

Introduction to Matlab Graphics:

General Programming

M-File Functions



S. Awad, Ph.D.
E.C.E. Department
University of Michigan-Dearborn



General Matlab Programming: M-File Functions



M-File Functions

- M-File Editor
- M-File Scripts
- General M-Functions
- Variable Input Argument Functions
- Variable Workspace
- Global Variables
- Persistent Variables
- Subfunctions

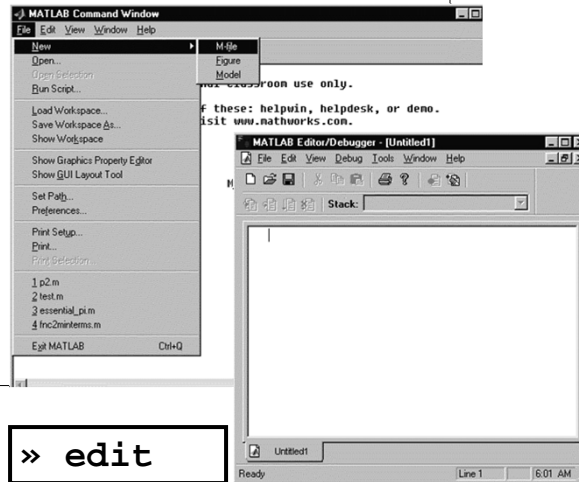
2



General Matlab Programming: M-File Functions

M-File Editor

- The M-File text editor can be opened from the **New** or **Open** options in the **File** menu
- Can also be opened from the **command line** by typing **edit**



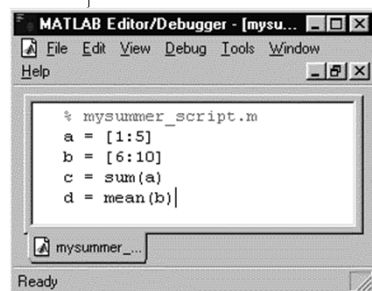
3



General Matlab Programming: M-File Functions

M-File Scripts

- Standard ASCII Text file consisting of commands that are executed in the base Matlab workspace
- The same as executing a group of commands from the command prompt

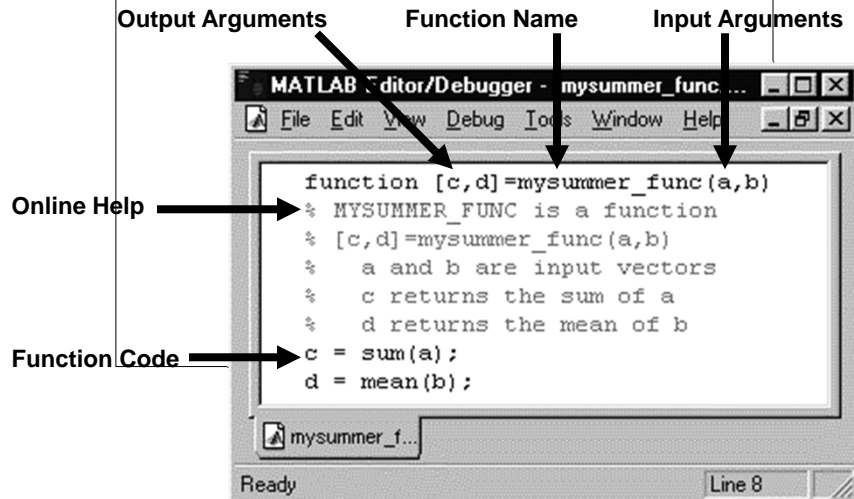


```
» mysummer_script
a =
     1     2     3     4     5
b =
     6     7     8     9    10
c =
    15
d =
     8
```

4

General Matlab Programming: M-File Functions

M-File Functions



5

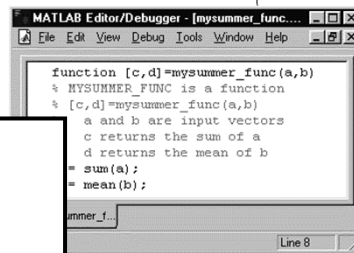
General Matlab Programming: M-File Functions

Running M-Functions

- Executed like any Matlab function

```
» help mysummer_func
MYSUMMER_FUNC is a function
[c,d]=mysummer_func(a,b)
a and b are input vectors
c returns the sum of a
d returns the mean of b

» [c d]=mysummer_func([1:5],[6:10])
c =
    15
d =
     8
```



6



General Matlab Programming: M-File Functions

Variable Input Arguments

- **varargin** is a **variable length input argument list** that allows any number of input arguments to a function
- **varargin** is a **cell array**
- **varargin** must be declared as the **last input argument** and collects all the inputs from that point onwards.
- **nargin** generally holds the number of function input arguments
- Inside the body of a user-defined function, **nargin** returns the **number of input arguments** that were **used to call the function**

7



General Matlab Programming: M-File Functions

Variable Input Example

- Exclude parenthesis for no inputs

```
» myvarin_func
Num of Inputs = 0
» myvarin_func(1)
Num of Inputs = 1
» myvarin_func(1,'a')
Num of Inputs = 2
» myvarin_func(1,'a','More')
Num of Inputs = 3
```

```
MATLAB Editor/Debugger - [myvarin_func.m - D:...
File Edit View Debug Tools Window Help
function myvarin_func( varargin )
str = num2str( nargin );
disp(['Num of Inputs = ' str]);
Line 3
```

8



General Matlab Programming: M-File Functions

Varargin Cell

- Note that the **varargin** is actually a **cell** containing **any possible data type**
- It is up to the developer to handle different data types and extract the information from the cell
- Control flow statements (ex. switch, if) are used to decipher the arguments

9



General Matlab Programming: M-File Functions

Variable Workspaces

- All variables are stored in a workspace
 - ◆ **Base** Command line & script files.
 - ◆ **Function** Defined and maintained within function (includes Persistent variables)
 - ◆ **Global** All functions can access.

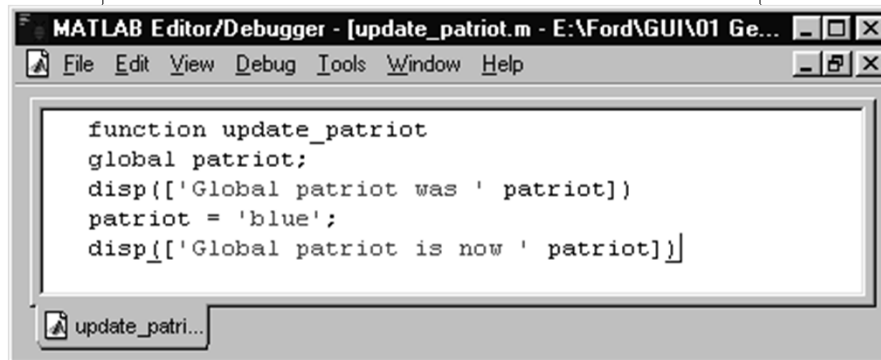
10



General Matlab Programming: M-File Functions

M-Function Global Variable

- Create a function to update a global variable **patriot** and set is equal to the string 'blue'



```
MATLAB Editor/Debugger - [update_patriot.m - E:\Ford\GUI\01 Ge...
File Edit View Debug Tools Window Help

function update_patriot
global patriot;
disp(['Global patriot was ' patriot])
patriot = 'blue';
disp(['Global patriot is now ' patriot])

update_patri...
```

11



General Matlab Programming: M-File Functions

Global Variable Example

```
» global patriot
» patriot = 'red'
patriot =
red

» update_patriot
Global patriot was red
Global patriot is now blue

» patriot
patriot =
blue
```

12



General Matlab Programming: M-File Functions

M-Function Persistent Variable

- A **persistent** variable is like a global variable that is only accessible to the function that created it

```
function update_persistent
persistent someval
if isempty(someval)
    someval='initialized'
else
    oldval=someval;
    someval=[oldval ' and more']
end
```

13



General Matlab Programming: M-File Functions

Update Persistent

```
» update_persistent
someval =
initialized

» update_persistent
someval =
initialized and more

» update_persistent
someval =
initialized and more and more
```

14



General Matlab Programming: M-File Functions

Clear and Mlock

- Use `clear` function to remove persistent or global variables from Matlab workspace
- `mlock` can be used to lock the currently running M-File in memory, thus making its persistent variables unclearable
- `munlock` returns them to a clearable state

15



General Matlab Programming: M-File Functions

Subfunctions

- More than one function can be defined within an M-File
- The **M-file** must have the **name of the first function**
- All other functions are considered **subfunctions**
- Subfunctions can **only be accessed by other functions in the same file**
- Each subfunction has its own **workspace**

16



General Matlab Programming: M-File Functions

Subfunction Example

- Create a subfunction that can only be called by `sub_example`

```
function sub_example(num)
    for n=1:num
        run_subfunction(n)
    end

    function run_subfunction(n)
        disp(['run subfunction entered '...
            num2str(n) ' time(s)']);
```

17



General Matlab Programming: M-File Functions

Execute sub_example

```
» sub_example(3)
run_subfunction entered 1 time(s)
run_subfunction entered 2 time(s)
run_subfunction entered 3 time(s)

» run_subfunction(1)
??? Undefined function or variable 'run_subfunction'.
```

- `sub_example` is accessible from the Matlab workspace but `run_subfunction` is not

18