# Polynomials

Some insights into what Maple's *solve* command does under the hood

*Erik Postma - software architect - epostma@maplesoft.com*

## Working with *solve*

- Find roots (zeroes) of the following expression:

$expr := 6\,x^2 - x - 2 :$
$plot(expr, x = -1..1)$
$fsolve(expr);$
$solve(expr = 0);$

- Replace $x$ by the cosine of $t$.

$expr := expr\big|_{x = \cos(t)}$
$plot(expr, t = -1..1)$
$solve(expr = 0);$

- Looks just as straightforward - but it isn't!

$plot(expr, t = -5..5)$
$solve(expr = 0, AllSolutions)$

- Issue 1: Periodicity
- Issue 2: Is $\arccos\left(\frac{2}{3}\right)$ really a solution? It just means "the number between 0 and $\pi$ whose cosine is $\frac{2}{3}$". It's another equation to solve!

- There is no "more elementary" way to represent the answer.
- This is just a convention: $\pi$ is also just a conventional name for $\arccos(-1)$; $\sqrt{2}$ is just a conventional name for the positive zero of $x^2 - 2$.

$solve(a \cdot \exp(a) = z, a);$

$map(print, [indices(FunctionAdvisor(\mathrm{LambertW}), pairs)]) :$

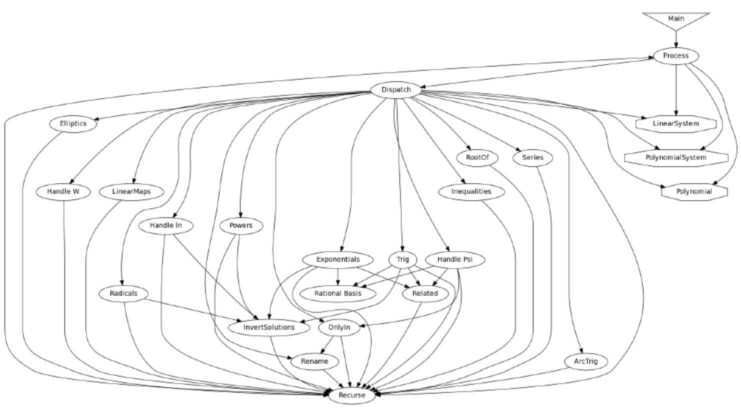
$$solve\left(6.132\,\cos(t)^2-\cos(t)\,-2.138=0\right);$$
$$solve\left(\frac{6132}{1000}\cos(t)^2-\cos(t)\,-\frac{2138}{1000}=0\right);$$
$$solve\left(6.132\,\cos(t)^2-\cos(t)\,-a=0,\,t\right);$$
$$solve\left(6.132\,\cos(t)^{2.1}-\cos(t)\,-a=0,\,t\right);$$

- "Most" polynomials of degree five and higher have no closed form solution. (For some reasonable measure, closed form solutions exist only for a measure-0 subset of the whole space. ) Hence what we saw above is the typical situation.
- Even if there exists a closed form solution, it doesn't always make you happy:

$$solve\left(x^9+3\,x^8+6\,x^7+5\,x^6+2\,x^5-3\,x^4-x^3+2\,x-2\right);$$

- What we really want solve to do is:
  – Rewrite our systems of equations to *simple* equations
  – If applicable, tell us the customary notation for the solution to such equations

# How does *solve* work?




- It all reduces to solving (systems of) polynomials in the end

# Solving single univariate polynomials

- Fundamental Theorem of Algebra: a non-constant univariate polynomial over a field $K$ has a root in an extension of $K$

$x^3 - 2$ has a root $\sqrt[3]{2}$
$x^2 + 1$ has a root $I$
$x^4 - 2x + 1$ has a root 1

- Given such a root $a$, we can *divide* by $x - a$ and get a polynomial that has the same set of roots except one occurrence of $a$ (using long division) :

$$evala\left(\frac{x^2+1}{x-I}\right) =$$
$$evala\left(\frac{x^3-2}{x-\sqrt[3]{2}}\right) =$$

- We can keep doing this as long as the polynomial is not constant, so any univariate polynomial of degree $n$ can be written as:

$$c\,(x - x_0)\,(x - x_1)\,\ldots\,(x - x_n)$$

- However, as we have seen, often the roots cannot be represented explicitly. In such a situation we factor the polynomial in as many factors with suitable coefficients as possible, and tell the user "it's the roots of these simpler factors". (For us, "suitable" = integer.)

$$factor(x^4 - 4x^2 + 4) =$$
$$solve(x^4 - 4x^2 + 4) =$$
$$factor(x^{10} - 2\cdot x^6 + 2\cdot x^5 + x^2 - 2\cdot x + 1) =$$
$$solve(x^{10} - 2\cdot x^6 + 2\cdot x^5 + x^2 - 2\cdot x + 1) =$$

- Factoring happens in many steps, with many tricks and shortcuts. Let's take an example.

$$f := x^8 + 3x^7 - 5x^6 - 25x^5 - 47x^4 - 47x^3 - 15x^2 + 5x + 2 :$$

- The first trick is to find if there are any *repeated factors*: if $f = g^2 \cdot h$. If so, then $\frac{d}{dx} f = 2\ g\ g'\,h + g^2 h' = g \cdot (2\,g'\,h + g\,h')$, and therefore $\frac{d}{dx} f$ and $f$ share a factor of $g$. If none of the factors are repeated ($f$ is *squarefree*), then $f$ and $\frac{d}{dx} f$ do not share any factors. This can be tested by computing the gcd:

$$fp := \frac{d}{dx} f =$$
$$gcd(f, fp) =$$
$$f2 := evala\left(\frac{f}{(x+1)^2}\right) =$$

- Now we know $f2$ is squarefree. The so-called *Landau-Mignotte bound* says that any (integer) factor of $p$ has coefficients that are, in an absolute sense, at most $LMB(p) = \left\lfloor \left( \binom{d-1}{\left\lfloor \frac{d}{2} \right\rfloor - 1} + \binom{d-1}{\left\lfloor \frac{d}{2} \right\rfloor} \right) \cdot \|p\|_2 \right\rfloor$, where $d = \left\lfloor \frac{degree(p)}{2} \right\rfloor$.

```
LMB := proc(p :: polynom, $)
local d, n;
    d := floor(degree(p)/2);
    n := norm(p, 2);
    return floor(binomial(d-1, floor(d/2)-1) +
                binomial(d-1, floor(d/2)) * n);
```

$LMB(f2) =$

- We will use finite fields: most simple algorithms for completely factoring polynomials reduce to factoring over finite fields, then build up the result in the original domain.
- If $f = g \cdot h$ is true over the integers, then equality also holds modulo any integer $m$ - so if there is a factorization over the integers, we will find it over the integers modulo $m$. Conversely, if we find a factorization over the integers modulo $m$, it may *not* correspond to a factorization over the integers:

$factor(x^2 + 2) =$
$Factor(x^2 + 2)$ **mod** $3 =$

- Demo here: use prime field $\mathbb{Z}/(p\mathbb{Z})$ with $p > 2 \cdot LMB(f2)$: then we know for each coefficient what the integer corresponding to it is.
- Best algorithm, but more complicated: use a small prime $p$, then "lift" factorization to rings $\mathbb{Z}/(p^n\mathbb{Z})$ with increasing $n$ until $p^n > 2 \cdot LMB(f2)$.
- Take $p := nextprime(2 \cdot LMB(f2)) =$ . Test that $f2$ is still squarefree if taken modulo $p$.

$Gcd(f2, diff(f2, x))$ **mod** $p =$

- Use: $x^{p^i} - x = \prod_{\substack{d \mid i \\ degree(g) = d \\ g\ irreducible \\ g\ monic}} g.$
- We can use this to find the product of all irreducible factors of degree 1, 2, ...: for $i = 1, 2, ...$, compute $gcd\left(f2, x^{p^i} - x\right)$, then divide $f2$ by the factor we just found.

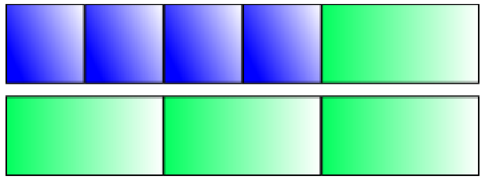
```
SplitDegrees := proc(f :: polynom,
                     x :: name,
                     p :: posint,
                     $)
local ff, lc, g, i, xpi, result;
     lc := lcoeff(f, x);
     ff := f / lc mod p;
     xpi := x;

     # Invariant: xpi = x^(p^i) mod ff
     # Invariant: ff is not divisible by irreducible
     #            factors of degree < i
     for i while degree(ff) >= 2*i do
          xpi := Powmod(xpi, p, ff, x) mod p;
          g := Gcd(ff, xpi - x) mod p;
          if g <> 1 then
               result[i] := g;
               ff := Quo(ff, g, x) mod p;
          end if;
     end do;

     if ff <> 1 then
          # Because of second invariant, ff must be
          # irreducible.
          result[degree(ff)] := ff;
```

$SplitDegrees(f2, x, \mathrm{p});$

- So we know that $f2$ has four linear factors and one quadratic factor over $\mathbb{Z}/(97\,\mathbb{Z})$.

$p2 := nextprime(p) =$
$SplitDegrees(f2, x, p2);$

- But only three quadratic factors over $\mathbb{Z}/(101\,\mathbb{Z})$!



- Since the factorization will be less coarse over the integers than over any prime field, we are better off with the three quadratic factors.
- However, we may be able to use the single quadratic factor found over $\mathbb{Z}/(97\,\mathbb{Z})$:

$rem(f2, x^2 + x + 2, x) =$
$f3 := quo(f2, x^2 + x + 2, x) =$

- This is indeed a valid factor over the integers, and we know it's irreducible because it was already irreducible over $\mathbb{Z}/(97\,\mathbb{Z})$.
- To find the $r := 2 :$ irreducible factors (say *f3a* and *f3b*) of *f3* over $\mathbb{Z}/(101\,\mathbb{Z})$ (which we know have degree $d := 2 :$):
- The field $\mathbb{Z}/(101\,\mathbb{Z})[X]/(f3)$ is a *direct sum* of two fields corresponding to *f3a* and *f3b*: a sum of two 2-dimensional vector spaces over $\mathbb{Z}/(101\,\mathbb{Z})$. So we can write any polynomial of degree 3 or less as a sum of a multiple of *f3a* and a multiple of *f3b* - but we don't know how.
- If we could get our hands on a multiple of *f3a*, we could find it by taking the gcd with *f3*.
- Take a *pseudorandom* element *g* of $\mathbb{Z}/(101\,\mathbb{Z})[X]/(f3)$ - that is, a polynomial of degree < $degree(f3) =$.

$g := Randpoly(degree(f3) - 1, x)$ **mod** $p2 =$

- Raise it to the power $\frac{p2^d - 1}{2} =$, modulo *f3* and modulo 101.

$gpow := Powmod\left(g, \frac{p2^d - 1}{2}, f3, x\right)$ **mod** $p2 =$

- Now for algebra tells us that the *f3a* component of *gpow* is equal to $+1$ for about half of the choices of *g* and equal to $-1$ for also about half of the choices. (There is also a small chance that it is 0.) The same is true for *f3b*.

$Gcd(gpow - 1, f3)$ **mod** $p2 =$
$Gcd(gpow + 1, f3)$ **mod** $p2 =$
$Gcd(gpow, f3)$ **mod** $p2 =$

- Bad luck? Try again.

$expand((x^2 + 18)\cdot(x^2 + 73) - f3)$ **mod** $p2 =$

- We now know that *if f3* has a factorization over the integers, it must be with factors congruent to $x^2 + 18$ and $x^2 + 73$ modulo 101.
- The Landau-Mignotte bound says that the absolute value of coefficients of factors of *f2*, and

therefore of $f3$, must be less than $LMB(f2) =$. So the candidate factorization is $(x^2 + 18) \cdot (x^2 + 73 - 101) =$.

- However, the coefficients must also be less than $LMB(f3) =$.

$expand((x^2 + 18) \cdot (x^2 - 28)) =$

- So $f3$ is irreducible over the integers.
- A full (integer) factorization of $f =$ is therefore $(x + 1)^2 \cdot (x^2 + x + 2) \cdot (x^4 - 10 \cdot x^2 + 1)$.

$factor(f) =$

# Solving systems of polynomials

- What does "solving a system of polynomials" mean?
- Much more complicated than single polynomials
- Redundancy
- Positive-dimensional components of a solution:

*restart*

$plots{:}\text{-}implicitplot3d(x^2 - y^2 z^2 + z^3, x = -0.5 .. 0.5, y = -2 .. 2, z = -1 .. 1, numpoints = 3 \cdot 10^3)$

$solve(\{x \cdot z = 0, y \cdot z = 0\});$

$plots{:}\text{-}display(plottools{:}\text{-}polygon([[-1, -1, 0], [-1, 1, 0], [1, 1, 0], [1, -1, 0]], color = red),$
$plottools{:}\text{-}line([0, 0, -1], [0, 0, 1], thickness = 3, color = black));$

- Several approaches: resultants, Gröbner basis, triangular decomposition/regular chains
- All take exponential amounts of time, or worse, in the worst case
- Resultants are a classical technique useful for theoretical results, but rarely used in practice these days
- For the rest, rewrite equations into some sort of normal form
- Gröbner bases are fairly well known; implementations in most major computer algebra systems (Maple has a well-regarded implementation of F4 by Faugère in its *Groebner* package)
- Triangular decomposition/regular chains: a similar idea, but a system is split into multiple simpler systems; a bit like row reduction for matrices - make each equation involve one pivot variable and only lesser variables than that

$$\begin{bmatrix} x^2 & y & z & -1 \\ x & y^2 & z & -1 \\ x & y & z^2 & -1 \end{bmatrix} \rightarrow \left\{ \begin{bmatrix} x & & -z & \\ & y & -z & \\ & & z^2 + 2z & -1 \end{bmatrix}, \begin{bmatrix} x & & & \\ & y & & \\ & & z & -1 \end{bmatrix}, \begin{bmatrix} x & & & \\ & y & & -1 \\ & & z & \end{bmatrix}, \begin{bmatrix} x & & & -1 \\ & y & & \\ & & z & \end{bmatrix} \right\}$$

$solve(\{x^2 + y + z = 1, x + y^2 + z = 1, x + y + z^2 = 1\}) =$

$\{x = 0, y = 0, z = 1\}, \{x = 0, y = 1, z = 0\}, \{x = 1, y = 0, z = 0\}, \{x = RootOf(\_Z^2 + 2\_Z - 1), y = RootOf(\_Z^2 + 2\_Z - 1), z = RootOf(\_Z^2 + 2\_Z - 1)\}$

$solve(\{x^2 + y + z = 1, x + y^2 + z = 1, x + y + z^2 = 1\}, 'explicit') =$

$\{x = 0, y = 0, z = 1\}, \{x = 0, y = 1, z = 0\}, \{x = 1, y = 0, z = 0\}, \{x = \sqrt{2} - 1, y = \sqrt{2} - 1, z = \sqrt{2} - 1\}, \{x = -1 - \sqrt{2}, y = -1 - \sqrt{2}, z = -1 - \sqrt{2}\}$

- I think the Maple package *RegularChains* is the only up to date implementation.

# Solving systems with inequalities and inequations (over real numbers)

- Inequalities: $a < b$ or $a \leq b$; inequations: $a \neq b$
- Just inequations are relatively easy to deal with - use the same theory as before
- Inequalities mean we need to solve systems over the real numbers only
- Theory much less well-developed: for a quadratic univariate polynomial $a\,x^2 + b\,x + c = 0$, we all know that the discriminant $b^2 - 4\,a\,c$ determines whether the polynomial has 0, 1, or 2 real solutions, but these pre-created rules don't exist for more complicated systems. This can now be done, using both Gröbner basis techniques and *RegularChains*.

```
with(RootFinding:-Parametric) :
cd := CellDecomposition([a x^2 + b x + c = 0], [x]);
NumberOfSolutions(cd);
map(print, CellDescription~(cd, [seq(1 ..12)])) :
```

- First cell: $c$-coordinate is between minus infinity and the first root of $c = 0$ (that is, $c < 0$), and $b < 0$, and $a < \frac{b^2}{4\,c}$.
- Second cell: same except $\frac{b^2}{4\,c} < a < 0$.
- Difficult to visualize volumes in 3D, but easy for 2D (that is, two parameters)

```
cd := CellDecomposition([x^3 + a·x^2 + b·x·y + a·b = 0, y^2 + b·y = a], [x, y]);
NumberOfSolutions(cd);
CellPlot(cd, samplepoints, symbolsize = 5, font = [HELVETICA, 15]);

cd:-SamplePoints[7] =
SampleSolutions(cd, %) =

solve([x^3 + a·x^2 + b·x·y + a·b = 0, y^2 + b·y = a], [x, y], parametric);
value(%) assuming b = 0
value(%) assuming a = 4;

solve([a·x^2 + b·x + c < 0], [x], parametric)

with(RegularChains) :
R := PolynomialRing([x, y, a, b]) :
RegularChains:-LazyRealTriangularize([x^3 + a·x^2 + b·x·y + a·b < 0, y^2 + b·y = a], R)
```