

第三讲 MATLAB的符号运算

—— matlab 不仅具有数值运算功能，还开发了在matlab环境下实现符号计算的工具包Symbolic Math Toolbox

符号运算的功能

- 符号表达式、符号矩阵的创建
- 符号线性代数
- 因式分解、展开和简化
- 符号代数方程求解
- 符号微积分
- 符号微分方程

一、符号运算的基本操作

1. 什么是符号运算

- 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。

- 特点:

- ① 运算对象可以是没赋值的符号变量

- ② 可以获得任意精度的解

- 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. 符号变量与符号表达式

$f = \text{'sin(x)+5x'}$

f —— 符号变量名

sin(x)+5x —— 符号表达式

$'$ —— 符号标识

❖ 符号表达式一定要用 $'$ 单引号括起来matlab才能识别。

' '的内容可以是符号表达式，也可以是符号方程。

例：

$f1='a*x^2+b*x+c'$ —— 二次三项式

$f2='a*x^2+b*x+c=0'$ —— 方程

$f3='Dy+y^2=1'$ —— 微分方程

※符号表达式或符号方程可以赋给符号变量，以后调用方便；也可以不赋给符号变量直接参与运算

3.符号矩阵的创建

数值矩阵 $A=[1,2;3,4]$

$A=[a,b;c,d]$ —— 不识别

①用matlab函数sym创建矩阵 (symbolic的缩写)

命令格式: $A=sym(' [\quad]')$

※ 符号矩阵内容同数值矩阵

※ 需用sym指令定义

※ 需用' '标识

例如: $A = \text{sym}('[a, 2*b; 3*a, 0]')$

$A =$

$[\quad a, 2*b]$

$[3*a, \quad 0]$

这就完成了一个符号矩阵的创建。

注意: 符号矩阵的每一行的两端都有方括号, 这是与 matlab 数值矩阵的一个重要区别。

②用字符串直接创建矩阵

- ❖ 模仿matlab数值矩阵的创建方法
- ❖ 需保证同一列中各元素字符串有相同的长度。

例： $A = \begin{bmatrix} 'a, 2*b' \\ '3*a, 0' \end{bmatrix}$

$A =$

$\begin{bmatrix} a, 2*b \end{bmatrix}$

$\begin{bmatrix} 3*a, 0 \end{bmatrix}$

③ 符号矩阵的修改

a. 直接修改

可用↑、←键找到所要修改的矩阵，
直接修改

b. 指令修改

- ❖ 用 $A1 = \text{sym}(A, *, *, 'new')$ 来修改。
- ❖ 用 $A1 = \text{subs}(A, 'new', 'old')$ 来修改
- ❖ $A1 = \text{subs}(S, 'old', 'new')$

例如: $A = \begin{bmatrix} a, 2*b \\ 3*a, 0 \end{bmatrix}$

$A1 = \text{sym}(A, 2, 2, '4*b')$

$A1 = \begin{bmatrix} a, 2*b \\ 3*a, 4*b \end{bmatrix}$

$A(2,2) = '4*b'$

$A3 = \begin{bmatrix} a, 2*b \\ 3*a, 4*b \end{bmatrix}$

$A2 = \text{subs}(A1, 'c', 'b')$

$A2 = \begin{bmatrix} a, 2*c \\ 3*a, 4*c \end{bmatrix}$

④ 符号矩阵与数值矩阵的转换

❖ 将数值矩阵转化为符号矩阵

函数调用格式: `sym(A)`

```
A=[1/3,2.5;1/0.7,2/5]
```

```
A =
```

```
    0.3333    2.5000
```

```
    1.4286    0.4000
```

```
sym(A)
```

```
ans =
```

```
[ 1/3, 5/2]
```

```
[10/7, 2/5]
```



❖ 将符号矩阵转化为数值矩阵

函数调用格式: `numeric(A)`

`A =`

`[1/3, 5/2]`

`[10/7, 2/5]`

`numeric(A)`

`ans =`

0.3333 2.5000

1.4286 0.4000



二、符号运算

1. 符号矩阵运算

数值运算中，所有矩阵运算操作指令都比较直观、简单。例如： $a=b+c$;
 $a=a*b$; $A=2*a^2+3*a-5$ 等。

而符号运算就不同了，所有涉及符号运算的操作都有专用函数来进行

符号矩阵运算的函数：

`symadd(a,d)` —— 符号矩阵的加

`symsub(a,b)` —— 符号矩阵的减

`symmul(a,b)` —— 符号矩阵的乘

`symdiv(a,b)` —— 符号矩阵的除

`sympow(a,b)` —— 符号矩阵的幂运算

`symop(a,b)` —— 符号矩阵的综合运算

例1: f='2*x^2+3*x-5'; g='x^2+x-7';

h= symadd(f,g)

h=

$3*x^2+4*x-12$

例2: f='cos(x)'; g='sin(2*x)';

symop(f,'/',g,'+',f,'*',g)

ans =

$\cos(x)/\sin(2*x)+\cos(x)*\sin(2*x)$

例1: $f=2*x^2+3*x-5; g=x^2+x-7;$

>> `syms x`

>> `f=2*x^2+3*x-5; g=x^2+x-7;`

>> `h=f+g`

`h = 3*x^2+4*x-12`

例2: $f=\cos(x);g=\sin(2*x);$

>> `syms x`

>> `f=cos(x);g=sin(2*x);`

>> `f/g+f*g`

`ans =`

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

符号运算函数:

`symsize` —— 求符号矩阵维数

`charpoly` —— 特征多项式

`determ` —— 符号矩阵行列式的值

`eigensys` —— 特征值和特征向量

`inverse` —— 逆矩阵

`transpose` —— 矩阵的转置

`jordan` —— 约当标准型

`simple` —— 符号矩阵简化

2. 任意精度的数学运算

在symbolic中有三种不同的算术运算：

1. 数值类型 matlab的浮点算术运算
2. 有理数类型 maple的精确符号运算
3. vpa类型 maple的任意精度算术
运算

- 浮点算术运算

$1/2+1/3$ ——(定义输出格式format long)

ans =

0.8333333333333333

- 符号运算

$\text{sym}(1/2)+(1/3)$

ans =

$5/6$ ——精确解



3. 符号微积分与积分变换

- $\text{diff}(f)$ — 对缺省变量求微分
- $\text{diff}(f,v)$ — 对指定变量 v 求微分
- $\text{diff}(f,v,n)$ — 对指定变量 v 求 n 阶微分
- $\text{int}(f)$ — 对 f 表达式的缺省变量求积分
- $\text{int}(f,v)$ — 对 f 表达式的 v 变量求积分
- $\text{int}(f,v,a,b)$ — 对 f 表达式的 v 变量在 (a,b) 区间求定积分

`int` ('被积表达式', '积分变量', '积分上限',
'积分下限') —— 定积分

——缺省时为不定积分

`mtaylor(f,n)` —— 泰勒级数展开

`ztrans(f)` —— Z变换

`Invztrans(f)` —— 反Z变换

`Laplace(f)` —— 拉氏变换

`Invlaplace(f)` —— 反拉氏变换

`fourier(f)` —— 付氏变换

`Invfourier(f)` —— 反付氏变换

例1.计算二重不定积分 $\iint x e^{-xy} dx dy$

F=int(int('x*exp(-x*y)','x'),'y')

F=

1/y*exp(-x*y)

例2.计算 $f='x*\exp(-x*10)'$ 的Z变换

F=ztrans(f)

F=

z*exp(-10)/(z-exp(-10))^2

```
>> syms x y
```

```
>> F=int(int(x*exp(-x*y),x),y)
```

```
F =
```

```
1/y*exp(-x*y)
```

```
>> syms x
```

```
>> f=x*exp(-x*10);
```

```
>> F=ztrans(f)
```

```
>> F=ztrans(x*exp(-x*10);
```

```
F =
```

```
z*exp(-10)/(z-exp(-10))^2
```

例3. 计算指数函数 e^{At} 。

用拉氏反变换法计算 e^{At} 的公式为：

$$e^{At} = \mathbf{L}^{-1}[(\mathbf{S}\mathbf{I} - \mathbf{A})^{-1}]$$

系统矩阵 $\mathbf{A} = \begin{pmatrix} 0 & 1 \\ -2 & -3 \end{pmatrix}$

结果：

$$e^{At} = \begin{pmatrix} -e^{-2t} + 2e^{-t} & -e^{-2t} + e^{-t} \\ 2e^{-2t} - 2e^{-t} & 2e^{-2t} - e^{-t} \end{pmatrix}$$

```
>> a=[0 1;-2 -3];
```

```
>> syms s
```

```
>> b=(s*eye(2)-a)
```

```
b =
```

```
[ s, -1]
```

```
[ 2, s+3]
```

```
>> B=inv(b)
```

```
[ (s+3)/(s^2+3*s+2), 1/(s^2+3*s+2)]
```

```
[ -2/(s^2+3*s+2), s/(s^2+3*s+2)]
```

```
>> b11=ilaplace(sym(b,1,1));b(1,1)=b11;
```

```
>> b12=ilaplace(sym(b,1,2));b(1,2)=b12;
```

```
>> b21=ilaplace(sym(b,2,1));b(2,1)=b21;
```

```
>> b22=ilaplace(sym(b,2,2));b(2,2)=b22;
```

```
>> b
```

```
b =
```

```
[ -exp(-2*t)+2*exp(-t),    exp(-t)-exp(-2*t)]
```

```
[ -2*exp(-t)+2*exp(-2*t), 2*exp(-2*t)-exp(-t)]
```

4.符号代数方程求解

matlab符号运算能够解一般的线性方程、非线性方程及一般的代数方程、代数方程组。当方程组不存在符号解时，又无其他自由参数，则给出数值解。

命令格式：

`solve(f)` —— 求一个方程的解

`Solve(f1,f2, ...,fn)` —— 求n个方程的解

例1. $f = ax^2 + bx + c$ 求解

$f = 'a*x^2 + b*x + c';$

- `solve(f)` —— 对缺省变量 x 求解

`ans =`

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

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

计算机
格式

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

一般格式

- `solve(f, 'b')` —— 对指定变量**b**求解

`ans =`

$$-(a*x^2+c)/x$$

例2. 符号方程 $\cos(x)=\sin(x)$

$\tan(2*x)=\sin(x)$ 求解

`f1=solve('cos(x)=sin(x)'),`

`f1 =`

$$1/4*\pi$$


```
f2=solve('tan(2*x)=sin(x)')
```

```
f2 =
```

matlab4.2的解

```
[ 0]
```

```
[acos(1/2+1/2*3^(1/2))]
```

```
[acos(1/2 -1/2*3^(1/2))]
```

```
f3=
```

matlab4.2的解

```
[ 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]
```

numeric(f2)

ans =

0

0 + 0.8314i

1.9455

numeric(f3)

ans =

0

3.1416

0 + 0.8314i

0 - 0.8314i

1.9455

-1.9455

matlab4.2与6.1的对比

例3. 解方程组

$$\begin{cases} x+y+z=1 \\ x-y+z=2 \\ 2x-y-z=1 \end{cases}$$

`g1='x+y+z=1',g2='x-y+z=2',g3='2*x-y-z=1'`

`f=solve(g1,g2,g3)`

`f=solve('x+y+z=1','x-y+z=2','2*x-y-z=1')`

`f =`

`z = 5/6, y = -1/2, x = 2/3`

```
f=solve('x+y+z=1','x-y+z=2','2*x-y-z=1')
```

```
f =
```

```
x: [1x1 sym] f.x
```

```
ans = 2/3
```

```
y: [1x1 sym] f.y
```

```
ans = -1/2
```

```
z: [1x1 sym] f.z
```

```
ans = 5/6
```

```
[x,y,z]=solve('x+y+z=1','x-y+z=2','2*x-y-z=1')
```

```
x = 2/3
```

```
y = -1/2
```

```
z = 5/6
```

5. 符号微分方程求解

—— 用一个函数可以方便地得到微分方程的符号解

符号微分方程求解指令：`dsolve`

命令格式：`dsolve(f,g)`

- `f` —— 微分方程，可多至12个微分方程的求解；`g`为初始条件
- 默认自变量为 '`x`',可任意指定自变量'`t`', '`u`'等
- 微分方程的各阶导数项以大写字母`D`表示

$\frac{dy}{dt}$ 或 $\frac{dy}{dx}$ y的一阶导数——Dy

$\frac{d^2y}{dt^2}$ 或 $\frac{d^2y}{dx^2}$ y的二阶导数——D2y

$\frac{d^n y}{dt^n}$ 或 $\frac{d^n y}{dx^n}$ y的 n 阶导数——Dny

[y1,y2...]=dsolve(x1,x2,...xn) —— 返回
微分方程的解

一阶微分方程

```
dsolve('Dx=y','Dy=x','x(0)=0','y(0)=1')
```

```
ans =
```

```
x(t) = sin(t), y(t) = cos(t)
```

二阶微分方程

```
dsolve('D2y=-a^2*y','y(0)=1','Dy(pi/a)=0')
```

```
ans =
```

```
cos(a*x)
```

例3. $\frac{d^2 y}{dx^2} + 2 \frac{dy}{dx} + 2y = 0$

$y(0) = 1, \frac{dy}{dx}(0) = 0$ 求该方程的解

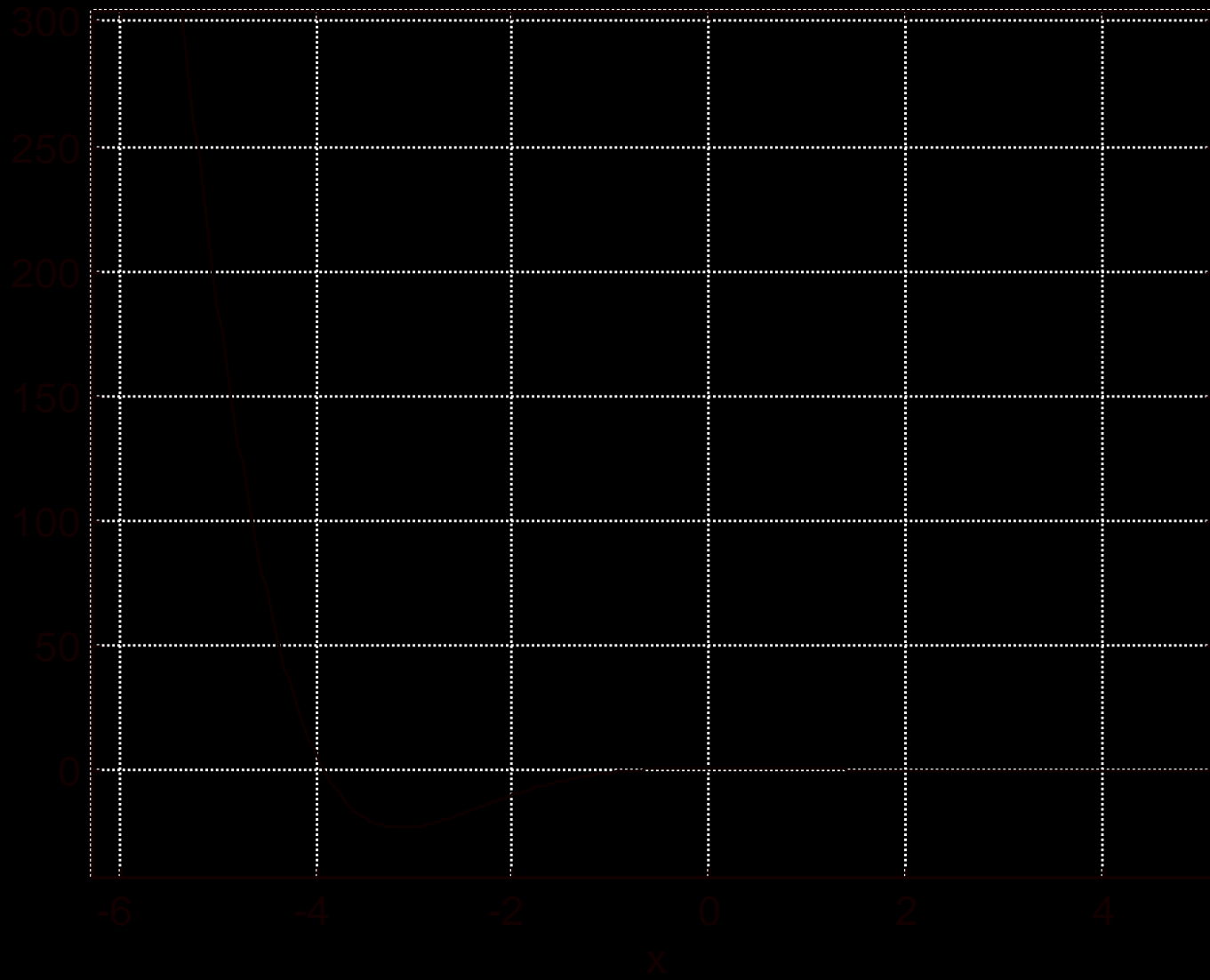
```
y=dsolve('D2y+2*Dy+2*y=0','y(0)=1','Dy(0)=0')
```

```
ans =
```

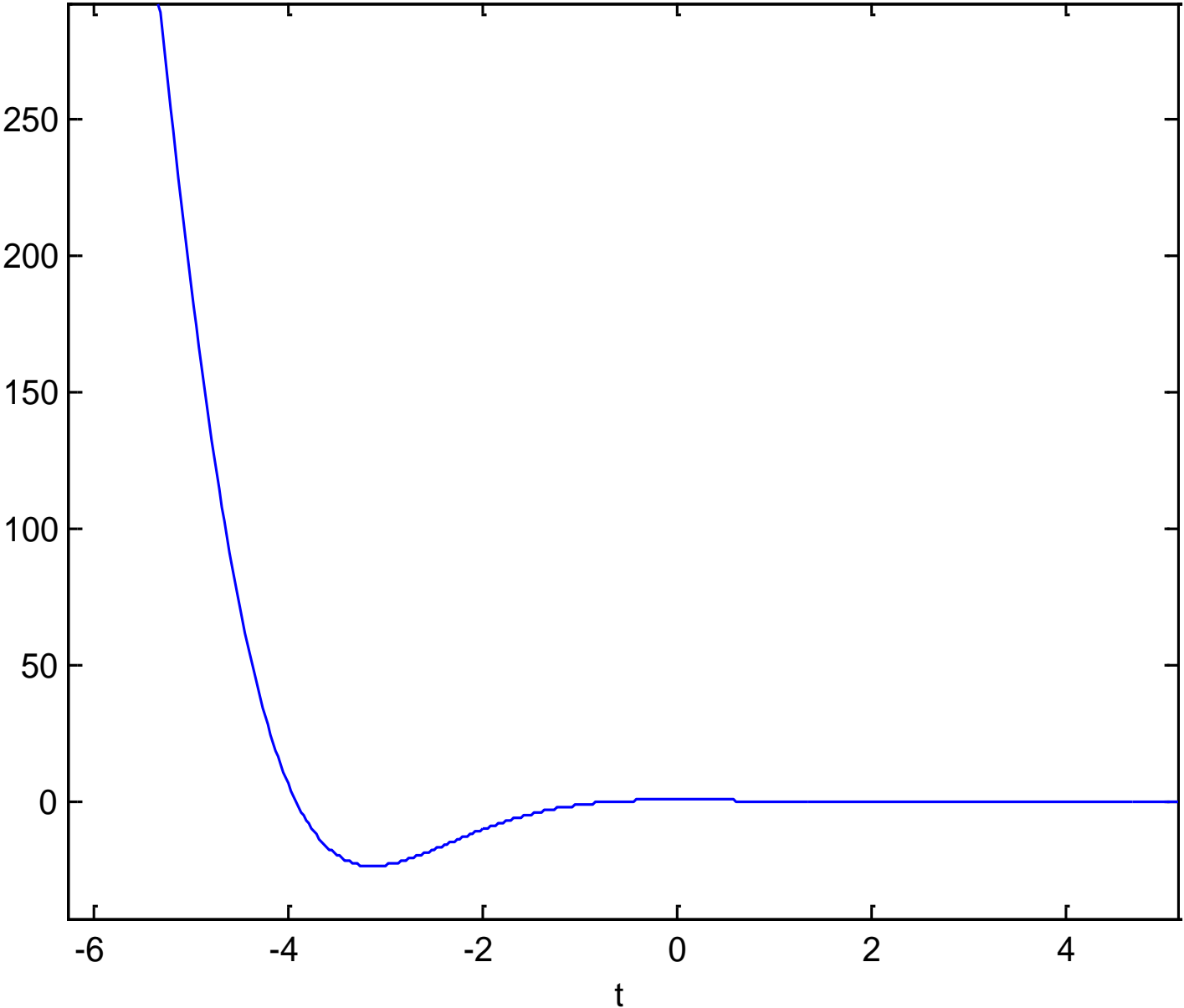
```
exp(-x)*cos(x)+exp(-x)*sin(x)
```

```
ezplot(y) —— 方程解y(t)的时间曲线图
```


$$\exp(-x) \cos(x) + \exp(-x) \sin(x)$$



$$\exp(-t) \sin(t) + \exp(-t) \cos(t)$$



三、maple函数——符号运算的扩展

maple——是专门进行数学运算的软件工具，具有超强的符号运算能力，提供了几乎包括所有数学领域的专用函数

matlab——依赖于maple的内核与函数库，扩展了自己的符号运算功能。

matlab还设计了对maple库函数的调用功能使得已有的maple数学功能，可以扩充matlab中,作为自身符号运算能力的扩展。

1. maple内核访问函数

可以访问maple内核的matlab函数:

maple —— 访问maple内核函数

mapleinit —— maple函数初始化

mpa —— maple函数定义

mhelp —— maple函数帮助命令

procread —— maple函数程序安装

①. maple 的调用格式

`maple('表达式')`

——将表达式送至maple内核，
返回符号表达式结果。

`maple ('函数', 变量1, 变量2)`

——调用maple函数，传递给定
变量。

例1. 展开5阶 bernoulli 多项式，计算 $x=3$ 时 bernoulli 数。

```
a:=maple('bernoulli(5,x)')
```

```
a =
```

```
-1/6*x+5/3*x^3+x^5-5/2*x^4
```

```
a:=maple('bernoulli(5,3)')
```

```
a =
```

```
85
```

例2. 化简三角函数式 $\sin^2x+\cos^2x$

```
a:=maple('simplify(sin(x)^2+cos(x)^2);')
```

a =

1

例4. 求 $f(t)=e^{-3t}\sin t$ 的拉式变换

```
f:=maple('laplace(exp(-3*t)*sin(t),t,s);')
```

f =

$1/((s+3)^2+1)$

例4. 寻找二次多项式的完全平方

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

```
a=maple('completesquare(x^2+2*x+2)')
```

```
a =
```

```
completesquare(x^2+2*x+2)
```



将工具包
装入内存

```
maple('with(student);')
```

```
a=maple('completesquare(x^2+2*x+2)')
```

```
a =
```

```
(x+1)^2+1
```


- maple软件中的所有函数，在初始化时并没有完全装入内存，可用readlib指令把库函数读入内存，或用with指令将应用工具包装入内存。
- 调用格式

```
maple('readlib(函数名);')
```

```
maple('with(工具包名);')
```

例5. 求 $\sin(x^2+y^2)$ 在 $x=0, y=0$ 处泰勒级数展开式，8阶截断。

```
maple('mtaylor(sin(x^2+y^2),[x=0,y=0],8)')
```

```
ans =
```

```
mtaylor(sin(x^2+y^2),[x = 0, y = 0],8)
```

```
maple('readlib(mtaylor);')
```

```
maple('mtaylor(sin(x^2+y^2),[x=0,y=0],8)')
```

```
ans =
```

```
x^2+y^2-1/6*x^6-1/2*y^2*x^4-1/2*y^4*x^2-  
1/6*y^6
```

2. mpa —— maple变量定义

- 任何一个matlab定义的函数f，可使用mpa语句直接调用，还可把f定义成maple变量v。
- maple的工作空间与matlab工作空间是相互独立的，所以f与v是属于不同工作空间中的变量
- mpa的调用格式：

`mpa('v',f)`

`mpa v f`

f为matlab工作空间中已存在的变量

例. 电磁力计算公式为 $F(I, x) = \frac{S\mu_0(NI)^2}{4x^2}$

试 $I=0.5$, $x=0.1$ 邻域展开泰勒级数, 3阶截

断, 令常数 $k = \frac{S\mu_0 N^2}{4}$, $F(I, x) = \frac{kI^2}{x^2}$

1. 直接调用

```
maple('readlib(mtaylor);')
```

```
maple('mtaylor(k*I^2/x^2,[I=0.5,x=0.1],3);')
```


注意： matlab符号运算时，可以识别matlab定义的符号变量，但在调用 maple 函数时，需将matlab变量定义为maple变量后，所调用的函数方可识别和执行

3.mhelp —— maple函数帮助命令

① mhelp 是协助检索maple库函数的专用命令

调用格式：mhelp 相关词条 →

工具词条
函数词条

例如：

mhelp intro —— maple介绍

mhelp maple —— maple命令格式

mhelp tutorial ——maple入门

mhelp index ——maple检索

mhhelp index 用于工具包检索

library —— maple标准库函数

packages —— 应用工具包

libmisc —— 其它库函数

statements —— maple语句描述

expressions —— maple表达式

datatypes —— maple数据格式

tables —— maple表格和阵列

procedures —— maple程序

misc —— maple其它应用

一般帮助文本主要包括以下部分

- FUNCTION—— 函数功能说明
- CALLING SEQUENCE—— 调用格式
- PARAMETERS —— 调用参数说明
- SYNOPSIS —— 语法说明
- EXAMPLES —— 应用举例
- SEE ALSO —— 相关词条

4.maple库函数

maple库函数共分四类

①maple内部函数：驻留函数任何条件下都可调用

mhhelp index[internal]

② maple的外部函数—读库定义部分：
调用时先执行读库命令，因此与内部函数一样可直接调用

mhhelp index[external]

③ maple的外部函数—读库装入部分
maple其余外部函数需要在执行前执行
`maple('readlib(函数名);')`命令将其
装入内存

mhhelp index[libmisc]

④ maple的惰性函数—不能直接调用，
还需一些函数如`mod`,`evala`,`evalf`等才能调用

mhhelp index[intert]

小 结

本节介绍了matlab语言的符号运算功能，通过学习应该掌握：

- 掌握如何创建、修改符号矩阵
- 掌握符号运算功能
- maple函数调用
- mhelp检索