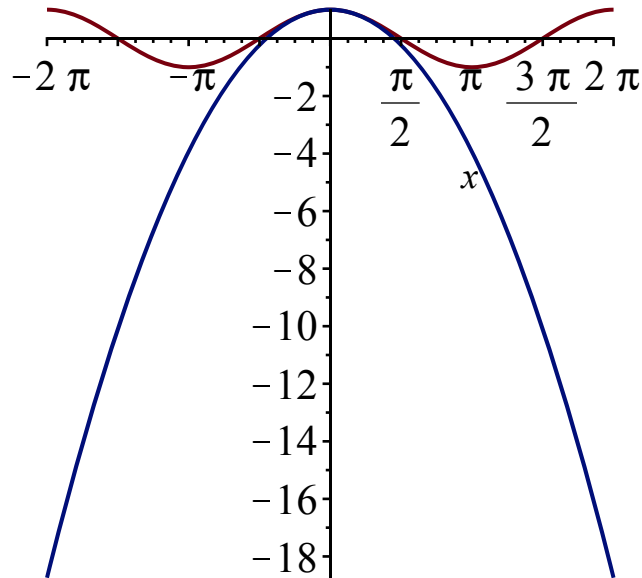


## Графики нескольких функций на одном рисунке

Сформулированная в заголовке этого документа задача допускает достаточно простое решение, если заметить, что первый параметр команды **plot** может представлять собой список.

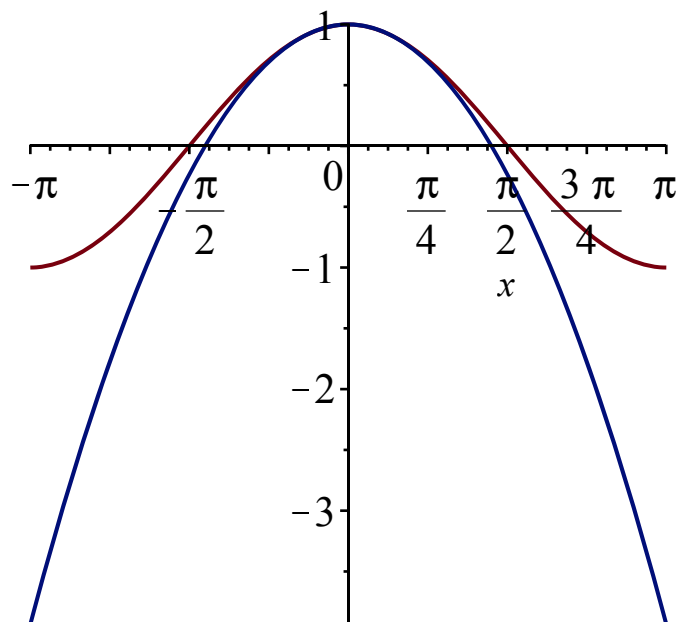
```
> plot([cos(x), 1-x^2/2]);
```



Каждый элемент списка трактуется как отдельная, явно заданная функция и все графики строятся на одном рисунке. Их параметры система настраивает по умолчанию.

Как и в случае единственного графика, рекомендуется указывать диапазон изменения аргумента. Это позволяет сосредоточиться на тех частях рисунка, которые представляют наибольший интерес.

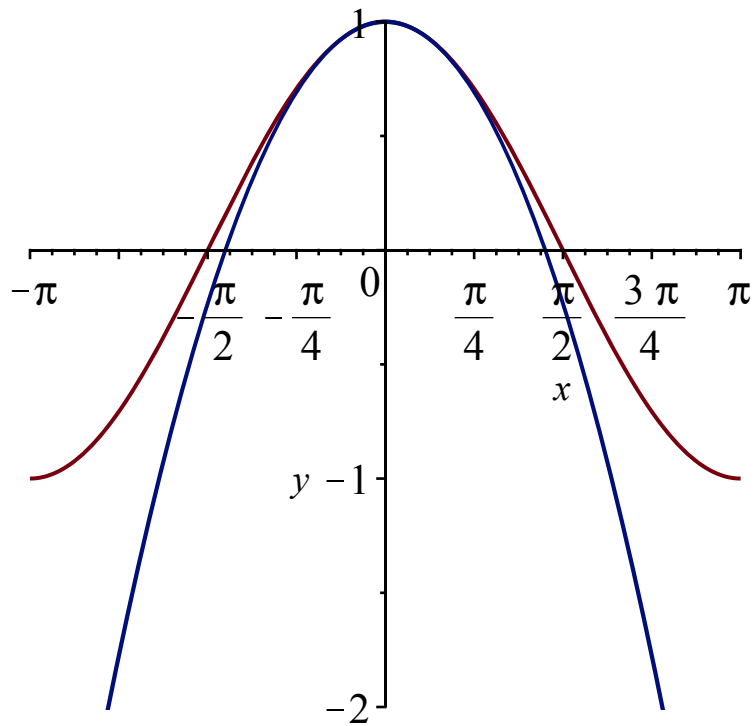
```
> plot([cos(x), 1-x^2/2], x=-Pi..Pi);
```



В данном случае визуально легко оценить размеры той области, в которой два первых члена степенного ряда хорошо аппроксимируют функцию  $\cos(x)$ .

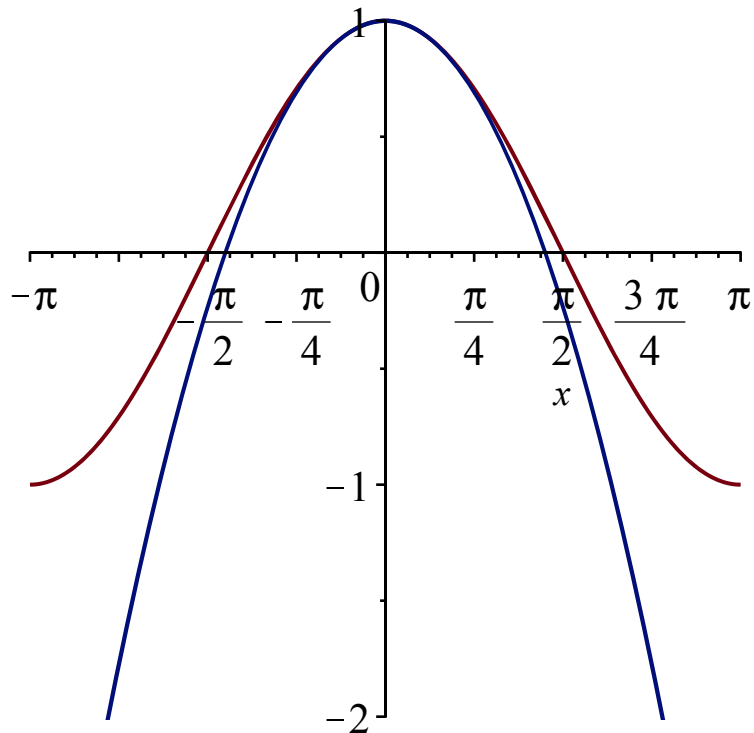
Ненужная часть графиков по оси ординат отсекается аналогичным образом путём добавления третьего аргумента в команду **plot**.

```
> plot([cos(x), 1-x^2/2], x=-Pi..Pi, y=-2..1);
```



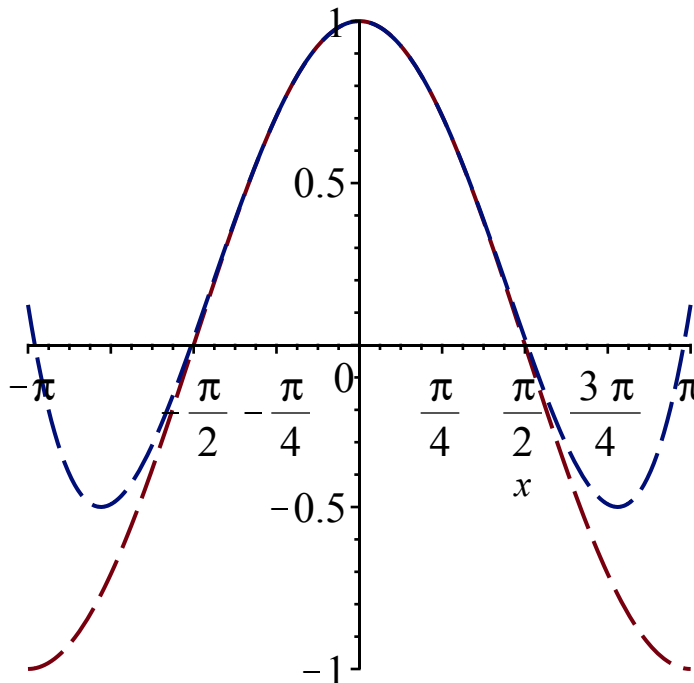
Последний аргумент можно задавать не только в виде уравнения, но и как обычный диапазон. В этом случае ось ординат остаётся без подписи.

```
> plot([cos(x), 1-x^2/2], x=-Pi..Pi, -2..1);
```



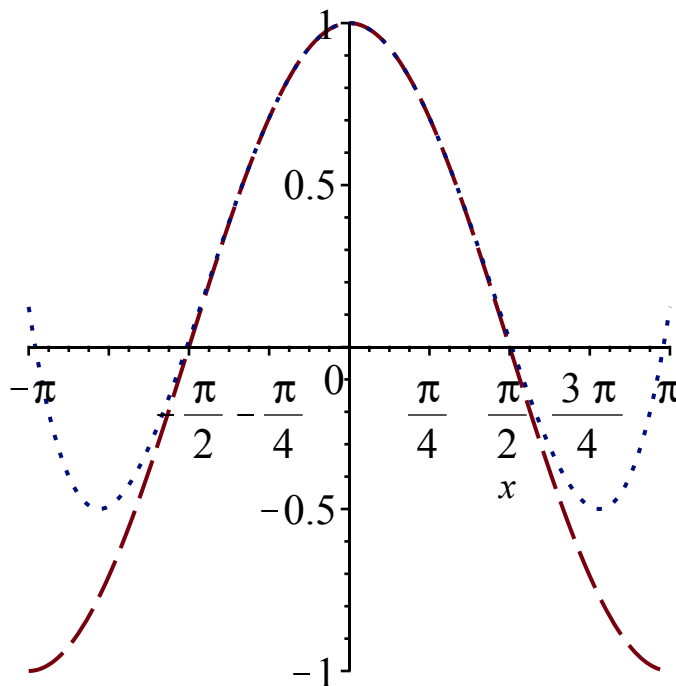
Если дополнительные опции команды **plot** использовать в том же виде, как и при построении графика только одной функции, то эти опции будут применены сразу ко всем графикам.

```
> plot([cos(x), 1-x^2/2+x^4/24], x=-Pi..Pi, linestyle=dash);
```



К счастью, их тоже можно задавать в виде списка и настраивать каждый график индивидуально.

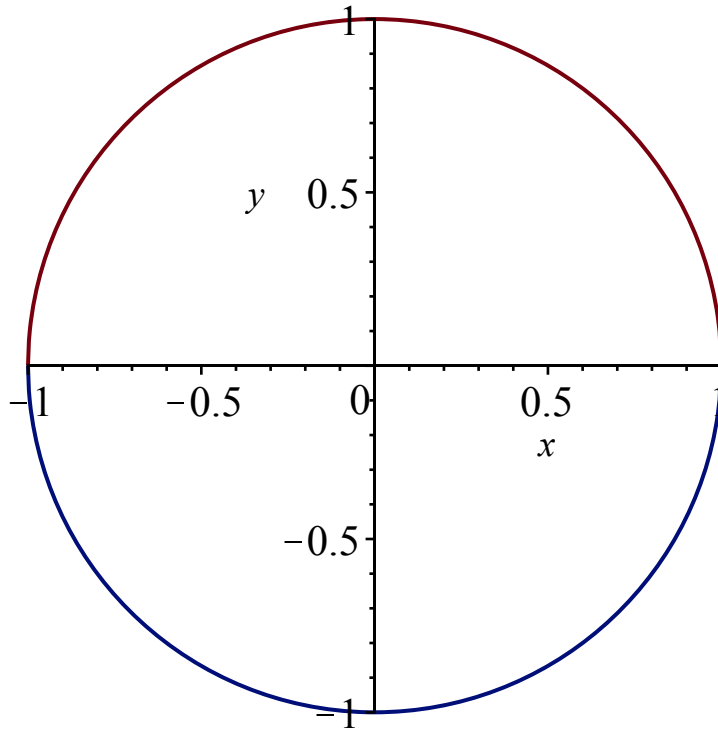
```
> plot([cos(x), 1-x^2/2+x^4/24], x=-Pi..Pi, linestyle=[dash, dot]);
```



Первый аргумент команды **plot** можно задавать не только в виде списка, но и как множество.

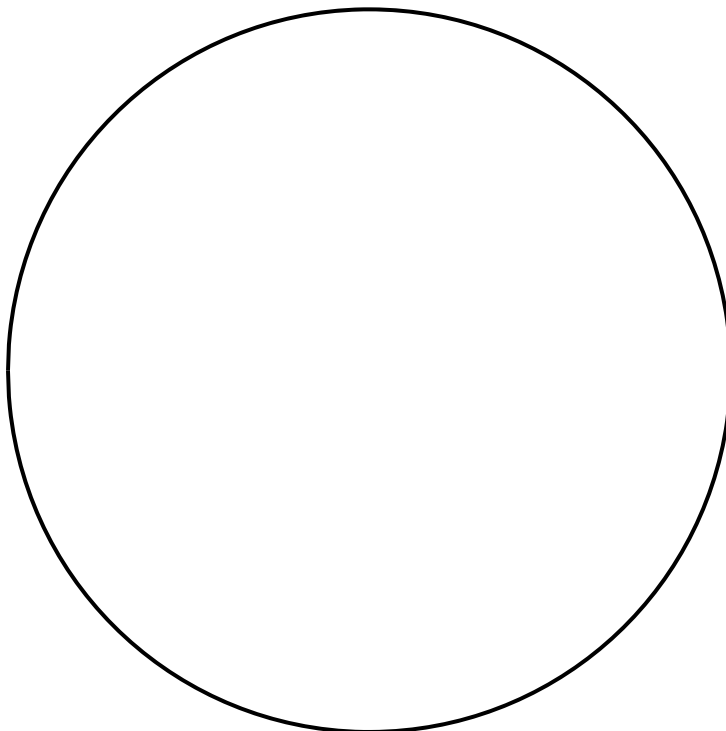
В этом примере из двух полуокружностей "склеена" окружность.

```
> plot({sqrt(1-x^2),-sqrt(1-x^2)},x=-1..1,y=-1..1,scaling=constrained);
```



Окрасив обе половинки в одинаковый цвет, можно избавиться от эффекта склейки

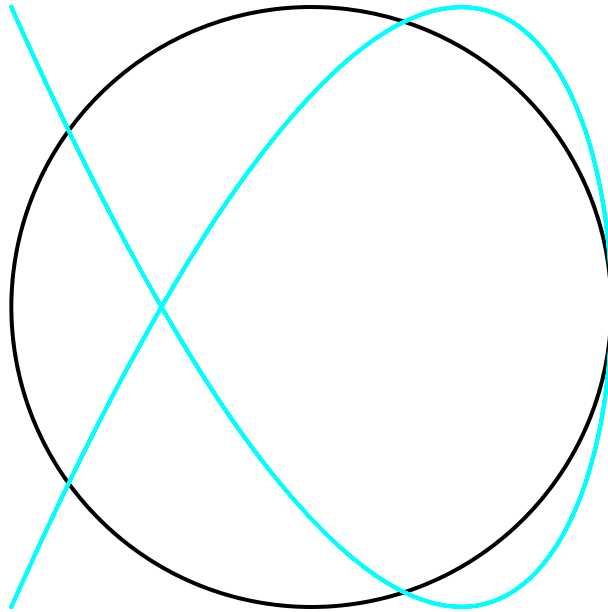
```
> plot({sqrt(1-x^2),-sqrt(1-x^2)},x=-1..1,y=-1..1,scaling=constrained,color=black,axes=none);
```



В документации к системе предупреждается, что если в опциях выполнена настройка параметров каждого отдельно взятого графика, то первый аргумент **plot** должен быть списком. Если же он будет множеством, то порядок присваивания значений опций не гарантируется.

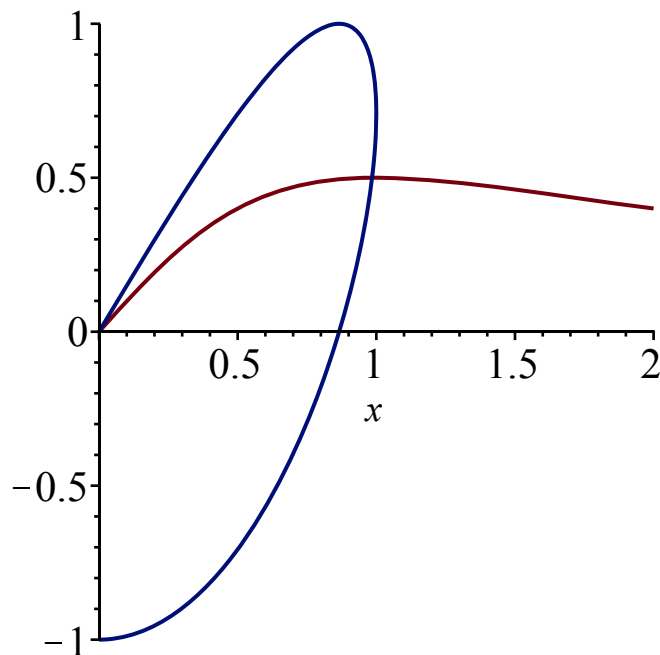
На одном рисунке можно отобразить несколько параметрически заданных кривых.

```
> plot([[sin(t),cos(t),t=0..2*Pi],[cos(2*t),sin(3*t),t=0..2*Pi]],  
       scaling=constrained,axes=none,color=[black,cyan]);
```



Список или же множество отображаемых зависимостей может содержать графики как явно, так и параметрически заданных функций.

```
> plot({x/(1+x^2),[sin(2*x),sin(3*x),x=0..Pi/2]},x=0..2);
```



На одном рисунке можно отобразить и несколько зависимостей, заданных в функциональной форме. В этом случае имя независимой переменной *не указывается!*

```
> F1:=x->abs(x+1)-abs(x-1);
```

$$F1 := x \mapsto |x + 1| - |x - 1|$$

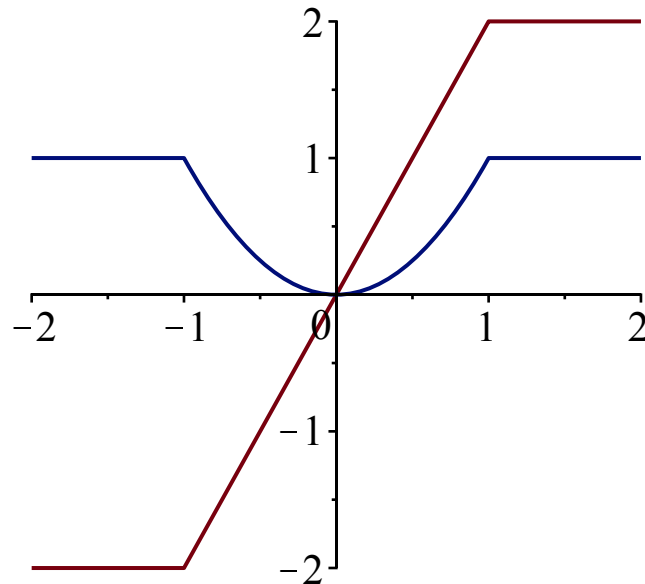
(1)

```
> F2:=x->`if`(abs(x)<1,x^2,1);
```

$$F2 := x \mapsto \text{if}(|x| < 1, x^2, 1)$$

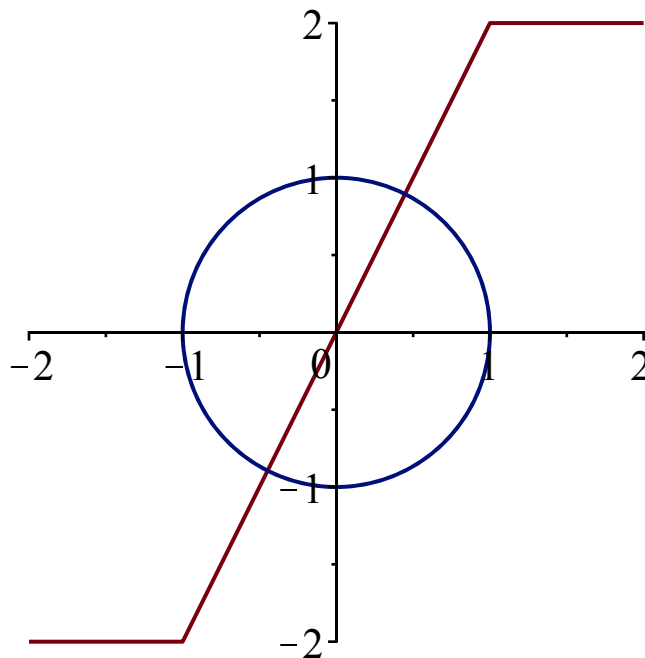
(2)

```
> plot([F1,F2],-2..2);
```



Синтаксис вызова команды **plot** допускает и такие, не совсем традиционные способы построения графиков.

```
> plot([F1,[sin(phi),cos(phi),phi=0..2*Pi]],-2..2);
```



Следует помнить, что некоторые опции отвечают за общий вид изображения, поэтому значением таких опций не могут быть списки.

В этом примере на одном рисунке показаны окружность и циклоида.  
Глобальные опции команды записаны во второй строке.

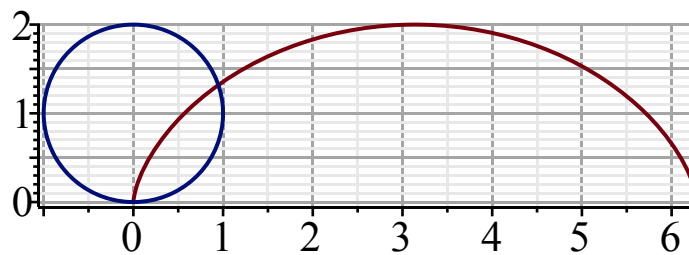
```
> x:=t->t-sin(t);  
y:=t->1-cos(t);
```

$$x := t \mapsto t - \sin(t)$$

$$y := t \mapsto 1 - \cos(t)$$

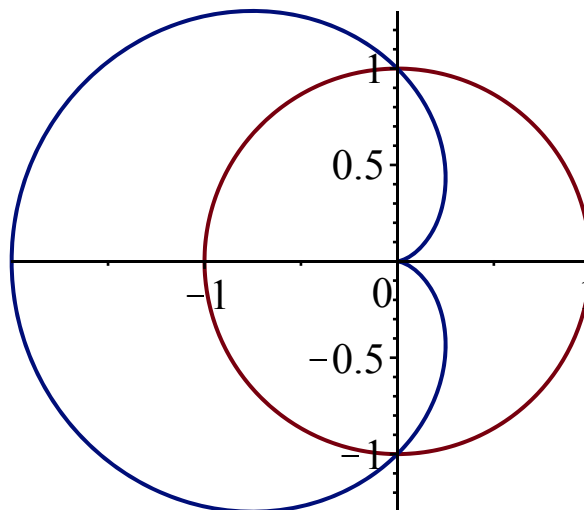
(3)

```
> plot([[x(t),y(t),t=0..2*Pi],[cos(t),1+sin(t),t=0..2*Pi]],  
scaling=constrained,axes=frame,gridlines=true);
```



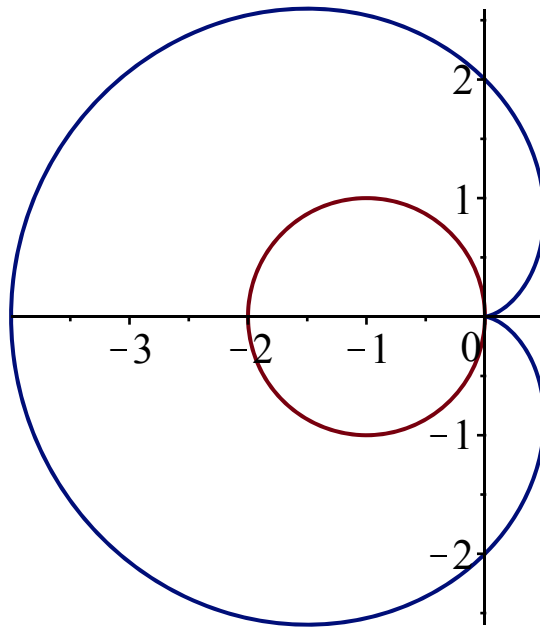
Имеется возможность построить несколько графиков функций,  
заданных в полярной системе координат.

```
> plot([1,1-cos(phi)],phi=0..2*Pi,coords=polar,scaling=constrained)  
;
```



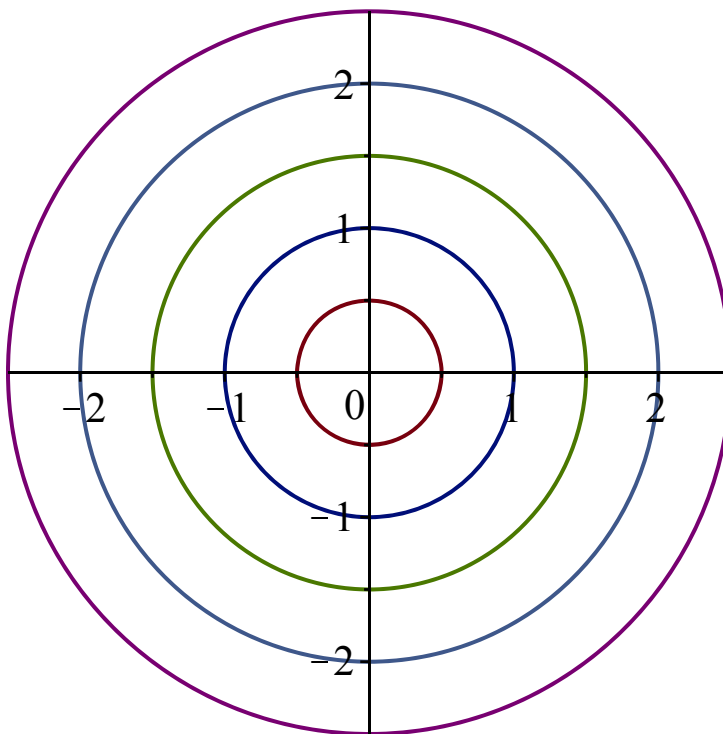
В полярной системе координат функции тоже могут задаваться параметрически.

```
> plot([-2*cos(phi),phi,phi=Pi/2..3/2*Pi],2*(1-cos(phi))],  
       phi=0..2*Pi,coords=polar,scaling=constrained);
```



В полярной системе координат линии с постоянным значением  $\rho$  представляют собой concentric окружности.

```
> plot([seq(j/2,j=1..5)],phi=0..2*Pi,coords=polar,scaling=  
       constrained);
```

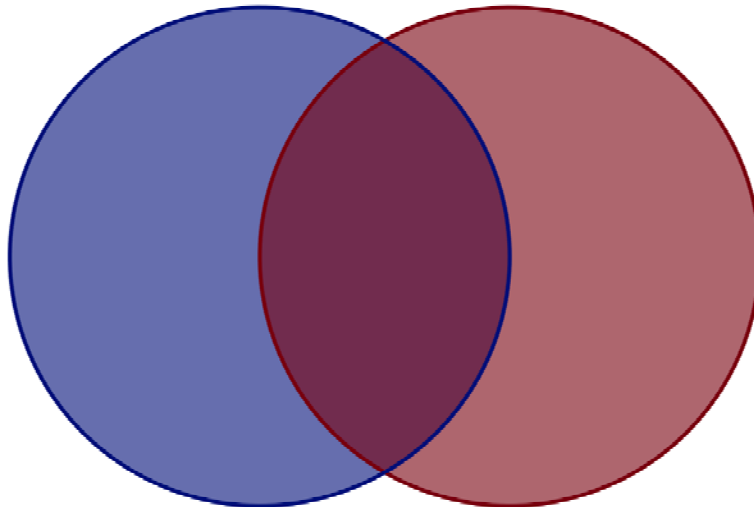


В декартовой системе координат реализован механизм закрашки области между кривой и осью абсцисс. Он работает и в том случае, когда кривых



несколько. Область пересечения закрашивается в промежуточный цвет.

```
> plot([ [1/2+cos(phi), sin(phi), phi=0..2*Pi],  
        [-1/2+cos(phi), sin(phi), phi=0..2*Pi]],  
       scaling=constrained, filled=true, axes=None);
```



На закрашенные области оказывает влияние значение опции *transparency*, которая принимает вещественные значения из диапазона от 0 до 1. Чем больше *transparency*, тем более прозрачными кажутся закрашенные области.

```
> plot([ [1/2+cos(phi), sin(phi), phi=0..2*Pi],  
        [-1/2+cos(phi), sin(phi), phi=0..2*Pi]],  
       scaling=constrained, filled=true, axes=None, transparency=0.7);
```

