

# 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

### Задание 1

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

```
figure (1) %номер графического окна, каждое задание будет в
отдельном графическом окне
x=0:0.1:2*pi % в дальнейшем подавлять вывод на экран компонент
вектора x (использовать ; после команды)
y=sin(x) % в дальнейшем подавлять вывод на экран
plot(x,y)
xlabel('x') % подписи осей
ylabel('y')
```

### Задание 2

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

```
figure (2)
x=0:0.1:2*pi; %попробуйте уменьшить величину шага с 0.1 до 0.01
%Как изменится график?
y=eval('sin(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.

```
figure (3)
x=0:0.1:2*pi;

%1й способ - использование поэлементных операций
y=x.*cos(x.^2);

%2й способ - использование eval(vectorize)
y=eval(vectorize('x*cos(x^2)'))

plot(x,y,'g--')
```

#### 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

```
figure (4)
%x=0:0.1:2*pi; % можно повторно не выполнять, есть в памяти
y=cos(x);
g=eval('cos(2*x)');
plot(x,y,'r-',x,g,'r--') % два графика в одних осях
legend('y=cos(x)', 'g=cos(2x)')
gtext('max=1') % щелкните в осях в точке расположения надписи
```

#### Задание 5

Изменить на базе задания 4 толщину пунктирной линии.

```
% 1-й способ
d=get(gca, 'children') % d - двумерный (по числу графиков) вектор
из дескрипторов графиков,
d=sort(d) % упорядочиваем по убыванию, теперь d(1), d(2)
дескрипторы первого и второго графиков (дескрипторы формируются
так, как в стеке, первый - помещается в конец и имеет меньшую
величину и т.д.)
set(d(2), 'linewidth', 2.5) % изменяем толщину линии, по умолчанию
linewidth=1 ('linewidth'-свойство поля структуры 2.5 - его
значение)

% 2-й способ
%создаем новое окно Figure для второго способа:
figure % создается окно со следующим номером или можно figure(n), n
- натуральное
d=plot(x,y,'r-',x,g,'r--');
set(d(2), 'linewidth', 2.5) % здесь синтаксис предыдущей команды
предполагает последовательную нумерацию графиков (первая тройка -
первый график и т.д.)
```

**Задание 6.** В новом окне построите графики функций  $y=\cos(x)$ ,  $f=(\cos(x))^2$  и  $g=x*\cos(2x)$ , на отрезке  $2*\pi \leq x \leq 2*\pi$ ; 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=-pi:0.1:2*pi
y=cos(x);
f=cos(x).^2;
g=eval(vectorize('x*cos(2*x)'));
YFG=[y' f' g'] % каждый столбец содержит ординаты
соответствующего графика
```

```

r=plot(x,YFG); % для x синтаксически не важно это вектор-строка
или столбец
%for k=1:3, set(r(k),'Color', [k*0.3 1-k*0.3 0.5]), end % если весь
оператор цикла помещаем в одну строку, то каждый синтаксический
элемент в нем отделяем запятой
% в ранних версиях MatLab mycolors{k} - с фигурными скобками
for k=1:3
    set(r(k),'Color',[k*0.3 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  $g=x*\exp(x)$ , сначала на интервале  $-\pi \leq x \leq \pi$ ; затем на  $-2*\pi \leq x \leq 2*\pi$ . Использовать одно графическое окно, а в нем организовать структуру из двух графиков, расположенных рядом.

```

%gcf %get current figure handle - работа с текущим графическим
окном, если его нет, создается новое графическое окно
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)

```
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
```

### Задание 11. Используем demos

Полный комплект демо-версий для изучения различных возможностей MatLab доступен, если набрать в командной строке demos. В командной строке наберите demos, откроется окно справки. В разделе Examples выберите Matlab-> 2D and 3D plots, откройте живой скрипт (Open Live script). Изучите примеры в скрипте.

В комментариях ответьте на вопросы:

- 1) Что делает команда linspace?
- 2) Выполните задание 10 с использованием команды **tiledlayout** вместо **subplot**.

### Задание 12. 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.