Introduction to MATLAB: Elementary Matrix Operations and Image Processing

Note:

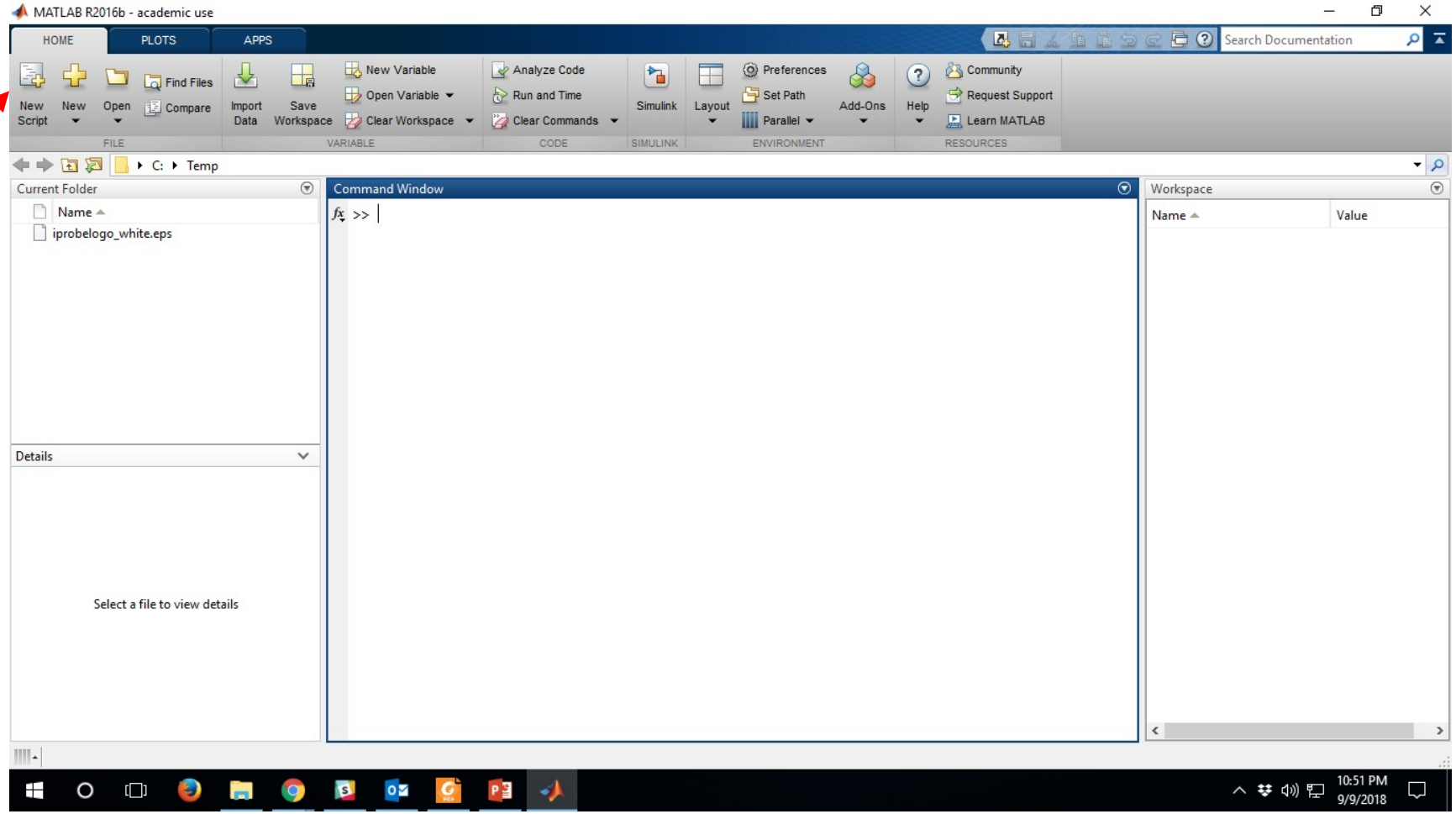
All EGR students can use Matlab in the RDS environment.

Instructions for connecting can be found here: https://www.egr.msu.edu/decs/help-support/how-to/connect-decs-remote-desktop-services-rds-servers

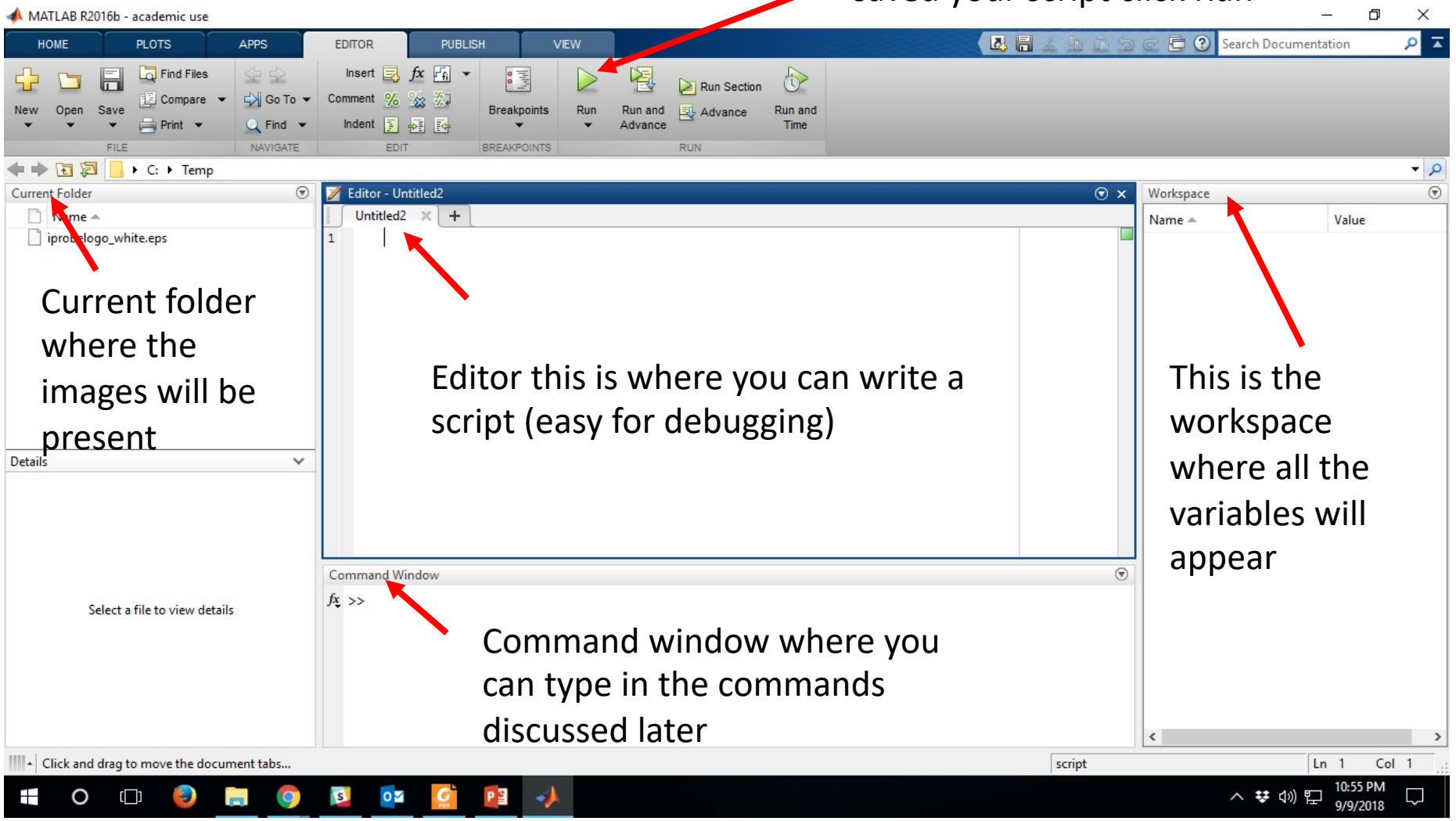
MATLAB

- Open MATLAB from the Start menu. Click on ‘New Script’

Click on
New Script



MATLAB Windows



Elementary Concepts about Vectors and Matrices

What is an array?

An array consists of numbers arranged in rows or columns or both. A 1-dimensional array is also called a vector.

Column Vector ➡ $\begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$ Row Vector ➡ $\begin{bmatrix} 1 & 2 & 3 & 4 \end{bmatrix}$

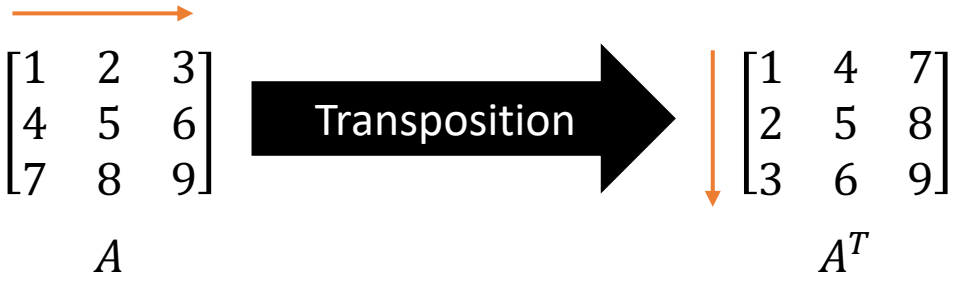
What is a matrix?

A matrix is a multidimensional array. It can have 2/3/more dimensions

2 x 2 Matrix ➡ $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ 2 x 3 Matrix ➡ $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$

Matrix Operations

Transposition



$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \quad A \qquad \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix} \quad A^T$$

Inverse

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad B \qquad \begin{bmatrix} -2 & 1 \\ 1.5 & -0.5 \end{bmatrix} \quad B^{-1}$$

$$B^{-1} = \frac{1}{\det(B)} \text{adj}(B)$$

$$\det(B) = |B| = (1\times4) - (2\times3) = -2 \qquad \text{adj}(B) = \begin{bmatrix} 4 & -2 \\ -3 & 1 \end{bmatrix}$$

Note: For the inverse to exist the matrix must be square and its determinant value ≠0

Some More Matrix Operations

Addition

$$\underset{A}{\begin{bmatrix}1 & -1\\ 8 & 5\end{bmatrix}} + \underset{B}{\begin{bmatrix}7 & 2\\ 1 & 0\end{bmatrix}} = \underset{C}{\begin{bmatrix}8 & 1\\ 9 & 5\end{bmatrix}}$$

Scalar Multiplication

$$\underset{A}{\begin{bmatrix}1 & 2 & 3\\ 4 & 5 & 6\\ 7 & 8 & 9\end{bmatrix}} \qquad \underset{2A}{\begin{bmatrix}2 & 4 & 6\\ 8 & 10 & 12\\ 14 & 16 & 18\end{bmatrix}}$$

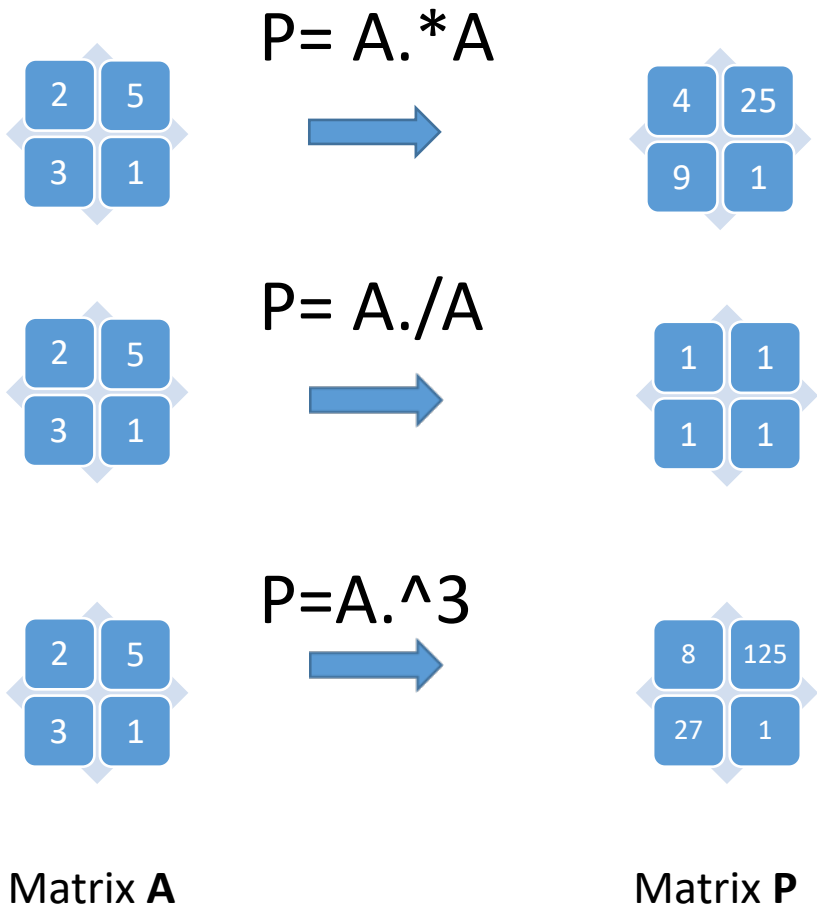
Matrix Multiplication

$$\begin{bmatrix}a & b\\ c & d\end{bmatrix} * \begin{bmatrix}e & f\\ g & h\end{bmatrix} = \begin{bmatrix}(ae+bg) & (af+bh)\\ (ce+dg) & (cf+dh)\end{bmatrix}$$

If size of matrix A is m x n and size of matrix B is n x p,
then their product will result in a matrix of size m x p

Note: In general, matrix multiplication is not commutative, ie., $A * B \neq B * A$

Performing element wise operations

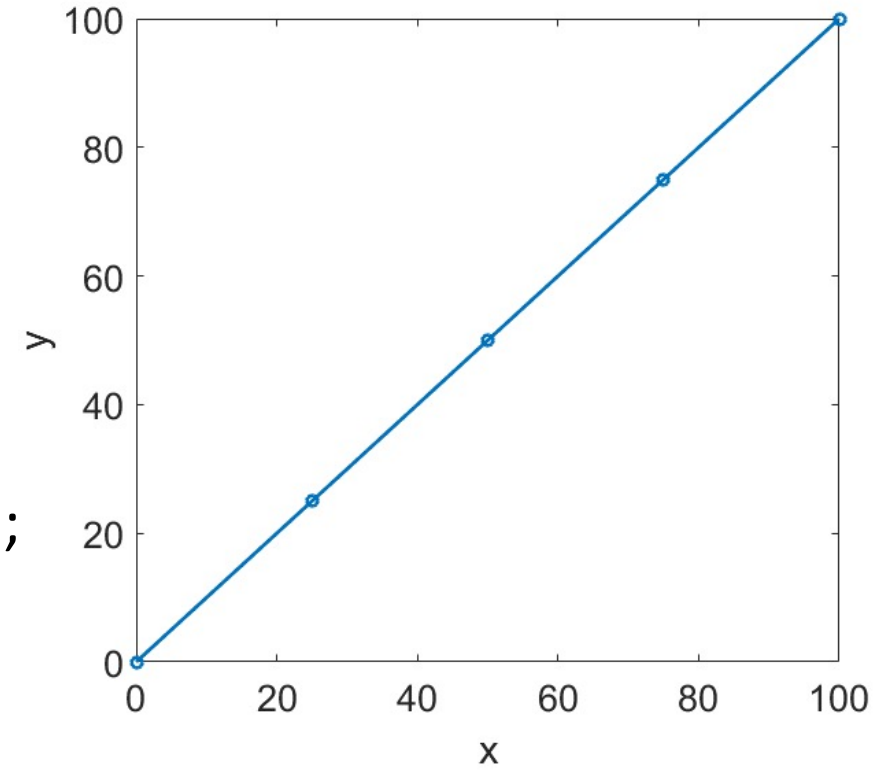


Here the dot operator (.) denotes element-wise operations

Plots

- To plot the graph of a function, define a function $y = f(x)$
- Specify the **range of values** for the variable x, for which the function is to be plotted
- Call the **plot** command, as **plot(x, y)**

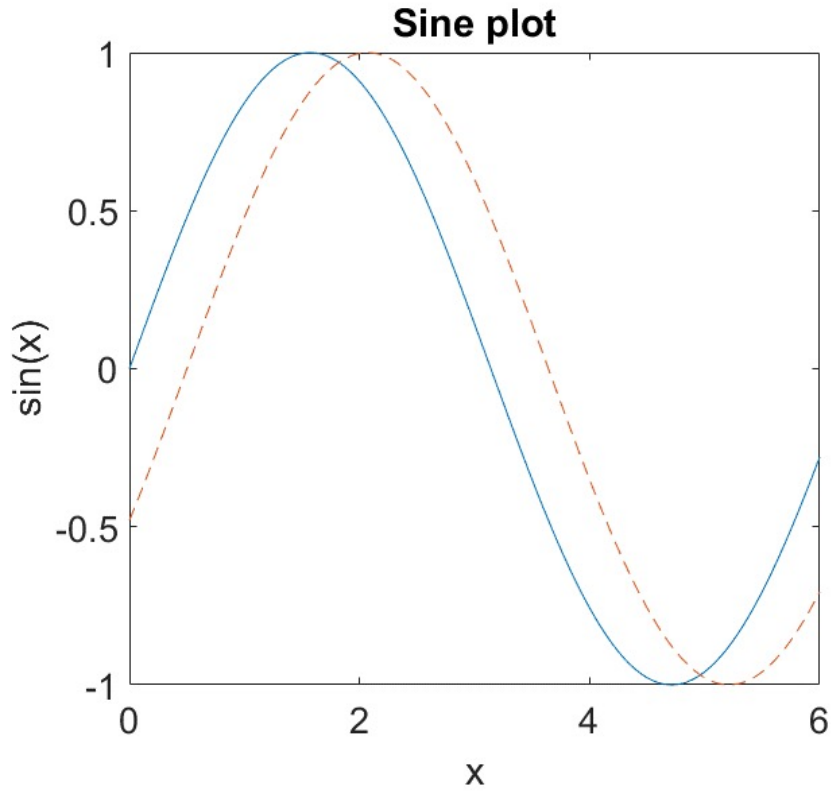
```
x= [0:5:100]; % initialvalue : increment: finalvalue
y=x;
plot(x,y,'-o','MarkerIndices',1:5:length(y),'LineWidth',2);
xlabel('x')
ylabel('y')
```



Plot Display Options

You can plot multiple graphs in the same figure and use different colors to distinguish between them. You can also add title to your plot

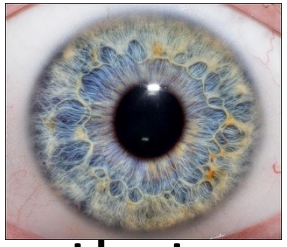
```
x = 0:pi/100:2*pi;
y1 = sin(x);
y2 = sin(x-0.5);

figure
plot(x,y1,x,y2,'--')
title('Sine plot')
xlabel('x')
ylabel('sin(x)')
```



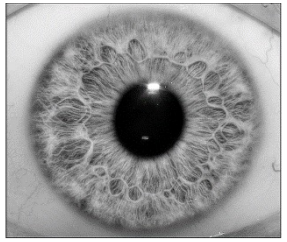
What is an Image?

- An image is made of "pixels“. A pixel is the basic unit in a digital image

- To represent color images: red, green and blue components must be specified separately for each pixel. Thus, the pixel ‘value’ is a vector of three numbers $[R \;\; G \;\; B]$

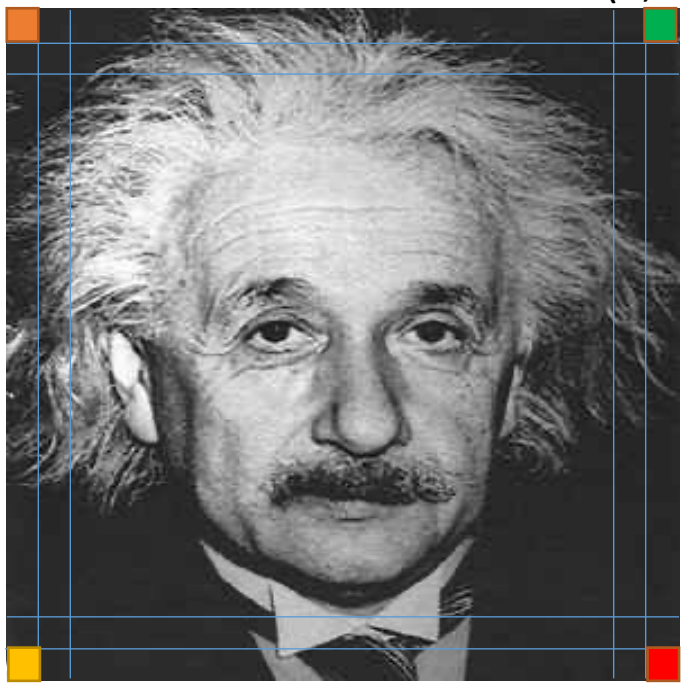


- For a grayscale image, the pixel value is a single number that represents the pixel intensity



Note: You can use MATLAB inbuilt function rgb2gray() to convert a color image to a grayscale image

Image Processing



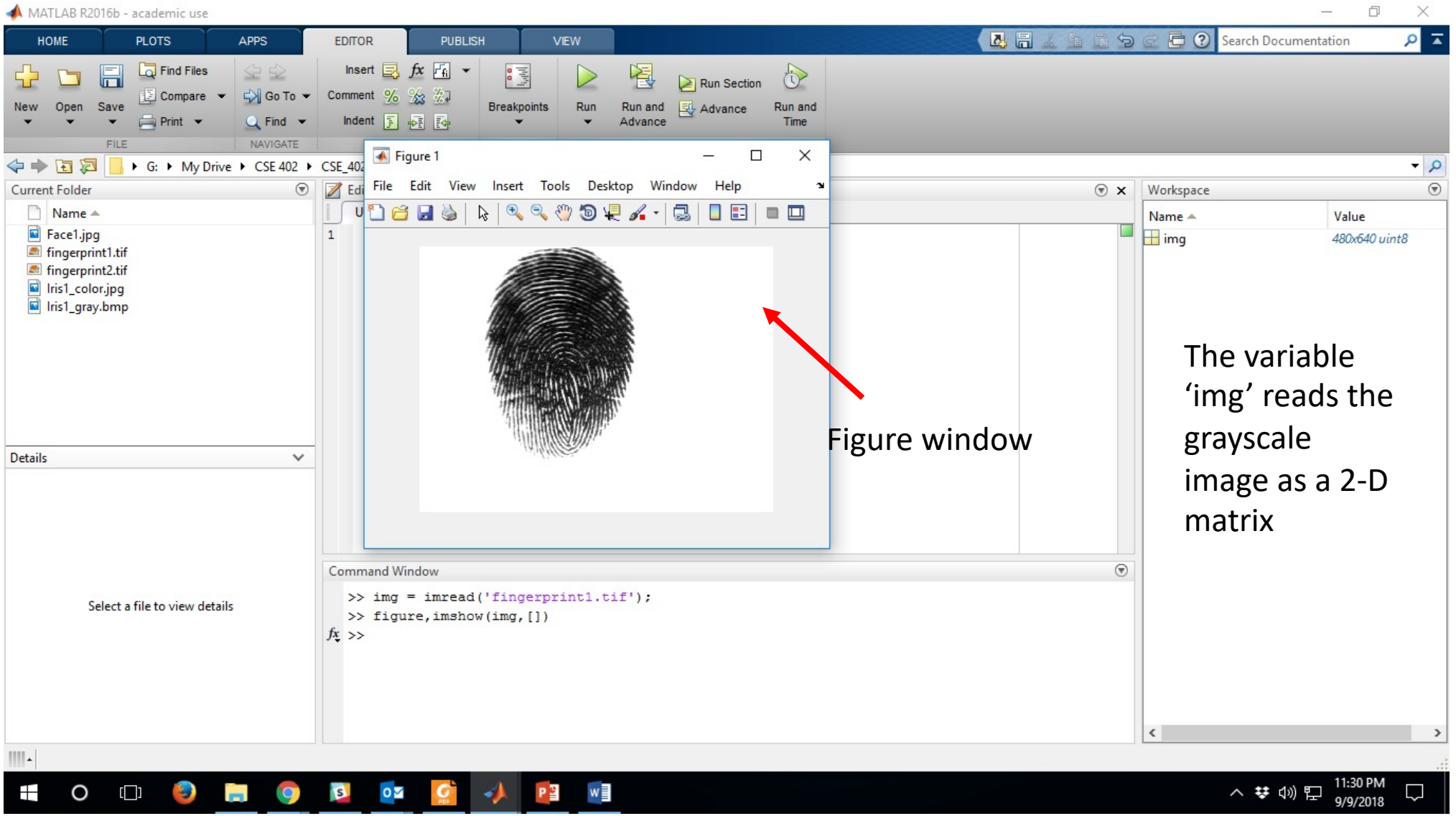
327 x 300 Matrix

$$\begin{bmatrix} 46 & \cdots & 59 \\ \vdots & \ddots & \vdots \\ 239 & \cdots & 41 \end{bmatrix}$$

Width: 300, Height 327

A grayscale image (left) and its corresponding 2-D matrix representation (right). Each element in the matrix represents a pixel intensity value at that pixel location

Reading an image in MATLAB



Digital Image Processing

Digital image processing performs operations to analyze and process a digital image to extract useful information from it

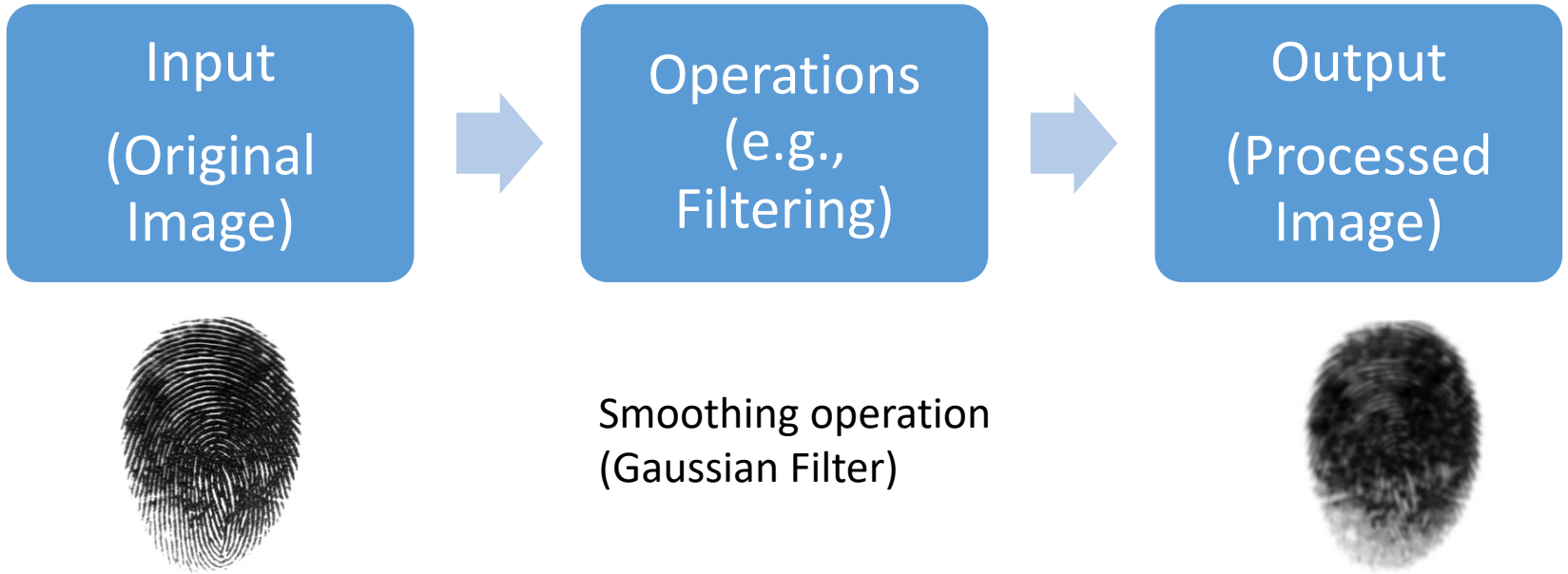


Image Operations

- img = imread('pout.tif') → reads the image into the matrix
- imshow(img) → displays the image
- size(img) → returns the size of the image
- imwrite(img, ‘fig1.png’) → writes the matrix as image to graphics file
- imfinfo(‘fig1.png’) → returns the information about the image in the file
- imshowpair(img1,img2,’montage’) → displays images side by side
- imgC = imfuse(imgA,imgB) →creates a composite image from two images, imgA and imgB

Image Filtering

- imfilter N-D filtering of multidimensional images
- fspecial Create predefined 2-D filter
- h = fspecial('average',hsize)
- h = fspecial('disk',radius)
- h = fspecial('laplacian',alpha)
- h = fspecial('motion',len,theta)
- h = fspecial('sobel')
- h = fspecial('gaussian',hsize,sigma)
- h = fspecial('log',hsize,sigma)
- h = fspecial('prewitt')

Note: A filter is a matrix and image filtering involves convolution operation

Creating a Filter and Applying to an Image

originalimg = imread(‘fingerprint1.tif');
imshow(originalimg)

h = fspecial(‘log‘,[3,3],0.4);
filteredimg = imfilter(originalimg, h,’replicate’);
figure, imshow(filteredimg)

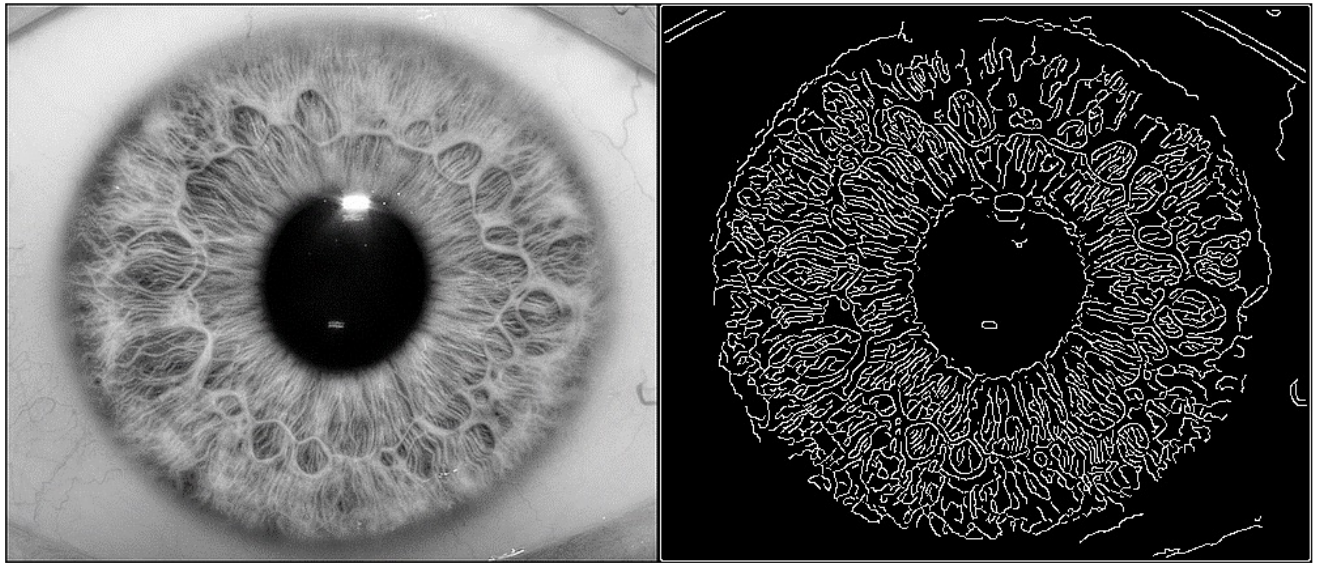
Edge detection in images

- Edge detection is an image processing technique used for finding object boundaries within images. It detects discontinuities in the intensities

- Common edge detection operators include
  - Sobel
  - Canny
  - Prewitt
  - Roberts

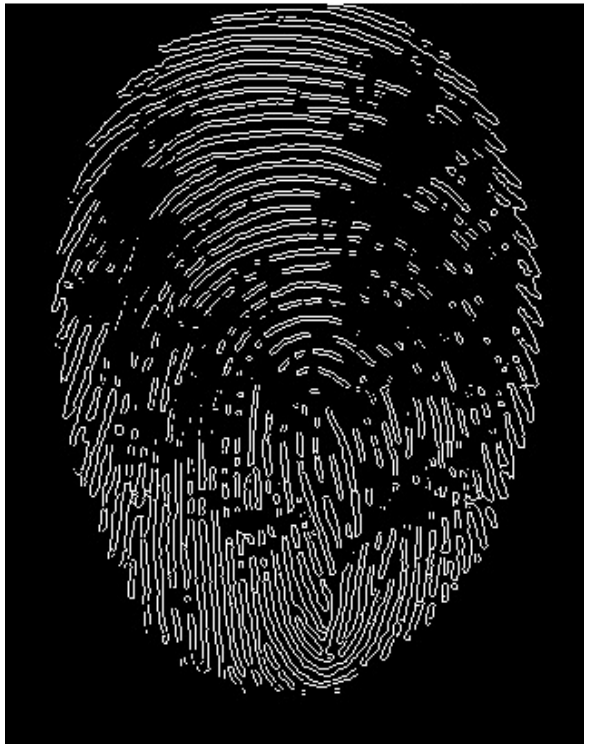
Comparison of edge detection using Canny & Prewitt

```
% Read the image into matrix

img = imread('iris_gray.bmp');

% Apply Canny edge operator

edge_img = edge(img,'Canny');

% Create a montage of the original
and the edge images

imshowpair(img,edge_img,'montage')
```

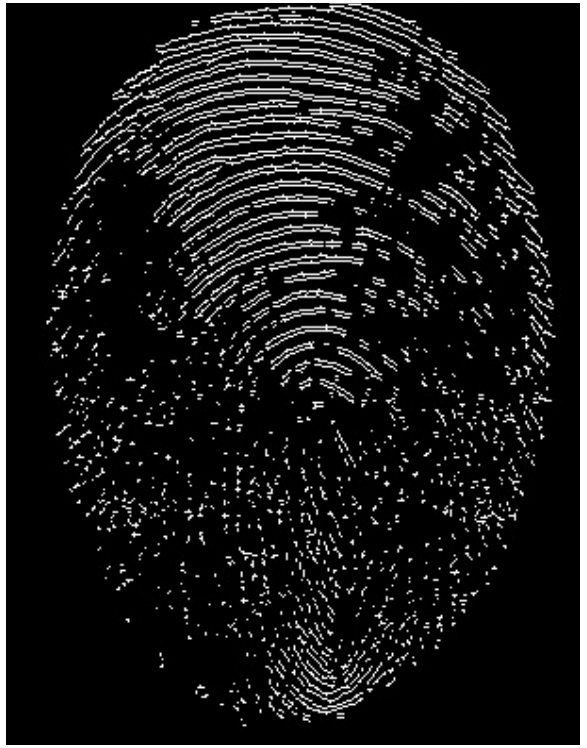


Direction of Edges

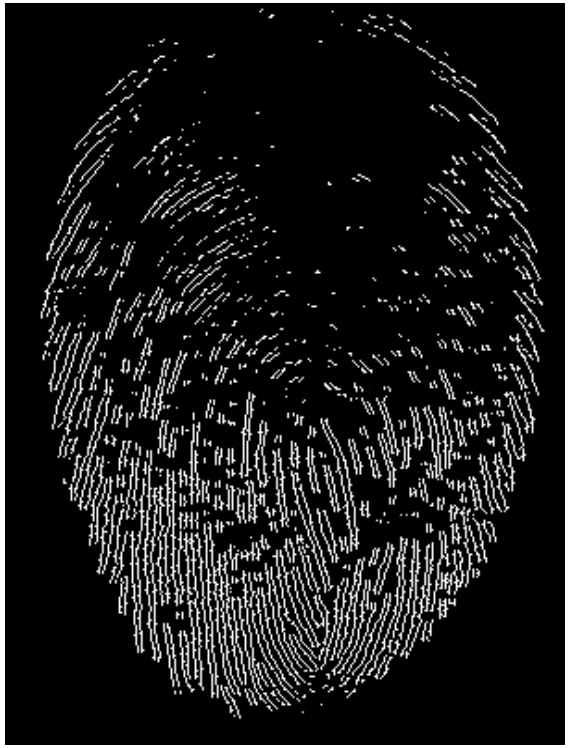
```
im=imread('fingerprint1.tif');
```



```
im_edge=edge(im,'Prewitt');
```



```
im_hori=edge(im,'Prewitt',
'horizontal');
```



```
im_vert=edge(im,'Prewitt',
'vertical');
```

MATLAB Exercise

The document describes the commands to be used for performing matrix operations and image processing operations

Some exercise problems need to be completed and the solutions need to be uploaded