ENGR 105: Introduction to Scientific Computing

Function vs. Scripts, Functions as Black Boxes, Interacting with Functions and Scripts, Discussion on Complexity, Variables, and Arrays

Dr. Graham. E. Wabiszewski

Additional resources for Essential Matlab

- M-files for the 4th ed. of Essential Matlab are located at: http://www.elsevierdirect.com/v2/companion.jsp?ISBN=9780123748836
- May or may not match with examples in 5th ed.
- I recently inquired with publisher to find out if files specific to 5th edition exist

Digital Version of Essential Matlab

- Do not contact anyone about availability of the digital edition - I will do this
- May be available today...or tomorrow!
- You can access it at: http://www.sciencedirect.com/science/book/9780123943989#ancpt005

Upcoming quizzes and HW

Lab quiz #1

- Wednesday 9/11
- Covers lecture slides up to 9/9, Ch. 1 and 2 of Essential Matlab, and "All I need to know...I learned in kindergarten"
- Taken at the beginning of lab - 10 min. time limit
- No Googling, looking up answers, using MATLAB during the quiz
- May NOT be taken before the start of lab
- Primarily multiple choice

Homework comments

HW #1

- Due by 11:59 pm on Wednesday 9/11
- Follow submission instructions

HW #1 / Problem #1

- You should produce an array of the same dimensions as the example provided
- The random numbers in your example...will be random!

Problem 1 (10 pts): Consider the following array of random numbers constructed in MATLAB with a *for* loop

```
for jj = 1:20
    randArr(jj) = rand;
end
```

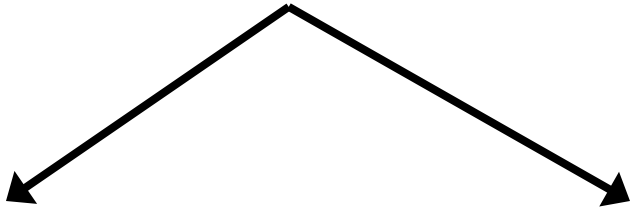
where `rand` assigns a random number.

Produce an array of random numbers, `randArr2`, with the same dimensions as in the example above but using a single line of code and without the use of a for loop. Helpful hint: check the MATLAB help section for `rand`.

Random question for candy

Functions vs. scripts

M-file: any set of instructions stored in a MATLAB file



Function

```
function d = myFunFunction(a,b,c)
% Takes scalar values a, b, and c,
modifies and sums the values

% Modifying the input values
var1 = 2*a;
var2 = b^2;
var3 = sqrt(c);

% Calculating the output
d = var1 + var2 + var3;
```

Script

```
% Define values of scalars a, b, and
c
a = 1;
b = 2;
c = 15;

% Modifying the scalar values a, b,
and c
var1 = 2*a;
var2 = b^2;
var3 = sqrt(c);

% Calculating the sum of the modified
scalar values
d = var1 + var2 + var3;
```

Commonalities between functions and scripts

- Both stored in M-files
- Both modified using the "Editor" window
- Standalone pieces of code
- Can be called from the command window or other functions

Why use scripts?

What is a Matlab "script"

- MATLAB M-file where all variables, constants, etc. are called within the file

Why develop code with a "script"

- You may want to develop multiple lines of code to solve a problem
- Consistently typing that code into the command window can become laborious / doesn't "save" your code
- You can rapidly comment out and adjust syntax to debug your code

Why use functions?

What is a function?

- A MATLAB M-file that starts with `function`
- Generally, you need to pass information into the function and the function returns one to several results

Why use functions?

- The function serves as a black box of code
- Its functionality has been tested / verified and it can be called at any time without questioning the output
- Long, complicated computations can be broken into multiple, smaller pieces of code

How to specify / use a function

Establishing a function requires a specific syntax

- For a single output and input

```
function output = myFunc(input)
```

- For multiple outputs and inputs

```
function [output1,output2,output3] =
myFunc(input1,input2,input3)
```

*Note: this would be on a single line

Naming / saving a function

- Must start with a letter from the alphabet (a through z)
- The name used to save the function needs to be the function name

Interacting with a function

How do I "call" a function

- Initiate function execution from the command line - make sure you include the inputs! (DEMO)
- Functions can be called from other functions (DEMO)

I called a function - where did my variables go?

Unless cleared, variables will persist for "scripts", but not for functions

- Whatever happens in a function, stays in a function
- For instance, a variable, myVar, could be invoked in myFunction1 and resulting in myOut, which is returned to another function, say myFunction2 - this function could also use myVar without retaining "memory" of myVar as invoked in myFunction1
- DEMO - retention of workspace variables for a "script" but not for a function

Inspecting variables in a function

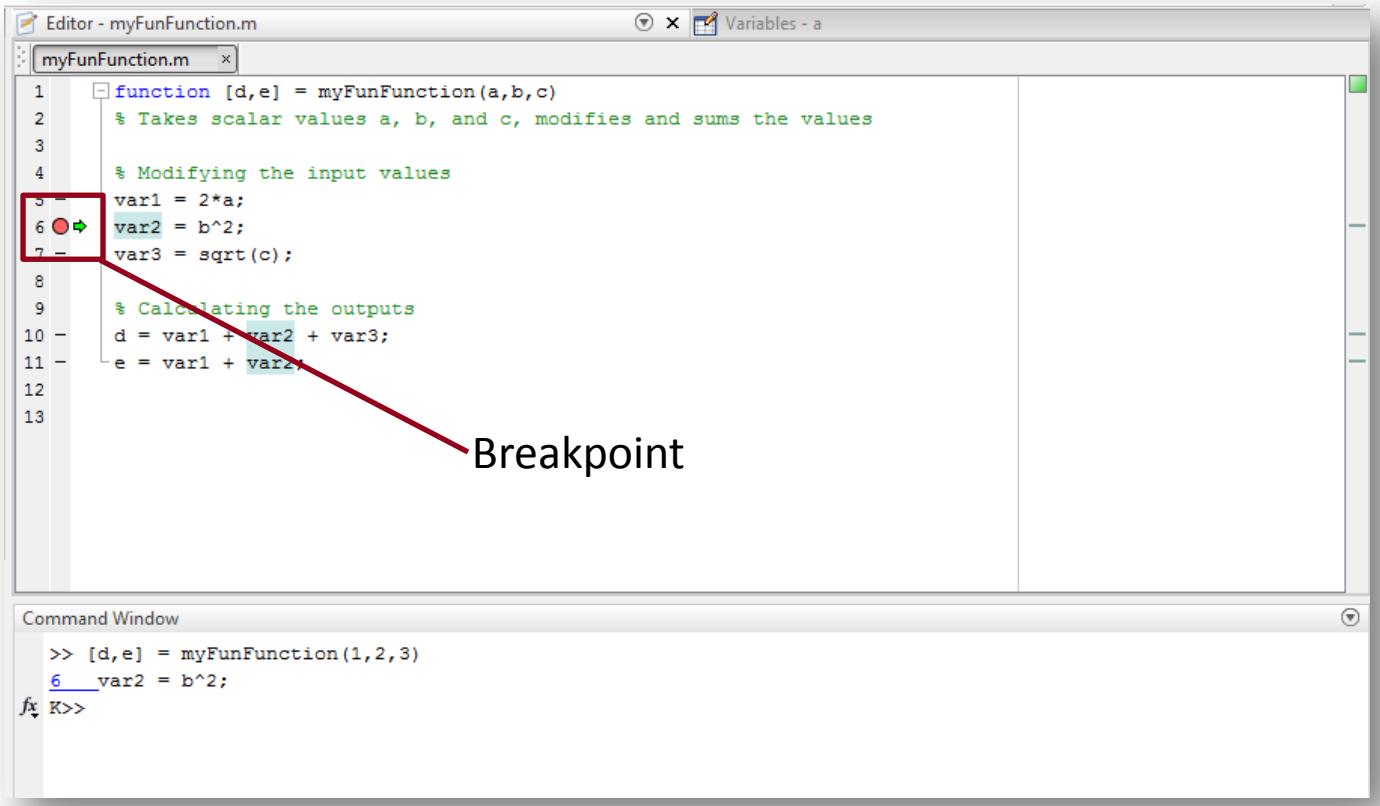
Option #1: Enable command line output (do not suppress output using ";")

- For instance, a function has the following line of code

```
var1 = 2*x+3*y+z^2  % Result will print to
the command line
var1 = 2*x+3*y+z^2; % Result will NOT print
to the command line
```

Inspecting variables in a function

Option #2: Enable a breakpoint



Cleaning up your MATLAB workspace and command line

Variables held in memory can be cleared using the `clear` command or by manually by selecting and deleting variables

- Do NOT invoke the `clear` command within a "function"
- `clear` can be called within a "script" or from the command line

The command window may be cleared using the `clc` command

Commenting functions / scripts

The general rule is at least one comment for every three lines of code

- Proper commenting helps others understand, add to, or debug your code
- Proper commenting helps the instructors grade your assignments

Commenting / uncommenting large swaths of code

- Highlight a selection or place your cursor on a line and press "ctrl+r" to comment a section or line of code, respectively
- "ctrl+t" uncomments the code

Solving real problems - start from simple and build to complex

In class discussion

Interacting with variables and arrays

In class discussion