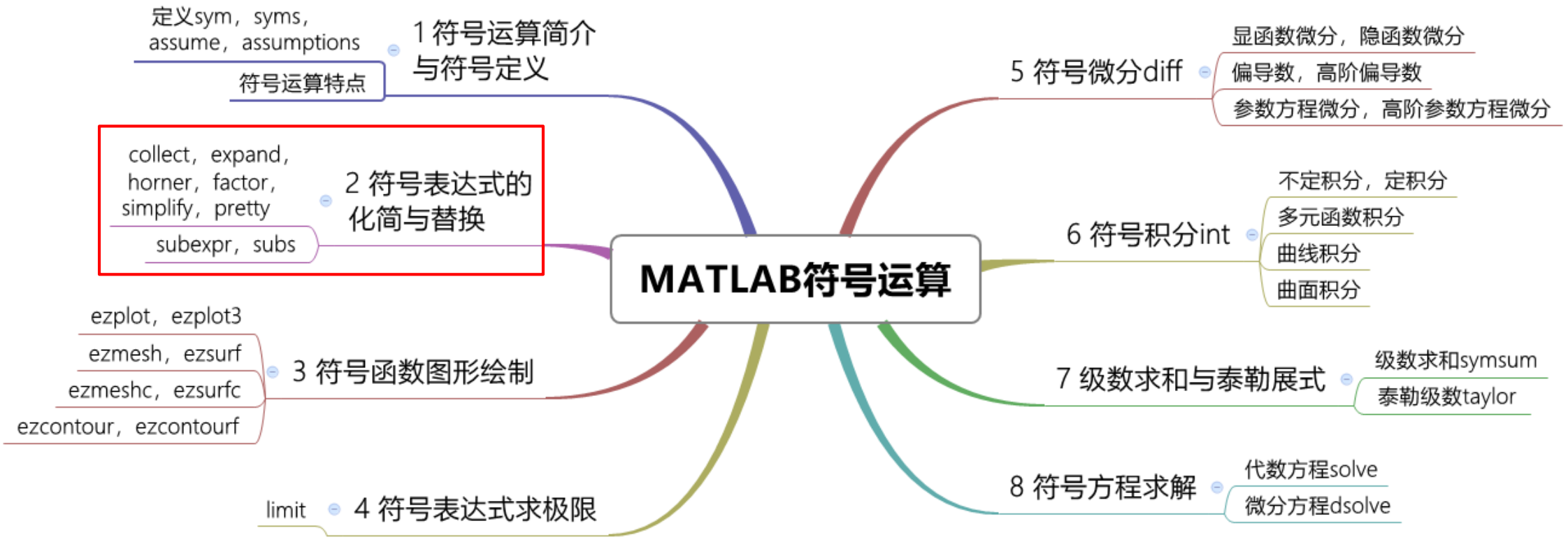
信陽師範学院
数学与统计学院
SCHOOL OF MATHEMATICS AND STATISTICS

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。

符号表达式的化简与替换

- collect、expand、horner、factor、simplify和pretty函数分别实现符号表达式的化简。
- subexpr 和subs分别用来实现变量替换。

collect

用来合并同类项

expand

用于符号表达式的展开

factor

实现因式分解功能

horner

将函数转化为嵌套格式，降低运算的复杂度

simplify

实现表达式的化简

pretty

漂亮地打印符号表达式

subexpr

自动将表达式中重复出现的字符串用变量替换

subs

将符号表达式中的某些符号变量替换为指定的新的变量

1. collect函数

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

- collect：函数用来合并同类项。
  - R = collect(S)：S可以是数组，数组的每个元素为符号表达式。该命令将S中的每个元素进行合并同类项。
  - R = collect(S,v)：对指定的变量v进行合并，如果不指定，则默认对x进行合并，或者由symvar函数返回的结果进行合并。
  - collect(P, expr)：收集符号表达式expr幂的P系数。如果P是向量或矩阵，则collect对P按元素顺序执行操作。如果expr是向量，则collect根据expr中的所有表达式查找系数。
- 例1：对下列函数合并同类项

$$f_1=\left(e^x+x\right)\left(x+2\right) \qquad f_2=x^2y+yx-x^2-2x \qquad f_3=a^2xy+abx^2+axy+x^2$$

$$f_4=2xi-3yi \qquad f_5=x\pi\left(\pi-y\right)+x\left(\pi+i\right)+3\pi y$$

1. collect函数

```
>> syms x
>> coeffs = collect((exp(x) + x)*(x + 2))   %未指定符号变量
coeffs =
   x^2 + (exp(x) + 2)*x + 2*exp(x)
>> symvar((exp(x) + x)*(x + 2), 1)  %查找表达式中的符号变量
ans =
   x
%收集特定变量的幂系数
>> syms x y
>> coeffs_x = collect(x^2*y + y*x - x^2 - 2*x, x)
coeffs_x =
   (y - 1)*x^2 + (y - 2)*x
>> coeffs_y = collect(x^2*y + y*x - x^2 - 2*x, y)
coeffs_y =
   (x^2 + x)*y - x^2 - 2*x
>> syms a b
%指定对x、y收集
>> coeffs_xy = collect(a^2*x*y + a*b*x^2 + a*x*y +
x^2, [x y])
coeffs_xy =
    (a*b + 1)*x^2 + (a^2 + a)*x*y
%根据i和pi收集系数
>> coeffs_i = collect(2*x*i - 3*i*y, i)
coeffs_i =
    (2*x - 3*y)*1i
>> coeffs_pi = collect(x*pi*(pi - y) + x*(pi + i) + 3*pi*y,
pi)
coeffs_pi =
   x*pi^2 + (x + 3*y - x*y)*pi + x*1i
```

1. collect函数

%符号表达式和函数的系数集合

```
>> syms x y
>> fh = expand(sin(x + 3*y));
coeffs_cosy = collect(fh, cos(y))
coeffs_cosy =
    (4*sin(x))*cos(y)^3 + (4*cos(x)*sin(y))*cos(y)^2 + (-
3*sin(x))*cos(y) - cos(x)*sin(y)
>> coeffs_sinxsiny = collect(fh, [sin(x) sin(y)])
coeffs_sinxsiny =
    (4*cos(y)^3 - 3*cos(y))*sin(x) + (4*cos(x)*cos(y)^2 -
cos(x))*sin(y)
>> syms y(x)
>> fh2 = y^2*x + y*x^2 + y*sin(x) + x*y;
>> coeffs_y = collect(fh2, y)
coeffs_y(x) =
    x*y(x)^2 + (x + sin(x) + x^2)*y(x)
```

%为矩阵的每个元素收集系数

```
>> syms x y
>> A = collect([(x + 1)*(y + 1), x^2 + x*(x -y); 2*x*y - x, x*y +
x/y], x)
A =
   [ (y + 1)*x + y + 1, 2*x^2 - y*x]
   [       (2*y - 1)*x, (y + 1/y)*x]
```

%收集函数调用的系数

```
>> syms a b c d e f x
>> F = a*sin(2*x) + b*sin(2*x) + c*cos(x) + d*cos(x) + e*sin(3*x)
+f*sin(3*x);
>> collect(F, 'sin')
ans =
   (a + b)*sin(2*x) + (e + f)*sin(3*x) + c*cos(x) + d*cos(x)
>> collect(F, {'sin' 'cos'})
ans =
   (c + d)*cos(x) + (a + b)*sin(2*x) + (e + f)*sin(3*x)
```

2. expand函数

- expand函数：用于符号表达式的展开。其操作对象可以是多种类型，如多项式、三角函数、指数函数等。
- 例2：求下列函数展开式

(1) $f_1(x,y)=(x+y)^3$

(2) $f_2(x,y)=\sin(x+y)$

(3) $f_3(x,y)=e^{x+y}$

(4) $f_4(x)=(\sin 3x-1)^2$

(5) $f_5=\ln\left(\frac{ab}{c}\right)^2$

```
>> syms x y;
>> f1 = (x+y)^3;
>> f1_exd = expand(f1)
f1_exd =
x^3 + 3*x^2*y + 3*x*y^2 + y^3
>> f2_exd = expand(sin(x+y))
f2_exd =
cos(x)*sin(y) + cos(y)*sin(x)
>> f3_exd = expand(exp(x+y))
f3_exd =
exp(x)*exp(y)
```

2. expand函数

```
>> syms x
>> sf = (sin(3*x) – 1)^2;
>> fep = expand(sf)
fep =
2*sin(x) + sin(x)^2 - 8*cos(x)^2*sin(x) -
8*cos(x)^2*sin(x)^2 + 16*cos(x)^4*sin(x)^2 + 1
```

%通过将“ArithmeticOnly”设置为true，抑制函数（如sin(3*x)）的扩展。

```
>> fep2 = expand(sf, 'ArithmeticOnly', true)
fep2 =
sin(3*x)^2 - 2*sin(3*x) + 1
```

```
>> syms a b c
>> fl = log((a*b/c)^2);
>> flg = expand(fl)
ans =
log((a^2*b^2)/c^2)
```

%通过将“IgnoreAnalyticConstraints”设置为true，应用标识来简化对数的输入。

```
>> flg2 = expand(fl,'IgnoreAnalyticConstraints',true)
flg2 =
2*log(a) + 2*log(b) - 2*log(c)
```

3. horner函数

- horner(p)函数：将函数转化为嵌套格式。嵌套格式在多项式求值中可以降低计算时间复杂度。
- horner(p,var)该格式目标版本2016b不支持。

```
>> syms x y;
>> fun = expand((x-2)^3)
fun =
    x^3 - 6*x^2 + 12*x - 8
>> funh = horner(fun)
funh =
   x*(x*(x - 6) + 12) – 8
>> fun1 = x^3+3*x+1;
>> fun2 = 3*y^2+4*y+7;
>> horfun = horner([fun1,fun2])
horfun =
    [ x*(x^2 + 3) + 1, y*(3*y + 4) + 7]
```

因式乘积的形式

$$f(x)=(x^2+x+1)(x^3+1)$$

```
>> fx = (x^2+x+1)*(x^3+1);
>> fxh = horner(fx)
fxh =
x*(x*(x*(x*(x + 1) + 1) + 1) + 1) + 1
```

4. factor函数

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

- factor函数：实现因式分解功能，如果输入的参数为正整数，则返回此数的素数因子。

```
>> F = factor(823429252)
F =
     2          2          59         283        12329
%大数字用单引号括起来转化为符号数值
>> F = factor(sym('82342925225632328'))
F =
    [ 2, 2, 2, 251, 401, 18311, 5584781]
>> F = factor(sym(-92465))  %负数
F =
    [ -1, 5, 18493]
>> F = factor(sym(112/81))   %分数形式
F =
   [ 2, 2, 2, 2, 7, 1/3, 1/3, 1/3, 1/3]
```

```
>> sym x;
>> fun = 4*x^3+x^4+8*x+5*x^2+6;
>> hfun = factor(fun)
hfun =
  [ x + 3, x + 1, x^2 + 2]
>> syms a b c d
>> y = -a*b^5*c*d*(a^2 - 1)*(a*d - b*c);
>> F = factor(y,[b c])
F =
  [ -a*d*(a - 1)*(a + 1), b, b, b, b, b, c, a*d - b*c]
```

4. factor函数

使用FactorMode参数选择特定的分解模式。在不指定因子分解模式的情况下对表达式进行因子分解。默认情况下，factor对有理数使用因子分解。在这种模式下，因子保持有理数的精确符号形式。

```
>> syms x
>> factor(x^3 + 2, x)
ans =
x^3 + 2
>> factor(x^3 + 2, x, 'FactorMode', 'real')
ans =
[ x + 1.259921049894873164767210607278 2, x^2 - 1.259921049894873164767210607278 2*x + 1.587401051968199474751705639272 3]
>> factor(x^3 + 2, x, 'FactorMode', 'complex')
ans =
[ x + 1.259921049894873164767210607278 2, x - 0.629960524947436582383605303639 11 + 1.091123635971721403560072614189 8i, x - 0.629960524947436582383605303639 11 - 1.091123635971721403560072614189 8i]
>> factor(x^3 + 2, x, ‘FactorMode’, ‘full’)  %2016b版本不支持
```

5. simplify函数

- simplify函数：实现表达式的化简，化简所选用的方法为Maple中的化简方法。
- 例3：化简如下函数

$$(1)\sin^2 x+\cos^2 x;$$

$$(2)e^{c\ln\sqrt{a+b}};$$

$$(3)\frac{x^2+5x+6}{x+2};$$

$$(4)\sqrt{16}.$$

```
>> syms x c a b;
>> s1 = simplify(sin(x)^2 + cos(x)^2)
s1 =
   1
>> s2 = simplify(exp(c*log(sqrt(a+b))))
s2 =
   (a + b)^(c/2)
>> s3 = simplify([(x^2+5*x+6)/(x+2),sqrt(16)])
s3 =
   [ x + 3, 4]
```

6. 符号表达式的替换

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

subexpr：该函数自动将表达式中重复出现的字符串用变量替换。

- [Y,SIGMA] = subexpr(X,SIGMA)：制定用符号变量SIGMA来代替符号表达式中重复出现的字符串。
- [Y,SIGMA] = subexpr(sol,'beta')：指定所用符号

subs：表示将符号表达式中的某些符号变量替换为指定的新的变量

- R = subs(S)，R = subs(S,new)：直接替换，变量为symvar识别的符号变量
- R = subs(S,old,new)：用新的符号变量new替换S中的变量，被替换的变量由old指定。

6. 符号表达式的替换——subexpr函数

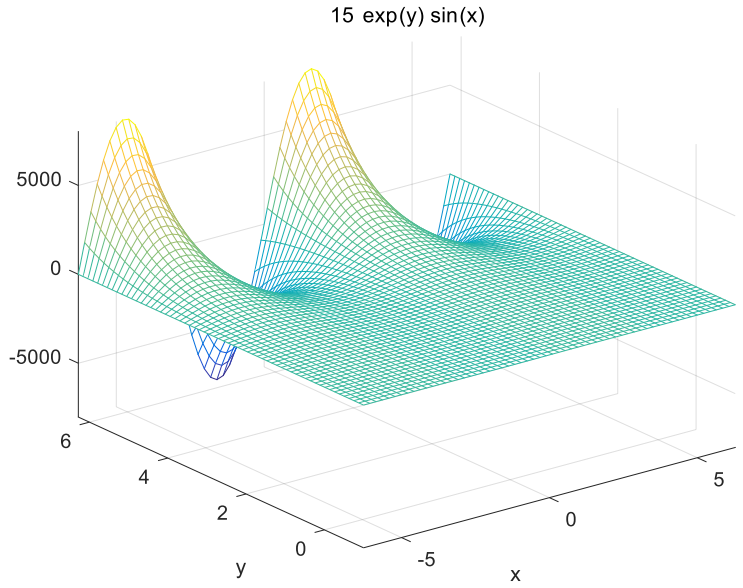
```
>> syms x a
>> sol=solve(x^3+a*x+1,x,'MaxDegree',3)
>> r = subexpr(sol)
sigma =
  (a^3/27 + 1/4)^(1/2) - 1/2
r =
    sigma^(1/3) - a/(3*sigma^(1/3))
    a/(6*sigma^(1/3)) - (3^(1/2)*(a/(3*sigma^(1/3)) +
sigma^(1/3))*1i)/2 - sigma^(1/3)/2
    (3^(1/2)*(a/(3*sigma^(1/3)) + sigma^(1/3))*1i)/2 +
a/(6*sigma^(1/3)) - sigma^(1/3)/2
>> pretty(r)  %pretty 漂亮地打印符号表达式（看起来是有分子分母的格式）
```

```
>> syms a b c x
>> solutions = solve(a*x^2 + b*x + c == 0, x)
solutions =
   -(b + (b^2 - 4*a*c)^(1/2))/(2*a)
   -(b - (b^2 - 4*a*c)^(1/2))/(2*a)
>> syms s
>>[abbrSolutions,s] = subexpr(solutions,s)
abbrSolutions =
   -(b + s)/(2*a)
   -(b - s)/(2*a)
s =
  (b^2 - 4*a*c)^(1/2)
```

6. 符号表达式的替换——subs函数

```
%替换上一题解中的a
>> sola = subs(sol,a,1)
>>solv = vpa(sola,15)
solv =
   -0.682327803828019
    0.34116390191401 - 1.161541399997 25i
    0.34116390191401 + 1.161541399997 25i
%指定替换为数值矩阵
>> syms a t
>> SA = subs(exp(a*t) + 1, a, -magic(3))
SA =
   [ exp(-8*t) + 1,   exp(-t) + 1, exp(-6*t) + 1]
   [ exp(-3*t) + 1, exp(-5*t) + 1, exp(-7*t) + 1]
   [ exp(-4*t) + 1, exp(-9*t) + 1, exp(-2*t) + 1]
```

```
>> syms x y a b
>> fh = a*b*x*y;
%指定替换为函数
>> fh1 = subs(subs(fh,x,sin(x)),y,exp(y))
fh1 =
   a*b*exp(y)*sin(x)
%嵌套替换为对应数值
>> fh2 = subs(subs(fh1,a,5),b,3)
fh2 =
   15*exp(y)*sin(x)
%绘制二维曲面图
>> ezmesh(fh2)
```




01
IDEA
02
LEARNING
03
CREATIVE
INSPIRATION
EDUCATION
BUSINESS
PHOTO BOOK

感谢聆听