MATH 3341: Introduction to Scientific Computing Lab

Libao Jin

University of Wyoming

February 05, 2020

Lab 02: Variables, Arrays, and Scripts

Script Files

A script file is simply a file that contains a chain of commands that you edit in a separate window, then execute with a single mouse click or command. This is where we can define variables, perform calculations and leave commands to remind us what the file calculates.

File Naming Conventions

- "The rules are exactly the same as for variable names: start with a letter, followed by letters or numbers or underscore, maximum 63 characters (excluding the .m extension), and must not be the same as any MATLAB reserved word."
- "None of the conventions matter to MATLAB itself: they only matter to the people writing the code, and the people maintaining the code (usually a much harder task), and to the people paying for the code (you'd be amazed how much gets written into contract specifications.)"

Reference:
`https://www.mathworks.com/matlabcentral/answers/30223-what-are-the-rules-for-naming-script-files`

Put Comments to Your Script File

```
% MATH 3341, Spring 2020
% Lab 02: Variables, Arrays, and Scripts
% Author: first_name last_name
% Date: 02/05/2020
```

Variables

Variables help us represent quantities or expressions in order to make their use and re-use more convenient.

Naming Variables

- Must start with a letter.
- Followed by letters (a-z, A-Z) or numbers (0-9) or underscores (_).
- Maximum 63 characters (excluding the .m extension).
- Must not be the same as any MATLAB reserved word.
- Space is not permitted.
- Case sensitive, i.e., `a ~= A`.

Naming Variables

- Be as descriptive as possible with your variable names.
- Avoid built-in function/variable names (reserved keywords) such as `pi`, `sin`, `exp`, etc.
- Check if a name is already in use: `which variableName` or `exist variableName`.

Naming Conventions

- snake_case: writing compound words or phrases in which the elements are separated with one underscore character (_) and no spaces, e.g. "foo_bar".
- camelCase: writing compound words or phrases such that each word or abbreviation in the middle of the phrase begins with a capital letter, with no intervening spaces or punctuation, e.g. "fooBar"
- Other conventions: Hungarian notation, positional notation, etc.

Reference: `https://en.wikipedia.org/wiki/Naming_convention_(programming)`

Default Variable Definitions

Command	Description
`pi`	variable defining π
`i` or `j`	imaginary number $i = \sqrt{-1}$

Arrays

Array, Vectors, and Matrices

- An array is a data form that can hold several values, all of one type.
- A vector is a 1-D array: we can define row vectors, column vectors.
- A matrix is a 2-D array.
- Also, we can define N-D array.

The general notation for a vector or matrix is a list of values enclosed in square brackets [] separated by commas (space) or semi-colons (or the combination).

Vectors

- Row vector, e.g., $x = \begin{bmatrix} 1 & 2 & 3 & 4 \end{bmatrix}$
  - `x = [1,2,3,4]`
  - `x = [1 2 3 4]`
- Column vector, e.g., $x = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$ or $x = \begin{bmatrix} 1 & 2 & 3 & 4 \end{bmatrix}^\top$
  - `x = [1;2;3;4]`
  - `x = [1 2 3 4]'` where `'` is the infix notation for transpose operation in MATLAB.

Matrices

- Define a matrix, e.g., $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$

```
A = [1,2;3,4]
```

Generate a Subarray using Slicing

- `a = [1,2,3;4,5,6;7,8,9]`
- `b = a(1,1)  % b = [1]`
- `c = a(:,1)  % c = [1;4;7]` same as `c = a(1:3,1)`
- `d = a(2:end,2:end)  % d = [5,6;8,9]`
  same as `d = a(2:3,2:3)`

Generate a 3-D Array using Slicing

- `A = [1,2;3,4]`
- `B = [5,6;7,8]`
- `C(:,:,1) = A` or `C(:,:,1) = [1,2;3,4]`
- `C(:,:,2) = B` or `C(:,:,2) = [5,6;7,8]`

Concatenate Arrays

- `a = [1,2,3]`
- `b = [4,5,6]`
- `c = [a,b]  % c = [1,2,3,4,5,6]`
- `d = [a;b]  % d = [1,2,3;4,5,6]`
- `e = [d;d]  % e = [1,2,3;4,5,6;1,2,3;4,5,6]`
- `f = [d,d]  % f = [1,2,3,1,2,3;4,5,6,4,5,6]`

String: Array of Characters

- `s = 'abc'`
- `t = ['a' 'b' 'c']`
- `s == t  % return logical 1`
- `[s t]   % return 'abcabc'`
- `[s;t]   % return ['abc';'abc']`

Cell Arrays

- `s1 = {'abc', 'def'}` vs. `t1 = ['abc', 'def']`
- `s2 = {'abc'; 'def'}` vs. `t2 = ['abs'; 'def']`
- `s3 = {'ab', 'cd'; 'ef', 'gh'}`
- `s3{1,1}  % 'ab'`
- `cell(n)`: create 1-D cell array of length `n`
- `cell(m,n)`: create 1-D cell array of size `m` by `n`

Functions for Vectors & Matrices

Command	Description
`linspace`	Linearly spaced vector
`logspace`	Logarithmically spaced vector
`colon` or `:`	Colon
`transpose` or `'`	Non-conjugate transpose of a vector
`eye`	Identity matrix
`ones`	Ones array
`zeros`	Zeros array
`rand`	Uniformly distributed pseudorandom numbers
`randn`	Normally distributed pseudorandom numbers
`magic`	Magic square
`diag`	Diagonal matrices and diagonals of a matrix
`reshape`	Reshape array
`size`	Size of array
`length`	Length of vector

Additional Functions and Commands

Command	Description
`iskeyword`	Check if input is a keyword
`who`	List current variables
`whos`	List current variables, long form
`which`	Locate functions and files
`clear`	Clear variables and functions from memory
`clc`	Clear command window
`clf`	Clear current figure
`close`	Close figure
`exist`	Check existence of variable/script/function/folder/class
`disp`	Display array

Useful Functions for Homework 1

Command	Description
`dot`	Vector dot product
`eig`	Eigenvalues and eigenvectors
`transpose` or `'`	Transpose
`fplot`	Plot 2-D function
`find`	Find indices of nonzero elements
`intmin`	Smallest integer value
`intmax`	Largest positive integer value
`realmin`	Smallest positive normalized floating point number
`realmax`	Largest finite floating point number

Useful MATLAB Shortcuts

- Windows shortcuts
  - Press `Ctrl` + `A` to select all
  - Press `Ctrl` + `I` to adjust indentation
  - Press `Ctrl` + `R` to comment
  - Press `Ctrl` + `T` to uncomment
- macOS shortcuts
  - Press `command` + `A` to select all
  - Press `command` + `I` to adjust indentation
  - Press `command` + `/` to comment
  - Press `command` + `T` to uncomment

LaTeX Primer

table Environment

```
\begin{table}[!hbtp]
  \caption{This is a table}
  \begin{tabular}{rcl}
  \toprule
  Column 1 & Column 2 & Column 3 \\
  \midrule
  1        & 1        & 1        \\
  12       & 12       & 12       \\
  123      & 123      & 123      \\
  \bottomrule
  \end{tabular}
\end{table}
```

table Environment

Table 1:This is a table

Column 1	Column 2	Column 3
1	1	1
12	12	12
123	123	123

figure Environment

```
\begin{figure}[!hbtp]
  \centering
  \includegraphics[height=0.3\textheight]{figure.pdf}
  \caption{Plot of $\sin{x}$}
  \label{fig:sin}
\end{figure}
```

generates

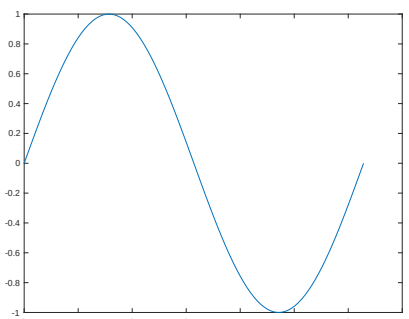

Figure 1:Plot of $\sin x$

\left and \right vs. \big, \Big, \Bigg

```
\begin{align*}
\|x\|_2 & = \big(\sum_{i = 1}^{n} x_i^2 \big)^{1/2},
\|x\|_2 = \Big(\sum_{i = 1}^{n} x_i^2 \Big)^{1/2}, \\
\|x\|_2 & = \Bigg(\sum_{i = 1}^{n} x_i^2 \Bigg)^{1/2},
\|x\|_2 = \left(\sum_{i = 1}^{n} x_i^2 \right)^{1/2}.
\end{align*}
```

generates

$$\|x\|_2 = \big(\sum_{i=1}^{n} x_i^2\big)^{1/2}, \|x\|_2 = \Big(\sum_{i=1}^{n} x_i^2\Big)^{1/2},$$

$$\|x\|_2 = \Bigg(\sum_{i=1}^{n} x_i^2\Bigg)^{1/2}, \|x\|_2 = \left(\sum_{i=1}^{n} x_i^2\right)^{1/2}.$$

Links

```
\href{https://www.google.com}{Google}
```

Google

Or simply

```
\url{https://www.google.com}
```

https://www.google.com

case Environment

```
$$
f(x) =
\begin{cases}
5 x + 4    & \text{if~} x \leq 1, \\
3 x^2 + 6 & \text{if~} x > 1
\end{cases}
$$
```

generates

$$f(x) = \begin{cases} 5x + 4 & \text{if } x \leq 1, \\ 3x^2 + 6 & \text{if } x > 1 \end{cases}$$

Cross-Reference

```
\begin{equation}
\label{eq:ls}
A \mathbf{x} = \mathbf{b}.
\end{equation}

The expression \eqref{eq:ls} is a linear system.
```

generates

$$A\mathbf{x} = \mathbf{b}. \tag{1}$$

The expression (1) is a linear system.

Cross-Reference

```
\begin{table}[!hbtp]
\caption{$y = 2x$}
\label{tab:xy}
  \begin{tabular}{cc}
  \toprule
  $x$ & $y$ \\
  \midrule
  $6$ & $12$ \\
  $7$ & $14$ \\
  $8$ & $16$ \\
  \bottomrule
  \end{tabular}
\end{table}
Table \ref{tab:xy} gives the result of $y = 2x$.
```

Cross-Reference

Table 2:$y = 2x$

x	y
6	12
7	14
8	16

Table 2 gives the result of $y = 2x$.