Projection for Cylindrical Algebraic Decomposition
An Annotated Bibliography

Christopher W. Brown, United States Naval Academy

April 29, 2025

1 Introduction

This document is an annotated bibliography of papers dealing with projection in Cylindrical Algebraic Decomposition (CAD). Its intended audience is people already familiar with the concepts and terminology of CAD, and the discussions appearing here are written with that audience in mind. Here are some important points to note when comparing and contrasting different projections:

1. There are (in my view) three families of projections: Collins-style, McCallum-style and Lazard-style projections. My organization of the references is organized this way. One could possibly add a fourth family based on triangular decomposition.

2. A projection operator cannot be wholly separated from the process of constructing cell representations from the projection. For example, using "Lazard projection" requires "Lazard lifting", while using the "model-based projection" requires a cell construction procedure that is "model-based", i.e. that starts with a sample point and then computes polynomials from it, rather than first computing polynomials and then constructing sample points from them. So, although the language of the paper may refer to a particular projection, implicitly it is also referring to the associated cell-construction process.

3. One issue that comes up when discussing different projections is *completeness*, which is generally taken to mean that, starting with a formula of the appropriate kind, the projection is always able to produce a truth-invariant CAD for the formula. In the model-based paradigm, "complete" typically means that there is a complete non-model-based projection such that the cell the model-based algorithm produces around the model point is a superset of the cell from the CAD the non-model-based projection would produce from the same formula that contains the model point.

4. An issue that is, in my view, often overlooked is the question of constructing formulas from CADs or of constructing formulas for individual cells. This issue depends on the target language for these formulas as well as the projection operator used and properties of the specific CAD from which a formula is to be constructed. Typically, these formulas are either required to be *Tarski formulas* or are allowed to also contain *indexed-root expressions*. In this work, we will say that a CAD-with-truth-values is *projection definable for X*, where "X" is a type of formula, e.g. "Tarski formula", if a formula of type "X" defining the union of the true cells can be constructed using only projection factors.

The remainder of this document consists of sections for the three families of projection operators: Collins, McCallum and Lazard. Cross-cutting issues, e.g., model-based versions of projections, appear based on which family of projection operator is used.

2 Collins-style projection

George Collins introduced CAD in the early 1970s, and described it in a series of presentations and papers:

[14] : G. E. Collins. "Efficient quantifier elimination for elementary algebra". Abstract presented at Symposium on Complexity of Sequential and Parallel Algorithms, Carnegie-Mellon University. May 1973

[16] : G. E. Collins. "Quantifier Elimination for Real Closed Fields by Cylindrical Algebraic Decomposition—Preliminary Report". In: *Proceedings of EUROSAM 74* 8.3 (Aug. 1974), pp. 80–90

[17] : G. E. Collins. "Quantifier Elimination for the Elementary Theory of Real Closed Fields by Cylindrical Algebraic Decomposition". In: *Lecture Notes In Computer Science*. Vol. 33. Reprinted in [**Caviness˙Johnson:98**]. Springer-Verlag, Berlin, 1975, pp. 134–183

His projection operator was based on *sign invariance*, and was coupled with a lifting procedure based on partial evaluation of polynomials (at rational and algebraic points) and univariate polynomial real root isolation. The projection adds coefficients, discriminants, resultants, subresultant coefficients and sub-discriminant coefficients[1] This projection/lifting scheme is *complete* in the sense that for any input Tarski formula, it produces a CAD with cells in which the polynomials from the formula have constant sign, and thus the formula has constant truth value. When the "augmented projection" is used, each cell has a defining Tarski formula that can be constructed from projection factors and

[1] Often overlooked is the fact that, when constructing a CAD for quantifier elimination, he included an even larger "augmented projection" to be used for projections in free-variable space to ensure that a solution formula consisting solely of polynomial sign constraints could be constructed.

their derivatives. However, I know of no implementation that uses the augmented projection. Without the augmented projection, the resulting CAD is not, in general, projection definable for Tarski formulas, though it is projection definable for Tarski formulas + indexed root expressions.

An improved version of the Collins projection was described by Hong:

[22] : H. Hong. "An Improvement of the Projection Operator in Cylindrical Algebraic Decomposition". In: *Proc. International Symposium on Symbolic and Algebraic Computation.* 1990, pp. 261–264

The Hong projection follows the basic principles of the Collins projection, but avoids some coefficients, subdiscriminant coefficients and subresultant coefficients by providing criteria that allow one to determine that they are unnecessary. This produces CADs with fewer cells, and does so in less time. The resulting projection/lifting scheme is still complete. Like the Collins projection without augmented projection, the Hong projection produces CADs that are not, in general, projection definable for Tarski formulas, though they are projection definable for Tarski formulas + indexed root expressions.

The first model-based projection was based on the Hong projection and introduced in:

[25] : Dejan Jovanović and Leonardo de Moura. "Solving Non-linear Arithmetic". In: *Automated Reasoning.* Ed. by Bernhard Gramlich, Dale Miller, and Uli Sattler. Vol. 7364. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, pp. 339–354. ISBN: 978-3-642-31364-6

[24] : Dejan Jovanovic, Clark Barrett, and Leonardo de Moura. "The design and implementation of the model constructing satisfiability calculus". In: *2013 Formal Methods in Computer-Aided Design.* Oct. 2013, pp. 173–180. DOI: `10.1109/FMCAD.2013.7027033`

Its authors dubbed it *the model-based projection*, but because we want to use that term to refer to a number of projection operators that act similarly, we will refer to it here as the *Jovanović and de Moura Projection.* Model-based projections disrupt the project-then-lift paradigm of Collins' original CAD construction algorithm and the variants that had followed it up to that point. In model-based projection, we start with a "model point", and the goal is to construct a (hopefully large) CAD cell containing the model point and satisfying some other properties. In the Jovanović and de Moura Projection, those other properties are the sign-invariance of an initial set of polynomials. This projection uses the model point to determine that certain polynomials that are included in the Hong projection can be left out. For example, in projecting polynomial $p := xz^2 + (x - y)z + xy - 1$ with respect to z using model point (1,1,1), the Hong projection would include x and $x - y$, the coefficient of the quadratic and linear terms of p as a polynomial in z. Since leading coefficient x is non-zero at the model point and x is included in the projection, the Jovanović and de Moura Projection deduces that coefficient $x - y$ can be left out of the projection without losing the desired properties of the cell that will be constructed around the model point.

Based on the Collins-Hong projection, Seidl and Sturm propose what they call a "generic" projection operator, which is related to the idea of open CAD (discussed in the section on McCallum-style projection).

[38] : A. Seidl and T. Sturm. "A generic projection operator for partial cylindrical algebraic decomposition". In: *Proc. International Symposium on Symbolic and Algebraic Computation.* Ed. by R. Sendra. 2003, pp. 240–247

The idea behind "generic" projection and "generic" CAD is that one constructs a partial CAD and a list L of polynomials containing only parameters, and the CAD of parameter space is sign-invariant for the polynomials in L, but only required to be truth-invariant (with respect to the quantified input formula) for cells in which all the elements of L are non-zero. The Collins-Hong projection includes a stopping criterion based on sequences of leading coefficients of the polynomials to be projected. In generic projection, when assuming the non-vanishing of a polynomial purely in parameters allows the stopping criterion to be used early, that polynomial is added to L. So one strategically chooses polynomials to add to L based on their allowing for a smaller projection.

3 McCallum-style projection

The McCallum projection is described in:

[28] : S. McCallum. "An Improved Projection Operator for Cylindrical Algebraic Decomposition". PhD thesis. University of Wisconsin-Madison, 1984

[27] : S. McCallum. "An Improved Projection Operation for Cylindrical Algebraic Decomposition of Three-dimensional Space". In: *Journal of Symbolic Computation* 5.1,2 (1988), pp. 141–161

[29] : S. McCallum. "An Improved Projection Operator for Cylindrical Algebraic Decomposition". In: *Quantifier Elimination and Cylindrical Algebraic Decomposition.* Ed. by B. Caviness and J. Johnson. Texts and Monographs in Symbolic Computation. Springer-Verlag, Vienna, 1998

The McCallum projection is based on *order invariance* and is coupled with a lifting procedure that follows Collins' original lifting procedure with two differences: (1) when a projection factor derived from a projection step (as opposed to appearing solely as an input polynomial) is nullified over a cell of positive dimension, the lifting procedure exits with "fail", and (2) when a projection factor p derived from a projection step is nullified over a cell c of dimension zero, instead of including the roots of $p(c, x)$ as sections in the decomposition of $c \times \mathbb{R}$, a set called *the delineating polynomials* are constructed, and their roots above c are used as sections. The set of delineating polynomials consists of all partial derivatives of p of some order k, where k is determined by p and c.

The McCallum projection includes coefficients, discriminants and resultants, and so is a subset of the Collins projection and the Hong projection. It has the advantage of producing a smaller projection. However, it is not complete. As described above, it may fail during the lifting process. (Note that if it returns without failure the decomposition is guaranteed to be correct.)

The question of solution formula construction is interesting. If the McCallum projection does not fail, and it does not need to use delineating polynomials, the resulting CAD will be projection definable for Tarski formulas + indexed root expressions, just as with Collins-style projection. However, if the CAD has a cell that is a section of a delineating polynomial and not a section of any projection factor, then that cell (and the sectors above and below it) cannot be described by a formula including only projection factors. However, it can be described by a formula using indexed roots of projection factors and an indexed-root expression for one of the delineating polynomials of which it is a section. Moreover, there is a procedure for refining the CAD until it is projection definable by Tarski formulas. This is described in [4].

Nullification of a projection factor is detected and dealt with during lifting. The technical report:

[7] : C. W. Brown. "The McCallum Projection, Lifting, and Order-Invariance". See `https://www.usna.edu/ComputerScience/cstech/reports/2005-02.pdf`. Sept. 2001

... describes techniques for determining, in some instances, that nullification does not invalidate projection/lifting, and it provides an alternative to McCallum's delineating polynomials for handling nullifications of projection factors over positive-dimensional regions.

An improved version of McCallum's projection, which we will refer to here as the Brown-McCallum projection, was described in

[5] : C. W. Brown. "Improved Projection for CAD's of $\mathbf{R}^3$". In: *Proc. International Symposium on Symbolic and Algebraic Computation.* 2000, pp. 48–53

[6] : C. W. Brown. "Improved Projection for Cylindrical Algebraic Decomposition". In: *Journal of Symbolic Computation* 32.5 (Nov. 2001), pp. 447–465

... which shows that when projecting polynomial $p(x_1, \ldots, x_n)$, the leading coefficient in x_n is the only coefficient required for McCallum's projection, provided that (1) some analysis of the system of all coefficients in x_n of p is done prior to projection, and (2) if that system has only single-point solutions, those points are added as "projection points". Of course if the system has a positive-dimensional set of solutions then McCallum's original projection returns "fail", as does the improved projection. The Brown-McCallum projection is a subset of McCallum's projection, but it requires the up-front analysis of systems of coefficients.

This can be done efficiently relative to the cost of including extra coefficients in projection, but adds to the complexity of implementations. Given a CAD resulting from the Brown-McCallum projection, a defining formula can be constructed from indexed-root expressions of projection factors plus indexed-root expressions for the coordinates of the "projection points".

McCallum's projection has been improved, specialized and adapted in a number of ways. In the following we consider several categories: "equational constraints", model-based versions, a "projective" version, and adaptations to "open" CAD construction.

3.1 Equational constraints and McCallum's projection

Improving the performance of CAD construction by exploiting "equational constraints" was suggested by Collins in

[15] : G. E. Collins. "Quantifier Elimination by Cylindrical Algebraic Decomposition - 20 Years of Progress". In: *Quantifier Elimination and Cylindrical Algebraic Decomposition.* Ed. by B. Caviness and J. Johnson. Texts and Monographs in Symbolic Computation. Springer-Verlag, 1998

... though only as a suggested avenue of inquiry. What constitutes an "equational constraint" is technical enough that I'll leave it to the papers — except to say that, for polynomial p, if the input formula has the property that $p(\alpha) = 0$ for any point α satisfying the formula, then p is an equational constraint for that formula. McCallum developed this idea and proved basic results about how to exploit equational constraints during projection and lifting.

[30] : S. McCallum. "On Projection in CAD-Based Quantifier Elimination with Equational Constraint". In: ed. by Sam Dooley. ISSAC '99. 1999, pp. 145–149

[31] : S. McCallum. "On propagation of equational constraints in CAD-based quantifier elimination". In: ed. by Bernard Mourrain. ISSAC '01. 2001, pp. 223–230

The first of these papers shows how an n-level equational constraint polynomial can be used to reduce the projection at level n. The second of these shows how, in the presence of multiple n-level equational constraints, new equational constraints at lower levels are created by the projection process. These are called propagated constraints, and while top-level equational constraint polynomials typically only need to be sign invariant within cells, propagated constraints need to be order invariant. So propagated constraints need to be handled differently than top-level constraints.

Further development of McCallum's approach, including results on how the complexity of CAD construction is improved by exploiting equational constraints, appears in:

[19] : Matthew England, Russell Bradford, and James H. Davenport. "Cylindrical algebraic decomposition with equational constraints". In: *Journal* *of Symbolic Computation* 100 (2020). Symbolic Computation and Satisfiability Checking, pp. 38–71. ISSN: 0747-7171. DOI: `https://doi.org/10.1016/j.jsc.2019.07.019`. URL: `https://www.sciencedirect.com/science/article/pii/S0747717119300859`

When there are multiple equational constraint polynomials at the same level, the original approach to equational constraints designates one of them as *the* constraint for projection purposes. An alternative is to try to formulate a projection operator that can leverage all constraints at that level to reduce the projection. The following two papers follow that approach for the case of two equational constraints.

[12] : Christopher W. Brown and Scott McCallum. "On using bi-equational constraints in CAD construction". In: *ISSAC '05: Proceedings of the 2005 international symposium on Symbolic and algebraic comp utation.* Beijing, China: ACM Press, 2005, pp. 76–83. ISBN: 1-59593-095-7. DOI: `http://doi.acm.org/10.1145/1073884.1073897`

[33] : Scott McCallum and Christopher W. Brown. "On delineability of varieties in CAD-based quantifier elimination with two equational constraints". In: *Proceedings of the 2009 international symposium on Symbolic and algebraic computation.* Seoul, Republic of Korea: ACM, 2009, pp. 71–78. ISBN: 978-1-60558-609-0. DOI: `10.1145/1576702.1576715`. URL: `http://portal.acm.org/citation.cfm?id=1576702.1576715`

The view of CAD construction from Collins' original paper is that we start with a set of polynomials and produce a CAD such that the sign of each input polynomial is invariant on each cell, commonly referred to as a *sign-invariant CAD*. However, we often start with an input formula, and require the CAD such that the formula has constant truth value within each cell, commonly referred to as a *truth-invariant CAD*. *Partial CAD*, introduced in H. Hong. "Improvements in CAD–based Quantifier Elimination". PhD thesis. The Ohio State University, 1990, gives an example of a CAD that is truth invariant but not necessarily sign invariant for the set of polynomials appearing in the input formula. A *Truth Table Invariant CAD* (TTICAD) is defined in terms of a set of formulas, and requires that the CAD is truth invariant for each formula in the set. One benefit of this view is that it allows one to exploit an equational constraint that only applies to only some of the formulas in the set. If one were, for example, ultimately interested in the disjunction of the elements of the set, such a constraint could not be exploited by the existing theory of equational constraints. This work is described in:

[3] : Russell Bradford et al. "Truth table invariant cylindrical algebraic decomposition". In: *Journal of Symbolic Computation* 76 (2016), pp. 1–35. ISSN: 0747-7171. DOI: `https://doi.org/10.1016/j.jsc.2015.11.002`. URL: `https://www.sciencedirect.com/science/article/pii/S0747717115001005`

Very recent work by Rizeng Chen generalizes the equational constraints projection, treating all constraints symmetrically, as opposed to designating one as a "pivot".

[13] : Rizeng Chen. *A Geometric Approach to Cylindrical Algebraic Decomposition.* 2025. arXiv: `2311.10515 [math.AG]`. URL: `https://arxiv.org/abs/2311.10515`

Thanks to Hoon Hong for pointing this out. A journal version should be forthcoming, but I do not have bibliographic information for this yet.

3.2 Model-based variants of McCallum's projection

The model-based projection introduced in Jovanović and de Moura's NLSAT algorithm was adaptation of The Collins-Hong projection to make use of the fact that, ultimately, only a single cell was being constructed and, since the sample point for that cell is given as input, its coordinates can give information about polynomials at or near that point. This same idea has been explored in the context of McCallum's projection as well. Strzebonski proposed a model-based version of McCallum's projection to be applied to the general problem of constructing CADs in:

[40] : Adam Strzeboński. "Cylindrical Algebraic Decomposition Using Local Projections". In: ISSAC '14. Kobe, Japan: ACM, 2014, pp. 389–396. ISBN: 978-1-4503-2501-1. DOI: `10.1145/2608628.2608633`. URL: `http://doi.acm.org/10.1145/2608628.2608633`

A model-based version of McCallum's projection applied specifically to the problem of constructing a single cell around a model point was given, first for open cells, and then for general cells in:

[8] : Christopher W. Brown. "Constructing a single open cell in a cylindrical algebraic decomposition". In: ISSAC '13. Boston, Maine, USA: ACM, 2013, pp. 133–140. ISBN: 978-1-4503-2059-7

[10] : Christopher W. Brown and Marek Košta. "Constructing a single cell in cylindrical algebraic decomposition". In: *Journal of Symbolic Computation* 70 (2015), pp. 14–48. ISSN: 0747-7171. DOI: `http://doi.org/10.1016/j.jsc.2014.09.024`. URL: `http://www.sciencedirect.com/science/article/pii/S0747717114000923`

In their introduction of Cylindrical Algebraic Coverings, Ábrahám et al. give a different model-based variant of the McCallum projection for the problem of constructing a "covering" by generalized intervals, rather than a single cell:

[1] : Erika Ábrahám et al. "Deciding the consistency of non-linear real arithmetic constraints with a conflict driven search using cylindrical algebraic coverings". In: *Journal of Logical and Algebraic Methods in Programming* 119 (2021), p. 100633

3.3 Projective delineability

Michel et al. propose a variant of McCallum's projection aimed at providing, not delineability, but "projective delineability". In the usual sense of delineability, we have a total ordering of cells decomposing $S \times \mathbb{R}$, where S is some base cell. In projective delineability, we have a cyclic ordering of cells decomposing $S \times \mathbb{P}$, where $\mathbb{P}$ is real projective space.

[36] : Lucas Michel et al. "On Projective Delineability". In: *2024 26th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC) : [Proceedings]*. 26. International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, Timisoara (Romania), 16 Sep 2024 - 19 Sep 2024. Pisctaway, NJ: IEEE, Sept. 16, 2024. DOI: `10.1109/SYNASC65383.2024.00015`. URL: `https://publications.rwth-aachen.de/record/1005631`

3.4 McCallum-style projection for "open" CAD

An "Open CAD" is a form of partial CAD in which only the full dimensional cells (which are exactly the cells that are open sets) matter, where "matter" is problem dependent. For example, in deciding the satisfiability of a formula in which all atoms are strict inequalities ($<$, $\leq$, or $\neq$), only the full dimensional cells "matter" because there is a solution if and only if there is an open set of solutions. So we can answer the question without considering any cell of less than full dimension. Projection (and lifting) becomes a lot easier when we restrict to only cells of full dimension. There have been several papers that have looked at adapting McCallum's projection to this easier case.

[32] : Scott McCallum. "Solving polynomial strict inequalities using cylindrical algebraic decomposition". In: *The Computer Journal* 36.5 (1993), pp. 432–438

[39] : A. Strzebonski. "An algorithm for systems of strong polynomial inequalities". In: *The Mathematica Journal* 4 (1994), pp. 74–77

[9] : Christopher W. Brown. "Model-based construction of Open Non-uniform Cylindrical Algebraic Decompositions". In: *CoRR* abs/1403.6487 (2014). URL: `http://arxiv.org/abs/1403.6487`

[2] : Philipp Bär et al. "Exploiting Strict Constraints in the Cylindrical Algebraic Covering". In: *CoRR* abs/2306.16757 (2023). DOI: `10.48550/ARXIV.2306.16757`. arXiv: `2306.16757`. URL: `https://doi.org/10.48550/arXiv.2306.16757`

Taking Open CAD as a starting point, related decompositions with even fewer sample points are described in:

[20] : Jingjun Han, Liyun Dai, and Bican Xia. "Constructing Fewer Open Cells by GCD Computation in CAD Projection". In: ISSAC '14. Kobe, Japan: ACM, 2014, pp. 240–247. ISBN: 978-1-4503-2501-1. DOI: 10.1145/2608628.2608676. URL: http://doi.acm.org/10.1145/2608628.2608676

[21] : Jingjun Han et al. "Open weak CAD and its applications". In: *Journal of Symbolic Computation* 80 (2017), pp. 785–816. ISSN: 0747-7171. DOI: https://doi.org/10.1016/j.jsc.2016.07.032. URL: https://www.sciencedirect.com/science/article/pii/S0747717116300797

4 Lazard-style projection

In the 1990s, Daniel Lazard proposed a projection that was a strict improvement on McCallum's projection in two ways: 1) that the projection set is a subset of McCallum's, and 2) that it is valid for any input set of polynomials, even when nullification occurs.

[26] : D. Lazard. "An Improved Projection for Cylindrical Algebraic Decomposition". In: *Algebraic Geometry and its Applications*. Ed. by C. L. Bajaj. Collections of Papers from Abhyankar's 60th Birthday Conference. Springer-Verlag, 1994, pp. 467–476

However, the original article contained an invalid proof, and it took about 25 years for a correct proof to be worked out (see below). This projection is based not on *sign invariance* or *order invariance*, but rather on the invariance of the *Lazard valuation* of projection factors in cells. Instead of *delineability* or *analytic delineability*, there is *Lazard delineability*. It also introduced a new lifting mechanism based not only on substitution and univariate real root isolation, but on *Lazard evaluation*. With these concepts, Lazard proposed a projection consisting of resultants, discriminants and leading and trailing coefficients of projection polynomials. This is a subset of McCallum's original projection, which included all coefficients (though not of the Brown-McCallum projection).

The completeness of Lazard's projection/lifting makes it attractive. It does, however, have the downside that CADs produced by Lazard's method are not, in general, projection definable for Tarski formulas or for Tarski formulas + indexed root expressions (or at least there is no algorithm at present). Instead, one needs to use "Lazard-valuation-indexed root expressions", which are non-standard. Though, in fairness, for many problem instances in which this issue occurs, McCallum-based methods would fail anyways.

The correctness of Lazard's projection was established in the two-paper sequence:

[34] : Scott McCallum and Hoon Hong. "On using Lazard's projection in CAD construction". In: *Journal of Symbolic Computation* 72 (2016), pp. 65–81. ISSN: 0747-7171. DOI: https://doi.org/10.1016/j.jsc.2015.02.001. URL: https://www.sciencedirect.com/science/article/pii/S0747717115000097

[35] : Scott McCallum, Adam Parusiński, and Laurentiu Paunescu. "Validity proof of Lazard's method for CAD construction". In: *Journal of Symbolic Computation* 92 (2019), pp. 52–69. ISSN: 0747-7171. DOI: `https://doi.org/10.1016/j.jsc.2017.12.002`. URL: `https://www.sciencedirect.com/science/article/pii/S0747717117301268`

Soon thereafter, a Lazard analogue to the Brown-McCallum projection was given, i.e. testable conditions under which trailing coefficients do not need to be included in Lazard's projection.

[11] : Christopher W. Brown and Scott McCallum. "Enhancements to Lazard's Method for Cylindrical Algebraic Decomposition". In: *Computer Algebra in Scientific Computing - 22nd International Workshop, CASC 2020, Linz, Austria, September 14-18, 2020, Proceedings.* Ed. by François Boulier et al. Vol. 12291. Lecture Notes in Computer Science. Springer, 2020, pp. 129–149. DOI: `10.1007/978-3-030-60026-6\_8`. URL: `https://doi.org/10.1007/978-3-030-60026-6%5C_8`

Given the success of specialized versions of McCallum's projection that take advantage of "equational constraints", one might hope that something similar could be developed for Lazard's projection. The following two papers make progress in that direction.

[37] : Akshar Nair, James Davenport, and Gregory Sankaran. "Curtains in CAD: Why Are They a Problem and How Do We Fix Them?" In: *Mathematical Software – ICMS 2020: 7th International Conference, Braunschweig, Germany, July 13–16, 2020, Proceedings.* Braunschweig, Germany: Springer-Verlag, 2020, pp. 17–26. ISBN: 978-3-030-52199-8. DOI: `10.1007/978-3-030-52200-1_2`. URL: `https://doi.org/10.1007/978-3-030-52200-1_2`

[18] : James Harold Davenport et al. "Lazard-style CAD and Equational Constraints". In: *Proceedings of the 2023 International Symposium on Symbolic and Algebraic Computation.* ISSAC '23. Tromsø, Norway: Association for Computing Machinery, 2023, pp. 218–226. ISBN: 9798400700392. DOI: `10.1145/3597066.3597090`. URL: `https://doi.org/10.1145/3597066.3597090`

References

[1] Erika Ábrahám et al. "Deciding the consistency of non-linear real arithmetic constraints with a conflict driven search using cylindrical algebraic coverings". In: *Journal of Logical and Algebraic Methods in Programming* 119 (2021), p. 100633.

[2] Philipp Bär et al. "Exploiting Strict Constraints in the Cylindrical Algebraic Covering". In: *CoRR* abs/2306.16757 (2023). DOI: `10.48550/ARXIV.2306.16757`. arXiv: `2306.16757`. URL: `https://doi.org/10.48550/arXiv.2306.16757`.

[3] Russell Bradford et al. “Truth table invariant cylindrical algebraic decomposition”. In: *Journal of Symbolic Computation* 76 (2016), pp. 1–35. ISSN: 0747-7171. DOI: https://doi.org/10.1016/j.jsc.2015.11.002. URL: https://www.sciencedirect.com/science/article/pii/S0747717115001005.

[4] C. W. Brown. “Guaranteed Solution Formula Construction”. In: *Proc. International Symposium on Symbolic and Algebraic Computation.* 1999, pp. 137–144.

[5] C. W. Brown. “Improved Projection for CAD's of $\mathbf{R}^3$”. In: *Proc. International Symposium on Symbolic and Algebraic Computation.* 2000, pp. 48–53.

[6] C. W. Brown. “Improved Projection for Cylindrical Algebraic Decomposition”. In: *Journal of Symbolic Computation* 32.5 (Nov. 2001), pp. 447–465.

[7] C. W. Brown. “The McCallum Projection, Lifting, and Order-Invariance”. See https://www.usna.edu/ComputerScience/cstech/reports/2005-02.pdf. Sept. 2001.

[8] Christopher W. Brown. “Constructing a single open cell in a cylindrical algebraic decomposition”. In: ISSAC '13. Boston, Maine, USA: ACM, 2013, pp. 133–140. ISBN: 978-1-4503-2059-7.

[9] Christopher W. Brown. “Model-based construction of Open Non-uniform Cylindrical Algebraic Decompositions”. In: *CoRR* abs/1403.6487 (2014). URL: http://arxiv.org/abs/1403.6487.

[10] Christopher W. Brown and Marek Košta. “Constructing a single cell in cylindrical algebraic decomposition”. In: *Journal of Symbolic Computation* 70 (2015), pp. 14–48. ISSN: 0747-7171. DOI: http://doi.org/10.1016/j.jsc.2014.09.024. URL: http://www.sciencedirect.com/science/article/pii/S0747717114000923.

[11] Christopher W. Brown and Scott McCallum. “Enhancements to Lazard's Method for Cylindrical Algebraic Decomposition”. In: *Computer Algebra in Scientific Computing - 22nd International Workshop, CASC 2020, Linz, Austria, September 14-18, 2020, Proceedings.* Ed. by François Boulier et al. Vol. 12291. Lecture Notes in Computer Science. Springer, 2020, pp. 129–149. DOI: 10.1007/978-3-030-60026-6_8. URL: https://doi.org/10.1007/978-3-030-60026-6%5C_8.

[12] Christopher W. Brown and Scott McCallum. “On using bi-equational constraints in CAD construction”. In: *ISSAC '05: Proceedings of the 2005 international symposium on Symbolic and algebraic comp utation.* Beijing, China: ACM Press, 2005, pp. 76–83. ISBN: 1-59593-095-7. DOI: http://doi.acm.org/10.1145/1073884.1073897.

[13] Rizeng Chen. *A Geometric Approach to Cylindrical Algebraic Decomposition.* 2025. arXiv: 2311.10515 [math.AG]. URL: https://arxiv.org/abs/2311.10515.

[14] G. E. Collins. “Efficient quantifier elimination for elementary algebra”. Abstract presented at Symposium on Complexity of Sequential and Parallel Algorithms, Carnegie-Mellon University. May 1973.

[15] G. E. Collins. “Quantifier Elimination by Cylindrical Algebraic Decomposition - 20 Years of Progress”. In: *Quantifier Elimination and Cylindrical Algebraic Decomposition.* Ed. by B. Caviness and J. Johnson. Texts and Monographs in Symbolic Computation. Springer-Verlag, 1998.

[16] G. E. Collins. “Quantifier Elimination for Real Closed Fields by Cylindrical Algebraic Decomposition—Preliminary Report”. In: *Proceedings of EUROSAM 74* 8.3 (Aug. 1974), pp. 80–90.

[17] G. E. Collins. “Quantifier Elimination for the Elementary Theory of Real Closed Fields by Cylindrical Algebraic Decomposition”. In: *Lecture Notes In Computer Science.* Vol. 33. Reprinted in [**Caviness˙Johnson:98**]. Springer-Verlag, Berlin, 1975, pp. 134–183.

[18] James Harold Davenport et al. “Lazard-style CAD and Equational Constraints”. In: *Proceedings of the 2023 International Symposium on Symbolic and Algebraic Computation.* ISSAC ’23. Tromsø, Norway: Association for Computing Machinery, 2023, pp. 218–226. ISBN: 9798400700392. DOI: 10.1145/3597066.3597090. URL: https://doi.org/10.1145/3597066.3597090.

[19] Matthew England, Russell Bradford, and James H. Davenport. “Cylindrical algebraic decomposition with equational constraints”. In: *Journal of Symbolic Computation* 100 (2020). Symbolic Computation and Satisfiability Checking, pp. 38–71. ISSN: 0747-7171. DOI: https://doi.org/10.1016/j.jsc.2019.07.019. URL: https://www.sciencedirect.com/science/article/pii/S0747717119300859.

[20] Jingjun Han, Liyun Dai, and Bican Xia. “Constructing Fewer Open Cells by GCD Computation in CAD Projection”. In: ISSAC ’14. Kobe, Japan: ACM, 2014, pp. 240–247. ISBN: 978-1-4503-2501-1. DOI: 10.1145/2608628.2608676. URL: http://doi.acm.org/10.1145/2608628.2608676.

[21] Jingjun Han et al. “Open weak CAD and its applications”. In: *Journal of Symbolic Computation* 80 (2017), pp. 785–816. ISSN: 0747-7171. DOI: https://doi.org/10.1016/j.jsc.2016.07.032. URL: https://www.sciencedirect.com/science/article/pii/S0747717116300797.

[22] H. Hong. “An Improvement of the Projection Operator in Cylindrical Algebraic Decomposition”. In: *Proc. International Symposium on Symbolic and Algebraic Computation.* 1990, pp. 261–264.

[23] H. Hong. “Improvements in CAD–based Quantifier Elimination”. PhD thesis. The Ohio State University, 1990.

[24] Dejan Jovanovic, Clark Barrett, and Leonardo de Moura. “The design and implementation of the model constructing satisfiability calculus”. In: *2013 Formal Methods in Computer-Aided Design.* Oct. 2013, pp. 173–180. DOI: 10.1109/FMCAD.2013.7027033.

[25] Dejan Jovanović and Leonardo de Moura. "Solving Non-linear Arithmetic". In: *Automated Reasoning*. Ed. by Bernhard Gramlich, Dale Miller, and Uli Sattler. Vol. 7364. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, pp. 339–354. ISBN: 978-3-642-31364-6.

[26] D. Lazard. "An Improved Projection for Cylindrical Algebraic Decomposition". In: *Algebraic Geometry and its Applications*. Ed. by C. L. Bajaj. Collections of Papers from Abhyankar's 60th Birthday Conference. Springer-Verlag, 1994, pp. 467–476.

[27] S. McCallum. "An Improved Projection Operation for Cylindrical Algebraic Decomposition of Three-dimensional Space". In: *Journal of Symbolic Computation* 5.1,2 (1988), pp. 141–161.

[28] S. McCallum. "An Improved Projection Operator for Cylindrical Algebraic Decomposition". PhD thesis. University of Wisconsin-Madison, 1984.

[29] S. McCallum. "An Improved Projection Operator for Cylindrical Algebraic Decomposition". In: *Quantifier Elimination and Cylindrical Algebraic Decomposition*. Ed. by B. Caviness and J. Johnson. Texts and Monographs in Symbolic Computation. Springer-Verlag, Vienna, 1998.

[30] S. McCallum. "On Projection in CAD-Based Quantifier Elimination with Equational Constraint". In: ed. by Sam Dooley. ISSAC '99. 1999, pp. 145–149.

[31] S. McCallum. "On propagation of equational constraints in CAD-based quantifier elimination". In: ed. by Bernard Mourrain. ISSAC '01. 2001, pp. 223–230.

[32] Scott McCallum. "Solving polynomial strict inequalities using cylindrical algebraic decomposition". In: *The Computer Journal* 36.5 (1993), pp. 432–438.

[33] Scott McCallum and Christopher W. Brown. "On delineability of varieties in CAD-based quantifier elimination with two equational constraints". In: *Proceedings of the 2009 international symposium on Symbolic and algebraic computation*. Seoul, Republic of Korea: ACM, 2009, pp. 71–78. ISBN: 978-1-60558-609-0. DOI: 10.1145/1576702.1576715. URL: http://portal.acm.org/citation.cfm?id=1576702.1576715.

[34] Scott McCallum and Hoon Hong. "On using Lazard's projection in CAD construction". In: *Journal of Symbolic Computation* 72 (2016), pp. 65–81. ISSN: 0747-7171. DOI: https://doi.org/10.1016/j.jsc.2015.02.001. URL: https://www.sciencedirect.com/science/article/pii/S0747717115000097.

[35] Scott McCallum, Adam Parusiński, and Laurentiu Paunescu. "Validity proof of Lazard's method for CAD construction". In: *Journal of Symbolic Computation* 92 (2019), pp. 52–69. ISSN: 0747-7171. DOI: https://doi.org/10.1016/j.jsc.2017.12.002. URL: https://www.sciencedirect.com/science/article/pii/S0747717117301268.

[36] Lucas Michel et al. "On Projective Delineability". In: *2024 26th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC) : [Proceedings]*. 26. International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, Timisoara (Romania), 16 Sep 2024 - 19 Sep 2024. Pisctaway, NJ: IEEE, Sept. 16, 2024. DOI: 10.1109/SYNASC65383.2024.00015. URL: https://publications.rwth-aachen.de/record/1005631.

[37] Akshar Nair, James Davenport, and Gregory Sankaran. "Curtains in CAD: Why Are They a Problem and How Do We Fix Them?" In: *Mathematical Software – ICMS 2020: 7th International Conference, Braunschweig, Germany, July 13–16, 2020, Proceedings*. Braunschweig, Germany: Springer-Verlag, 2020, pp. 17–26. ISBN: 978-3-030-52199-8. DOI: 10.1007/978-3-030-52200-1_2. URL: https://doi.org/10.1007/978-3-030-52200-1_2.

[38] A. Seidl and T. Sturm. "A generic projection operator for partial cylindrical algebraic decomposition". In: *Proc. International Symposium on Symbolic and Algebraic Computation*. Ed. by R. Sendra. 2003, pp. 240–247.

[39] A. Strzebonski. "An algorithm for systems of strong polynomial inequalities". In: *The Mathematica Journal* 4 (1994), pp. 74–77.

[40] Adam Strzeboński. "Cylindrical Algebraic Decomposition Using Local Projections". In: ISSAC '14. Kobe, Japan: ACM, 2014, pp. 389–396. ISBN: 978-1-4503-2501-1. DOI: 10.1145/2608628.2608633. URL: http://doi.acm.org/10.1145/2608628.2608633.