GETTING STARTED WITH DATA: MATLAB

WiO Workshop

01.28.2016

Presenter: Kelsey Miller

MATLAB

MATrix LABoratory

- Multi-paradigm, numerical computing environment
- 4th gen programming language
- Allows matrix manipulation, function and data plotting, algorithm implementation, user interface creation, and interfacing with other languages including 3rd gen languages: C, C++, Java, Fortran and Python

An Intro to **MATLAB**

MATLAB

An Intro to **MATLAB**

In layman's terms:

- Powerful, user-friendly tool for data analysis that plays well with other languages

What We'll Be Covering

- Loading data into MATLAB
- Organizing data in your workspace
- Using for loops to deal with large data sets
- Exporting your data

- load
- importdata
- xlsread
- imread
- fitsread

Loading Data into MATLAB

load

- Function: Loads variables from a file into the workspace (output in double)

- Syntax:

load filename

A = load('filename')

A = load('filename', variables)

A = load('filename', '-format')

A = load('filename', '-format', variables)

- Example:

exampleload.m

- Function: Loads variables from a file into the workspace. Can be used to load image files (output in uint8) and can read header information in ASCII file

- Syntax:

(paste from clipboard)

A = importdata('-pastespecial')

(import from folder)

A = importdata('filename')

A = importdata('filename','delimiterIn')

A = importdata('filename' ,'delimiterIn','headerlinesIn')

- Example:

exampleimportdata.m

importdata

xlsread

- Function: Reads Microsoft Excel spreadsheet file (output in double).

- Syntax:

A = xlsread('filename')

A = xlsread('filename', sheet)

A = xlsread('filename', range)

A = xlsread('filename', sheet, range)

[data, txt, raw] = xlsread('filename', sheet, range)

- Example:

examplexlsread.m

- Function: Read an image from a graphics file

(Formats supported: BMP, CUR, GIF, HDF4, ICO, JPEG, JPEG 2000, PBM, PCX, PGM, PNG, PPM, RAS, TIFF, XWD.) (Output type depends on file bit depth.)

- Syntax:

A = imread('filename')

A = imread('filename' , 'format')

[A , map] = imread('filename')

imread

(For information about graphics file)

info = imfinfo('filename')

- Example:

exampleimread.m

fitsread

- Function: Reads data from a FITS file (Flexible Image Transport System)*

- Syntax:

data = fitsread('filename')

info = fitsinfo('filename')

- Example:

examplefitsread.m

*http://www.mathworks.com/help/matlab/import_export/importing-flexible-image-transport-system-fits-files.html

- Navigating the command window
- Calling/partitioning selected data
- Cell arrays
- Data cubes

Organizing Data

Navigating the Command Window

- Print working directory

  pwd

- Change directory

  cd directoryname

  cd ..

- Create new directory

  mkdir directoryname

- Remove directory

  rmdir directoryname s

- Copy files

  copyfile ***filename-to-be-copied foldername-to-copy-to***

- Move files

  movefile ***filename-to-be-moved foldername-to-move-to***

- List files

  ls OR ***dir*** ***directoryname***

- Search specific file type

  ls ****.filetype***

- Delete files

  delete ***filename***

Navigating the Command Window

Calling/Partitioning Selected Data

- Calling specific column from matrix A

  x = A(: , column number)

- Calling specific row from matrix A

  x = A(row number, :)

- Call specific value from matrix A

  x = A(row number, column number)

- Assigning data to new matrix B
  - Create matrix B

    B = zeros (number of rows, number of columns)
  - Fill matrix B with data from matrix A

    B(: , column number) = A(: , column number)

  - <u>Example:</u>

  example_SelectingData.m

Calling/Partitioning Selected Data

Cell Arrays

- Create a cell array to hold multiple matrices, images, etc. all of various sizes

  A = cell(number of rows, number of columns)

  Can create multi-dimensional arrays:

  A = cell(m, n, p, o, q ……)

- Fill cell array

  Assign data to first cell in simple 2D array

  A{ 1, 1 } = data

- <u>Example:</u>

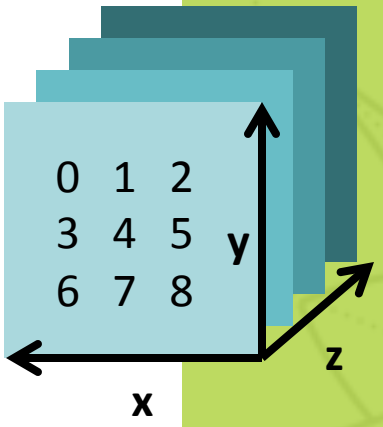
  example_CellArray.m

Data Cubes

- Create a ‘cube’ matrix to contain multiple data sets of the same size. NOTE: Not limited to 3 dimensions

  Create empty 3D data cube

  A = zeros(y, x, z)



- Fill data cube

  Assign data to first position in data cube

  A(:, :, 1) = data

- Useful for group statistics and single image stats on large numbers of images when used in a for loop (covered in the following chapter)

- Example Problem: Finding the mean value and standard deviation of a single pixel across multiple images

Data Cubes

M = mean(DataCube(row num, column num, :))

S = std(DataCube(row num, column num, :))

- Example:

  example_DataCube.m

- Using *for* loops to load multiple files
- Using *for* loops in data analysis

Looping Data

Loading Multiple Files

- Useful to use *for* loops when loading large number of files into your workspace

- Think ahead when coming up with your data naming scheme:
  - Good example:

    data_1.fits

    data_2.fits

    …
- Allows for easy implementation of a *for* loop in both loading and organizing your data files in your workspace

- Sample syntax:

Loading data into a Data Cube

```
N = number of files to be loaded
DataCube = zeros(num rows, num cols, N)

for k = 1 : N
        filename = horzcat( 'file_',num2str(k),'.txt')
        DataCube( :, :, k ) = load(filename)
end
```

Loading Multiple Files

Loading Multiple Files

- Sample syntax:

Loading data into a Cell Array

```
N = number of files to be loaded
CellArray = cell(1,N)

for k = 1 : N
        filename = horzcat( 'file_',num2str(k),'.txt')
        CellArray{ k } = load(filename)
end
```

- Example:

example_LoadingMultipleFiles.m

- Using *for* loops to iterate through data cubes

- Example Problem: Finding the mean of every image stored in a data cube

```
N = number of files in data cube
Mvector = zeros(1,N)
for k = 1 : N
        Mvector(k) = mean2(DataCube( :, :, k ))
end
```

- Example:

  example_DataCube.m

Data Analysis

- save
- saveas
- xlswrite
- fitswrite
- Communicating with Zemax

Exporting Data

save

- Function: Saves workspace variables to .mat file or ASCII

- Syntax:

save filename

save filename variables

save('filename', 'variables', '-format')

save('filename', 'variables', '-append')

- Example:

examplesave.m

saveas

- Function: Saves figure to specific file format. Can be saved as .m, FIG, JPEG, PNG, PDF, TIF, BMP, etc. (Saves figure to a .fig file if the file extension is unspecified.)

- Syntax:

(current figure)

saveas(gcf, filename)

saveas(fig, filename)

saveas(fig, filename, formattype)

- Example:

examplesaveas.m

xlswrite

- Function: Writes data to Microsoft Excel spreadsheet file

- Syntax:

xlswrite('filename', variable)

xlswrite('filename', variable, sheet number, range)

xlswrite('filename', variable, sheet number)

xlswrite('filename', variable, range)

- Example:

examplexlswrite.m

fitswrite

- Function: Writes an image to a FITS file

- Syntax:

fitswrite(imagedata, 'filename')

- Example:

examplefitswrite.m

Communicating with Zemax

- Function: Uses Zemax's built in DDE (Dynamic Data Exchange) server to communicate with MATLAB as a client application.

- Can talk both ways between MATLAB and Zemax
  - Push lens data from MATLAB into the Zemax LDE (Lens Data Editor)
  - Pull data/plots from Zemax into the MATLAB workspace

- Example:

ZemaxMatlabCom.m

- Functionality:
  - Atmosphere, Sun and Astronomy
  - Radiometry, Photometry, Spectral Filtering, Signal Chain and Processing
  - Transfer Functions: OTF, MTF, Spatial Filtering, etc.
  - Image Processing and Exploitation
  - Input and Output
  - Optical Modelling, Analysis and Testing
  - Surveillance
  - Internal Zemax Commands
  - Plotting/Miscellaneous

Communicating with Zemax

- Reference:

http://www.zemax.com/support/resource-center/knowledgebase/how-to-talk-to-zemax-from-matlab

Communicating with Zemax

Examples URL:

wp.optics.arizona.edu/womeninoptics/

STUDENT OPTICS CHAPTER
COLLEGE OF OPTICAL SCIENCES - THE UNIVERSITY OF ARIZONA

Workshops in MATLAB & Python

Offered by SOCk

Mid-to-end of February

- Covering MATLAB toolboxes and publication workflow (e.g. function handles, memory allocation, interfacing with LaTeX)
- Replacing both Mathematica and MATLAB with Python

But Wait….
There's MORE!