

Advanced Programming Techniques in MATLAB

Eric Johnson
Application Engineer

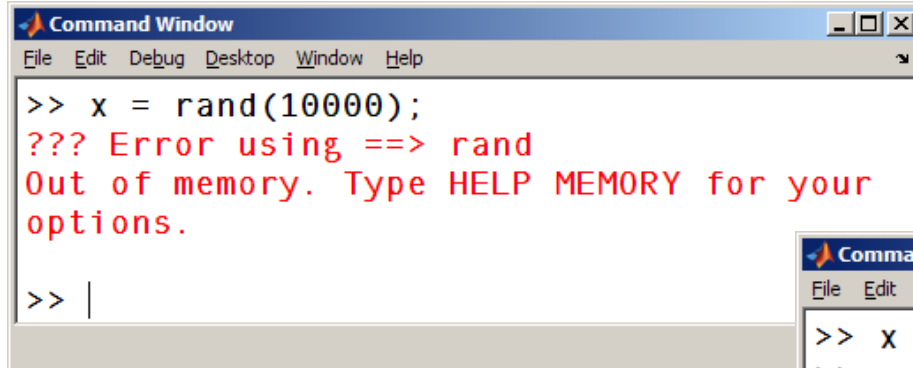
Agenda – Advanced MATLAB

- ***Memory Usage in MATLAB***
 - When is data copied?
 - How does MATLAB store data in memory?
- ***Functions of All Types***
 - Introduction/review of MATLAB function types
 - Nested functions, anonymous functions, function handles, etc.
 - Applications of nested functions
 - Solving optimization problems
 - Building a graphical user interface for volume visualization
 - Building 2-figure GUIs (time-permitting)

Agenda – Advanced MATLAB

- ***Memory Usage in MATLAB***
 - When is data copied?
 - How does MATLAB store data in memory?

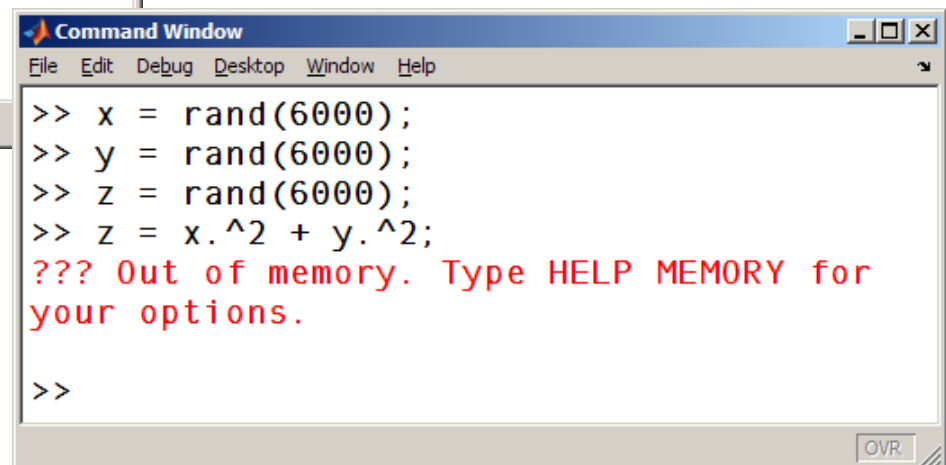
Have you ever had an “Out of Memory” error?



```

>> x = rand(10000);
??? Error using ==> rand
Out of memory. Type HELP MEMORY for your
options.

>> |
    
```



```

>> x = rand(6000);
>> y = rand(6000);
>> z = rand(6000);
>> z = x.^2 + y.^2;
??? Out of memory. Type HELP MEMORY for
your options.

>>
    
```

Memory is Often OS-Bound

- **32-bit** systems have **4GB** of addressable process memory
 - Part of it is reserved by the OS, leaving the application < 4GB
 - Windows XP (default): 2GB
 - Windows XP with /3gb switch: 2GB + 1GB
 - Linux/UNIX/Mac: ~3GB
- **64-bit** systems allow **16TB** of addressable memory
- Data needs to be defined in **contiguous memory**
 - Fragmented memory may cause “out of memory” errors.

When is data copied?

Function calls

```
function y = foo(x,a,b)
```

```
    a(1) = a(1) + 12;
```

```
    y = a * x + b;
```

When does MATLAB copy memory upon calling a function?

```
y = foo(1:3,2,4)
```

- i.e., `x = 1:10000`, `a = 2`, `b = 4`



When is data copied?

In-Place optimizations

When does MATLAB perform calculations “**in-place**”?

y = 2*x + 3;

x = 2*x + 3;

NOT in-place

in-place

- **Output variable name same as input variable name**
- **Element-wise computation**

In-place Optimizations

What happens during “**in-place**” operations?

```
x = rand(5000);
```

```
y = rand(5000);
```

```
%% NOT In-Place
```

```
y = sin(sqrt(2*x.^5+3*x+4));
```

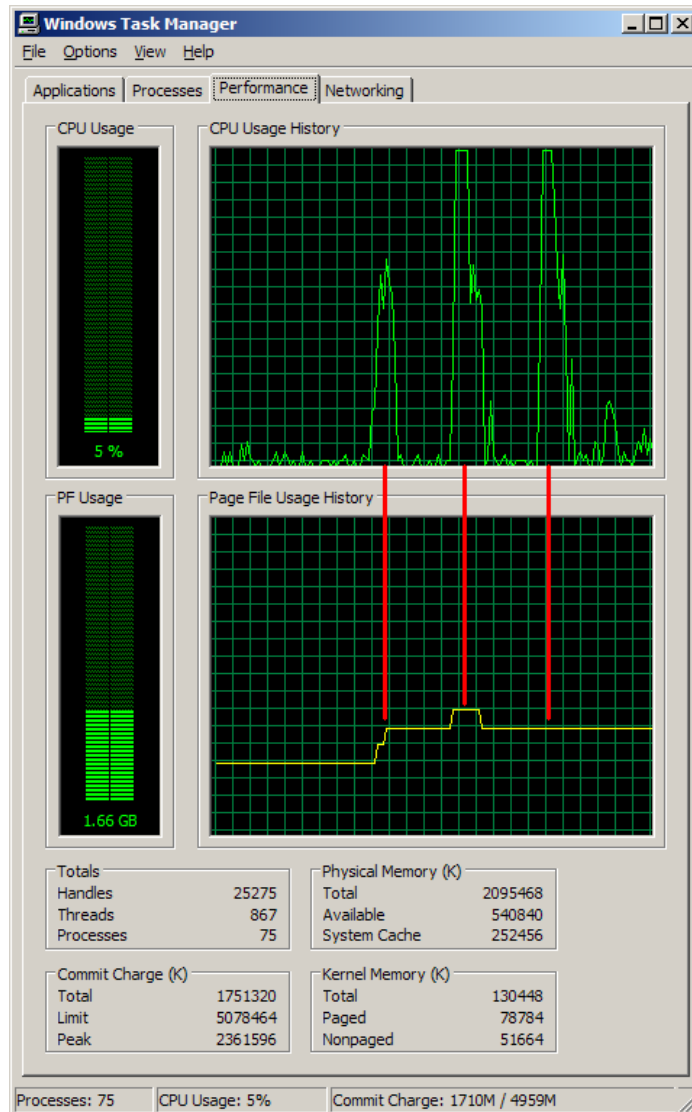


```
%% In-Place
```

```
x = sin(sqrt(2*x.^5+3*x+4));
```



testInPlaceSimple.m



```
%% Create Variables (Pre-Allocate)
```

```
x = rand(5000);
```

```
y = rand(5000);
```

```
%% NOT In-Place
```

```
y = sin(sqrt(2*x.^2+3*x+4));
```

```
%% In-Place
```

```
x = sin(sqrt(2*x.^2+3*x+4));
```


How does MATLAB store data?

Numeric data types

Floating-point numbers

<code>double</code>	8 bytes	(<i>default</i>)
<code>single</code>	4 bytes	

Integers

<code>int8, uint8</code>	1 byte
<code>int16, uint16</code>	2 bytes
<code>int32, uint32</code>	4 bytes
<code>int64, uint64</code>	8 bytes
<code>logical</code>	1 byte
<code>char</code>	2 bytes

How does MATLAB store data?

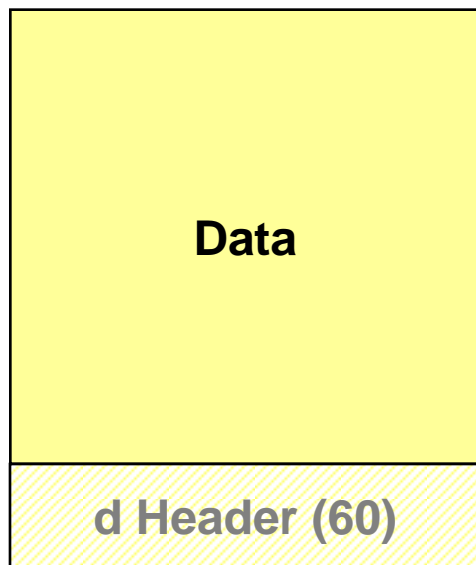
Container overhead

```
d = [1 2]           % Double array
dcell = {d}         % Cell array containing "d"
dstruct.d = d       % Structure containing "d"
```

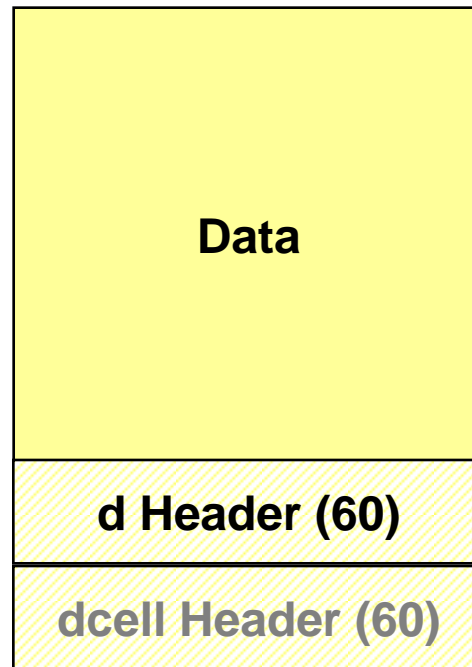
```
whos
```

How does MATLAB store data?

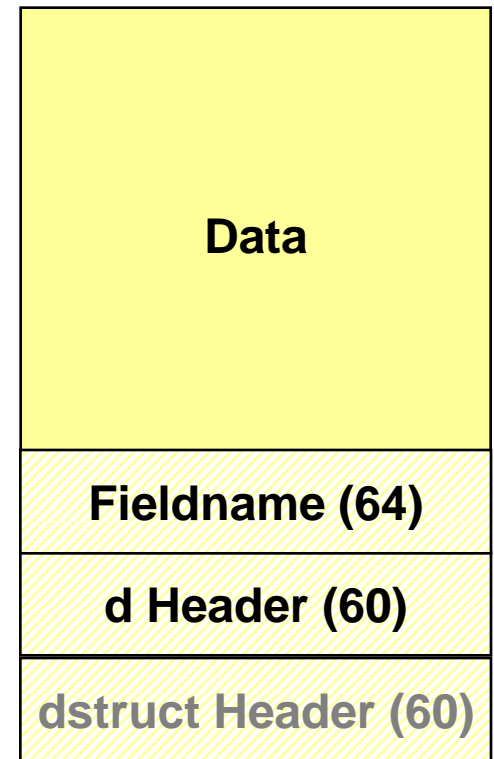
Container overhead



`d = [1 2]`



`dcell = {d}`



`dstruct.d = d`

More on Memory Overhead

Structure of Arrays vs. Array of Structures

Which costs more memory?

Structures of arrays

OR

Array of structures?

For example, let's look at an RGB image...



Structure of Arrays vs. Array of Structures

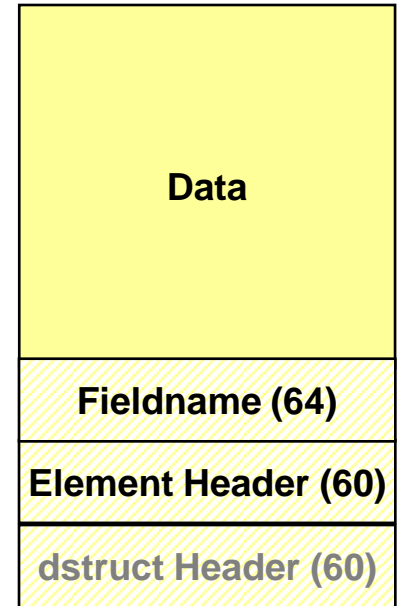
im1

			im1.blue	30	35	30	31	36
				25	22	21	28	41
		im1.green	94		24	22	26	31
			91		28	28	25	33
im1.red	82		90		30	33	30	31
	80		89	90	91	93	89	33
	78		86	88	89	88	88	
	74	74	80	85	83			
	75	75	77	80	81			

im2

im2(1,1).pixel	im2(1,2).pixel	im2(1,3).pixel
82 94 30	82 97 35	79 95 30
im2(2,1).pixel	im2(2,2).pixel	im2(2,3).pixel
80 91 25	81 94 22	81 95 21

⋮



...

structmem2.m

How does MATLAB store data?

Structures

```
s.A = rand(4000,3200);
```

```
s.B = rand(4000,3200);
```

```
sNew = s;
```

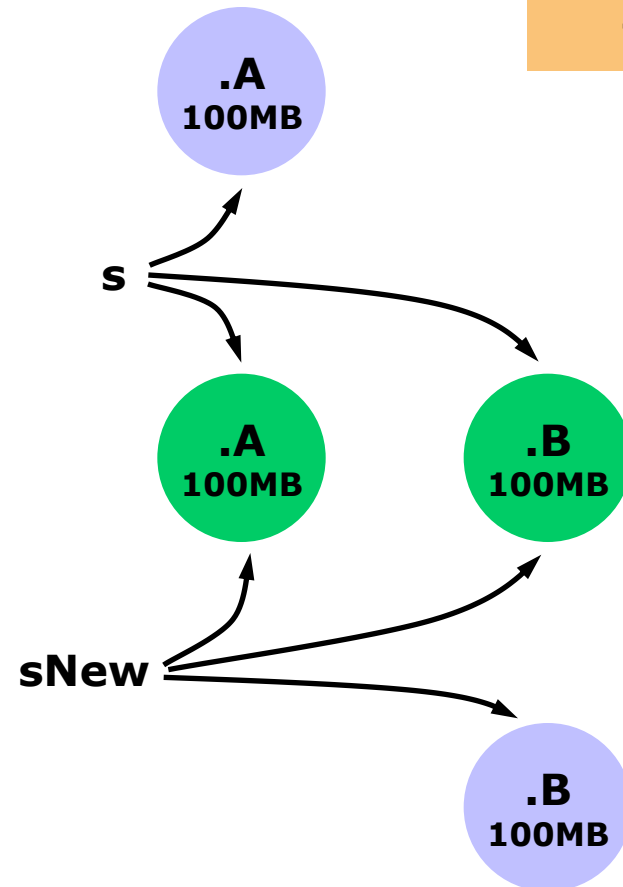
```
s.A(1,1) = 17;
```

```
sNew.B(1,1) = 0;
```

```
sNew = s;
```

Memory Used

300 MB



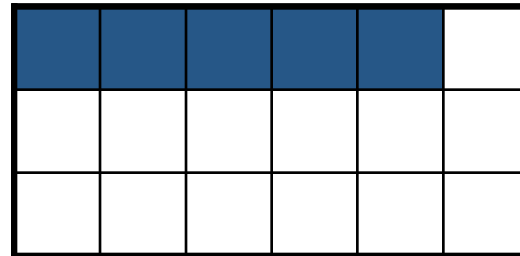
structmem1.m

Summary of Memory Usage in MATLAB

- When is data copied?
 - “Lazy” copy: only when necessary (copy on write)
 - Never, if operation can be performed “in-place”
 - “in-place” is faster because memory is not copied
- How does MATLAB store data?
 - Every array has overhead
 - Structures and cell arrays are containers which can hold multiple arrays
- ***Pre-allocation of memory***
- ***Vectorization***

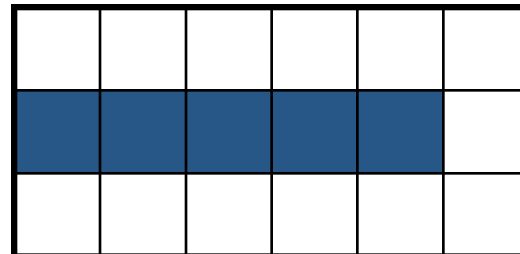
Pre-Allocation of Memory

```
for id = 1:10000
    a(id) = myFcn(id);
end
```



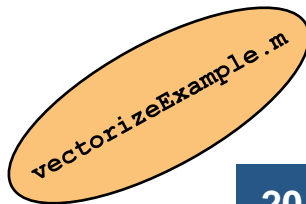
vs.

```
a = nan(1, 10000);
for id = 1:10000
    a(id) = myFcn(id);
end
```



Vectorization

- MATLAB is a matrix (vector)-based language
 - *supports vectorized operations (**faster** in general)*
- Interpreter manages memory
 - *makes MATLAB safer, but **requires more memory***
- Can **de-vectorize** to **reduce memory** requirements in sacrifice of speed
- Process in **column blocks**, rather than row blocks



MATLAB and Memory: Additional Resources

- Recorded Webinar:
 - “Handling Large Data Sets Efficiently in MATLAB”
 - “Speeding Up MATLAB Applications”
 - www.mathworks.com → Recorded Webinars
- Memory Management Guide (***Tech Note 1106***)
 - `>> doc memory`
 - Search www.mathworks.com for “1106”
- Avoiding Out of Memory Errors (***Tech Note 1107***)
 - Search www.mathworks.com for “1107”

Agenda – Advanced MATLAB

- ***Functions of All Types***
 - Introduction/review of MATLAB function types
 - Nested functions
 - Anonymous functions
 - Function handles
 - etc.
- Applications
 - Solving optimization problems
 - Building a graphical user interface for volume visualization
 - Building 2-figure GUIs (time permitting)

Function Quiz

Function Quiz

- Let's see how much *you* know about functions in MATLAB
- This quiz covers
 - Anonymous functions
 - Nested functions
 - Function handles
 - Regular (“simple”) functions
 - Subfunctions
- This quiz does not cover
 - MEX functions, private functions, social functions...

Quiz

Nested Functions

myGUI

myCallback

This creates
two
workspaces

Nested functions
are defined
inside a parent
function

```
function myGUI
...
% Pushbutton
hb = uicontrol('Style', 'pushbutton', ...
               'Callback', @myCallback );

% Checkbox
hc = uicontrol('Style', 'checkbox');

% Nested Callback
function myCallback(h, e)
    I = imread('starry_night.jpg');
    checked = get(hc, 'Value') > 0;
    ...
end
end
```

Importantly, a nested function can
access and modify the parent workspace

Function Quiz Review

- Regular (“simple”) functions
 - Function handles
 - Anonymous functions
 - Subfunctions
 - Nested functions
-
- Q: Which of these function types can be incorporated into a single file?

Nested Function Applications

- Building a graphical user interface for volume visualization
- Solving optimization problems
- Building 2-figure GUIs (Optional)

Application 1: Solving Optimization Problems

- We get many posts on `comp.soft-sys.matlab` about optimization. The problems fall into several categories. The one we will address today is:
 - How to include extra parameters to define the objective function

Optimization Example (unconstrained)

Objective function:

$$ax_1^2 + bx_1x_2 + cx_2^2 + d|x_2 - x_1|$$

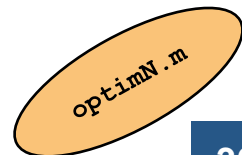
a, b, c – “Regular” parameters

d – Additional parameter that might alter the type of problem by making the objective function non-smooth

Optimization Example Using Nested Functions

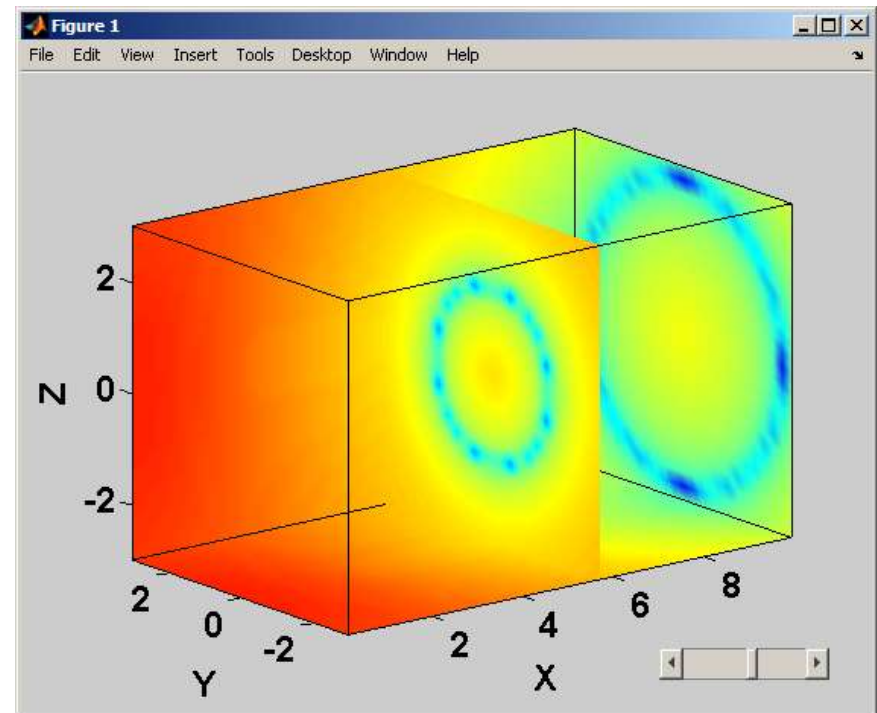
Use nested functions in place of subfunctions

- Start with code in `optimS.m` (end up with code in `optimN.m`)
 - Simplify the interfaces, with nesting
 - Objective function
 - `function f = objfun(x)`
 - Output function
 - `function stop = ... myoutputfcn(x,optimvalues,state)`



Application 2: Building a Graphical User Interface for Volume Visualization

- Application: Building a custom tool for volume visualization
- This example illustrates:
 - Using function handles to export nested functions
 - Using nested functions for object callbacks



Application 3 (Optional): Creating Two-Figure GUIs

This is a frequent comp.soft-sys.matlab request (several times each month).

Challenges:

- Managing handles
- Sharing information between figures

Nested functions can simplify and improve performance.

Summary and Advice

Anonymous Functions

- Create simple functions without creating m-files:

```
fh = @(x,y) a*sin(x)*cos(y)
```

- Useful for “fun-funs” (function functions)
 - Optimization
 - Solving differential equations
 - Numerical integration
 - Plotting
 - Array functions (cellfun, structfun, ...)
- Convenient for simple callbacks
- Embed data in function calls, to help reduce “copy-paste” coding

```
getXLSDData = @(worksheet) xlsread('records.xls', worksheet);
```

Summary and Advice

Nested Functions

Embed one function within another, with shared, persistent workspace:

```
function main
    function nest
    end
end
```

- Useful for “fun-funs” (function functions)
- Very convenient for callbacks (shared workspace)
- Encapsulate functionality and data (export as function handles)

MATLAB Function Types: Additional Resources

- **MATLAB Digest - September 2005**
 - **Dynamic Function Creation with Anonymous and Nested Functions**
<http://www.mathworks.com/company/newsletters/digest/2005/sept/dynfunctions.html>
- **Examples**
 - **Anonymous functions**
<http://www.mathworks.com/products/matlab/demos.html?file=/products/demos/shipping/matlab/anondemo.html>
 - **Nested Functions**
<http://www.mathworks.com/products/matlab/demos.html?file=/products/demos/shipping/matlab/nesteddemo.html>
- **The Art of MATLAB Blog**
 - **Look at Category: Function Handles**
<http://blogs.mathworks.com/loren/>

Summary

- Memory Usage in MATLAB
 - When is data copied?
 - How does MATLAB store data in memory?
- Functions of All Types
 - Introduction/Review of MATLAB function types
 - Applications of nested functions
 - Solving optimization problems
 - Building a graphical user interface for volume visualization
 - Building 2-figure GUIs (optional)