

MATLAB

Introduction

CS101 Lecture #21

Administrivia

Administrivia

- ❖ Homework #8 is out;
- ❖ Due on next Monday, Dec. 11th.

Warmup Quiz

Question #1

```
import numpy as np
import math
tmax = 10
dt = 0.01
nt = int( tmax/dt ) + 1
x = np.zeros( (nt,) )
for i in range( 0, tmax, dt ):
    vx = x[ i-1 ] / np.sin( i * math.pi)
    x[ i+1 ] = x[ i ] + vx * dt
```

Which uncaught error will halt this code?

- A ZeroDivisionError
- B TypeError
- C SyntaxError
- D IndexError

Question #1

```
import numpy as np
import math
tmax = 10
dt = 0.01
nt = int( tmax/dt ) + 1
x = np.zeros( (nt,) )
for i in range( 0, tmax, dt ):
    vx = x[ i-1 ] / np.sin( i * math.pi )
    x[ i+1 ] = x[ i ] + vx * dt
```

Which uncaught error will halt this code?

- A ZeroDivisionError
- B TypeError ★(dt has to be integer)
- C SyntaxError
- D IndexError

Question #2

```
x = np.ones( 10 )
for i in range( 10 ):
    try:
        ???
    except Exception:
        print( 'Error on step %d.' % i )
        continue
```

Which of the following candidates for ??? would *not* produce an error message?

- A `x += x[i+1]`
- B `x[i] /= 0`
- C `x[-i-1] = sum(x[:i])`
- D `x[10-i] = sum(x[:i])`

Question #2

```
x = np.ones( 10 )
for i in range( 10 ):
    try:
        ???
    except Exception:
        print( 'Error on step %d.' % i )
        continue
```

Which of the following candidates for ??? would *not* produce any error message?

- A `x += x[i+1]` index error
- B `x[i] /= 0` ★ (np.dtype('float') handles division by zero without throwing an error)
- C `x[-i-1] = sum(x[:i])` ★
- D `x[10-i] = sum(x[:i])` index error

MATLAB

What is MATLAB?

- ❖ Programming language + environment (an IDE).
- ❖ Proprietary, owned and maintained by MathWorks.
- ❖ Dates from late 1970s, under active development.
- ❖ Was an influence on NumPy/matplotlib, so will be familiar.

Why MATLAB?

- Designed for engineering and scientific computing
- Famous for its *matrix-centric* computation
- Ideal for:
 - Linear algebra
 - Image processing
 - Numerical analysis
 - Simulation, Graphical plot, etc.
- Many toolboxes available.
- Excellent documentation: MATLAB Central.

Basics

- ✦ Literals, variables, operators

5

- ✦ Expressions

```
a = 3 * 2
```

```
b = 1 + a
```

- ✦ Semicolon suppresses output (mutes): ;

```
b = b ^ 2;
```

- ✦ ans is default result.

```
a / 4
```

- ✦ disp(ans) displays the value of ans.
- ✦ whos a shows type, size (as an array), and number of Bytes of variable a in workspace.

Numeric types

- MATLAB implements:
 - integers
 - floating-point numbers
 - complex numbers
 - logical (boolean, 0/1)
- in 8-, 16-, 32-, and 64-bit versions.

Array types

- Arrays are the fundamental type in MATLAB:

```
a = [ 1 2 3 ];
```

- Arrays are indexed using parentheses:

```
b = a( 1 );
```

```
c = a( end );
```

- **MATLAB counts from one, not zero!**

Multidimensional arrays

- More dimensional arrays use semicolons to separate rows:

```
A = [ 1 2 3 ; 4 5 6 ];
```

- Arrays are indexed using parentheses and commas:

```
a = A( 1,2 );
```

- Helper functions are available:

```
B = zeros( 3,3 ) + ones( 3,3 ) + eye( 3,3 ) ;
```

Question

$$\begin{pmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \end{pmatrix}$$

Which of the following will produce this array?

A `[1 1 1] ; [2 2 2]`

B `[1 1 1 ; 2 2 2]`

C `[1 2] ; [1 2] ; [1 2]`

D `[1 2 ; 1 2 ; 1 2]`

E `[[1 1 1] , [2 2 2]]`

Question

$$\begin{pmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \end{pmatrix}$$

Which of the following will produce this array?

A `[1 1 1] ; [2 2 2]`

B `[1 1 1 ; 2 2 2]` ★

C `[1 2] ; [1 2] ; [1 2]`

D `[1 2 ; 1 2 ; 1 2]`

E `[[1 1 1] , [2 2 2]]`

Question

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$$

Which of the following will access 4 in this array?

A `A(1,0)`

B `A[2,1]`

C `A(2,1)`

D `A(1)(0)`

Question

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$$

Which of the following will access 4 in this array?

A `A(1,0)`

B `A[2,1]`

C `A(2,1)` ★

D `A(1)(0)`

Array operations

```
% basic mathematics:  
A = ( ones( 3,3 ) + 1 ) / 2  
B = sin( ones( 3,3 ) * pi )  
C = B' % transpose with '  
  
% matrix multiplication:  
D = eye( 3,4 ) * ones( 4,5 )
```

Question

$$\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$$

Which of the following will produce this array?

A `3*ones(2,2) - 2*eye(2,2)`

B `2*ones(2,2) + eye(2,2)`

C `3*ones(2,2) - eye(2,2)`

D `ones(2,2) + eye(2,2)`

Question

$$\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$$

Which of the following will produce this array?

A `3*ones(2,2) - 2*eye(2,2)`

B `2*ones(2,2) + eye(2,2)`

C `3*ones(2,2) - eye(2,2)`

D `ones(2,2) + eye(2,2)` ★

Array operations

```
% concatenating arrays  
A = [ eye( 3,4 ), eye( 3,5 );  
      ones( 2,4 ), ones( 2, 5) ]
```

Array operations

```
% concatenating arrays  
A = [ eye( 3,4 ), eye( 3,5 );  
      ones( 2,4 ), ones( 2, 5) ]
```

- ❖ horizontal concatenation by ',' or ' '
- ❖ vertical concatenation by ';'
- ❖ array size must agree!

Question

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}$$

How can we produce this array?

- A `[[1 3 5] [2 4 6]]`
- B `[[1 2] [3 4] [5 6]]`
- C `[[1 3 5] ; [2 4 6]]`
- D `[[1 2] ; [3 4] ; [5 6]]`

Question

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}$$

How can we produce this array?

A `[[1 3 5] [2 4 6]]`

B `[[1 2] [3 4] [5 6]]`

C `[[1 3 5] ; [2 4 6]]`

D `[[1 2] ; [3 4] ; [5 6]]` ★

Matrix v. element operations

```
A = 2 * ones( 2,2 )  
B = A * eye( 2,2 ) %matrix multiplication  
C = A .* eye( 2,2 ) %elementwise multiply
```

- ❖ Denoted by the dot . in front of the operator.
- ❖ Matrix dimensions must agree.

Scripts and Function

- ❖ MATLAB uses .m files for two purposes: scripts and functions.
- ❖ Use the built-in editor to create these.
- ❖ Use 'edit' command to open or edit a (.m) file from cmd window

Run a script

- ❖ Scripts contain a sequence of commands to run
- ❖ From command window: `filename` (no `'.m'`)
- ❖ Be aware of your Present Working Directory (`pwd`).
- ❖ Use `addpath` and `genpath` to add folder/sub-folders to your search paths.

Functions

- ❖ A function must be defined in a file of the same name.

```
function [outputs] = function_name(inputs)
    % ...
    % ...
end
```

- ❖ Multiple input/output variables separated by ' , '
- ❖ No explicit return statements
- ❖ Takes the value of output variable(s) that is produced in the function

$$T_C = \frac{100}{180}(T_F - 32)$$

File CelsiusFromFahrenheit.m:

```
function Tc = CelsiusFromFahrenheit(Tf)
    Tc = (Tf - 32) * ( 100/180 );
end
```


Commenting in scripts

- Comments are indicated as follows:

```
% this is a comment  
%{  
    this  
    is  
    a  
    block comment  
%}
```

Strings

- ❖ Also supports strings, not as convenient as numbers (python is a better choice for handling strings)

Strings

- ❖ Also supports strings, not as convenient as numbers (python is a better choice for handling strings)
- ❖ Manages string as an array of the 'char' type

Strings

- ❖ Also supports strings, not as convenient as numbers (python is a better choice for handling strings)
- ❖ Manages string as an array of the 'char' type
- ❖ Enclosed with single quotes (only!).

```
s = 'a brown fox';  
class(s)  
size(s)
```

String formatting and print

- Print formatted strings with `sprintf`, `fprintf`

```
s = sprintf( '%d %f\n', 5, sin(pi/4) );
```

%output to screen

```
fprintf('%d %f\n', 5, sin(pi/4));
```

%output to file

```
fp = fopen('newfile.txt', 'w')
```

```
fprintf(fp, '%d %f\n', 5, sin(pi/3));
```

```
fclose(fp)
```

Look for more information

- ❖ `help fprintf`
 - displays help information to the screen
- ❖ `doc fprintf`
 - opens matlab's documentation book

Look for more information

- ❖ `help fprintf`
 - displays help information to the screen
- ❖ `doc fprintf`
 - opens matlab's documentation book

Tutorial: <http://zichengl.net/cmpt469/matlabtutorial.m>