

# Introduction to MATLAB

数学建模

李景治

2025 Spring



- ① MATLAB 概述
- ② 基本语法与操作
- ③ 自学 MATLAB
- ④ MATLAB 在数学建模中的应用

- ① MATLAB 概述
- ② 基本语法与操作
- ③ 自学 MATLAB
- ④ MATLAB 在数学建模中的应用

# 简介

**MATLAB** (Matrix Laboratory) 是一种广泛使用的高级编程语言和开发环境，主要用于数值计算、数据分析、可视化和算法开发。它最初由 MathWorks 公司于 1984 年发布，具有强大的数学计算和可视化功能，广泛应用于工程、科学、金融、经济等多个领域。

# 主要特点

- 有着丰富的内置函数和工具箱 (toolbox)。
- 基本元素是矩阵和数组，许多算法的实现都基于矩阵运算。
- 支持高效的数值计算和符号计算。
- 提供了丰富的绘图功能。

# MATLAB 主界面



- ① MATLAB 概述
- ② 基本语法与操作
- ③ 自学 MATLAB
- ④ MATLAB 在数学建模中的应用

# 标量的创建和运算

## 标量的创建

```
a = 3.14;    % 直接赋值。  
b = 7 + 3;   % 使用表达式创建标量  
c = exp(1);  % 从函数创建标量
```

## 标量的运算

```
sum = a + b;    % 加法  
diff = a - b;   % 减法  
prod = a * b;   % 乘法  
quot = a / b;   % 除法  
power = a^b;    % 幂运算  
sqrt_a = sqrt(a); % 开方运算
```

此外还可以使用其他的内置函数进行运算。



# 向量的创建和运算

创建行向量：元素之间用空格或逗号分隔。

```
v1 = [1 2 3 4 5];      % 使用空格  
v2 = [1, 2, 3, 4, 5]; % 使用逗号
```

创建列向量：元素之间用分号分隔，或者转置行向量。

```
v3 = [1; 2; 3; 4; 5]; % 使用分号  
v4 = [1 2 3 4 5]';    % 转置
```

使用 colon 运算符 (:) 创建向量：创建一个从一个起始值到终止值，且步长固定的向量。

```
v5 = 0:2:10; % 创建一个从0到10，步长为2的行向量
```

使用 linspace 创建向量：创建一个在指定区间内均匀分布的向量。

```
% 在[0, 10]区间内生成5个均匀分布的点  
v6 = linspace(0, 10, 5);
```

向量的加减法：加法和减法的前提是两个向量的维度必须相同。

```
v7 = [1, 2, 3];  
v8 = [4, 5, 6];  
v9 = v6 + v7;    % 向量加法  
v10 = v6 - v7;   % 向量减法
```

## 元素群运算

```
x.*y=[x1*y1 x2*y2 x3*y3] % 元素群乘积
x./y=[x1/y1 x2/y2 x3/y3] % 元素群右除，右边的y做分母
x.\y=[y1/x1 y2/x2 y3/x3] % 元素群左除，左边的x做分母
x.^5=[x1^5 x2^5 x3^5]      % 元素群乘
2.^x=[2^x1 2^x2 2^x3]      % 元素群乘幂
x.^y=[x1^y1 x2^y2 x3^y3] % 元素群乘幂
```

## 向量的元素索引与切片

```
v = [10, 20, 30, 40, 50];
elem = v(3);      % 获取第3个元素
sub_v = v(2:4);   % 获取从第2个到第4个元素
```

# 矩阵的创建和运算

使用方括号创建矩阵，元素用空格或逗号分隔，不同行用分号分隔。

```
A = [1 2 3; 4 5 6]; % 2 行 3 列
```

创建特殊矩阵：

- `zeros(m, n)`：创建一个  $m$  行  $n$  列的零矩阵。
- `ones(m, n)`：创建一个  $m$  行  $n$  列的全 1 矩阵。
- `eye(n)`：创建一个  $n$  行  $n$  的单位矩阵。

## 创建随机矩阵：

- `rand(m, n)`: 创建一个  $m$  行  $n$  列的矩阵，元素为 0 到 1 之间的随机数。
- `randn(m, n)`: 创建一个  $m$  行  $n$  列的矩阵，元素为均值为 0、标准差为 1 的正态分布随机数。

## 使用 `reshape` 改变矩阵的维度：

`reshape` 用于将一个矩阵转换为指定维度的矩阵，元素的顺序会按列填充。

```
H = 1:9; % 创建一个从 1 到 9 的行向量  
I = reshape(H, 3, 3); % 将行向量转换为 3x3 的矩阵
```

矩阵的加减法：

矩阵的加法与减法是元素对应相加和相减。要求两个矩阵的大小必须相同。

矩阵乘法有两种常见的形式：普通矩阵乘法（也叫线性代数乘法）和元素级乘法（Hadamard 乘法）。

- 普通矩阵乘法：矩阵 A 的列数要等于矩阵 B 的行数。
- 元素级乘法：使用.\* 运算符。

```
A = [1, 2; 3, 4];
```

```
B = [5, 6; 7, 8];
```

```
% 普通矩阵乘法
```

```
C = A * B;    % 结果为 [19, 22; 43, 50]
```

```
% 元素级乘法
```

```
D = A .* B;   % 结果为 [5, 12; 21, 32]
```

## 矩阵的除法：

- 左除 “\”：  
求矩阵方程  $AX=B$  的解，解为  $X=A\backslash B$
- 右除 “/”：  
求矩阵方程  $XA=B$  的解，解为  $X=B/A$

## 矩阵的索引与切片

```
A = [1, 2, 3; 4, 5, 6; 7, 8, 9];  
% 获取第2行第3列的元素  
elem = A(2, 3); % 结果为 6  
% 获取第1行的所有元素  
row1 = A(1, :); % 结果为 [1, 2, 3]  
% 获取第2列的所有元素  
col2 = A(:, 2); % 结果为 [2; 5; 8]  
% 获取矩阵的子矩阵  
sub_matrix = A(1:2, 2:3);  
% 获取第1到第2行, 第2到第3列的子矩阵
```



# 关系运算符和逻辑运算符

运算符	含义	运算符	含义
<	小于	~=	不等于
<=	小于或等于	&	与
>	大于	或	或
>=	大于或等于	~	非
==	等于		

# 控制语句

## 条件语句

```
if x > 5
    disp('x大于5');
elseif x == 5
    disp('x等于5');
else
    disp('x小于5');
end
```

## 循环结构

- for 循环

```
for i = 1:5  
    disp(i);  
end
```

- while 循环

```
i = 1;  
while i <= 5  
    disp(i);  
    i = i + 1;  
end
```

# 函数

- 定义函数

MATLAB 函数需要保存为一个文件，文件名与函数名相同。

```
function result = addNumbers(a, b)
    result = a + b;
end
```

- 调用内置函数

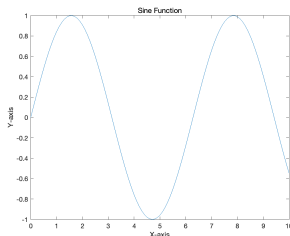
```
y = sin(pi); % 计算sin(pi)
meanVal = mean([1, 2, 3, 4, 5]); % 计算均值
```

# 绘图与数据可视化

绘制简单图形：

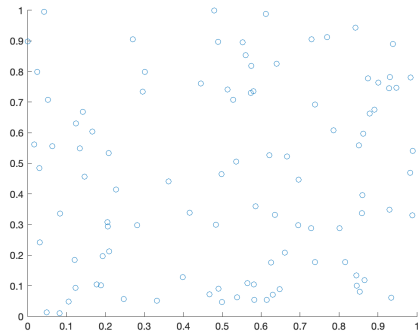
```
x = 0:0.1:10;  
y = sin(x);  
plot(x, y);  
title('Sine Function');  
xlabel('X-axis');  
ylabel('Y-axis');
```

% 绘制x与y的关系图  
% 设置标题  
% 设置X轴标签  
% 设置Y轴标签



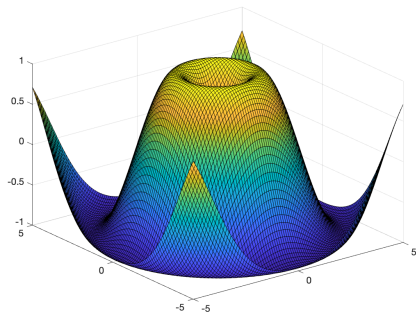
绘制散点图：

```
x = rand(1, 100); % 随机生成100个点  
y = rand(1, 100);  
scatter(x, y); % 绘制散点图
```



绘制 3D 图：

```
[X, Y] = meshgrid(-5:0.1:5, -5:0.1:5);  
Z = sin(sqrt(X.^2 + Y.^2));  
surf(X, Y, Z); % 绘制三维表面图
```



## m 文件

M 文件是包含 MATLAB 代码的文本文件，文件扩展名为.m。m 文件有两种类型：

- 脚本（script）文件：脚本文件是一种没有输入输出参数的文件，包含一系列 MATLAB 命令。脚本中的代码按顺序执行，结果会直接影响工作区的变量。
- 函数（function）文件：函数文件包含一个或多个可以接受输入和返回输出的函数。函数文件通常用于封装可重复使用的代码块，以便在其他代码中调用。



对于函数文件，必须以 `function` 关键字开头，后跟函数名、输入参数和输出参数。每个函数文件只能包含一个主函数（主函数的文件名必须与函数名一致），但是可以包含多个局部子函数。

例：

```
% 函数文件：addNumbers.m
function sum = addNumbers(a, b)
    % 计算两个数的和
    sum = a + b;
end
```

这个函数可以通过在命令窗口或其他 M 文件中调用：

```
result = addNumbers(3, 5);
% 调用addNumbers函数，返回结果8
```

- ① MATLAB 概述
- ② 基本语法与操作
- ③ 自学 MATLAB
- ④ MATLAB 在数学建模中的应用

# MATLAB 的帮助系统

MATLAB 的帮助系统非常强大且易于使用，适合自学。从总体上看大致分为三类：

- 命令窗口查询帮助命令
- MATLAB 文档
- MATLAB 示例

# 命令窗口查询帮助命令

命令字	功能
help	显示帮助系统所有项目，用目录形式列出
help < 函数名 >	查询与函数有关的帮助
lookfor < 关键词 >	查询包含关键词的有相关函数
demo	打开演示窗口
info	显示 MATLAB 的一般信息
whatsnew	列出 MATLAB 的新有信息

# MATLAB 文档

## 打开方式

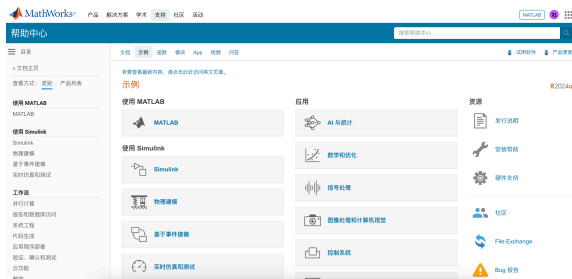
- ① 命令行输入 doc 或 doc + 函数名
- ② 主页 -> 帮助 -> 文档
- ③ MATLAB 官网 -> 帮助中心 -> 文档



# MATLAB 示例

## 打开方式

- ① 命令行输入 demo
- ② 主页 -> 帮助 -> 示例
- ③ MATLAB 官网 -> 帮助中心 -> 示例



# MATLAB 的课程

此外，MATLAB 的官网中有着不同类型的课程可供学习。

## 定制学习路径

过滤依据

类型

课程 74  
入门之旅(入门课程) 24  
课程规划 14

使用 MATLAB

所有 MATLAB 课程 42  
MATLAB 快速入门 5  
语言基础知识 14  
数据导入和分析 16  
图形 4  
编程 13  
App 构建 1  
软件开发工具 7

使用 Simulink

所有 Simulink 课程 14  
Simulink 快速入门 6  
建模和仿真 2  
物理建模 7

排序依据 精选

< 1 到 10 个结果, 共 88 个 >



还是 MATLAB 新手?从以下课程开始学习:  
**MATLAB 入门之旅**  
2% | 2 小时 | [语言](#)  
MATLAB 基础知识快速入门。



还是 Simulink 新手?从以下课程开始学习:  
**Simulink 入门之旅**  
2 小时 | [语言](#)  
Simulink 基础知识快速入门。



**MATLAB 核心技能**  
学习路径: 4 门课程  
提升您的 MATLAB 核心技能, 自信应对新挑战。



**Simulink Fundamentals**  
8 小时 | [语言](#)  
Learn how to use Simulink, a graphical simulation tool for modeling dynamic physical systems.

- ① MATLAB 概述
- ② 基本语法与操作
- ③ 自学 MATLAB
- ④ MATLAB 在数学建模中的应用



## Example1: 多项式拟合

要对表 1 中的数据进行拟合。

1	2	3	4	5	6	7	8	9
9	7	6	3	-1	2	5	7	20

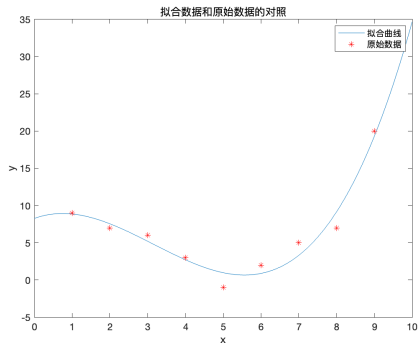
表 1: 实例数据

拟合的命令如下:

```
x = [1 2 3 4 5 6 7 8 9];  
y = [9 7 6 3 -1 2 5 7 20];  
P = polyfit(x,y,3);  
xi = 0:.1:10;  
yi = polyval(P,xi);  
plot(xi,yi,x,y,'r*')  
xlabel('x')  
ylabel('y')  
title('拟合数据和原始数据的对照')  
legend('拟合曲线','原始数据')
```

`polyfit(X,Y,N)`: 多项式拟合, 返回降幂排序的多项式系数。  
`polyval(P,xi)`: 计算多项式的值。

得到的拟合曲线与原始数据的对照如图所示。



## Example2: 求解线性规划问题

$$\begin{aligned} \min z &= 2x_1 + 3x_2 + x_3 \\ \text{s.t.} \quad &\begin{cases} x_1 + 4x_2 + 2x_3 \geq 8 \\ 3x_1 + 2x_2 \geq 6 \\ x_1, x_2, x_3 \geq 0 \end{cases} \end{aligned}$$

编写 MATLAB 程序如下:

```
c = [2 ; 3 ; 1 ];  
a= [1,4,2;3,2,0];  
b = [8 ; 6 ];  
[x, y] = linprog(c, - a, - b, [], [], zeros (3,1))
```

linprog 是 MATLAB 内置的非线性规划求解器函数。  
结果为

找到最优解。

最优解为：  $x_1 = 0.8$ ,  $x_2 = 1.8$ ,  $x_3 = 0.0$

最优值为：  $z = 7.0$