



Пакеты научных вычислений

Лекция 3. Графика в MatLab

Курбатова Наталья Викторовна, к.ф.-м.н.,
доцент кафедры математического
моделирования, мехмат, ЮФУ



Содержание:

Графические объекты (ГО) :

- системные и handle
- ГО как дескрипторы;
ГО как структуры

2D, 3D графика, заданная:

- строкой или символами
- параметрически
- неявно с помощью handle function

Графические примитивы:

- Line, Rectangle, *Surface*
- Text
- Image



Схема иерархии классов ГО

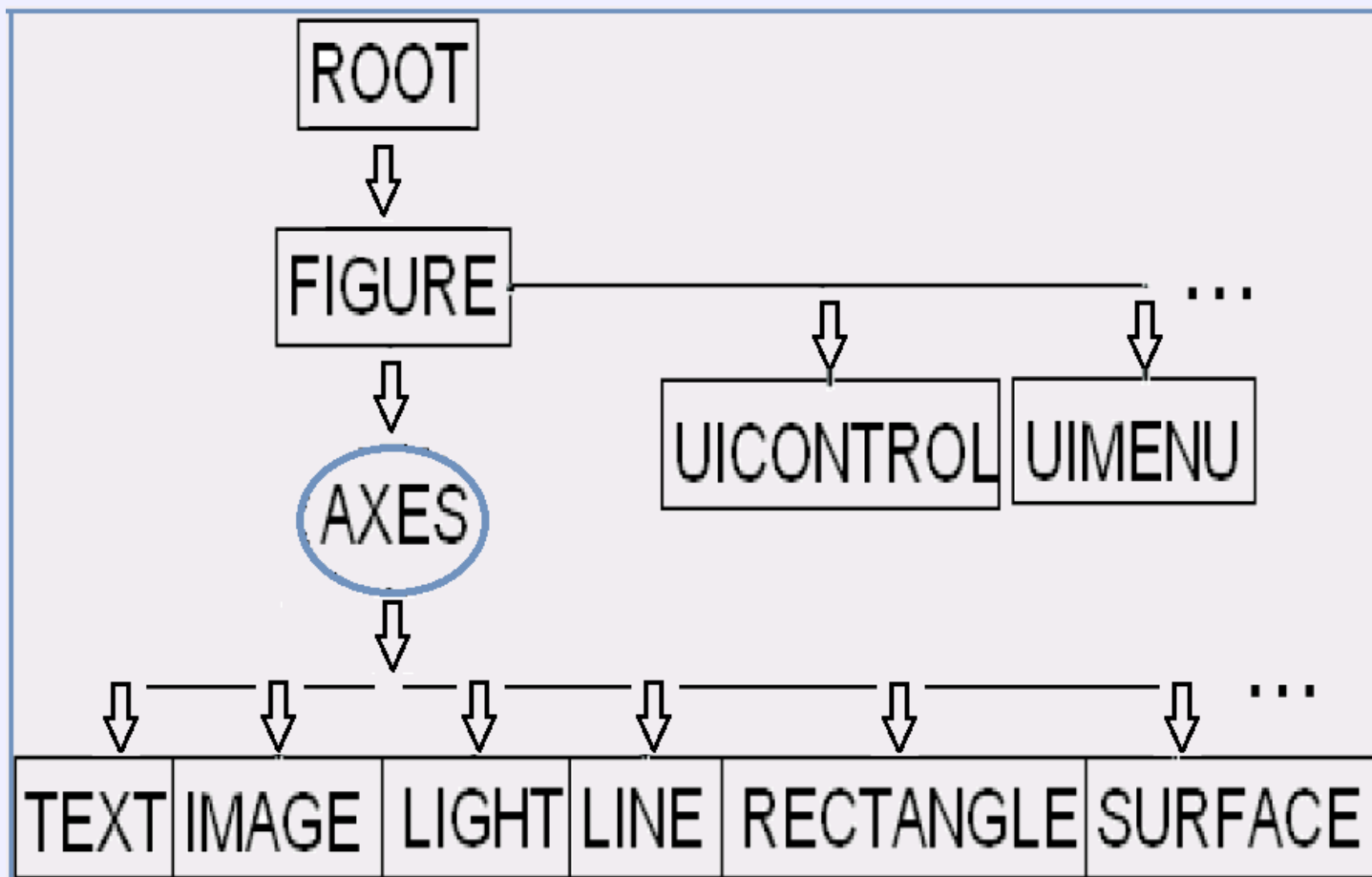




Схема иерархии классов ГО

Дескрипторы текущих графических объектов:

- **Root** ~ 0 ~ Command Window
- **Figure** ~ gcf ~ GraphicCurrentFigure (дескрипторы – натуральные числа, по умолчанию – окна Figure создаются последовательно)
- **Axes** ~ gca ~ GraphicCurrentAxes (пользовательские дескрипторы осей – **handle**)
- **gco** ~ GraphicCurrentObject

Определение возможных свойств:

set(0), set(gcf), set(gca), set(gco)

0, gcf, gca, gco – системные дескрипторы

Изменение свойств:

set(0,'PropertyName',value)

Получение свойств:

Value=get(0,'PropertyName')...



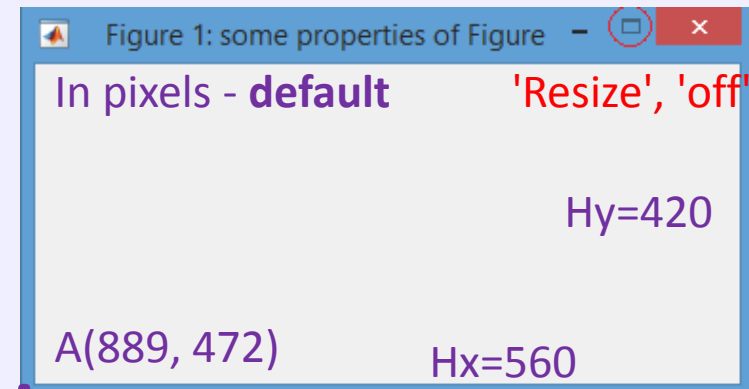
Важные свойства figure

```
>>set(gcf) % myfigure - 1
    MenuBar: {'none' 'figure'}
    Name: { }
    Resize: {'on' 'off'}
    NextPlot: {'new' 'add' 'replace' 'replacechildren'}
    Units: {'inches' 'centimeters' 'characters' 'normalized' 'points'
'pixels'}
```

```
>> set(1, 'MenuBar', 'none', 'Resize', 'off', 'NextPlot', 'add', 'Name', ...
    'some properties of Figure') % NextPlot', 'add'~ hold on
```

```
>>sizefigure=get(gcf,'position')
    sizefigure = 889 472 560 420
>> r=get(0, 'screensize')
    r = 1 1 1920 1080
```

```
>>figurecolor=get(gcf, 'color') %[R G B]
    figurecolor= 0.9400 0.9400 0.9400
```





Построение нескольких графиков в одном окне

I. Построение нескольких графиков одной функцией:

```
plot(x,g, lineSpec1,x,y, lineSpec2, x,f, ...) )
```

LineSpec - спецификатор линии; задаёт Color, linestyle, Marker (цвет, стиль линии, маркер) ; lineSpec \in Char (f.e., 'm--*')

II. Графики последовательно помещаются в Figure:

% **Example** Using char

```
clear
```

```
n=50; x=(linspace(-pi,pi,150))' % столбец
```

```
y='exp(sin(x))/3'; y=eval(y); % выполнили строку
```

```
r=plot(x,y,'r--') , hold on % 1-й способ
```

```
g=abs(sin(x)); t=plot(x,g, 'c:'), % 2-й способ
```

```
set([r,t],'linewidth',2);
```

```
M=[sin(2*x), cos(x)]; % M – матрица из колонок (?)
```

```
p=plot(x,M), grid on % 3-й способ
```

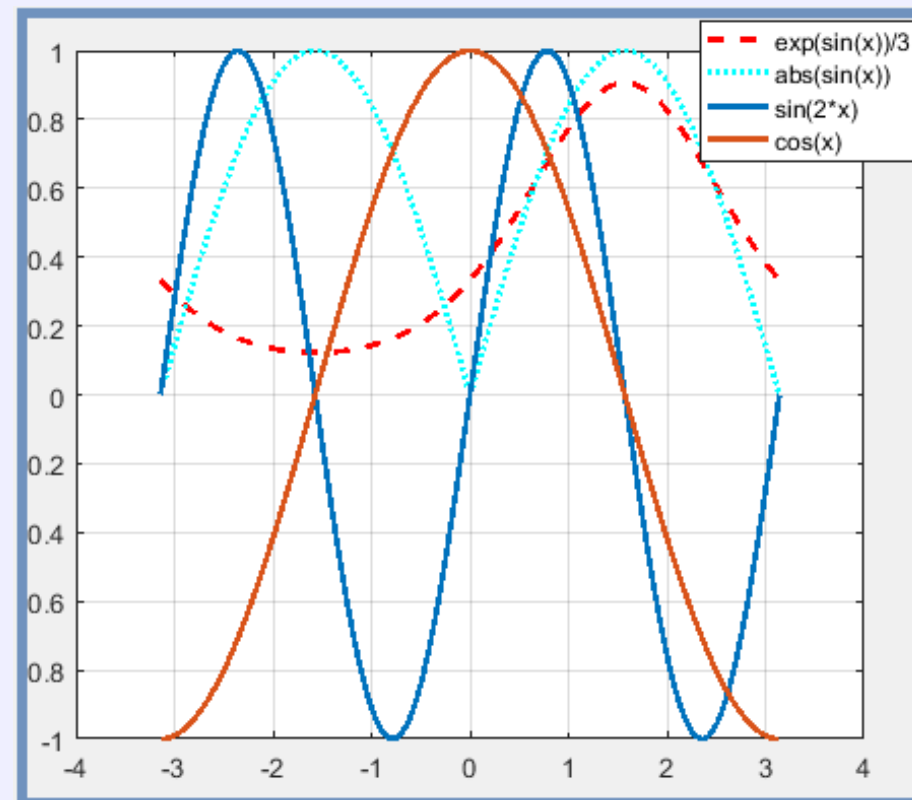
```
set(p,'linewidth',2);
```

```
legend('exp(sin(x))/3','abs(sin(x))',...
```

```
'sin(2*x)', 'cos(x)')
```

Объясните!

```
gr3=plot(x,y,'r-',x,g,'b-.',x,f,'m:');
```

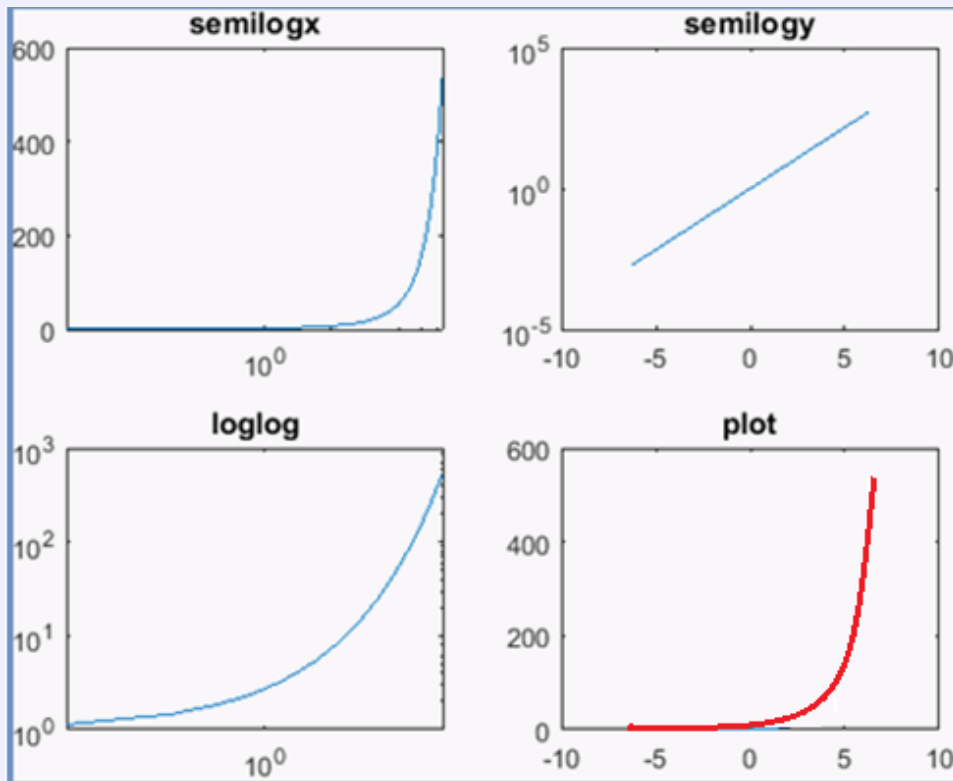


Example



Графические функции, учитывающие рост (x,y...)

Логарифмическая шкала для единственной оси или всех:



Помним, что свободны от регистра поля функции set и get:

```
>>set(hll, 'color ', 'red ')
```

```
clear
n=50
x=linspace(-2*pi,2*pi,50)
% Logarithmic scale for single or for all axes:
figure % обязательно?
y=exp(x);

% Question: Compare the three functions,
% choose the best one (for this example),
% Title of each window is the
% plotting function

subplot(2,2,1), semilogx(x,y), title('semilogx')
subplot(2,2,2), semilogy(x,y), title('semilogy')
subplot(2,2,3), loglog(x,y), title('loglog'),
subplot(2,2,4), hll=plot(x,y), hll.Color= 'red ',
```

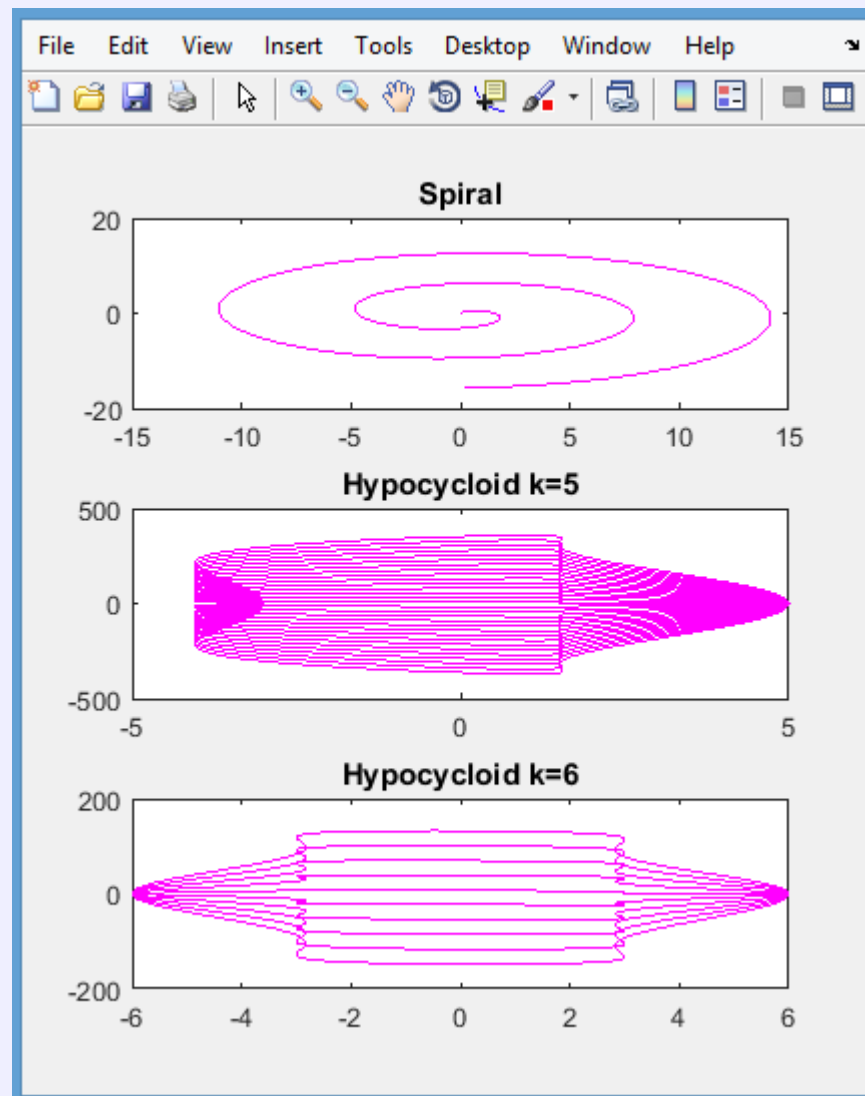


Графики функций, заданных параметрически

```
subplot(3,1,1); t=0:0.01:5*pi;  
x=t.*sin(t); y=t.*cos(t);  
plot(x,y,'m-'), title('Spiral')
```

```
subplot(3,1,2); t=0:0.01:30*pi;  
k=5; r=1;  
x = (k-1)*r*cos(t) + r*cos((k-1)*t);  
y = (k-1)*t.*sin(t) - r*sin((k-1)*t);  
plot(x,y,'m-'), title('Hypocycloid k=5')
```

```
subplot(3,1,3); t=0:0.01:10*pi;  
k=6; r=1;  
x = (k-1)*r*cos(t) + r*cos((k-1)*t);  
y = (k-1)*t.*sin(t) - r*sin((k-1)*t);  
plot(x,y,'m-'), title('Hypocycloid k=6')
```





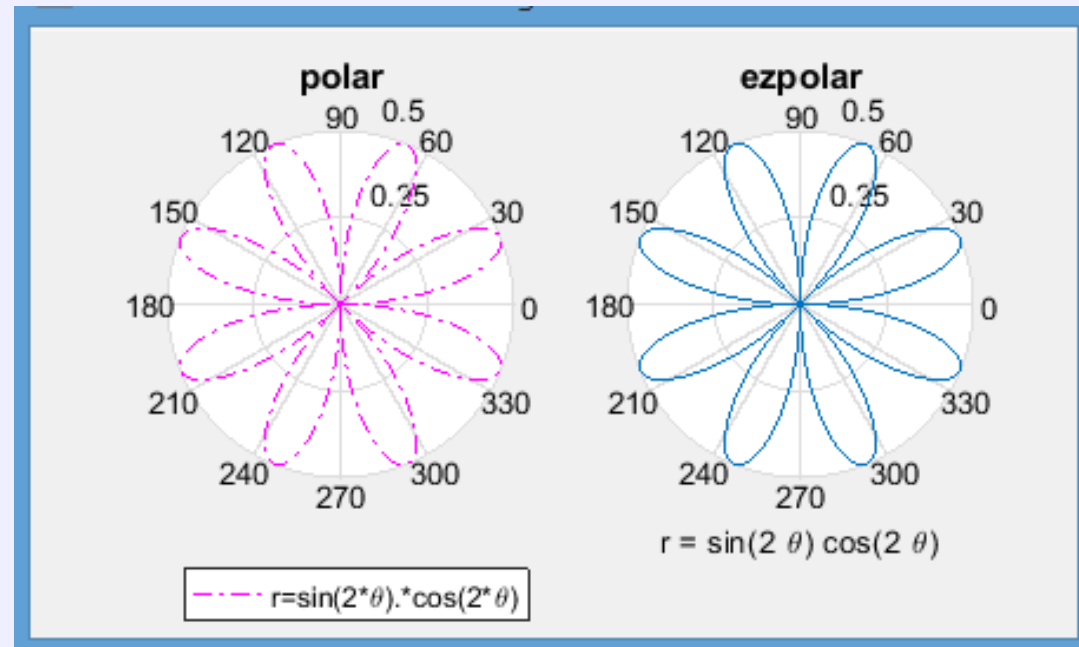
Графики функций, заданных в полярных координатах

POLAR , EZPOLAR

polar(THETA, RHO, S)

```
figure  
theta = 0:0.01:2*pi;  
rho = sin(2*theta).*cos(2*theta);  
  
subplot(1,2,1), polar(theta,rho, 'm-.'),  
legend('sin(2*\theta).*cos(2*\theta)')  
title('polar')  
  
subplot(1,2,2),  
ezpolar('sin(2*theta).*cos(2*theta)'),  
title('ezpolar')
```

polar – для новых версий ML!!!
pollarplot – устарела!



Аргументы EZPOLAR – строки или
символьные переменные, а не
векторы класса Double!



Примитив: Line. 3D Line~plot3

title(...) - общий заголовок

sgtitle(...) - заголовок для окна subplot

figure

title('3D plot')

sgtitle('3D and XOY - projection')

X=sort(3*rand(1,1000));

Y=sort(2*rand(1,1000));

subplot(2,1,1)

h=line(X,Y,'linewidth',2); grid on

Z=sin(X).*exp(Y); ? **Объясните синтаксис!**

subplot(2,1,2)

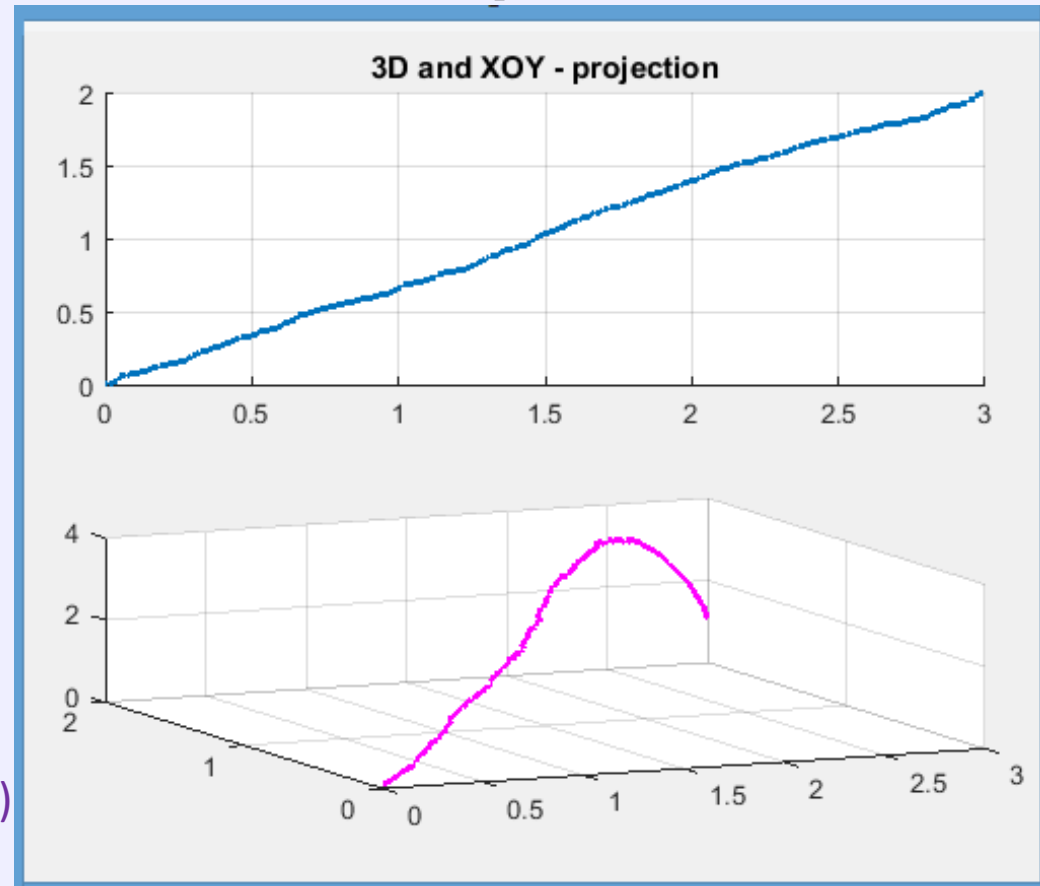
plot3(X,Y,Z,'linewidth',2,'color','magenta')

polyline 3d (try rotate)

grid on

plot3(X,Y,Z) % 3d - line (try rotate!)

3D plot





Примитив: Rectangle.

'Curvature' = {[0 0] – прямоугольник; [1 1] - эллипс [k,r] – скругленные вершины, $0 < k, r < 1$ }

- 1) figure
- 2) subplot(3,1,1)
- 3) rectangle('Curvature',[0 0],'Position',[100 100 100 400],...
'EdgeColor', 'blue', 'linewidth', 3) %creates a rectangle
- 4) subplot(3,1,2)
- 5) rectangle('Curvature',[1 1],'Position',[50 50 100 100],...
- 6) 'facecolor', 'red') %creates an ellipse, circle
- 7) subplot(3,1,3)
- 8) g=rectangle('Curvature',[.3 .3],'Position',[900 200 200 300],...
- 9) 'facecolor',[0.2 0.5 1]) % creates a rectangle with rounded corners
- 10) get(g) % Изучите свойства примитива!

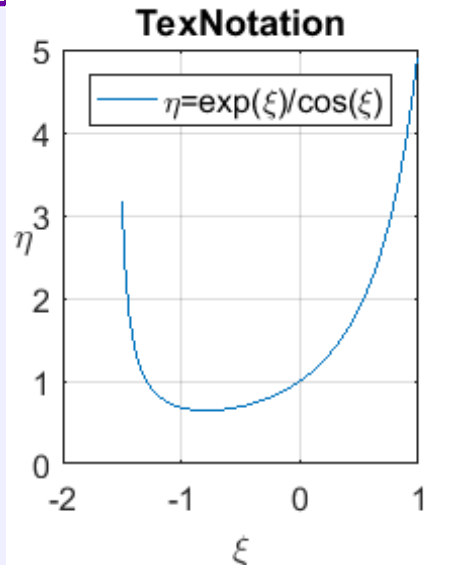
rectangle() %создаёт квадрат единичной длины

**axis square % вставьте эту команду после 6-й строки, уйдет вытянутость;
Коэффициент вытянутости экрана: $r(4)/r(3)=H_y/H_x$, см.слайд 5**

Идём смотреть? 11



Использование нотации LaTeX. Функция заполнения цветом - Fill



1) $x=-1.5:0.03:1$; $y=\exp(x)/\cos(x)$ % x-radian

2) `plot(x,y);`

%% Дизайн графика:

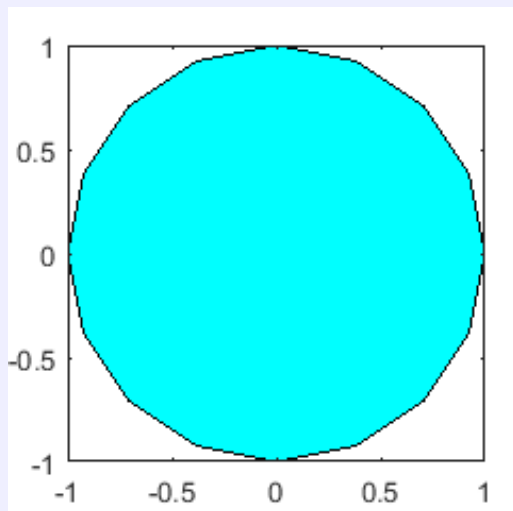
2) `lg=legend('\eta=\exp(\xi)/\cos(\xi)'), set(lg,'fontsize',12);`

3) `t=title('TexNotation');`

4) `xi=xlabel('\xi','fontname','latex')`

5) `eta=ylabel('\eta','fontname','latex');` % `set(eta,'rotation',90)`

6) `set(gca,'fontsize',12)` % **ВСЕ** Надписи на осях 12pt



1) `t = (1/16:1/16:1)*2*pi; x = cos(t); y = sin(t);`

%% Create a closed figure and fill by cyan function:

2) `fill(x,y, 'c')` % закрашивает замкн.линию цветом

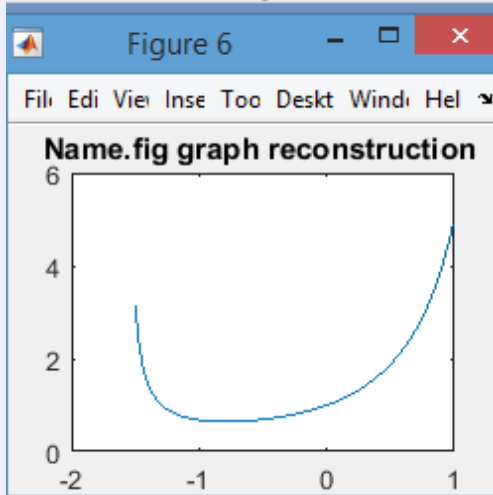
3) `axis square` % **убирается эффект вытянутости экрана**

**Что будет, если изменим шаг и начало кривой:
 $1/16 \rightarrow 1/100$?**



Импортирование hload! *.fig – источник данных!

До



После

- 1) `hfig = hgload('Name.fig') % load Name.fig in Folder`
- 2) `figure(hfig); % визуализация графика`
- 3) `haxe=get(hfig,'Children') % Графики в осях!`
`%% Тут haxe содержит одну line – но аналогично и для большего количества линий`
- 5) `ChAxes=get(haxe,'Children') % ищем line!`
- 6) `x=get(ChAxes(2),'XData'); % X-coordinates of line`
- 7) `y=get(ChAxes(2),'YData'); % Y-coordinates of line`
- 8) `figure, plot(x,y) % check line!`
- 9) `title('name.fig graph reconstruction')`

CHaxes =

2×1 cell array

[0×0 GraphicsPlaceholder]

[1×1 Line]



fplot(Fun,Limits,lineSpec) -

nvkurbatova@sfedu.ru

Fun – anonymous function, имеет тип *function_handle* содержит один исполняемый оператор

Limits – интервал области задания функции

lineSpec – спецификатор линии; задаёт *Color, LineStyle, Marker*-цвет, стиль линии, маркер ; *lineSpec* \in Char (f.e., 'm--*') – **повторяем!**

Пример:

```
subplot(1,2,2)
```

```
f = @(x,n) exp(sin(n*x))*cos(x);
```

```
fplot(@(x)f(x,10),[0 2*pi],'r','linewidth',1.5)
```

Warning: Function fails on array inputs. Use element-wise operators to increase speed

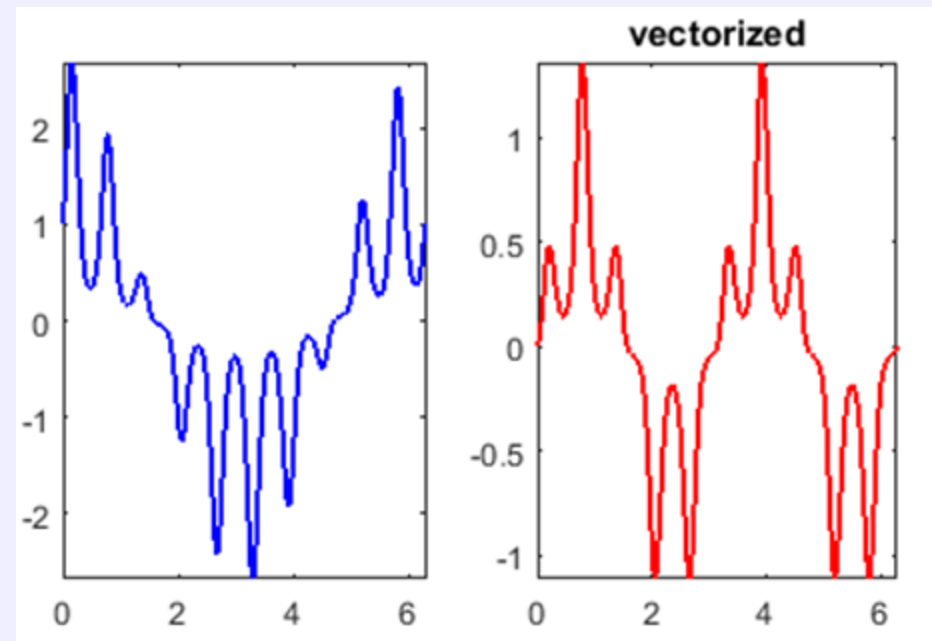
%% Используем векторные операции:

```
subplot(1,2,1)
```

```
f = @(x,n) exp(sin(n*x)).*cos(x).*sin(x);
```

```
fplot(@(x)f(x,10),[0 2*pi],'r','linewidth',1.5)
```

```
title('vectorized')
```





ezplot(Fun,Limits,lineSpec) fimplicit ~ implicitplot (anonymous,...)

%% ezplot

figure

```
r=ezplot('x^2+(y-abs(x)^(1/2))^2=1'), hold on, grid on
set(r,'linewidth',3,'color','red'); title(' ')
```

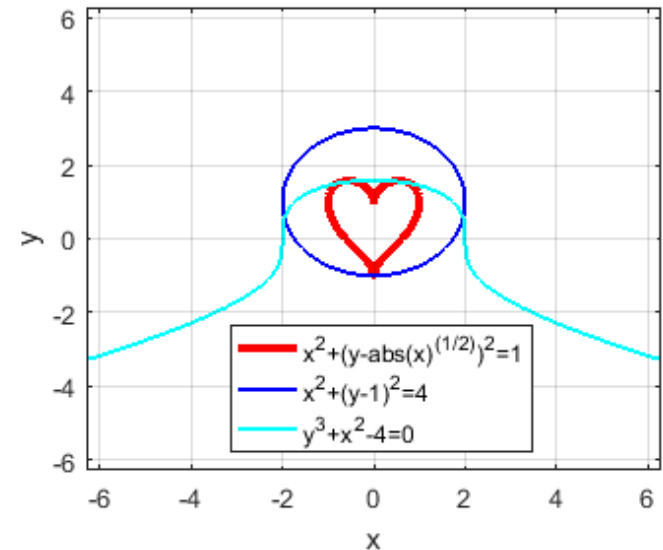
% автоматически в заголовок помещается строка первой функции с дескриптором r → title(' ')

```
rf=ezplot('x^2+(y-1)^2=4',[-3,3]); title(' ')
```

```
ra=ezplot(@(x,y) y^3+x^2-4);
```

```
set(ra,'linewidth',1.5,'color','cyan'); title(' ');
```

```
legend('x^2+(y-abs(x)^(1/2))^2=1',...
       'x^2+(y-1)^2=4','y^3+x^2-4=0')
```



%% implicitplot - (New!!!)

% For function $f(x,y) = 0$

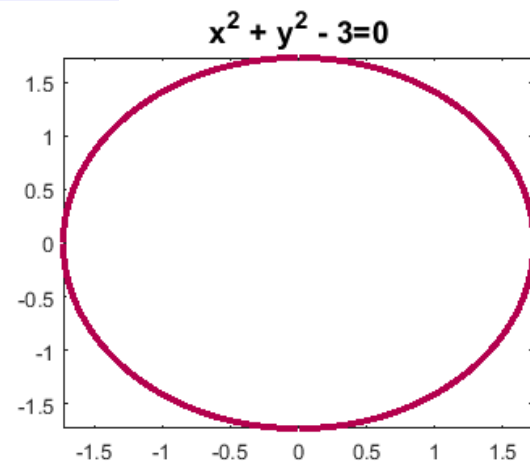
figure

```
fp = fimplicit(@(x,y) x.^2 + y.^2 - 3)
```

```
fp.LineWidth = 3 % Important
```

```
fp.Color=[0.7 0 0.3] % Register!
```

```
title('x^2 + y^2 - 3=0','fontsize',14)
```





Примитив Text: `text(x,y,characteristic,opts)`

%% I.TEXT . Построение экстремальных точек

```
x = linspace(-2,3); % 100 point - default  
y = 3*x.^3-6*x.^2; % vectorized!  
plot(x,y,'c-','linewidth',1.5) % простейший способ
```

%% Надпись о нулевой производной

% задаём, оценив **визуально**:

```
xt = [-0.2 1.5]; yt = [5 -7]; str = 'dy/dx = 0';  
text(xt,yt,str,'fontsize',12), hold on  
% т.к. добавим ещё что-нибудь!
```

%% II.TEXT. Условия для min и max:

```
% yim1<yi,yi>yip1 yim1>yi,yi<yip1 % yim1~ yi minus 1  
% yim1 ~ y{i-1}; yi~yi; yip1~y{i+1}
```

% вспомогательные массивы для поиска min и max:

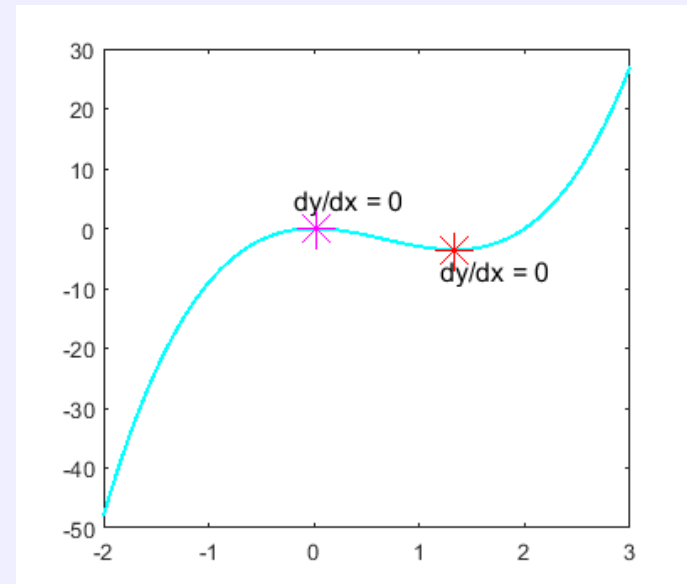
```
yi=y(2:end-1) ; yip1=y(3:end); yim1=y(1:end-2)  
% обозначения: yi=y{i}; yim1= y{i-1}; yip1=y{i+1}  
% в фигурных скобках индексы в нотации Latex, не для MatLab!
```

% III.TEXT – Найдём и обозначим точки экстремума:

```
trmax=and(yim1<yi,yi>yip1) %max condition  
indexmax=find(trmax)
```

```
trmin=and(yim1>yi,yi<yip1) %min condition  
indexmin=find(trmin) % у нас единственный min  
xmin=x(indexmin+1)% min  
ymin=y(indexmin+1); plot(xmin,ymin,'r*','markersize',16)
```

```
xmax=x(indexmax+1)% max  
ymax=y(indexmax+1); plot(xmax,ymax, 'm*','markersize',16)
```





Интерактивный текст. GTEXT

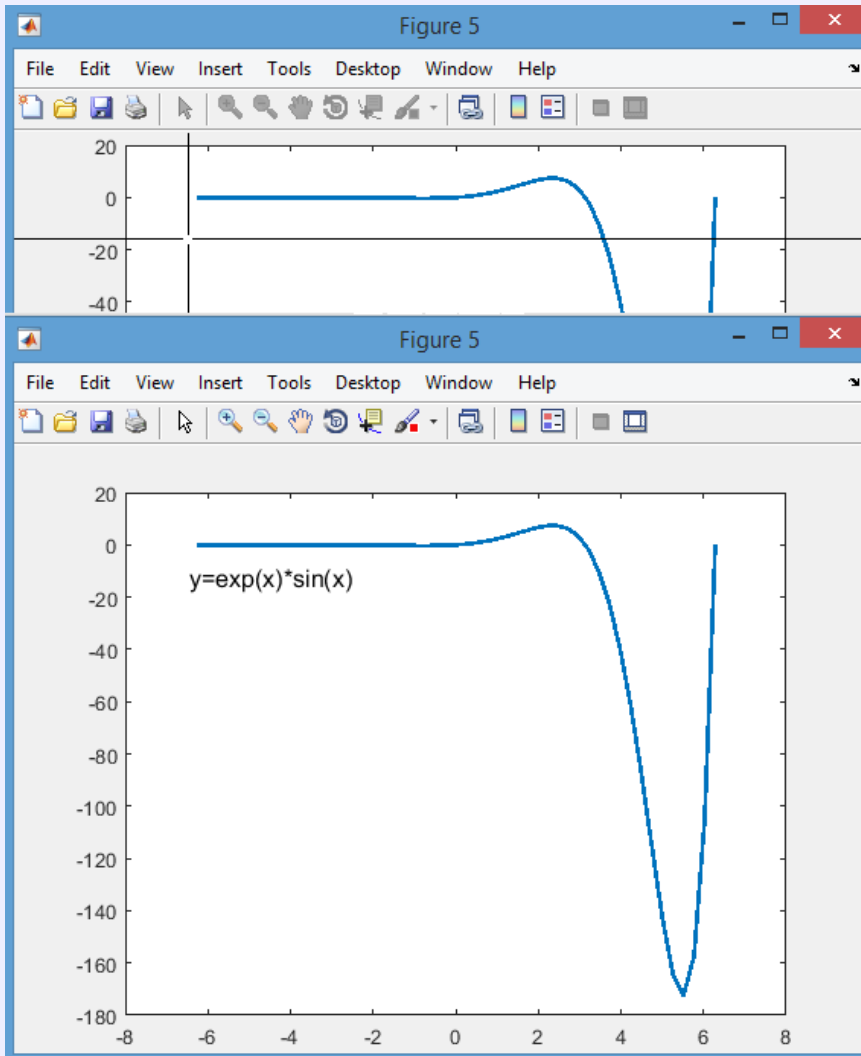


График задан строкой Char

% Using char

n=50;

x=linspace(-2*pi,2*pi,n);

y='exp(x)*sin(x)';

% or y='exp(x).*sin(x)'; eval(y);

plot(x,eval(vectorize(y)),'linewidth',2)

%% интерактивно выбрать

координаты для надписи:

gtext('y=exp(x)*sin(x)','fontsize',12)

Ещё способы построения точек для plot

x=linspace(-2*pi,2*pi,50)

y='exp(x).*sin(x) '
y=eval(y);

~

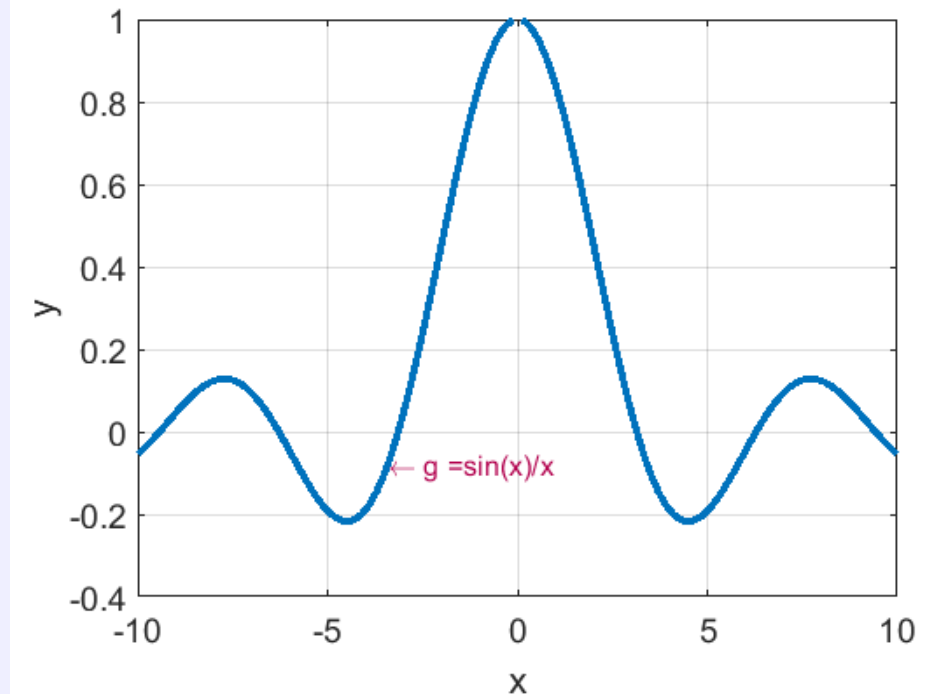
y=exp(x).*sin(x)



Надписи с изысками (стрелки - arrow)

```
clear
x=-10:0.1:10;
g='sin(x)/x'; % строка функции
% выполнить векторизованную строку:
f=eval(vectorize(g));
n=length(x);
p1=plot(x,f);
set(p1,'linewidth',3);
%% почему n/3?
t=text(x(fix(n/3)), f(fix(n/3)),...
       '\leftarrow g =sin(x)/x');
t.FontSize=12 % Remember - Register!
t.Color=[0.7  0.0  0.3] % [r g b]
set(gca,'fontsize',14)% все надписи осей увеличили (14pt)!
```

```
grid on % задана опция сетки на графике
xlabel('x')
ylabel('y')
```



fix – округляет в направлении нуля
round – округляет к ближайшему целому
floor – округляет в направлении $-\infty$
ceil – округляет к в направлении $+\infty$



ПОВЕРХНОСТИ. Сетки Meshgrid. Конструкторы MESH и SURF

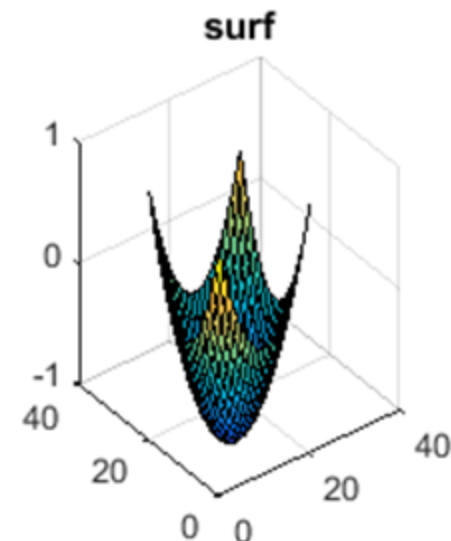
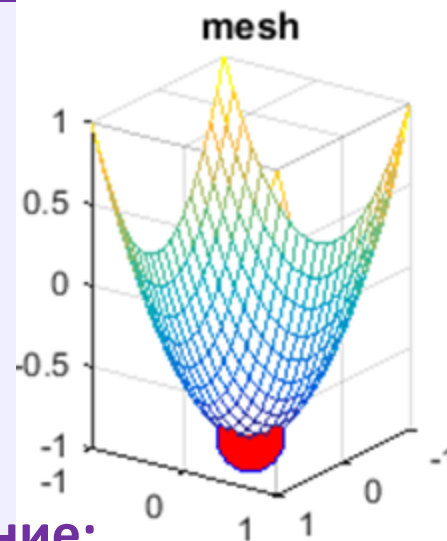
%%Построение сетки в осях XOY:

- 1) `[X,Y]=deal(-1:0.1:1); [XX,YY]=meshgrid(X,Y); % точки сетки в XOY`
- 2) `Z=XX.^2+YY.^2-1; % векторизованная функция поверхности`
- 3) `[minzJ,Jmin]=min(min(Z)); % minzJ - minValue in column Jmin of Z`
- 4) `[minzI,Imin]=min(min(Z'));` % minzI - minValue minzI in Imin row of Z
- 5) `minzJ=min(Z); [minValueJ,Jmin]=min(minzJ);`
- 6) `subplot(1,2,1); rm=mesh(XX,YY,Z); title('mesh'); % see, using full size of window`
- 7) `hold on; Xmesh=get(rm,'xdata'); Ymesh=get(rm,'ydata'); Zmesh=get(rm,'zdata');`
- 8) `r=plot3(XX(Imin,Jmin),YY(Imin,Jmin),minValueJ,'bO');` % minimal Value
- 9) `set(r,'markersize',20,'MarkerFaceColor','red');`
- 10) `subplot(1,2,2), surf(Z), title('surf')`

Результат в окне mesh – после интерактивного вращения!

mesh – прозрачная сетка;

surf – непрозрачная сетка;



Поэлементное присваивание:

`[X,Y]=deal(a1:step1:b1, a2:step2:b2)` – одномоментное присваивание



Функции `fmesh`, `fsurf` в классе `SYMBOLIC`, `mesh`, `surf` - Double. Примеры

1-th: symbolic

```
clear, syms x y  
z=@(x,y)x.^2.*y, s=diff(z,1,x)  
fsurf(z), hold on  
it isn't possible change to ColorMap!  
r=fsurf(s,'FaceColor','interp','EdgeColor',[1 0 0])  
legend('z','zprime')
```

2-th: numeric

```
clear, clf % clean figure  
[x,y]=deal(0:0.05:1); [x,y]=meshgrid(x,y)  
[n,m]=size(x), z=x.^2+y.^2-1  
s=diff(z,1), surf(x,y,z),hold on  
surf(x(1:n-1,:),y(1:n-1,:),s), colorbar
```

**Найдите соответствия построенных объектов
В предложенном коде и на слайде 21!**

**Присылайте ответы в тимс или на почту
СЕГОДНЯ!**

3-th: transparent

```
clear; clf  
syms x y;  
Z=x.^2+sin(x).*y.^2  
S=diff(Z,x)  
[x,y]=meshgrid(-8:0.5:8)  
mesh(x,y,eval(S))
```

4-th: `fmesh`

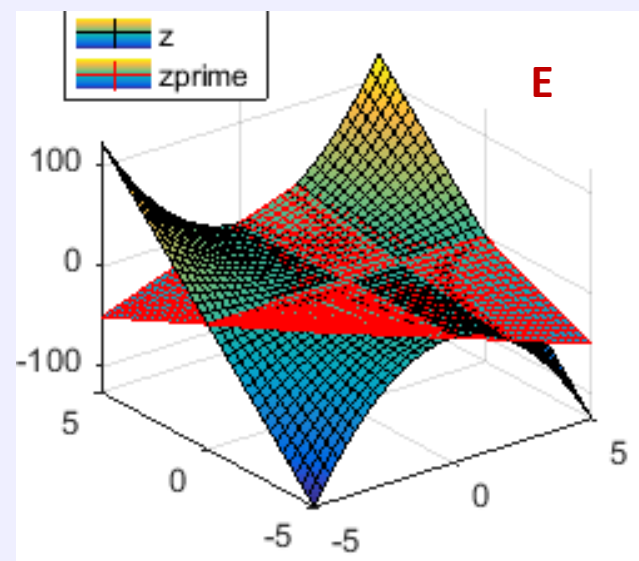
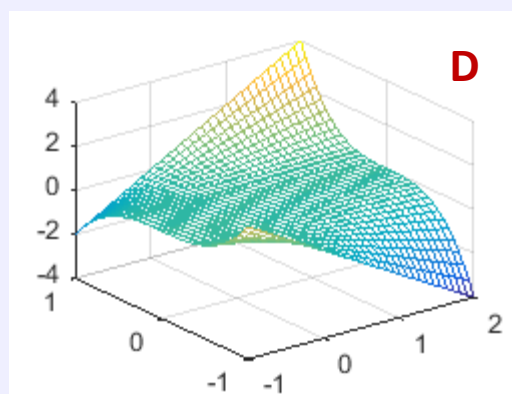
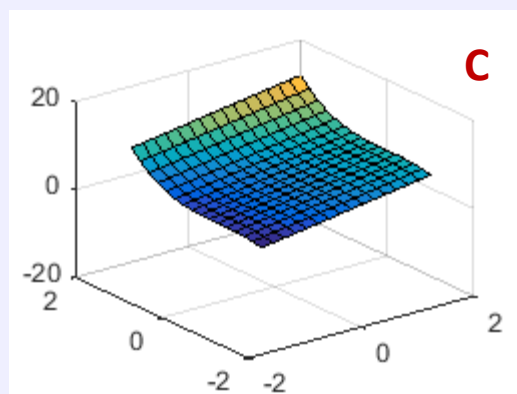
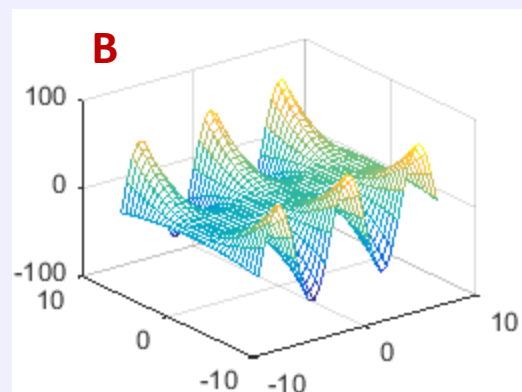
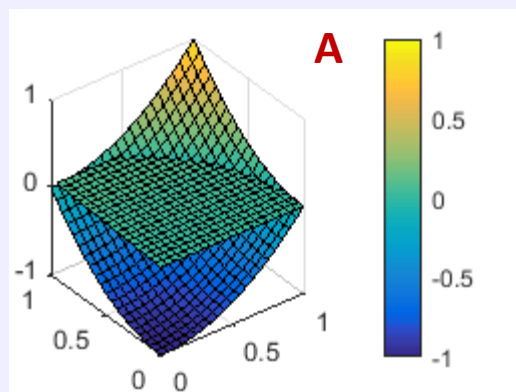
```
clear; clf, syms x y; Z=x^2*y^3,  
Z1=diff(Z,x)  
fmesh(Z1,[-1 2 -1 1]) %[ax bx ay by]
```

5-th: mixed approach

```
clear; clf  
syms x y;  
Z=x^2+x*y^3  
Z1=diff(Z,x)  
[x,y]=meshgrid(-1:0.2:2,-1:0.2:2)  
surf(x,y,eval(vectorize(char(Z1))))
```

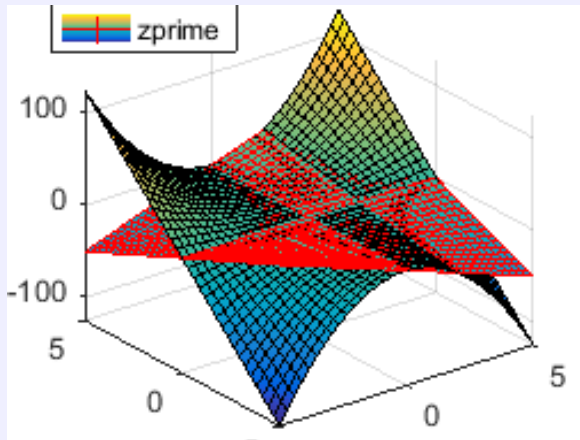


Примеры : графические результаты сл.20





I/O графических объектов/handle graphics



H=figure

hs=fsurf(s,'FaceColor','interp','EdgeColor',[1 0 0])

hgsave(**hs**, 'namefile.fmt'); %fmt=png,jpg,eps,...

hgsave(**hs**, ' mynamefile'); % save as mynamefile.fig

clear, **hgload**('namefile.fmt'); % checking **hgload**

savefig(**H**, 'anyname.fig') % save figure with all graph

clear % all

%% delete figures from n to m:

delete(n:m) % без пропусков!



IMAGE

nvkurbatova@sfedu.ru

```
subplot(1,3,1)
```

```
C = [0 2 4 6; 8 10 12 14; ...  
     16 18 20 22];
```

```
image(C); colorbar % шкала
```

```
load trees % defined by X and map
```

```
% and forest.tif defined by ????
```

```
[X2,map2] = imread('forest.tif');
```

```
subplot(1,3,2), imshow(X,map)
```

```
subplot(1,3,3), imshow(X2,map2)
```

```
% Display a grayscale image, adjust  
(отрегулируйте) the display range
```

```
I = imread('pout.tif');
```

```
h = imshow(I,[0 80]);
```

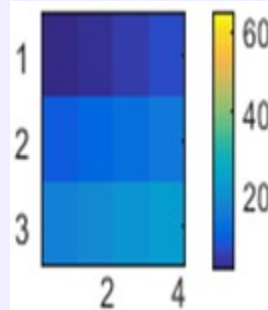
```
% Display a grayscale image
```

```
RI = imref2d(size(I));
```

```
RI.XWorldLimits = [0 3]; % rangeX
```

```
RI.YWorldLimits = [2 5]; % rangeY
```

```
imshow(I,RI);
```



Синтаксис:

% **I**) чтение фото с расширением FMT:

```
[X,MAP] = imread(FILENAME,FMT)
```

imshow(X, map) визуализация изображения

% **II**) визуализация изображения

image (Y) % заданного матрицей Y

imref2d(size(I)) – привязка 2D изображения к координатам

% IN ADDITION!!!

%% Convert to gray color

```
I = imread('example.tif');
```

```
imshow(I) % colored
```

figure

```
J = rgb2gray(I);
```

```
imshow(J)
```





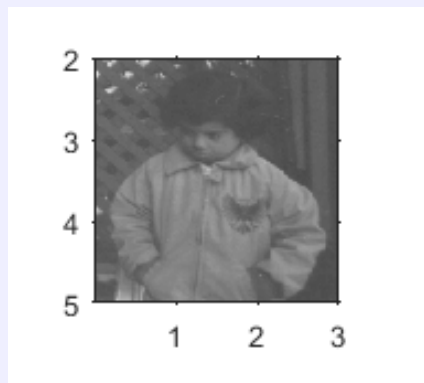
Colormap их использование

R2023a: MATLAB Online limits image display resolution!

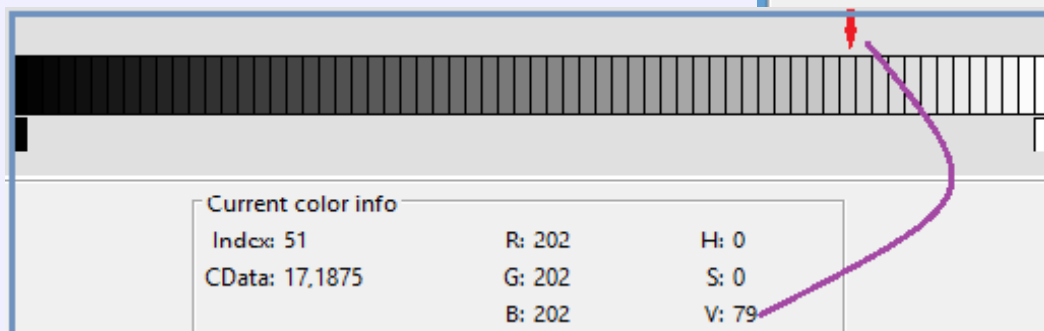
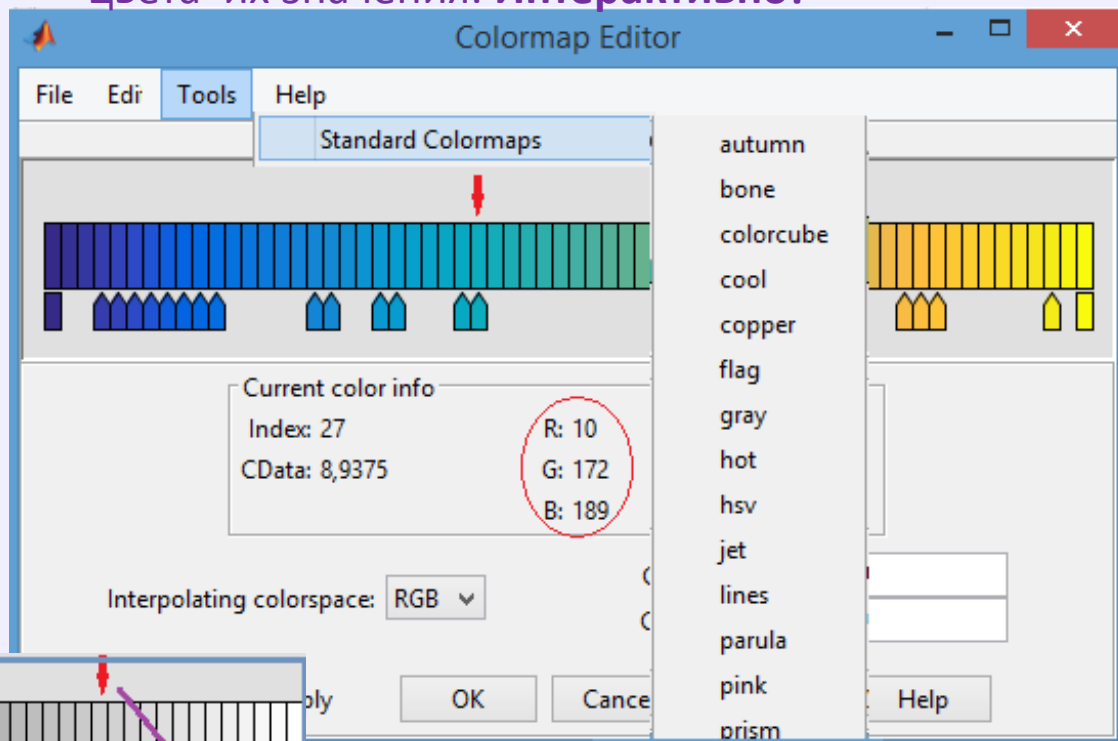
>> colormapeditor

HSV - Hue, Saturation Value — тон, насыщенность, значение.

`h = imshow(I,[0 80]); % см.слайд23`
% нет белого
% цвета!
% 80 – max



Здесь можно **выбрать** и задать (RGB HSV) конкретные цветовые карты и выбрать любимые цвета их значения. **Интерактивно!**





Выбор палитры и редактор цвета

- **>> colormapeditor**

Наберите в командной строке - откройте редактор выбора палитры и цвета

- **Определите свой (родителей и т.д.) цвет**

R – день рождения; G- месяц; B – год, например, последние две (три) цифры

Успехов!



Спасибо за внимание!

*“Люди, цветы и бабочки прекрасны своей
хрупкостью и разнообразием!” NV*