Nary a Variety

or a
Different Kind of Fun

UNK
SciMath / MathSci
Colloquium

23 October 2008

Barton Willis
Department of Mathematics
University of Nebraska at Kearney
Kearney, Nebraska 68849, USA

Maxima's solve function is weak

Solution set is too *small*:

```
(%i1) solve(sin(x)=0,x);
Some solutions will be lost.
(%o1) [x=0]
```

Solution set is too *big*:

```
(%i3) solve(sqrt(x)=-1 + %i,x);
(%o3) [x=-2*%i]
```

☞ $-1 + i \notin \text{range}(\sqrt{})$, so (%o3) should be the empty set.

Solution set is silly (unable to solve):

```
(%i1) solve(x + sqrt(x) = 2,x);
(%o1) [x=2-sqrt(x)]
```

Let's warm up our carbon-based calculators

Let's find the *complex* solutions to $x + \sqrt{x} = 2$. Let $g = \sqrt{x}$.

$$\{x \in \mathbf{C} \,|\, x + \sqrt{x} = 2\} = \{x + g = 2, g = \sqrt{x}\}.$$

Every solution to $g = \sqrt{x}$ is a solution to $g^2 = x$, but *some* solutions to $g^2 = x$ *aren't* solutions to $g = \sqrt{x}$; so

$$\subset \{x + g = 2, g^2 = x\}.$$

(Notice the subset!) Solving the polynomial system gives

$$= \{x = 4, g = -2\} \cup \{x = 1, g = 1\}.$$

Reinstating the set builder "preamble" $(x \in \mathbf{C})$ gives

$$= \{4, 1\}.$$

Some solutions might be spurious (the subset!); so we need to check each *candidate* solution:

$$1+\sqrt{1}=1+1=2, \quad \checkmark$$
$$4+\sqrt{4}=4+2\neq 2. \quad \times$$

We've shown that $\{x \in \mathbf{C} \,|\, x+\sqrt{x}=2\}=\{1\}$.

- We solved an *algebraic* equation by "polynomializing" it.
- There are algorithms for solving polynomial systems.

Question OK, I see how to polynomialize powers; what about other expressions?

Fact If $g = \max\{x, 5\}$ either $g - x = 0$ or $g - 5 = 0$. So

$$\begin{aligned}\{x \in \mathbf{R} \mid \max\{x,5\} = 10 - x\} &= \{g = 10 - x, g = \max\{x, 5\}\},\\ &\subset \{g = 10 - x, (g - x)(g - 5) = 0\}.\end{aligned}$$

Also, we can *almost* polynomialize trigonometric terms:

$$\begin{aligned}\{\cos(x) = 1/2\} &= \left\{\frac{1}{2} = \frac{1}{2}\left(e^{ix} + e^{-ix}\right)\right\},\\ &= \left\{\frac{1}{2} = \frac{1 + z^2}{2z}, z = e^{ix}\right\},\\ &\subset \{z^2 - z + 1 = 0, z = e^{ix}\}.\end{aligned}$$

- It's not a polynomialization ($z = e^{ix}$), but it's close enough.
- In the last line, the subset sign could be equality. But that takes some logic (need to show that $z \neq 0$).

The principal power (function)

Every nonzero complex number has a *unique* polar form $re^{i\theta}$, where $r \in \mathbf{R}_{\geq 0}$ and $\theta \in (-\pi, \pi]$. For a number in polar form, define

$$\arg\left(re^{i\theta}\right) = \begin{cases} 0 & \text{if } r = 0 \\ \theta & \text{if } r \neq 0 \end{cases}.$$

And for any $\alpha \in \mathbf{R}$ define the *principal power*

$$\left(re^{i\theta}\right)^{\alpha} = r^{\alpha} e^{i\alpha\theta}.$$

- Since $r \geq 0$ and α is real, we know what r^{α} means.
- $\text{range}\,(\sqrt{\ }) = \{z \in \mathbf{C} \mid \arg(z) \in (-\pi/2, \pi/2]\}$. Other powers have similar (wedge-like) ranges.

Let's try to fix Maxima's solve

We'd like a method that will solve equations such as:

$$x + \sqrt{x} - 2 = 0, \quad (1)$$
$$\max\{x, 2\} + x = 8, \quad (2)$$
$$|x| = a, \quad (3)$$
$$\cos(x) = 1/2. \quad (4)$$

- I'm only interested in *exact* solutions.
- Maxima has code for solving polynomial systems.
- Our scheme:

$$\text{equations} \longrightarrow \text{polynomials} \longrightarrow \text{solution} + \text{check}.$$

We'll bow out gracefully when we can't do the first step.

The case of the spurious solution*

Polynomializing $x^{3/2} + x = -3/8$ and solving shows that one candidate solution is

$$x = -\frac{\left(27\sqrt{113} - 803\right)^{\frac{2}{3}} - 4 \cdot 2^{\frac{1}{3}} \left(27\sqrt{113} - 803\right)^{\frac{1}{3}} + 52 \cdot 2^{\frac{2}{3}}}{12 \cdot 2^{\frac{1}{3}} \left(27\sqrt{113} - 803\right)^{\frac{1}{3}}}.$$

- Who would like to check this solution? Volunteers?
- Checking is sometimes harder than solving!

*Tribute to Erle Gardner and others, *The Case of the Spurious Sister*, 1959.

Theorem There is an algorithm that crunches every vanishing *algebraic* number to zero.

Theorem For numbers involving powers, roots, trigonometric functions, and logarithms, there is no algorithm that crunches every vanishing number to zero.

- It's not that we haven't *discovered* such an algorithm, such an algorithm doesn't exist. See (Daniel) Richardson's theorem (1968).
- Checking an *inequality* is generally easier than checking an *equality*.
- Let's append *inequalities* to the polynomialization process so we can check inequalities instead of equalities.

Polynomialization with constraints

We can avoid spurious solutions by appending constraints; for powers, the constraints are *inequalities:*

$$\{x + \sqrt{x} = 2\} = \left\{x + g = 2, g^2 = x, -\pi/2 < \text{arg}(g) \leq \pi/2\right\}$$

For absolute values, maximum, and minimum, the constraints are *equalities:*

$$\{|x| = 42\} = \left\{g = 42, g^2 = x^2, \text{arg}(g) = 0\right\},$$
$$\{\text{min}\{x, 1\} = 6\} = \{g = 6, (g - x)(g - 1) = 0, \text{arg}(g) = 0\}.$$

Fun with polynomializing

Let $x = \sqrt{5} + \sqrt{6}$. Polynomializing gives

$$\{x = g_0 + g_1, 5 = g_0^2, 6 = g_1^2\}.$$

Using resolvents to eliminate all but x yields

$$x^4 - 22x^2 + 1 = 0.$$

Solving for x gives four possible values for x. The solution that matches $x = \sqrt{5} + \sqrt{6}$ is $x = \sqrt{2\sqrt{30} + 11}$. So

$$\sqrt{5} + \sqrt{6} = \sqrt{2\sqrt{30} + 11}.$$

There are algorithms that de-nest *some* nested radicals.

Keepin' it real is hard to do

The equation $x^3 + x^2 - 16x - 9 = 0$ has *three real solutions*. One solution is

$$x = -\frac{\left(\sqrt{3}\,i + 1\right)\left(9\sqrt{17081}\,i + 97\sqrt{3}\right)^{\frac{1}{3}}}{6 \cdot 2^{\frac{1}{3}} \cdot 3^{\frac{1}{6}}} + \frac{49 \cdot 2^{\frac{1}{3}} \cdot 3^{\frac{1}{6}}\left(\sqrt{3}\,i - 1\right)}{6 \cdot \left(9 \cdot \sqrt{17081}\,i + 97\sqrt{3}\right)^{\frac{1}{3}}} - \frac{1}{3}$$

- Who would like to show that this solution is purely real?
- When solving over the reals, let's not be overly eager to reject a solution because it looks non-real.
- See *casus irreducibilis* (rejecting non-reals isn't easy.)

Solution sets can be messy

We have

$$\{x \in \mathbf{R} \mid |x| = a\} = \begin{cases} \{-a, a\} & \text{if } \arg(a) = 0 \\ \varnothing & \text{otherwise} \end{cases}.$$

And

$$\{x \in \mathbf{R} \mid mx = b\} = \begin{cases} \mathbf{R} & \text{if } m = 0, b = 0 \\ \varnothing & \text{if } m = 0, b \neq 0 \\ \{b/m\} & \text{otherwise} \end{cases}.$$

Representation of solutions

(a) Every solution is a *union* object. That makes iterating over solutions straightforward.

(b) Every member of a *union* object is either a list containing a solution or a conditional expression.

(c) To represent a nonfinite set, we'll introduce unique(!) identifiers:

$$\%Z_k = \text{arbitary integer},$$
$$\%C_k = \text{arbitary complex},$$
$$\%R_k = \text{arbitary real},$$

where k is an integer. All such identifiers are automatically entered into the Maxima database.

A simplifying union

A union object automatically:

(a) sorts and deletes redundant members:

$$\text{union}(b, a, a, b) \rightarrow \text{union}(a, b),$$

(b) uses associativity to *flatten:*

$$\text{union}(a, \text{union}(b, c)) \rightarrow \text{union}(a, b, c),$$

(c) expunges the empty set:

$$\text{union}(a, \text{union}()) \rightarrow \text{union}(a).$$

Maxima predicates

- A *predicate* is a boolean valued function.
- Maxima uses short-circuit logic (stop looking when the truth value is known);
- consequently, Maxima is *not* allowed sort or delete redundant members of a conjunction:

```
(%i1) is( (1 > 3) and 1/0 = 42);
(%o1) false

(%i2) a and a and not(a);
(%o2) a and a and not(a)
```

- For the solver, we'd like to have nonshort-circuit logic (so we can better simplify boolean valued expressions).
- Substitution into a Maxima conditional requires help:

```
(%i1) if (a < b) then 6 else 8;
(%o1) if a<b then 6 else 8

(%i2) subst([a=1, b = 2], %);
(%o2) if 1<2 then 6 else 8

(%i3) ev(%);
(%o3) 6
```

Also, the function ‘ev’ is a known trouble maker.

The Cure: I wrote my own simplifying logical functions.

❐ Disadvantages of one-method-for-all

Solutions can be more messy than expected; compare

$$\{x \in \mathbf{C} \,|\, x^{5/2} = -1\} = \{e^{-2i\pi/5}, e^{2i\pi/5}\},$$

(short, easy to check) to the cute, but incomprehensible

$$\{x \in \mathbf{C} \,|\, x^{5/2} = -1\} = \left\{ \frac{5^{\frac{1}{4}}\sqrt{\sqrt{5}-1}\,i}{2\sqrt{2}} - \frac{\sqrt{5}}{4} - \frac{1}{4}, \text{c.c.} \right\}.$$

Also, the solve process can wander in unexpected ways.

☞ If your math teacher asks you to *simplify*, ask your teacher to define *simplify.*

- The solver can needlessly struggle with equations that are easy to solve; for example

$$w = 1, \quad y = 42w + \exp(w + \sqrt{w}).$$

(A triangular system of equations.)

❐ The Lambert solver

One solution to $y\,\mathsf{exp}(y) = x$ is $y = W(x)$, where W is the *Lambert* function.

There is a pre-processor that looks for terms of the form

$$\mathsf{blob}^\star \times \mathsf{blob} \times \mathsf{exp}(\mathsf{blob}),$$

where blob* is a constant blob, and blob is a nonconstant blob. This method misses opportunities such as

$$\{\mathsf{log}(y) + y = \mathsf{log}(x)\} = \{y\,\mathsf{exp}(y) = x\} = \{y = W(x)\}.$$

❐ Cute (well, maybe) features

Solve for non-atoms:

```
(%i2) to_poly_solve(set(x^2 + x*y = 1, x*y=5), set(x^2, x*y));
(%o2) %union([x^2=-4,x*y=5])
```

Conjugate solver:

```
(%i7) to_poly_solve(z - %i/ conjugate(z)= 42,z);
(%o7) %union([z=%i/42-(sqrt(777923)-882)/42],
             [z=%i/42+(sqrt(777923)+882)/42])
```

Singular solutions

A solution that makes a jacobian singular is a *singular* solution; let's find the singular solutions to

$$ax^2 + y - b = 0, bx + y = 0,$$

where a and b are parameters. The jacobian is

$$\begin{bmatrix} \partial_x \left(ax^2 + y - b\right) & \partial_y \left(ax^2 + y - b\right) \\ \partial_x \left(bx + y\right) & \partial_y \left(bx + y\right) \end{bmatrix} = \begin{bmatrix} 2ax & 1 \\ b & 1 \end{bmatrix}.$$

The jacobian is singular* when $2ax - b = 0$.

*And no, I didn't use the determinant.

Solving (equations $\cup$ determinant of jacobian).

$$ax^2 + y - b = 0, bx + y = 0, 2ax - b = 0 \tag{5}$$

for x, y, and b gives *some* singular solutions. The singular solution(s) is (are)

$$\{x = 0, y = 0, b = 0\} \cup \{x = -2, y = -8a, b = -4a\}.$$

But wait! That's not all! Eqs. (5) have singular solutions! The jacobian is

$$\begin{bmatrix} \partial_x\left(ax^2 + y - b\right) & \partial_y\left(ax^2 + y - b\right) & \partial_b\left(ax^2 + y - b\right) \\ \partial_x\left(bx + y\right) & \partial_y\left(bx + y\right) & \partial_b\left(bx + y\right) \\ \partial_x\left(2ax - b\right) & \partial_y\left(2ax - b\right) & \partial_b\left(2ax - b\right) \end{bmatrix} = \begin{bmatrix} 2\,a\,x & 1 & -1 \\ b & 1 & x \\ 2\,a & 0 & -1 \end{bmatrix}$$

This jacobian is singular when $2a + b = 0$. So additional singular solutions:

$$ax^2 + y - b = 0, bx + y = 0, 2ax - b = 0, 2a + b = 0. \tag{6}$$

The solution is $\{x = z_0, y = 0, a = 0, b = 0\}$, where $z_0 \in \mathbf{C}$. The singular solution is a union of set-valued conditional expressions:

$$\begin{cases} \{x = 0, y = 0\} & \text{if } b = 0 \\ \varnothing & \text{otherwise} \end{cases} \cup \begin{cases} \{x = -2, y = -8a\} & \text{if } b = -4a \\ \varnothing & \text{otherwise} \end{cases}$$
$$\cup \begin{cases} \{x = z_0, y = 0\} & \text{if } (a = 0) \wedge (b = 0) \\ \varnothing & \text{otherwise} \end{cases}.$$

Two examples

```
(%i2) eq : set(max(x,x*y) - min(x^2 - y^2) = 9, abs(x-y)=3);
(%o2) {y^2+max(x,x*y)-x^2=9,abs(y-x)=3}

(%i3) to_poly_solve(%,set(x,y));
(%o3) %union([x=-9,y=-6],[x=0,y=-3],[x=0,y=3],[x=9,y=6])

(%i8) to_poly_solve(a*x^2 + x+1,x);
(%o8) %union([x=(sqrt(1-4*a)-1)/(2*a)],[x=-(sqrt(1-4*a)+1)/(2*a)])

(%i9) to_poly_solve(a*x^2 + x+1,x, 'parameters = ['a]);
(%o9) %union(%if(a=0,[x=-1],%union()),%if(4*a-1=0,[x=-2],%union()),%i
[x=(sqrt(1-4*a)-1)/(2*a)],%union()),%if(a#0 and 4*a-1#0,
[x=-(sqrt(1-4*a)+1)/(2*a)],%union()))
```

```
(%i10) tex(%)$
```

Typeset (slightly hand edited TEX), we have

$$\begin{cases} [x = -1] & \text{if } a = 0 \\ \varnothing & \text{otherwise} \end{cases}$$
$$\cup \begin{cases} [x = -2] & \text{if } 4\,a - 1 = 0 \\ \varnothing & \text{otherwise} \end{cases}$$
$$\cup \begin{cases} \left[x = \frac{\sqrt{1-4\,a}-1}{2\,a}\right] & \text{if } (a \neq 0) \wedge (4\,a - 1 \neq 0) \\ \varnothing & \text{otherwise} \end{cases}$$
$$\cup \begin{cases} \left[x = -\frac{\sqrt{1-4\,a}+1}{2\,a}\right] & \text{if } (a \neq 0) \wedge (4\,a - 1 \neq 0) \\ \varnothing & \text{otherwise} \end{cases}.$$

Things to work on

- To prevent infinite loops, the solver bails out when the recursion depth is too large (user settable, but unsmart). Maybe this is necessary, or maybe the solver needs additional logic.
- The Lambert solver misses opportunities.
- Solve over reals / complex needs work.
- Lots of possibilities: solve over integers, solve over extended reals, solve over (square) matrices, ...
- Make the solver user extensible—say on the fly tell it that the solution to $x = \log(y + \exp(y))$ is $y = \mathrm{UNK}(x)$.
- The predicates of conditional expressions are simplified in their own (mostly) empty environment (the global fact database is ignored).

In short and thanks

- blobification $\in$ abstraction.
- Working on the solver is fun and relaxing, but it's not research (it's a different kind of fun).
- My solver rarely strays from

$$\text{equations} \to \text{polynomials} + \text{constraints} \to \text{solution}.$$

- An associative and commutative function is sometimes said to be *nary*; the solution set of a polynomial system is a *variety*.
- My thanks to Maxima developers Robert Dodier and Stavros Macrakis for valuable guidance and patience with my flip-flops.