

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

USING DERIVATIVES IN PYTHON

You now have an introduction to the basics of Python (substitution, simplifying expressions, solving equations, and plotting graphs). In class, you have also started learning (or will soon learn) shortcut rules for derivatives. We will now begin to apply the basics of Python, learn how to take derivatives in Python, and solve typical Calculus problems in Python.

NOTE: The example here (and in future overviews) is NOT a copy/paste to solve the problems in lab. However, it will USE many of the features you will use to solve your problems, such as (in this case) differentiating functions, finding slopes and equations of tangent lines, solving equations, and plotting.

EXAMPLE:

Given $f(x) = x/(x+1)$:

- Find the derivative, $f'(x)$.
- Find the equation of the line tangent to f at $x=1$. Plot f and your tangent line on the same graph to check your answer. Use the domain $0 \leq x \leq 2$.
- Find the equations of the lines tangent to f which also pass through the point $(1,2)$.

Part a) is straightforward-take the derivative. In Python, the **diff** command finds your derivative.

```
In [2]: x=symbols('x')
        f=x/(x+1)
        df=diff(f,x)
        print("f'(x)=",df)
        # NOTICE that I used double-quotes for the text so I could type f'(x).
        # Remember single OR double-quotes work, but print('f'(x)=',df) will give a
        # n error! (Can you see why?)

        # We can also use our normal Python syntax of VARIABLE.COMMAND(OPTIONS).
        # I also simplified the answer to make it look more like how you would diff
        # erentiate by hand
        df=f.diff(x)
        print("The simplified version of f'(x)=",df.simplify())
```

```
f'(x)= -x/(x + 1)**2 + 1/(x + 1)
The simplified version of f'(x)= (x + 1)**(-2)
```

Before we solve parts b) and c), a **VERY IMPORTANT** public announcement:

Just because we are using a computer does NOT mean we don't have to think!!! In fact, the easiest way to approach most of these Python calculus problems is to ask yourself:

- 1) How would I solve this problem by hand?
- 2) What Python command(s) will allow me to do each step?

So how do we find the equation of a tangent line? To find the equation of a line, you need a point and a slope.

STEP 1) Substitute the x-value (x_0) into the function to get y_0 , the y-value (Key Python command: **subs**)

STEP 2) Find the derivative of f (Key Python command: **diff**, though we already did this step in part a)

STEP 3) Substitute the x-value into the derivative to get m , the slope (Key Python command: **subs**)

STEP 4) Use the formula $y - y_0 = m(x - x_0)$, or $y = y_0 + m(x - x_0)$

STEP 5) This problem also asks you to plot the function and tangent line (Key Python command: **plot**)

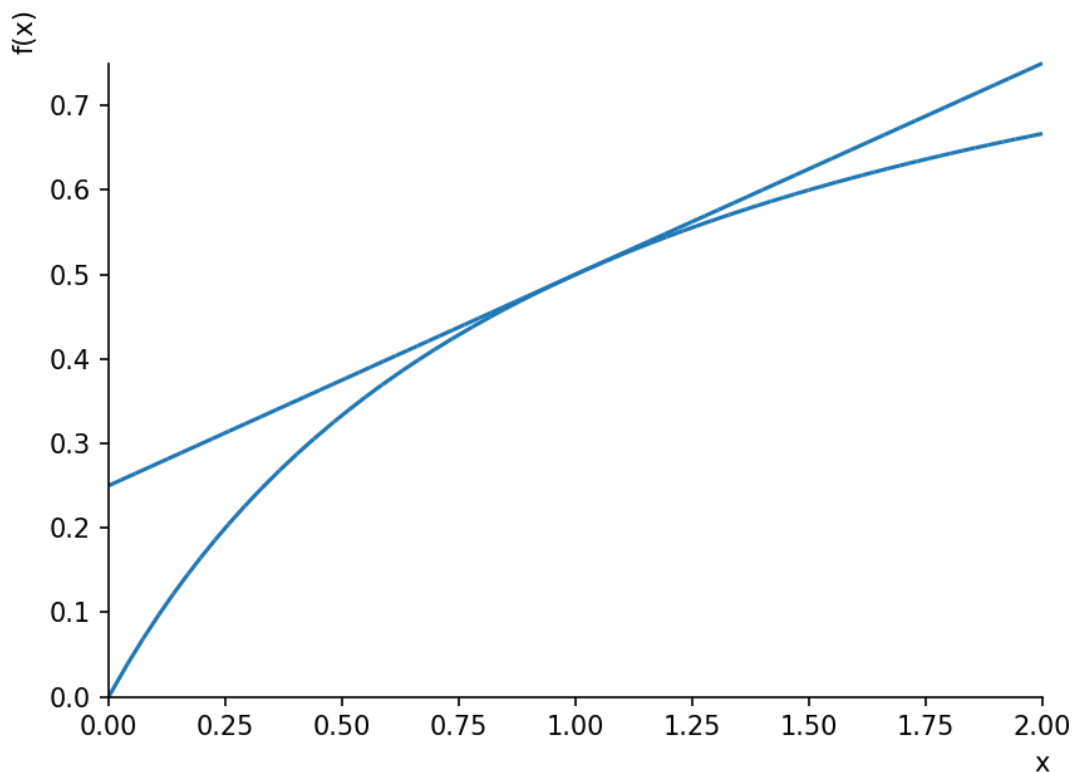
```
In [7]: # We have already defined f, but we should define x0 before we go to step 1
x0=1
y0=f.subs(x,x0)
# Step 2: already done, but we'll repeat it here to follow our algorithm above
df=f.diff(x)
# Step 3
m=df.subs(x,x0)
# Step 4: using the second version of the formula since we want to plot the function later
y=y0+m*(x-x0)
print('The equation of the line tangent to f at x=1 is y=',y)
```

The equation of the line tangent to f at $x=1$ is $y = x/4 + 1/4$

Now we want to plot. Remember in Jupyter to always run this command by itself before each plot so you get a graph and it doesn't appear on a previous graph instead.

```
In [8]: matplotlib notebook
```

```
In [10]: # To plot multiple graphs in one command, put the function and x-range in p  
arentheses. Recall this is called a tuple.  
plot( (f,(x,0,2)) ,(y,(x,0,2)) )  
# Notice that the line does appear to be tangent to the curve, so we have l  
ikely gotten the right answer!
```



```
Out[10]: <sympy.plotting.plot.Plot at 0x1b7a4921c88>
```

For part c, there are two strategies:

STRATEGY 1:

- 1) Find the equation of the tangent line at $x=a$ (define 'a' as symbolic and follow the steps in part b)
- 2) Substitute the point (1,2) into the equation (Key Python command: **subs**)
- 3) Solve for a (Key Python command: **solve**)

STRATEGY 2:

- 1) The slope of the tangent line at $x=a$ is $f'(a)$. (So do steps 2-3 in part b, call the slope m_1)
- 2) The slope can also be calculated since you know two points on the line: $(a, f(a))$ and $(1,2)$. (Define m_2 as change in y over change in x)
- 3) Set the slopes equal to each other and solve for a (Key Python command: **solve**)

Once you know a, you can find the equation as done in part b.

```
In [14]: #Solve using Strategy 1
# Strategy 1, Step 1
a,x=symbols('a x') #This is how you create multiple symbolic variables. Ca
n also do each on a separate line
# PAY ATTENTION to where there ARE commas (left side) and where there ARE N
OT commas (right hand side)
f=x/(x+1)
x0=a
# Notice Strategy 1, step 1 requires you to do part b again
# Part b, step 1
y0=f.subs(x,x0)
# Part b, step 2
df=diff(f,x)
# Part b, step 3
m=df.subs(x,x0)
# Part b, step 4
y=y0+m*(x-x0)
print('The equation of the tangent line is y=',y)
#HINT: It's a good idea to run your code so far to make sure there are no e
rrors!

# Strategy 1, Step 2
eqn=y.subs(x,1)-2 #NOTE that the y is on the other side, so we actually sol
ve y - 2 = 0
# Strategy 1, Step 3
avals=solve(eqn,a)
print(avals) # This Lets us see how many solutions there are
```

```
The equation of the tangent line is y= a/(a + 1) + (-a + x)*(-a/(a + 1)**2
+ 1/(a + 1))
[-2 - sqrt(3), -2 + sqrt(3)]
```

```
In [15]: # Solve using Strategy 2
# Strategy 2, Step 1: Requires steps 2-3 of part b
a,x=symbols('a x') #Not necessary, but not a bad idea in case you redefine
d x or a previously!
f=x/(x+1)
# Part b, Step 2
df=diff(f,x)
# Part b, Step 3
m1=df.subs(x,a)
# Strategy 2, Step 2
m2=(f.subs(x,a)-2)/(a-1)
print('The equation to solve is',m1,'=',m2) # Not required, but a good ide
a to check your work so far
# Strategy 2, Step 3
avals=solve(m1-m2,a)
print(avals)
```

The equation to solve is $-a/(a + 1)**2 + 1/(a + 1) = (a/(a + 1) - 2)/(a - 1)$
 $[-2 - \text{sqrt}(3), -2 + \text{sqrt}(3)]$

We see the 'a' values agree in both Strategies. Now find the equations of the tangent lines (repeat step b twice, changing x0 each time).

```
In [16]: x0=avals[0] #The FIRST number in the List-REMEMBER from the previous Lab th
at Python starts counting Lists at 0!
# Part b, Step 1
y0=f.subs(x,x0)
# We've done step 2 several times now, so we'll skip it here. Remember 'd
f' is the derivative.
# Part b, Step 3
m=df.subs(x,x0)
# Part b, Step 4
eqn1=y0+m*(x-x0)
print('The first tangent line is y=',eqn1,'Or approximately y=',eqn1.evalf
())
# Notice the "evalf" command to get decimal approximations.

# We now repeat the process, redefining x0
# ALSO: it is better programming practice if you give distinct names to you
r variables.
# Example: x0a and x0b, y0a and y0b, etc.
x0=avals[1] # The SECOND number in the List
y0=f.subs(x,x0)
m=df.subs(x,x0)
eqn2=y0+m*(x-x0)
print('The second tangent line is y=',eqn2,'Or approximately y=',eqn2.evalf
())
```

The first tangent line is $y = (1/(-\sqrt{3}) - 1) - (-2 - \sqrt{3})/(-\sqrt{3}) - 1)^2 * (x + \sqrt{3}) + 2 + (-2 - \sqrt{3})/(-\sqrt{3}) - 1$ Or approximately $y = 0.133974596215561*x + 1.86602540378444$

The second tangent line is $y = (-(-2 + \sqrt{3})/(-1 + \sqrt{3})^2 + 1/(-1 + \sqrt{3})) * (x - \sqrt{3}) + 2 + (-2 + \sqrt{3})/(-1 + \sqrt{3})$ Or approximatel
 $y = 1.86602540378444*x + 0.133974596215562$

In []: