

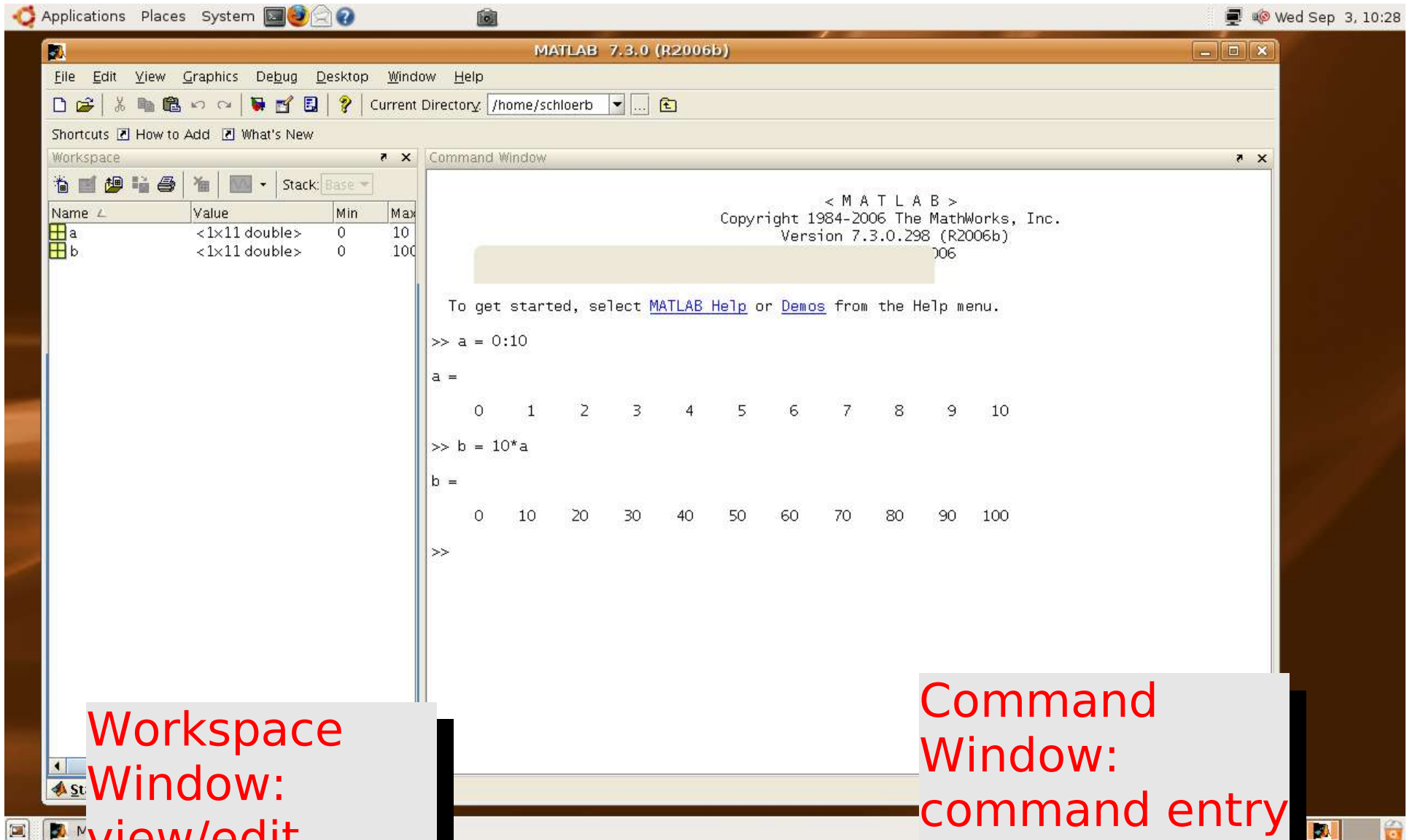
MATLAB I

Computational Physics

MATLAB (you need to know) Lecture 1

Outline

- MATLAB Environment
- MATLAB Mathematics
 - Variables
 - Mathematical operations
 - Colon Operator
- MATLAB Script



Workspace
Window:
view/edit
variable arrays

Command
Window:
command entry
and results

Useful Windows

- **Command:** Issue MATLAB commands and view printed results
- **Workspace:** View arrays that are present in memory
- **Help:** help with MATLAB commands
- **Editor/Debugger:** Write, Edit, Save, Run, Debug MATLAB scripts.
- **Profiler:** Measure time consumed in program steps.

Command Line Input

Help

- help followed by command or function name will give information about that command.
- Examples:
 - help fprintf - produces help information for fprintf, which is used later in this lecture.
 - help plot - produces help information on the plot function

MATLAB

Variables

- Fundamental Data Type: *Matrix*

$$C = \begin{bmatrix} 0 & 1 & 1 \\ 2 & 3 & -1 \\ 0 & 0 & 1 \end{bmatrix}$$

```
C = [0 1 1; 2 3 -1; 0 0 1];
```

- Special Cases

- 1x1 Matrix = scalar

```
Z = 1;
```

- 1xN Matrix = row vector

```
a = [0 3 4];
```

- Nx1 Matrix = column vector

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

MATLAB

Mathematics Operations

- All matrices defined as they are used; no strict typing as in C.
- Operations
 - $+$ $-$ $*$ $/$ are *matrix* operations.
 - Special "." operation works element-by-element: ($.+$ $.-$ $.*$ $./$)
- ' (apostrophe) operator: Hermitian Conjugate (transpose of complex conjugate)
- .' operator: Transpose of Matrix

Matrix Operations

Examples

```
>> t = [1 2 3 4 5];
```

Define Row Vector

```
>> z = t*t;
```

Try to multiply
row x row ...

```
??? Error using ==> *
```

doesn't work!!

Inner matrix dimensions must agree.

```
>> w = [1; 2; 3; 4; 5;];
```

Make Column Vector

```
>> z = t*w
```

row x column works!
(NOTE: no ; at end of
line causes result to
print)

```
z =
```

```
55
```

```
>> x = t.*t
```

example of
.* operation

```
x =
```

```
1 4 9 16 25
```

The Colon Operator

- Define a vector:

```
t = [1 2 3 4 5 6 7 8 9 10]
```

```
t = 1:10;
```

- Define

```
x=[0 0.2 0.4 0.6 0.8]
```

```
x = 0:0.2:0.8;
```

- Access a row or column in a matrix:

```
A = [1 2 3; 4 5 6; 7 8 9]
```

```
first_row = [1 2 3]
```

```
first_col = [1; 4; 7;]
```

```
% A is a matrix
```

```
% row vector:
```

```
first_row = A(1 , :);
```

```
% column vector:
```

```
first_col = A(:, 1);
```

MATLAB *clear* Command

- MATLAB creates new vectors and matrices as you "request" them.
- These remain in the environment until you quit MATLAB.
- Sometimes it is necessary to enter *clear* command to reinitialize.

```
>> a = 1:2
a =
    1    2
>> a(5) = 12
a =
    1    2    0    0   12
>> clear
>> a
??? Undefined function
>> a = 1:2
a =
    1    2
```

MATLAB M-Files

- M-Files allow us to carry out tasks requiring many steps using MATLAB.
- Two types of M-Files in MATLAB
 - **Script**: a file containing a sequence of MATLAB statements executed one at a time as if typed at the keyboard.
 - **Function**: a file containing the definition of a new MATLAB function with specific inputs and outputs.
- For now, let's deal with scripts.

Example of a Script

Script is saved in file:
simple.m

```
% a simple script example
% Author: F. P. Schloerb
% multiplies a number supplied by the user by x2

x = input('Enter a Number: ')

y = 2*x
```

Comments begin with % character. Useful to explain what script does.

Input function requests a value

Then we multiply by 2

File Edit Debug Desktop Window Help

- New
- Open... Ctrl+O
- Close Command Window
- Import Data...
- Save Workspace As...
- Set Path...
- Preferences...
- Page Setup...
- Print...
- Print Selection...
- 1 /home/...uff/ph281/fall.m
- 2 /home/schloerb/newgeom.m
- 3 /home/schloerb/fall.m
- 4 /home/schloerb/gh.m
- Exit MATLAB Ctrl+Q

Current Directory: /home/schloerb/Mystuff/ph281

Command Window

```
>> cd Mystuff/ph281/
>> fall
Enter initial height (m): 500
Here are results:
t      x
0.0    500.00
1.0    495.10
2.0    480.40
3.0    455.90
4.0    421.60
5.0    377.50
6.0    323.60
7.0    259.90
8.0    186.40
9.0    103.10
10.0    10.00
Enter initial height (m): 500
Here are results:
t      x
---    -
0.0    500.00
1.0    495.10
2.0    480.40
3.0    455.90
4.0    421.60
5.0    377.50
6.0    323.60
7.0    259.90
8.0    186.40
9.0    103.10
10.0    10.00
>>
```

Select M-File under New to Edit Scripts



Matlab 7.3

Editor - /home/schloerb/Mystuff/ph281/fall.m

File Edit Text Go Cell Tools **Debug** Desktop Window Help

Open M-Files when Debugging
 Step F10
 Step In F11
 Step Out Shift+F11
Run F5
 Go Until Cursor
 Set/Clear Breakpoint F12
 Set/Modify Conditional Breakpoint...
 Enable/Disable Breakpoint
 Clear Breakpoints in All Files
 Stop if Errors/Warnings...
 Exit Debug Mode

```

1 % falling ball demo script
2 % Author: F. P. Schloerb
3 % computes solution of
4 %   x(t) = x(0) - 1/2
5 %
6
7 % define an array of times
8 % array t spans times from 0 to 10
9 t = 0:1:10;
10
11 % get the initial height
12 h = input('Enter initial height (m): ');
13
14 % compute the array of positions
15 % note use of .* operator
16 x = h - 0.5 * 9.8 * t.*t;
17
18 % plot the result - with labels for axes
19 plot(t,x);
20 xlabel('time (s)');
21 ylabel('position (m)');
22 title('Falling Ball');
23
24 % display results in a table that looks nice
25 fprintf('Here are results:\n')
26 fprintf('  t      x\n')
27 fprintf(' ---  -----\n')
28 for i=1:11
29     fprintf('%5.1f %10.2f\n',t(i),x(i))
30 end
31
  
```

stack: Base

mes, t
on of t arrays

script Ln 28 Col 11 DVR

Select Run or Save and Run to run Script from Editor Window