MATH 151 Lab 2

TA: Nida Obatake

In [1]:
```
from sympy import *
from sympy.plotting import (plot, plot_parametric,plot3d_parametric_surf
ace, plot3d_parametric_line,plot3d)
```

In [2]:
```
matplotlib notebook
```

1 Let $x = \cos(t)$ and $y = 3 + \sin(t)$.

a) Eliminate the parameter t to find a Cartesian equation of the curve involving x and y.

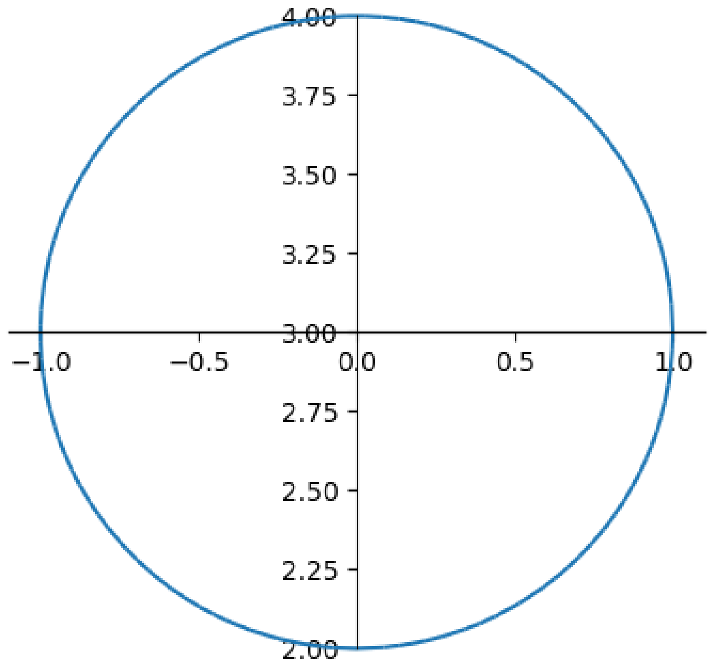
b) Sketch the graph of the curve represented by the parametric equations. Use an interval of $t \in [0, 2\pi]$ and create equal axes. Use the example found at http://calclab.math.tamu.edu/Python/Equal_Axes_Example.pdf (http://calclab.math.tamu.edu/Python/Equal_Axes_Example.pdf) for help.

c) Given this graph, rearrange the equation in part a) (by hand) to produce a more familiar equation of the graph in part b). Enter you answer in a print command.

In [3]:
```
x,y,t=symbols('x y t')
# 1a)
eq1 = x-cos(t)
eq2 = y-(3+sin(t))
tofx = solve(eq1,t)
#print(tofx) #we'll take the positive expression for t, so the second co
mponent
carteq = eq2.subs(t,tofx[1]) #equation.subs(variablereplaced,newvalue)
print("A Cartesian equation of the curve is {} = 0.".format(carteq))
# For each problem, you may insert additional cells as needed (under "In
sert" above)
```

```
A Cartesian equation of the curve is y - sqrt(-x**2 + 1) - 3 = 0.
```

In [4]:
```
# 1b)
graph = plot_parametric(cos(t),3+sin(t),(t,0,2*pi)) #plot_parametric(RHS of x eqn, RHS of y eqn, (var, min, max))
fig = graph._backend.fig
ax = graph._backend.ax
ax.set_aspect("equal")
```



In [5]:
```
# 1c)
print("Rearranging the Cartesian equation from 1a) by hand, we get the equation x^2+(y-3)^2=1 (a circle centered at (0,3) with radius 1).")
```

```
Rearranging the Cartesian equation from 1a) by hand, we get the equatio
n x^2+(y-3)^2=1 (a circle centered at (0,3) with radius 1).
```

2 Simplify the expression $\sin(2\arcsin(x))$. In a print command, explain how this answer would be obtained by hand.

In [6]:
```
# Insert code for problem #2 here.
print("The expression sin(2arcsin(x)) simplifies to {}.".format(trigsimp
(sin(2*asin(x)))))
print("To do this by hand, draw reference triangle with side x, hypotenu
se 1, and other side sqrt(1-x**2), then use the trig identity sin(2x)=2s
in(x)cos(x).")
```

```
The expression sin(2arcsin(x)) simplifies to 2*x*sqrt(-x**2 + 1).
To do this by hand, draw reference triangle with side x, hypotenuse 1,
and other side sqrt(1-x**2), then use the trig identity sin(2x)=2sin(x)
cos(x).
```

3 Given $f(x) = \dfrac{x^3 - 1}{\sqrt{x} - 1}$:

a) Use a **for** statement to evaluate the function at $x = 0.9, 0.99, 0.999$, then at $x = 1.1, 1.01, 1.001$ to numerically estimate $\lim_{x \to 1} f(x)$.

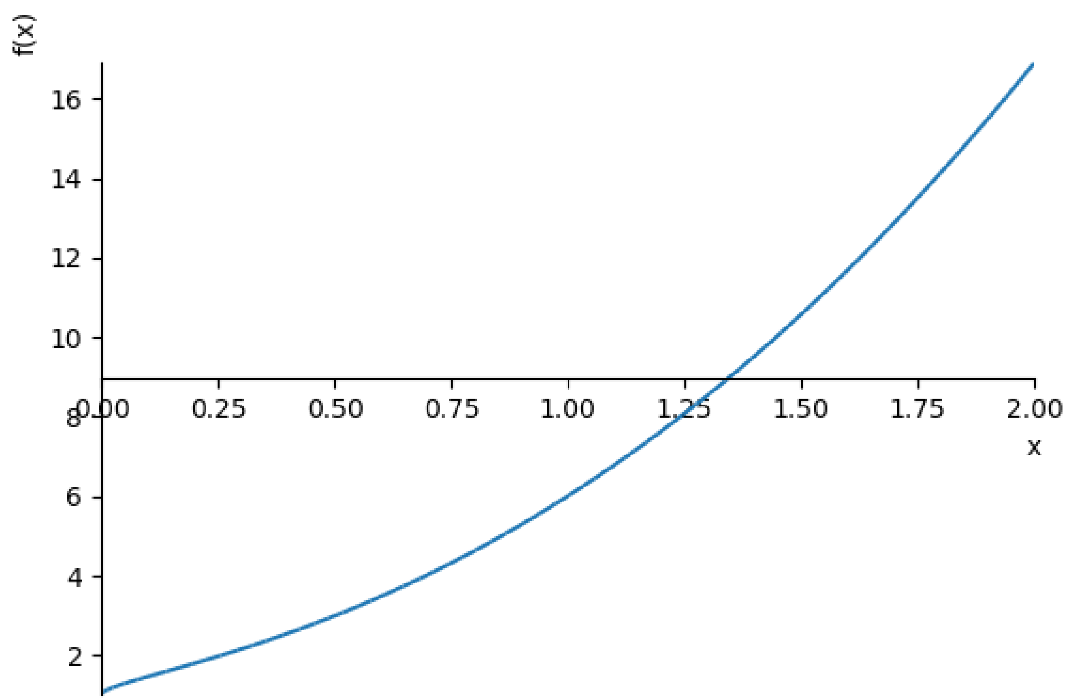
b) Plot f on the interval $[0, 2]$ to graphically verify your estimate in part a).

c) Compute $\lim_{x \to 1} f(x)$ exactly.

In [7]:
```
# 3a)
f = (x**3-1)/(sqrt(x)-1)
for i in [0.9, 0.99, 0.999]:
    print(f.subs(x,i))
print("As x approaches 1 from the left, the function values seem to be a
pproaching 6.")
for j in [1.1, 1.01, 1.001]:
    print(f.subs(x,j))
print("As x approaches 1 from the right, the function values seem to be
 approaching 6.")
```

```
5.28093173771689
5.92531218695038
5.99250312468795
As x approaches 1 from the left, the function values seem to be approac
hing 6.
6.78155728744320
6.07531281195596
6.00750312531191
As x approaches 1 from the right, the function values seem to be approa
ching 6.
```

In [38]:
```
# 3b)
plot(f,(x,0,2)) #plot_parametric(RHS of x eqn, RHS of y eqn, (var, min,
 max))
```



Out[38]: <sympy.plotting.plot.Plot at 0x115416860>

In [14]:
```
# 3c)
print("The limit is {}.".format(limit(f,x,1))) #limit(f(x), x, x0) compu
tes the limit of f(x) as x approaches x0
```

The limit is 6.