Maxima 介绍

李东风 制作

2014.12.

目录

本节目录

Maxima 是什么

- 计算机代数软件可以进行多项式化简、分解因式、分式化简、通分、积分、微分等公式推导。这种运算又称为“符号计算”，与“数值计算”相对。
- 著名的计算机代数系统有 Maple、Mathematica、Maxima。
- Maxima 是一个免费的计算机代数软件，可安装在各种不同的操作系统中。
- 具有计算机代数系统必须的各种功能。
- 用 Lisp 语言编制，是一个有悠久历史的软件，开发历史可追溯至 1968 年到 1982 年之间在 MIT 开发的 DOE Macsyma，后来转由 Texas 大学 William F. Schelter 教授维护，在 1998 年转为 GNU GPL 版权协议。

Maxima 是什么

- 计算机代数软件可以进行多项式化简、分解因式、分式化简、通分、积分、微分等公式推导。这种运算又称为“符号计算”，与“数值计算”相对。
- 著名的计算机代数系统有 Maple、Mathematica、Maxima。
- Maxima 是一个免费的计算机代数软件，可安装在各种不同的操作系统中。
- 具有计算机代数系统必须的各种功能。
- 用 Lisp 语言编制，是一个有悠久历史的软件，开发历史可追溯至 1968 年到 1982 年之间在 MIT 开发的 DOE Macsyma, 后来转由 Texas 大学 William F. Schelter 教授维护，在 1998 年转为 GNU GPL 版权协议。

Maxima 是什么

- 计算机代数软件可以进行多项式化简、分解因式、分式化简、通分、积分、微分等公式推导。这种运算又称为“符号计算”，与“数值计算”相对。
- 著名的计算机代数系统有 Maple、Mathematica、Maxima。
- Maxima 是一个免费的计算机代数软件，可安装在各种不同的操作系统中。
- 具有计算机代数系统必须的各种功能。
- 用 Lisp 语言编制，是一个有悠久历史的软件，开发历史可追溯至 1968 年到 1982 年之间在 MIT 开发的 DOE Macsyma, 后来转由 Texas 大学 William F. Schelter 教授维护，在 1998 年转为 GNU GPL 版权协议。

Maxima 是什么

- 计算机代数软件可以进行多项式化简、分解因式、分式化简、通分、积分、微分等公式推导。这种运算又称为“符号计算”，与“数值计算”相对。
- 著名的计算机代数系统有 Maple、Mathematica、Maxima。
- Maxima 是一个免费的计算机代数软件，可安装在各种不同的操作系统中。
- 具有计算机代数系统必须的各种功能。
- 用 Lisp 语言编制，是一个有悠久历史的软件，开发历史可追溯至 1968 年到 1982 年之间在 MIT 开发的 DOE Macsyma, 后来转由 Texas 大学 William F. Schelter 教授维护，在 1998 年转为 GNU GPL 版权协议。

Maxima 是什么

- 计算机代数软件可以进行多项式化简、分解因式、分式化简、通分、积分、微分等公式推导。这种运算又称为“符号计算”，与“数值计算”相对。
- 著名的计算机代数系统有 Maple、Mathematica、Maxima。
- Maxima 是一个免费的计算机代数软件，可安装在各种不同的操作系统中。
- 具有计算机代数系统必须的各种功能。
- 用 Lisp 语言编制，是一个有悠久历史的软件，开发历史可追溯至 1968 年到 1982 年之间在 MIT 开发的 DOE Macsyma, 后来转由 Texas 大学 William F. Schelter 教授维护，在 1998 年转为 GNU GPL 版权协议。

安装与运行

- 在 MS Windows 系统中，可以下载安装 wxMaxima 软件 (http://http://andrejv.github.io/wxmaxima/index.html)。
- wxMaxima 运行界面是一个窗口形式，在窗口中逐行输入命令，命令以分号结尾，用 Shift+RETURN 键运行每行命令。命令用 $ 结尾则不显示命令的结果。
- 可以用% 访问上一个公式的输出；用%i1, %o1 等访问历史的输入与输出。
- 用`%th(k)`访问向上数第 k 个结果。
- 用冒号定义一个变量。

安装与运行

- 在 MS Windows 系统中，可以下载安装 wxMaxima 软件 (http://http://andrejv.github.io/wxmaxima/index.html)。
- wxMaxima 运行界面是一个窗口形式，在窗口中逐行输入命令，命令以分号结尾，用 Shift+RETURN 键运行每行命令。命令用 $ 结尾则不显示命令的结果。
- 可以用% 访问上一个公式的输出；用%i1, %o1 等访问历史的输入与输出。
- 用`%th(k)`访问向上数第 k 个结果。
- 用冒号定义一个变量。

安装与运行

- 在 MS Windows 系统中，可以下载安装 wxMaxima 软件 (http://http://andrejv.github.io/wxmaxima/index.html)。
- wxMaxima 运行界面是一个窗口形式，在窗口中逐行输入命令，命令以分号结尾，用 Shift+RETURN 键运行每行命令。命令用 $ 结尾则不显示命令的结果。
- 可以用% 访问上一个公式的输出；用%i1, %o1 等访问历史的输入与输出。
- 用`%th(k)`访问向上数第 k 个结果。
- 用冒号定义一个变量。

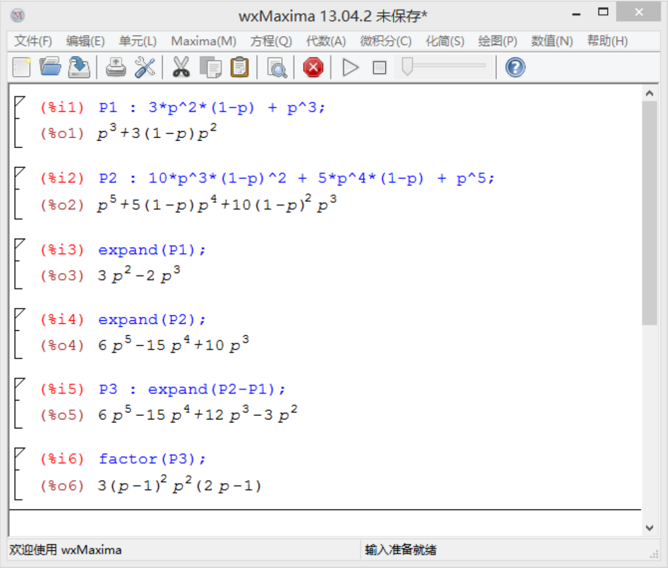
安装与运行

- 在 MS Windows 系统中，可以下载安装 wxMaxima 软件 (http://http://andrejv.github.io/wxmaxima/index.html)。
- wxMaxima 运行界面是一个窗口形式，在窗口中逐行输入命令，命令以分号结尾，用 Shift+RETURN 键运行每行命令。命令用 $ 结尾则不显示命令的结果。
- 可以用% 访问上一个公式的输出；用%i1, %o1 等访问历史的输入与输出。
- 用`%th(k)` 访问向上数第 k 个结果。
- 用冒号定义一个变量。

安装与运行

- 在 MS Windows 系统中，可以下载安装 wxMaxima 软件 (http://http://andrejv.github.io/wxmaxima/index.html)。
- wxMaxima 运行界面是一个窗口形式，在窗口中逐行输入命令，命令以分号结尾，用 Shift+RETURN 键运行每行命令。命令用 $ 结尾则不显示命令的结果。
- 可以用% 访问上一个公式的输出；用%i1, %o1 等访问历史的输入与输出。
- 用`%th(k)` 访问向上数第 k 个结果。
- 用冒号定义一个变量。

运行界面示例


wxMaxima 13.04.2 未保存*
文件(F) 编辑(E) 单元(L) Maxima(M) 方程(Q) 代数(A) 微积分(C) 化简(S) 绘图(P) 数值(N) 帮助(H)
(%i1) P1 : 3*p^2*(1-p) + p^3;
(%o1) $p^3+3(1-p)p^2$
(%i2) P2 : 10*p^3*(1-p)^2 + 5*p^4*(1-p) + p^5;
(%o2) $p^5+5(1-p)p^4+10(1-p)^2p^3$
(%i3) expand(P1);
(%o3) $3p^2-2p^3$
(%i4) expand(P2);
(%o4) $6p^5-15p^4+10p^3$
(%i5) P3 : expand(P2-P1);
(%o5) $6p^5-15p^4+12p^3-3p^2$
(%i6) factor(P3);
(%o6) $3(p-1)^2p^2(2p-1)$
欢迎使用 wxMaxima
输入准备就绪

精确计算

- 四则运算符号用 `+ - * /`, 乘方用 `^` 表示。
- Maxima 中不带小数点的四则计算结果保存为有理式，不作近似。如

```
12/5 + 6*(2^3 + 1/11);
```

结果为

$$\frac{2802}{55}$$

精确计算

- 四则运算符号用 `+ - * /`, 乘方用 `^` 表示。
- Maxima 中不带小数点的四则计算结果保存为有理式，不作近似。如

```
12/5 + 6*(2^3 + 1/11);
```

结果为

$$\frac{2802}{55}$$

近似计算

- 为了求以上结果的近似值，用 numer 函数开关，或 float 函数，如

```
float(%);
```

或

```
12/5 + 6*(2^3 + 1/11), numer;
```

结果为 50.94545454545455。

- 这样，Maxima 可以当作一个高级计算器使用。

近似计算

- 为了求以上结果的近似值，用 numer 函数开关，或 float 函数，如

```
float(%);
```

或

```
12/5 + 6*(2^3 + 1/11), numer;
```

结果为 50.94545454545455。

- 这样，Maxima 可以当作一个高级计算器使用。

数值类型

- Maxima 的数值类型有整数、有理数、普通浮点数、高精度的浮点数。
- 普通浮点数如 1.2, 1.2E-3。
- 高精度浮点数如 1.2B0, 1.2B-3, 有效位数用变量 fpprec 控制。
- 函数 `float(x)` 把 x 转为浮点型，函数 `bfloat(x)`x 转为高精度浮点型。
- 有高精度浮点数参加运算的算式结果为高精度浮点数；没有高精度浮点数但有浮点数算式结果为浮点数。

数值类型

- Maxima 的数值类型有整数、有理数、普通浮点数、高精度的浮点数。
- 普通浮点数如 1.2, 1.2E-3。
- 高精度浮点数如 1.2B0, 1.2B-3, 有效位数用变量 fpprec 控制。
- 函数 `float(x)` 把 `x` 转为浮点型，函数 `bfloat(x)x` 转为高精度浮点型。
- 有高精度浮点数参加运算的算式结果为高精度浮点数；没有高精度浮点数但有浮点数算式结果为浮点数。

数值类型

- Maxima 的数值类型有整数、有理数、普通浮点数、高精度的浮点数。
- 普通浮点数如 1.2, 1.2E-3。
- 高精度浮点数如 1.2B0, 1.2B-3, 有效位数用变量 fpprec 控制。
- 函数 `float(x)` 把 `x` 转为浮点型，函数 `bfloat(x)x` 转为高精度浮点型。
- 有高精度浮点数参加运算的算式结果为高精度浮点数；没有高精度浮点数但有浮点数算式结果为浮点数。

数值类型

- Maxima 的数值类型有整数、有理数、普通浮点数、高精度的浮点数。
- 普通浮点数如 1.2, 1.2E-3。
- 高精度浮点数如 1.2B0, 1.2B-3, 有效位数用变量 fpprec 控制。
- 函数 `float(x)` 把 `x` 转为浮点型，函数 `bfloat(x)x` 转为高精度浮点型。
- 有高精度浮点数参加运算的算式结果为高精度浮点数；没有高精度浮点数但有浮点数算式结果为浮点数。

数值类型

- Maxima 的数值类型有整数、有理数、普通浮点数、高精度的浮点数。
- 普通浮点数如 1.2, 1.2E-3。
- 高精度浮点数如 1.2B0, 1.2B-3, 有效位数用变量 fpprec 控制。
- 函数 `float(x)` 把 `x` 转为浮点型，函数 `bfloat(x)x` 转为高精度浮点型。
- 有高精度浮点数参加运算的算式结果为高精度浮点数；没有高精度浮点数但有浮点数算式结果为浮点数。

特殊常数

- 用%e、%pi 表示 e 和 π, 用%i 表示虚数单位。
- inf, minf, 分别表示 $+\infty$, $-\infty$, infinity 表示复数无穷大。

特殊常数

- 用`%e`、`%pi` 表示 e 和 π, 用`%i` 表示虚数单位。
- `inf`, `minf`, 分别表示 $+\infty$, $-\infty$, `infinity` 表示复数无穷大。

复数

- 用%i 表示虚数单位，如此可以输入复数，如

```
z1 : 5 + 3*%i;
```

- 用 realpart 和 imagpart 返回复数的实部和虚部。
- 用 conjugate 返回复数的复共轭。
- 用 abs 求复数模，用 carg 和求辐角。

复数

- 用%i 表示虚数单位，如此可以输入复数，如

```
z1 : 5 + 3*%i;
```

- 用 realpart 和 imagpart 返回复数的实部和虚部。
- 用 conjugate 返回复数的复共轭。
- 用 abs 求复数模，用 carg 和求辐角。

复数

- 用%i 表示虚数单位，如此可以输入复数，如

```
z1 : 5 + 3*%i;
```

- 用 realpart 和 imagpart 返回复数的实部和虚部。
- 用 conjugate 返回复数的复共轭。
- 用 abs 求复数模，用 carg 和求辐角。

复数

- 用%i 表示虚数单位，如此可以输入复数，如

```
z1 : 5 + 3*%i;
```

- 用 realpart 和 imagpart 返回复数的实部和虚部。
- 用 conjugate 返回复数的复共轭。
- 用 abs 求复数模，用 carg 和求辐角。

- `rectform` 把复数表为直角坐标形式，如

```
rectform(z1);
```

显示为

$$3\,i+5$$

- `polarform` 把复数表为直角坐标形式，如

```
polarform(z1);
```

显示为

$$\sqrt{34}\,e^{i\,\mathrm{atan}\left(\frac{3}{5}\right)}$$

- rectform 把复数表为直角坐标形式，如

```
rectform(z1);
```

显示为

$$3\,i+5$$

- polarform 把复数表为直角坐标形式，如

```
polarform(z1);
```

显示为

$$\sqrt{34}\,e^{i\,\mathrm{atan}\left(\frac{3}{5}\right)}$$

列表

- 用方括号把逗号分隔的数值组合在一起作为一个列表，如

```
[2,3,4] + [7,8,9]
```

结果为

```
[9,11,13]
```

本节目录

多项式和分式计算

多项式合并

- 考虑如下的一个推导问题。甲、乙比赛，每局比赛甲赢的概率 $p \in (0.5, 1)$。
- 问：三局两胜还是五局三胜对甲有利？
- 三局两胜中甲赢的概率为

$$
\begin{aligned}
&P(\text{三局两胜中甲赢}) = P(\text{甲赢 2 局}) + P(\text{甲赢 3 局}) \\
=&C_3^2 p^2(1-p) + p^3 = 3p^2 - 2p^3,
\end{aligned}
$$

- 五局三胜中甲赢的概率为

$$
\begin{aligned}
&P(\text{五局三胜中甲赢}) \\
=&P(\text{甲赢 3 局}) + P(\text{甲赢 4 局}) + P(\text{甲赢 5 局}) \\
=&C_5^3 p^3(1-p)^2 + C_5^4 p^4(1-p) + p^5 = 10p^3 - 15p^4 + 6p^5,
\end{aligned}
$$

多项式合并

- 考虑如下的一个推导问题。甲、乙比赛，每局比赛甲赢的概率 $p \in (0.5, 1)$。
- 问：三局两胜还是五局三胜对甲有利？
- 三局两胜中甲赢的概率为

$$P(\text{三局两胜中甲赢}) = P(\text{甲赢 2 局}) + P(\text{甲赢 3 局})$$
$$= C_3^2 p^2 (1-p) + p^3 = 3p^2 - 2p^3,$$

- 五局三胜中甲赢的概率为

$$P(\text{五局三胜中甲赢})$$
$$= P(\text{甲赢 3 局}) + P(\text{甲赢 4 局}) + P(\text{甲赢 5 局})$$
$$= C_5^3 p^3 (1-p)^2 + C_5^4 p^4 (1-p) + p^5 = 10p^3 - 15p^4 + 6p^5,$$

多项式合并

- 考虑如下的一个推导问题。甲、乙比赛，每局比赛甲赢的概率 $p \in (0.5, 1)$。
- 问：三局两胜还是五局三胜对甲有利？
- 三局两胜中甲赢的概率为

$$P(\text{三局两胜中甲赢}) = P(\text{甲赢 2 局}) + P(\text{甲赢 3 局})$$
$$= C_3^2 p^2 (1-p) + p^3 = 3p^2 - 2p^3,$$

- 五局三胜中甲赢的概率为

$$P(\text{五局三胜中甲赢})$$
$$= P(\text{甲赢 3 局}) + P(\text{甲赢 4 局}) + P(\text{甲赢 5 局})$$
$$= C_5^3 p^3 (1-p)^2 + C_5^4 p^4 (1-p) + p^5 = 10p^3 - 15p^4 + 6p^5,$$

多项式合并

- 考虑如下的一个推导问题。甲、乙比赛，每局比赛甲赢的概率 $p \in (0.5, 1)$。
- 问：三局两胜还是五局三胜对甲有利?
- 三局两胜中甲赢的概率为

$$
\begin{aligned}
& P(\text{三局两胜中甲赢}) = P(\text{甲赢 2 局}) + P(\text{甲赢 3 局}) \\
=& C_3^2 p^2 (1-p) + p^3 = 3p^2 - 2p^3,
\end{aligned}
$$

- 五局三胜中甲赢的概率为

$$
\begin{aligned}
& P(\text{五局三胜中甲赢}) \\
=& P(\text{甲赢 3 局}) + P(\text{甲赢 4 局}) + P(\text{甲赢 5 局}) \\
=& C_5^3 p^3 (1-p)^2 + C_5^4 p^4 (1-p) + p^5 = 10p^3 - 15p^4 + 6p^5,
\end{aligned}
$$

- 在 wxMaxima 中用如下两个命令定义了这两个概率:

```
P1 : 3*p^2*(1-p) + p^3;
P2 : 10*p^3*(1-p)^2 + 5*p^4*(1-p) + p^5;
```

- 用 expand 函数可以把多项式合并同类项化简:

```
expand(P1);
expand(P2);
```

- 如下命令计算两个多项式的差并合并同类项:

```
P3 : expand(P2-P1);
```

- 结果得到两个概率的差为

$$6p^5 - 15p^4 + 12p^3 - 3p^2$$

- 在 wxMaxima 中用如下两个命令定义了这两个概率:

```
P1 : 3*p^2*(1-p) + p^3;
P2 : 10*p^3*(1-p)^2 + 5*p^4*(1-p) + p^5;
```

- 用 `expand` 函数可以把多项式合并同类项化简:

```
expand(P1);
expand(P2);
```

- 如下命令计算两个多项式的差并合并同类项:

```
P3 : expand(P2-P1);
```

- 结果得到两个概率的差为

$$6p^5 - 15p^4 + 12p^3 - 3p^2$$

- 在 wxMaxima 中用如下两个命令定义了这两个概率:

```
P1 : 3*p^2*(1-p) + p^3;
P2 : 10*p^3*(1-p)^2 + 5*p^4*(1-p) + p^5;
```

- 用 expand 函数可以把多项式合并同类项化简:

```
expand(P1);
expand(P2);
```

- 如下命令计算两个多项式的差并合并同类项:

```
P3 : expand(P2-P1);
```

- 结果得到两个概率的差为

$$6p^5 - 15p^4 + 12p^3 - 3p^2$$

- 在 wxMaxima 中用如下两个命令定义了这两个概率:

```
P1 : 3*p^2*(1-p) + p^3;
P2 : 10*p^3*(1-p)^2 + 5*p^4*(1-p) + p^5;
```

- 用 expand 函数可以把多项式合并同类项化简:

```
expand(P1);
expand(P2);
```

- 如下命令计算两个多项式的差并合并同类项:

```
P3 : expand(P2-P1);
```

- 结果得到两个概率的差为

$$6p^5 - 15p^4 + 12p^3 - 3p^2$$

多项式因式分解

- 以上的两种赛制甲获胜概率之差简单地化为

$$3p^2(2p^3 - 5p^2 + 4p - 1),$$

- 为判断其是否在 $p \in (0.5, 1)$ 总为正值，可以分解因式。
- Maxima 用 `factor` 函数分解因式，如:

```
factor(P3);
```

- 结果为

$$3\,(p-1)^2\,p^2\,(2\,p-1)\,.$$

在 $p \in (0.5, 1)$ 为正值。

多项式因式分解

- 以上的两种赛制甲获胜概率之差简单地化为

$$3p^2(2p^3 - 5p^2 + 4p - 1),$$

- 为判断其是否在 $p \in (0.5, 1)$ 总为正值，可以分解因式。
- Maxima 用 `factor` 函数分解因式，如：

```
factor(P3);
```

- 结果为

$$3\,(p-1)^2\,p^2\,(2\,p-1)\,.$$

在 $p \in (0.5, 1)$ 为正值。

多项式因式分解

- 以上的两种赛制甲获胜概率之差简单地化为

$$3p^2(2p^3 - 5p^2 + 4p - 1),$$

- 为判断其是否在 $p \in (0.5, 1)$ 总为正值，可以分解因式。
- Maxima 用 `factor` 函数分解因式，如:

```
factor(P3);
```

- 结果为

$$3\,(p-1)^2\,p^2\,(2\,p-1)\,.$$

在 $p \in (0.5, 1)$ 为正值。

多项式因式分解

- 以上的两种赛制甲获胜概率之差简单地化为

$$3p^2(2p^3 - 5p^2 + 4p - 1),$$

- 为判断其是否在 $p \in (0.5, 1)$ 总为正值，可以分解因式。
- Maxima 用 `factor` 函数分解因式，如:

```
factor(P3);
```

- 结果为

$$3\,(p-1)^2\,p^2\,(2\,p-1)\,.$$

在 $p \in (0.5, 1)$ 为正值。

推导结果导出

- wxMaxima 的结果显示为数学公式形式，选中结果，可以选“编辑—复制为 LaTeX”菜单，把结果导出到 LaTeX 文件中。
- 也可以把公式复制为纯文本、图片。

推导结果导出

- wxMaxima 的结果显示为数学公式形式，选中结果，可以选“编辑—复制为 LaTeX”菜单，把结果导出到 LaTeX 文件中。
- 也可以把公式复制为纯文本、图片。

分式通分合并

- 考虑如下分式:

```
R1 : (x+1)^2/(x-1) + 1/(x+1);
```

结果为

$$\frac{(x+1)^2}{x-1}+\frac{1}{x+1}$$

- expand 函数可以分别展开分子分母中的多项式，但不能通分合并:

```
expand(R1);
```

结果为

$$\frac{1}{x+1}+\frac{x^2}{x-1}+\frac{2\,x}{x-1}+\frac{1}{x-1}$$

分式通分合并

- 考虑如下分式:

```
R1 : (x+1)^2/(x-1) + 1/(x+1);
```

结果为

$$\frac{(x+1)^2}{x-1} + \frac{1}{x+1}$$

- expand 函数可以分别展开分子分母中的多项式，但不能通分合并:

```
expand(R1);
```

结果为

$$\frac{1}{x+1} + \frac{x^2}{x-1} + \frac{2\,x}{x-1} + \frac{1}{x-1}$$

- 函数 ratexpand 可以把分式的分子分母中多项式合并同类项，然后通分，结果表示为分子每个不同幂次单独一个分式的加法：

```
ratexpand(R1);
```

结果为

$$\frac{x^3}{x^2-1}+\frac{3\,x^2}{x^2-1}+\frac{4\,x}{x^2-1}$$

- 函数 ratsimp 把分式之和通分合并变成一个分式：

```
ratsimp(R1);
```

结果为

$$\frac{x^3+3\,x^2+4\,x}{x^2-1}$$

- 函数 ratexpand 可以把分式的分子分母中多项式合并同类项，然后通分，结果表示为分子每个不同幂次单独一个分式的加法：

```
ratexpand(R1);
```

结果为

$$\frac{x^3}{x^2-1}+\frac{3\,x^2}{x^2-1}+\frac{4\,x}{x^2-1}$$

- 函数 ratsimp 把分式之和通分合并变成一个分式：

```
ratsimp(R1);
```

结果为

$$\frac{x^3+3\,x^2+4\,x}{x^2-1}$$

- factor 函数可以对分式的分子分母分别进行因式分解，如

```
factor(%);
```

结果为

$$\frac{x\left(x^2+3\,x+4\right)}{(x-1)\,(x+1)}$$

本节目录

三角函数化简

三角函数化简

- Maxima 用如下函数进行三角函数化简:
- `trigexpand` 和差化积;
- `trigreduce` 积化和差;
- `trigsimp` 利用 $\sin^2 x + \cos^2 x = 1$ 之类化简;
- `trigrat` 把三角有理分式化为 sin 和 cos 的线性函数。

三角函数化简

- Maxima 用如下函数进行三角函数化简:
- `trigexpand` 和差化积;
- `trigreduce` 积化和差;
- `trigsimp` 利用 $\sin^2 x + \cos^2 x = 1$ 之类化简;
- `trigrat` 把三角有理分式化为 sin 和 cos 的线性函数。

三角函数化简

- Maxima 用如下函数进行三角函数化简:
- `trigexpand` 和差化积;
- `trigreduce` 积化和差;
- `trigsimp` 利用 $\sin^2 x + \cos^2 x = 1$ 之类化简;
- `trigrat` 把三角有理分式化为 sin 和 cos 的线性函数。

三角函数化简

- Maxima 用如下函数进行三角函数化简:
- `trigexpand` 和差化积;
- `trigreduce` 积化和差;
- `trigsimp` 利用 $\sin^2 x + \cos^2 x = 1$ 之类化简;
- `trigrat` 把三角有理分式化为 sin 和 cos 的线性函数。

三角函数化简

- Maxima 用如下函数进行三角函数化简:
- `trigexpand` 和差化积;
- `trigreduce` 积化和差;
- `trigsimp` 利用 $\sin^2 x + \cos^2 x = 1$ 之类化简;
- `trigrat` 把三角有理分式化为 sin 和 cos 的线性函数。

和差化积

- 考虑如下三角函数有理式:

```
T1 : sin(2*x)/cos(x) + cos(2*x);
```

结果显示为

$$\frac{\sin\left(2\,x\right)}{\cos\left(x\right)}+\cos\left(2\,x\right)$$

- trigexpand 把 $\sin(2x)$ 展开为 $2\sin x\cos x$, 把 $\cos(2x)$ 展开为 $\cos^2 x-\sin^2 x$:

```
T2 : trigexpand(T1);
```

结果显示为

$$-\sin\left(x\right)^2+2\sin\left(x\right)+\cos\left(x\right)^2$$

和差化积

- 考虑如下三角函数有理式:

```
T1 : sin(2*x)/cos(x) + cos(2*x);
```

结果显示为

$$\frac{\sin\left(2\,x\right)}{\cos\left(x\right)}+\cos\left(2\,x\right)$$

- **trigexpand** 把 $\sin(2x)$ 展开为 $2\sin x\cos x$, 把 $\cos(2x)$ 展开为 $\cos^2 x-\sin^2 x$:

```
T2 : trigexpand(T1);
```

结果显示为

$$-\sin\left(x\right)^2+2\,\sin\left(x\right)+\cos\left(x\right)^2$$

- 上述结果中同时有 $\cos^2 x$ 和 $\sin^2 x$，用 `trigsimp` 化简

```
trigsimp(%);
```

结果为

$$2\sin(x) + 2\cos(x)^2 - 1$$

积化和差

- 如下程序中

```
T3 : trigreduce(T2);
```

$\sin^2 x$, $\cos^2 x$ 被化为一次式, 结果显示为

$$\frac{\cos(2\,x)+1}{2}+\frac{\cos(2\,x)}{2}+2\sin(x)-\frac{1}{2}$$

- 用 expand 简单地合并同类项:

```
expand(%);
```

结果为

$$\cos(2\,x)+2\sin(x)$$

- 用 trigsimp 或 trigrat 也可以合并同类项。

积化和差

- 如下程序中

```
T3 : trigreduce(T2);
```

$\sin^2 x$, $\cos^2 x$ 被化为一次式, 结果显示为

$$\frac{\cos(2\,x)+1}{2}+\frac{\cos(2\,x)}{2}+2\sin(x)-\frac{1}{2}$$

- 用 `expand` 简单地合并同类项:

```
expand(%);
```

结果为

$$\cos(2\,x)+2\sin(x)$$

- 用 `trigsimp` 或 `trigrat` 也可以合并同类项。

积化和差

- 如下程序中

```
T3 : trigreduce(T2);
```

$\sin^2 x$, $\cos^2 x$ 被化为一次式, 结果显示为

$$\frac{\cos(2\,x)+1}{2}+\frac{\cos(2\,x)}{2}+2\sin(x)-\frac{1}{2}$$

- 用 `expand` 简单地合并同类项:

```
expand(%);
```

结果为

$$\cos(2\,x)+2\sin(x)$$

- 用 `trigsimp` 或 `trigrat` 也可以合并同类项。

三角函数有理式化简

- 对 T1

$$\frac{\sin(2\,x)}{\cos(x)}+\cos(2\,x)$$

这样的三角函数有理分式，用 `ratsimp` 可以直接化简为三角函数线性式, 或分子、分母都是三角函数线性项的有理式:

```
trigrat(T1);
```

结果为

$$\cos(2\,x)+2\sin(x)$$

- 又如，

```
T4 : sin(x)^2 / cos(x) + cos(2*x);
```

为

$$\cos(2\,x) + \frac{\sin(x)^2}{\cos(x)}$$

- 用 trigrat 作用:

```
T4 : sin(x)^2 / cos(x) + cos(2*x);
```

结果为

$$\frac{\cos(3\,x) - \cos(2\,x) + \cos(x) + 1}{2\cos(x)}$$

- 又如,

```
T4 : sin(x)^2 / cos(x) + cos(2*x);
```

为

$$\cos\left(2\,x\right)+\frac{\sin\left(x\right)^2}{\cos\left(x\right)}$$

- 用 `trigrat` 作用:

```
T4 : sin(x)^2 / cos(x) + cos(2*x);
```

结果为

$$\frac{\cos\left(3\,x\right)-\cos\left(2\,x\right)+\cos\left(x\right)+1}{2\cos\left(x\right)}$$

- 三角函数有理分式 T4 又可以用 trigexpand 和 trigsimp 作用:

```
trigexpand(%);
```

结果为

$$\frac{-3\cos\left(x\right)\,\sin\left(x\right)^2+\sin\left(x\right)^2+\cos\left(x\right)^3-\cos\left(x\right)^2+\cos\left(x\right)+1}{2\cos\left(x\right)}$$

```
trigsimp(%);
```

结果为

$$\frac{2\cos\left(x\right)^3-\cos\left(x\right)^2-\cos\left(x\right)+1}{\cos\left(x\right)}$$

本节目录

函数和微积分

常用初等函数

sqrt()	abs()	max()	min()	sign()	
exp()	log()				
sin()	cos()	tan()	cot()	sec()	csc()
asin()	acos()	atan()	acot()	asec()	acsc()
sinh()	cosh()	tanh()	coth()	sech()	csch()
asinh()	acosh()	atanh()	acoth()	asech()	acsch()

组合函数

- 用 $x!$ 计算阶乘，如

```
5!;
```

结果为 120。

- 用 binomial(n, k) 计算 C_n^k，如

```
binomial(5,2);
```

结果为 10。

组合函数

- 用 $x!$ 计算阶乘，如

```
5!;
```

结果为 120。

- 用 `binomial`(n, k) 计算 C_n^k, 如

```
binomial(5,2);
```

结果为 10。

自定义变量和自定义函数

- 用冒号定义变量，可以是数值，也可以是代数式，如:

```
g : 9.8;
P1 : 3*p^2*(1-p) + p^3;
```

- 用:= 定义函数，如

```
f(x,y) := 1/2*g*x^2*y;
```

结果显示为

$$f(x, y) := \frac{1}{2} g x^2 y$$

- 调用如

```
f(2,10);
```

结果为 196.0。

自定义变量和自定义函数

- 用冒号定义变量，可以是数值，也可以是代数式，如:

```
g : 9.8;
P1 : 3*p^2*(1-p) + p^3;
```

- 用:= 定义函数，如

```
f(x,y) := 1/2*g*x^2*y;
```

结果显示为

$$\mathrm{f}\left(x, y\right) := \frac{1}{2}\, g\, x^2\, y$$

- 调用如

```
f(2,10);
```

结果为 196.0。

自定义变量和自定义函数

- 用冒号定义变量，可以是数值，也可以是代数式，如:

```
g : 9.8;
P1 : 3*p^2*(1-p) + p^3;
```

- 用:= 定义函数，如

```
f(x,y) := 1/2*g*x^2*y;
```

结果显示为

$$\mathrm{f}\left(x, y\right) := \frac{1}{2}\, g\, x^2\, y$$

- 调用如

```
f(2,10);
```

结果为 196.0。

代换

- 带有未知数的式子可以用逗号格式把未知数代换为数值或其他代数式，如

```
P1 : 3*p^2*(1-p) + p^3;
P1, p=0.5;
P1, p=1;
```

后面两式的结果分别为 0.5 和 1。

- 用 $\frac{1}{2} + a$ 替换 p，如

```
expand(P1), p=1/2 + a;
```

结果为

$$-2\,a^3 + \frac{3\,a}{2} + \frac{1}{2}$$

级数和

- 用 sum 函数表示级数和，如

```
sum(i^2, i, 1, 10);
```

表示 $\sum_{i=1}^{10} i^2$，结果为 385。

- 而

```
sum(i^2, i, 1, n);
```

显示为

$$\sum_{i=1}^{10} i^2$$

级数和

- 用 sum 函数表示级数和，如

```
sum(i^2, i, 1, 10);
```

表示 $\sum_{i=1}^{10} i^2$，结果为 385。

- 而

```
sum(i^2, i, 1, n);
```

显示为

$$\sum_{i=1}^{10} i^2$$

级数和化简

- 用 simpsum 进行级数化简，如

```
%, simpsum;
```

结果为

$$\frac{2\,n^3+3\,n^2+n}{6}$$

连乘积

- $\prod_{i=m}^{n} i^2$ 可以表示为 `product(i^2, i, m, n)`。
- 用 `simpproduct` 化简连乘积，如

```
product(i, i, 1, n), simpproduct;
```

结果为

$$n!$$

连乘积

- $\prod_{i=m}^{n} i^2$ 可以表示为 `product(i^2, i, m, n)`。
- 用 `simpproduct` 化简连乘积，如

```
product(i, i, 1, n), simpproduct;
```

结果为

$$n!$$

微分

- diff(f(x), x, n) 求 $\frac{d^n f(x)}{dx^n}$。
- 如

```
diff(sin(x)*exp(2*x), x, 1);
```

结果为

$$2\, e^{2\, x}\, \sin\left(x\right) + e^{2\, x}\, \cos\left(x\right)$$

- 又

```
diff(sin(x)*exp(2*x), x, 2);
```

结果为

$$3\, e^{2\, x}\, \sin\left(x\right) + 4\, e^{2\, x}\, \cos\left(x\right)$$

微分

- diff(f(x), x, n) 求 $\frac{d^n f(x)}{dx^n}$。
- 如

```
diff(sin(x)*exp(2*x), x, 1);
```

结果为

$$2\, e^{2\, x}\, \sin\left(x\right) + e^{2\, x}\, \cos\left(x\right)$$

- 又

```
diff(sin(x)*exp(2*x), x, 2);
```

结果为

$$3\, e^{2\, x}\, \sin\left(x\right) + 4\, e^{2\, x}\, \cos\left(x\right)$$

微分

- diff(f(x), x, n) 求 $\frac{d^n f(x)}{dx^n}$。
- 如

```
diff(sin(x)*exp(2*x), x, 1);
```

结果为

$$2\,e^{2\,x}\sin\left(x\right)+e^{2\,x}\cos\left(x\right)$$

- 又

```
diff(sin(x)*exp(2*x), x, 2);
```

结果为

$$3\,e^{2\,x}\sin\left(x\right)+4\,e^{2\,x}\cos\left(x\right)$$

不定积分

- `integrate(f(x), x)` 计算不定积分。
- 如

```
integrate(x / (1+x)^2, x);
```

结果为

$$\log(x+1)+\frac{1}{x+1}$$

不定积分

- `integrate(f(x), x)` 计算不定积分。
- 如

```
integrate(x / (1+x)^2, x);
```

结果为

$$\log(x+1) + \frac{1}{x+1}$$

定积分

- integrate(f(x), x, a, b) 计算定积分 $\int_a^b f(x)\,dx$。
- 如

```
integrate(x*sin(x)*exp(-x), x, 0, %pi);
```

结果为

$$\frac{(\pi+1)\ e^{-\pi}}{2}+\frac{1}{2}$$

- 又如

```
integrate(exp(-1/2*x^2), x, minf, inf);
```

结果为 $\sqrt{2}\sqrt{\pi}$。

定积分

- `integrate(f(x), x, a, b)` 计算定积分 $\int_a^b f(x)\,dx$。
- 如

```
integrate(x*sin(x)*exp(-x), x, 0, %pi);
```

结果为

$$\frac{(\pi+1)\ e^{-\pi}}{2}+\frac{1}{2}$$

- 又如

```
integrate(exp(-1/2*x^2), x, minf, inf);
```

结果为 $\sqrt{2}\sqrt{\pi}$。

定积分

- `integrate(f(x), x, a, b)` 计算定积分 $\int_a^b f(x)\,dx$。
- 如

```
integrate(x*sin(x)*exp(-x), x, 0, %pi);
```

结果为

$$\frac{(\pi+1)\ e^{-\pi}}{2}+\frac{1}{2}$$

- 又如

```
integrate(exp(-1/2*x^2), x, minf, inf);
```

结果为 $\sqrt{2}\sqrt{\pi}$。

本节目录

方程

一元方程

- 用 solve(eqn, x) 求解关于未知数 x 的一元方程，如

```
eq1 : a*x^2 + b*x + c = 0;
```

显示为

$$a\,x^2+b\,x+c=0$$

- 求解是符号运算:

```
eq1 : a*x^2 + b*x + c = 0;
```

结果为

$$[x=-\frac{\sqrt{b^2-4\,a\,c}+b}{2\,a},x=\frac{\sqrt{b^2-4\,a\,c}-b}{2\,a}]$$

一元方程

- 用 solve(eqn, x) 求解关于未知数 x 的一元方程，如

```
eq1 : a*x^2 + b*x + c = 0;
```

显示为

$$a\,x^2+b\,x+c=0$$

- 求解是符号运算:

```
eq1 : a*x^2 + b*x + c = 0;
```

结果为

$$[x=-\frac{\sqrt{b^2-4\,a\,c}+b}{2\,a},x=\frac{\sqrt{b^2-4\,a\,c}-b}{2\,a}]$$

- 以上一般解代入具体参数计算，如

```
%, a=1, b=2, c=1;
```

结果为

$$[x = -1, x = -1]$$

- 又如，以下的三次方程 (注意可以只给出方程左端)

```
solve(x^3 + 2*x^2 + 3*x + 4, x);
```

结果为

$$\left[x=-\frac{5\left(\frac{\sqrt{3}\,i}{2}-\frac{1}{2}\right)}{9\left(\frac{5\sqrt{2}}{3^{\frac{3}{2}}}-\frac{35}{27}\right)^{\frac{1}{3}}}+\left(\frac{5\sqrt{2}}{3^{\frac{3}{2}}}-\frac{35}{27}\right)^{\frac{1}{3}}\left(-\frac{\sqrt{3}\,i}{2}-\frac{1}{2}\right)-\frac{2}{3},\right.$$

$$x=\left(\frac{5\sqrt{2}}{3^{\frac{3}{2}}}-\frac{35}{27}\right)^{\frac{1}{3}}\left(\frac{\sqrt{3}\,i}{2}-\frac{1}{2}\right)-\frac{5\left(-\frac{\sqrt{3}\,i}{2}-\frac{1}{2}\right)}{9\left(\frac{5\sqrt{2}}{3^{\frac{3}{2}}}-\frac{35}{27}\right)^{\frac{1}{3}}}-\frac{2}{3},$$

$$\left.x=\left(\frac{5\sqrt{2}}{3^{\frac{3}{2}}}-\frac{35}{27}\right)^{\frac{1}{3}}-\frac{5}{9\left(\frac{5\sqrt{2}}{3^{\frac{3}{2}}}-\frac{35}{27}\right)^{\frac{1}{3}}}-\frac{2}{3}\right]$$

二元方程组

- 二元方程求解也使用 `solve`，其中方程和未知数都用方括号表示的列表表示。
- 如

```
eq1 : y + 2*c = 0;
eq2 : 2*x*y - c*y = 5;
```

对应于

$$y + 2\,c = 0$$

$$2\,x\,y - c\,y = 5$$

二元方程组

- 二元方程求解也使用 solve，其中方程和未知数都用方括号表示的列表表示。
- 如

```
eq1 : y + 2*c = 0;
eq2 : 2*x*y - c*y = 5;
```

对应于

$$y + 2\,c = 0$$

$$2\,x\,y - c\,y = 5$$

- 求解如

```
solve([eq1, eq2], [x, y]);
```

  结果为

$$[[x = \frac{2\,c^2 - 5}{4\,c}, y = -2\,c]]$$

- 结果表示为解的列表，每个解又是包含 x, y 两部分的列表。

- 求解如

```
solve([eq1, eq2], [x, y]);
```

结果为

$$[[x = \frac{2\,c^2 - 5}{4\,c}, y = -2\,c]]$$

- 结果表示为解的列表，每个解又是包含 x, y 两部分的列表。

- 一个二元二次方程组的例子:

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

结果为

$$
\begin{aligned}
&[[x = 0.51535959522949, y = 1.94039270687237],\\
&[x = 0.3167481152931\, i - 0.74342140598106,\\
&\quad y = -0.48506249405943\, i - 1.13846246879373 1],\\
&[x = -0.3167481152931\, i - 0.74342140598106,\\
&\quad y = 0.48506249405943\, i - 1.13846246879373 1],\\
&[x = 2.971483220309511, y = 0.3365322722219]]
\end{aligned}
$$

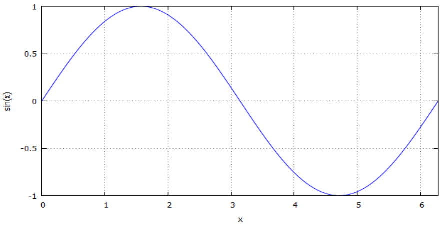
本节目录

图形

曲线图

- 函数的曲线图使用 plot2d 函数，用如

```
plot2d(sin(x), [x, 0, 2*%pi]);
```



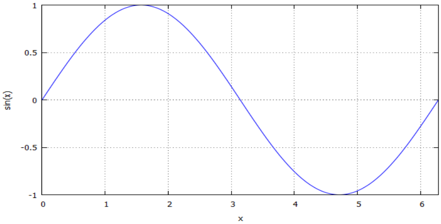
- 可以指定纵坐标范围，如

```
plot2d(sin(x), [x, 0, 2*%pi],
  [y, -1.2, 1.2]);
```

曲线图

- 函数的曲线图使用 plot2d 函数，用如

```
plot2d(sin(x), [x, 0, 2*%pi]);
```



- 可以指定纵坐标范围，如

```
plot2d(sin(x), [x, 0, 2*%pi],
  [y, -1.2, 1.2]);
```

图形文件

- 作图结果可以复制到 Windows 剪贴板，然后粘贴到 Word 中，或粘贴到画图程序中再保存为图形文件。
- 也可以直接保存为图形文件，如

```
plot2d(sin(x), [x, 0, 2*%pi],
  [gnuplot_term, png],
  [gnuplot_out_file, "test-save.png"]);
```

图形文件

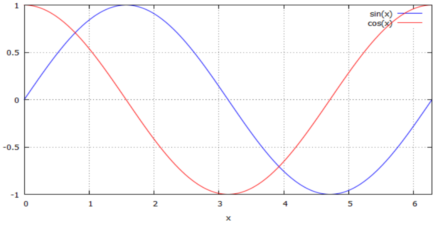
- 作图结果可以复制到 Windows 剪贴板，然后粘贴到 Word 中，或粘贴到画图程序中再保存为图形文件。
- 也可以直接保存为图形文件，如

```
plot2d(sin(x), [x, 0, 2*%pi],
  [gnuplot_term, png],
  [gnuplot_out_file, "test-save.png"]);
```

多条曲线

- 多条曲线只要指定多个函数，如

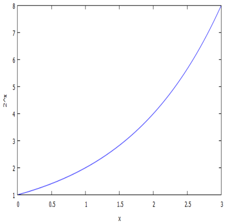
```
plot2d([sin(x), cos(x)], [x, 0, 2*%pi]);
```

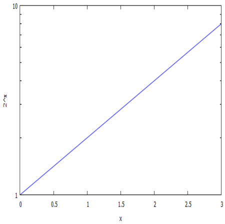


对数坐标轴

- plot2d 中加选项 [logy] 使得 y 轴为对数轴，加选项 [logx, logy] 使得 x, y 轴都为对数轴，如

```
plot2d(2^x, [x, 0, 3]);
plot2d(2^x, [x, 0, 3], [logy]);
```





参数方程作图

- 在 plot2d 中把要绘图的函数用 [parametrix, y(t), x(t), [t, tmin, tmax]] 表示，可以进行参数方程作图。
- 例如，如下参数方程表示的曲线

$$\begin{cases} x(t) = cos(t), \\ y(t) = sin(3t), \end{cases} \quad t \in [0, 2\pi]$$

- 可以作图如下:

```
plot2d([parametric, cos(t), sin(3*t),
   [t, 0, 2*%pi]],
   [x, -1.2, 1.2], [y, -1.2, 1.2],
   [nticks, 200]);
```

参数方程作图

- 在 plot2d 中把要绘图的函数用 [parametrix, y(t), x(t), [t, tmin, tmax]] 表示，可以进行参数方程作图。
- 例如，如下参数方程表示的曲线

$$\begin{cases} x(t) = cos(t), \\ y(t) = sin(3t), \end{cases} \quad t \in [0, 2\pi]$$

- 可以作图如下:

```
plot2d([parametric, cos(t), sin(3*t),
  [t, 0, 2*%pi]],
  [x, -1.2, 1.2], [y, -1.2, 1.2],
  [nticks, 200]);
```

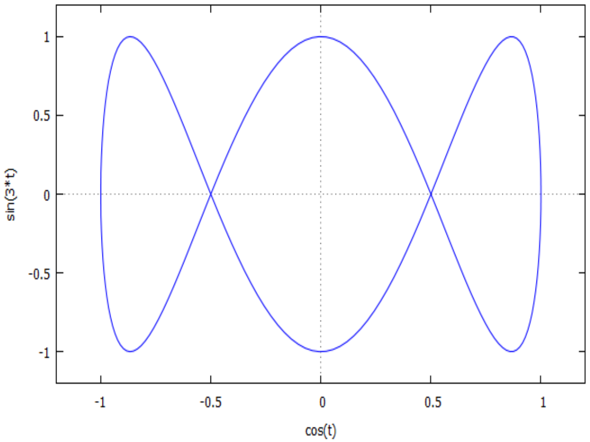
参数方程作图

- 在 plot2d 中把要绘图的函数用 [parametrix, y(t), x(t), [t, tmin, tmax]] 表示，可以进行参数方程作图。
- 例如，如下参数方程表示的曲线

$$\begin{cases} x(t) = cos(t), \\ y(t) = sin(3t), \end{cases} \quad t \in [0, 2\pi]$$

- 可以作图如下:

```
plot2d([parametric, cos(t), sin(3*t),
  [t, 0, 2*%pi]],
  [x, -1.2, 1.2], [y, -1.2, 1.2],
  [nticks, 200]);
```


1
0.5
0
-0.5
-1
sin(3*t)
-1
-0.5
0
0.5
1
cos(t)

- 极坐标参数曲线可以化为直角坐标后参数作图，如

$$z(t)=\left[e^{\cos t}-2\cos 4t-\sin^5\frac{t}{12}\right]e^{it},\ \ t\in[-8\pi,8\pi],$$

- 程序如

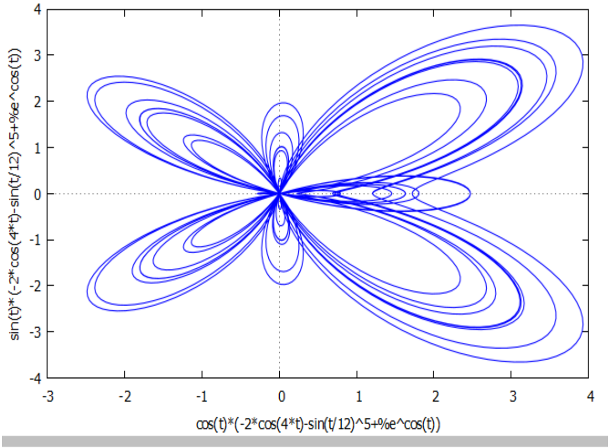
```
r : (exp(cos(t)) - 2*cos(4*t) - sin(t/12)^5);
plot2d([parametric, r*sin(t), r*cos(t),
  [t, -8*%pi, 8*%pi]],
  [nticks, 2000]);
```

- 极坐标参数曲线可以化为直角坐标后参数作图，如

$$z(t) = \left[e^{\cos t} - 2\cos 4t - \sin^5 \frac{t}{12} \right] e^{it}, \ \ t \in [-8\pi, 8\pi],$$

- 程序如

```
r : (exp(cos(t)) - 2*cos(4*t) - sin(t/12)^5);
plot2d([parametric, r*sin(t), r*cos(t),
   [t, -8*%pi, 8*%pi]],
   [nticks, 2000]);
```


sin(t)*(-2*cos(4*t)-sin(t/12)^5+%e^cos(t))
cos(t)*(-2*cos(4*t)-sin(t/12)^5+%e^cos(t))
4
3
2
1
0
-1
-2
-3
-4
-3
-2
-1
0
1
2
3
4

三维曲面图

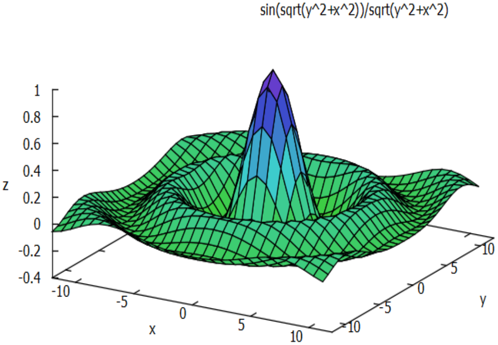
- 三维图用 plot3d。
- 如

```
plot3d(sin(sqrt(x^2+y^2))/sqrt(x^2+y^2),
    [x, -12, 12], [y, -12, 12])$
```

三维曲面图

- 三维图用 plot3d。
- 如

```
plot3d(sin(sqrt(x^2+y^2))/sqrt(x^2+y^2),
     [x, -12, 12], [y, -12, 12])$
```


sin(sqrt(y^2+x^2))/sqrt(y^2+x^2)
z
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-10
-5
0
5
10
x
-10
-5
0
5
10
y

三维参数曲面图

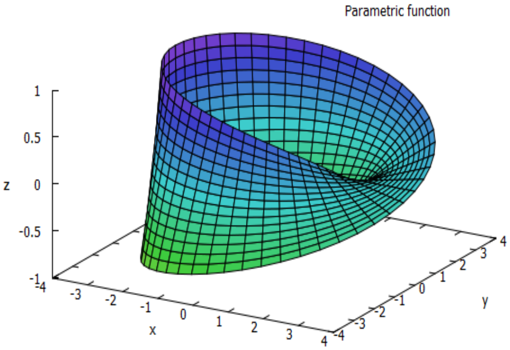
- 三维图用 plot3d 和参数方程，有两个参数。
- 如

```
plot3d([cos(x)*(3 + y*cos(x/2)),
        sin(x)*(3 + y*cos(x/2)),
        y*sin(x/2)],
        [x, -%pi, %pi], [y, -1, 1],
        ['grid, 50, 15])$
```

三维参数曲面图

- 三维图用 plot3d 和参数方程，有两个参数。
- 如

```
plot3d([cos(x)*(3 + y*cos(x/2)),
        sin(x)*(3 + y*cos(x/2)),
        y*sin(x/2)],
        [x, -%pi, %pi], [y, -1, 1],
        ['grid, 50, 15])$
```


Parametric function
z
1
0.5
0
-0.5
-1
x
-4
-3
-2
-1
0
1
2
3
4
y
-4
-3
-2
-1
0
1
2
3
4