



Matlab符号计算

第8讲



MATLAB的符号运算

- **MATLAB**: 世界顶级的数值计算能力
- **MATLAB**强大的**符号运算**能力来自:

MATLAB

+ **Symbolic Math Toolbox**

+ **Maple**

基本符号工具箱、扩展符号工具箱



1、符号运算基础

■ 符号对象的定义

- MATLAB用来定义符号对象的指令有两个：**sym**和**syms**。其常用格式如下：

- **f=sym (arg)**

将**数字、字符串或表达式**arg转换成符号对象f。

- **syms('arg1', 'arg2', ...)**

将**字符**arg1, arg2, ...定义为基本符号对象

- **syms arg1, arg2, ...**

其作用同上。



符号对象的定义

■ 举例符号对象的定义

```
>>syms('x', 'y')  % or syms x y
```

```
% define symbol objects x and y
```

```
>>whos x y
```

Name	Size	Bytes	Class
x	1x1	126	sym object
y	1x1	126	sym object

Grand total is 4 elements using 252 bytes

```
>>f=x^2 - y^2;  % f sym object ?
```



符号对象的定义（续）

```
>> whos x y f
```

Name	Size	Bytes	Class
f	1x1	138	sym object
x	1x1	126	sym object
y	1x1	126	sym object

Grand total is 12 elements using 390 bytes

Note: 由符号对象表达式构成的仍然是一个符号对象。



符号对象的定义（续）

例：用sym指令将一个表达式定义为一个符号对象

```
clear
```

```
f_e=sym('a*x^2+b*x+c');
```

```
whos
```

Name	Size	Bytes	Class
f_e	1x1	146	sym object

Grand total is 12 elements using 146 bytes

Note: 在符号运算中，倘若事先没有对表达式的独立变量符号进行定义，那么MATLAB会自动检测哪些字符是符号函数，哪些是符号变量。



基本符号运算和操作

- 符号表达式与通常的算术式一样，可进行四则运算。

举例：符号表达式的四则运算

```
syms x y a b
```

```
f1=sin(x)+cos(y);
```

```
f2=a-b;
```

```
f12=f1+f2
```

```
f12=
```

```
sin(x)+cos(y)+a-b
```

```
f1*f2
```

```
ans =
```

```
(sin(x)+cos(y))*(a-b)
```

基本符号运算和操作（续）

■ 合并同类项，指令：**collect(s), collect(s,v)**

举例：

```
clear
```

```
syms x y
```

```
f = x^2*y+y*x-x^2-2*x;
```

```
collect(f)
```

```
ans =
```

```
(y-1)*x^2+(y-2)*x
```

```
collect(f,'y')
```

```
ans=
```

```
(x^2+x)*y-x^2-2*x
```




基本符号运算和操作（续）

■ **因式分解**，指令：**factor**

举例：

```
clear
```

```
syms x y
```

```
factor(x^2-y^2)
```

```
ans =
```

```
(x+y)*(x-y)
```

```
factor(x^3-y^3)
```

```
ans =
```

```
(x-y)*(x^2+y*x+y^2)
```

```
factor(x^4-y^4)
```

```
ans =
```

```
(x-y)*(x+y)*(x^2+y^2)
```



基本符号运算和操作（续）

■ **化简**指令：simplify和simple

举例：simplify的使用

clear

syms x y

simplify(cos(x)^2+sin(x)^2)

ans =

1

simplify(cos(x)^2-sin(x)^2)

ans =

cos(2*x)



基本符号运算和操作（续）

举例：simple化简符号表达式

```
>> simple(cos(x)^2-sin(x)^2)
```

simplify:

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

radsimp:

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

combine(trig):

```
cos(2*x)
```

factor:

```
-(sin(x)-cos(x))*(sin(x)+cos(x))
```

expand:

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

combine:

```
cos(2*x)
```

...



基本符号运算和操作（续）

convert(exp):

$(1/2*\exp(i*x)+1/2/\exp(i*x))^2+1/4*(\exp(i*x)-1/\exp(i*x))^2$

convert(sincos):

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

convert(tan):

$(1-\tan(1/2*x)^2)^2/(1+\tan(1/2*x)^2)^2-4*\tan(1/2*x)^2/(1+\tan(1/2*x)^2)^2$

collect(x):

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

mwcossin:

$1-2*\sin(x)^2$

ans =

cos(2*x) %最后的答案



基本符号运算和操作（续）

- Simple的2种调用格式（三角函数公式验证）

```
[r, how]=simple(cos(x)^2-sin(x)^2)
```

```
r =
```

```
cos(2*x)
```

```
how =
```

```
combine(trig)
```

```
clear
```

```
y=sym( '2*sin(x)*cos(x)' )
```

```
%把字符表达式转换为符号变量
```

```
y =
```

```
2*sin(x)*cos(x) %如何化简?
```

```
y=simple(y)
```

```
y =
```

```
sin(2*x)
```

基本符号运算和操作（续）

验证三角等式

$$\sin \varphi_1 \cos \varphi_2 - \cos \varphi_1 \sin \varphi_2 = \sin(\varphi_1 - \varphi_2)$$

```
syms fai1 fai2;
```

```
f=simple(sin(fai1)*cos(fai2)-cos(fai1)*sin(fai2))
```

```
f =
```

```
sin(fai1-fai2)
```

■ 简单符号表达式操作指令还有(help可获得帮助):

expand(展开)、**horner**(嵌套形式)、**numden**(找到分子和分母)



例：

```
[n,d] = numden(sym(4/5))
```

```
n =
```

```
4
```

```
d =
```

```
5
```

```
syms x y
```

```
[n,d] = numden(x/y + y/x)
```

```
n = x^2+y^2 , d = y*x
```

符号矩阵的创建与运算

■ 使用sym指令创建符号数组

举例：

```
a1=sym('[1/3 1/2 sqrt(2)]');
```

or

```
b1= '[1/3 1/2 sqrt(2)] ';
```

```
a2=sym(b1);
```

```
a1, a2
```

```
a1 =
```

```
[ 1/3, 1/2, sqrt(2)]
```

```
a2 =
```

```
[ 1/3, 1/2, sqrt(2)]
```

```
b1 class ?
```

```
>> whos a1 a2 b1
```

Name	Size	Bytes	Class
a1	1x3	270	sym object
a2	1x3	270	sym object
b1	1x20	40	char array

符号矩阵的创建与运算（续）

举例：创建符号常数数组

```
>> a=reshape(1:9, 3, 3)
```

```
a =
```

```
    1    4    7
```

```
    2    5    8
```

```
    3    6    9
```

```
>> s_a=sym(a)
```

```
s_a =
```

```
[ 1, 4, 7]
```

```
[ 2, 5, 8]
```

```
[ 3, 6, 9]
```

```
>> s_b=1./s_a
```

```
s_b =
```

```
[ 1, 1/4, 1/7]
```

```
[ 1/2, 1/5, 1/8]
```

```
[ 1/3, 1/6, 1/9]
```

```
>> whos a s_a s_b % class?
```

Name	Size	Bytes	class
a	3x3	72	double array
s_a	3x3	622	sym object
s_b	3x3	654	sym object



符号矩阵的创建与运算（续）

举例：创建一般符号数组

```
>> clear
>> s_b1=sym('[x^2 x*y 1/x 1/y]')
s_b1 =
[ x^2, x*y, 1/x, 1/y]
>> s_b2=sym('[y^2 x*y 1/x 1/y]')
s_b2 =
[ y^2, x*y, 1/x, 1/y]
>> s_c=[s_b1;s_b2]    %使用方括号2-D数组
s_c =
[ x^2, x*y, 1/x, 1/y]
[ y^2, x*y, 1/x, 1/y]
```

符号矩阵的创建与运算（续）

```
>>s_c*6      %乘以普通标量
ans =
[ 6*x^2, 6*x*y, 6/x, 6/y]
[ 6*y^2, 6*x*y, 6/x, 6/y]
>>syms x
>>s_c*x      %乘以符号标量
ans =
[ x^3, x^2*y, 1, x/y]
[ x*y^2, x^2*y, 1, x/y]
>> s_c.*s_c  %符号数组的点乘
ans =
[ x^4, x^2*y^2, 1/x^2, 1/y^2]
[ y^4, x^2*y^2, 1/x^2, 1/y^2]
Note: s_c.*s_c等同于s_c.^2
```



■ 小结

- 符号对象的定义: `sym`和`syms`
 - 符号常数
 - 符号对象表达式
- 符号表达式运算与操作
 - 四则运算
 - 操作指令: `collect`、`factor`、`simple`等
- 符号数组的创建与运算
 - 使用`sym`指令
 - 符号数组运算

2、符号微积分

- 求极限（极限微积分的基础和出发点）
- 符号极限指令：**limit**
 - `limit(f,x,a)` % 计算自变量 $x \rightarrow a$ 时，函数 f 的极限值.
 - `limit(f,a)` % 计算由 `findsym(f)`确定的自变量 $\rightarrow a$ 时，
函数 f 的极限值.
 - `limit(f)` % 计算 $a = 0$ 时，函数 f 的极限值.
 - `limit(f,x,a,'right')` or `limit(f,x,a,'left')`
 - 计算 $x \rightarrow a$ 时的右极限和左极限
 - **Note:**函数 f ，在MATLAB中是关于符号对象 x 的符号表达式。



符号极限 (续)

■ 演示求极限

```
syms x y t h
```

```
f=sin(x)/x;
```

```
limit(f)
```

```
ans = 1
```

```
limit((1+2*t/x)^(3*x), x, inf)
```

```
ans = exp(6*t)
```

```
limit(1/x, x, 0, 'right')
```

```
ans = Inf
```

```
limit(1/x, x, 0, 'left')
```

```
ans = -Inf
```

```
limit((sin(x+h)-sin(x))/h, h, 0)
```

```
% sin(x)导数的概念
```

```
ans = cos(x)
```

函数的符号微分与求导

■ MATLAB符号微分指令: **diff**

■ 举例:

■ 定义一个函数

```
>> syms x y
```

```
>> f=x^2+x*y+y^2+sin(x)+cos(y);
```

■ %对x一阶导数 ?

```
>> dx1=diff(f)
```

```
dx1 = 2*x+y+cos(x)
```

■ 对x二阶导数 ?

```
>> dx2=diff(f, 2) dx2 = 2-sin(x)
```

■ 对y一阶导数 ?

```
>> dy1=diff(f, y)
```

```
dy1 = x+2*y-sin(y)
```

■ 对y二阶导数 ?

```
>> dy2=diff(f, y, 2)
```

```
dy2 = 2-cos(y)
```

Note: about “diff”

在不指定自变量的情况下，系统对x自变量求导；如函数（符号表达式）中没有x，系统将对最接近x的符号变量求导。



函数的符号微分与求导（续）

- MATLAB能否计算函数的3阶、4阶或高阶导数？
- The answer is “Yes”
 - 计算函数f的3阶导数
`diff(f,3)` or `diff(f, 'y', 3)`
 - 计算函数f的4阶导数
`diff(f,4)` or `diff(f, 'y', 4)`
 - 计算函数f的n阶导数 ($n=1,2,3,\dots$)
`diff(f,n)` or `diff(f, 'y', n)`



MATLAB如何进行函数的符号积分？

- 符号积分指令：**int**---计算不定积分和定积分

- 求解符号不定积分

- 计算 $1/x$ 、 $1/x^2$ 、 $\sin(w*t)$ 的不定积分？

```
>> syms x w t
```

```
>> int(1/x)
```

```
ans =
```

```
log(x)
```

```
>> int(1/x^2)
```

```
ans =
```

```
-1/x
```

```
>> int(sin(w*t), t) % 对t积分
```

```
ans =
```

```
-1/w*cos(w*t)
```

函数的符号积分（续）

■ 求解符号定积分？

验证积分 $\int_{-\tau/2}^{\tau/2} A e^{-i\omega t} dt = A\tau \cdot \frac{\sin \frac{\omega\tau}{2}}{\frac{\omega\tau}{2}}$

```
>> clear
```

```
syms a t tao w;
```

```
>> yf=int(a*exp(-i*w*t), t, -tao/2, tao/2)
```

```
yf=-i*a*(-exp(-1/2*i*w*tao)+exp(1/2*i*w*tao))/w
```

```
>> Yf=simple(yf)
```

```
Yf = 2*a*sin(1/2*w*tao)/w
```



3、符号代数方程的求解

- 代数方程的特点

- 含变量的等式

- 代数运算：加、减、乘、除、乘方、开方

- MATLAB符号方法求解代数方程的指令：

- 线性方程组的求解：**linsolve**

- 一般代数方程组的解：**solve**

这里指举例说明solve的使用

符号代数方程的求解（续）

- 举例：求方程组 $u*y^2+v*z+w=0$, $y+z+w=0$ 关于 y, z 的解

```
>> S=solve('u*y^2+v*z+w=0','y+z+w=0','y','z')
```

```
S =
```

```
  y: [2x1 sym]
```

```
  z: [2x1 sym]
```

```
>> disp('S.y'),disp(S.y)
```

```
S.y
```

```
-1/2/u*(-2*u*w-v+(4*u*w*v+v^2-4*u*w)^(1/2))-w
```

```
-1/2/u*(-2*u*w-v-(4*u*w*v+v^2-4*u*w)^(1/2))-w
```

```
>> disp('S.z'),disp(S.z)
```

```
S.z
```

```
1/2/u*(-2*u*w-v+(4*u*w*v+v^2-4*u*w)^(1/2))
```

```
1/2/u*(-2*u*w-v-(4*u*w*v+v^2-4*u*w)^(1/2))
```



符号代数方程的求解（续）

■ 求 $(x+2)^2=2$ 的解

```
>> clear all
```

```
>> s=solve('(x+2)^x=2','x')
```

```
s =
```

```
.69829942170241042826920133106081
```

```
>> s=solve('(y+2)^y=2','y')
```

```
s =
```

```
.69829942170241042826920133106081
```

```
>> s=solve('(y+2)^a=2','a')
```

```
s =
```

```
log(2)/log(y + 2)
```