>>
```



Графики функций, заданных параметрически

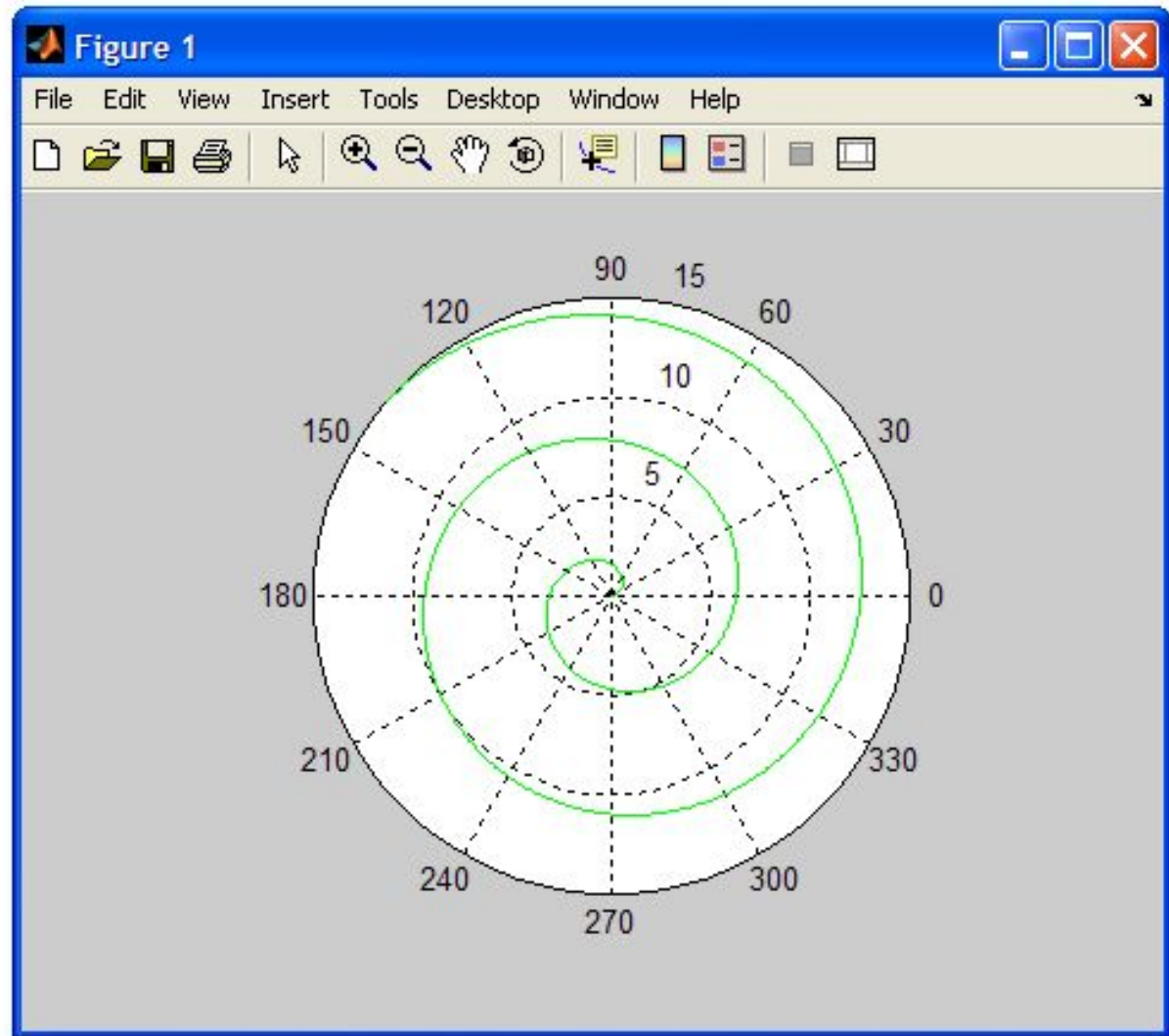
- Графики параметрических функций часто возникают в физических приложениях
- Независимая переменная t в этом случае имеет смысл времени, x и y – координаты
- Для построения динамического графика можно использовать функцию $\text{comet}(x, y)$

Функции в полярной СК

- Строятся аналогично графикам функций в декартовой системе
- Для построения используется команда *polar*

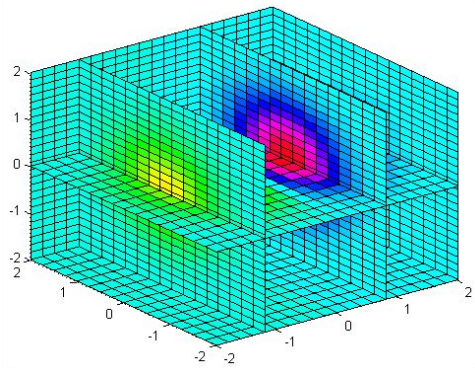
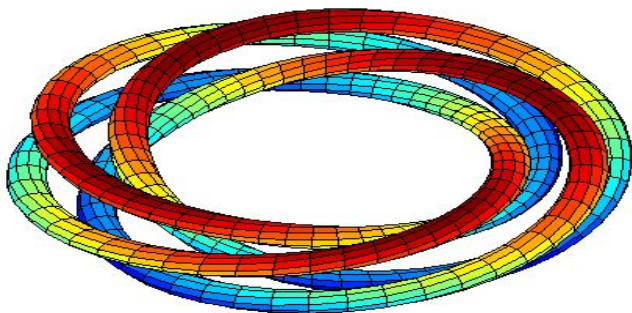
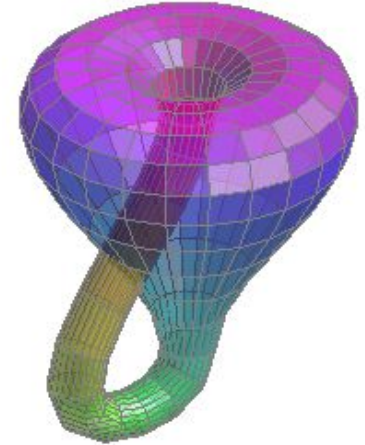
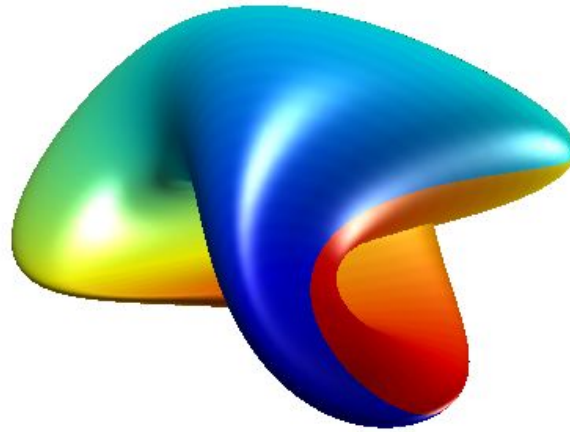
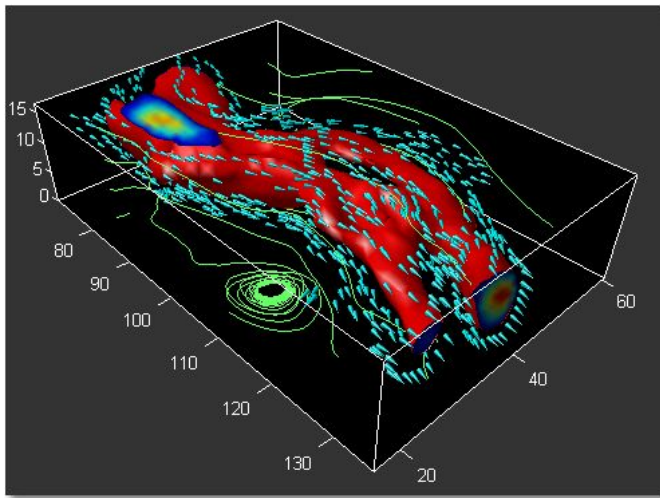
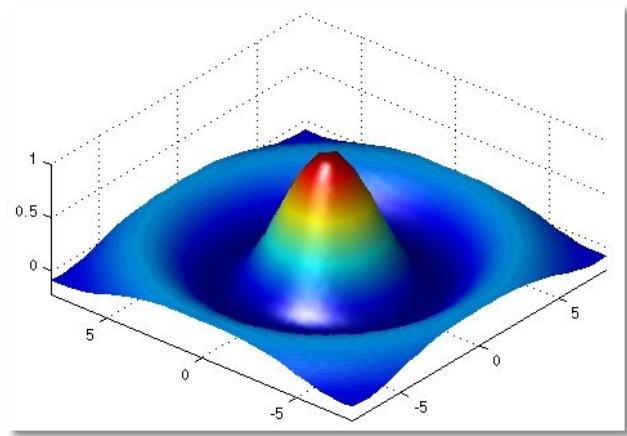
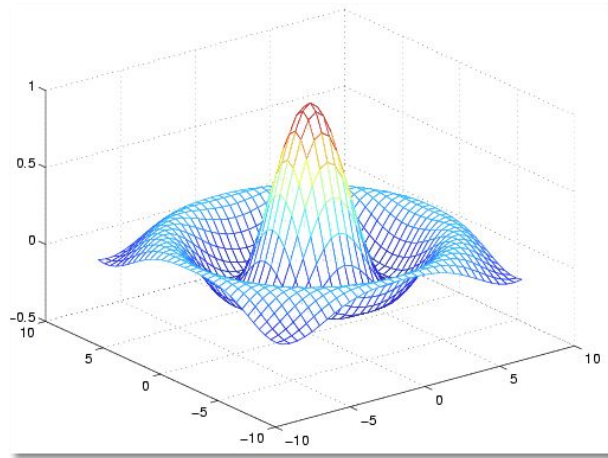
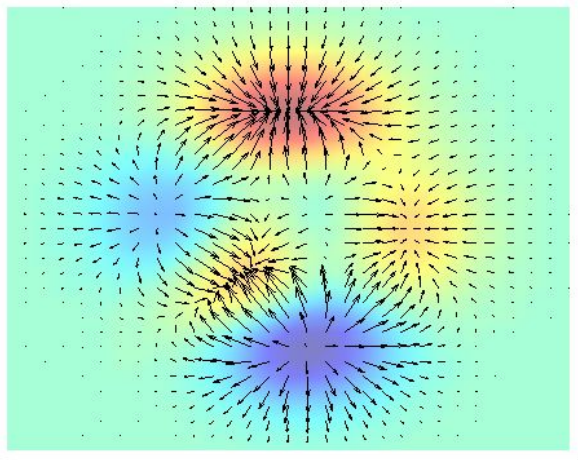
Функции в полярной СК

```
>> phi = 0: .01: 15;  
>> r = phi;  
>> polar(phi, r, 'g')  
>>
```



Трёхмерная (3D-) графика

- Построение
 - поверхностей
 - контурных диаграмм (линии равного уровня)
 - 3D-линий
 - векторных полей
 - скалярных полей
 - и др.

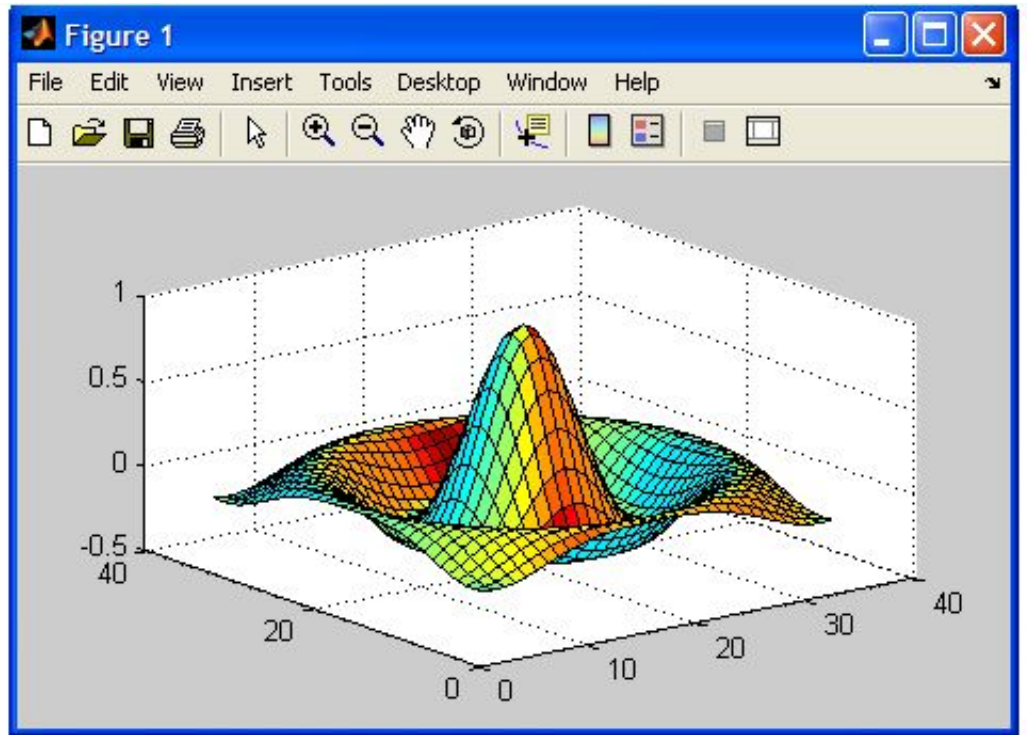


Построение 3D-поверхности

- Рассмотрим пример:

построить поверхность $f(x,y) = \sin(r)/r$, где
 $r = \text{sqrt}(x^2 + y^2)$


```
>> [X,Y] = meshgrid(-8:.5:8);  
>> R = sqrt(X.^2 + Y.^2) + eps;  
>> Z = sin(R)./R;  
>> surf1(Z)  
>>
```



- Функция `meshgrid` возвращает две матрицы – X и Y – которые определяют область построения функции
- Если диапазоны по X и Y разные, то функции передаются два диапазона
- Собственно поверхность выводится функцией `surf1`

Функции для построения поверхностей

<i>Функция</i>	<i>Для чего используется</i>
<code>mesh, surf</code>	Построение поверхностей
<code>meshc, surfc</code>	Строит поверхность и контурную диаграмму под ней
<code>meshz</code>	Поверхность на «пьедестале»
<code>surfl</code>	Подсвеченная поверхность
<code>contour</code>	Контурная диаграмма
<code>plot3</code>	Трёхмерная линия (параметрическое задание)
<code>comet3</code>	Движение по трёхмерной линии

- О других графических функциях можно узнать в системе помощи Matlab