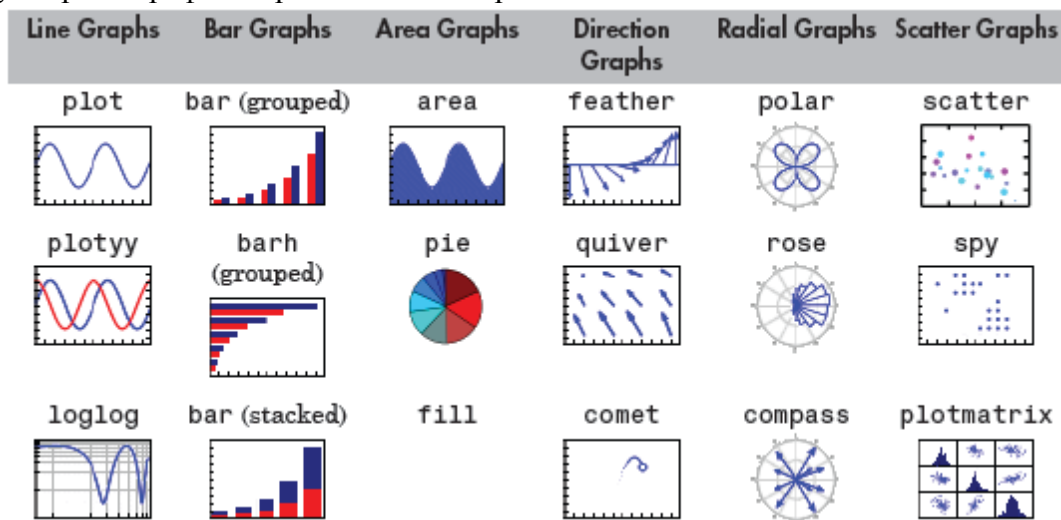


Графики функций (plot, fplot, ezplot)

Пакет MATLAB располагает большими возможностями для визуализации результатов вычислений. Прежде всего, это вывод численных результатов в виде диаграмм, графиков плоских и пространственных фигур, трехмерных поверхностей и некоторых геометрических фигур. Пользователю предоставлена возможность строить графики и диаграммы в приложении, а в случае необходимости редактировать их в графическом редакторе. Некоторые из вариантов двумерной графики представлены на рис.:



Графики функции одной переменной

Для вывода графика функции $y = f(x)$ необходимо сформировать вектор точек, в которых вычисляется функция $f(x)$, и сформировать вектор значений функции в этих точках. После этого следует обратиться к функции **plot**, которая может вывести несколько графиков в одно окно:

```
x=[0:0.02:1]; %формируем вектор точек для ф-ции y
y=x.*(1-x)+0.1; %задаем функцию y
x1=[-0.5:0.05:1.5]; %вектор точек для ф-ции y1=(x1+0.5)/5)
plot(x,y,x1,(x1+0.5)/5) %вывод графиков двух ф-ций y и y1
```

После выполнения этих команд MATLAB сформирует (если нет других окон) окно с заголовком **Figure 1**, разместит в нем стандартное меню и линейку инструментов, выделит в окне прямоугольное поле с графиками функций, сделает разметку осей (рис. 1). В том случае, если уже есть какие-нибудь графические окна, то функция **plot** будет выводить графики в текущее графическое окно. Новое окно для вывода графиков можно вызвать командой **figure**.

Функция **plot** перед выводом графика очищает текущее графическое окно (если оно есть). После выполнения в командной строке (или в программе пользователя) команды **hold on** все последующие графики, формируемые функцией **plot**, будут выводиться на уже существующие в графическом окне оси. Отмена режима "наложения" графиков осуществляется командой **hold off**. Масштабно-координатная сетка наносится командой **grid on**, а убирается командой **grid off**. Подписи к осям делают функции **xlabel** и **ylabel**, а заголовок – **title**. Аргументом этих функций является текстовая переменная или последовательность символов в апострофах. Рис. 1 соответствуют команды

```
grid on;
xlabel('Time');
ylabel('Function');
```

```
title('Graphics')
```

Для создания легенды используется функция `legend('Name1','Name2',...)`. Число аргументов у этой функции должно соответствовать числу линий на графике. Легенде на рис. 1 соответствует команда `legend('f1','f2',-1)`. Более подробно о дополнительных возможностях функции **legend** можно ознакомиться в справочной системе.

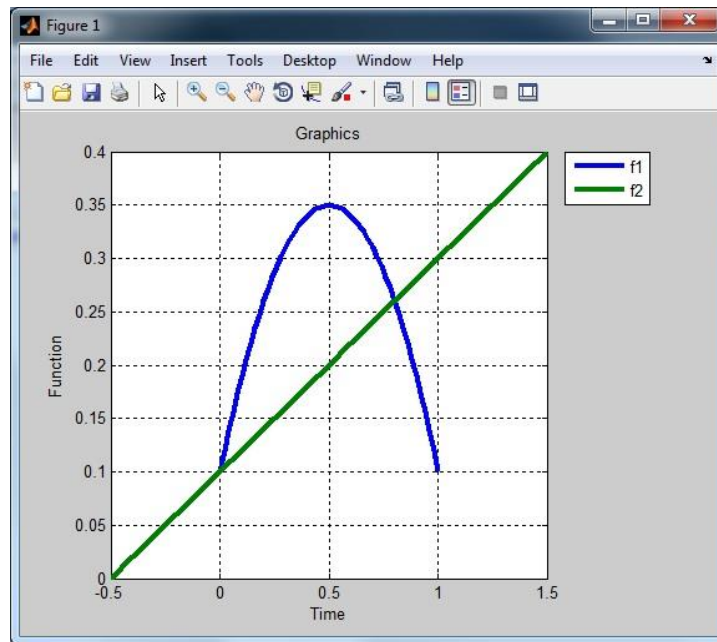


Рис. 1.

Синтаксис обращения к функции **plot**:

```
plot(x,y)
plot(x, y, LineSpec, 'Property', Value)
Lines = plot(x, y, LineSpec, 'Property', Value)
```

В переменной `LineSpec` в апострофах указываются в соответствии с табл. 1 цвет линии, тип маркера и тип линии. Например, для сплошной линии красного цвета с маркером "кружок" `LineSpec` будет иметь вид `'ro-'`. В переменной `LineSpec` задаются свойства линии, задаваемой парой векторов (x,y) , за которой она стоит в обращении к функции `plot`. Указатель на `LineSpec` не является обязательным и может для некоторых линий отсутствовать.

Свойства линии `Property` приведены в табл. 2. Часть из них можно описать в переменной `LineSpec`. Более подробное описание можно найти в справочной системе. Свойство линии `Property` задается одновременно для всех линий, формируемых функцией `plot`. Использование команды **hold on** позволяет выводить графики поочередно с требуемыми свойствами линий и маркеров.

Например, после выполнения команд

```
x=linspace(0,1,50);
x1=[-0.5:0.05:1.5];
y=x.*(1-x)+0.1;
plot(x,y,'xr',x1,(x1+0.5)/5,'b','LineWidth',2,'MarkerSize',8)
grid on
```

В текущее графическое окно будет выведен график функции $y = x(1 - x) + 0.1$, заданной на промежутке $[0, 1]$ в виде линии красного цвета с маркером на ней в виде крестика ('xr'), и график функции $y = (x_1 + 0.5)/5$, заданной на промежутке $[-0.5, 1.5]$ в виде линии синего цвета ('b') (рис. 2). Толщина линий 2pt, размер маркера 8pt.

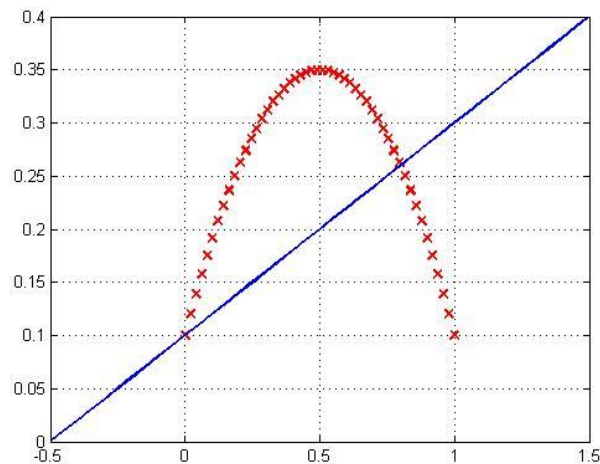


Рис. 2.

Графическое окно

Окно для вывода графиков является графическим объектом, которому можно присвоить имя. Задается оно с помощью функции **figure**:

Fig=figure('PropertyName', PropertyValue, ...).

Эта же функция при наличии нескольких окон делает окно **Fig** текущим:

figure(Fig)

Описание и свойства окна ('PropertyName') приведены в табл. 3.

Пример.

Последовательность команд

```
Poz=[20 40 620 540];
Fig=figure('Position',Poz,'NumberTitle','off','Name','Graphics')
x=linspace(0,2*pi,500);
y=exp(-0.5*x).*sin(4*pi*x);
z=exp(-0.5*x);
plot(x,y,x,z,':r',x,-z,':r','LineWidth',2)
grid on
```

создает графическое окно с именем Fig шириной 600 пикселей и высотой 500 пикселей, с надписью Graphics в верхнем правом углу и отсутствующим номером. Левый нижний угол окна находится на расстоянии 20 пикселей от левого нижнего угла монитора и на высоте 40 пикселей. Затем формирует функции $y = e^{-0.5x} \sin(4\pi x)$ и $z = e^{-0.5x}$ на промежутке $[0, 2\pi]$ и выводит график функции $y = e^{-0.5x} \sin(4\pi x)$ в виде сплошной линии синего цвета и графики функций $z = e^{-0.5x}$ и $z = -e^{-0.5x}$ в виде пунктирных линий красного цвета в графическое окно Fig (рис. 3).

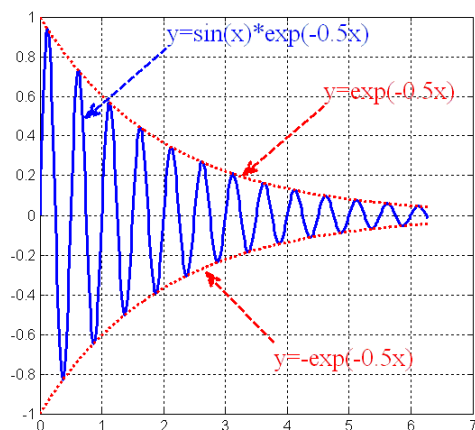


Рис. 3.

Табл. 1.

Цвет*		Тип маркера		Тип линии	
y	желтый	•	точка	—	сплошная
m	розовый	○	кружок	:	пунктирная
c	голубой	x	крестик	—.	штрих-пунктирная
r	красный	+	знак "плюс"	— —	штриховая
g	зеленый	*	звездочка	пробел	без линии
b	синий	s	квадрат		
w	белый	d	ромб		
k	черный	p	пятиконечная звезда		
		h	шестиконечная звезда		

* либо вектор [R G B]

Табл. 2.

Название свойства	Назначение	Значение
Color	Задает цвет линии	
LineStyle	Задает тип линии	
LineWidth	Определяет толщину линии в пунктах	Положительное число
Marker	Задает тип маркера	Табл. 1
MarkerEdgeColor	Задает цвет границы маркера	Табл. 1
MarkerFaceColor	Задает цвет маркера	Табл. 1
MarkerSize	Задает размер маркера в пунктах	Положительное число

Табл. 3.

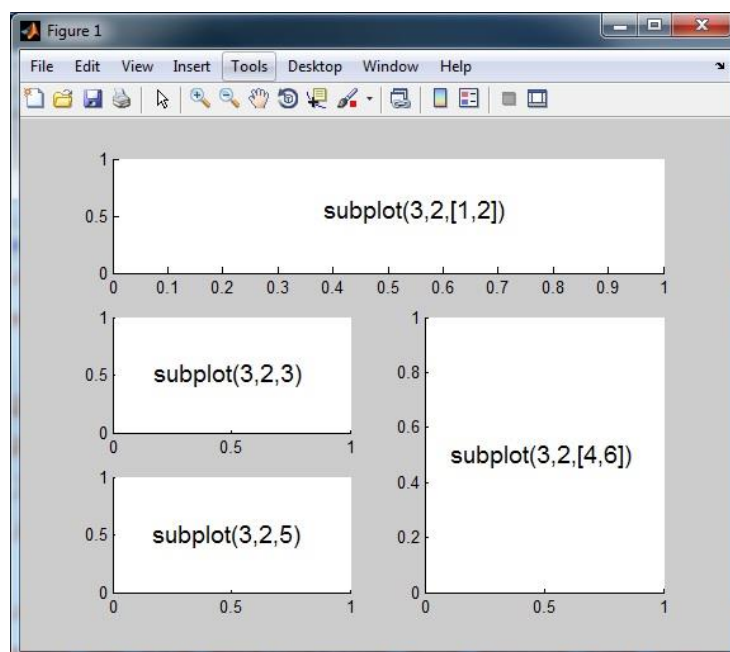
Свойство	Назначение	Значение
----------	------------	----------

Position	Указывает на местоположение и размеры окна на экране	Вектор-строка из четырех чисел: координаты левого нижнего угла окна, его ширина и высота
MenuBar	Задаёт режим вывода меню графического окна	'figure' 'non'
Name	Указывает на необходимость добавить текст в название графического окна	Последовательность символов в апострофах
NumberTitle	Определяет вывод номера в названии фигуры	'on' 'off'

В одном графическом окне можно создать нескольких осей с помощью функции **subplot**. Варианты синтаксиса обращения к этой функции:

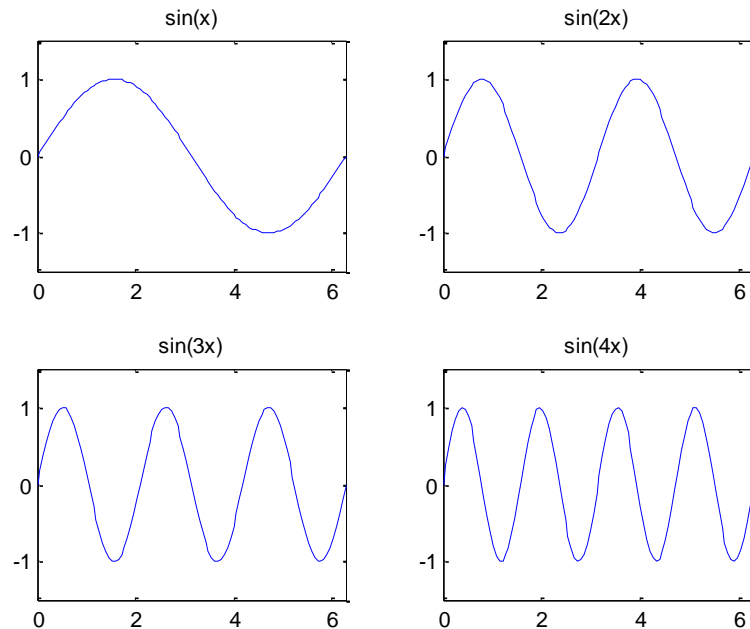
```
subplot(n,m,k);
subplot(n,m,k, 'replace');
subplot(n,m,[k l]);
Name=subplot(...)
```

Эта функция в графическом окне формирует матрицу для вывода графических осей размерами $m \times n$ (m строк, n столбцов) с физическими номерами ячеек от 1 до $m \times n$ (слева направо и сверху вниз). При обращении **subplot(n,m,k)** в ячейке k сформируются оси, на которые можно вывести графики. При обращении **subplot(n,m,k, 'replace')** в k -ой ячейке графического окна уничтожаются старые оси. После выполнения команды **subplot(n,m,[k l])** графические оси выводятся в k -ую и l -ую ячейки одновременно. На рис. приведен пример формирования в графическом окне шести "ячеек" (три строки и два столбца – **subplot(3,2,...)**). После выполнения команды **subplot(3,2,[1 2])** объединяются первая и вторая ячейки, а после выполнения команды **subplot(3,2,[4 6])** – четвертая и шестая.



Пример:

```
x = linspace(0,2*pi);  
subplot(2,2,1);  
plot(x,sin(x)); axis([0 2*pi -1.5 1.5]); title('sin(x)');  
subplot(2,2,2);  
plot(x,sin(2*x)); axis([0 2*pi -1.5 1.5]); title('sin(2x)');  
subplot(2,2,3);  
plot(x,sin(3*x)); axis([0 2*pi -1.5 1.5]); title('sin(3x)');  
subplot(2,2,4);  
plot(x,sin(4*x)); axis([0 2*pi -1.5 1.5]); title('sin(4x)');
```



Варианты задания осей координат: **axis square**, **axis tight**, **axis equal**, **axis xy**, **axis ij**.

Подробная справка по функции **plot** находится по адресу:
<http://www.mathworks.com/help/matlab/ref/plot.html>

Построение графика вектор-функции

График функции одной переменной, описанной в файл-функции **Name.m**, на отрезке $[x_{min}, x_{max}]$ строит также и функция **fplot**:

```
fplot('Name', [x_min, x_max], tol, LineSpec)
```

где **tol** – точность, **LineSpec** – спецификация кривых (толщина линии, маркер, цвет – табл. 1).
tol и **LineSpec** – не обязательные аргументы.

При обращении

```
[x,y]=fplot('Name', [x_min, x_max], tol, LineSpec)
```

в *Workspace* формируются вектор-столбец **x** узловых точек и вектор-столбец **y** значений функции в этих точках.

Функция **fplot** может построить несколько графиков, если выходным аргументом функции **Name** является вектор:

```
function F=Name(x)  
F=[f1(x), ..., fn(x)];
```

По умолчанию первый график рисуется синим цветом, второй – зеленым, третий – красным.

Пример. Построить графики функций $f_1(x) = \sin x$ и $f_2(x) = x^2 - x - 1$ на промежутке $[0, \pi]$.

Вариант решения:

Описание файл-функции, вычисляющей f_1 и f_2 :

```
function F=Fun(x)
F=[sin(x), x.*x-x-1];
```

Графики будут построены после выполнения команды

```
fplot('Fun', [0, pi])
```

или

```
fplot(@Fun, [0, pi])
```

Если заранее не задавать файл-функцию, вычисляющую f_1 и f_2 , то график может быть построен после выполнения команды

```
fplot(' [sin(x), x^2-x-1] ', [0, pi])
```

Построение графика функции, заданной в неявном виде

График функции, заданной уравнением $f(x, y) = 0$, в прямоугольнике $[x_{min}, x_{max}, y_{min}, y_{max}]$ (по «умолчанию» $[-2\pi, 2\pi, -2\pi, 2\pi]$) строится с помощью функции **ezplot**:

```
ezplot(@(x,y) f(x,y), [x_min, x_max, y_min, y_max])
```

(допустимы и другие варианты обращения к ф-ции **ezplot**, см. **help**)

Пример. Построить конхоиду Никомеда, задаваемую уравнением $(x - a)^2(x^2 + y^2) = b^2x^2$ с параметрами $a = 0.4$ и $b = 1$ в прямоугольнике $[-1, 2, -4, 4]$:

Вариант решения:

```
a=0.4; b=1;
```

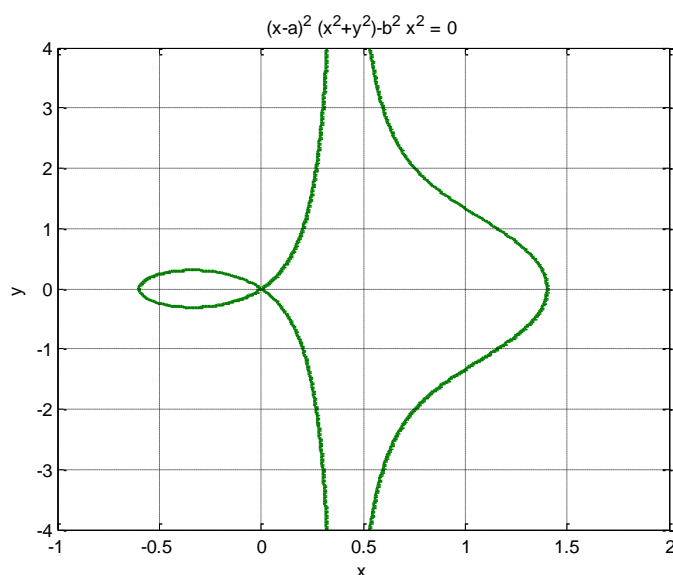
```
F=@(x,y) (x-a).^2*(x.^2+y.^2)-b^2*x.^2;
```

```
C=ezplot(F, [-1 2 -4 4]);
```

```
grid on
```

```
set(C, 'LineWidth', 3, 'Color', [0 0 1]) %задание свойств линии
```

График конхоиды приведен на рис.



Иногда требуется сравнить поведение двух функций, значения которых сильно отличаются друг от друга. График функции с небольшими значениями практически сливается с осью абсцисс, и установить его вид не удастся. В этой ситуации помогает функция **plotyy**, которая выводит

графики в окно с двумя вертикальными осями, имеющими подходящий масштаб. При этом цвет графика совпадает с цветом соответствующей ему оси ординат. Синтаксис:

```
plotyy(x1,y1,x2,y2)
plotyy(x1,y1,x2,y2,'function1','function2')
```

Пример:

```
x = 0:0.01:20;
y1 = 200*exp(-0.05*x).*sin(x);
y2 = 0.8*exp(-0.5*x).*sin(10*x);

figure % новое графическое окно
[hAx,hLine1,hLine2] = plotyy(x,y1,x,y2);

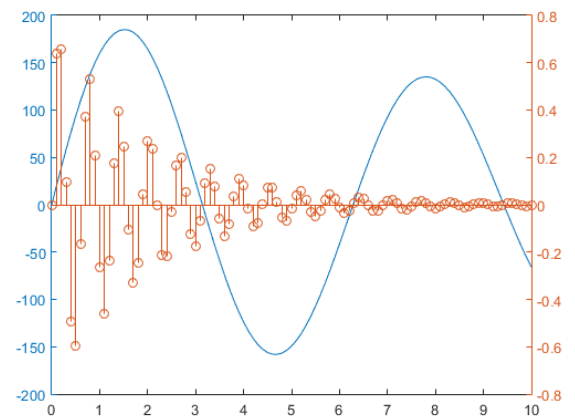
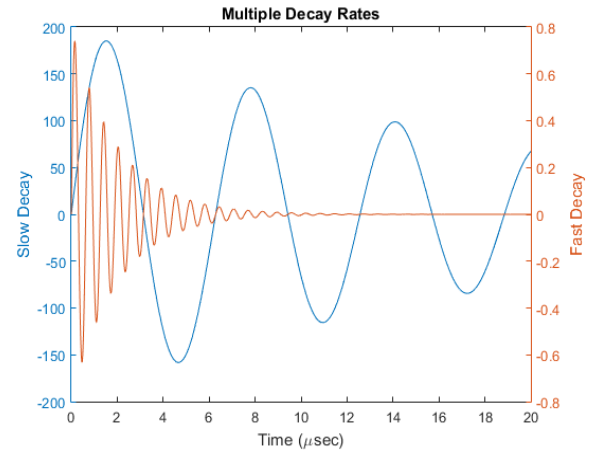
title('Multiple Decay Rates')
xlabel('Time (\musec)')

ylabel(hAx(1),'Slow Decay') % левая ось y
ylabel(hAx(2),'Fast Decay') % правая ось y

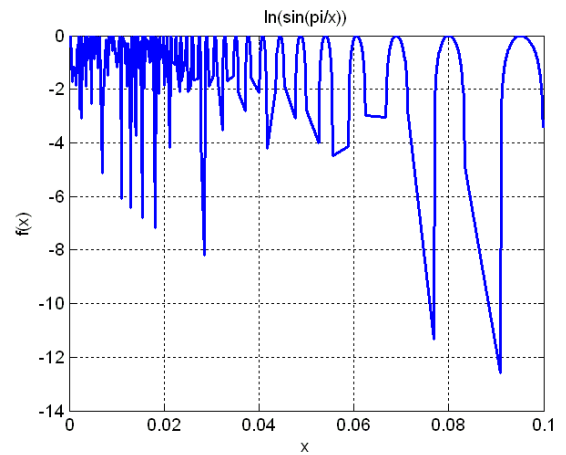
%{B версиях до R2015a
set(get(ax(1), 'YLabel'), 'String', 'Low
Frequency')
set(get(ax(2), 'YLabel'), 'String', 'High
Frequency')
%}

hLine1.LineStyle = '--'; % обращение к
элементу объекта через «.» - начиная с
версии R2015a. До этого – функция set.
hLine2.LineStyle = ':'; % верно начиная с
версии R2015a

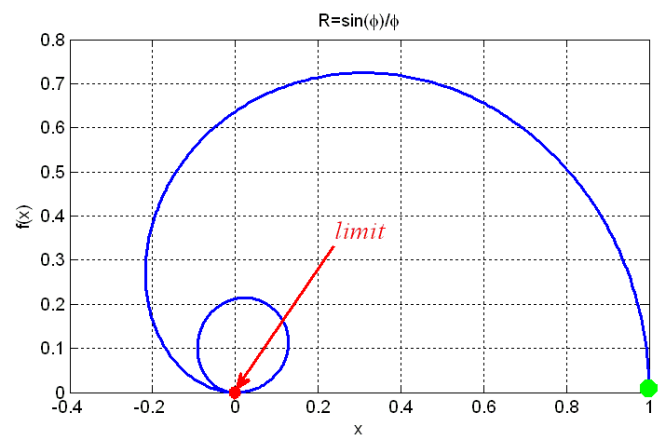
plotyy(x,y1,x,y2,'plot','stem') % комбинация разных стилей графиков
```



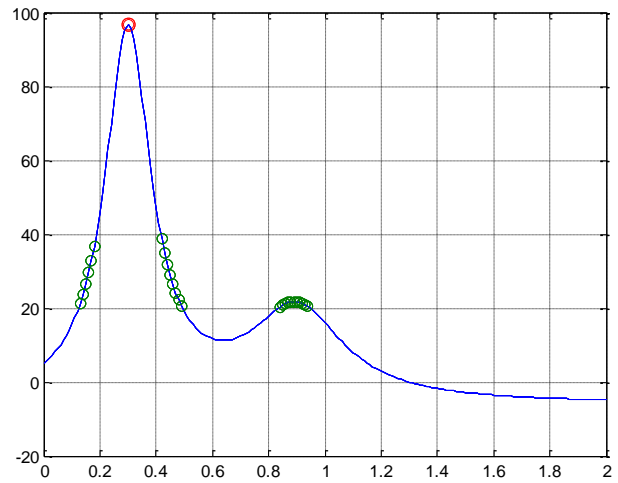
Пример 1: Построить график функции $y = \ln\left(\sin\frac{\pi}{x}\right)$



Пример 2: Построить график функции $r = \frac{\sin \varphi}{\varphi}$ в ДПСК



Пример: построить график встроенной функции `humps(x)`. Указать на графике точку, соответствующую максимальному значению функции. Указать на графике множество значений функции, лежащих в интервале $[20, 40]$.



Задание для самостоятельной работы: построить графики функций:

1. $y = x + \frac{1}{x^2}$

2. Серпантин Ньютона $y = \frac{2x}{1+x^2}$,

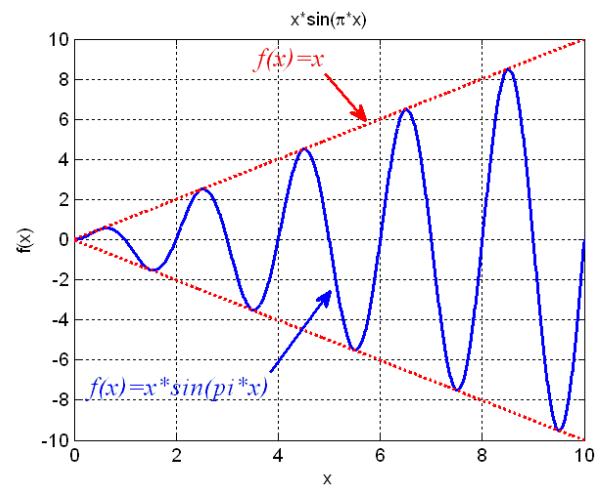
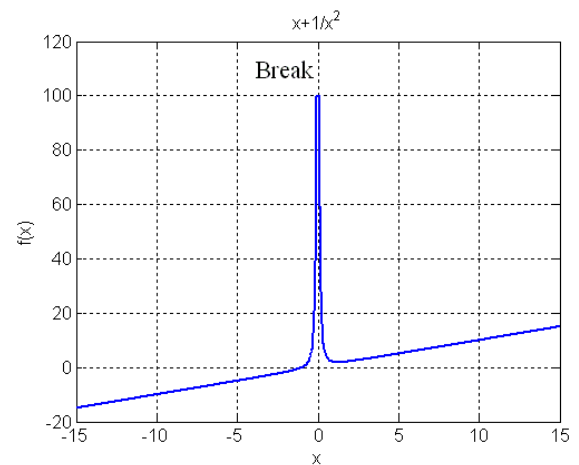
3. $y = (x-2)\sqrt{\frac{1+x}{1-x}}$,

4. $y = \ln(x^2 - 4)$,

5. $y = \sqrt{\sin(\pi\sqrt{x})}$,

6. $y = (x+|x|)\sqrt{x\sin^2 \pi x}$

7. $y = x \cdot \sin \pi x$



8. $r = \frac{a}{a + (\varphi - \pi/2)^n} (b - \sin k\varphi - \cos m\varphi)$, $\varphi \in [-\pi/2, 3\pi/2]$

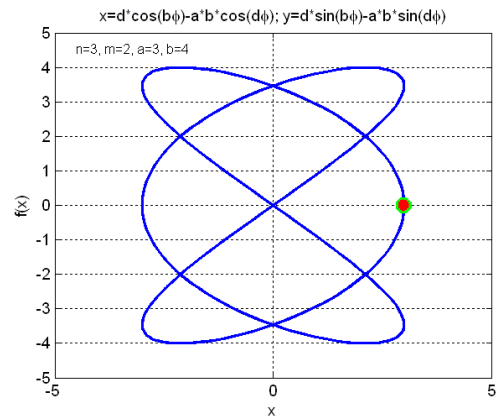
$$a=100, b=2, n=4, k=14.$$

9.

$$x = \left(1 + \frac{n}{m}\right) \cos \frac{n}{m} \varphi - a \frac{n}{m} \cos \left(1 + \frac{n}{m}\right) \varphi,$$

$$y = \left(1 + \frac{n}{m}\right) \sin \frac{n}{m} \varphi - a \frac{n}{m} \sin \left(1 + \frac{n}{m}\right) \varphi,$$

$$\varphi \in [0, 2m\pi]$$



10.

$$x = a \sin(n\varphi + \varphi_0),$$

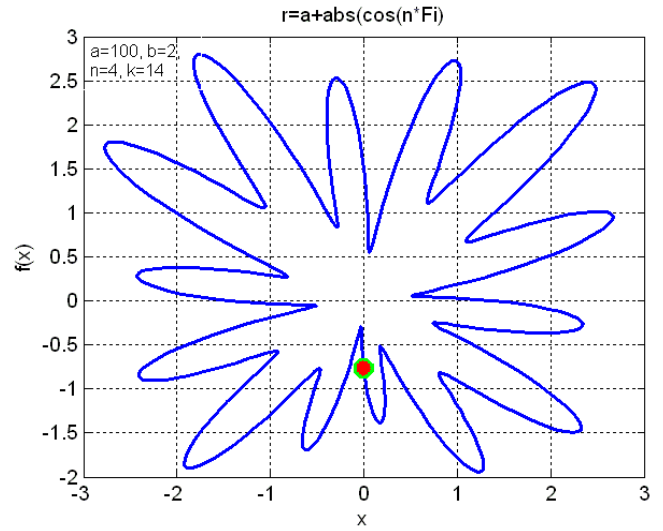
$$y = b \sin(m\varphi).$$

11.

$$x = \cos t + a \cos mt + b \sin nt$$

$$y = \sin t + a \sin mt + b \cos nt,$$

$$t \in [-5, 5], a = \frac{1}{2}, b = \frac{1}{3}, m = 7, n = 17$$



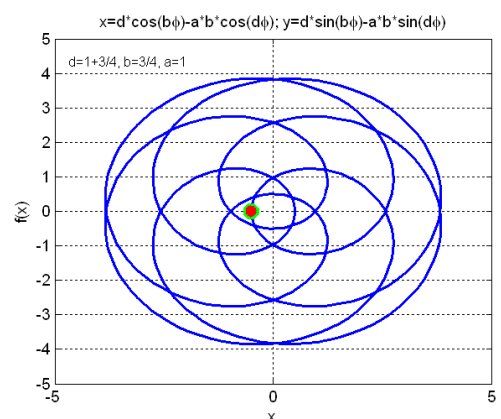
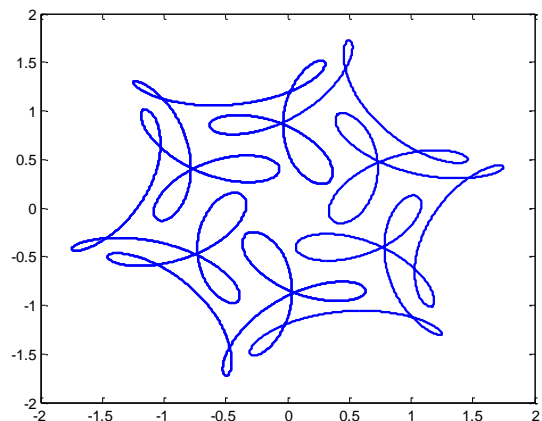
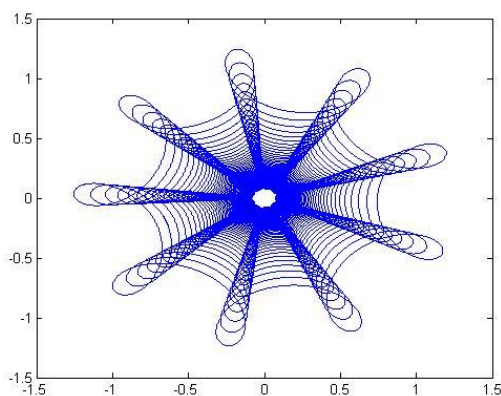
12. Построить множество точек $\left(\frac{x}{k}, \frac{y}{k}\right),$

$$k \in [1, s] - \text{целые числа.}$$

$$x = \cos t - a \cos mt + b \sin nt$$

$$y = \sin t + a \sin mt + b \cos nt,$$

$$t \in [-5, 5], a = \frac{1}{4}, b = \frac{1}{16}, m = 8, n = 8.$$



13. Построить графики функций:

$y = \frac{1}{x} - \frac{1}{x-1} + \frac{1}{x-2}$	$r = a \cos \varphi - b$	$x = \cos n\varphi \cos^m \varphi,$
$y = \frac{2x}{1+x^2}$	$r = 1/\varphi$	$y = \sin n\varphi \sin^m \varphi.$
$y = \sqrt{\cos(\pi x^2)}$	$x = a(b \cos \varphi + \varphi \sin \varphi),$	$n = 2, m = 2.$
$y = \frac{\sqrt{x}}{\sin \pi x}$	$y = a(\sin \varphi - c\varphi \cos \varphi).$	$x = \sin m\varphi \cos(n\varphi + \varphi_0),$
$y = \arccos(2 \sin x)$	$a = 2; b = 1; c = 1;$	$y = \sin m\varphi \sin(n\varphi + \varphi_0).$
$y = \operatorname{ctg} \pi x + \arccos(2^x)$	$a = 2; b = 1; c = 0.1;$	$\varphi \in [0, 2\pi]$
$y = e^{-0.2x} \sin x$	$r = \sin \frac{k}{2n-1} t, \quad t \in [0, 2\pi(2n-1)]$	$\varphi_0 = \pi/4, n = 1, m = 2.$
$y = e^{0.2x} \cos x$	$k = 2, \quad n = 2$	$x = \sin m\varphi \cos(n\varphi + \varphi_0),$
$y = x^2 \sin^2 \pi x$	$R = a + \cos n\varphi , \quad \varphi \in [-\pi, \pi]$	$y = \cos m\varphi \sin(n\varphi + \varphi_0).$
	$a = 1, \quad n = 4$	$\varphi \in [0, 2\pi]$

14. Найти, каким уравнением задаётся кривая и построить график этой кривой:

- | | |
|--|--|
| 1) лемниската Бернулли | 19) нефроида |
| 2) Розы Гранди | 20) строфоида |
| 3) улитка Паскаля | 21) циссоида Диоклеса |
| 4) полукубическая парабола
(парабола Нейля) | 22) спираль Ферма (обе ветви) |
| 5) спираль Архимеда (три витка вокруг
начала координат) | 23) спираль Галилея |
| 6) трезубец Ньютона | 24) кривая <i>лист щавеля</i> |
| 7) конхоида Никомеда | 25) трисектриса Маклорена |
| 8) инволюта окружности | 26) трилистник |
| 9) Декартов лист | 27) кардиоида |
| 10) локон Аньези | 28) эвольвента правильного
треугольника. |
| 11) цепная линия | 29) циклоида |
| 12) эвольвента окружности | 30) Бабочка Темпла Фея |
| 13) эвольвента квадрата | 31) Гипоциклоида |
| 14) гиперболическая спираль | 32) овал Кассини |
| 15) трактриса | 33) лист Хабенихта |
| 16) спираль Кейли | $R(t) = 1 + 7 \cos at + 4(\sin at)^2 + 3(\sin at)^4$ |
| 17) логарифмическая спираль | $, a = 8.$ |
| 18) дельтоид | |