

# Matlab与简单数学模型

Vanilla\_Yukirin

Vanilla-Yukirin/matlab-math-modeling

2025 年 11 月 13 日



# 目录

- 1 Matlab介绍
  - Matlab与Python的对比
  - Matlab的版本选择与安装
- 2 Matlab的基础使用
  - 认识Matlab基本界面
  - Matlab基本语法
  - Matlab基础绘图
- 3 Matlab简单应用
  - 递推加速
  - 数值计算
- 4 简单数学模型的介绍
  - 线性回归
  - 线性规划模型与求解
  - 复杂规划模型与启发式算法求解
- 5 课后习题
- 6 总结与答疑



# Matlab简介

Matlab（矩阵实验室）是由美国MathWorks公司开发的一种高级技术计算语言和交互式环境，广泛应用于科学计算、数据分析、算法开发和可视化等领域。



# 选Matlab还是Python

## Matlab

- 内置数学计算函数和工具箱、交互式工作区、工程领域标准。不需要会环境配置，安装好后开箱即用
- 文档齐全，内置帮助文档（按F12即可查看函数详细用法），相比去问AI会更加精确且全面
- 付费软件，昂贵的授权费，每个工具箱还需要单独付费<sup>a</sup>，很多学校没有资金订阅

<sup>a</sup>其实我们学校之前是有订阅过的，但是嘛.....

## Python

- 有numpy、sklearn、matplotlib等科学计算库、jupyter notebook交互式工作区，但是需要掌握环境管理/虚拟环境（anaconda、uv、venv等）
- 教程齐全。作为一门热门语言，网络上教程很多，问LLM也能得到不错的答案
- 免费软件，免费、开源、社区活跃，每个人都能自由的使用（freeware）



# 为什么建议数模初学者用Matlab

- **Matlab** 是工程领域的计算器+画图板+方程求解器——5分钟从数据到图表，无需折腾环境配置。安装完立即算矩阵、画图、解方程，让大家专注算法而非工具；
- 而 **Python** 需要先学虚拟环境管理，对完全零基础的同学是额外负担。
- 装 **Python** 的编程环境常见的环境问题（依赖冲突、版本不兼容）往往劝退初学者。



# Matlab版本选择

Matlab相邻版本之间差别并不大。且近年更新的高级功能做数模一般用不上，所以只要不用过于老的版本（推荐使用  $\geq 2016b$ ），语法都是通用的。

**ab**尾缀的含义：

- **a**: 上半年发布的版本，通常在3月份发布
- **b**: 下半年发布的版本，通常在9月份发布

通常来说，一个a版本发布后，经过几个月的使用和反馈，MathWorks会修复一些Bug，并在同年秋季的b版本中使其更加稳定。所以推荐使用**b**尾缀的版本。

接下来的教程是基于Matlab R2024b的，如果你的版本不是2024b也没有关系，操作完全相同。



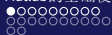
# Matlab安装演示

自行下载Matlab，网盘链接在钉钉群中

如果下载速度过慢，可以带u盘下课找我拷贝

*TODO: 安装Matlab过程截图*





# Matlab基本界面介绍

双击Matlab图标，等待一会儿，就会出现白色的Matlab软件窗口。

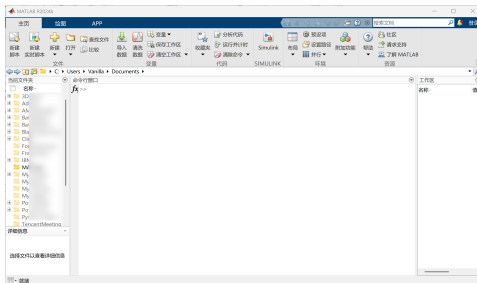


图: Matlab R2024b 界面示例





图: Matlab R2024b 界面示例



# Matlab工作区

工作区，顾名思义就是进行工作的地方。建议先更换工作区到准备好的地方，做好文件管理，不要到处都散乱着代码文件。

通常，初次启动Matlab时，默认路径可能是C:\Users\用户名\Documents\MATLAB，建议更改为自己准备好的工作目录。

个人习惯在比赛目录下新建一个code或者workspace文件夹专门用来存放代码





认识Matlab基本界面

# 更换工作区目录

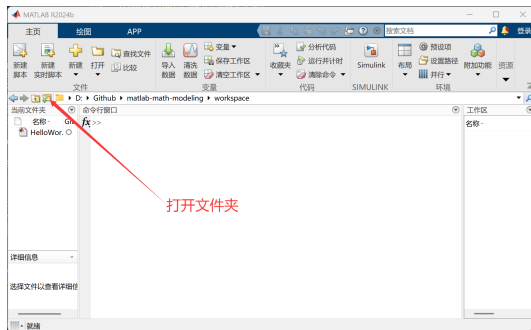


图: 更换当前工作目录



# 命令行窗口

命令行窗口的左上角会有三个大于号：>>，表示等待用户输入命令。

命令行窗口是 **Matlab** 最常用的窗口，用于输入命令、执行命令、查看结果。

可以直接在命令行窗口中写一些简单的代码，也可以把它当作一个**全能计算器**。

```
>> format long % set long format
```

```
>> pi
```

```
ans =
```

```
3.141592653589793
```

```
>> pi^2
```

```
ans =
```

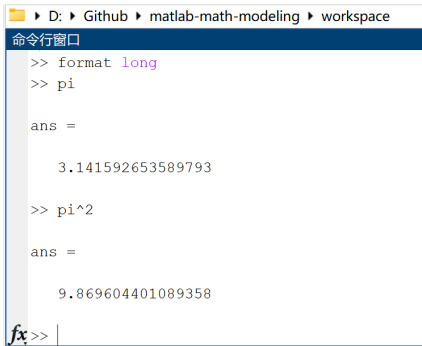
```
9.869604401089358
```

```
>>
```





# 命令行窗口



```
>> format long
>> pi

ans =

    3.141592653589793

>> pi^2

ans =

    9.869604401089358

fx>>
```

图: 命令行窗口



# 文件命名规范

MATLAB 文件名必须以字母开头，最多包含 63 个字母数字字符或下划线。

也就是说：

- 不能以数字开头
- 不能包含中文
- 不能用减号 -，应使用下划线 \_

养成良好的文件命名习惯，可以帮助我们在数模比赛中快速的找到文件。在数模比赛中，由于会有很多道子问题，所以个人推荐这样命名，既规整又有含义：

- `problem1_GA.m` — 问题一，使用遗传算法
- `problem2_draw.m` — 问题二，画图相关代码



# 运行代码

新建 HelloWorld.m 文件，写入如下内容，然后用编辑器的运行按钮或命令行运行。

文件: HelloWorld.m

```
clc; % clear console
clear; % clear variables
n=input("Please input a number: ");
n
fprintf("Hello, World!\n")
```

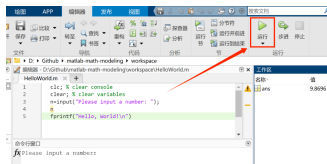
常见运行方式：

- ① 上方的“运行”按钮
- ② 选中代码，按 F9（在命令行中执行所选）
- ③ 在行号左侧点击出现的运行蓝条

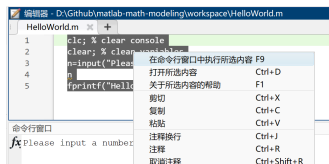


## 认识Matlab基本界面

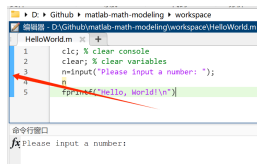
## 运行代码示例



(a) 运行按钮



(b) 选中代码按F9/右键运行



(c) 左侧点击运行（运行段）

运行后命令行会显示“Please input a number: ”，  
输入一个数字并回车后，会输出你输入的数字并打印Hello, World!。





# 从矩阵开始

Matlab 的名字是“矩阵实验室”，内部数据以矩阵为主，直接面向矩阵进行运算。

文件: Section2\_1\_matrix.m

```
clc;clear; % 清空控制台和变量
```

```
A=[1,2,3;4,5,6;7,8,9] % 创建矩阵A，是一个3行3列的矩阵
```

```
% [] 表示构建一个矩阵
```

```
% 逗号或空格用来分隔同一行不同列的元素
```

```
% 分号用来分隔不同行（同一列）的元素，或者说表示换行
```

```
A(1:2,:) % 拆：取A的前两行
```

```
% 矩阵后面连着一个括号，代表取矩阵的某个部分
```

```
% 第一个参数 1:2 表示“从1到2”，即第1行和第2行
```

```
% 第二个参数 : 表示“所有”
```

```
% 所以这里取的是A的前两行，所有列
```

```
B=[A;10,11,12] % 添：在A的下面添加一行新数据，构成新矩阵B
```

```
% 把新行 [10 11 12] 用分号和A连接，表示把新行添加到A的下面
```





# 向量是特殊的矩阵

向量是特殊的矩阵，可以看作是只有一行或者一列的矩阵。

同理，标量可以看作是1行1列的矩阵。这也是为什么说Matlab中万物皆矩阵

文件: Section2\_2\_vector.m

```
clc;clear
```

```
x = [1 2 3 4 5]           % 一行五列→ 行向量
```

```
y = [1; 2; 3; 4; 5]       % 五行一列→ 列向量
```

```
a = [1,2,3,4,5,6,7,8,9]   % 手写会很累
```

```
b = 1:1:10                % 从1开始，步长1，到10结束→ [1 2 3 4 5 6 7 8 9 10]
```

```
d = 0:2:20                % 从0开始，步长2，到20结束→ [0 2 4 6 8 10 12 14 16 18 20]
```

```
e = 10:-1:1               % 从10开始，步长-1，到1结束→ [10 9 8 7 6 5 4 3 2 1]
```

步长可正可负，会构造一个等差数列。可以用这种方法快速的构造等间隔的数轴坐标、时间序列等。



# 常用的矩阵创建方法

文件: Section2\_3\_create.m

```
clc;clear
```

```
a=[1,2,3,4,5,6,7,8,9] % 创建一个向量
```

```
aa=a' % 加'是转置, 行→列
```

```
b=[1,2,3;4,5,6;7,8,9] % 创建一个矩阵
```

```
c=1:1:10 % 创建一个从1到10的向量
```

```
e=eye(4) % 生成4维 (4*4) 的单位矩阵I (对角线为1)
```

```
z=zeros(1,4) % 生成1行4列的全零矩阵
```

```
o=ones(4,1) % 生成4行1列的全1矩阵
```

```
% 随机矩阵
```

```
r=rand(4) % 生成4*4的0-1范围内的随机矩阵
```

```
rn=randn(4) % 生成4*4的均值为0, 方差为1的正态分布随机矩阵
```

```
ri=randi([1,10],2,4) % 生成2*4的随机整数矩阵 (1到10之间)
```

```
% 对角矩阵
```

```
d=diag([1,2,3]) % 对角线上是1,2,3的3×3矩阵
```

```
d=diag(b) % 提取b的对角线元素
```



# 常用的矩阵创建方法

文件: Section2\_3\_create.m

% 三角矩阵

**U=triu(b)**

% 上三角矩阵 (下三角变为0)

**L=tril(b)**

% 下三角矩阵 (上三角变为0)

% 生成相同维度的矩阵

**size(b)**

% 会输出b的尺寸 [3 3]

**f=zeros(size(b))**

% 生成一个和b矩阵尺寸一样的矩阵

% **f=zeros(size(b,1),size(b,2))** 和 **f=zeros(height(b),width(b))** 同理

% 重复矩阵

**R= repmat([1 2],2,3)**

% 把 [1 2] 重复成2行3列块

% 常用序列

**x=linspace(0,10,100)**

% 0到10之间等距100个点, 无需手动设置步长

**y=logspace(1,3,5)**

%  $10^1$  到  $10^3$  之间对数间隔5个点





# 变量名和ans





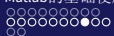
# 分号与常用输出方式





# 矩阵的基本运算





# Matlab中的循环与条件语句



# Matlab中的函数定义与调用







# 绘图的作用

绘图是数模与科研中必不可少的一环。

绘图可以更方便的观察数据，也可以进行简单的数据处理。模型求解也通常需要用图像进行辅助展示。







# 总结

- 1 Matlab介绍
  - Matlab与Python的对比
  - Matlab的版本选择与安装
- 2 Matlab的基础使用
  - 认识Matlab基本界面
  - Matlab基本语法
  - Matlab基础绘图
- 3 Matlab简单应用
  - 递推加速
  - 数值计算
- 4 简单数学模型的介绍
  - 线性回归
  - 线性规划模型与求解
  - 复杂规划模型与启发式算法求解
- 5 课后习题
- 6 总结与答疑



# 总结

今天晚上和周六周日，我会在数模5群里面看大家的问题

大家可以把不懂的地方发到群里，我和数模协会的同学会尽量帮大家解答



# 行内等宽: ...

这里演示行内等宽: `for i = 1:n`, 适合短的、简单的代码或变量名。



# 使用 ...

演示 listings 行内: `disp('hello')`, 适合需要 listings 语法高亮的短代码片段。