



Декрипторная графика

MATLAB

Лекция #4

Пустовалова О.Г.
доцент. каф. мат.мод.
ИММИКН ЮФУ

Содержание

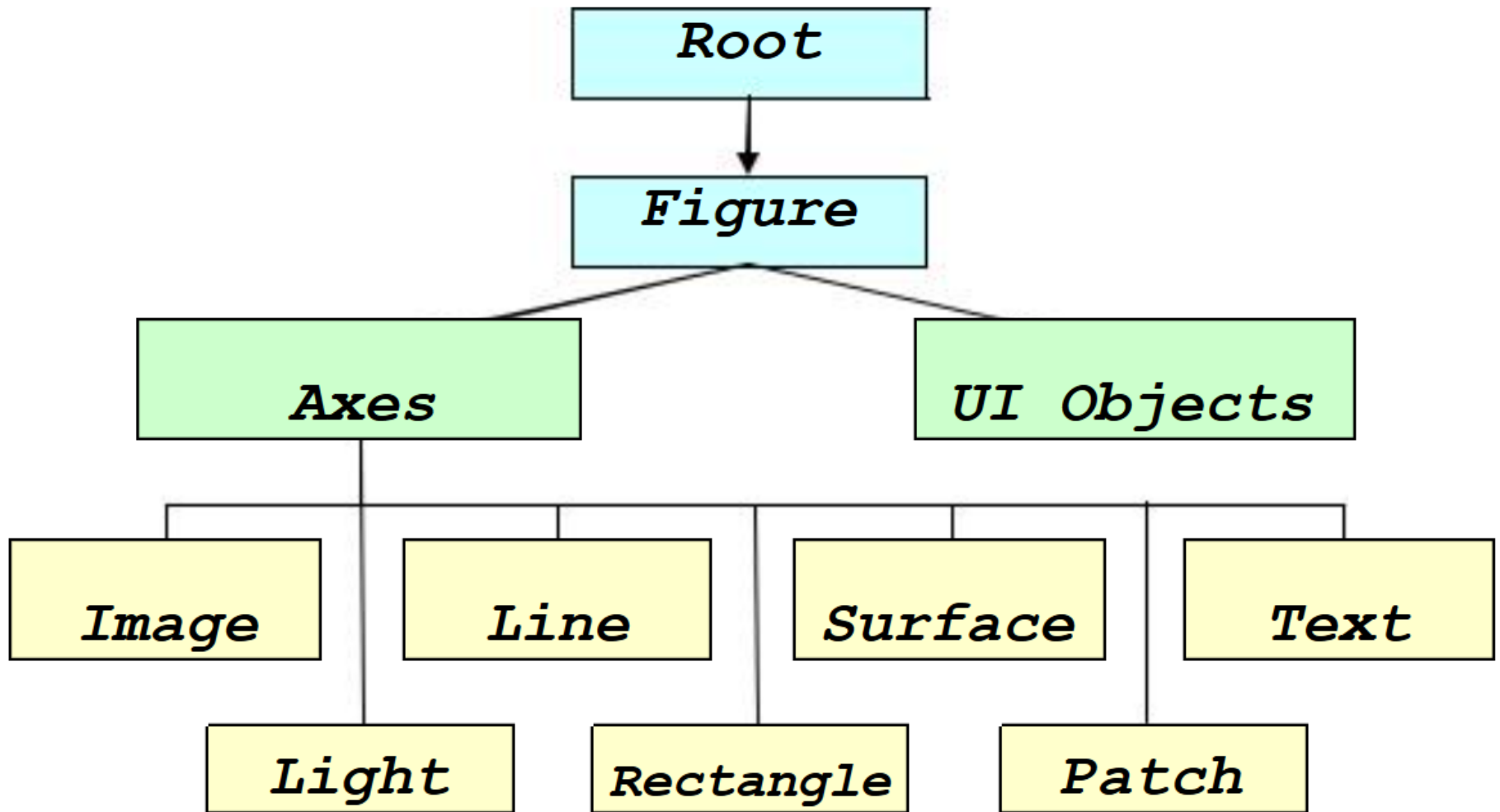
- **Объекты дескрипторной графики**
- **Назначение свойств. Команда set**
- **Команда get**
- **Изменение свойств объекта**
- **Примеры**

Дескрипторная графика

Графические средства MATLAB базируются на низкоуровневой графике, которая называется **дескрипторной** (описательной), или **handle** графикой.

Центральным понятием дескрипторной графики является
графический объект

Иерархия объектов дескрипторной графики MATLAB



Объекты дескрипторной графики

✓ `root`

корень – первичный объект, ~ экран компьютера;

✓ `figure`

рисунок – объект создания графического окна;

✓ `uicontrol`

элемент управления, определенный пользователем – объект создания элемента пользовательского интерфейса;

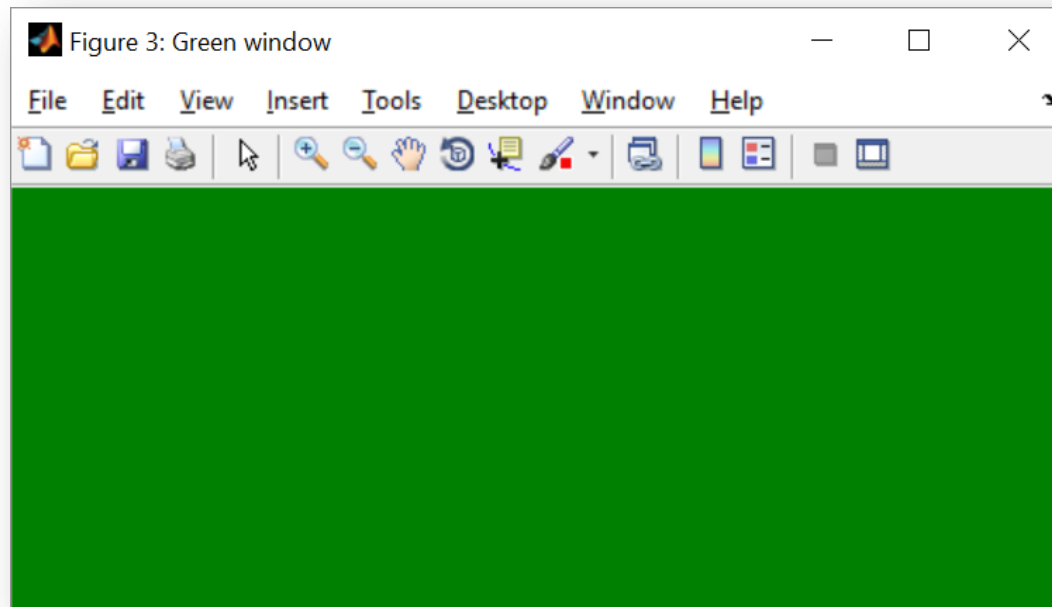
✓ `axes`

оси – объект, задающий область расположения графика в окне объекта `figure`;

Объекты дескрипторной графики. Пример. **figure**

```
hf=figure;
```

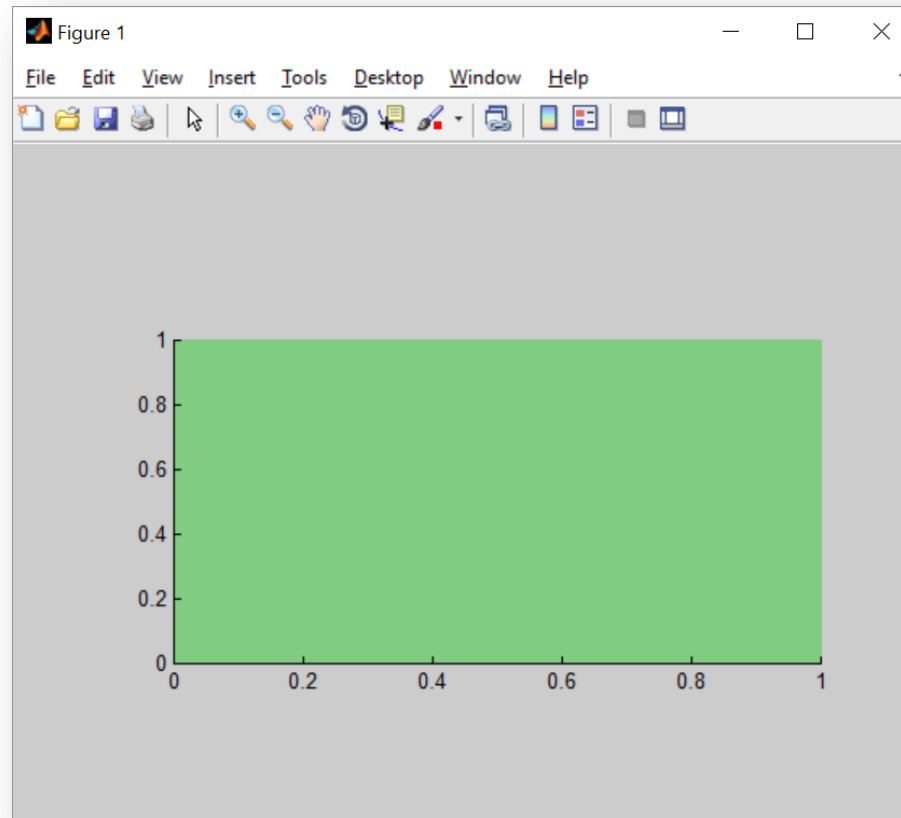
```
set(hf, 'Position', [50 50 500 200], ...  
      'Color', [0 0.5 0],  
      'Name', 'Green window');
```



Объекты дескрипторной графики. Пример. **axes**

```
hA=axes('Color',[0.5, 0.8, 0.5]);
```

```
set(hA,'Units','pixels',  
      'Position',[100, 100, 400, 200]);
```



Объекты дескрипторной графики

✓ `uimenu`

(определенное пользователем меню) – объект создания меню;

✓ `uicontextmenu`

(определенное пользователем контекстное меню) – объект создания контекстного меню;

✓ `image`

(образ) – объект создания растровой графики;

✓ `line`

(линия) – объект создания линии;

Объекты дескрипторной графики

✓ `patch`

объект создания закрасенных фигур;

✓ `rectangle`

прямоугольник – объект создания закрасенных
прямоугольников;

✓ `surface`

поверхность – объект создания поверхности;

✓ `text`

текст – объект создания текстовых надписей;

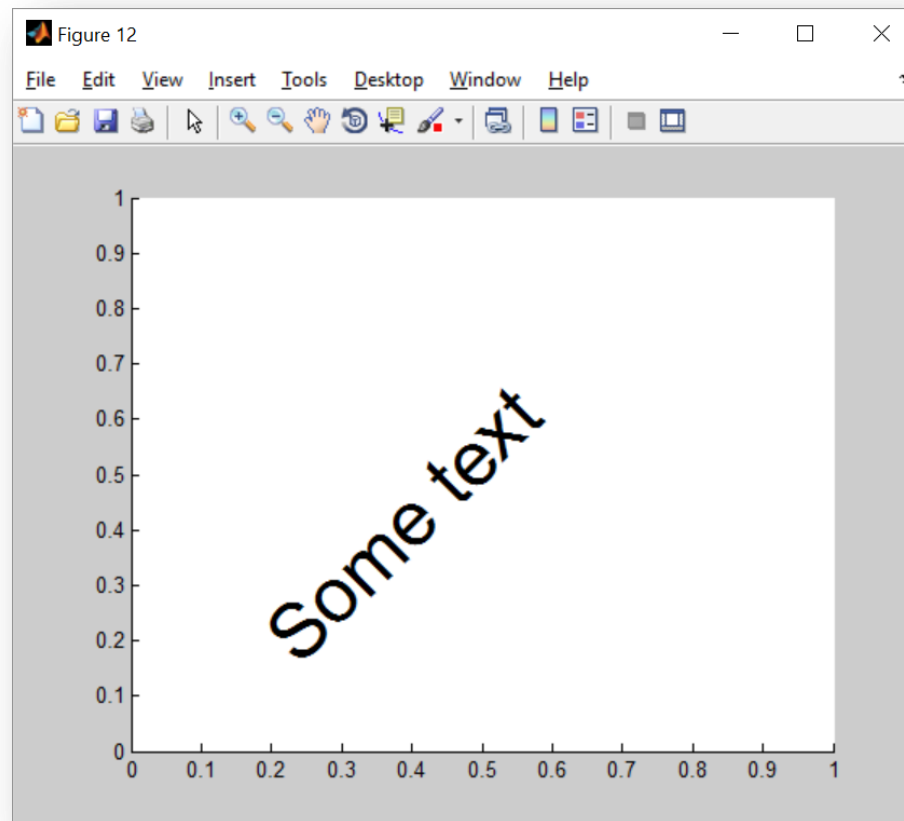
✓ `light`

свет – объект создания эффектов освещенности.

Объекты дескрипторной графики. Пример. **text**

```
hT=text(0.2, 0.2, 'Some text');
```

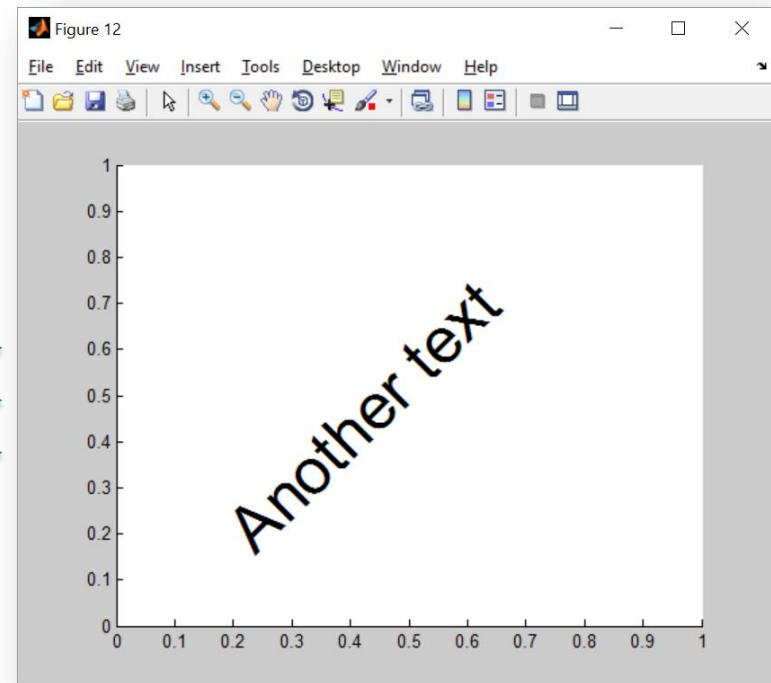
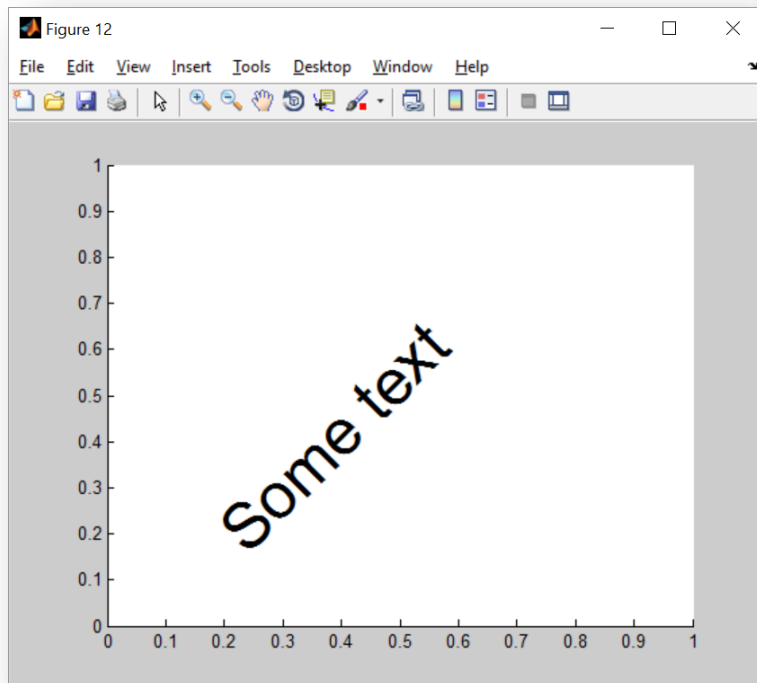
```
set(hT, 'FontName', 'Arial', 'FontSize', 36, 'Rotation', 45);
```



Объекты дескрипторной графики. Пример. **text**

```
hT=text(0.2, 0.2, 'Some text');  
set(hT, 'FontName', 'Arial', 'FontSize', 36, 'Rotation', 45);
```

```
pause(3)  
set(hT, 'String', 'Another text');
```



Создание графического окна и управление им

✓ **figure**

открыть чистое графическое окно;

✓ **gcf**

получить дескриптор графического окна `figure`;

✓ **clf**

очистить графическое окно;

✓ **shg**

показать ранее свернутое графическое окно;

✓ **close**

закреть графическое окно;

✓ **refresh**

обновить графическое окно.

Создание координатных осей и управление ими

✓ **axes**

создать оси координат;

✓ **box**

построить прямоугольник вокруг рисунка;

✓ **cla**

убрать построения axes;

✓ **gca**

получить дескриптор графического объекта axes;

✓ **hold**

сохранить оси координат;

✓ **ishold**

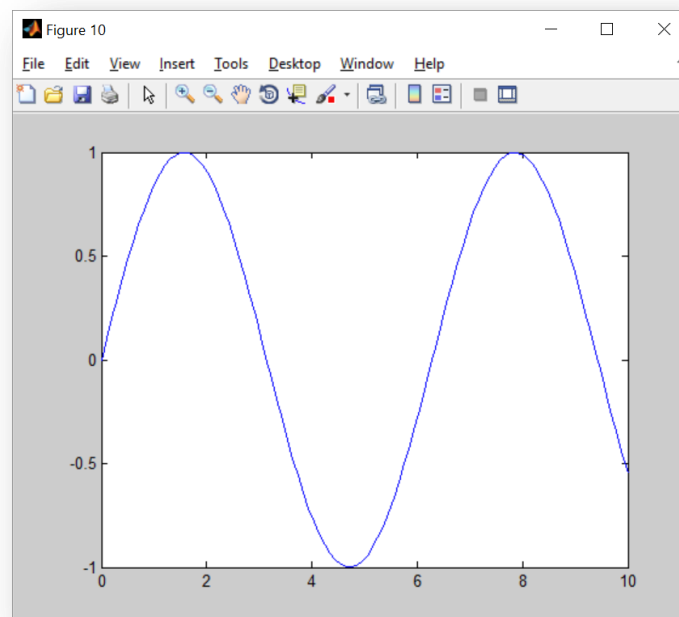
проверка статуса hold (1, если оси сохранены, и 0 в противоположном случае).

Объекты дескрипторной графики. Пример. `gca`

```
x = linspace(0,10);  
y = sin(x);  
plot(x,y);
```

`% получили дескриптор осей`

```
hA=gca;  
set(hA, 'xtick', [0:2:10], 'ytick', [-1:0.5:1]);
```



Дескрипторы объектов

Дескриптор (описатель) – число-идентификатор объекта.

Дескриптор объектов **root** всегда равен **0**.

Дескриптор **figure**– это **целое число** ~ номер графического окна.

Дескрипторы других объектов – это числа с плавающей запятой.

```
A=magic(5);
```

```
h=plot(A)
```



```
h =
```

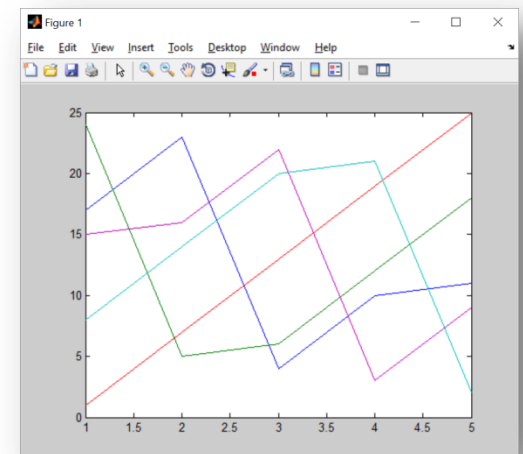
```
174.0044
```

```
175.0039
```

```
176.0039
```

```
177.0039
```

```
178.0039
```



Операции над графическими объектами

✓ **set**

установка свойств графического объекта;

✓ **get**

вывод свойств графического объекта;

✓ **reset**

восстановить свойства графического объекта по умолчанию;

✓ **delete**

удалить созданный графический объект;

Операции над графическими объектами. **get**

```
clc  
clear  
figure(1)  
x = linspace(0,10);  
y = sin(x);
```

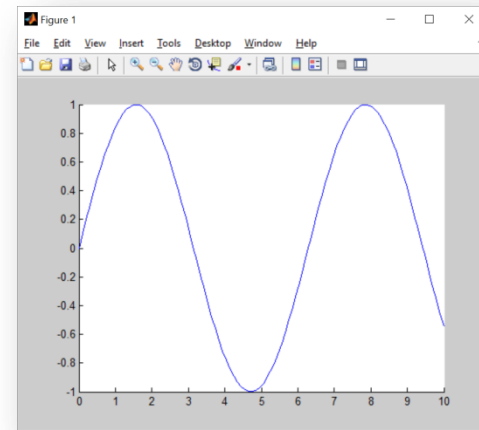
```
h = line(x,y)
```



```
h = 173.0049
```

дескриптор

```
get(h)
```



Операции над графическими объектами. **get**

```
x = linspace(0,10);  
y = sin(x);  
h=line(x,y)
```

```
get(h)      h =
```

```
173.0054
```



```
DisplayName =  
Annotation = [ (1 by 1) hg.Annotation array]  
Color = [0 0 1]  
LineStyle = -  
LineWidth = [0.5]  
Marker = none  
MarkerSize = [6]  
MarkerEdgeColor = auto  
MarkerFaceColor = none  
XData = [ (1 by 100) double array]  
YData = [ (1 by 100) double array]  
ZData = []
```

Операции над графическими объектами. **get**

get (h)

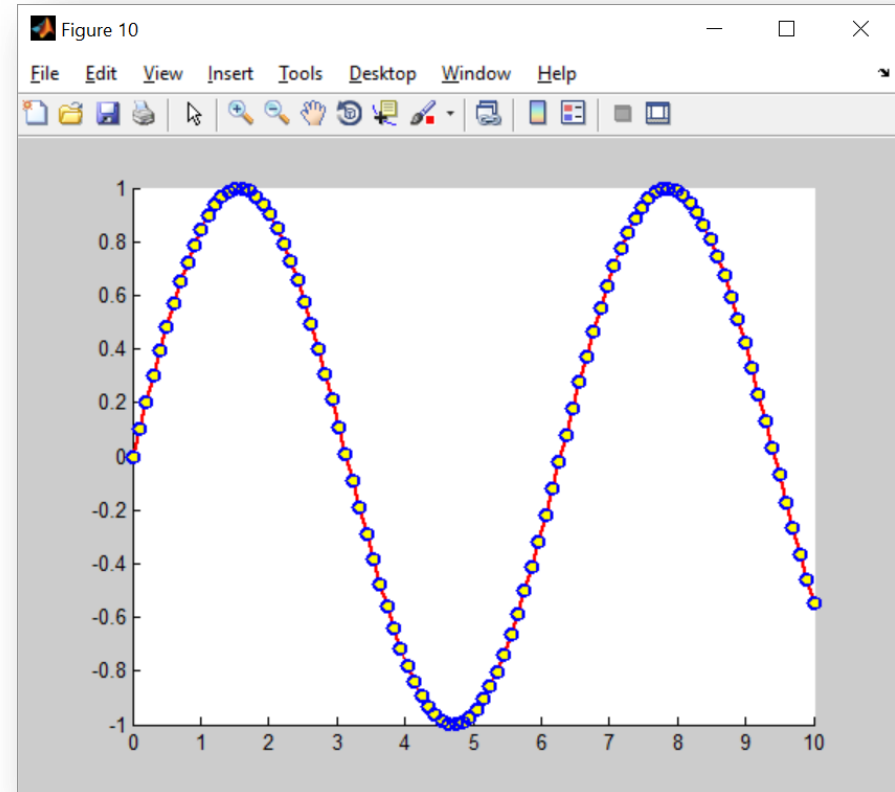


```
BeingDeleted = off
ButtonDownFcn =
Children = []
Clipping = on
CreateFcn =
DeleteFcn =
BusyAction = queue
HandleVisibility = on
HitTest = on
Interruptible = on
Parent = [174.005]
Selected = off
SelectionHighlight = on
Tag =
Type = line
UIContextMenu = []
UserData = []
Visible = on
```

Изменение свойств объекта. Команда **set**

```
clc
clear
figure(10)
x = linspace(0,10);
y = sin(x);
h=line(x,y)
get(h)

set(h, 'Color', 'red')
set(h, 'LineWidth', 2)
set(h, 'Marker', 'o')
set(h, 'MarkerSize', 6)
set(h, 'MarkerEdgeColor', 'b')
set(h, 'MarkerFaceColor', 'y')
```

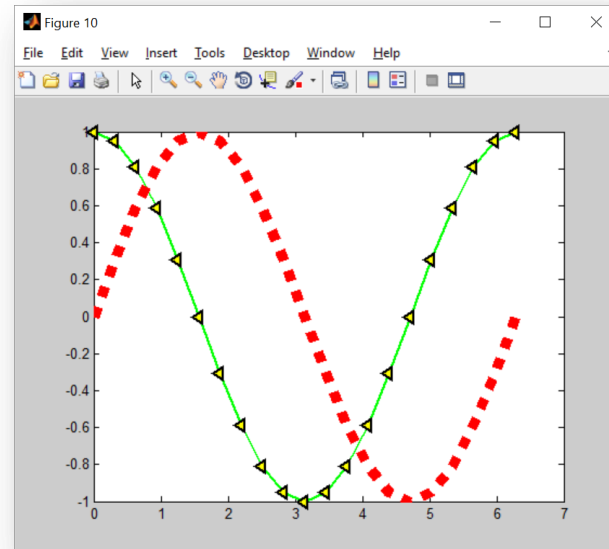


Изменение свойств объекта. Команда **set**

```
x = 0:pi/10:2*pi;  
f = cos(x);  
g = sin(x);  
h = plot(x,f,x,g);
```

```
set(h(1), 'Color', 'g',  
         'LineWidth', 2,  
         'Marker', '<',  
         'MarkerSize', 8,  
         'MarkerEdgeColor', 'k',  
         'MarkerFaceColor', 'y')
```

```
set(h(2), 'Color', 'r', 'LineWidth', 8, 'LineStyle', ':')
```



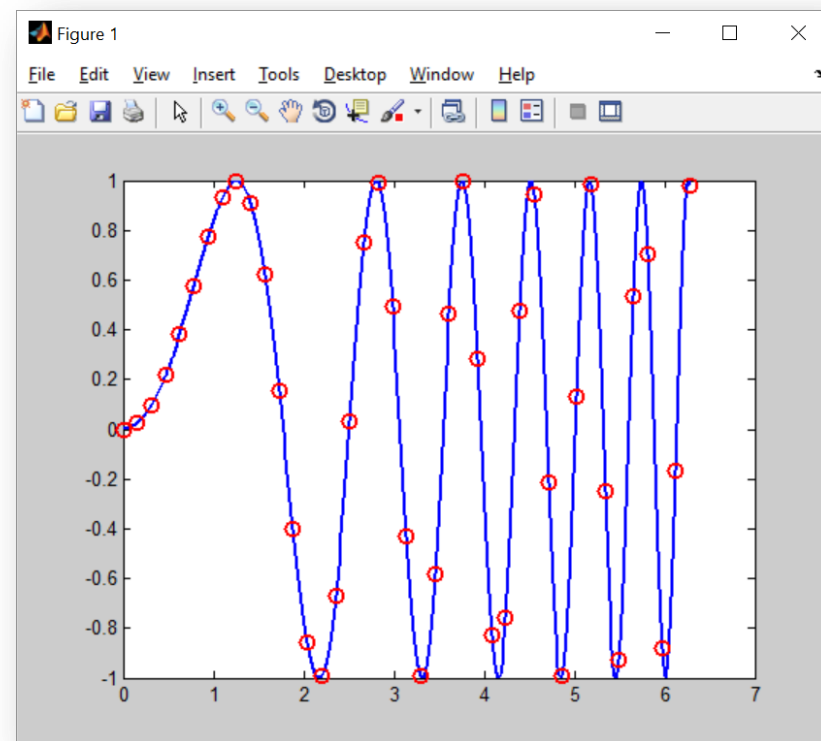
Изменение свойств объекта. Команда **set**

```
x = 0:pi/200:2*pi;  
y = sin(x.^2);
```

```
h = plot(x, y, x(1:10:end), y(1:10:end))
```

```
set(h,  
    'LineWidth', 2,  
    'Color', 'b')
```

```
set(h(2),  
    'LineStyle', 'none',  
    'Marker', 'o',  
    'MarkerSize', 8,  
    'MarkerEdgeColor', 'r')
```



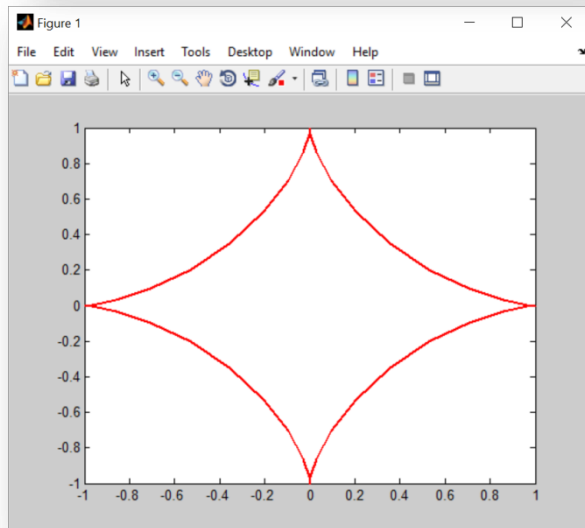
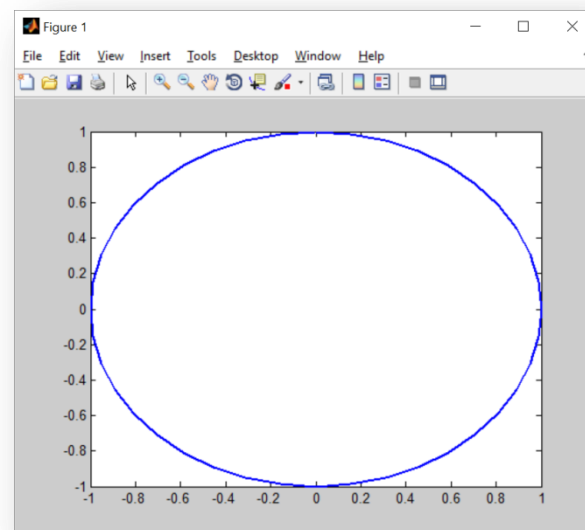
Изменение свойств объекта. Команда **set**

```
t = 0:pi/20:2*pi;  
x = sin(t);  
y = cos(t);
```

```
h = plot(x, y);  
set(h, 'LineWidth', 2)
```

```
pause(2);
```

```
x = sin(t).^3;  
y = cos(t).^3;  
set(h,  
    'XData', x,  
    'YData', y,  
    'Color', 'r')
```

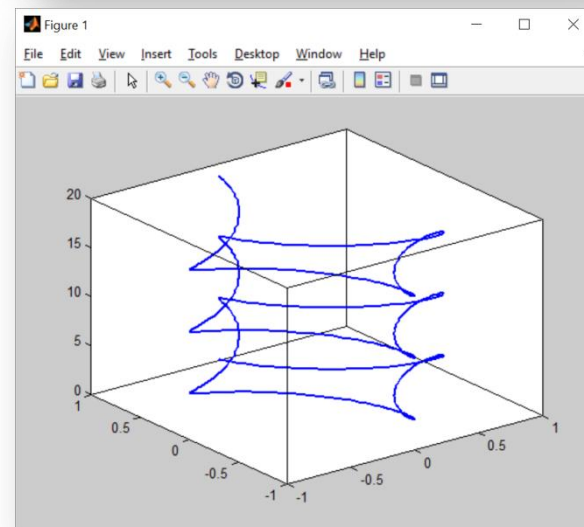
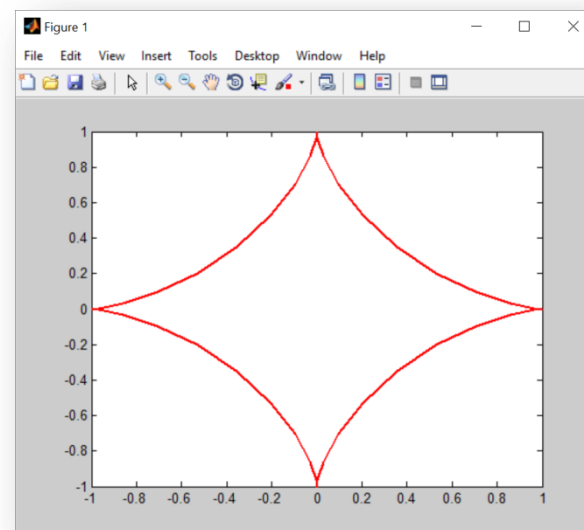


Изменение свойств объекта. Команда **set**

```
t = 0:pi/80:6*pi;  
x = sin(t).^3;  
y = cos(t).^3;  
  
h = plot(x, y,  
         'Color', 'r',  
         'LineWidth',2);
```

```
pause(2);
```

```
z = t;  
set(h,  
     'ZData', z,  
     'Color', 'b')  
view(3)
```

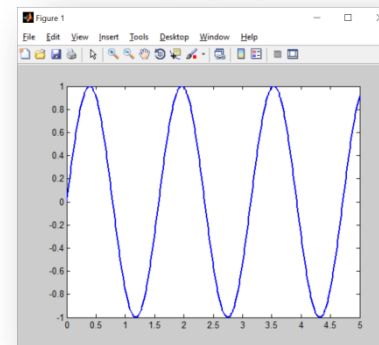
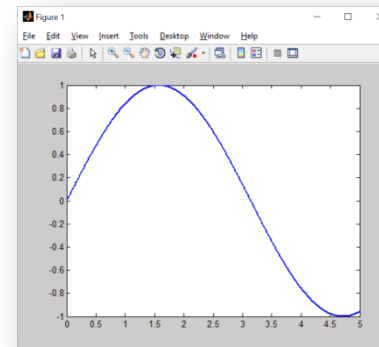


Изменение свойств объекта. Команда **set**

```
x = 0:0.01:5;  
y = sin(x);  
h = plot(x, y,  
         'XDataSource', 'x',  
         'YDataSource', 'y',  
         'LineWidth', 2)'
```

```
pause(0.5)
```

```
for k=1:7  
    y = sin(k*x);  
    refreshdata(h)  
    pause(0.5)  
end
```



XDataSource, YDataSource, ZDataSource - ассоциирование данных (для построения линии) с переменными.

После изменения значения переменных можно обновить график при помощи функции refreshdata.

Изменение свойств объекта. Команда **set**

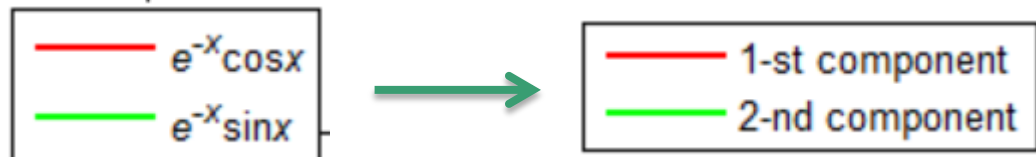
```
x = 0:pi/30:2*pi;  
f = exp(-x).*cos(x);  
g = exp(-x).*sin(x);
```

```
h1 = plot(x, f,  
          'DisplayName', '{\ite}^{-\itx}cos{\itx}',  
          'Color', 'r', 'LineWidth', 2)
```

```
hold on
```

```
h2 = plot(x, g,  
          'DisplayName', '{\ite}^{-\itx}sin{\itx}',  
          'Color', 'g', 'LineWidth', 2)
```

```
legend('show')
```



```
pause(2)
```

```
set(h1, 'DisplayName', '1-st component')
```

```
set(h2, 'DisplayName', '2-nd component')
```

Операции над графическими объектами

✓ `gco`

возвращает дескриптор текущего графического объекта;

✓ `gcbo`

возвращает дескриптор объекта, чья функция в данный момент выполняется;

✓ `gcbf`

возвращает дескриптор окна, содержащего объект, функция которого в данный момент выполняется

Операции над графическими объектами

✓ `drawnow`

выполнить очередь задержанных графических команд;

✓ `findobj`

найти объекты с заданными свойствами;

✓ `copyobj`

скопировать объект и порожденные им объекты.

Примеры

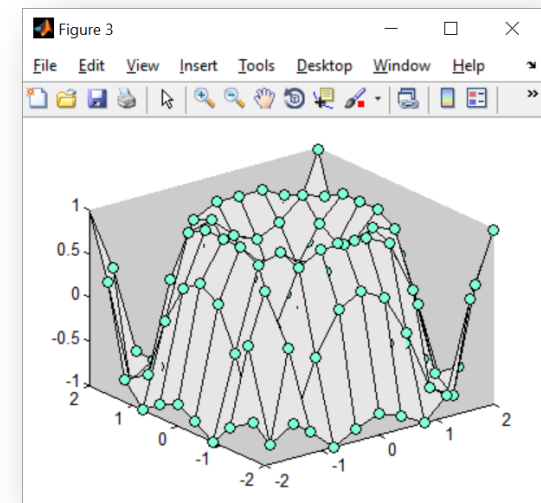
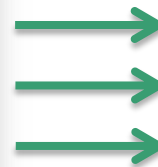
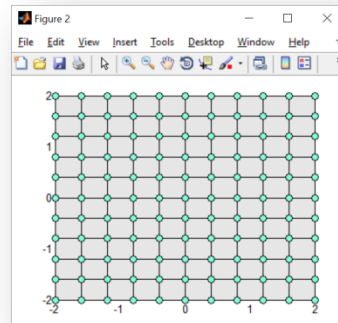
```
[x,y] = meshgrid([-2:.4:2]);  
Z =sin(x.^2+y.^2);
```

```
fh = figure('Position',[350 275 400 300],'Color','w');
```

```
ah = axes('Color',[.8 .8 .8],'XTick',[-2 -1 0 1 2],...  
         'YTick',[-2 -1 0 1 2]);
```

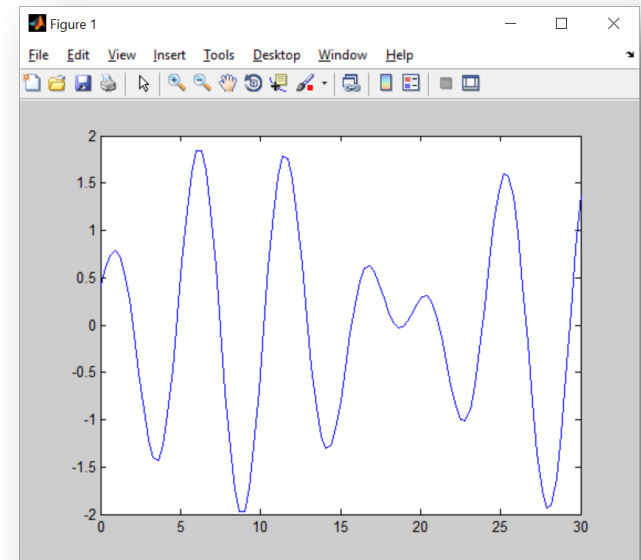
```
sh = surface('XData',x,'YData',y,'ZData',Z,...  
           'FaceColor',get(ah,'Color')+.1,...  
           'EdgeColor','k','Marker','o',...  
           'MarkerFaceColor',[.5 1 .85]);
```

`view(3)`



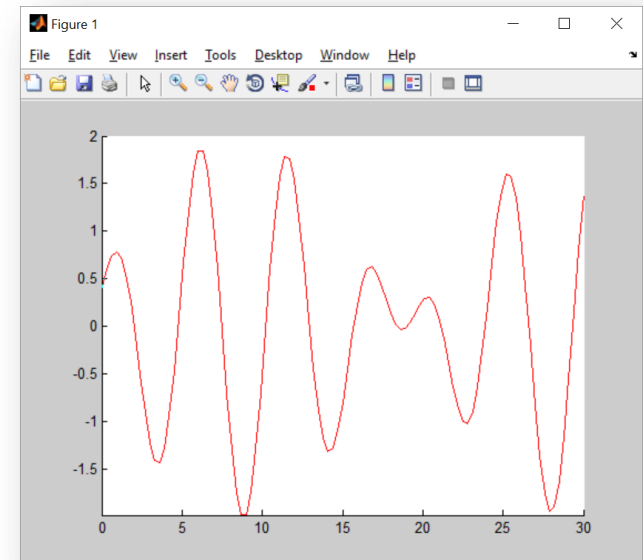
Пример. Анимация 1

```
clear
t=0:0.1:100;
x=0:0.3:30;
k=1.3;
w=0.9;
n=length(t);
y=cos(k*x-w*t(1))+cos(x-t(1));
figure
plot(x,y);
for i=2:n;
    y=cos(k*x-w*t(i))+cos(x-t(i));
    plot(x,y);
    drawnow
end;
```



Пример. Анимация 2

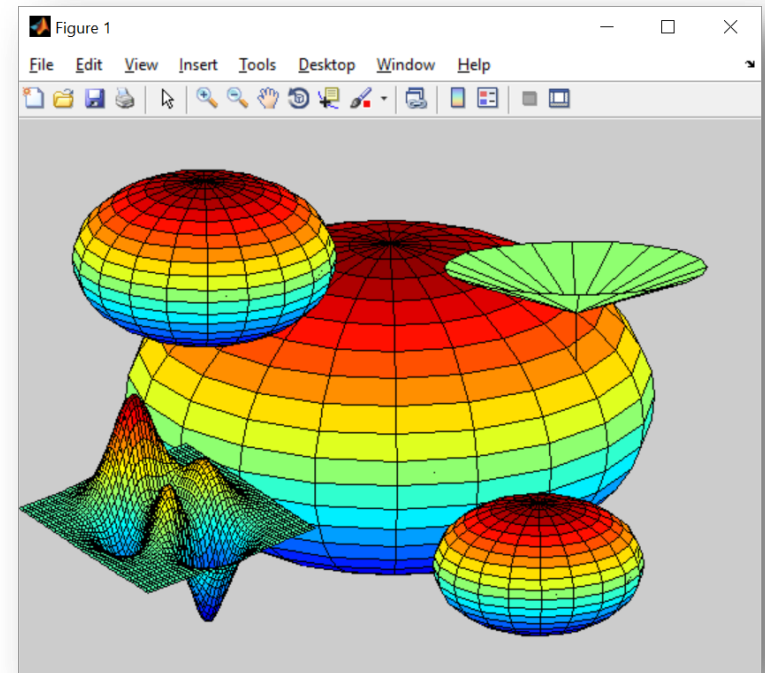
```
t=0:0.1:100;  
x=0:0.3:30;  
k=1.3;  
w=0.9;  
n=length(t);  
y=cos(k*x-w*t(1))+cos(x-t(1));  
axlim=[min(x) max(x) min(y) max(y)];  
figure  
lh=line(x,y);  
set(lh,'color','r');  
axis(axlim);  
set(lh,'erasemode','xor');  
for i=2:n  
    y=cos(k*x-w*t(i))+cos(x-t(i));  
    set(lh,'XData',x,'YData',y);  
    drawnow;  
end
```



Примеры

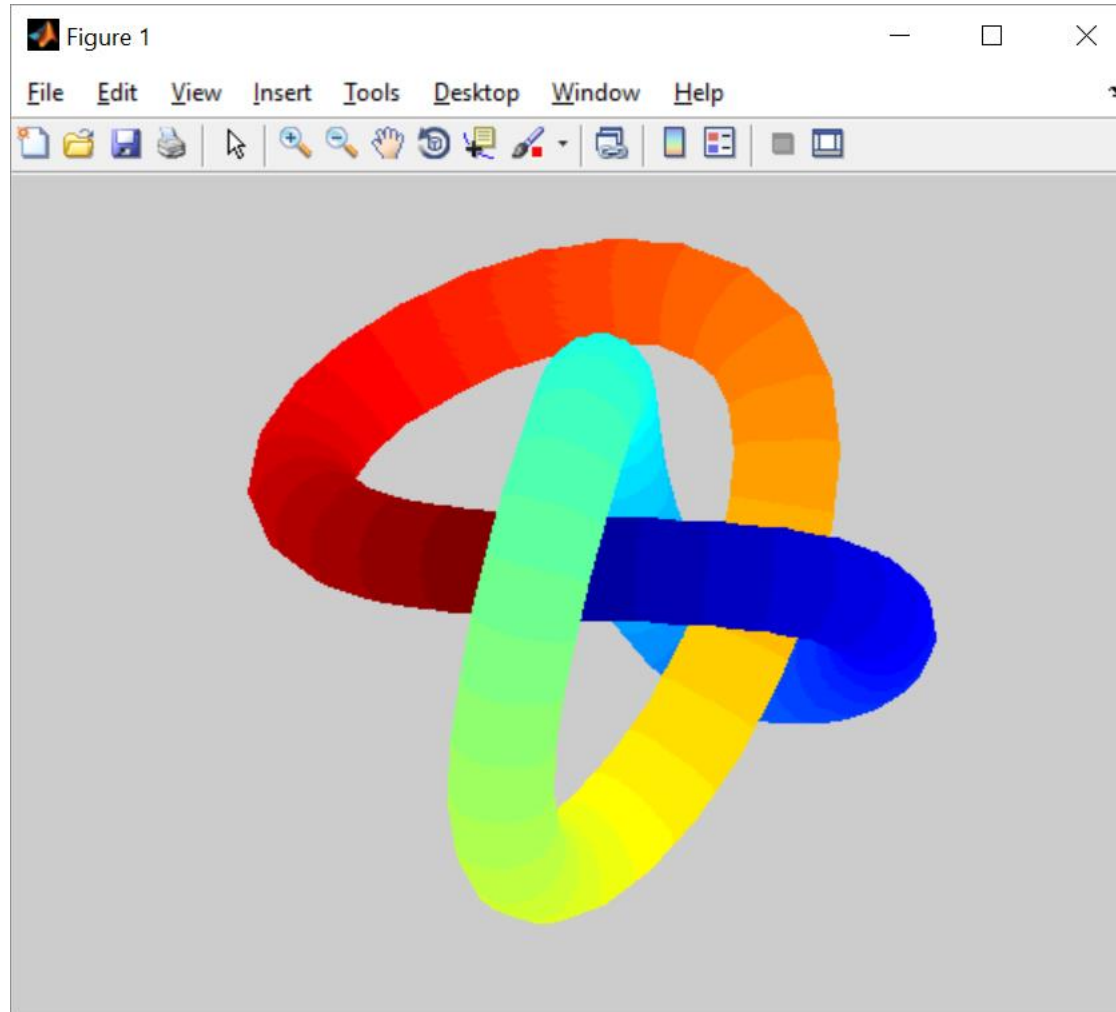
```
h(1) = axes('Position',[0 0 1 1]); sphere  
h(2) = axes('Position',[0 0 .4 .6]); peaks;  
h(3) = axes('Position',[0 .5 .5 .5]); sphere  
h(4) = axes('Position',[.5 0 .4 .4]); sphere  
h(5) = axes('Position',[.5 .5 .5 .3]);cylinder([0 0 0.5])
```

```
set(h, 'Visible', 'off')  
set(gcf, 'Renderer', 'painters')
```



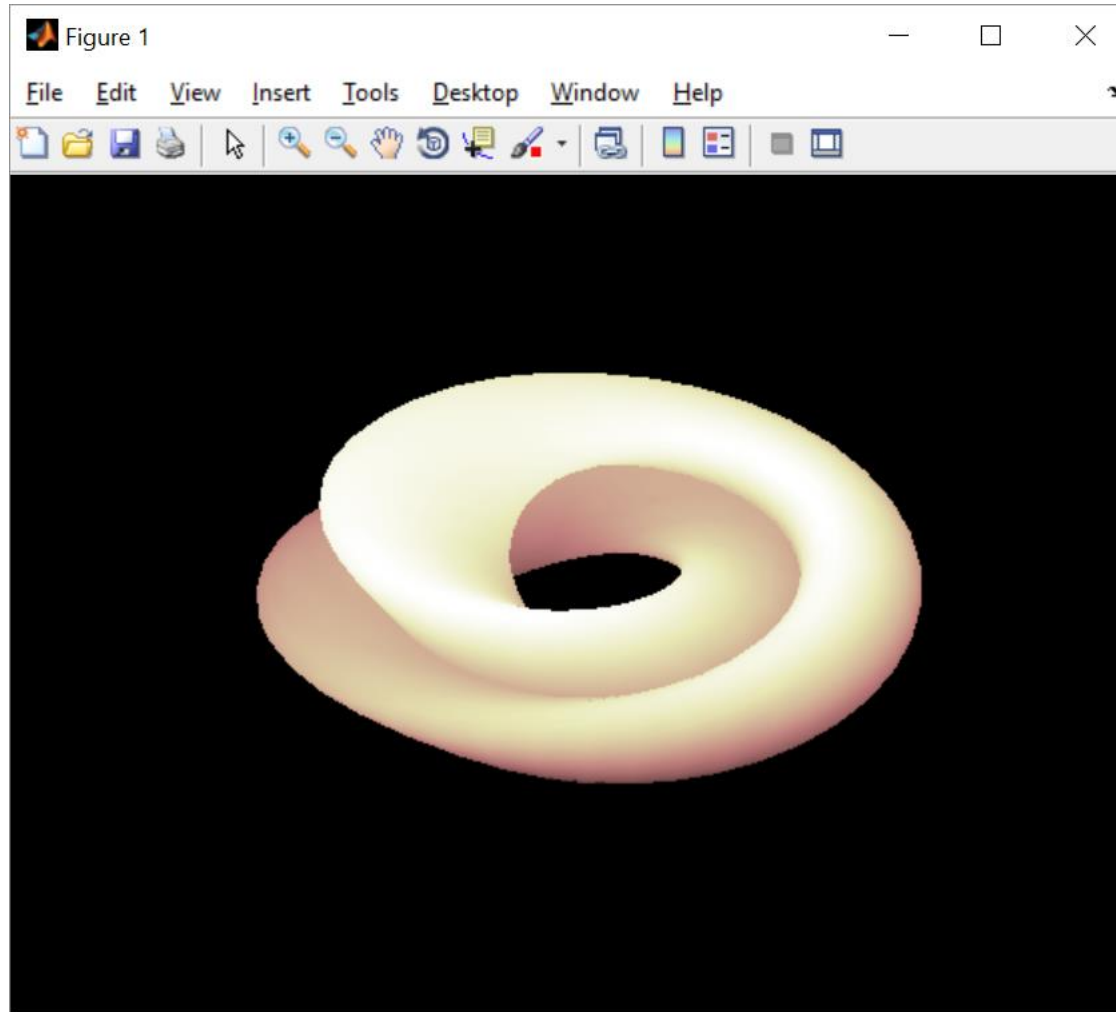
Галерея трехмерной графики. Завязанный узел

knot



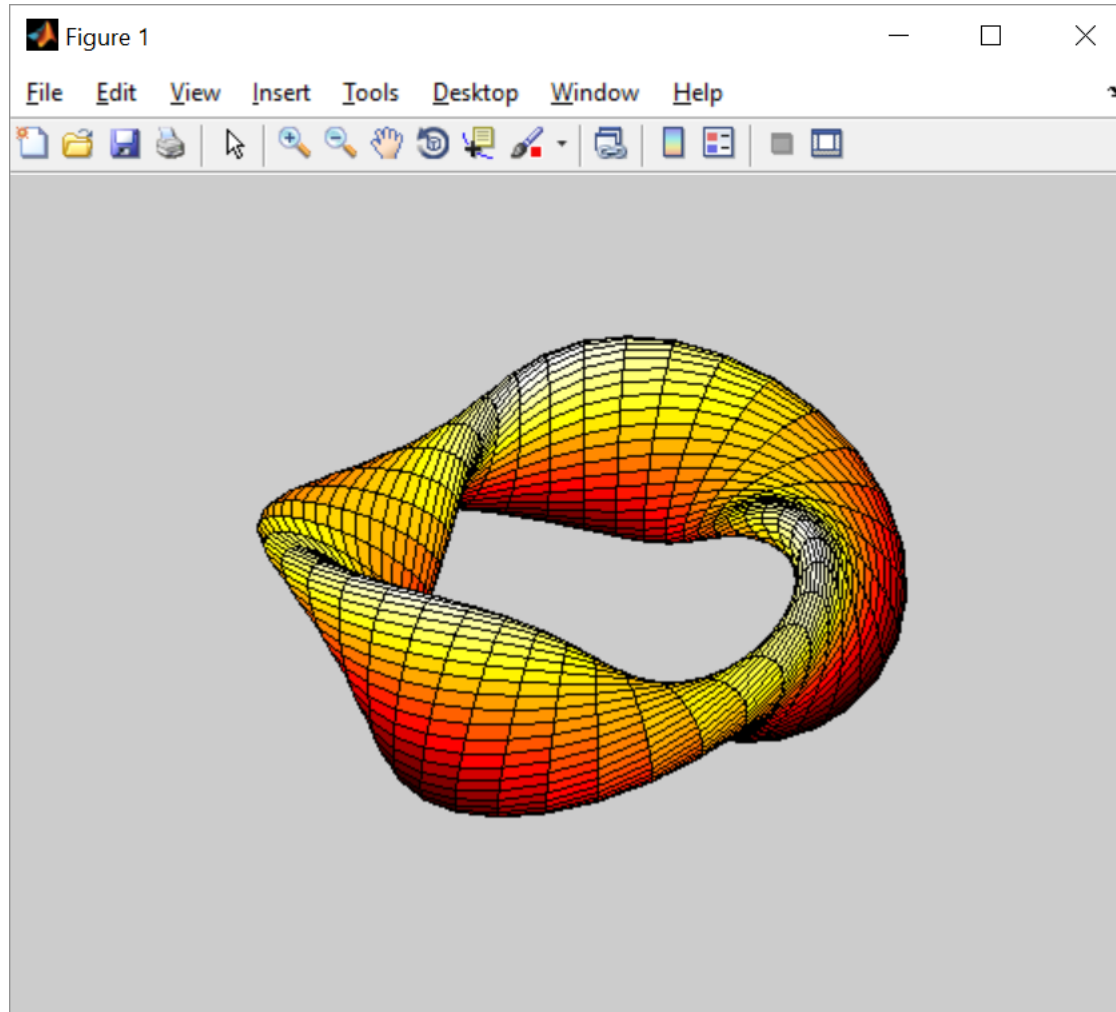
Галерея трехмерной графики. Объемное кольцо

`klein1`



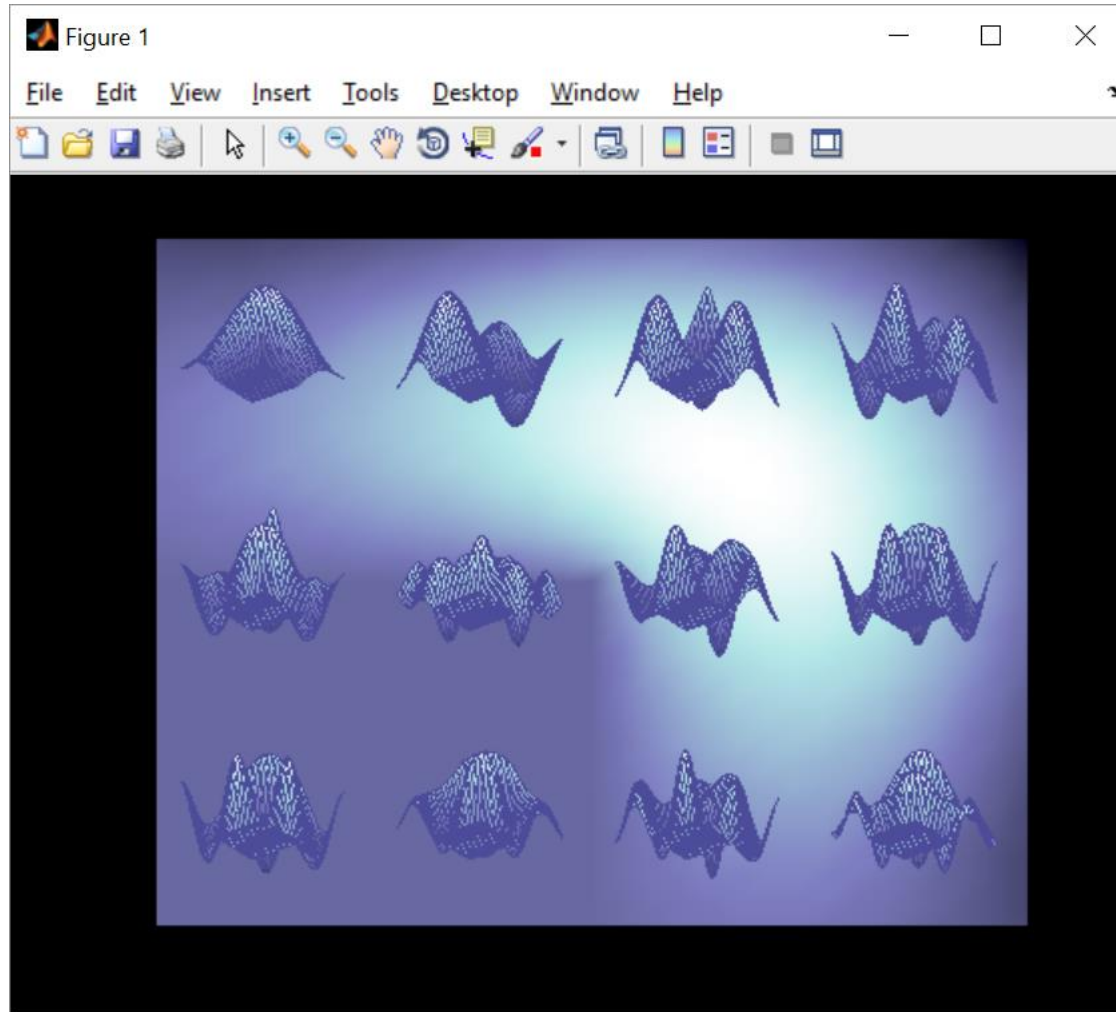
Галерея трехмерной графики. Объемное кольцо Мебиуса

kruller



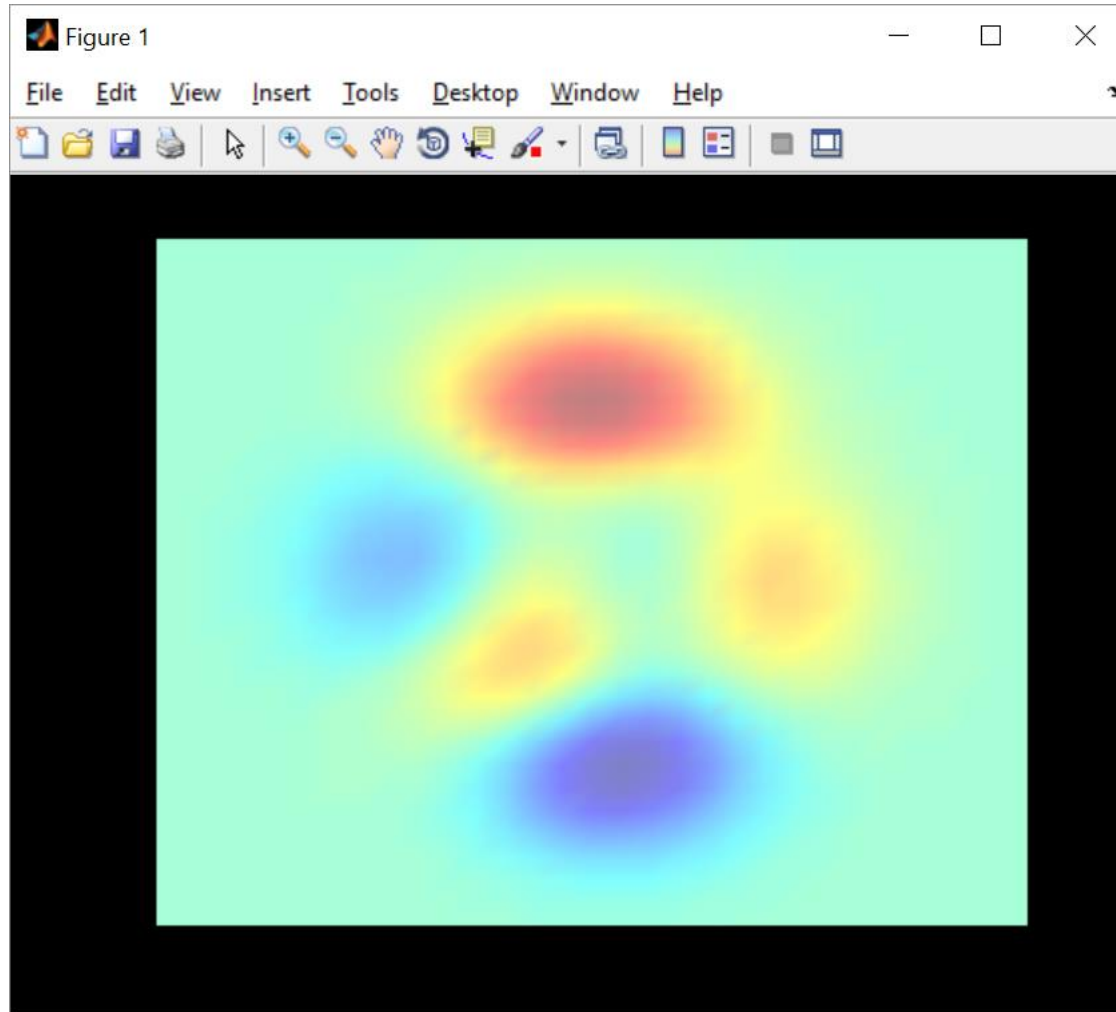
Галерея трехмерной графики. Фазы анимации

modes



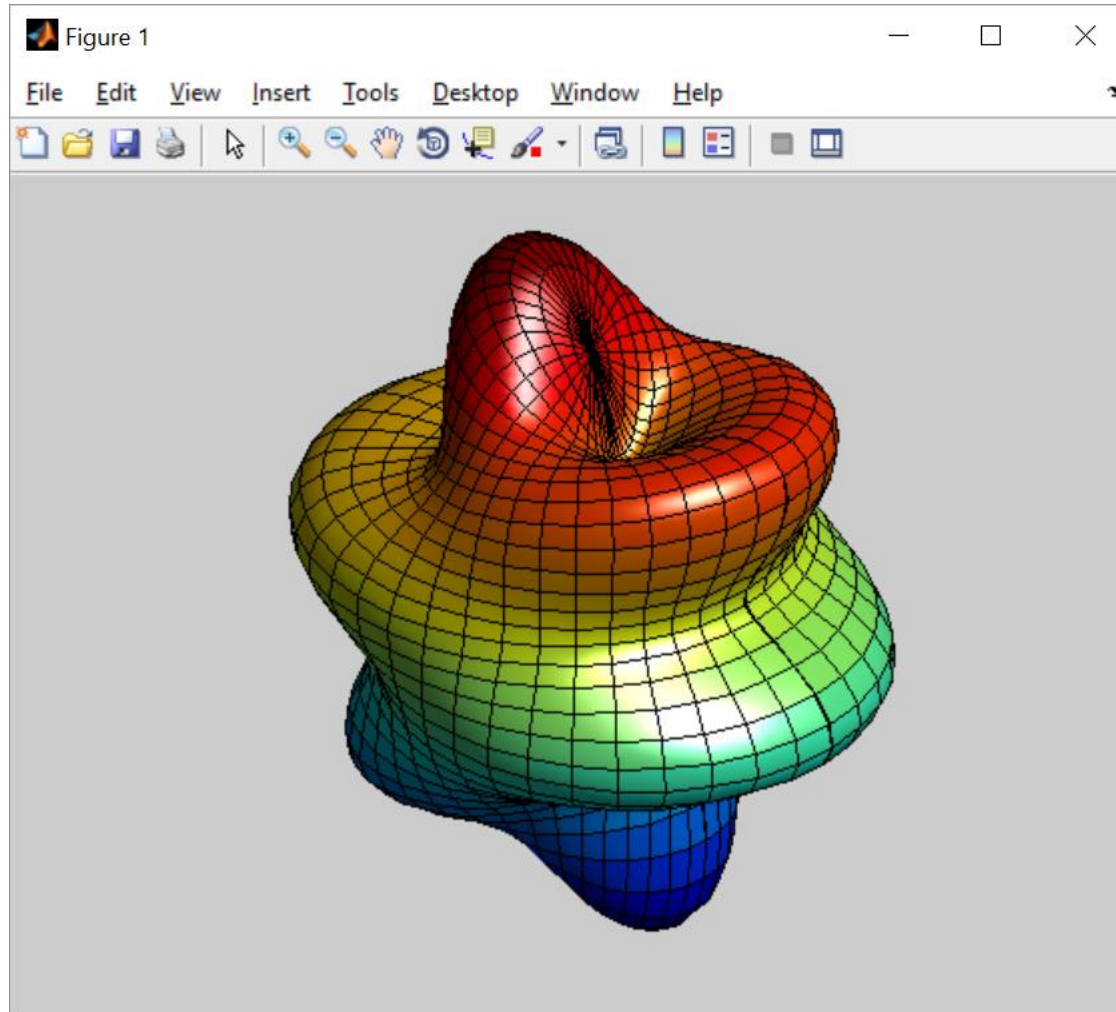
Галерея трехмерной графики. Векторное поле

`quivdemo`



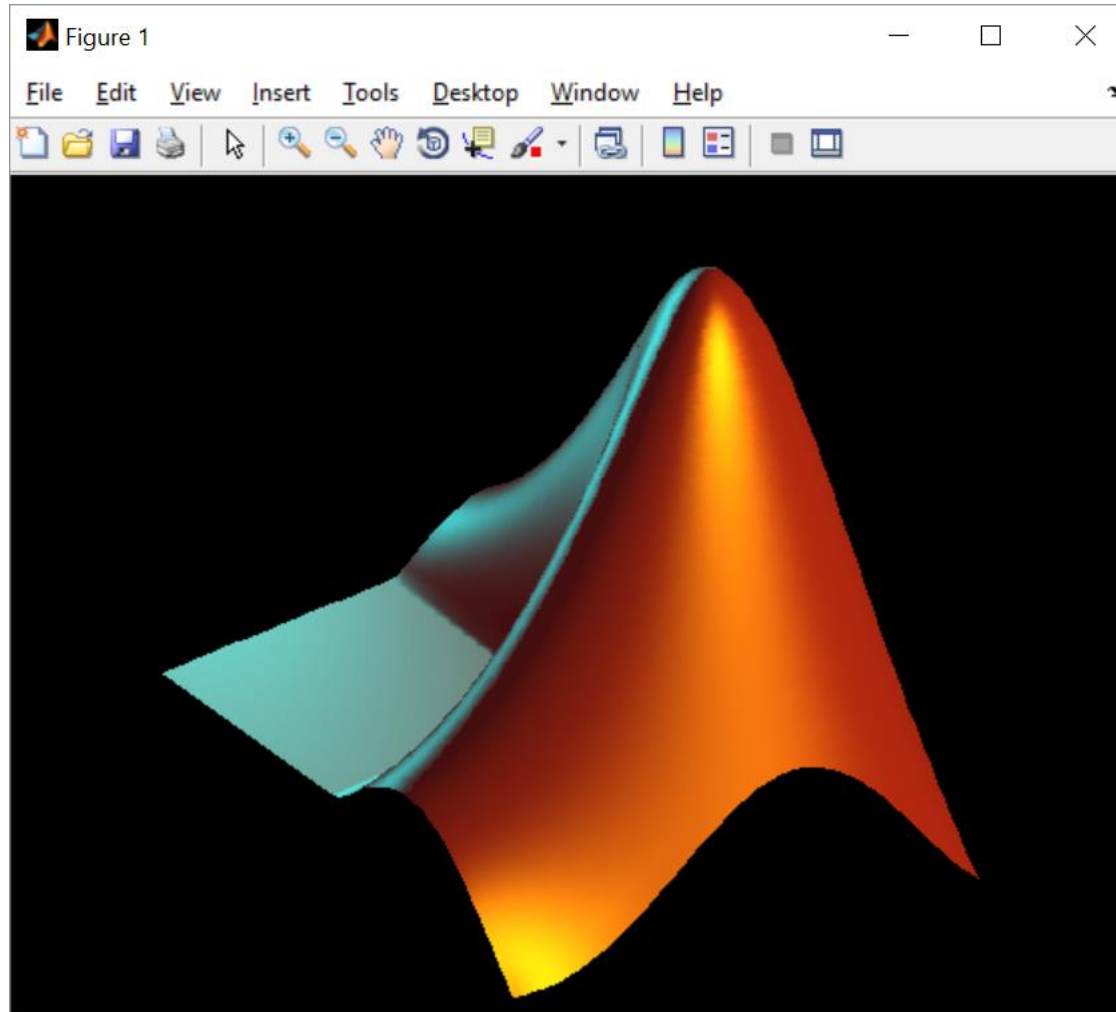
Галерея трехмерной графики. Улитка

spharm2



Галерея трехмерной графики. Логотип

Logo





Спасибо за внимание!