Lecture 1: Introduction

Math 98

Introduction

Instructor: Andrew Shi
Email: andrewshi@math.berkeley.edu (Do NOT use bCourses mail)
Website:
https://math.berkeley.edu/∼andrewshi/math98/sp22/98-sp22.html

Lectures, Grading and Deliverables

1. 5 Lectures (prerecorded and asynchronous)
   - Every Tu/Th, starting Tues (01/25)
2. 4 Homeworks
   - Due Tu/Th at 11:59pm, 7 days after corresponding lecture.
   - Graded largely on effort and completion
   - Assignment Submission: on bCourses
     - ★ Solutions appear 15 minutes after deadline.
   - Can be somewhat flexible with deadlines (within reason)
     - ★ But in your best interest to complete course quickly.
3. 1 Project
   - Graded for accuracy.
   - Expectation is that you get this *completely* correct as warmup for 128a PAs.
4. Receiving a P in this class should not be hard
   - Each HW worth 15 points, Project worth 40.
   - Must receive minimum of 70% on each HW and project.
   - Start early, work hard, and seek help on the assignments.

Who should take this course

1. If you have no/minimal programming experience:
   - This is the target audience of this course.
   - We will build things up from scratch.
   - I also expect many of you to also feel a bit nervous about programming (and that's OK!)
2. If you have extensive programming experience in another language:
   - This course likely won't be a good use of your time.
   - Easy to pick up MATLAB syntax yourself. Work through an online tutorial. (see suggestions on my webpage)
3. For everyone else in between.....
   - Take a look at some of the homeworks and see if you could mostly code them up in another language.

Why should take this course

Why take this course? (vs. some other online course)

1. Condensed focus that covers the necessary material for Math 128a.
2. Assignments that are well aligned with Math 128a.
   - Especially the final project.
3. Availability of customized help (from me!) in office hours
   - Even if you can figure out the problem, it is extremely useful to get feedback on your thought process and areas for improvement. So make sure to ask for it!

There is no real need to formally enroll for 1 unit (unless you want it). I will be happy to grant access to the course material to auditors and speak with them in office hours.

Remote Instruction

This course is completely asynchronous.

- All the lecture slides and accompanying videos are available. Go through the videos in order.
  - Mix of Coursera videos and my own content.
  - Should have MATLAB open to follow along and to do "in-class" exercises.

Office Hours

- By appointment, email within 24 hours (see webpage for ZOOM ID).
  - Very useful for us to talk through thought process.
  - Highly encouraged for you to get help.

Course Expectations

This is a fairly intense course. In 3 weeks you will go from no programming experience to being able to complete a 128A Programming Assignment.

- I expect you to set aside a block of $\sim$2-3 hours at least every other day to practice programming (including the lectures).
  - Like taking a foreign language class. Immersion.
- This will pay off later in the course!! (and in life).

Important: Practice. The best way to learn how to code is to **write lots of code**.

Webpage

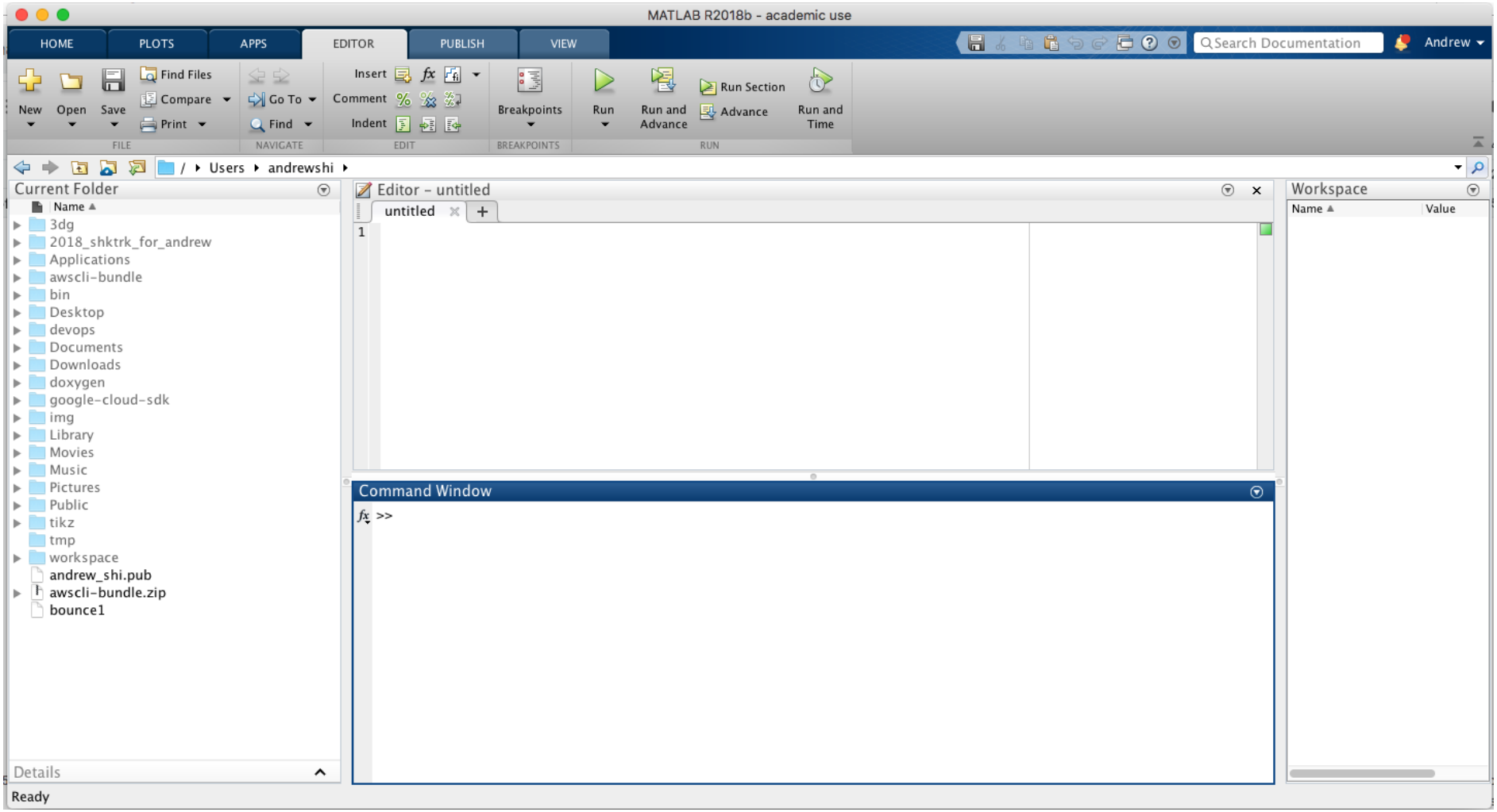
Go over items on webpage:
https://math.berkeley.edu/∼andrewshi/math98/sp22/98-sp22.html

Per Section 102.23: Course Materials of the UC Berkeley Student Code of Conduct - don't post any of these materials to CourseHero (or any similar sites).

Downloading MATLAB

- MATLAB is available for free for UC Berkeley students via a campus license.
- Make sure to download ASAP in case issues arise.

MATLAB Intro: Opening up MATLAB



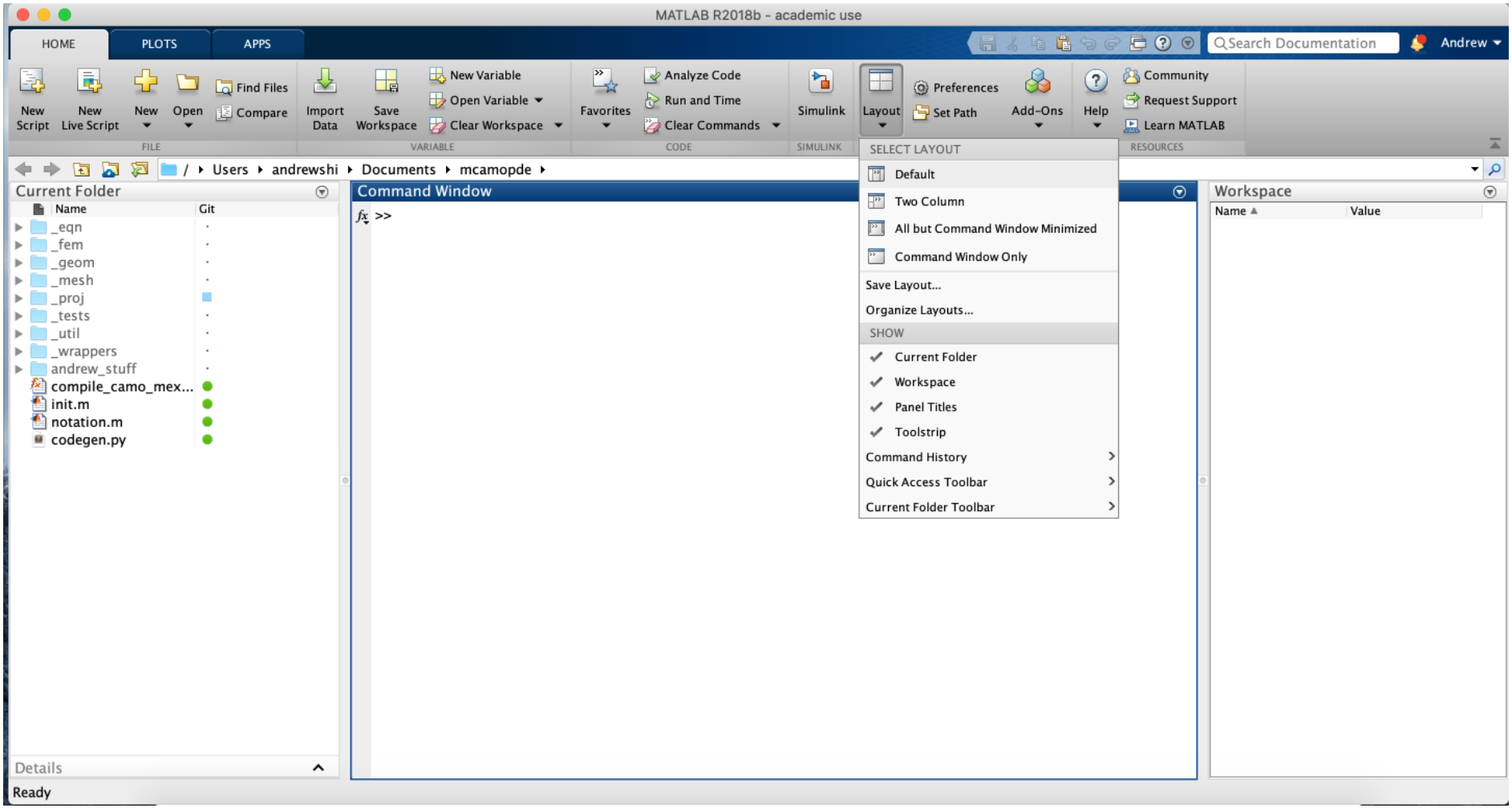
MATLAB Intro: Getting started with MATLAB

Matlab has five features:

1. **Current Folder** shows the files that MATLAB is accessing. By default MATLAB cannot execute files contained in other folders.
2. **Command Window** Here we can define variables, perform calculations, and much more.
3. **Workspace** is a MAT-file containing the variables you have defined in your session.
4. **Editor** allows us to save collections of commands as M-files.
5. **Command History** can be accessed using the up arrow.

It's OK if this doesn't make much sense yet.....we will revisit this.

MATLAB Intro: Reset Layout



If something happens to this, you can reset the layout to the default configuration.

MATLAB Intro: Command Window

To begin with, think of MATLAB as a giant calculator.
Operations: $+, -, *, /, \hat{\cdot}, \exp(\cdot), \sqrt{\cdot}, \log(\cdot)$

```
>> 2 - 3*(1+2)/2
ans =
-2.5000
>> 3^4
ans =
81
>> log(4)
ans =
1.3863
>> sqrt(9.01)
ans =
3.0017
```

MATLAB Intro: Help

Wait, what kind of logarithm is that?

```
>> help log
 log    Natural logarithm.
    log(X) is the natural logarithm of the elements of X.
    Complex results are produced if X is not positive.

    See also log1p, log2, log10, exp, logm, reallog.

    Reference page for log
    Other functions named log
```

help is going to be your best friend when using MATLAB.

MATLAB Intro: MATLAB Documentation

- The online documentation is also excellent with many examples.
- Google search "matlab log"
  - https://www.mathworks.com/help/matlab/ref/log.html
- Knowing where to find the answer is much more important than knowing the answer.

MATLAB Intro: Built in MATLAB Functions

- Many built in functions in MATLAB to do math.

`cos(x)`	Cosine	`abs(x)`	Absolute value
`sin(x)`	Sine	`sign(x)`	Signum function
`tan(x)`	Tangent	`max(x)`	Maximum value
`acos(x)`	Arc cosine	`min(x)`	Minimum value
`asin(x)`	Arc sine	`ceil(x)`	Round towards $+\infty$
`atan(x)`	Arc tangent	`floor(x)`	Round towards $-\infty$
`exp(x)`	Exponential	`round(x)`	Round to nearest integer
`sqrt(x)`	Square root	`rem(x)`	Remainder after division
`log(x)`	Natural logarithm	`angle(x)`	Phase angle
`log10(x)`	Common logarithm	`conj(x)`	Complex conjugate

Variables: ans

The workspace shows variables that have been defined in the current session. In particular, `ans` is by default the value of the last arithmetic computation we made. We can check the value of a variable by entering its name in the command window.

```
>> 1 + 3
ans =
4
>> ans
ans =
4
>> 3 * 8
ans =
24
>> ans
ans =
24
```

Variables: Defining your own

We can define our own variables, too! Variable names must start with a letter and can contain letters, digits, and underscores. MATLAB is case sensitive but all built-in MATLAB functions use lowercase letters, so if you use at least one capital letter in your variable names you can be sure to avoid any name conflicts.

```
>> x1 = 5.337
>> my_variable = "howdy"
>> frodoBaggins33 = sqrt(2)*pi
>> x2 = x1 + 1
```

Variables: Built in Variables I

There are many built in variables too.

```
>> pi
ans =
3.1416
>> eps
ans =
2.2204e-16
```

You can override these if you want....

```
>> pi = 4
>> pi
pi =
4
```

.....but this is usually a bad idea. Try to avoid ambiguity.

Variables: Built in variables II

There are two more special built in variables: `Inf` and `NaN` (not a number)

```
>> 1/0
ans =
Inf
>> -1/0
ans =
-Inf
>> 0/0
ans =
NaN
>> Inf/Inf
ans =
NaN
```

What is `Inf/0`? What about `exp(exp(7))`? Compare to `exp(exp(6))`.

Variables: Using informative names

Give your variables informative names. Good.

```
>> radius = 4
>> area = pi*radius^2
area =
50.2655
```

Bad.

```
>> foo = 4
>> blah = pi*foo^2
blah =
50.2655
```

Variables

Use a semicolon to suppress the output of a command. Using `disp` will suppress the `ans =` text, but will also not save the output to `ans`. Multiple commands can be placed on a line separated by semicolons.

```
>> x = 5; y = 6; disp(x+y)
11
```

Use SHIFT-ENTER to start a new line. Use ellipses (...) and SHIFT-ENTER to continue a line.

```
>> sqrt(5 + 7 + ...
13)
ans =
5
```

Variables

Use `clear` to clear all variables from the workspace. Use `clear VAR1 VAR2` to clear specific variables `VAR1` and `VAR2`.

```
>> x = 5; clear x;
>> x
Undefined function or variable 'x'.
```

Use `clc` to clear the command line.

Formatting: long and short

Doesn't `pi` have more digits than this?

```
>> pi
ans =
3.1416
```

This is just a formatting/visual thing. Try format long to show 15 digits.

```
>> format long
pi
ans =
3.141592653589793
```

Then try format short.

```
>> format short
pi
ans =
3.1416
```

Formatting: compact and loose

Go into MATLAB and do stuff in the command window. Do you notice all those blank lines in between the command window output?
Experiment with `format compact` and `format loose`.

Relations

The following statements will take value 0 (if false) or 1 (if true)

- $a < b$: a less than b
- $a > b$: a greater than b
- $a <= b$: a less than or equal to b
- $a >= b$: a greater than or equal to b
- $a == b$: a equal to b (note the doubled equals sign!)
- $a \sim= b$: a not equal to b

Logical Statements

- `and(a,b)` or equivalently `a & b`
- `or(a,b)` or equivalently `a | b`
- `not(a)`
- `xor(a,b)`

What do the commands `&&` and `||` do?

Boolean Expressions

A boolean expression is any expression involving relations or logical statements:

$$((4 <= 100)|(-2 > 5))\&(\texttt{true}|\sim\texttt{false})$$

Boolean expressions evaluate to 1 for true and 0 for false. Note that 0 and 1 are just numbers and are not in a separate class for logicals.

```
>> 5 + true
ans =
6
```

The order of operations is as follows:

1. negation
2. relations
3. and
4. or

Editor

Okay, it's finally time to make a proper script! In the command line, enter

```
>> edit Hello.m
```

Or New → Script (Top Left)

- An m-file is a file containing a script for MATLAB—a sequence of instructions for the computer.
- The name of the file must have the format `filename.m`.
  - Filenames are case sensitive (like everything in MATLAB).
  - `filename.m` and `Filename.m` are different!
- For MATLAB to execute the file, it must be saved in the Current Directory.

Two ways to run a script:

- Open up the script and hit "Run".
- Type the name of the script into the Command Window.

Commenting with %

Except within a string, everything following `%` is a comment. Comments do not get executed when the program runs, but can make the code easier to read by providing information about its organization and usage.

Comments in the beginning lines of a program will be revealed when using the command `help`.

`help` (as well as `doc`) is also invaluable when learning how to use various MATLAB functions.