

# 符号数学工具 (Symbolic Math Toolbox)

## 一、符号矩阵和运算

### 1. 符号矩阵的四则运算

Matlab 把符号矩阵的四则运算简化为与数值矩阵完全相同的运算方式，其运算符为：加 (+)、减 (-)、乘 (×)、除 (/、\ ) 等或：符号矩阵的和 (symadd)、差 (symsub)、乘 (symmul)。

例 1-58 >>A=sym('[1/x, 1/(x+1);1/(x+2), 1/(x+3)]')  
>>B=sym('[x, 1; x+2, 0]')  
>>C=B-A  
>>D=A\B %相当于  $A^{-1}*B$ , 非方阵时为伪逆 区别 B/A 是  $B*A^{-1}$

则显示:

C=  
x-1/x 1-1/(x+1)  
x+2-1/(x+2) -1/(x+3)  
D=  
-6\*x-2\*x^3-7\*x^2 1/2\*x^3+x+3/2\*x^2  
6+2\*x^3+10\*x^2+14\*x -2\*x^2-3/2\*x-1/2\*x^3

### 2. 其他基本运算

符号矩阵的其他一些基本运算包括转置 (')、行列式 (det)、逆 (inv)、秩 (rank)、幂 (^) 和指数 (exp 和 expm) 等都与数值矩阵相同

### 3. 将数值矩阵转化为符号矩阵

函数 **sym**

格式 B=sym(A) %将 A 转化为符号矩阵 B

例 1-59

>> A=[2/3,sqrt(2),0.222;1.4,1/0.23,log(3)]  
A =  
0.6667 1.4142 0.2220  
1.4000 4.3478 1.0986  
>> B=sym(A)  
B =  
[ 2/3, sqrt(2), 111/500]  
[ 7/5, 100/23, 4947709893870346\*2^(-52)]

### 4. 符号矩阵的简化

符号工具箱中提供了符号矩阵因式分解、展开、合并、简化及通分等符号操作函数。

#### (1) 因式分解

函数 **factor**

格式 factor(s) %符号表达式 s 的因式分解函数

说明 S 为符号矩阵或符号表达式，常用于多项式的因式分解。

例 1-61 将  $x^9-1$  分解因式

在 Matlab 命令窗口键入:

```
syms x  
factor(x^9-1)
```

则显示: ans =

$(x-1)*(x^2+x+1)*(x^6+x^3+1)$

例 1-62 问 “ $\lambda$ ” 取何值时，齐次方程组 
$$\begin{cases} (1-\lambda)x_1 - 2x_2 + 4x_3 = 0 \\ 2x_1 + (3-\lambda)x_2 + x_3 = 0 \\ x_1 + x_2 + (1-\lambda)x_3 = 0 \end{cases}$$
 有非 0 解？

解：系数矩阵的行列式值为 0，因此在 Matlab 编辑器中建立 M 文件：

```
syms k
A=[1-k -2 4;2 3-k 1;1 1 1-k];
D=det(A)
factor(D)
```

其结果显示如下：

```
D =
-6*k+5*k^2-k^3
ans =
-k*(k-2)*(-3+k)
```

从而得到：当  $k=0$ 、 $k=2$  或  $k=3$  时，原方程组有非 0 解。

(2) 符号矩阵的展开

函数 **expand**

格式：expand(s) %符号表达式 s 的展开函数

说明：s 为符号矩阵或表达式。常用在多项式的因式分解中，也常用于三角函数，指数函数和对数函数的展开中。

例 1-63 将  $(x+1)^3$ 、 $\sin(x+y)$  展开

在 Matlab 编辑器中建立 M 文件：

```
syms x y
p=expand((x+1)^3)
q=expand(sin(x+y))
```

则结果显示为

```
p =
x^3+3*x^2+3*x+1
q =
sin(x)*cos(y)+cos(x)*sin(y)
```

(3) 同类式合并

函数 **Collect**

格式 Collect(s,v) %将 s 中的变量 v 的同幂项系数合并

Collect(s) % s 是矩阵或表达式，此命令对由命令 findsym 函数返回的默认变量进行同类项合并。

(4) 符号简化

函数 **simple** 或 **simplify** %寻找符号矩阵或符号表达式的最简型

格式 simple(s) % s 是矩阵或表达式

[R,how]=simple(s) %R 为返回的最简形，how 为简化过程中使用的主要方法。

说明 Simple(s)将表达式 s 的长度化到最短。若还想让表达式更加精美，可使用函数 Pretty。

格式 Pretty(s) %使表达式 s 更加精美

例 1-64 计算行列式 
$$\begin{vmatrix} 1 & 1 & 1 & 1 \\ a & b & c & d \\ a^2 & b^2 & c^2 & d^2 \\ a^4 & b^4 & c^4 & d^4 \end{vmatrix}$$
 的值。

在 Matlab 编辑器中建立 M 文件：

```
syms a b c d
A=[1 1 1 1;a b c d;a^2 b^2 c^2 d^2;a^4 b^4 c^4 d^4];
d1=det(A)
d2=simple(d1) %化简表达式 d1
```

```
pretty(d2)      % 让表达式 d2 符合人们的书写习惯
则显示结果如下:
d1 =
    b*c^2*d^4-b*d^2*c^4-b^2*c*d^4+b^2*d*c^4+b^4*c*d^2-b^4*d*c^2-a*c^2*d^4+a*d^2*c^4+a*b
^2*d^4-a*b^2*c^4-a*b^4*d^2+a*b^4*c^2+a^2*c*d^4-a^2*d*c^4-a^2*b*d^4+a^2*b*c^4+a^2*b^4*d-
a^2*b^4*c-a^4*c*d^2+a^4*d*c^2+a^4*b*d^2-a^4*b*c^2-a^4*b^2*d+a^4*b^2*c
d2 =
    (-d+c)*(b-d)*(b-c)*(-d+a)*(a-c)*(a-b)*(a+c+d+b)
    (-d+c)(b-d)(b-c)(-d+a)(a-c)(a-b)(a+c+d+b)
```

## 二、符号表达式

- 符号表达式是代表数字、函数、算子和变量的MATLAB字符串，或字符串数组。不要要求变量有预先确定的值，符号方程式是含有等号的符号表达式。如表2.1所列。

表2.1

符号表达式	在MATLAB中的表示
$\frac{1}{2x^n}$	'1/(2*x^n)'
$y = \sqrt{2x}$	y='1/sqrt(2*x)'
$\cos(x^2) - \sin(2x)$	'cos(x^2)-sin(2*x)'
$M = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$	M=sym('[a, b; c, d]')
$\int_a^b \frac{x^3}{\sqrt{1-x}} dx$	f=int('x^3/sqrt(1-x)','a','b')

注意：加单引号”表示是一个符号表达式而不是数字表达式。

- 符号函数：让用户操作上述符号表达式的方法和函数，比如：

```
>> f=sym('x^2')
f =
x^2
>> M=sym(' [a, b; c, d] ')      %   create a symbolic matrix M
M=
[a, b]
[c, d]
```

注意：1.建立符号数组时，最好用上函数sym

2.许多符号函数可以自动将字符转变为符号表达式。例如diff cos(x) 也可以，

```
>> diff x^2+3*x+5 % the argument is equivalent to 'x^2+3*x+5'
```

- 符号变量是符号表达式中的变量，如：'x+3\*y+z'中的x,y,z。

```
>> symvar('a*x+y')      %返回自变量
ans =
```

'a'  
'x'  
'y'

### 三、符号表达式运算（目的：更方便的构造符号表达式）

标准代数运算：

例如：给定两个函数

$$f = 2x^2 + 3x - 5 \quad g = x^2 - x + 7$$

```
>> f= sym('2*x^2+3*x-5')    %  define the symbolic expression
f =
2*x^2+3*x-5
>> g= sym('x^2-x+7')
g =
x^2-x+7
>> a=f+g
a =
3*x^2+2*x+2
>> b=f-g
b =
x^2+4*x-12
>> c=f.*g
c =
(2*x^2+3*x-5)*(x^2-x+7)
>> d=f./g
d =
(2*x^2+3*x-5)/(x^2-x+7)
```

高级运算：

- compose: 把 $f(x)$ 和 $g(x)$ 复合成 $f(g(x))$ ;
  - finverse: 求函数的逆函数;
  - symsum: 求表达式的序列和。
- compose  
给定表达式

```
f = 1/(1+x^2)    g = sin(x)    h = 1/(1+u^2)    k = sin(v)
>> syms x y u v
>> f = 1/(1+x^2); g = sin(y); h = 1/(1+u^2); k = sin(v);
%  create the four expression
>> compose(f,g)  %  find an expression for f(g(x))
ans =
```

```

1/(1+sin(y)^2)
>> compose(g,f) % find an expression for g(f(x))
ans=
sin(1/(1+x^2))

```

### ● finverse

表达式譬如 $f(x)$ 的函数逆 $g(x)$ ，满足 $g(f(x))=x$ 。例如， $e^x$ 的函数逆是 $\ln(x)$ ，因为 $\ln(e^x)=x$ 。 $\sin(x)$ 的函数逆是 $\arcsin(x)$ ，函数 $\frac{1}{\tan(x)}$ 的函数逆是 $\arcsin(\frac{1}{x})$ 。函数`finverse`返回表达式的函数逆。如果解不是唯一就给出警告。

```

>> syms x a b c d z
>> finverse(1/x) % the inverse of 1/x is 1/x since '1/(1/x)=x'
ans=
1/x

```

### ● symsum

求表达式的序列和有四种形式：

`symsum(f)`返回  $\sum_0^{x-1} f(x)$ ；

`symsum(f,'s')`返回  $\sum_0^{s-1} f(s)$ ，

`symsun(f,a,b)`返回  $\sum_a^b f(x)$ ；

`symsun(f,'s',a,b)`返回  $\sum_a^b f(s)$ 。

例子：  $\sum_1^{\infty} \frac{1}{(2n-1)^2}$ ，返回应是  $\frac{\pi^2}{8}$ 。

```

>> syms n
>> symsum(1/(2*n-1)^2,1,inf)
ans=
1/8*pi^2

```

## 求符号表达式的函数值

为了得到数字，需要使用函数`eval`来转换字符串。

```

>> syms x
>> f=symsum(x^2)
>> x=2
>> eval(f)
ans =
1.0000

```

## 四、微积分（Calculus）函数

微分和积分广泛地用在许多工程学科，这部分的主要函数有：

**diff(): Differentiate.**（微分）

**int(): Integrate**（积分）

**jacobian(): Jacobian matrix**（雅可比行列式）

**limit(): Limit of an expression.**（极限）

**symsum(): Summation of series**（序列和）

**taylor(): Taylor series expansion**（泰勒级数）

### 微分和差分:diff

```
>> f='a*x^3+x^2-b*x-c' % define a symbolic expression
f=
a*x^3+x^2-b*x-c
>> diff(f) % differentiate with respect to the default variable x
ans=
3*a*x^2+2*x-b
>> diff(f,'a') % differentiate with respect to a
ans=
x^3
>> diff(f,2) % differentiate twice with respect to x
ans=
6*a*x+2
>> diff(f,'a',2) % differentiate twice with respect to a
ans=
0
>> m=[(1:8).^2] % create a vector
M=
1 4 9 16 25 36 49 64
>> diff(M) % find the differences between elements
ans=
3 5 7 9 11 13 15
```

如果diff的表达式或可变参量是数值，MATLAB就非常巧妙地计算其数值差分；如果参量是符号字符串或变量，MATLAB就对其表达式进行微分。

### 积分int:

格式: **int(f)**，其中f是一符号表达式，它力图求出另一符号表达式F使**diff(F)=f**。

注意：积分或逆求导不一定是以封闭形式存在，或存在但软件也许找不到，或者软件可明显地求解，但超过内存或时间限制。当MATLAB不能找到逆导数时，它将返回未经计算的命令。

```
>> int('log(x)/exp(x^2)') % attempt to integrate
```

```
ans=
```

```
log(x)/exp(x^2)
```

积分函数的多种形式:

$\left\{ \begin{array}{l} \text{int(f): 相对于缺省的独立变量x求逆导数} \\ \text{int(f,'s'): 相对于符号变量s积分} \\ \text{int(f,a,b)和int(f, 's', a, b): a, b是数值, 求解符号表达式从a到b的定积分} \\ \text{int(f,'m','n')和形式int(f, 's', 'm', 'n'): 其中m, n是符号变量, 求解符号表达式从} \end{array} \right.$

m到n的定积分。

```
>> f='sin(s+2*x)' % crate a symbolic function
```

```
f=
```

```
sin(s+2*x)
```

```
>> int(f) % integrate with respect to x
```

```
ans=
```

```
-1/2*cos(s+2*x)
```

```
>> int(f, 's') % integrate with respect to s
```

```
ans=
```

```
-cos(s+2*x)
```

```
>> int(f, pi/2, pi) % integrate with respect to x from pi/2 to pi
```

```
ans=
```

```
-cos(s)
```

```
>> int(f, 's', pi/2, pi) % integrate with respect to s from pi/2 to pi
```

```
ans=
```

```
2*cos(x)^2-1-2*sin(x)*cos(x)
```

```
>> int(f, 'm', 'n') % integrate with respect to x from m to n
```

```
ans=
```

```
-1/2*cos(s+2*n)+1/2*cos(s+2*m)
```

**diff** 和 **int** 均可以对符号数组的每一个元素进行运算。

```
>> F=sym( '[a*x,b*x^2;c*x^3,d*s]' ) % create a symbolic array
```

```
F =
```

```
[ a*x, b*x^2]
```

```
[ c*x^3, d*s]
```

```
>> int(F) % ubtegrate the array elements with respect to x
```

```
ans=
```

```
[1/2*a*x^2, 1/3*b*x^3]
```

```
[1/4*c*x^4, d*s*x]
```

## 五、方程求解

求解单个变量的代数方程: 利用 **solve** 函数

1.没有=号的表达式, 用**solve**将其置成等于0。

```
>> solve('a*x^2+b*x+c') % solve for the roots of the quadratic equation
```

```
ans=
[1/2/a*(-b+(b^2-4*a*c)^1/2)]
[1/2/a*(-b-(b^2-4*a*c)^1/2)]
```

注：方程有2个解。

2.对其他非缺省变量求解，在 solve 中指定出该变量。

```
>> solve('a*x^2+b*x+c','b') % solve for b
ans=
-(a*x^2+c)/x
```

3.对带有等号的方程求解。

```
>> f=solve('cos(x)=sin(x)') % solve for x
f=
1/4*pi
>> t=solve('tan(2*x)=sin(x)')
t =
[
0]
[
pi]
[
atan(1/2*(-2*3^(1/2))^(1/2),1/2+1/2*3^(1/2))]
[
atan(-1/2*(-2*3^(1/2))^(1/2),1/2+1/2*3^(1/2))]
[
atan(1/2*2^(1/2)*3^(1/4)/(1/2-1/2*3^(1/2)))+pi]
[
-atan(1/2*2^(1/2)*3^(1/4)/(1/2-1/2*3^(1/2)))-pi]
```

单个常微分方程求解：利用 dsolve 函数

dsolve 函数的句法：用字母D来表示求微分，D2，D3等等表示二阶、三阶微分，并以此来设定方程。任何D后所跟的字母为因变量。方程  $\frac{d^2 y}{dx^2}=0$  用符号表达式 D2y=0 来表示。独立变量可以指定或由 symvar 规则选定为缺省。

例1，求一阶方程  $dy/dx=1+y^2$  的解，其通解为：

```
>> dsolve('Dy=1+y^2') % find the general solution
ans=
tan(t+C1) % C1是积分常数
```

给定初值  $y(0)=1$  的方程的解：

```
>> dsolve('Dy=1+y^2','y(0)=1') % add an initial condition
y=
tan(t+1/4*pi)
```

指定独立变量：

```
>> dsolve('Dy=1+y^2','y(0)=1','v') % find solution to dy/dv
ans=
tan(v+1/4*pi)
```

例2：求二阶微分方程的解，方程有两个初始条件：

$$\frac{d^2 y}{dx^2} = \cos(2x) - y \quad \frac{dy}{dx}(0)=0 \quad y(0)=1$$



```
>> y=dsolve('D2y=cos(2*x)-y','Dy(0)=0','y(0)=1','x')
y=
(1/6*cos(3*x)-1/2*cos(x))*cos(x)+(1/2*sin(x)+1/6*sin(3*x))*sin(x)+4/3*cos(x)

>> y=simple(y) % y looks like it can be simplified
y=
4/3*cos(x)-1/3*cos(2*x)
```

例3：求解下列形式的微分方程

$$\frac{d^2 y}{dx^2} - 2 \frac{dy}{dx} - 3y = 0$$

通解为： `>> y=dsolve('D2y-2*Dy-3*y=0','x')`  
`y =`

$$C1 \cdot \exp(-x) + C2 \cdot \exp(3x)$$

加上初始条件：  $y(0)=0$  和  $y(1)=1$  可得到：

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

$$1/(\exp(-1)-\exp(3)) \cdot \exp(-x) - 1/(\exp(-1)-\exp(3)) \cdot \exp(3x)$$

```
>> y=simple(y) % this looks like a candidate for simplification
y=
```

$$(\exp(-x)-\exp(3x))/(\exp(-1)-\exp(3))$$

```
>> pretty(y) % pretty it up
```

$$\frac{\exp(-x)-\exp(3x)}{\exp(3)-\exp(-1)}$$