# Iterated Resultants in Cylindrical Algebraic Decomposition

**James H. Davenport** (University of Bath)
and Matthew England (Coventry University)

8th International Workshop on Satisfiability Checking and Symbolic Computation, 28th July 2023
Tromsø, Norway

Supported by EPSRC DEWCAD Project, *Pushing Back the Doubly-Exponential Wall of Cylindrical Algebraic Decomposition* (EP/T015713/1 and EP/T015748/1).

# Definition and Plan



The **resultant** of two polynomials is a polynomial formed of their coefficients that is equal to zero if and only if the two original polynomials have a common root.

Iterated resultants are a key ingredient in Cylindrical Algebraic Decomposition (CAD) [Col75]. Thus also in the SMT tools developed based on CAD technology, e.g. NLSAT/MCSAT [JdM12] and Cylindrical Algebraic Coverings [ADEK21].

[Col75, pp. 177–178] suggests that iterated resultants, where there are "common ancestors" tend to factor. We present here some prelinary ideas on optimisations emitting from this, in the context of SC-Square technology.

# Outline

# Outline

# Inertia Forms



Let $A$ be a set of $r$ homogeneous polynomials $F_1, \ldots, F_r$ in $x_1, \ldots, x_n$, with indeterminate coefficients. An integral polynomial $T$ in these indeterminates (that is, $T \in \mathbb{Z}[A]$) is called an *inertia form* for $F_1, \ldots, F_r$ if $x_i^\tau T \in (F_1, \ldots, F_r)$, for suitable $i$ and $\tau$.

The inertia forms comprise an ideal $I$ of $\mathbb{Z}[A]$, and van der Waerden1950 showed that $I$ is a prime ideal of this ring. It follows from these observations that we may take the ideal $I$ of inertia forms to be a resultant system for the given $F_1, \ldots, F_r$ in the sense that for special values of the coefficients in $K$, the vanishing of all elements of the resultant system is necessary and sufficient for there to exist a non-trivial solution to the system $F_1 = 0, \ldots, F_r = 0$ in some extension of $K$.

# Homogenous Multipolynomial Resultant



Consider $n$ homogeneous polynomials in $n$ variables. Let $F_1, \ldots, F_n$ be $n$ generic homogeneous forms in $x_1, \ldots, x_n$ of positive total degrees $d_1, \ldots, d_n$. I.e. every possible coefficient of each $F_i$ is a distinct indeterminate, and the set of all coefficients is $A$. Let $I$ denote the ideal of inertia forms for $F_1, \ldots, F_n$.

McCallum and Winkler proved the following.
[MW18, Proposition 5]: $I$ is a nonzero principal ideal of $\mathbb{Z}[A]$: $I = (R)$, for some $R \neq 0$. $R$ is uniquely determined up to sign. We call $R$ the (generic multipolynomial) resultant of $F_1, \ldots, F_n$.
[MW18, Proposition 6] The vanishing of $R$ for particular $F_1, \ldots, F_n$ with coefficients in a field $K$ is necessary and sufficient for the existence of a non-trivial zero of the system $F_1 = 0, \ldots, F_n = 0$ in some extension of $K$.

# The Multivariate Resultant



For a given non-homogeneous $f(x_1, \ldots, x_{n-1})$ over $K$ of total degree d, we may write $f = H_d + H_{d-1} + \cdots + H_0$, where the $H_j$ are homogeneous of degree $j$. Then $H_d$ is known as the leading form of $f$. Recall that the homogenization $F(x_1, \ldots, x_n)$ of $f$ is defined by $F = H_d + H_{d-1}x_n + \cdots + H_0 x_n^{d_n}$.

Let $f_1, \ldots, f_n$ be particular non-homogeneous polynomials in $x_1, \ldots, x_{n-1}$ over $K$ of positive total degrees $d_i$, and with leading forms $H_{i,d_i}$. We set $\mathrm{res}(\mathrm{f}_1, \ldots, \mathrm{f}_\mathrm{n}) = \mathrm{res}(\mathrm{F}_1, \ldots, \mathrm{F}_\mathrm{n})$ , where $F_i$ is the homogenization of $f_i$ to define the multivariate resultant of $n$ non-homogeneous polynomials in $n-1$ variables.

# Properties of the Multivariate Resultant



[MW18, Proposition 7]: The vanishing of $\mathrm{res}(\mathrm{f}_1, \ldots, \mathrm{f}_\mathrm{n})$ is necessary and sufficient for

either the forms $H_{i,d_i}$ to have a common nontrivial zero over an extension of $K$,

or the polynomials $f_i$ to have a common zero over an extension of $K$.

Observe that the common zeros of the $f_i$ correspond to the affine solutions of the system, whereas the nontrivial common zeros of the leading forms correspond to the projective solutions on the hyperplane at infinity.

# Outline

1. Iterated vs Multivariate Resultants
   - Theory
   - Example

2. Optimisations
   - Discarding Spurious Factors
   - Detecting Spurious Factors

# Example



Consider these polynomials:

$$
\begin{aligned}
f &= y^2 + z^2 + x + z - 1, \\
g &= -x^2 + y^2 + z^2 - 1, \\
h &= x^2 + y + z.
\end{aligned}
$$

Under variable ordering $z \succ y \succ x$ we may calculate the iterated resultant $\mathrm{res_y(res_z(f,g), res_z(f,h))}$ as

$$
\begin{aligned}
&= 5x^8 + 16x^7 + 14x^6 - 2x^5 - 12x^4 - 8x^3 + 3x^2 + 2x \\
&= \underbrace{x\left(5x^3 + 6x^2 - 3x - 2\right)}_{\text{spurious}} \underbrace{\left(x^2 + x + 1\right)\left(x^2 + x - 1\right)}_{\text{genuine}} . \qquad (1)
\end{aligned}
$$

# Genuine vs Spurious



The roots of the factors labelled as "genuine" are

$$\{x : \exists y \exists z f(x,y,z) = g(x,y,z) = h(x,y,z) = 0\}, \qquad (2)$$

whereas the roots of the factors labelled as "spurious" are

$$\begin{aligned} \{x : \exists y (\exists z_1 f(x,y,z_1) = g(x,y,z_1) = 0 \wedge \\ \exists z_2 \neq z_1 f(x,y,z_2) = h(x,y,z_2) = 0)\}. \end{aligned} \qquad (3)$$

They are "spurious" in the sense that they do not form part of any true root of all three polynomials. Nevertheless, they are $x$ values above which the topology changes, so cannot always be discarded.

Note that there will always be a neat factorisation (over $\mathbb{Z}$ if that was the original ring) into "genuine" versus "spurious".

# Alternative Resultant Combinations



Instead of $\mathrm{res_y}(\mathrm{res_z}(f, g), \mathrm{res_z}(f, h))$ we may calculate:

- $\mathrm{res_y}(\mathrm{res_z}(f, g), \mathrm{res_z}(g, h))$

$$\begin{aligned} &= 5x^8 + 16x^7 + 18x^6 + 8x^5 - 5x^4 - 8x^3 - 2x^2 + 1 \\ &= \underbrace{\left(x^2 + x + 1\right)\left(x^2 + x - 1\right)}_{\text{genuine}} \underbrace{\left(5x^4 + 6x^3 + x^2 - 1\right)}_{\text{spurious}}. \end{aligned} \tag{4}$$

- $\mathrm{res_y}(\mathrm{res_z}(f, h), \mathrm{res_z}(g, h))$

$$\begin{aligned} &= 2x^4 + 4x^3 + 2x^2 - 2 \\ &= 2\underbrace{\left(x^2 + x + 1\right)\left(x^2 + x - 1\right)}_{\text{genuine}}. \end{aligned} \tag{5}$$

# Gröbner Basis to Reveal Multivariate Resultant (Slide 9/18)

Consider the Gröbner Basis,

$$GB_{\texttt{plex}}(f, g, h) = \left\{x^4 + 2x^3 + x^2 - 1, y - x, x^2 + x + z\right\}. \quad (6)$$

We see that the basis polynomial univariate in $x$ divides all three of the iterated resultants we computed. In fact, it is the multivariate resultant $\mathrm{res}(\mathrm{f}, \mathrm{g}, \mathrm{h})$. That will happen in general.

In this example, it happened to be one of the iterated resultants (5), but that need not happen in general.

# Variable Ordering



Earlier we used $z \succ y \succ x$. If instead we use $x \succ y \succ z$ we have:

$$\mathrm{res_y}(\mathrm{res_x}(f,g),\mathrm{res_x}(f,h)) = (z^2-1)^2, \tag{7}$$

$$\mathrm{res_y}(\mathrm{res_x}(f,g),\mathrm{res_x}(g,h)) = (z^2-1)^4, \tag{8}$$

$$\mathrm{res_y}(\mathrm{res_x}(h,g),\mathrm{res_x}(f,h)) = (z^2-1)^4, \tag{9}$$

$$GB_{\mathtt{plex(x,y,z)}}(f,g,h) = \left\{z^2-1, y^2+y+z, x-y\right\}. \tag{10}$$

I.e. no spurious roots were uncovered with this ordering.

CAD variable ordering is known to greatly effect the complexity of CAD both in practice [dRE22] and theory [BD07]. Is the introduction of spurious factors in some orderings but not others a significant contributing factor?

# Outline

1. Iterated vs Multivariate Resultants
   - Theory
   - Example
2. Optimisations
   - Discarding Spurious Factors
   - Detecting Spurious Factors

# CAD with Multiple Equational Constraints (Slide 11/18)

McCallum [McC01] optimised CAD for multiple equation constraints (ECs) i.e. the case when

$$\Phi \equiv f_1 = 0 \wedge f_2 = 0 \wedge \cdots f_k = 0 \wedge \overline{\Phi}(f_{k+1}, \ldots, f_m). \qquad (11)$$

[McC01] proved that we need only take those resultants that involve one designated EC, say $f_1$ in the first projection. Then at the next projection $\mathrm{res}(\mathrm{f}_1, \mathrm{f}_2)$ is another EC and we can proceed similarly.

For such input we are only interested in the genuine zeros, since away from these the formula will be uniformly false and so any further refinement is unnecessary.
Thus any $\mathrm{res}_{\mathrm{x}_{\mathrm{n}-1}}(\mathrm{res}_{\mathrm{x}_\mathrm{n}}(\mathrm{f}_1, \mathrm{f}_2), \mathrm{res}_{\mathrm{x}_\mathrm{n}}(\mathrm{f}_1, \mathrm{f}_\mathrm{i}))$ can be replaced by $\mathrm{res}(\mathrm{f}_1, \mathrm{f}_2, \mathrm{f}_\mathrm{i})$ in the second projection, and so on.

# Improvements to Complexity



If the $f_i$ have degree $d$ in each $x_i$, then an iterated resultant after $k$ eliminations has degree $O((2d)d^{2^k})$ (doubly exponential), whereas $\mathrm{res}(f_1, \ldots, f_k)$ has degree $O\left(d^k\right)$ (the Bézout bound).

We note that [EBD15] observed that use of $k$ equational constraints reduces the double exponent of $m$ from $n$ to $n-k$; the present observations show that the same reduction applies to the double exponent of $d$, at least *inasmuch as the nested resultants are concerned*.

**Work to be done:** Prove the same conclusions would apply to equational constraints with the Lazard projection [DNSU23]. There are challenges with "curtains" [Nai21] (regions of nullification).

# Cylindrical Algebraic Coverings



In CAC [ADEK21], each polynomial has (at least one) explicit reason for being where it is in the computation.

For example, $\mathrm{res}_{\mathrm{x_n}}(\mathrm{f_1}, \mathrm{f_2})$ might be in the computation because of a specific root $\alpha$, where it is the case for $x_{n-1} > \alpha$ (until the next point) the regions ruled out by $f_1$ and $f_2$ overlap, whereas for $x_{n-1} < \alpha$ we need a further reason to rule out regions. The same might be true of $\mathrm{res}_{\mathrm{x_n}}(\mathrm{f_1}, \mathrm{f_3})$, needed because of a specific root $\beta$. Then $\mathrm{res}_{\mathrm{x_{n-1}}}(\mathrm{res}_{\mathrm{x_n}}(\mathrm{f_1}, \mathrm{f_2}), \mathrm{res}_{\mathrm{x_n}}(\mathrm{f_1}, \mathrm{f_3}))$ tracks where $\alpha$ and $\beta$ meet. Hence in this context we are interested only in genuine roots, and so we could replace the iterated resultant by $\mathrm{res}(\mathrm{f_1}, \mathrm{f_2}, \mathrm{f_3})$.

**Work to be done:** Work this through precisely with an implementation of CAC.

# Outline

# Detecting Spurious Factors



How to know if a factor is "spurious" or "genuine". Any alternative to manually checking for whether they lead to common zeros?

In some cases we can discard factors with based on their degree, when this breaches the Bézout Bound on the true multivariate resultant. I.e., if $\mathrm{res_y}(\mathrm{res_z}(f, g), \mathrm{res_z}(f, h))$ has an irreducible factor of degree $> d^3$, it *must* be spurious and can be discarded.

Since it is common for CAD implementation to factor polynomials, this is a cheap, albeit incomplete, test.

# Example of Detecting Spurious Factors



Three 3-variable polynomials created randomly in Maple to have total degree 5:

$$\begin{aligned} f &= -34x^2z^3 - 20y^5 + 7x^2y^2 - 43y^3z + 63x + 16z, \\ g &= 13xz^4 - 27z^4 - 21xy^2 + 30yz - 42x - 81, \\ h &= -65xz^4 + 13z^5 + 30x^3z + 17xy^3 + 25yz + 78. \end{aligned}$$

Then $res_y(res_z(f,g), res_z(f,h))$ factors into a constant times two irreducible polynomials: one of degree 378 and the other of degree 89. With no further computation we can identify the first as spurious since its degree is greater than $5^3 = 125$. The second could be genuine, or be another spurious factor: we check manually to see it indeed genuine.

# Bones of a Detection Algorithm



Work through factors in turn:

- if degree is above the bound then discard;
- if below then analyse if there is a genuine multiple root above.
  - If so, mark as genuine and reduce the bound by the degree.
  - Should the bound be set to zero discard all remaining factors.

**Work to be done:** Implement and experiment with this.

# More Work to be Done I



We have only looked at the resultants, not the discriminants, and indeed only at resultants of resultants. Undoubtedly something similar can be said about e.g. $\mathrm{res_y}(\mathrm{res_z}(f,g),\mathrm{disc_z}(f))$.

A complete solution for resultants of discriminants, discriminants of resultants and discriminants of discriminant would bring the double exponent in the degree complexity down entirely.

# More Work to be Done II



In the example iterated resultant (5), the “genuine” part had two factors, one with no real roots. I.e. Even the “genuine” part may still be overkill for *real* geometry.

Can we:

a) detect that a factor has no real components; and
b) use this to further reduce the polynomials? Furthermore,
c) can we make any meaningful statement about the complexity implications of this?

# The End

## Thanks for Listening
## Any Questions?

# Bibliography I

E. Ábrahám, J.H. Davenport, M. England, and G. Kremer.
Deciding the Consistency of Non-Linear Real Arithmetic Constraints with a Conflict Driven Search Using Cylindrical Algebraic Coverings.
*Journal of Logical and Algebraic Methods in Programming Article 100633*, 119, 2021.

C.W. Brown and J.H. Davenport.
The Complexity of Quantifier Elimination and Cylindrical Algebraic Decomposition.
In C.W. Brown, editor, *Proceedings ISSAC 2007*, pages 54–60, 2007.

# Bibliography II

G.E. Collins.
Quantifier Elimination for Real Closed Fields by Cylindrical Algebraic Decomposition.
In *Proceedings 2nd. GI Conference Automata Theory & Formal Languages*, pages 134–183, 1975.

James Harold Davenport, Akshar Sajive Nair, Gregory Kumar Sankaran, and Ali Kemal Uncu.
Lazard-style cad and equational constraints.
In *Proceedings of the 2023 International Symposium on Symbolic and Algebraic Computation*, ISSAC '23, pages 218–226, New York, NY, USA, 2023. Association for Computing Machinery.

# Bibliography III

Tereso del Río and Matthew England.
New heuristic to choose a cylindrical algebraic decomposition variable ordering motivated by complexity analysis.
In François Boulier, Matthew England, Timur M. Sadykov, and Evgenii V. Vorozhtsov, editors, *Computer Algebra in Scientific Computing*, pages 300–317, Cham, 2022. Springer International Publishing.

M. England, R. Bradford, and J.H. Davenport.
Improving the Use of Equational Constraints in Cylindrical Algebraic Decomposition.
In D. Robertz, editor, *Proceedings ISSAC 2015*, pages 165–172, 2015.

# Bibliography IV

D. Jovanović and L. de Moura.
Solving Non-Linear Arithmetic.
In *Proceedings IJCAR 2012*, pages 339–354, 2012.

S. McCallum.
On Propagation of Equational Constraints in CAD-Based Quantifier Elimination.
In B. Mourrain, editor, *Proceedings ISSAC 2001*, pages 223–230, 2001.

McCallum, Scott and Winkler, Franz.
Differential resultants.
*ITM Web Conf.*, 20:01005, 2018.

# Bibliography V

A.S. Nair.
*Exploiting Equational Constraints to Improve the Algorithms for Computing Cylindrical Algebraic Decompositions*.
PhD thesis, University of Bath, 2021.