

Initiate useful plotting commands:

```
> with(plots);
```

```
[animate, animate3d, animatecurve, arrow, changecoords, complexplot, complexplot3d,  
conformal, conformal3d, contourplot, contourplot3d, coordplot, coordplot3d, densityplot,  
display, dualaxisplot, fieldplot, fieldplot3d, gradplot, gradplot3d, implicitplot, implicitplot3d,  
inequal, interactive, interactiveparams, intersectplot, listcontplot, listcontplot3d,  
listdensityplot, listplot, listplot3d, loglogplot, logplot, matrixplot, multiple, odeplot, pareto,  
plotcompare, pointplot, pointplot3d, polarplot, polygonplot, polygonplot3d,  
polyhedra_supported, polyhedraplot, rootlocus, semilogplot, setcolors, setoptions,  
setoptions3d, spacecurve, sparsematrixplot, surfdata, textplot, textplot3d, tubeplot]
```

(1)

Define cardioid as a function depending on the angle:

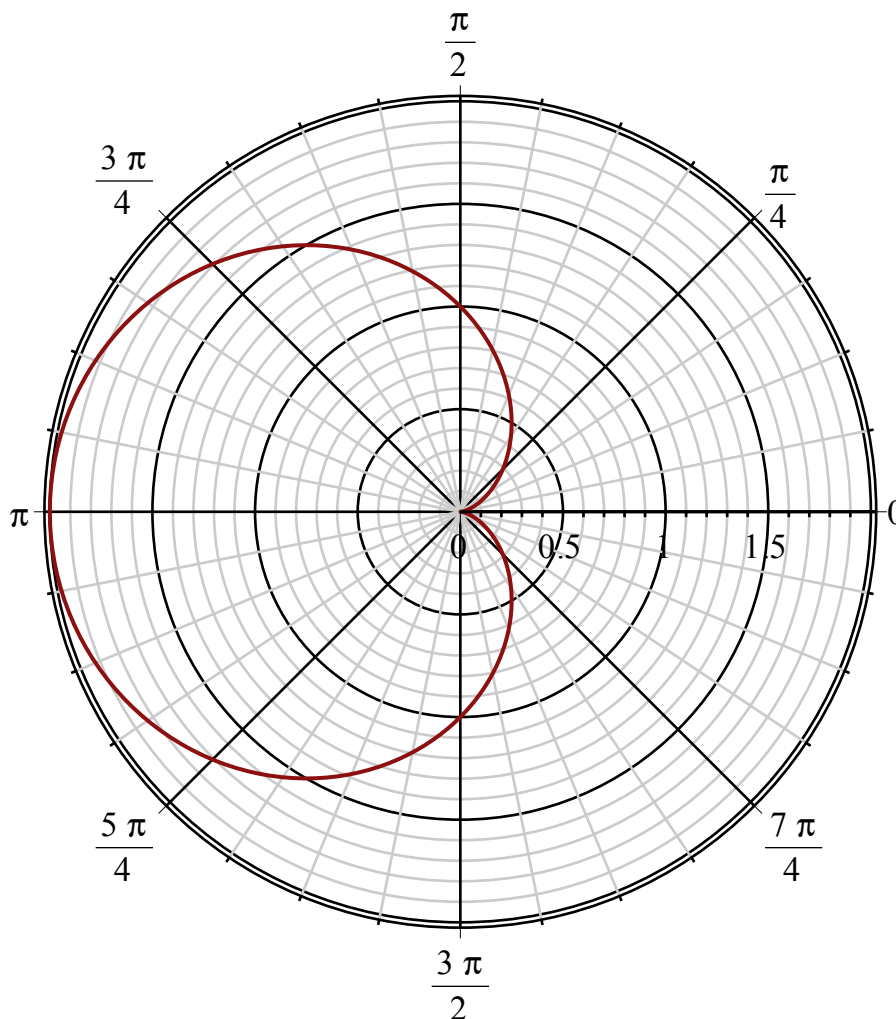
```
> r := 1 - cos(theta);
```

$$r := 1 - \cos(\theta)$$

(2)

Plot cardioid in polar coordinates with polarplot:

```
> polarplot([r(theta)], theta = -Pi..Pi);
```



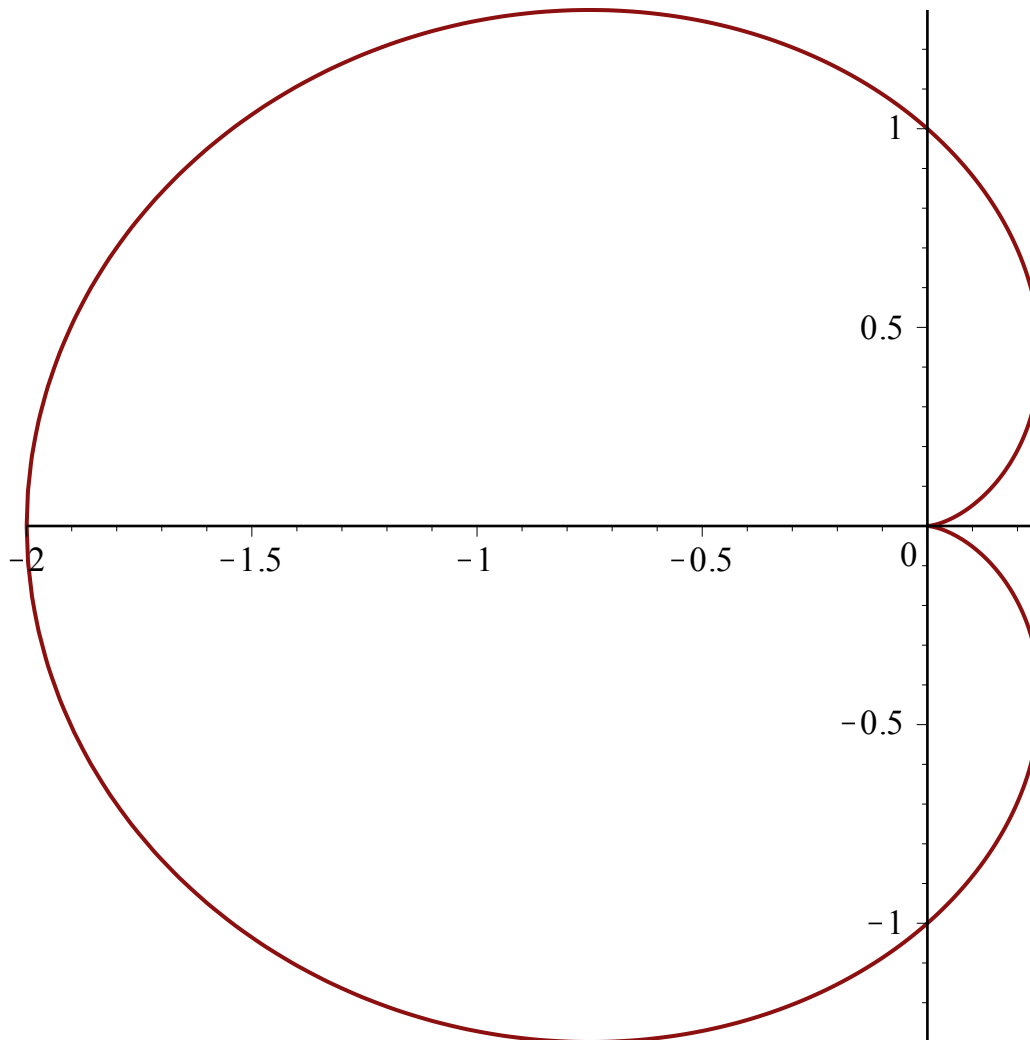
Define cardioid implicitly using a function depending on both variables:

```
> f(r, theta) := r + cos(theta) - 1;  
f := (r, theta) -> r - 1 + cos(theta)
```

(3)

Plot cardioid with implicitplot in polar coordinates:

```
> implicitplot(f(r, theta) = 0, r = 0 .. 2, theta = -Pi .. Pi, numpoints = 20000,  
coords = polar);
```



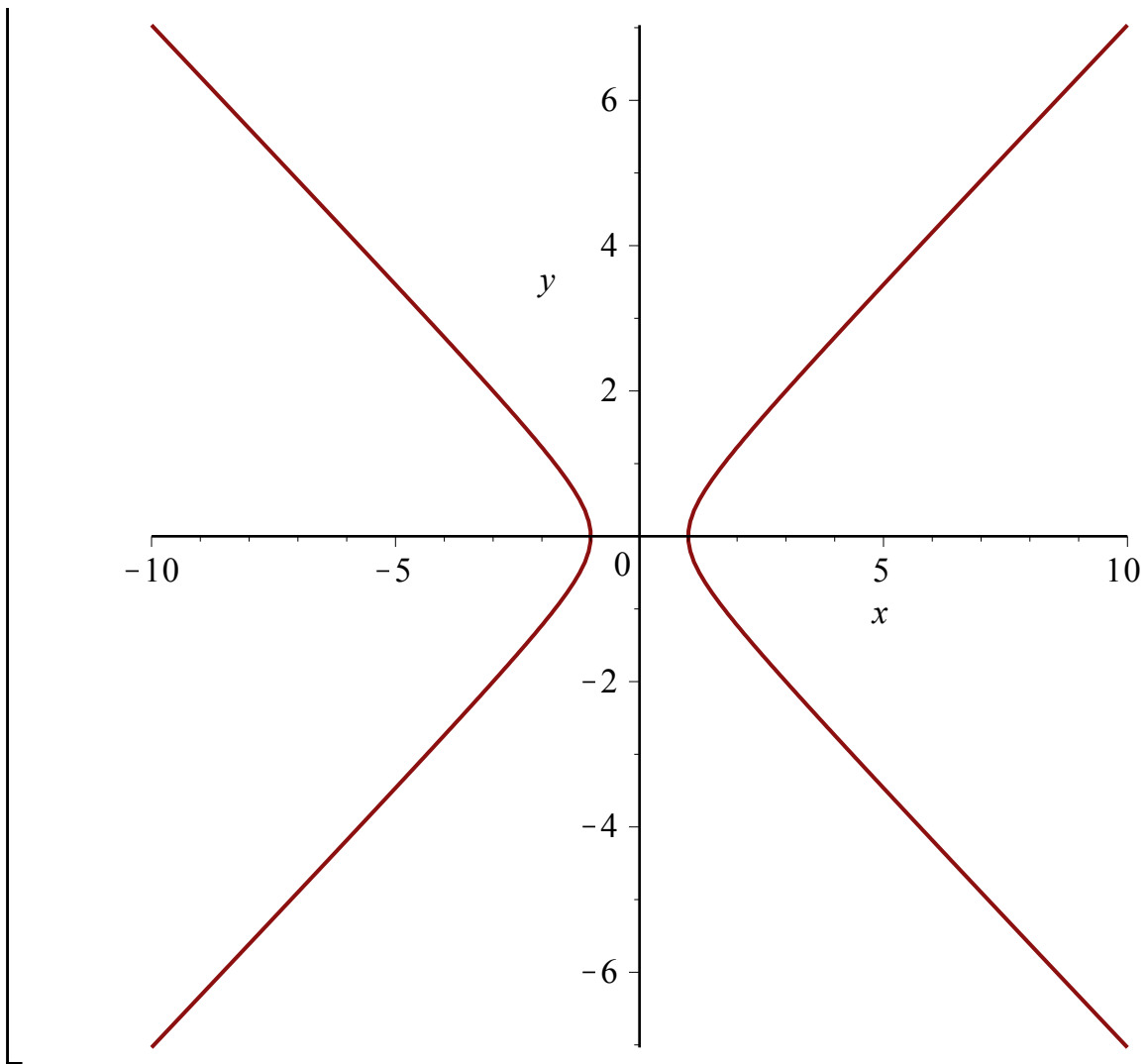
Variant: define hyperbola implicitly:

```
> g(x, y) := x^2 - 2 * y^2 - 1;  
g := (x, y) -> x^2 - 2 * y^2 - 1
```

(4)

Plot hyperbola with implicitplot (in cartesian coordinates):

```
> implicitplot(g(x, y) = 0, x = -10 .. 10, y = -10 .. 10, numpoints = 20000);
```



[ Good command to know for resetting all parameters and loaded packages:  
[> `restart;`