MATLAB TUTORIAL

CIS 581
ANVITH EKKATI
Variables and Workspace

- **Scripts vs Command Window**
- \( a = <\text{value}> \) — defines a variable
- `clear` — clears all variables in the workspace
- `clear <\text{var name}>` — clears only the specified variable
- Use the command window to quickly run code, and check variables in the workspace — useful for debugging — `clc` to clear the command window text
- Note: `;` suppresses output for a line
- Comments start with a `'#'`. Use `'##'` to denote a region. `'#{'` to start and `'}'` to end multiline comments.
Arrays and Matrices

- \(a = [10 \ 20 \ 30 \ 40 \ 50];\)
- \(b = [10 : 10 : 50]; b = [50 : -10 : 10];\)
- \(c = \text{zeros}(5);\)
- \(d = \text{ones}(5);\)
- \(e = \text{eye}(5);\)
- \(f = \text{rand}(5, 3);\)
- Matrix addition (+), multiplication (*)
- Point-wise multiplication
- ‘:’ operator for indexing matrices, and flattening — column major
- uint8([1.1 2.8]) — Convert to integer
- Array indexing, slicing
For Loops

```matlab
for i = 1:10
    i
end

for i = 1:2:10
    i
end

for i = [1 2 7]
    i
end

Performance is improved if arrays are pre-allocated before a for loop

for i = 1:10
    a(i) = i;
end

a = zeros(10, 1)
for i = 1:10
    a(i) = i;
end

a
```
If Conditions

if A==B
    'A is equal to B'
elseif A>B
    'A is greater than B'
else
    'A is neither equal to B, nor greater than B'
end
Logical Operators

- Can be used to compare corresponding elements in arrays: \( A > B \)
- Can be used to compare every element in an array with a number: \( A == 0; B ~= 0; \) \( \% \) \( ~ \) is the not operator
- AND: \( A \& B \)
- OR: \( A | B; \)
Save and Load — mat files

- `save('<filename>.mat')` — Save entire workspace
- `save('<filename>.mat', 'var1', 'var2', ...)` — Save only certain variables
- `save('<filename>.mat')` — Load all variables in mat file
- `load('<filename>.mat', 'var1', 'var2', ...)` — Load only required variables from mat file
- `load` will overwrite any variables with the same name which are already in the workspace
Plotting

- imread
- imshow
- imagesc
- figure
- rgb2gray
- hold
- quiver
Function Handles

redLine = @(x, y) plot(x, y, '-r', 'LineWidth', 5);
redLine([size(vertical_sobel, 2), 0],[0, size(vertical_sobel, 1)]);
Functions

Functions operate in their own workspace

```matlab
function [ A, B ] = custom_func( C, D )
%CUSTOM_FUNC Summary of this function goes here
% Detailed explanation goes here

    A = C + D;
    B = custom_func_local(C, D);

end
```

Local functions vs Global Functions
Debugging

dbstop if error
Important Functions

- interp2
- conv2
- norm
- size
- vertcat/horzcat
- find
- reshape
- permute
- inv
- transpose, Ex: A’
- Matlab has a lot of common array operations as functions
- help
bsxfun

```
A = [1 2 10; 3 4 20; 9 6 15];
C = bsxfun(@minus, A, mean(A));
```
MATLAB OnRamp