EPSRC
Polaris House, North Star Avenue, Swindon, Wiltshire,
United Kingdom SN2 1ET
Telephone +44 (0) 1793 444000
Web http://www.epsrc.ac.uk/

Standard PROPOSAL

Document Status: With Council
EPSRC Reference: EP/J003247/1

Computer Science

Organisation where the Grant would be held

Organisation	University of Bath	Research Organisation Reference:	JHD5910
Division or Department	Computer Science		

Project Title [up to 150 chars]

Real Geometry and Connectedness via Triangular Description

Start Date and Duration

a. Proposed start date	01 October 2011	b. Duration of the grant (months)	48

Applicants

Role	Name	Organisation	Division or Department	How many hours a week will the investigator work on the project?
Principal Investigator	Professor James Davenport	University of Bath	Computer Science	3.89
Co-Investigator	Dr Russell Bradford	University of Bath	Computer Science	2.86

Objectives

List the main objectives of the proposed research in order of priority [up to 4000 chars]

Cylindrical algebraic decompositions are a very useful computational tool for understanding the geometry, and in particular the connectedness, of the real solutions of systems of equations and inequalities. The investigators, and many others, have used such decompositions to solve many practical problems which can be reduced to connectedness questions. However, as the PI and colleagues [10, 19] have previously shown, computing them is inherently doubly exponential in the number of variables. Recent Canadian work [26] has produced an alternative means of computing cylindrical algebraic decompositions via triangular decompositions. In 2010 the Canadians and the PI [12] showed that, under suitable assumptions, the highest (complex) dimensional component of a system can be computed in singly-exponential time, and have produced encouraging experimental results.

Hence the fundamental aim of this project is to investigate, both in theory and in practice, how [12] and [26] interact. This is therefore incremental, in that it builds on these papers and the PI's previous work on the complexity of cylindrical algebraic decomposition, but also transformative in that it offers the possibility of breaking through the doubly-exponential barrier, and producing algorithms which solve practical problems in singly-exponential time.

In somewhat more detail, we have 5 objectives in the following logical order. 2 is the most important, but relies on 1. 3-5 will be corollaries of the first two.

1. Understanding the Complexity

The apparent contradiction between singly and doubly exponential is resolved by the fact that n iterations of a process where the output is singly-exponential in the input will give a doubly-exponential answer. But precisely where in the process does the doubly-exponential complexity manifest itself? Over the complexes, it is possible to perform triangular decomposition in singly-exponential time, so the fact that we are solving over the reals is clearly important. Understanding this would lead to a better understanding of [26] and its inherent complexity.

2. Adding laziness to cylindricity

To what extent is it possible to perform the MakeCylindrical operation of [26] on the sort of lazy structure output by [12]? Note that [13] proposes significant improvements to the border polynomial construct of [12], which is what defines those parts which are deferred, i.e. about which we are going to be lazy.

3. Linguistic Refinement

Though it makes no difference to the asymptotic complexity, allowing a wider range of connectives, e.g. "less than or equal to" as well as "less than", makes a substantial difference in practice with earlier versions of cylindrical algebraic decomposition. Is the same true here? Can we quantify the improvement? What happens if we allow the construct "x is not equal to y", rather than coding it as "x is less than y or x is greater than y"? Furthermore, as well as allowing these in the input, can we use them in the output with a view to generating more natural output?

4. Application to cuts

Both branch cuts in the simplication application, and obstacles in the robotics application, tend to be defined by parts of lines and curves, rather than the whole line/curve. [20] shows that in certain applications this can lead to a simpler decomposition: this direction needs to be explored in greater generality.

5. Further optimisations

We would like to implement the theoretical improvements in [13] to [12], and see how they behave, both directly in the triangular decomposition setting and with respect to cylindrical algebraic decomposition.

Summary

Describe the proposed research in simple terms in a way that could be publicised to a general audience [up to 4000 chars]. Note that this summary will be automatically published on EPSRC's website in the event that a grant is awarded.

Connectedness, as in "can we get there from here", is a fundamental concept, both in actual space and in various abstract spaces. Consider a long ladder in a right-angled corridor: can it get round the corner? Calling it a corridor implies that it is connected in actual three-dimensional space. But if we consider the space of configurations of the ladder, this is determined by the position and orientation of the ladder, and the `corridor' is now the requirement that no part of the ladder run into the walls - it is not sufficient that the ends of the ladder be clear of the walls. If the ladder is too long, it may have two feasible positions, one in each arm of the corridor, but there may be no possible way to get from one to the other. In this case we say that the configuration space of the ladder is not connected: we can't get the ladder there from here, even though we can get each end (taken separately, which is physically impossible) from here to there. Connectedness in configuration space is therefore the key to motion planning. These are problems human beings (especially furniture movers, or people trying to park cars in confined spaces) solve intuitively, but find very hard to explain. Note that the ladder is rigid and three-dimensional, hence its position is determined by the coordinates of three points on it, so configuration space is nine-dimensional.

Connectedness in mathematical spaces is also important. The square root of 4 can be either 2 or -2: we have to decide which. Similarly, the square root of 9 can be 3 or -3. But, if 4 is connected to 9 in our problem space (whatever that is), we can't make these choices independently: our choice has to be consistent along the path from 4 to 9. When it is impossible to make such decisions totally consistently, we have what mathematicians call a `branch cut' - the classic example being the International Date Line, because it is impossible to assign `day' consistently round a globe. In previous work, we have shown that several mathematical paradoxes reduce to connectedness questions in an appropriate space divided by the relevant branch cuts. This is an area of mathematics which is notoriously difficult to get right by hand, and

mathematicians, and software packages, often have internal inconsistencies when it comes to branch cuts.

The standard computational approach to connectedness, which has been suggested in motion planning since the early 1980s, is via a technique called cylindrical algebraic decomposition. This has historically been computed via a "bottom-up" approach: we first analyse one direction, say the x-axis, decomposing it into all the critical points and intermediate regions necessary, then we take each (x,y)-cylinder above each critical point or region, and decompose it, then each (x,y,z) above each of these regions, and so on. Not only does this sound tedious, but it is inevitably tedious - the investigators and others have shown that the problem is extremely difficult (doubly exponential in the number of dimensions).

Much of the time, notably in motion planning, we are not actually interested in the lower-dimensional components, since they would correspond to a motion with no degrees of freedom, rather like tightrope-walking. Recent Canadian developments have shown an alternative way of computing such decompositions via so-called triangular decompositions, and a 2010 paper (Moreno Maza in Canada + Davenport) has shown that the highest-dimensional components of a triangular decomposition can be computed in singly-exponential time. This therefore opens up the prospect, which we propose to investigate, of computing the highest-dimensional components of a cylindrical decomposition in singly-exponential time, which would be a major breakthrough in computational geometry.

Academic Beneficiaries

Describe who will benefit from the research [up to 4000 chars].

We see three distinct classes of academic beneficiaries.

a) Those working in (computational) real geometry and its complexity.

b) There are many people who do not realise that the geometry of space induced by the branch cuts of the various functions they are dealing with is fundamental to the correctness or otherwise of algebraic simplifications. Computer algebra systems either ignore these difficulties, and make incorrect `simplifications', or are too conservative and do not make desired simplifications. The Bath team have previously worked on this problem [1-4], but realising the underlying decompositions was a significant stumbling block in that work.

c) Those working in other fields, notably robotics and associated areas, where classification and connectedness of real spaces is important, or would be important if only effective (meaning both theoretically efficient and practically implemented) algorithms were available.

The theoreticians in class (a) will benefit from our published results, which will appear in key refereed conferences, and in a summative journal paper. We would hope that the recasting of the problem of real system solving which this project achieves will help many other researchers to move forward. The more practical ones will also benefit from having an implementation available, in a widely available computer algebra system (Maple), with which they will be able to experiment.

Users in class (b) will benefit firstly from having an implementation of better simplification in Maple, whose authors are keen to exploit this technology. Here 'better' means that the system never proposes an incorrect simplification, and on a wide class of naturally-occurring problems can always decide whether a simplification is valid or has a counter-example. In a wider sense, this will also "up the ante" on simplification, and we can expect other systems to follow suit.

Users in class (c) will benefit from having the necessary tools for exploring connectedness available in Maple.

Impact Summary

Impact Summary (please refer to the help for guidance on what to consider when completing this section) [up to 4000 chars]

Real geometry, i.e. the geometry of space as we perceive it without complex numbers, is vital to much of mathematics and many applications despite being less studied/taught (because it is harder?) than complex geometry. Solving problems in real geometry tends to lead to a lot of special cases (see (2) in the case for support), and is tricky to do by hand. Hence there are potentially many application areas for better algorithmic methods for real geometry and connectedness. In particular the question of interest is "connectedness within the reals": it is no good telling a robot to turn through an imaginary angle.

A) The theory of cylindrical algebraic decomposition was originally invented to solve the quantifier elimination problem: given a sentence containing "for all " or "there exists" clauses, produce an equivalent sentence without such clauses. Note that many sentences which appear not to contain such clauses in fact do: the [false over the reals] statement "sin is an invertible function" is really "for all y there exists an x such that y = sin(x)". Hence there are numerous applications in logic and computational mathematics which can benefit from better cylindrical algebraic decomposition: both in the U.K. and world-wide.

B) Motion planning for robots and similar devices such as automatically-steered cars, can be seen, as in [32], as "connectedness in configuration space", i.e. "can we get there from here", where "here" and "there" are configurations of the robot, rather than simply spatial points. This "configuration space" is typically of much higher dimension than one might expect: a rigid body in three-dimensional space is defined by three points, i.e. nine dimensions in all. Hence it is very important to have algorithms which behave efficiently as the number of dimensions grows, and this project should produce such algorithms. Motion planning is one of the areas we have worked on in the past [17], and we intend to pursue exemplar case studies ourselves, as well as disseminating our work via the IMRCs and appropriate conferences.The benefits of better motion planning are incalculable, ranging from better housework robots to the "Factory of the Future". Reconfiguring a "Factory of the Future" requires off-line motion planning, a "motion compiler", which would be a major gain: achievable via the use of the Maple software we will deliver. Further down the line one might envisage on-line motion planning, where our improvements in algorithmic performance would be even more significant.

C) Many mathematicians, and users of mathematics, have problems with apparently-simple function definitions. It is impossible to define 'square root' as a continuous function on the whole complex plane, since the argument of the square root of x is half the argument of x, and therefore the square root of 1 could be either 1 or -1, and neither choice is consistent with continuity. Hence one has to introduce 'branch cuts', and say that the definition is discontinuous as one crosses this cut. These cuts mean that many formulae are not in fact correct, either for special values ("log(1/x)=-log(x)" is not true when x=-1), or in much of the complex plane ("sqrt(x**2-1)=sqrt(x-1)sqrt(x+1)" is only 50% true). The Bath team [1-8] have realised that this is a question of connectivity of the complement of the branch cuts. This has been implemented, and provides a much better (guaranteed never to state that an incorrect identity is true) simplification algorithm than previous ones. However, it is limited by the connectivity question, and the breakthrough envisaged in this project will make the connectivity approach to simplification much more feasible. Maplesoft (see letter of support) would be very interested in this, and where one leading vendor goes, others will follow.

All categories of users will benefit from our twin-track approach of scientific publication and software dissemination via the world-wide Maple computer algebra system.

Summary of Resources Required for Project

Financial resources

Summary fund heading	Fund heading	Full economic Cost	EPSRC contribution	% EPSRC contribution
Directly Incurred	Staff	123669.00	98935.20	80
	Travel & Subsistence	31600.00	25280.00	80
	Other Costs	9296.00	7436.80	80
	Sub-total	**164565.00**	**131652.00**	
Directly Allocated	Investigators	65651.95	52521.56	80
	Estates Costs	19934.00	15947.20	80
	Other Directly Allocated	4680.00	3744.00	80
	Sub-total	**90265.95**	**72212.76**	
Indirect Costs	Indirect Costs	116337.00	93069.60	80
Exceptions	Staff	47565.00	47565.00	100
	Other Costs	12131.00	12131.00	100
	Sub-total	**59696.00**	**59696.00**	
	Total	**430863.95**	**356630.36**	

Summary of staff effort requested

	Months
Investigator	8.75
Researcher	36
Technician	0
Other	0
Visiting Researcher	0
Student	42
Total	86.75

Other Support

Details of support sought or received from any other source for this or other research in the same field.

Awarding Organisation	Awarding Organisation's Reference	Title of project	Decision Made (Y/N)	Award Made (Y/N)	Start Date	End Date	Amount Sought / Awarded (£)
University of Waterloo (Canada)	Mark Giesbrecht	Davenport Sabbatical	Y	Y	15/02/2009	15/07/2009	10000

Staff

Directly Incurred Posts

			EFFORT ON PROJECT							
Role	**Name /Post Identifier**	**Start Date**	**Period on Project (months)**	**% of Full Time**	**Scale**	**Increment Date**	**Basic Starting Salary**	**London Allowance (£)**	**Super-annuation and NI (£)**	**Total cost on grant (£)**
Researcher	Researcher (postdoc)	01/01/2012	36	100	Research07	01/04/2012	31671	0	7429	123669
									Total	123669

Applicants

Role	Name	Post will outlast project (Y/N)	Contracted working week as a % of full time work	Total number of hours to be **charged** to the grant over the duration of the grant	Average number of hours per week **charged** to the grant	Rate of Salary pool/banding	Cost estimate
Principal Investigator	Professor James Davenport	Y	100	684	3.9	110166	45669
Co-Investigator	Dr Russell Bradford	Y	100	504	2.9	65421	19983
						Total	65652

Exceptions

Role	**Name /Post Identifier /Institution**	**Start Date**	**London Allowance (£)**	**Fees**	**Stipend**
Project Student	Research Student / University of Bath	01/10/2011	No	12131.00	47565.00

Travel and Subsistence

Destination and purpose		Total £
Outside UK	Closing workshop: with conference TBC	7000
Outside UK	ISSAC 2012 - Grenoble	3000
Outside UK	ISSAC 2013 - North America, +Brown/U.W.O.	4500
Outside UK	ISSAC 2014 - Europe?	4000
Outside UK	ISSAC 2015 (Far East??)	4000
Outside UK	MEGA 2013 (Europe?)	2000
Outside UK	MEGA 2015 (Europe?)	1000
Outside UK	Visit to Canada (U.W.O. & Maplesoft)	3600
Within UK	Moreno Maza visit to Bath (with ISSAC 2014)	500
Within UK	UK robotics dissemination to IMRC etc. (two)	1000
Outside UK	International robotics dissemination	1000
	Total £	31600

Other Directly Incurred Costs

Description	Total £
Closing workshop organisational costs	3000
2xDell Optiplex 980 with 8GB memory (RO+Student): needed to run development version of Maple	1922
Toshiba Portege R700-185 laptop: needed to run development version of Maple: for demonstrations and visits to collaborators.	1374
Recruitment costs (postdoc and student)	3000
Total £	9296

Other Directly Allocated Costs

Description	Total £
Pool staff costs	4680
Total £	4680

Project Partners: details of partners in the project and their contributions to the research. These contributions are in addition to resources identified above.

1	**Name of partner organisation**		**Division or Department**	**Name of contact**	
Maplesoft			Director of Research	Dr Juergen Gerhard	
Direct contribution to project			**Indirect contribution to project**		
	Description	**Value £**		**Description**	**Value £**
cash			use of facilities/ equipment		
equipment/materials	Use of Development Version (notional cost)	1	staff time	Advisory Group	4000
secondment of staff			other		
other			Sub-Total		4000
Sub-Total		1		Total Contribution	4001

Total Contribution from all Project partners	£4001

OTHER INFORMATION

Reviewers

1	Name	Organisation	Division or Department	Email Address
Professor Alex Wilkie		The University of Manchester	Mathematics	awilkie@maths.manchester.ac.uk

Reviewers

2	Name	Organisation	Division or Department	Email Address
Professor Lawrence Paulson		University of Cambridge	Computer Laboratory	LP15@cam.ac.uk

Reviewers

3	Name	Address	Town	Email Address
Professor Marie-Francoise Coste-Roy		IRMAR	RENNES	marie-francoise.roy@univ-rennes1.fr

Pathways to Impact

Who will benefit from this research?

The immediate impact will be on the computer algebra community, through the development of better algorithms and better understanding the the complexity of cylindrical algebraic decomposition, and its approach to Real Geometry. Beyond that, there are as many applications as mathematics has itself.

- Formal mathematics and formal proof. As one of the leading arithmetic verifiers at Intel has said:

  > It wasn't until I came to formalise these identities [of elementary functions] that I realised how messy the side-conditions could become. [30, and attached letter of support]

  These identities *can* be automatically handled, and the side-conditions automatically derived, by the method of [4], but in practice this currently founders on the rock of cylindrical algebraic decomposition for many examples.

- Robotics and motion planning. This can theoretically be done by the method of [32], but again in practice this currently founders on the rock of cylindrical algebraic decomposition for many examples. It is worth noting that we are only interested[1] in connectivity of components of full dimension, so this application is intrinsically suitable for the approach of [12].

- Algebraic simplification is a key issue for computer algebra. As Maplesoft write in their letter of support

  > it goes without saying that we *would* have improved our simplification methods if we could, so this additional insight, translated into usable code, will be most interesting to us.

  Because of the competitive nature of the computer algebra market, improved functionality in Maple coming from this project will automatically lead to interest in these algorithms from other vendors.

[1]Connectivity via components of lower dimension would correspond to a move where there was absolutely no tolerance, and thus infeasible in practice.

Pathways to Impact

How will they benefit from this research?

There are essentially two routes to benefit: from the mathematics and algorithmics, and from access to an actual implementation. The first will be achieved through the standard scientific dissemination routes, while the second will be achieved by building our software implementation on top of Maple and the existing tools in Maple, and we would expect to follow the experience of the University of Western Ontario, where their code (which included contributions from the PI) made its way into production Maple quite quickly.

What will be done to ensure that they have the opportunity to benefit from this research?

The research programme proper is planned to finish in December 2014. We propose a subsequent 9-month dissemination phase (during the initial period of which the research student will also be completing their thesis). We propose two main activities in this period, which cannot reasonably be done while the research is still ongoing.

1. A workshop, probably organised as a satellite activity to an existing conference such as ISSAC or CICM, in which we, and others such as Brown and the Canadians, would explore, and talk about, the results of this research and associated results. We would naturally extend invitations to the other U.K. teams using real algebraic geometry, notably the MetiTarski group at Cambridge, and in Edinburgh.

2. A major journal paper, pulling together the individual results of the researchers, which will have appeared in various conference papers written during the research phase of the project. This is a technique we have previously employed, where [4] pulled together the results of [1, 2, 5, 3].

3. In terms of dissemination into robotics and advanced manufacturing, we will work with the EPSRC-funded Innovative Manufacturing Research Centres, especially the Bath one, which is thematically as well as geographically closest to our interests, to ensure that there is proper dissemination into that community.

Pathways to Impact

Real Geometry and Connectedness via Triangular Description

Track Record and Case for Support

PI: Professor James Davenport holds a joint appointment between Mathematical Sciences and Computer Science at Bath. He has worked in computer algebra since 1976, and in Cylindrical Algebraic Decomposition since 1985 [16, which is still cited, as the Davenport–Mahler–Mignotte bound, today]. His attempts to apply this to robot-motion planning, on the lines of [32], were frustrated by the complexity of the cylindrical decompositions produced [17, 18]. He has subsequently produced important *lower* bound results on the complexity of Cylindrical Algebraic Decomposition [10, 19].

He has twice had major visits to the Ontario Research Centre for Computer Algebra[1], two months in 2000 (U.W.O., where he was the first holder of the Ontario Research Chair in Computer Algebra) and five in 2009 (Waterloo). The first of these sparked his interest in the verified simplification problem [15, 14] and the second introduced him to the U.W.O. work of [26]. In particular he contributed to the interface design for this work [11], and saw, at first hand, how this made its way into the production version (14) of Maplesoft's flagship product, the computer algebra system Maple. Both [11] and [12] are cited in the β version (15) of Maple, so there was a timelag of less than a year between [12] being *submitted* for publication and its appearing in the β version of the product: a process we aim to emulate.

CI: Dr Russell Bradford has worked in computer algebra since 1984. He has been interested in the verified simplification problem since 1992, authoring one of the earliest papers on the subject [6]. Since then he worked with Davenport on the subject (see "Team", below). He has a long history of international research collaborations: for example, the Distributed Systems Laboratory at the University of Calgary where he was involved with the design of virtualised appliance delivery for WestGrid (who supply computing services for the Universities of Western Canada).

Closer to home, his research spans the range from the abstract mathematical end of computer science to the hard technology of applying parallelism to the analysis of sound [9].

Maple Maple is a major computer algebra system, with which the team has had a great deal of experience. It is the key delivery technology for the project, and is the means by which the Canadian team deliver their software. As a computer algebra system, Maple is also an ideal setting to explore the applications to simplification. Their development version is already used by U.W.O., so giving us a copy (quoted as £1 on JES, since it is not commercially available) will therefore greatly aid the collaboration with U.W.O. as well as with Maplesoft itself.

Team This pair have held a successful EPSRC grant (GR/R84139/01) on the simplification problem, which produced [1]–[4], [8, 7], and have continued to work on the problem with a jointly-supervised research student [20].

This project differs from (GR/R84139/01) in the amount of cooperation required with Maplesoft and the University of Western Ontario. The PI is an experienced collaborative project manager, having chaired several EU projects, and numerous PRINCE-2 projects in the University of Bath, and this experience will be invaluable in managing this project.

Why Bath?

Bath is a research-oriented university, and both Mathematical Sciences and Computer Science

[1] A joint laboratory between the University of Waterloo, University of Western Ontario and Waterloo Maplesoft.

scored highly in the last RAE, which makes it a natural home for this project.

In the U.K., there is a certain amount of interest in real decision procedures such as quantifier elimination at Edinburgh [31] and in the theory of real functions (also closely related to simplification) [Paulson's system MetiTarski [21]]. Both groups have in fact asked us for advice on the theory, and practical implementation, of cylindrical algebraic decomposition.

There is also work on the more theoretical aspects of decision procedures, which indeed extend beyond the algebraic to the Pfaffian (differential equation) case, both at Bath (Vorobjov) and elsewhere (Macintyre, Wilkie), but this work is not aimed at practical implementation.

Hence the Bath team is uniquely situated between the pure theoreticians on the one hand, and the primarily application solvers on the other.

Team References

[1] J.C. Beaumont, RJB, and JHD. Better Simplification of Elementary Functions Through Power Series. In J.R. Sendra, editor, *Proceedings ISSAC 2003*, pages 30–36, 2003.

[2] J.C. Beaumont, RJB, JHD, and N. Phisanbut. A Poly-Algorithmic Approach to Simplifying Elementary Functions. In J. Gutierrez, editor, *Proceedings ISSAC 2004*, pages 27–34, 2004.

[3] J.C. Beaumont, RJB, JHD, and N. Phisanbut. Adherence is Better Than Adjacency. In M. Kauers, editor, *Proceedings ISSAC 2005*, pages 37–44, 2005.

[4] J.C. Beaumont, RJB, JHD, and N. Phisanbut. Testing Elementary Function Identities Using CAD. *AAECC*, 18:513–543, 2007.

[5] J.C. Beaumont, RJB, and N. Phisanbut. Practical Simplification of Elementary Functions Using CAD. In *Proceedings A3L*, pages 35–40, 2005.

[6] RJB. Algebraic Simplification of Multiple-Valued Functions. In *Proceedings DISCO '92*, pages 13–21, 1993.

[7] RJB, R.M. Corless, JHD, D.J. Jeffrey, and S.M. Watt. Reasoning about the Elementary Functions of Complex Analysis. *Annals of Mathematics and Artificial Intelligence*, 36:303–318, 2002.

[8] RJB and JHD. Towards Better Simplification of Elementary Functions. In T. Mora, editor, *Proceedings ISSAC 2002*, pages 15–22, 2002.

[9] RJB, R. Dobson, and J.P. ffitch. Sliding with a Constant Q. In *Proc. Int. Conf. Digital Audio Effects (DAFx-08)*, pages 363–369, Espoo, Finland, Sep 1-4 2008.

[10] C.W. Brown and JHD. The Complexity of Quantifier Elimination and Cylindrical Algebraic Decomposition. In C.W. Brown, editor, *Proceedings ISSAC 2007*, pages 54–60, 2007.

[11] C. Chen, JHD, J. May, M. Moreno Maza, B. Xia, R. Xiao, and Y. Xie. User Interface Design for Geometrical Decomposition Algorithms in Maple. `http://www.activemath.org/workshops/MathUI/09/proc/Davenport-et-al-UI-design-geometric-decomposition-MathUI09.pdf`, 2009.

[12] C. Chen, JHD, 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.

[13] C. Chen, JHD, M. Moreno Maza, B. Xia, and R. Xiao. Computing with Semi-Algebraic Sets Represented by Triangular Decomopoposition. *Submitted to ISSAC 2011*.

[14] R.M. Corless, JHD, D.J. Jeffrey, G. Litt, and S.M. Watt. Reasoning about the Elementary Functions of Complex Analysis. In John A. Campbell and Eugenio Roanes-Lozano, editors, *Proceedings Artificial Intelligence and Symbolic Computation*, pages 115–126, 2001.

[15] R.M. Corless, JHD, D.J. Jeffrey, and S.M. Watt. According to Abramowitz and Stegun. *SIGSAM Bulletin 2*, 34:58–65, 2000.

[16] JHD. Computer Algebra for Cylindrical Algebraic Decomposition. *TRITA-NA-8511*, 1985.

[17] JHD. On a "Piano Movers" Problem. *SIGSAM Bulletin 1/2*, 20:15–17, 1986.

[18] JHD. Robot Motion Planning. *Geometric Reasoning*, pages 43–55, 1989.

[19] JHD and J. Heintz. Real Quantifier Elimination is Doubly Exponential. *J. Symbolic Comp.*, 5:29–35, 1988.

[20] N. Phisanbut, RJB, and JHD. Geometry of Branch Cuts. *Communications in Computer Algebra*, 44:132–135, 2010.

Case for Support

Cylindrical Algebraic Decomposition (CAD) was invented in the 1970s [27], and is a conceptually very powerful technique, solving previously unsolved, or intractable, problems in, or using, real geometry and connectedness. [26] introduced a radically different approach to CAD, via triangular decompositions. [12] produced new algorithms for triangular decomposition, in which the highest dimensional information could be extracted more efficiently than computing the complete answer (the 'lazy' approach). We wish to explore how this lazy approach interacts with CAD in the style of [26], and the benefits this has for various applications.

1 Introduction

CAD has been applied in numerous areas, of which we name three:

A) in formal reasoning, quantifier elimination and the solution of decision problems (historically the first)
B) in robotics, particularly motion planning [32]
C) in (computerised) mathematics, analysis of branch cuts and the simplification of elementary functions [1]–[4].

Practical uptake has nevertheless been limited for three reasons.

Availability CAD has historically been implemented only in specialised algebra systems, such as QEPCAD [24], and these have proved difficult to interface to. Indeed, the first widely-available implementation was in Maple of the new method from [26].

Interface CAD is not widely taught, and the data structure produced is not natural for most users. In addition, the data structure is normally very large, since it describes, not merely the full-dimensional cells, but also their boundaries, and the boundaries of the boundaries, and so on. For example, a 3-dimensional cube has 6 surfaces, bounded by 12 lines and 8 corners. In 4 dimensions, each hypercube has 8 hypersurfaces, 24 surfaces, 32 lines and 16 corners. In [11], we introduced a `piecewise` representation, which is much more natural for most users, and allows us to 'hide the details'[2].

[2] For 'bottom-up' CAD, they are still computed, but one way of seeing this project is to avoid computing them at all.

Running Time CAD is, both theoretically and practically, very expensive. Ideas such as partial CAD [28] reduce the running time substantially for quantifier elimination applications, but are not universally applicable and do not improve the theoretical complexity. Simplification [25] can be a great help in practice, but has no theoretical analysis. In addition the results can be tedious to analyse, in view of the number of lower-dimensional components. In [3] (Application C), we can avoid analysing them but they are currently still produced. In motion-planning (Application B) they are unnecessary, but are again produced. Hence a method only computing the full-dimensional components would be very desirable.

What Has Changed?

There are two answers to this question, depending on whether one wants to look at technology push, or application pull.

Push CAD has been around for 35 years [27], but for 34 of them there was essentially only one way of computing them, basically the bottom-up method of Collins. This method is intrinsically doubly-exponential in the number of variables. In 2009 [26] introduced an alternative perspective on the problem, via triangular decomposition, which can be seen as a more holistic approach to the problem. However, the then-known triangular decomposition techniques must have the same doubly-exponential complexity for computing a CAD, even though over $\mathbf{C}$ the complexity is singly-exponential in one (lazy) formulation, and unknown in others. In 2010, [12] introduced a lazy technique for the triangular sets, in which the doubly-exponential work could be deferred. If transferring this to CAD is feasible, this would be a major breakthrough.

Pull [31] shows that the decision problems are important in practice, but the algebra is a limiting factor. [16] shows that CAD (as originally conceived) is infeasible for motion-planning applications, but there are increasing demands for motion-planning in contexts such as assisted living. The applicants' work on the verified simplification problem (Application C) has shown the limitations of

CAD as a building block, since the bottom-up method is inherently doubly-exponential in the number of variables. But no alternative to the applicants' work has emerged, and the demands for verified simplification continue [30]. In short, applications exist that can benefit from this technology.

2 Background

2.1 Quantifier Elimination over the reals

Quantifier Elimination is a fundamental problem in formal reasoning: given a statement

$$P(x_1,\ldots,x_m) = Q_1y_1\ldots Q_ny_nf(x_1,\ldots,x_m,y_1,\ldots,y_n), \tag{1}$$

where the Q_i are either $\forall$ or $\exists$, produce a quantifier-free equivalent formula $g(x_1,\ldots,x_m)$. The problem obviously depends on the language of discourse: we consider **real algebraic geometry**, where f, g etc. are Boolean combinations of polynomial equations $p(x_1,\ldots,x_m,y_1,\ldots,y_n) = 0$ and inequalities $q(x_1,\ldots,x_m,y_1,\ldots,y_n) > 0$. In this language $\exists x : ax^2+bx+c=0$ reduces to $b^2-4ac \geq 0$, but only when $a \neq 0$, and the full set of special cases is

$$\left\{ \begin{array}{ll} b^2-4ac \geq 0 & a \neq 0 \\ b \neq 0 & a = 0 \\ c = 0 & a = b = 0 \end{array} \right. . \tag{2}$$

This class of problems was solved by Tarski [33], but the complexity of his solution was indescribable[3]. A better (but nevertheless doubly exponential in the number of variables[4] $m+n$, i.e. $\delta^{O(2^{m+n})}$), solution had to await the concept of CAD [27] described in the next section.

In fact the problem of quantifier elimination is intrinsically doubly exponential: both [10, 19] produce families of P depending on n, with fixed m, where the length of g is doubly exponential in n, since the solutions consists of a doubly-exponential number of isolated points. In both cases, the construction increases n by adding a block $\forall y_1\ldots,\forall y_k\exists y_{k+1}\ldots\exists y_l$ to a previous formula. Since $\exists x\exists y$ is equivalent to $\exists y\exists x$, and similarly for $\forall$, we extend $\exists$ and $\forall$ to operate on blocks of variables, so that, if $\mathbf{x} = (x_1,\ldots,x_k)$, $\exists\mathbf{x}$ is equivalent to $\exists x_1\ldots\exists x_k$ etc. We can then reformulate (1) as (3), where we do as much combining of blocks of quantifiers as we can.

$$P(x_1,\ldots,x_m) = \mathbf{Q}_0\mathbf{y}_0\ldots\mathbf{Q}_a\mathbf{y}_af(x_1,\ldots,x_m,y_1,\ldots,y_n), \tag{3}$$

where now $\mathbf{Q}$ is either $\exists$ or $\forall$.

[3] In the formal sense, that there was no elementary function which could describe it, i.e. no tower of exponentials of fixed height would suffice!

[4] For simplicity we consider the polynomial degree δ of the input. In principle the number of polynomials, and the height H of the coefficients also figure in the complexity formulae, but they behave roughly like δ.

2.2 Cylindrical Algebraic Decomposition

This technique, introduced by [27], takes a set of polynomials $p_i^{(1)}(y_1,\ldots,y_N)$ (where for convenience we write y_{n+i} for x_i, and N for $m+n$), and produces a **sampled CAD** (sCAD) of $\mathbf{R}^1,\ldots,\mathbf{R}^N$ which is sign-invariant for the p_i. Each cell $C_i^{(k)}$ of $\mathbf{R}^{N-k+1}$ is defined by

$$f_{i,1}^{(k)}(y_k,\ldots,y_N)\,\sigma_{i,1}^{(k)}\,0,\ldots,f_{i,n_i}^{(k)}(y_k,\ldots,y_N)\,\sigma_{i,n_i}^{(k)}\,0, \tag{4}$$

where the $\sigma_{i,j}^{(k)}$ are either $=$ or $>$, such that:

1. Each $\mathbf{R}^{N-k+1}$ is the disjoint union of the $C_i^{(k)}$;
2. The $C_i^{(k)}$ are connected and non-empty — this being demonstrated by a **sample point** $s_i^{(k)} \in C_i^{(k)}$
3. The projection of each $C_i^{(k)}$ onto $\mathbf{R}^{N-k}$ is one of the $C_i^{(k+1)}$ (and the sample point of $C_i^{(k)}$ projects to the sample point of $C_i^{(k+1)}$);
4. Each $C_i^{(1)}$ is **sign-invariant** for the $p_j^{(1)}(y_1,\ldots,y_N)$, i.e. each p_j is identically positive, identically negative or identically zero throughout each $C_i^{(1)}$.

The solution to the quantifier elimination problem is then the disjunction of the formulae (4) for those cells C_i^{n+1} (i.e. involving only $x_1,\ldots,x_m$) for which P is true, which is determined by investigating the truth of f at the sample points lying above the sample point of C_i^{n+1}, and combining the truth values according to the nature of the quantifiers Q_i ($\forall$ means that it must be true for all sample points, $\exists$ means that it must be true at one of the sample points). Collins' algorithm for producing such a decomposition proceeds in three phases:

Projection Repeatedly reduce the set $p_i^{(k)}(y_k,\ldots,y_N)$ to a set $p_i^{(k+1)}(y_{k+1},\ldots,y_N)$ such that a sCAD of $\mathbf{R}^{N-k}$ sign-invariant for $p_i^{(k+1)}(y_{k+1},\ldots,y_N)$ can be lifted to a sCAD of $\mathbf{R}^{N-k+1}$ sign-invariant for $p_i^{(k)}(y_k,\ldots,y_N)$;

Solution of the problem in $\mathbf{R}^1$, i.e. isolating all the roots of the $p_i^{(N)}(y_N)$ (the roots, and the rationals separating them, becoming the sample points in $\mathbf{R}^1$);

Lifting the sCAD of $\mathbf{R}^{N-k}$ sign-invariant for $p_i^{(k+1)}(y_{k+1},\ldots,y_N)$ to a sCAD of $\mathbf{R}^{N-k+1}$ sign-invariant for $p_i^{(1)}(y_k,\ldots,y_N)$.

The examples of [10, 19] show that the total degree of the input to the solution phase can indeed be doubly exponential in n. It has also been remarked (e.g. [16]) that a CAD is overkill: not only does it solve the problem given, but also *any* other quantifier elimination problem involved the same variables in the same order, and the same polynomials (but possibly different σ, different quantifiers, and different Boolean formulae). In particular the decomposition is cylindrical (point 3 above) for all k ($0 < k < N$), whereas in fact quantifier elimination only needs cylindricity at the boundaries between quantifier blocks in (3). In particular quantifier elimination does not require cylindricity among the x_i: it is the induction process in building a sCAD that requires it.

It should be noted that a CAD of $\mathbf{R}^N$ is automatically a decomposition of $\mathbf{R}^N$ into **connected** sets each sign-invariant for the $p_i^{(1)}$, together with sufficient adjacency information to deduce connected components sign-invariant for the $p_i^{(1)}$. Such a decomposition (and this is the only currently implemented way of producing it) has many computational uses, ranging from robot motion planning (Application B) [32] to verification of functional identities (Application C) [4]. These applications do not actually require *any* cylindricity. In practice [4, 17] these applications find the size and cost of a CAD a significant obstacle: in [17] even the example of a one-dimensional ladder in a rectangular corridor produced polynomials of degree 800.

2.3 Recent Developments

[26] recently proposed a completely different algorithm for computing a CAD in three very different phases:

InitialPartition partition $\mathbf{C}^N$ into sets C_i on which the $p_i^{(1)}$ are invariant (always zero or never zero);

MakeCylindrical the decomposition $\{C_i\}$, in the sense that for any $j < N$, the projections of any two C_i and $C_{i'}$ onto the last j coordinates are either disjoint or equal;

MakeSemiAlgebraic deduce from this a CAD on $\mathbf{R}^N$ sign-invariant for the $p_i^{(1)}$.

If necessary explicit sample points are generated.

The complexity of this algorithm has not been studied, but it has to be at least doubly exponential in N [10, 19], and is unlikely to be worse than this. All phases rely heavily on **triangular decomposition** [23].

We have recently produced [12] an algorithm which, for a set S of equations and inequalities $p_i\,\sigma_i\,0 \subset \mathbf{Z}[y_1\ldots y_n]$ (where $\sigma_i \in \{=,\neq,\geq,>\}$) produces a description (in terms of regular semi-algebraic systems — RSAS — the equivalent for $\mathbf{R}$ of regular chains) of the zero set of S, $\mathcal{Z}(S)$. The algorithm is doubly exponential in N, and, since it is likely that this problem encodes [19], this seems inevitable. What is more interesting is that we ([12]) have a lazy variant of this algorithm. Let d be the dimension of the variety in $\mathbf{C}^N$ defined by the equalities of S. Then, in **singly** exponential time, this algorithm produces:

1. a set (possibly empty, in the case that the dimension over $\mathbf{R}$ of $\mathcal{Z}(S) < d$) R_i of RSAS such that $\bigcup_i \mathcal{Z}(R_i) \subset \mathcal{Z}(S)$ — the complement $\mathcal{Z}(S)\setminus\bigcup_i \mathcal{Z}(R_i)$ is of dimension $< d$;
2. $G \subset \mathbf{Z}[y_1\ldots y_n]$, referred to as the 'fingerprint polynomial set', and essentially "where things might go wrong", such that $\mathcal{Z}(S)\setminus\bigcup_i \mathcal{Z}(R_i) \subset \mathcal{Z}(G=0)$ and $\dim_{\mathbf{C}} \mathcal{Z}(G=0) < d$. In the example of (2), this would be $a=0$. Hence the part of $\mathcal{Z}(S)$ not represented by the R_i can be computed by a recursive call to this algorithm with argument $S\cup\{g_i=0 \mid g_i\in G\}$ (and therefore a smaller d).

This therefore sidesteps the doubly-exponential barrier of [10, 19], since their examples are zero-dimensional, and N iterations of a singly-exponential algorithm can give us the doubly-exponential growth in degree required by these examples.

3 The major challenge

Informally, this can be simply stated: is it possible to merge lazy triangular decomposition [12] and CAD via triangular decomposition [26] to produce an algorithm which does some kind of "lazy" CAD in the same sense as [12] is “lazy”, i.e. producing the high-dimensional cells of the CAD and deferring the computation of the lower-dimensional ones? It is not yet clear what the precise *formal* description of a “lazy CAD” would be, and this would need to be developed.

CAD was designed for logic (application A) and has since been applied elsewhere.

In the applications to robotics (B), we tend to have a certain number of equalities, which represent rigid constraints on the robot, and inequalities, which represent legal positions. In [16], considering a line in a 2-dimensional corridor, there is the length of the ladder as a rigid constraint. Hence the legitimate positions lie in a 3-dimensional subspace. A 2-dimensional or lower space represents a constraint with no leeway, and hence is of no interest in practical motion planning.

In the application (C) to analysis of branch cuts [4], only the full-dimensional cells are needed to determine whether the identity is “true except on a set of measure zero", which is all that is needed for many applications. Adherence [3] will often determine whether such an identity is totally true. Even if this doesn’t apply, G is a pointer to where the problems might arise, and in general knowing this allows further simplification (item 4 below).

We indicated earlier that there were three main obstacles to the application of CAD. Let us indicate how our triangular set approach will address these.

Availability Our methods will be available for, and indeed distributed with, the widely-used Maple computer algebra package.

Interface Building on [11], we will have a more natural interface to CAD, which will also be more comprehensible since the lower-dimensional special cases will be at least hidden, and in many cases not even evaluated.

Running Time As stated above, many applications do not require these lower-dimensional components, and by not even computing them we should save time, hopefully by an exponential factor.

4 Methodology

The project will proceed on two, interacting, tracks, essentially ‘Practical’ (Work package 2), led by Bradford, and ‘Theoretical’ (Work package 3), led by Davenport. Work package 1 is project initiation and package 4 is the research student’s research training, culminating in the thesis.

WP2 — Practical. This work package will be undertaken by the research officer, under the guidance of Bradford and Davenport.

WP3 — Theoretical. This work package will be undertaken by the research student, under the guidance of Davenport and Bradford.

Within these work packages, there are five issues to be explored. **P** indicates an issue to be explored in the ‘practical’ work package (2), and **T** one for the ‘theoretical’ work package (3). We use **p** and **t** to indicate lesser importance of these packages.

1. **Understanding the Complexity. P** Does the problem solved by [12] actually encode the examples of [19, 10] (and hence the doubly exponential behaviour of the non-lazy algorithm is inherent) or not? In the latter case, it would seem to be (though again this would need research) the MakeCylindrical step of the algorithm of [26] which accounts for the doubly exponential nature of these algorithms when applied to [19]. **t** This would lead to better understanding of the complexity of [26].
2. **Adding laziness to cylindricity. P** Is it possible to adapt the MakeCylindrical step of the algorithm of [26] so as only to provide cylindricity corresponding to the blocks in (3)? If so, this would be a significant step forward for *practical* quantifier elimination, and *possibly* bring the complexity of the algorithm closer to the theoretical lower bound of $\delta^{O(N2^a)}$. It would certainly be of great advantage to the other applications of a CAD that don’t need cylindricity as such, including applications B and C.
   This is non-trivial, since the standard argument that shows that the cells are connected relies on cylindricity, inducting from the (trivial) connectedness of the components of a decomposition of $\mathbf{R}^1$. However, previous attempts at reducing the cylindricity requirement (e.g. [22]) have foundered on the fact that the projection phase of [27] has to guarantee cylindricity during the lifting phase, but it is too early at that stage to know what the cylindricity requirements will be. Since [26] fundamentally does not work the same way, the issue is worth revisiting.

3. **Linguistic Refinement. T** As stated above, the algorithm of [27] does not distinguish between equalities and inequalities during the construction of a sCAD: this only matters during the solution of the quantifier elimination problem. Modern implementations (such as QEPCAD [24]) take advantage of this to some extent, using the ideas of [28]), but it is fair to say that this is not systematic. [26], aiming to produce a complete CAD in the sense of [27], does not distinguish either. **p** [12] makes a four-fold distinction of the connective being $=, \neq, >, \geq$ ($\neq$ and $\geq$ are logically redundant, but pragmatically useful). How can this be used to improve the CAD produced?
4. **Application to cuts. P/T** The other applications of CAD tend to deal in sections of curves rather than complete curves. For example the branch cut of $\log(x + iy)$ (and therefore $\sqrt{x + iy}$) is [4]

$$y = 0 \wedge x < 0, \tag{5}$$

and the corridor considered in [17] consists of $y = 0 \wedge x < 0$, $y = 1 \wedge x < -1$, $x = 0 \wedge y > 0$ and $x = -1 \wedge y > 1$. We have already observed [20] that this form of constraint lends itself to pre-processing. For example, in the purported identity

$$\sqrt{z^2 - 1} \stackrel{?}{=} \sqrt{z - 1}\sqrt{z + 1} \tag{6}$$

the translation of (5) gives us

$$xy = 0 \wedge x^2 - y^2 - 1 < 0. \tag{7}$$

This gives us 29 cells to investigate with [26], and 36 with [24]. Replacing it with the equivalent (pseudo-dividing the inequality by the equality)

$$xy = 0 \wedge -y^4 - y^2 < 0, \tag{8}$$

reduces the CADs to 21 cells with [26], and 22 or 24 with [24]. In more dimensions we would expect the effect to be more striking.

This idea could already be applied in the algorithm of [12], since more reductions might be possible in $S \wedge \{g_i = 0 \mid g_i \in G\}$ than were possible simply in S. Again, this might be very useful in practice despite the theoretical barrier of [10, 19]. It could also be applied in any merged version of [12, 26], since preconditioning could be applied during the recursive reductions as well as at the start.
5. **Further optimisations. P** The PI and the Canadian team have recently [13] proposed various improvements to [12], not all of which have yet been implemented. We would like to implement these, and then see how they behave in practice.

5 Project Management

We will have weekly meetings of the project team. These will be followed up with quarterly reports to the Advisory Group.

5.1 Advisory Group

We propose an international Advisory Group to steer the project. Modern tele-conferencing tools are at the stage where we do not propose physical meetings, but will have a formal rendezvous, with an agreed summary and plan of future work after three months and every subsequent year. We propose the following members, and letters of support are attached.

- Jürgen Gerhard, Waterloo Maplesoft.
- Marc Morena Maza, University of Western Ontario. He is the team leader of the Canadian team, and collaboration with his team is very important to the project.
- C.K. Yap, City University of New York, representing application area B.
- Chris Brown, USNA and author of QEPCAD, probably the leading implementation of [28], representing the theory of CAD, and Application A.
- John Harrison, Intel, for the logical difficulties of branch cuts in application area C.

5.2 Risk Management

We see various potential risks in the project: both external (1–3) and internal (4–5). The risks identified are manageable.

1. Failure to recruit. It is possible that we would be unable to recruit either a research officer or a research student. This seems unlikely, given both the current economic state and the investigators' numerous contacts with Europe and beyond. Perhaps more likely is that the research officer would be unable to start on the planned date, but this would not be a serious barrier if this start date should slip by three months, and even six would be possible. The University of Bath is already

receiving applications from suitable research students in the general area.

2. Weakened collaboration with Maplesoft. This is clearly important to the project's software dissemination. Clearly Maplesoft will make its decisions in its commercial interest, but we note that they have incorporated the Western Ontario code in the past, and that their presentation "New Features in Maple 14" at ISSAC 2010 dwelt largely on [11]. Hence it is likely that Maplesoft will incorporate successful software arising from this project. However, this is *not* vital to the dissemination of the theoretical side of the project, and indeed the software could still be disseminated as a separate piece of Maple source code.
3. Weakened collaboration with University of Western Ontario. This has proved very successful in the past, not only in papers [11, 12], but also in telepresence at seminars etc. There is no reason to believe that this will change. However, if for any reason it should, we observe that Maple is not a closed system, we have and can develop, the Maple code, and that development of our software *could* proceed independent of any collaboration with Western Ontario.
4. Inability to define a 'lazy' CAD. this would be worrying, and we would need to try to *prove* that such a concept did not exist. This would then at least provide a negative result, and show, even more fundamentally than [19, 10], the limitations of CAD.
5. Inability to define 'cylindrical by blocks' as in issue 2 above. This would be a blow, and again would indicate that CAD is not as powerful as we would hope. We think this outcome is unlikely, as the known bad cases for CAD [19, 10] depend on having many small blocks.

As can be seen, though there are external risks, they are not fatal to the project, and we have mitigation strategies. The existence of internal risks is inherent in research, but we observe that there are alternative, admittedly negative, results that could emerge.

Other References

[21] B. Akbarpour and L.C. Paulson. MetiTarski: An Automatic Theorem Prover for Real-Valued Special Functions. *J. Automated Reasoning*, 44:175–205, 2010.

[22] D.S. Arnon, G.E. Collins, and S. McCallum. An Adjacency Algorithm for Cylindrical Algebraic Decompositions of Three-dimensional Space. *J. Symbolic Comp.*, 5:163–187, 1988.

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

[24] C.W. Brown. QEPCAD B: a program for computing with semi-algebraic sets using CADs. *ACM SIGSAM Bulletin 4*, 37:97–108, 2003.

[25] C.W. Brown and A. Strzeboński. Black-Box/White-Box Simplification and Applications to Quantifier Elimination. In S.M. Watt, editor, *Proceedings ISSAC 2010*, pages 69–76, 2010.

[26] C. Chen, M. Moreno Maza, B. Xia, and L. Yang. Computing Cylindrical Algebraic Decomposition via Triangular Decomposition. In J. May, editor, *Proceedings ISSAC 2009*, pages 95–102, 2009.

[27] 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.

[28] G.E. Collins and H. Hong. Partial Cylindrical Algebraic Decomposition for Quantifier Elimination. *J. Symbolic Comp.*, 12:299–328, 1991.

[29] D.Yu. Grigoriev. Complexity of Deciding Tarski Algebra. *J. Symbolic Comp.*, 5:65–108, 1988.

[30] J. Harrison. Difficulty of Side-Conditions. *Private Communication 4 June 2010*.

[31] G.O. Passmore and P.B. Jackson. Combined Decision Techniques for the Existential Theory of the Reals. In J. Carette *et al.*, editor, *Proceedings Intelligent Computer Mathematics*, pages 122–137, 2009.

[32] J.T. Schwartz and M. Sharir. On the "Piano-Movers" Problem: II. General Techniques for Computing Topological Properties of Real Algebraic Manifolds. *Adv. Appl. Math.*, 4:298–351, 1983.

[33] A. Tarski. A Decision Method for Elementary Algebra and Geometry, 2nd ed. *Univ. Cal. Press*, 1951.

Justification of Resources

Investigators Davenport will be the principal investigator, with Bradford as co-investigator.

Preparation/Research Phases (first 39 months) Based on past experience, We request 3 hours/week for Bradford and Davenport. In terms of the major research packages, Bradford will concentrate on WP2 (practical) and Davenport on WP3 (theoretical), building on [10, 19]. For applications, Davenport will focus on B, building on [16], while Bradford will focus on C, building on [6] and [1]–[4]. Because of the co-operative nature of this project, we request an additional 1 hour/week for Davenport as Project Manager and responsible for liaison with the Canadian team and other advisors.

Dissemination Phase (last 9 months) We request 3 weeks for Davenport (workshop organiser) and 2 weeks for Bradford.

Supervision is covered by the student's fees.

Research student The project has numerous open-ended research questions, notably refinement (question 3) and handling of sections rather than whole curves (question 4). These would form suitable topics for a PhD thesis, but the student will need research training. There is substantial background, as far as we know only taught at Bath in the U.K., for a research student to master before he/she can be productive. We therefore aim for a 3.5 year maths-style studentship. There is plenty of specific initial training both available and necessary in the first year of the studentship, not least the PI's course "CM30070 — Computer Algebra" taught in the first semester. In the past the PI has run a seminar series on Triangular Decomposition, with the Canadian team participating via Skype, and this would also be appropriate.

Postdoc. We will need a researcher of post-doctoral experience to develop the code required and perform the appropriate experiments to validate the work of this project. It is unlikely (though not impossible) that we could recruit a suitable postdoctoral researcher from the U.K., given the very small computer algebra community here, but Davenport is in regular contact with the major European centres, and finding a suitable candidate should not be too difficult.

Travel The International Symposium on Symbolic and Algebraic Computation (ISSAC) is the major outlet for research in computer algebra. It also has a strong application track: for example in 2010 we heard about the application of a parameterized quantified SAT solving at Daimler Cars.

- ISSAC 2012 (Europe) PI+2
- Also in 2012 a visit to Moreno Maza in Western Ontario/ Maplesoft in Waterloo PI+1 (almost certainly the postdoc., who may stay longer at Western Ontario).

- ISSAC 2013 North America + visits to Moreno Maza and Brown (Annapolis) PI+2.
- ISSAC 2014 We assume Europe and factor in a visit by Moreno Maza (who normally attends ISSAC) to Bath PI+3.
- ISSAC 2015 We assume the Far East PI+CI.
- The (roughly) biennial Méthodes Algébriques en Géometrie Algébrique (MEGA) is also a major (anglophone, despite the name) conference. MEGA 2011 is too early, but we should certainly aim to present at MEGA 2013 (PI+1) and 2015 (PI) — probably in Europe.
- External dissemination, e.g. robotics: since this will occur 3+ years from now, we cannot name conferences, but expect two UK workshops and one European.
- A closing workshop forms a major part of the dissemination plan. In view of the pressure on people's time and diaries, we will try to schedule this alongside an existing conference, aiming for a European one (possibly MEGA 2015 if the timing is right). We request travel and subsistence for the the research team and two invited speakers (£7000), as well as organisational costs of the workshop, such as venue hire (£3000). By placing it alongside an existing conference, we anticipate an attendance of 30–50.

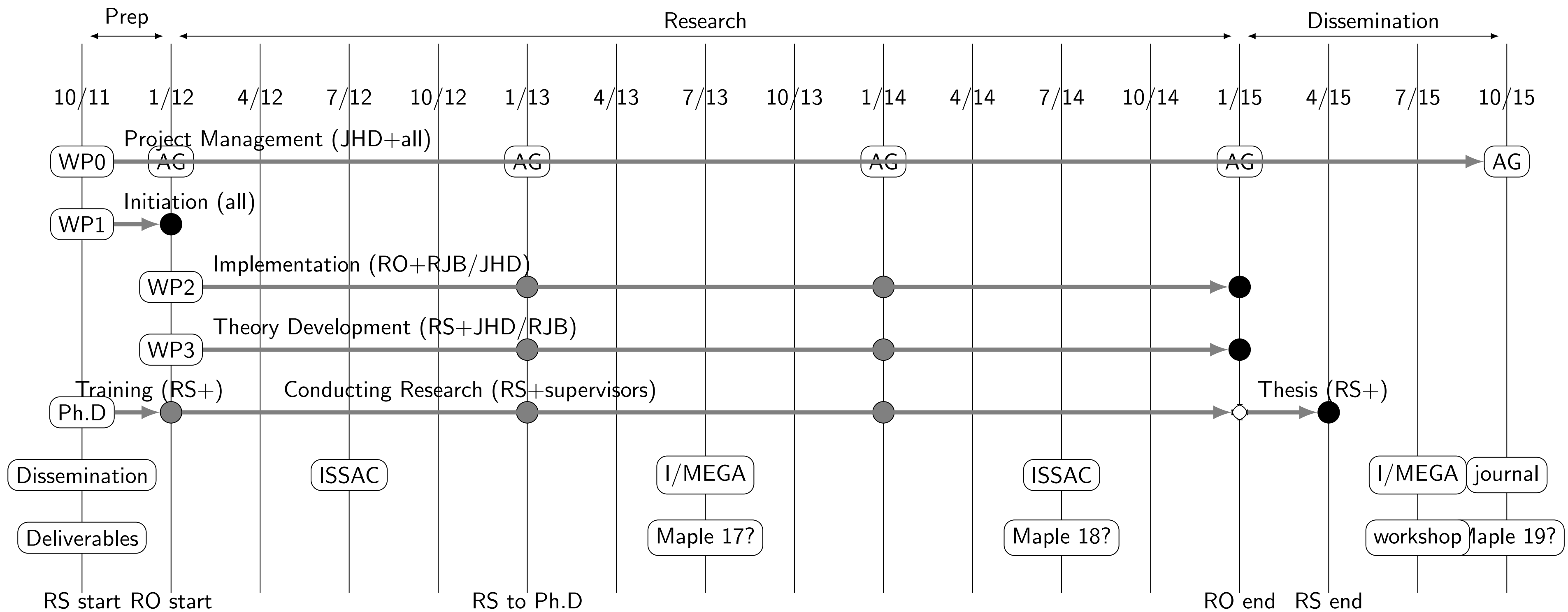
Other Directly Incurred Costs £3000 is requested to cover the advertising and interviewing costs associated with the recruitment of the Research student and the Postdoc.

We will need two reasonably powerful (more in terms of memory than CPU) PCs for the postdoc and research student: these will need to be dedicated to the project, as they will be running the specially-licenced development version of Maple, so a pool computer is not possible.

We will also need a laptop for demonstration and taking to U.W.O. and Maplesoft, again running the development version of Maple. We will also naturally meet the U.W.O. team at conferences, so being able to work together on a shared develeopment version will be extremely productive.

Other Directly Allocated Costs It will be far more efficient to have these machines installed and maintained by the Department's support team, and we are requesting a person-month of computer officer time to purchase, configure and keep updating.

Justification of Resources



key	people		events
RJB	Russell Bradford (CI)	**AG**	Formal Rendezvous of Advisory Group
JHD	James Davenport (PI)	**ISSAC**	International Symposium on Symbolic and Algebraic Computation
RO	Research Officer (post-doctoral)	**I/MEGA**	ISSAC and Méthodes Effectives en Géometrie Algébrique
RS	Research Student	**RS to Ph.D**	Research Student transfers from M.Phil. to Ph.D. status

Maple release numbers are conjectural dates for rendezvous with the Maple release cycle.