# 第二章　微积分运算

微积分是数学学习的重点和难点之一，而微积分运算是 Maple 最为拿手的计算之一，任何解析函数，Maple 都可以求出它的导数来，任何理论上可以计算的积分，Maple 都可以毫不费力的将它计算出来. 随着作为数学符号计算平台的 Maple 的不断开发和研究，越来越多的应用程序也在不断地创设.

## 1 函数的极限和连续

### 1.1 函数和表达式的极限

在 Maple 中，利用函数 **limit** 计算函数和表达式的极限. 如果要写出数学表达式，则用惰性函数 **Limit**. 若 a 可为任意实数或无穷大时，求 $\lim\limits_{x\to a} f(x)$ 命令格式为: limit(f,x=a); 求 $\lim\limits_{x\to a^+} f(x)$ 时的命令格式为 limit(f, x=a, right); 求 $\lim\limits_{x\to a^-} f(x)$ 时的命令格式为 limit(f, x=a, left); 请看下述例子：

```
> Limit((1+1/x)^x,x=infinity)=limit((1+1/x)^x,x=infinity);
```

$$\lim_{x\to\infty}\left(1+\frac{1}{x}\right)^x = \mathrm{e}$$

```
> Limit((x^n-1)/(x-1),x=1)=limit((x^n-1)/(x-1),x=1);
```

$$\lim_{x\to 1}\frac{x^n-1}{x-1} = n$$

```
> Limit(x^x,x=0,right)=limit(x^x,x=0,right);
```

$$\lim_{x\to 0+} x^x = 1$$

```
> Limit(abs(x)/x,x=0,left)=limit(abs(x)/x,x=0,left);
```

$$\lim_{x\to 0-}\frac{|x|}{x} = -1$$

```
> Limit(abs(x)/x,x=0,right)=limit(abs(x)/x,x=0,right);
```

$$\lim_{x\to 0+}\frac{|x|}{x} = 1$$

```
> limit(abs(x)/x,x=0);
```

$$undefined$$

对于多重极限计算, 也用limit. 命令格式为: limit(f, points, dir); 其中, points是由一系列方程定义的极限点, dir(可选项)代表方向: left(左)、right(右)等. 例如:

```
> limit(a*x*y-b/(x*y),{x=1,y=1});
```

$$a-b$$

```
> limit(x^2*(1+x)-y^2*((1-y))/(x^2+y^2),{x=0,y=0});
```

$$undefined$$

由于多重极限的复杂性，很多情况下 limit 无法找到答案，此时，不应轻易得出极限不存在的结论，而是应该利用数学基础判定极限的存在性，然后再寻找别的可行的方法计算极限(如化 n 重根限为 n 次极限等)。如下例就是化二重极限为二次极限而得正确结果：

```
> limit((sin(x+y)/(sin(x)*sin(y)),{x=Pi/4,y=Pi/4}));
```

$$\operatorname{limit}\left(\frac{\sin(x+y)}{\sin(x)\sin(y)}, \left\{x=\frac{1}{4}\pi, y=\frac{1}{4}\pi\right\}\right)$$

```
> limit(limit(sin(x+y)/(sin(x)*sin(y)),x=Pi/4),y=Pi/4);
```

$$2$$

## 1.2 函数的连续性

### 1.2.1 连续

在 Maple 中可以用函数 **iscont** 来判断一个函数或者表达式在区间上的连续性. 命令格式为:

iscont(expr, x=a..b, 'colsed'/'opened'**);**

其中, closed 表示闭区间, 而 opened 表示开区间(此为系统默认状态).

如果表达式在区间上连续, iscont 返回 true, 否则返回 false, 当 iscont 无法确定连续性时返回 FAIL. 另外, iscont 函数假定表达式中的所有符号都是实数型. 颇为有趣的是, 当给定区间[*a,b*] (*a*>*b*)时, iscont 会自动按[*b,a*]处理.

```
> iscont(1/x,x=1..2);
```

$$true$$

```
> iscont(1/x,x=-1..1,closed);
```

$$false$$

```
> iscont(1/(x+a),x=0..1);
```

$$FAIL$$

```
> iscont(ln(x),x=10..1);
```

$$true$$

### 1.2.2 间断

函数 **discont** 可以寻找函数或表达式在实数域的间断点, 当间断点周期或成对出现

时，Maple 会利用一些辅助变量予以表达，比如，**_Zn**~(任意整数)、**_NZn**~(任意自然数)和 **Bn**~(一个二进制数, 0 或者 1)，其中 n 是序号．判定 f(x)间断点的命令为:

discont(f, x);

```
> discont(ln(x^2-4),x);
```

$$\{-2, 2\}$$

```
> discont(arctan(1/2*tan(2*x))/(x^2-1),x);
```

$$\{-1, 1, \frac{1}{2}\pi\_Z1\sim+\frac{1}{4}\pi\}$$

```
> discont(round(3*x-1/2),x);
```

$$\{\frac{1}{3}+\frac{1}{3}\_Z1\}$$

函数round为“四舍五入”函数，上例并非一目了然，对其进一步理解可借助于函数plot或下面给出的fdiscont例子。

另一个寻找间断点的函数**fdiscont**是用数值法寻找在实数域上的间断点．命令格式为:

fdiscont(f, domain, res, ivar, eqns);

其中，f表示表达式或者，domain表示要求的区域，res表示要求的分辨率，ivar表示独立变量名称, eqns表示可选方程.

```
> fdiscont(GAMMA(x/2), x=-10..0, 0.0001);
```

[-10.0000086158831731 .. -9.99994148230377888 , -8.00006570776243642 .. -7.99994267738026022 ,
-6.00006150975869534 .. -5.99992641140166860 , -4.00004327130417892 .. -3.99994355004940650 ,
-2.00005917347370676 .. -1.99993152974711540 , -.0000687478627751661410 .. .0000124812251913941740 ]

```
> fdiscont(arctan(1/2*tan(2*x))/(x^2-1),x=-Pi..Pi);
```

[-1.00024845147850305 .. -.999333284093140484 , -.785872546183299604 .. -.785043522184549426 ,
.785028868651463486 .. .785693822920970008 , .999744688307389716 .. 1.00074853662123409 ,
2.35588567826642148 .. 2.35643817620602248 ]

```
> fdiscont(abs(x/10000),x=-1..1,0.000001);
```

[ ]

```
> fdiscont(tan(10*x),x=0..Pi,0.01,newton=true);
```

[.157079632679489656 , .471238898038468967 , .785398163397448280 , 1.09955742875642758 ,
1.41371669411540690 , 1.72787595947438644 , 2.04203522483336552 , 2.35619449019234484 ,
2.67035375555132414 , 2.98451302091030346 ]

```
> fdiscont(round(3*x-1/2),x=-1..1);
```

[-1.00003155195758886 .. -.999634596231187556 , -.667056666250248842 .. -.666327819216380068 ,
-.333731406929595187 .. -.333002495225188489 , -.000262904384890887231 .. .000298288004691702826 ,
.332976914927844203 .. .333778498831551751 , .666340020328179072 .. .667141797110034518 ,
.999646728223793524 .. 1.00008356747731275 ]

# 2 导数和微分

## 2.1 符号表达式求导

利用 Maple 中的求导函数 **diff** 可以计算任何一个表达式的导数或偏导数, 其惰性形式 **Diff** 可以给出求导表达式, **$**表示多重导数. 求 expr 关于变量 x1, x2, …, xn 的(偏)导数的命令格式为:

diff(expr, x1, x2, …, xn);

diff(expr, [x1, x2, …, xn]);

其中, expr 为函数或表达式, x1, x2, …, xn 为变量名称.

有趣的是, 当 n 大于 1 时, diff 是以递归方式调用的:

diff(f(x), x, y)=diff(diff(f(x), x), y)

```
> Diff(ln(ln(ln(x))),x)=diff(ln(ln(ln(x))),x);
```

$$\frac{\partial}{\partial x}\ln(\ln(\ln(x))) = \frac{1}{x\ln(x)\ln(\ln(x))}$$

```
> Diff(exp(x^2),x$3)=diff(exp(x^2),x$3);
```

$$\frac{\partial^3}{\partial x^3}\mathrm{e}^{(x^2)} = 12\,x\,\mathrm{e}^{(x^2)} + 8\,x^3\,\mathrm{e}^{(x^2)}$$

```
> diff(x^2*y+x*y^2,x,y);
```

$$2\,x + 2\,y$$

```
> f(x,y):=piecewise(x^2+y^2<>0,x*y/(x^2+y^2));
```

$$\mathrm{f}(x,y) := \begin{cases} \dfrac{x\,y}{x^2+y^2} & x^2+y^2 \neq 0 \\ 0 & otherwise \end{cases}$$

```
> diff(f(x,y),x);
```

$$\begin{cases} \dfrac{y}{x^2+y^2} - \dfrac{2\,x^2\,y}{(x^2+y^2)^2} & x^2+y^2 \neq 0 \\ 0 & otherwise \end{cases}$$

```
> diff(f(x,y),x,y);
```

$$\begin{cases} \dfrac{1}{x^2+y^2} - \dfrac{2\,y^2}{(x^2+y^2)^2} - \dfrac{2\,x^2}{(x^2+y^2)^2} + \dfrac{8\,x^2\,y^2}{(x^2+y^2)^3} & x^2+y^2 \neq 0 \\ 0 & otherwise \end{cases}$$

```
> normal(%);
```

$$\begin{cases} -\dfrac{x^4 - 6x^2y^2 + y^4}{(x^2+y^2)^3} & x^2 + y^2 \neq 0 \\ 0 & otherwise \end{cases}$$

函数 diff 求得的结果总是一个表达式, 如果要得到一个函数形式的结果, 也就是求导函数, 可以用 D 算子. D 算子作用于一个函数上, 得到的结果也是一个函数. 求 f 的导数的命令格式为: D(f);

值得注意的是, f 必须是一个可以处理为函数的代数表达式, 它可以包含常数、已知函数名称、未知函数名称、箭头操作符、算术和函数运算符.

复合函数表示为 f@g, 而不是 f(g), 因此 D(sin(y))是错误的, 正确的应该是 D(sin@y).

D 运算符也可以求高阶导数, 但此时不用$, 而用两个@@.

D 运算符并不局限于单变量函数, 一个带指标的 **D** 运算符 D[i](f)可以用来求偏导函数, D[i](f)表示函数 f 对第 i 个变量的导函数, 而高阶导数 D[i,j](f)等价于 **D[i](D[j](f))**.

```
> g:=x->x^n*exp(sin(x));
```

$$g := x \to x^n\, e^{\sin(x)}$$

```
> D(g);
```

$$x \to \frac{x^n\, n\, e^{\sin(x)}}{x} + x^n \cos(x)\, e^{\sin(x)}$$

```
> Diff(g,x)(Pi/6)=D(g)(Pi/6);
```

$$\left(\frac{\partial}{\partial x} g\right)\left(\frac{1}{6}\pi\right) = 6\,\frac{\left(\frac{1}{6}\pi\right)^n n\, e^{(1/2)}}{\pi} + \frac{1}{2}\left(\frac{1}{6}\pi\right)^n \sqrt{3}\, e^{(1/2)}$$

```
> D(D(sin));
```

$$-\sin$$

```
> (D@@2)(sin);
```

$$-\sin$$

```
> f:=(x,y,z)->(x/y)^(1/z);
```

$$f := (x, y, z) \to \left(\frac{x}{y}\right)^{\left(\frac{1}{z}\right)}$$

```
> Diff(f,y)(1,1,1)=D[2](f)(1,1,1);
```

$$\left(\frac{\partial}{\partial y} f\right)(1, 1, 1) = -1$$

D 运算符和函数 diff 的差别:

1） D 运算符计算运算符的导数，而 diff 计算表达式的导数;

2） D 的参数和结果是函数型运算符，而 diff 的参数和结果是表达式;

3） 将含有导数的表达式转换为 D 运算符表达式的函数为: convert(expr,D);

```
> f:=diff(y(x),x$2):
```

$$f := \frac{\partial^2}{\partial x^2}\,\mathrm{y}(x)$$

```
> convert(f,D);
```

$$(\mathrm{D}^{(2)})(y)(x)$$

4） 将 D(f)(x)表达式转换为 diff(f(x),x)形式的命令: convert(expr,diff,x);

```
> f:=D(y)(x)-a*D(z)(x);
```

$$f := \mathrm{D}(y)(x) - a\,\mathrm{D}(z)(x)$$

```
> convert(f,diff,x);
```

$$\left(\frac{\partial}{\partial x}\,\mathrm{y}(x)\right) - a\left(\frac{\partial}{\partial x}\,\mathrm{z}(x)\right)$$

D 运算符可以计算定义为程序的偏导数，此即 Maple 自动求导功能(详细内容参看第 6 章).

下面我们讨论在积分学当中的一个微妙的漏洞，在大多数计算机代数系统中都会出现这个问题，甚至于在许多教科书和积分表中这种情况也是长期存在。

```
> f:=1/(2+sin(x));
```

$$f := \frac{1}{2+\sin(x)}$$

```
> F:=int(f,x);
```

$$F := \frac{2}{3}\sqrt{3}\,\arctan\left(\frac{1}{3}\left(2\tan\left(\frac{1}{2}x\right)+1\right)\sqrt{3}\right)$$

```
> limit(F,x=Pi,right), limit(F,x=Pi,left);
```

$$-\frac{1}{3}\pi\sqrt{3},\ \frac{1}{3}\pi\sqrt{3}$$

关于函数 f(x)的积分仅在一些区间上是正确的，因为 F 是不连续的，虽然由微积分的基本定理可知当 f 连续时 F 应该是连续的。进一步的讨论 F 的不连续点：

```
> discont(F,x);
```

$$\{2\pi\_Z2 + \pi\}$$

因此，F 在 $x = n\pi$ 处有跳跃间断点。

在对多元函数 f(x,y)求混合偏导数时，Maple 总自以为是 $\frac{\partial f}{\partial x\partial y} = \frac{\partial f}{\partial y\partial x}$，这一点在

f(x,y)连续的情况下当然正确，但不连续时不正确。一个典型的例子是：

```
> f(x,y):=piecewise(x^2+y^2<>0,x*y*(x^2-y^2)/(x^2+y^2));
```

$$f(x, y) := \begin{cases} \dfrac{x\,y\,(x^2-y^2)}{x^2+y^2} & x^2+y^2 \neq 0 \\ 0 & \mathit{otherwise} \end{cases}$$

**********************************

```
> normal(diff(f(x,y),x,y));
```

$$\begin{cases} \dfrac{x^6+9\,x^4\,y^2-9\,x^2\,y^4-y^6}{(x^2+y^2)^3} & x^2+y^2 \neq 0 \\ 0 & \mathit{otherwise} \end{cases}$$

```
> normal(diff(f(x,y),y,x));
```

$$\begin{cases} \dfrac{x^6+9\,x^4\,y^2-9\,x^2\,y^4-y^6}{(x^2+y^2)^3} & x^2+y^2 \neq 0 \\ 0 & \mathit{otherwise} \end{cases}$$

因此，使用 Maple 进行科学计算时，一定要先运用数学理论和方法对问题进行简单推导，然后再利用 Maple 辅助计算，切不可把所有的事情都交给 Maple，如果那样的话会出现错误甚至是低级的错误。

## 2.2 隐函数求导

隐函数或由方程(组)确定的函数的求导，使用命令 **implicitdiff**. 假定 f, f1,⋯,fm 为代数表达式或者方程组, y, y1,⋯,yn 为变量名称或者独立变量的函数, 且 m 个方程 f1,⋯,fm 隐式地定义了 n 个函数 y1,⋯,yn, 而 u, u1,⋯,ur 为独立变量的名称, x, x1,⋯,xk 为导数变量的名称. 则：

(1) 求由 f 确定的 y 对 x 的导数:

implicitdiff(f,y,x);

(2) 求由 f 确定的 y 对 x1,⋯,xk 的偏导数:

implicitdiff(f,y,x1,⋯,xk);

(3) 计算 u 对 x 的导数, 其中 u 必须是给定的 y 函数中的某一个

implicitdiff({f1,⋯,fm},{y1,⋯,yn},u,x);

(4) 计算 u 对 x1,⋯,xk 的偏导数

implicitdiff({f1,⋯,fm},{y1,⋯,yn},u,{x1,⋯xk});

(5) 计算 u 的高阶导数

implicitdiff({f1,⋯,fm},{y1,⋯,yn},{u1,⋯,ur}, x1,⋯,xk);

implicitdiff(f,y,x)命令的主要功能是求隐函数方程 f 确定的 y 对 x 的导数, 因此, 输入的 f 必须是 x 和 y 或者代数表达式的方程(其中代数表达式为 0). 第二个参数 y 指定了非独立变量、独立变量或常数, 如果 y 是名称, 就意味着非独立变量, 而所有其他出现在

输入的 f 和求导变量 x 中名称以及不看作是常数类型的变量，统统视作独立变量处理. 如果方程 f1,…,fm 是超定的, implicitdiff 返回 FAIL. 例如:

```
> f:=exp(y)-x*y^2=x;
```

$$f := e^y - x\,y^2 = x$$

```
> implicitdiff(f,y,x);
```

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

```
> g:=x^2+y^3=1;
```

$$g := x^2 + y^3 = 1$$

```
> implicitdiff(g,z,x);
```

$$FAIL$$

如果是对多元函数求多个偏导数，结果将用偏微分形式给出. 可以给定最后一个可选参数来确定结果的表达形式，默认情况下或者给定 **notation=D**，这时结果中的微分用 D 运算符表示，否则可以给定 **notation=Diff**，这样给出的结果中的微分运算符和使用 Diff 时相同，即用 ∂ 来表示. 试作以下实验:

$$\begin{cases} x = \cos u \cos v \\ y = \cos u \sin v \\ z = \sin u \end{cases} \quad 求 \frac{\partial^2 z}{\partial x^2}$$

```
> f:=x=cos(u)*cos(v);
```

$$f := x = \cos(u)\cos(v)$$

```
> g:=y=cos(u)*sin(v);
```

$$g := y = \cos(u)\sin(v)$$

```
> h:=z=sin(u);
```

$$h := z = \sin(u)$$

```
> implicitdiff ({f,g,h}, {z(x,y), u(x,y), v(x,y)}, {z}, x, x, notation=Diff);
```

$$\left\{ \left(\frac{\partial^2}{\partial x^2} z\right)_y = -\frac{\sin(v)^2 \sin(u)^2 + 1 - \sin(v)^2}{\sin(u)^3} \right\}$$

## 2.3 函数的极值

### 2.3.1 函数的极值

极值包含两种情形：极大值和极小值。在 **Maple** 中，有两个求函数极值的命令: minimize, maximize，命令格式如下:

**minimize (expr, vars, range);**

**maximize (expr, vars, range);**

```
> expr1:=x^3-6*x+3:
  minimize(expr1,x=-3..3);
```

$$-6$$

```
  maximize(expr1,x=-3..3);
```

$$12$$

```
> minimize(tanh(x),x,`infinite`);
```

$$-1$$

```
  maximize(tanh(x),x,`infinite`);
```

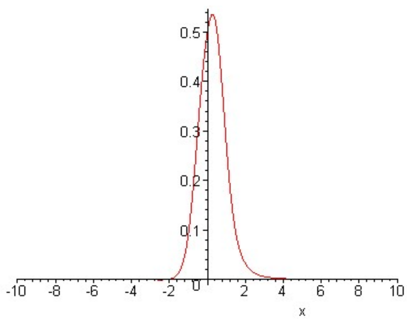
$$1$$

虽然，minimize 和 maximize 这两个命令很好用，但对于一些特殊的函数而言，这两个指令不但有可能无法求得极值，还有可能给我们错误的解. 因此，在 Maple 下求极值最好的方法是先作图(鼠标右键点击函数解析式选择 plots 命令即可)，由图上找出极值的大概位置，然后再由 Maple 提供的各种指令来求解. 下面一个例子是关于函数极大极小值的求解问题，此处，图形提供了做题的部分思路，尤其是求驻点时：

```
> f:=(x+2)/(3+(x^2+1)^3);
```

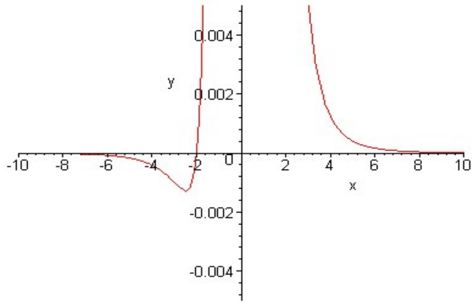
$$f := \frac{x+2}{3+(x^2+1)^3}$$

```
> plot(f,x=-10..10);
```




从上图可以看出，f(x)函数从区域-2 到 4 之间有极值出现，且极大值小于 1. 为了更清楚的了解函数图像性质，我们在 plot 命令中加入因变量 y 的变化范围。由此可看出 f(x)与 x 轴有交点，且在-2 附近有一极小值：

```
> plot(f,x=-10..10,y=-0.005..0.005);
```




进一步应用导数性质求解该问题：

```
> d:=diff(f,x);
```

$$d := \frac{1}{3+(x^2+1)^3} - \frac{6\,(x+2)\,(x^2+1)^2\,x}{(3+(x^2+1)^3)^2}$$

```
> simplify(d);
```

$$-\frac{-4+5\,x^6+9\,x^4+3\,x^2+12\,x^5+24\,x^3+12\,x}{(4+x^6+3\,x^4+3\,x^2)^2}$$

由图形可见，极值“可能”出现在 x=-2 和 x=0 附近，可用下述语句求出确切极点，然后使用 eval 命令求出相应的极值：

```
> xmin:=fsolve(d=0,{x=-2});
```

$$xmin := \{x = -2.485165927\}$$

```
> xmax:=fsolve(d=0,{x=0});
```

$$xmax := \{x = .2700964050\}$$

```
> Digits:=4:
> Xmin:=eval(f,xmin);
```

$$Xmin := -.001302$$

```
> Xmax:=eval(f,xmax);
```

$$Xmax := .5360$$

### 2.3.2 条件极值

有时候，我们还会遇到在条件 $q(x,y)=0$ 下计算函数 $f(x,y)$ 的极大值和极小值，这就是条件极值．在求解时需要先构造一个函数 $g(x,y)=f(x,y)+\mu q(x,y)$ ( $\mu$ 称为拉格朗日乘子)，然后将 $g(x,y)$分别对 $x$ 和 $y$ 求导，得到联立方程组，求解方程组即可得到函数 $f(x,y)$ 的极大值和极小值．

下面求解 $f(x,y)=x^2+y^2$ 在条件 $q(x,y)=x^2+y^2+2x-2y+1$ 下的极大值和极小值.

```
> f:=x^2+y^2:
> q:=x^2+y^2+2*x-2*y+1:
> g:=f+mu*q;
```

$$g := x^2+y^2+\mu\,(x^2+y^2+2\,x-2\,y+1)$$

```
> exp1:=diff(g,x); exp2:=diff(g,y);
```

$$\mathrm{exp1} := 2\,x+\mu\,(2\,x+2)$$

$$\mathrm{exp2} := 2\,y+\mu\,(2\,y-2)$$

```
> exp3:=solve({q=0,exp1,exp2},{x,y,mu});
```

$$\mathrm{exp3} := \{y=\mathrm{RootOf}(2\,\_Z^2-4\,\_Z+1),\, x=-\mathrm{RootOf}(2\,\_Z^2-4\,\_Z+1),\, \mu=1-2\,\mathrm{RootOf}(2\,\_Z^2-4\,\_Z+1)\}$$

```
> allvalues(exp3);
```

$$\{y=1+\frac{1}{2}\sqrt{2},\,\mu=-1-\sqrt{2},\,x=-1-\frac{1}{2}\sqrt{2}\},\,\{\mu=-1+\sqrt{2},\,y=1-\frac{1}{2}\sqrt{2},\,x=-1+\frac{1}{2}\sqrt{2}\}$$

```
> subs({x=-1-1/2*2^(1/2),y=1+1/2*2^(1/2)},f):
> fmax:=evalf(%);
```

$$fmax := 5.828427124$$

```
> subs({x=-1+1/2*2^(1/2),y=1-1/2*2^(1/2)},f):
> fmin:=evalf(%);
```

$$fmin := .1715728755$$

# 3 积分运算

## 3.1 不定积分

Maple 有许多内建的积分算法，一般地，用 **int** 求不定积分. 命令格式为:

int(expr,x);

```
> Int(x^2*arctan(x)/(1+x^2),x)=int(x^2*arctan(x)/(1+x^2),x);
```

$$\int\frac{x^2\arctan(x)}{1+x^2}\,dx=\frac{1}{8}\ln(1+I\,x)^2+\frac{1}{2}I\left(-x+\frac{1}{2}I\ln(1-I\,x)\right)\ln(1+I\,x)+\frac{1}{8}\ln(1-I\,x)^2+\frac{1}{2}I\,x\ln(1-I\,x)-\frac{1}{2}\ln(1+x^2)$$

```
> int(x/(x^3-1),x);
```

$$\frac{1}{3}\ln(x-1)-\frac{1}{6}\ln(1+x+x^2)+\frac{1}{3}\sqrt{3}\arctan\left(\frac{1}{3}(2\,x+1)\sqrt{3}\right)$$

```
> int(exp(-x^2),x);
```

$$\frac{1}{2}\sqrt{\pi}\ \mathrm{erf}(x)$$

```
> Int(ln(x+sqrt(1+x^2)),x);
```

$$\int \ln(x+\sqrt{1+x^2})\,dx$$

```
> value(%)+c;
```

$$\ln(x+\sqrt{1+x^2})\,x-\sqrt{1+x^2}+c$$

```
> int(exp(-x^2)*ln(x),x);
```

$$\int \mathrm{e}^{(-x^2)}\ln(x)\,dx$$

可以看出, Maple 求不定积分的结果中没有积分常数, 这一点需要注意. 但是, 这有一定好处的, 尤其当对结果作进一步处理时, 由于 Maple 符号计算的特点, 引入积分常数相当于引入一个变量, 对于计算极为不便. Maple 中不定积分的计算过程为:

(i) 首先, Maple 用传统方法处理一些特殊的形式, 如多项式、有理式、形如 $\left(\sqrt{a+bx+cx^2}\right)^n$ 和 $Q(x)\left(\sqrt{a+bx+cx^2}\right)^n$ 的根式, 以及形如 $\mathrm{Pn}(x)\cdot\ln x$ 或 $\frac{P_1(x)}{Q_1(x)}\bullet\ln\frac{P_2(x)}{Q_2(x)}$ 的表达式;

(ii) 如果传统方法难以奏效, Maple 将应用 **Risch-Norman** 算法, 以避免在包含三角函数和双曲函数的积分中引入复指数和对数;

(iii) 如果仍然无法得到答案, Maple 将采用 **Risch** 算法, 这将无法避免地在结果表达式中引入有关积分变量的 RootOf 的表达式;

(iv) 如果最终还是没有找到解析表达式, Maple 会把积分式作为结果返回.

## 3.2 定积分

定积分与不定积分的计算几乎一样, 只是多了一个表示积分区域的参数. 在[a,b]上求 f 的定积分的命令格式为:

int(f, x=a..b);

```
> Int(1/(1+x^2),x=-1..1)=int(1/(1+x^2),x=-1..1);
```

$$\int_{-1}^{1}\frac{1}{1+x^2}\,dx=\frac{1}{2}\pi$$

抛物线 $y^2=2px$ 与 $x^2=2py$ 所围图形的面积计算过程如下:

```
> assume(p>0);
```

```
> int(sqrt(2*p*x)-x^2/(2*p),x=0..2*p);
```

$$\frac{4}{3}p^2$$

Maple 中的定积分是怎样完成的呢？最简单的想法是：按照 Newton-Leibnize 定理，先求出被积函数的任一个原函数，再求其在积分限上的增量即可得定积分．试看下例(其中的函数 **rhs** 用来获取等式右边部分—**right hand side**，相应地，左边部分为 **lhs**)：

```
> Int(1/x^2,x)=int(1/x^2,x);
```

$$\int \frac{1}{x^2}\,dx = -\frac{1}{x}$$

```
> subs(x=1,rhs(%))-subs(x=-1,rhs(%));
```

$$-2$$

显然，这是错误的，在上述积分中含有一个暇点 $x$=0，积分是发散的，对此，Maple 是知道的：

```
> Int(1/x^2,x=-1..1)=int(1/x^2,x=-1..1);
```

$$\int_{-1}^{1} \frac{1}{x^2}\,dx = \infty$$

在大多数情况下，Maple 通过查表和形式匹配，或者利用特殊函数的导数来求定积分．

```
> Int( exp(-x^2)*ln(x)^2, x=0..infinity )=int( exp(-x^2)*ln(x)^2, x=0..infinity );
```

$$\int_{0}^{\infty} e^{(-x^2)} \ln(x)^2\,dx = \frac{1}{16}\pi^{(5/2)} + \frac{1}{8}\sqrt{\pi}\,\gamma^2 + \frac{1}{2}\sqrt{\pi}\,\gamma\ln(2) + \frac{1}{2}\sqrt{\pi}\ln(2)^2$$

```
>evalf(rhs(%));
```

$$1.947522181$$

```
>Int(sin(x)/x,x=-1..1)=int(sin(x)/x,x=-1..1);
```

$$\int_{-1}^{1} \frac{\sin(x)}{x}\,dx = 2\,\mathrm{Si}(1)$$

```
> evalf(rhs(%));
```

$$1.892166141$$

```
> Int(sin(x^2),x=-infinity..infinity)=int(sin(x^2),x=-infinity..infinity);
```

$$\int_{-\infty}^{\infty} \sin(x^2)\,dx = \frac{1}{2}\sqrt{2}\,\sqrt{\pi}$$

```
> evalf(rhs(%));
```

$$1.253314137$$

在返回一个未求值的定积分的情况下，可以对积分式调用 evalf 来获得数值积分.命令格式为:

evalf(int(f, x=a..b));

evalf(Int(f, x=a..b, Digits, flag);

其中, f 为被积函数, x 积分变量, a..b 积分区间, Digits 表示需要精度的位数, flag 指定要使用的数值方法的名称. 值得注意的是, 上述命令格式第一式 int 中的 i 可以大写也可以小写(输出结果略有形式上的不同), 第二式的 I 必须大写.

Maple 中默认的数值积分方法是 **Clenshaw-Curitis** 4 阶方法; 当收敛很慢(由于存在奇点)时, 系统将试着用广义的级数展开和变量代换取消积分的奇异性; 如果存在不可去奇点, 则改而采用自适应双指数方法. 在数值精度不高的情况下(比如 Digits≤15), 采用自适应的牛顿—柯特斯方法就够了. 通过指定 **evalf/int** 语句的第 4 个参数, 可以选择积分方法. 可供选择的有 3 种方法:

**_Ccquad** -- Clenshaw-Curitis 4 阶方法

**_Dexp** -- 自适应双指数方法

**_Ncrule** -- 牛顿—柯特斯方法

```
> evalf(Int(1/sqrt(x),x=0..2,15,_Dexp));
```

$$2.82842712474619$$

```
> evalf(Int(sin(x)/x,x=0..1,20,_NCrule));
```

$$.94608307036718301494$$

```
> evalf(Int(sin(x)*ln(x+1),x=0..1));
```

$$.2265353653$$

```
> evalf(Int(sin(x)*ln(x+1),x=0..1));
```

$$.2265353653$$

前面述及函数 value 的主要功能是对惰性函数求值，但它与 evalf 是有区别的，试通过下例体会 value 与 evalf 功能的不同之处：

```
> P:=Int(x^2*sin(sin(x)),x=0..Pi);
```

$$P := \int_0^{\pi} x^2 \sin(\sin(x))\, dx$$

```
> value(P);
```

$$\int_0^{\pi} x^2 \sin(\sin(x))\, dx$$

```
> evalf(P);
```

$$5.289745102$$

### 3.3 其它积分方法

Maple有丰富的内建积分算法，除了上述**int**命令外，另外一些算法也非常有用．本节简述其中几种较为有用的算法.

#### 3.3.1 三角和双曲积分

三角和双曲积分主要有下述几种:

$$S_i(x) = \int_0^x \frac{\sin t}{t} dt$$

$$C_i(x) = gamma + \ln(x) + \int_0^x \frac{\cos t - 1}{t} dt$$

$$Ssi(x) = S_i - \frac{\pi}{2}$$

$$S_{hi}(x) = \int_0^x \frac{\sinh t}{t}$$

$$C_{hi}(x) = gamma + \ln(x) + \int_0^x \frac{\cosh t - 1}{t} dt$$

上述函数在Maple中的调用格式分别为: Si(x); Ci(x); Ssi(x); Shi(x); Chi(x); 其中x为表达式(复数).

函数Si, Ssi和Shi是完整的，函数Ci和Chi在原点处有一个对数极点，在负实半轴上有一个分支截断点.

```
> int(sin(x)/x,x=0..1);
```

$$\mathrm{Si}(1)$$

```
> evalf(%);
```

$$.9460830704$$

```
> Ci(3.14159+1.23*I);
```

$$-.02624028922 - .4706897380\ I$$

```
> Ssi(2002.118);
```

$$.0003014039122$$

```
> evalf(Shi(Pi));
```

$$5.469640347$$

```
> evalf(Chi(3+4*I));
```

$$-2.077477803 + 2.142816206\ I$$

```
> convert(Ci(x),Ei);
```

$$-\frac{1}{2}\mathrm{Ei}(1, I\,x) - \frac{1}{2}\mathrm{Ei}(1, -I\,x) + \frac{1}{2} I\,(\mathrm{csgn}(x) - 1)\,\mathrm{csgn}(I\,x)\,\pi$$

#### 3.3.2 Dirac 函数和 Heaviside 阶梯函数

Dirac 函数和 Heaviside 函数主要应用于积分变换或者求解微分方程，也可以用来表示分段连续函数．其定义分别如下:

$$\mathrm{Dirac}(t)=\begin{cases}0 & t\neq 0\\ \infty & t=0\end{cases}\qquad \mathrm{Heaviside}(t)=\begin{cases}0 & t<0\\ 1 & t>0\end{cases}$$

命令格式为:

Dirac(t); # Dirac函数(t=0时为无穷大，其余处处为0)

Dirac(n,t); # Dirac函数的n阶导数

Heaviside(t); #Heaviside函数(t<0时为0, t>0时为1, t=0时无意义)

```
> Int(Dirac(t),t=-infinity..infinity)=int(Dirac(t),t=-infinity..infinity);
```

$$\int_{-\infty}^{\infty}\mathrm{Dirac}(t)\,dt=1$$

```
> Int(Dirac(t),t)=int(Dirac(t),t);
```

$$\int\mathrm{Dirac}(t)\,dt=\mathrm{Heaviside}(t)$$

```
> Diff(Heaviside(t),t)=diff(Heaviside(t),t);
```

$$\frac{\partial}{\partial t}\mathrm{Heaviside}(t)=\mathrm{Dirac}(t)$$

### 3.3.3 指数积分

对于非负整数n，指数积分Ei(n,x)在实部Re(x)>0上定义为:

$$E_i(n,x)=\int_1^{\infty}\frac{e^{-xt}}{t^n}dt$$

单参数的指数积分是一个Cauchy主值积分，只对实参数x有如下定义:

$$E_i(x)=PV-\int_{-\infty}^{x}\frac{e^t}{t}dt$$

特别地，当x<0时，有: $E_i(x)=-E_i(1,-x)$

Ei(1,x)可以解析延拓到除了0点之外的整个复平面. 对于所有的这些函数，0是一个分支点，负实半轴是分支截断. 分支截断上的值要满足函数在增加参数的方向上是连续的条件.

指数函数和不完全GAMMA函数有如下关系:

$$E_i(n,x)=x^{n-1}GAMMA(1-n,x)$$

```
> Ei(1,1.); #=evalf(Ei(1,1));
```

$$.2193839344$$

```
> simplify(Ei(1,I*x)+Ei(1,-I*x));
```

$$-2\,\mathrm{Ci}(x)-I\pi+I\pi\,\mathrm{csgn}(x)$$

```
> expand(Ei(5,x));
```

$$\frac{1}{4}e^{(-x)}-\frac{1}{12}x\,e^{(-x)}+\frac{1}{24}x^2\,e^{(-x)}-\frac{1}{24}x^3\,e^{(-x)}+\frac{1}{24}x^4\,\mathrm{Ei}(1,x)$$

```
> evalf(Ei(1));
```

$$1.895117816$$

```
> int(exp(-3*t)/t,t=-x..infinity);
```

$$\mathrm{Ei}(1,-3\,x)$$

```
> int(exp(-3*t)/t,t=-x..infinity,CauchyPrincipalValue);
```

$$-\mathrm{Ei}(3\,x)$$

上述最后两例的结果大相径庭，原因是在最后一例中出现了“CauchyPrincipal Value”一选项，这一命令的主要功能是通知int将间断点的左右极限作为极限来处理，此时，独立变量按相同的速度接近间断点.

3.3.4 **对数积分**

对数积分Li(x)的定义为:

$$L_i(x)=PV-\int_0^x(1/\ln t)\mathrm{d}t=Ei(\ln x)\qquad(x\ge 0)$$

其中，PV-int表示Cauchy主值积分. 该函数只对实数参数 $x\ge 0$ 有定义，它给出了小于或等于x的素数的一个近似值.

```
> Li(2002.);    #对数积分在 2002.0 处的值
```

$$315.0723560$$

```
> nops(select(isprime, [$1..2002]));    # 小于或等于 2002 的实数中素数个数
```

$$303$$

```
> convert(Li(x),Ei);        #对数积分转换为指数积分
```

$$\mathrm{Ei}(\ln(x))$$

3.3.5 **椭圆积分**

所谓椭圆积分是形如 $\int_a^b R(x,y^{1/2})dx$ 的积分，其中R是一个有理数，y是3次或4次多项式，这是椭圆积分的代数形式. 除此之外还有三角形式、双曲三角等形式.

椭圆积分可以用初等函数项和椭圆函数项，如EllipticF，EllipticE和EllipticPi表示成它们的Legendre标准形式.

```
> ans:=int(sqrt(1+x^4)/(1-x^4),x=0..1/3);
```

$$ans:=-\frac{1}{8}\sqrt{2}\,(\ln(2)+\ln(\sqrt{41}-3))+\frac{1}{8}\sqrt{2}\,(\ln(2)+\ln(3+\sqrt{41}\,))$$
$$-\frac{1}{4}\sqrt{2}\,\arctan\left(\frac{1}{6}\sqrt{82}\,\sqrt{2}\right)+\frac{1}{8}\pi\,\sqrt{2}$$

```
> evalf(ans,20);
```

$$.33457573315002445140$$

```
> assume(0<k,k<1);
```

```
> int(x^2/sqrt((1-x^2)*(1-k^2*x^2)),x=0..k);
```

$$\frac{\mathrm{EllipticF}(k,k)}{k^2}-\frac{\mathrm{EllipticE}(k,k)}{k^2}$$

```
> int(1/sqrt(-(x-1)*(x-2)*(x-3)),x=0..1/2);
```

$$\sqrt{2}\ \mathrm{EllipticF}\left(\frac{2}{5}\sqrt{5},\frac{1}{2}\sqrt{2}\right)-\sqrt{2}\ \mathrm{EllipticF}\left(\frac{1}{3}\sqrt{2}\sqrt{3},\frac{1}{2}\sqrt{2}\right)$$

```
> evalf(%);
```

$$.270099742$$

### 3.3.5 换元积分法和分部积分法

换元积分法是积分计算中一种重要而实用的方法. 在Maple中, 对被积函数施行变量代换的命令是changevar, 该命令在工具包student中, 须先调用student工具包. 命令格式为:

changevar(s, f);

changevar(s, f, u);

changevar(t, g ,v);

其中, s是形式为h(x)=g(u)的一个将x定义为u的函数的表达式, f为积分表达式(如Int(F(x),x=a..b);), u为新的积分变量名称, t为定义的多元变量代换的方程组, g为二重或者三重积分, v为新变量的列表.

Changevar函数对积分、求和或者极限实现变量代换. 第1个参数s是用旧变量定义新变量的一个方程, 如果包含了两个以上的变量, 新变量必须放置在第3个参数位置, 而第2个参数是一个要被替换的表达式, 一般包含Int, Sum或者Limit等非求值形式(应尽量使用这种形式以便最后用value求值).

当问题为二重或三重积分时, 定义多元变量代换的方程由一个集合给出, 而新变量由一个列表给出.

```
> with(student):
> changevar(cos(x)+1=u,Int((cos(x)+1)^3*sin(x),x),u);
```

$$\int -u^3\,du$$

```
> changevar(x=sin(u),Int(sqrt(1-x^2),x=a..b),u);
```

$$\int_{\arcsin(a)}^{\arcsin(b)}\sqrt{1-\sin(u)^2}\,\cos(u)\,du$$

```
> changevar({x=r*cos(t),y=r*sin(t)},Doubleint(1,x,y),[t,r]);
```

$$\iint |r|\,dt\,dr$$

分部积分法(integration by parts)通过调用**student**工具包中的**intparts**来完成:

```
> with(student):
```

```
> int(x*exp(-a^2*x^2)*erf(b*x),x);
```

$$\int x\,\mathrm{e}^{(-a^2 x^2)}\,\mathrm{erf}(b\,x)\,dx$$

```
> intparts(%,erf(b*x));
```

$$-\frac{1}{2}\,\frac{\mathrm{erf}(b\,x)\,\mathrm{e}^{(-a^2 x^2)}}{a^2}-\int -\frac{\mathrm{e}^{(-b^2 x^2)}\,b\,\mathrm{e}^{(-a^2 x^2)}}{\sqrt{\pi}\,a^2}\,dx$$

```
> value(%);
```

$$-\frac{1}{2}\,\frac{\mathrm{erf}(b\,x)\,\mathrm{e}^{(-a^2 x^2)}}{a^2}+\frac{\frac{1}{2}\,b\,\mathrm{erf}(\sqrt{b^2+a^2}\,x)}{a^2\,\sqrt{b^2+a^2}}$$

## 3.3 重积分和线积分

在 Maple 中，重积分的形式函数有 **Doubleint**(二重)和 **Trippleint**(三重)，均在 **student** 工具包中，应用前需调用 student 工具包，它们适用于定积分和不定积分，可用 **value** 来获得积分结果的解析表达式. 命令格式为:

Doubleint(g, x, y);

Doubleint(g, x, y, Domain);

Doubleint(g, x = a..b , y = c..d );

Tripleint(g, x, y, z)

Tripleint(g, x, y, z, Domain)

Tripleint(g, x = a..b, z = e..f, y  = c..d )

其中, g 为积分表达式, x, y, z 为积分变量, Domain 为积分区域.

```
> with(student):
> Doubleint(f(x,y),x,y);
```

$$\iint \mathrm{f}(x,y)\,dx\,dy$$

比较以下两个实验:

```
> Doubleint(x+y,x=0..1,y=1..exp(x)):
  %=value(%);
```

$$\int_1^{\mathrm{e}^x}\int_0^1 x+y\,dx\,dy=\frac{1}{2}\,\mathrm{e}^x-1+\frac{1}{2}\,(\mathrm{e}^x)^2$$

```
> Doubleint(x+y,y=1..exp(x),x=0..1):
  %=value(%);
```

$$\int_0^1 \int_1^{e^x} x+y\,dy\,dx = -\frac{1}{4}+\frac{1}{4}(e)^2$$

在这两个形式函数中，我们还可以加入一个可选的参数，用来表示积分区域(通常用 S 表示二维区域，用 Ω 表示三维区域). 注意：在 Maple 中，这个参数仅仅用来做形式上的表示，不可以用来求值.

```
> Tripleint(x^2*y^2*z^2,x,y,z,Omega);
```

$$\iiint_{\Omega} x^2\,y^2\,z^2\,dx\,dy\,dz$$

在 Maple 中还有一个计算用参数方程形式表示的第一型曲线积分的函数—Lineint，它也在 student 工具包中. 下面通过一个实例说明这一函数的用法.

例：求曲线积分 $\int_C y^2 ds$，其中 C 为摆线 $x=a(t-\sin t), y=a(1-\cos t), 0\le t\le 2\pi$ 的一拱.

```
> with(student):
Lineint(y^2,x=a*(t-sin(t)),y=a*(1-cos(t)),t=0..2*Pi);
value(%);
```

$$\int_0^{2\pi} a^2\,(1-\cos(t))^2 \sqrt{\left(\frac{\partial}{\partial t} a\,(1-\cos(t))\right)^2+\left(\frac{\partial}{\partial t} a\,(t-\sin(t))\right)^2}\,dt$$

$$\frac{256}{15}\,\frac{a^4}{\sqrt{a^2}}$$

## 3.4 利用辅助手段积分

机器终归是机器，再聪明的机器也无法彻底代替人脑，Maple 也一样，它只能作为我们数学计算、推证的助手. 下面通过例子来体会 Maple 的真正用处.

例：求广义积分 $\int_0^{\infty} e^{-cx^2} dx$，其中 c>0.

按照常规，我们会通过下面的语句进行计算:

```
> Int(exp(-c*x^2), x=0..infinity) = int(exp(-c*x^2), x=0..infinity);
```

但 Maple 告诉我们，由于无法确定常数 $c$ 的正负号，因而无法确定积分是否收敛.

解决这一问题的办法是，通过 assume 设定 c 的取值范围:

```
> assume(c>0);
> Int(exp(-c*x^2),x=0..infinity)=int(exp(-c*x^2),x=0..infinity);
```

$$\int_0^{\infty} e^{(-c\sim x^2)}\,dx = \frac{1}{2}\frac{\sqrt{\pi}}{\sqrt{c\sim}}$$

解决这一问题的另一方法是假设 c 是另一个参数 p 的绝对值(**c:=abs(p)**)，这样 c 就自然是一个非负的参数了.

```
> with(student):
> int(x*exp(-a^2*x^2)*erf(b*x),x);
```

$$\int x\,e^{(-a^2 x^2)}\,\mathrm{erf}(b\,x)\,dx$$

```
> intparts(%,erf(b*x));
```

$$-\frac{1}{2}\frac{\mathrm{erf}(b\,x)\,e^{(-a^2 x^2)}}{a^2} - \int -\frac{e^{(-b^2 x^2)}\,b\,e^{(-a^2 x^2)}}{\sqrt{\pi}\,a^2}\,dx$$

```
> value(%);
```

$$-\frac{1}{2}\frac{\mathrm{erf}(b\,x)\,e^{(-a^2 x^2)}}{a^2} + \frac{\frac{1}{2}\,b\,\mathrm{erf}(\sqrt{b^2+a^2}\,x)}{a^2\sqrt{b^2+a^2}}$$

其中, erf(x)为误差函数(**error function**)，定义为: $erf(x) = \dfrac{2}{\sqrt{\pi}}\displaystyle\int_0^x e^{-t^2}\,dt$ .

**例**: 求证 $\displaystyle\int_0^{2\pi}\frac{1}{1+3\sin t^2}dt = \pi$

在 Maple7 中，该积分直接利用 Maple 计算也可完成证明：

```
> Int(1/(1+3*sin(t)^2),t=0..2*Pi)=int(1/(1+3*sin(t)^2),t=0..2*Pi);
```

$$\int_0^{2\pi}\frac{1}{1+3\sin(t)^2}\,dt = \pi$$

这里，我们将其作为一个例子，试图说明应用有关数学理论知识转化问题，然后再利用 Maple 进行辅助计算的方法和技巧。

由复变函数理论知道, 此积分可用围道积分(contour integration)来求解. 首先, 我们把被积函数写成复变量 $z = e^{it}$ 的形式, 然后把原问题转化成围道积分, 再求得结果. 过程为:

```
> p:=1/(1+3*sin(t)^2);
```

$$p := \frac{1}{1+3\sin(t)^2}$$

```
> convert(p,exp);
```

$$\frac{1}{1-\frac{3}{4}\left(e^{(I\,t)}-\frac{1}{e^{(I\,t)}}\right)^2}$$

```
> factor(%)/diff(exp(I*t),t);
```

$$\frac{4\,I\,e^{(I\,t)}}{(3\,(e^{(I\,t)})^2-1)\,((e^{(I\,t)})^2-3)}$$

```
> g:=subs(exp(I*t)=Z,%);
```

$$g := \frac{4\,I\,Z}{(3\,Z^2-1)\,(Z^2-3)}$$

```
> solve(denom(g)=0,Z);
```

$$\sqrt{3},\ -\sqrt{3},\ \frac{1}{3}\sqrt{3},\ -\frac{1}{3}\sqrt{3}$$

```
> readlib(residue):
> residue(g,Z=1/3*sqrt(3));
```

$$\frac{-1}{4}\,I$$

```
> residue(g,Z=-1/3*sqrt(3));
```

$$\frac{-1}{4}\,I$$

```
> 2*pi*I*(%+%%);
```

$$\pi$$

其中，函数residue(f, x=a) 计算表达式f对变量$x$在a点附近的代数残差(algebraic residue)，残差定义为f的Laurent级数中(x-a)^(-1)的系数.

# 4 级　数

## 4.1 数值级数和函数项级数求和以及审敛法

我们可以用 Maple 中的函数 **sum** 方便地求得级数的和，无论是有限项还是无穷项，常数项级数还是函数项级数．相应地，和式的形式函数是 Sum．求连乘积使用命令 **product**.

```
> Sum(1/(4*k^2-1), k=1..infinity)= sum(1/(4*k^2-1), k=1..infinity);
```

$$\sum_{k=1}^{\infty} \frac{1}{4k^2-1} = \frac{1}{2}$$

```
> Sum(i^2,i=1..n)=sum(i^2,i=1..n);
```

$$\sum_{i=1}^{n} i^2 = \frac{1}{3}(n+1)^3 - \frac{1}{2}(n+1)^2 + \frac{1}{6}n + \frac{1}{6}$$

```
> Product(1/k^2,k=1..n)=product(1/k^2,k=1..n);
```

$$\prod_{k=1}^{n} \frac{1}{k^2} = \frac{1}{\Gamma(n+1)^2}$$

Maple 对级数求和的方法如下:

(i) 多项式级数求和用贝努利级数公式: $\sum_{k=0}^{n-1} k^m = \frac{1}{m+1} C_{m+1}^{k} B_k n^{m+1-k}$，其中，贝努利数 $B_k$ 由以下隐式递推公式定义: $B_0 = 1, \sum_{k=0}^{m} C_{m+1}^{k} B_k = 0$

(ii) 有理函数级数求和是用 Moenck 方法，得到的结果是一个有理函数加伽玛函数(Polygamma function) $\psi$ 及其导数的项;

(iii) Gosper 算法是 Risch 算法的离散形式，它被用到求包含级乘和乘幂的级数和上;

(iv) 计算无穷项级数的和有时会用到超比级数.

收敛或发散是级数的重要性质，在这里主要以绝对收敛的比值审敛法(ratio test for absolute convergence)为例说明 Maple 的使用，其余类推.

绝对收敛的比值审敛法的数学原理是:

设 $\sum a_k$ 为不含 0 的交错级数，并令 $\rho = \lim\limits_{k\to\infty} \frac{|a_{k+1}|}{|a_k|}$，则:

1） 若 $\rho$<0，级数绝对收敛;

2） 若 $\rho$>0 或 $\rho=\infty$，级数发散;

3） 若 $\rho$=1，待定

例: 判定级数 $\sum_{k=1}^{\infty} \frac{(-1)^k 2^k}{k!}$ 是否绝对收敛.

```
> f:=k->(-1)^k*12^k/(k!);
```

$$f := k \to \frac{(-1)^k\, 12^k}{k!}$$

```
> r:=simplify(abs(f(k+1))/abs(f(k)));
```

$$r := 12\frac{1}{|k+1|}$$

```
> Limit(r,k=infinity);
```

$$\lim_{k\to\infty} 12\frac{1}{|k+1|}$$

```
> value(%);
```

$$0$$

由此可见，r=0<1，级数绝对收敛．事实上，还可以用 sum 对级数求和，确定级数收敛于一个定值:

```
> sum(f(k),k=1..infinity);
```

$$\mathrm{e}^{(-12)}(1-\mathrm{e}^{12})$$

## 4.2 幂级数

幂级数的有关计算在专门的工具包 **powseries** 中，这个工具包含有生成和处理幂级数的各种常用工具．如果我们已知一个幂级数的系数，就可以用函数 **powcreate** 来生成它，其参数是系数所满足的方程或方程组，格式为:

```
> with(powseries);
> powcreate(t(n)=3^sqrt(n));
```

没有任何结果显示出来，事实上，此时, Maple 已经按照要求把该幂级数的系数函数赋给了 t(n)，用下述命令即可看出:

```
> t(2);
```

$$3^{(\sqrt{2})}$$

显然，这样的级数很不直观，更多的时候我们需要幂级数的截断表达式(truncated power series form)，此时可以通过该工具包中的 **tpsform** 命令完成，这是一个很有用的 Maple 函数:

```
> tpsform(t,x,8);
```

$$1+3x+3^{(\sqrt{2})}x^2+3^{(\sqrt{3})}x^3+3^{(\sqrt{4})}x^4+3^{(\sqrt{5})}x^5+3^{(\sqrt{6})}x^6+3^{(\sqrt{7})}x^7+\mathrm{O}(x^8)$$

但是，大多数情况下，我们并不知道幂级数的系数，而只知道幂级数的和的解析表达式，需要把它展开成和式．对于一些常用的函数，**powseries** 工具包中有一些函数可以生成对应的幂级数，比如 sin(p)、cos(p)、exp(p)、ln(p)、sqrt(p)的幂级数可以分别通过 powsin(p)、powcos(p)、powexp(p)、powlog(p)、powsqrt(p)来得到，其中, p 可以是单变量函数或表达式甚至级数.

```
> t:=powlog(1+x+x^2):  tpsform(t,x,6);
```

$$x+\frac{1}{2}x^2-\frac{2}{3}x^3+\frac{1}{4}x^4+\frac{1}{5}x^5+\mathrm{O}(x^6)$$

对于多项式，可以用 **powpoly(expr, x)** 得到对应的幂级数，其中 **expr** 是多项式, x 是

变量. 而任意函数的表达式可以通过函数 **evalpow** 来获得其幂级数形式.

```
> t:=evalpow(1/(1-3*x+2*x^2)):
> tpsform(t,x,6);
```

$$1+3x+7x^2+15x^3+31x^4+63x^5+O(x^6)$$

掌握了级数的有关生成后, 可以对这些级数进行运算了, **powseries** 工具包中具有对幂级数的各种运算: 加(**powadd**)、减(**subtract**)、乘(**multiply**)、除(**quotient**)、求负(**negative**)、求倒数(**inverse**)、复合(**compose**)、求逆(**reversion**)、求导(**powdiff**)、积分(**powint**)等. 下面通过实例学习这些运算.

```
> restart:with(powseries):
powcreate(t(n)=t(n-1)/n,t(0)=1):
powcreate(v(n)=v(n-1)/2,v(0)=1):
s:= powadd(t, v):   tpsform(s, x, 7);
```

$$2+\frac{3}{2}x+\frac{3}{4}x^2+\frac{7}{24}x^3+\frac{5}{48}x^4+\frac{19}{480}x^5+\frac{49}{2880}x^6+O(x^7)$$

```
> p:=multiply(t,v):  tpsform(p,x,7);
```

$$1+\frac{3}{2}x+\frac{5}{4}x^2+\frac{19}{24}x^3+\frac{7}{16}x^4+\frac{109}{480}x^5+\frac{331}{2880}x^6+O(x^7)$$

```
> q:=quotient(t,v):  tpsform(q,x,7);
```

$$1+\frac{1}{2}x-\frac{1}{12}x^3-\frac{1}{24}x^4-\frac{1}{80}x^5-\frac{1}{360}x^6+O(x^7)$$

```
> w:=powdiff(t):  tpsform(u,x,6);
```

$$1+x-2x^2+x^3+x^4-2x^5+O(x^6)$$

```
> u:=powint(v):  tpsform(u,x,6);
```

$$x+\frac{1}{4}x^2+\frac{1}{12}x^3+\frac{1}{32}x^4+\frac{1}{80}x^5+O(x^6)$$

## 4.3 泰勒级数和劳朗级数

在 Maple 中, 可以用命令 **taylor** 方便快捷地得到一个函数或表达式在一点的任意阶 Tayloe 展开式, 而一般级数展开命令为 series. 命令格式为:

taylor(expr,eqn/nm,n);

series(expr, eqn, n);

其中, expr 表示表达式, eqn/nm 表示方程(如 x=a)或名称(如 x), n(非负整数)表示展开阶数.

在调用 taylor 或 **series** 级数时, 只需要指定有待展开的表达式、展开点、展开的阶数就可以了. 如果不给定展开点, 默认为 0 点, 如果不指定展开阶数时默认为 6 阶(命令 **order** 可获取截断级数的展开阶数). 另外, series 函数可以展开更一般的截断函数, 比如 laurent 级数等, 它会根据情况决定展开成什么类型级数.

```
> taylor(sin(tan(x))-tan(sin(x)),x=0,19);
```

$$-\frac{1}{30}x^7-\frac{29}{756}x^9-\frac{1913}{75600}x^{11}-\frac{95}{7392}x^{13}-\frac{311148869}{54486432000}x^{15}+O(x^{17})$$

```
> series(GAMMA(x),x=0,3);
```

$$x^{-1}-\gamma+\left(\frac{1}{12}\pi^2+\frac{1}{2}\gamma^2\right)x+\left(-\frac{1}{3}\zeta(3)-\frac{1}{12}\pi^2\gamma-\frac{1}{6}\gamma^3\right)x^2+O(x^3)$$

```
> order(%);
```

$$3$$

如果需要限制展开为 lauren 级数, 可以使用 numapprox 工具包中的 laurent 函数, 否则, series 有可能会得出一些更为广义的级数, 比如 Puisseux 级数:

```
> with(numapprox);
> laurent(exp(x), x=0,10);
```

$$1+x+\frac{1}{2}x^2+\frac{1}{6}x^3+\frac{1}{24}x^4+\frac{1}{120}x^5+\frac{1}{720}x^6+\frac{1}{5040}x^7+\frac{1}{40320}x^8+\frac{1}{362880}x^9+O(x^{10})$$

```
> series(1/(x*(1+sqrt(x))),x=0,3);
```

$$\frac{1}{x}-\frac{1}{\sqrt{x}}+1-\sqrt{x}+x-x^{(3/2)}+x^2-x^{(5/2)}+O(x^3)$$

Maple 在计算截断级数上还具有一个有用的特性, 即待展开的表达式或函数不一定要有解析表达式, 而只要可以求导就能计算.

```
> Int(exp(x^3),x)=int(exp(x^3),x);
> series(rhs(%),x=0);
```

$$\int e^{(x^3)}\,dx=-\frac{1}{3}(-1)^{(2/3)}\left(\frac{2}{3}\frac{x\,(-1)^{(1/3)}\,\pi\sqrt{3}}{\Gamma\left(\frac{2}{3}\right)(-x^3)^{(1/3)}}-\frac{x\,(-1)^{(1/3)}\,\Gamma\left(\frac{1}{3},-x^3\right)}{(-x^3)^{(1/3)}}\right)$$

$$x+\frac{1}{4}x^4+O(x^7)$$

与前面介绍的幂级数不同, 这里可以直接对截断形式的级数进行求导和积分运算:

```
> sin_series:=series(sin(x),x);
```

$$sin\_series:=x-\frac{1}{6}x^3+\frac{1}{120}x^5+O(x^6)$$

```
> diff(sin_series,x);
```

$$1-\frac{1}{2}x^2+\frac{1}{24}x^4+O(x^5)$$

```
> int(%,x);
```

$$x-\frac{1}{6}x^3+\frac{1}{120}x^5+O(x^6)$$

关于多元级数需调用函数 **mtaylor**, 结果是一般的多项式而非截断函数, 自然就可以进行多项式的运算了. 命令格式为:

mtaylor(f,v);

mtaylor(f,v,n);

mtaylor(f,v,n,w);

其中, f 表示代数表达式, v 表示方程的列表或集合, n(可选, 默认值为 6)表示截断阶数的非负整数, w(可选)表示自然数列表, 权变量列表.

```
> mtaylor(sin(x^2+y^2),[x,y],8);
```

$$x^2+y^2-\frac{1}{6}x^6-\frac{1}{2}y^2x^4-\frac{1}{2}y^4x^2-\frac{1}{6}y^6$$

```
> whattype(%);
```

$$+$$

更进一步, 可以用 **normal** 函数对展开式进行正则化(其更主要的功能是对有理函数化简), 用 **coeff** 获得展开式的系数, 与多项式不同的是, 对于级数, **coeff** 只能获得主变量的幂系数. 实际上, 更为方便的是, 可以利用库函数 **coeftayl**, 不展开就能获得泰勒级数的系数, 计算 ***f*** 在 **x=x₀** 点的泰勒展开式中 $(x-x_0)^k$ 的系数的命令格式为:

**coeftayl (f, x=x₀, k)**;

```
> normal((f(x)^2-1)/(f(x)-1));
```

$$f(x)+1$$

```
> normal(sin(x*(x+1)-x));
```

$$\sin(x^2)$$

```
> normal(1/x+x/(x+1),expanded);
```

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

```
> coeftayl(series(sin(x),x=0,20),x=0,11);
```

$$\frac{-1}{39916800}$$

```
> coeftayl(series(exp(x),x=0,30),x=1,13);
```

$$\frac{56874039553217}{1302866478266056704000000}$$

## 5 积分变换

无论在数学理论研究还是在数学应用中，积分变换都是一种非常有用的工具. 积分变换就是将一个函数通过参变量积分变为另一个函数. 常用的积分变换包括拉普拉斯变换(**Laplace transforms**)、傅里叶变换(**Fourier transforms**)、梅林变换(**Melin transforms**)以及汉克尔变换(Hankel transforms)等. 函数 f 的积分变换的定义如下:

$$T(f)(s)=\int_a^b f(t)K(s,t)dt$$

其中 ***K*** 为变换核. 具体如下:

| 变换名称 | 定　义 | 变换命令 | 逆变换命令 |
|---|---|---|---|
| 拉普拉斯 | $\int_0^{\infty} f(t)e^{-st}dt$ | **laplace(f(t), t, s)** | **invlaplace(f(t), t, s)** |
| 傅 里 叶 | $\int_{-\infty}^{+\infty} f(t)e^{-ist}dt$ | **fourier(f(t), t,s)** | **invfourier(f(t), t,s)** |
| 梅　　林 | $\int_0^{\infty} f(t)e^{s-1}dt$ | **melin(f(t), t,s)** | **invmelin(f(t), t,s)** |

这些函数都在 **inttrans** 工具包中, 使用时用 **with** 引入.

## 5.1 拉普拉斯变换

形如多项式或者一些特定函数(如 Diract function、Heaviside function、Bessel function 等)组成的有理表达式或它们的和, 都可以用 Maple 积分变换工具包 **inttrans** 中的函数 **laplace** 求得其拉普拉斯变换, 相应的逆变换用 **invlaplace**.

```
> t^2-exp(t)+sin(a*t);
```

$$t^2-e^t+\sin(a\,t)$$

```
> with(inttrans):
> laplace(%%,t,s);
```

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

```
> invlaplace(%,s,t);
```

$$t^2-e^t+\sin(a\,t)$$

所有的积分变换都主要应用于微分和积分方程领域。由于拉普拉斯变换的导数和积分性质, 它被大量地应用到微分方程和积分方程的求解上. 作为拉普拉斯变换的一个应用, 下面求解 Volterra 积分方程:

```
> with(inttrans):
> e:=u(t)+int((t-x)*u(x),x=0..t)-cos(t);
```

$$e:=u(t)+\int_0^t (t-x)\,u(x)\,dx-\cos(t)$$

```
> laplace(e,t,s);
```

$$\mathrm{laplace}(\mathrm{u}(t), t, s) + \frac{\mathrm{laplace}(\mathrm{u}(t), t, s)}{s^2} - \frac{s}{s^2+1}$$

下面要从这个简单的代数方程中解出 **laplace(e,t,s)**来，值得注意的是要解出来的不是一个变量，而是一个子式。当然，我们可以先用 **subs** 将这个子式代换为一个变量，再用 **solve** 求解。但在 Maple 中有一个更方便的命令：库函数 **isolate** 可直接从方程是解出这个子式来：

```
> readlib(isolate)(%,laplace(e,t,s));
```

$$\mathrm{laplace}(\mathrm{u}(t), t, s) + \frac{\mathrm{laplace}(\mathrm{u}(t), t, s)}{s^2} - \frac{s}{s^2+1} = 0$$

然后求解变换后的结果 laplace(u(t),t,s)：

```
> solve(%,laplace(u(t),t,s));
```

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

作反变换即可得到 u(t)，也就是原方程的解：

```
> u(t):=invlaplace(%,s,t);
```

$$\mathrm{u}(t) := \left(-\frac{1}{2}\,t\,\sin(t) + \cos(t)\right)\mathrm{Dirac}(t)$$

再看另外一类积分方程的 Laplace 变换求解：

```
> int_eqn:=int(exp(a*x)*f(t-x),x=0..t)+b*f(t)=t;
```

$$int\_eqn := \int_0^t \mathrm{e}^{(a\,x)}\,\mathrm{f}(t-x)\,dx + b\,\mathrm{f}(t) = t$$

对其进行拉普拉斯变换，得到 f(t)的变换 L(f)(s)的方程：

```
> laplace(%,t,s);
```

$$\frac{\mathrm{laplace}(\mathrm{f}(t), t, s)}{s-a} + b\,\mathrm{laplace}(\mathrm{f}(t), t, s) = \frac{1}{s^2}$$

```
> readlib(isolate)(%,laplace(f(t),t,s));
```

$$\mathrm{laplace}(\mathrm{f}(t), t, s) = \frac{1}{s^2\left(\frac{1}{s-a} + b\right)}$$

再对上式做逆变换，即可得所求：

```
> invlaplace(%,s,t);
```

$$f(t)=\frac{a\,t}{-1+b\,a}-\frac{2\sinh\left(\frac{1}{2}\frac{(-1+b\,a)\,t}{b}\right)e^{\left(1/2\,\frac{(-1+b\,a)\,t}{b}\right)}}{(-1+b\,a)^2}$$

## 5.2 傅里叶变换

Maple 工具包 **inttrans** 中的函数 **fourier** 可以求表达式的傅里叶变换:

```
> with(inttrans):
> 1/(1+t^2);
```

$$\frac{1}{1+t^2}$$

```
> fourier(%,t,omega);
```

$$e^{\omega}\,\pi\,\mathrm{Heaviside}(-\omega)+e^{(-\omega)}\,\pi\,\mathrm{Heaviside}(\omega)$$

上式的结果中出现了 **Heaviside** 函数，它的导数是 **Diract** 函数，它在负半轴上取 0, 在正半轴上取 1. 一般地，可对其做逆变换:

```
> invfourier(%,omega,t);
```

$$-\frac{1}{(1+I\,t)\,(-1+I\,t)}$$

```
> normal(evalc(%));
```

$$\frac{1}{1+t^2}$$

对于一些函数，比如三角函数、Diract 函数、Heaviside 函数和 Bessel 函数等, Maple 会利用卷积定理、查表、定积分等方法求它们的傅里叶变换:

```
> fourier(BesselJ(0,t),t,omega);
```

$$2\,\frac{-\mathrm{Heaviside}(\omega-1)+\mathrm{Heaviside}(\omega+1)}{\sqrt{1-\omega^2}}$$

我们甚至可以扩充 Maple 里的傅里叶变换表，尽管 Maple 内部的公式表可能比任何一本教科书都要完全. 例如，定义函数 f(t)的傅里叶变换为 $F(s)/(1+s^2)$, 然后利用 Maple 中的函数 **addtable** 将其加到变换表中去:

```
> addtable(fourier,f(t),F(s)/(1+s^2),t,s);
> fourier(exp(3*I*t)*f(2*t),t,omega);
```

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

**addtable** 是 **inttrans** 中的函数，它不仅可以用来扩充傅里叶变换的变换表，对其他的积分也适用. 其第一个参数是积分变换名，第二个是函数，第三个是该函数经过变换后的函数，后两个参数是变换前后函数的自变量.

函数 **fourier** 和 **invfourier** 是用来计算符号表达式的连续傅里叶变换的，对于离散域上的数值傅里叶变换，可以调用快速傅里叶变换(Fast Fourier Transformation)函数 **FFT**，相应的逆变换函数是 **iFFT**.

对于一个长度为 N 的离散数值向量 $x=[x_0, x_1, \cdots\cdots, x_{N-1}]$，其傅里叶变换 $X=[X_0, X_1, \cdots\cdots, X_{N-1}]$的定义如下:

$$X_k = \sum_{j=1}^{N-1} x_j e^{\frac{2\pi ijk}{N}}$$

其中，$0 \le k \le N-1$. 快速傅里叶变换的算法决定了 N 必须为 2 的整数次幂. 作为例子，我们取 $N = 2^3$ 来计算实数序列[1, 1, 1, 1, -1, -1, -1, -1]的傅里叶变换. 和 **fourier** 不同, **FFT** 与 **iFFT** 都是函数 (Maple 7 以前的版本是库函数)，而不是 **inttrans** 工具包中的函数.

**FFT** 是用来对复数序列进行傅里叶变换的，需要分别提供数据的实部和虚部.

```
> x:=array([1,1,1,1,-1,-1,-1,-1]);
```

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

```
> y:=array([0,0,0,0,0,0,0,0]);
```

$$y := [0, 0, 0, 0, 0, 0, 0, 0]$$

```
> FFT(3,x,y);
```

$$8$$

```
> print(x);
```

$$[0, 2.000000001, 0., 1.999999999, 0, 1.999999999, 0., 2.000000001]$$

```
> print(y);
```

$$[0, -4.828427122, 0., -.828427124, 0, .828427124, 0., 4.828427122]$$

FFT 的第一个参数表明了序列的元素个数，上面的 3 表示要换的数是 $2^3$ 个，后两上参数分别是变换的实部和虚部，它们的数据类型都是数组(**array**)，可以用函数 **array** 从数据列表生成. 有关数组的显示用 **print** 命令. 从上面的结果可以看出，FFT 把变换的结果直接赋给了输入的数组，而不是把结果作为返回值返回.

再用 **iFFT** 检验变换的正确性:

```
> iFFT(3,x,y);
```

$$8$$

```
> print(x);
```

$$[1.000000000, .9999999990, .9999999995, .9999999985, -1.000000000, -.9999999990, -.9999999995, -.9999999985]$$

```
> print(y);
```

$$[0., -.2500000000\ 10^{-9}, 0., .2500000000\ 10^{-9}, 0., .2500000000\ 10^{-9}, 0., -.2500000000\ 10^{-9}]$$

这里的结果是将实部与虚部分开的，我们可以用 zip 函数把他们合成为复数形式:

```
> zip((a,b)->a+b*I,x,y);
```

$[1.000000000 + 0.\ I, .9999999990 - .2500000000\ 10^{-9}\ I, .9999999995 + 0.\ I, .9999999985 + .2500000000\ 10^{-9}\ I,$
$-1.000000000 + 0.\ I, -.9999999990 + .2500000000\ 10^{-9}\ I, -.9999999995 + 0.\ I, -.9999999985 - .2500000000\ 10^{-9}\ I]$

**zip** 的第一个参数是一个二元函数，它将后两个向量中的每一个元素按此函数结合成一个新的向量.

## 5.4 其他积分变换

在 Maple 中还有一些其他的积分变换，它们的调用格式和前面的基本一样.

| 变　换 | 定　义 | Maple 中的函数 |
|---|---|---|
| 傅里叶余弦变换 | $\sqrt{\frac{2}{\pi}}\int_0^{\infty} f(t)\cos(st)dt$ | **fourierscos(f(t),t,s)** |
| 傅里叶正弦变换 | $\sqrt{\frac{2}{\pi}}\int_0^{\infty} f(t)\sin(st)dt$ | **fourierssin(f(t),t,s)** |
| 汉克尔变换 | $\int_0^{\infty} f(t)\sqrt{st}BesslJ(v,st)dt$ | **hankel(f(t), t, s, nu)** |
| 希尔伯特变换 | $\frac{1}{\pi}\int_{-\infty}^{+\infty}\frac{f(t)}{t-s}dt$ | **hilbert(f(t), t, s)** |

下面通过几个例子说明这些变换的应用：

① Hankel 变换：

```
> with(inttrans):
> assume(k,integer,k>0):
> hankel(sqrt(t^2),t,s,k);
```

$$2\frac{\sqrt{2}\ \Gamma\left(\frac{1}{2}k+\frac{5}{4}\right)}{s^2\ \Gamma\left(\frac{1}{2}k-\frac{1}{4}\right)}$$

② Fourier正余弦变换

```
> assume(a>0):
> fouriercos(Heaviside(a-t),t,s);
```

$$\frac{\sqrt{2}\,\sin(a\,s)}{\sqrt{\pi}\,s}$$

```
> fouriercos(%,s,t);
```

$$\text{Heaviside}(a-t)$$

```
> fouriersin(Heaviside(a-t),t,s);
```

$$2\,\frac{\sqrt{2}\,\sin\left(\frac{1}{2}\,a\,s\right)^2}{\sqrt{\pi}\,s}$$

③ Hilbert 变换

```
> assume(k,integer,k>0):
> hilbert(Dirac(x)+sin(k*x)/x,x,y);
```

$$\frac{-1+\pi\cos(y\,k)-\pi}{y\,\pi}$$

④ Mellin 变换

```
> mellin(1/(1+t),t,s);
```

$$\pi\csc(\pi\,s)$$

```
> mellin(ln(1+t),t,s);
```

$$\frac{\pi\csc(\pi\,s)}{s}$$