

Matlab Primer

Lecture 02a

Optical Sciences 330

Physical Optics II

William J. Dallas

January 12, 2005



Introduction

- The title MATLAB stands for Matrix Laboratory. This software package (from The Math Works, Inc.) is designed specifically to perform operations on matrices. The package consists of a command interpreter, a command language and a collection of routines which can be exercised by the interpreter. The commands can either be entered line-by-line in the command window or they can be collected in a text file to be executed by a single line entered in the command window. All computations internal to MATLAB are done in double precision.



Inline Help

To access the inline help, type "help" in the command window followed by the term in question. For example, if you wish to find out the syntax of the IF statement, you can type

```
help if
```

on the command line. A brief explanation will then appear in the command window.



HTML Help

- To access the html help, select Help on the menu-bar of the main Matlab window, then select "MATLAB help". Alternatively, you can press the "F1" function key on your keyboard.



Scripts or M-files

- MATLAB programs can be typed in line-by-line at the command prompt.
- M-file: type the commands line-by-line into a text file, using for instance the Matlab editor, saving to a script file. At the command prompt, type the name of the file or press the run button on the Matlab editor toolbar.
- A comment can be inserted by beginning a line with the percent (%) character.



Search Path

- Matlab will find M-files only if they reside on the current directory, or are listed in your search path. You can examine and modify your search path by selecting “File” on the main Matlab menu bar, then selecting “Set Path”.



Variables

- All variables are matrices. Scalars are 1×1 matrices; vectors are $1 \times N$ or $N \times 1$ matrices. The matrices may be of type real, complex, or string. Note that variable names are case sensitive.
- There are two special matrices we will use often, the matrix of dimensions $M \times N$ that is filled with zeroes or ones:

`zeros(M,N)` filled with zeroes

`ones(M,N)` filled with ones



Operators I

- **Assignment**

= Assign the value on the right to the variable on the left.

- **Semicolon**

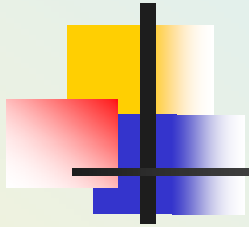
The semicolon is very important, though not strictly an operator. It suppresses the echo of entries to the command screen. For example, suppose you assign a large matrix to second, equally large, matrix. Unless you enter a semicolon on the line after the assignment, all of all of matrix values will be written to the screen. Should this happen, you can interrupt the operation by hitting ***CNTRL_C***.



Operators II

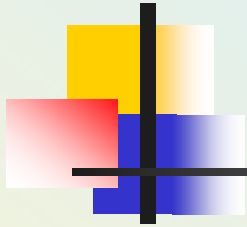
- The arithmetic binary operators, operators that combine two variables, exist in MATLAB as two families: the matrix operators and the array operators.
- Arithmetic Operators of Identical Matrix and Array Form

+	addition
-	subtraction



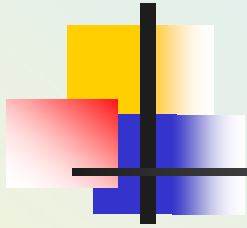
Arithmetic Matrix-Operators

- * multiplication
- / right division
- \ left division
- ^ raising to a power
- ' transpose with conjugate



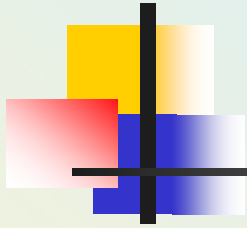
Arithmetic Array-Operators

- `.*` multiplication
- `./` right division
- `.\` left division
- `.^` raising to a power
- `.'` transpose with no conjugate



Relational Operators

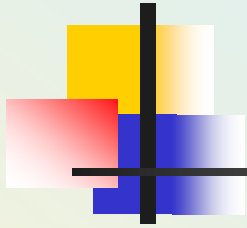
- $<$ less than
- $>$ greater than
- $<=$ less than or equal to
- $>=$ greater than or equal to
- $==$ equal to
- \neq not equal to



Flow-Control Structures I

Loop

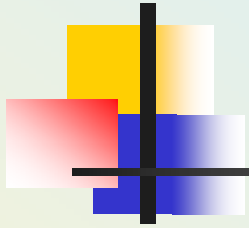
- While: Repeats the group of statements within the command's scope while the condition contained in the control variable is true.
- For: Repeats the group of statements within the command's scope for a specific number of times
- End: Delimits the scope of a "FOR" or "WHILE" statement
- Break: Terminates the execution of a loop.



Flow-Control Structures II

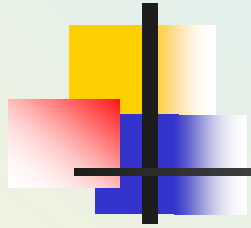
Branch

- If: Conditionally executes a group of statements within the command's scope depending on the logical value of the control expression.
- Elseif: Specifies an alternative action when the "IF" control variable is false and the "ELSEIF" control variable is true.
- Else: Specifies an alternative action when the "IF" control variable is false and (if present) the "ELSEIF" control variable is false.
- End: Delimits the scope of an "IF" statement.
- Return: Return control to the invoking function.



Input/Output Commands

- Interactive Input
- Graphical Output



Interactive Input

- **Input:** Displays the prompt in the text string on the command screen and then waits for input from the keyboard.
- **Menu:** Generates a menu of choices for user input.
- **Pause:** Stops execution and waits for a user response, or the elapsing of a timer.
- **Keyboard:** Invoke the keyboard as if it were a Script-file.



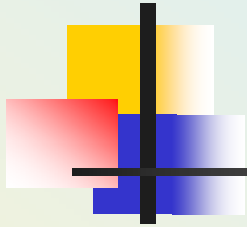
Disk Commands I

- **Directory Commands**

The directory commands available in MATLAB are identical to those in DOS. For instance, the command "dir" lists the files present on your current directory. The command "c:" will change your present directory to the C disk. The command "cd" will change your directory.

- **Save**

The form most useful to us will be "save file_name variable_name". This command will save "variable_name" to disk in the file "file_name" with the extension "mat" located on your current directory.



Disk Commands II

- **Load**

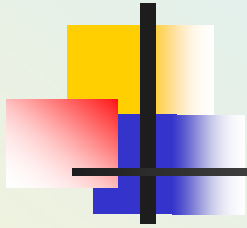
The "load" command retrieves variables from disk. The form most useful is "load file_name". The expression "file_name" is the name of a file with the extension "mat".

- **Imwrite**

Write an image in a given format. The formats can, among others, be bmp, jpg, tif, pcx, pgm, png, or ppm.

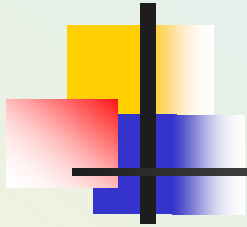
- **Imread**

Read an image in a given format. See Imwrite for some of the formats.



Maxima and Minima

- Finding extrema will be of immediate importance to us in normalizing matrices for display. These functions determine the vector \mathbf{y} in a matrix \mathbf{x} , or scalar y in a vector \mathbf{x} containing the extremum:
 - $y = \max(\mathbf{x})$; maximum over the columns
 - $y = \min(\mathbf{x})$; minimum over the columns
- More usually we wish to have the scalar y containing the extremum in a matrix \mathbf{x} :
 - $y = \max(\max(\mathbf{x}))$; maximum over the entire matrix
 - $y = \min(\min(\mathbf{x}))$; minimum over the entire matrix



Fourier Transforms

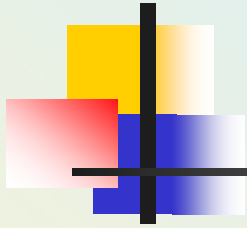
- There are three steps in doing a discrete Fourier transform or DFT:

quadrant-swap, FFT, quadrant-swap

The quadrant swapping routine is the same for all dimensions, but the FFT routine differs for one and two dimensions. We assume an array \mathbf{x} of the appropriate dimensionality.

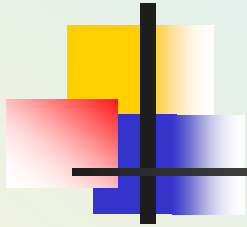
- Complete 2-D FFT: $y = \text{fftshift}(\text{fft2}(\text{fftshift}(x)));$

- Complete 2-D inv. FFT: $y = \text{fftshift}(\text{ifft2}(\text{fftshift}(x)));$



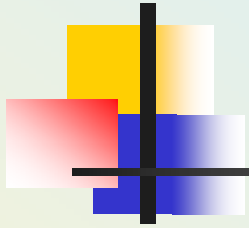
Displaying an Image I

- To open figure window number one, or to select it if it is already open
 - `figure (1);`
- For monochrome images you set a graytone color map from 0-63 with
 - `colormap(gray);`
- Alternate 256-level colormap:
 - Generate:
`Ramp = (0:255)/255.0; Gray256 = zeros(256,3);`
`Gray256(:,1)=Ramp; Gray256(:,2)=Ramp; Gray256(:,3)=Ramp;`
 - Load: `colormap(Gray256);`



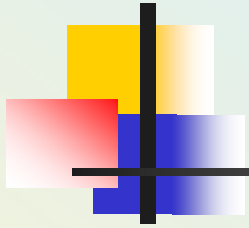
Displaying an Image II

- If the data you wish to display are contained in the array "MyImage" and the values have been normalized to the depth of your color table you may use
 - `image (x);`
- If the data you wish to display are contained in the variable " MyImage " and wish to let Matlab take care of the gray-scaling
 - `imagesc (MyImage);`
- To close figure 1
 - `close (1);`



Manipulating the Workspace

- To show the variables and their storage-space allocations in the workspace
 - whos
- To clear the workspace
 - clear



Examples in the Lecture Notes

- Matrix filling
- Control-flow structure



Always

- Comment your code
- Name yourself as author, give credit if the code is derived from another source
- Put the date latest revision in your code