Introduction to MATLAB

Xiaoshuang Li

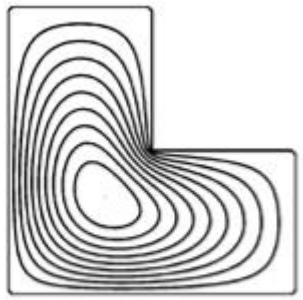
Outline

- A brief history of MATLAB
- Why MATLAB ?
- Installation
- Learn & Try
- Operations

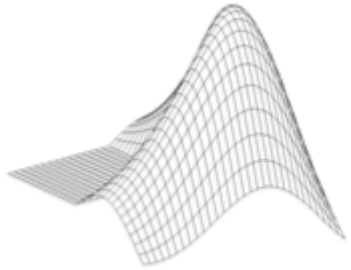
A brief history of MATLAB

- Matrix Laboratory

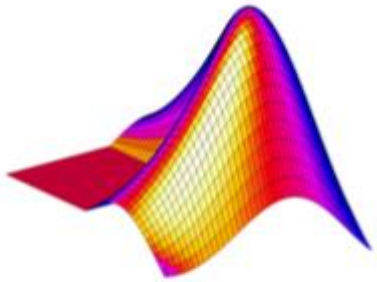
A proprietary multi-paradigm programming language and numerical computing environment developed by MathWorks.



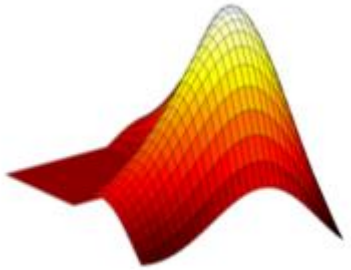
1984



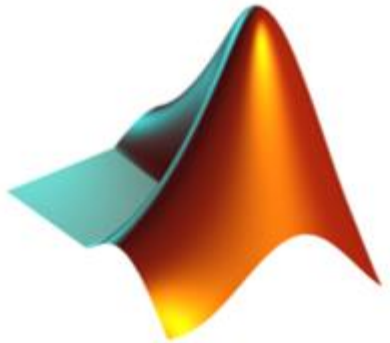
1987



1992



1994



1997

A brief history of MATLAB

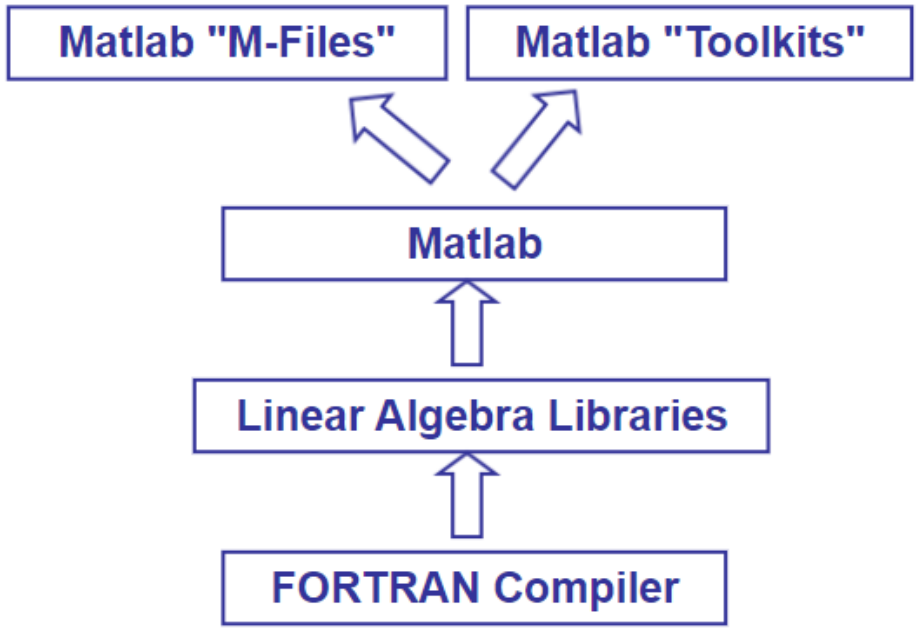
- Matrix Laboratory
  - Example: solve a linear system A*x=b, solve for x.

```
x = A \ b
```

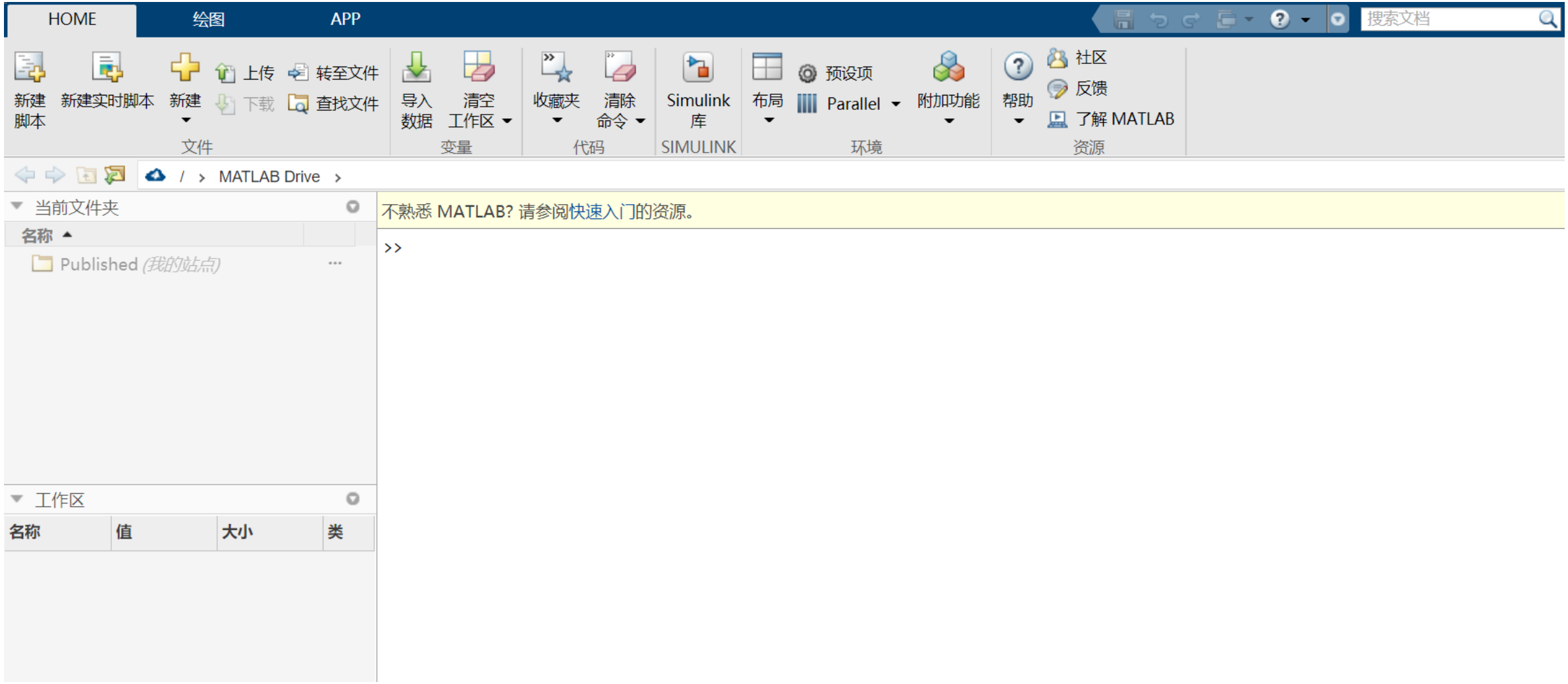
```
C.... factor the A matrix
      CALL SGEFA(A, N, N, IPVT, INFO)
C.... copy B vector into X vector
      CALL SCOPY(N, B, 1, X, 1)
C.... solve the system of equations
      CALL SGESL(A, N, N, IPVT, X, 0)
```

```
C Solve B = A*X for X.
C N is dimension of vectors and matrix
C Does not use row interchange, scaling.
     SUBROUTINE LINSYS(N, A, X, B, TMP)
     INTEGER N
     DOUBLE PRECISION A(N,N), X(N), B(N)
     DOUBLE PRECISION TMP(N), RATIO
C... Forward elimination
     DO 13 J=1,N-1
       DO 12 I=J+1,N
         RATIO = -A(I,J)/A(J,J)
         A(I,*) = A(I,*) +RATIO*ROW(J,*)
         DO 11 K=J+1,N
  11     A(I,K) = A(I,K) + RATIO*A(J,K)
         A(I,J) = 0.0
         X(I) = X(I) + RATIO*X(J)
  12   CONTINUE
  11 CONTINUE
          Continued...
```

```
C... Backwards substitution
     X(N) = X(N)/A(N,N)
     DO 21 I=N-1,1,-1
       TMP = X(I)
       DO 20 J=I+1,N
  20   TMP = TMP - A(I,J)*X(J)
       X(I) = TMP/A(I,I)
  21 CONTINUE
     RETURN
     END
```



A brief history of MATLAB



Why MATLAB ?

- MATLAB vs. Python
  - Computing Environment + MATLAB language vs. A general-purpose programming language
  - Toolboxes vs. Packages
  - Everything about MATLAB is designed specifically for engineers and scientists.

    Function names and signatures are familiar and memorable.
    The matrix-based MATLAB language lets you express math directly.
    Documentation is written for engineers and scientists, not computer scientists.
  - Paid vs. Free

Installation of MATLAB

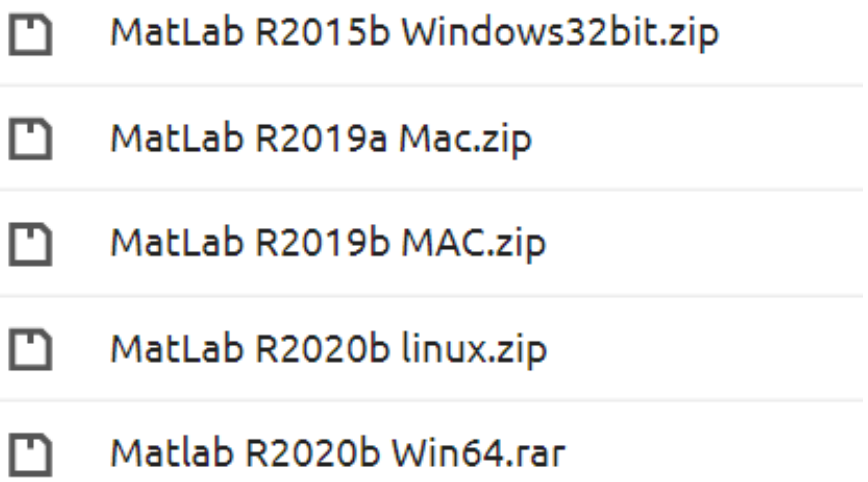
- Online

Compile online

- Installation

MATLAB software in library (Campus only)

  - Register a account
  - Download



Installation of MATLAB

- Installation
  - Install

    Installation Guide (Campus only）

    Authorization for Linux and Mac (Campus only）

    You should register an account with XXX@shanghaitech.edu.cn.
    You need to manually enter the key

MATLAB Activation Key：17005-33008-37294-86396-93404

Installation of MATLAB

- Install

Toolboxes to be installed

MATLAB Audio Toolbox
Communications Toolbox
Control System Toolbox
DSP System Toolbox
Image Processing Toolbox
Signal Processing Toolbox
Statistics and Machine Learning Toolbox
Symbolic Math Toolbox

Learn & Try

- Simple Operation Learning

MATLAB-onramp course

Learning Resources(Campus only）

Guide

Signal Processing

Example. mlx

Exercises_1 .mlx

Operations

- Variables
- Variable names must begin with a letter.
- Names can include any combinations of letters, numbers and underscores.
- Maximum length for a variable name is 63 characters.
- MATLAB is case sensitive. That is the variable name *A* is different than the variable name *a*.
- Avoid the following names: pi (π), and all built-in MATLAB function names such as length, char, size, plot, break, cos, log, …
- Use 1i or 1j instead of single i or j to represent $\sqrt{-1}$.
- It is good programming practice to name your variables to reflect their function in a program rather than using generic x, y, z variables.

Operations

- Array vs. Matrix Operations
  - Execute element by element

  +, - ,.*, ./, .\

  Consider two matrix

$$\mathrm{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \qquad \mathrm{B} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

$$A.* B = \begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} \\ a_{21}b_{21} & a_{22}b_{22} \end{bmatrix}$$

Operations

- Array vs. Matrix Operations
  - Follow the rules of linear algebra

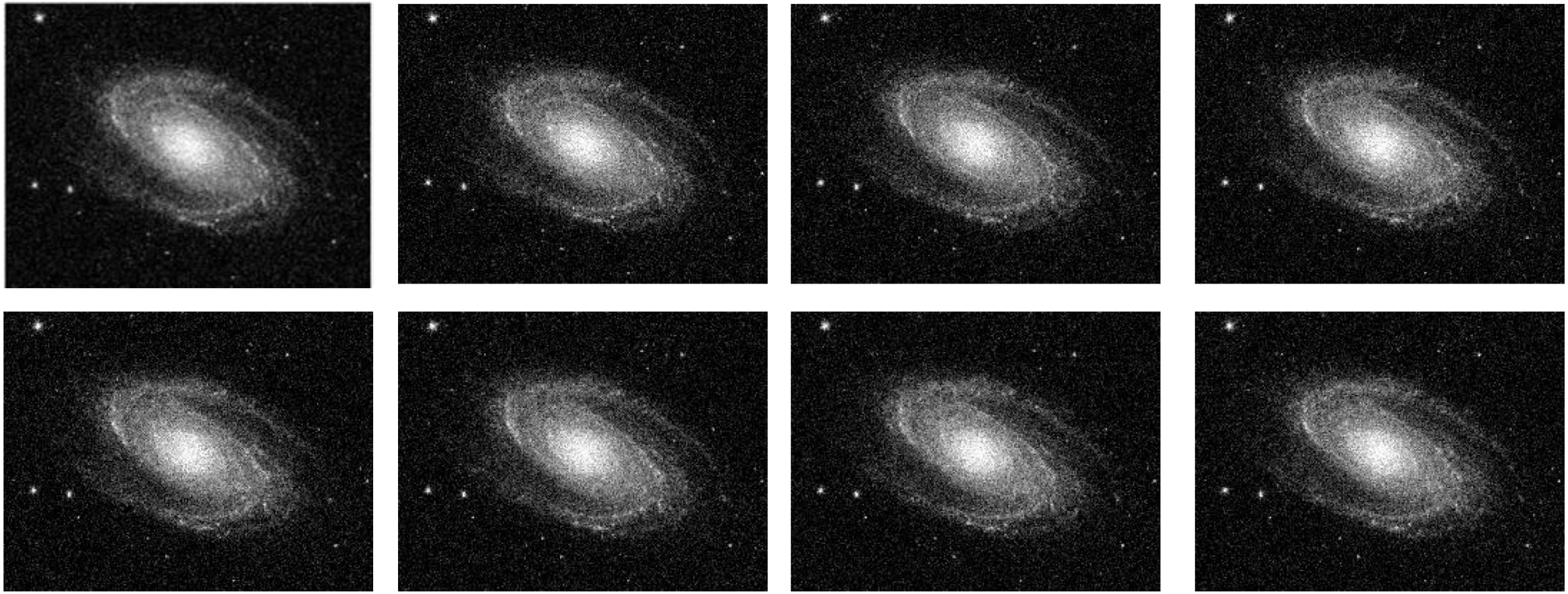
*, /, \, ^, ‘

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \end{bmatrix}$$

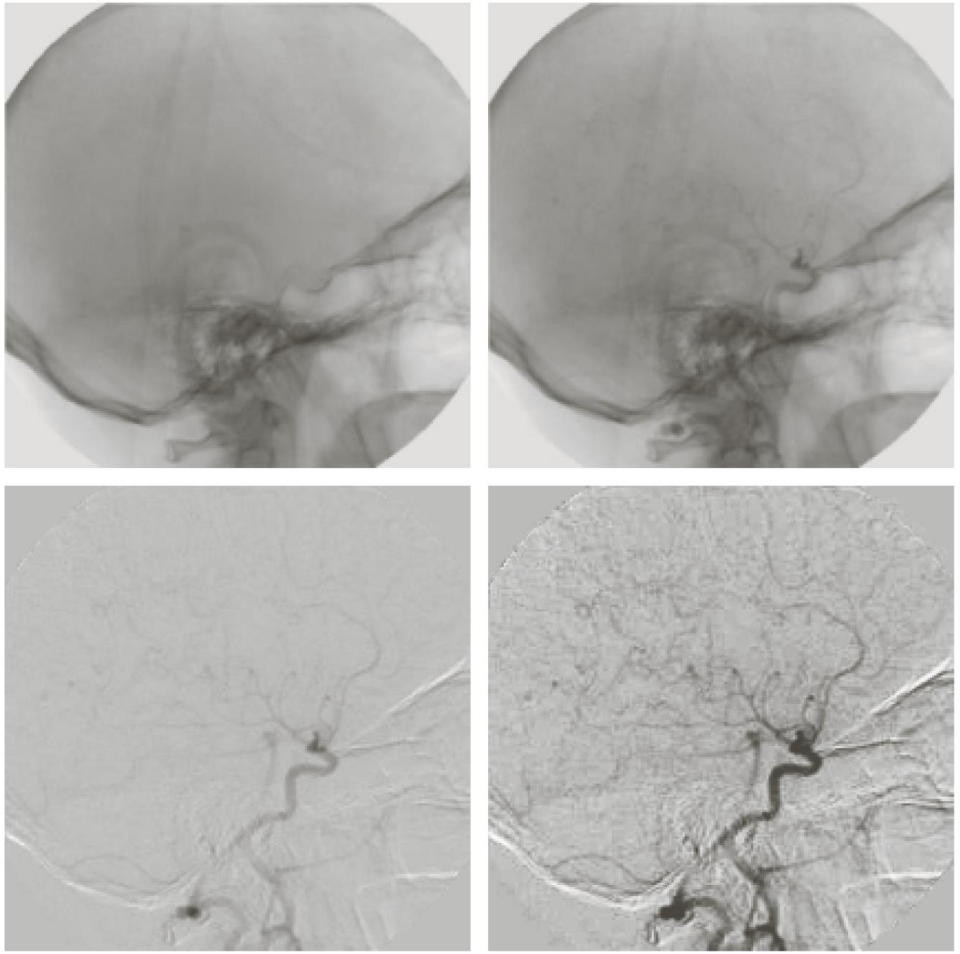
Operations

Task 1: Add eight images together.

$$s(x,y) = f(x,y) + g(x,y)$$



Operations



$$d(x,y) = f(x,y) - g(x,y)$$

Operations

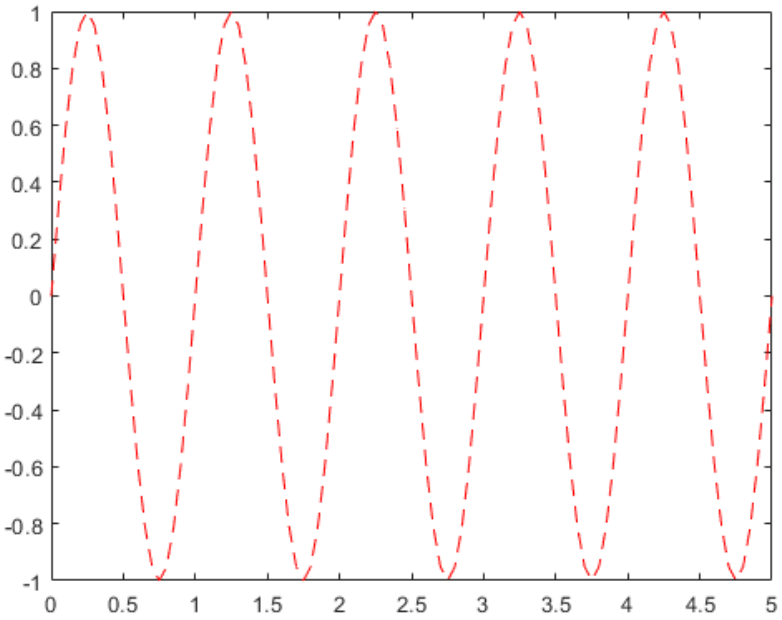
$$p(x, y) = f(x, y) \times g(x, y)$$

Operations

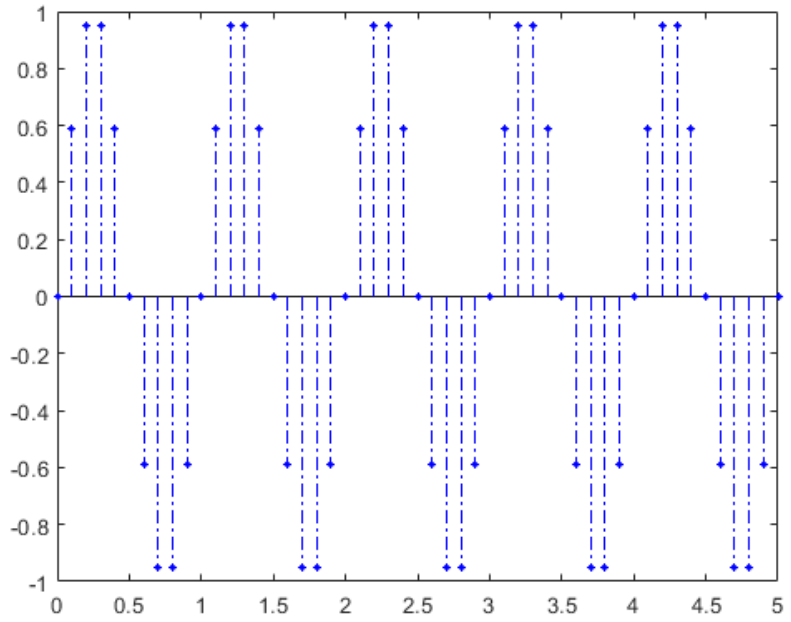
- Something for Signal Processing
- Signals
  - sin
  - cos
  - exp
  - ones
  - zeros
- Interactive
  - input
  - disp
- Symbol
  - ;
  - :
  - ,
- Others
  - clear
  - clf
  - help

Operations

- Draw a picture
  - plot



  - stem



xlabel, ylabel, title, axis, hold on/off, subplot

Operations

- Structure
  - Sequential

```
a = 5;
b = 10;
c = a+b;
```

- Loop

```
for index = values
    statements
end
```

```
while expression
    statements
end
```

- Branch

```
if expression-1
    statements-1
elseif expression-2
    statement-2
...
else
    statement-n
end
```

```
switch expression
    case value-1
        statement-1
    case value-2
        statement-2
    ...
    otherwise
        statement-n
end
```

Homework

- Generate pdf file and upload it to gradescope.
- Need to display both code and results.
- Check if the code is complete.

Encourage discussion in the office hour, if there are code and theoretical questions.

Everyone will have different problems during installation, you should learn to use the Internet and cooperate with each other.

Exercises_1 .mlx

Submission：Before Oct. 23, 0:00