

9/3/2019 --  
HH16, HH15, HH20 ... bad license file. HH02, HH09, HH10, HH03, HH19, HH31, old licen  
HH18 seems good. No one **else** is awake.

>  
>  
>  
>

stuff here is some polynomial  $x^3 + y$  and thats that. or maple `plot(x^2, x=1..2)`

>

types of things... **x** is a name. so is `jimmyJoeBob`

> `jimmyJoeBob := 4;`

`jimmyJoeBob := 4` (1)

strings are a bunch of charcters

> `something := "something in the way she moves"`

`something := "something in the way she moves"` (2)

> `something[5]`

"t" (3)

>

numbers come in many forms

> `27, 8.0123, sqrt(5), 1 + I*sqrt(2), 0, 0, 5`

`27, 8.0123, sqrt(5), 1 + I*sqrt(2), 0, 0, 5` (4)

> `somelist := %`

`somelist := 27, 8.0123, sqrt(5), 1 + I*sqrt(2), 0, 0, 5` (5)

I can refer to the 2nd element. The thing above is actually a (finite) sequence, but whatever.

> `somelist[2]`

8.0123 (6)

can box stuff up into a set or a list... which is a kind of array.

> `{somelist}`

`{0, 5, 27, 8.0123, sqrt(5), 1 + I*sqrt(2)}` (7)

> `mylist := [somelist]`

`mylist := [27, 8.0123, sqrt(5), 1 + I*sqrt(2), 0, 0, 5]` (8)

sets have no fixed order, no repeated elements, lists fixed order, repeats are ok.

> `mylist[4]`

$1 + I\sqrt{2}$  (9)

> `mylist[10]`

Error, invalid subscript selector

Theres lots more types.

To make a sequence by computing...

> `seq(i^2, i=1..12)`

`1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144` (10)

let's write a function that given n and m, gives the squares between n and m inclusive.

This is not it.

```
> f := x → x2
                                     f := x ↦ x2
(11)
```

```
> f(2);
                                     4
(12)
```

```
> g := (x, y) → x2 + sin(y)
                                     g := (x, y) ↦ x2 + sin(y)
(13)
```

```
> g(3,  $\frac{\text{Pi}}{4}$ )
                                     9 +  $\frac{\sqrt{2}}{2}$ 
(14)
```

```
> seq(T2, T=1..12)
                                     1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144
(15)
```

```
> h := (m, n) → seq(i2, i=m..n)
Warning, `i` is implicitly declared local to procedure `h`
                                     h := (m, n) ↦ seq(i2, i=m..n)
(16)
```

```
> h(5, 8)
                                     25, 36, 49, 64
(17)
```

```
> h(8, 5)
```

```
>
```

```
>
```

```
> h := proc(m, n)
    local i;
    return(seq(i2, i=m..n));
end
                                     h := proc(m, n) local i; return seq(i2, i=m..n) end proc
(18)
```

semicolon ends a statement. colon (:) ends and doesn't write answer.

```
> myseq := h(1, 100) :
> myseq[23]
                                     529
(19)
```

```
> h := proc(m, n)
    local i;
    if (m ≥ n) then
        print("are you sure? m is bigger than n", m, n);
    fi;
    return(seq(i2, i=m..n));
end:
```

```
h := proc(m, n)
    local i;
    if n ≤ m then print("are you sure? m is bigger than n", m, n) end if;
    return seq(i2, i=m..n)
end proc
(20)
```

```
> h(3, 2)
                                     "are you sure? m is bigger than n", 3, 2
(21)
```

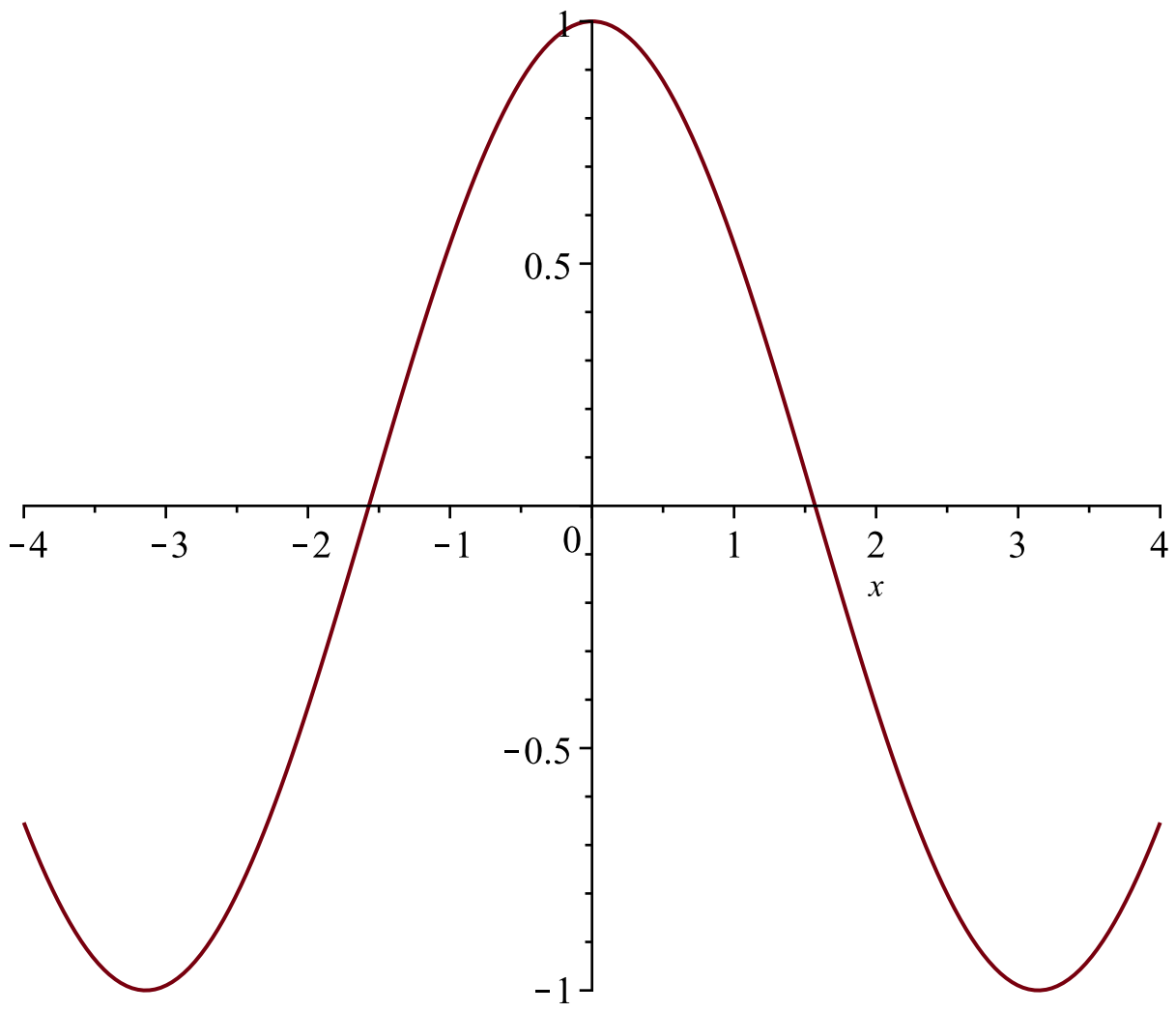
```
> h(2, 5)
```

4, 9, 16, 25

```
>  
>  
>  
>
```

```
some words on plotting
```

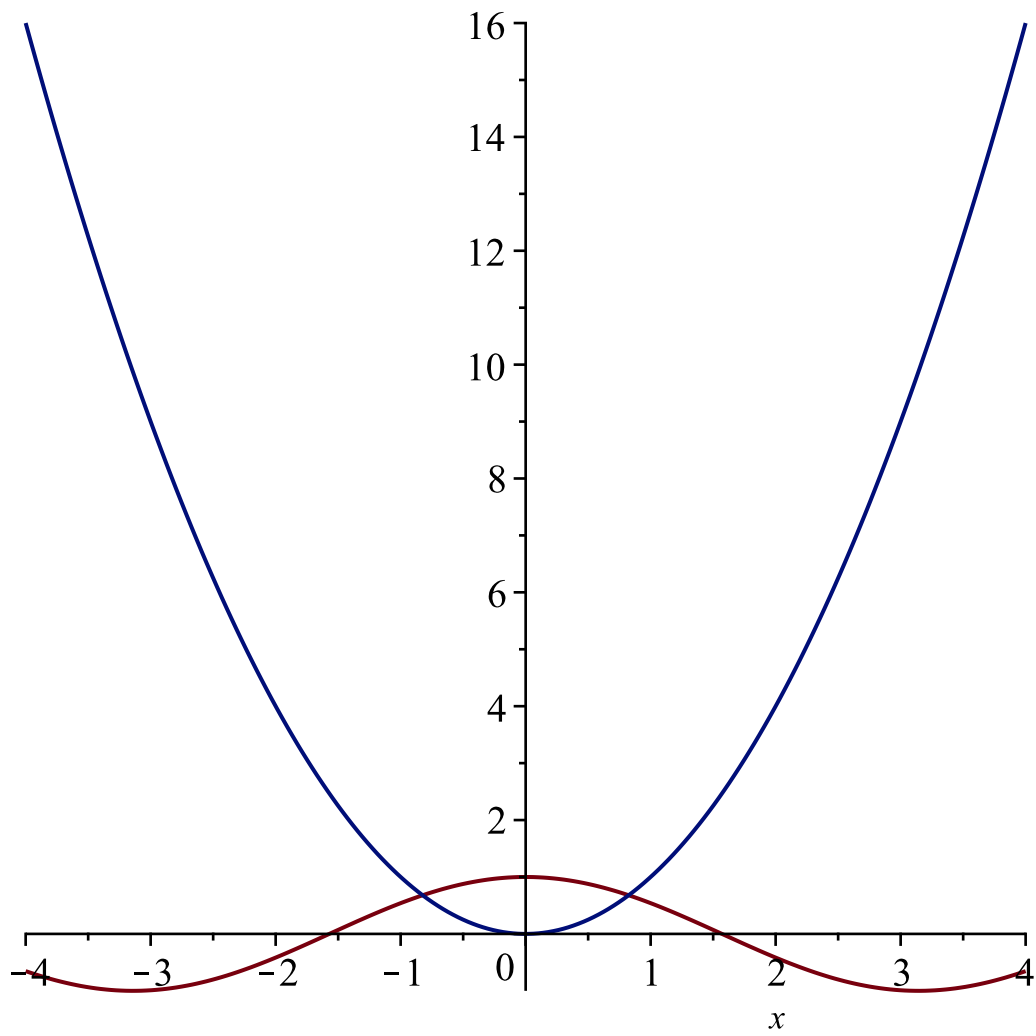
```
> plot(cos(x), x=-4..4)
```



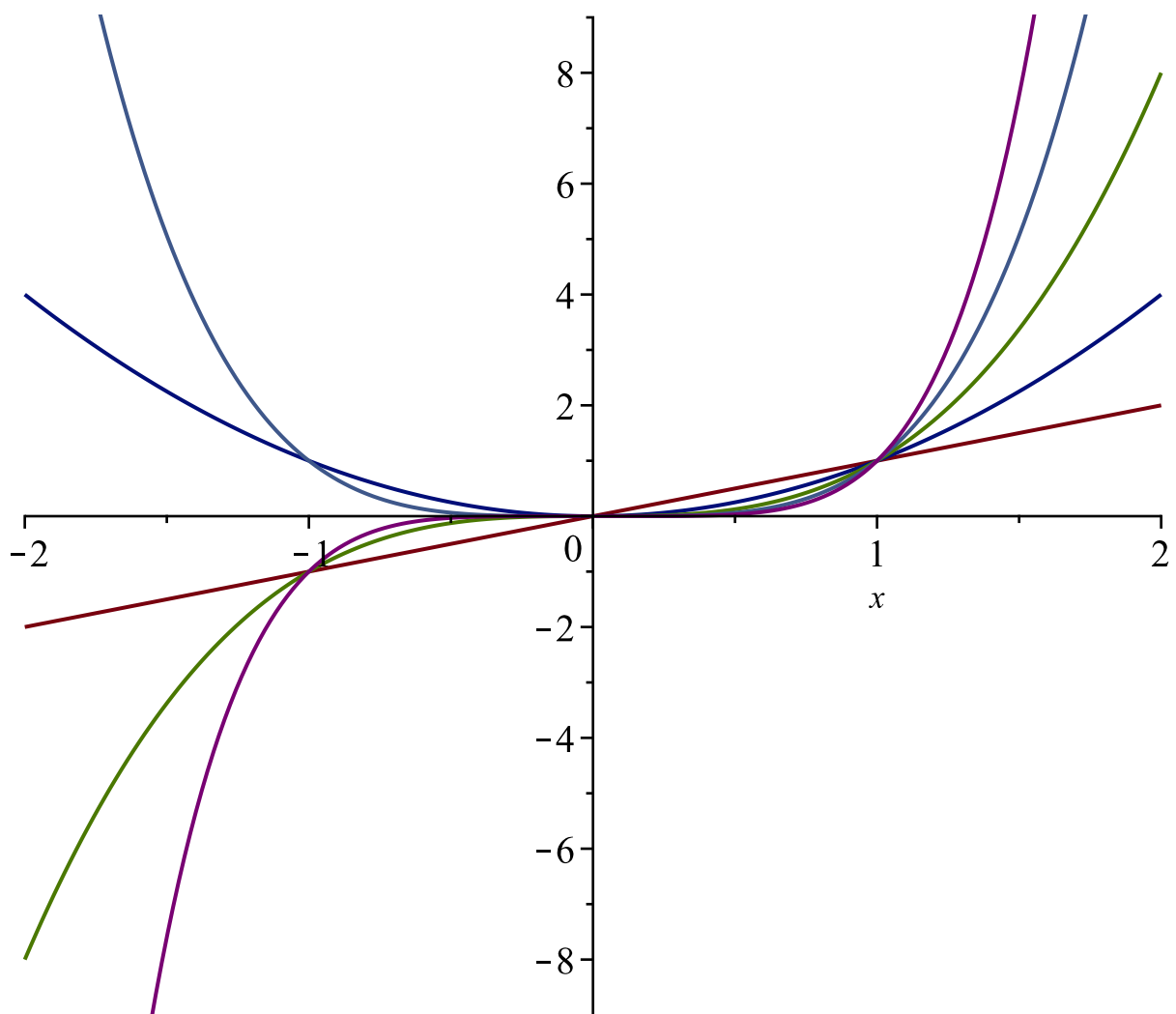
```
> plot(cos(x), x^2, x=-4..4)
```

Error, (in plot) unexpected options: [x^2, x = -4 .. 4]

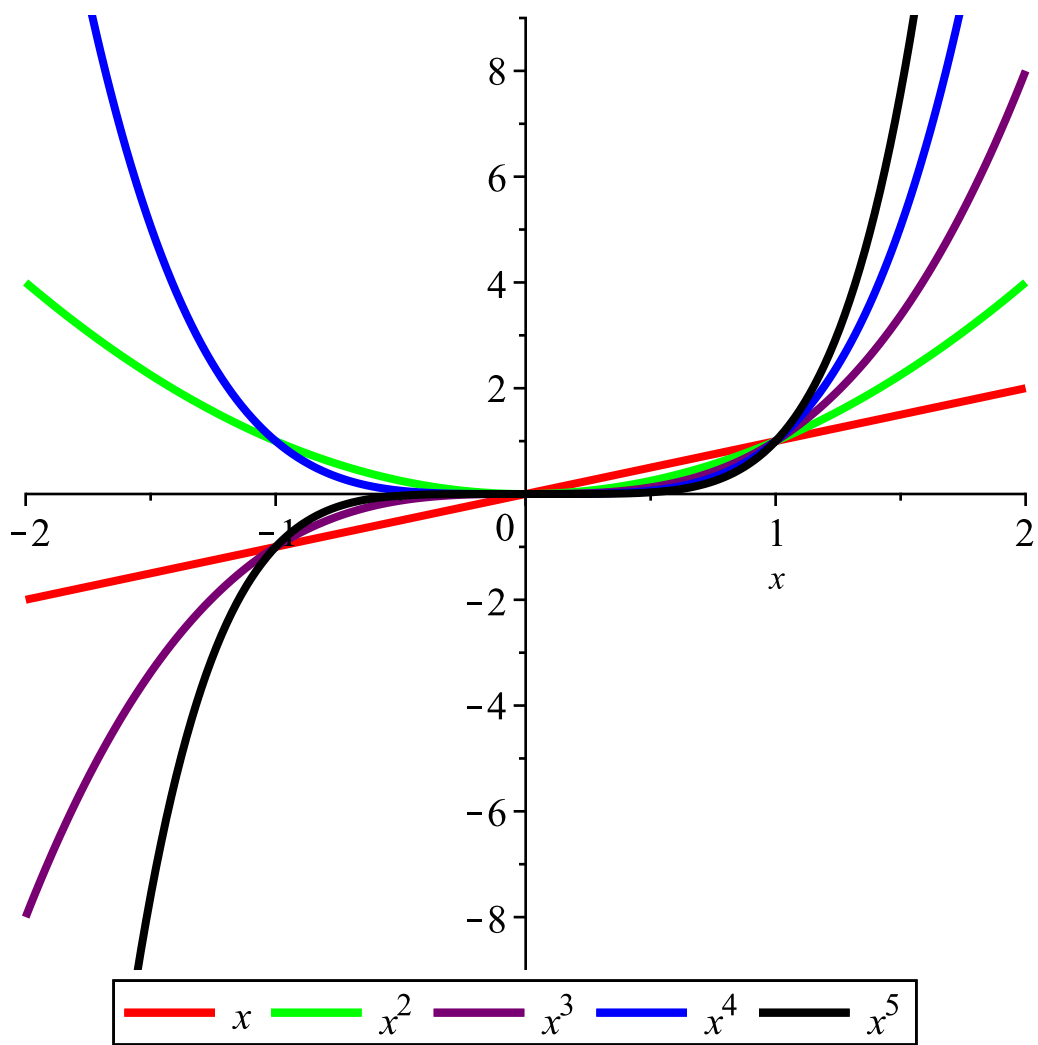
```
> plot([cos(x), x^2], x=-4..4)
```



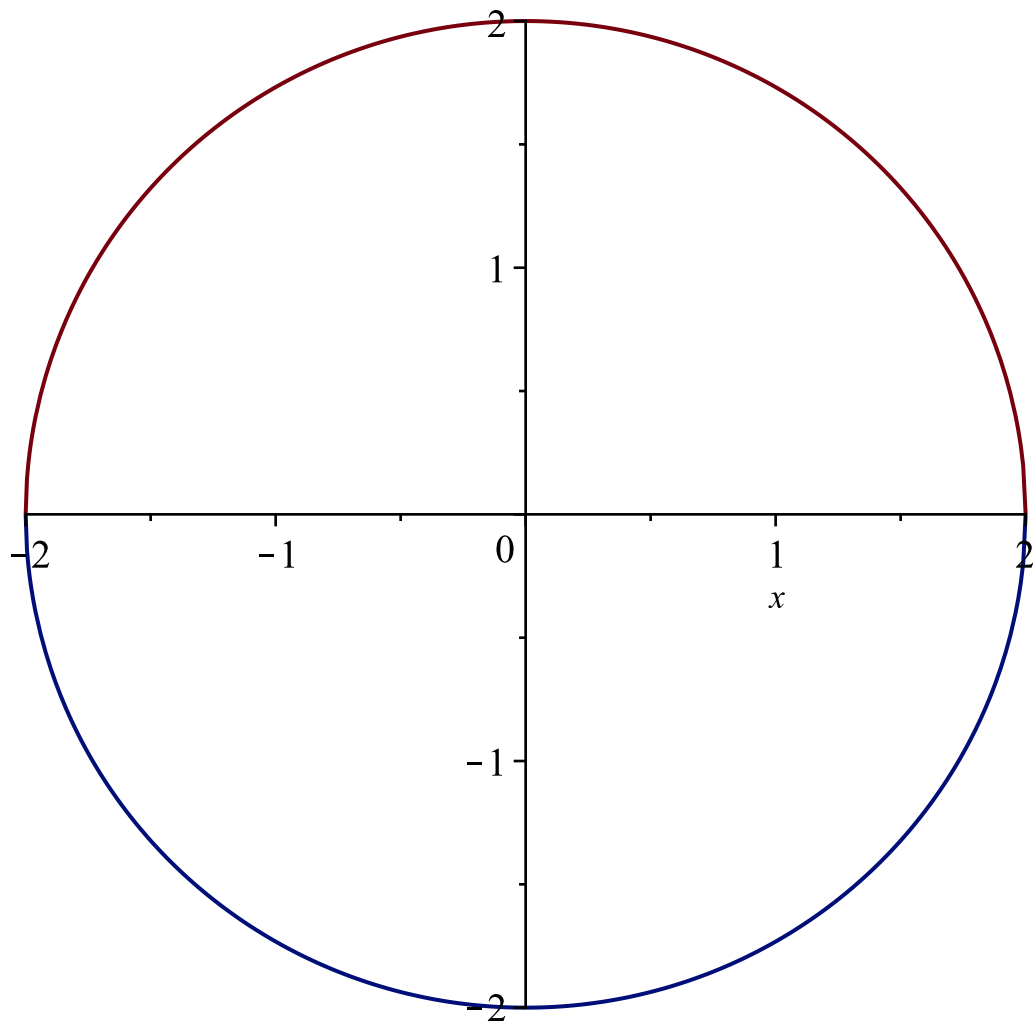
```
> plot([seq(x^n, n = 1 .. 5)], x = -2 .. 2);
```



```
> plot([seq(x^n, n = 1 ..5)], x = -2 ..2, thickness = 3, color = [red, green, purple, blue, black], legend  
= [seq(x^n, n = 1 ..5)]);
```

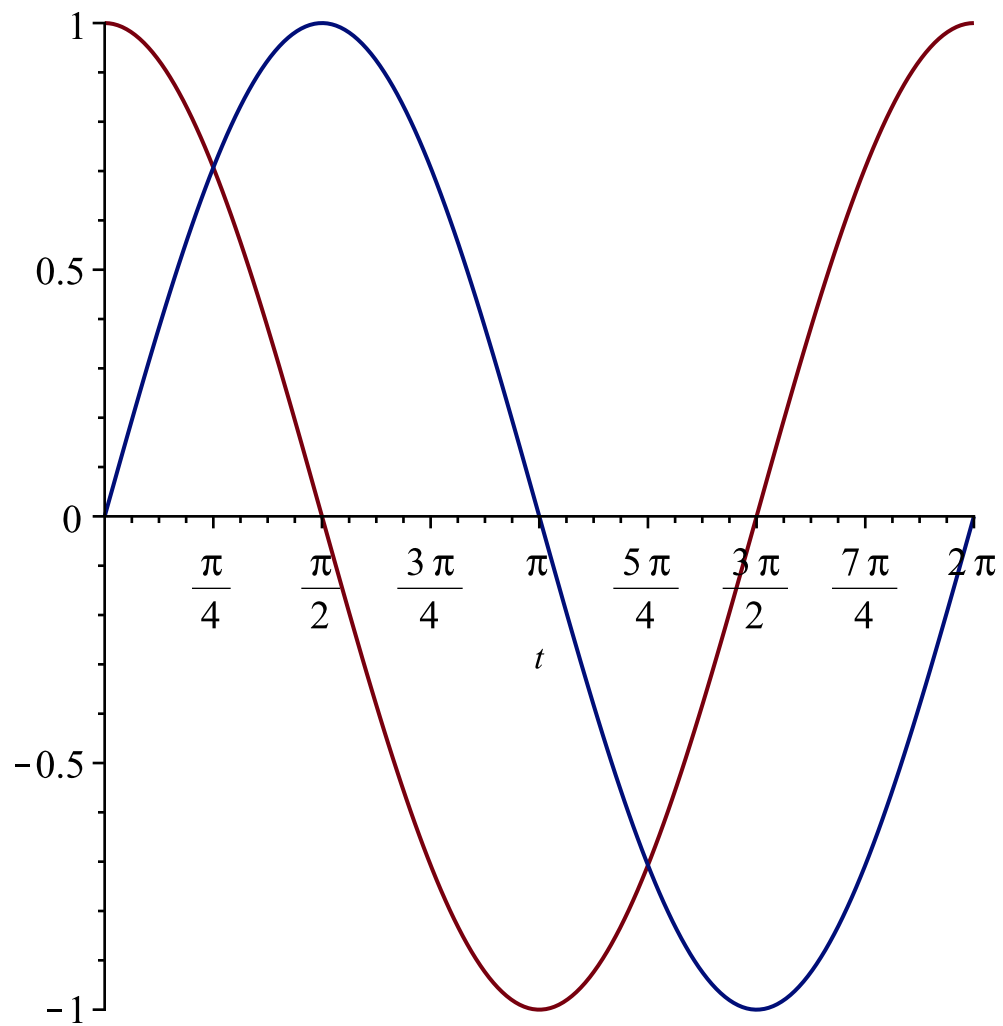


```
> plot([sqrt(4 - x^2), -sqrt(4 - x^2)], x=-2..2)
```



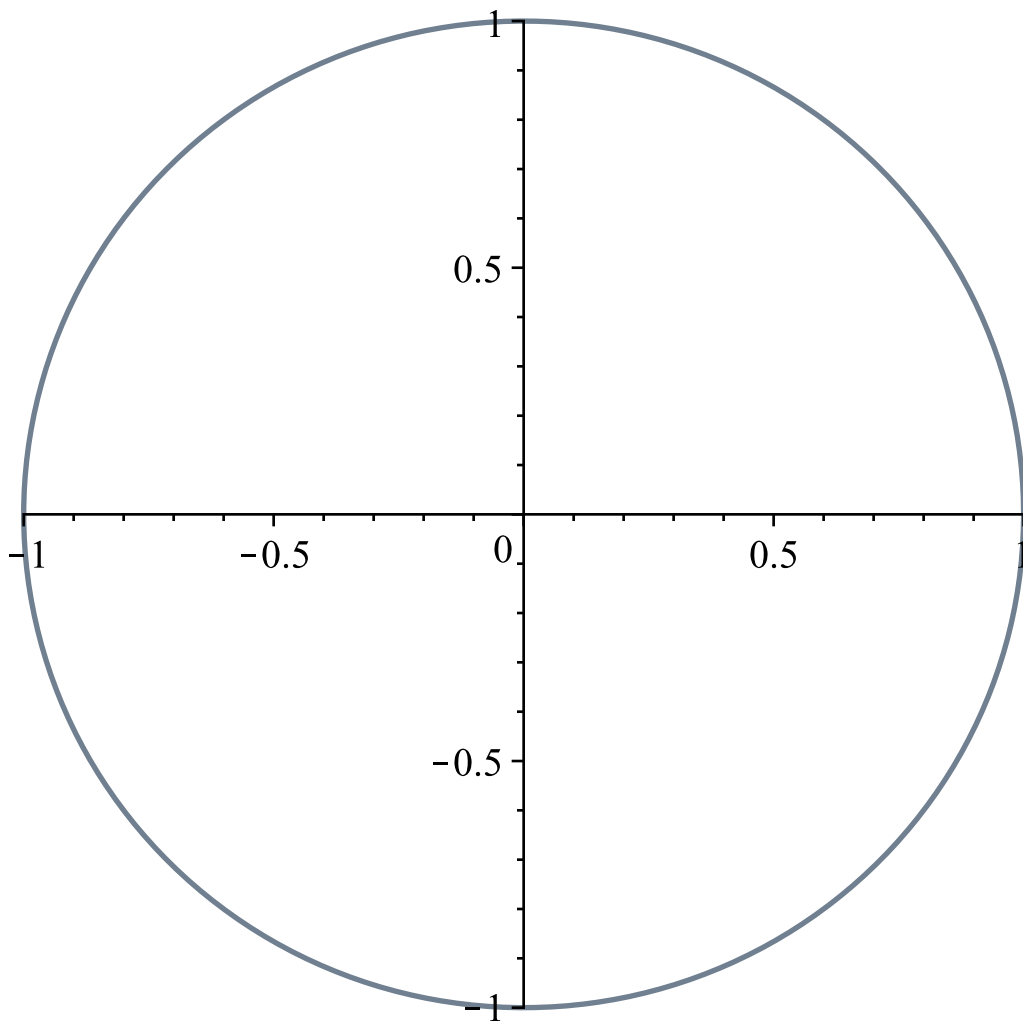
not what I want, but close

> `plot( [cos(t), sin(t)], t=0 ..2·Pi)`

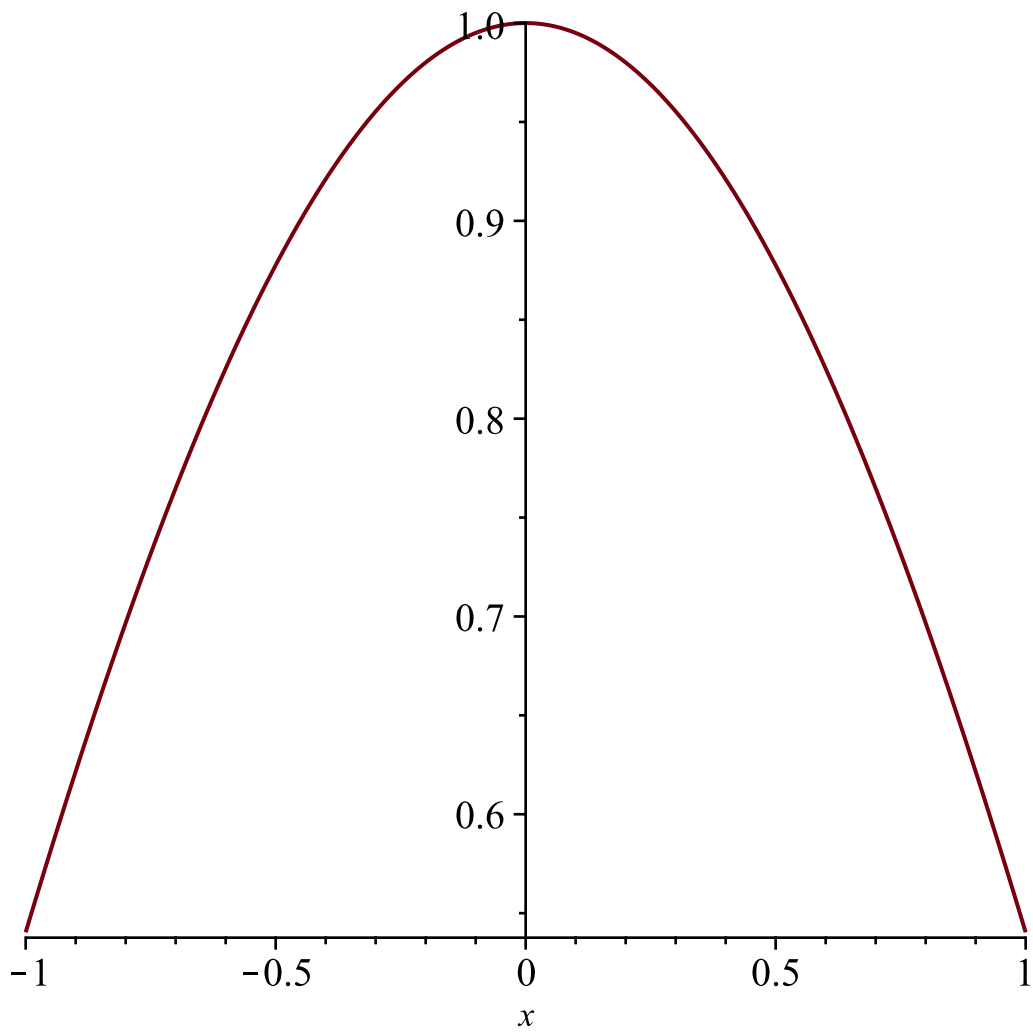


```
> cpic := plot( [cos(t), sin(t), t=0..2·Pi], thickness=2, color="SlateGrey")
```



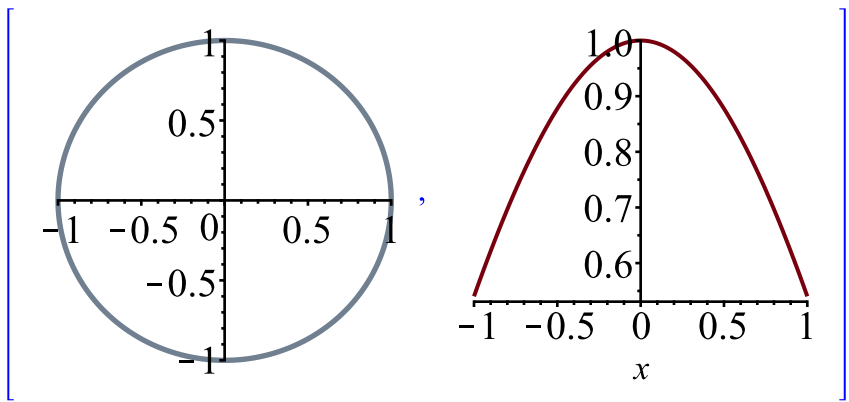


```
> rpic := plot(cos(x), x=-1..1);
```

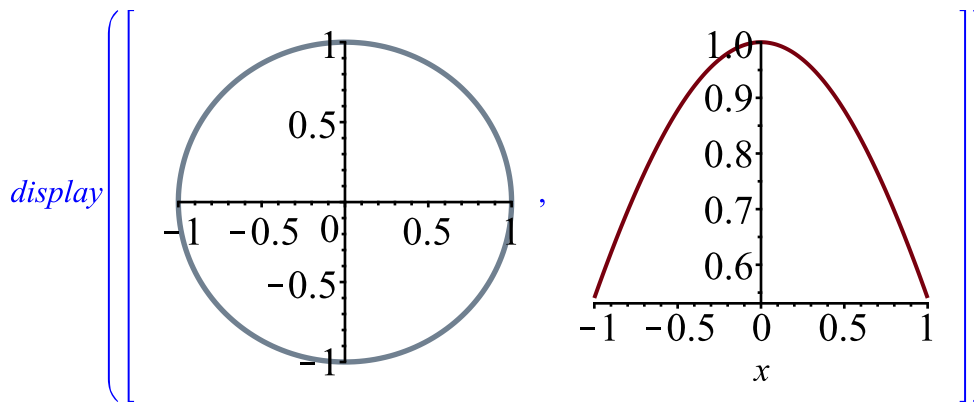


to show together, use display

> [cpic, rpic]



> display([cpic, rpic])

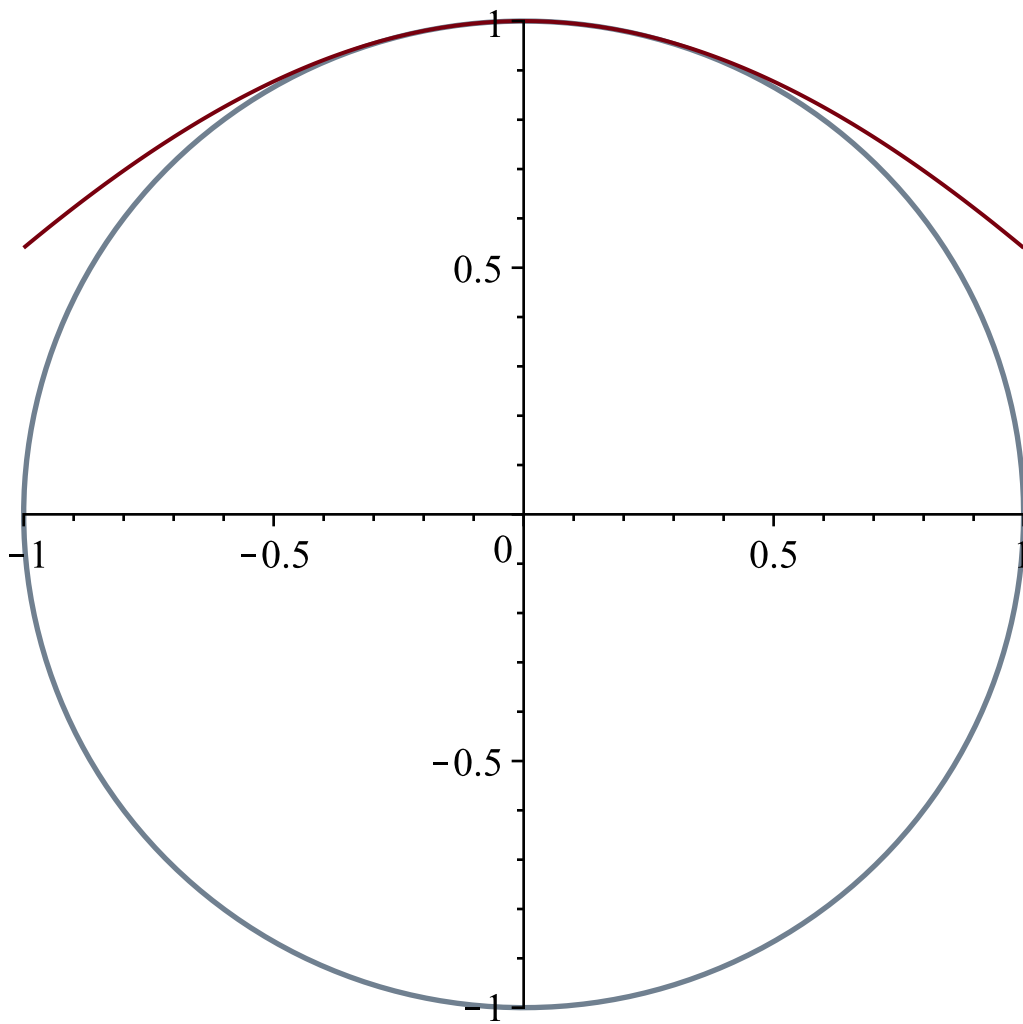


> *with(plots)*

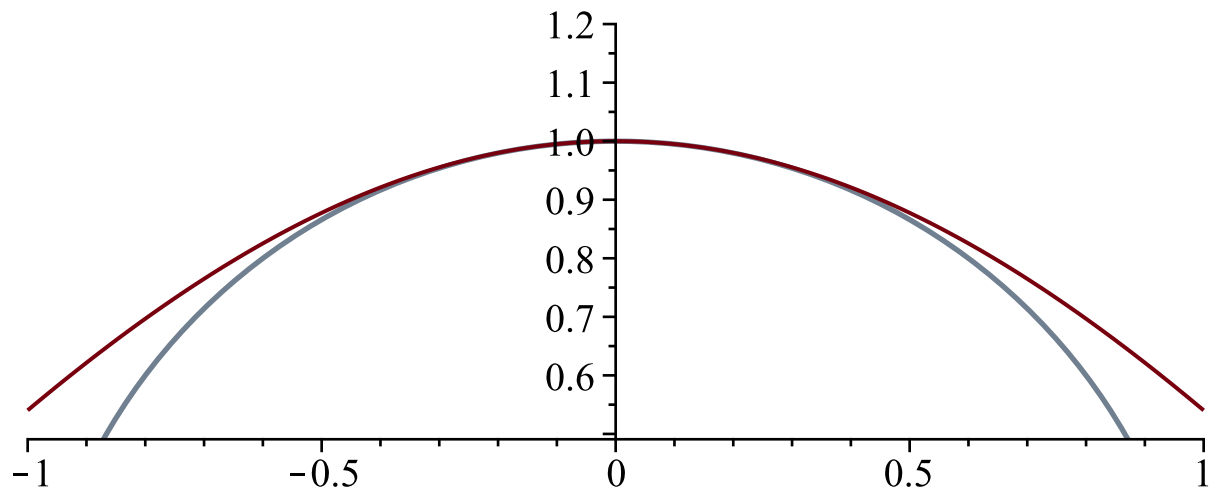
[*animate, animate3d, animatecurve, arrow, changecoords, complexplot, complexplot3d, conformal, conformal3d, contourplot, contourplot3d, coordplot, coordplot3d, densityplot, display, dualaxisplot, fieldplot, fieldplot3d, gradplot, gradplot3d, implicitplot, implicitplot3d, inequal, interactive, interactiveparams, 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, shadebetween, spacecurve, sparsematrixplot, surfdata, textplot, textplot3d, tubeplot*]

> *display([cpic, rpic])*

(23)



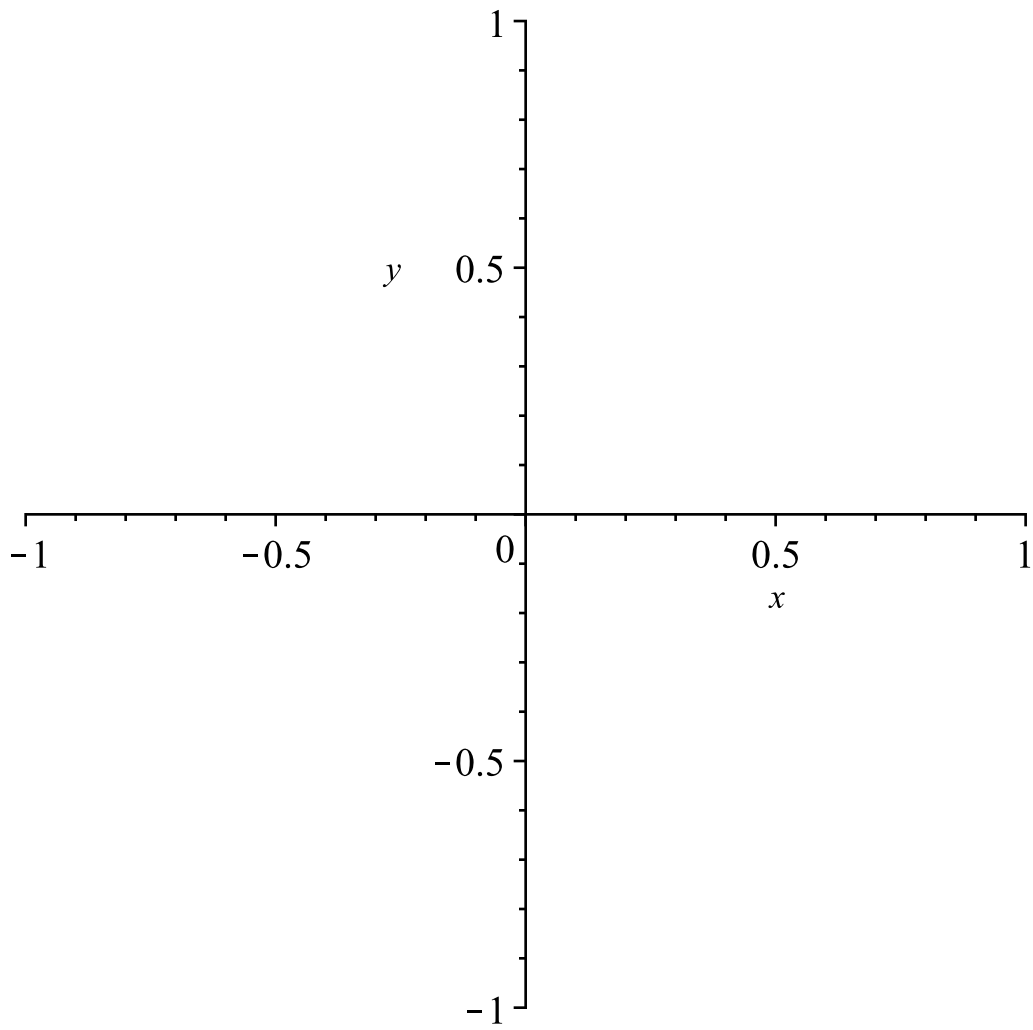
```
> display([cpic, rpic], view = [-1 .. 1, 0.5 .. 1.2], scaling = constrained)
```



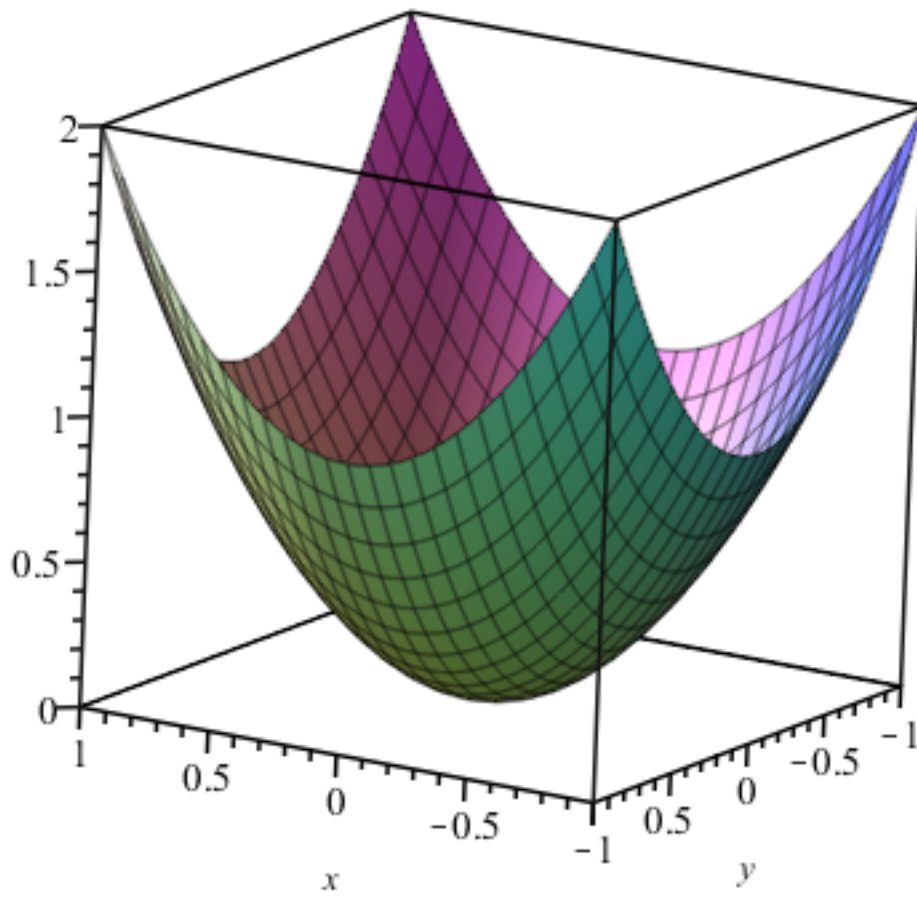
function of two variables

```
> plot(x^2 + y^2, x=-1..1, y=-1..1)
```

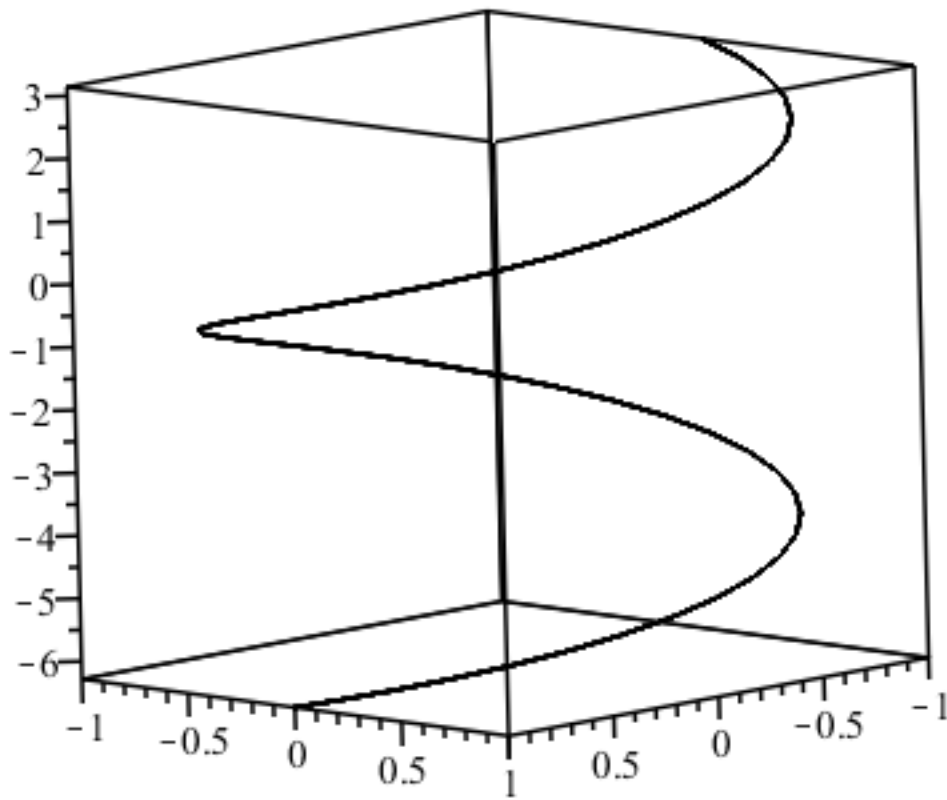
Warning, expecting only range variable x in expression  $x^2 + y^2$  to be plotted but found name y



```
> plot3d(x^2 + y^2, x=-1..1, y=-1..1)
```



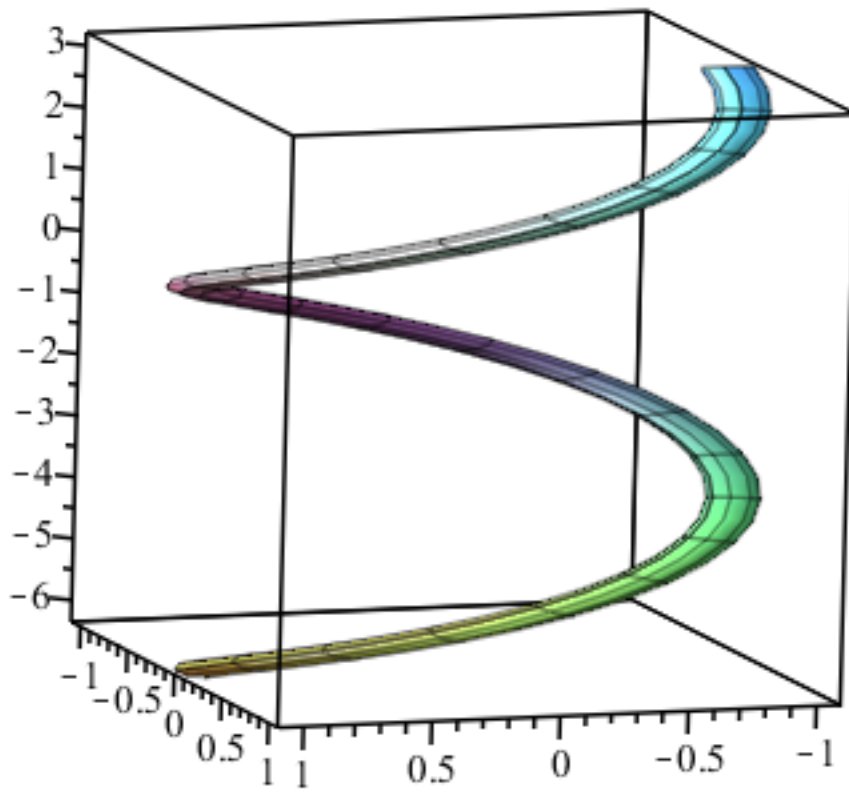
```
> plot3d( [cos(t), sin(t), t], t=-2 Pi..Pi, axes = boxed)
```



```
> ?tubeplot
```

```
> tubeplot([cos(t), sin(t), t], t=-2 Pi ..Pi, radius = .1, axes = boxed)
```





> `tubeplot`( $\left[ \cos(t), \sin(t), \frac{t}{5} \right], t = -2 \text{ Pi} .. \text{Pi}, \text{radius} = .1, \text{axes} = \text{boxed}, \text{scaling} = \text{constrained}$ )

