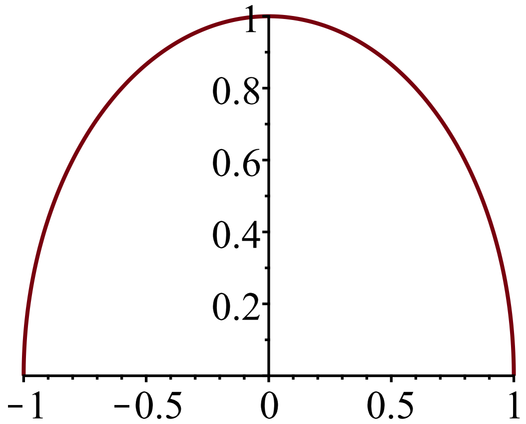
Графики функций в криволинейных системах координат

Декартова система координат является наиболее популярной, но не единственно возможной системой координат, в которой Maple умеет строить графики. В двумерном случае поддерживаются более 10 различных координатных систем.

Среди криволинейных координатных систем наиболее известна полярная система координат. В ранних версиях Maple для построения графиков функций в полярной системе координат использовались специальные команды из графических библиотек системы. В настоящее время рекомендуемый подход состоит в использовании команды **plot** с опцией *coords*. Никаких дополнительных библиотек подключать уже не надо.

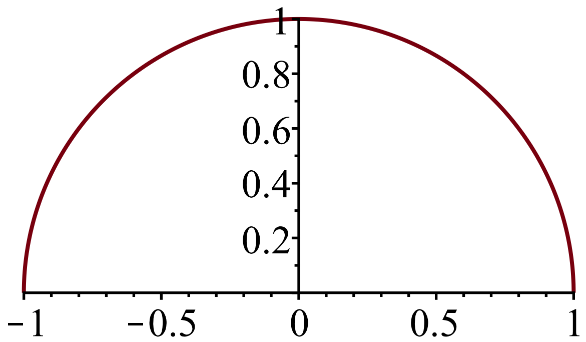
Например, если мы хотим построить полуокружность достаточно задать значение радиуса, указав, что будет использована полярная система координат.

```
> plot(1,x=0..Pi,coords=polar);
```



Все ранее изученные опции команды **plot** сохраняют работоспособность, поэтому для корректного отображения геометрического объекта всегда желательно использовать опцию *scaling*.

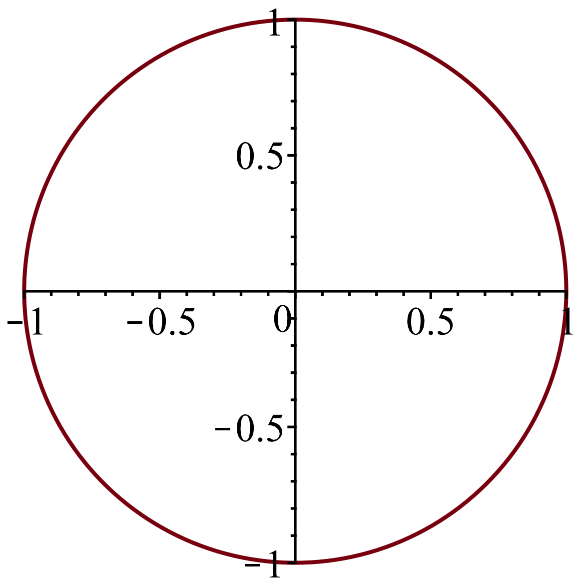
```
> plot(1,x=0..Pi,coords=polar,scaling=constrained);
```



В полярной системе координат явно заданные функции определяются зависимостью длины радиус-вектора ρ от полярного угла ϕ, то есть $\rho = \rho(\phi)$. Таким образом, аргумент *x* из предыдущих примеров следует понимать как полярный угол.

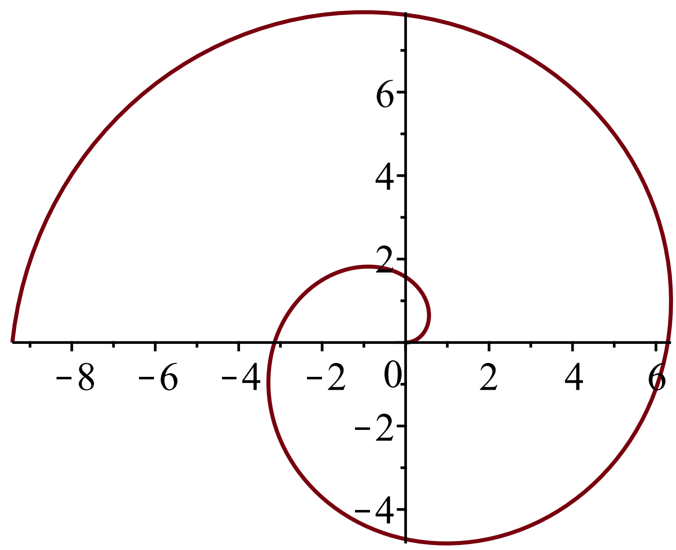
Для того, чтобы не возникало коллизий его лучше всего и обозначать соответствующей переменной. Тогда для рисования окружности можно воспользоваться командой

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



Знаменитая спираль Архимеда, которая определяется соотношением $\rho = \phi$, наглядно иллюстрирует роль аргумента команды **plot** при наличии опции *coords*.

```
> plot(phi,phi=0..3*Pi,coords=polar,scaling=constrained);
```



Поддерживается функциональная форма вызова команды **plot** при использовании опции *coords*.

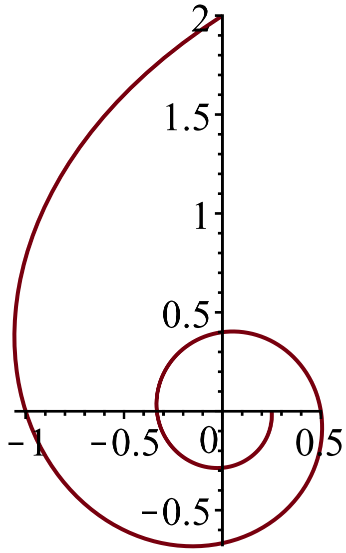
Предположим, что уравнение гиперболической спирали задано в виде функции

```
> Spiral:=phi->Pi/phi;
```

$$Spiral := \phi \to \frac{\pi}{\phi} \tag{1}$$

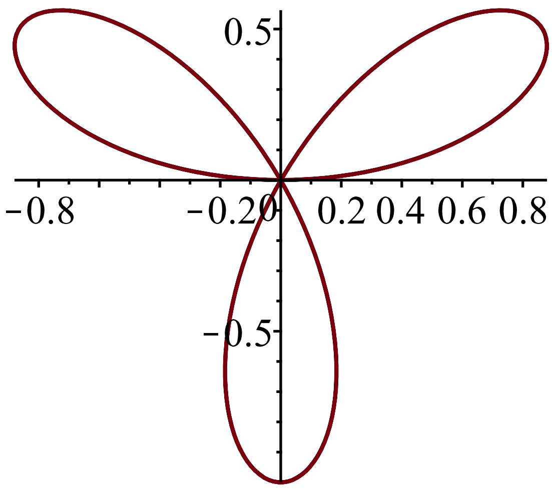
При вызове команды **plot** аргумент этой функции не указывается!

```
> plot(Spiral,Pi/2..4*Pi,coords=polar,scaling=constrained);
```



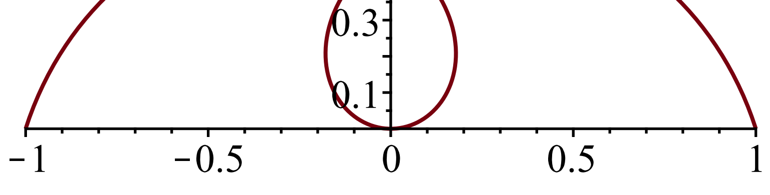
Использование полярной системы координат позволяет строить кривые весьма причудливой формы с помощью выражений совсем простого вида. Это пример, так называемой, трёхлепестковой розы.

```
> plot(sin(3*phi),phi=0..2*Pi,coords=polar,scaling=constrained);
```



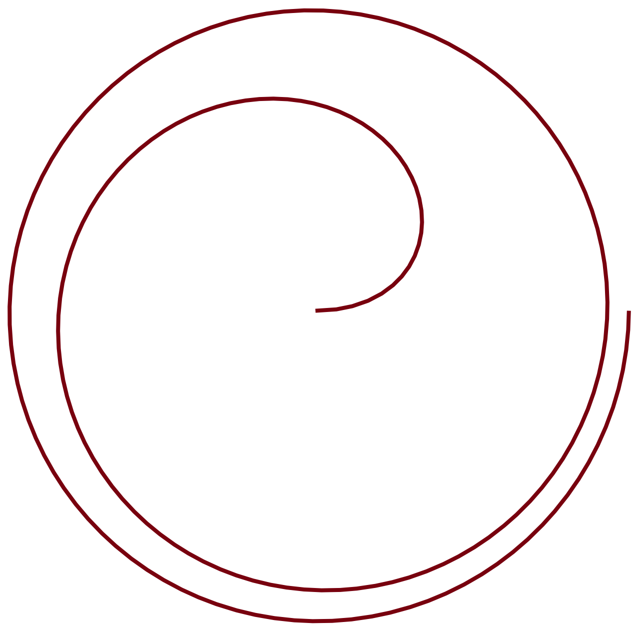
Реализованные в системе алгоритмы таковы, что допускают использование выражений с отрицательными значениями ρ. Если для некоторого ϕ величина $\rho(\phi)$ отрицательна, то проходящая через полярный центр линия постоянного азимута продолжается в противоположном направлении и на ней отмечается точка на расстоянии $|\rho(\phi)|$ от полярного центра. Эта особенность поясняет симметричный характер приводимой ниже кривой.

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



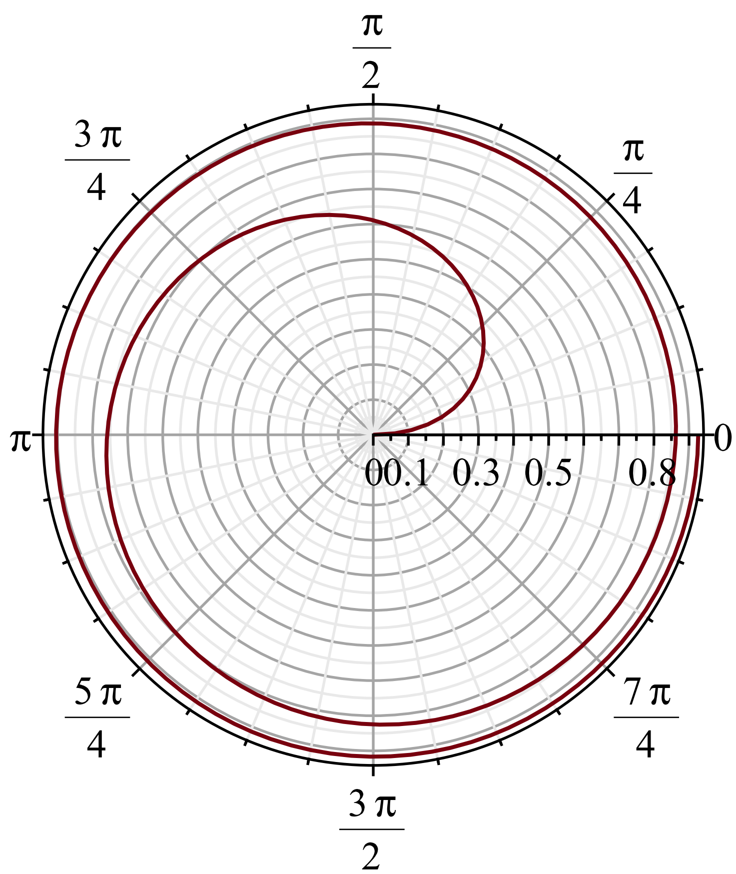
По умолчанию отображаемая система координат остаётся декартовой, но от неё можно избавиться. Для этого следует использовать уже известную опцию *axes*.

```
> plot(phi/(1+phi),phi=0..4*Pi,coords=polar,scaling=constrained,
      axes=none);
```



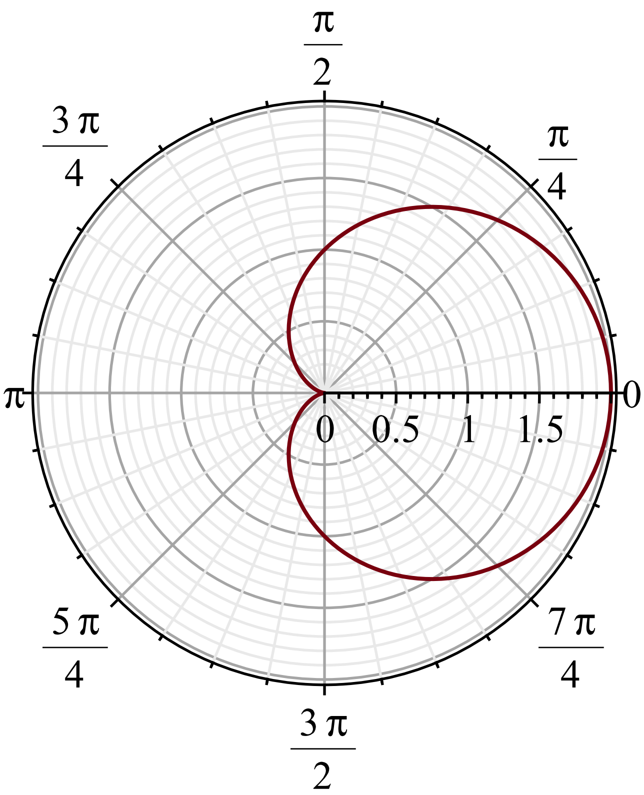
При решении некоторых задач координатную сетку всё же лучше сохранять. Для полярной системы координат Maple предлагает удобное решение в виде опции *axiscoordinates.* Указав в качестве значения этой опции название координатной системы, получаем изображение пригодное для детального анализа кривой.

```
> plot(phi/(1+phi),phi=0..4*Pi,coords=polar,scaling=constrained,
      axiscoordinates=polar);
```



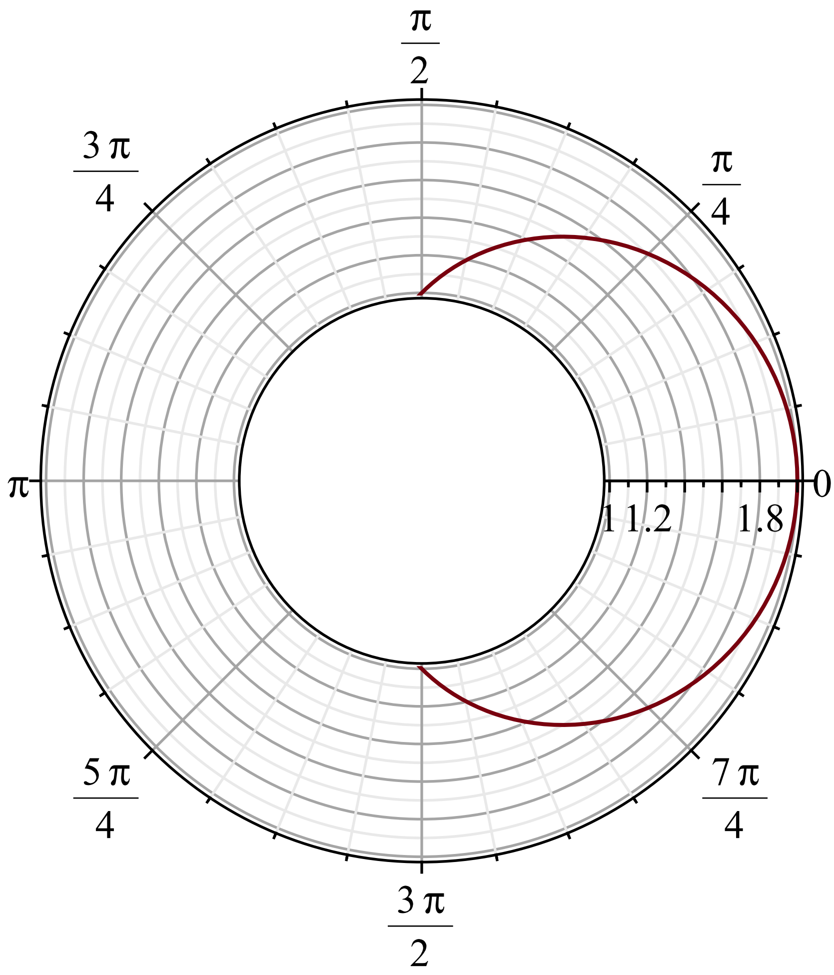
Так будет выглядеть кардиоида при наложении на неё координатной сетки.

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



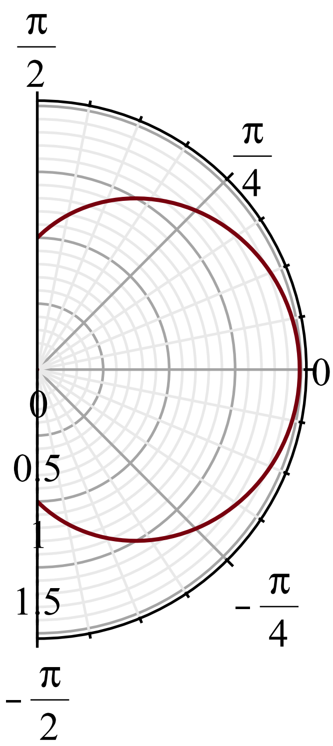
Координатная сетка может быть нарисована только в некоторой ограниченной области, параметры которой задаются с помощью опции *coordinateview*. Значением этой опции является список из двух диапазонов. Первый из этих диапазонов определяет границы параметра ρ, в которых отображается как координатная сетка, так и сама кривая.

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



Второй диапазон задаёт границы изменения азимута. Стоит отметить, что он не обязан находиться внутри диапазона, который использовался для рисования графика.

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

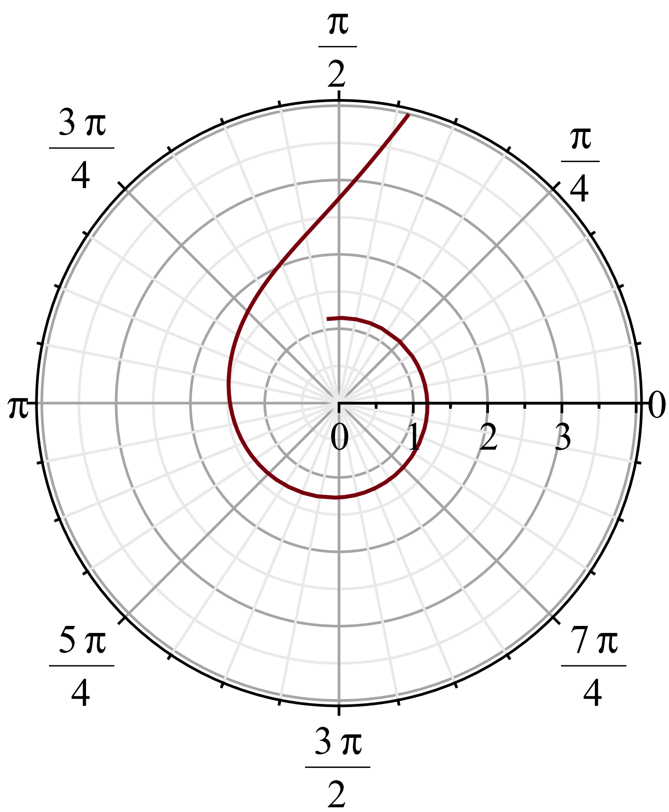


В полярных координатах можно строить графики не только явно заданных функций. Параметрически заданные кривые рисуются также, как и в декартовых координатах.

В качестве примера построим график кривой вида $\phi = \frac{\rho}{\rho - 1}$ при $1 < \rho$.

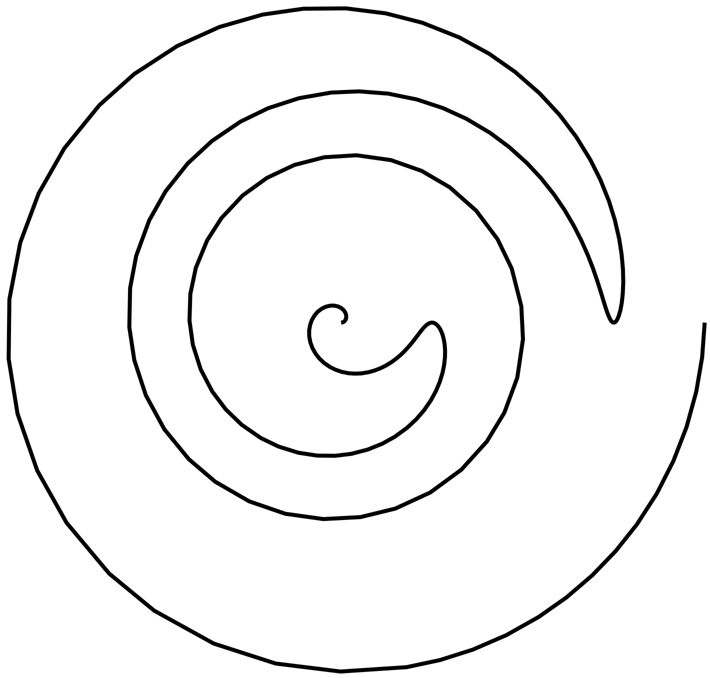
В первом элементе списка содержится зависимость от параметра величины ρ, во втором $-\phi$.

```
> plot([rho,rho/(rho-1),rho=8/7..4],coords=polar,scaling=constrained,
      axiscoordinates=polar);
```



Ещё один пример параметрически заданной кривой: зависимость вида $\phi = 2\,\pi \sin(\rho)$.

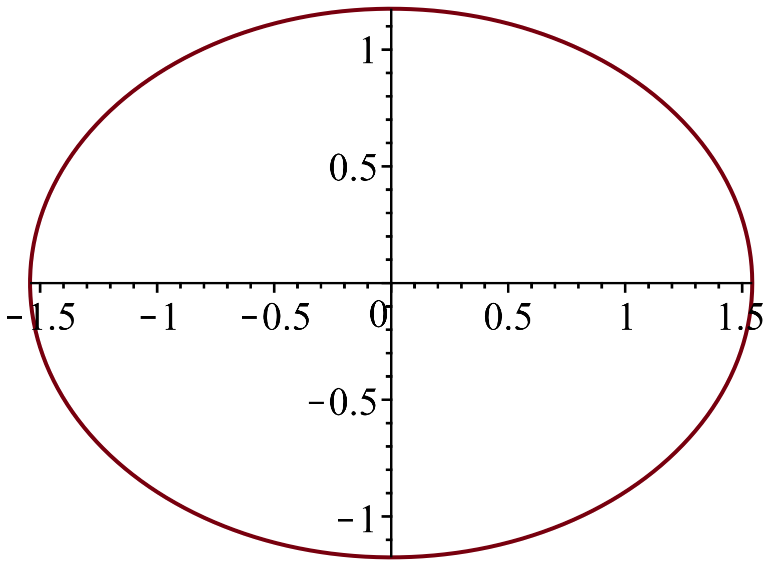
```
> plot([rho,2*Pi*sin(rho),rho=0..2*Pi],coords=polar,
      scaling=constrained,axes=none,color=black);
```



Другие криволинейные координатные системы используются крайне редко.

Аналогом единичной окружности в эллиптической системе координат является эллипс с полуосями равными $\cosh(1)$ и $\sinh(1)$.

```
> plot(1,x=0..2*Pi,coords=elliptic,scaling=constrained);
```



В этом нетрудно убедиться непосредственно из рисунка, если заметить, что

```
> evalf(cosh(1));evalf(sinh(1));
```

$$1.543080635$$
$$1.175201194 \quad \textbf{(2)}$$