



信阳师范学院
数学与统计学院
SCHOOL OF MATHEMATICS AND STATISTICS

第7章 MATLAB符号运算

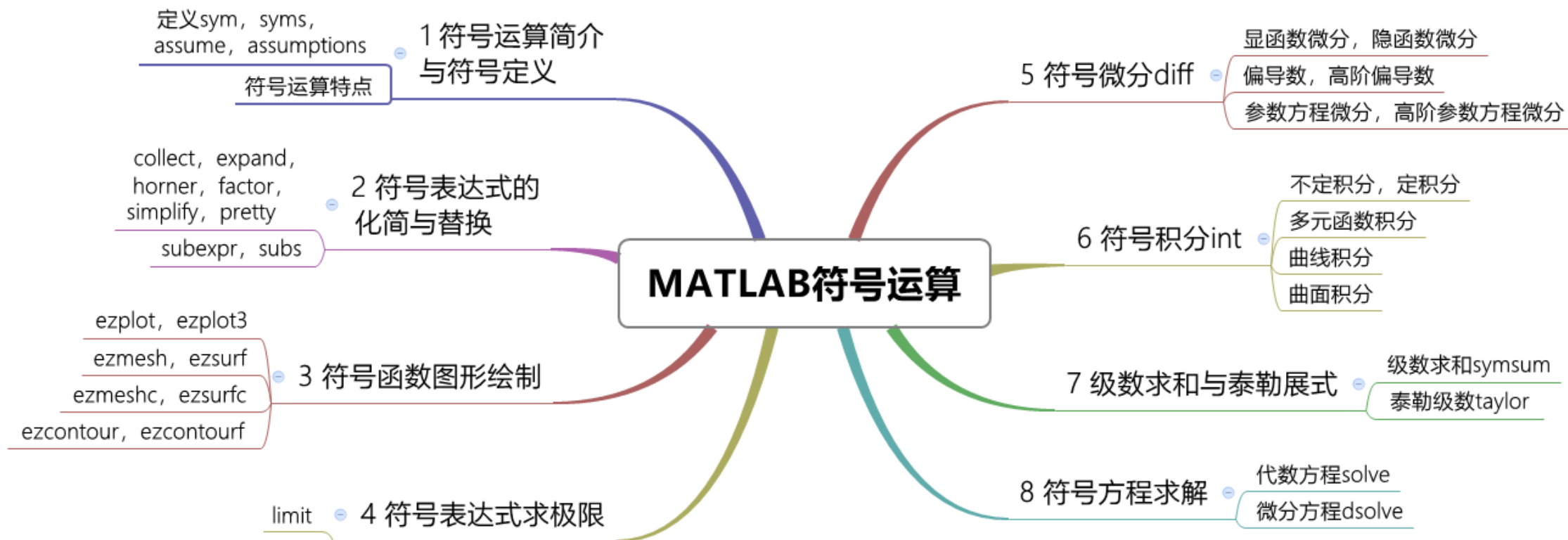


讲授人：牛言涛



日期：2020年3月22日

第7章 MATLAB符号运算思维导图



符号运算与数值运算的区别：数值计算的表达式、矩阵变量中**不允许有未定义**的自由变量，而符号计算**可以含有未定义的符号变量**。符号计算存放的是精确数据，耗存储空间，运行速度慢，但结果精度高；数值计算则是以一定精度来计算的，计算结果有误差，但是运行速度快。

1. 符号运算简介

- 自然科学理论分析中的公式、关系式及其推导是符号计算要解决的问题。
- MATLAB数值计算的对象是数值，而符号计算的对象则是非数值的符号字符串。
- 符号运算与数值运算的区别
 - 数值运算中必须先对变量赋值，然后才能参与运算。
 - 符号运算无须事先对独立变量赋值，运算结果以标准的符号形式表达。

代数方程: $ax^2 + bx + c = 0 (a \neq 0)$

微分方程: $y' = ay$, 解的形式为 $y(t) = e^{at+c} = c_1 e^{at}$

函数导数: $\frac{d}{dx}(\cos^2 x) = -2 \cos x \sin x$

定积分: $\int_a^b x^2 dx = \frac{1}{3}(b^3 - a^3)$

1. 符号运算简介

- 符号计算可以得到问题精确地完备解，但是计算量大且表达形式庞大；数值计算可以快速的处理很多实际应用中的问题，但是一般只能得到近似的局部解。
- 数值计算在处理病态问题时，收敛往往较慢容易出错。符号计算能给出精确结果，这一特点为用户提供了良好的使用环境，可避免由舍入误差引起的“病态问题”。

符号运算的特点：

- (1) 符号计算是精确计算
- (2) 运算对象可以是没赋值的符号变量
- (3) 符号计算的计算速度较慢
 - Symbolic Math Toolbox——符号运算工具包通过调用Maple软件实现符号计算的。
 - Maple软件——主要功能是符号运算，它占据符号软件的主导地位。

2. 符号运算常见定义函数和精度显示

函数	函数描述	举例
sym	用来建立单个符号变量或表达式，也可以用a = ‘x’ 来创建。 <u>符号变量要先定义，后引用！！</u>	>> a=sym('x') >> c=sym('(1+sqrt(3))/2')
syms	一次可定义多个符号变量，变量间用空格而不要用逗号分隔。 <u>syms 符号变量名1 符号变量名2 … 符号变量名n</u>	>> syms a b c d x >> fun = a*exp(b*x)-sin(c+d*x);
findsym symvar	用于确定一个表达式中的符号变量， <u>推荐用symvar</u>	>> symb = symvar(fun) symb =a,b,c,d,x
subs	可以将符号表达式中的符号变量用数值代替，或者用符号函数替换，也可嵌套替换变量	>> fun1 = subs(subs(fun,a,1),b,5) >> fun2 = subs(fun1,c, 'log(x)') >> fun3 = subs(subs(fun2,d,3),x,2) fun3 = exp(10) - sin(log(2) + 6)
digits	设定所用数值的精度, digits(D)： 设置数值的精度为D位。 The default is 32 digits.	>> digits(55)
vpa	进行可控精度运算, vpa(S,D)： 显示符号表达式S在精度D下的值。	>> digits(55) >> vpa('1/2+1/3') >> p = vpa('pi',100)

3. sym函数

- 符号对象中的符号常量、变量、函数和表达式，可以用`sym`和`syms`函数创建。
- 使用`class`函数测试建立的操作对象为何种操作对象类型及是否为符号对象类型。
- `sym`函数：可以生成单个的符号变量
- `var = sym('var', set)`：创建一个符号变量，并设置符号对象的格式，`set`可以不选
 - `set`: integer, rational, real, or positive, 用`assume`替代
 - `sym('var', 'clear')`：清除先前设置的符号变量，用`assume`替代

```
>> x = sym('x','real'); %实数
>> y = sym('y','positive'); %正数
>> z = sym('z','integer'); %整数
>> t = sym('t','rational'); %有理数
>> assumptions %显示影响符号变量、表达式或函数的假设
ans =
[ in(t, 'rational'), in(x, 'real'), 0 < y, in(z, 'integer')]
```

```
>> assume([x y z t],'clear') %设置符号对象的假设
>> assumptions
ans =
Empty sym: 1-by-0
```

assume与assumptions函数

sym

Create symbolic variables, expressions, functions, matrices

[collapse all in page](#)

! The syntaxes `sym(A,set)` and `sym(A,'clear')` for a symbolic object `A` in the MATLAB® workspace have been removed. Use `assume(A,set)` and `assume(A,'clear')` instead.

Support of character vectors that are not valid variable names and do not define a number will be removed in a future release. To create symbolic expressions, first create symbolic variables, and then use operations on them. For example, use `syms x; x + 1` instead of `sym('x + 1')`, `exp(sym(pi))` instead of `sym('exp(pi)')`, and `syms f(var1,...varN)` instead of `f(var1,...varN) = sym('f(var1,...varN)')`.

Common Assumptions

Set an assumption using the associated syntax.

Assume 'x' is	Syntax
real	<code>assume(x,'real')</code>
rational	<code>assume(x,'rational')</code>
positive	<code>assume(x,'positive')</code>
positive integer	<code>assume(x,{'positive','integer'})</code>
less than -1 or greater than 1	<code>assume(x<-1 x>1)</code>
an integer from 2 through 10	<code>assume(in(x,'integer') & x>2 & x<10)</code>
not an integer	<code>assume(~in(z,'integer'))</code>
not equal to 0	<code>assume(x ~= 0)</code>
even	<code>assume(x/2,'integer')</code>
odd	<code>assume((x-1)/2,'integer')</code>
from 0 through 2π	<code>assume(x>0 & x<2*pi)</code>
a multiple of π	<code>assume(x/pi,'integer')</code>

```
>> syms n
>> assume(n,'integer')
>> assumptions
ans =
in(n, 'integer')
>> syms x
>> assume(n<x & x<42)
>> assumptions(n)
ans =
[ n < x, x < 42]
>> syms m
>> assume(1<m<3)
>> assumptions([m x])
ans =
[ 1 < m, m < 3, n < x, x < 42]
```

assume与assumptions函数

```
>> assumptions
ans =
[ n < x, x < 42, 1 < m, m < 3]
>> assume([m n x],'clear')
>> assumptions
ans =
Empty sym: 1-by-0
%定义x为负实数
>> syms x
>> assume(x,'real')
>> assumeAlso(x<0)
>> assumptions(x)
ans =
[ x < 0, in(x, 'real')]
>> assume(x,'clear')
```

%影响表达式和函数的假设

```
>> syms a b c
>> expr = a*exp(b)*sin(c);
>> assume(a+b > 3 & in(a,'integer') & in(c,'real'))
>> assumptions(expr)
```

```
ans =
[ 3 < a + b, in(a, 'integer'), in(c, 'real')]
```

%影响函数的假设

```
>> syms f(a,b,c)
>> assumptions(f)
ans =
[ 3 < a + b, in(a, 'integer'), in(c, 'real')]
>> assume([a b c],'clear')
>> assumptions
ans = Empty sym: 1-by-0
```

%恢复旧假设

```
>> syms m k positive
>> syms x(t)
>> dsolve(m*diff(x,t,t) == -k*x, x(0)==0)
ans = C3*sin((k^(1/2)*t)/m^(1/2))
>> tmp = assumptions; %存储旧假设
>> assume([m k],'clear') %清楚, 求解不受约束
>> dsolve(m*diff(x,t,t) == -k*x, x(0)==0)
ans = C6*exp(-(t*(-k*m)^(1/2))/m) -
C6*exp((t*(-k*m)^(1/2))/m)
>> assume(tmp) %恢复旧假设
```


3. sym函数——符号型数据变量的创建

- `Num = sym(Num,flag)`: 将一个数值转换为符号形式, 输入参数flag为转换的符合对象应该符合的格式类型

- 'r': 最接近有理表示, 为系数默认设置
- 'e': 带估计误差的有理表示
- 'f': 十六进制浮点表示
- 'd': 最接近的十进制浮点精确表示

```
r = sym(pi)
```

```
%r = pi
```

```
f = sym(pi,'f')
```

```
%f = 884279719003555/281474976710656
```

```
d = sym(pi,'d')
```

```
%d = 3.1415926535897931159979634685442
```

```
e = sym(pi,'e')
```

```
%e = pi - (198*eps)/359
```

```
%Create Large Symbolic Numbers
```

```
%When creating symbolic numbers with 15 or more digits,  
use quotation marks to accurately represent the numbers
```

```
inaccurateNum = sym(111111111111111111)
```

```
inaccurateNum =
```

```
11111111111111110656
```

```
accurateNum = sym('111111111111111111')
```

```
accurateNum =
```

```
111111111111111111
```

3. sym函数——创建符号数值

- 将数值转换为符号数字或表达式。
- 在子表达式上使用sym而不是整个表达式以获得更好的精度。
- 在整个表达式上使用sym是不准确的，因为MATLAB首先将表达式转换为浮点数，这样就失去了准确性。sym不能总是恢复这个丢失的准确性。

```
>> inaccurate1 = sym(1/1234567)
inaccurate1 =
7650239286923505/9444732965739290427392
>> inaccurate1 = 1/sym(1234567)
inaccurate1 =
1/1234567
>> inaccurate2 = sym(sqrt(1234567))
inaccurate2 =
4886716562018589/4398046511104
```

```
>> inaccurate2 = sqrt(sym(1234567))
inaccurate2 =
1234567^(1/2)
>> inaccurate3 = sym(exp(pi))
inaccurate3 =
6513525919879993/281474976710656
>> inaccurate3 = exp(sym(pi))
inaccurate3 =
exp(pi)
```

3. sym函数——创建符号矩阵

- $A = \text{sym}('A', \text{dim})$: 创建一个矢量或矩阵的符号变量
- $A = \text{sym}('A', \text{set})$: 创建一个符号矩阵, set用于设置矩阵的维数
- $\text{sym}(A, 'clear')$: 清除前面已创建的符号矩阵, 用assume替代
- $Af(\text{arg1}, \dots, \text{argN}) = \text{sym}('f(\text{arg1}, \dots, \text{argN})')$: 根据f指定的输入参数arg1,...,argN创建符号变量f(arg1,...,argN)

```
>> A = sym('A',[3 4])
```

```
A =
```

```
 [ A1_1, A1_2, A1_3, A1_4]
```

```
 [ A2_1, A2_2, A2_3, A2_4]
```

```
 [ A3_1, A3_2, A3_3, A3_4]
```

```
>> B = sym('x%d%d',4) %格式化
```

```
B =
```

```
 [ x11, x12, x13, x14]
```

```
 [ x21, x22, x23, x24]
```

```
 [ x31, x32, x33, x34]
```

```
 [ x41, x42, x43, x44]
```

```
%高维矩阵
```

```
>> A = sym('a',[2 2 2])
```

```
A(:, :, 1) =
```

```
 [ a1_1_1, a1_2_1]
```

```
 [ a2_1_1, a2_2_1]
```

```
A(:, :, 2) =
```

```
 [ a1_1_2, a1_2_2]
```

```
 [ a2_1_2, a2_2_2]
```

```
>> A = sym('A%d%d',[2 2],'positive')
```

```
>> sol = solve(A(1,1)^2-1, A(1,1))
```

```
%sol = 1
```

```
>> assumptions(A)
```

```
%[ 0 < A21, 0 < A11, 0 < A22, 0 < A12]
```

```
>> assume(A,'clear');
```

```
>> assumptions(A)
```

```
%ans = Empty sym: 1-by-0
```

```
>> sol = solve(A(1,1)^2-1, A(1,1))
```

```
%sol = -1 1
```

3. sym函数——创建符号函数



- 从匿名函数句柄创建符号表达式

```
>> h_expr = @(x)(sin(x) + cos(x));
```

```
>> sym_expr = sym(h_expr)
```

```
sym_expr =
```

```
cos(x) + sin(x)
```

```
>> ezplot(sym_expr) %符号函数ezplot绘制曲线
```

%帕斯卡矩阵：由杨辉三角形表组成的矩阵称为帕斯卡(Pascal)矩阵。

```
>> h_matrix = @(x)(x*pascal(3));
```

```
>> sym_matrix = sym(h_matrix)
```

```
sym_matrix =
```

```
[ x,  x,  x]
```

```
[ x, 2*x, 3*x]
```

```
[ x, 3*x, 6*x]
```

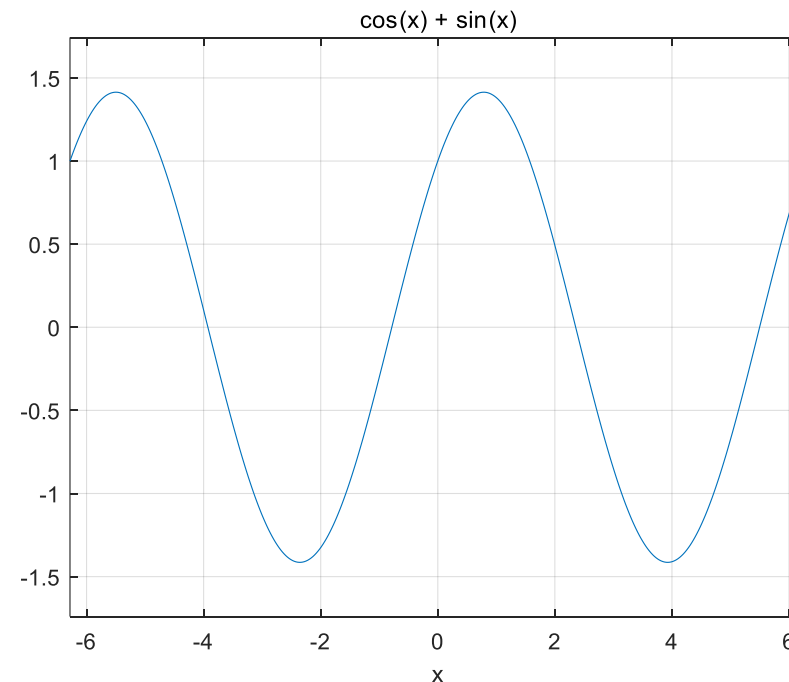
```
>> subs(sym_matrix,symvar(sym_matrix),2) %使用函数subs替换符号x为2, 并使用  
函数symvar识别符号变量x
```

```
ans =
```

```
[ 2, 2,  2]
```

```
[ 2, 4,  6]
```

```
[ 2, 6, 12]
```



4. syms函数简介

- syms函数：可以创建任意多个符号变量
 - `syms var...varN`：创建符号变量var...varN
 - `syms var...varN set`：set指定符号对象的格式：real, positive, integer, or rational.
 - `syms var...varN clear`：清除前面已经定义好的符号对象
 - `syms f(arg1,...,argN)`：创建符号函数f，函数中包含多个符号变量

```
>> syms x y integer %创建整型符号变量
```

```
>> fh = x*sin(y) + y*sin(x);
```

```
>> syms s(t) f(x,y) %创建符号函数
```

```
>> f(x,y) = x + 2*y;
```

```
>> val = f(1,2)
```

```
val =
```

5

```
>> syms x y
```

```
>> m1 = [1,2+x,1;3-x,1,4+y;1,2+y,0]
```

```
m1 =
```

```
[ 1, x + 2, 1]
```

```
[ 3 - x, 1, y + 4]
```

```
[ 1, y + 2, 0]
```

4. syms函数简介

%用矩阵作为公式创建符号函数

```
>> syms x
>> f(x) = [x x^3; x^2 x^4]
f(x) =
    [ x, x^3]
    [ x^2, x^4]
>> A = f(2)
A =
    [ 2, 8]
    [ 4, 16]
>> y = f([1 2 3; 4 5 6])
y =
2×2 cell 数组
    [2×3 sym]    [2×3 sym]
    [2×3 sym]    [2×3 sym]
```

```
>> Y2 = y{2}
```

```
Y2 =
    [ 1, 4, 9]
    [ 16, 25, 36]
```

%符号计算的运算符

```
>> syms x
>> A = sym([x^2 3; 4 * x cos(x)]);
>> B = sym([1/x^2 2*x; 3 x^2+x]);
>> C = A + B
C =
    [ 1/x^2 + x^2,      2*x + 3]
    [ 4*x + 3, x + cos(x) + x^2]
>> J = A / B
```

%点乘与乘法

```
>> syms a b c d e f g h;
>> A = sym([a,b;c,d]);
>> B = sym([e,f;g,h]);
>> C = A*B
C =
    [ a*e + b*g, a*f + b*h]
    [ c*e + d*g, c*f + d*h]
>> D = A.*B
D =
    [ a*e, b*f]
    [ c*g, d*h]
```



感谢聆听
