

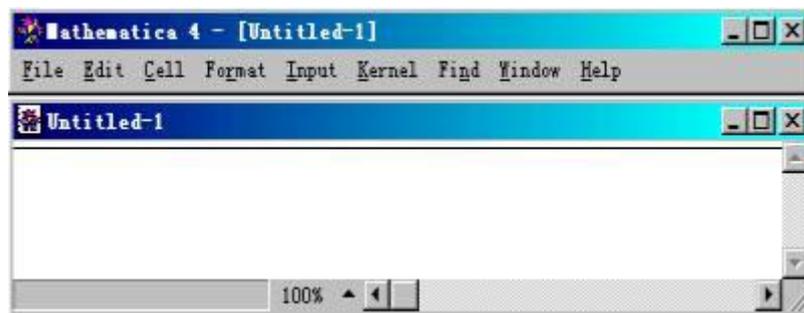
第 1 章 Mathematica 概述	
1. 运行和启动	介绍如何启动 Mathematica 软件，如何输入并运行命令
2. 表达式的输入	介绍如何使用表达式
3. 帮助的使用	如何在 mathematica 中寻求帮助。
第 2 章 Mathematica 的基本量	
1. 数据类型和常量	mathematica 中的数据类型和基本常量
2. 变量	变量的定义，变量的替换，变量的清除等
3. 函数	函数的概念，系统函数，自定义函数的方法
4. 表	表的创建，表元素的操作，表的应用
5. 表达式	表达式的操作
6. 常用符号	经常使用的一些符号的意义
第 3 章 Mathematica 的基本运算	
1. 多项式运算	多项的四则运算，多项式的化简等
2. 方程求解	求解一般方程，条件方程，方程数值解以及方程组的求解
3. 求积求和	求积与求和
第 4 章 函数作图	
1. 二维函数作图	一般函数的作图，参数方程的绘图。
2. 二维图形元素	点，线等图形元素的使用
3. 图形样式	图形的样式，对图形进行设置
4. 图形的重绘和组合	重新显示所绘图形，将多个图形组合在一起。
5. 三维图形的绘制	三维图形的绘制，三维参数方程的图形，三维图形的设置。
第 5 章 微积分的基本操作	
1. 函数的极限	如何求函数的极限
2. 导数与微分	如何求函数的导数，微分。
3. 定积分与不定积分	如何求函数的不定积分和定积分，以及数值积分。
4. 多变量函数的微分	如何求多元函数的偏导数，微分
5. 多变量函数的积分	如何计算重积分
6. 无穷级数	无穷级数的计算，敛散性的判断
第 6 章 微分方程的求解	
1. 微分方程的解	微分方程的求解
2. 微分方程的数值解	如何求微分方程的数值解

值解	
第 7 章 Mathematica 程序设计	
1. 模块	模块的概念和定义方法
2. 条件结构	条件结构的使用和定义方法
3. 循环结构	循环结构的使用
第 8 章 Mathematica 中的常用函数	
1. 运算符和一些特殊符号，系统常数	常用的和不常用一些运算符，和系统定义的一些常量及其意义
2. 代数运算	表达式相关的一些运算函数
3. 解方程	和方程求解有关的一些操作
4. 微积分相关函数	关于求导，积分，泰勒展开等相关的函数
5. 多项式函数	多项式的相关函数
6. 随机函数	能产生随机数的函数函数
7. 数值函数	和数值处理相关的函数，包括一些常用的数值算法
8. 表相关函数	创建表，表元素的操作，表的操作函数
9. 绘图函数	二维绘图，三维绘图，绘图设置，密度图，图元，着色，图形显示等函数
10. 流程控制函数	

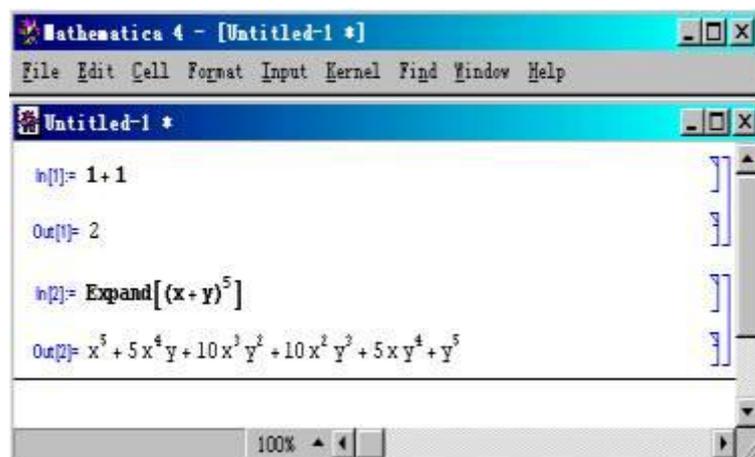
1.1.1 Mathematica 的启动和行

Mathematica 是美国 Wolfram 研究公司生产的一种数学分析型的软件，以符号计算见长，也具有高精度的数值计算功能和强大的图形功能。

假设在 Windows 环境下已安装好 Mathematica 4.0, 启动 Windows 后, 在“开始”菜单的“程序”中单击  Mathematica 4, 就启动了 Mathematica 4.0, 在屏幕上显示如图的 Notebook 窗口, 系统暂时取名 Untitled-1, 直到用户保存时重新命名为止



输入 $1+1$, 然后按下 $\text{Shif}+\text{Enter}$ 键, 这时系统开始计算并输出计算结果, 并给输入和输出附上次序标识 $\text{In}[1]$ 和 $\text{Out}[1]$, 注意 $\text{In}[1]$ 是计算后才出现的; 再输入第二个表达式, 要求系统将一个二项式展开, 按 $\text{Shift}+\text{Enter}$ 输出计算结果后, 系统分别将其标识为 $\text{In}[2]$ 和 $\text{Out}[2]$. 如图



在 Mathematica 的 Notebook 界面下, 可以用这种交互方式完成各种运算, 如函数作图, 求极限、解方程等, 也可以用它编写像 C 那样的结构化程序。在 Mathematica 系统中定义了许多功

能强大的函数，我们称之为内建函数 (built-in function)，直接调用这些函数可以取到事半功倍的效果。这些函数分为两类，一类是数学意义上的函数，如：绝对值函数 $Abs[x]$ ，正弦函数 $Sin[x]$ ，余弦函数 $Cos[x]$ ，以 e 为底的对数函数 $Log[x]$ ，以 a 为底的对数函数 $Log[a, x]$ 等；第二类是命令意义上的函数，如作函数图形的函数 $Plot[f[x], \{x, xmin, xmax\}]$ ，解方程函数 $Solve[eqn, x]$ ，求导函数 $D[f[x], x]$ 等。



必须注意的是：

Mathematica 严格区分大小写，一般地，内建函数的首写字母必须大写，有时一个函数名是由几个单词构成，则每个单词的首写字母也必须大写，如：求局部极小值函数 $FindMinimum[f[x], \{x, x0\}]$ 等。第二点要注意的是，在 Mathematica 中，函数名和自变量之间的分隔符是用方括号 “[]”，而不是一般数学书上用的圆括号 “()”，初学者很容易犯这类错误。

如果输入了不合语法规则的表达式，系统会显示出错信息，并且不给出计算结果，例如：要画正弦函数在区间 $[-10, 10]$ 上的图形，输入 $plot[Sin[x], \{x, -10, 10\}]$ ，则系统提示“可能有拼写错误，新符号 ‘plot’ 很像已经存在的符号 ‘Plot’”，实际上，系统作图命令 “Plot” 第一个字母必须大写，一般地，系统内建函数首写字母都要大写。再输入 $Plot[Sin[x], \{x, -10, 10\}]$ ，系统又提示缺少右方括号，并且将不配对的括号用蓝色显示，如图

```
Mathematica 4 - [Untitled-1 *]
File Edit Cell Format Input Kernel Find Window Help

Untitled-1 *
In[4]:= plot[Sin[x], {x, -10, 10}]

General::spell1 :
Possible spelling error: new symbol name "plot" is
similar to existing symbol "Plot".

Out[4]= plot[Sin[x], {x, -10, 10}]

In[5]:= Plot[Sin[x], {x, -10, 10}]

Syntax::bktncp :
Expression "Plot[Sin[x], {x, -10, 10}" has no closing "]".

Plot[Sin[x], {x, -10, 10}]

100%
```

一个表达式只有准确无误，方能得出正确结果。学会看系统出错信息能帮助我们较快找出错误，提高工作效率。完成各种计算后，点击 File->Exit 退出，如果文件未存盘，系统提示用户存盘，文件名以“.nb”作为后缀，称为 Notebook 文件。以后想使用本次保存的结果时可以通过 File->Open 菜单读入，也可以直接双击它，系统自动调用 Mathematica 将它打开。

1.1.2 表达式的输入

Mathematica 提供了多种输入数学表达式的方法。除了用键盘输入外，还可以使用工具样或者快捷方式键入运算符、矩阵或数学表达式。

1. 数学表达式二维格式的输入

Mathematica 提供了两种格式的数学表达式。形如 $x/(2+3x)+y*(x-w)$ 的称为一维格式，形如 $\frac{x}{2+3x} + \frac{y}{x-w}$ 的称为二维格式。

你可以使用快捷方式输入二维格式，也可用基本输入工具栏输入二维格式。下面列出了用快捷方式输入二维格式的方法

按键	数学运算	数学表达式
x Ctrl+ / 2	分式	$\frac{x}{2}$
Ctrl+ ^ 2	n 次方	x^n
Ctrl +2 x	开 n 次方	$\sqrt[n]{x}$
x Ctrl+_ 2	下标	x_2

如果要取消二维格式输入 按下 Ctrl+SPACE (空格) 例如输入数学表达式

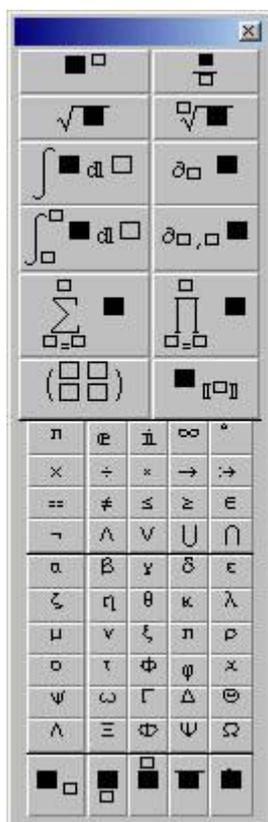
$$(x+1)^4 + \frac{a_1}{\sqrt[2]{2x+1}}$$

可以按如下顺序输入按键:

(, x, +, 1,), Ctrl+^, ->, +, a, Ctrl+_, 1, ->, Ctrl+/, Ctrl+2, 2, x, +, y, ->, ->

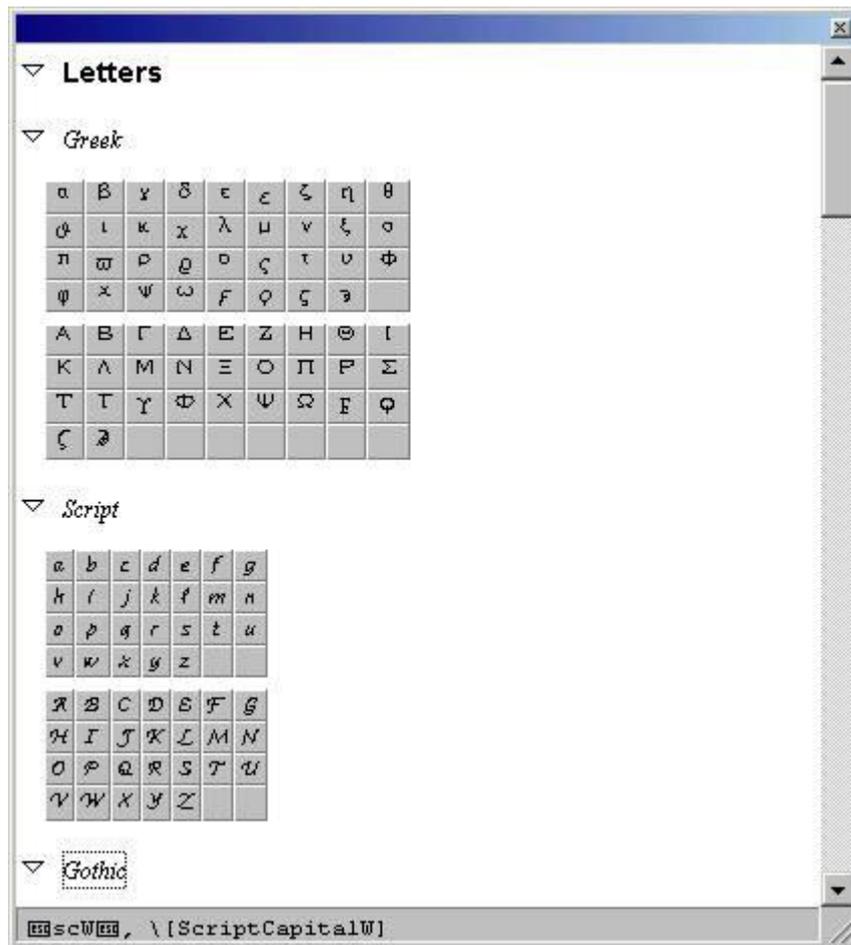
另外也可从 FILE 菜单中激活 **Plaettes->Basic Input** 工具栏, 也可输入, 并且使用工具栏可输入更复杂的数学表达式。

如下图:



2. 特殊字符的输入

Mathematica 还提供了用以输入各种特殊符号的工具样。基本输入 工具样包含了常用的特殊字符 (上图), 只要单击这些字符按钮即可输入。若要输入其它的特殊字符或运算符号, 必须使用从 FILE 菜单中选取 **Complete Characters** 工具栏, 如下图:



单击符号后即可输入。

1.2. Mathematica 的联机帮助系统

用 Mathematica 的过程中，常常需要了解一个命令的详细用法，或者想知系统中是否有完成某一计算的命令，联机帮助系统永远是最详细、最方便的资料库。

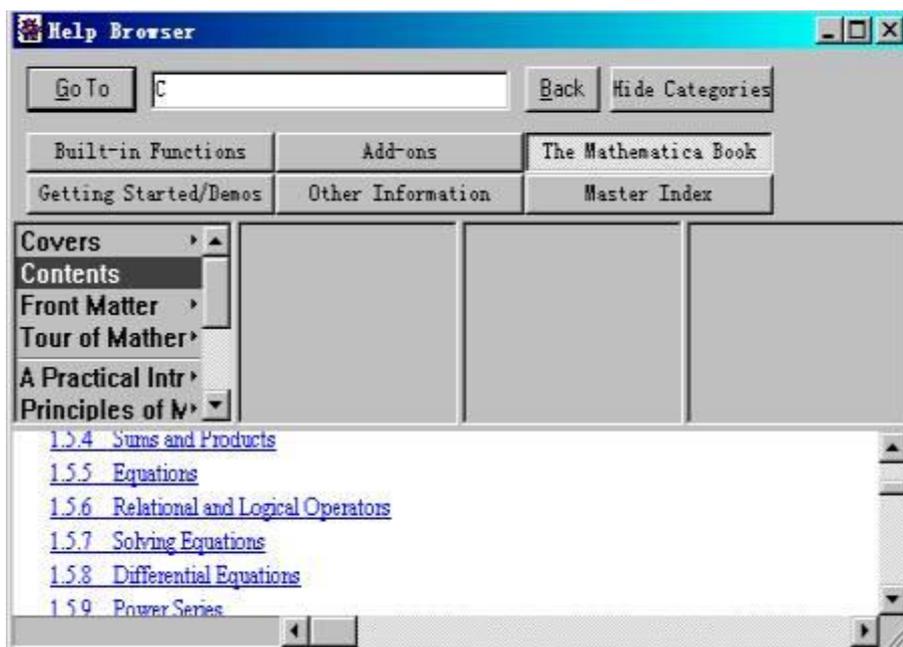
1. 获取函数和命令的帮助

在 Notebook 界面下，用 `?` 或 `??` 可向系统查询运算符、函数和命令的定义和用法，获取简单而直接的帮助信息。例如，向系统查询作图函数 Plot 命令的用法？`Plot` 系统将给出调用 Plot

的格式以及 Plot 命令的功能（如果用两个问号“??”，则信息会更详细一些）。?Plot* 给出所有以 Plot 这四个字母开头的命令

2. Help 菜单

任何时候都可以通过按 F1 键或点击帮助菜单项 Help Browser，调出帮助菜单，如图

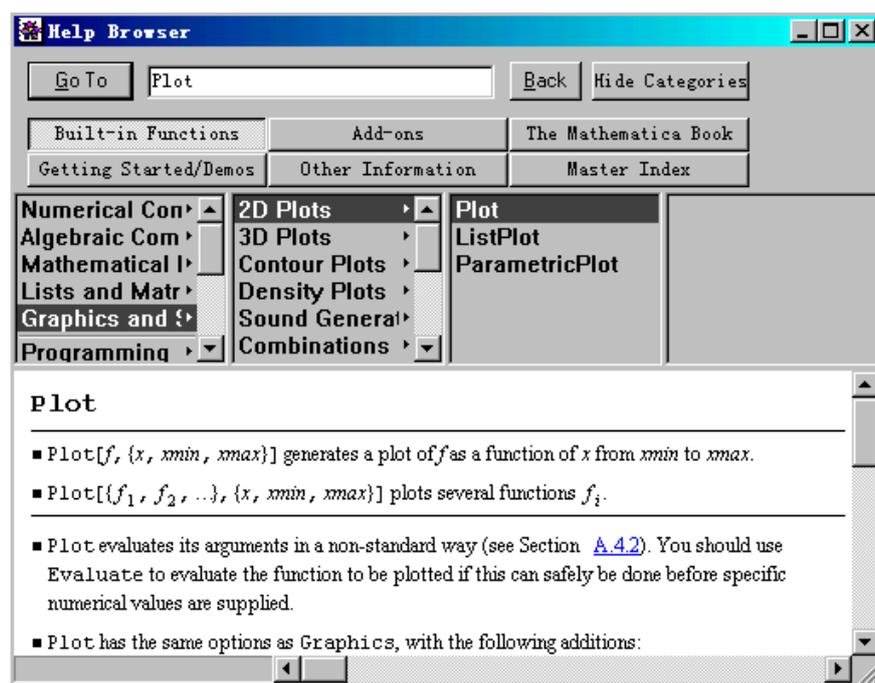


所示，其中的各按钮用途如表所示

Built-in Function	内置函数，按数值计算、代数计算、图形和编程分类存放
Add-ons	有程序包 (Standard Packages) MathLink Library 等内容
The Mathematica Book	一本完整的 Mathematica 使用手册
Getting Started/Demos	初学者入门指南和多种演示
Other Information	菜单命令的快捷键，二维输入格式等
Master Index	按字母命令给出命令、函数和选项的索引表

如果要查找 Mathematica 中具有某个功能的函数，可以通过帮助菜单中的 Mathematica 使用手册，通过其目录索引可以快速定位到自己要找的帮助信息。例如：需要查找 Mathematica

中有关解方程的命令，单击“The Mathematica Book”按钮，再单击“Contents”，在目录中找到有关解方程的节次，点击相应的超链接，有关内容的详细说明就马上调出来了。如果知道具体的函数名，但不知其详细使用说明，可以在命令按钮 Goto 右边的文本框中键入函数名，按回车键后就显示有关函数的定义、例题和相关联的章节。例如，要查找函数 Plot 的用法，只要在文本框中键入 Plot，按回车键后显示如图的窗口，

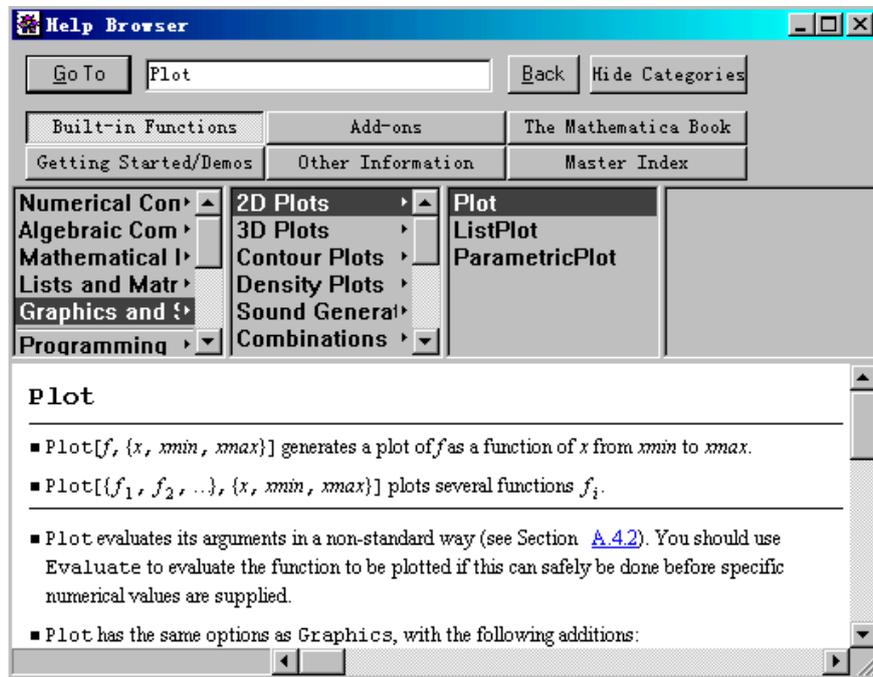


再按回车键，则显示 Plot 函数的详细用法和例题。如果已经确知 Mathematica 中有具有某个功能的函数，但不知具体函数名，可以点击 Built-in Functions 按钮，再按功能分类从粗到细一步一步找到具体的函数，例如，要找画一元函数图形的函数，点击 Built-in Functions ->Graphics and Sound->2D Plots->Plot，找到 Plot 的帮助信息。

如果知道具体的函数名，但不知其详细使用说明，可以在命令按钮 Goto 右边的文本框中键入函数名，按回车键后就显示有关函数的定义、例题和相关联的章节。例如，要查找函数 Plot 的用法，只要在文本框中键入 Plot，按回车键后显示如图 1-5 的窗口，再按回车键，则显示 Plot 函数的详细用法和例题。

如果已经确知 Mathematica 中有具有某个功能的函数，但不知具体函数名，可以点击 Built-in Functions 按钮，再按功能分类从粗到细一步一步找到具体的函数，例如，要找画一元函数图形

的函数，点击 Built-in Functions ->Graphics and Sound->2D Plots->Plot，找到 Plot 的帮助信息。



2.1 数据类型和常数

1 数值类型

在 Mathematic 中，基本的数值类型有四种：整数，有理数、实数和复数

如果你的计算机的内存足够大，Mathematica 可以表示任意长度的精确实数，而不受所用的计算机字长的影响。整数与整数的计算结果仍是精确的整数或是有理数。例如：2 的 100 次方是一个 31 位的整数：

```
In[1]:=2^100
```

```
Out[1]=1267650600228228229401496703205376
```

在 Mathematica 中允许使用分数，也就是用有理数表示化简过的分数。当两个整数相除而又不能整除时，系统就用有理数来表示，即有理数是由两个整数的比来组成如：

```
Ln[2]:=12345/5555
```

```
Out[2]=2469/1111
```

实数是用浮点数表示的，Mathematica 实数的有效位可取任意位数，是一种具有任意精确度的近似实数，当然在计算的时候也可以控制实数的精度。实数有两种表示方法：一种是小数点另外一种是用指数方法表示的。如：

```
ln[3]:=0.239998
```

```
Out[3]=0.23998
```

```
ln[4]:=0.12*10^11
```

```
Out[4]=0.12*10^11
```

实数也可以与整数，有理数进行混合运算，结果还是一个实数。

```
Ln[5]:=2+1/4+0.5
```

```
Out[5]=2.75
```

复数是由实部和虚部组成。实部和虚部可以用整数，实数，有理数表示。在 Mathematica 中，用 i 表示虚数单位如：

```
Ln[6]:=3+0.7i
```

```
Out[6]:=3+0.7i
```

2. 不同类型数的转换

在 Mathematica 的不同应用中，通常对数字的类型要求是不同的。例如在公式推导中的数字常用整数或有理数表示，而在数值计算中的数字常用实数表示。在一般情况下在输出行 Out[n] 中，系统根据输入行 In[n] 的数字类型对计算结果做出相应的处理。如果有一些特殊的要求，就要进行数据类型转换。

在 Mathematica 中的提供以下几个函数达到转换的目的：

<code>N[x]</code>	将 x 转换成实数
<code>N[x, n]</code>	将 x 转换成近似实数，精度为 n
<code>Rationalize[x]</code>	给出 x 的有理数近似值
<code>Rationalize[x, dx]</code>	给出 x 的有理数近似值，误差小于 dx

[举例]

```
In[1]=N[5.3, 20]

Out[1]=1.66666666666666666667

In[2]:=N[%, 10]

Out[2]=1.66666667
```

二行输出是把上面计算的结果变为 10 位精度的数字。%表示上一输出结果。

```
Ln[3]=Rationalize[%]

Out[3]=5/3
```

3. 数学常数

Mathematica 中定义了一些常见的数学常数，这些数学常数都是精确数，例如表示圆周率。

Pi	表示 π =
3.14159.....	
E	自
然对数的底, e=2.71828.....	
Degree	$\pi/180$
i	虚数单位
Infinity	无穷大 ∞
-infinity	负的无穷大 $-\infty$
GoldenRatio	黄金分割数 0.61803

数学常数可用在公式推导和数值计算中。在数值计算中表示精确值：如：

```

n[1]:=Pi^2

Out[1]=  $\pi^2$ 

ln[2]:=Pi^2/N

Out[2]=9.86961

```

4. 数的输出形式

在数的输出中可以使用转换函数进行不同数据类型和精度的转换。另外对一些特殊要求的格式还

可以使用如下的格式函数：

NumberForm[expr, n] 以 n 位精度的实数形式输出实数 expr

`ScientificFormat[expr]` 以科学记数法输出实数 `expr`

`EngineergForm[expr]` 以工程记数法输出实数 `expr`

例如:

```
In[1]:=N[Pi^30,30]
8.21289330402749581586503585434×1014

In[2]:=NumberForm[%,10]

Out[2]//NumberForm=8.212893304×1014
```

下面的函数输出幂值可被 3 整除的实数

```
In[3]=EngineeringForm[%]

Out[6]//EngineeringForm= 821.289×1012
```

2.2 变量

1. 变量的命名

Mathematica 中内部函数和命令都是以大写字母开始的标示符。为了不会与它门混淆，我们自定义的变量应该是以小写字母开始，后跟数字和字母的组合，长度不限。例如：`a12`, `ast`, `aST` 都是合法的，而 `12a`, `z*a` 是非法的。另外在 Mathematica 中的变量是区分大小写的 在 Mathematica 中，变量不仅可以存放一个数值，还可以存放表达式或复杂的算式。

2. 给变量赋值

在 Mathematica 中用等号 `=` 为变量赋值。同一个变量可以表示一个数值，一个数组，一个表达式，甚至一个图形。如：

```
Ln[1]:=x=3
```

```
Out[1]=3
```

```
Ln[2]:=x^2+2x
```

```
Out[2]=15
```

```
Ln[3]:=x=%+1
```

```
Out[3]=16
```

对不同的变量可同时赋不同的值例如：

```
Ln[4]:={u, v, w}={1, 2, 3}
```

```
Out[4]={1, 2, 3}
```

```
Ln[5]:=2u+3v+w
```

```
Out[5]=11
```

对于已定义的变量，当你不再使用它是，为防止变量值的混淆，可以随时用`=.`清除他的值，如果变量本身也要清除用函数`Clear[x]`例如

```
Ln[6]:=u=.
```

```
Ln[7]:=2u+v
```

```
Out[7]=2+2u
```

3. 变量的替换

在给定一个表达式时其中的变量可能取不同的值，这是可用变量替换来计算表达式的不同值。方法为用`expr/.`例如：

```
Ln[1]:=f=x/2+1
```

```

Out[1]= 1 +  $\frac{x}{2}$ 

Ln[2]:=f/. x->1

Out[2]=  $\frac{3}{2}$ 

Ln[3]:=f/. ->2

Out[3]=3

```

如果表达式中有多个变量也可以同时替换方法为例如有两个：

```

expr/. {x->xval, y->yval}

Ln[4]:=(x+y)(x-y)^2/. {x->3, y->1-a}

Out[4]= (4-a)(2+a)^2

```

2.3 函数

1. 系统函数

在 Mathematic 中定义了大量的数学函数可以直接调用，这些函数其名称一般表达了一定的意义，可以帮助我们理解。下面是几个常用的函数：

不比 x 大的最大整数
不比 x 小的最小整数
符号函数
接近 x 的整数
x 绝对值
x1 , x2, x3……. 中的最大值

<code>x1, x2, x3, ...]</code>	<code>x1, x2, x3, ...]</code> 中的最小值
	$0 \sim 1$ 之间的随机函数
<code>al, xmax]</code>	$0 \sim xmax$ 之间的随机函数
<code>al, {xmin, xmax}]</code>	$xmin \sim xmax$ 之间的随机函数
	指数函数
	自然对数函数 $\ln x$
	以 b 为底的对数函数
<code>s[x], Tan[x], Csc[x], Sec[x], Cot[x]</code>	三角函数 (变量是以弧度为单位的)
<code>osh[x], Tanhx[x], Csch[x], Sech[x], Coth[x]</code>	双曲函数
<code>], ArcCoth[x]</code>	双曲函数
	的余数, 余数与 n 的符相同
<code>m, n]</code>	m/n 的整数部分
<code>, n3, ...]</code> 或 <code>GCD[s]</code>	$n1, n2, \dots$ 的最大公约数, s 为一数集合
<code>, ...]</code> 或 <code>LCM[s]</code>	$n1, n2, \dots$ 的最大公倍数, s 为数据集合
	n 的阶程
	n 的双阶程

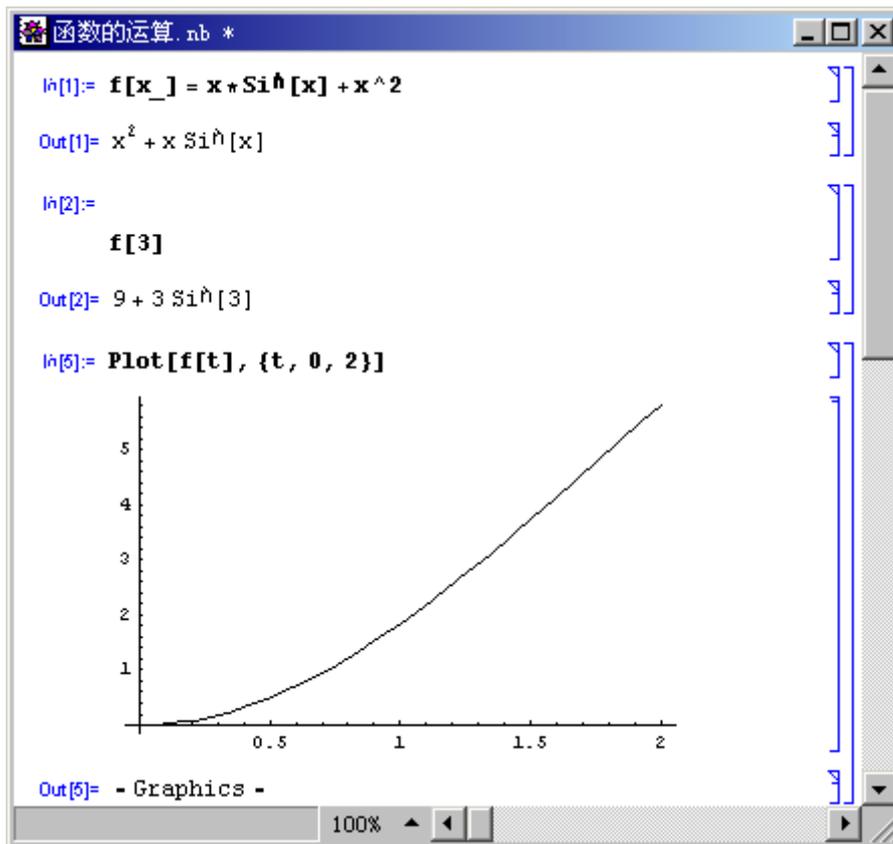
Mathematica 中的函数与数学上的函数有些不同的地方, Mathematica 中函数是一个具有独立功能的程序模块, 可以直接被调用。同时每一函数也可以包括一个, 或多个参数, 也可以没有参数。参数的数据类型也比较复杂。更加详细的可以参看系统的帮助, 了解各个函数的功能和使用方法是学习 Mathematica 软件的基础

2. 函数的定义

(1) 函数的立即定义

立即定义函数的语法如下 `f[x_]=expr` 函数名为 f , 自变量为 x , $expr$ 是表达式。在执行时会把 $expr$ 中的 x 都换为 f 的自变量 x (不是 $x_$)。函数的自变量具有局部性, 只对所在的函数起作用。函数执行结束后也就没有了, 不会改变其它全局定义的同名变量的值。请看下面的例子

定义函数 $f(x)=x*\sin x+x^2$ 对定义的函数我们可以求函数值, 也可绘制它的图形。



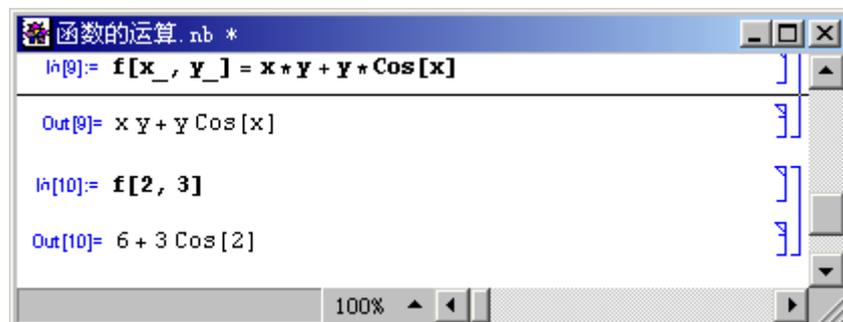
对于定义的函数我们可以使用命令 `Clear[f]` 清除掉而 `Remove[f]` 则从系统中删除该函数。

(2) . 多变量函数的定义

也可以定义多个变量的函数，格式为 `f[x_, y_, z_, ...]=expr` 自变量为 `x, y, z, ...`，相应的 `expr` 中的自变量

会被替换。例如定义函数

$$f(x, y) = xy + y \cos x$$



(3) . 延迟定义函数

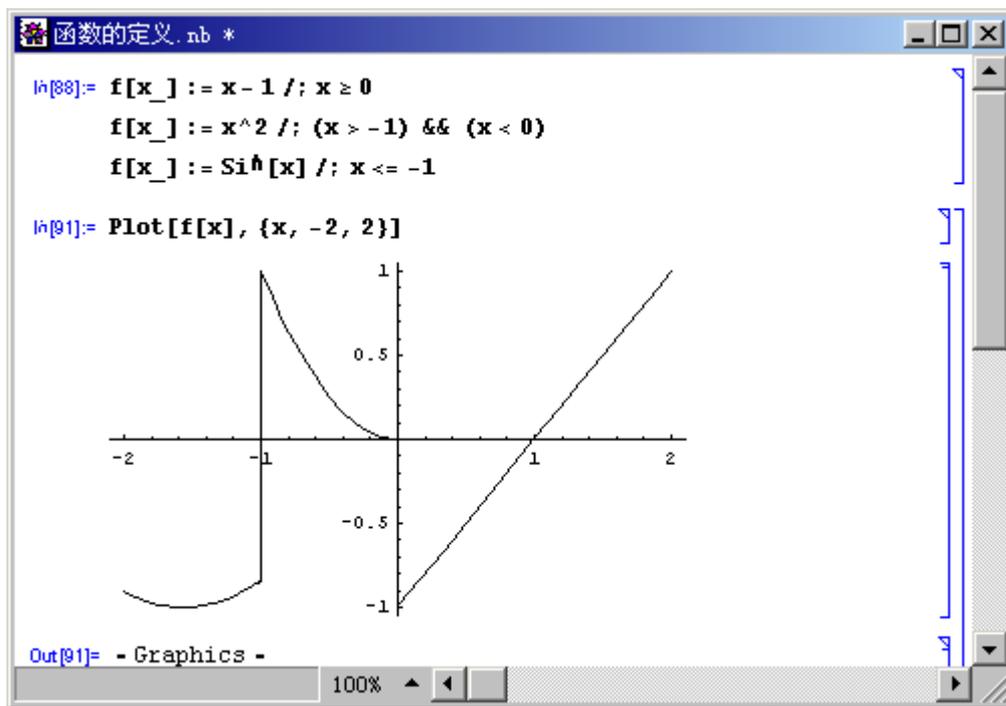
延迟定义函数从定义方法上与即时定义的区别为“=”与“:=”延迟定义的格式为 $f[x_]:=expr$ 其他操作基本相同。那么延迟定义和即时定义的主要区别是什么？即时定义函数在输入函数后立即定义函数并存放在内存中并可直接调用。延时定义只是在调用函数时才真正定义函数。

(4). 使用条件运算符定义和 If 命令定义函数

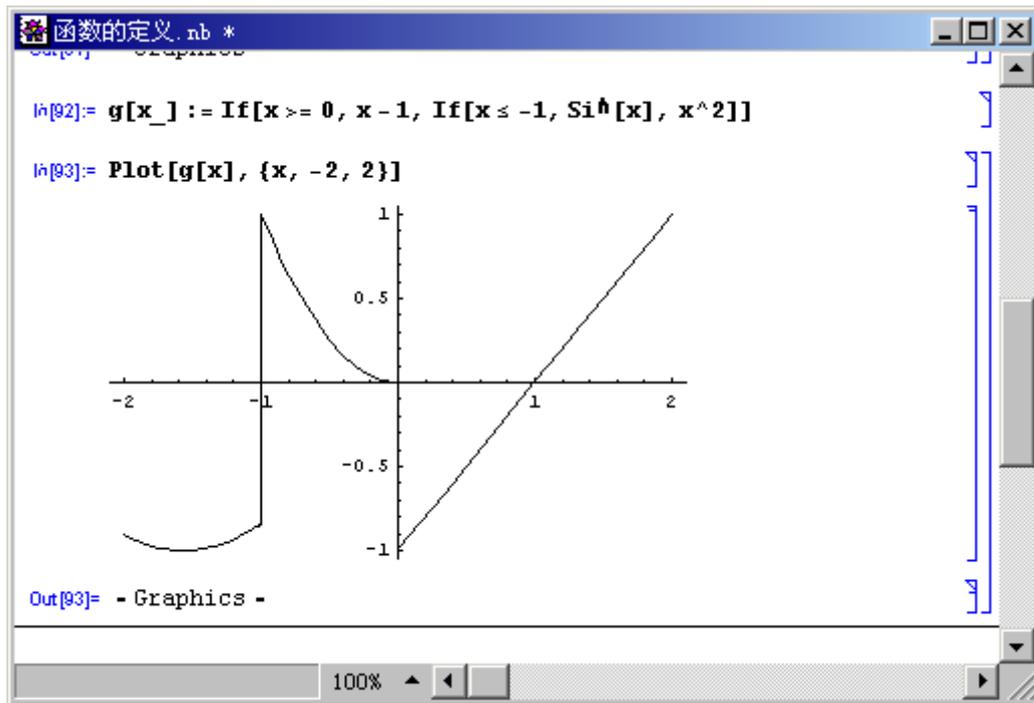
如果要定义如：

$$f(x) = \begin{cases} x-1 & x > 0 \\ x^2 & 0 \geq x > -1 \\ \text{Sin}x & x \leq -1 \end{cases}$$

这样的分段函数应该如何定义，显然要根据 x 的不同值给出不同的表达式。一种办法是使用条件运算符，基本格式为 $f[x_]:=expr/condition$ ，当 $condition$ 条件满足时才把 $expr$ 赋给 f 。下面定义方法，通过图形可以验证所定义函数的正确性



当然使用 If 命令也可以定义上面的函数，If 语句的格式为 If[条件，值 1，值 2] 如果条件成立取“值 1”，否则取“值 2”，下面用 If 语句的定义结果



可以看出用 If 定义的函数 $g(x)$ 和前面函数 $f(x)$ 相同，这里使用了两个 If 嵌套。逻辑性比较强关于其他的条件命令的进一步讨论请看后面的章节。



2.4 表

将一些相互关联的元素放在一起，使它们成为一个整体。既可以对整体操作，也可以对整体中的一个元素单独进行操作。在 Mathematica 中这样的数据结构就称作表 (List)。表主要有三个用法：表 $\{a, b, c\}$ 可以表示一个向量；表 $\{\{a, b\}, \{c, d\}\}$ 可表示一个矩阵。

1. 建表

在表中元素较少时，可以采取直接列表的方式列出表中的元素，如 $\{1, 2, 3\}$ 。请看下面的操作

```
Ln[1]:={1, 2, 3}
```

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

下面是符号表达式的列表

```
Ln[2]:=1+%x+x^%
Out[2]={1+2x, 1+2x+x2, 1+3x+x2}
```

下面是对列表中的表达式对 x 求导

```
Ln[3]:=D[%, x]
Out[3]={2, 2+2x, 3+2x}

Ln[4]:=%/ . x->1
Out[4]={2, 4, 5}
```

如果表中的元素较多时，可以用建表函数进行建表。

```
Table[f, {I, min, max, step}] 以 step 为步长给出 f 的数值表，i 由 min 变到 max，
Table[f, {min, max}]          给出 f 的数值表，I 由 min 变到 max 步长为 1
Table[f, max]                 给出 max 个 f 的表
Table[f, {I, imin, imax}, {j, jmin, jmax}, ...] 生成一个多维表
TableForm[list]               以表格格式显示一个表
Range[n]                      生成一个 {1, 2, ..., n} 的列表
Range[n1, n2, d]              生成 {n1, n1+d, n1+d, ..., n2} 的列表
```

下面给出 x 乘 i 的值的表， i 的变化范围为 $[2, 6]$ ：

```
Ln[1]:=Table[x*i, {i, 2, 6}]
Out[1]={2x, 3x, 4x, 5x, 6x}

Ln[2]:=Table[x^2, {4}]
Out[2]={x2, x2, x2, x2}
```

用 Range 函数生成一个序列数

```
Ln[3]:=Range[10]
```

```
Out[3]={1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

下面这个序列是以步长为 2，范围从 8 到 20

```
Ln[4]:=Range[8, 20, 2]
```

```
Out[4]={8, 10, 12, 14, 16, 18, 20}
```

上面的参数变化都是只有一个，也可制成包括多个参数的表，下面生成一个多维表：

```
Ln[5]:=Table[2i+j, {i, 1, 3}, {j, 3, 5}]
```

```
Out[5]={{5, 6, 7}, {7, 8, 9}, {9, 10, 11}}
```

使用函数 TableForm 可以以表格的方式输出

```
Ln[6]:=TableForm
```

```
Out[6]//TableForm=5 6 7
```

```
7 8 9
```

```
9 10 11
```

2. 表的元素的操作

当 t 表示一个表时， $t[[i]]$ 表示 t 中的第 i 个子表。如果 $t=\{1, 2, a, b\}$ 那么 $t[[3]]$ 表示“a”。如：

```
In[1]:=t=Table[I+2, j{I, 1, 3}, {j, 3, 5}]
```

```
Out[1]={{7, 9, 11}, {8, 10, 12}, {9, 11, 13}}
```

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

```
Out[2]={8, 10, 12}
```

对于表的操作 Mathematica 提供了丰富的函数，详细的可以查阅后面的附录或者系统帮助。

2.5 表达式

1 表达式的含义

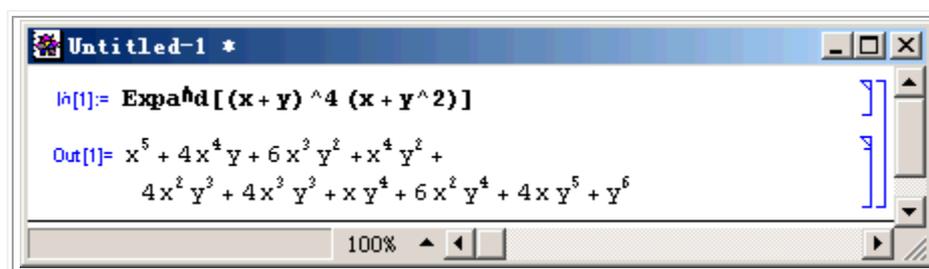
Mathematica 能处理数学公式,表以及图形等多种数据形式。尽管他们从形式上看起来不一样,但在 Mathematica 内部都被看成同种类型,即都把他们当作表达式的形式。Mathematica 中的表达式是由常量、变量、函数、命令、运算符和括号等组成,他最典型的形式是 $f[x, y]$

2. 表达式的表示形式

在显示表达式时,由于需要的不同,有时我们需要表达式的展开形式,有时又需要其因子乘积的形式。在我们计算过程中可能得到很复杂的表达式,这时我们又需要对它们进行化简。常用的处理这种情况的函数。变换表达式表示形式函数

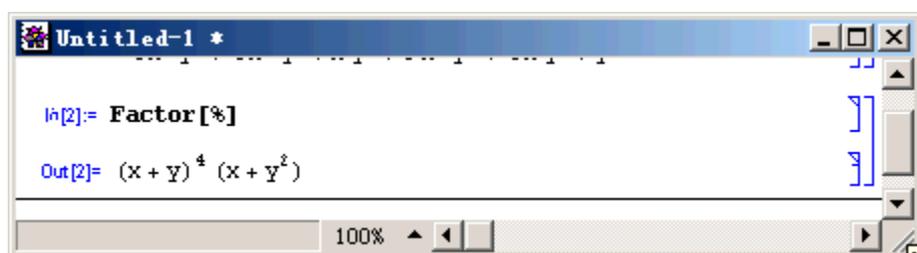
表达式表示形式函数	意义
Expand (expr)	按幂次升高的顺序展开表达式
Factor (expr)	以因子乘积的形式表示表达式
Simplify (expr)	进行最佳的代数运算,并给出表达式的最少项形式

表达式 $(x+y)^4 (x+y^2)$ 展开:



```
Untitled-1 *
In[1]:= Expand[(x+y)^4 (x+Y^2)]
Out[1]:= x^5 + 4x^4Y + 6x^3Y^2 + x^4Y^2 +
         4x^2Y^3 + 4x^3Y^3 + xY^4 + 6x^2Y^4 + 4xY^5 + Y^6
```

还原上面的表达式为因子乘积的形式:



```
Untitled-1 *
In[2]:= Factor[%]
Out[2]:= (x+Y)^4 (x+Y^2)
```

多项式表达式的项数较多，比较复杂，在显示时显得比较杂乱，而且在计算过程中没有必要知道全部的内容；或表达式的项很有规律，没有必要打印全部的表达式的结果，Mathematica 提供了一些命令，可将它缩短输出或不输出

命令	意义
command	执行命令 command，屏幕上不显示结果
expr/Short	显示表达式的一行形式
Short (expr,n)	显示表达式的 n 行形式命令后加分号“;” 不打印结果

将表达式 $(1+x)^{30}$ 展开，并仅显示一行有代表项的式子：

```

多项式.nb *
In[5]:= Expand[(1+x)^30] // Short
Out[5]//Short=
1 + 30 x + 435 x^2 + 4060 x^3 + <<24>> + 435 x^28 + 30 x^29 + x^30
    
```

将上式分成三行的形式展开：

```

多项式.nb *
In[6]:= Short[%, 3]
Out[6]//Short=
1 + 30 x + 435 x^2 + 4060 x^3 + 27405 x^4 + 142506 x^5 + <<19>> +
142506 x^25 + 27405 x^26 + 4060 x^27 + 435 x^28 + 30 x^29 + x^30
    
```

把代数表达式变换到你所需要的形式没有一种固定的模式，一般情况下，最好的办法是进行多次实验，尝试不同的变换并观察其结果，再挑出你满意的表示形式。

3. 关系表达式与逻辑表达式

我们已经知道“=”表示给变量赋值。现在我们来学习一些其它的逻辑与关系算子。关系表达式是最简单的逻辑表达式，我们常用关系表达式表示一个判别条件。例如： $x>0, y=0$ 。关系表达式的一般形式是：**表达式+关系算子+表达式**。其中表达式可为数字表达式、字符表达式或意义更广泛的表达式，如一个图形表达式等。在我们实际运用中，这儿的表达式常常是数字表达式或字符表达式。下面出 Mathematica 中的各种关系算子。

$x==y$	相等
$x!=y$	不相等
$x>y$	大于
$x>=y$	大于或等于
$x<y$	小于
$x<=y$	小于等于
$x==y==z$	都相等
$x!=y!=z$	都不相等
$x>y>z, \text{etc}$	严格递减

给变量 x, y 赋值，输出后以变量的值，如：

```
Ln[1]:=x=2;y=9
Out[1]=9;
Ln[2]:=x>y
Out[2]=false;
```

下面是比较两个表达式的大小

```
Ln[3]:=3^2>y+1
Out[3]=True
```

用一个关系式只能表示一个判定条件，要表示几个判定条件的组合，必须用逻辑运算符将关系表达式组织在一起，我们称表示判定条件的表达式为逻辑表达式。

下面是常用的逻辑运算和它们的意义

!	非
&&	并
	或
Xor	异或
If	条件

LogicalExpand[Expr] 展开逻辑表达式，例如下面的例子说明它们的应用

```
Ln[4]:=3x^2<Y+1&&3^2==y
```

```
Out[4]=false
```

```
Ln[5]:=3x^2+1||3^2==y
```

```
Out[5]=True
```

常用的符号

(term) 圆括号用于组合运算

f[x] 方括号用于函数

{ } 花括号用于列表

[[i]] 双括号用于排序

% 代表最后产生的结果

%% 倒数第二次的算结果

%%(k) 倒数第 k 次的计算结果

%n 例出行 Out[n])的结果 (用时要小心)

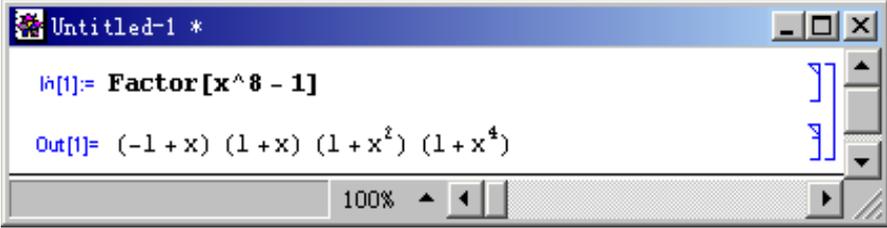
3.1 多项式的表示形式

可认为多项式是表达式的一种特殊的形式,所以多项式的运算与表达式的运算基本一样,表达式中的各种输出形式也可用于多项式的输出。Mathematica 提供一组按不同形式表示代数式的函数。

<code>Expand[poly]</code>	按幂次展开多项式 poly
<code>Expand[poly]</code>	全部展开多项式 poly
<code>ExpandAll[poly]</code>	全部展开多项式 poly
<code>Factor[poly]</code>	对多项式 poly 进行因式分解
<code>FactorTerms[poly, {x, y, ...}]</code>	按变量 x, y, ...进行分解
<code>Simplify[poly]</code>	把多项式化为最简形式
<code>FullSimplify[poly]</code>	把多项式展开并化简
<code>Collect[poly, x]</code>	把多项式 poly 按 x 幂展开
<code>Collect[poly, {x, y, ...}]</code>	把多项式 poly 按 x, y, ... 的幂次展开

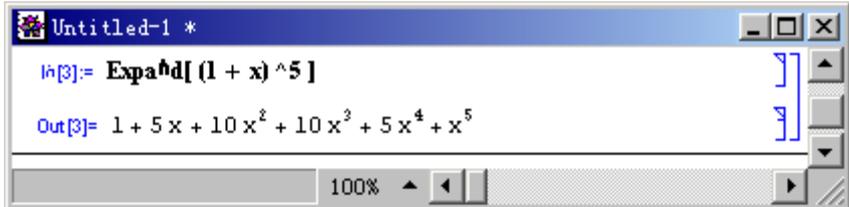
1. 下面是一些例子

(1). 对 x^8-1 进行分解



```
Untitled-1 *
In[1]:= Factor[x^8 - 1]
Out[1]= (-1 + x) (1 + x) (1 + x^2) (1 + x^4)
```

(2). 展开多项式 $(1+x)^5$



```
Untitled-1 *
In[3]:= Expand[(1 + x)^5]
Out[3]= 1 + 5 x + 10 x^2 + 10 x^3 + 5 x^4 + x^5
```

(3). 展开多项式 $(1+x+3y)^4$

```
In[4]:= Expand[(1 + x + 3 y)^4]
Out[4]= 1 + 4 x + 6 x^2 + 4 x^3 + x^4 + 12 y + 36 x y + 36 x^2 y + 12 x^3 y +
54 y^2 + 108 x y^2 + 54 x^2 y^2 + 108 y^3 + 108 x y^3 + 81 y^4
```

(4). 化简 $(2+x)^4(1+x)^4(3+x)^3$

```
In[7]:= Simplify[Expand[(2 + x)^4 (1 + x)^4 (3 + x)^3]]
Out[7]= (3 + x)^3 (2 + 3 x + x^2)^4
```

2. 多项式的代数运算

多项式的运算有加、减、乘、除运算： $+$ ， $-$ ， $*$ ， $/$ 下面通过例子说明。

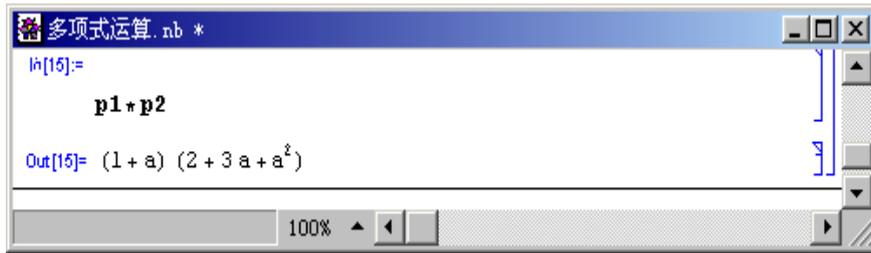
(1). 多项式的加运算 a^2+3a+2 与 $a+1$ 相加（后面例子中也使用这两个多项式运算

```
In[10]:= p1 = a^2 + 3 a + 2; p2 = a + 1; p1 + p2
Out[11]= 3 + 4 a + a^2
```

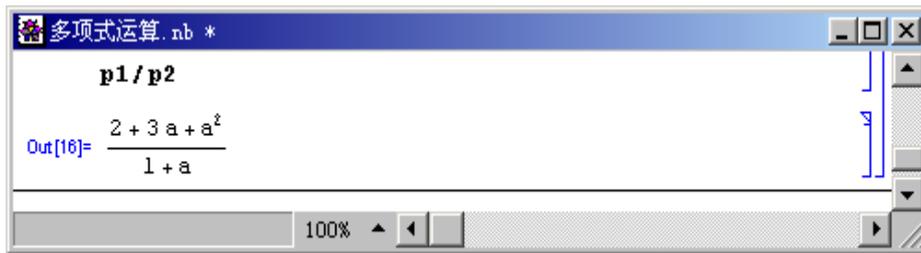
(2). 多项式相减

```
In[14]:= p2 - p1
Out[14]= -1 - 2 a - a^2
```

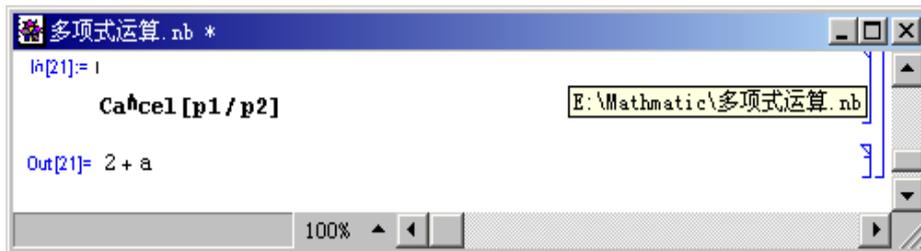
(3). 多项式相乘



(4). 多项式相除



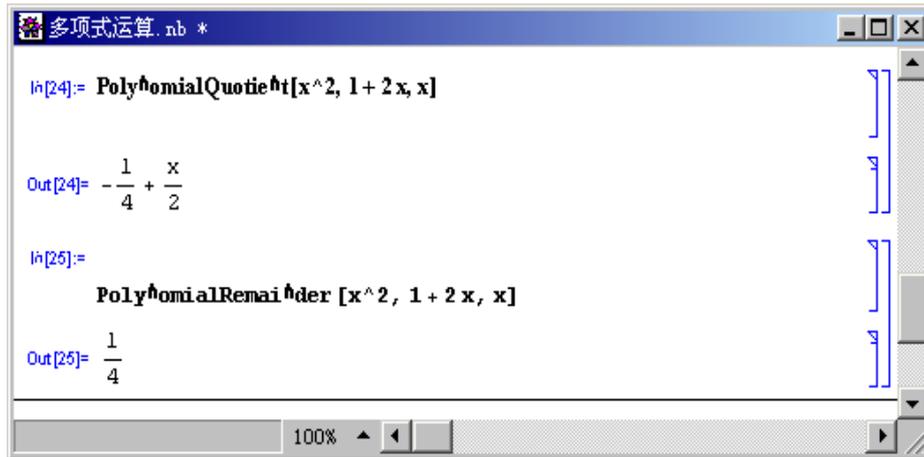
(5). 另外使用 Cancel 函数可以约去公因式



两个多项式相除，总能写成一个多项式和一个有理式相加 Mathematic 中提供两个函数

PolynomialQuotient 和 PolynomialRemainder 分别返商式和余式。

例如：



3. 2 方程及其根的表达

因为 Mathematica 把方程看作逻辑语句。在数学方程式表示为形如“ $x^2-2x+1=0$ ”的形式。在 Mathematica 中“=”用作赋值语句，这样在 Mathematica 中用“==”表示逻辑等号，则方程应表示为“ $x^2-2x+1==0$ ”。方程的解同原方程一样被看作是逻辑语句。例如用 Roots 求方程 x^2-3x+2 的根显示为



这种表示形式说明 x 取 1 或 2 均可。而用 Solve[] 可得解集形式。

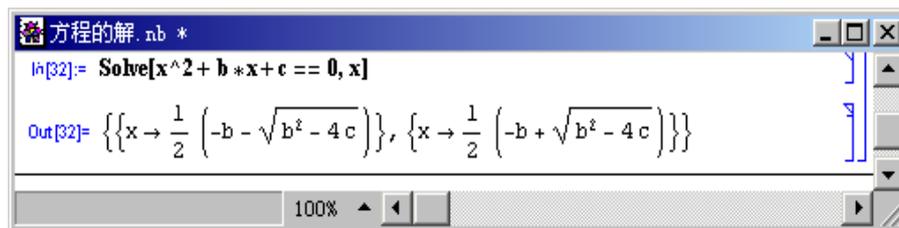


1 求解一元代数方程

下面是常用的一些方程求解函数

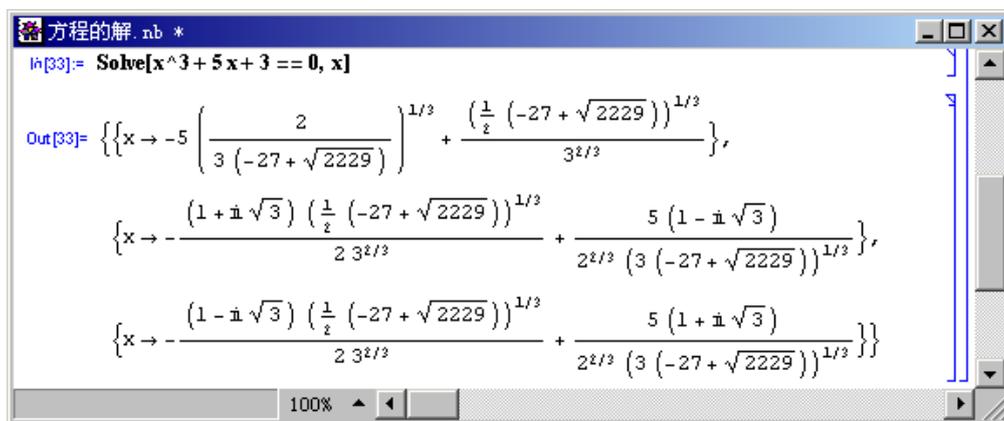
Solve[lhs==rhs, vars]	给出方程的解集
NSolve[lhs==rhs, vars]	直接给出方程的数值解集
Roots[lhs==rhs, vars]	求表达式的根
FindRoot[lhs==rhs, {x, x0}]	求 x=x0 时, 方程的解值

先看 Solve 函数例子



Solve 函数可处理的主要方程是多项式方程。Mathematica 总能对不高于四次的方程进行精确求解，对于三次或四次方程，解的形式可能很复杂。

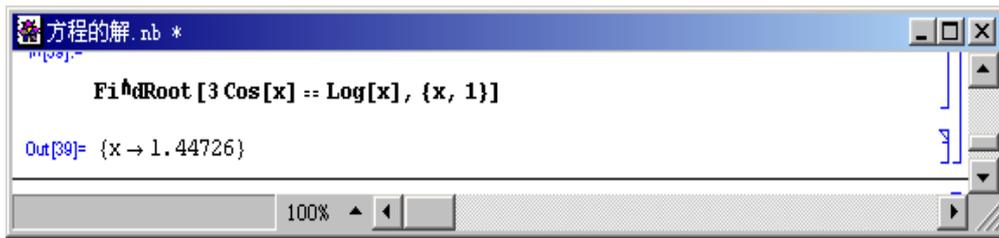
例如求 $x^3 + 5x + 3 = 0$



这时可用 N 函数近似数值解。

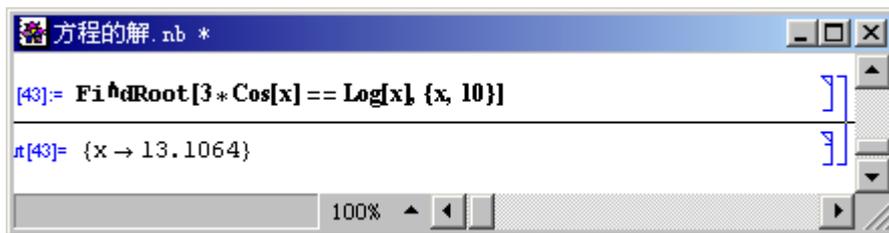
当方程中有一些复杂的函数时，Mathematica 可能无法直接给出解来。在这种情况下我们可用 FindRoot[] 来求解，但要给出起始条件。

例如：求 $3\text{Cos}x = \log x$ 的解

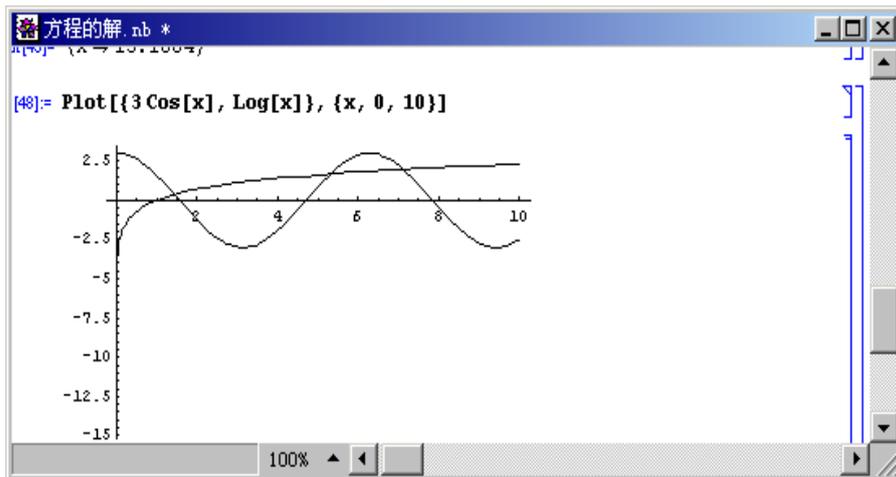


但只能求出 $x=1$ 附近的解，如果方程有几个不同的解，当给定不同的条件时，将给出不同的解。

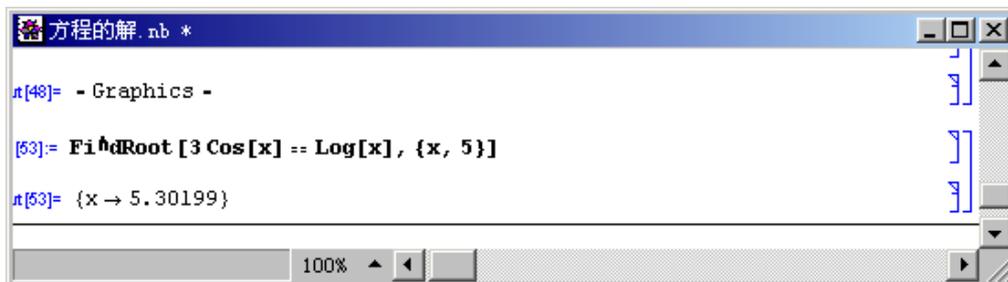
如上例若求 $x=10$ 附近的解命令为：



因此确定解的起始位置是比较关键，一种常用的方法是，先绘制图形观察后再解



如上例通过图形可断定在 $x=5$ 附近有另一根

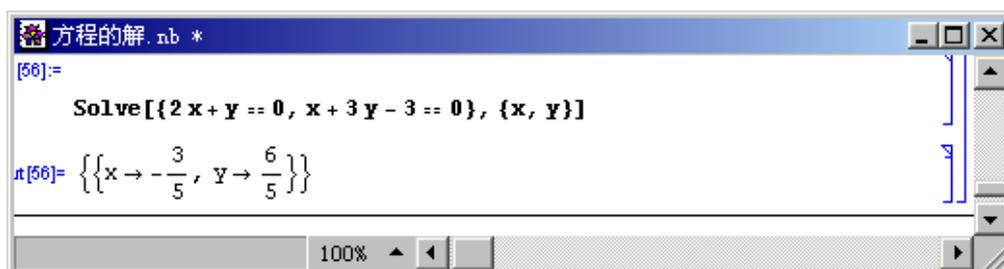


2. 求方程组的根

使用 Solve 和 NSolve, FindRoot 也可求方程组的解, 只是使用时格式略有不同下面给出一个

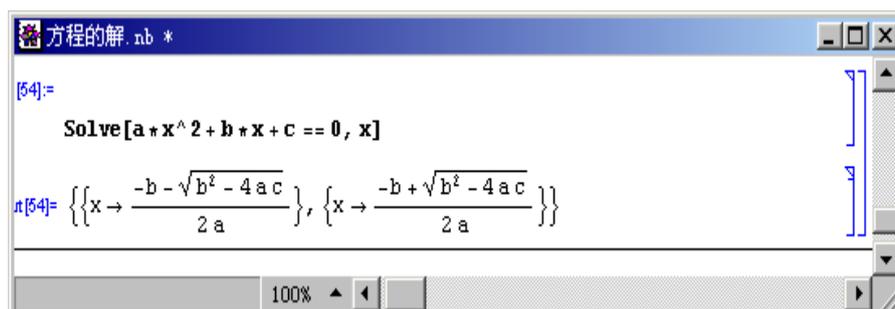
Solve 函数的例子:

求解:
$$\begin{cases} 2x + y = 0 \\ x + 3y - 3 = 0 \end{cases}$$



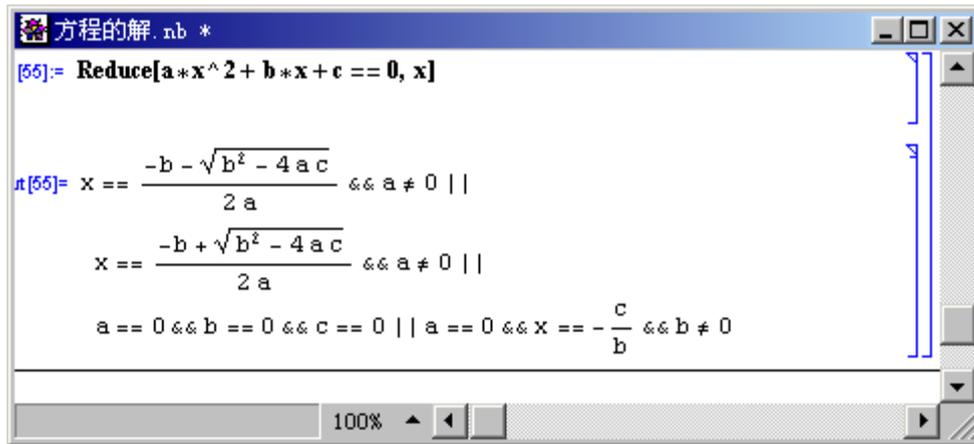
3 求方程的全解

如果我们求 $ax^2+bx+c=0$ 的根我们用 Solve 函数解的结果是:



这显然是不合理的, 因为对不同的 a, b, c 方程的解有不同的情况, 而上面只是给出部分解如果要

解决这个问题可用 Reduce 命令, 它可根据, a, b, c 的取值给出全部值。



因此 Solve, Roots 只给出方程的一般解，而 Reduce 函数数可以给出方程的全部可能解。

4. 解条件方程

在作方程计算时，可以把一个方程看作你要处理的主要方程，而把其他方程作为必须满足的辅助

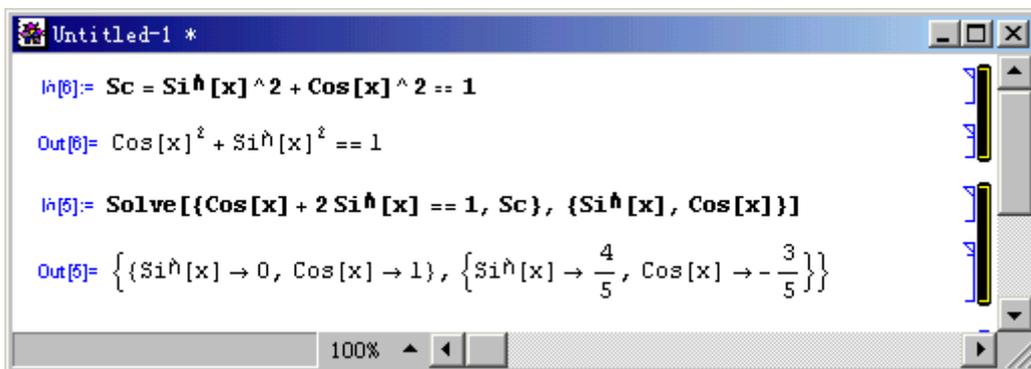
条件，你将会发现这样处理很方便。譬如在求解像 $x^4 + bx^2 + c = 0$ 这样的方程时，通常我们

采用 $x^2 = y$ 的代换方法使求解方程

得到简化。在 Mahematica 中，我们通常是首先命名辅助条件组，然后用名字把辅助条件包含在

你要用函数 Solve[] 求解的方程组中。

用 Sc 定义方程： $\text{Sin}^2 x + \text{Cos}^2 x = 1$ ，在这种条件下，求解方程。



3.3 求和与求积

在 Mathematica 中，数学上的各式符号 \sum 用 Sum 表示，连乘 \prod 用 Product 表示。
下面列出求

各与求积函数的形式和意义：

Sum[f, {i, imin, imax}] 求和 $\sum_{i=imin}^{imax} f$

Sum[f, {i, imin, imax, di}] 以步长 di 增加 i 求和

Sum[f, {i, imin, imax}, {j, jmin, jmax}] 嵌套求和 $\sum_{i=imin}^{imax} \sum_{j=jmin}^{jmax} f$

Product[f, {i, imin, imax}] 求积 $\prod_{i=imin}^{imax} f$

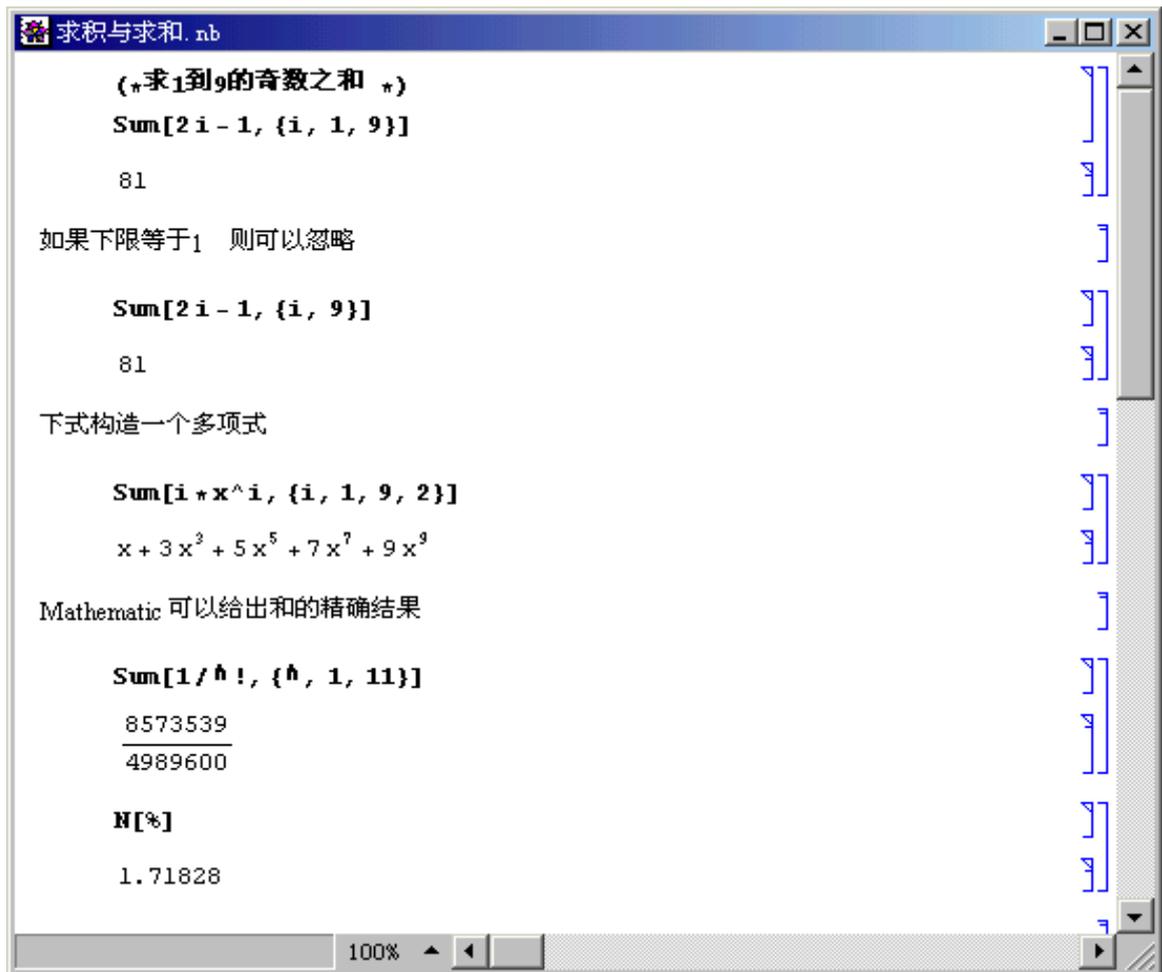
Product [f, {i, imin, imax, di}] 以步长 di 增加 i 求和

Product[f, {I, imin, imax}, {j, jmin, jmax}] 嵌套求积 $\prod_{i=imin}^{imax} \prod_{j=jmin}^{jmax} f$

Nsum[f, {i, imin, Infinity}] 求 $\sum_{i=imin}^{\infty} f$ 近似值

NProduct[f, {i, imin, Infinity}] 求 $\prod_{i=imin}^{\infty} f$ 近似值

一些例子



4.1 基本的二维图形

Mathematica 在直角坐标系中作一元函数图形用下列基本命令。

`Plot[f, {x, xmin, xmax}, option->value]` 在指定区间上按选项定义值画出函数在直角坐标系中的图形。

`Plot[{f1, f2, f3, ...}, {x, xmin, xmax}, option->>value]` 在指定区间上按选项定义值同时画出多个函数在直角坐标系中的图形

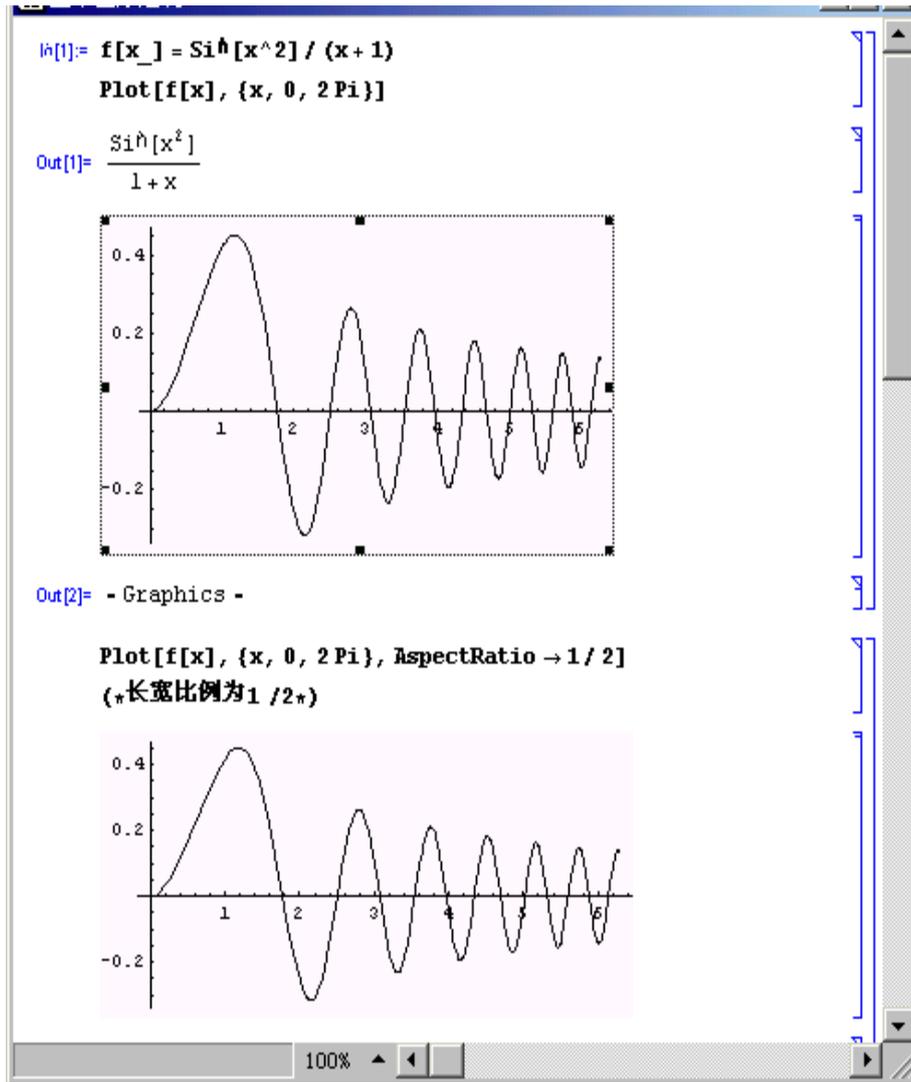
Mathematica 绘图时允许用户设置选项值对绘制图形的细节提出各种要求。例如：要设置图形的高宽比，给图形加标题等。每个选项都有一个确定的名字，以“选项名->选项值”的形式

放在 Plot 中的最右边位置，一次可设置多个选项，选项依次排列，用逗号隔开，也可以不设置选项，采用系统的默认值。

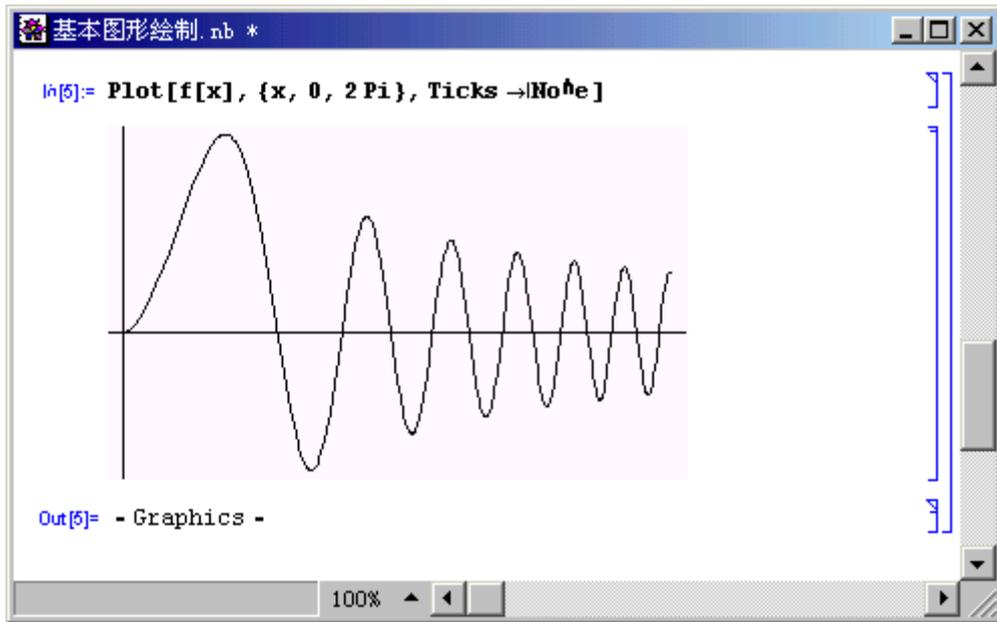
选项	说明	默认值
AspectRatio	图形的高、宽比	1/0.618
AxesLabel	给坐标轴加上名字	不加
PlotLabel	给图形加上标题	不加
PlotRange	指定函数因变量的区间	计算的结果
PlotStyle	用什么样方式作图（颜色，粗细等）	值是一个表
PlotPoint	画图时计算的点数	25

1. 举例

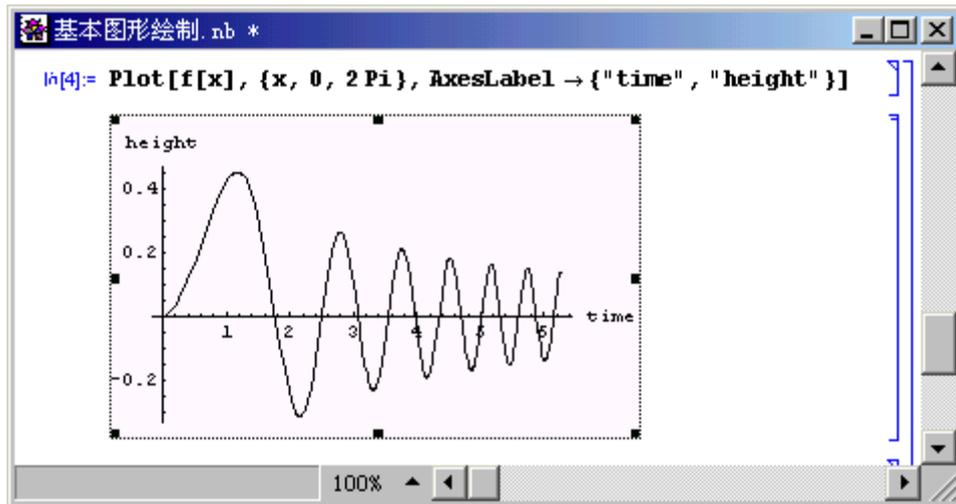
(1). 例如绘制 $f(x) = \frac{\sin x^2}{x+1}$ 的图形。



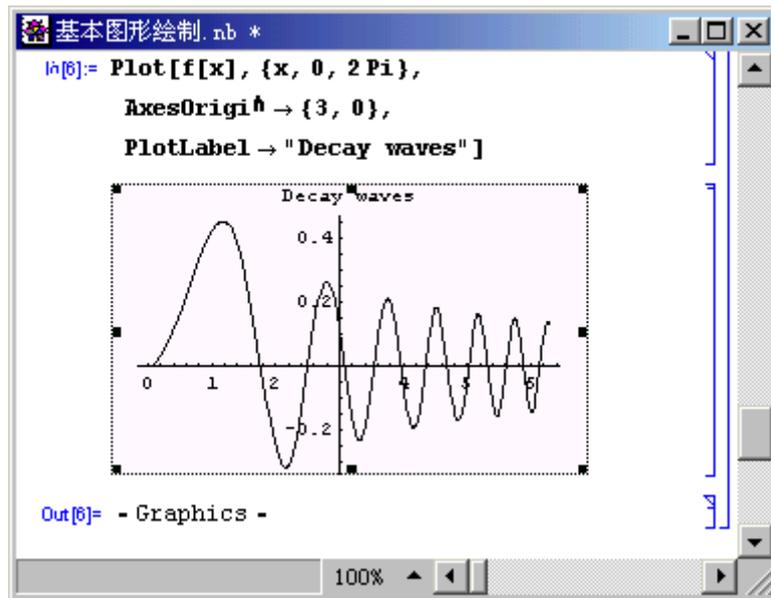
(2). 如果要取消刻度可以使用 Ticks 选项



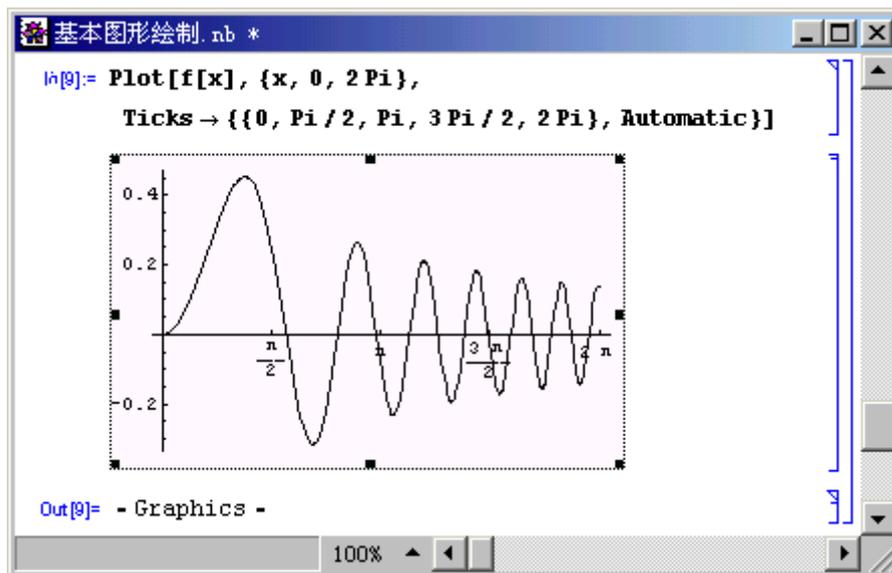
(3). 如果要标注坐标名称 x 轴为“Time”, y 轴为“Height”



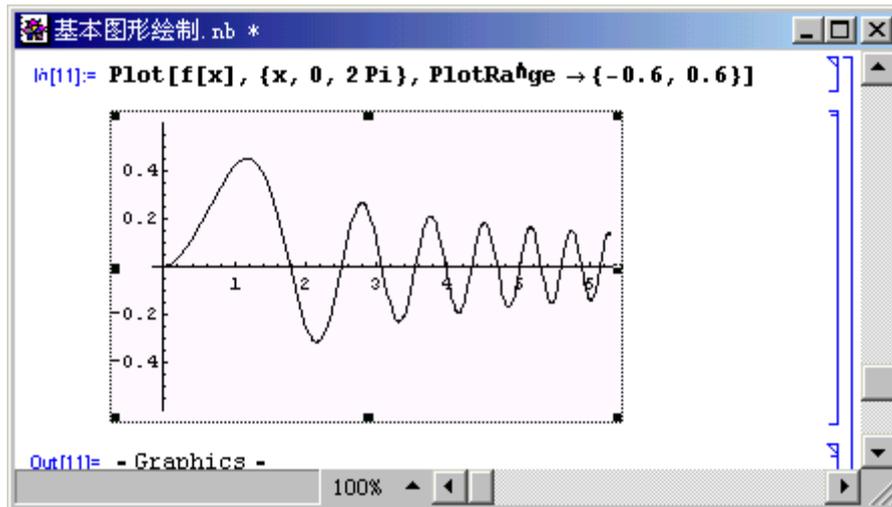
(4). 将坐标交点 (3, 0), 并标注图形名称。



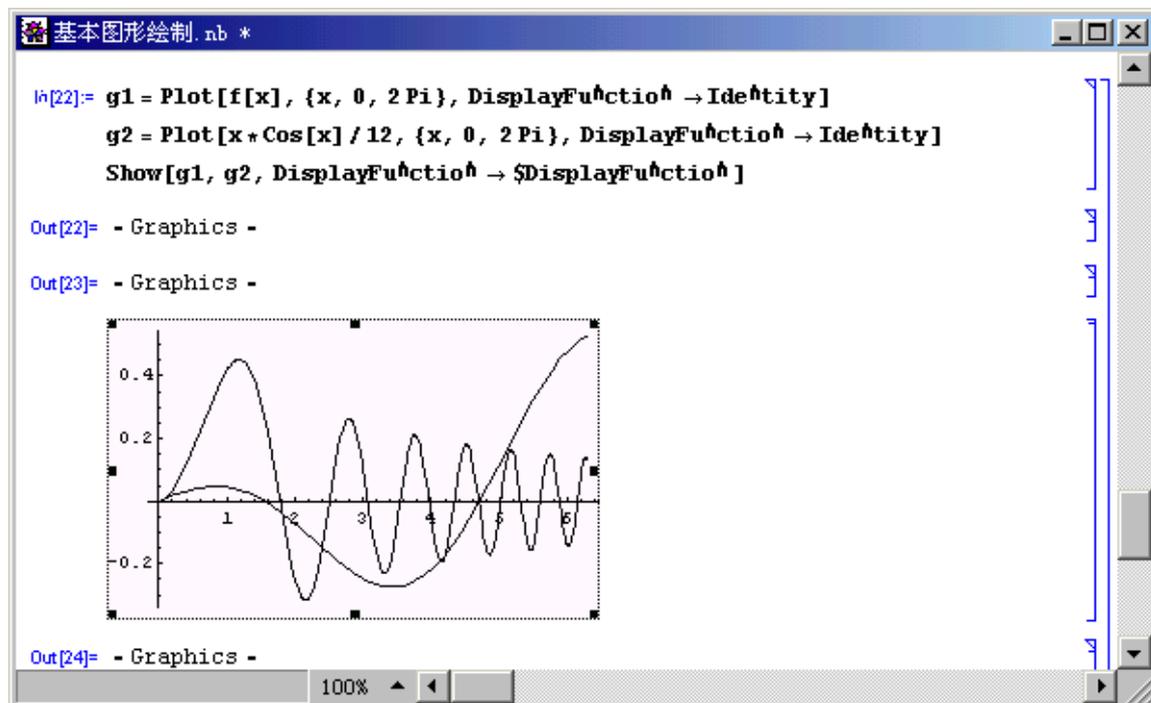
(5). 修改 x 方向的刻度, y 轴方向的刻度则用默认值。



(6). 定义 y 轴的绘图范围



(7). 另外我们也可以将图形结果定义给变量，但不显示图形，后用 Show 命令显示。

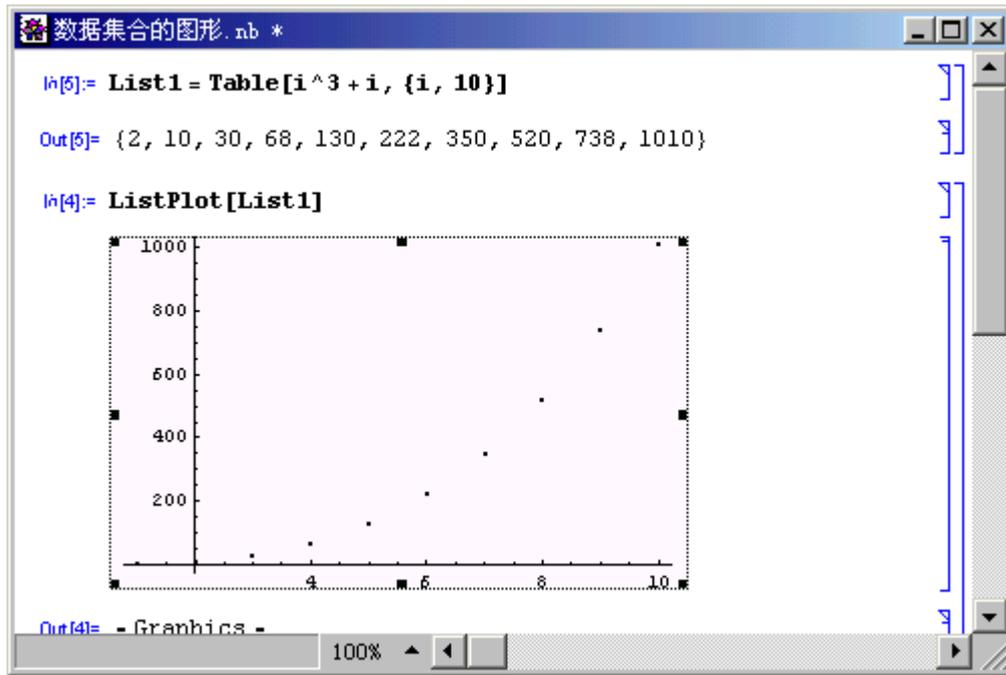


2. 数据集合的图形

Mathematica 用于绘数字集合的图形的命令与前面介绍的绘函数图形的命令是相似的。如下：

ListPlot[{y1, y2, ...}]	绘出在 x 的值为 1, 2... 时 y1, y2, ... 的图形
ListPlot[{{x1, y1}, {x2, y2}, ...}]	绘出离散点 (xi, yi)
ListPlot[List, PlotJoined -> True]	把离散点连成曲线

(1). 下面举例说明下面是一个离散数据的集合的图形

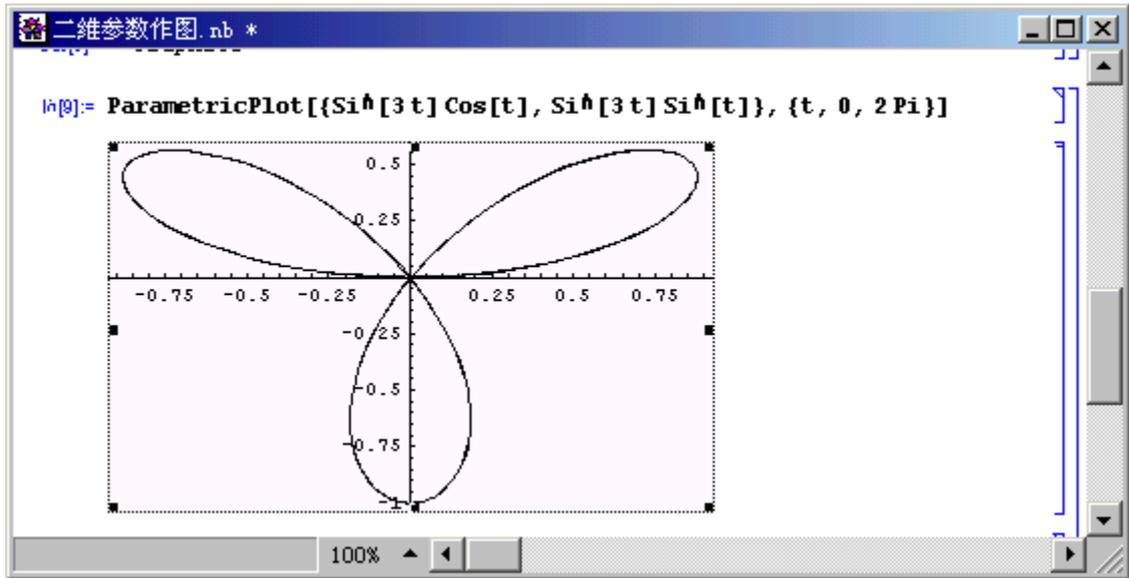


3. 二维参数作图

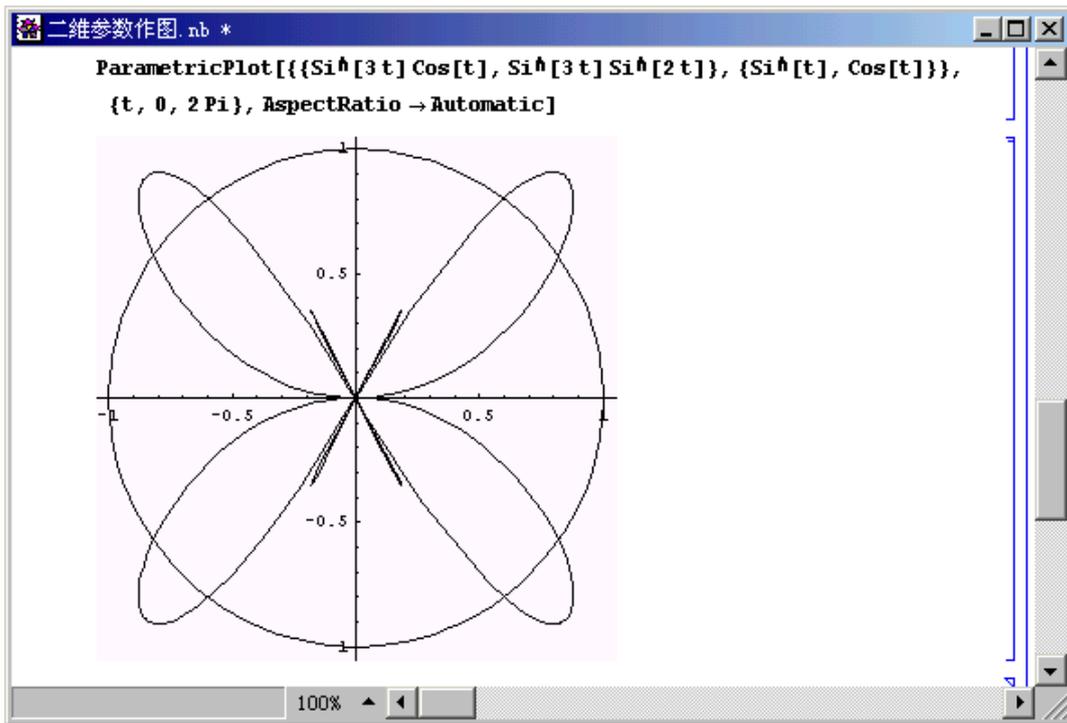
前面我们使用 Plot 命令可以绘出直角坐标系下的函数图形,使用 ParametricPlot 可以绘制参数曲线下面给出 ParametricPlot 的常用形式

ParametricPlot[{fx, fy}, {t, tmin, tmax}]	绘出参数图
ParametricPlot[{fx, fy}, {gx, gy}, ..., {t, tmin, tmax}]	绘出一组参数图
ParametricPlot[{fx, fy}, {t, tmin, tmax}, AspectRatio->Automatic]	设法保持曲线的形

(1). 绘制参数方程
$$\begin{cases} x = \sin 3t \cos t \\ y = \sin 3t \sin t \end{cases}$$
 的图形



(2). 下面将一个圆与上面参数绘在同一个坐标下，并保证图形的形状正确。



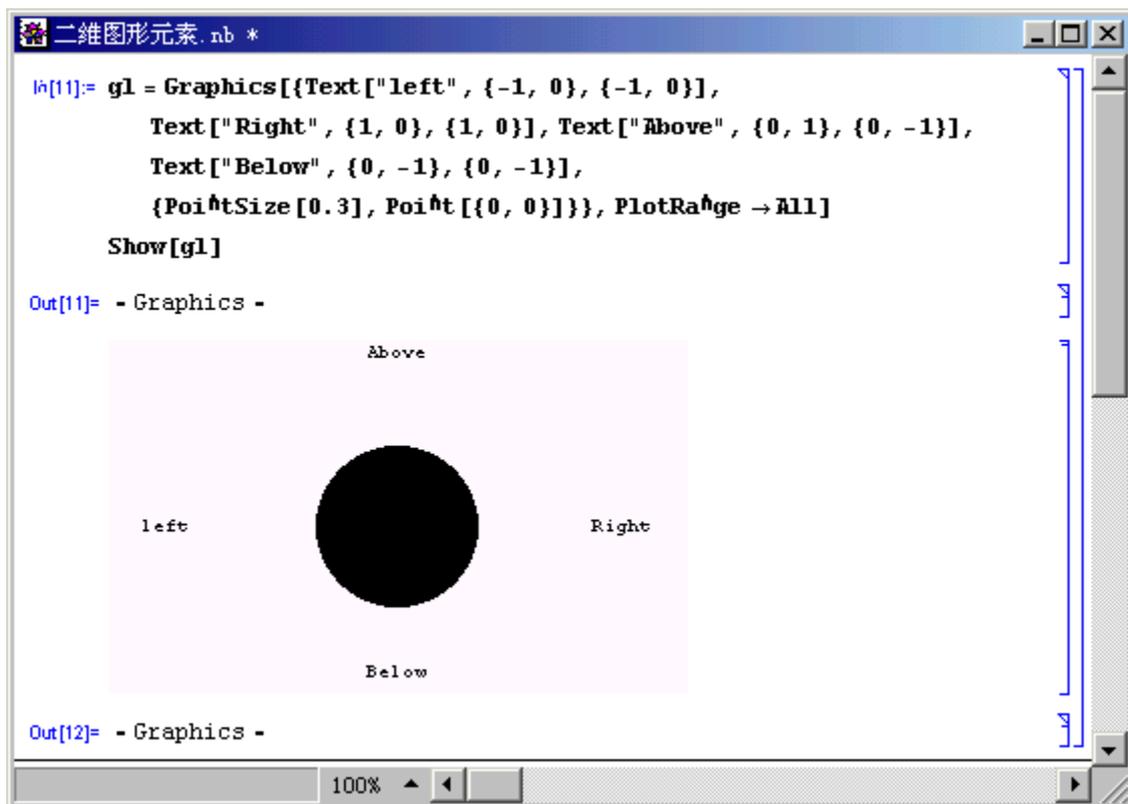
4. 2 二维图形元素

用图形元素绘图适合于绘制结构复杂的图形。Mathematica 中还提供了各种如绘制点、线段、圆弧等函数。同样我们可先用 Graphics 作出平面图形的表达式，再用 Show 显示守成的图形。

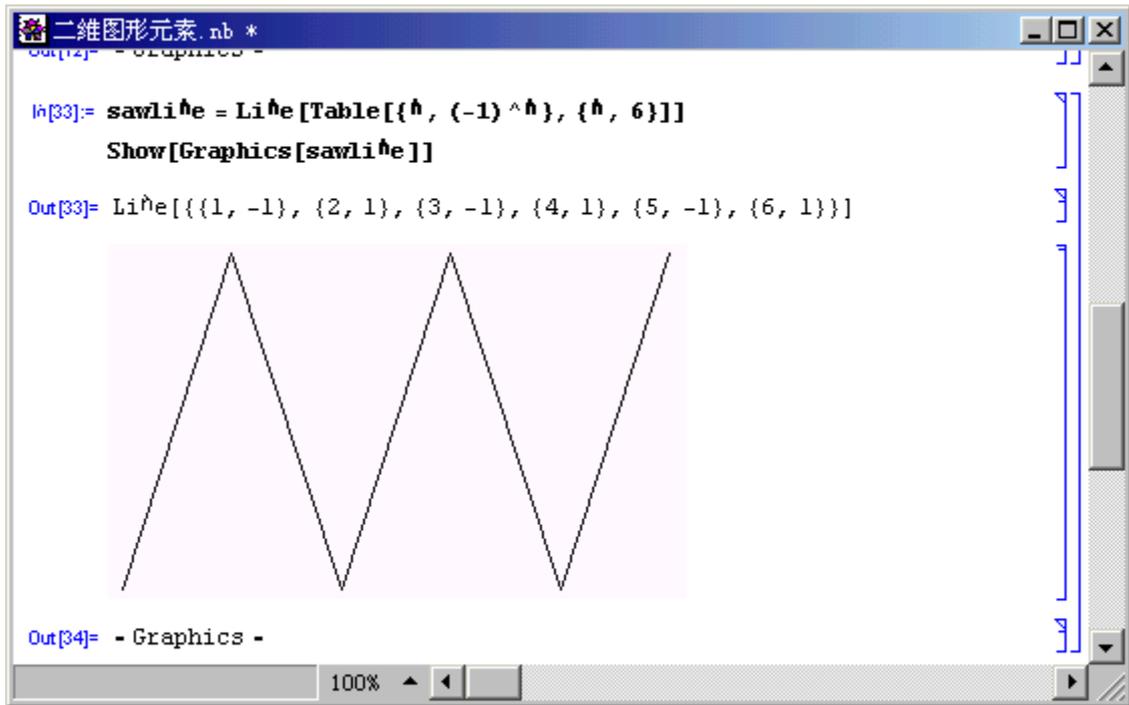
下面给出在 Mathematica 中常用的二维图形元素。

Point[[x, y]]	点
Line[{{x1, y1}, {x2, y2}, ...}]	线段
Rectangle[{xmin, ymin}, {xmax, ymax}]	填充矩阵
Polygon	[{{x1, y1}, {x2, y2}, ...}] 填充多边形
Circle[{x, y}, r]	圆
Circle[{x, y}, {rx, ry}]	半轴分别为 rx, ry 的椭圆
Circle[{x, y}, r, {theta1, theta2}]	圆弧
Circle[{x, y}, {rx, ry}, {theta1, theta2}]	椭圆弧
Disk[{x, y}, r]	填充圆
Raster[{{a11, a12, ...}, {a21, ...}, ...}]	灰度在 0 到 1 之间的灰层组
Text[Expr, {x, y}]	文本大小

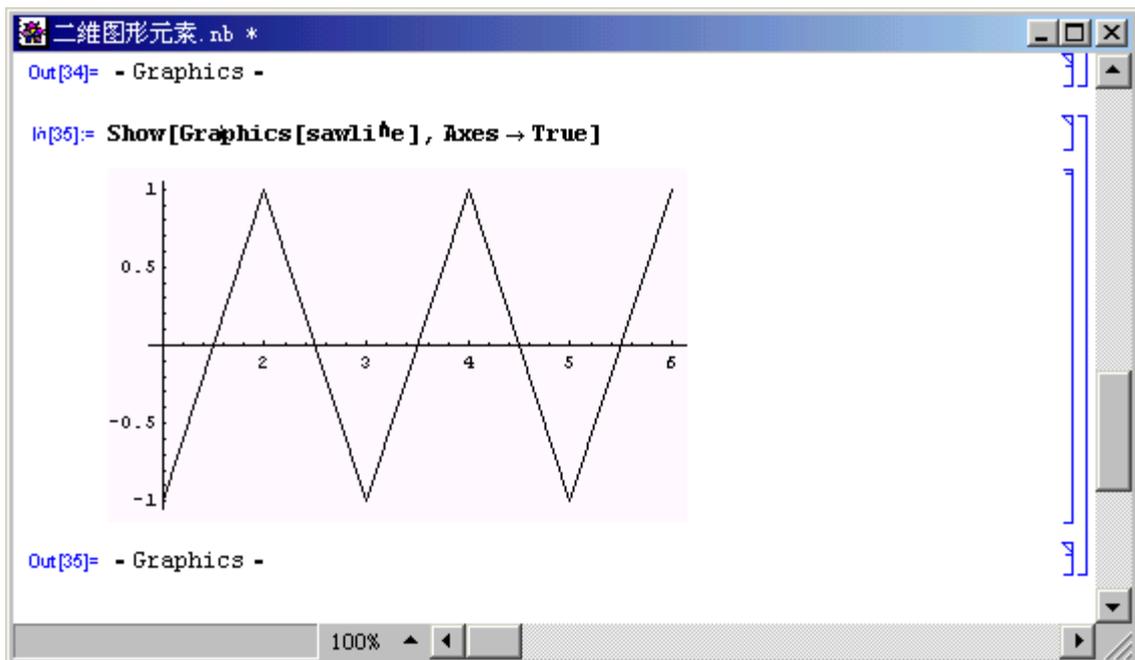
下图绘出一个有颜色和大小点，且在图形四周插入文本



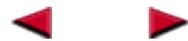
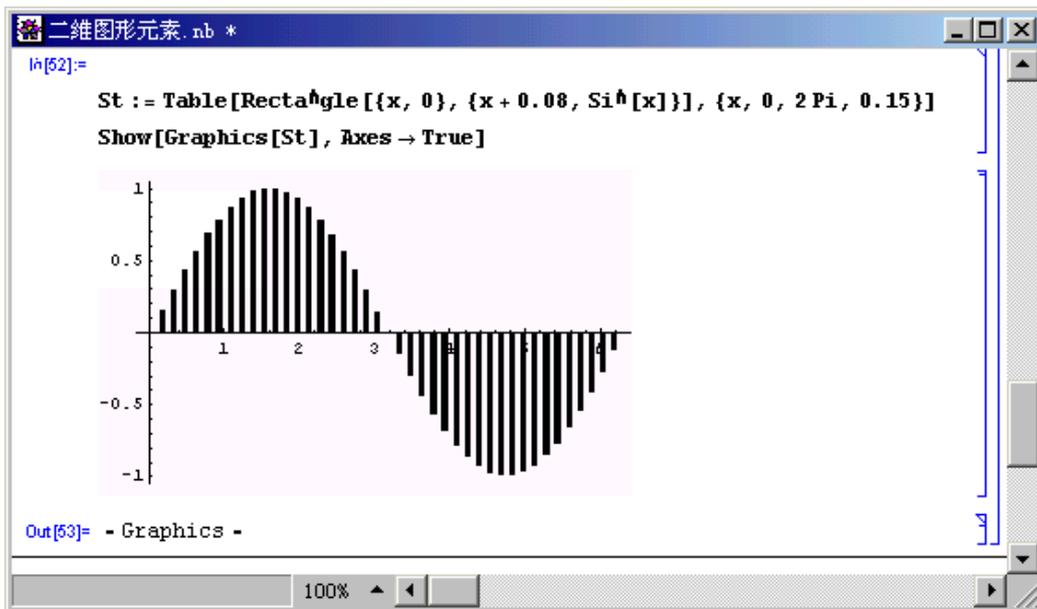
下面绘制一些有线条组成的图形



当然也可以添加坐标轴下面的例子，说明了这一点。



下面的例子，是说明了 Rectangle 的图形绘制，例子中用一些小矩形逼近正弦曲线与 x 轴所成面积。程序中生成一个图形集合并显示出来。



4.3 图形的样

式

我们称图形的颜色、曲线的形状和宽度等特性为图形样式。在本节中，我们就图形的各种样式，尤其是曲线的样式进行学习。

下面给出选项用于设置图形样式。

GrayLevel[]	灰度介于 0(黑)到 1(白)之间
RGBColor[r, g, b]	由红、绿、蓝组成的颜色，每种色彩取 0 到 1 之间的数
Hue[A]	取 0 到 1 之间的色彩
Hue[h, s, b]	指定色调，位置和亮度的颜色，每项介于 0 到 1 之间
PointSize[d]	给出半径为 d 的点，单位是 Plot 的一个分数
AbsolutePointSize[d]	给出半径为 d 的点(以绝对单位量取)
Thickness[w]	给所有线的宽度 w，单位是 Plot 的分数
AbsoluteThickness[w]	给所有线的宽度 w，(以绝对单位量取)
Dashing[w1, w2, ...]	给所有线为一系列虚线，虚线段的长度为 w1, w2, ...
Absolutedashing[{w1, w2, ...}]	以绝对单位给出虚线长度

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/988057107073007000>