第六章 符号计算

6.1 符号对象的创建

6.1.1 符号对象的生成

符号计算工具箱定义了一种新的 MATLAB 数据类型，叫做符号对象。在 MATLAB 内部，符号对象的数据存储格式是符号字符串。sym 函数用于创建符号对象，包括符号常量、符号变量和符号表达式。

【调用格式】

f = sym(arg)	把数值、字符串或者表达式 arg 转换为符号对象 f
f = sym(argn, flagn)	把数值或者数值表达式 argn 转换为 flagn 格式的符号对象
f = sym('argv', flagv)	把字符串'argv'按照 flagv 的格式转换为符号对象
syms argv1 argv2 ...	定义 argv1、argv2 等为符号对象
syms arg1 arg2 ... flagv	定义 arg1、arg2 等为 flagv 格式的符号对象

【说明】

1. 对于数值或者数值表达式 argn，flagn 可以取以下值：
   'd' 最接近的十进制浮点精度表示
   'e' 带估计误差的有理表示
   'f' 十六进制浮点数表示
   'r' 最接近的有理表示，MATLAB 的缺省表示方法
2. 对于字符串变量名 argv，flagv 可以取以下“限定”项：
   'positive' “正实数”符号变量
   'real' “实数”符号变量
   'unreal' “非实数”符号变量
3. syms 是 sym 函数的简化书写方式，各符号对象之间只能用空格分开。

6.1.2 符号常量

用 sym 函数可以定义符号常量对象，包括符号标量对象和符号常量数组对象，定义符号常量对象的同时也可以指定数值常量的表示方法。

例 6.1.1 符号常量的定义

```
>>a=[1/3, sqrt(5), pi+sqrt(2)];                %定义数值数组
>>s1=sym([1/3, sqrt(5), pi+sqrt(2)], 'd');     %用十进制方式表示符号常量
>>s2=sym([1/3, sqrt(5), pi+sqrt(2)]);          %用最接近的有理方式表示符号常量
>>s3=sym('[1/3, sqrt(5), pi+sqrt(2)]');        %绝对准确的符号数值表示，输入为字符串
```

6.1.3 符号变量与符号表达式

1. 定义符号变量和符号表达式

例 6.1.2 符号变量和符号表达式的定义

```
>> x = sym('x', 'real');        %定义实数符号变量 x
```

```
>> sym y real;                %定义实数符号变量 y
>> z=x+i*y;                   %定义符号表达式对象 z
>> conj(z)                    %符号变量求共轭复转置
>>f = z*conj(z);              %符号表达式对象
>> f=simple(f)                %符号表达式对象化简
f =
  x^2+y^2
```

例 6.1.3 符号变量与符号矩阵

```
>> syms  a b c;                          %定义符号变量 a，b，c
>> A=[a,b,c; b,c,a; c,a,b ];              %定义符号矩阵 A
>> sum(A(:,1));                          %求矩阵 A 第一列的元素的和
>> sum(A(1,:))==sum(A(:,2))              %符号对象的关系运算
>> det(A);                               %矩阵求行列式
>> syms  alpha  beta;                    %定义符号变量 alpha 和 beta
>> A(1,3)=beta;                          %矩阵元素赋值
>> A=subs(A,a,alpha)                     %符号表达式的替换操作，矩阵 A 中的 a 用 alpha 代替
```

2. 符号表达式中自变量的确定

在数学表达式或者数学函数中，一般都含有自变量。为了便于进行数学运算，通常要显示指定表达式中的自变量，如果不指定自变量，MATLAB 会根据上下文关系，识别表达式中默认的自变量（独立自由的符号变量）。识别表达式中自变量的基本原则是：按照字母表中靠近小写字母 x 的顺序识别，最靠近字母 x 的变量被第一个识别为自变量。MATLAB 还提供了自变量识别函数 findsym。

【调用格式】

findsym(exp)　　识别表达式 exp 中所有的自由符号变量

findsym(exp, n)　　识别表达式 exp 中最靠近 x 的 n 个独立自由变量

【说明】

1. 表达式可以是符号矩阵，此时变量识别是对整个矩阵进行的。
2. 函数识别的是“独立的”“自由的”符号变量，符号常量或者非独立的符号变量无法被识别。
3. 识别次序是按照靠近 x 的远近进行的，区分大小写，总认为大写字母距离 x 的距离大于所有小写字母。

例 6.1.4 符号表达式中自变量的识别

```
>> syms a b x X Y;                       %定义符号变量
>> k=sym(2.5);
>> z=sym('c*sqrt(alpha)+y*sin(beta)');   %定义符号表达式对象（变量）z
>> exp=a*z*X+k*Y+b^x                     %定义符号表达式变量 exp
exp =
a*(c*alpha^(1/2)+y*sin(beta))*X+5/2*Y+b^x
??? >> k=sym(2.5);                       %定义符号常量 k，无法被识别
>> findsym(exp)                          %自动识别所有自由独立变量，k 为常量，z 为非独立
>>findsym(exp,3)                         %识别 exp 中前 3 个靠近 x 的独立自由变量
```

6.1.4 符号数学函数

在 MATLAB 中，可以定义表示数学函数的符号对象，既能建立具有详细运算关系的函数，又能建立抽象数学函数。定义符号数学函数有 2 种方法：

1. 用符号表达式
2. 用函数 M 文件（在函数 M 文件中用符号变量作为输入变量）

例 6.1.5 用符号表达式定义符号数学函数

```
>> syms x y z                    %定义函数自变量
>> r = sqrt(x^2 + y^2 + z^2);    %定义函数 r
>> t = atan(y/x);                %定义函数 t
>> f = sin(x*y)/(x*y);           %定义函数 f
```

例 6.1.6 用函数 M 文件来定义符号数学函数，下列代码定义了函数 $\mathrm{sinc}(x)=\frac{\sin(x)}{x}$。

```
function z = sinc(x)
%SINC 符号函数
%       sin(x)/x   数学计算公式
%       接受符号变量作为输入变量
if isequal(x,sym(0))          %如果自变量值为符号 0，则函数值为 1
    z = 1;
else
    z = sin(x)/x;
end
```

例 6.1.7 建立抽象的符号数学函数

```
>> f = sym('f(x)')               %建立抽象函数 f(x)
>> syms x h;
>> df = (subs(f, x, x+h) -f)/h   %建立抽象函数 df，表示 f(x)的导数表达式
df =
 (f(x+h)-f(x))/h
>> g=subs(df, 'f', 'sin')        %建立 sin(x)的导数函数的定义
 g =
 ((sin)(x+h)-(sin)(x))/h
>> limit(g,h,0)                  %根据导数定义，求 sin(x)的导数
 ans =
 cos(x)
```

6.1.5 符号数值计算的精度

数值计算受计算机字长的限制，每次数值计算都会产生截断误差。在 MATLAB 中，数值计算的精度大约为 16 位数字。对于符号计算来说，只要能获得解析结果，其计算结果是绝对准确的，不包含任何误差。但是当将符号数值对象转换成数值数据时，就会产生误差，涉及到转换精度问题。MATLAB 的符号计算工具箱提供 3 种算术运算：

1. 数值运算（16 位数字精度）
2. 符号运算（绝对准确）
3. 任意精度运算（用户指定运算精度）

有关符号运算精度的函数及其调用方法如下：

【调用格式】

n = digits 获取当前采用的数值计算精度

digits(n) 设置数值计算精度的有效位为 n，除非再次设定，否则始终有效

xs = vpa(x) 在当前精度下，给出变量 x 的数值符号结果 xs

xs = vpa(x,n) 在 n 位精度下，给出变量 x 的数值符号结果 xs

【说明】

1. 相对精度位数 n 表示有效数字位数。

2. x 可以是符号常量，也可以是数值对象。

3. 返回值 xs 是符号结果，一定是符号对象。

例 6.1.8 符号数值计算精度

```
>> z = 1.0e-16
>> x = 1.0e+2
>> digits(14)                %设置数值运算精度为 14 位
>> y = vpa(x*z+1)            %在 14 位精度下计算 y
```

6.1.6 符号对象与其他数据类型之间的转换

数值、字符和符号是 MATLAB 中 3 种不同的数据类型。MATLAB 提供了不同数据类型之间的转换指令。表 6.1.1 介绍了 MATLAB 中实现符号对象到数值、字符串之间的转换指令。

表 6.1.1 符号对象和其他数据类型之间的转换指令表

函数名称	功能	函数名称	功能
char	将符号对象转换为字符串	uint32	将符号对象转换为 32 位无符号整数
double	将符号对象转换为双精度数值	uint64	将符号对象转换为 64 位无符号整数
int8	将符号对象转换为 8 位整数	single	将符号对象转换为单精度数值
int16	将符号对象转换为 16 位整数	sym2poly	将符号多项式转换为数值系数向量
int32	将符号对象转换为 32 位整数	poly2sym	多项式系数向量转换为符号多项式
int64	将符号对象转换为 64 位整数	vpa	转换为符号运算结果
uint8	将符号对象转换为 8 位无符号整数	sym	转为为符号对象
uint16	将符号对象转换为 16 位无符号整数	pretty	转换为易读的显示方式

例 6.1.9 多项式和符号对象之间的转换

```
>> sym x;
>> f = x^4-3*x^2+3-5*x          %定义符号多项式对象
>> p1=sym2poly(f)               %将符号多项式对象转换为数值系数向量
p1 =
 1     0    -3    -5     3
>> p1s=poly2str(p1, 'y');       %将多项式数值系数向量转换为多项式字符串格式
>> f2=poly2sym(p1);             %将多项式数值系数向量转换为多项式符号对象
>> pretty(f2, 's');             %将多项式符号对象显示为易读格式
```

6.2 符号对象的代数运算

6.2.1 符号对象的运算

1.符号对象的基本代数运算

符号对象的基本代数运算和普通数值变量一样，可以使用算术运算符、关系运算符（仅能用==和！=），其运算符的定义和数值运算相同。

例 6.2.1 符号矩阵的基本代数运算

```
>>syms t;
>>G = [cos(t), sin(t); -sin(t), cos(t)];
>>A = G*G;                    %符号矩阵的乘法
>>A = simple(A);              %化简为最简表达式
```

```
>>I = G.' *G;                          %符号数组乘法
>>I = simple(I);                       %化简为最简表达式
I =
[ 1, 0]
[ 0, 1]
>>F= [cos(t), -sin(t); sin(t), cos(t)];
>>(F+G)/2;                             %符号矩阵加法和符号矩阵与标量的除法
```

2.符号对象的函数运算

数值计算使用的函数基本上也可以用于符号计算，包括三角函数（atan2 除外）、指数函数、对数函数（log2、log10 除外）、复数函数（angle 除外）、线性代数函数和矩阵函数。

例 6.2.2 符号矩阵的函数运算

```
>>H = hilb(3) ;          %生成 3×3 的希尔伯特数值矩阵
>>H = sym(H);            %将数值矩阵转为符号常数矩阵
>>inv(H) ;               %符号矩阵求逆
>>det(H)                 %符号矩阵求行列式的值
ans =
 1/2160
>>syms s
>>H(1,1) = s;            %将符号元素赋值为符号变量 s，使矩阵变为非奇异矩阵
>>Z = det(H)             %带有符号变量的符号矩阵求行列式的值
Z =
 1/240*s-1/270
>>sol = solve(Z)         %求行列式的根
sol =
 8/9
>>H = subs(H,s,sol);     %将矩阵行列式的根代入符号矩阵，使矩阵变为奇异矩阵
>>det(H) ;               %奇异矩阵的行列式值
>>inv(H) ;               %奇异矩阵求逆
>> eig(H);               %符号矩阵的特征值
```

6.2.2 符号表达式分解、展开与化简

MATLAB 提供了符号表达式的因式分解、展开和化简函数。

【调用格式】

collect(expr, v)	合并符号表达式 expr 中符号对象 v 的同类项系数
expand(expr)	对表达式 expr 进行多项式、三角函数、指数对数等函数展开
factor(expr)	对符号表达式 expr 做因式分解
horner(expr)	把多项式 expr 分解为嵌套形式
[n,d]=numden(expr)	提取表达式最小分母公因子 d 和相应的分子多项式 n
simplify(expr)	用多种恒定变换对表达式 expr 进行综合化简
simple(expr)	把 expr 化简为最简表达式

【说明】上述表达式 expr 可以是符号矩阵，此时函数对符号矩阵的每个元素进行相应操作。

例 6.2.3 符号表达式的展开和分解

```
>>f=sym ('(x^3-6*x^2+10*x-5)+(x-1)');     %定义原始的符号表达式
>> fc=collect(f);                         %符号表达式合并同类项
>> ff=factor(f);                          %符号表达式因式分解
```

>> **fh=horner(f);** %符号表达式的嵌套分解

>> **fe=expand(fh);** %符号表达式的展开

>> **factor(1025);** %正整数的质数分解，1025＝5×5×41

例 6.2.4 写出矩阵 $\begin{bmatrix} \frac{1}{4} & \frac{1}{2x-1}+\frac{1}{x+1} \\ \frac{2}{x^2-1} & 3x+2 \end{bmatrix}$ 各元素的分子多项式和分母多项式

>> **syms x;**

>> **A=[1/4, 1/(2*x-1)+1/(x+1); 2/(x^2-1), 3*x+2];**

>> **[n,d]=numden(A);** %提取表达式最小分母公因子 d 和相应的分子多项式 n

>> **pretty(simplify(n./d));**

例 6.2.5 化简 $f(x)=\sqrt[3]{\frac{1}{x^3}+\frac{3}{x^2}+\frac{3}{x}+1}$

解：用 simplify 进行多次化简也得不到最简结果，用 simple 化简可以得到最简结果

>> **syms x;**

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

>> **sfy1=simplify(f);**

>> **sfy2=simplify(sfy1);**

>> **sp1=simple(f);**

>> **sp2=simple(sp1)**

sp2 =

1+1/x

6.2.3 符号表达式的置换操作

MATLAB 提供了子表达式的置换函数。通常在这几种情况下使用子表达式的置换函数：第一，符号计算结果中多次出现同一个表达式，为了阅读方便，可以把这个表达式用符号变量来置换；第二，可以用符号常量对象置换表达式中的自变量，实现表达式求值；第三，通过置换某些表达式可以生成新的表达式。

1.自动置换函数

【调用格式】

[RS,vn]=subexpr(S,vn) 用符号变量 vn 置换 S 中的子表达式，并重写 S 为 RS。

[RS,vn]=subexpr(S, 'vn')

例 6.2.6 对一元三次方程 $ax^3+bx^2+cx+d=0$ 的符号解进行子表达式的置换。

>> **t = solve('a*x^3+b*x^2+c*x+d = 0');**

>> **[r,s] = subexpr(t,'s')**

2.通用置换函数

【调用格式】

RS=subs(S, old, new) 用 new 置换 S 中 old，生成 RS

【说明】

(1) old 是被替换的子表达式，可以是符号变量，也可以是字符串表达式。

(2) new 是替换 old 的值，可以是符号常量、符号变量、符号表达式，也可以是数值。

(3) 如果要替换多个子表达式，则 old 和 new 为细胞数组。

例 6.2.7 通用置换函数。

>> **syms a x;**

>> **f=a*cos(x)+1;**

```
>> f1=subs(f, 'cos(x)', sym('y'));          %用符号变量替换
>> f2=subs(f, {a, x}, {3, sym('pi/4')})     %用符号常量替换，
>> f3=subs(f, {a, x}, {3, pi/4});           %用双精度数替换
>> f4=subs(f, {a, x}, {0:3, 0:pi/3:pi});    %用双精度数组替换
>> f5=subs(f, x , sym('exp(-t)'))           %用符号表达式替换
 f5 =
 a*cos(exp(-t))+1
```

6.2.4 符号函数的反函数

【调用格式】

g = finverse(f, v)　　求指定自变量为 v 的函数 f(v)的反函数 g(v)

g = finverse(f)　　求函数 f 对缺省的自变量（由 findsym 确定）的反函数 g

例 6.2.8 求 $f(x)=\sqrt{x^2+1}$ 的反函数。

```
>> syms x;   f=(x^2+1)^(1/2);
>> g=finverse(f);
```

6.2.5 符号函数的复合函数

【调用格式】

fg = compose(f, g, x, y, z)　　对 $f(x)$ 和 $g(y)$ 求复合函数 $fg(z)=f(g(y))\big|_{y=z}$

fg = compose(f,g)　　对 $f(\cdot)$ 和 $g(\cdot)$ 求复合函数 $fg=f(g(\cdot))$，自变量由 findsym 确定

例 6.2.9 求复合函数。

```
>> syms x y z t u;
>> f = 1/(1 + x^2); g = sin(y);   h = x^t;   p = exp(-y/u);
>> compose(f,g);
>> compose(f, g, t);
>> compose(h, g ,x, z)
ans =
sin(z)^t
>> compose(h, g, t, z);
>> compose(h, p, x, y, z);
>> compose(h, p, t, u, z)
ans =
x^exp(-y/z)
```

6.3 符号微积分

6.3.1 符号微分和雅可比矩阵

求导数、高阶导数、偏导数是常见的数学运算，MATLAB 提供这方面的符号微分函数。

【调用格式】

df= diff(f, v, n) 求 $\frac{d^n f(v)}{dv^n}$，即求函数 $f(v)$ 对 v 的 n 阶导数

R = jacobian(f,v) 求多元向量函数 $f(v)$ 的雅可比矩阵，即 $\left[\frac{\partial f_i(v)}{\partial v_j}\right]_{m\times n}$

例 6.3.1 $f(a,t,x)=\begin{bmatrix} a & t^2 \\ t\sin(x) & \log(x) \end{bmatrix}$，求 $\frac{d}{dx}f$、$\frac{d^2}{dt^2}f$ 和 $\frac{d^2}{dxdt}f$

```
>> syms a t x;
>> f=[a, t^2; t*sin(x), log(x)];
>> dfx=diff(f,x)                    %矩阵 f 对 x 的一阶导数，也可写为 diff(f)
dfx =
[          0,              0]
[ t*cos(x),          1/x]
>> df2t=diff(f,t,2);                %矩阵 f 对 t 的二阶导数
>> dfxt=diff(diff(f,x),t)
dfxt =
[         0,          0]
[ cos(x),          0]
```

例 6.3.2 $f(x,y,z)=\begin{bmatrix} xyz \\ y \\ x+z \end{bmatrix}$，求其雅可比矩阵

```
>> syms x y z;
>> f = [x*y*z; y; x+z];
>> v=[x,y,z];
>> R = jacobian(f,v)
R =
[ y*z,   x*z,   x*y]
[    0,     1,      0]
[    1,     0,      1]
```

6.3.2 函数极限

函数的极限是通过导数来定义的，limit 函数用于求函数极限。

【调用格式】

limit(f, x, a) 求 $\lim_{x\to a} f(x)$

limit(f, x, a, 'right') 求 $\lim_{x\to a^+} f(x)$

limit(f, x, a, 'left') 求 $\lim_{x\to a^-} f(x)$

例 6.3.3 求极限运算。

```
>> limit(sin(x)/x,x,0);
>> limit(1/x,x,0,'right');
>> limit(1/x,x,0,'left');
>> limit((sin(x+h)-sin(x))/h,h,0)          %sin(x)导函数的定义
ans =
```

cos(x)

>> **v = [(1 + a/x)^x, exp(-x)];**

>> **limit(v,x,inf)**

ans =

[exp(a), 0]

6.3.3 符号积分

int 函数用于求定积分、不定积分和多重积分的符号解。

【调用格式】

S= int(f, v) 求不定积分 $\int f(v)dv$，符号计算结果不带积分常数

S= int(f, v, a, b) 求定积分 $\int_a^b f(v)dv$

rsums(f, ra, rb) 用 Riemann 和求符号函数 $f(x)$ 在[ra,rb]区间上的近似积分

【说明】

1. 当 f 为矩阵时，将对矩阵的每个元素做积分运算。
2. v 缺省时，积分对 findsym 确认的变量进行。
3. 积分上下限 a 和 b 可以是任何数值和表达式
4. rsums 函数绘制求积分的曲线，根据用户的选择来确定最终的近似积分值

例 6.3.4 计算 $\int \begin{bmatrix} e^t & t^4 \\ 1 & \sin(at) \end{bmatrix} dt$

>> **syms a t;**

>> **f=[exp(t), t^4; 1, sin(a*t)];**

>> **int(f,t)**

ans =

[exp(t), 1/5*t^5]

[t, -1/a*cos(a*t)]

例 6.3.5 求多重积分 $\int_1^2 \int_{\sqrt{x}}^{x^2} \int_{\sqrt{xy}}^{xy} (x^2+y^2+z^2)dzdydx$

>> **syms x y z;**

>> **V=int(int(int(x^2+y^2+z^2, z, sqrt(x*y), x*y), y, sqrt(x), x^2), x, 1, 2)**

V =

-6072064/348075*2^(1/2)+14912/4641*2^(1/4)+64/225*2^(3/4)+51643291/795600

>> **vpa(V)**

ans =

44.540012164287862574338076759024

6.3.4 符号序列求和

symsum 函数实现符号序列的求和运算。

【调用格式】

s= symsum(f, v, a, b)　　求 $\sum_{v=a}^{b} f(v)$，a 和 b 为整数，b 可以取无穷大

例 6.3.6 计算 $\sum_{t=0}^{t}\begin{bmatrix} t & k^2 \end{bmatrix}$ 和 $\sum_{k=0}^{\infty}\frac{x^k}{k!}$

```
>> syms t k x;
>> f1=[t,k^2];
>> f2=x^k/sym('k!');
>> simple(symsum(f1, t, 0, t))
ans =[ 1/2*t*(t+1),     (t+1)*k^2]
>> simple(symsum(f2 ,k, 0, inf))
 simplify:
 exp(x)
```

6.4 符号方程求解

6.4.1 符号代数方程组的解

代数方程包括线性、非线性和超越方程等，solve 函数用于符号代数方程求解。

【调用格式】

g = solve('eq1', 'eq2' ,..., 'eqn', 'var1', 'var2' ,..., 'varn')

g = solve(exp1, exp2 ,..., expn, var1, var2 ,..., varn)

【说明】

1. 'eq1', 'eq2' ,..., 'eqn'是字符串表达式的方程，或者是字符串表达式，如果它们是不含等号的字符串表达式，则相当于方程'eqi＝0'
2. 'var1', 'var2' ,..., 'varn'是用字符串表示的方程中的变量名
3. exp1, exp2 ,..., expn 只能是符号表达式，不能是带有等号的表达式方程。
4. var1, var2 ,..., varn 只能是符号变量。
5. 返回值 g 是一个结构体数据，方程组的解用 g.var1，g.var2，... ，g.varn 表示。
6. 在无法求得解析解的时候，给出数值解。

例 6.4.1 方程组 $\begin{cases} uy+vz^2+w=0 \\ y+z+v=0 \end{cases}$，分别求其关于 y,z 和关于 u,v,w 的解

```
>> s1=solve('u*y+v*z^2+w=0','y+z+v=0','y','z');   %方程组关于 y,z 的解，从结果来看方程有 2 组解
s1 =
    y: [2x1 sym]
    z: [2x1 sym]
>> y=s1.y；z=s1.z;
%方程组关于 u,v,w 的解，2 个方程 3 个变量，方程有无穷多组解，其中 w 为自由变量
>> s2=solve('u*y+v*z^2+w=0','y+z+v=0','u','v','w');
>> u=s2.u；  v=s2.v；w=s2.w;
```

例 6.4.2 求方程组 $\begin{cases} xy^2+z^2=0 \\ y-z=1 \\ x^2-5x+6=0 \end{cases}$ 的解

解：【写法 1】

>> **s=solve('x*y^2 +z^2', 'y – z=1', 'x^2 - 5*x + 6');**

>> **s.x;s.y;s.z**

【写法 2】

>> **syms x y z;**

>> **s=solve(x*y^2+z^2, y-z-1, x^2-5*x+6);**

>> **s=solve(x*y^2+z^2=0, y-z=1, x^2-5*x+6=0);** %错误写法，不能含有符号方程

6.4.2 符号微分方程

【调用格式】

g = dsolve('eq1', 'eq2' ,..., 'eqn', 'cond1', 'cond2', ... , 'condn', 'v')

【说明】

1. 'eq1', 'eq2' ,..., 'eqn'是微分方程，只能用字符串形式。微分方程是函数必须的输入变量。
2. 'cond1', 'cond2' ,..., 'condn'是初始条件，也只能用字符串形式。
3. 'v'定义了微分方程的独立变量名（微分方程解中的自变量名），只能用字符串形式。默认的独立变量名为 t。
4. 微分方程字符串中，Dny 表示 y 的 n 阶导数，Dy 表示 y 的一阶导数。
5. 返回值 g 是一个结构体变量，要引用其成员才能得到方程的解。

例 6.4.3 求微分方程组 $\begin{cases} \dfrac{dx}{dt}=y \\ \dfrac{dy}{dt}=-x \end{cases}$ 的解。

>> **s=dsolve('Dx=y,Dy=-x');**

>> **x=s.x**

x =

-C1*cos(t)+C2*sin(t)

>> **y=s.y**

y =

C1*sin(t)+C2*cos(t)

例 6.5.4 求微分方程 $xy''-3y'=x^2$ 的解 $y(x)$，初始条件为 $y(0)=0, y(1)=1$

>> **s=dsolve('x*D2y-3*Dy=x^2', 'y(0)=0, y(1)=1', 'x')**

s =

4/3*x^4-1/3*x^3

例 6.5.5 求微分方程 $y''+y^2=1$ 的解，初始条件为 $y(0)=0$

>> **y = dsolve('(Dy)^2 + y^2 = 1','y(0) = 0')**

y =

sin(t)

-sin(t)

6.5 积分变换

MATLAB 的符号计算支持积分变换，对傅氏变换、拉氏变换和 Z 变换都提供了相应的符号计算函数。

6.5.1 傅立叶变换及其反变换

【调用格式】

Fw = fourier(ft, t, w) 求时域函数 ft 的傅氏变换 Fw

ft = ifourier(Fw, w, t) 求频域函数 Fw 的傅立叶反变换 ft

【说明】：ft 是以时间 t 为自变量的时域函数；Fw 是以角频率 w 为自变量的频域函数；输入参数 t 和 w 可以省略。

例 6.5.1 求函数 $f(t)=e^{-x^2}$ 的傅氏变换 $F(w)$

```
>> syms t;
>> ft= exp(-t^2);
>> Fw=fourier(ft)
Fw =
    pi^(1/2)*exp(-1/4*w^2)
```

例 6.5.2 求函数 $f(t)=\begin{cases} e^{-(t-x)} & t \geq x \\ 0 & t \geq x \end{cases}$ 的傅氏变换 $F(w)$，其中 x 是参数。

```
>> syms t w x;   ft=exp(-(t-x))*sym('heaviside(t-x) ');
>> Fw= simple(fourier(ft, t, w))      %fourier(ft) 和 fourier(ft, t)都会产生歧义
Fw =
1/(1+i*w)/exp(i*x*w)
```

【说明】MATLAB 的 Maple 提供了一些特殊函数，这些函数不是 MATLAB 的函数，因此不能在 MATLAB 中直接调用，必须用函数 syms 生成特殊函数的对象才能被 MATLAB 的符号计算所识别。

heaviside(t-a) Maple 的单位阶跃函数 $u(t-a)$

dirac(t-a) Maple 的单位脉冲函数 $\delta(t-a)$

k! Maple 的 k 的阶乘函数

sym('heaviside(t)') 将 Maple 的 heaviside(t)函数变为符号函数对象

6.5.2 拉普拉斯变换及其反变换

【调用格式】

Fs = laplace(ft, t, s) 求时域函数 ft 的拉氏变换 Fw

ft = ilaplace (Fs, s, t) 求频域 Fs 的拉氏反变换 ft

【说明】：ft 是以时间 t 为自变量的时域函数；Fs 是以复频率 s 为自变量的频域函数；输入参数 t 和 s 可以省略。

例 6.5.3 求函数 $f(t)=e^{-5t}\sin(3t)$ 的拉氏变换 $F(s)$。

```
>> syms t ;
>> ft=exp(-5*t)*sin(3*t);
>> Fs=laplace(ft)
```

Fs =

1/3/(1/9*(s+5)^2+1)

例 6.5.4 控制系统的闭环传递函数为

$$G(s)=\frac{4}{s^2+2s+4}$$

求初始条件为零时系统的单位阶跃响应。

```
>> cs=sym('4/(s^2+2*s+4)')*laplace(sym('Heaviside(t)'));
>> ft=ilaplace(cs)
ft =
1-exp(-t)*cos(3^(1/2)*t)-1/3*3^(1/2)*exp(-t)*sin(3^(1/2)*t)
```

6.5.3 Z 变换及其反变换

【调用格式】

Fz = ztrans(fn, n, z) 求时域序列 fn 的 Z 变换 Fz

Fn = iztrans (Fz, z, n) 求频域序列 Fz 的 Z 反变换 fn

例 6.5.5 某针对单位速度信号设计的最小拍控制系统的闭环脉冲传递函数为 $G(z)=2z^{-1}-z^{-2}$，求系统的单位速度响应，并观察其最小拍的拍数。

```
>> cz=sym('2/z-1/z^2')*ztrans(sym('t'))        %系统的单位速度响应的 Z 变换表达式
cz =
(2/z-1/z^2)*z/(z-1)^2
>> fn=iztrans(cz)                              %单位速度响应序列表达式
fn =
-charfcn[1](n)+n
>> subs(fn,'n',0:10)                           %求 0 到 10 个采样周期内的响应值
ans =
     0     0     2     3     4     5     6     7     8     9    10
```

从 10 个采样周期内的响应值可以看到：系统的单位速度响应从第 2 个采样周期后就完全跟随上了输入信号，因此跟踪拍数为 2。

6.6 Maple 的应用

6.6.1 经典特殊函数的调用

自从 MathWorks 公司购买了 Waterloo Maple 公司的符号计算工具 Maple 的使用权后，在 MATLAB 平台上集成了符号计算工具箱（Symbolic Math Toolbox），利用 MAPLE 引擎提供了对符号计算的良好支持。

Maple 定义了许多工程上常用的“经典特殊函数”，可以通过 mfun 函数计算这些经典特殊函数的值。经典特殊函数的使用步骤如下：

1.获取所需要的经典特殊函数名

运行 doc mfunlist 即可获得经典特殊函数 HTML 格式的列表，表中详细的说明了各个函数的函数名以及它们的数学定义，我们可以选择所需的函数。也可以在命令窗口运行 mfunlist 命令来获取经典特殊函数的文本列表。

2.通过 mfun 函数计算经典特殊函数的值

【调用格式】

Y = mfun('fun',v1,v2,v3,v4)　以输入变量 v1，v2，v3，v4 调用经典特殊函数 fun

【说明】：函数名 fun 只能用字符串表示，输入变量可以是函数 fun 能够接收的合法数据。

例 6.6.1 计算 $\int_0^2 \frac{\sinh(t)}{t} dt$

解：通过 doc mfunlist 可知 Maple 用 $Shi(z) = \int_0^z \frac{\sinh(t)}{t} dt$ 函数表示上述积分形式

```
>> r = mfun('Shi',2)
r =
2.5016
>> int('sinh(t)/t',0,2)          %int 函数得到一个用经典特殊函数构成的表达式
ans =
Shi(2)
```

6.6.2 Maple 函数的调用

可以用 maple 函数调用 Maple 提供的各种符号计算函数。

【调用格式】

R = maple(MapleStatement)　运行 Maple 格式的语句 MapleStatement

R = maple(fun, v1, v2, ...)　运行以 v1 等为输入变量的 Maple 中的 fun 函数

【说明】：MapleStatement 必须是复合 Maple 格式的语句，其格式和 MATLAB 语句有区别；返回值 R 是字符串，不是符号对象。

例 6.7.2 求递推方程 $f(n) = f(n-1) + n$ 的通解 $f(k)$

解：Maple 中用 rsolve 来求递推方程的解，可以用 2 种方式调用 rsolve

```
>> Fk = maple('rsolve(f(n)=f(n-1)+n, f(k));')      %第一种调用格式
>> Fk = maple('rsolve' , 'f(n)=f(n-1)+n', 'f(k)')   %第二种调用格式
Fk =
f(0)-k-1+(k+1)*(1/2*k+1)
>> Fks=simple(sym(Fk))
Fks =
f(0)+1/2*k+1/2*k^2
```

6.6.3 Maple 工具的帮助系统

MATLAB 集成了符号计算工具 Maple。Maple 提供了大量的函数和指令用以实现符号计算，通过 Maple 的帮助系统获取这些函数的使用方法是非常重要的。Maple 帮助系统是通过 mhelp 命令引出的，可以进行 3 类帮助信息的搜索。

1. Maple 帮助索引

【调用格式】 **mhelp index**

【说明】可以调出 Maple 帮助的分类目录，在命令窗口中显示 Maple 帮助的总目录，里面列出有哪些具体的分类帮助条目。

2. Maple 分类帮助

【调用格式】 **mhelp index[*category*]**

【说明】获取 category 分类中的帮助信息条目。*category* 是 mhelp index 命令列出的分类名称，通常包括 function、expression、misc、packages、procedure、statement。例如：mhelp index[function]会得到 Maple 的函数列表。

3. Maple 具体函数帮助

【调用格式】

mhelp fun_name 获取名字为 fun_name 的 Maple 函数的帮助信息

小结

符号运算具有强大的功能，其特点是：

(1) 运算不受到计算误差累计的限制；

(2) 可以列出完全封闭的解析解；

(3) 计算指令简单，表达式符合数学上的描述方法；

(4) 计算速度慢。

符号运算对控制系统建模、系统分析与计算具有极为重要的意义，为复杂运算和公式推导提供有力的工具。通过本章的学习应该掌握如何创建、修改符号对象，符号运算功能和 maple 函数调用。