**Mathematica入门教程**

**Mathematica的基本语法特征**

　　如果你是第一次使用Mathematica，那么以下几点请你一定牢牢记住：

Mathematica中大写小写是有区别的，如Name、name、NAME等是不同的变量名或函数名。

系统所提供的功能大部分以系统函数的形式给出，内部函数一般写全称，而且一定是以大写英文字母开头，如Sin[x],Conjugate[z]等。

乘法即可以用\*，又可以用空格表示，如2 3＝2\*3＝6 ,x y,2 Sin[x]等；乘幂可以用“^”表示，如x^0.5,Tan[x]^y。

自定义的变量可以取几乎任意的名称，长度不限，但不可以数字开头。

当你赋予变量任何一个值，除非你明显地改变该值或使用Clear[变量名]或“变量名=.”取消该值为止，它将始终保持原值不变。

一定要注意四种括号的用法：()圆括号表示项的结合顺序，如(x+(y^x+1/(2x)));[]方括号表示函数，如Log[x],BesselJ[x,1]；{}大括号表示一个“表”(一组数字、任意表达式、函数等的集合)，如{2x,Sin[12 Pi],{1+A,y\*x}}；[[]]双方括号表示“表”或“表达式”的下标，如a[[2,3]]、{1,2,3}[[1]]=1。

Mathematica的语句书写十分方便，一个语句可以分为多行写，同一行可以写多个语句（但要以分号间隔）。当语句以分号结束时，语句计算后不做输出（输出语句除外），否则将输出计算的结果。

**一.数的表示及计算**

1.在Mathematica中你不必考虑数的精确度，因为除非你指定输出精度，Mathematica总会以绝对精确的形式输出结果。例如：你输入

In[1]:=378/123，系统会输出Out[1]:=126/41，如果想得到近似解，则应输入

In[2]:=N[378/123,5],即求其5位有效数字的数值解，系统会输出Out[2]:=3.073

2，另外Mathematica还可以根据你前面使用的数字的精度自动地设定精度。

　　Mathematica与众不同之处还在于它可以处理任意大、任意小及任意位精度的数值，如100^7000,2^(-2000)等数值可以很快地求出，但在其他语言或系统中这是不可想象的，你不妨试一试N[Pi,1000]。

Mathematica还定义了一些系统常数，如上面提到的Pi(圆周率的精确值)，还有E(自然对数的底数)、I(复数单位)，Degree(角度一度，Pi/180)，Infinity(无穷大)等，不要小看这些简单的符号，它们包含的信息远远大于我们所熟知的它们的近似值，它们的精度也是无限的。

二.**“表”及其用法**

“表”是Mathematica中一个相当有用的数据类型，它即可以作为数组，又可以作为矩阵；除此以外，你可以把任意一组表达式用一个或一组{}括起来，进行运算、存储。可以说表是任意对象的一个集合。它可以动态地分配内存，可以方便地进行插入、删除、排序、翻转等等几乎所有可以想象到的操作。

　　如果你建立了一个表，你可以通过下表操作符[[]](双方括号)来访问它的每一个元素，如我们定义table={2,Pi,Sin[x],{aaa,A\*I}}为一个表，那么table[[1]]就为2，table[[2]]就是Pi，而table[[3,1]]表示嵌套在table中的子表{aaa,A\*I}的第一个元素即aaa，table[[3,2]]表示{aaa,A\*I}第二个元素即A\*I。总之，表每一层次上并列的部分用逗号分割，表可以无穷嵌套。

你可以通过Append[表,表达式]或Prepend[表,表达式]把表达式添加到表的最前面或最后面，如Append[{1,2,3},a]表示{1,2,3,a}。你还可以通过Union[表1，表2，......],Jion[表1,表2,......]来把几个表合并为一个表，二者不同在于Union在合并时删除了各表中重复的元素，而后者仅是简单的合并；你还可以使用Flatten[表]把表中所有子表"抹平"合并成一个表，而Patition[表，整数n]把表按每n个元素分段作为子表，集合成的表。如Flatten[{1,2,{Sin[x],dog},{{y}}}]表示{1,2,Sin[x],y},而Partition[{1,2,Sin[x],y},2]把表每两个分段，结果为{{1,2},{Sin[x],y}}；还可以通过Delete[表，位置]、Insert[表，位置]来向表中按位置插入或删除元素，如要删除上面提到的table中的aaa,你可以用Delete[table,{3,1}]来实现；Sort[表]给出了表中各元素的大小顺序，Reverse[表]、RotateLeft[表，整数n]、RotateRight[表，整数n]可以分别将一个表进行翻转、左转n个元素、右转n个元素等操作，Length[表]给出了表第一个层次上的元素个数，Position[表，表达式]给出了表中出现该表达式的位置，Count[表，表达式]则给出表达式出现的次数。各种表的操作函数还有很多，这里就不再一一介绍了。

**三.图形函数**

Mathematica的图形函数十分丰富，用寥寥几句就可以画出复杂的图形，而且可以通过变量和文件存储和显示图形，具有极大的灵活性。

　 图形函数中最有代表性的函数为Plot[表达式，{变量，下限，上限}，可选项]，(其中表达式还可以是一个"表达式表"，这样可以在一个图里画多个函数)；变量为自变量；上限和下限确定了作图的范围；可选项可要可不要，不写系统会按默认值作图，它表示对作图的具体要求。例如Plot[Sin[x],{x,0,2\*Pi},AspectRatio-1]表示在0<x<2Pi的范围内作函数Sin[x]的图象，AspectRatio为可选项，表示图的x向y向比例，AspectRatio-1表示纵横比例为1:1，如果不写这一项，系统默认比例为1:GodenRatio,即黄金分割的比例(注意，可选项的写法为可选项名-可选项值)，Plot还有很多可选项，如PlotRange表示作图的值域，PlotPoint表画图中取样点的个数，越大则图越精细，PlotStyle来确定所画图形的线宽、线型、颜色等特性，AxesLabel表式在坐标轴上作标记等等。

**.二维函数作图**

**Plot[函数f，{x，xmin，xmax}，选项]**

**在区间{x，xmin，xmax}上，按选项的要求画出函数f的图形**

**Plot[{函数1，函数2}，{x，xmin，xmax}，选项]**

**在区间{x，xmin，xmax}上，按选项的要求画出几个函数的图形**

#### 

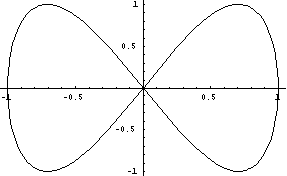
图一.用Plot生成x\*Sin[1/x]的图形

#### Image82

**.二维参数画图函数**

**ParametricPlot[{x[t],y[t]},{t,t0,t1},选项] 画一个X轴,Y轴坐标为{x[t],y[t]},参变量t在[t0,t1]中的参数曲线**

图二.用**ParametricPlot**生成Image83的图形



**.三维函数作图**

**Plot3D[f[x,y],{x,x0,x1},{y,y0,y1},选项]**

**在区域**Image85**上,画出空间曲面f[x,y].**

图3.用Plot3D生成的Sin[x]\*Cos[y]的三维图形

#### Image86

除Plot,二维参数方程作图的ParametricPlot[{x(t),y(t)},{t,下限,上限}，可选项]、三维作图的Plot3D[二维函数表达式，{变量1，下限，上限}, {变量2，下限，上限}，可选项}]、三维参数方程作图的ParametricPlot3D[{x(u,v),y(u,v),z(u,v)},{u,下限,上限},{v,下限,上限}，可选项]外,还有画二维等高线图ContourPlot[二元表达式，{变量1，下限，上限}, {变量2，下限，上限}，可选项}]、画二维密度图的DensityPlot[二元表达式，{变量1，下限，上限}, {变量2，下限，上限}，可选项}]等等不一而足。

　　除使用上述函数作图以外，Mathematica还可以象其他语言一样使用图形元语言作图，如画点函数Point[x,y],画线函数Line[x1,y1,x2,y2],画圆的Circle[x,y,r]，画矩形和多边形的Rectangle和Polygon,字符输出的Text[字符串,输出坐标]，还有颜色函数RGBColor[red,green,blue]、Hue[],GrayLevel[gray]来描述颜色的亮度、灰度、饱和度，用PointSize[相对尺度]、Thickness[相对尺度]来表示点和线的宽度。总之Mathematica可以精确地调节图形的每一个特征。

**四.数学函数的用法**

Mathematica系统内核提供了丰富的数学计算的函数，包括极限、积分、微分、最值、极值、统计、规划等数学的各个领域，复杂的数学问题简化为对函数的调用，极大地提高了解决问题的效率。

　 Mathematica提供了所有的三角、反三角、双曲、反双曲、各种特殊函数(如贝塞尔函数系、椭圆函数等)，各种复数函数(如Im[z],Re[z],Conjugate[z], Abs[z],Arg[z])，各种随机函数(如Random[n]可以通过不同的参数产生任意范围内整型、实型任意分布的随机数)，矩阵运算函数(如求特征值特征向量的EigenVector[],EigenValue[],求逆的Inverse[]等)。

　　Mathematica还提供了大量数学操作的函数，如取极限的Limit[f[x],{x,a}],求微分的D[f[x],x],全微分的Dt[f[x],x],不定积分的Integrate[f[x],x]和定积分的Integrate[f[x],{x,a,b}],解任意方程的Solve[lhs=rhs,x]及微分方程的DSolve[lhs=rhs,x],解幂级数和付立叶展开的Series[f[x]]，Fourier[f[x]]及其逆变化InverseSeries,InverseFourier, 求和函数Sum[],求积函数Product[]，以上函数均可以适用于多维函数或多维方程。

　　Mathematica中还有相当数量的数值计算函数，最常用的是N[表达式,整数]可以求出表达式精确到指定有效数字的数值解，还有如数值求积分的NIntegrate[],求方程数值根的NSolve[]和NDSolve[],最小、最大值的NFindMinimum[]和NFindMaximum[]等等。

Mathematica还有各种表达式操作的函数，如取分子、分母的 Numerator[expr] , Denormator[expr],取系数的Coefficient[expr],因式分解的Factor[expr],以及展开的Expand[expr]和ExpandAll[expr],表达式化简的Simplify[expr]等。expr代表一个任意的表达式。

**. 求极限**

**计算函数极限Image87的一般形式是:**

**Limit[expr,x->x0] x->x0时函数的极限**

**Limit[expr,x->x0,Direction->-1] x->Image88时函数的极限**

**Limit[expr,x->x0, Direction->1] x->Image89时函数的极限**

In[1]:= Image90

Out[1]:=1

**. 微商和微分**

在Mathematica中能方便地计算任何函数表达式的任意阶微商(导数).如果f是一元函数,D[f,x]表示Image91;如果f是多元函数,D[f,x]表示Image92.微商函数的常用形式如下:

**D[f,x]** Image93Image92

#### In[1]:=D[x^x,x]

Out[1]:=Image99

下面列出全微分函数Dt的常用形式及其意义:

**Dt[f]** 全微分

**Dt[f,x]** 全导数

**Dt[f,x1,x2,…]** 多重全导数

In[1]:=Dt[x^2+y^2]

Out[1]:=

**. 不定积分和定积分**

#### 不定积分

**Integreate**函数主要计算只含有1“简单函数”的被积函数. “简单函数”包括有理函数、指数函数、对数函数和三角函数与反三角函数。不定积分一般形式如下：

Integrate[f，x] 计算不定积分

Integrate[f，x，y] 计算不定积分

Integrate[f，x，y，z] 计算不定积分Image107

In[1]：= **Image108**

Out[1]：= Image109

In[2]：= **Image110**

Out[2]：= Image111

**2．定积分**

计算定积分的命令和计算不定积分是同一个**Integrate**函数，在计算定积分时，除了要给出变量外还要给出积分的上下限。当定积分算不出准确结果时，用**N[%]**命令总能得到其数值解**.Nintegrate**也是计算定积分的函数,其使用方法和形式和**Integrate**函数相同.用**Integrate**函数计算定积分得到的是准确解,**Nintegrate**函数计算定积分得到的是近似数值解.计算多重积分时,第一个自变量相应于最外层积分放在最后计算.

Integrate[f,{x,a,b}] 计算定积分Image112

NIntegrate[f,{x,a,b}] 计算定积分Image112

Integrate[f,{x,a,b},{y,c,d}] 计算定积分Image113

NIntegrate[f,{x,a,b},{y,c,d}] 计算定积分Image113

In[1]:= **Image114**

Out[1]:= Image115

In[2]:= **Image116**

Out[2]:= Image117

In[3]:= **Image118**

Out[3]:= Image119

**. 幂级数**

幂级数展开函数Series的一般形式:

**Series[expr,{x,x0,n}] 将expr在x=x0点展开到n阶的级数**

**Series[expr,{x,x0,n},{y,y0,m}] 先对y展开到m阶再对x展开n阶幂级数**

#### 用Series展开后,展开项中含有截断误差Image120

In[1]:= **Image121**

Out[1]:= Image122

In[2]:= **Image123**

Out[2]:= **Image124**

In[3]:= **Image125**

Out[3]:= Image126

**. 常微分方程**

#### 求解常微分方程和常微分方程组的函数的一般形式如下:

**Dsolve[eqns,y[x],x]** 解y(x)的微分方程或方程组eqns,x为变量

**Dsolve[eqns,y,x]** 在纯函数的形式下求解

**NDsolve[eqns,y[x],x,{xmin,xmax}]** 在区间{xmin,xmax}上求解变量x的数的形式下求解常微分方程和常微分方程组eqns的数值解

In[1]:= **Image127**

Out[1]:= Image128

In[2]:= **Image129**

Out[2]:= Image130

In[3]:= **Image131**

#### Out[3]:= Image132

**.线性代数**

### 定义向量和矩阵函数

#### 定义一个矩阵,可用函数Table或Array.当矩阵元素能用一个函数表达式时,用函数Table在定义矩阵大小的同时也给每个矩阵元素定义确定的值.用函数Range只能定义元素为数值的向量.Array只能用于定义向量、矩阵和张量,并规定矩阵和张量的元素下标从1开始.Array的一般形式: Array[向量元素名,n,f] 定义下标从f开始的有n个元素的向量,当f是1时可省略. Array[矩阵元素名,{m,n}] 定义m行n列的矩阵.其中:矩阵元素名是一个标识符,表示矩阵元素的名称,当循环范围是{u,v,w}时定义一个张量. Table[表达式f,循环范围] 表达式f表示向量或矩阵元素的通项公式;循环范围定义矩阵的大小. 循环范围的一般形式:{循环变量名,循环初值,循环终值,循环步长}. 在Array或Table的循环范围表示方法略有区别.请在下面的实例中注意观察.

In[1]:= **Image133**

Out[1]:= **Image134(\*矩阵每一行元素用一对{}括起来\*)**

In[2]:= **Image135**

Out[2]:= **Image136**

In[3]:= **Image137(\*IndentityMatrix[n]生成n维矩阵\*)**

Out[3]:= **Image138**

In[4]:= **Image139(\*生成对角元素为表元素的对角矩阵\*)**

Out[4]:= **Image140**

In[5]:= **Image141(\*TableForm[m]或MatrixForm[m]按矩阵形式输出m\*)**

Out[5]:= Image142

一个矩阵可用一个变量表示,如In[2]所示U是一个矩阵,则U[[I]]表示U的第I行的N个元素;Transpose[U][[j]]表示U的第J行的M个元素;U[[I,j]]或a[I,j]表示U的第I行第J列元素;U[[{i1,i2,…,ip},{j1,j2,…,jq}]]表示由行为{i1,i2,…,ip}和列为{j1,j2,…,jq}组成的子矩阵.

### 矩阵的运算符号和函数

|  |  |
| --- | --- |
| 表达式 | 意义 |
| A+c | A为矩阵,c为标量,c与A中的每一个元素相加 |
| A+B | A,B为同阶矩阵或向量,A与B的对应元素相加 |
| cA | A为矩阵,c为标量,c与A中的每个元素相乘 |
| U.V | 向量U与V的内积 |
| A.B | 矩阵A与矩阵B相乘,要求A的列数等于B的行数 |
| Det[M] | 计算矩阵M的行列式的值 |
| Transepose[M] | M的转置矩阵(Image143或Image144) |
| Inverse[M] | 计算矩阵M的逆矩阵(Image145) |
| Eigenvalus[A] | 计算矩阵A的全部(准确解)特征值 |
| Eigenvalus[N[A]] | 计算矩阵A的全部(数值解)特征值 |
| Eigenvectors[A] | 计算矩阵A的全部(准确解)特征向量 |
| Eigenvectors[N[A]] | 计算矩阵A的全部(数值解)特征向量 |
| Eigensystem[A] | 计算矩阵A的所有(准确解)特征值和特征向量 |
| Eigensystem[N[A]] | 计算矩阵A的所有(数值解)特征值和特征向量 |

### 方程组求解函数

#### 在Mathematica中用LinerSolve[A,B],求解满足AX=B的一个解.如果A的行列式不为零,那么这个解是方程组的唯一解; 如果A的行列式是零,那么这个解是方程组的一个特解,方程组的全部解由基础解系向量的线性组合加上这个特解组成. NullSpace[A]计算方程组AX=0的基础解系的向量表,用LinerSolve[A,B]和NullSpace[A]联手解出方程组AX=B的全部解. Mathematica中还有一个美妙的函数RowReduce[A],它对A的行向量作化间成梯形的初等线性变换.用RowReduce可计算矩阵的秩,判断向量组是线性相关还是线性无关和计算极大线性无关组等工作.

|  |  |
| --- | --- |
| 解方程组函数 | 意义 |
| RowReduce[A] | 作行的线性组合化简A,A为m行n列的矩阵 |
| LinerSolve[A,B] | 求解满足AX=B的一个解,A为方阵 |
| NullSpace[A] | 求解方程组AX=0的基础解系的向量表, A为方阵 |

例:已知A=Image99Image146,计算A的秩,计算AX=0的基础解系.

#### In[1]:= Image147In[2]:= Image148

Out[2]:= **Image149(\*显然,A的秩是2\*)**

#### In[3]:= Image150

Out[3]:= **Image151(\*A的两个线性无关解\*)**

#### 

**五.程序流程控制**

　　循环语句有For[赋初值，循环条件，增量语句，语句块]表示如果满足循环条件，则执行语句块和增量语句，直到不满足条件为止，While[test,block]表明如果满足条件test则反复执行语句块block,否则跳出循环，Do[block,{i,imin,imax,istep}]与前者功能是相同的。还有Goto[lab], Label[lab]提供了程序中无条件跳转，Continue[]和Break[]提供了继续循环或跳出循环的控制，Catch[语句块1]和Throw[语句块2]提供了运算中对异常情况的处理。另外，在程序中书写注释可以用一对"(\*　 \*)"括起来，注释可以嵌套。

**六.其他**

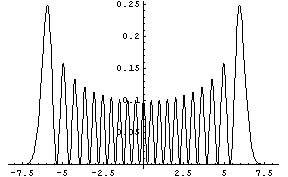
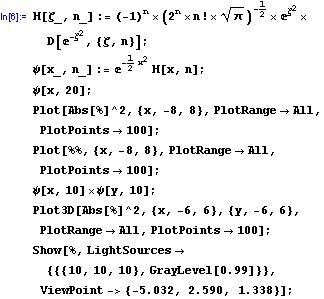
　　1. 使用帮助，Mathematica的帮助文件提供了Mathematica内核的基本用法的说明，十分详细，可以参照学习。

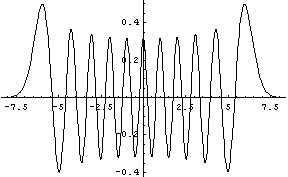
　　2. 你可以使用"? 符号名"或"??符号名"来获得关于该符号(函数名或其他)的粗略或详细介绍。符号名中还可以使用通配符，例如?M\*，则系统将给出所有以M开头的关键词和函数名，再如??For将会得到关于For语句的格式和用法的详细情况。

　　3. 在Mathematica的编辑界面中输入语句和函数，确认光标处于编辑状态(不断闪烁)，然后按Insert键来对这一段语句进行求值。如果语句有错，系统将用红色字体给出 出错信息，你可以对已输入的语句进行修改，再运行。如果运行时间太长，你可以通过Alt+.(Alt+句号)来中止求值。

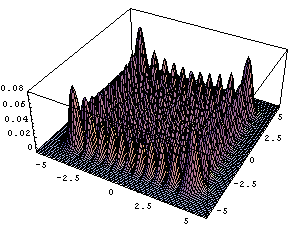
　　4. 对函数名不确定的，可先输入前面几个字母(开头一定要大写)，然后按Ctrl+K，系统会自动补全该函数名。

**七.应用例子**

量子一维、二维简谐振子问题

量子一维简谐振子图像

#### 

量子二维简谐振子图像