General Coding Concepts

MATH-151: Mathematical Algorithms in Matlab

August 23, 2023

STATEMENTS AND COMMANDS

- We communicate with the computer using a series of **statements** or **commands** that instruct the computer what to do

    - It will do *exactly* and *only* what you command it to do

Some examples of commands we'll make use of

- Arithmetic operators

    - 6 + 3
    - 48 - 13
    - 6*17
    - 4096/4
    - 12^2

- Control statements

    - if ... else ...
    - for ...
    - while ...

- Mathematical functions

    - sin(pi)
    - cos(pi/4)
    - exp(2)
    - log(1)
    - sqrt(120)

- Presentation statements

    - display(18/3 + 7)
    - plot(x,y)

VARIABLES AND ASSIGNMENT

- Commands are great, but they aren't very useful as displayed on the previous slide. The real power of computers is the ability to keep track of many values as we perform many operations!

- To do this, instead of using numbers directly, we often will apply those values to variables. For example

  - x = 7; tells the computer that wherever it sees x to use 7 instead.

  - Similarly we can set y = 10; and see how they interact!

```
>> x = 7;
>> y = 10;
>> x + y
ans =
    17
>> x*7
ans =
    49
```

- This is called assigning a value to a variable. A very important use is that we can use those variables throughout our code, and even use it to update the variable itself!

  - For example x = x + 2; is interpreted as "take x, add 2 to it, and that is our new x"

  - In Matlab = is used for assignment, so its a little different than the = you've seen in the past. Some languages use <- for assignment to make this more clear.

```
>> x = 7
x =
    7
>> x = x + 2
x =
    9
>> x = x*2 + x/3
x =
    21
```

THE END [OF A STATEMENT]

- There are two ways to indicate to Matlab that a statement is complete.

    - End of the line
        - Press the enter key! Matlab will see an invisible character called the "Line Feed" and know the statement is complete
        - Matlab will perform this statement **and print the result to the Command Window**

    - Semicolon (;)
        - A Semicolon indicates to Matlab that a command is finished and **do not print result to Command Window**
        - Eventually, most of your commands will end with a semicolon.

```
>> x = 7;
>> y = 10;
>> x + y
ans =
    17
>> x*7
ans =
    49
```

COMMENTS

- When writing code, it is always important to remember that you will have to pass it along to someone else at some point. So it is good to make sure your code is easy to read and understand
    - You'll probably find that "other person" is yourself at a later date!
- Comments are lines in code that are ignored by the computer, but allow us to describe in words what we intend for the code to do
- In Matlab, we use the symbol % to tell the computer that everything afterwards on that line is a comment
- Some helpful guidelines on effective commenting
    - Describe what variables represent (also use descriptive variable names as your codes become more complicated)
    - Focus on **what** code is doing, rather than **how** it is doing it

CONFUSERS

- Here's two quotes I keep in mind when computing
    - "I call this a confuser, because it confuses you into thinking it has the right answer"
    - "All models are wrong, but some are useful"
- Let's suppose I ask Matlab to compute $\sqrt{2}\sqrt{2} - 2$. How about $\sin(\pi)$?

```
>> format long
>> sqrt(2)*sqrt(2) - 2
ans =
    4.440892098500626e-16
```

```
>> sin(pi)
ans =
    1.224646799147353e-16
```

- Computers can only store a finite number of digits, so irrational numbers like π and $\sqrt{2}$ will always be just a *little* wrong. Also, most functions (like the $\sin(x)$) are represented by a finite number of terms in their Taylor Series expansions.
- Computer models are still very practical and useful, just keep in mind that there is always error!

Installing Matlab

- In that class we will be focusing on Matlab as our coding language

Here's the Matlab install link again!
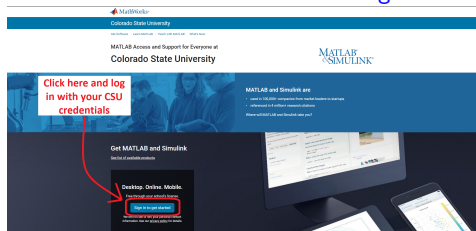
- The install process asks which toolboxes you want to install. This class will not require any of them, but here are some that I've used the most
  - Image Processing Toolbox
  - Navigation Toolbox
  - Parallel Computing Toolbox
  - Signal Processing Toolbox
  - Statistics and Machine Learning Toolbox
- Installation can take a while! It took me about an hour!

MATLAB DEMO

I am going to show you a quick script in Matlab to help show you around the program.
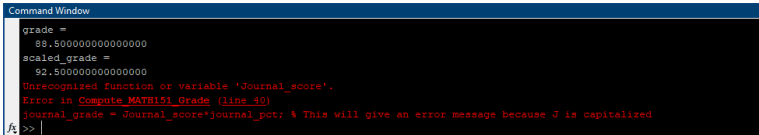
SCRIPTS

- One of the primary ways we will interact with Matlab is through **scripts**
    - Click the "New Script" button on the command window to create a new script
- These are files, {Filename}.m, that contain a collection of statements for Matlab, so you don't need to enter them one by one into the Command Window
- When you want to **execute** a script, there is a run button toward the top of the Editor window.
    - You can also press F5 on your keyboard

ERRORS

- An error occurs when Matlab tries to process a statement, but cannot interpret it

- It will respond by beeping angrily and posting an **error message** to the command window. Telling you where the error occurred, and what's wrong

- In this case it is trying to use Journal_score, but the variable doesn't exist.

- The variable journal_score, does exist though! Variable names are case-sensitive

- A lot of your time coding will be fixing errors. Don't get discouraged by them, they are a learning opportunity!

    - A lot of time they are typos though...