[6.46] solve() and nsolve() ignore constraints with local variables

(Applies to AMS 2.05)

Constraint expressions are used with *solve()* and *nsolve()* to limit the solution range for the independent variable. This shortens the solution time, and also narrows the search to the desired root if the function has more than one. For example, the constraint expression "|x>1 and x<2" in this example

```
nsolve(f(x)=0,x)|x>1 and x<2
```

should return a single solution between x=1 and x=2.

In July 2001, David Dannenmiller posted an *nsolve()* example to the 89/92+ discussion group, where the constraints were ignored and *nsolve()* returned a root outside of the solution bounds. It turns out that David found a particular constraint expression which causes *nsolve()* to return a solution which is a root, but is outside of the solution bounds. In particular, the constraint is expressed with a local variable, and the constraint bound is expressed as a>x instead of x<a, where 'a' is a local variable.

So, if you use *solve()* or *nsolve()* in a program or function, and you use local variables in the constraint expression, be sure to list the independent variable *first* in the contraint expression, to the left of the conditional operator:

Do this: `nsolve(f(x)=0,x)|x<a and x<b`

Don't do this: `nsolve(f(x)=0,x)|a>x and x<b`

As an example, consider this function:

$$f(x) = x^3 - 6x^2 + 11x - 6 + \sin(3x)$$

which has these three real roots when f(x)=0:

x1 = 0.71379958487236
x2 = 1.1085563021699
x3 = 2.14042730947

Suppose we use *nsolve()* in a function to find the third root, x3. The function might look like this:

```
f()
Func
local a,b

2→a
2.2→b

return nsolve(x^3-6*x^2+11*x-6+sin(3*x)=0,x)|x>a and x<b

EndFunc
```

This is a contrived example because you would not actually use such a function, to solve a single function for a single root. A more realistic application would solve a more complicated function with variable coefficients, and the constraint limits *a* and *b* would be found by some estimating method.

As shown, the constraint is |x>a and x<b. This table shows the returned results for various other constraint expressions

Constraint	Result	Comment
x>a and x<b	x3	
a<x and x<b	x1	Wrong!
a<x and b>x	x1	Wrong!
x>a and b>x	x3	

Note that if numeric limits are used, instead of local variables, then *nsolve()* returns the correct results regardless of the constraint ordering. For example, *nsolve()* will return the correct root x3 if the contraint is |2<x and x<2.2.

solve() shows the same behavior as *nsolve()*, except that it takes much longer to return the incorrect roots, and the correct root is often included as one of multiple solutions.