Computer Algebra through Maple and Reduce

James H. Davenport[1]
University of Bath
J.H.Davenport@bath.ac.uk
http://staff.bath.ac.uk/masjhd/JHD-CA.pdf

3 August 2017

[1] Thanks to EU H2020-FETOPEN-2016-2017-CSA project $\mathcal{SC}^2$ (712689) and the many partners on that project: www.sc-square.org

Maple and Reduce

Not the only options: Mathematica, Maxima, SAGE etc, in polynomial-based (calculus-oriented) computer algebra. More specialised SINGULAR and CoCoA.
MAGMA and GAP in group-theory

Reduce
: 45 years old; LISP-based; now public-domain; recursive structure (by default); expansion (by default)

From:
: `http://reduce-algebra.sourceforge.net/`

Maple
: 35 years old; C kernel; commercial product; distributed structure (by default); explicit expansion

“expand” and “simplify”

expand	Apply $a * (b + c) \Rightarrow a * b + a * c$ etc. exhaustively
simplify	“Looking at the standard textbooks on Computer Algebra Systems (CAS) leaves one even more perplexed: it is not even possible to find a proper definition of the problem of simplification” [Car04].
Query 1	Does $\frac{x^2-1}{x-1}$ simplify to $x + 1$?
Answer 1	Normally, but $x = 1$?
Query 2	Does $\frac{x^{1000}-1}{x-1}$ simplify to $x^{999} + \cdots + 1$?
Answer 2	For consistency, yes, but ouch!
Query 3	Does $\sqrt{1-x}\sqrt{1+x}$ simplify to $\sqrt{1-x^2}$?
Answer 3	Yes (but most systems won’t)
Query 4	Does $\sqrt{x-1}\sqrt{x+1}$ simplify to $\sqrt{x^2-1}$?
Answer 4	No: consider $x = -2$.
Query 5	Working mod p, does $x^p - x$ simplify to 0?
Answer 5	No as polynomials, yes as functions $\mathbf{F}_p \to \mathbf{F}_p$

Polynomials in one variable $\mathbf{Z}[x]$ or $\mathbf{Q}[x]$

$a_n x^n + \cdots + a_1 x + a_0$ with $a_n \neq 0$

Obvious Array $[a_0, a_1, \ldots, a_n]$ — Dense

But should $x^{1000000} - 1$ really take megabytes?

And this really won't scale to multivariates

So $((n, a_n), \ldots (1, a_1), (0, a_0))$ *all* $a_i \neq 0$ — Sparse

e.g. $((1000000, 1), (0, -1))$ for $x^{1000000} - 1$

While we might use dense in specific algorithms, all systems are sparse at top-level.

Sparse Complexity Theory is a challenge

Complexity in terms of degrees d_p is easy: $d_{f+g} \le \max(d_f, d_g)$; $d_{fg} = d_f + d_g$; $d_{f/g} = d_f - d_g$.
Number of terms t_f looks OK: $t_{f+g} \le t_f + t_g$, $t_{fg} \le t_f t_g$.
But $\frac{x^n-1}{x-1} = x^{n-1} + x^{n-2} + \cdots + x + 1$: $t_{f/g}$ is unbounded
GCD is equally bad and $t_{\gcd(f,g)}$ is unbounded[Sch03]:

$$\begin{aligned}\gcd(x^{pq}-1, x^{p+q}-x^p-x^q+1) &= \frac{(x^p-1)(x^q-1)}{x-1} \\ &= \underbrace{x^{p+q-1} + x^{p+q-2} \pm \cdots - 1}_{2\min(p,q) \text{ terms}},\end{aligned}$$

Theorem ([Pla77])

It is NP-hard to determine whether two sparse polynomials (in the standard encoding) have a non-trivial common divisor.

Conjecture ([DC10])

"Essentially", all bad cases are variants of $x^n - 1$

A general problem: gcd computation

$$\begin{aligned} A(x) &= x^8 + x^6 - 3x^4 - 3x^3 + 8x^2 + 2x - 5; \\ B(x) &= 3x^6 + 5x^4 - 4x^2 - 9x - 21. \end{aligned}$$

The first elimination gives $A - (\frac{x^2}{3} - \frac{2}{9})B$, that is

$$\frac{-5}{9}x^4 + \frac{127}{9}x^2 - \frac{29}{3},$$

and the subsequent eliminations give

$$\frac{50157}{25}x^2 - 9x - \frac{35847}{25}$$

$$\frac{93060801700}{1557792607653}x + \frac{23315940650}{173088067517}$$

and, finally,

$$\frac{7610300007338478950486 91}{866031281304672289 00}.$$

All rather large fractions considering where we started.

Work over $\mathbf{Z}$ instead? Cross-multiply

$$
\begin{aligned}
A(x) &= x^8 + x^6 - 3x^4 - 3x^3 + 8x^2 + 2x - 5;\\
B(x) &= 3x^6 + 5x^4 - 4x^2 - 9x - 21.
\end{aligned}
$$

$$-15\,x^4 + 381\,x^2 - 261,$$

$$6771195\,x^2 - 30375\,x - 4839345,$$

$$500745295852028212500\,x + 1129134141014747231250$$

and

$$7436622422540486538114177255855890572956445312500.$$

Again, this is a number, so $\gcd(A, B) = 1$.

Work modulo 5

$$\begin{aligned} A(x) &= x^8 + x^6 - 3x^4 - 3x^3 + 8x^2 + 2x - 5; \\ B(x) &= 3x^6 + 5x^4 - 4x^2 - 9x - 21. \end{aligned}$$

$$\begin{aligned} A_5(x) &= x^8 + x^6 + 2x^4 + 2x^3 + 3x^2 + 2x; \\ B_5(x) &= 3x^6 + x^2 + x - 1; \\ C_5(x) &= \mathrm{rem}(A_5(x), B_5(x)) = A_5(x) + 3(x^2+1)B_5(x) = 4x^2 + 3; \\ D_5(x) &= \mathrm{rem}(B_5(x), C_5(x)) = B_5(x) + (3x^4 + 4x^2 + 3)C_5(x) = x; \\ E_5(x) &= \mathrm{rem}(C_5(x), D_5(x)) = C_5(x) + xD_5(x) = 3. \end{aligned}$$

But anything that divides A and B over $\mathbf{Z}$ also does so mod 5, so $\gcd(A, B) = 1$.
How to generalise?

Relate $\gcd(f, g)$ and $\gcd(f \pmod p, g \pmod p)$?

Pathology: p might divide leading coefficients of both f and g: all bets are off.

Avoid!

p too small: The common factor might be $x + 7$, but with $p = 5$ I'll only see $x + 2$.

Solution: $p > 2\,\text{max coefficient in}\ \gcd(f, g)$

p misleading: $\gcd(x - 2, x + 3) = x - 2 = x + 3 \pmod 5$

Solution: Check the result and try different p

Theorem: Only finitely many misleading p: divisors of $\mathrm{res}(f, g)$.

lc: We don't know what the leading coefficient should be

How big should p be?

$$\begin{array}{rcl} f = & x^5+3x^4+2x^3-2x^2-3x-1 & = (x+1)^4(x-1); \\ g = & x^6+3x^5+3x^4+2x^3+3x^2+3x+1 & = (x+1)^4(x^2-x+1); \\ \gcd = & x^4+4x^3+6x^2+4x+1 & = (x+1)^4. \end{array}$$

Theorem (Landau–Mignotte[Lan05, Mig74])

Every coefficient of the g.c.d. of $f = \sum_{i=0}^{\alpha} a_i x^i$ and $g = \sum_{i=0}^{\beta} b_i x^i$ (with a_i and b_i integers) is bounded by

$$2^{\min(\alpha,\beta)} \gcd(a_\alpha, b_\beta) \min\left(\frac{1}{|a_\alpha|}\sqrt{\sum_{i=0}^{\alpha} a_i^2}, \frac{1}{|b_\beta|}\sqrt{\sum_{i=0}^{\beta} b_i^2}\right).$$

And 2 is best possible [Mig81], even though it's often overkill.

How to check $h = \gcd(f, g)$?

Theorem

We can never undershoot: a common divisor produced this way is a greatest common divisor.

Divide: Does h divide *both* f and g? Possibly expensive if fails

CrossMultiply: Produce A, B: $Ah = f$, $Bh = g$, and check the multiplications

But: these only certify common divisor.

Bézout: There are C, D such that $Cf + Dg = h$:

Certificate: (A, B, C, D) are a certificate for h.

2 is often overkill

Could try smaller p first, and if they don't work, try larger ones.
Or we can recycle these.

Theorem (Chinese Remainder)

If we know $f \pmod{p}$ and $f \pmod{q}$ we can determine $f \pmod{pq}$.

Hence we take small primes p_i until $\prod_{p_i \text{ good}} \geq$

$$2^{\min(\alpha,\beta)+1} \gcd(a_\alpha, b_\beta) \min\left(\frac{1}{|a_\alpha|}\sqrt{\sum_{i=0}^{\alpha} a_i^2}, \frac{1}{|b_\beta|}\sqrt{\sum_{i=0}^{\beta} b_i^2}\right).$$

Modular (CRT) GCD algorithm: $\gcd(A, B)$ [Col71, Bro71]

$M :=$`Landau_Mignotte_bound`(A, B); $g := \gcd(\mathrm{lc}(A), \mathrm{lc}(B))$;
$p :=$ `find_prime`(g); $D := g$`modular_gcd`(A, B, p);
if $\deg(D) = 0$ **then** **return** 1
$N := p$; # N is the modulus we will be constructing
while $N < 2M$ **repeat** (*)
  $p :=$ `find_prime`(g);
  $C := g$`modular_gcd`(A, B, p);
  if $\deg(C) = \deg(D)$
   then $D :=$ Chinese(C, D, p, N); $N := pN$;
   else **if** $\deg(C) < \deg(D)$
     # C proves that D is based on primes of bad reduction
     if $\deg(C) = 0$ **then** **return** 1
     $D := C$; $N := p$;
   else #D proves that p is of bad reduction, so we ignore it
$D := \mathrm{pp}(D)$; # In case multiplying by g was overkill
Check that D divides A and B, and return it
If not, all primes must have been bad, and we start again

CRT GCD algorithm: gcd(A, B) Early success

```
[Prelude as before]
while N < 2M repeat                    (*)
      p := find_prime(g);
      C := gmodular_gcd(A, B, p);
      if deg(C) = deg(D)
        then  if  C = D (mod p) and pp(D) divides A and B
                  then  return  pp(D)
              D := Chinese(C, D, p, N);  N := pN;
        else  if deg(C) < deg(D)
                # C proves that D is based on primes of bad reduction
                if deg(C) = 0 then  return 1
                D := C;  N := p;
        else    #D proves that p is of bad reduction, so we ignore it
D := pp(D);     # In case multiplying by g was overkill
Check that D divides A and B, and return it
If not, all primes must have been bad, and we start again
```

Polynomials in several variables

A fundamental choice. Note that we always use sparse encoding.

Recursive $\mathbf{Z}[y][x]$ e.g.
$$x^2(y^2+2y+1)+x(2y^2+4y+2)+x^0(y^2+2y+1):$$

Reduce (except that x^0 is suppressed)

Distributed $\mathbf{Z}[x,y]$ e.g.
$$\underbrace{x^2y^2}_{D=4}+\underbrace{2x^2y+2xy^2}_{D=3}+\underbrace{x^2+4xy+y^2}_{D=2}+\underbrace{2x+2y}_{D=1}+\underbrace{1}_{D=0}$$

Maple In the `Poly` format [MP14], after `expand`

But why not
$$\underbrace{y^2x^2}_{D=4}+\underbrace{2y^2x+2yx^2}_{D=3}+\underbrace{y^2+4yx+x^2}_{D=2}+\underbrace{2y+2x}_{D=1}+\underbrace{1}_{D=0}$$

Or $$\underbrace{x^2y^2+2x^2y+x^2}_{D_x=2}+\underbrace{2xy^2+4xy+2x}_{D_x=1}+\underbrace{y^2+2y+1}_{D_x=0}$$

Or ... (there are many orderings: Gröbner base theory).

GCD in several variables

The naïve algorithms, when run in $\mathbf{Z}[\ldots][x]$, suffer growth in $\mathbf{Z}[\ldots]$ as we reduce x, just as univariates did.

Basically same solution: as well as working modulo (several small) p_i, we work modulo (several) $y - v_i$

We still have Chinese Remainder Theorem, theorems that guarantee the algorithms work, good bounds (much better than Landau–Mignotte) etc.

Pragmatically, the complexity isn't bad for dense polynomials — same league as division (maybe 10–100 times worse), but much worse for sparse polynomials (if the answer is non-trivial)

Hence we want algorithms that avoid gcd where possible, but we shouldn't be afraid of doing it when necessary

In particular

Differentiation $f = \sum a_i x^i$, then $f' = \sum i a_i x^{i-1}$ (pure algebra)

Note that if $f = f_1 f_2^2$, then $f' = f_1' f_2^2 + f_1 f_2' f_2$, so $f_2 | \gcd(f, f')$
If the f_i are square-free and relatively prime, $f_2 = \gcd(f, f')$.
And in general, if $f = \prod f_i^i$ (f_i square-free and relatively prime), then $\prod_{i>1} f_i^{i-1} = \gcd(f, f')$; $\prod_i f_i = \frac{f}{\gcd(f,f')}$;
$f_1 = \frac{\prod_i f_i}{\gcd(\prod_i f_i, \prod_{i>1} f_i^{i-1})}$ etc.
Hence we can recover the f_i by gcd alone (in fact, there are smarter ways[Yun76]).
This is known as *square-free decomposition*. In theory, we end up with more polynomials which might be larger, but in practice

- if it doesn't find anything it's cheap
- if it does find something, the gain is almost always worth it
- Theory-wise, McCallum's (M, D) notation makes it manageable [McC84]

Factoring

quadratics $ax^2 + bx + c$: factors iff $b^2 - 4ac$ is a square

cubic $ax^3 + bx^2 + cx + d$: must have a linear factor $a'x + d'$ with $a'|a, d'|d$

$$\frac{1}{6}\sqrt[3]{36\,bc - 108\,d - 8\,b^3 + 12\,\sqrt{12\,c^3 - 3\,c^2b^2 - 54\,bcd + 81\,d^2 + 12\,db^3}} - \frac{2c - \frac{2}{3}b^2}{\sqrt[3]{36\,bc - 108\,d - 8\,b^3 + 12\,\sqrt{12\,c^3 - 3\,c^2b^2 - 54\,bcd + 81\,d^2 + 12\,db^3}}} - \frac{1}{3}b.$$

quartic Well, there's a formula, but I can't remember it: maybe trial and error?

quintics etc. No formula

The quartic formula

$x^4 + bx * c + cx + d$ after a transformation

$$\frac{\sqrt{6}}{12}\sqrt{\frac{-4\,b\sqrt[3]{-288\,db + 108\,c^2 + 8\,b^3 + 12\sqrt{-768\,d^3 + 384\,d^2b^2 - 48\,d\ldots}}}{\sqrt[3]{-288\ldots}}}$$

$$\begin{aligned}
S &:= \sqrt{-768\,d^3 + 384\,d^2b^2 - 48\,db^4 - 432\,dbc^2 + 81\,c^4 + 12\,c^2b^3} \\
T &:= \sqrt[3]{-288\,db + 108\,c^2 + 8\,b^3 + 12\,S} \\
U &:= \sqrt{\frac{-4\,bT + T^2 + 48\,d + 4\,b^2}{T}}
\end{aligned}$$

return $\displaystyle \frac{\sqrt{6}}{12}U + \frac{\sqrt{6}}{12}\sqrt{\frac{-\left(8\,bTU + UT^2 + 48\,Ud + 4\,Ub^2 + 12\,c\sqrt{6}\,T\right)}{TU}}$

Factoring mod p (small) is $O(d^3)$

If a polynomialis irreducible mod p it's irreducible: great.
But a generic (therefore irreducible) polynomial only has a $1/d$ chance of being irreducible mod p
However, it will factor differently modulo different primes,e.g. a degree 4 might factor as $f_3 \times f_1$ modulo p_1, and $g_2 \times h_2$ modulo p_2
Hence in fact that polynomial must be irreducible over $\mathbf{Z}$
[Mus78] states 5 primes suffice for generic polynomials: in theory there's also a $\log \log d$ term, and [PPR15] suggest 7 primes.

However, that's for generic polynomials

Particular cases might need more, or even not be provable irreducible.
$x^4 + 1$ *is* irreducible, but always factors as $g_2 \times h_2$ (or more splitting) modulo p
Statistically (taking random polynomials of degree d and coefficients $\leq H$, and letting $H \to \infty$) this never happens, but in real life it does, especially when manipulating algebraic numbers

OK, but we still have the Chinese Remainder Theorem?

Consider x^4+3. This factors as

$$x^4+3=\left(x^2+2\right)(x+4)(x+3) \quad \bmod 7$$

$$x^4+3=\left(x^2+x+6\right)\left(x^2+10\,x+6\right) \quad \bmod 11. \tag{1}$$

So the first has too much decomposition, and we consider

$$x^4+3=\left(x^2+2\right)\left(x^2+5\right) \quad \bmod 7, \tag{2}$$

obtained by combining the two linear factors.
Chinese Remainder Theorem dilemma: do we pair $\left(x^2+x+6\right)$ with $\left(x^2+2\right)$ or $\left(x^2+5\right)$? Both *are* feasible.

$$x^4+3=\left(x^2+56\,x+72\right)\left(x^2-56\,x-16\right) \quad \bmod 77, \tag{3}$$

$$x^4+3=\left(x^2+56\,x+61\right)\left(x^2-56\,x-5\right) \quad \bmod 77: \tag{4}$$

both of which are correct. The difficulty in this case is that, while polynomials over $\mathbf{Z}_7$ have unique factorization, as do those over $\mathbf{Z}_{11}$ (and indeed modulo any prime), polynomials over $\mathbf{Z}_{77}$ (or any product of primes) do not, as (3) and (4) demonstrate.

We need a different technique

Hensel's Lemma lets us take a factorisation modulo p and lift it to one mod p^2, and then one mod p^3 (or indeed p^4) and so on, *and the lifting is unique* (as long as the polynomial is square-free)

1. Factor modulo several (up to 7) p
2. Piece together
3. (return irreducible if possible)
4. Take the best p,
5. lift to $p^n > 2$Landau–Mignotte
6. Combine these factors to make factors over the integers

This also works for multivariates, but it's an expensive process

Gröbner Bases [Buc65]

Think distributed in $R[x_1,\ldots,x_n]$, fix an order $\prec$ on monomials and sort that way, leading monomial ($\mathrm{lm}(f)$) of f first.

If $\mathrm{lm}(g)$ divides $\mathrm{lm}(f)$ then g reduces f: $f \to^g f - \frac{\mathrm{lt}(f)}{\mathrm{lt}(g)} g$.

$S(f,g) := \frac{\mathrm{lt}(g)}{\gcd(\mathrm{lm}(f),\mathrm{lm}(g))} f - \frac{\mathrm{lt}(f)}{\gcd(\mathrm{lm}(f),\mathrm{lm}(g))} g$

Theorem

The following conditions are equivalent

1. $\forall f,g \in G, S(f,g) \stackrel{*}{\to}^G 0$. *This is known as the S-Criterion.*
2. *If* $f \stackrel{*}{\to}^G g_1$ *and* $f \stackrel{*}{\to}^G g_2$, *then* g_1 *and* g_2 *differ at most by a multiple in* R, *i.e.* $\stackrel{*}{\to}^G$ *is essentially well-defined.*
3. $\forall f \in \mathit{Ideal}(G), f \stackrel{*}{\to}^G 0$.
4. $\mathit{Ideal}(\mathrm{lm}(G)) = \mathit{Ideal}(\mathrm{lm}(\mathit{Ideal}(G)))$.

Then G is called a Gröbner Base. Completely reduced Gröbner bases are unique

Purely Lexicographic GB $\approx$ Triangular Matrices

Purely lex = "consider degrees in x_1, break ties by degree in x_2, etc."

$$\begin{gathered} p_n(x_n) \\ p_{n-1,1}(x_{n-1}, x_n), \ldots, p_{n-1,k_{n-1}}(x_{n-1}, x_n) \\ \vdots \\ p_{1,1}(x_1, \ldots), x_n), \ldots, p_{1,k_1}(x_1, \ldots, x_n) \end{gathered}$$

This gives us a back-substitution process (for finitely many zeros)
Solve for x_n, for each root, solve the lowest-degree $p_{n-1,i}$ not to vanish for x_{n-1}, continue
$p_{i,j}$ vanishes iff its leading coefficient does [Gia89, Kal89].

At
`http://staff.bath.ac.uk/masjhd/Slides/SC2School2017/`
in Maple worksheet (executable) and PDF (readable) formats.

GB3 "cyclic 3" A Gröbner base in either `tdeg` or `plex` shows the solutions: 6.

GB4 "cyclic 4" A Gröbner base in `plex` shows that d is undetermined. If we spot the repeated factor, the solutions drop out easily enough: two one-dimensional curves (but we've lost the multiplicity information).

GB5 "cyclic 5" A Gröbner base in `plex` shows that each variable is determined. However, the Gianni–Kalkbrener process is quite complicated (70 solutions).

Cyclic-n has finitely many solutions iff n is square-free [Bac89].

Nonlinear Polynomial Systems are hard

1. $x^2-1, y^2-1, (x-1)(y-1)$ defines 3 points of the plane, 2 when $x=1$ and 1 when $x=-1$. not equiprojectable
2. $(x-y-1)(x-3), (x-y-1)(y-1)$ defines the line $x=y+1$ *and* the point $(3,1)$. not equidimensional
3. $x^2+y^2=0$ defines two lines in **C**, but a point in **R**. $\mathbf{C} \neq \mathbf{R}$
4. Gröbner bases can be doubly-exponential in degree, comparied with the input [MR13]. Is this rare?

Maybe the problem is that we are insisting on a universal solution.

Triangular Sets/Regular Chains [Wu89, ALM99]

Every polynomial has a different main variable. Not always possible: $x^2-1, y^2-1, (x-1)(y-1)$

But if we did have this, reading off the solutions would be easy

So have several regular chains: $\{x-1, y^2-1\}, \{x+1, y-1\}$

Important technical conditions: every lc is invertible with respect to the rest of the chain

Not much is known about the complexity theoretically, but in practice the special cases kill you. So why do them? [CDM+10]

Regular Grains: worked examples

At
`http://staff.bath.ac.uk/masjhd/Slides/SC2School2017/`
in Maple worksheet (executable) and PDF (readable) formats.

RC
: The eamples GB4 and GB5 from Groebner bases

LRT
: An example of `LazyRealTRiangularize`, where the special cases are wrapped up in further, unevaltiated, calls to `LazyRealTRiangularize`

Questions?

P. Aubry, D. Lazard, and M. Moreno Maza.
On the Theories of Triangular Sets.
J. Symbolic Comp., 28:105–124, 1999.

J. Backelin.
Square multiples *n* give infinitely many cyclic *n*-roots.
Technical Report 8, Matematiska Institutionen Stockholms Universitet, 1989.

W.S. Brown.
On Euclid's Algorithm and the Computation of Polynomial Greatest Common Divisors.
J. ACM, 18:478–504, 1971.

Bibliography II

B. Buchberger.
Ein Algorithmus zum Auffinden des Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal.
PhD thesis, Math. Inst. University of Innsbruck, 1965.

J. Carette.
Understanding Expression Simplification.
In J. Gutierrez, editor, *Proceedings ISSAC 2004*, pages 72–79, 2004.

C. Chen, J.H. Davenport, J.P. May, M. Moreno Maza, B. Xia, and R. Xiao.
Triangular Decomposition of Semi-algebraic Systems.
In S.M. Watt, editor, *Proceedings ISSAC 2010*, pages 187–194, 2010.

G.E. Collins.
The Calculation of Multivariate Polynomial Resultants.
J. ACM, 18:515–532, 1971.

J.H. Davenport and J. Carette.
The Sparsity Challenges.
In S. Watt *et al.*, editor, *Proceedings SYNASC 2009*, pages 3–7, 2010.

P. Gianni.
Properties of Gröbner bases under specializations.
In *Proceedings EUROCAL 87*, pages 293–297, 1989.

M. Kalkbrener.
Solving systems of algebraic equations by using Gröbner bases.
In *Proceedings EUROCAL 87*, pages 282–292, 1989.

E. Landau.
Sur Quelques Théorèmes de M. Petrovic Relatif aux Zéros des Fonctions Analytiques.
Bull. Soc. Math. France, 33:251–261, 1905.

S. McCallum.
An Improved Projection Operation for Cylindrical Algebraic Decomposition.
PhD thesis, University of Wisconsin-Madison Computer Science, 1984.

M. Mignotte.
An Inequality about Factors of Polynomials.
Math. Comp., 28:1153–1157, 1974.

Bibliography

V

M. Mignotte.
Some Inequalities About Univariate Polynomials.
In *Proceedings SYMSAC 81*, pages 195–199, 1981.

M. Monagan and R. Pearce.
POLY : A new polynomial data structure for Maple 17.
In R. Feng *et al.*, editor, *Proceedings Computer Mathematics*, pages 325–348, 2014.

E.W. Mayr and S. Ritscher.
Dimension-dependent bounds for Gröbner bases of polynomial ideals.
J. Symbolic Comp., 49:78–94, 2013.

D.R. Musser.
On the efficiency of a polynomial irreducibility test.
J. ACM, 25:271–282, 1978.

D.A. Plaisted.
Sparse Complex Polynomials and Irreducibility.
J. Comp. Syst. Sci., 14:210–221, 1977.

R. Pemantle, Y. Peres, and I. Rivin.
Four random permutations conjugated by an adversary generate S_n with high probability.
Random Structures & Algorithms, 49:409–428, 2015.

A. Schinzel.
On the greatest common divisor of two univariate polynomials, I.
In *A Panorama of number theory or the view from Baker's garden*, pages 337–352. C.U.P., 2003.

W. Wu.
A zero structure theorem for polynomial-equations-solving and its applications.
In *Proceedings EUROCAL 87*, 1989.

D.Y.Y. Yun.
On Square-free Decomposition Algorithms.
In R.D. Jenks, editor, *Proceedings SYMSAC 76*, pages 26–35, 1976.