nvkurbatova@sfedu.ru

Scientific Computer Software

(MatLab)

Topic 4. Graphics in MatLab.
Topic 5. Interactive Graphics and features

Kurbatova Natalia Viktorovna

nvkurbatova@sfedu.ru

Contents:

Graphics objects (GO) :

- system and handle;
- descriptive GO ;
- GO as structure.

2D, 3D the graphics set using:

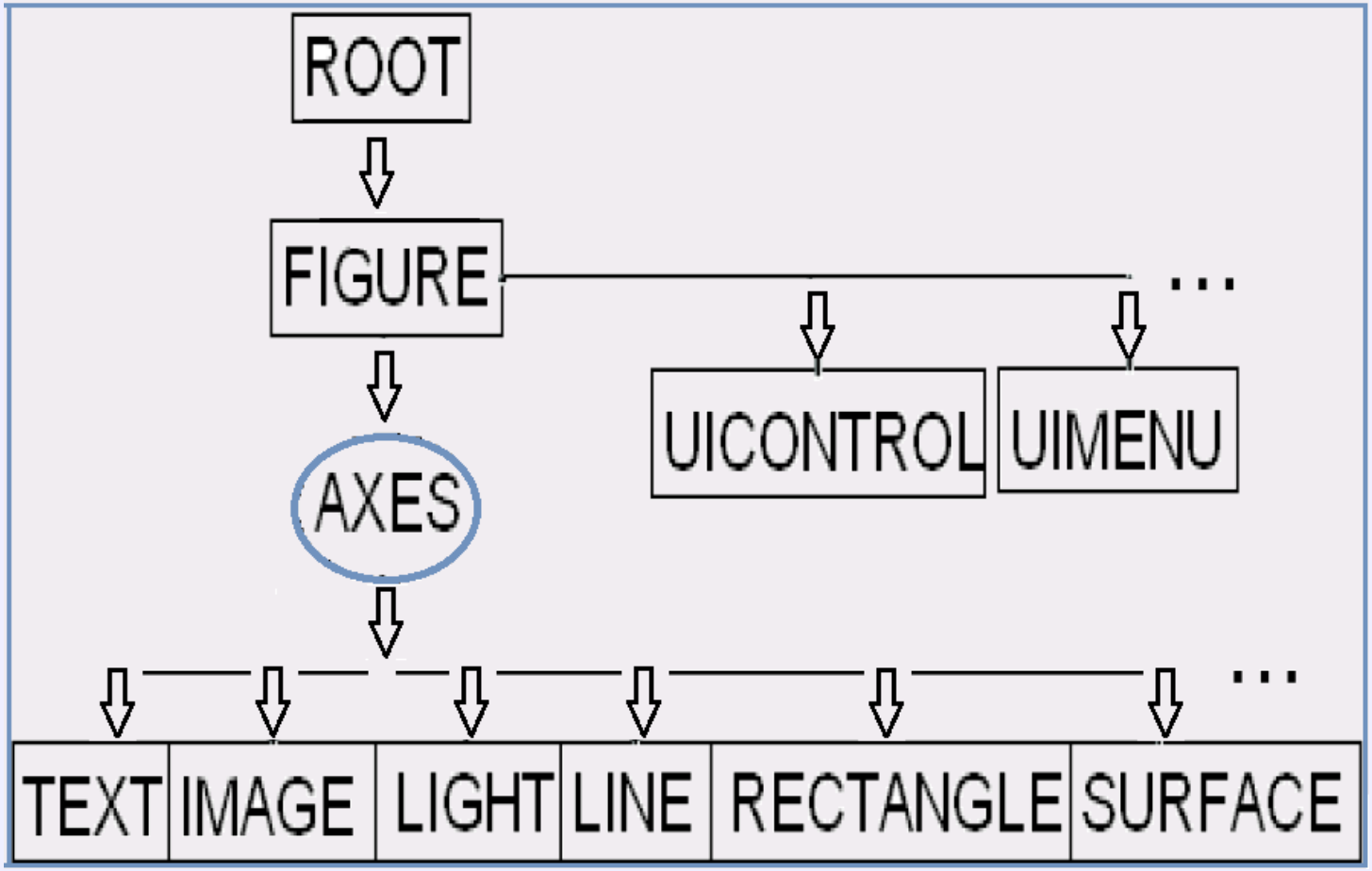
- by string or Symbolic form;
- in parametric form;
- an implicit function or handle function.

Graphic primitives :

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

nvkurbatova@sfedu.ru

The scheme of the hierarchy of classes of GO



nvkurbatova@sfedu.ru

Descriptors GO

Descriptors of current GP:

- **root ~ 0 ~Command Window**
- **figure ~ gcf ~G**raphic**C**urrent**F**igure (gcf – natural numbers, default – windows Figure are created sequentially)
- **axes ~ gca ~ G**raphic**C**urrent**A**xes (**handle –** user's objects)
- **gco ~ G**raphic**C**urrent**O**bject

How to define possible properties:

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

0, cgf, gca, gco – system descriptors

Setting properties:

set(0,'PropertyName',value)

…

Getting properties:

Value=get(0,'PropertyName')…

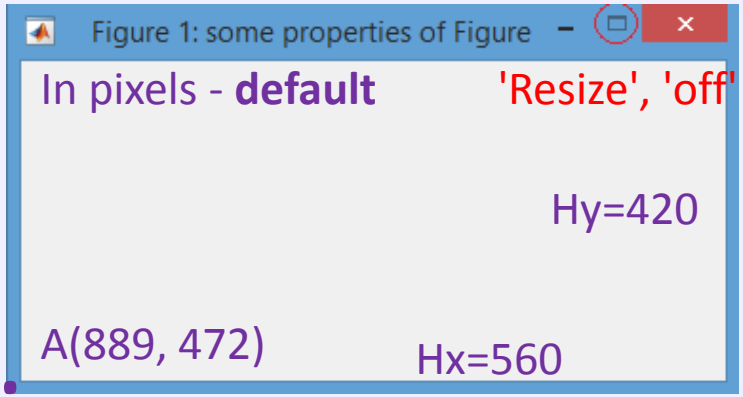
nvkurbatova@sfedu.ru

Figure properties

```
>>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
```



nvkurbatova@sfedu.ru

Multiple charts in one window

I. Plotting multiple graphs with a single function:

plot(x,g, lineSpec1,x,y, lineSpec2, x,f, …))

LineSpec - Color, linestyle, Marker ; lineSpec ∈ Char (f.e., 'r--.') **???**

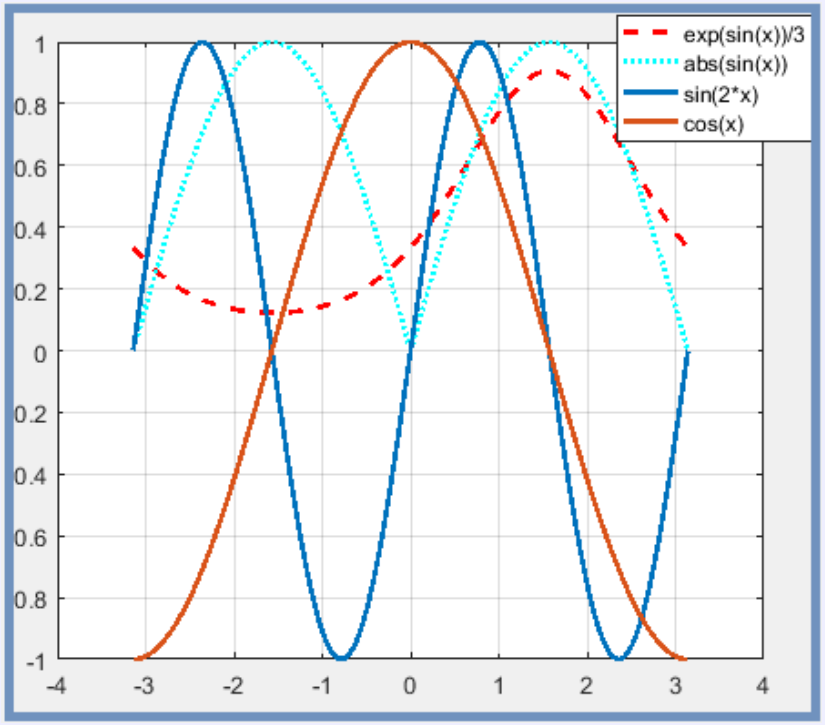
Explain!

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

II. The graphs are sequentially placed in the figure:

```
% Example Using char
 clear; n=50;  x=(linspace(-pi,pi,150)) ' % column
y='exp(sin(x))/3';  y=eval(y); % evaluate char
r=plot(x,y,'r--') , hold on          % 1-th way
g=abs(sin(x)); t=plot(x,g, 'c:'), % 2-th way
    set([r,t],'linewidth',2);
M=[sin(2*x), cos(x)]; % M – matrix of columns (?)
p=plot(x,M), grid on                 % 3-th way
  set(p,'linewidth',2);
  legend('exp(sin(x))/3','abs(sin(x))', 'sin(2*x)', 'cos(x)')
```

Example

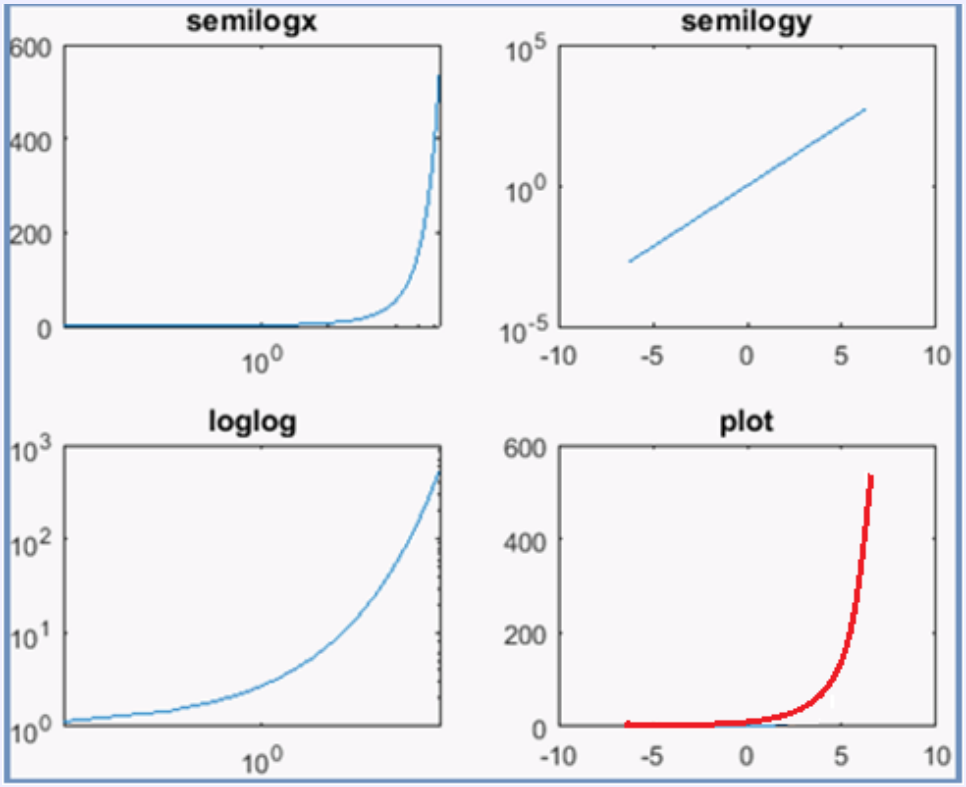


III. Vertical line with constant x-value:

xline(x) or xline(x,LineSpec,labels)

Taking into account the growth of functions

Logarithmic scale for a single axis or for all:



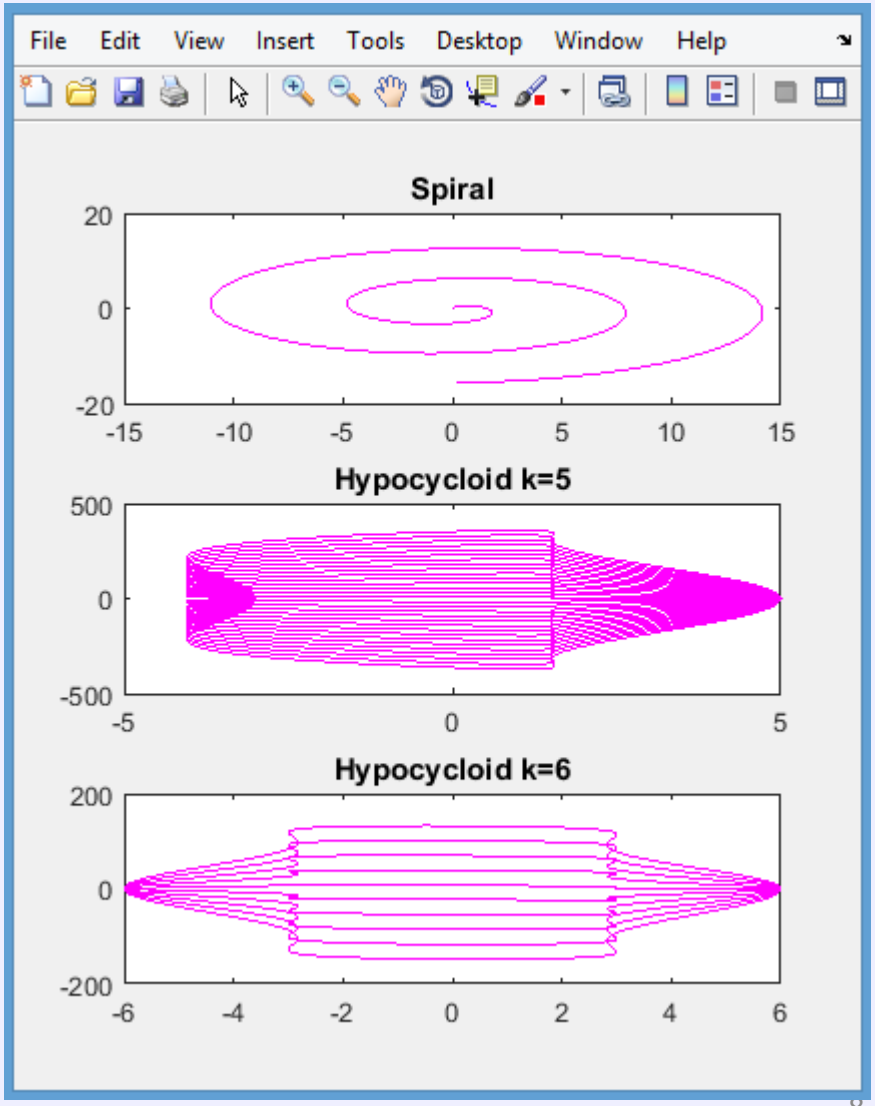

```
clear
n=50
x=linspace(-2*pi,2*pi,50)
% Logarithmic scale for single or for all axes:
figure % necessary?
y=exp(x);

% 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 ',
```

Remember that the function **set** and **get** fields are case-free :

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

Graphs of functions defined parametrically

```
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')
```



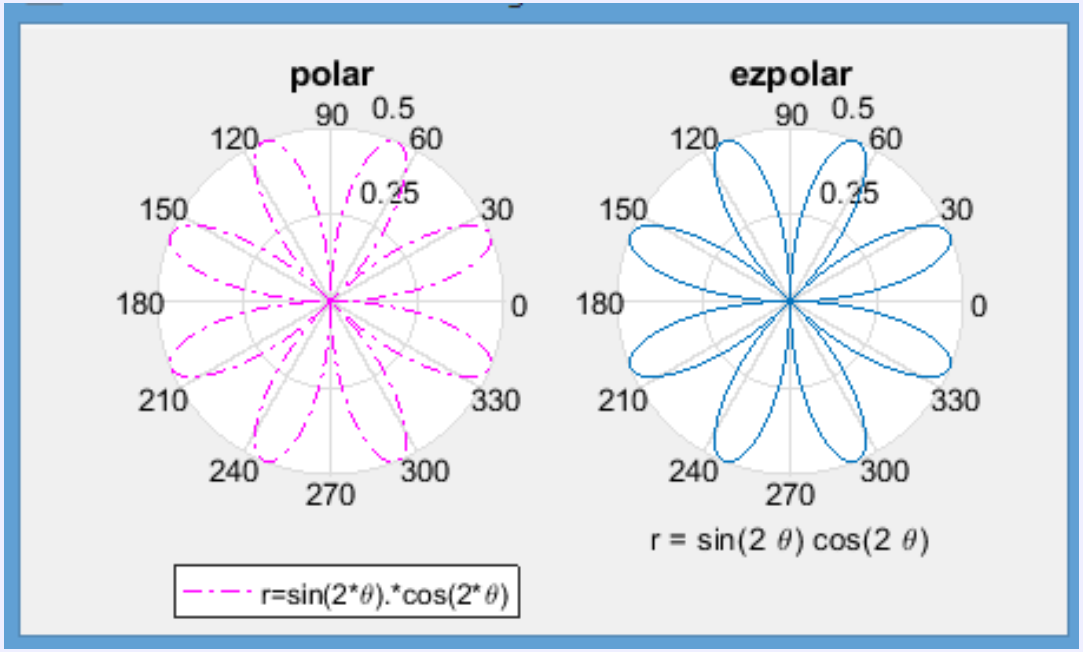
Graphs of functions given in polar coordinates

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 – for new ML versions!!!
pollarplot – old!

POLAR , EZPOLAR



Parameters of EZPOLAR – CHAR or Symbolic

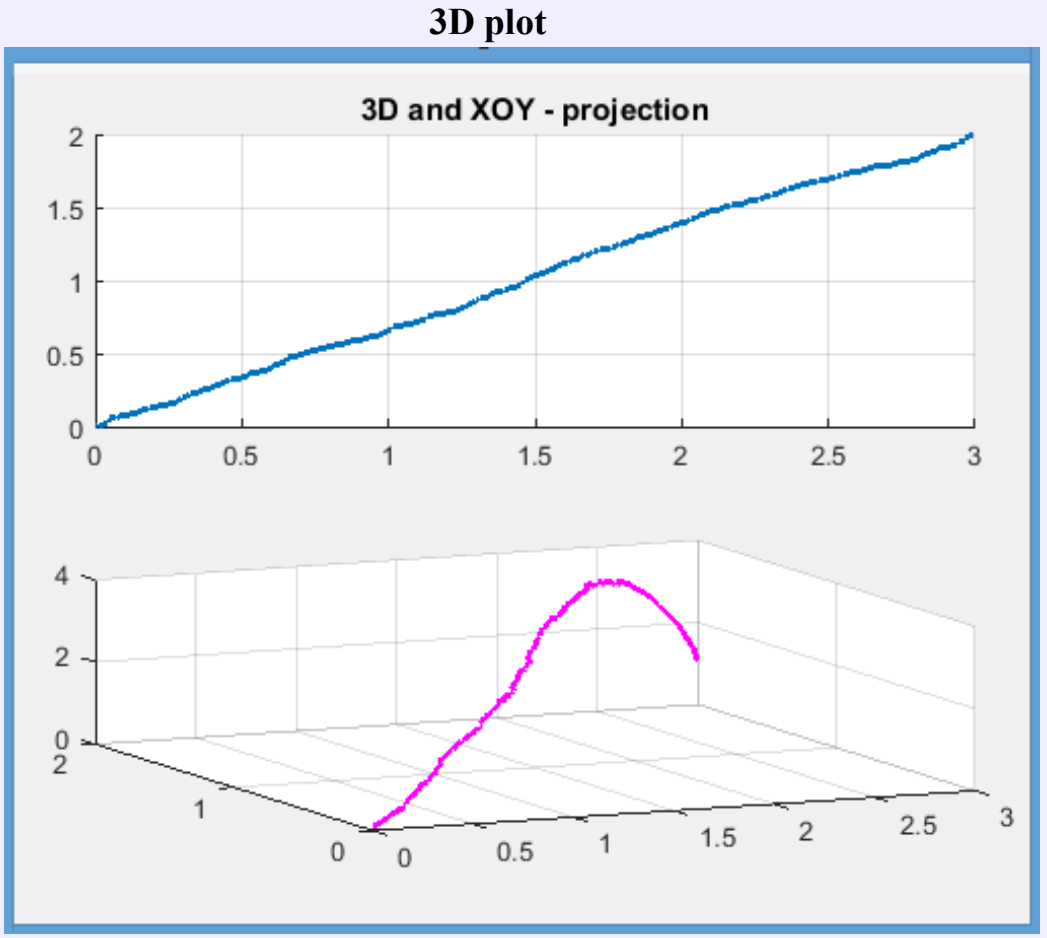
nvkurbatova@sfedu.ru

Primitives: Line. 3D Line~plot3

title(…) -
sgtitle(…) – title for subplot

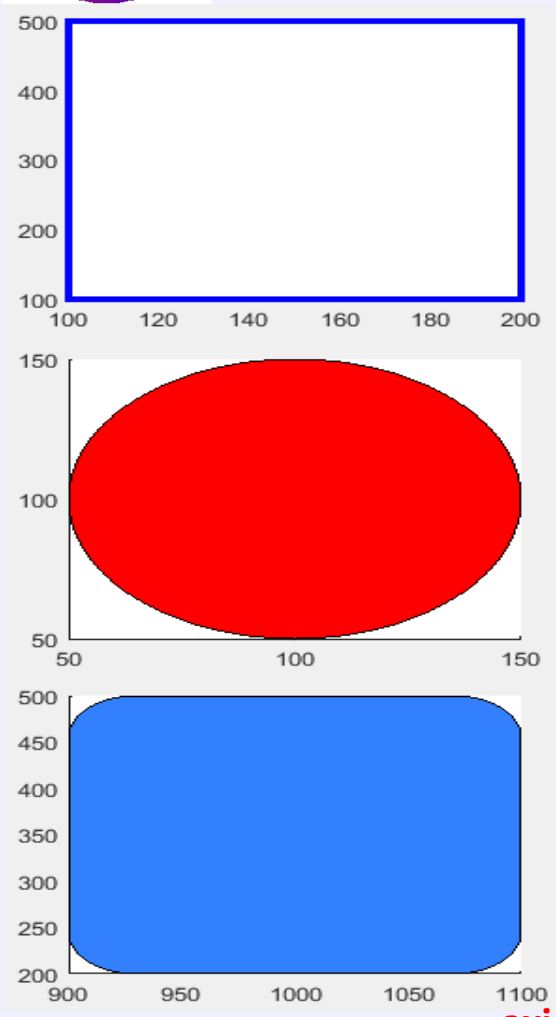
```
figure
title ('3D plot')
 sgtitle('3D and XOY - projection')
   X=sort(3*rand(1,1000));
   Y=sort(2*rand(1,1000));

subplot(2,1,1)
h=line(X,Y,'linewidth',2);  grid on
Z=sin(X).*exp(Y);  ? Explain!

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!)



nvkurbatova@sfedu.ru

Primitive: Rectangle.



'Curvature' ={[0 0] – Rectangle**; [1 1]** – ellipse; **[k,r]** – rounded vertices**, 0< k,r<1}**

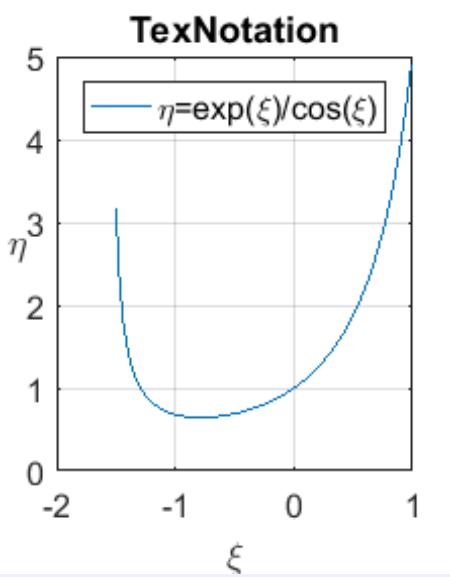
1) figure
2) subplot(3,1,1)
3) rectangle('Curvature',[0 0],'Position',[100 100 100 400],... '**EdgeColor**','blue','linewidth',3) %creates a rectangle
4) subplot(3,1,2)
5) rectangle('Curvature',[1 1],'Position',[50 50 100 100],...
6) '**facecolor**','red') %creates an ellipse, circle
7) subplot(3,1,3)
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) % Study the properties of the primitive**!**

rectangle() % creates a square of unit length

axis square ***% you need to insert this command after line 6, the elongation will disappear.***

Screen elongation coefficient : r(4)/r(3)=Hy/Hx

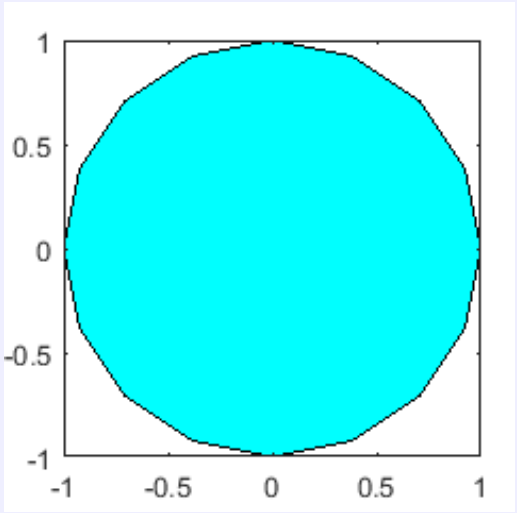
Using Notation of LaTeX. Color filling function - fill



1) x=-1.5:0.03:1; y=exp(x)./cos(x) % x-radian
2) plot(x,y);

%% Graphic design:

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) % **text writing of** 12pt



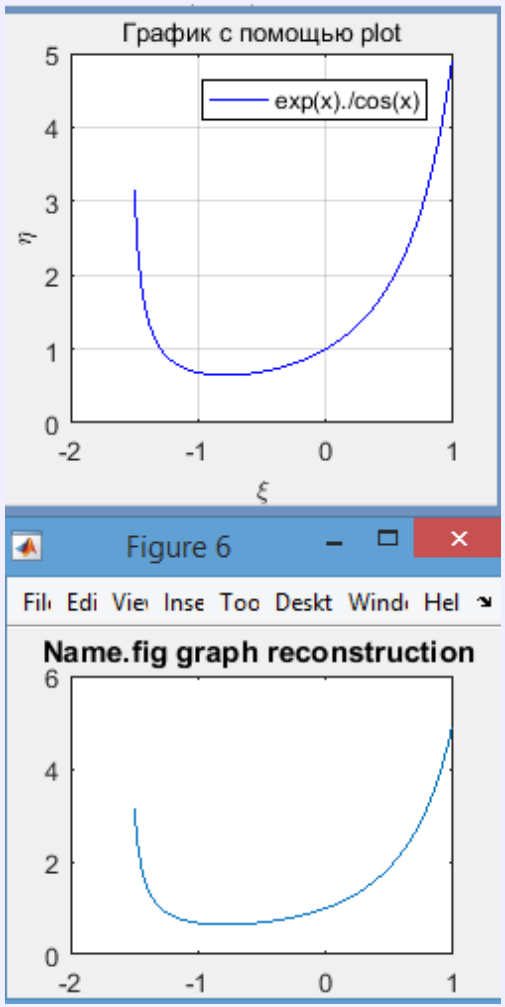
1) t = (1/16:1/16:1)'*2*pi; x = cos(t); y = sin(t);

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

2) fill(x,y, 'c') % fills in a closed line with color
3) axis square % the effect of elongation of the screen is removed

Import - hgload! *.fig – data source!

Before



After

1) hfig = hgload('Name.fig') % load Name.fig in Folder
2) figure(hfig); % visualization
3) haxe=get(hfig,'Children') % Graphs!

%% haxe contains single **line –** the same approach for multiple lines

5) ChAxes=get(haxe,'Children')% **line finding!**

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, has *function_handle* type and contains the single the executable operator

Limits – function definition interval

lineSpec – the specifier: *Color, LineStyle, Marker*; *lineSpec* ∈ Char (f.e., 'm--*') – repeat!

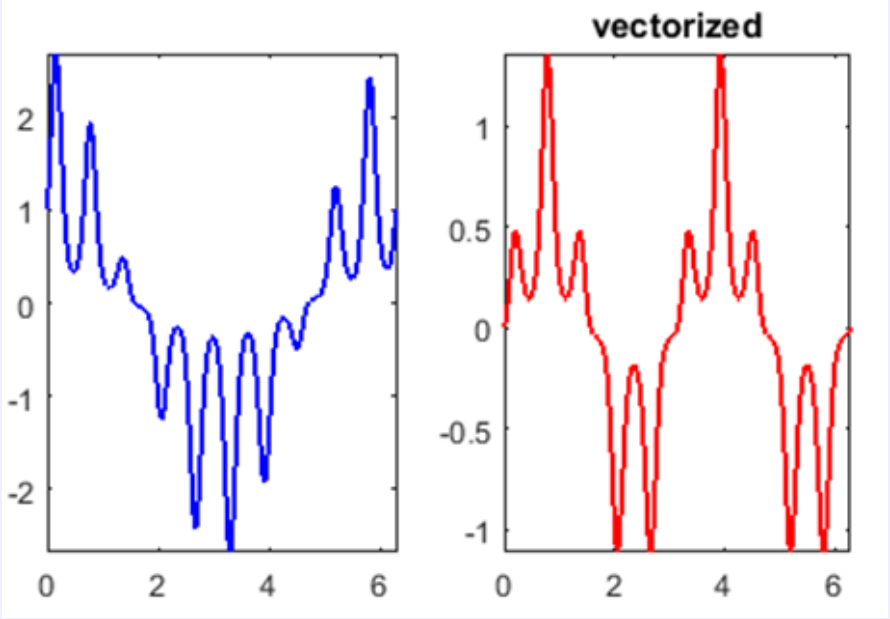
Example:

```
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

% % Vector or element-by-element operations:

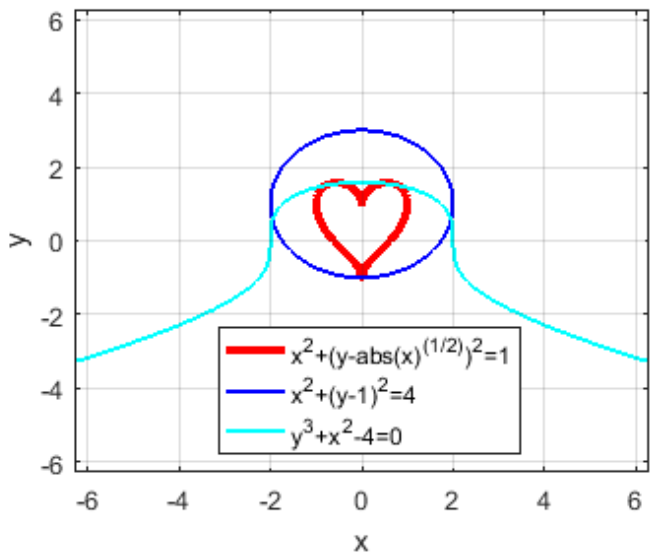
```
subplot(1,2,1)
f = @(x,n) exp(sin(n*x)).*cos(x).*sin(x);
fplot(@(x)f(x,10),[0 2*pi],'r','linewidth',1.5)
 title('vectorized')
```



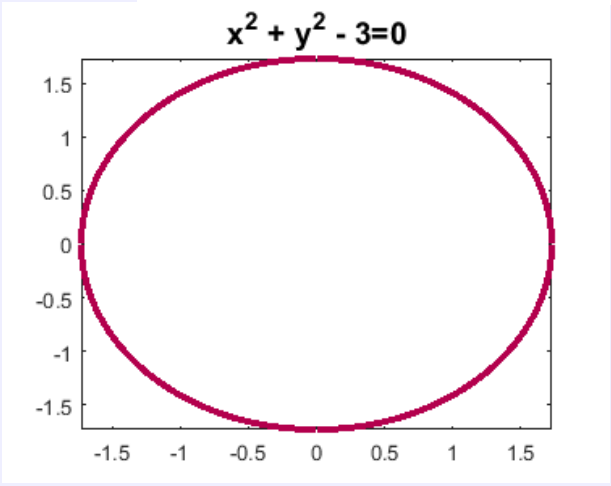
nvkurbatova@sfedu.ru

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

```
%% ezplot;   fplot  - for new versions!!!
figure
r=ezplot('x^2+(y-abs(x)^(1/2))^2=1'), hold on, grid on
set(r,'linewidth',3,'color','red' ); title(' ')
% the line of the first function with a descriptor is
automatically placed in the header 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)
```



nvkurbatova@sfedu.ru

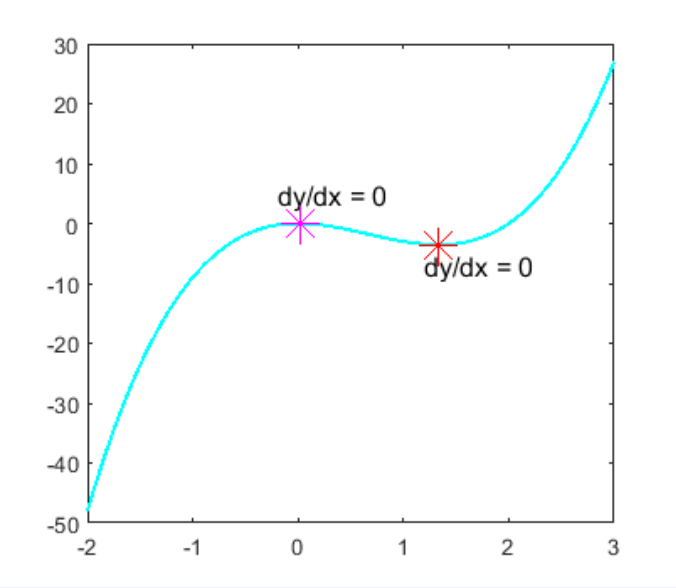
Primitive Text: text(x,y,characteristic,opts)

%% I.TEXT . Extreme points are added:

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

%% The inscription about the zero derivative
      is set by evaluating visually:
xt = [-0.2   1.5];  yt = [5 -7]; str = 'dy/dx = 0';
text(xt,yt,str,'fontsize',12),  hold on
%  may be will add something?!

%% II.TEXT.  Conditions for min  and  max:

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

% reserve array for min and  max finding :

yi=y(2:end-1) ; yip1=y(3:end); yim1=y(1:end-2)
%  notations: yi=y_{i};  yim1= y_{i-1};  yip1=y_{i+1}
%in curly brackets indexes in notation Latex, not forMatLab!
```

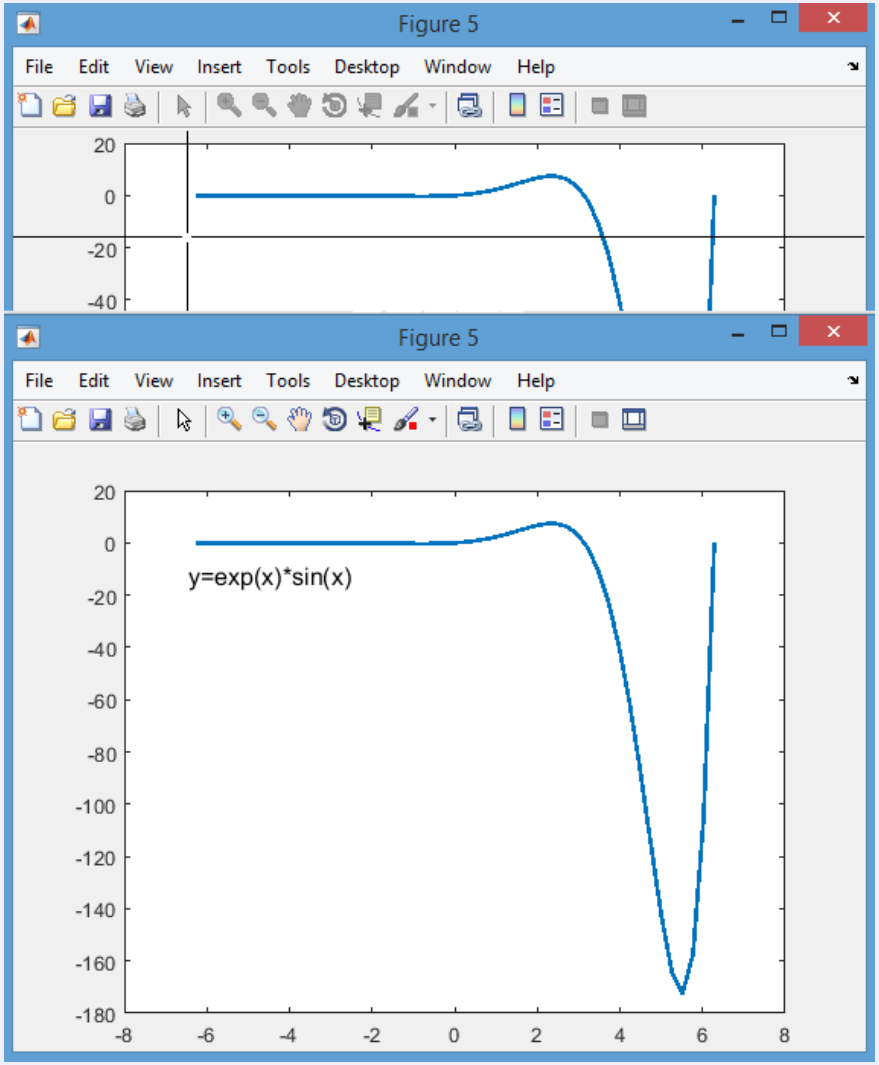
% III.TEXT – Find and note extreme points:

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

trmin=and(yim1>yi,yi<yip1) %min condition
indexmin=find(trmin) % here single 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)
```



nvkurbatova@sfedu.ru

Interactive GTEXT



Given as 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)
%% interactive choice:
gtext('y=exp(x)*sin(x)','fontsize',12)
```

Some other ways for plot

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

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

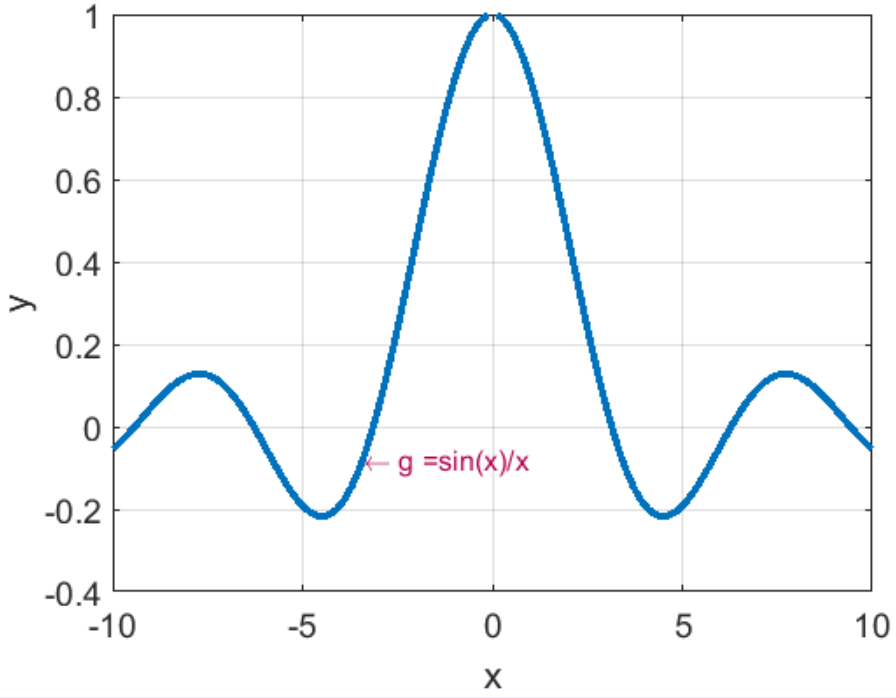
~

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

nvkurbatova@sfedu.ru

Explanations for graph

```
clear
x=-10:0.1:10;
g='sin(x)/x'; % characteristics
% evaluate string:
f=eval(vectorize(g));
n=length(x);
p1=plot(x,f);
set(p1,'linewidth',3);
%% Why 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)%  all text of gca is (14pt)!

grid on %
xlabel('x')
ylabel('y')
```



rounding up:

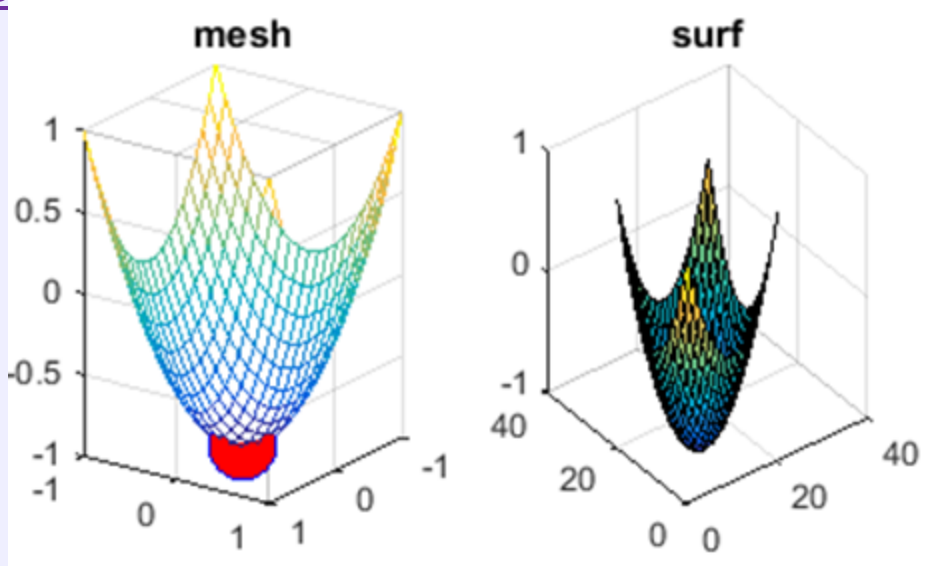
fix – округляет в направлении нуля
round – округляет к ближайшему целому
floor – округляет в направлении $-\infty$
ceil – округляет к в направлении $+\infty$

Surface. Constructors MESH и SURF

%% Grid creation in XOY:

```
1)  [X,Y]=deal(-1:0.1:1);   [XX,YY]=meshgrid(X,Y); % grid pointsXOY
2)  Z=XX.^2+YY.^2-1;        %  vectorized surface function
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)  minzJJ=min(Z);    [minValueJ,Jmin]=min(minzJJ);
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')
```

Result after rotation is in window mesh



mesh – transparent grid;

surf – not a transparent grid;

Element by element assignment :

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

Functions fmesh, fsurf in class SYMBOLIC, mesh, surf - Double. Examples

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
```

Find the matches of the constructed objects (on slide 21) in the proposed code and send your answers by the team or by email TODAY!

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

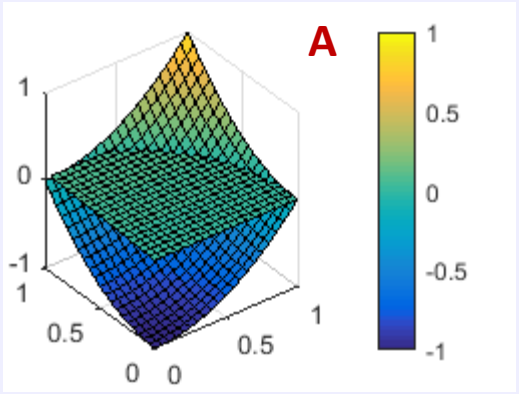
```
clear; clf, syms x y; Z=x^2*y^3,
Z1=diff(Z,x)
fmesh(Z1,[-1 2 -1 1]) %[ax bx ay by]
```

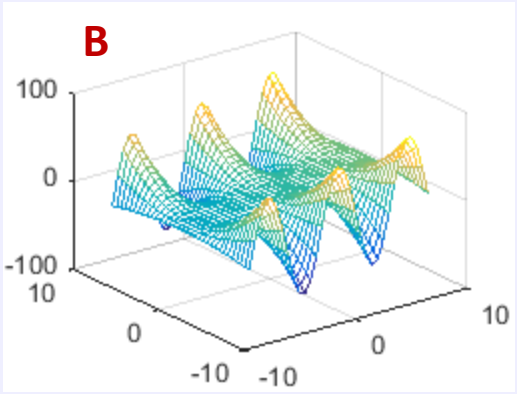
5-th: mixed approach

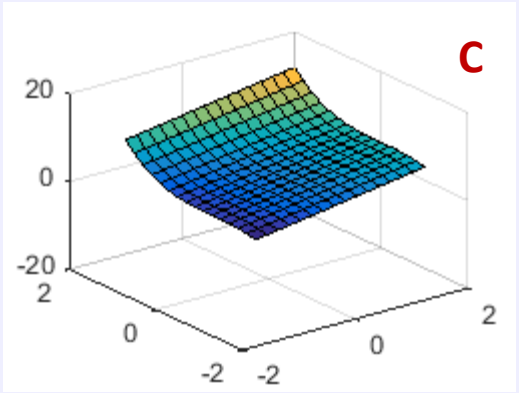
```
clear; clf
syms x y;
Z=x^2+x*y^3
Z1=diff(Z,x)
[x,y]=meshgrid(-1:0.2:2,-1:0.2:2)
surf(x,y,eval(vectorize(char(Z1))))
```

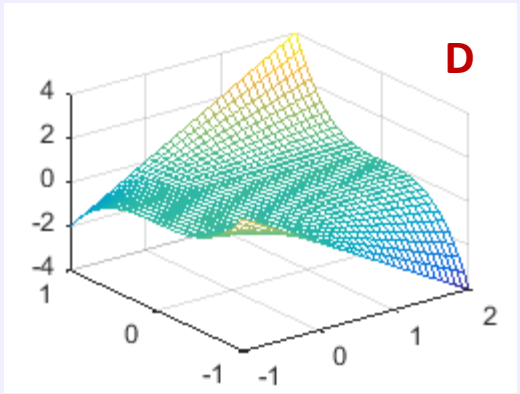
nvkurbatova@sfedu.ru

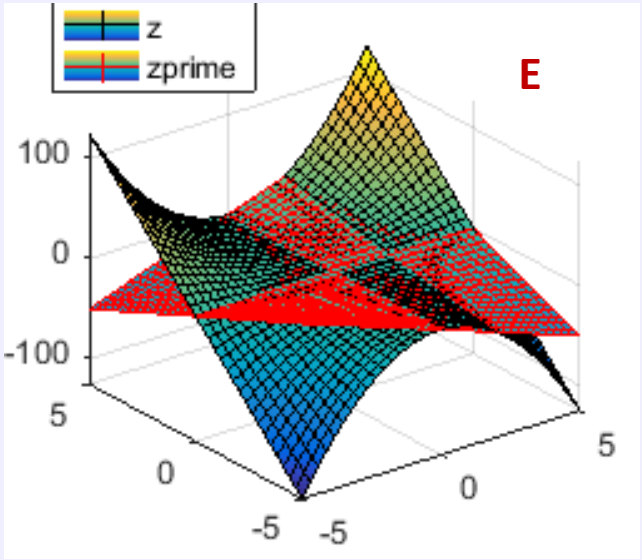
Graphs built using the code on slide 20





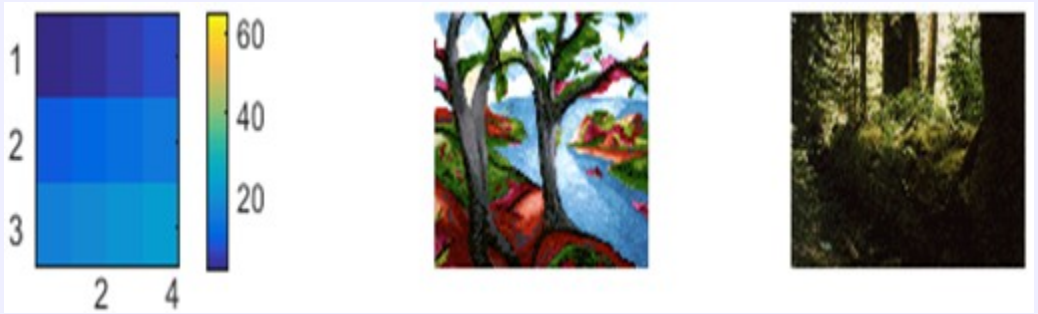






nvkurbatova@sfedu.ru

IMAGE




```
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.tiff 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)reading pfoto *.FMT:
[X,MAP] = imread(FILENAME,FMT)
imshow(X, map) % visualization
% image visualization
image **(Y) %** given matrix Y
imref2d(size(I)) – binding a 2D image to coordinates

```
% IN ADDITION!!!
%% Convert to gray color
I = imread('example.tif');
imshow(I) % colored
figure
J = rgb2gray(I);
imshow(J)
```

Colormap - using

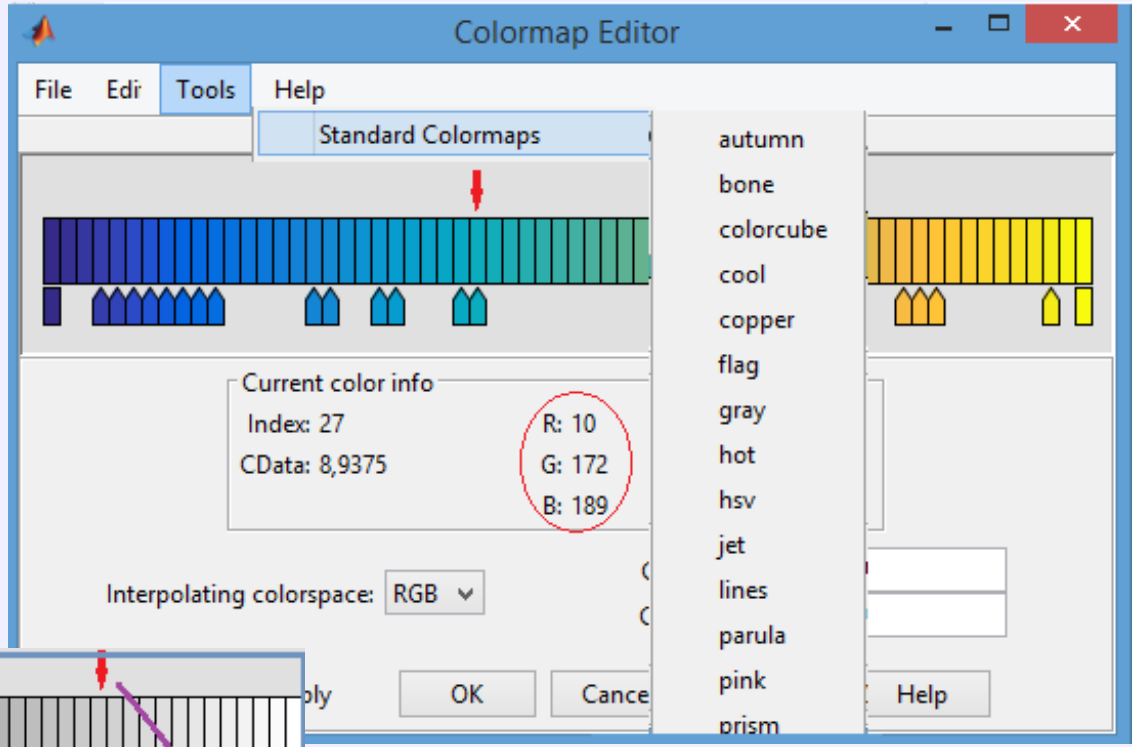
R2023a: MATLAB Online limits image display resolution!

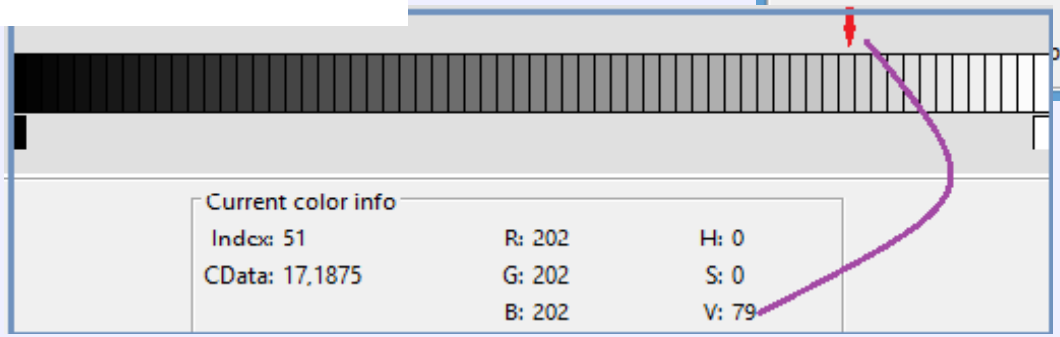
You can set (RGB HSV…) colormap and choose favorite colors. **Interactive!**

>> colormapeditor

HSV - Hue, **S**aturation **V**alue — tone, saturation, value

```
h = imshow(I,[0 80]); % see.slide 20
        % without white color here
        % 80 – max
```





Palette selection and color editor

- **>> colormapeditor**

  Enter in the command line - open the palette and color selection editor

- **Define your color**

  R – birthday; G- month; B – year, for example, the last two (three) digits

Success to you!

nvkurbatova@sfedu.ru

Спасибо за внимание!

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

People, flowers and butterflies are beautiful for their fragility and diversity! – it seems to me