Matlab: Variables and Functions

- Peter Adany
  - peter@cresis.ku.edu
  - Room 314
  - www.cresis.ku.edu/~peter

CReSIS

Matlab: Variables & Functions

How do you use Matlab?

Accessing Matlab help

Variables & data types

Functions – built in and diy

Some useful functions

Some examples

How do you use Matlab?

- **Starting Matlab**
  - Opens the command window
  - Shows the workspace, directory etc…
  - Open a file like MatlabTutorial.m in the text editor or start a new file
    - The file name can't start with a number.

How do you use Matlab?

- Running Matlab:
  - Write a script in the editor and then run it. [1]
  - The code compiles, then runs line by line.
  - You can also just type things in the command window.

Accessing Matlab Functions

- Matlab Help
  - The main help reference
- From the command window:
  - Type "help *function*"[2]

- Online: the Mathworks page
  - google "matlab help"
    - http://www.mathworks.com/access/helpdesk/help/techdoc/matlab.shtml

Variables

Types

- The most basic type of variable in Matlab is the matrix, or array.
- Arrays[3]
  - Most common format for storing values
  - Strings such as 'hello' can be treated as arrays
  - Variables with a single value are treated as a [1x1] array and so on.

- Structs[4]
  - Used for data organization

- Cells[5]
  - Also used for data organization

Variables

- Scope
  - Local
    - All variables are local by default unless you declare them as global or persistent.
  - Global
    - All functions that also declare the variable as global can see the same value
  - Persistent
    - While running a program with repeated calls to the same function, the function will remember the old value each time it is called.

- Data type:
  - double, string, integer, binary, boolean etc…

Functions

- Many functions are built in.[6]
- The following formats apply:
  - function(input);
  - output = function(input);
  - [output1 output2] = function(in1, in2);

Functions

- Writing a new function in the editor[7]
  - The first line indicates that it is a function instead of a script file.
  - Save as "*functionname.m*" in work dir.

Some Useful Functions[8]

- *linspace(), logspace()*
- *min(), max()*
- *find()*
- *reshape()*
- *circshift()*
- *round(), floor(), ceil()*
- *eval()*

Some Useful Functions

- *linspace(), logspace()*
  - Creates a linear array or log-spaced array
  - Similar to using 0:0.1:1

Some Useful Functions

- *min(), max()*
  - Can be used to get min or max value from a variable
  - Also useful for 'clipping' arrays within specified bounds

Some Useful Functions

- *find(x >= y)*
  - Returns the indices where the test is true.

Some Useful Functions

- *reshape()*
  - *Converts [N x M] array into [N' x M']*
  - *Useful to reorganize values to make computation easier*

Some Useful Functions

- *circshift()*
  - *Rotates data values within array*
  - *Useful for implementing delays, etc*

Some Useful Functions

- *eval()*
  - *Forces matlab to read the argument as a line of code*
  - *Useful when you don't know what a variable name will be, etc...*

Using Functions in Matlab

- Tips:
  - Avoid *for* loops
    - line-by-line operation is slow
    - Use higher level functions instead
    - If you do use a loop, pre-allocate all the memory
      - Ie, DataOut = zeros(1,N);
  - Make it simple
    - There is a function out there for nearly everything

Example: FM Chirp Signal

- Generate a frequency sweep signal:
  - Time duration of 0.5 s
  - Sampling frequency of 44.1 kHz
  - Sweep from 25 to 500 Hz
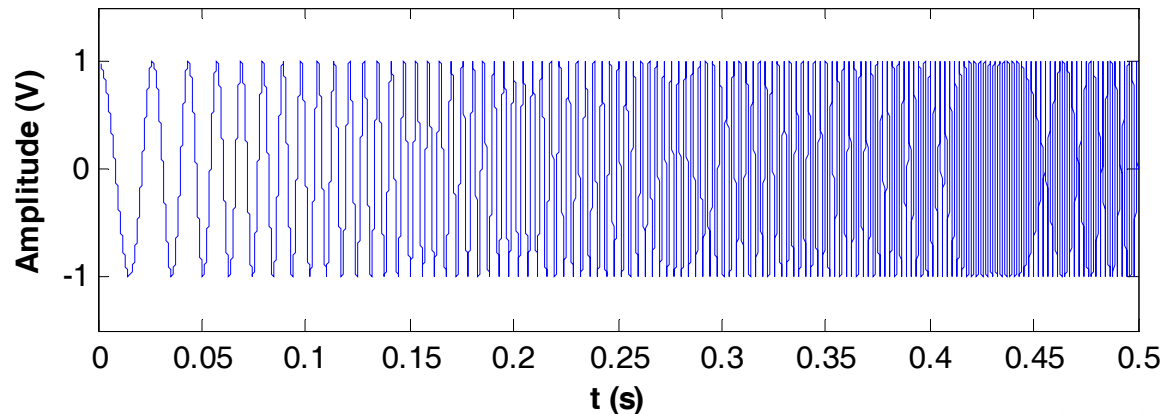
Questions?

Thank You

- # Peter Adany
  - peter@cresis.ku.edu