第 12 讲 符号计算基础与符号微积分

（第 9 章　MATLAB 符号计算）

目的：

一、掌握符号对象申明方法。

二、掌握符号表达式的运算法则。

三、掌握求符号函数极限和导数的方法。

四、掌握求符号函数不定积分和定积分的方法。

除了数值运算外，matlab 还提供了**符号运算工具箱**以便进行符号运算，**为了和一般数值运算加以区分，在使用符号运算前需要将变量或数值对象申明为符号对象。如果不加以申明，进行符号运算时会报错。**

一、符号对象的申明：

1、使用 syms 申明**“符号变量”**

>>syms x y z；中间用空格隔开,可以在工作区看到 x,y,z 的类型是 sym 型

>>a=x+y;

>>b=x-y;　%由符号变量运算产生的 a,b 也是 sym 型

>>a*b

ans = (x + y)*(x - y)

2、使用 syms 申明**“符号函数”**（类型是 symsfun 型）

>>syms x y z;　　　或者　　　>>syms s(x,y,z);

>>s(x,y,z)=x+y+z;　　　　　　>>s(x,y,z)=x+y+z;

从工作区可以看到，s 的类型是 symsfun 型，定义好符号函数后可以用于求值：

>>s(1,2,3)

ans = 6

注：结果数字 6 的类型是 sym 型，符号数字和数值数字看起来一样，但在系统里不是同一种类型。

3、使用 syms 申明符号矩阵

```
>> syms a b;
>> A=[a b; 1 2]
A =
[a, b]
[1, 2]
```

注：用 syms 定义的符号对象默认是复数，为了得到实数符号对象，可以使用下面方式：

```
>> syms c d real;  % real 实数、positive 正数、integer 整数
>>B=[c,d;1,2]
B =
[c, d]
[1, 2]
```

以上两种方式定义的矩阵 A、B，一个是复矩阵，一个是实矩阵。

数学上复数矩阵求转置有两种结果，一种是普通转置

例如：$\begin{bmatrix}1+2i & 2-3i\\ i & 1+i\end{bmatrix}^T=\begin{bmatrix}1+2i & i\\ 2-3i & 1+i\end{bmatrix}$

另一种叫共轭转置（也叫埃尔米特共轭）

例如：$\begin{bmatrix}1+2i & 2-3i\\ i & 1+i\end{bmatrix}^T=\begin{bmatrix}1-2i & -i\\ 2+3i & 1-i\end{bmatrix}$

在 matlab 中用 A’ 表示求 A 的共轭转置，用 A.’表示求 A 的一般转置，如果 A 矩阵是实矩阵，那么 A’=A.’。所以，如果对上面定义的 A、B 求转置，结果是

```
>> A'
ans =
[conj(a), 1]
[conj(b), 2]
```

```
>>B'
ans=
[c, 1]
[d, 2]
```

其中 conj()是求元素的共轭。

```
>> A.'                              >>B.'
  ans =                              ans=
        [a,  1]                           [c,  1]
        [b,  2]                           [d,  2]
```

>> syms A [2,3] %注：A 与[2,3]间有空格，该命令生成下面的矩阵

```
>>A
A =
[A1_1, A1_2, A1_3]
[A2_1, A2_2, A2_3]
```

4、符号矩阵的运算

对符号矩阵，也可以进行诸如算术运算、求秩、求逆矩阵、行列式、关系运算、逻辑运算等等。

```
>>det(A)
ans=2*a-b
>> A^-1  %或者 inv(A)
ans =
[ 2/(2*a - b), -b/(2*a - b)]
[-1/(2*a - b),  a/(2*a - b)]
```

二、常用的符号函数和运算

1、symvar(s)：从符号表达式中找出所有符号变量。

```
>> syms s(x,y,z)
>> s(x,y,z)=x*y*z;
>> symvar(s)
ans =[x, y, z]
```

注：可以用 symvar(s,1)找出 s 的默认第一自变量。

例如：s=2*y+x+3; symvar(s,1)结果是 x，即优先 x，虽然表达式中 y 在前。

但如果是：s(y,x)=x^2+y+1;symvar(s,1)结果是 y，因为 s(y,x)中 y 在前。

2、poly2sym 与 sym2poly：多项式转符号和符号转多项式。

```
>>p=[1,2,3];
>>y=poly2sym(p)
y=x^2 + 2*x + 3
>> syms s(x)
>>s(x)=x^3+2*x+5;
>>q=sym2poly(s)
q = 1      0      2      5
```

3、factor(s)：因式分解。factor 也可以将整数分解为质因素的乘积，例如 factor（10），结果为 2，5。

```
>>p=[1,0,0,-1];
>>s=poly2sym(p);
>>y=factor(s)
y =
[x - 1, x^2 + x + 1]
```

4、expand(s):多项式展开。

例如 u=expand((x^2 + y^2)*(x + y)*(x - y))；结果是 u=x^4-y^4。

5、collect(s)：按 s 的默认第一自变量合并同类项。

collect(x^2*y + y*x - x^2 - 2*x);结果是(y-1)*x^2+(y-2)*x

注：这里 s 的默认第一自变量是 x，可以用 symvar(s,1)查看。

collect(s,expr)：按 s 指定的对象合并同类项。

collect(x^2*y + y*x - x^2-2*x,y);结果是(x^2+x)*y-x^2-2*x

collect(2*x*i - 3*i*y,i);结果是(2*x - 3*y)*1i

collect(2*x*pi+x*y*pi+1,pi)

结果是：(2*x + x*y)*pi + 1。

6、[n,d]=numden(s)：通分求分子分母。结果是对 s 通分后的分子 n 和分母 d。

```
>> s=1/(x+y)-1/(x-y);
>>[n,d]=numden(s)
n =-2*y
d =(x + y)*(x - y)
```

7、finverse(s):求反函数

```
>>s=2x+3;
>>finverse(s);
```

ans=x/2-3/2;%这里要特别注意，答案中的 x/2-3/2 中的 x 是以前的因变量 s。

例：$y=\ln\left(x+\sqrt{x^2+1}\right)$，反函数是 $x=\dfrac{e^y+e^{-y}}{2}=\dfrac{e^{-y}\left(e^{2y}+1\right)}{2}$

```
>>syms x
>>y=log(x+sqrt(x^2+1));
>>finverse(y)
ans =(exp(-x)*(exp(2*x) - 1))/2。
```

8、simplify：化简函数。simplify(expr)或者 simplify(expr,name,value)；

expr 如果是符号矩阵，则简化其每个元素。举例：

```
>>syms x;
>>expr=cos(x)^2-sin(x)^2;
>>s=simplify(expr)
s =cos(2*x)
```

注：simplify(expr,name,value)使用说明

simplify(expr,'all',true):显示所有等效表达式，默认是 false 只显示最后结果。

simplify(expr,'steps',n):显示化简到第 n 步的最后结果。

结合使用：simplify(expr,'all',true,'steps',n):显示化简路上所有等效结果（到第 n 步）

simplify(expr,'IgnoreAnalyticConstraints',true)：利用对数函数的性质合并有关对数函数的和差次幂等；但对含复数的表达式可能出错，所以如果表达式含复数就不要设置为 true，默认值是 false。

9、compose():求复合函数。

compose(f,g)等同于 f(g)：将 f 中的默认第一自变量替换为 g，举例

>> syms x y t; >> f(x,y)=x*2+3*y; >> g(t)=6*t; >> s=compose(f,g)

s(t, y) = 12*t + 3*y

>> syms x y t; >> f(x,y)=x*2+3*y; >> g(t)=6*t+y; >> s=compose(f,g)

s(t, y) =12*t + 5*y

注：可以用 symvar(f,1)查看 f 的默认第一自变量。

compose(f,g,x,z):将 f 中的自变量 x 替换为 g(z)，此时 g 中的自变量被替换成了 z。

compose(f,g,x,y,z):将 f 中的自变量 x 替换为 g，g 中的自变量 y 被替换为了 z。

10、limit()：求极限或者左右极限

求极限 $\lim\limits_{x\to a} s$：limit(s,x,a)

求左极限 $\lim\limits_{x\to a^-} s$：limit(s,x,a,'left')

求右极限 $\lim\limits_{x\to a^+} s$：limit(s,x,a,'right')

求 $\lim\limits_{x\to 1}\frac{xy}{x^2+y^2}$

>>syms x y;

>>s1=x*y/(x^2+y^2);

>>limit(s1,x,1);

ans =y/(y^2 + 1)

求 $\lim \frac{|x|}{x}$

```
>>s2=abs(x)/x;
>>limit(s2,x,0)
ans =NaN
>>limit(s2,x,0,'left')
ans = -1
>>limit(s2,x,0,'right')
ans = 1
```

求 $\lim_{x\to 1}\frac{\sin x-\sin 1}{x-1}$ $(=\cos 1)$

```
>>syms x;
>>f=(sin(x)-sin(1))/(x-1);
>>s=limit(f,x,1)
ans=Nan
```

结果是 Nan 而不是 cos1，这是由于 Matlab 计算极限时将 sin1 的值用高精度的近似值替代了，但精度再高也只是近似值，这个近似值无法使分子 $\lim_{x\to 1}(\sin x-\sin 1)=0$，从而此时分母趋于 0，分子由于近似值的原因不能趋于 0，于是出现了 Nan 这个结果。

遇到这种情况可以用下面方式处理：

```
>>syms x a;
>>f=(sin(x)-sin(a))/(x-a);
>>s(a)=limit(f,x,a);
>>s(1)
ans =cos(1)
```

11、diff():求导和偏导

diff(f)求 f 对默认第一自变量的一阶导。

diff(f,n)求 f 对默认第一自变量的 n 阶导。

可以用 symvar(f,1)确定默认自变量。

diff(f,var,n)求 f 对指定自变量 var 的 n 阶偏导。

diff(f,x,y)表示 f 先对 x 求偏导再对 y 求偏导。

diff(f,x,x,x,y)表示 f 先对 x 求 3 次偏导，再对 y 求一次偏导。

12、int()：求函数的不定积分与定积分，

int(f)：求不定积分；

int(f,x)：求函数 f 对指定变量 x 的不定积分。例

```
>>syms x y;
>>f=x*y;
>> int(f,y)
ans=(x*y^2)/2
```

int(f,a,b)：求函数 f 对默认变量在区间[a,b]内的定积分。例：

```
>>syms x;
>>f=exp(x)*(1+exp(x))^2;
>>d=int(f, 0, log(2))
 d =
(exp(6243314768165359/9007199254740992)*(3*exp(6243314768165359/9007199254740992) + exp(6243314768165359/4503599627370496) + 3))/3 - 7/3
```

这个结果没有化简，显得非常复杂，这是因为各版本定积分的积分结果可能有不同显示，此时可以用 double(d)或者 vpa(d,n)控制结果的显示精度。比如 d=vpa(d,5)，或者 d=double(d)或者 d=round(d,5),此时结果是 d =6.3333。

int(f,x,a,b)：求 f 对指定变量在区间[a,b]内的定积分，a,b 可以是符号表达式。

例：求 $\int_{x}^{x+1} xydy=\frac{x(2x+1)}{2}$

```
>>syms x ,y;
>>f=x*y;
>>s1=int(f,y,x,x+1)
s1=(x*(2*x + 1))/2
```

matlab 中没有专用于求解符号重积分的函数，所以要计算符号重积分，可以叠加使用 int 函数，比如，计算 $\int_0^1 dx \int_0^x xydy$

```
>>syms x y;
>>s=int(int(x*y,y,0,x),x,0,1);
```

这相当于先计算 s1=int(x*y,y,0,x)，然后对这个结果再计算 int(s1,x,0,1)。

13、subs():替代函数。将表达式中的指定变量替换成对应元素并进行运算。

```
>>syms x y z;
>>s=x*y+z^2;
>>subs(s,[x,y,z],[1,2,3])  %用 1 替代 x，2 替代 y，3 替代 z，并计算 s 的值。
ans =11
>>s=subs(s,x,z)     %用 z 替代 x
s =  z^2 + y*z
```

14、MatlabFunction()：将符号型函数转为数值型匿名函数句柄。

```
syms x y;
>>y=cos(x)*exp(x)+x^2;
>>f=matlabFunction(y);
f = 包含以下值的 function_handle: @(x)exp(x).*cos(x)+x.^2
```

注：进行数值计算时常常需要用到数值型的函数句柄。相对于定义函数句柄，定义符号函数更符合我们的使用习惯。所以在需要使用函数句柄时，我们可以使用符号方式定义函数，然后再用 matlabFunction 转为想要用的函数句柄。

15、latex()：将符号型表达式转为对应的 latex 语句。

latex 是一套跨平台的文字排版系统，非常适合用来生成复杂的数学公式和表格，但学习 latex 需要花费一定精力，比如 cos(x)*exp(x)+x^2 对应的 latex 语句是：{\mathrm{e}}^x\,\cos\left(x\right)+x^2 ，显然为了使用 latex 排版系统，需要学习它的语法规则，是一件很麻烦的事。

matlab 提供了 latex()函数，可以帮我将 matlab 中的符号表达式，直接转

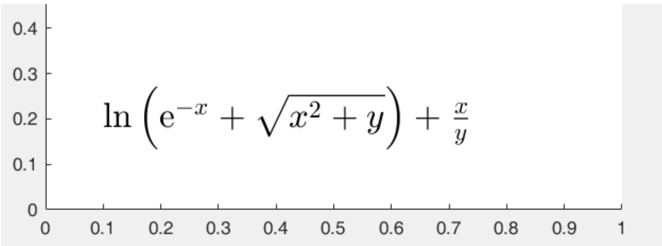
为对应的 latex 语句(是一个字符串)。例如：

```
>>clear;
>>syms x y;
>>y=x/y+log(exp(-x)+sqrt(x^2+y))
>>s=latex(y)
s =' \ln\left({\mathrm{e}}^{-x}+\sqrt{x^2+y}\right)+\frac{x}{y}'
```

得到对应的 latex 语句后，我们可以使用 text 文本框将符号表达式在图中显示出来。接上：

```
>> s1=['$',s,'$']; %在字符串 s 的两头加上$符号，这是显示 latex 语句要求的。
>> text('interpret','latex','string',s1,'fontsize',20,'position',[0.1,0.2]);
% 在图中产生一个文本框，指定用 latex 解释器编译字符串 s1，字体大小设置为 20，文本框的起始显示位置是[0.1,0.2]。
```

显示效果如下：



所以使用 latex 可以把符号表达式显示为我们平时熟悉的样式，这在符号表达式复杂的时候特别有用。**下面将执行 latex 的语句编辑成了一个名为 mylatex 的 m 函数文件，以后想要用 latex 方式显示某个符号对象 s 时，只需执行 mylatex(s)即可。**

```
function mylatex(mysym)
%输入参数 mysym 是符号型对象;
%如果想要显示多个符号对象，可以用中括号连接,将 mysym 设置为[f,g]。
s=latex(simplify(mysym));
s1=['$',s,'$'];
text('string',s1,'Interpreter','latex','FontSize',20,'Position',[0.2,0.2]);
xlim([0,15]);
end
```

练习 1：利用符号表达式求值

已知 $z=\dfrac{x+1}{\sqrt{3+x}-\sqrt{y}}$，求 z(6,5)

```
syms x y;
z(x,y)=(x+1)/(sqrt(3+x)-sqrt(y));
>> z(6,5)
ans =-7/(5^(1/2) - 3)
```

练习 2：分解因式

(1) x^4-y^4　　　　(2) 5135

练习 3：化简表达式

(1) $\sin a\cos b-\cos a\sin b$　(2) $\dfrac{4x^2+8x+3}{2x+1}$

练习 4：符号矩阵运算

已知

$$P_1=\begin{bmatrix}0&1&0\\1&0&0\\0&0&1\end{bmatrix},\quad P_2=\begin{bmatrix}1&0&0\\0&1&0\\1&0&1\end{bmatrix},\quad A=\begin{bmatrix}a&b&c\\d&e&f\\g&h&k\end{bmatrix}$$

完成下列运算：

(1) $B=P_1\cdot P_2\cdot A$

(2) B 的逆矩阵并验证结果。

(3) 包括 B 矩阵主对角线元素的下三角阵。

(4) B 的行列式值。

练习 5：用符号方法求下列极限或导数

(1) $\lim\limits_{x\to 0}\dfrac{x(e^{\sin x}+1)-2(e^{\tan x}-1)}{\sin^3 x}$　　(2) $\lim\limits_{x\to -1^+}\dfrac{\sqrt{\pi}-\sqrt{\arccos x}}{\sqrt{x+1}}$

(3) $y=\dfrac{1-\cos(2x)}{x}$, 求y', y''

(4) 已知$A=\begin{bmatrix}a^x & t^3\\ t\cos x & \ln x\end{bmatrix}$, 分别求$\dfrac{\mathrm{d}A}{\mathrm{d}x},\dfrac{\mathrm{d}^2A}{\mathrm{d}t^2},\dfrac{\mathrm{d}^2A}{\mathrm{d}x\mathrm{d}t}$

(5) 已知 $f(x,y)=(x^2-2x)e^{-x^2-y^2-xy}$，求 $\left.\frac{\partial y}{\partial x},\frac{\partial^2 f}{\partial x\partial y}\right|_{x=0,y=1}$

练习 6：用符号方法求下列积分（必要时对结果进行化简）

(1) $\int\frac{\mathrm{d}x}{1+x^4+x^8}$　　(2) $\int\frac{\mathrm{d}x}{(\arcsin x)^2\sqrt{1-x^2}}$

```
>>clear; syms x;
>>f1=1/(1+x^4+x^8);  %第（1）题
>>s1=simplify(int(f1))
s1 =
-(3^(1/2)*(atan((3^(1/2)*x)/(x^2 - 1)) - atanh((3^(1/2)*x)/(x^2 + 1))))/6
>>f2=1/(asin(x)^2*sqrt(1-x^2)); %第（2）题
>>s2=simplify(int(f2))
s2 = -1/asin(x)
```

(3) $\int_0^{+\infty}\frac{x^2+1}{x^4+1}\mathrm{d}x$　　(4) $\int_0^{\ln 2}e^x(1+e^x)^2\mathrm{d}x$

```
>> clear; syms x;  %第(3)题
>> f3=(x^2+1)/(x^4+1);
>> s3=int(f3,x,0,inf)
s3 = (pi*2^(1/2))/2
>> f4=exp(x)*(1+exp(x))^2;  %第(4)题
>>s4= simplify(int(f4,0,log(2)))
s4 =
(exp(6243314768165359/9007199254740992)*(3*exp(6243314768165359/9007199254740
992) + exp(6243314768165359/4503599627370496) + 3))/3 - 7/3
>>round(s4,3)  %或者 vpa(s4,5)
ans=6.333
```

（5）$\int_0^a dy\int_0^{\sqrt{a^2-y^2}}\left(x^2+y^2\right)dx$ （6）$\int_0^1 dx\int_0^{1-x}dy\int_0^{1-x-y}xyzdz$

ans=$\frac{\pi\cdot a^4}{8}$ ans=$\frac{1}{720}$

练习 7：已知方程组$\begin{cases}F\left(x,y,u,v\right)=0\\G\left(x,y,u,v\right)=0\end{cases}$，$F,G$ 可微，用符号运算验证雅可比行列式$\begin{vmatrix}x_u' & x_v'\\ y_u' & y_v'\end{vmatrix}$与$\begin{vmatrix}u_x' & u_y'\\ v_x' & v_y'\end{vmatrix}$互为倒数。

分析：将方程组中的x,y看作u,v的函数，两边对u求偏导：

$\begin{cases}F_1x_u'+F_2y_u'+F_3\cdot 1+F_4\cdot 0=0\\G_1x_u'+G_2y_u'+G_3\cdot 1+G_4\cdot 0=0\end{cases}$，解出：

$$x_u'=-\frac{\begin{vmatrix}F_3 & F_2\\ G_3 & G_2\end{vmatrix}}{\begin{vmatrix}F_1 & F_2\\ G_1 & G_2\end{vmatrix}},\quad y_u'=-\frac{\begin{vmatrix}F_1 & F_3\\ G_1 & G_3\end{vmatrix}}{\begin{vmatrix}F_1 & F_2\\ G_1 & G_2\end{vmatrix}};$$

同理两边对v求偏导可解出：

$$x_v'=-\frac{\begin{vmatrix}F_4 & F_2\\ G_4 & G_2\end{vmatrix}}{\begin{vmatrix}F_1 & F_2\\ G_1 & G_2\end{vmatrix}}\quad y_v'=-\frac{\begin{vmatrix}F_1 & F_4\\ G_1 & G_4\end{vmatrix}}{\begin{vmatrix}F_1 & F_2\\ G_1 & G_2\end{vmatrix}}$$

将方程组中的u,v看作的函数x,y，两边对x求偏导：

$\begin{cases}F_1\cdot 1+F_2\cdot 0+F_3\cdot u_x'+F_4\cdot v_x'=0\\G_1\cdot 1+G_2\cdot 0+G_3\cdot u_x'+G_4\cdot v_x'=0\end{cases}$，解出：

$$u_x'=-\frac{\begin{vmatrix}F_1 & F_4\\ G_1 & G_4\end{vmatrix}}{\begin{vmatrix}F_3 & F_4\\ G_3 & G_4\end{vmatrix}},\quad v_x'=-\frac{\begin{vmatrix}F_3 & F_1\\ G_3 & G_1\end{vmatrix}}{\begin{vmatrix}F_3 & F_4\\ G_3 & G_4\end{vmatrix}}$$

两边对y求偏导可解出：

$$u_y' = -\frac{\begin{vmatrix} F_2 & F_4 \\ G_2 & G_4 \end{vmatrix}}{\begin{vmatrix} F_3 & F_4 \\ G_3 & G_4 \end{vmatrix}}, \quad v_y' = -\frac{\begin{vmatrix} F_3 & F_2 \\ G_3 & G_2 \end{vmatrix}}{\begin{vmatrix} F_3 & F_4 \\ G_3 & G_4 \end{vmatrix}}$$

```
clear;
syms f1 f2 f3 f4 g1 g2 g3 g4;
xu=-det([f3 f2;g3,g2])/det([f1 f2;g1,g2]);
yu=-det([f1 f3;g1,g3])/det([f1 f2;g1,g2]);
xv=-det([f4 f2;g4,g2])/det([f1 f2;g1,g2]);
yv=-det([f1 f4;g1,g4])/det([f1 f2;g1,g2]);

ux=-det([f1,f4;g1,g4])/det([f3 f4;g3,g4]);
vx=-det([f3,f1;g3,g1])/det([f3 f4;g3,g4]);
uy=-det([f2,f4;g2,g4])/det([f3 f4;g3,g4]);
vy=-det([f3,f2;g3,g2])/det([f3 f4;g3,g4]);

xyuvjocbi=simplify(det([xu,xv;yu,yv]));
uvxyjocbi=simplify(det([ux,uy;vx,vy]));
mylatex([xyuvjocbi,uvxyjocbi]);
```

$$\left(\begin{array}{cc} \frac{f_3\,g_4-f_4\,g_3}{f_1\,g_2-f_2\,g_1} & \frac{f_1\,g_2-f_2\,g_1}{f_3\,g_4-f_4\,g_3} \end{array} \right)$$