2018 年 5 月 1 日

Maxima/wxMaxima
符号计算软件

吕 荐 瑞

暨南大学数学系

软件简介

Maxima 是一个符号计算软件，而 wxMaxima 是它的图形化前端。
下载地址：http://maxima.sourceforge.net

1. 基本运算
2. 代数运算
3. 微积分计算
4. 函数绘图
5. 数论计算
6. 进阶应用

1.1 基本操作

1.2 四则运算

1.3 幂次运算

1.4 数学常数

1.5 初等函数

1.6 集合运算

基本操作

打开 wxMaxima，输入命令，比如如下命令

```
(1+2)*(3-4)/5;
```

然后按 Shift+Enter 运行命令，就得到结果。

文档格式

在保存 wxMaxima 文档时，可以选择保存为 .wxm 格式或者 .wxmx 格式。这两者的区别在于：

- .wxm 格式不保存命令的输出结果；
- .wxmx 格式保存命令的输出结果。

1.1 基本操作

1.2 四则运算

1.3 幂次运算

1.4 数学常数

1.5 初等函数

1.6 集合运算

四则运算

前面的命令包含了四则运算：加（+），减（-），乘（*），除（/）：

```
(1+2)*(3-4)/5;
```

注意：乘号在任何时候都不能省略。因此上述命令去掉 * 号是错误的：

```
(1+2)(3-4)/5;
```

四则运算

默认情形，运算结果使用分数来表示的：

```
(1+2)*(3-4)/5;
```

如果需要用小数来表示，可以在命令后面加上 numer 选项：

```
(1+2)*(3-4)/5,numer;
```

1.1 基本操作

1.2 四则运算

1.3 幂次运算

1.4 数学常数

1.5 初等函数

1.6 集合运算

幂次运算

幂次用 ^ 符号表示。例如，下列命令

```
3^2-2^3+8^(1/3);
```

可以计算出 $3^2 - 2^3 + 8^{1/3} = 3$。
特别地，二次根号也可用 sqrt 命令。例如：

```
sqrt(5)*sqrt(5);
```

可以计算出 $\sqrt{5} \cdot \sqrt{5} = 5$。

1.1 基本操作

1.2 四则运算

1.3 幂次运算

1.4 数学常数

1.5 初等函数

1.6 集合运算

数学常数

常用的数学常数用下面方式表示：

- 圆周率用 %pi 表示
- 虚数单位用 %i 表示
- 自然对数的底用 %e 表示

因此，输入下列命令

```
%e^(%pi*%i);
```

可以计算出 $e^{\pi i} = -1$。

1.1 基本操作

1.2 四则运算

1.3 幂次运算

1.4 数学常数

1.5 初等函数

1.6 集合运算

初等函数

常用的初等函数用下面方式表示：

- 绝对值函数用 abs 表示
- 自然指数函数用 exp 表示
- 自然对数函数用 log 表示
- 三角函数用 sin，cot 等表示
- 反三角函数用 asin，acot 等表示

因此下列命令计算 $\arcsin 0 + \arccos 0 = \pi/2$：

```
asin(0)+acos(0);
```

1.1 基本操作

1.2 四则运算

1.3 幂次运算

1.4 数学常数

1.5 初等函数

1.6 集合运算

集合运算

定义一个集合：

```
A: {1,2,3};
```

定义集合的另一种方法：

```
B: set(1,3,5);
```

求两个集合的交：

```
intersect(A,B);
```

求两个集合的并：

```
union(A,B);
```

1. 基本运算
2. 代数运算
3. 微积分计算
4. 函数绘图
5. 数论计算
6. 进阶应用

2.1 求和运算

2.2 多项式运算

2.3 方程求解

2.4 矩阵运算

求和运算

在 Maxima 中用 sum 命令作求和运算。例如下面的命令：

```
sum(k^2,k,1,100);
```

计算了 $\sum_{k=1}^{100} k^2$，即

$$1^2+2^2+3^2+\cdots+100^2=338350$$

求和运算

将前面的命令修改为如下命令：

```
sum(k^2,k,1,n),simpsum;
```

我们计算了 $\sum_{k=1}^{n} k^2$，即

$$1^2+2^2+3^2+\cdots+n^2=\frac{2n^3+3n^2+n}{6}$$

注意此时需要加上 simpsum 选项来化简求和。

2.1 求和运算

2.2 多项式运算

2.3 方程求解

2.4 矩阵运算

多项式运算

展开多项式：

```
expand((x+y+x*y)*(x^2+y+1));
```

分解多项式：

```
factor(x^3+x^2*y^2+x*y+y^3);
```

多项式运算

将假分式表示为多项式与真分式之和：

```
partfrac((x^3+3)/(x^2-1),x);
```

这个命令计算出 $\dfrac{x^3+3}{x^2-1} = x - \dfrac{1}{x+1} + \dfrac{2}{x-1}$.

2.1 求和运算

2.2 多项式运算

2.3 方程求解

2.4 矩阵运算

方程求解

解高次方程：

```
solve(x^3+2*x^2-x+1=0,x);
```

解方程组：

```
solve([a+b=3,a*b=2],[a,b]);
```

注意：乘号在任何情形都不能省略，即使是在一个常数和变量相乘时。

2.1 求和运算

2.2 多项式运算

2.3 方程求解

2.4 矩阵运算

矩阵运算

定义矩阵（变量后面加冒号表示定义一个变量）：

```
A: matrix(
    [2,7,6],
    [9,5,1],
    [4,3,8] );

B: matrix(
    [1,2,3],
    [4,5,6],
    [7,8,9] );
```

矩阵运算

作矩阵运算，比如 $2A + 3B - AB$：

```
2*A+3*B-A.B;
```

其中，矩阵乘法用 . 表示.

求 A 的转置 A^{T}：

```
transpose(A);
```

矩阵运算

求方阵 A 的幂次 A^2：

```
A^^2;
```

其中，矩阵的幂次用 ^^ 表示。

求方阵 A 的行列式 $|A|$：

```
determinant(A);
```

矩阵运算

求 A 的伴随矩阵 A^*：

```
adjoint(A);
```

求 A 的逆矩阵 A^{-1}：

```
invert(A);
```

这等同于下列幂次命令：

```
A^^-1;
```

矩阵运算

求 A 的秩 $r(A)$：

```
rank(A);
```

矩阵运算

求 A 的特征多项式：

```
charpoly(A,x);
```

也可以同时将特征多项式展开：

```
expand(charpoly(A,x));
```

矩阵运算

求 A 的特征值：

```
eigenvalues(A);
```

输出的结果中，先显示 A 的几个特征值，再显示各个特征值的重数.

矩阵运算

求 A 的特征向量：

```
eigenvectors(A);
```

输出的结果中，先显示 A 的特征值及其重数，然后才是矩阵的各个特征值对应的特征向量.

矩阵运算

将 A 相似对角化：

```
simtran(A);
```

输出的结果中，先显示 A 的特征值及其重数，然后才是变换矩阵.

矩阵运算

对 A 中行向量作施密特正交化：

```
gramschmidt(A);
```

1. 基本运算
2. 代数运算
3. 微积分计算
4. 函数绘图
5. 数论计算
6. 进阶应用

3.1 极限运算

3.2 导数运算

3.3 积分运算

3.4 微分方程

极限运算

下面的命令求出极限 $\lim_{n\to\infty}\left(1-\frac{1}{n}\right)^n$：

```
limit((1-1/n)^n,n,inf);
```

正无穷大和负无穷大分别用 inf 和 minf 表示。

3.1 极限运算

3.2 导数运算

3.3 积分运算

3.4 微分方程

导数运算

一阶导数:

```
diff(%e^(-2*x),x);
```

二阶导数:

```
diff(%e^(-2*x),x,2);
```

偏导数:

```
diff(y^2*%e^(-2*x),x);
```

3.1 极限运算

3.2 导数运算

3.3 积分运算

3.4 微分方程

积分运算

不定积分：

```
integrate(x^2*%e^x,x);
```

定积分：

```
integrate(x^2*%e^x,x,0,1);
```

3.1 极限运算

3.2 导数运算

3.3 积分运算

3.4 微分方程

微分方程

下面命令求出一阶微分方程 $y' + y = x$ 的通解：

```
ode2('diff(y,x)+y=x,y,x);
```

注意导数命令 diff 前面为英文单引号'。
然后用下面的命令可以从初始值求出特解：

```
ic1(%, x=0, y=1);
```

在 Maxima 中，% 表示上一个命令的计算结果。

微分方程

下面命令求出二阶微分方程 $y'' + y' = x$ 的通解：

```
ode2('diff(y,x,2)+'diff(y,x)=x,y,x);
```

然后用下面的命令可以从初始值求出特解：

```
ic2(%, x=0, y=1, 'diff(y,x)=2);
```

1. 基本运算
2. 代数运算
3. 微积分计算
4. 函数绘图
5. 数论计算
6. 进阶应用

4.1 二维图形

4.2 三维图形

二维图形

在 Maxima 中用 plot2d 命令画曲线图形。例如：

```
plot2d(x*sin(1/x),[x,-1/10,1/10]);
```

二维图形

用 plot2d 命令也可以同时画几条曲线的图形。例如：

```
plot2d([sin(x),x],[x,-10,10]);
```

二维图形

用 plot2d 命令还可以画参数曲线的图形。例如：

```
plot2d([parametric,sin(t),cos(t)],
       [t,0,2*%pi]);
```

在 Maxima 中，在一条命令中间可以随意换行。

二维图形

画隐函数的曲线，需要先载入 implicit_plot 包：

```
load(implicit_plot);
```

然后设定首次采样和再次采样的格点数：

```
ip_grid:[100,100];
ip_grid_in:[10,10];
```

现在就可以画隐函数的曲线了，例如：

```
implicit_plot(y^2=x^3-3*x+1,
                    [x,-4,4],[y,-4,4]);
```

4.1 二维图形

4.2 三维图形

三维图形

在 Maxima 中用 plot3d 命令画曲面图形。例如：

```
plot3d(x^2-y^2,[x,-1,1],[y,-1,1]);
```

画出的图形可以用鼠标拖动任意旋转。

1. 基本运算
2. 代数运算
3. 微积分计算
4. 函数绘图
5. 数论计算
6. 进阶应用

5.1 整除与同余

5.2 素数的判别

5.3 二次剩余

5.4 递归序列

整除与同余

计算两个数的最大公因数：

```
gcd(24,32);
```

计算两个数的最小公倍数：

```
lcm(24,32);
```

整除与同余

计算一个数的所有因数之和：

```
divsum(28);
```

用中国剩余定理解同余式组：$\begin{cases} x \equiv 2 \pmod 3 \\ x \equiv 3 \pmod 5 \\ x \equiv 2 \pmod 7 \end{cases}$

```
chinese([2,3,2], [3,5,7]);
```

整除与同余

计算 $110 \pmod 7$：

```
mod(110,7);
```

计算 $3^{15} \pmod 7$：

```
power_mod(3,15,7);
```

计算 $3^{-1} \pmod{41}$：

```
inv_mod(3,41);
```

5.1 整除与同余

5.2 素数的判别

5.3 二次剩余

5.4 递归序列

素数的判别

判别一个数是否为素数：

```
primep(27);
```

寻找一个数之前的最大素数：

```
prev_prime(27);
```

寻找一个数之后的最小素数：

```
next_prime(27);
```

素数的判别

计算欧拉函数 $\phi(9)$:

```
totient(9);
```

分解因式:

```
factor(2^63-1);
```

5.1 整除与同余

5.2 素数的判别

5.3 二次剩余

5.4 递归序列

二次剩余

计算 Jacobi 符号 $\left(\frac{37603}{48611}\right)$:

```
jacobi(37603,48611);
```

5.1 整除与同余

5.2 素数的判别

5.3 二次剩余

5.4 递归序列

递归序列

计算斐波那契数列 F_n 的前几项：

```
fib(1); fib(2); fib(3); fib(4);
fib(5); fib(6); fib(7); fib(8);
```

或者可以用 map 函数：

```
map(fib, [1,2,3,4,5,6,7,8]);
```

1. 基本运算
2. 代数运算
3. 微积分计算
4. 函数绘图
5. 数论计算
6. 进阶应用

6.1 变量定义

6.2 函数定义

6.3 条件循环

6.4 输入输出

6.5 继续学习

变量定义

变量后面加 ：表示定义一个变量。
例如，先定义 $a = 12345$：

```
a: 12345;
```

再定义 $b = 54321$：

```
b: 54321;
```

然后计算两者的平方和 $a^2 + b^2$：

```
a^2+b^2;
```

在 Maxima 中，每个命令用分号 ；结束，而多个命令可以写在同一行。例如，之前的例子

```
a:12345;
b:54321;
a^2+b^2;
```

可以简单的写成一行：

```
a:12345; b:54321; a^2+b^2;
```

如果将命令结束的 ；改为 $ 号，则不显示该命令的输出结果。

6.1 变量定义

6.2 函数定义

6.3 条件循环

6.4 输入输出

6.5 继续学习

函数定义

下面的命令定义了一个函数（用 := 可以定义一个函数，注意上面定义变量只用 :）

```
maximum(a,b) :=
    (if a>=b then c:a else c:b, c);
```

注意函数定义里的最后一个语句是该函数的返回值。运行此命令后就可以用下面语句调用此函数：

```
maximum(2.71, 2.718);
```

6.1 变量定义

6.2 函数定义

6.3 条件循环

6.4 输入输出

6.5 继续学习

条件循环

Maxima 也可以自己编程增强其功能，因为它提供了各种条件判断和循环命令，例如：

```
for a:1 thru 9 step 2 do display(a);

for i:1 while i<=9 do display(i^2);
```

6.1 变量定义

6.2 函数定义

6.3 条件循环

6.4 输入输出

6.5 继续学习

输出变量

用 disp 和 display 命令可显示变量的值. 假设

```
d:3; a:"hello";
```

则用 `disp(d^2, a);` 将显示下列结果

```
9
"hello"
```

而用 `display(d^2, a);` 将显示下列结果

```
d = 9
a = "hello"
```

输出变量

用 print 命令可在同一行显示多个变量值. 例如

```
print(d^2, a)$
```

将显示下列结果

```
9 "hello"
```

还可以用 printf 命令格式化输出。例如

```
printf(false, "~d and ~a", d^2, a);
```

将显示（其中 ~d 和 ~a 分别表示整数和字符串）

```
9 and "hello"
```

写入文件

下面的代码用于将多个平方数写入文件

```
with_stdout("d:/somefile.txt",
  for n:1 thru 10 do
      printf(true, "~d ", n^2)
);
```

此时 d:\somefile.txt 文件的内容如下

```
1 4 9 16 25 36 49 64 81 100
```

写入文件

还有另一种方法，实现将多个平方数写入文件

```
s: openw("d:/somefile.txt");
for n:1 thru 10 do
    printf(s, "~d ", n^2);
close(s);
```

注意 printf 的第一个参数的写法和之前的不同.

读取文件

读取文件和写入文件的方法类似. 比如

```
s: openr("somefile.txt");
while stringp(line:readline(s)) do
    print(line);
close(s);
```

将把 somefile.txt 文件的内容逐行显示出来.

6.1 变量定义

6.2 函数定义

6.3 条件循环

6.4 输入输出

6.5 继续学习

继续学习

- 找一些极限或积分试试，看有哪些 Maxima 不能求出来。
- 可以看看 Maxima 的帮助文档，里面有非常多的命令。