MATLAB 语言及应用

MATLAB Language and Its Applications

第五章 符号计算
Ch.5 Symbolic Calculus

刘世东
Shidong Liu

School of Physics and Physical Engineering
Qufu Normal University

May 24, 2021

本章目录

MATLAB 符号计算的基本说明

MATLAB 本身以数值计算为引擎驱动, 并没有符号演算的能力. 2008 年之前, MATLAB 的符号计算以 Maple 为引擎, 2008 年以后改用 MuPad. 因使用了不同的符号引擎, 导致 MATLAB 高低版本之间的符号计算可能不一致.

MATLAB 符号计算是通过集成在 MATLAB 中的符号数学工具箱（Symbolic Math Toolbox）来实现的。和别的工具箱有所不同，该工具箱不是基于矩阵的数值分析，而是使用字符串来进行符号分析与运算。

MATLAB 的符号数学工具箱可以完成几乎所有的符号计算功能: 符号表达式的计算、复合、化简，符号矩阵的计算，符号微积分、符号函数画图，符号代数方程求解，符号微分方程求解等。

符号计算比数值计算耗费资源, 故无特别要求, 实际使用过程中最好采用数值计算避免符号计算.

符号对象的创建

创建单个变量 MATLAB 提供了两种变量定义方法 sym 和 syms. 其中 syms 是 sym 的 shorthand 法.

```
syms x
y = sym('y')
```

第一句创建了 x (符号) 变量并存储在工作空间 (其值为 x): (准确的说法: 创建了符号 x, 存储在变量 x 中, 但是我们往往就是称作符号变量.)

第二句创建 y (符号) 变量, 其值为 y; (若没有等号前面的 y, 则创建符号变量 ans, 其值为 y)

创建多个变量

用 syms, 可以一次性创建多个符号变量, 例如: a, b, and c.

```
syms a b c % 只能用空格
```

当创建的符号变量有多个值时, 使用 syms 不方便, 可以使用 sym, 如创建符号变量 A: 值为 a1, a2, ..., a20.

```
A = sym('a', [1 20])
```

创建符号数值/字

```
sym(1/3) % 创建了符号数字 1/3 结果为 ans = 1/3
consNum = sym(1/3) % 符号数字 1/3，赋值给符号变量consNum = sym(1/3)
```

sym创建符号数字时，可以没有引号！符号数字只能用sym创建. 若在命令窗口中直接输入 1/3，结果为 ans = 0.3333. 符号数字是精确的值. 如

```
s1 = sin(sym(pi)) % 之前学的函数可以直接对符号数值进行运算，如果能给出精确值的话
s2 = sin(pi)
%运行结果
s1 = 0 % 注意 s1 是符号变量
s2 = 1.2246e-16 % s2 是数值
% 通过whos看它们的属性
```

数值/变量转化为符号使用sym即可. 例如:

```
t = 0.1;
sym(t) % 注意没有引号，没有引号代表转化
%运行结果，是数值的一个近似的精确符号值.
ans =
1/10
```

创建符号表达式:

创建常数表达式 $\phi = (1 + \sqrt{5})/2$

```
phi = (1 + sqrt(sym(5)))/2; % 符号与双精度数字的混合运算
```

可以是表达式中的任意一个数字符号化, 例如 `phi = (sym(1)+ sqrt(5))/2;` 是等价的. 在以后的符号计算中, 可以使用 phi 精确的表达式黄金分割比. 如, 计算 `phi^2` 或 `phi+1` 等等

```
syms a b c x % 创建变量
f = a*x^2 + b*x + c; % 创建表达式
```

创建符号数值或者符号表达式的数值为常数时, 用 sym. (Tip To create a symbolic number, use the sym command. Do not use the syms function to create a symbolic expression that is a constant. For example, to create the expression whose value is 5, enter f = sym(5). The command f = 5 does not define f as a symbolic expression.)

NOTE: 符号表达式不同于符号函数, 符号函数可以作为函数进行运算, 符号表达式本质上是一个整体的符号变量.

创建符号函数

符号函数的创建可以使用syms:

```
syms f(x, y)
```

此语句同时定义了符号函数 f 和符号变量 x 和 y. 可分两步进行:

```
syms x y % 创建符号变量
f(x, y) = x^3*y^3 % 赋值给函数f, 同时创建了符号函数f
```

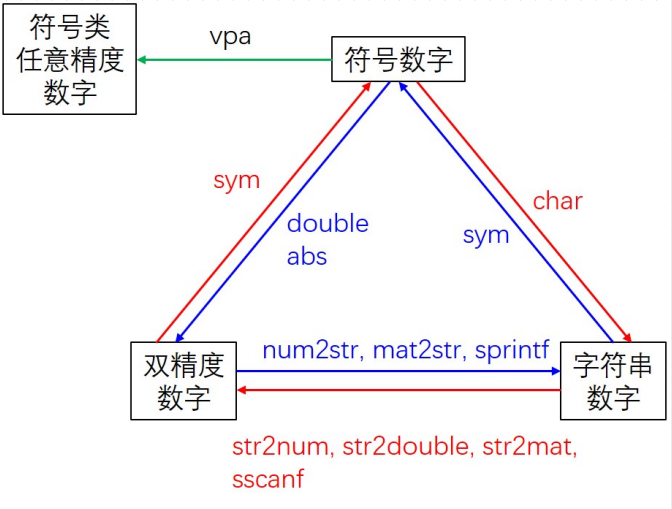
可以比较符号表达式的区别 `f = x^3*y^3`.

创建符号函数之后, 可以对其进行微分, 积分, 化简等操作. 例如

```
d2fy = diff(f, y, 2)
%运行结果
d2fy(x, y) = % 注意: d2fy 也是符号函数
6*x^3*y
```

```
f(y + 1, y) % x=y+1 代入符号表达式 try f(1,1)
ans = %运行结果
y^3*(y + 1)^3
```

转换指令


符号类
任意精度
数字
vpa
符号数字
sym
double
abs
sym
char
双精度
数字
num2str, mat2str, sprintf
字符串
数字
str2num, str2double, str2mat,
sscanf

符号数字与双精度数字之间的转换

```
% 定义符号数字
sa = sym(1/3); % 精确数字，不谈精度
sb = sym(pi)+sqrt(5);
% 定义双精度数字，有效数字为16位
da = 1/3;
db = pi + sqrt(5);
% 数值转符号，用最接近的有利分数表示
da2sa = sym(da)
db2sb = sym(db)
%运行结果
da2sa = 1/3
db2sb = 189209612611719/35184372088832 % 注意这个分数...
% 符号数字到双精度数字
sa2da = double(sa);
sb2da = double(sb);
%运行结果
sa2da = 0.3333
sb2da = 5.3777
```

符号数字与字符数字之间的差异

```
% 定义符号数字
sa = sym(1/3); % 精确数字，不谈精度
sb = sym(pi)+sqrt(5);
% 定义字符数字
stra = '1/3'; % 里面的任何字符都是字符串的元素，包括空格
strb = 'pi + sqrt(5)';
% 字符串转符号，用最接近的有利分数表示
stra2sa = sym(stra)% 这种转换已经不再建议使用，会warning
strb2sb = sym(strb)
%运行结果
stra2sa = 1/3
strb2sb = pi + 5^(1/2)
```

字符与数值之间的转换从略...

符号数字的任意精度表达式

数值在计算过程中会产生截断误差, 并且这种误差会不算累计;

符号计算是在完全准确的情况下进行的, 但是计算资源大, 从而耗时. 有时为了加快速度, 可以将符号计算转化为变精度计算.

MATLAB 存在以下相关命令

- `digits` 显示当前环境下十进制符号数字的有效位数
- `digits(n)` 把十进制符号数字有效位数设定为 n
- `xs = vpa(x)` 根据表达式 x 得到 `digits` 指定精度下的符号数字 xs, xs 仍然是符号数字, 虽然形式上跟普通数值数字之间没有区别
- `xs = vpa(x,n)` 根据表达式 x 得到 n 位有效数字的符号数字 xs

```
sa = sym(1/3);
xa = vpa(sa)
% 运行结果
xa = 0.33333333333333333333333333333333
```

操作指令

Choose Function to Rearrange Expression

Type of Transformation	Function
Combine Terms of Same Algebraic Structures	combine
Expand Expressions	expand
Factor Expressions	factor
Extract Subexpressions from Expression	children
Collect Terms with Same Powers	collect
Rewrite Expressions in Terms of Other Functions	rewrite
Compute Partial Fraction Decompositions of Expressions	partfrac
Compute Normal Forms of Rational Expressions	simplifyFraction
Represent Polynomials Using Horner Nested Forms	horner

有些教材中的指令是早期版本的, 新版本 MATLAB 有的已经完全停止使用或者赋予其他含义, 有的已经不建议使用.

符号表达式的显示 pretty

符号表达式的显示，默认采用 MATLAB 式的显示。`pretty` 允许用户将符号表达式显示为符合一般数学表达习惯的数学表达式。

`pretty` is not recommended. Use Live Scripts instead. Live Scripts provide full math rendering while pretty uses plain-text formatting. BUT, Live Scripts are expensive.

```
syms a b c d e f g x
f=a*x^3+b*x^2+c*x+d+e*x^-1+f*x^-2+g*x^-3
pretty(f)
% pretty 结果
                3       2    e     f     g
d + c x + a x  + b x  + - + -- + --
                        x     2     3
                             x     x
% 也不好看且不是很便于辨认
```

组合相同的代数结构 combine

$\boxed{Y = \text{combine}(S)}$: rewrites products of powers in the expression S as a single power.

```
syms x y z
combine(x^y*x^z) % 结果 ans =x^(y + z) 合并了相同的底数
```

```
syms x y
combine(x^(3)*x^y*x^exp(sym(1))) % ans = x^(y + exp(1) + 3)
```

符号表达式合并 collect

R=collect(S,x)：将表达式 S 中 x 的相同次幂的项合并。S 可以是一个表达式，也可以是一个符号矩阵，返回表达式 R.

```
syms x
f = (x-1)^3
collect(f)
% 运行结果
ans = x^3 - 3*x^2 + 3*x - 1 % 相当于进行了展开
```

```
syms x y t
f2=x*cos(t)+y*sin(t)+(x*2+2*x*y+3*y*2)*t;
collect(f2,x)
%运行结果
ans = (cos(t) + t*(2*y + 2))*x + 6*t*y + y*sin(t)

% try change x into y
```

符号表达式展开

R=expand(S)：将表达式 S 中的各项进行展开。如果 S 包含函数，则利用恒等变形将它写成相应的和的形式。该函数多用于多项式，有时也用于三角函数、指数函数和对数函数。

```
syms x y
f1=(x-1)^2*(y-1);
expand(f1)
% 运行结果
ans = 2*x + y - 2*x*y + x^2*y - x^2 - 1
```

```
syms x
f = sin(x+y);
expand(f)
 % 运行结果
ans = cos(x)*sin(y) + cos(y)*sin(x)
```

符号表达式嵌套 horner

horner(P): 将多项式进行嵌套表达.

```
syms x
horner(x^3 - 6*x^2 + 11*x - 6)
%运行结果
ans = x*(x*(x - 6) + 11) - 6
```

公因子提取置换 subexpr

[r,sigma] = subexpr(expr): 重写 expr 符号表达式为 r, sigma 为公因子. 注意, 此时 expr 中不应该含有 sigma 变量 (sigma 是默认的公因子).

[r,var] = subexpr(expr,'var'): 同上, 区别只是公因子变量为 var.

```
syms a b c
f = [-(b + (b^2 - 4*a*c)^(1/2))/(2*a); -(b - (b^2 - 4*a*c)^(1/2))/(2*a)];
% 二元一次方程的通解
[newf, sigma] = subexpr(f)
% 运行结果
newf = [-(b + sigma)/(2*a); -(b - sigma)/(2*a)]
sigma = (b^2 - 4*a*c)^(1/2)
```

采用 [r, var] = subexpr(expr,'var')形式

```
[newf2, gongYinZi] = subexpr(f, 'GongYinZi')
% 运行结果
newf2 = [-(GongYinZi + b)/(2*a); (GongYinZi - b)/(2*a)]
gongYinZi = (b^2 - 4*a*c)^(1/2)
```

通用置换命令 subs

subs(s,old,new): 将表达式 s 中的old 部分(注意不一定是变量) 用 new 替换, 并进行计算.

变量替换

```
syms a b
f = a + b;
subs(f, a, 4)
% 运行结果
ans = b + 4
```

部分置换

```
syms x y
f2 = x*y^2;
subs(f2, x*y, 5)
% 运行结果
ans = 5*y % NOT 25
```

多个替换

```
syms a b
f3 = sin(a) + cos(b);
subs(f3,[a,b],[sym('xyz'),6])
% 运行结果
ans = cos(6) + sin(xyz)
```

- 注意替换数字和符号变量的区别: 符号变量须先定义.
- 多个替换时数组也可以用胞元. 即subs(f3,{a,b},{sym('xyz'),6}).
- 当将第 3 例中的 a 和 b 均替换为 pi 时, 给出的结果是 -1 (subs 要进行计算, 但是计算的结果仍然是符号).

More see doc and Related Examples there!

极限 limit

MATLAB 中，用函数limit来求表达式的极限.

limit(expr,x,a): 求表达式 expr 中 x 趋于 a 的极限值.

limit(expr,x,a,'left'): 求表达式左趋近 a 的极限值. 右趋近, 则 left 换成right. 一般极限求算.

```
syms x h
lim1 = limit(sin(x)/x) % 可以先定义 f = sin(x)/x
lim2 = limit((sin(x + h) - sin(x))/h, h, 0) % 其实就是求sin(x)的导数.
%运行结果
lim1 = 1
lim2 = cos(x) % 注意 lim1 和 lim2 都是符号
```

左右求导的区别

```
syms x
limr = limit(1/x, x, 0, 'right')
limf = limit(1/x, x, 0, 'left')
% 运行结果 (我重新手动排版了)
limr = inf      liml = -inf
```

导数 diff

diff 可以完成一元或多元函数的任意阶数的微.

diff(f,x,n): 对表达式 x 自变量进行 n 阶求导. n 缺省时为 1.

diff(f, x1, x2,.., xn) 对表达式按 xi 顺序一次求导 (可以用来 n 阶求导).

$$f = \exp(x \sin y) + \ln z$$

一. 只对一个变量求导

```
syms x y z % 一阶求导
f=exp(x*sin(y))+log(z);
dfx = diff(f, x)
dfy = diff(f, y)
dfz = diff(f, z)
% 运行结果
dfx = exp(x*sin(y))*sin(y)
dfy = x*exp(x*sin(y))*cos(y)
dfz = 1/z
```

```
syms x y z % 二阶求导
f=exp(x*sin(y))+log(z);
dfx2 = diff(f, x ,2)
dfy2 = diff(f, y, 2)
dfz2 = diff(f, z, 2)
% 运行结果
dfx2 = exp(x*sin(y))*sin(y)^2
dfy2 = x^2*exp(x*sin(y))*cos(y)^2-x*exp(x*sin(y))*sin(y)
dfz2 = -1/z^2
```

二. 多个变量求导: 仍然针对上一个表达式

```
syms x y z % 二阶求导
f=exp(x*sin(y))+log(z);
dfx2 = diff(f, x ,2)
dfxx = diff(f, x, x)
twoEqual = isequal(dfx2, dfxx)
% 运行结果
dfx2 = exp(x*sin(y))*sin(y)^2
dfxx = exp(x*sin(y))*sin(y)^2
twoEqual = 1 % 是逻辑数  logical
```

注意 `dfx2 = diff(f, x ,2)` 和 `dfxx = diff(f, x, x)` 结果是一样的..

```
dfxy = diff(f, x, y)
dfxyz = diff(f, x, y, z)
% 运行结果
dfxy = exp(x*sin(y))*cos(y) + x*exp(x*sin(y))*cos(y)*sin(y)
dfxyz = 0
```

雅可比行列式求导

在多元偏微分中经常使用雅可比行列式求导

$$J = \frac{\partial(x\,,y\,,z)}{\partial(r\,,\lambda\,,\phi)} = \begin{vmatrix} \frac{\partial x}{\partial r}, \frac{\partial x}{\partial \lambda}, \frac{\partial x}{\partial \phi} \\ \frac{\partial y}{\partial r}, \frac{\partial y}{\partial \lambda}, \frac{\partial y}{\partial \phi} \\ \frac{\partial z}{\partial r}, \frac{\partial z}{\partial \lambda}, \frac{\partial z}{\partial \phi} \end{vmatrix}$$

jacobian(f,v): f 对 v 求雅可比矩阵. 第 (i,j) 个元素为 $\partial f(i)/\partial v(j)$.

```
syms x y z
f = [x*y*z, y^2, x + z];% 可以为行向量, 也可以为列向量
v = [x, y, z]; % 同上
jacobian(f,v)
%运行结果
ans =
[ y*z, x*z, x*y]
[   0, 2*y,   0]
[   1,   0,   1]
```

符号积分 int More see doc int

符号的定积分和不定积分函数为 int

一. 一元积分: int(expr,var,a,b) 对表达式 expr 中的变量 var 进行积分, 积分区间为 [a,b]. [a,b] 缺省时为不定积分, 积分结果中不带常数.

```
syms x
f = 2*x +1;
fb = int(f,x)
fd = int(f,0,1)
% 运行结果
fb = x*(x + 1) % 不定积分结果不带积分常数
fd = 2 % 此结果为符号
```

无法给出解析积分结果的, 不进行计算, 可以用vpa进行 '数值' 显示:

```
syms x
F = int(cos(x)/sqrt(1 + x^2), x, 0, 10) % 运行结果是自身, 如下
F = int(cos(x)/(x^2 + 1)^(1/2), x, 0, 10)
% 为得到结果可以进行数值积分, 可以用vpa进行数值计算
vpa(F,5) % 5位有效数字, 结果 ans = 0.37571
```

二. 多重积分

$$I=\int_1^2\int_{\sqrt{x}}^{x^2}\int_{\sqrt{xy}}^{x^2y}(x^2+y^2+z^2)\,\mathrm{d}z\,\mathrm{d}y\,\mathrm{d}x$$

```
syms x y z
f = x^2 + y^2 + z^2;
% 三步法
Iz = int(f,z, sqrt(x*y), x^2*y);
Iy = int(Iz, y, sqrt(x),x^2);
I = int(Iy, x, 1, 2)
% 一步法
I = int(int(int(f,z, sqrt(x*y), x^2*y),y, sqrt(x),x^2),x,1,2)
% 通过vpa得到最后的值.
vpa(I)
% 运行结果
I = (14912*2^(1/4))/4641 - (6072064*2^(1/2))/348075 + (64*2^(3/4))/225 + ...
    1610027357/6563700

ans = 224.92153573331143159790710032805 %这是一个符号
```

上题积分的数值积分形式

```
f = @(x,y,z)x.^2+y.^2+z.^2;
ymin = @(x)sqrt(x);
ymax = @(x)x.^2
zmin =@(x,y)sqrt(x.*y);
zmax = @(x,y)x.^2.*y;
I = integral3(f,1,2,ymin,ymax,zmin,zmax)
% 运行结果
I = 224.9215 % 这是一个数值   y

% 与符号积分结果对比: 224.92153573331143159790710032805
```

泰勒级数展开 taylor

taylor(f,var,a) 表达式 f 对变量 var=a 处泰勒级数展开, a 缺省则在 0 处展开, 默认展开到 6 阶

```
syms x
taylor(exp(x)) %运行结果 ans = x^5/120 + x^4/24 + x^3/6 + x^2/2 + x + 1
taylor(sin(x)) % 运行结果 ans = x^5/120 - x^3/6 + x
taylor(sin(x),1) % 运行结果如下:
ans = sin(1) - (sin(1)*(x - 1)^2)/2 + (sin(1)*(x - 1)^4)/24 + cos(1)*(x - 1) - ...
    (cos(1)*(x - 1)^3)/6 + (cos(1)*(x - 1)^5)/120
```

修改阶数 taylor(f,var,'ExpansionPoint', a, 'Order',n): a 处 n 阶 taylor 级数展开.

```
syms x
taylor(sin(x), 'ExpansionPoint', 0, 'Order',8)
% 运行结果 ans = - x^7/5040 + x^5/120 - x^3/6 + x
```

多元泰勒级数展开

```
syms x y
f = y*exp(x - 1) - x*log(y);
taylor(f, [x, y], [1, 1], 'Order', 3) % 运行结果 ans = x + (x - 1)^2/2 + (y - 1)^2/2
```

级数求和

F = symsum(f,k,a,b): 计算级数的和, 级数由 f 表达, 对变量 k 求和, 求和区间为 [a,b].

```
syms k x
S1 = symsum(k^2, k, 0, 10)
S2 = symsum(1/k^2, k, 1, Inf)
S3 = symsum(x^k/factorial(k), k, 0, Inf) % factorial 表示阶乘
% 运行结果
S1 = 385
S2 = pi^2/6
S3 = exp(x)
```

More see

Choose Function to Rearrange Expression

Numeric to Symbolic Conversion

Operators and Elementary Operations

Conversion Between Symbolic and Numeric

Symbolic Computations in MATLAB

Symbolic Math Toolbox

Getting Started with Symbolic Math Toolbox

它们的周边等等...

微分方程的符号解

S = dsolve(eqn,cond): 求解微分方程 (组) eqn, eqn 是符号等式, 用 diff 函数和 == 表示. cond 表示初始条件或者边界条件, 可以缺省. S 表示返回的解 (结构体), 也可以写为 [y1,..,yN]

一. 一阶微分方程: 符号求解 $\mathrm{d}y/\mathrm{d}t = ay$, 不给定初始条件.

```
syms a y(t) %定义符号变量a 和 符号函数 y(t), 其自变量为 t.
eqn = diff(y,t) == a*y; % eqn是一个变量 表示的是微分表达式 diff(y,t) == a*y
dsolve(eqn)
% 运行结果
ans = C1*exp(a*t) % 注意运行结果给出的是精确解, 因为本题没有给出初始条件, 故解有常数
```

若上述方程初值为 $y(t=0)=5$, 则

```
syms y(t) a
eqn = diff(y,t) == a*y; cond = y(0) == 5; % 定义微分方程和初始条件, 注意时 == 而不是 =
ySol(t) = dsolve(eqn,cond) % 运行结果如下, 这样求得的结果是符号函数, 可求值 ySol(2)
ySol(t) = 5*exp(a*t) % 也可以让结果是符号表达式, 即上一步为 ys
```

NOTE: 初始条件和微分方程的定义都是用 ==

二. 高阶微分方程 $\mathrm{d}^2y/\mathrm{d}t^2 = ay$

```
syms y(t) a
eqn = diff(y,t,2) == a*y; % 也可以是 diff(y,t,t) == a*y
ySol(t) = dsolve(eqn)
ySol(t) = C5*exp(a^(1/2)*t) + C6*exp(-a^(1/2)*t)
```

上述方程初始条件 $y(0) = 1, y'(0) = 2$

```
syms y(t) a
eqn = diff(y,t,2) == a*y;
dy = diff(y,t); % 可以任意定义变量名，并非必须dy. 曾记否: diff后仍是符号函数
cond1 = y(0) == 1;
cond2 = dy(0) == 2;
ySol(t) = dsolve(eqn, cond1, cond2)
% 运行结果
ySol(t) = (exp(a^(1/2)*t)*(a^(1/2) + 2))/(2*a^(1/2)) + (exp(-a^(1/2)*t)*(a^(1/2) - ...
    2))/(2*a^(1/2))
```

三. 非线性微分方程: $(\mathrm{d}y/\mathrm{d}t + y^2)^2 = 1$, 初值 $y(0) = 0$

```
syms y(t)
y(t) = dsolve((diff(y,t) + y^2)^2 == 1, y(0) == 0)
% 运行结果, 给出两个解
y(t) =
tanh(t)
-tan(t)
```

四. 微分方程组 $\mathrm{d}x/\mathrm{d}t = y, \quad \mathrm{d}y/\mathrm{d}t = -x$

```
syms x(t) y(t)
eq1 = diff(x,t) == y;
eq2 = diff(y,t) == -x;
S = dsolve(eq1,eq2) % 运行结果如下
S = % S是结构体 可以通过S.x和S.y查看相应的表达式
struct with fields:
y: [1×1 sym]
x: [1×1 sym]
```

四. 微分方程没有解或者没有解析解时，给出提示，返回空符号, 如

```
syms y(x)
dsolve(exp(diff(y)) == 0)
%运行结果如下
Warning: Explicit solution could not be found.
> In dsolve (line 201)
ans = [ empty sym ] % 结果为空
```

符号解微分方程 (组) 的一般步骤

1. 创建符号函数即变量
2. 定义微分方程, 并指定初值条件或者边界条件
3. dsolve 求解

另: 得到的函数解可以用fplot直接画图.

doc dsolve

More About: Solve a Single Differential Equation (后有例题可用)

代数方程的符号解

S = solve(eqn,var): 解方程 (组) eqn, 自变量 (们) 为 var.

一. 代数方程

方程 $3 * x = 9$.

```
syms x
eq = 3*x==9;
solve(eq,x) % 运行结果 ans = 3
```

给出精确解 3.

方程 $ax^2 + bx + c = 0$

```
syms x a b c
solve(a*x^2+b*x+c==0, x) %运行结果
ans =
-(b + (b^2 - 4*a*c)^(1/2))/(2*a)
-(b - (b^2 - 4*a*c)^(1/2))/(2*a)
```

非线性或超越方程一般无法给出解析解, 结果用特殊函数表示或近似数值

例如方程 $x + e^x = 2$

```
syms x
x0 = solve(x+exp(x)==2) %运行结果
x0 = 2 - lambertw(0, exp(2)) % lambertw 特殊函数: 朗伯W函数 (Lambert W Function)
xnum = vpa(x0) % 结果ans = 0.44285440100238858314132799999934
```

可以通过 double(xnum)+ exp(double(xnum))- 2 判断结果 (运行结果为 0)

二. 代数方程组

```
syms x y a
[solx,soly] = solve(x^2*y^2 == 0, x-y/2 == a)
%运行结果，两组解
solx =
0
a
soly =
-2*a
0
```

doc solve Related Examples

傅里叶变换及其逆变换

傅里叶变换常应用于信号处理科学. 傅里叶变换的数学表达式

$$\begin{cases} F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t}\,\mathrm{d}t \\ f(t) = \dfrac{1}{2\pi}\int_{-\infty}^{\infty} F(\omega) e^{i\omega t}\,\mathrm{d}t \end{cases}$$

完成变换的方法有两种, 一种是直接利用符号积分进行计算, 另一种是理论 MATLAB 内置的专门函数 fourier 和 ifourier 进行计算.

一. 傅里叶变换: fourier(f,var,transVar): 对函数 f 进行傅里叶变换, var 是自变量, transVar 是变换变量.

```
syms t a w
assume(a > 0);
f = exp(-a*abs(t)); % 对称指数函数的傅里叶变换
fourier(f,t,w)
% 运行结果
ans = (2*a)/(a^2 + w^2)
```

二. 傅里叶逆变换: `ifourier (F,var,transVar)`: 对函数 F 进行傅里叶逆变换, var 是自变量, transVar 是变换变量.

对上述傅里叶变换作逆变换:

```
syms t, a, w
assume(a>0)
F = (2*a)/(a^2 + w^2)
ifourier(F,w,t)
% 运行结果
ans = exp(-a*abs(t))
```

拉普拉斯变换及其逆变换

拉普拉斯变换的数学定义

$$\begin{cases} F(s) = \int_0^{\infty} f(t)e^{-st}\,\mathrm{d}t \\ f(t) = \dfrac{1}{2\pi j}\int_{c-i\infty}^{c+i\infty} F(s)e^{st}\,\mathrm{d}s \end{cases}$$

完成拉普拉斯变换同样有两种方法, 积分 int 和内置专门函数 laplace 和 ilaplace

一. 拉普拉斯变换: laplace(f,var,transVar): 对函数 f 进行拉普拉斯变换, var 是自变量, transVar 是转换变量.

```
syms t a s
f = exp(-a*t); % 对指数衰减进行拉普拉斯变换
laplace(f,t,s)
% 运行结果
ans = 1/(a + s)
```

二. 拉普拉斯逆变换: `ilaplace(F,var,transVar)`: 对函数 F 进行拉普拉斯逆变换, var 是自变量, transVar 是变换变量.

对上述拉普拉斯变换作逆变换:

```
syms t a s
assume(t>0)
F = 1/(a + s);
ilaplace(F,s,t)
% 运行结果
ans = exp(-a*t)
```

Z 变换及其逆变换

一个离散因果序列的 Z 变换及其逆变换为

$$\begin{cases} F(z) = \sum_{n=0}^{\infty} f(n) z^{-n} \\ f(n) = Z^{-1}\{F(z)\} \end{cases}$$

涉及 Z 逆变换的方法有很多, 常见的有三种: 幂级数展开法, 部分分式展开法和围线积分法. MATLAB 采用的是围线积分法. 相应的数学表达式为 $f(n) = \frac{1}{i2\pi} \oint F(z) z^{n-1} \, \mathrm{d}z$. 指令:

Z 变换: ztrans(f,var,transVar): 对 f 序列进行 Z 变换, var 子自变量, transVar 是转换变量.

Z 逆变换: ztrans(F,var,transVar): 对 F 进行 Z 逆变换, var 子自变量, transVar 是转换变量.

```
syms x n z % Z变换
fn = x^n/factorial(n); % x^n/n! 指数函数泰勒级数
Fz = ztrans(fn, n, z)
% 运行结果
Fz = exp(x/z)
```

```
syms x n z % Z逆变换
Fz = exp(x/z)
fni = iztrans(Fz, z, n)
% 运行结果
fni = x^n/factorial(n)
```

符号卷积

卷积的数学公式

$$y(t) = \int_0^t u(\tau)h(t-\tau)\,\mathrm{d}\tau$$

MATLAB 没有提供专门的符号卷积函数. 但是我们仍然有多种方法进行符号卷积的计算. 一. 积分函数 int, 二. 拉普拉斯变换或者傅里叶变换.

一. 积分法:

```
syms T t tau
ut = exp(-t); % 输入函数
ht = exp(-t/T)/T; % 冲激函数
utau = subs(ut,t,tau);
htau = subs(ht,t,t-tau);
uh = utau*htau;
yt = int(uh, tau, 0,t)
% 运行结果
yt = -(exp(-t) - exp(-t/T))/(T - 1)
```

二. 拉普拉斯变换法:

原理是: 卷积的拉普拉斯变换等于拉普拉斯变换的乘积.

```
syms T t s
ut = exp(-t); % 输入函数
ht = exp(-t/T)/T; % 冲激函数

Lyt = laplace(ut,t,s)*laplace(ht,t,s);

yt = ilaplace(Lyt,s,t)
% 运行结果
yt = exp(-t/T)/(T - 1) - exp(-t)/(T - 1)
```

致谢

本课程参考资料包括但不仅限于 (排名不分先后):

1. MATLAB 帮助文档
2. Experiments with MATLAB, C.Moler
3. MATLAB 高效编程技巧与应用: 25 个案例分析, 吴鹏 (rocwoods)
4. 精通 MATLAB R2011a, 张志勇
5. MATLAB 2014a 完全自学一本通, 刘浩/韩晶
6. Elementary Mathematical and Computational Tools for Electrical and Computer Engineers using MATLAB, Jamal T.Manassah
7. Physical Modeling in MATLAB, Allen B.Downey
8. 一些网络资源 (知乎, MATLAB 中文论坛等)
9. ...