简单来说，solve函数可以进行以下情况的求解：

（1）等式：单/多变量+线性/非线性 ；（2）不等式

（是MATLAB doc solve的全部翻译，将常用部分标注彩色）

（唉，以后绝不这样干了）

语法

S = solve(eqn,var)example

S = solve(eqn,var,Name,Value)example

Y = solve(eqns,vars)

Y = solve(eqns,vars,Name,Value)example

[y1,...,yN] = solve(eqns,vars)example

[y1,...,yN] = solve(eqns,vars,Name,Value)

[y1,...,yN,parameters,conditions] = solve(eqns,vars,'ReturnConditions',true)example

Description

一些函数

vpa 设置数值的精度（有效数字位数、保留的小数点位数）

subs 符号替换（用数字来替换符号变量）

ezplot 简单地画出函数的图形/曲线（显函数fun(x)、隐函数fun2(x,y)=0）

isAlways 一个判断函数（返回logical 1，表示true）

pretty 漂亮地打印符号表达式（看起来是有分子分母的格式）

举例

1.%% 求解单变量方程

%-----例子1------

syms x

eqn=sin(x)==1;

solve(eqn,x)

%-----例子2------

syms x

eqn=sin(x)==1;

[solx,params,conds]=solve(eqn,x,'ReturnConditions',true)

%-----例子3---------------

%如果返回empty，则表明解不存在。如果返回empty+warning，则解可能存在，但是solve找不到

syms x

solve(3\*x+2,3\*x+1,x)

2.%% 求解多变量方程

%---例1-----------------

%为了避免求解方程时对符号参数产生混乱，需要指明一个等式中需要求解的变量。

%如果不指明的话，solve函数就会通过symvar选择一个变量（认为该变量是要求解的变量）

clc,clear

syms a b c x

sola=solve(a\*x^2+b\*x+c==0,a) %待求解的变量是a

sol=solve(a\*x^2+b\*x+c==0) %待求解的变量是x

%----例子2--------------

%当求解的变量大于1个时，你声明变量的顺序就是slove返回解的顺序

syms a b

[b,a]=solve(a+b==1,2\*a-b==4,b,a)

3.%% solve返回的解带有：参数&条件

%为了返回一个方程的完整的解（即解中含有的参数，及对参数的限制），需要指定ReturnConditions 为：true

%---例子1：关于解的约束----

clc,clear

syms x

S=solve(sin(x)==0 ,x,'ReturnConditions',true);

S

S.x

S.parameters

S.conditions

%为了找到x的数值解，以一个值（利用函数subs）代替k。用函数isAlways检验该值是否满足关于k的限制

%检验k=4是否满足in(k, 'integer')

isAlways(subs(S.conditions,S.parameters,4))

%isAlways返回的是logical 1(true)，这意味着：对于k而言，4是一个合法值。

%利用4代替k，得到x的一个解。利用函数vpa获得该逼近的数值解(vpa设置数值精度：保留几位有效数字、几位小数)

solx=subs(S.x,S.parameters,4)

vpa(solx)

%为0<x<2\*pi寻找一个k的合法解，符合下面的前提：条件（S.conditions）、利用solve求解k的约束条件。替代解x中的k的值

assume(S.conditions)

solk=solve(S.x>0,S.x<2\*pi,S.parameters)

solx=subs(S.x,S.parameters,solk)

4.%% 求解方程组（为变量分配解）------------

%当求解方程组的时候，利用多个输出项对应求解的输出变量。

%solve返回一个符号数组（为每个相互独立的变量）

%-----例子1--------------

syms a u v

[sola,solu,solv]=solve(a\*u^2+v^2==0,u-v==1,a^2+6==5\*a,a,u,v)

solutions=[sola,solu,solv]

%----例子2------------

syms x y z

[solx,soly,solz]=solve(35\*(y-x)==0,-7\*x-x\*z+28\*y==0,x\*y-3\*z==0,x,y,z)

solutions=[solx,soly,solz]

5.%% 返回方程组完整的解（包括：参数和约束条件）

%需要指定ReturnConditions 为：true

%输出则要多附加两项：parameters 、conditions

clc,clear

syms x y

[solx,soly,params,conditions]=solve(sin(x)==cos(2\*y) , x^2==y , [x,y],'ReturnConditions',true)

solutions=[solx,soly]

6.%% 返回数值解

%解析解（analytical solution）：用严格的公式表示的解。

%数值解（numerical solution）：无法用严格的公式表示，是采用某种计算方法（有限元、逼近、插值）得到的。

%symbolic solver无法找到精确的用符号表示的解，因此在调用numeric solver之前会事先声明（warning）。因为等式不是多项式，所以想要找到全部可能的解需要很长时间。

%numeric solver不会尽力去找等式的全部numeric solution，它仅仅返回它找到的第一个解。

clc,clear

syms x

solve(sin(x)==x^2-1,x)

%验证上面的等式确实有一个正值解：画出等式的左右两部分的曲线

ezplot(sin(x),-2,2)

hold on

ezplot(x^2-1,-2,2)

hold off

%也可以直接用函数vpasolve求出数值解（需要定义（寻找）解的范围）

vpasolve(sin(x)==x^2-1,x,[0 2 ])

7.%% 求解不等式

%solve能求解满足约束条件的不等式

%需要指定ReturnConditions 为：true。这样可以返回解中涉及到的任何参数和约束条件

% x>0

% y>0

% x^2+y^2+xy<1

clc,clear

syms x y

S=solve(x^2+y^2+x\*y<1,x>0,y>0 , [x,y], 'ReturnConditions',true);

solx=S.x

soly=S.y

params=S.parameters

conditions=S.conditions

%利用subs和isAlways检验u=7/2和v=1/2是否满足约束条件

isAlways(subs(S.conditions,S.parameters,[7/2,1/2]))

%isAlways返回loogical 1（true）表示这些值满足约束条件。将这两个参数的值带入(函数subs)S.x和S.y中，找到一个x和y的解

solx=subs(S.x,S.parameters,[7/2,1/2])

soly=subs(S.y,S.parameters,[7/2,1/2])

%用函数vpa得到解的数值形式

vpa(solx)

vpa(soly)

8.%% 返回实数解

clc,clear

syms x

solve(x^5==3125,x)

%如果仅仅需要一个实数解，那么就把选项Real设置成true

solve(x^5==3125,x,'Real',true)

9.%% 返回一个解（主值Principal）

%不是返回一个无限多元素的周期解的集合，而是选择其中的最为实际的3个解（实际的、实用性的be most practical）

syms x

solve(sin(x)+cos(2\*x)==1,x)

%利用选择PrincipalValue设置为true选择一个解（主值）

solve(sin(x)+cos(2\*x)==1,x,'PrincipalValue',true)

10.%% 应用简化规则来缩短结果

%solve默认是不对解采用简化规则的，但是这些解从数学上来讲，不总是正确的。这样以来，solve就不能symbolically求解方程了。

clc,clear

syms x

solve(exp(log(x)\*log(3\*x))==4 , x)

%将IgnoreAnalyticConstraints （忽略解析约束）设置为true，这样就会应用简化规则，有可能让solve找到一个结果。

%简化规则的目的就是为了找到一个解。

%但是也不是任何情况下都可以应用简化规则，因此，应用简化规则后，应该对解的正确与否进行核实

S=solve(exp(log(x)\*log(3\*x))==4 , x,'IgnoreAnalyticConstraints',true)

11.%% 忽略有关变量的假设

%sym和syms函数可以让你对符号变量进行假设（设置assumptions）。例如，可以声明x为正值

clc,clear

syms x positive

%那么，在上述假设下，求得的解只能是符合假设的解

solve(x^2+5\*x-6==0,x)

%如果想要得到方程全部的解，则需要将IgnoreProperties 设置为true

solve(x^2+5\*x-6==0,x,'IgnoreProperties ',true)

%为了后续计算，清除之前的假设

syms x clear

12.%% 数值逼近符号解（that Contain RootOf）

%当求解多项式的时候，solve可能返回包含RootOf的解。为了数值逼近这些解，可以采用vpa函数。

clc,clear

syms x

s=solve(x^4+x^3+1==0,x)

%因为解中没有参数，所以可以采用vpa进行数值逼近

vpa(s)

13.%% 求解高阶的多项式等式

%当求解高阶的多项式方程的时候，solve可能采用 RootOf表示求得的解

clc,clear

syms x a

solve(x^4+x^3+a==0,x)

%为了得到方程的显式解，尝试调用带有参数MaxDegree的solve函数。该选项规定了多项式最大的degree，solve以此标准返回显式解。

%默认值是3。增大该数值，就可以得到高阶多项式的显式解。

s=solve(x^4+x^3+a==0,x,'MaxDegree',4)

pretty(s)

————————————————

版权声明：本文为CSDN博主「彩陶瓜」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/ldj1208/article/details/51893293