



# 数学软件

# Mathematica

---

——使用入门

# 主要内容

---

- 符号计算系统
- Mathematica 基本用法
- 基本运算
- 符号计算
- 图形功能
- 程序设计（略）

# 符号计算系统

---

## ■ 符号计算系统

- 也称为**计算机代数**
- 以推理为主，是一个表示数学知识和数学工具的系统
- 与代数计算、算法设计、机器学习、自动推理等紧密联系
- 一般由系统内核、符号计算语言和若干软件包组成
- 通常包括符号计算、数值计算、图形演示和程序设计

借助计算机速度快的特点，帮助人们完成在短时间内无法完成的公式推导计算。

# 符号计算系统

---

## ■ 常见的符号计算软件

### ● 商业软件

Mathematica, Maple, MuPAD, MathCad

### ● 免费软件

Axiom, CoCoA, Derive, Euler, GAP, Maxima, ...

更多符号计算软件参见课程主页相关网络链接。

### ● 符号计算软件的两种运行方式：[交互方式](#)和[程序方式](#)

# Mathematica 介绍

---

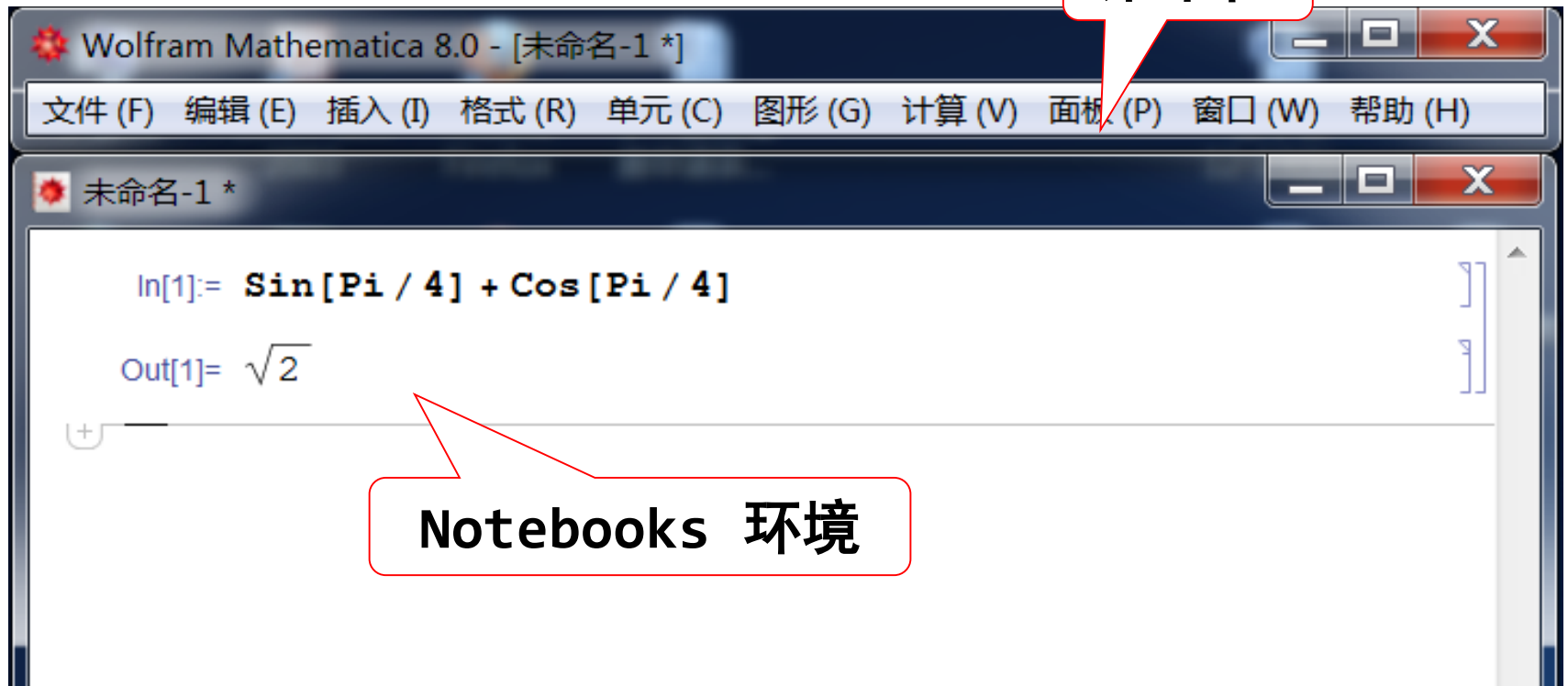
- Mathematica 由美国 Wolfram Research 公司于 1988 年推出，目前的最新版本为 Mathematica 8.04
- Mathematica 很好地结合了数值和符号计算引擎、图形系统、编程语言、文本系统以及与其他应用程序的高级连接，是目前使用最广泛的数学软件之一，也是世界上符号计算系统中最强大的两个系统之一。
- Mathematica 有简体中文版：中文界面和中文帮助
- 可在官方网站上下载试用版

# Mathematica 欢迎界面



# Mathematica 工作界面

- Mathematica 的工作平台：自带的 Notebooks 环境
- Mathematica 工作界面



- 在 Notebooks 环境下输入命令，系统执行后返回结果

最好最完整的学习材料：系统自带的帮助和实例演示

# Mathematica 使用

---

## ■ Mathematica 语言规则

- ① 所有命令和内置函数都是以**大写字母**开始
- ② 函数的参数是在**方括号**中给出
- ③ 乘法运算符可以用**空格**代替（不建议这么做）
- ④ 内置的函数名通常都很长，使用函数的名字的全拼



# Mathematica 使用

---

## ■ Mathematica 命令的执行（简称**执行键**）

**Shift + Enter**

- 可以运行单个命令或语句
- 也可以运行多个语句：输入全部语句后再按**执行键**
- 命令（语句）分隔符：回车或分号
- 如果不需要显示运行结果：在语句后面加分号

## ■ 输入和输出标识符：**In[n]：** 和 **Out[n]**

```
In[1]:= Sin[Pi/4]+Cos[Pi/4]
```

```
Out[1]=  $\sqrt{2}$ 
```

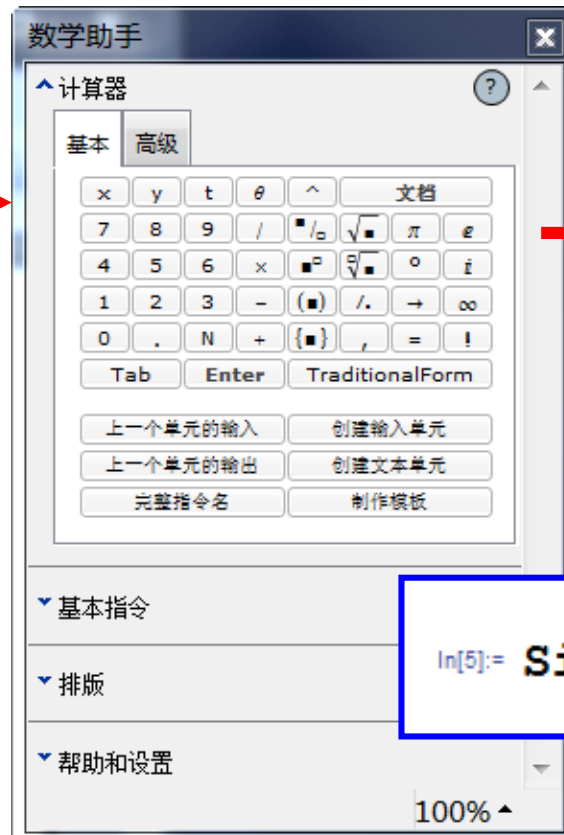
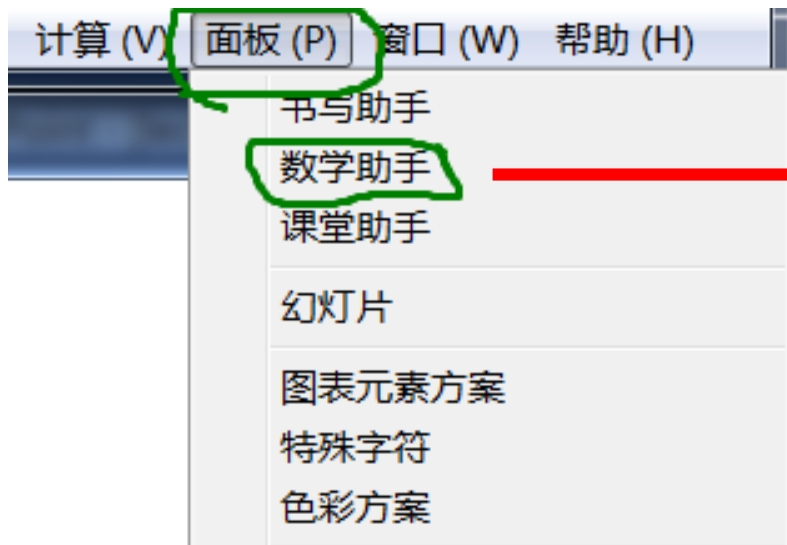
# 数学公式的输入

## ■ 数学公式的输入

- 直接输入

```
In[1]:= Sin[Pi/4]+Cos[Pi/4]
```

- 借助 Mathematica 提供的面板 (Palettes)



```
In[5]:= Sin  $\left[\frac{\pi}{4}\right]$  + Cos  $\left[\frac{\pi}{4}\right]$ 
```

# 一些常用符号

<code>()</code>	运算的结合
<code>f[x]</code>	函数取值
<code>{}</code>	列表
<code>[[k]]</code>	分量
<code>%</code>	最后一次的计算结果
<code>%%</code>	倒数第二次的计算结果
<code>%%%(k)</code>	倒数第 k 次的计算结果
<code>%k</code>	第 k 次计算结果, 即 <code>Out[k]</code> 的值
<code>?name</code>	显示系统变量、命令或函数的简短介绍
<code>??name</code>	显示系统变量、命令或函数的全部信息
<code>(*comments*)</code>	注解
<code>Ctrl+K</code>	命令补全功能

# 内部常数

## ■ Mathematica 中的数：普通数字和内部常数

- 整数，有理数，实数（任意精度和机器精度），复数
- 常用内部常数

Degree	角度到弧度的转换系数， $\text{Pi}/180$
E	自然对数的底， $2.71828\dots$
EulerGamma	Euler 常数： $\lim_{n \rightarrow \infty} (1 + \frac{1}{2} + \dots + \frac{1}{n} - \ln n) = 0.577215\dots$
GoldenRatio	黄金分割数： $\frac{1}{2}(1 + \sqrt{5})$
I	虚部单位
Infinity	无穷大
Pi	圆周率

# 基本运算符

## ■ 基本运算：

### ● 算术运算

+ - * /	加减乘除
^	幂
!	阶乘（运算级别比加减乘除和幂运算高）

### ● 比较运算与逻辑运算

== > < >= <= !=
&&    ! Xor

# 变量与表达式

---

## ■ 变量

- 通常字母开头，后面可以跟字母与数字
- 长度不限
- 区分大小写
- 可以包含希腊字母或中文，如“数学”
- 用户自定义变量建议都用小写，避免与系统自带函数冲突
- 变量赋值： **变量名=表达式**

## ■ 表达式

- Mathematica 中一切皆为表达式
- 算术表达式，关系表达式，逻辑表达式，复合表达式

# 变量操作相关函数

<code>x=y=a</code>	给多个变量赋相同的值
<code>Unset[x]</code> 或 <code>x=.</code>	清除变量的值
<code>Clear[x1,x2,...]</code>	清除变量
<code>Print[x1,x2,...]</code>	打印变量的值
<code>expr/.{x-&gt;a,y-&gt;b,...}</code>	变量替换（变量本身没有被赋值）

例：已知  $f(x,y)=2x+y$ , 计算  $f(2,3)$

```
In[1]:= Clear[x,y];  
In[2]:= f=2*x+y;  
In[3]:= f./{x->2,y->3} (* f(2,3) 的值 *)  
In[3]:= f./{2->5} (*把 2 替换成 5*)
```

# 数的基本运算

## ● Mathematica 中的实数分精确数和双精度数

<code>N[x,n]</code>	<code>x</code> 的带 <code>n</code> 位有效数字的近似值
<code>N[x]</code>	<code>x</code> 的双精度近似值
<code>IntegerPart[x]</code>	整数部分
<code>FractionalPart[x]</code>	小数部分
<code>Floor[x]</code>	取整：不大于 <code>x</code> 的最大整数
<code>Round[x]</code>	取整：四舍五入
<code>Ceil[x]</code>	取整：不小于 <code>x</code> 的最小整数
<code>Precision[expr]</code>	显示计算精度

在 Mathematica 中，当输入的式子中所有数字都为整数时，输出的结果是精确的；但若式子中含有小数点，则通常以近似方式输出运算结果。



# 数字进制的转换

<code>b^^xxxxx</code>	输入一个 b 进制数 ( $2 \leq b \leq 36$ )
<code>BaseForm[x,b]</code>	十进制数 x 的 b 进制形式
<code>IntegerString[x,b]</code>	同上，但写成字符串形式
<code>FromDigits[str]</code>	从字符串构造整数
<code>FromDigits[list,b]</code>	从 b 进制数字列表中构造整数
<code>IntegerDigits[x,b]</code>	十进制数 x 的 b 进制数字列表

# 内置函数

---

- Mathematica 具有超过 3000 个内置函数，具体见“帮助” → “参考资料中心” 的左下角处的 “函数索引”
- Mathematica 函数命名规则
  - ① 第一个字母大写，后面跟小写字母，如 `Sin[x]`，`Log[x]`
  - ② 大多数函数名与数学中的名称相同
  - ③ 当函数名分为几段时，每一段的头一个字母大写，后面的用小写字母，如：`ArcSin[x]`
- 寻求帮助： `??函数名`

```
In[1]:= ??Sin
```

# 常用初等函数

<code>Abs[x]</code>	绝对值
<code>Sign[x]</code>	符号函数
<code>Power[x,y]</code>	幂函数 $x^y$
<code>Sqrt[x]</code>	平方根
<code>Exp[x]</code>	以 e 为底的指数函数
<code>Log[x], Log[b,x]</code>	以 e 和 b 为底的对数函数
<code>Factorial[n]</code>	$n!$
<code>Factorial2[n]</code>	$n!!$
<code>GCD[n1,n2,...]</code>	最大公约数
<code>GCD[list]</code>	列表中所有数的最大公约数
<code>LCM[n1,n2,...], LCM[list]</code>	最小公倍数
<code>Max[x1,x2,...], Max[list]</code>	求最大值
<code>Min[x1,x2,...], Min[list]</code>	求最小值

# 常用初等函数

<code>Re[x], Im[x]</code>	提取实部和虚部
<code>Conjugate[x]</code>	取共轭
<code>Arg[x]</code>	辐角
<code>Mod[m,n]</code>	m 除以 n 的余数
<code>Quotient[m,n]</code>	m 除以 n 的整数商
<code>Sin[x], Cos[x], ...</code>	三角函数
<code>ArcSin[x], ArcCos[x], ...</code>	反三角函数
<code>Sinh, Cosh, ...</code>	双曲函数
<code>ArcSinh, ArcCosh, ...</code>	反双曲函数
<code>Prime[k]</code>	第 k 个素数
<code>PrimeQ[n]</code>	判断 n 是否为素数
<code>Binomial[n,m]</code>	二项式系数 $C_n^m$

# 随机函数

<code>Random[]</code>	生成 $[0, 1]$ 内的一个随机实数
<code>Random[Real, a]</code>	生成 $[0, a]$ 内的一个随机实数
<code>Random[Real, {a,b}]</code>	生成 $[a, b]$ 内的一个随机实数
<code>Random[Integer]</code>	随机给出整数 0 或 1
<code>Random[Integer, {a,b}]</code>	生成 $[a, b]$ 内的一个随机整数
<code>Random[Complex]</code>	生成一个 $[0, 1] \times [0, 1]$ 内的一个随机复数

```
In[1] := Random[Real, 10]
```

```
In[2] := Random[Integer, {1, 100}]
```

# 打印函数

---

<code>Print[x,y,...]</code>	在屏幕输出变量的值
-----------------------------	-----------

```
In[1]:= x=5; Print["x=",x]
```

# 函数调用方式

- 标准方式：函数名[变量列表]

```
In[1] := x=Pi/4; Sin[x]
```

- 其它方式

```
In[2] := x=Pi/4; x//Sin      (* 后缀形式 *)
```

```
In[3] := x=Pi/4; Sin@x      (* 前缀形式 *)
```

```
In[4] := 5~Mod~3            (* 中缀形式, 等价于 Mod[5,3] *)
```

# 自定义函数

- Mathematica 允许用户自定义函数，一般格式为

函数名[自变量名1\_, 自变量名2\_, ...]:= 表达式

- 这里函数名与变量名的规定相同
- 方括号中的每个自变量名后都要有一个下划线 “\_”
- 中间的 “:=” 为定义号
- 注意符号表达式与函数的区别

<code>f[x_]:=2*x-3</code>	定义一个一元函数
<code>f[x_,y_]:=Log[x/y]-Power[x,y]</code>	定义一个二元函数
<code>Clear[f]</code>	清除自定义的函数

- 自定义函数前，最好先清除自变量的值，否则可能会出现意想不到的错误



# 自定义函数

---

```
In[1]:= f=2*x+y      (* 符号表达式 *)
```

```
In[2]:= g[x_,y_]:=2*x+y  (* 函数 *)
```

```
In[3]:= f[2,3]      (* 结果是什么? *)
```

```
In[4]:= g[2,3]      (* 这个结果又是什么? *)
```

# 字符串

## ■ 字符串：用双引号括起来的字符序列

### ● 字符串相关函数

<code>Characters[str]</code>	转化为字符列表
<code>StringJoin[s1,s2,...]</code>	字符串合并
<code>s1&lt;&gt;s2&lt;&gt;...</code>	字符串合并
<code>StringLength[str]</code>	字符串长度
<code>StringSplit[str]</code>	按空白字符分割字符串
<code>ToExpression[str]</code>	转化为表达式
<code>ToString[expr]</code>	将表达式转化为字符串

### ● 更多字符串相关函数参见 “[参考资料中心](#)”

# 列表

## ■ 列表

- 是 Mathematica 的基本对象，可用来表示集合，数组等
- 分为标准列表和稀疏列表
  - 标准列表：  
用大括号括起来的有限个元素，元素之间用逗号分隔

```
In[1]:= x={1,2,"hello",{1,0}};
```

- 列表中的元素可以是不同类型的任意 Mathematica 对象
- 列表可以嵌套，如 矩阵
- 稀疏列表：通常由 `SparseArray` 来定义

Mathematica 提供了上千个列表操作函数，参见 “[参考资料中心](#)”

# 列表生成

- 枚举法：利用大括号，直接输入
- 利用 **Array**, **Range**, **Table** 等函数

<code>Array[f,n]</code>	生成一维列表 $\{f[1],f[2],\dots,f[n]\}$
<code>Array[f,{n1,n2,...}]</code>	生成多维列表（嵌套），这里的 $f$ 为函数
<code>Range[a,b,h]</code>	生成等差数列列表： $a$ 为首项， $h$ 为公差，最后一项不超过 $b$ ； $a$ 和 $h$ 的缺省值为 1
<code>Table[expr,{n}]</code>	生成 $n$ 元列表 $\{expr,expr,\dots,expr\}$
<code>Table[expr,{i,a,b,h}]</code>	$\{expr \mid i \text{ 在 } Range[a,b,h] \text{ 中变化}\}$
<code>Table[expr,{i,list}]</code>	$\{expr \mid i \text{ 在列表 } list \text{ 中变化}\}$

- **Table** 中的 **expr** 一般给的是通项公式

<code>RandomInteger[range,n]</code>	生成 $n$ 个伪随机整数， $range$ 表示取值范围
<code>RandomReal[range,n]</code>	生成 $n$ 个伪随机实数， $n$ 缺省值为 1

# 列表生成举例

---

```
In[1]:= x=Array[Sin,3]
```

```
Out[1]= {Sin[1], Sin[2], Sin[3]}
```

```
In[2]:= y=Table[Sin[k],{k,3}]
```

```
Out[2]= {Sin[1], Sin[2], Sin[3]}
```

```
In[3]:= f=RandomReal[{1,5},2]
```

```
Out[3]= {4.68541, 2.86979}
```

## 生成二维列表

```
In[4]:= z=Table[i*j,{i,3,5},{j,2,4,2}]
```

```
Out[4]= {{6, 12}, {8, 16}, {10, 20}}
```

# 列表分量

<code>list[[k]]</code>	第 k 个分量
<code>list[[-k]]</code>	倒数第 k 个分量
<code>list[[i]][[j]]</code>	第 i 个分量的第 j 个分量
<code>list[[i,j]]</code>	第 i 个分量的第 j 个分量
<code>list[{{i,j,...}}]</code>	{list[[i]],list[[j]], ... }
<code>First[list]</code>	第一个分量
<code>Last[list]</code>	最后一个分量
<code>Take[list,k]</code>	前 k 个分量
<code>Take[list,-k]</code>	最后 k 个分量
<code>Take[list,{i}]</code>	{list[[i]]}
<code>Take[list,{i,j}]</code>	{list[[i]],list[[i+1]], ..., list[[j]]}
<code>list[[i;;j]]</code>	同上
<code>Take[list,{i,j,h}]</code>	{list[[i]],list[[i+h]], ... }
<code>list[[i;;j;;h]]</code>	同上

# 列表修改

<code>Drop[list,{k}]</code>	删除第 k 个分量
<code>Drop[list,k]</code>	删除前 k 个分量
<code>Drop[list,-k]</code>	删除最后 k 个分量
<code>Drop[list,{i,j,h}]</code>	删除 <code>list[[i]],list[[i+h]], ...</code>
<code>Rest[list]</code>	删除第一个分量
<code>Most[list]</code>	删除最后一个分量
<code>Delete[list,k]</code>	删除第 k 个分量
<code>Delete[list,-k]</code>	删除倒数第 k 个分量
<code>Delete[list,{i,j, ...}]</code>	删除 <code>list[[i,j,...]]</code>
<code>Insert[list,x,k]</code>	在第 k 个位置插入 x
<code>Prepend[list,x]</code>	将 x 插入到 list 的最前面
<code>PrependTo[list,x]</code>	将 x 插入到 list 的最前面, 并将结果赋给 list
<code>Append[list,x]</code>	将 x 插入到 list 的最后面
<code>AppendTo[list,x]</code>	将 x 插入到 list 的最前面, 并将结果赋给 list

# 列表运算

<code>Sort[list]</code>	从小到大排序
<code>Reverse[list]</code>	将列表中的元素顺序倒过来
<code>Apply[Plus,list]</code>	求列表中所有元素之和
<code>Apply[Times,list]</code>	求列表中所有元素之积
<code>Total[list]</code>	求列表中所有元素之和
<code>Length[list]</code>	列表元素的个数

- 当函数作用在列表上时，表示作用在每个分量上

```
In[1] := x=Range[0,Pi,Pi/4];
```

```
In[2] := y=Sin[x]
```

- 利用列表可以同时给多个变量赋值

```
In[3] := {x,y}={2,3}
```



# 矩阵定义

- 矩阵是列表的一种，可用 `Array`, `Table` 等函数创建

`Array[函数名, 取值范围]`

`Array[f,n]`

生成向量  $\{f[1], f[2], \dots, f[n]\}$

`Array[f,{m,n}]`

生成  $m$  行  $n$  列的矩阵  $f[i,j]$

```
In[1]:= Clear[i,j]; f[i_,j_]:=1/(i+j-1);
```

```
In[2]:= A=Array[f,{3,3}]
```

# 矩阵定义

`Table[通项公式, {循环范围}, {循环范围}, ...]`

## ● 循环范围表示方法

<code>{i,a,b,h}</code>	i 从 a 到 b, 步长为 h, 最后一项不超过 b
<code>{i,a,b}</code>	缺省步长为 1
<code>{i,b}</code>	缺省首项为 1
<code>{k}</code>	重复 k 次
<hr/>	
<code>Table[expr, {i,a,b,h}]</code>	向量
<code>Table[expr, {i,a1,b1,h1}, {j,a2,b2,h2}]</code>	矩阵

# 矩阵举例

```
In[1] := x=Table[0,{5}] (* 长度为5的零向量 *)
```

```
In[2] := A=Table[0,{5},{5}] (* 5阶零矩阵 *)
```

```
In[3] := A=Table[1/(i+j-1),{i,3},{j,3}]
```

```
In[4] := B=Table[x+y,{x,2,6,2},{y,3,12,3}]
```

```
In[5] := P=Table[Binomial[i,j],{i,0,4},{j,0,i}];
```

```
In[6] := TableForm[P] (* 杨辉三角形 *)
```

# 矩阵分量

<code>MatrixForm[A]</code>	按矩阵方式显示
<code>TableForm[A]</code>	按表格方式显示（没有括号）
<code>Dimensions[A]</code>	矩阵的维数（行数和列数）
<code>Dimensions[A,k]</code>	矩阵的前 k 重维数
<hr/>	
<code>A[[i,j]]</code>	矩阵分量 $a_{ij}$
<code>A[[All,j]]</code>	矩阵第 $j$ 列
<code>Take[A,{i1,i2,...},{j1,j2,...}]</code>	子矩阵

```
In[1] := Take[A,{1,2},{2,3}]/MatrixForm
```

# 特殊矩阵

<code>IdentityMatrix[n]</code>	$n$ 阶单位矩阵
<code>ConstantArray[c, {m, n}]</code>	生成 $m$ 行 $n$ 列的常数矩阵，元素都为 $c$
<code>DiagonalMatrix[list]</code>	以列表 <code>list</code> 中的元素为对角线的对角矩阵
<code>HilbertMatrix[n]</code>	$n$ 阶的 Hilbert 矩阵
<code>RandomInteger[range, {m, n}]</code>	$m$ 行 $n$ 列的伪随机整数矩阵 <code>range</code> 表示元素取值范围
<code>RandomReal[range, {m, n}]</code>	$m$ 行 $n$ 列的伪随机实数矩阵

```
In[4] := A=RandomReal[{0,1},{3,4}]
```

```
In[5] := Dimensions[A,1] (* 矩阵的行数 *)
```

# 矩阵运算

设  $A, B$  是矩阵,  $a$  是标量,  $b$  是向量

$A+B, A-B, A+a, A-a$	矩阵加减
$A \cdot B$	普通矩阵乘积
$A * B, A / B$	对于分量相乘或相除
$\text{Det}[A], \text{Inverse}[A]$	行列式与矩阵的逆
$\text{Transpose}[A]$	转置 (不取共轭)
$\text{ConjugateTranspose}[A]$	共轭转置
$\text{MatrixRank}[A]$	矩阵的秩
$\text{Eigenvalues}[A]$	特征值
$\text{Eigenvectors}[A]$	特征向量
$\text{Eigensystem}[A]$	特征值和特征向量
$\text{LinearSolve}[A, b]$	解线性方程组 $Ax=b$

# 符号计算

---

- 多项式运算 (略)
- 解代数方程 (略)
- 解微分方程 (略)
- 计算极限
- 计算导数
- 计算积分

# 计算极限

<code>Limit[f,x-&gt;a]</code>	$x$ 趋向于 $a$ 时 $f$ 的极限
<code>Limit[f,x-&gt;a,Direction-&gt;1]</code>	左极限
<code>Limit[f,x-&gt;a,Direction-&gt;-1]</code>	右极限

例：计算  $\lim_{x \rightarrow 0} \frac{\sin x}{x}$  和  $\lim_{x \rightarrow 0^-} \frac{1}{x}$

```
In[1]:= Limit[Sin[x]/x, x->0]
```

```
In[2]:= Limit[1/x, x->0, Direction->1]
```



# 计算导数

$D[f, x]$	计算 $f$ 关于 $x$ 的导数
$D[f, x, y]$	二重偏导数
$D[f, x, y, \dots]$	多重偏导数
$D[f, \{x, n\}]$	$n$ 重导数

例：计算  $\frac{d^2}{dx^2}(\sin x \tan x)$  和  $\frac{\partial}{\partial x} \frac{\partial}{\partial y} \sin(x^2 y^3)$

```
In[1] := D[Sin[x]*Tan[x], {x, 2}]
```

```
In[2] := D[Sin[x*x*y^3], x, y]
```

# 计算积分

<code>Integrate[f,x]</code>	计算 $f$ 关于 $x$ 的不定积分（省略积分常数）
<code>Integrate[f,x,y]</code>	二重积分（积分顺序自右向左）
<code>Integrate[f,{x,a,b}]</code>	定积分，积分区间为 $[a,b]$
<code>NIntegrate[f,{x,a,b}]</code>	数值积分

例：计算  $\int 3ax^2 dx$  和  $\iint 2x + y dx dy$

```
In[1]:= Integrate[3*a*x*x, x]
```

```
In[2]:= Integrate[2*x+y, x, y] (* 注意积分顺序 *)
```

例：计算  $\int_0^{\pi} \sin \sin x dx$  的精确值和近似值

```
In[3]:= Integrate[Sin[Sin[x]], {x,0,Pi}]
```

```
In[4]:= NIntegrate[Sin[Sin[x]], {x,0,Pi}]
```

# 作图

---

## ■ 二维曲线做图

- 函数作图
- 参数方程作图
- 极坐标作图
- 散点图

## ■ 三维做图

- 三维曲线作图
- 三维曲面函数作图
- 三维曲面参数方程作图

# 二维函数作图

■ 曲线方程： $f = f(x)$ ,  $x \in [a, b]$

<code>Plot[f, {x, a, b}]</code>	画 $f$ 关于 $x$ 的图像，绘图区间为 $[a, b]$
<code>Plot[f, {x, a, b}, options-&gt;val]</code>	带绘图选项
<code>Plot[{f1, f2, ...}, {x, a, b}]</code>	在一个绘图区域做多个函数的图像

## ● 常用作图选项

<code>AspectRatio</code>	图形的高宽比，缺省为 0.618
<code>AxesLabel</code>	坐标轴标注，缺省为不加标注
<code>AxesStyle</code>	坐标轴的粗细
<code>PlotLabel</code>	标题，缺省为不加
<code>PlotPoint</code>	作图时计算的点数，缺省为 25
<code>BaseStyle</code>	坐标轴属性：颜色，标注字体大小
<code>PlotStyle</code>	指定线型，颜色，粗细（最常用的选项）

# 作图选项

## ● BaseStyle 和 PlotStyle 取值

线型	Dashed, Dotted, DotDashed
粗细	Thin, Thick, Thickness[w] ( $w \in [0,1]$ , 通常小于 0.1)
颜色	Red, Blue, White, Green, Yellow, Black, ... RGBColor[r,g,b]
字体大小	FontSize->大小

```
In[1]:= Plot[Sin[x^2]/(x+1), {x,0,2*Pi}]
```

```
In[2]:= Plot[Sin[x^2]/(x+1), {x,0,2*Pi},  
  AxesLabel->{x,f[x]},  
  AxesStyle->Thick,  
  PlotLabel->y==sin[x]/(x+1),  
  BaseStyle->{RGBColor[0.6,0.6,0.2], FontSize->15},  
  PlotStyle->{Blue,Thick,Dashed}]
```

# 作图选项

## ● 更多作图选项

Filling->Axis	填充曲线与 $x$ 轴所围区域
Ticks->None	取消刻度
Axes->None	取消坐标轴
Frame->True	加边框
FrameLabel	边框标注

```
In[3]:= Plot[Sin[x^2]/(x+1), {x,0,2*Pi},  
           Filling->Axis,  
           Frame->True,  
           BaseStyle->{FontSize->15},  
           PlotStyle->{Blue,Thick,Dashed}]
```

可输入命令 `Options[Plot]` 查看 `Plot` 的所有作图选项

# 参数方程作图

---

■ 曲线方程:  $x = x(t), y = y(t), t \in [a, b]$

---

`ParametricPlot[{x,y},{t,a,b}]`

参数方程, 绘图区域  $t \in [a, b]$

---

```
In[3]:= ParametricPlot[{Sin[t],Sin[2*t]}, {t,0,2*Pi},  
BaseStyle->{FontSize->15},  
PlotStyle->{Blue,Thick,Dashed}]
```

● 该函数也可以同时绘制多个图像

---

`ParametricPlot[{{x1,y1},{x2,y2},...}, {t,a,b}]`

---

# 参数方程作图

---

```
In[4]:= ParametricPlot[{{2*Cos[t],2*Sin[t]},  
    {2*Cos[t],Sin[t]}, {Cos[t],2*Sin[t]},  
    {Cos[t],Sin[t]}}, {t,0,2*Pi},  
    PlotStyle->{{Blue,Thick}, {Red,Dashed},  
        {Green,Dotted}, {Black,DotDashed}}]
```



# 极坐标方程作图

■ 曲线极坐标方程:  $r = r(\theta)$ ,  $\theta \in [\alpha, \beta]$

`PolarPlot[r, {θ, α, β}]`

极坐标方程作图, 可带绘图选项

```
In[1]:= PolarPlot[2, {t, 0, 2*Pi}]
```

```
In[2]:= PolarPlot[t, {t, 0, 3*Pi}]
```

```
In[3]:= PolarPlot[{0.5, 0.5+1/24*Sin[12*t]}, {t, 0, 2*Pi}]
```

```
In[4]:= PolarPlot[Exp[Cos[t-Pi/2]]-2*Cos[4*(t-Pi/2)]  
+Sin[(t-Pi/2)/12]^5, {t, 0, 36*Pi},  
Axes->None]
```

# 散点图

- 给的数据点:  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$

<code>ListPlot[{{x1,y1},{x2,y2},...}]</code>	散点图
<code>ListPlot[{y1,y2,...}]</code>	数据点: $\{(1, y_1), (2, y_2), \dots\}$
<code>ListPlot[list, Joined-&gt;True]</code>	画过数据点的连线
<code>ListLinePlot[list]</code>	带连线的散点图
<code>ListPolarPlot[list]</code>	极坐标下的散点图

- 常用选项

<code>PointSize[a]</code>	点的大小, 通过 <code>PlotStyle</code> 来设置
<code>PlotMarkers-&gt;{g,size}</code>	点的形状和大小 (可通过面板 Palettes 输入)

# 散点图

---

```
In[1]:= list=Table[{x,Sin[x]},{x,-Pi,Pi,Pi/10}];  
ListPlot[list,BaseStyle->{FontSize->15},  
PlotStyle->{PointSize[0.03]}]
```

```
In[2]:= ListPlot[list,PlotMarkers->{"♠", 25}]
```

```
In[3]:= ListPolarPlot[{Sin[Range[0, 12*Pi, Pi/20]],  
0.8*Ssin[Range[0, 12*Pi, Pi/20]]},  
PlotMarkers->{"@",12},  
Joined->True, Axes->None]
```

# 三维曲线作图

■ 三维曲线方程:  $x = x(t)$ ,  $y = y(t)$ ,  $z = z(t)$ ,  $t \in [a, b]$

`ParametricPlot3D[{x,y,z},{t,a,b},选项]`

三维曲线绘图

```
In[1]:= ParametricPlot3D[{t*Sin[t],t*Cos[t],t},  
  {t,0,100},PlotStyle->{Blue,Thickness[0.005]}]
```

# 三维曲面作图

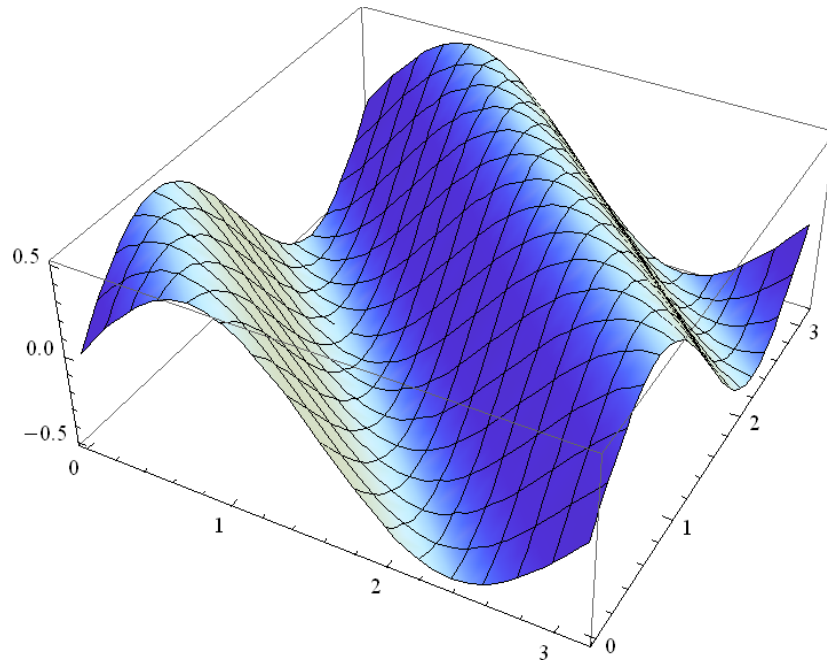
■ 曲面方程:  $z = z(x, y), x \in [a, b], y \in [c, d]$

```
Plot3D[z, {x, a, b}, {y, c, d}]
```

三维曲面绘图

例: 画  $z(x, y) = \sin(x + y) \cos(x + y)$  的立体图

```
In[1]:= Plot3D[Sin[x+y]*Cos[x+y], {x, 0, Pi}, {y, 0, Pi}]
```

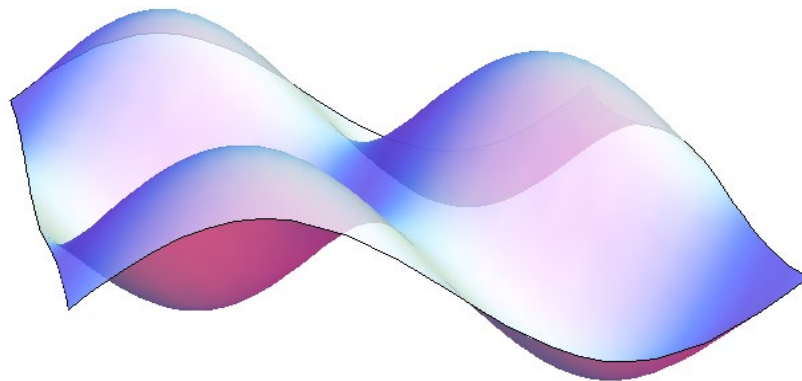


# 三维曲面作图

## ● 常用选项

<code>Boxed-&gt;False</code>	去除立体方框
<code>Mesh-&gt;None</code>	去除网格
<code>BoxRatios</code>	立体方框的比例, 缺省为 <code>{1,1,0.4}</code>
<code>Opacity[数字]</code>	透明度, 缺省不透明, 通过 <code>PlotStyle</code> 来设置

```
In[2]:= Plot3D[Sin[x]*Cos[2*y], {x,-3,3}, {y,-2,2},  
Boxed->False, Axes->None, Mesh->False,  
PlotStyle -> {Opacity[0.8]}]
```



# 三维曲面参数方程作图

## ■ 三维曲面参数方程:

$$x = x(u, v), \quad y = y(u, v), \quad z = z(u, v), \quad u \in [a, b], \quad v \in [c, d]$$

---

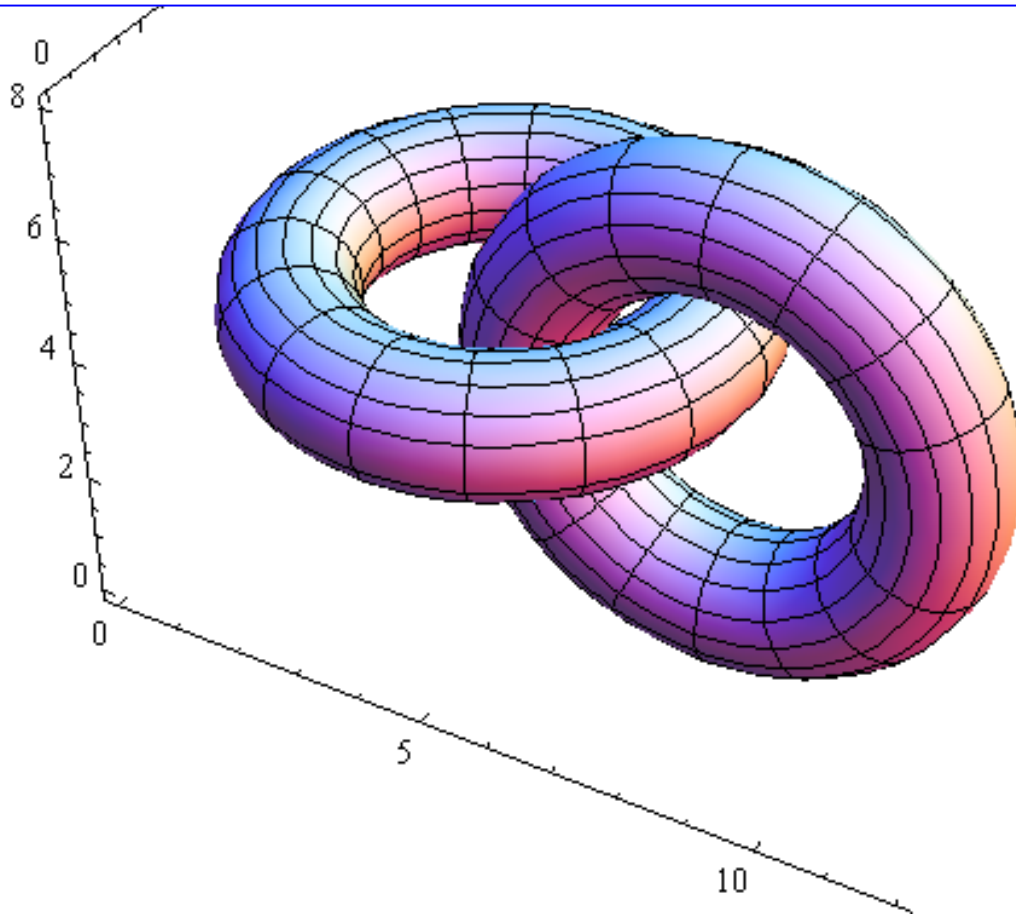
`ParametricPlot3D[{x,y,z},{u,a,b},{v,c,d},选项]`

---

```
In[1]:= ParametricPlot3D[{3*Sec[u]*Cos[v],3*Sec[u]*Sin[v],  
                        5*Tan[u]},  
                        {u,-Pi/3,Pi/3},{v,0,2*Pi},  
                        Boxed->False]
```

# 三维做图演示

```
In[1]:= ParametricPlot3D[{  
  {4+(3+Cos[v])*Sin[u],4+(3+Cos[v])*Cos[u],4+Sin[v]},  
  {8+(3+Cos[v])*Cos[u],3+Sin[v],4+(3+Cos[v])*Sin[u]}},  
  {u,0,2*Pi},{v,0,2*Pi},Boxed->False]
```





# 图形组合

## ■ 图形组合：将多个图形放在一个绘图区域中

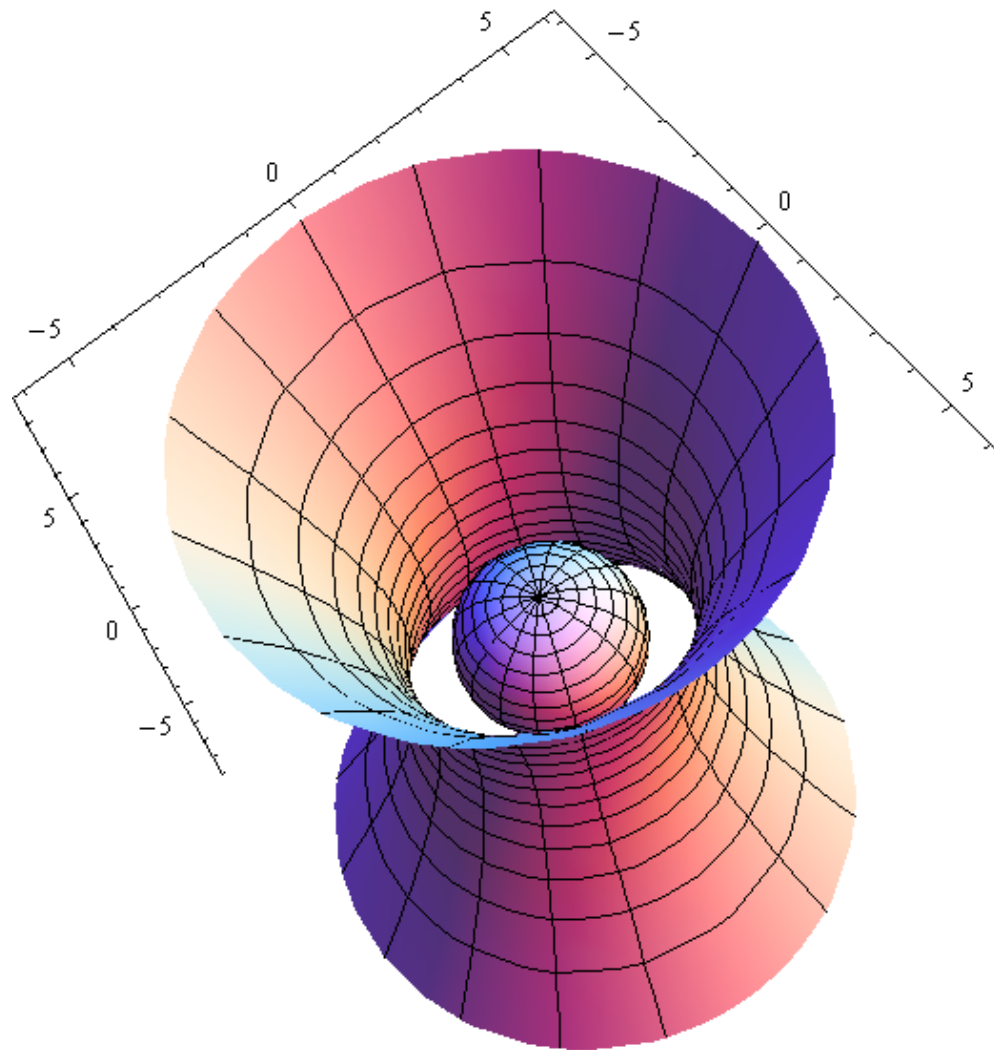
<code>Show[pic]</code>	显示图形表达式
<code>Show[pic,options-&gt;values]</code>	按指定的选项显示图形表达式
<code>Show[pic1,pic2,...]</code>	将多个图形放在一个绘图区域中

```
In[1]:= pic1=ParametricPlot3D[{3*Sec[u]*Cos[v],  
    3*Sec[u]*Sin[v],5*Tan[u]},  
    {u,-Pi/3,Pi/3},{v,0,2*Pi},Boxed->False];  
(* 单页双曲面 *)
```

```
In[2]:= pic2=ParametricPlot3D[{2*Sin[u]*Cos[v],  
    2*Sin[u]*Sin[v], 2*Cos[u]},  
    {u,0,Pi},{v,0,2*Pi}]; (* 球面 *)
```

```
In[3]:= Show[pic1,pic2] (* 组合两个图形 *)
```

# 图形组合



# 动画做图演示

---

```
In[1]:= Manipulate[Plot[Sin[n*x]/(x+1),{x,0,2*Pi},  
  PlotStyle->{RGBColor[(10-n)/10,n/10,0],  
  Thickness[0.01]}],  
  {n,1,20,1}]
```