

Mathematica 参考

jypan@mathecnu

一、Mathematica 基础.....	1
1 基本语法规则.....	1
1.1 Mathematica 中的一些常用符号.....	1
1.2 命令和语句的执行方式.....	1
2 Mathematica 中的数与基本运算.....	1
2.1 Mathematica 常用内部常数.....	1
2.2 Mathematica 中数的类型.....	1
2.3 Mathematica 中的精确数与近似数.....	2
2.4 关于数的一些基本运算.....	2
2.5 不同进制的数的转换.....	2
3 基本运算.....	2
3.1 算术运算.....	2
3.2 关系运算与逻辑运算.....	3
4 变量与表达式.....	3
4.1 变量.....	3
4.2 表达式.....	3
4.3 变量操作.....	3
5 字符串.....	3
5.1 字符串操作.....	4
6 列表.....	4
6.1 列表的生成.....	4
6.2 列表分量.....	4
6.3 列表修改.....	5
6.4 列表运算.....	5
7 矩阵.....	6
7.1 矩阵的生成.....	6
7.2 特殊矩阵.....	6
7.3 矩阵分量.....	6
7.4 矩阵运算.....	7
8 函数.....	8
8.1 常用初等函数.....	8
8.2 随机函数.....	9
8.3 自定义函数.....	9
二、符号计算.....	10
1 多项式运算.....	10
2 代数方程求解.....	10
3 微分方程求解.....	10
4 级数运算.....	10
5 计算极限.....	10
6 计算导数.....	10
7 计算积分.....	10
三、作图.....	11
1 二维曲线做图.....	11
1.1 函数作图.....	11
1.2 参数方程作图.....	12

1.3 极坐标方程作图.....	12
1.4 散点作图.....	12
2 三维曲线做图.....	13
2.1 参数方程作图.....	13
3 三维曲面做图.....	13
3.1 函数作图.....	13
3.2 参数方程作图.....	14
3.3 球坐标作图.....	14
4 图形的重现与组合.....	14
5 动画.....	15
四、程序设计.....	15

一、Mathematica 基础

1 基本语法规则

- 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
- ⑤ 输入和输出标识符：`In[n]:=` 和 `Out[n]=`

1.1 Mathematica 中的一些常用符号

<code>()</code>	运算的结合	<code>%</code>	最后一次的计算结果
<code>f[]</code>	函数取值	<code>%%</code>	倒数第二次计算结果
<code>{}</code>	列表	<code>%%(k)</code>	倒数第 k 次计算结果
<code>[[k]]</code>	列表的分量	<code>%k</code>	第 k 次计算结果，即 <code>Out[k]</code>
<code>!cmd</code>	执行 Dos 命令	<code>(*comments*)</code>	注解
<code>!!filename</code>	显示文件的内容	<code>Print[x,y,...]</code>	屏幕输出函数
<code>?name</code>	显示系统变量、命令或函数的相关信息		
<code>??name</code>	显示系统变量、命令或函数的全部信息		

1.2 命令和语句的执行方式

- ① 命令或语句输入结束后按 `Shift + Enter`（简称执行键）即可执行；
- ② 运行多个语句：输入全部语句后再按执行键；
- ③ 如果不需要显示运行结果：在语句后面加分号；
- ④ 命令（语句）分隔符：回车或分号；
- ⑤ 长语句可以分多行输入，直接按回车键换行即可。

2 Mathematica 中的数与基本运算

Mathematica 中的数分普通的数和内部常数。

2.1 Mathematica 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...
EulerGamma	Euler 常数 $\lim_{n \rightarrow \infty} \left(1 + \frac{1}{2} + \dots + \frac{1}{n} - \ln n \right) = 0.577217\dots$
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

2.2 Mathematica 中数的类型

Mathematica 的数有：整数、有理数、实数和复数：

- ① 整数：Integer，具有任意长度的精确数；
- ② 有理数：Rational，用最简分数表示，有理数是精确数，输入方式“分子/分母”；
- ③ 实数：Real，是指除了整数和有理数之外的所有实数。与一般高级语言不同的是，这里

的实数分任意精度和机器精度；

- ④ 复数：Complex，带虚数单位 I，实部和虚部可以是整数、有理数或实数。

2.3 Mathematica 中的精确数与近似数

- ① 精确数：整数、有理数、数学常数以及函数在自变量取整数、有理数、数学常数时的函数值；如 12, 2/3, Sin[3], Pi
 ② 近似数：带有小数点的数；如 12.0, 3.14159
 ③ 如果参与运算的数带有小数点，则运算结果通常为近似数，显示的时候带 6 位有效数字，实际计算时具有机器精度。

2.4 关于数的一些基本运算

N[x,n]	提取 x 的近似值，带 n 位有效数字
N[x]	提取 x 的近似值，机器精度
Floor[x]	取整：不大于 x 的最大整数
Ceil[x]	取整：不小于 x 的最小整数
Round[x]	取整：四舍五入
Precision[expr]	查看表达式计算结果的精度
IntegerPart[x]	提取 x 的整数部分
FractionalPart[x]	提取 x 的小数部分

2.5 不同进制的数的转换

b[^]xxxx	输入一个 b 进制数， $2 \leq b \leq 36$ ，并输出相应的十进制数
BaseForm[x,b]	十进制数 x 的 b 进制表示
IntegerString[x,b]	十进制数 x 的 b 进制表示，写成字符串形式
FromDigits["str"]	从数字字符串中构造一个整数
FromDigits[list]	从十进制数字列表中构造一个整数
FromDigits[list,b]	从 b 进制数字列表中构造一个整数
IntegerDigits[x,b]	十进制数 x 的 b 进制数字列表

3 基本运算

3.1 算术运算

+	-	加、减	运算优先级
*	/	乘、除	
.		矩阵乘积	
^		幂	
+=	-- *= /=	运算后赋值	由低到高
++	--	自加 1、自减 1	
!	!!	阶乘、双阶乘	
()		运算的结合	

不同类型的数参与运算，其运算结果的类型为：

- ① 如果有复数，则计算结果为复数类型；
 ② 如果没有复数，但有实数，则计算结果为实数类型；

- ③ 如果没有复数和实数，但有分数，则计算结果为有理数类型；
- ④ 如果只有整数，则计算结果为整数类型或有理数类型。

3.2 关系运算与逻辑运算

< <= > >= == !=	比较运算
x1 == x2 == x3 == ...	全部相等
x1 != x2 != x3 != ...	两两不等
a<b<c	a<b && b<c
&& ! xor	逻辑运算

- ① 关系运算的优先级低于算术运算；逻辑运算的优先级低于关系运算；
- ② 当关系表达式成立时，取值为 True；不成立时，取值为 False；如果不能确定，则原样输出，表示取值为非真非假；
- ③ 多个比较运算符可以同时使用，如：ac 等价于 (a<b) && (b>c)

4 变量与表达式

4.1 变量

- ① 变量名通常以字母开头，后面跟字母、数字、下划线，不能含空格；
- ② 变量名的长度不限；
- ③ 变量名区分大小写；
- ④ 变量名中也可以包含希腊字母或中文，如“数学”，“码头”；
- ⑤ 变量名不能以数字开头，否则将理解成数字与变量的乘积；如 3ab 等价于 3*ab
- ⑥ 用户定义变量时，建议使用小写字母，以免与系统内置函数重名；
- ⑦ 变量不必事先声明，其类型可以不断改变，取决于其所存数据的类型。

4.2 表达式

- ① Mathematica 中几乎所有对象都是表达式；
- ② 基本表达式有：算术表达式，关系表达式，逻辑表达式；

4.3 变量操作

x=expr	变量赋值
Unset[x] 或 x=.	清除变量的值
Print[x1,x2,...]	打印变量的值
Clear[x1,x2,...]	清除变量
expr/.{x->a,y->b,...}	变量替换（注：变量本身没有被赋值）

建议：使用变量前先清除其中的内容

Mathematica 中的变量名还可以用作代数中的数学符号：在命令或程序中出现的变量名，如果该变量名所代表的变量没有被赋值，则它就作为一个数学符号参与数学公式的推导和运算；如果该变量被赋值了，则用该变量所赋的值参与对应数学公式的推导和运算。因此，如果用户在做符号运算时，所使用的符号变量已经被赋值，则会出现意想不到的错误。

5 字符串

字符串：用双引号括起来的字符序列。

5.1 字符串操作

<code>Characters[str]</code>	转化为字符列表
<code>StringJoin[str1,str2,...]</code>	字符串合并
<code>str1 <> str2 <> ...</code>	字符串合并
<code>StringLength[str]</code>	打印变量的值
<code>StringSplit[str]</code>	根据空白字符分隔字符串
<code>ToExpression[str]</code>	转化为表达式
<code>ToString[expr]</code>	将表达式转化为字符串

6 列表

- ① 是 Mathematica 的基本对象，可用来表示集合，数组等；
- ② 可分为标准列表和稀疏列表；
- ③ 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；
稀疏列表：通常由 `SparseArray` 来定义；
- ④ 列表中的元素可以是不同类型的任意 Mathematica 对象；
- ⑤ 列表可以嵌套，形成多维列表，如矩阵；
- ⑥ 当函数作用在列表上时，采用的是数组运算，即作用在每个分量上。

6.1 列表的生成

<code>{x1,x2,...}</code>	枚举法，直接输入
<code>Array[f,n]</code>	生成一维列表{f[1],f[2],...,f[n]}
<code>Array[f,{n1,n2,...}]</code>	生成多维列表，f 为函数
<code>Range[a,b,h]</code>	生成一个等差数列构成的列表：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, ... , expr}
<code>Table[expr,{i,a,b,h}]</code>	{expr i 在 Range[a,b,h]中变化}
<code>Table[expr,{i,list}]</code>	{expr i 在 list 中变化}，expr 为通项公式
<code>RandomInteger[range,n]</code>	生成 n 个伪随机整数列表，range 表示范围
<code>RandomReal[range,n]</code>	生成 n 个伪随机实数列表，n 的缺省值为 1

6.2 列表分量

<code>list[[k]]</code>	列表 list 的第 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（嵌套列表）
<code>list[[i,j]]</code>	第 i 个分量的第 j 个分量（嵌套列表）
<code>list[{{i,j,...}}]</code>	{list[[i]],list[[j]], ... }
<code>First[list], Last[list]</code>	第一个和最后一个分量
<code>Take[list,k], Take[list,-k]</code>	前 k 个和最后 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>Take[list,{i,j,h}]</code>	{list[[i]],list[[i+h]], ... } (步长为 h)
<code>list[[i;;j;;h]]</code>	同上，h 可以省略，缺省值为 1

6.3 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]]</code> , <code>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>Delete[list,{{i1,j1, ...}, {i2,j2,...}, ...}]</code>	删除多个分量
<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 <code>list</code> 的最前面
<code>PrependTo[list,x]</code>	将 x 插入到 <code>list</code> 的最前面, 并将结果赋给 <code>list</code>
<code>Append[list,x]</code>	将 x 插入到 <code>list</code> 的最后面
<code>AppendTo[list,x]</code>	将 x 插入到 <code>list</code> 的最前面, 并将结果赋给 <code>list</code>

6.4 列表运算

<code>Sort[list]</code>	从小到大排列
<code>Sort[list,p]</code>	用排序函数 p 对元素排序
<code>Sort[list,f]</code>	排序方式根据 f 应用到每个元素的结果
<code>Ordering[list]</code>	列表 <code>list</code> 中元素按 <code>Sort[list]</code> 顺序排列的位置
<code>Ordering[list,n]</code>	列表中前 n 个元素按 <code>Sort[list]</code> 顺序排列的位置
<code>Ordering[list,-n]</code>	列表中后 n 个元素按 <code>Sort[list]</code> 顺序排列的位置
<code>Ordering[list,n,p]</code>	列表中前 n 个元素按 <code>Sort[list,p]</code> 排列的位置
<code>Length[list]</code>	列表 <code>list</code> 中元素的个数
<code>Reverse[list]</code>	将列表中的元素反过来排列
<code>Permutations[list]</code>	列表中所有元素的所有排列
<code>Permutations[list,n]</code>	列表中不超过 n 个元素的所有排列
<code>Permutations[list,{n}]</code>	列表中 n 个元素的所有排列
<code>MemberQ[list,a]</code>	判断 a 是否在列表中
<code>Subset[list]</code>	列表的所有子集
<code>Subset[list,n]</code>	列表的所有不超过 n 个元素的子集
<code>Subset[list,{n}]</code>	列表的所有含 n 个元素的子集
<code>Flatten[list]</code>	将列表压缩为一维列表
<code>Partition[a,n]</code>	将列表拆分成若干长度为 n 的子列表
<code>Partition[a,{n1,n2,...}]</code>	将列表拆分成 $n_1 \times n_2 \times \dots$ 大小的字块
<code>Union[list1,list2,...]</code>	合并列表并排序, 删除重复元素
<code>Intersection[a1,a2,...]</code>	计算交集并排序, 删除重复元素
<code>Apply[Plus,list]</code>	计算列表中所有元素的和

Apply[Times,list]	计算列表中所有元素的乘积
Total[list]	计算列表中所有元素的和

7 矩阵

向量是一维列表，矩阵是二维列表。

7.1 矩阵的生成

矩阵可以通过 Array, Table 等函数生成。

Array[函数名,变量取值范围]	
Array[f,n]	生成长度为 n 的向量 {f[1],f[2],...,f[n]}
Array[f,{m,n}]	生成 m×n 的矩阵，元素为 f[i,j]
Table[通项公式,{循环范围},...]	
Table[expr,{i,a,b,h}]	生成向量 {expr i=a:h:b}
Array[expr,{i,a1,b1,h1},{j,a2,b2,h2}]	生成 m×n 的矩阵

注：Table 中的循环范围的一般表示方法为

{i,a,b,h}	i 从 a 到 b，步长为 h，最后一项不超过 b
{i,a,b}	步长缺省为 1
{i,b}	首项缺省为 1
{k}	重复 k 次

7.2 特殊矩阵

IdentityMatrix[n]	n 阶单位矩阵
ConstantArray[c,{m,n}]	常数矩阵: $a_{ij} = c$
DiagonalMatrix[list]	以列表 list 中的元素为对角线的对角矩阵
HilbertMatrix[n]	n 阶 Hilbert 矩阵
HilbertMatrix[{m,n}]	m×n 的 Hilbert 矩阵
ToeplitzMatrix[n]	n 阶 Toeplitz 矩阵: $a_{ij} = i - j + 1$
ToeplitzMatrix[list]	以 list 的元素为第一列的对称 Toeplitz 矩阵
ToeplitzMatrix[list1,list2]	指定第一列和第一行的 Toeplitz 矩阵
HankelMatrix[n]	n 阶 Hankel 矩阵: $a_{ij} = i - j + 1$
HankelMatrix[list]	以 list 的元素为第一列的 Hankel 矩阵
HankelMatrix[list1,list2]	指定第一列和最后一行的 Hankel 矩阵
RotationMatrix[θ]	平面逆时针旋转 θ 所对应的 2 阶矩阵
RotationMatrix[θ,w]	空间绕 w 逆时针旋转 θ 所对应的 3 阶矩阵

7.3 矩阵分量

A[[i,j]]	提取 a_{ij}
A[[All,j]]	提取第 j 列
A[[i,All], A[[i]]]	提取第 i 行
Take[A,{i1,i2,...},{j1,j2,...}]	提取一个子矩阵

7.4 矩阵运算

MatrixForm[A]	按矩阵形式输出 A
TableForm[A]	按表格形式输出 A
Length[v]	元素个数
Dimensions[A]	矩阵的维数（行数和列数）
Dimensions[A,k]	矩阵的前 k 重维数
VectorQ[v]	是否为向量
MatrixQ[A]	是否为矩阵

A+B, Plus[A,B]	矩阵或向量相加
A-B, Subtract[A,B]	矩阵或向量相减
-A, Minus[A]	负矩阵或负向量
A+a, A-a	矩阵（向量）每个分量都与数相加
A.B	矩阵普通乘积
x.y, Dot[x,y]	向量内积
Cross[x,y]	向量外积
MatrixPower[A,n]	矩阵的幂
MatrixExp[A]	矩阵的指数函数
A*B, Times[A,B]	对应分量相乘（数组运算）
A/B, Divide[A,B]	对应分量相除（数组运算）
A^n, Power(A,n)	对应分量的幂（数组运算）
Det[A]	行列式
Inverse[A]	矩阵的逆
Transpose[A]	矩阵转置
ConjugateTranspose[A]	共轭转置
MatrixRank[A]	矩阵的秩
Eigenvalues[A]	特征值
Eigenvectors[A]	特征向量
Eigensystem[A]	特征值和特征向量
Norm[A]	矩阵的 2 范数
Norm[A,1], Norm[A,Infinity]	1 范数和无穷范数
Norm[x,p]	向量的 p 范数, $p \geq 1$, 缺省为 2 范数
Tr[A]	矩阵的迹
LinearSolve[A,b]	解线性方程组 $Ax = b$
NullSpace[A]	矩阵的零空间的一组基
Normalize[x]	向量单位化
Orthogonalize[A]	将矩阵 A 的行向量标准正交化
LUdecomposition[A]	LU 分解
Choleskydecomposition[A]	Cholesky 分解
Hessenbergdecomposition[A]	Hessenberg 化

JordanDecomposition[A]	Jordan 标准型
QRDecomposition[A]	QR 分解
SchurDecomposition[A]	Schur 分解
SingularValueDecomposition[A]	SVD 分解

8 函数

Mathematica 提供了超过 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 系统内置函数的函数名一般使用数学中的英文单词，只要输入相应的函数名，就可以方便地使用这些函数。内部函数既有数学中常用的函数，又有工程中用的特殊函数。

用户也可以自己定义新的函数。

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 当函数名分为几段时，每一段的头一个字母大写，后面的字母用小写，如：ArcSin[x]
- ③ 查看某个函数的用法：??函数名，如 ??Sin

Mathematica 函数调用方式

- ① 标准方式：函数名[变量列表]，如：Sin[Pi/3], Mod[5,2]
- ② 后缀方式：变量//函数名，如：Pi/3//Sin
- ③ 前缀方式：函数名@变量，如：Sin@(Pi/3)
- ④ 中缀方式：变量~函数名~变量，如：5~Mod~2

注：使用前缀方式时，函数只作用在@后面的第一个变量上，如 Sin@Pi/3 等价于 Sin[Pi]/3，正确使用方法为 Sin@(Pi/3)；而使用后缀方式时，函数作用在前面的所有表达式上，如 1-Pi/3//Sin 等价于 Sin[1-Pi/3]，正确使用方法为 1-(Pi/3//Sin)。

8.1 常用初等函数

Abs[x]	绝对值
Sign[x]	符号函数
Power[x,y]	幂函数 x^y
Sqrt[x]	平方根
Exp[x]	以 e 为底的指数函数
Log[x], Log[b,x]	以 e 和 b 为底的对数函数
Factorial[n]	$n!$
Factorial2[n]	$n!!$
GCD[n1,n2,...]	最大公约数
GCD[list]	列表中所有数的最大公约数
LCM[n1,n2,...], LCM[list]	最小公倍数
Max[x1,x2,...], Max[list]	求最大值
Min[x1,x2,...], Min[list]	求最小值
Re[x], Im[x]	提取实部和虚部
Conjugate[x]	取共轭
Arg[x]	辐角
Mod[m,n]	m 除以 n 的余数

Quotient[m,n]	m 除以 n 的整数商
Sin[x], Cos[x], ...	三角函数
ArcSin[x], ArcCos[x], ...	反三角函数
Sinh, Cosh, ...	双曲函数
ArcSinh, ArcCosh, ...	反双曲函数
Prime[k]	第 k 个素数
PrimeQ[n]	判断 n 是否为素数
Binomial[n,m]	二项式系数 C_n^m

8.2 随机函数

Random[]	生成[0,1]内的一个随机实数
Random[Real, a]	生成[0,a]内的一个随机实数
Random[Real, {a,b}]	生成[a,b]内的一个随机实数
Random[Integer]	随机给出整数 0 或 1
Random[Integer, {a,b}]	生成[a,b]内的一个随机整数
Random[Complex]	生成一个[0,1]×[0,1]内的一个随机复数

8.3 自定义函数

函数名[自变量名 1_, 自变量名 2_, ...]:= 表达式

- ① 这里函数名与变量名的规定相同
- ② 方括号中的每个自变量名后都要有一个下划线“_”
- ③ 中间的“:=”为定义号
- ④ 表达式中一般都含有自变量

f[x_]:=2*x-3	定义一个一元函数
f[x_,y_]:=Log[x/y]-Power[x,y]	定义一个二元函数
Clear[f]	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。

注：函数与符号表达式的区别！

二、符号计算

1 多项式运算

略

2 代数方程求解

略

3 微分方程求解

略

4 级数运算

略

5 计算极限

<code>Limit[f,x->a]</code>	计算极限 $\lim_{x \rightarrow a} f(x)$ ，这里 f 是符号表达式
<code>Limit[f,x->a,Direction->1]</code>	左极限
<code>Limit[f,x->a,Direction->-1]</code>	右极限

6 计算导数

<code>D[f,x]</code>	计算 $f'(x)$ ，这里 f 是符号表达式
<code>D[f,x,y]</code>	二重偏导数 $\frac{\partial}{\partial x} \frac{\partial}{\partial y} f(x)$
<code>D[f,x,y,...]</code>	多重偏导数
<code>D[f,{x,n}]</code>	n 重导数 $f^{(n)}(x)$
<code>D[f,{x,y,...}]</code>	计算梯度 $\left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \dots \right]$

7 计算积分

<code>Integrate[f,x]</code>	计算 $\int f(x) dx$ ，省略积分常数
<code>Integrate[f,x,y]</code>	二重积分 $\iint f(x,y) dydx$ （自右向左）
<code>Integrate[f,x,y,...]</code>	多重积分
<code>Integrate[f,{x,a,b}]</code>	定积分 $\int_a^b f(x) dx$
<code>Integrate[f,{x,a,b},{y,c,d},...]</code>	多重定积分
<code>Integrate[f,{x,y,...}]</code>	计算梯度 $\int_a^b f(x) dx$
<code>NIntegrate[f,{x,a,b},{y,c,d},...]</code>	数值积分

三、作图

1 二维曲线做图

1.1 函数作图

已知曲线方程: $y = f(x), x \in [a, b]$

<code>Plot[f, {x, a, b}]</code>	作 $f(x)$ 的图形, 绘图区间 $[a, b]$
<code>Plot[f, {x, a, b}, options]</code>	带绘图选项 (后面详细介绍各选项)
<code>Plot[{f1, f2, ...}, {x, a, b}]</code>	同时画多个图形, 也可以带选项

常用绘图选项

<code>AspectRatio</code>	图形的高宽比, 缺省约为 6.18, 如: <code>AspectRatio->1</code>
<code>AxesLabel</code>	坐标轴标注, 缺省不加标注, 如: <code>AxesLabel->{x, f[x]}</code>
<code>PlotLabel</code>	标题, 缺省没有标题, 如: <code>PlotLabel->{y==f[x]}</code>
<code>PlotPoint</code>	作图时取的样本点个数, 缺省为 25, 如: <code>PlotPoint->50</code>
<code>PlotRange</code>	指定 y 的范围
<code>AxesStyle</code>	坐标轴属性 (含标签和刻度), 如颜色, 粗细等, 可以取多个值
<code>BaseStyle</code>	指定基本样式
<code>PlotStyle</code>	图形的属性, 如颜色, 线型, 粗细等

AxesStyle, BaseStyle 和 PlotStyle 的取值

线型	<code>Dashed, Dotted, DotDashed, Dashing[{w1, w1, ...}]</code>
粗细	<code>Thin, Thick, Thickness[w]</code> ($w \in [0, 1]$, 通常小于 0.1)
颜色	<code>Red, Blue, Black, White, Green, Yellow, Gray, Cyan, Magenta, Brown, Orange, Pink, Purple, LightRed, LightBlue, ...</code> <code>Transparent</code>
颜色	<code>RGBColor[r, g, b]</code> ($r, g, b \in [0, 1]$)
字体大小	<code>FontSize->大小</code> , 如 <code>FontSize->15</code>

作图举例:

```
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}]
```

更多选项

<code>Filling</code>	填充, 取值可以是 <code>Axis, Bottom, Top, ...</code>
<code>Background->颜色</code>	设置背景颜色
<code>Ticks->None</code>	取消刻度
<code>Axes->None</code>	不显示坐标轴
<code>Frame->True</code>	加上边框
<code>FrameLabel</code>	边框标注

WorkingPrecision	画图时内部计算使用的精度
-------------------------	--------------

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

1.2 参数方程作图

已知曲线参数方程: $x = x(t), y = y(t), t \in [a, b]$

ParametricPlot[{x,y},{t,a,b}]	参数方程作图, 可带选项
ParametricPlot[{{x1,y1},{x2,y2},...},{t,a,b}]	同时画多个图形

```
In[1]:= ParametricPlot[{Sin[t],Sin[2*t]},{t,0,2*Pi},
           BaseStyle->{FontSize->15},
           PlotStyle->{Blue,Thick,Dashed}]
```

```
In[2]:= 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}}]
```

1.3 极坐标方程作图

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

PolarPlot[r,{θ,α,β}]	极坐标方程作图, 可带选项
-----------------------------	---------------

```
In[1]:= PolarPlot[5,{t,0,2*Pi}, PlotStyle->{Blue,Thick}]
```

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

```
In[3]:= PolarPlot[Exp[Cos[t-Pi/2]]-2*Cos[4*(t-Pi/2)]
           +Sin[(t-Pi/2)/12]^5,{t,0,36*Pi},Axes->None]
```

1.4 散点作图

已知数据点: $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$

ListPlot[{{x1,y1},{x2,y2},...}]	散点图, 可带选项
ListLinePlot[{y1,y2,...}]	数据点为 $(1, y_1), (2, y_2), \dots, (n, y_n)$
ListPlot[list,Joined->True]	画通过数据点的折线
ListPolarPlot[list]	在极坐标下的散点图
ListLinePlot[list]	带连线的散点图

常用选项

PointSize[size]	点的大小, 需通过 PlotStyle 来设置
PlotMarkers->{marker,size}	设置点的类型和大小, 可通过面板输入特殊字符

```
In[1]:= list=Table[{Cos[2*k*Pi/5],Sin[2*k*Pi/5]},{k,3,18,3}];
           ListLinePlot[list,AspectRatio->1,PointSize->Large]
```

```

In[2]:= ListPolarPlot[Table[t, {t, 0, 2*Pi, Pi/20}]]
        PolarPlot[t, {t, 0, 2*Pi}]
In[3]:= ListPolarPlot[{Sin[Range[0, 12*Pi, Pi/20]],
                      0.8*Sin[Range[0, 12*Pi, Pi/20]]},
                      Joined→True, Axes→None]

In[4]:= ListPolarPlot[{Sin[Range[0, 12*Pi, Pi/20]],
                      0.8*Sin[Range[0, 12*Pi, Pi/20]]},
                      PlotMarkers→{"@", 12},
                      Joined→True, Axes→None]

```

2 三维曲线做图

2.1 参数方程作图

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

<code>ParametricPlot3D[{x,y,z},{t,a,b}]</code>	三维曲线作图
<code>ParametricPlot3D[{{x1,y1,z1},{x2,y2,z2},...},{t,a,b}]</code>	同时画多个图

常用绘图选项

<code>Boxed->False</code>	去除立体方框
<code>BoxRatios</code>	立体方框的比例, 缺省值为 <code>{1,1,0.4}</code>

3 三维曲面做图

3.1 函数作图

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

<code>Plot3D[z,{x,a,b},{y,c,d},options]</code>	三维曲面作图, 也可以同时画多个图
--	-------------------

常用绘图选项

<code>Boxed->False</code>	去除立体方框
<code>BoxRatios</code>	立体方框的比例, 缺省值为 <code>{1,1,0.4}</code>
<code>Mesh->None</code>	去除网格, 其它常用取值有 <code>All, Full</code>
<code>MeshStyle</code>	设置网格元素的属性, 如颜色, 透明度等
<code>RegionFunction</code>	通过函数指定绘图区域
<code>BoundaryStyle</code>	图形边界属性 (可与 <code>RegionFunction</code> 联合画两个曲面的交线)
<code>Opacity[a]</code>	设置透明度, $a \in [0, 1]$

```

In[1]:= Plot3D[Sin[x+y]*Cos[x+y], {x, 0, Pi}, {y, 0, Pi},
              Mesh→None, Boxed→False]

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

In[3]:= Plot3D[{x^2+y^2, 4-2*x^2-y^2}, {x, -2, 2}, {y, -2, 2},
              PlotStyle→{{Opacity[0.6], Red}, {Opacity[0.6], Blue}},
              RegionFunction→Function[{x, y, z}, x^2+y^2≤z≤4-2*x^2-y^2],

```

`BoundaryStyle→{Red,Thickness[0.01]}`

(* 两个抛物面及其交线 *)

3.2 参数方程作图

已知曲面参数方程: $x = x(u, v), y = y(u, v), z = z(u, v), u \in [a, b], v \in [c, d]$

<code>ParametricPlot3D[{x,y,z},{u,a,b},{v,c,d}]</code>	三维曲面参数方程作图
--	------------

注: 也可以同时画多个图

```
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]
```

```
In[2]:= 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]
```

3.3 球坐标作图

直角坐标与球坐标关系:

$$x = r \sin \theta \cos \varphi, y = r \sin \theta \sin \varphi, z = r \cos \theta$$

$$r = \sqrt{x^2 + y^2 + z^2}, \theta = \arctan \frac{\sqrt{x^2 + y^2}}{z} = \arccos \frac{z}{r}, \varphi = \arctan \frac{y}{x}$$

<code>SphericalPlot3D[r,{θ,α1,β1},{φ,α2,β2}]</code>	球坐标作图
<code>SphericalPlot3D[{r1,r2,...},{θ,α1,β1},{φ,α2,β2}]</code>	同时画多个图

```
In[1]:= SphericalPlot3D[{1,2,3},{u,0,Pi},{v,0,3*Pi/2}]
```

```
In[2]:= SphericalPlot3D[v,{u,0,Pi},{v,0,2*Pi},
BoxRatios→{1,1,1}]
```

4 图形的重现与组合

<code>Show[pic]</code>	显示图形表达式
<code>Show[pic,options→values]</code>	添加选项
<code>Show[pic1,pic2,...]</code>	将多个图形放在一个绘图区域中
<code>GraphicsGrid[{{p11,p12,...},{p21,p22,...},...}]</code>	按矩阵方式显示图形

```
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 *)
```

```
In[4]:= pic=Table[Plot[Sin[x+t],{x,0,2*Pi}],{t,0,3}];
```

```
In[5]:= GraphicsGrid[{{pic[[1]],pic[[2]]},{pic[[3]],pic[[4]]}}]
```


5 动画

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

四、程序设计

待续