第 3 章 微积分

3.1 求极限

计算函数极限 $\lim_{x->x_0} f(x)$ 的一般形式是：**Limit [** f [x], x -> x0 **]**

下列计算函数极限的三种形式：

Limit [expr, x -> x0**]** x -> x_0 时函数 expr 的极限

Limit [expr, x -> x0, Direction –> 1**]** x -> x_0^- 时函数 expr 的极限

Limit [expr, x -> x0, Direction –> –1**]** x -> x_0^+ 时函数 expr 的极限

例如：计算 1) $\lim_{x->0}\frac{\sin x}{x}$ 2) $\lim_{x->\infty}\frac{x^2-1}{4x^2-7x+1}$ 3) $\lim_{x->0}\sin\frac{1}{x}$

4) $\lim_{x->0^+}\frac{1}{x}$ 5) $\lim_{x->0^-}\frac{1}{x}$ 6) $\lim_{x->1} xg(x)$

7) $\lim_{x->0}\frac{\tan x - x}{x^3}$

例题：

```
Limit[Sin[x]/x, x->0]
Limit[(x^2-1)/(4x^2-7x +1), x->Infinity]
Limit[Sin[1/x], x ->0]
Limit[1/x, x ->0, Direction ->1]
Limit[1/x, x -> 0, Direction -> -1]
Limit[x g[x], x ->1]
Limit[E^x^x - E^x^(2 x), x->Infinity]
```

3.2 微商和微分

3.2.1 微商（导数）

微商函数	功 能
D[f, x]	计算偏导数 $\frac{\partial}{\partial x} f$

D[f, x1, x2,....]	计算多重导数 $\frac{\partial}{\partial x1}\frac{\partial}{\partial x2}\ldots f$
D[f, {x, n}]	计算 n 阶导数 $\frac{\partial^n}{\partial x^n} f$
D[f, x, NonConstants ->{v1,v2,....}]	计算 $\frac{\partial}{\partial x} f$ 其中 v1,v2,....依赖于 x

表 3-1

例如：1) 计算 x^{x^x} 的一阶导数　　2) $\frac{\partial}{\partial x \partial y} z\sin(x^2 y^2)$

3) 计算 cx^n 的二阶导数

```
D[x^x^x, x]
D[z Sin[x^2 y^2],x,y]
D[c x^n,{x,2}]
D[z[1]^2+Sin[z[1]+z[2]],z[1]]    (* 以 z[1]为变量求导 *)
D[x^2+y^2, x, NonConstants ->{y}]    (* y 是 x 的函数 *)
```

3.2.2 全导数

✧ 全微分函数 Dt

全微分函数 Dt	说 明
Dt[f]	全微分 df
Dt [f, x]	全导数 $\frac{df}{dx}$
Dt [f, x1, x2,...]	多重全导数 $\frac{d}{dx1}\frac{d}{dx2}\ldots f$
Dt [f, x, Constants->{c1,c2,...}]	全导数，说明 ci 为常数（即 $\frac{d(ci)}{dx}$ =0）
y/: Dt [y, x] = 0	置 $\frac{dy}{dx}$=0

表 3 - 2

例如： 1) 计算 $\frac{\partial(x^2+y^2)}{\partial x}$　　2) 计算 $d(x^2+y^2)$

3) 全导数 $\frac{d}{dx}(x^2+y^2)$　　4) $\frac{d}{dx}(x^2+y^2+z^2)$

```
D[x^2+y^2, x]
Dt[x^2+y^2]
Dt[x^2+y^2, x]
```

```
Dt[x^2+y^2+z^2, x]
Dt[x^2+y^2+z^2, x,Constants –>{z}]
```
(* 说明 z 为常数，即 Dt[z,x]=0 *)

设 $u = e^{a\theta}\cos(a\ln r)$ ，证明 ： $\frac{\partial^2 u}{\partial r^2}+\frac{1}{r^2}\frac{\partial^2 u}{\partial \theta^2}+\frac{1}{r}\frac{\partial u}{\partial r}=0$

```
u = Exp[a s]Cos[a Log[r]];
D[u,{r, 2}] + 1/r^2 D[u,{s,2}]+1/r D[u, r]
Simplify[%]
```

令 $F(x)=\int_1^{x^2} e^{\cos t}dt$,计算 $F'(x)$

```
F′[x]
```

✧ **抽象函数的导数表示**

在表达式中也能使用微积分中计算导数的撇号标记。例如：

f'[x] 单变量函数的一阶导数

f'''[x] 单变量函数的三阶微分

```
f[x_]:=Sin[x^2]
```
（* 定义函数 f (x)= x 2 *）

```
f[x]+f'[x]+f''[x]
D[x g[2x], x]
```

```
D[h[x,y],x,x,y]
```
(*计算 $\frac{\partial}{\partial x^2}\frac{\partial}{\partial y}h^{(3)}(x,y)$ *)

3.2.3 定义导数

在 *Mathematica* 中定义函数的一阶导数的方式就像定义函数一样，例如：f '[x_]:= hf[x] x_ 表示 x 是函数变量，相当于函数定义中的形式参量，有关定义函数的详细描述请看第 7 章。定义函数的高阶导数或多个参量的导数要用函数 Derivative。

下列定义导数的一般形式：

f '[x]:= rhs 定义 f 的一阶导数

Derivative [n][f][x]:= rhs 定义 f 的 n 阶导数

Derivative [n1,n2,...][g][x1,x2,...]:= rhs 定义相对于 g 的多个参量的导数

例如：

```
In[1]:= f'[x_]:= h f[x]+g f[x]
In[2]:= D[f[x^2],x]
```

3.3 不定积分和定积分

3.3.1 积分计算

函数 Integrate [f, x] 计算不定积分 $\int f(x)dx$，输出结果中省略积分常数。Integrate 函数假定任何不显含积分变量的对象与积分变量无关，都当常数处理，因而 Integrate 很象微分 D 的逆。

Integrate [f, x**]** 计算不定积分 $\int f(x)dx$

Integrate [f, x, y **]** 计算不定积分 $\int dx\int f(x,y)dy$

Integrate [f, x, y, z**]** 计算不定积分 $\int dx\int dy\int f(x,y,z)dz$

例如：计算 1) $\int 3ax^2dx$ 2) $\int 3f'(x)f^2(x)dx$

3) $\iint(3x^2+2y)dxdy$ 4) $\int\sqrt{tgx}dx$

```
Integrate[3a x^2, x]
Integrate[3f'[x]f[x]^2, x]
Integrate[3x^2+2y, x, y]
Integrate[Sqrt[Tan[x]],x]
```

积分 $\int\sqrt{\cos x}dx$ 的结果是椭圆函数。

```
Integrate[Sqrt[Cos[x]],x]
Integrate[Sin[a x]/x,{x,0,Infinity}]
```

给出明确的附加条件 Im[a]为零，积分值也随之确定。

```
Integrate[Sin[a x]/x,{x,0,Infinity},Assumptions -> Im[a] == 0]
```

3.3.2 不定积分的计算范围

```
In[1]:= Integrate[Sin[Sin[x]],x]      (*原函数不是初等函数 *)
Out[1]= Integrate[Sin[Sin[x]],x]      （*算不出结果按原输入命令输出 *）
```

有时 *Mathematica* 积出被积函数的一部分，将剩余部分用 Integrate[g,x] 表示。

Mathematica 除了能计算简单积分外，还能计算一类如贝塞尔 BesselJ[n,z]、 Gamma[z] 等二三十个数学物理特殊函数。这些特殊函数极大地扩充了 *Mathematica* 的积分范围。

例如：Gamma[z]函数定义的积分 $\Gamma(z)=\int_0^\infty t^{a-1}e^{-t}dt$

Beta [a,b]函数定义的积分　　$B(a,b)=\dfrac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}=\int_0^1 t^{a+1}(1-t)^{b-1}dt$

还可以通过定义函数或程序包来增强 *Mathematica* 积分的功能。

3.3.3 定积分计算

计算定积分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 Integrate 函数相同。Integrate 按牛顿-莱布尼兹公式 $\int_a^b f(x)dx=F(b)-F(a)$ 计算定积分得到的是准确解，NIntegrate 用数值积分公式计算定积分得到的是近似数值解。计算多重积分时，按照从右至左的次序计算积分。

积分函数	说　明
Integrate [f, {x, a, b}**]**	计算定积分 $\int_a^b f(x)dx$ 的准确解
NIntegrate [f, {x, a, b}**]**	计算定积分 $\int_a^b f(x)dx$ 的数值解
Integrate [f, {x, a, b},{y, c, d}**]**	计算定积分 $\int_a^b dx\int_c^d f(x,y)dy$ 的准确解
NIntegrate [f, {x, a, b},{y, c, d}**]**	计算定积分 $\int_a^b dx\int_c^d f(x,y)dy$ 的数值解

表 3 - 3

例如：计算定积分 $\int_0^1(\cos^2 x+\sin^3 x)dx$

```
Integrate[Cos[x]^2+Sin[x]^3,{x,0,1}]
N[Integrate[Cos[x]^2+Sin[x]^3,{x,0.,1}],20]
N[Integrate[Cos[x]^2+Sin[x]^3,{x,0,1}],20]
NIntegrate[Cos[x]^2+Sin[x]^3, {x, 0,1}]
```

计算定积分 $\int_b^a dx\int_0^x(x+y)dy$

```
Integrate[2x+2y,{x,0,a},{y,0,b}]
```

3.4 幂级数

3.4.1 幂级数展开

幂级数展开函数 Series 的一般形式：

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

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

```
Series[Sin[2x],{x,0,6}]
Series[f[x],{x,0,3}]
```

在 x =∞ 处展开

```
Series[Exp[1/x],{x,Infinity,3}]
```

对有分数指数的函数展开。

```
Series[Exp[Sqrt[x]],{x,0,3}]
```

Series 在处理多元函数幂级数时，同 Integrate 和 Sum 等一样，从最后一个变量到第一个变量逐个展开。

```
Series[Cos[x]Cos[y],{x,0,3},{y,0,3}]
```

用求和函数 Sum 也能算出一些幂级数,例如:

```
Sum[x^n/n!,{n,0,Infinity}]
```

e^x

有些幂级数只能用特殊函数表示.

```
Sum[x^n/(n!^2),{n,0,Infinity}]
Sum[x^k/k!,{k,0,n}]
```

3.4.2 幂级数的运算

✧ 幂级数的简单运算

Mathematica 能对幂级数进行多种运算，当一个普通表达式和一个幂级数进行运算时，*Mathematica* 尽可能将表达式的各项并入幂级数中。*Mathematica* 保留幂级数的次数与初始幂级数的次数基本一致，正如数学运算中保留实数精度的方式。函数 Normal [幂级数]，去掉截断误差项 O[x]^n，将幂级数转化为一般表达式。例如：

```
t = Series[Log[x+1],{x,0,4}]
%^2
 D[%,x]
Normal[%]
SeriesCoefficient[%,2]    (*给出 2 次项系数 *)
```

✧ 幂级数的复合和反演

对于已经展开的幂级数，如果要继续对它的每一项再做幂级数展开，称为幂级数复合。用“ 幂级数 1 /. x –> 幂级数 2 ”可完成幂级数复合运算。也可以直接对复合函数做幂级数展开。下面 In[3] 的工作量与 In[1]和 In[2] 的工作等价。

设 y(x) = $1+x+\frac{x^2}{2}+\frac{x^3}{6}+\frac{x^4}{24}+O[x]^5$，这时 y 是 x 的函数 。如何计算反函数 x(y) 的表达式 ？请用函数 InverseSeries [幂级数, y]。

下列幂级数的复合和反演函数的一般形式和实例：

幂级数 1 **/. x –>** 幂级数 2　　幂级数复合

InverseSeries [幂级数, y**]**　　以变量 y 表示反演幂级数

InverseSeries [幂级数**]**　　用原变量表示反演幂级数

例题：

```
t=Series[Sin[x],{x,0,5}]
%/. x -> Series[Sin[x],{x,0,5}]
Series[Sin[Sin[x]],{x,0,5}]
InverseSeries[t]
% /. x -> t
InverseSeries[t,y]
```

3.4.3 残数计算

Limit [expr, x -> x0]计算当 x 趋于 x0 表达式 expr 的极限值。当极限值为无穷大时，则计算 expr 在 x 在 x0 处的残数（留数）。残数（residue）是表达式 expr 在 x0 处幂级数展开中 $(x-x0)^{-1}$ 的系数。

Residue 的一般形式：

Residue [expr, {x, x0}**]**　　计算表达式 expr 在 x0 处的残数

例，分别计算 $\frac{\cos x}{x^3}$ 在 x = 0 的残数和 x 趋于 0 的极限；

```
In[1]:={Residue[Cos[x]/x^3,{x,0}],Limit[Cos[x]/x^3,x -> 0]}
```

Out[1]= $\{-\frac{1}{2},\infty\}$

也可用幂级数展开计算残数。

```
In[2]:=Series[Cos[x]/x^3,{x,0,3}]
          (* 函数在解析点处残数为 0 *)
In[3]:={Residue[Sin[x]/x,{x,0}], Limit[Sin[x]/x, x -> 0]}
Out[3]={0,1}
```

3.5 微分方程

3.5.1 常微分方程

DSolve 解线性与非线性的常微分方程，以及联立常微分方程组。在没有给定方程的初值条件情况下，所得的解包括了待定的系数 C[1]，C[2]等。

用 DSolve 函数得到的是常微分方程的准确解。

NDSolve 求解常微分方程的数值解，这时要给出求解区间{x, xmin, xmax}。

一般形式：

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，纯函数部分请看第 7 章。

NDSolve [eqns, y[x], {x, xmin, xmax}**]**

在区间{xmin, xmax}上求解变量是 x 的常微分方程或联立常微分方程组 eqns。

例如：解微分方程 y ' (x)= ay(x) 。

```
In[1]:= DSolve[y'[x] = =a y[x],y[x],x]
```

Out[1]= $\{\{y[x] -> E^{ax} C[1]\}\}$

解微分方程 y ' (x)= a y (x) ， 边界条件：y(0)=1。

```
ss=DSolve[{y'[x] = =a y[x],y[0]= =1},y[x],x]
Plot[y[x]/.ss,{x,-20,20}
```

解联立常微分方程组：$\begin{cases} x(t) = -y'(t) \\ y(t) = -x'(t) \end{cases}$

```
DSolve[{x[t]==-y'[t],y[t]==-x'[t]},{x[t],y[t]},t]
```

当用 DSolve 求 y[x]时，y[x]以替换规则形式：y[x] –> expr ,在含有微分的表达式中不能替换 y ' [x] ；如果用 DSolve 求 y ， 直接可替换 y ' [x]等项。

```
DSolve[y'[x]== x+y[x],y[x],x]
DSolve[y'[x]== x+y[x],y,x]
y''[x]+y[x] /. %
DSolve[y''''[x] == y[x],y[x],x]
```

DSolve 解出 Bernoulli 方程,

```
In[7]:=DSolve[y'[x] - x y[x]^2 - y[x] == 0,y[x],x]
```

3.5.2 偏微分方程

DSolve 不仅能解单变量的常微分方程，也能解多个变量的偏微分方程。

DSolve [{eqn1,eqn2,…}, y [x1, x2,…], {x1,x2,…}**]** 计算偏微分方程组的准确解

DSolve [eqn, y [x1, x2,…], {x1,x2,…}**]** 解偏微分方程 y [x1, x2,…]

DSolve [eqn, y, {x1,x2,…}**]** 解偏微分方程的纯函数 y

NDSolve [{eqn1,eqn2,…}, y [x1, x2,…], {x1,x2,…}**]** 计算偏微分方程组的数值解

计算有两个独立变量的偏微分方程 $\frac{\partial y(x_1,x_2)}{\partial x_1}+\frac{\partial y(x_1,x_2)}{\partial x_2}=\frac{1}{x_1x_2}$

```
DSolve[D[y[x1,x2],x1]+D[y[x1,x2],x2] == 1/(x1 x2),
       y[x1,x2],{x1,x2}]
```

计算 $x_1\frac{\partial y(x_1,x_2)}{\partial x_1}+x_2\frac{\partial y(x_1,x_2)}{\partial x_2}=e^{x_1x_2}$

由于 y 和它的微分满足一个线性方程，所以它的通解是存在的。

```
DSolve[x1 D[y[x1, x2], x1] + x2 D[y[x1, x2], x2]
                == Exp[x1 x2], y[x1, x2], {x1, x2}]
```

下面的非线性偏微分方程存在一个通解。

```
DSolve[D[y[x1,x2],x1]+ D[y[x1,x2],x2]
        == Exp[y[x1,x2]],y[x1,x2],{x1,x2}]
```

3.5.3 矢量运算

在工程计算中，常需要对矢量作运算。矢量运算函数放在 Calculus 的子 VectorAnalysis 程序包中，运算前用运算符“**<<**”调入该程序包。表 3 - 4 列出部分矢量运算函数及其使用说明。

函 数	说 明
<<Calculus`VectorAnalysis`	载入向量运算包
SetCoordinates[system[names]]	设定坐标系（Cartesian,Cylindrical,Spherical 等）
Div[f]	计算矢量场 f 在缺省坐标系下的散度 $\nabla\cdot f$
Curl[f]	计算矢量场 f 在缺省坐标系下的旋度 $\nabla\times f$
Grad[f]	计算标量 f 在缺省坐标系下的梯度 ∇f
Laplacian[f]	计算标量 f 在缺省坐标系下的 $\nabla^2 f$
Biharmonic[f]	计算标量 f 在缺省坐标系下的 $\Delta^2 f$
Div[f,coordsys],Curl[f,coordsys],…	按给定的坐标系计算散度、旋度…

表 3 - 4

调入向量运算程序包，

```
In[1]:=<<Calculus`VectorAnalysis`
```

设定球面坐标系，

```
In[2]:= SetCoordinates[Spherical[r, theta, phi]]
Out[2]=Spherical[r, theta, phi]
```

计算梯度，

```
In[3]:=Grad[r^2 Sin[theta]]
Out[3]={2 r Sin[theta], r Cos[theta], 0}
```

计算散度，

```
In[4]:=Div[{x^2y, y, z}, Cartesian[x, y, z]]
Out[4]=2 + 2 x y
```

计算拉普拉斯算子。

```
In[5]:=Laplacian[x + y^3, Cartesian[x, y, z]]
```

3.6 积分变换

3.6.1 Laplace 变换

函数 *f(t)*的 Laplace 变换为 $\int_0^\infty f(t)e^{-st}dt$ 。函数 F(s)的 Laplace 反变换为 $\frac{1}{2\pi i}\int_{\gamma-i\infty}^{\gamma+i\infty}F(s)e^{st}ds$ 。

Laplace 的数学形式与 Fourier 变换非常相似，但变换结果差别很大。因为它对原函数 *f(t)*的要求相对比较弱，而对于某些问题 Laplace 变换比 Fourier 变换适应面更广。

下列 Laplace 变换函数常用形式：

LaplaceTransform[expr, t, s**]** expr 的 Laplace（拉普拉斯）变换

InverseLaplaceTransform[expr, t, s**]** expr 的 Laplace 反变换

例如对 $t^3\cos t$ 做 Laplace 变换，得到

```
In[1]:=LaplaceTransform[t^3 Cos[t],t,s]
```

再做 Laplace 反变换，则

```
In[2]:= InverseLaplaceTransform[%,s,t]
```

甚至一些简单的变换也常会包含特殊的函数。

```
In[3]:=LaplaceTransform[1/(1 + t^2),t,s]
```

Laplace 变换能将积分和微分运算转化为基本代数运算。

也可做高维 Laplace 变换或反变换。

LaplaceTransform [expr, {t1,t2,…}, {s1,s2,…}**]** expr 的高维 Laplace 变换

InverseLaplaceTransform [expr, {s1,s2,…}, {t1,t2,…}**]** expr 的高维 Laplace 反变换

3.6.2 Fourier 变换

在 *Mathematica* 中，函数 f(t)的 Fourier 变换为 $\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} f(t)e^{iwt}dt$，函数 F(t)的 Fourier 反变换为 $\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} F(w)e^{-iwt}dw$。

下列 Laplace 变换函数常用形式：

FourierTransform [expr, t, w**]** expr 的 Fourier 变换

InverseFourierTransform [expr, w, t**]** expr 的 Fourier 反变换

```
FourierTransform[1/(1 + t^4),t,ω]
InverseFourierTransform[%, ω, t]
```

在不同的应用领域对 Fourier 变换有不同的定义形式，用选项 FourierParameters 可以指定其中一种。

Fourier 变换形式	说 明
FourierSinTransform[expr, t, w]	Fourier 正弦变换
FourierCosTransform[expr, t, w]	Fourier 余弦变换
InverseFourierSinTransform[expr, w, t]	Fourier 正弦反变换
InverseFourierCosTransform[expr, w, t]	Fourier 余弦反变换

表 3 - 5

3.6.3 Z 变换

函数 *f(n)*的 Z 变换为的 $\sum_{n=0}^{\infty} f(n)z^{-n}$。Z 变换后常得到复变量 Z 的函数 ，因此称为 Z 变换。它是一种有效的离散的拉普拉斯形式。主要应用数字信号处理、控制论和解微分方程等离散系统分析中。可以认为它产生某种类似于组合数学和数论中常用的生成函数。

F(z)的逆 Z 变换为 $\frac{1}{2\pi i}\oint F(z)z^{n-1}dz$。

常用形式：

ZTransform[expr,n,z**]** expr 的 Z 变换

InverseZTransform[expr,z,n**]** expr 的逆 Z 变换

例题：

```
ZTransform[1/n!,n,z]
InverseZTransform[%,z,n]
```