MATLAB for Engineers

Mauricio Villarroel

Centre for Doctoral Training in Healthcare Innovation
Institute of Biomedical Engineering
Department of Engineering Science
University of Oxford

Healthcare Innovation
CENTRE FOR DOCTORAL TRAINING
Research Councils UK
Digital Economy
Transforming Business and Society

"Supported by the RCUK Digital Economy Programme grant number EP/G036861/1"

May 31st, 2012

Outline

1. The MATLAB environment

2. Naming convention

3. Data types

4. Import/export data

5. Practical session

Outline

1 The MATLAB environment

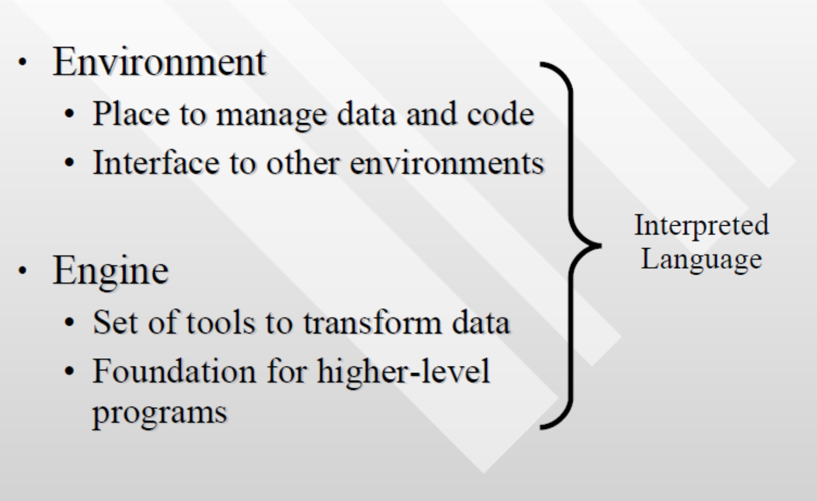
2 Naming convention

3 Data types

4 Import/export data

5 Practical session

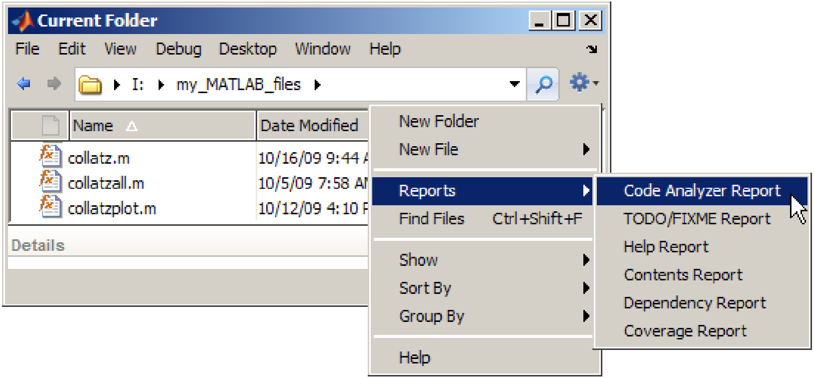
The MATLAB environment



- Environment
  - Place to manage data and code
  - Interface to other environments
- Engine
  - Set of tools to transform data
  - Foundation for higher-level programs

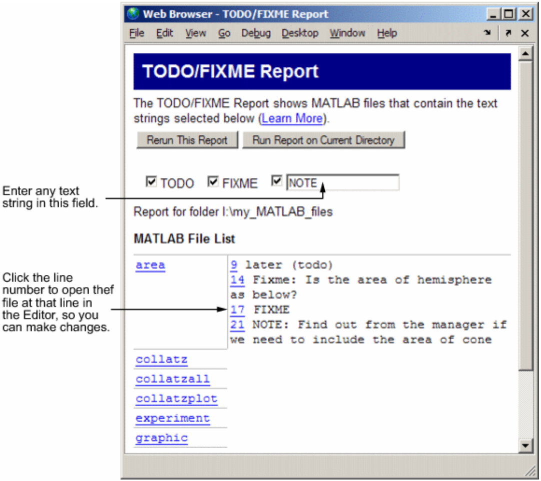
Interpreted Language

Directory reports



- Help you refine code and improve their performance.
- Useful for checking the quality of files BEFORE you SHARE them with others

TODO/FIXME report



Web Browser - TODO/FIXME Report

File Edit View Go Debug Desktop Window Help

TODO/FIXME Report

The TODO/FIXME Report shows MATLAB files that contain the text strings selected below (Learn More).

Rerun This Report
Run Report on Current Directory

☑ TODO ☑ FIXME ☑ NOTE

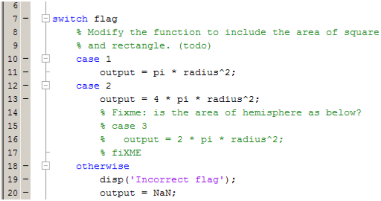
Enter any text string in this field.

Report for folder I:\my_MATLAB_files

MATLAB File List

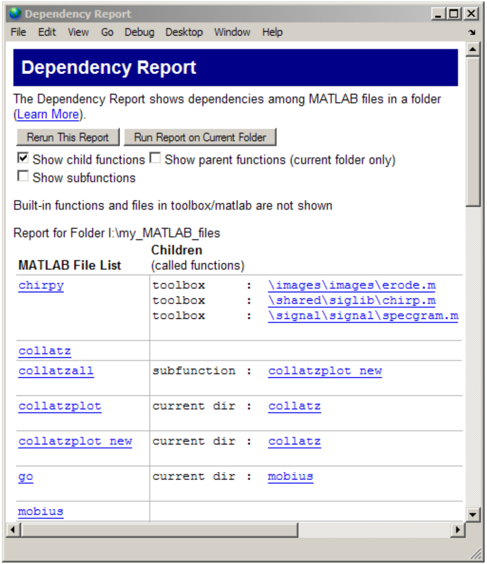
area	9 later (todo) 14 Fixme: Is the area of hemisphere as below? 17 FIXME 21 NOTE: Find out from the manager if we need to include the area of cone
collatz	
collatzall	
collatzplot	
experiment	
graphic	

Click the line number to open thef file at that line in the Editor, so you can make changes.



```
 6
 7   switch flag
 8       % Modify the function to include the area of square
 9       % and rectangle. (todo)
10       case 1
11           output = pi * radius^2;
12       case 2
13           output = 4 * pi * radius^2;
14           % Fixme: is the area of hemisphere as below?
15           % case 3
16           %   output = 2 * pi * radius^2;
17           % fiXME
18       otherwise
19           disp('Incorrect flag');
20           output = NaN;
```

Dependency report



Dependency Report

The Dependency Report shows dependencies among MATLAB files in a folder (Learn More).

☑ Show child functions ☐ Show parent functions (current folder only)
☐ Show subfunctions

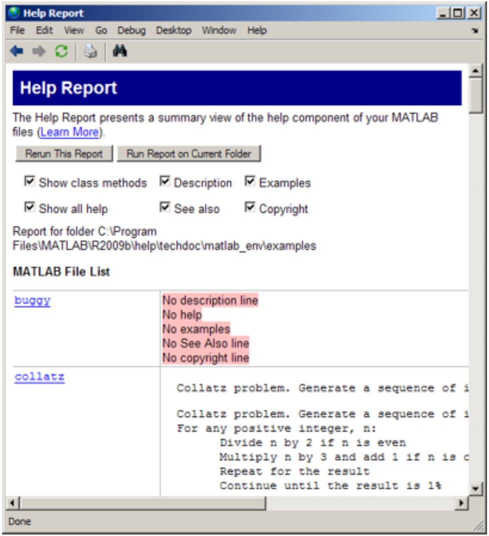
Built-in functions and files in toolbox/matlab are not shown

Report for Folder I:\my_MATLAB_files

MATLAB File List	Children (called functions)
chirpy	toolbox : \images\images\erode.m toolbox : \shared\siglib\chirp.m toolbox : \signal\signal\specgram.m
collatz	
collatzall	subfunction : collatzplot_new
collatzplot	current dir : collatz
collatzplot_new	current dir : collatz
go	current dir : mobius
mobius	

- Which files in the folder are required by other files in the folder.
- If any files in the current folder will fail if you delete a file.
- If any called files are missing from the current folder.

Help report


Help Report
File Edit View Go Debug Desktop Window Help
Rerun This Report
Run Report on Current Folder
Done

Help Report

The Help Report presents a summary view of the help component of your MATLAB files (Learn More).

☑ Show class methods ☑ Description ☑ Examples

☑ Show all help ☑ See also ☑ Copyright

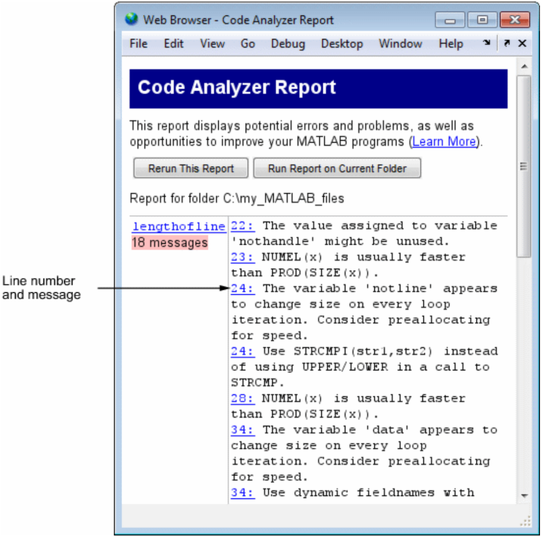
Report for folder C:\Program Files\MATLAB\R2009b\help\techdoc\matlab_env\examples

MATLAB File List

buggy	No description line No help No examples No See Also line No copyright line
collatz	Collatz problem. Generate a sequence of i Collatz problem. Generate a sequence of i For any positive integer, n: Divide n by 2 if n is even Multiply n by 3 and add 1 if n is c Repeat for the result Continue until the result is 1%

- Summary view of the help component of your MATLAB files.
- Assist you in identifying undocumented files.

Code analyzer


Web Browser - Code Analyzer Report
File Edit View Go Debug Desktop Window Help
Rerun This Report
Run Report on Current Folder
Line number and message

Code Analyzer Report

This report displays potential errors and problems, as well as opportunities to improve your MATLAB programs (Learn More).

Report for folder C:\my_MATLAB_files

lengthofline
18 messages

```
22: The value assigned to variable
'nothandle' might be unused.
23: NUMEL(x) is usually faster
than PROD(SIZE(x)).
24: The variable 'notline' appears
to change size on every loop
iteration. Consider preallocating
for speed.
24: Use STRCMPI(str1,str2) instead
of using UPPER/LOWER in a call to
STRCMP.
28: NUMEL(x) is usually faster
than PROD(SIZE(x)).
34: The variable 'data' appears to
change size on every loop
iteration. Consider preallocating
for speed.
34: Use dynamic fieldnames with
```

- Displays potential errors and problems in your code
- Opportunities for improvement

Source control

- Windows
  - Microsoft Visual SourceSafe
  - Microsoft Common Source Control standard
- Unix
  - IBM Rational
  - Concurrent Version System (CVS)
  - ChangeMan/PVCS
  - Revision Control System (RCS)

SubVersion

- Centralized version control system
- Download from: http://subversion.apache.org
- Available in the lab through: https://ibme-web.eng.ox.ac.uk/svn/**project_name**
- Clients for Windows:
  - TortoiseSVN (http://tortoisesvn.tigris.org)
  - RapidSVN (http://rapidsvn.tigris.org)
  - VisualSVN (http://www.visualsvn.com/visualsvn)
- Clients for Unixes
  - KDE SVN (KDE distribution)
  - QSvn (http://www.anrichter.net/projects/qsvn)

Section summary

- Use the tools: MATLAB "Can" help you refine your code and improve performance with not too much effort.
- Use a source control system to manage your code.
- BackUP your code/data
- Let us REUSE your code

Outline

1. The MATLAB environment
2. Naming convention
3. Data types
4. Import/export data
5. Practical session

Variable naming conventions

- It's all about consistency!
- Variable names in mixed case starting with lower case: *line*, *startSample*, *titleFontSize*
  (Other alternative: Use underscores but watch out with Matlab's LaTeX interpreter)
- Short vs long names: Check variable scope: "*elementIndex*" vs "*i*"
- Application domain: "*fs*" = Sampling frequency

Variable naming conventions

- The prefix n should be used for variables representing the number of objects:
  nFiles, nSegments, nFrames
- Singular vs plural variables, choose one consistently.
- Variables representing a single entity number can be suffixed by “*No*” or prefixed by “*i*” :
  iFile, fileNo, iByte, byteNo (Choose one)
- Avoid using a keyword or special value name for a variable name:
  filter, mean, ...

Variable naming conventions - Constants

- Named constants should be all uppercase using underscore to separate words:
  MAX_ITERATIONS, COLOR_RED
- Constants can be prefixed by a common type name. This gives additional information on which constants belong together and what concept the constants represent:
  COLOR_RED, COLOR_GREEN, COLOR_BLUE

Variable naming conventions - Structures

- Structure names should begin with a capital letter. It helps to distinguish between structures and ordinary variables.
- The name of the structure is implicit, do not include it in a fieldname:
  Use Segment.length
  Avoid Segment.segmentLength

Variable naming conventions - functions

- Names of functions should be written in lower case. It helps to have the function and its m-file names the same. (Portability to other OSes): *getname(.), computetotalwidth(.)*
- The prefix "*compute*" can be used in methods where something is computed.
- The prefix "*find*" can be used in methods where something is looked up.
- The prefix "*initialize*" can be used where an object or a concept is established.
- The prefix "*is*" should be used for boolean functions.

Variable naming conventions

Avoid cryptic code, favor readability vs apriori optimization (use the profiler):

```
if (value>=lowerLimit)&(value<=upperLimit)&
   ~ismember(value, valueArray)
    :
end
```

Should be replaced by:

```
isValid = (value >= lowerLimit) & (value <= upperLimit);
isNew   = ~ismember(value, valueArray);

if (isValid & isNew)
    :
end
```

Section summary

- It's all about consistency!
- Names should be meaningful in the **application** domain, not the **implementation** domain
- Favor readability vs optimization
- Document your code to provide better understanding after a long interval of time.
- Let us REUSE your code

Outline

1 The MATLAB environment

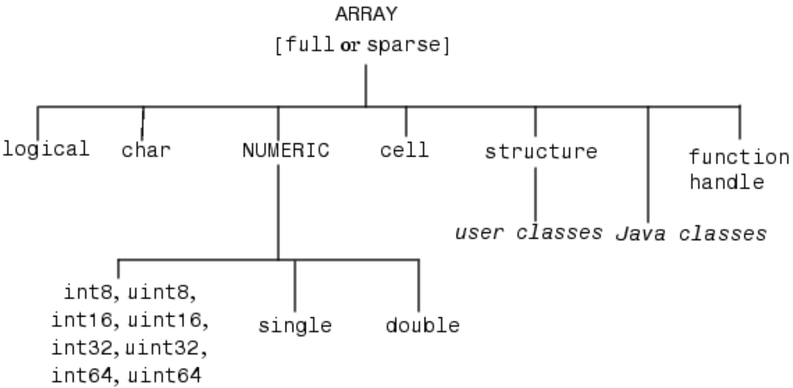
2 Naming convention

3 Data types

4 Import/export data

5 Practical session

MATLAB Data types



ARRAY
[full or sparse]

logical

char

NUMERIC

cell

structure

function handle

user classes

Java classes

int8, uint8,
int16, uint16,
int32, uint32,
int64, uint64

single

double

Structures

- N-D matrix of a special data type
- All structs in the array have the same number of fields.
- All structs have the same field names.
- Fields of the same name in different structs can contain different types or sizes of data.
- Example: *files* = *dir* returns Nx1 vector of structures with fields:

```
files(8).name = 'testscript.m'
files(8).date = '21-Dec-2007␣14:35:25'
files(8).bytes = 376
files(8).isdir = false
files(8).datenum = 733397.6079282408
```

- Fields can contain any data type

Structures - examples

- Time series:

```
       Name: ’Position’
       Time: [5x1 double]
       Data: [5x2 double]
Annotations: [5x1 double]
```

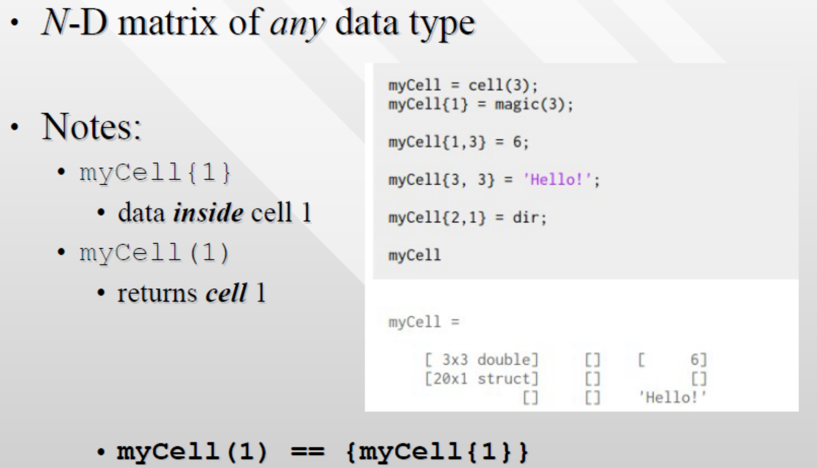
- Dynamic field reference:

```
S = struct('A', [1 2], 'B',[3 4 5]);
SNames = fieldnames(S);
for loopIndex = 1:numel(SNames)
    stuff = S.(SNames{loopIndex})
end
```

Structures - useful commands

Command	Description
fieldnames	Field names of structure
isa	Determine if item is object of given class
isequal	Determine if arrays are numerically equal
isfield	Determine if item is structure array field
isstruct	Determine if item is structure array
orderfields	Order fields of a structure array
rmfield	Remove structure fields
setfield	Sets contents of field
struct	Create a structure array
struct2cell	Convert struct to cell array
cell2struct	Convert cell array to structure

Cell arrays



- *N*-D matrix of *any* data type
- Notes:
  - `myCell{1}`
    - data ***inside*** cell 1
  - `myCell(1)`
    - returns ***cell*** 1

```
myCell = cell(3);
myCell{1} = magic(3);

myCell{1,3} = 6;

myCell{3, 3} = 'Hello!';

myCell{2,1} = dir;

myCell
```

```
myCell =

    [ 3x3 double]     []    [     6]
    [20x1 struct]     []          []
               []     []    'Hello!'
```

- **`myCell(1) == {myCell{1}}`**

Cell arrays - useful commands

- **cellfun** : Apply a function to each element in a cell array. Syntax:

```
[A1,...,Am] = cellfun(func,C1,...,Cn)
```

- Examples: Compute the mean of each vector in cell array C.

```
C = {1:10, [2; 4; 6], []};

averages = cellfun(@mean, C)
```

This code returns

```
averages =
    5.5000    4.0000       NaN
```

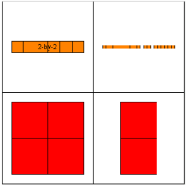
Cell arrays - useful commands

- **cellplot**: Graphically display the structure of cell arrays

```
cellplot(c,'legend')
```

- Examples: Consider the following 2-by-2 cell array:

```
c{1,1} = '2-by-2';
c{1,2} = 'eigenvalues of eye(2)
c{2,1} = eye(2);
c{2,2} = eig(eye(2));
```



Cell arrays - useful commands

Command	Description
cell	Construct cell array
cellstr	Create cell array of strings from character array
cell2mat	Convert cell array of matrices into single matrix
celldisp	Display cell array contents
class	Return object's class name (e.g., cell)
deal	Deal inputs to outputs
isa	Detect object of given class (e.g., cell)
iscell	Determine if item is cell array
iscellstr	Determine if item is cell array of strings
isequal	Determine if arrays are numerically equal
mat2cell	Divide matrix up into cell array of matrices
num2cell	Convert numeric array into cell array

Section summary

- Aprox. 15 fundamental data types in MATLAB.
- Data types are in the form of a matrix or array.
- You can define your own data types using MATLAB classes
- Work/interface with several programming languages: C, Java

Outline

1. The MATLAB environment
2. Naming convention
3. Data types
4. **Import/export data**
5. Practical session

Workflow

Matlab is great for data analysis, but...

- First have to get data INto Matlab!
  csv files, Excel, binary, etc...
- Have to get data OUT
  either as figure or analyzed data
- Sometimes need other applications to do our work

Supported file formats

- Matlab native *.mat files (Mostly incompatible with other programs)
- Text:
  - White space delimited numbers
  - Comma or other delimited characters
  - Mixture of strings and numbers
- Spreadsheets:
  - Microsoft Excel on Windows
  - OpenDocument
- XML
- Physionet's WFDB (separate toolbox)
- Several image, video and audio files

Importing data

- **importdata** command
  - Uses file extension if possible to determine file type
  - If no recognizable file extension, assumes delimited data
  - File type determines data type
- **uiimport** command
  - GUI interface to import data
- **textscan** command
  - Imports to cell arrays
  - Use for non-standard data formats and large files

Exporting data

Export options:

- MAT-Files
- Text Data Files
- XML Documents
- Excel Spreadsheets
- Scientific Data Files: CDF
- Images
- Audio and Video
- Binary Data with Low-Level I/O

Import/Export - useful commands

Command	**Description**
load/save	Load/save data from/into MAT-file workspace
textscan	Read formatted data from text file or string
dlmread/dlmwrite	Read/write ASCII-delimited file of numeric data from/into matrix
xlsread/xlswrite	Read/write Microsoft Excel spreadsheet file
xmlread/xmlwrite	Read/save XML document and return Document Object Model node
imread/imwrite	Read/save image from/into graphics file

Section summary

- Favor open and well-documented data formats.
- If unsure, use the import/export wizard.

Outline

1 The MATLAB environment

2 Naming convention

3 Data types

4 Import/export data

5 Practical session

The PTB Diagnostic ECG Database

Physikalisch-Technische Bundesanstalt (PTB), the National Metrology Institute of Germany.

The ECGs in this collection were obtained using a non-commercial, PTB prototype recorder with the following specifications:

- 12 standard ECG leads, plus 3 orthogonal Frank leads
- Digital resolution: 16 bits
- 1 kHz synchronous sampling frequency for all channels

The PTB Diagnostic ECG Database

Task: Write a function/functions to plot a given length of data and a specific named lead from a record in PTB database

Files given:

- s0016lre.csv : Comma separated value raw ECG data
- s0016lre.hea : Record's metada information in Physionet's format.

Reference:

- Physionet header file format: http://www.physionet.org/physiotools/wag/header-5.htm