# Mathematical Modeling and Data Analysis

## Lecture 0 Introduction to Programming Languages: Matlab, Python and Julia

> *The only way to learn programming is to do programming.*

### 0.1 Matlab

- 界面布局
- 变量

```
% 数
a = 1
b = 2
% 矩阵
b = [1,2]

c = [1,2;
     3,4]
% 矩阵“组合拼接”

bb = [b,b]

b2b = [b;b]

% 逻辑值

0 < 1 # 值是1

0 > 1 # 值是0

% 一些定义方式
t = 1:3:7 % 1~7，间隔step为3
t = linspace(0,10,11) % 0~10，分11个点
```

```
t = [0.1,0.2,0.3]
```

- 基本计算

```
a = a*2
a = a/2
a = a + 1
a = a - 1
```

- 矩阵“组合拼接”

```
a = 'Hello'
b = 'word'
c = 'a'

ab = [a b]

a_b = [a;c b]
```

- 函数文件

```
% circle.m文件
function [s,p] = circle(r)
s = pi*r^2;
p = 2*pi*r;

% 在同一文件夹内可调用此函数
[a,b] = circle(5)
```

- 函数句柄

```
f @(x,y) = x^2 + y^2
f(3,4)
```

- 函数可以嵌套

```
[a,b] = circle(f(3,4))
```

- for循环

```
for n = 1:1:10 % 1~10，步长为1
    disp(n)
end

myfriends = ["Ted", "Robyn", "Barney", "Lily", "Marshall"]

for friend = myfriends
    disp(friend)
end
```

- 字符串也可以用 for 遍历

```
% 循环可以嵌套
m = 5;
n = 5;
A = zeros(m,n);

for j = 1:n
    for i = 1:m
        A(i, j) = i + j;
    end
end
% 但在Matlab中尽量用矩阵计算，因为算法得到过优化
```

- 条件循环

```
n = 1001

if mod(n,2) == 1
    disp('Odd')
elseif mod(n,2) == 0
    disp('Even')
else
        disp('Not an Int')
end
```

- 矩阵

```
A = toeplitz([1,2,zeros(1,3)]
A(1,2)
A(1,:)
A(:,2)
A(:,1:3)
B = A(1:2,:)*A(:,1:2)

d = zeors(5,5,3);
```

- 结构体
  - 像python的字典

```
MMDA = struct('stu',{{'ZhangSan','LiSi'}},'grade',[98 99]);
MMDA.stu(1)
MMDA.stu{1}
MMDA.grade(2)
```

- 元胞

```
c = {} % 声明一个空元胞
c = cell(2,2);
c{1,1} = @(x)x^2;
c{1,2} = 5;
c{2,1} = 'MMDA';
c{2,2} = zeros(2,3);

a = c{1,1}
a(2)
```

- 包Packages

```
% Just download some mfile, and add the path to MATLAB
  add path
%% Linear algebra in MATLAB
%%%%%%%%%%%%%%%%%%%%%% Generate matrix %%%%%%%%%%%%%%%%%%%%%%
% First let's define a random matrix
A = rand(4,3)

% Define a vector of ones
x = ones(1,4)
x = [1,zeros(1,3)]

% Define a eye matrix
x = eye(4)

% Define a toeplitz matrix
K = toeplitz([1,2,3,0])

%%%%%%%%%%%%%%%%%%%%%%%%%% Operation %%%%%%%%%%%%%%%%%%%%%%%%%%
% Multiplication
b = A*x

a = [1,2,3];

c = a*a'
```

```
c = a'*a
c = a.*a

% Transposition
A.'

% conjugate transpose
A = [1+i,1-i;
    2+i,2-i]
A'

%  Solving linear systems
% The problem Ax=b for A is solved by the \ function. (or inv function)
% Ax = b

x = inv(A)*b
x = A\b

% LU Factorization
a = toeplitz([2,1,0,0])
[l,u] = lu(a)

% QR Factorization
[q,r] = qr(a)

% svd Factorization
[U,S,V] = svd(a)

% svd Factorization
[m,n] = eig(a)
```

- 绘图

```
% 2D图像
x = linspace(1,10,10);
y1 = x.^2;          % .^2获得每一个元素自己平方后的数组（矩阵）
y2 = 1\x;

%scatter plot
scatter(x,y1,'o')

%line plot
plot(x,y1,'r-^',x,y2,'b--*')

```

```
xlabel('x')
ylabel('y')
title('Simple plot')
xlim([1,7])

[ax,ay1,ay2] = plotyy(x,y1,x,y2)
```

- 子图

```
subplot(2,2,1);
x = linspace(0,10);
y1 = sin(x);
plot(x,y1)
title('Subplot 1: sin(x)')

subplot(2,2,4);
y2 = sin(5*x);
plot(x,y2)
title('Subplot 2: sin(5x)')
```

- 3D图

```
plot3(A)

%mesh plot
x=1:0.1:10;
y=1:0.1:10;
[x,y] = meshgrid(x,y);
z=x.^2-y.^2;
surf(x,y,z)
```

- 矩阵的绘制

```
spy(S) % 看矩阵元素的分布位置
```

- 文件的读写（没细讲）

```
data = load('Mat_1.txt');
fid = fopen('Mat_1.txt','a')
fprintf(fid,'Hello,Eason\r\n');
fclose(fid)
```

    - 建议具体问题具体查询，不同文件类型不一样
- 重点介绍一下图片的读取

```
pic = imread('bigtiger.jpeg')
imshow(pic)
imwrite(pic,'bigtiger1.jpeg') % 写到另一个文件里
```

- 补充知识点
  - clear 清空变量
  - clc 清空命令行

## 0.2 Python

如果没有安装其他东西，可以用python自带的IDLE（不推荐），pycharm，jupyter notebook都可以用，今天主要用jupyter notebook讲解。

- 语法比较自然

```
x = 3
y = 5
print(x + y)
print(x, y)
```

- 变量交换

```
# 不简洁的方法
z = x
x = y
y = z
print(x, y)

# 更简洁的方法
x, y = y, x
print(x, y)
```

- 单、双、三引号

```
print('"123"')
print("'1'23")
print('''Hello
world
!''')
```

- 定义函数（注意**缩进**）

```
def plus(x,y):
        res = x + y
        print(res)

plus(1,100)
```

- 循环

```
for i in 'hello':
        print(i, end = ' ') # end不写默认是换行符\n
```

- range

```
range(1, 10, 2) # 1, 3, 5, 7, 9, 范围1~9（range包括左边不包括右边），步长step=2

s = 0
for i in range(1, 10, 2):
        s += i # 就是s = s + i
print(s)
```

- 循环嵌套

```
for i in range(1,4):
        for j in range(1,5):
                print("*", end = ' ')
        print()
```

- 判断语句

```
number = eval(input('你的彩票号码是: '))
if number == 456456:
        print('中奖啦')
else:
        print('很遗憾')
```

  - if也可以嵌套
- 数据结构
  - number, tuple, list, dict, set

```
s = 100
x = 99.36

ss = (1,2,3) # 元组tuple 元素不能更改
m = ss[1]
print(m) # ss[0]是1

ss = [1,2,3] # 列表list 元素可以更改
ss.append(4) # 在末尾添加一个数4
print(ss)

xx = {'Mike':23, 'Tom':19} # 字典dict
print(xx['Mike'])
```

```
14
15 scores = {1, 2, 3} # 集合set
```

- 包package

```
1  import numpy as np
2  import scipy.linalg as la # 有很多线性代数功能
3  f = np.array([[0,1],[1,2]]) # 矩阵
4  f2 = la.inv(f) # inv求逆
5  print(f2)
6  la.lu(f) # LU分解
7
8  # toesplitz矩阵
9  m = np.zeros([1, 25])
10 m[0,0] = 2
11 m[0,1] = -1
12 k = la.toeplitz(m)
13 print(k)
14
15 b = la.kron(a1, k) + la.kron(k, a1) # la里有各种运算，用到的时候
   再查即可
16
17 import matplotlib.pyplot as plt # 用于画图
18
19 x = [1 ,2 ,3 ,4]
20 y = [1 ,2 ,3 ,4]
21 plt.plot(x, y, 'r*-')
22 plt.xlabel('x')
23 plt.ylabel('y')
24 plt.show()
25
26 fig = plt.figure()
27 ax = fig.add_subplot(111)
28 ax.matshow(k, vmin=-1, vmax=2, cmap='jet') # jet,magma
29 plt.show()
```

## 0.3 Julia

在Julia中按”]”，然后add 需要的包（IJulia等）

- 小试牛刀

```
1 println("Hello world!")
```

- Jupyter notebook的快捷命令
  - 键盘tab+a、tab+b分别在上面或下面加一个cell

    - 键盘shift+enter运行
- 变量

```
my_answer = 42

typeof(my_answer)

my_pi = 3.14159

typeof(my_pi)

pi # Julia里自己也有pi

alpha = 1

\alpha(press "tab") = 1 # Julia里可以用希腊字母以及表情😺（卡通图案），可以欢乐

\beta = 2

letters = [\alpha, \beta]

😺 = "smiley cat"

typeof(😺)
🍎 = 1
🍑 = 2
🍆 = 3
🍎 + 🍑 == 🍆
```

- 基本运算

```
sum = 3 + 7
difference = 10 - 3
product = 20*3
quotient = 10/5
modulus = 10%2
```

- 数据结构

```
# 元组tuple
myfavoriteanimals=("dog", "cat", "monkey")
myfavoriteanimals[1] # "dog"
# tuple元素不能更改

# NamedTuples
```

```
myfavoriteanimals = (bird="penguins", mamal="cats")
myfavoriteanimals.bird # "penguins"

# 字典Dict，key、value对儿
myphonebook = Dict(
    "Chen" => "111-222-3333",
    "Gu" => "444-555-6666")

myphonebook["Chen"]

for key in keys(myphonebook)
        println(key, "->", myphonebook[key])
end

# 数组Arrays
fibonacci = ]1, 1, 2, 3, 5, 8, 13]
mix = [1, 1, 2, 3, "chen", "gu"]

push!(mix, "5") # 末尾加个字符串"5"
pop!(mix) # pop出末尾元素

# 二维数组（矩阵）、三维数组
rand(4,3)
rand(4,3,2)
```

- 循环Loops

```
myTAs = ["aa", "bb", "cc", "dd"]

# while loops
i = 1
while 1 <= length(MyTAs)
        TA=myTAs[i]
        println("Hi, $TA, it's great to see you!")
        i += 1
end

# for loops
for i \in 1:10
        TA=myTAs[i]
        println("Hi, $TA, it's great to see you!")
end
```

- 一些生成矩阵的方法

```
m,n = 5, 5
A = fill(0, (m,n))

B = fill(0, (m,n))
for j in 1:n, i in 1:m # 比过往语言两层for简洁
                B[i,j] = i + j
        end
end
A

C = [i + j for i \in 1:m, j \in 1:n] # 更简洁

square_arr = [x^2 for x in 1:100]
```

- 条件判断Conditionals

```
N = 15
if (N % 3 ==0) && (N % 5 ==0)
        println("FizzBuzz")
elseif N % 3 ==0
        println("Fizz")
else
        println("Buzz")
end

# 支持a ? b : c语法
x, y = 3, 4
(x>y) > x : y # 取较大值
```

- 函数Functions

```
function sayhi(name)
        println("Hi, $name, it's great to see you!")
end

function sayhi(number::Int64)
        println(number)
end

sayhi(16)
sayhi("xiaoming")

sayhi2 = name -> println("haha $name") # 匿名的函数
sayhi2("monkey")

map(sayhi2, [1,2,3]) # map函数可以放入函数作为参数
```

```
map(x->x^3, [1,2,3]) # 可以放匿名的函数

# 加一个点.
function f(x)
        x^2
end

v = rand(3,3)

f(v) # matrix multiplication

f.(v) # element-wise
```

- 包Packages

```
using ToeplitzMatrices

v = [0 for i in 1:10]
v[1] = 2
v[2] = -1
SymmetricToeplitz(v)

using SpecialMatrices
K10 = Strang(10) # 因为这本书是Gilbert Strang写的，所以他把自己名字定义成了K矩阵

# Solving Ku=f

using LinearAlgebra

lu(K10) # lu分解

svd(K10) # svd分解

det(K10) # 求det

import Pkg; Pkg.add("BenchmarkTools") # 现场装一下这个包
using BenchmarkTools

F = ones(10,1)
u = K10\f # 直接得到结果
```

- 作图Plotting

```julia
using Plots
globaltemperatures = [14.4, 14.5, 14.8, 15.2, 15.5, 15.8]
numpirates = [45000, 20000, 15000, 5000, 400, 17]

plot(numpirates, globaltemperatures, label = "line", lw = 1,
color = :red)
scatter!(numpirates, globaltemperatures, label = "points")
# 感叹号意味着在原图上加画一个图
xlabel!("Number of Pirates")
ylabel!("Global Temperature (C)")
title!("Influence of pirate population on...")

heatmap(K10, color = :blues) # 可以用heatmap看矩阵
```

- Julia is fast
  - 代码略