

# Лабораторная работа 6. Использование графиков. Решение и графический анализ нелинейных уравнений и систем

## Графический функционал, конструкторы для построения графиков

### 1) Конструктор `plot`

Значения функций, заданы векторами точек:

- `plot(X, Y, format)`  $X$  – вектор абсцисс,  $Y$  – вектор ординат графика функции, *format* – спецификация линии;
- `plot(X, Z)`  $X$  – вектор абсцисс,  $Z$  – матрица, векторы-столбцы – ординаты различных функций;
- `plot(Z)`  $Z$  – матрица, векторы-столбцы – ординаты различных функций, значения на оси абсцисс совпадают с номером элемента в векторе (номером строки);

### 2) Конструкторы `ezplot` и `fplot`

- $f$  – выражение функции, задано строкой: `f='x*sin(x)'`  
`ezplot(f, [xmin, xmax, ymin, ymax])`

если второй аргумент `ezplot` не указан, то по умолчанию считается, что  $x \in [-2\pi, 2\pi]$ , ( $y \in [-2\pi, 2\pi]$  в случае функции двух переменных); конструктор `ezplot` считается устаревшим, вместо него рекомендуется использовать `fplot`.

- $f$  – анонимная функция, например: `f=@(x) sin(x)/x`  
`fplot(f)`  
`fplot(f, [xmin, xmax])`

`[xmin, xmax]` – диапазон изменения аргумента, если интервал не указан, то по умолчанию  $x \in [-5, 5]$ . Для конструктора `fplot` функция может быть задана и строкой, но при таком использовании может быть выдано предупреждение, поэтому лучше задавать анонимную функцию.

### 3) Графики в текущих осях: `hold`

`hold on`

`hold off`

`hold on` добавляет все новые графики в текущие оси, `hold off` очищает предыдущие графики и настройки для осей. Для Matlab версии R2019b и выше см. также функции `tiledlayout` и `nexttile` для задания разного расположения осей (по типу subplot)

### 4) Две вертикальных оси на одном графике (можно использовать для двух функций с разным качественным поведением): `plotyy` и `yyaxis`

`plotyy(x1, Y1, x2, Y2)`

Команда `plotyy` считается *устаревшей*, вместо нее рекомендуется использовать `yyaxis` (см. Пример ниже).

`yyaxis left`

`yyaxis right`

`plot(x, y)`

При использовании двух вертикальных осей на одном графике обычные способы настройки цветов кривых (например, с помощью **set**) не изменяют цвет вертикальных осей. Для того чтобы цвета вертикальных осей соответствовали цветам кривых, используйте команду **colororder** для графического окна (т.е. после команды **figure**). Данная команда доступна в версии Matlab, начиная с R2019b. `colororder({'red','blue'})`

### Примеры

```
% Пример 1. График  $\phi$ -и на интервале [-5,5]
fplot(@(x) sin(x))
fplot('x*sin(x)') % поэлементные операции не требуются, но если
задавать функцию как строку, будет выдано предупреждение

% Пример 2. График параметрической кривой
xt = @(t) cos(3*t);
yt = @(t) sin(2*t);
fplot(xt,yt)

% Пример 3. Графики трех синусоид в одних осях разными цветами, с
заданием толщины линии
fplot(@(x) sin(x+pi/5), 'Linewidth',2);
hold on
fplot(@(x) sin(x-pi/5), '--or');
fplot(@(x) sin(x), '-.*c')
hold off

% Пример 4. Редактирование свойств линии после создания графика
fp = fplot(@(x) sin(x))
fp.LineStyle = ':';
fp.Color = 'r';
fp.Marker = 'x';
fp.MarkerEdgeColor = 'b';

% Пример 5. Графики двух функций с разным качественным поведением,
с двумя разными вертикальными осями
% plotyy
x = 0:0.01:20;
y1 = 200*exp(-0.05*x).*sin(x);
y2 = 0.8*exp(-0.5*x).*sin(10*x);
figure (51)
plotyy(x,y1,x,y2)

% yyaxis
x = 0:0.01:10;
y = sin(3*x);
z = sin(3*x).*exp(0.5*x);
figure (52)
colororder({'red','blue'})
yyaxis left
plot(x,y)
yyaxis right
plot(x,z)
```

```
ylim([-150 150])
```

### Задание 1

Постройте графики функций  $f=\sin(x)\cos^2(x)$  и  $g=\exp(\cos(2x))$ , на отрезке  $x \in [-2\pi, 2\pi]$ , сформировав предварительно вектор абсцисс и матрицу ординат двух функций.

Графики следует различать стилем линии и толщиной. Добавить легенду и сетку.

*Указание.* Использовать **plot**

### Задание 2

Постройте график  $y = x*\sin(x)$ , где функция  $y$  задана строкой, пунктирной линией, цвет линии содержит красного – 20%; зеленого - 50%; синего – 80%, нанесите сетку в осях, снабдите легендой.

*Указание.* Использовать **ezplot**, **set**.

### Задание 3

Построить график функции  $g = \exp(\sin(x))$  для  $x \in [-\pi, \pi]$ , где функция  $g$  – анонимная.

Интерактивно надписать «extreme» на графике у позиции экстремальной точки.

*Указание.* Использовать **fplot**, **gtext**.

### Задание 4

Фигуры Лиссажу в параметрической форме имеют вид:

$$x = \sin(mt), y = \cos(nt), t \in [-\pi, \pi].$$

Построить две фигуры Лиссажу (т.е. с двумя парами значений  $m$  и  $n$ ) в одних осях разным цветом, указать в легенде значения параметров  $m$ ,  $n$  для каждой из них. В заголовке указать: «Фигуры Лиссажу».

*Указание.* Использовать **fplot** или **polar**

### Задание 5

Построить графики функций  $y=\sin(x)$  и  $g=\sin(x)*\exp(x)$  в одних осях ( $x \in [-3\pi, 3\pi]$ ), так, чтобы качественное поведение одной функции не подавлялось ростом другой. Снабдить поясняющей информацией.

*Указание.* Использовать **plotyy** или **yyaxis**

## **Решение нелинейных уравнений или систем (определение точек пересечения линий)**

1) Поиск корня нелинейной функции **fzero**

```
x = fzero(fun, x0)
```

```
x = fzero(fun, x0, options)
```

**fzero** ищет корень функции **fun**, меняющей знак при пересечении оси абсцисс (т.е. корень функции  $x^2$  эта команда не найдет), в окрестности заданной точки  $x_0$ . Функция **fun** должна быть задана как анонимная (@) или как строка.

2) Решение системы нелинейных уравнений **fsolve** (для использования этой функции должен быть установлен модуль Optimization Toolbox).

```
x = fsolve(fun, x0)
```

```
x = fsolve(fun, x0, options)
```

**fsolve** ищет решение системы нелинейных уравнений, записанной в виде  $F(x) = 0$ , где  $F(x)$  – вектор-функция, например  $F(x)=(f_1(x), f_2(x))$ . Вектор-функция должна быть

задана как функция Matlab (см. примеры ниже). В случае, если ищется решение одного нелинейного уравнения, то  $F(x)$  как функцию Matlab задавать не требуется.

### Примеры

```
% Пример 1. Корень  $\phi$ -и  $\sin(x)$  в окрестности точки  $x=3$  (число  $\pi$ )
fun = @sin; % function
x0 = 3; % initial point
x = fzero(fun,x0)
```

```
% Пример 2. Корень  $\phi$ -и  $\cos(x)$  на заданном интервале [1,2]
fun = @cos; % function
x0 = [1 2]; % initial interval
x = fzero(fun,x0)
```

```
% Пример 3 Корень  $\phi$ -и  $x^2$  в окрестности точки  $x=0.1$ 
fzero('x*x',0.1) % выдаст NaN
```

```
% Пример 4 Система двух уравнений вида  $f_1(x)=0$ ,  $f_2(x)=0$ 
fun = @root2d;
x0 = [0,0]; %начальное приближение
x = fsolve(fun,x0)
```

```
function F = root2d(x)
F(1) = exp(-exp(-(x(1)+x(2)))) - x(2)*(1+x(1)^2);
F(2) = x(1)*cos(x(2)) + x(2)*sin(x(1)) - 0.5;
End
```

```
% Пример 5. Найти решение нелинейной системы как точку пересечения
двух кривых  $y_1=\exp(-x)*\sin(x)$ ,  $y_2=x-2$ ; отметить эти точки на
графике
```

```
y1=@(x)exp(-x)*sin(x) % можно задать как аноним
y2=@(x)x-2
%y2='x-2' % или как строку
```

```
fplot(y1,[-3,3]),hold on
fplot(y2,[-3,3],'r')
```

```
% Примерные координаты точек пересечения можно определить по
графикам. Это приблизительно -3, -1, 2.
```

```
% Для использования fzero нужно рассмотреть функцию вида  $y_2-y_1$  и
точки ее пересечения с осью абсцисс
```

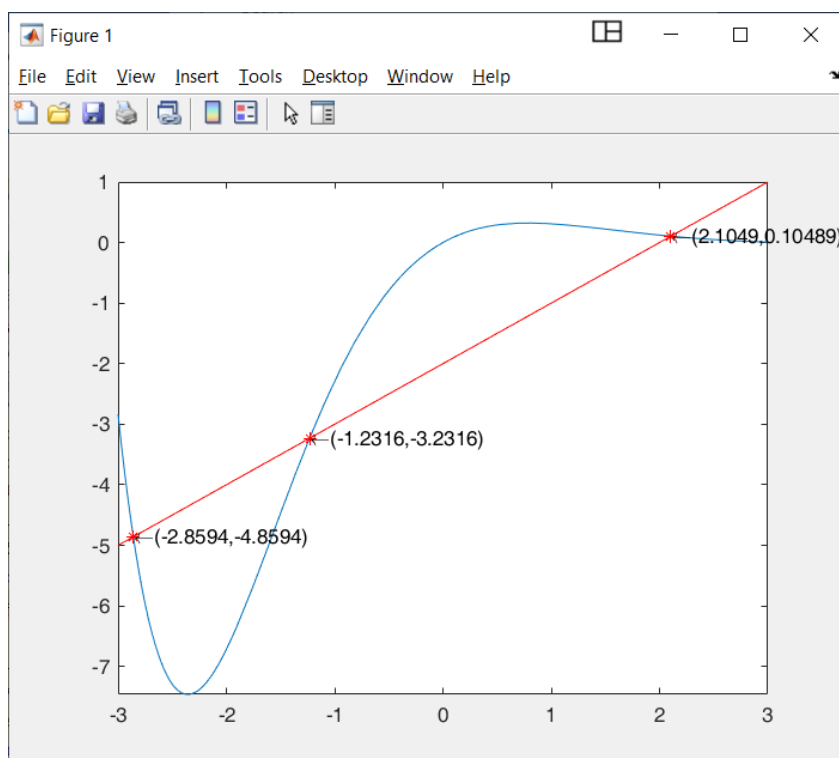
```
g='exp(-x)*sin(x)-(x-2) '
x0=fzero(g,-3)
x1=fzero(g,-1)
x2=fzero(g,2)
```

```
for x=[x0,x1,x2]
y=y2(x) % вычисление координаты  $y$ , если функция  $y_2$  задана как
аноним
% y=eval(y2) % если  $y_2$  задана как строка, то нужен eval
plot(x,y,'r*') % изображение точки пересечения в виде красной
звездочки
xy=['(',num2str(x),',',num2str(y),')']% по умолчанию num2str
выведет 4 знака после десятичной точки
%Чтобы вывести меньше знаков, указать второй аргумент в num2str
```

```

% num2str(x,2) % 2 знака после десятичной точки
text(x,y,['\leftarrow' xy ]) % текст с координатами точки
end

```



### Задание 6

Решить уравнение  $\sqrt{x+2} = 2-x$ . Для подбора интервалов поиска корней вывести график. В комментарии написать ответ (значения корней).

Использовать функции **fplot**, **fzero** или **fsolve**, подписать значения корней на графике функцией **text**.

### Задание 7

Найти все корни тригонометрического уравнения  $\sin^4 x - \cos^4 x = 1/2$  на интервале  $[-3,3]$ . Для подбора интервалов поиска корней вывести график. В комментарии написать ответ (значения корней).

Использовать функции **fplot**, **fzero** или **fsolve**, подписать значения корней на графике функцией **text**.

### Задание 8

Найти все точки пересечения двух линий  $y = x \cdot \sin(8x)$  и  $y = x^5 - x + 0.5$  на отрезке  $[-1,1]$ , предварительно определив абсциссы этих точек как корни функции  $g = x \cdot \sin(8x) - (x^5 - x + 0.5)$ . Отметить точки пересечения линий на графике (стиль маркера и размер выберите самостоятельно).

Использовать функции **fplot**, **fzero** или **fsolve**, подписать координаты точек пересечения функцией **text**.

### Задание 9

Найти все точки пересечения двух линий  $y = \sin(\exp(x))$  и  $y = 0.6x - 0.5$  на отрезке  $[0,3]$ , отметить эти точки на графике (стиль маркера и размер выберите самостоятельно).

Использовать функции **fplot**, **fzero** или **fsolve**, подписать координаты точек пересечения функцией **text**.

## Задание 10

Решить систему уравнений. Представить графическую интерпретацию решений. Использовать функции **fplot**, **fzero** или **fsolve**, подписать координаты точек пересечения функцией **text**.

$$\begin{cases} y = x * \sin(x) \\ y = 1 - x^2 \end{cases}$$

### Полиномы

- 1) Конструктор полинома по его коэффициентам **poly2sym** (команда из модуля Symbolic Math Toolbox)

**p = poly2sym(c)**

**p = poly2sym(c, var)**

**poly2sym** создает символьный полином **p** из вектора коэффициентов **c**. Если **c = [c1,c2,...,cn]**, то команда **p = poly2sym(c)** создаст полином вида  $c_1x^{n-1}+c_2x^{n-2}+\dots+c_n$ . По умолчанию имя переменной полинома – **x**, но можно задать другое имя в переменной **var**.

```
p = poly2sym(sym([1/2, -1/3, 1/4]))
```

```
p = poly2sym([0.75, -0.5, 0.25])
```

- 2) Получение коэффициентов полинома **sym2poly** (команда из модуля Symbolic Math Toolbox)

**c = sym2poly(p)**

**sym2poly** выдает вектор коэффициентов **c** с символьного полинома **p**. Возвращаемый вектор **c = [c1 c2...cn]** будет содержать все коэффициенты, включая нулевые. Если  $p=c_1x^{n-1}+c_2x^{n-2}+\dots+c_n$ , то команда **c = sym2poly(p)** возвратит **c = [c1 c2 ... cn]**.

```
syms x
```

```
c = sym2poly(1/2*x^3 - 2/3*x - 5)
```

- 3) Вычисление корней (в том числе комплексных) для полинома, заданного своим коэффициентами: **roots**

**r = roots(p)**

**roots** возвращает корни полинома, где **p** – вектор всех коэффициентов полинома, начиная с коэффициента при старшей степени: **roots ([pn,...,p1,p0])** для полинома вида  $p_nx^n+ p_{n-1}x^{n-1}+\dots+p_1x+p_0=0$ .

Решение уравнения  $3x^2-2x-4=0$ :

```
p = [3 -2 -4];
```

```
r = roots(p)
```

- 4) Вычисление коэффициентов полинома по заданным корням: **poly**

**p = poly(r)**

**p = poly(A)**

**poly** возвращает коэффициенты полинома **p** по вектору корней **r**, или характеристический многочлен для матрицы **A**. Для корректного определения полинома каждый корень перечисляется с учетом его кратности:

```
y=poly([1, 1, 3, 7, 9])
```

5) Вычисление значения полинома, заданного своими коэффициентами, в заданной точке: **polyval**

**y = polyval(p, x)**

**polyval** выдает значение полинома, заданного вектором коэффициентов p, в точке x. Вычисление значения полинома  $p(x)=3x^2+2x+1$  в точках  $x=5, 7, 9$ .

```
p = [3 2 1];
```

```
x = [5 7 9];
```

```
y = polyval(p, x)
```

### **Задание 11**

Найти все корни полинома пятого порядка  $p(x)=x^5+x^2-10x-4.5$ . Обозначить полученные вещественные корни на графике  $p(x)$ , выделив их размером и формой маркера, подписать их значения на графике. Определить точность, с которой найдены корни.

*Указание.* Использовать **sym2poly**, **roots**. Вещественные корни найти с помощью **fzero**. Задать точность для **fzero** можно в параметре **options**:

```
options=optimset('TolX',1e-6);
```

```
x1=fzero(p, x0, options)
```

Сравните со значениями, полученными с точностью, заданной по умолчанию (для этого вывести значения в длинном формате: `format long`).

### **Задание 12**

Найти вид полинома пятого порядка, если известны его корни: 1 - кратности 2; а также простые корни: 3, 7, 9. Построить график функции полученного полинома. Выделить корни и подписать их значения на графике.

*Указание.* Использовать **poly**, **poly2sym**.