

数字图像处理

MATLAB简介

赵毅力 ylzhao@swfu.edu.cn

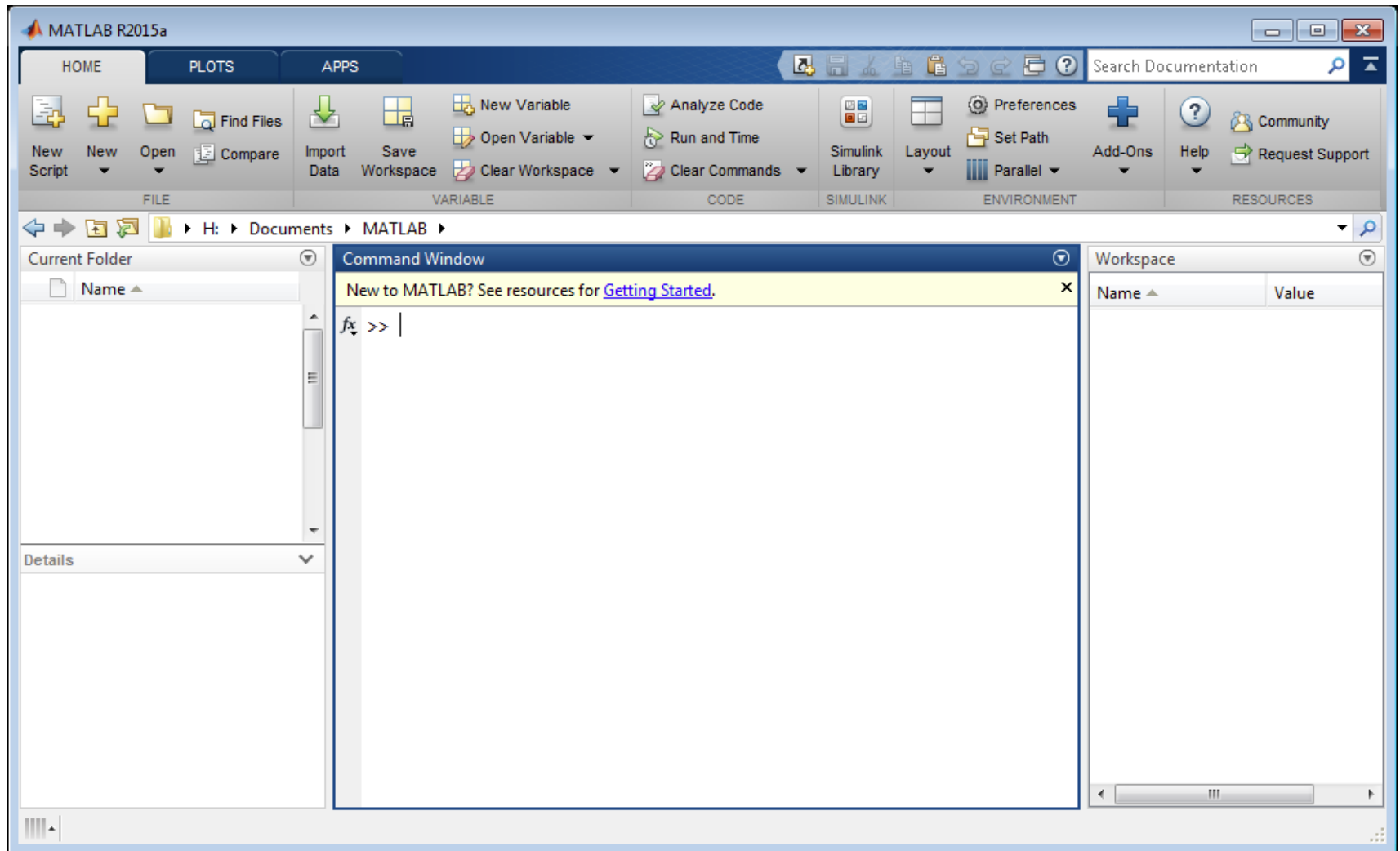
大数据与智能工程学院

西南林业大学

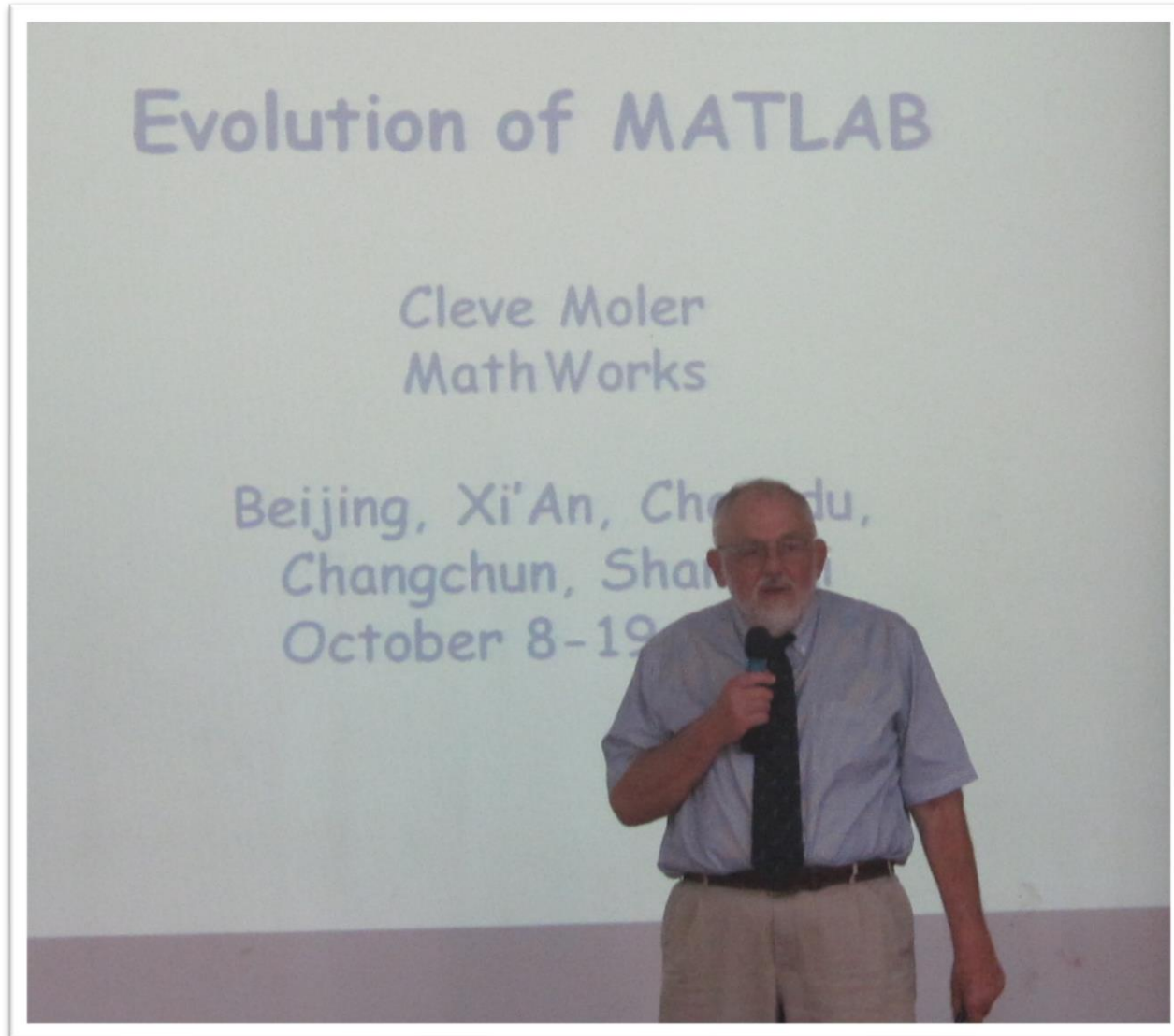
MATLAB

- MATLAB是一个科学计算和仿真模拟的软件平台，由一系列工具箱组成：
 - Aerospace Toolbox
 - Antenna Toolbox
 - Bioinformatics Toolbox
 - Communications System Toolbox
 - Computer Vision System Toolbox
 - ***Image Processing Toolbox***
 - Control System Toolbox
 - Curve Fitting Toolbox
 - Data Acquisition Toolbox
 - ...
- <http://cn.mathworks.com/>

MATLAB - Matrix Laboratory



Evolution of MATLAB



变量赋值

- $a = 1$
- $b = 2$
- $c = a + b$
- $d = \cos(a)$
- $\sin(a)$
- $e = a * b;$

向量和矩阵

- 矩阵是二维数组
- 1×1 的矩阵称为**标量**
- 只有一行的矩阵称为**行向量**
- 只有一列的矩阵成为**列向量**

输入矩阵

- $A = [16\ 3\ 2\ 13; 5\ 10\ 11\ 8; 9\ 6\ 7\ 12; 4\ 15\ 14\ 1]$

A =

16 3 2 13

5 10 11 8

9 6 7 12

4 15 14 1

一些常用的矩阵生成函数

- `magic` – 魔方矩阵
- `zeros` – 全零矩阵
- `ones` – 全一矩阵
- `eye` – 单位矩阵
- `rand` – 均匀分布矩阵
- `randn` – 正态分布矩阵

矩阵和数组

- `A1 = [1 2 3 4] % 行向量`
- `A2 = [1;2;3;4] % 列向量`
- `B = [1 2 3; 4 5 6; 7 8 9] % 矩阵`
- `Z = zeros(5, 1)`
- `C = A + 10`
- `D = sin(A)`
- `E = B' % 转置`
- `F = inv(B) % 逆矩阵`
- `P = B * E % 矩阵乘法`
- `W = B .* B % 矩阵点乘`
- `A.^3 % 元素点乘`

矩阵连接

- $A = [1 \ 2 \ 3; 4 \ 5 \ 6; 7 \ 8 \ 9]$
- $A1 = [A, A]$ % 水平方向连接
- $A2 = [A; A]$ % 垂直方向连接

矩阵求和

- `sum(A)`

ans =

34 34 34 34

指定矩阵计算的维度

- `sum(A, 2)`

ans =

34

34

34

34

矩阵转置

- A'

ans =

16	5	9	4
3	10	6	15
2	11	7	14
13	8	12	1

对角矩阵

- `sum(diag(A))`

ans =

34

magic函数

- `M = magic(4)`

M =

16	2	3	13
5	11	10	8
9	7	6	12
4	14	15	1

矩阵重排序

- $C = M(:, [1\ 3\ 2\ 4])$

C =

16 3 2 13

5 10 11 8

9 6 7 12

4 15 14 1

M =

16 2 3 13

5 11 10 8

9 7 6 12

4 14 15 1

删除矩阵中的行或列

- `A = magic(4)`
- `X = A`
- `X(:,2) = []`

矩阵索引

- `A = magic(4)`
- `A(4, 2)` % 行和列下标
- `A(8)` % 线性索引
- `A(4, 5)` % 线性索引超出矩阵维度?
- `A(1:3, 2)` % 使用冒号来引用多个元素
- `A(3, :)`
- `C = 0:10:100` % 范围
- `D = 0:10`

逻辑数组

- `A = [1 3 0 5 7];`
- `L = logical(A);`
- `A(L)`

逻辑运算

逻辑运算	含义
$A \& B$, <code>and(A, B)</code>	逻辑与
$\sim A$, <code>not(A)</code>	逻辑非
$A \mid B$, <code>or(A, B)</code>	逻辑或
<code>xor(A, B)</code>	逻辑异或
<code>all(A)</code>	判断是否全零
<code>any(A)</code>	判断是否有非零元素
<code>logical(A)</code>	把数值矩阵转换为逻辑矩阵
<code>islogical(A)</code>	判断矩阵是否为逻辑矩阵

逻辑下标

- `A = magic(4)`
- 问题：将矩阵A中所有非素数全部设置为0
- `isprime(A)`

find 函数

- find : 寻找矩阵中的非零元素及其索引
- `A = magic(4)`
- 问题：搜索矩阵A中的所有素数及其位置？
- (1) 首先将矩阵A中所有非素数置为零；
- (2) 使用find函数寻找素数的位置。

关系运算

关系运算	含义
<	小于
<=	小于等于
>	大于
>=	大于等于
==	相等
~=	不相等

特殊的函数

<code>pi</code>	3.14159265...
<code>i</code>	Imaginary unit, $\sqrt{-1}$
<code>j</code>	Same as <code>i</code>
<code>eps</code>	Floating-point relative precision, $\varepsilon = 2^{-52}$
<code>realmin</code>	Smallest floating-point number, 2^{-1022}
<code>realmax</code>	Largest floating-point number, $(2 - \varepsilon)2^{1023}$
<code>Inf</code>	Infinity
<code>NaN</code>	Not-a-number

复数

- $\text{sqrt}(-1)$
- $C = [3+4i, 4+3j; -i, 10j]$

省略号的用法

- $s = 1 - 1/2 + 1/3 - 1/4 \dots$
- $+ 1/5 - 1/6 + 1/7$

字符串

- `mytext = 'Hello, world!'`
- `otherText = 'You''re right'`
- `longText = [myText, ' – ', otherText] % 字符串连接`
- `f = 71; c = (f - 32) / 1.8;`
- `tempText = ['Temperature is ', num2str(c), 'C']`

字符串输出

- `str = 'MATLAB';`
- `display(str);`
- `fprintf('%s\n', str);`

字符串转换

- `n1 = 3.14;`
- `s1 = num2str(n1);`
- `n2 = str2num(s1);`

工作区变量

- whos % 查看工作区所有变量信息
- whos A % 查看工作区变量A的信息
- save(myfile.mat, A) % 将变量A保存为.mat文件
- clear % 清除工作区所有变量
- clear A % 清除工作区变量A
- load(myfile.mat) % 从.mat文件加载变量

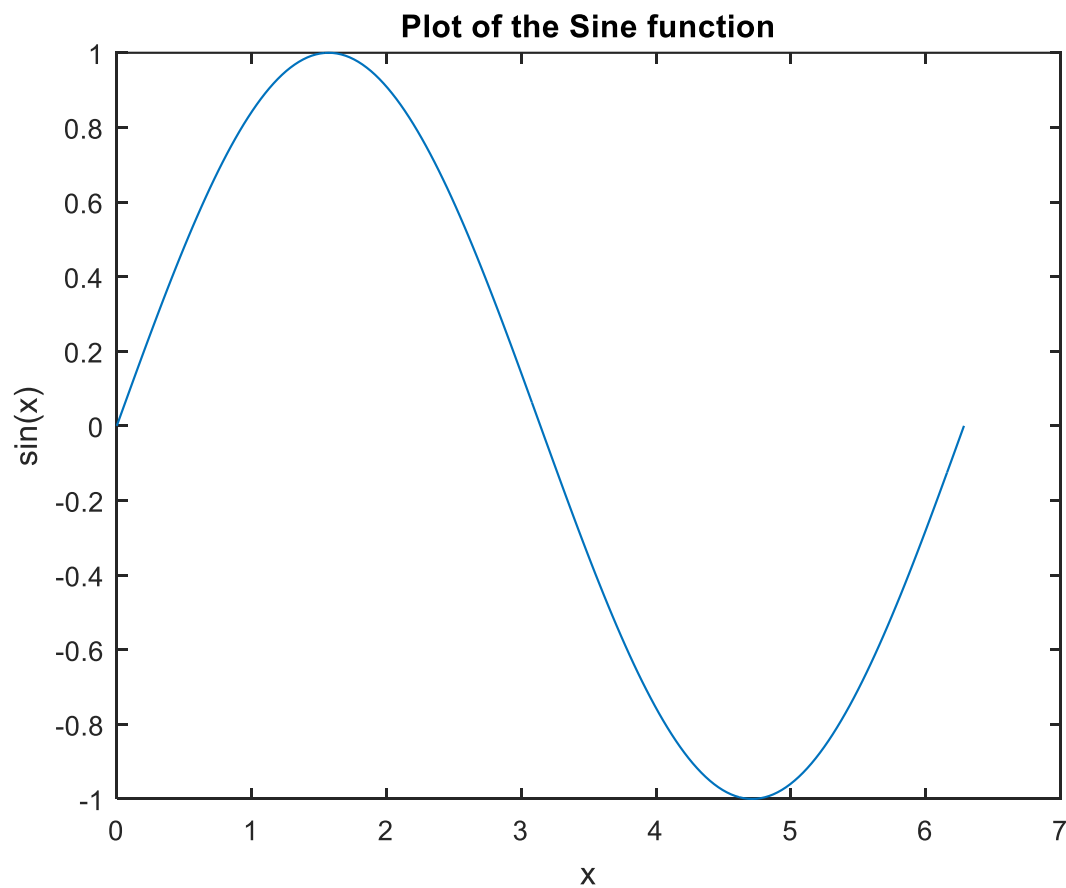
函数调用

- `A = [1 3 5];`
- `maxA = max(A)`
- `C = [10 6 4];`
- `maxC = max(A, C)`
- `[maxA, loc] = max(A)` % 计算最大值及其位置
- `clc` % 清除命令行的内容

绘制二维图形

- `x = 0:pi/100:2*pi;`
- `y = sin(x)`
- `plot(x, y)`
- `xlabel('x')`
- `ylabel('sin(x)')`
- `title('Plot of the Sine function')`

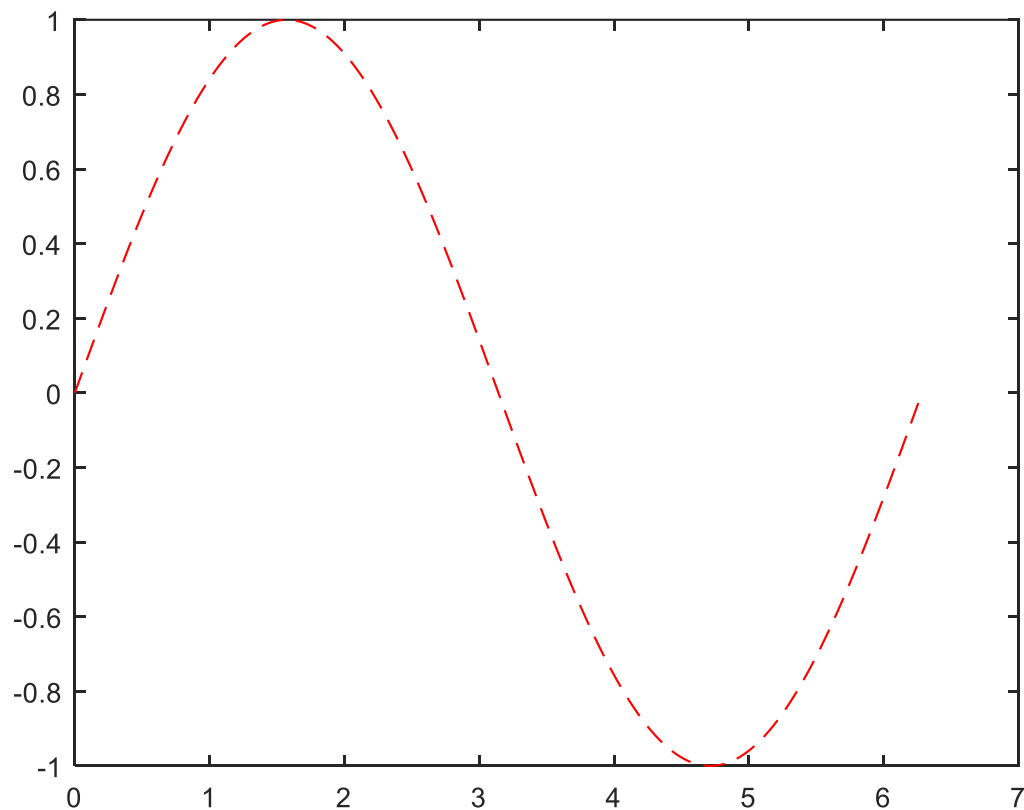
绘制Sine函数图形



绘制Sine函数图形

- `plot(x, y, 'r--')`

绘制Sine函数图形



线型

Specifier	Line Style
-	Solid line
--	Dashed line
:	Dotted line
-.	Dash-dot line

标记形状

Specifier	Marker
o	Circle
+	Plus sign
*	Asterisk
.	Point
x	Cross
s	Square

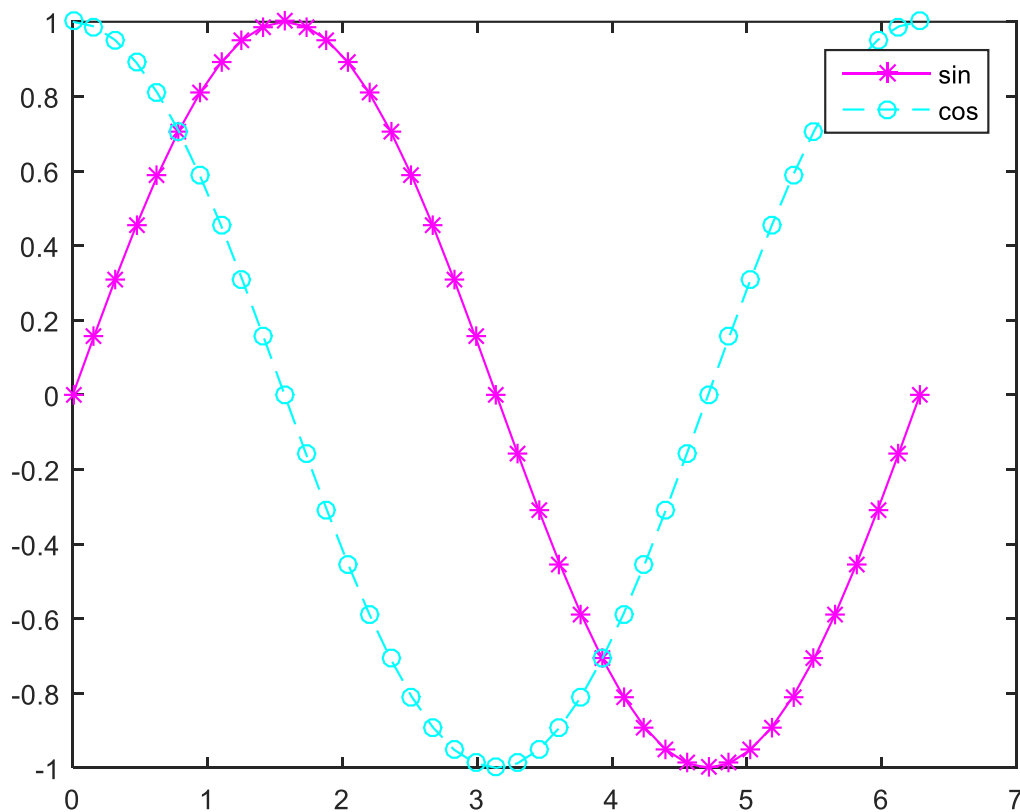
绘制颜色

Specifier	Color
y	yellow
m	magenta
c	cyan
r	red
g	green
b	blue
w	white
k	black

Hold on / hold off

- `x = 0:pi/20:2*pi;`
- `y = sin(x);`
- `plot(x, y, 'm-*)`
- `hold on`
- `y2 = cos(x);`
- `plot(x, y2, 'c--o')`
- `legend('sin', 'cos')`
- `hold off`

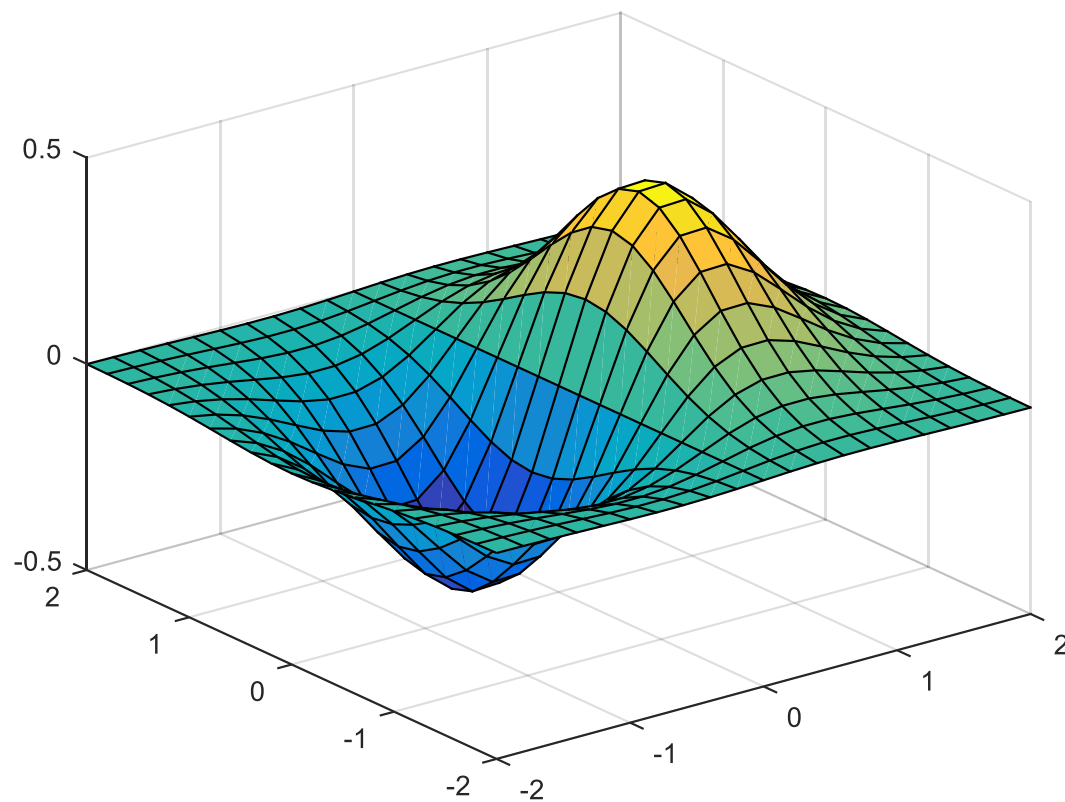
绘制正弦和余弦函数图形



绘制三维图形

- `[X, Y] = meshgrid(-2:.2:2);`
- `Z = X .* exp(-X.^2 - Y.^2);`
- `surf(X, Y, Z)`

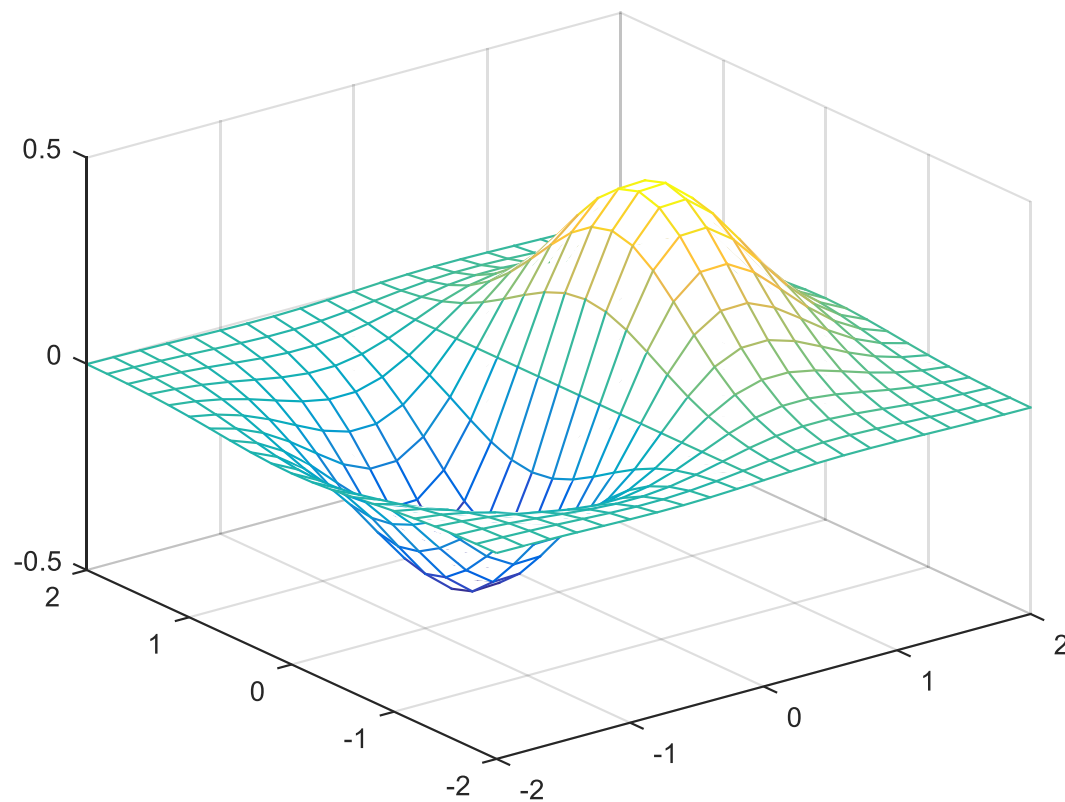
绘制三维图形



绘制三维图形

- `[X, Y] = meshgrid(-2:.2:2);`
- `Z = X .* exp(-X.^2 - Y.^2);`
- `mesh(X, Y, Z)`

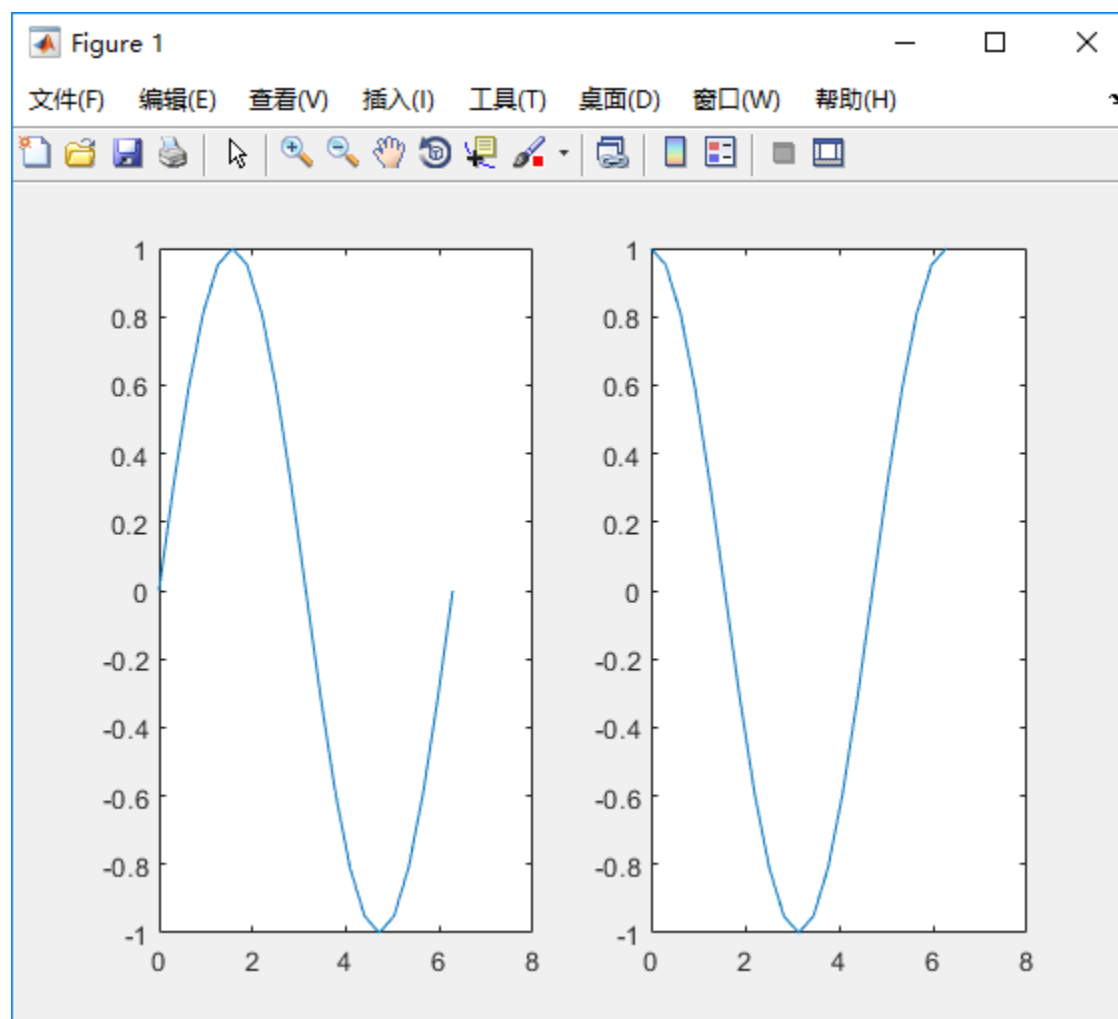
绘制三维图形



子图 - subplot

- `t = 0:pi/10:2*pi;`
- `w1 = sin(t);`
- `w2 = cos(t);`
- `subplot(1,2,1); plot(t,w1); title('sin');`
- `subplot(1,2,2); plot(t,w2); title('cos');`

子图 - subplot



程序设计与脚本

- 使用MATLAB实现快速排序算法

流程控制

if ... elseif ... else ...

for

while

switch ... case ... otherwise ...

break

continue

try ... catch ...

脚本位置

设置搜索路径

帮助和文档

doc mean

help mean

数组类型

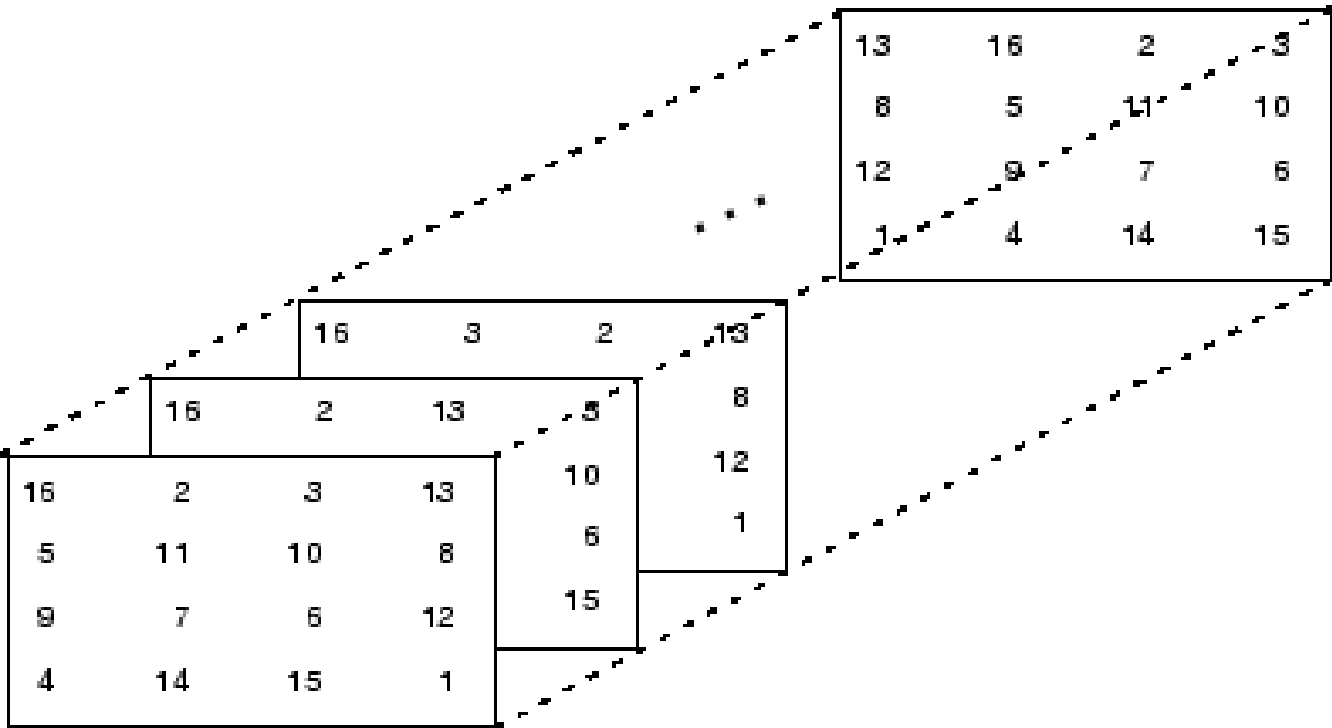
多维数组

细胞数组

多维数组

- `p = perms(1:4);`
- `A = magic(4);`
- `M = zeros(4, 4, 24);`
- `for k = 1 : 24`
- `M(:, :, k) = A(:, p(k, :));`
- `end`

多维数组



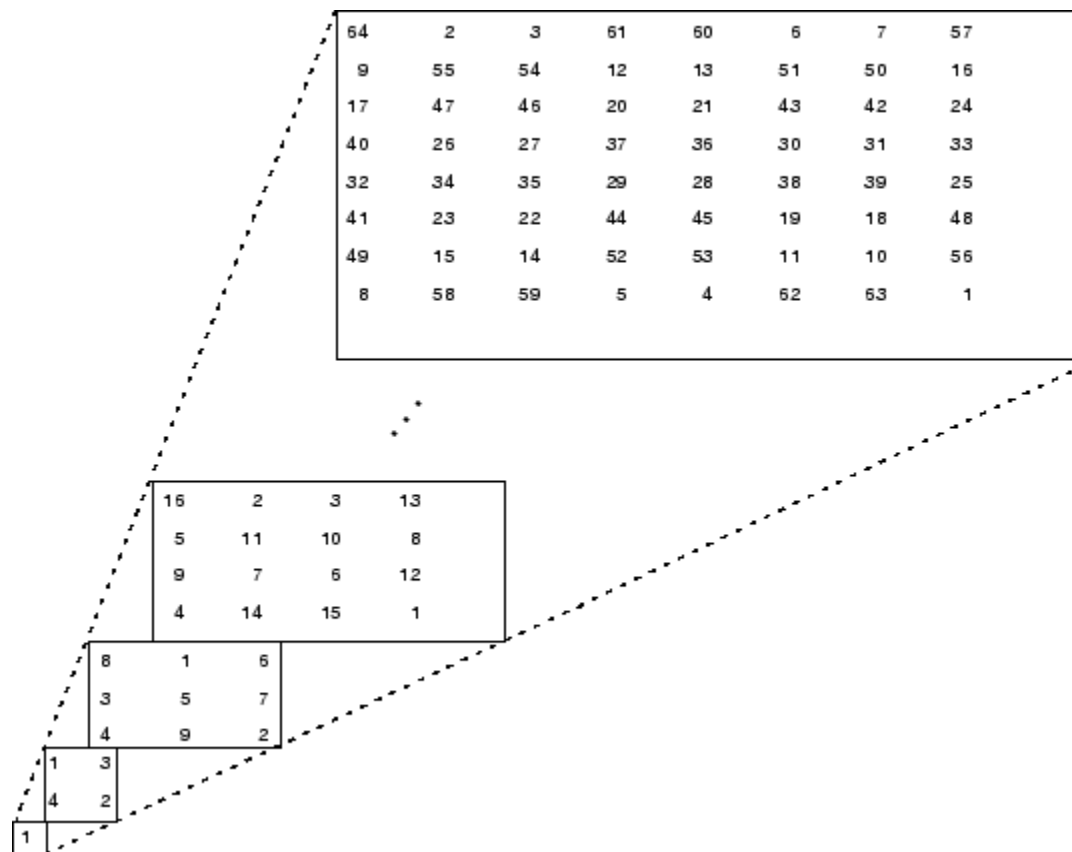
多维数组求和

- `sum(M, 1)`
- `sum(M, 2)`
- `sum(M, 3)`

细胞数组

- `M = cell(8, 1);`
- `for n = 1:8`
- `M{n} = magic(n);`
- `end`

细胞数组



结构体

- `S.name = 'Ed Plum';`
- `S.score = 83;`
- `S.grade = 'A+';`
- `S(2).name = 'Toni Miller';`
- `S(2).score = 91;`
- `S(2).grade = 'A-';`
- `S(3) = struct('name', 'Jerry Garcia', 'score', 70,... 'grade', 'C')`

结构体

- `scores = [S.score];`
- `names = {S.name};`
- `[N1 N2 N3] = S.name;`

函数

```
function f = fact(n)
%FACT    Compute the factorial of n
%    f = fact(n) computes the factorial of n
f = prod(1:n);
```

函数句柄

- `h = @fact`
- `h(4)`

匿名函数

- `sqr = @(x) x.^2`
- `a = sqr(5)`
- `b = sqr([1 2 3])`

匿名函数

- 计算积分

$$\int_{x=0}^{\infty} e^{-x^2} (\ln x)^2$$

- `q = integral(fun, xmin, xmax)`
- `fun = @(x) exp(-x.^2).*log(x).^2`
- `q = integral(fun, 0, Inf)`

代码优化

数组预分配

循环向量化

数组预分配

```
function y = sinfun1(M)
x = 0:M-1;
for k = 1:numel(x)
    y(k) = sin(x(k) / (100*pi));
end
```

```
tic; sinfun1(20000); toc
tic; sinfun2(20000); toc
```

```
function y = sinfun2(M)
x = 0:M-1;
y = zeros(1, numel(x));
for k = 1:numel(x)
    y(k) = sin(x(k) / (100*pi));
end
```

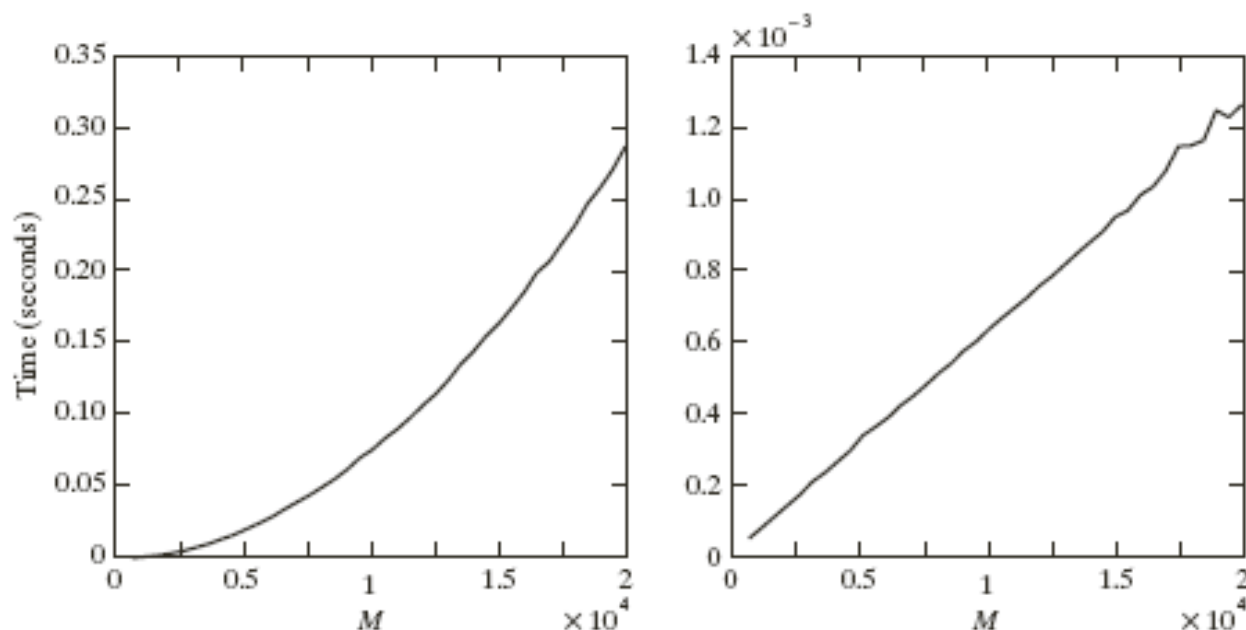
```
M = 20000;
f1 = @() sinfun1(M);
f2 = @() sinfun2(M);
t1 = timeit(f1);
t2 = timeit(f2);
```


数组预分配

a b

FIGURE 1.3

(a) Approximate execution times for function `sinfun1` as a function of M . (b) Approximate times for function `sinfun2`. The glitches were caused by interval variations in memory paging. The time scales in (a) and (b) are different.



循环向量化

```
function y = sinfun3(M)  
x = 0:M-1;  
y = sin(x ./ 100*pi);
```