What is MATLAB?

• MATrix LABoratory
  – High-level language for numerical computation, visualization, and application development
  – Functions for integrating MATLAB based algorithms with external applications and languages such as C, Java, .NET, and Microsoft® Excel®
  – $99 for students
  – www.mathworks.com
What to do with MATLAB?
The MATLAB Alternatives

• Octave
  – Initially developed by John Eaton at the University of Wisconsin–Madison.
  – Supports for Linux, BSD (FreeBSD & OpenBSD), MacOS X, Cygwin and MinGW.
The MATLAB Alternatives

- **FreeMat**
  - Developed by Demetrios et al.
  - Supports Windows, Linux and Mac OS X.
  - [http://freemat.sourceforge.net](http://freemat.sourceforge.net)
The MATLAB Alternatives

• Scilab
  – Developed by Scilab Enterprises.
  – Available under GNU/Linux, Mac OS X and Windows XP/Vista/7/8.
  – [http://www.scilab.org](http://www.scilab.org)
Getting into MATLAB

MATLAB desktop keyboard shortcuts, such as Ctrl+S, are now customizable. In addition, many keyboard shortcuts have changed for improved consistency across the desktop.

To customize keyboard shortcuts, use Preferences. From there, you can also restore previous default settings by selecting “R2009a Windows Default Set” from the active settings drop-down list. For more information, see Help.

Click here if you do not want to see this message again.
• `variblename = expression`

```plaintext
>> mynum = 7
mynum =
    7
>>
>> 6 + 3
ans =
    9
>> res = 6 + 3
res =
    9
>> clear
```
**Commands Relate to Variables**

- **who**
  - shows variables that have been defined in this Command Window. (this just shows the names of the variables)

- **whos**
  - shows variables that have been defined in this Command Window. (this shows more information on the variables, similar to what is in the Workspace Window)

- **clear**
  - clears out all variables so they no longer exist.

- **clear variablename**
  - clears out a particular variable.

- **clc**
  - clear Command Window.
Operators

- Numeric Expressions
  - + addition
  - – negation, subtraction
  - * multiplication
  - / division (divided by e.g. 10/5 is 2)
  - \\ division (divided into e.g. 5\10 is 2)
  - ^ exponentiation (e.g., 5^2 is 25)

$$\gg 4 + 5 * 3$$

ans =

19

$$\gg (4 + 5) * 3$$

ans =

27
Built-In Functions and Help

• help elfun
  – Elementary math functions.

• help ops
  – Operators and special characters.

• help functionname
  – To find out what a particular function does.

• e.g.
  – \( \text{abs}(-4) \)  \( \text{result} = 4 \)
  – \( \text{rem}(13,5) \)  \( \text{result} = 3 \)
Constants

- **pi**
  - \(3.14159\ldots\)

- **i**
  - \(\sqrt{-1}\)

- **j**
  - \(\sqrt{-1}\)

- **inf**
  - Infinity \(\infty\)

- **NaN**
  - stands for “not a number”; e.g., the result of \(0/0\)
General Commands

• **imread**
  – reads an image.

• **figure**
  – creates a figure on the screen.

• **imshow(g)**
  – displays the matrix \( g \) as an image.

• **pixval on**
  – turns on the pixel values in our figure.

• **impixel(i,j)**
  – returns the value of the pixel \((i,j)\).

• **imfinfo**
  – information about graphics file.
Vectors and Matrices

• Creating Row Vectors

\[
\text{>> } r = [1 2 3 4] \\
r = \\
1 2 3 4 \\
r = [1,2,3,4] \\
r = \\
1 2 3 4 \\
\]

• Creating Column Vectors

\[
\text{>> } c = [1;2;3] \\
c = \\
1 \\
2 \\
3 \\
\text{>> } r = 1:3; \\
c = r' \\
c = \\
1 \\
2 \\
3 \\
\]
Colon Operator & *linspace* Func.

```matlab
>> vec = 1:5
vec =
    1 2 3 4 5

>> nv = 1:2:9
nv =
    1 3 5 7 9
```

- **linspace function**
- creates a linearly spaced vector; `linspace(x,y,n)`

```matlab
>> ls = linspace(3,15,5)
ls =
    3 6 9 12 15
```
Matrix Variables

• Creating Matrix Variables

\[
\begin{align*}
\text{>> mat} &= [4 \ 3 \ 1; \ 2 \ 5 \ 6] \\
\text{mat} &= \\
&= \begin{bmatrix}
4 & 3 & 1 \\
2 & 5 & 6
\end{bmatrix}
\end{align*}
\]

\[
\begin{align*}
\text{>> mat} &= [2:4; \ 3:5] \\
\text{mat} &= \\
&= \begin{bmatrix}
2 & 3 & 4 \\
3 & 4 & 5
\end{bmatrix}
\end{align*}
\]

\[
\begin{align*}
\text{>> randint(2,4,[10,30])} \\
\text{ans} &= \\
&= \begin{bmatrix}
29 & 22 & 28 & 19 \\
14 & 20 & 26 & 10
\end{bmatrix}
\end{align*}
\]
2D Matrix

- 2D matrix has M rows and N columns
- Example
  
  >>> m=zeros(400,200);
  >>> m(1:50,1:100) = 1;

- Note:
  - Index 1 to M, not 0 to M-1
  - Origin of coordinate system is in upper left corner
  - Row index increases from top to bottom
  - Coordinate system is rotated in respect to ‘standard’ x-y coordinate system
• Display a 2D matrix as an image
  
  \[ \text{>> } m = \text{zeros}(400,200); \]
  
  \[ \text{>> } m(1:50,1:100) = 1; \]

• `imshow` is recommended by MATLAB for image processing

• Image has \( M \times N \) pixels = picture elements = dots on display

• Test the following commands

  \[ \text{>> } \text{imshow}(m); \]
  
  \[ \text{>> } \text{axis on}; \text{colorbar}; \]
Save Image Matrix

• Standard image format (tiff, png, jpeg, bmp, gif…)
  – save as tif, png, jpg, bmp, gif, …
    • >File>Export in Figure Window
      >> imwrite(m,'example.tif','tif')
      >> imwrite(m,'example.png','png')
      >> imwrite(m,'example.jpg','jpg')
      >> imwrite(m,'example.bmp','bmp')
      >> imwrite(m,'example.gif','gif')

• Save in Matlab Figure Format:
  – writes out example.fig
    • >File>Save As> in Figure Window
Loading 2D Data and Images

• Load matrix from standard image format
  
  \[
  m_{\text{in}} = \text{imread('example.tif', 'tif')}
  \]
  
  – To check on an image:
  
  \[
  \text{imfinfo('example.tif', 'tif')}
  \]

• Load figure from Matlab Figure Format (*.fig):
  
  – >File>Open> ‘example.fig’ in Figure Window

• Check loaded matrices
  
  \[
  \text{whos}
  \]
Some Common Image Formats

• **TIF (TIFF)**
  – Very common, compatible, all purpose image format
  – Allows for lossless compression

• **JPEG**
  – Allows for lossy compression (small file size)
  – Very common for internet

• **PNG (portable network graphics)**
  – Good for saving MATLAB images for importing into Microsoft documents such as Word

• **EPS (Encapsulated Postscript)**
  – Graphics format rather than an image format
  – Great for best quality print results
  – Large file size

• **DICOM**
  – The medical imaging format
  – Lots of header information (patient name & ID, imaging parameters, exam date, …) can be stored
  – Allows for lossy and lossless compression
  – Matlab function ‘dicomread’, ‘dicomwrite’
• Create a matrix that evaluates 2D Gaussian: \( \exp(-\pi/2 (x^2 + y^2)/\pi^2) \)

\[
\text{>> ind = [-32:1:31] / 32;}
\]
\[
\text{>> [x,y] = meshgrid(ind,-1*ind);}
\]
\[
\text{>> z = exp(-pi/2*(x.^2+y.^2)/(.25.^2));}
\]
\[
\text{>> imshow(z)}
\]
\[
\text{>> colorbar}
\]
References

• Book
  – “MATLAB - A Practical Approach” by Stormy Attaway

• Publication
  – “A Comparative Evaluation of MATLAB, OCTAVE, FREEMAT, and SCILAB for Research and Teaching”

• Lecture Note
  – “Introduction to Image Processing with MATLAB” by Wally Block
Do you have any questions?

Thank you

Junmo An
jman@cs.uh.edu
http://mrl.cs.uh.edu