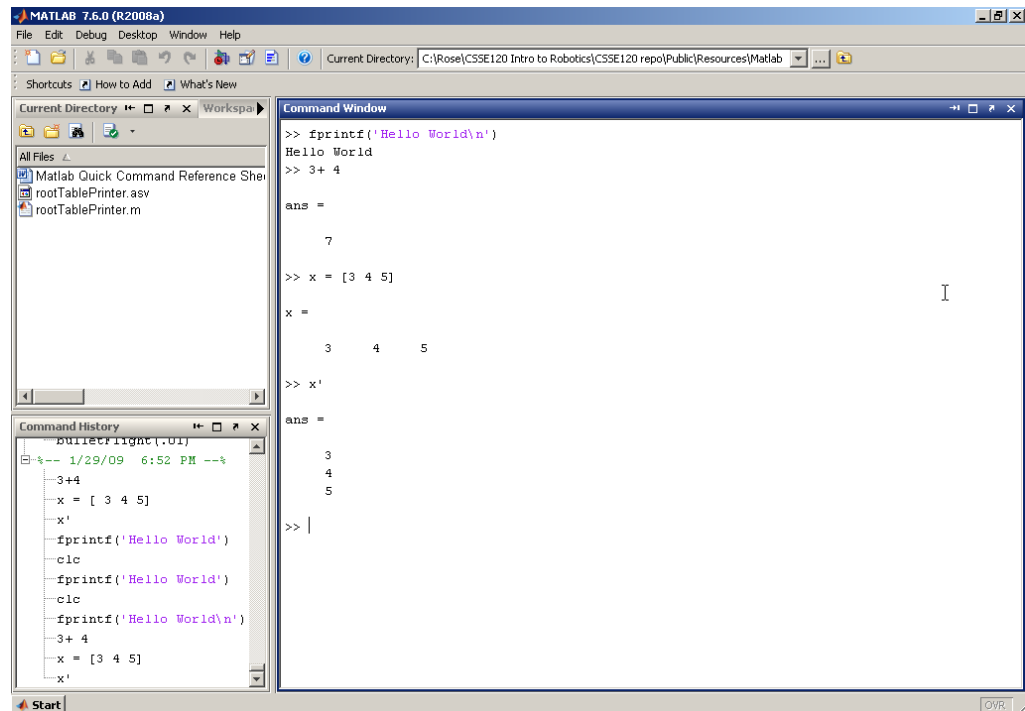


MATLAB INTRO: PREPARATION FOR ME323

CSSE 120—Rose Hulman Institute of Technology

What is MATLAB?

- Numerical Computing Environment &
- Programming Language
- Made by The MathWorks Inc.



The screenshot displays the MATLAB 7.5.0 (R2008a) environment. The Command Window on the right shows the following execution:

```
>> fprintf('Hello World\n')
Hello World
>> 3+4
ans =
    7
>> x = [3 4 5]
x =
    3    4    5
>> x'
ans =
    3
    4
    5
>> |
```

The Command History window at the bottom left shows a list of previous commands and their outputs, including the same sequence of operations shown in the Command Window.

How is MATLAB similar to Python?



- Interactive mode for quick tests
- Programming mode for writing code
 - ▣ Similar to Python's IDLE environment
 - Python has .py files for code
 - MATLAB uses .m files for code
- Similar programming concepts to Python...
 - ▣ Variables, functions, if, for, while, etc.

How is MATLAB different from Python?

- MATLAB = "**matrix laboratory**"
 - MATLAB defaults to use a 2D matrix of numbers (of type double) for as many things as possible
 - Many built in functions without loading libraries
 - Arrays are 1 based not 0 based
 - MATLAB actually has good help docs 😊
 - MATLAB is pricey! Ballpark \$5000 the day you stop going to Rose to have a personal copy of MATLAB.
 - Used heavily in industry. Very common.

Sample comparison code



- The first program we looked at in C was a print root table function. Let's see the syntax in Matlab.
 - ▣ Review see code in C and Python first
 - ▣ See how MATLAB would code the root table problem

```
from math import *

def printRootTable(n):
    for i in range(1,n):
        print "%2d %7.3f" % (i, sqrt(i))

def main():
    printRootTable(10)

main()
```

Parallel examples in Python and C.

```
#include <stdio.h>
#include <math.h>

void printRootTable(int n) {
    int i;
    for (i=1; i<=n; i++) {
        printf(" %2d %7.3f\n", i, sqrt(i));
    }
}

int main() {
    printRootTable(10);
    return 0;
}
```

rootTable in MATLAB

```
% David Fisher  
% Jan 28, 2009  
%  
% Prints a root table of values
```

```
- function rootTablePrinter  
    % Clear the screen  
    clc  
  
    %Call the rootTable function  
    rootTable(10)
```

```
- function rootTable(n)  
- for i = 1:n  
    fprintf('The sqrt of %3d is %7.3f\n',i,sqrt(i))  
- end
```

Comments



- % will comment a line in MATLAB
- Similar to # in Python
 - ▣ Similar to // in C

- First comment in a file is used as the help message.

if statement



if-Statement Structure:

```
if (a<0)
```

```
    x = 1
```

```
end
```

Must have an “end” statement, no :, tabbing is for looks only, () around condition

elif statement



```
if (a<0)
```

```
    x=1
```

```
elseif(a>0)
```

```
    x=2
```

```
else
```

```
    x=3
```

```
end
```

elif is done as elseif (one word)

while loops



```
while (k>0)
```

```
    k=k-1
```

```
end
```

Similar changes to the if statement, can still use the “break” statement to exit early if needed

for statement



```
for k=1 : 0.001 : 10
    x=x+5
end
```

Compare to range in Python:

```
for k in range(1,10,0.001):
    x = x+5
```

MATLAB for loop, $k = \text{first} : \text{increment} : \text{last}$

(could omit increment to default to 1, like Python)

Functions in MATLAB

```
function [output1,output2] = practice(input1, input2)

output1 = input1 * 5;
output2 = input2 * 10;

end
```

Easy to return multiple values, no “return” statement
Autoruns first function which should have same name
as .m file

Inputs and outputs are optional

- **function** testFunction
 - No inputs or outputs
- **function** [x] = testFunction2
 - Only 1 output called x
- **function** testFunction3(n)
 - Only 1 input called n
- **function** [y] = testFunction4(a,b,c)
 - 3 inputs a, b, c and 1 output y
- If the primary function has inputs call from command line, if no inputs needed you can hit Run (or F5)

MATLAB scripts vs MATLAB functions



- MATLAB scripts

- No code word function just code
- All variables visible in workspace
- No subfunctions at all

- MATLAB functions

- First line of code is `function [outputs] = name (inputs)`
- Subfunctions allowed in same .m file
- Variable scope limited to function

Hands on MATLAB script .m files

- Make a .m script file the does the following
- Save your script as practiceScript.m

```
clc
```

```
x = 3+4
```

```
for i = 1:5
```

```
    i
```

```
end
```

Things to try with that script file

- View the output
- Add a ; to the end of the `i;` line
- After running the code type in `x` or `i` in the interactive mode command line
- View the variable in the Workspace
- Notice how the value is stored as a 2D matrix of type double that has 1 row and 1 column.
Remember MATLAB!

Hands on MATLAB function .m files

- One of the first functions we made in Python was a factorial function.
- Make a program that has an m file called “factorialTable.m”
- Make a subfunction called factorial(n) that returns the n! value
 - A little help on the subfunction:

```
function [result] = factorial(n)
result = 1
...
```

Debugger



- In this case MATLAB is more like Eclipse than IDLE
- MATLAB has an easy to use debugger

- Add a breakpoint to the start of your factorialTable code (first line in the factorialTable function)
- Step into the code

Fun quick keys/Shortcuts

- Comment line - Ctrl k
- Uncomment line - Ctrl t
- Select All/Auto Indent - Ctrl a Ctrl I
- Run .m file - F5
- Autocomplete - Tab
- Save - Ctrl s
- Standard copy, cut, paste

Built-in MATLAB functions



- Let's learn about help in MATLAB
- Type in “help prod”

- Read about prod
- What does `prod(1:4)` do?
- What about `prod(1:n)` for you function factorial?
- Click on “doc prod” from the “help prod” text.
 - ▣ Excellent help documents in MATLAB

Help in MATLAB



- Go to the help menu -> Product Help

- In the Search Results tab, look for some things:
 - while
 - function (second item in search)
 - mean
 - whos
 - bench

- Click on the Contents tab -> Getting Started

Matrix operations

- Make a matrix to play with:

- $X = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$

- Or in a different syntax for the same result

- $X = [1 \ 2 \ 3; 4 \ 5 \ 6; 7 \ 8 \ 9]$

- $X = [1,2,3;4,5,6;7,8,9]$

- $X = [1, 2, 3; 4, 5, 6; 7, 8, 9]$

- $X = \begin{bmatrix} 1 & 2 & 3; \\ 4 & 5 & 6; \\ 7 & 8 & 9 \end{bmatrix}$

Get the matrix element

- Get the element of x in row 2 column 3
 - $x(2,3)$
- Set the element of x at row 2 column 3
 - $X(2,3) = 17$
- Get the first column of elements (All rows, column 1)
 - $x(:,1)$
- Slice the matrix to get the 2 by 2 upper left corner
 - $x(1:2,1:2)$
- Similar to Python list slicing but base 1.

Changing the size of the matrix

- Add a new column to our 3 by 3, X matrix
 - $X(:,4) = [10; 11; 12]$
- Add a new row to our 3 by 4, X matrix
 - $X(4,:) = [13\ 14\ 15\ 16]$

Doesn't throw an array out of bounds error, just works and expands the matrix for the new index

- Get the size of the matrix
 - $[R,C] = \text{size}(X)$

Vector operations



- Simple vector syntax
- `t = 1:10`
- `t = 1: 0.01: 10`
- Get the first 5 elements of `t`
 - `t(1:5)`
- Get the last 5 elements of `t`
 - `t(end-4:end)`
- Get the vector length
 - `length(t)`

Plotting in MATLAB

- All plots are based on points, unlike Maple
- Make a vector of x values
- Make a vector of y values
- Plot x vs y
- Sample:
 - $x = -\pi:0.1:\pi;$
 - $y = \sin(x);$
 - `plot(x,y)`
 - Now try `plot(x,y,'b.')`

Changing the step size

- Try a worse resolution:

- `x = -pi:0.5:pi;`

- `y = sin(x);`

- `plot(x,y,'b.')`

- Try a better resolution:

- `x = -pi:0.001:pi;`

- `y = sin(x);`

- `plot(x,y,'b.')`

- Use ***help plot*** to make a Black Dashed line

Sample Projectile Ball Problem

- Suppose we have a ball that we are throwing and we want to plot the position of the ball.
- We know the initial velocity of the ball, the angle of the initial velocity.
- We want a plot of the ball and the time when the ball hits the ground.
 - Store each time step into a matrix
 - Row 1 – Time
 - Row 2 – X position
 - Row 3 – Y position
 - Assume ideal world with only gravity

I was kind enough to start you off



- Go to Angle and download some code to get you started.

Continued Projectile Ball Problem

- #1. Solve for the default case first
 - Ball initial velocity = 5 m/s
 - Angle of throw = 30 degrees

- #2. Solve for any case
 - Make a function say `projectileBall` that takes two inputs (`initialVelocity`, `launchAngle`), plots the ball, and returns the time of the flight [`flightTime`]

Information about ME1 23



- [http://www.rose-hulman.edu/ME1 23/](http://www.rose-hulman.edu/ME123/)