

Лабораторная работа 2

Работа с графическими объектами в Matlab

Все задания выполнить в одном скрипт-файле, каждое задание – в отдельной секции (использовать Insert section break (Ctrl+Alt+Enter) во вкладке Editor->Section или %%).

Следует использовать глобальные переменные предыдущих заданий, не повторяя команд, если не задана очистка оперативной памяти clear.

Все команды выполнять пошагово, чтобы отследить особенности выполнения.

Указание к выполнению заданий. Выполните предлагаемый код в Matlab, в комментариях возле каждой строки напишите результат выполнения команд, что делает каждая команда. Дополнительно ответьте на вопросы, указанные в комментариях.

Задание 1

Постройте график функции $y=\cos(x)$, $x\in[0,2\pi]$ и подпишите оси

```
figure (1) % создание графического окна с номером 1
x=0:0.1:2*pi % в дальнейшем можно подавлять вывод на экран
компонент вектора x (использовать ; после команды)
y=sin(x)
y2=y'
plot(x,y)
xlabel('x') % подписи осей
ylabel('y')
plot(x,y2) % работает ли и почему
```

Задание 2

Постройте график функции $y=\cos(2x)$, $x\in[0,2\pi]$ красным цветом сплошной линией, выделив опорные точки звёздочкой, полагая, что **функция задана строкой**.

```
figure (2)
x=0:0.5:2*pi; %попробуйте уменьшить величину шага с 0.5 до 0.1
%Как изменится график?
y=eval('cos(2*x)');
plot(x,y,'r-*')
```

Задание 3

Постройте график функции $y=x\cdot\cos(x^2)$, $x\in[0,2\pi]$ зеленым цветом пунктирной линией, затем измените цвет и стиль на черную точечную линию.

```
figure (3)
x=0:0.1:2*pi;
%1й способ - использование поэлементных операций
y=x.*cos(x.^2);
plot(x,y,'g--')
```

```
%2й способ - использование eval(vectorize)
y=eval(vectorize('x*cos(x^2)'))
plot(x,y,'g--')% Измените цвет и стиль: черная точечная линия
```

Задание 4

Постройте графики функций $y=\cos(x)$ и $g=\cos(2x)$, $x\in[0,2\pi]$ красным цветом сплошной линией и пунктирной, поместив оба графика в одни оси и снабдив легендой. Интерактивно поместите текст $\max=1$ у первого максимума. Затем измените толщину пунктирной линии.

```
figure (4)
x=0:0.1:2*pi; % можно повторно не выполнять, есть в памяти
y=cos(x);
g=eval('cos(2*x)');
d=plot(x,y,'r-',x,g,'r--') % два графика в одних осях
legend('y=cos(x)', 'g=cos(2x)')
d(1) % что это за объект?
d(2) % что это за объект?
gtext('max=1') % разместить надпись интерактивно в figure(4) -
щелкнуть на осях
set(d(2),'linewidth', 2.5) % изменить толщину второй линии
```

Задание 5

Изменить на базе задания 4 толщину и цвет графика кривой $\cos(x)$, используя активные оси в figure.

```
f=get(gca, 'children') % используем текущие оси в figure
whos f % почему f размера 3X1?
set(f(3),'linewidth', 1.5)
% дополнительно измените цвет линии
```

Задание 6. В новом окне построьте графики функций $y=\cos(x)$, $f=(\cos(x))^2$ и $g=x*\cos(2x)$, на промежутке $[-2\pi,2\pi]$, предварительно очистив оперативную память. Все графики построить сгенерированным цветом с использованием формата RGB. Вместо легенды оси снабдить заголовком с перечнем функций. Сохранить графический файл с именем myplot.fig (File->Save as (figure))

```
clear
figure (6)
x=-2*pi:0.1:2*pi
y=cos(x);
f=cos(x).^2;
g=eval(vectorize('x*cos(2*x)'));
YFG=[y' f' g'] % что здесь формируется?
r=plot(x,YFG); % как plot работает со вторым аргументом в данном
случае?
yfg=[y;f;g] % что здесь формируется?
r=plot(x, yfg); % как plot работает со вторым аргументом в данном
случае?
```

```

for k=1:3
    set(r(k), 'Color', [k*0.2 1-k*0.3 0.5])
end
title('y=cos(x) f=(cos(x))^2 g=x*cos(2x)')

```

Задание 7.

Построить графики функций $y=\sin(x)$ и $g=x*\exp(x)$ на промежутке $[-2\pi, 2\pi]$.
Использовать одно графическое окно, а в нем организовать структуру из двух графиков, расположенных рядом.

```

figure (7)
x=-2*pi:0.2:2*pi;
y=sin(x);
g=eval(vectorize('x*exp(x)'))
subplot(1,2,1), plot(x,y,'b-^',x,g,'m-*') % subplot - строит
матричную структуру графических окон
subplot(1,2,2), plotyy(x,y,x,g) % построение двух графиков с
разным порядком роста, используются две оси ординат

```

Задание 8. Как извлечь файлы графиков из *.fig (в координатной форме)

Выбрать в папке требуемый файл (File->Open (figure)), например, myplot.fig из задания 6 (или можно выбрать любые figure, в которых есть оси и графики, чтобы оси стали активными).

Извлечь массивы координат точек имеющихся графиков; gca - активные оси, в которых размещены графики

```

r=get(gca, 'children') % gca - графические текущие оси; графики -
потомки осей
xy=[]
for i=1:length(r)
    xplots{i}=get(r(i), 'xdata')
    yplots{i}=get(r(i), 'ydata')
    xy=[xy(:, :) xplots{i}', yplots{i}']
end

figure (8)
plot(xy(:, 1), xy(:, 2:2:2*length(r)))
% сохраните матрицу графиков xy в двоичном файле и в текстовом.
save xy

```

Задание 9. Устаревшая функция ezplot (Easy-to-use function plotter).

Постройте график функции $\sin(x)/x$, задать цвет в RGB с преобладанием красного.
Использовать функцию **ezplot**.

```

clear
figure(9)
% ezplot - easy plot

```

```

% функция строит график выражения или функции, заданной строкой, на
интервале [-2*pi,2*pi]
% вектор независимой переменной x задавать не требуется
r=ezplot('sin(x)/x') % Что добавляет функция ezplot на график?
set(r,'color',[0.8 0.2 0.5])

```

Задание 10. Построение поверхностей

Построить поверхность $z=x^2+y^2-1$ если $-1 \leq x \leq 1$; $-1 \leq y \leq 1$; двумя способами и поместить в разные окна (с помощью subplot и с помощью tiledlayout).

```

clear
figure (10)

x=-1:0.1:1
y=-1:0.1:1
[xx,yy]=meshgrid(x,y) % построение сетки в плоскости хоу
z='xx^2+yy^2 -1'
zz=vectorize(z);
pointsurf=eval(zz);
subplot(1,2,1)
surf(pointsurf); % изучить различие этого конструктора и mesh

subplot(1,2,2)
mesh(pointsurf); % изучить различие этого конструктора и surf

figure(101)
% сделайте то же самое с помощью tiledlayout вместо plot

```

Задание 11. Построение примитивов

Познакомьтесь с помощью справки Help с синтаксисом и возможностями использования примитивов line, text; rectangle. Постройте в новом графическом окне три графика один под другим (с помощью subplot или tiledlayout). В каждом графике разместить по одному указанному графическому примитиву.

Задание 12. Изучите справку по командам histogram (гистограмма), boxchart (блочная диаграмма, «ящик с усами»), bubblechart (пузырьковая диаграмма), scatter (график рассеяния), pie (круговая диаграмма), heatmap (тепловая карта). Справку лучше всего смотреть в справочной системе и на сайте Matlab, если она недоступна, то можно здесь https://docs.exponenta.ru/matlab/referencelist.html?type=function&category=pie-charts-bar-plots-and-histograms&s_tid=CRUX_topnav (но перевод некачественный!)

Придумайте и выполните по одному практическому примеру для каждой команды. Используйте ИИ для создания примеров; дополнительно изучите, что означают эти виды диаграмм и для чего используются. Напишите ответ в комментариях.