

Iterated Resultants in CAD

James H. Davenport¹, Matthew England²

¹University of Bath, Faculty of Science, Department of Computer Science, Bath, UK

²Coventry University, Centre for Computational Science and Mathematical Modelling, Coventry, UK

Abstract

Cylindrical Algebraic Decomposition (CAD) by projection and lifting requires many iterated univariate resultants. It has been observed that these often factor, but to date this has not been used to optimise implementations of CAD. We continue the investigation into such factorisations, writing in the specific context of SC^2 .

Keywords

Cylindrical Algebraic Decomposition, Resultant, Gröbner Basis

1. Introduction

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. Resultants are a widely used tool in symbolic computation, and in satisfiability checking over non-linear arithmetic. In particular, they are a key ingredient of Cylindrical Algebraic Decomposition (CAD) [1] which in its traditional projection and lifting form requires many iterated univariate resultant calculations.

[1, pp. 177–178] suggests that iterated resultants, where there are “common ancestors” tend to factor. This was apparently responded to by van der Waerden in a letter [2], which alas we have not seen, but the letter’s contents are taken up again in [3]. There are further developments in [4, 5]. [3] is based on the theory in [6], which [7] notes has been deleted from more recent editions (such as [8]). [4] is based on [7]. Despite this factorisation being observed since the inception of CAD, we are not aware of any optimisations in CAD implementations in regards to it.

The purpose of this paper is to look at the connections of results on such factorisations with Cylindrical Algebraic Decomposition (CAD) [1] and also Cylindrical Algebraic Coverings (CAC) [9], a recent algorithm that was formed out of the SC^2 community via a reworking of CAD theory to better suit the SMT context.

For CAD, we assume that we are constructing a CAD for a specific Boolean formula Φ , rather than just a set of polynomials. For CAC, we again assume we are looking for SAT/UNSAT for a specific Boolean formula Φ .

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

✉ masjhd@bath.ac.uk (J. H. Davenport); Matthew.England@coventry.ac.uk (M. England)

🌐 <https://people.bath.ac.uk/masjhd> (J. H. Davenport); <https://matthewengland.coventry.domains> (M. England)

🆔 0000-0002-3982-7545 (J. H. Davenport); 0000-0001-5729-3420 (M. England)



© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

2. Theory

We are grateful to [10] for a clear exposition of the results in [7], which we have borrowed.

Definition 1. *Given r homogeneous polynomials F_1, \dots, F_r in x_1, \dots, x_n , with indeterminate coefficients comprising a set A , an integral polynomial T in these indeterminates (that is, $T \in \mathbf{Z}[A]$) is called an inertia form for F_1, \dots, F_r if $x_i^\tau T \in (F_1, \dots, F_r)$, for suitable i and τ .*

Van der Waerden observes that the inertia forms comprise an ideal I of $\mathbf{Z}[A]$, and he shows further 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, \dots, 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 $F_1 = 0, \dots, F_r = 0$ in some extension of K .

Now consider the case in which we have n homogeneous polynomials in the same number n of variables. Let F_1, \dots, F_n be n generic homogeneous forms in x_1, \dots, x_n of positive total degrees d_1, \dots, d_n . That is, every possible coefficient of each F_i is a distinct indeterminate, and the set of all such indeterminate coefficients is denoted by A . Let I denote the ideal of inertia forms for F_1, \dots, F_n . Proofs of the following two propositions may be found in [11].

Proposition 1. *[10, Proposition 5] I is a nonzero principal ideal of $\mathbf{Z}[A]$: $I = (R)$, for some $R \neq 0$. R is uniquely determined up to sign. We call R the (generic multipolynomial) resultant of F_1, \dots, F_n .*

Proposition 2. *[10, Proposition 6] The vanishing of R for particular F_1, \dots, F_n with coefficients in a field K is necessary and sufficient for the existence of a non-trivial zero of the system $F_1 = 0, \dots, F_n = 0$ in some extension of K .*

The above considerations also lead to the notion of a resultant of n non-homogeneous polynomials in $n - 1$ variables. For a given non-homogeneous $f(x_1, \dots, x_{n-1})$ over K of total degree d , we may write $f = H_d + H_{d-1} + \dots + H_0$, where the H_j are homogeneous of degree j . Then H_d is known as the leading form of f . Recall that the homogenization $F(x_1, \dots, x_n)$ of f is defined by $F = H_d + H_{d-1}x_n + \dots + H_0x_n^d$. Let f_1, \dots, f_n be particular non-homogeneous polynomials in x_1, \dots, x_{n-1} over K of positive total degrees d_i , and with leading forms H_{i,d_i} . We set $\text{res}(f_1, \dots, f_n) = \text{res}(F_1, \dots, F_n)$, where F_i is the homogenization of f_i . Then we have the following (see proof in [11]).

Proposition 3. *[10, Proposition 7] The vanishing of $\text{res}(f_1, \dots, f_n)$ 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.

3. Iterated Resultants: An Example

Consider these polynomials:

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

3.1. First variable ordering

Under variable ordering $z \succ y \succ x$ we may calculate the iterated resultant:

$$\begin{aligned} \text{res}_y(\text{res}_z(f, g), \text{res}_z(f, h)) &= 5x^8 + 16x^7 + 14x^6 - 2x^5 - 12x^4 - 8x^3 + 3x^2 + 2x \\ &= \underbrace{x(5x^3 + 6x^2 - 3x - 2)}_{\text{spurious}} \underbrace{(x^2 + x + 1)(x^2 + x - 1)}_{\text{genuine}}. \end{aligned} \quad (1)$$

We define the meaning of the labels below. An alternative computational path may have calculated similarly

$$\begin{aligned} \text{res}_y(\text{res}_z(f, g), \text{res}_z(g, h)) &= 5x^8 + 16x^7 + 18x^6 + 8x^5 - 5x^4 - 8x^3 - 2x^2 + 1 \\ &= \underbrace{(x^2 + x + 1)(x^2 + x - 1)}_{\text{genuine}} \underbrace{(5x^4 + 6x^3 + x^2 - 1)}_{\text{spurious}}. \end{aligned} \quad (2)$$

The final choice would have been to calculate,

$$\begin{aligned} \text{res}_y(\text{res}_z(f, h), \text{res}_z(g, h)) &= 2x^4 + 4x^3 + 2x^2 - 2 \\ &= 2 \underbrace{(x^2 + x + 1)(x^2 + x - 1)}_{\text{genuine}}. \end{aligned} \quad (3)$$

Up to constants (3) divides (2) and (1), but this need not happen in general. What does happen in general is that, if we consider a Gröbner Basis,

$$\text{Basis}_{\text{plex}}(f, g, h) = \{x^4 + 2x^3 + x^2 - 1, y - x, x^2 + x + z\}, \quad (4)$$

then we see that the basis polynomial in x only divides all three iterated resultants and in fact is $\text{res}(f, g, h)$ in the sense of §2. In this example, it is also (3), but again this need not happen in general.

The labels above are made in regards to the roots of the tagged resultant factors. The roots of the part we have labelled as “genuine” are

$$\{x : \exists y \exists z f(x, y, z) = g(x, y, z) = h(x, y, z) = 0\}, \quad (5)$$

whereas the roots of the part we have labelled as “spurious” are

$$\{x : \exists y (\exists z_1 f(x, y, z_1) = g(x, y, z_1) = 0 \wedge \exists z_2 \neq z_1 f(x, y, z_2) = h(x, y, z_2) = 0)\}. \quad (6)$$

They are “spurious” in the sense that they do not go on to form true triple roots. Nevertheless, they are x values above which the topology changes, so they cannot always be discarded. Note that §2 implies that there is always a neat factorisation (over \mathbf{Z} if that was the original ring) into “genuine” versus “spurious”.

3.2. Second variable ordering

What happens if we take the variables in a different order? In ordering $x \succ y \succ z$ we have:

$$\text{res}_y(\text{res}_x(f, g), \text{res}_x(f, h)) = (z^2 - 1)^2, \quad (7)$$

$$\text{res}_y(\text{res}_x(f, g), \text{res}_x(g, h)) = (z^2 - 1)^4, \quad (8)$$

$$\text{res}_y(\text{res}_x(h, g), \text{res}_x(f, h)) = (z^2 - 1)^4, \quad (9)$$

and

$$\text{Basis}_{\text{plex}(x,y,z)}(f, g, h) = \{z^2 - 1, y^2 + y + z, x - y\}. \quad (10)$$

I.e. no spurious roots were uncovered with this ordering. The question of CAD variable ordering is well studied and known to greatly effect the complexity of CAD both in practice [12] and theory [13]. The introduction of spurious factors in some orderings but not others may be a significant contributing factor to this.

4. When Can Spurious Factors be Discarded?

This section is not a complete classification on when spurious factors may be discarded, but it is a start.

4.1. During CAD with multiple equational constraints

McCallum [14] introduced the concept of multiple equation constraints, i.e. the case when

$$\Phi \equiv f_1 = 0 \wedge f_2 = 0 \wedge \cdots \wedge f_k = 0 \wedge \bar{\Phi}(f_{k+1}, \dots, f_m). \quad (11)$$

Here McCallum projects just $\text{res}_{x_n}(f_1, f_i)$ and $\text{disc}_{x_n}(f_i)$ (as well as various coefficients, which do not contribute to the degree explosion).

But since $f_1 = 0$ and $f_2 = 0$, we know that $\text{res}_{x_n}(f_1, f_2) = 0$ also. Hence all the $\text{res}_{x_n}(f_1, f_i)$ are equational constraints in x_1, \dots, x_{n-1} . Thus the next projection is

$$\text{res}_{x_{n-1}}(\text{res}_{x_n}(f_1, f_2), \text{res}_{x_n}(f_1, f_i)), \quad (12)$$

$\text{res}_{x_{n-1}}(\text{res}_{x_n}(f_1, f_2), \text{disc}_{x_n}(f_i))$ and numerous discriminants.

In this case, we are only interested in the genuine zeros, as away from these the formula will be uniformly false and thus further refinement is unnecessary. So we can replace (12) by $\text{res}(f_1, f_2, f_i)$.

If the f_i have degree d in each x_i , then the equivalent of (12) after k eliminations (i.e. eliminating all equational constraints) has degree $O((2d)d^{2^k})$ (doubly exponential), whereas $\text{res}(f_1, \dots, f_k)$ has degree $O(d^k)$ (the Bézout bound). We note that [15] 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*.

Though it would have to be proved, it seems very likely that the same conclusions would apply to equational constraints with the Lazard projection [16]. Here, there are challenges with “curtains” [17], which are the same as the regions of nullification in [18].

4.2. During CAC

In CAC [9], each polynomial has (at least one) explicit reason for being where it is in the computation. For example, $\text{res}_{x_n}(f_1, f_2)$ might be in the computation because of a specific root α , 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 $\text{res}_{x_n}(f_1, f_3)$, needed because of a specific root β . Then (12) tracks where α and β meet. Hence in this context we are interested only in genuine roots, and again we can replace (12) by $\text{res}(f_1, f_2, f_i)$.

We would need to work this through precisely with an implementation of CAC, which has yet to be done.

5. Detecting Spurious Factors

In the examples above the factors were marked as “spurious” or “genuine” via manual analysis to see if the roots of the factors led to common zeros or not. Are there alternatives to such manual detection?

We note that 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 $\text{res}_y(\text{res}_z(f, g), \text{res}_z(f, h))$ 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, if incomplete, test.

Example 1. *For example, the following three 3-variable polynomials were 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 $\text{res}_y(\text{res}_z(f, g), \text{res}_z(f, h))$ 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 may check manually that it is indeed genuine.

In an example where we have multiple factors below the bound we could work through them in turn keeping count of the sum of degrees of genuine factors as we uncover them, in each case reducing the degree bound accordingly for any further factors to be investigated as genuine.

6. Conclusions

There is much to be done to develop these ideas.

1. In §4.1, we have only looked at the resultants, not the discriminants, and indeed only at resultants of resultants. Undoubtedly something similar can be said about, for example

$$\text{res}_y(\text{res}_z(f, g), \text{disc}_z(f)), \tag{13}$$

but we have not explored this fully yet. We observe that, in the case of the polynomials from Example 1, (13) is a perfect square, and this seems to be true in general. We would need a complete solution for resultants of discriminants, discriminants of resultants and discriminants of discriminants in order to need to remove the caveat in italics towards the end of §4.1.

2. As stated in §4.2, the “genuine parts of resultants” idea would need to be worked through an implementation of CAC.
3. If we look at (3), we see that this polynomial, which is the “genuine” part, factors further, and one factor has no real roots. Hence this factor can be discarded, though there is not much benefit, since we are at the univariate phase. Nevertheless, this shows that even the “genuine” part may still be overkill for *real* geometry. Can we
 - a) detect that a factor of a resultant etc. 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?

Acknowledgements

Both authors are supported by the UK’s EPSRC, via the DEWCAD Project, *Pushing Back the Doubly-Exponential Wall of Cylindrical Algebraic Decomposition*; grant numbers EP/T015713/1 and EP/T015748/1.

We are also grateful to Gregory Sankaran and Ali Uncu for many useful conversations.

References

- [1] G. Collins, Quantifier Elimination for Real Closed Fields by Cylindrical Algebraic Decomposition, in: Proceedings 2nd. GI Conference Automata Theory & Formal Languages, volume 33 of *Springer Lecture Notes in Computer Science*, 1975, pp. 134–183. doi:10.1007/3-540-07407-4_17.
- [2] B. van der Waerden, About [1], Private communication to G.E. Collins (1975).
- [3] S. McCallum, Factors of Iterated Resultants and Discriminants, *J. Symbolic Comp.* 27 (1999) 367–385. doi:10.1006/jSCO.1998.0257.
- [4] L. Busé, B. Mourrain, Explicit Factors of Some Iterated Resultants and Discriminants, *Math. Comp.* 78 (2009) 345–386. doi:10.1090/S0025-5718-08-02111-X.
- [5] D. Lazard, S. McCallum, Iterated Discriminants, *J. Symbolic Comp.* 44 (2009) 1176–1193. doi:10.1016/j.jSC.2008.05.006.
- [6] B. van der Waerden, *Modern Algebra Vol. II* (trans. F. Blum), Frederick Ungar (1950).
- [7] J. Jouanolou, Le Formalisme du Résultant, *Advances in Mathematics* 90 (1991) 117–263. doi:10.1016/0001-8708(91)90031-2.
- [8] B. van der Waerden, *Modern Algebra Vol. II* (trans. F. Blum and J.R. Schulenberger), Frederick Ungar (1970).
- [9] E. Ábrahám, J. Davenport, M. England, 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* 119 (2021), Article 100633. doi:10.1016/j.jlamp.2020.100633.
- [10] S. McCallum, F. Winkler, Differential Resultants, *ITM Web of Conferences Article* 01005 20 (2018). doi:10.1051/itmconf/20182001005.
- [11] S. McCallum, F. Winkler, Resultants: Algebraic and Differential, Technical Report RISC18-08 Johannes Kepler University, 2018.
- [12] T. del Rio, M. England., New Heuristic to Choose a Cylindrical Algebraic Decomposition Variable Ordering Motivated by Complexity Analysis, in: F. Boulier, M. England, T.M. Sadykov, and E.V. Vorozhtsov (Ed.) *Computer Algebra in Scientific Computing (Proc. CASC 2022)*, volume 13366 of *Springer Lecture Notes in Computer Science*, 2022, pp. 300–317. doi:10.1007/978-3-031-14788-3_17.
- [13] C. Brown, J. Davenport, The Complexity of Quantifier Elimination and Cylindrical Algebraic Decomposition, in: C. Brown (Ed.), *Proceedings ISSAC 2007*, 2007, pp. 54–60. doi:10.1145/1277548.1277557.
- [14] S. McCallum, On Propagation of Equational Constraints in CAD-Based Quantifier Elimination, in: B. Mourrain (Ed.), *Proceedings ISSAC 2001*, 2001, pp. 223–230. doi:10.1145/384101.384132.
- [15] M. England, R. Bradford, J. Davenport, Improving the Use of Equational Constraints in Cylindrical Algebraic Decomposition, in: D. Robertz (Ed.), *Proceedings ISSAC 2015*, 2015, pp. 165–172. doi:10.1145/2755996.2756678.
- [16] J. Davenport, A. Nair, G. Sankaran, A. Uncu, Lazard-style CAD and Equational Constraints, in: G. Jeronimo (Ed.), *Proceedings ISSAC 2023*, 2023, pp. 218–226. doi:10.1145/3597066.3597090.
- [17] A. Nair, Curtains in Cylindrical Algebraic Decomposition, Ph.D. thesis, University of Bath, 2021. URL: <https://researchportal.bath.ac.uk/en/studentTheses/curtains-in-cylindrical-algebraic-decomposition>.
- [18] S. McCallum, An Improved Projection Operation for Cylindrical Algebraic Decomposition, Ph.D. thesis, University of Wisconsin-Madison Computer Science, 1984. URL: <https://www.proquest.com/openview/5a1e6630f4ac77995c62a2fb31f0e9ac/1>.