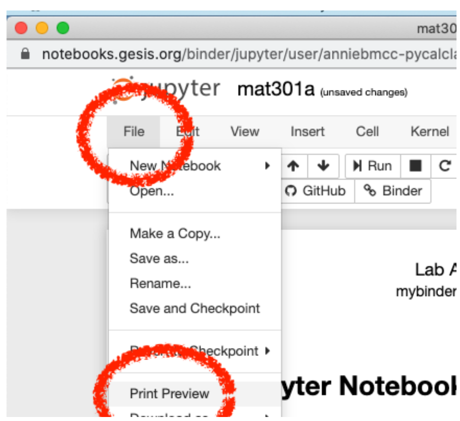
MAT301 Lab Directions

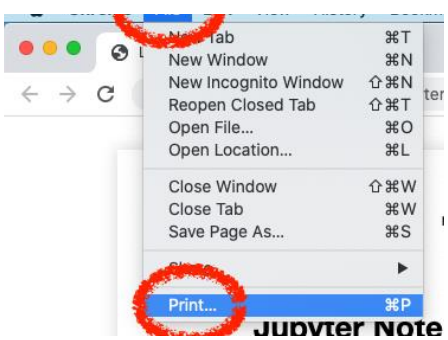
1. Click link in table of contents
   Wait for page to load

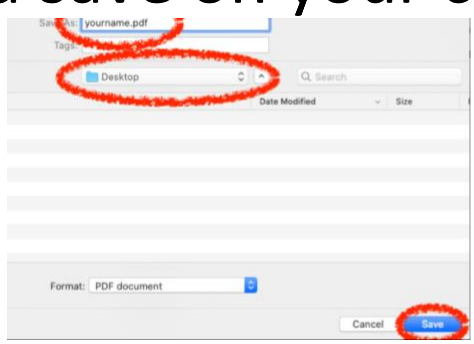
2. Run each cell that says RUN
   Type your answer to each EXERCISE

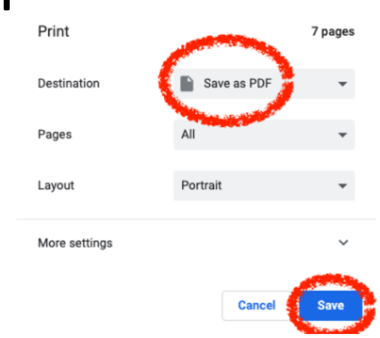
3. Open print preview



4. Print to PDF and save on your computer







5. Upload your PDF to blackboard

Table of Contents

Lab A: Using Python as a Calculator
mybinder.org/v2/gh/anniebmcc/pycalclab/master
2020 Summer — Calculus 1
Dr Matthew H Sunderland

Jupyter Notebooks

A1. **RUN** the following "code cell" (gray rectangle with `In[ ]` next to it), by CLICKING the code cell and pressing SHIFT+RETURN. Notice that only the last result will display.

In [ ]:
```
1 + 2 + 3
50 - 3
100*5
```

A2. **RUN** the following. As always, only the last result displays, but the last result has 2 parts because of the comma.

In [ ]:
```
1 + 2 + 3
50 - 3, 1000*1000
100*5, 7*7
```

A3. The "+" on the toolbar adds a code cell. The "scissors" deletes a cell.

Python arithmetic + - * / **

A4. **RUN** the following.

In [ ]:
```
3 + 10*5,  5**2,  27/10
```

A5. **EXERCISE.**
a) What does each of the 5 arithmetic operations do?
b) Do spaces around the 5 operations matter, or is it just style?

In [ ]:
```
# TYPE YOUR ANSWERS BELOW
#
#  a) + is
#     - is
#     * is
#     / is
#    ** is
#
#  b)
```

Python # and =

A6. **RUN** the following. You will notice python ignores everything after #

In [ ]:
```
# This is a comment
1 + 1 # This is also a comment
```

A7. **RUN** the following. Notice we assign variables using = Assignment itself does NOT produce output.

In [ ]:
```
a = 10
a
```

In [ ]:
```
b = 20
```

In [ ]:
```
a = 18
b = 21
c = a - b
c
```

A8. **RUN** the following. Notice you can assign multiple variables at once with a comma.

In [ ]:
```
x, y = 100, 500
x
```

In [ ]:
```
a,b,c = 3,4,5
a + b/c
```

A9. **RUN** the following. See that we can compute $\frac{(2-3)*-3}{-1+2}$ all at once (1st cell below), or we can assign variables to help us (2nd cell below).

In [ ]:
```
(2 - 3)*-3/(-1 + 2)
```

In [ ]:
```
top = (2 - 3)*-3
bottom = -1 + 2
top/bottom
```

A10. **EXERCISE.** Assign variables to help you compute $3 - \frac{3^2-2\cdot 3}{2\cdot 3-2}$

In [ ]:
```
# Type your answer below and press SHIFT+ENTER
```

Order of Operations

A11. **RUN** the following. Notice $a - b * c = a - (b * c)$, but they do not equal $(a - b) * c$.

In [ ]:
```
a,b,c = 3,4,5

a - b*c,   a - (b*c),   (a - b)*c
```

A12. **EXERCISE.** In each row, identify NON-equivalent choice. For example, the answer to (1) is $(a - b) * c$ because $a - b * c = a - (b * c)$

(1)	$a - b * c$	$a - (b * c)$	$(a - b) * c$
(2)	$a * (b - c)$	$(a * b) - c$	$a * b - c$
(3)	$a / b + c$	$a / (b + c)$	$(a / b) + c$
(4)	$(a + b) / c$	$a + (b / c)$	$a + b / c$
(5)	$a ** (b * c)$	$(a ** b) * c$	$a ** b * c$
(6)	$a * (b ** c)$	$a * b ** c$	$(a * b) ** c$
(7)	$a/b ** c$	$(a/b) ** c$	$a/(b ** c)$
(8)	$a ** b/c$	$(a ** b)/c$	$a ** (b/c)$
(9)	$(3 - 3) - 3$	$3 - 3 - 3$	$3 - (3 - 3)$
(10)	$(2 ** 3) ** 2$	$2 ** (3 ** 2)$	$2 ** 3 ** 2$
(11)	$6 / 3 / 2$	$6 / (3 / 2)$	$(6 / 3) / 2$

In [ ]:
```
# TYPE YOUR ANSWERS BELOW.
#
#  (1)   (a - b)*c
#  (2)
#  (3)
#  (4)
#  (5)
#  (6)
#  (7)
#  (8)
#  (9)
# (10)
# (11)
```

A13. **RUN** the following example, where we add 2 sets of parentheses which show the order of the 2 operations.

In [ ]:
```
1 + 3/5
```

In [ ]:
```
(1 + (3/5))
```

A14. **EXERCISE.** Add 4 sets of parentheses, which show the order of the 4 operations.

In [ ]:
```
7 - 3 ** 2/9 + 4
```

In [ ]:
```
# Type your answer below and press SHIFT+ENTER
```

A15. **EXERCISE.** Assign `a,b,c = 4,5,8` and then evaluate $\dfrac{a^b - c/b}{c-a}, \dfrac{a^{c-b}}{c-b}, \dfrac{a^{3/2}}{b}, \dfrac{a-b(c-a)}{c-a}$

In [ ]:
```
# Type your answer below and press SHIFT+ENTER
```

Making python functions

A16. **RUN** the following.

In [ ]:
```
def g(x):
    return x**2

g(7)
```

In [ ]:
```
def h(n): return n + 100

h(7)
```

A17. **EXERCISE.** Make the function $P(x) = x^2 - 2x + 1$ and find $P(P(7))$.

In [ ]:
```
# Type your answer below and press SHIFT+ENTER
```

Built-in `%pylab` functions

Python	Math notation	Meaning
`abs(x)`	$\|x\|$	absolute value
`sqrt(x)`	$\sqrt{x}$	square root
`exp(x)`	e^x	exponential function
`log(x)`	$\ln x$	natural logarithm
`sin(x)`	$\sin x$	sine
`arcsin(x)`	$\sin^{-1} x$	inverse sine
`radians(x)`		converts degrees to radians

A18. **RUN** the code cells below. The command `%pylab` only needs to be run once per lab; it loads "built-in functions" (from python packages numpy and matplotlib).

In [ ]:
```
%pylab

sqrt(49)
```

In [ ]:
```
pi, exp(1), sin(pi/2)
```

A19. **EXERCISE.** Evaluate

1. $\sin 40^\circ$
2. $\sin^2 65^\circ$
3. $e^{(10-8.5)/3}$
4. $\arcsin(\sin(3\pi/4))$

Note. Python uses radians for all angle measurements, so you need to convert any degrees to radians.

In [ ]:
```
# Type your answer below and press SHIFT+ENTER
```

Making an array with `r_[ ]`

A20. **RUN** the following. (If you get an error, go back and run A17.) The function `r_[ ]` can make an array of numbers of your choice. We will need arrays for graphing (Lab B).

In [ ]:
```
x = r_[2,3,4,5,10]
x**3
```

A21. **EXERCISE.** Use `r_[ ]` to store the numbers 2,3,5,7,11 in an array named `x`. Find `x*x`.

In [ ]:
```
# Type your answer below and press SHIFT+ENTER
```

Making an array with `r_[a:b:stride]`

A22. **RUN** the following. In general, `r_[a:b]` will list integers from a up to but *not* including b. A missing a is the same as 0.

In [ ]:
```
r_[5:10]
```

In [ ]:
```
r_[:5]
```

A23. **EXERCISE.** Use `r_[a:b]` to make the array 1,2,3,4,5,6,7,8,9

In [ ]:
```
# Type your answer below and press SHIFT+ENTER
```

A24. **RUN** the following. In general, `r_[a:b:stride]` spaces out your numbers by the amount `stride` .

In [ ]:
```
r_[0:100:2]
```

A25. **EXERCISE.** Use `r_[a:b:stride]` to make the array $1, 3, 5, \ldots, 99$

In [ ]:
```
# Type your answer below and press SHIFT+ENTER
```

Making an array with `linspace(a,b,n)`

A26. **RUN** the following. Observe that `linspace(a,b,n)` lists n numbers from a to b inclusive. This is useful for generating a lot of evenly-spaced numbers, such as when graphing (Lab B). Observe that `linspace(a,b)` lists 50 numbers from a to b inclusive.

In [ ]:
```
linspace(0,10,6)
```

In [ ]:
```
linspace(0,10)
```

A27. **EXERCISE.** Use `linspace(a,b,n)` to make the array $1, 1.5, 2, 2.5, 3, 3.5, 4$

In [ ]:
```
# Type your answer below and press SHIFT+ENTER
```

A28. **EXERCISE.**
Convert average body temperature $98.6^{\circ} F$ to Celsius using $C = 5/9(F - 32)$.

In [ ]:
```
# Type your answer below and press SHIFT+ENTER
```

A29. **RUN** the following.
Notice that `x` and `y` are arrays,
`c_[x,y]` puts them into a table.

In [ ]:
```
x = r_[:10]
y = x**2
c_[x,y]
```

A30. **EXERCISE.**
Use `r_` to make an array of Fahrenheit values `x` $= -100, -80, -60, \ldots, 100.$
Make the corresponding array of Celsius values `y`
Use `c_` to put `x` and `y` into a table.

In [ ]:
```
# Type your answer below and press SHIFT+ENTER
```

Lab B: Plotting Graphs in Python
2020 Summer — Calculus 1
Dr Matthew H Sunderland

B1. If you are reading this as a PDF,
open the live notebook remotely on Binder [here (https://mybinder.org/v2/gh/mattsunderland/pycalclab/master)](https://mybinder.org/v2/gh/mattsunderland/pycalclab/master).

Plotting with `plot`

B2 Example. To graph $f(x) = x^2$ over $[-2, 2]$ by hand, make an xy table: choose some x values,

x	-2	-1	0	1	2
y					

and then use f to compute the corresponding y values.
Graphing in python is similar. **Run the following (SHIFT+ENTER), then run it again.**

In [ ]:
```
%pylab inline

x = r_[-2, -1, 0, 1, 2]
y = r_[4, 1, 0, 1, 4]
plot(x,y)
```

B3. Run the code cell below. Notes:

- We make the `x` with `linspace(a,b,n)` , which gives `n` numbers evenly spaced `a` to `b` inclusive (see A30)
- We make the `y` by doing arithmetic on `x`
- `plot(x,y)` makes the plot
- `title()` adds a title
- `grid()` add a grid
- `r_[[x,y]]` displays an xy table

In [ ]:
```
x = linspace(-2,2,9)
y = x**2

plot(x,y)
title('$f(x) = x^2$ plotted with 9 points')
grid()

r_[[x,y]]
```

B4 Example. Run the following cells, which graph $f(x) = e^x$ over the interval $[0, 7]$.

- `r_[a:b:stride]` gives you the numbers **from** a **up to but *not* including** b spaced `stride` apart (see A28)
- `exp(x)` is how you write e^x in python (see A21)

In [ ]:
```
x = r_[:7]
y = exp(x)
plot(x,y)
```

B5. Run the following.
When we change the x we must recompute the y;
there are two ways to do it (compare B4 to B3).

In [ ]:
```
x = r_[:10]
plot(x, exp(x))
```

B6 Exercise. We want to graph $y = \cos 4x$ over $[0, \pi]$ with a step size of `pi/10` .

i. Which command gives the desired values for x? (a) x = 0:pi/10:pi (b) x = 0:pi:pi/10 (c) x = linspace(0,pi)
ii. Which gives the correct answer for y? (a) y = cos(4x) (b) y = cos4*x (c) y = cos(4*x)
iii. Plot the graph.
iv. Redo your plot from iii. using `x = linspace(0,pi)`
v. Which plot looks more like the plot of a cosine curve?
(a) The first one, (b) the second one, (c) both of them.

In [ ]:
```
# i. Type your answer in this comment:
# ii. Type your answer in this comment:

# iii. Type and run your code here.
```

In [ ]:
```
# iv. Type and run your code here.

# ii. Type your answer in this comment:
```

B7 Exercise.

i. Plot the function $f(x) = e^{\cos x}$ over the interval $[0, 2\pi]$.

ii. What command generates a sufficient number of values for x?
(a) `linspace(0,2*pi)` (b) `linspace(0,100,2*pi)`
(c) `r_[0:2*pi]` (d) `r_[0:0.01:2*pi]`

iii. Which command will generate the corresponding yvalues?
(a) `exp^cos(x)` (b) `e^cos(x)` (c) `exp(cos(x))` (d) `exp(x)cos(x)`

In [ ]:
```
# i. Type and run your code here.

# ii. Type your answer in this comment:
# iii. Type your answer in this comment:
```

Doing arthmetic on arrays

B8. Run the following.
We make numpy arrays with `r_` or `linspace`
Numpy arrays "know" how to do "elementwise" arithmetic.
Warning: x^2 is written `x**2`.

In [ ]:
```
x = r_[1:5]
x, 10 - x, x + 10, 10*x, x**2, 12/x, x**x, 10**x
```

B9. Run the following.

In [ ]:
```
# We can add arrays of the same shape (same length)

x = r_[10, 20, 50, 100]
y = r_[3, 0, 7, -1]
x + y
```

In [ ]:
```
# We can add an array (x) and a scalar (y)

x = r_[10, 20, 50, 100]
y = 100
x + y
```

In [ ]:
```
# We CANNOT add arrays of DIFFERENT shape

x = r_[10, 20, 50, 100]
y = r_[3, 0, 7]
x + y
```

B10 Examples. Run the cells below, which plot the following.

- $y = \sin x + \cos 3x$ over the domain $[0, 2\pi]$
- $y = e^{-x/2} \cos 6x$ over the domain $[0, 10\pi]$
- $y = 1/(x^2 - 1)$ over the domain $[2, 5]$

In [ ]:
```
x = linspace(0,2*pi)
y = sin(x) + cos(3*x)
plot(x,y)
```

In [ ]:
```
x = linspace(0, 10*pi, 300)
y1 = exp(-x/2) # Here we break up the
y2 = cos(6*x)  # computation into
y = y1*y2      # bite-sized pieces
plot(x,y)
```

In [ ]:
```
x = r_[2:5.1:0.1]
y = 1/(x**2 - 1)
plot(x,y)
```

B11 Exercise. Define a, b, and c by

```
a = r_[1:21:2]
b = r_[1:11]
c = r_[1:12:2]
```

Which of the following is/are defined?
(a) `b+c` (b) `a + b` (c) `a./ b` (d) `a * b` (e) `a ^ 2`

In [ ]:
```
# Type your answer in this comment:
```

B12 Example. Let x be the array 1,2,3. Write Python commands to compute x^3.
The output you get should be `array([ 1,  8, 27])`.

In [ ]:
```
x = r_[1,2,3]

# Write out and run your code here.
x**3
```

B13 Exercise. Let x be the array 1,2,3.

i. Write Python commands to compute $\cos x \sin x$.
You should get `array([ 0.45464871, -0.37840125, -0.13970775])`

ii. Write Python commands to compute $\sin^2 x$.
You should get `array([0.70807342, 0.82682181, 0.01991486])`

iii. Write Python commands to compute $\sin x^2$.
You should get `array([ 0.84147098, -0.7568025 ,  0.41211849])`

iv. Write Python commands to compute $7x^2 \sin \frac{1}{7x^2}$.
You should get `array([0.99660211, 0.99978743, 0.99995801])`

v. Write Python commands to compute $x - \frac{\cos x - \sin x}{\sin x + \cos x}$
You should get `array([1.2179581 , 4.68770694, 1.66751188])`

vi. Write Python commands to compute $\frac{1}{10}(x - \frac{x^{3/2}}{10})^2$
You should get `array([0.081     , 0.29486292, 0.61523085])`

```
In [ ]: # i. Write out and run your code here.
```

```
In [ ]: # ii. Write out and run your code here.
```

```
In [ ]: # iii. Write out and run your code here.
```

```
In [ ]: # iv. Write out and run your code here.
```

```
In [ ]: # v. Write out and run your code here.
```

```
In [ ]: # vi. Write out and run your code here.
```

B14 Exercise.
i. Graph the function $f(x) = \sin(\frac{\pi}{2}x) + \sin(\frac{2}{5}\pi x)$ over the interval $[0, 40]$.

ii. How many peaks (relative maxima) does your graph B21 have?
(a) 2 (b) 3 (c) 4 (d) 5 (e) none of the above

iii. The function in B21 is periodic; how many periods are graphed in $[0, 40]$?
(a) 2 (b) 3 (c) 4 (d) 5 (e) none of the above

iv. Estimate from your graph B21 the value of $f(10)$ to 1 decimal point.

In [ ]:
```
# i. Make your graph here.

# ii. Type your answer in this comment:

# iii. Type your answer in this comment:

# iv. Type your answer in this comment:
```

B15 Exercise.
i. Graph the function $f(x) = \cos^2 x - \sin^2 x$ over the interval $[-2\pi, 2\pi]$. Use 50 points in the domain.
ii. Does the graph B24 resemble any graph that you are familiar with?
(a) $\cos 2x$ (b) $\cos x/2$ (c) $\cos x$

In [ ]:
```
# i. Make your graph here.

# ii. Type your answer in this comment:
```

B16 Exercise.
i. Plot the polynomial function $f(x) = x^3 - 20x^2 + 10x - 1$ over the interval $[-10, 10]$.
ii. What is the approximate range for the y-axis?
(a) $[-10, 10]$ (b) $(-10, 10)$ (c) $[-3100, 0]$ (d) $[0, 2\pi]$

In [ ]:
```
# i. Make your graph here.

# ii. Type your answer in this comment:
```

B17 Exercise. We wish to investigate when (if) the is positive.
We can't readily tell from our graph B27 so we will replot over a smaller domain.

i. Which of these domains seems appropriate for this task?
(a) $[0, 500]$ (b) $[0, 10]$ (c) $[-1, 1]$ (d) $[0, 2\pi]$

ii. Replot the graph over the selected domain. Turn on the grid using `grid()`

iii. From your graph, which of these x values have $f(x) > 0$? Indicate all that apply:
(a) 0 (b) 0.25 (c) 0.50 (d) 0.75

In [ ]:
```
# i. Type your answer in this comment:

# ii. Make your graph here.

# iii. Type your answer in this comment:
```