# The DEWCAD Project: Pushing Back the Doubly Exponential Wall of Cylindrical Algebraic Decomposition

R. Bradford[1], J.H. Davenport[1], M. England[2], A. Sadeghimanesh[2], A. Uncu[1]

[1]University of Bath, U.K.
{R.Bradford, J.H.Davenport, aku21}@bath.ac.uk
[2]Coventry University, U.K.
{Matthew.England, Amirhossein.Sadeghimanesh}@coventry.ac.uk

## Abstract

This abstract seeks to introduce the ISSAC community to the DEWCAD project, which is based at Coventry University and the University of Bath, in the United Kingdom. The project seeks to push back the Doubly Exponential Wall of Cylindrical Algebraic Decomposition, through the integration of SAT/SMT technology, the extension of Lazard projection theory, and the development of new algorithms based on CAD technology but without producing CADs themselves. The project also seeks to develop applications of CAD and will focus on applications in the domains of economics and bio-network analysis.

## 1 Cylindrical Algebraic Decomposition

### 1.1 Introduction

*Cylindrical Algebraic Decomposition* (CAD), was developed by Collins in the 1970s. A CAD is a *decomposition* of ordered $\mathbb{R}^n$ space into cells (i.e. connected subsets) which are *semi-algebraic* so each is described by a sequence of polynomial constraints. Each constraint involves one further variable (locally cylindrical cells) meaning that the bounds of cells can be read easily. For any pair of cells the projections with respect to the given ordering are either equal or disjoint (global cylindricity). See [11] for a detailed exposition.

A CAD is produced relative to Tarski formulae, i.e. logical formulae whose atoms are sign conditions on non-linear polynomial constraints with integer coefficients. The CAD is produced to be *truth-invariant* for the formulae, commonly achieved through being sign-invariant for the polynomials involved. A CAD may then be used to give intuitive descriptions of the solutions of such formulae, and to solve associated problems like real Quantifier Elimination (given a quantified formulae produce an unquantified one which is logically equivalent).

CAD has a large range of potential applications, such as proving collisions of autonomous vehicles impossible [29]; artificial intelligence to pass a university entrance exam [3]; the derivation of optimal numerical schemes [16]; and structural design to minimise the weight of trusses [12].

However, CAD has worst case complexity doubly exponential in the number of indeterminates (both quantified and unquantified variables) [10], meaning that as problem sizes rise you inevitably *hit the doubly exponential wall*. It is thanks to over 40 years of extensive research that it may be used for applications like those above. We seek to push the doubly exponential wall back further still to bring new applications within scope.

### 1.2 Recent Developments

For most of its history CAD has followed an algorithmic framework of projection (identifying key polynomials needed to form the decomposition) and lifting (constructing the cells incrementally by dimension). But in recent years there have been a variety of alternative frameworks proposed: e.g. cylindrical decompositions of complex space which are then refined to CADs [13]; exploiting Boolean structure in the algebraic procedures [7], [15]; [28].

This has led to the development of algorithms and structures which relax the definition of CAD itself. Non-uniform CAD (NuCAD) maintains the locally cylindrical cell descriptions but relaxes the global cylindricity condition of cells being arranged in cylinders [9]. Cylindrical Algebraic Coverings (CACs) [2] produce cells which are arranged cylindrically but may overlap (so they form a covering, not a decomposition). Both offer significant savings over an actual CAD.

These new approaches have been inspired by the search based algorithms employed by SAT/SMT solvers, most notably the NLSAT algorithm [17] which determines satisfiability of Tarski formulae through a combination of Boolean search and the construction of cylindrical cells to rule out portions of the search space generalised from failure at a model point. In turn, the SMT community [4] has started to show interest in the algorithms of computer algebra which may be adapted for their solvers. The SC$^2$ initiative seeks to draw these communities together [1].

## 2 The DEWCAD Project

The DEWCAD project is funded by the UK's Engineering and Physical Sciences Research Council. It runs 2021−2025 and employs the authors, at Coventry University and the University of Bath in the United Kingdom.

### 2.1 Our Research Objectives

Our initial aim is the implementation of CAD infrastructure in Maple that can support both the use of CAD for traditional quantifier elimination, and also CAD as a theory solver for SMT as in [18]. We hypothesise that it will be more efficient to integrate SAT into computer algebra that the reverse, given (a) the complexity of algebraic procedures and the potential to benefit from decades of development of underlying sub algorithms and (b) the software engineering traditions within the SAT community that allow for standardised I/O and easier code reuse.

Our next objective is to reuse that infrastructure to implement the CAD-like algorithms discussed above [17], [9], [2] in Maple. These approach all relax the requirements of CAD in different ways and it is not clear which is superior: comparing them in a common system will allow for more meaningful conclusions on the underlying algorithms. There is also substantial scope for theory development on all such algorithms. Further, the experiments in [2] suggested there are substantial sets of problems on which each of the algorithms may excel giving rise to the potential of a portfolio solver (analogous to say [30]) and the use of machine learning for this and other choices [14].

CAD projection identifies those polynomials in less variables whose zeros represent changes in behaviour of the input polynomials. Simplifications to CAD projection have been critical to historic improvements. For a long time the best (i.e. smallest) operator was that of McCallum [19], which has been developed into a family of operators specialised to logical structure in the input e.g. [20], [7], [15]. However, all members of this family could fail for a small class of input types. Recently, the Lazard projection[1] has been shown to avoid such failure while being no more expensive than [19] except on cases where it failed. However, it remains to extend this theory into the wider family, with the initial work of [25], [26] to be continued in this project.

[1] first suggested in the 1990s by Lazard but verified recently by McCallum and collaborators [21], [22], [8]

### 2.2 New Application Domains

As noted above, CAD has a great many applications throughout the sciences and engineering. The DEWCAD project seeks to focus in detail on two emerging ones.

Bio-chemical network analysis has seen increasing use of computer algebra techniques, with recent examples including [6] and [27]. In both of these studies it was a combination of CAD with other algebraic techniques that allowed for an exact solution. Also, in both studies a comparison with solutions from numerical methods was made which showed the potential for numerical methods to make errors through floating point accumulation or insufficient sampling.

Recently CAD has also been shown to have use within economics [23] [24]. These range from the educational (Chicago now uses QE technology in its economics undergraduate curriculum) to the topical (QE used to demonstrate a gap in the reasoning of a Nobel Laureate).

### 2.3 Project partners

The DEWCAD project plans to work with a range of partners: the software company Maplesoft on implementation within Maple; Ábráham (RWTH Aachen) on integration with SMT; Brown (US Naval Academy) on search based algorithms; McCallum (Macquarie University) on the development of Lazard projection; Mulligan (U. Chicago) on economics applications; the SYMBIONT project [5] on bio-chemical network applications. We welcome collaboration with other members of the ISSAC community who have an interest in any of the topics described here or other potential CAD applications that may come into scope if the doubly exponential wall were pushed back further still.

## Acknowledgements

Bradford, Davenport and Uncu acknowledge the support of EPSRC Grant EP/T015713/1, while England and Sadeghimanesh acknowledge the support of EPSRC Grant EP/T015748/1.

## References

[1] E. Ábrahám, J. Abbott, B. Becker, A.M. Bigatti, M. Brain, B. Buchberger, A. Cimatti, J.H. Davenport, M. England, P. Fontaine, S. Forrest, A. Griggio, D. Kroening, W.M. Seiler, and T. Sturm. $\mathsf{SC}^2$: Satisfiability checking meets symbolic computation. In M. Kohlhase, M. Johansson, B. Miller, L. de Moura, and F. Tompa, editors, *Intelligent Computer Mathematics: Proc. CICM 2016*, LNCS 9791, pages 28–43. Springer, 2016. URL: https://doi.org/10.1007/978-3-319-42547-4_3.

[2] 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. *J. Logical and Algebraic Methods in Programming*, 119:100633, 2021. URL: https://doi.org/10.1016/j.jlamp.2020.100633.

[3] N.H. Arai, T. Matsuzaki, H. Iwane, and H. Anai. Mathematics by machine. In *Proc. 39th International Symposium on Symbolic and Algebraic Computation*, ISSAC '14, pages 1–8. ACM, 2014. URL: https://doi.org/10.1145/2608628.2627488.

[4] C. Barrett, R. Sebastiani, S.A. Seshia, and C. Tinelli. Satisfiability modulo theories. In A. Biere, M. Heule, H. van Maaren, and T. Walsh, editors, *Handbook of Satisfiability (Vol. 185 Frontiers in Artificial Intelligence and Applications), Chapter 26*, pages 825–885. IOS Press, 2009. URL: https://doi.org/10.3233/978-1-58603-929-5-825.

[5] F. Boulier, F. Fages, O. Radulescu, S.S. Samal, A. Schuppert, W. Seiler, T. Sturm, S. Walcher, and A. Weber. The SYMBIONT project: Symbolic methods for biological networks. *ACM Commun. Comput. Algebra*, 52(3):67–70, 2019. URL: https://doi.org/10.1145/3313880.3313885.

[6] R. Bradford, J.H. Davenport, M. England, H. Errami, V. Gerdt, D. Grigoriev, C. Hoyt, M. Košta, O. Radulescu, T. Sturm, and A. Weber. Identifying the parametric occurrence of multiple steady states for some biological networks. *J. Symbolic Computation*, 98:84–119, 2020. URL: https://doi.org/10.1016/j.jsc.2019.07.008.

[7] R. Bradford, J.H. Davenport, M. England, S. McCallum, and D. Wilson. Truth table invariant cylindrical algebraic decomposition. *J. Symbolic Computation*, 76:1–35, 2016. URL: http://dx.doi.org/10.1016/j.jsc.2015.11.002.

[8] C. Brown and S. McCallum. Enhancements to lazard's method for cylindrical algebraic decomposition. In F. Boulier, M. England, T.M. Sadykov, and E.V. Vorozhtsov, editors, *Computer Algebra in Scientific Computing*, LNCS 12291, pages 129–149. Springer, 2020. URL: https://doi.org/10.1007/978-3-030-60026-6_8.

[9] C.W. Brown. Open non-uniform cylindrical algebraic decompositions. In *Proc. 2015 International Symposium on Symbolic and Algebraic Computation*, ISSAC '15, pages 85–92. ACM, 2015. URL: https://doi.org/10.1145/2755996.2756654.

[10] C.W. Brown and J.H. Davenport. The complexity of quantifier elimination and cylindrical algebraic decomposition. In *Proc. 2007 International Symposium on Symbolic and Algebraic Computation*, ISSAC '07, pages 54–60. ACM, 2007. URL: https://doi.org/10.1145/1277548.1277557.

[11] B. Caviness and J. Johnson. *Quantifier Elimination and Cylindrical Algebraic Decomposition*. Texts & Monographs in Symbolic Computation. Springer-Verlag, 1998. URL: https://doi.org/10.1007/978-3-7091-9459-1.

[12] A.E. Charalampakis and I. Chatzigiannelis. Analytical solutions for the minimum weight design of trusses by cylindrical algebraic decomposition. *Archive of Applied Mechanics*, 88(1):39–49, 2018. URL: https://doi.org/10.1007/s00419-017-1271-8.

[13] C. Chen, M. Moreno Maza, B. Xia, and L. Yang. Computing cylindrical algebraic decomposition via triangular decomposition. In *Proc. 2009 International Symposium on Symbolic and Algebraic Computation*, ISSAC '09, pages 95–102. ACM, 2009. URL: https://doi.org/10.1145/1576702.1576718.

[14] M. England. Machine learning for mathematical software. In J.H. Davenport, M. Kauers, G. Labahn, and J. Urban, editors, *Mathematical Software – Proc. ICMS 2018*, LNCS 10931, pages 165–174. Springer, 2018. URL: https://doi.org/10.1007/978-3-319-96418-8_20.

[15] M. England, R. Bradford, and J.H. Davenport. Cylindrical algebraic decomposition with equational constraints. *J. Symbolic Computation*, 100:38–71, 2020. URL: https://doi.org/10.1016/j.jsc.2019.07.019.

[16] M. Erascu and H. Hong. Real quantifier elimination for the synthesis of optimal numerical algorithms (Case study: Square root computation). *J. Symbolic Computation*, 75:110–126, 2016. URL: https://doi.org/10.1016/j.jsc.2015.11.010.

[17] D. Jovanovic and L. de Moura. Solving non-linear arithmetic. In B. Gramlich, D. Miller, and U. Sattler, editors, *Automated Reasoning: 6th International Joint Conference (IJCAR)*, LNCS 7364, pages 339–354. Springer, 2012. URL: https://doi.org/10.1007/978-3-642-31365-3_27.

[18] G. Kremer and E. Ábrahám. Fully incremental CAD. *J. Symbolic Computation*, 100:11–37, 2020. URL: https://doi.org/10.1016/j.jsc.2019.07.018.

[19] S. McCallum. An improved projection operation for cylindrical algebraic decomposition. In B. Caviness and J. Johnson, editors, *Quantifier Elimination and Cylindrical Algebraic Decomposition*, Texts & Monographs in Symbolic Computation, pages 242–268. Springer-Verlag, 1998. URL: https://doi.org/10.1007/978-3-7091-9459-1_12.

[20] S. McCallum. On projection in CAD-based quantifier elimination with equational constraint. In *Proc. 1999 International Symposium on Symbolic and Algebraic Computation*, ISSAC '99, pages 145–149. ACM, 1999. URL: https://doi.org/10.1145/309831.309892.

[21] S. McCallum and H. Hong. On using Lazard's projection in CAD construction. *J. Symbolic Computation*, 72:65–81, 2016. URL: https://doi.org/10.1016/j.jsc.2015.02.001.

[22] S. McCallum, A. Parusińiski, and L. Paunescu. Validity proof of Lazard's method for CAD construction. *J. Symbolic Computation*, 92:52–69, 2019. URL: https://doi.org/10.1016/j.jsc.2017.12.002.

[23] C. Mulligan, R. Bradford, J.H. Davenport, M. England, and Z. Tonks. Non-linear real arithmetic benchmarks derived from automated reasoning in economics. In A.M. Bigatti and M. Brain, editors, *Proc. 3rd Workshop on Satisfiability Checking and Symbolic Computation* ($\mathsf{SC}^2$ *2018)*, CEUR Workshop Proceedings 2189, pages 48–60, 2018. URL: http://ceur-ws.org/Vol-2189/.

[24] C.B. Mulligan, J.H. Davenport, and M. England. TheoryGuru: A Mathematica package to apply quantifier elimination technology to economics. In J.H. Davenport, M. Kauers, G. Labahn, and J. Urban, editors, *Mathematical Software – Proc. ICMS 2018*, LNCS 10931, pages 369–378. Springer, 2018. URL: https://doi.org/10.1007/978-3-319-96418-8_44.

[25] A. Nair, J.H. Davenport, and G. Sankaran. On benefits of equality constraints in lex-least invariant cad. In J. Abbott and A. Griggio, editors, *Proc. 4th Workshop on Satisfiability Checking and Symbolic Computation (*$\mathsf{SC}^2$ *2019)*, CEUR Workshop Proceedings 2460, 2019. URL: http://ceur-ws.org/Vol-2460/.

[26] A. Nair, J.H. Davenport, and G. Sankaran. Curtains in CAD: Why are they a problem and how do we fix them? In A. Bigatti, J. Carette, J.H. Davenport, M. Joswig, and T. de Wolff, editors, *Mathematical Software – ICMS 2020*, LNCS 12097, pages 17–26. Springer, 2020. URL: https://doi.org/10.1007/978-3-030-52200-1_2.

[27] G. Röst and A. Sadeghimanesh. Exotic bifurcations in three connected populations with Allee effects. *Preprint on bioRxiv*, 2021. URL: https://doi.org/10.1101/2021.02.03.429609.

[28] A. Strzeboński. Cylindrical algebraic decomposition using local projections. *J. Symbolic Computation*, 76:36–64, 2016. URL: https://doi.org/10.1016/j.jsc.2015.11.018.

[29] T. Sturm and A. Tiwari. Verification and synthesis using real quantifier elimination. In *Proc. 36th International Symposium on Symbolic and Algebraic Computation*, ISSAC '11, pages 329–336. ACM, 2011. URL: https://doi.org/10.1145/1993886.1993935.

[30] L. Xu, F. Hutter, H.H. Hoos, and K. Leyton-Brown. SATzilla: Portfolio-based algorithm selection for SAT. *J. Artificial Intelligence Research*, 32:565–606, 2008. URL: https://doi.org/10.1613/jair.2490.