



Citation for published version:

Davenport, JH 2022, 'Digital Collections of Examples in Mathematical Sciences', Paper presented at 8th European Congress of Mathematicians, Portorož, Slovenia, 20/06/21 - 26/06/21.

Publication date:
2022

Document Version
Peer reviewed version

[Link to publication](#)

University of Bath

Alternative formats

If you require this document in an alternative format, please contact:
openaccess@bath.ac.uk

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Digital Collections of Examples in Mathematical Sciences

James H. Davenport

Some aspects of Computer Algebra (notably Computation Group Theory and Computational Number Theory) have some good databases of examples, typically of the form “all the X up to size n ”. But most of the others, especially on the polynomial side, are lacking such, despite the utility they have demonstrated in the related fields of SAT and SMT solving. We claim that the field would be enhanced by such community-maintained databases, rather than each author hand-selecting a few, which are often too large or error-prone to print, and therefore difficult for subsequent authors to reproduce.

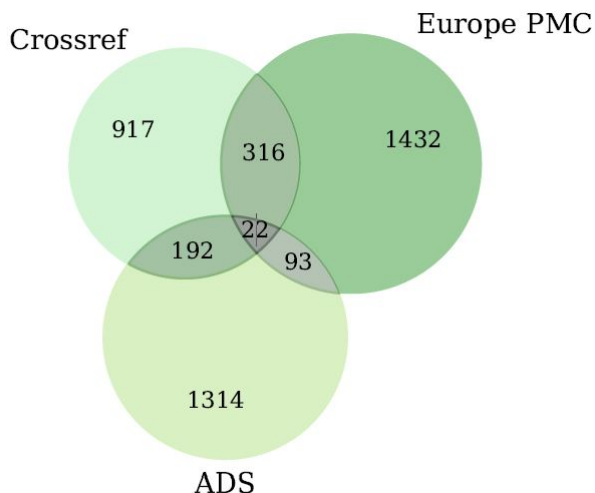
1 Introduction

Mathematicians have long had useful collections, either of systematic data or examples. One of the oldest known such is the cuneiform tablet known as Plimpton 322, which dates back to roughly 1800BC: see [23, pp. 172-176], or a more detailed treatment in [42, 50]. This use of systematic tables of data spawned the development on logarithmic, trigonometric and nautical tables: Babbage’s Difference Engine was intended to mechanise the production of such tables. But there were also tables of purely mathematical interest: the author recalls using an 1839 table of logarithms and what are now known as Zech logarithms [59] (but in fact they go back at least to [41]), i.e. tables of the function $\log x \mapsto \log(1+x)$, at least over \mathbf{R} : Jacobi’s table [34] was modulo p^n for all the prime powers $p^n < 1000$.

1.1 Data Citation

Citation and referencing is an important point of modern scholarship — Harvard-style referencing is generally attributed to [43], and the history of *Science Citation Index* is described in [29]. It is well-understood, and practically all research students, and many undergraduates, get lessons in article citation practices.

Keywords: Benchmarking, Citation, OpenMath
MSC 2020: 00A35, 12-04, 20-04

Figure 1.1. Overlaps between data citation harvesters [56, Figure 5]

Despite the success of article citation, data citation is a mess in practice [56]: only 1.16% of dataset DOIs in Zenodo are cited¹ (and 98.5% of these are self-citations). It is still a subject of some uncertainty: [36, 46] and significant changes are still being proposed [25]. Worse, perhaps, it is poorly harvested: see Figure 1.1. Assuming independence and looking at the overlap statistics, we can estimate that there are between 4,000–20,000 data sets waiting to be cited. In such circumstances, *de facto* people cite a paper if they can find one.

2 Pure Mathematics

2.1 Online Encyclopedia of Integer Sequences

This database [52] can be said to have “colonised the high ground” in mathematics: mathematicians from all sub-disciplines use it. It has evolved from a private enterprise, for a long time at <http://www.research.att.com/~njas/sequences>, to a system maintained by a foundation, and now at <https://oeis.org/>. The recommended citation is “OEIS Foundation Inc. (2022), The On-Line Encyclopedia of Integer Sequences, published electronically at https://oeis.org, [date]”, but the author had originally to search the website to find it!

¹In contrast, 60% of papers in Natural Science and Engineering *had* a citation in the next two years [39, 49].

2.2 Group Theory

The Classification of Finite Simple Groups, as well as being a *tour de force* in mathematics, also means that we have a complete database here. In most other areas, we have to be content with “small” databases.

An example of this is the transitive groups acting on n points, where various authors have contributed: [17] ($n \leq 11$); [51] ($n = 12$); [16] ($n = 14, 15$); [32] ($n = 16$); [33] ($17 \leq n \leq 31$); [18] ($n = 32$). These are available in the computer algebra system GAP (and MAGMA), except that (for reasons of space) $n = 32$ isn’t in the default build for GAP.

These are really great resources (if that’s what you want), but how does one cite this resource: “[55, transgrp library]”?

There are several other libraries such as primitive groups. But it could be argued that (finite) Group Theory is “easy”: for a given n there are a finite number and we “just” have to list them.

2.3 L -functions and Modular Forms

The L -functions and Modular Forms Database, known as LMFDB and hosted at lmfdb.org is a third example of mathematical databases. The recommended citation, “The LMFDB Collaboration, The L -functions and modular forms database, <http://www.lmfdb.org>, 2021” is directly linked from the home page, which is a good model to follow.

Computation in this area had a long history, from [9] and [54] to the current database, which is the work of a significant number of people. The early computations gave rise to the Birch–Swinnerton-Dyer Conjectures [10], now a Clay Millennium Prize topic. The current computations are in active use by mathematicians: see Poonen’s remarks in [28].

3 SAT and SMT Solving

3.1 SAT Solving

SAT solving is normally seen as solving a Boolean expression written in Conjunctive Normal Form (CNF).

The 3-SAT problem is: given a 3-literals/clause CNF satisfiability problem,

$$\underbrace{(l_{1,1} \vee l_{1,2} \vee l_{1,3})}_{\text{Clause 1}} \wedge (l_{2,1} \vee l_{2,2} \vee l_{2,3}) \wedge \cdots \wedge (l_{N,1} \vee l_{N,2} \vee l_{N,3}), \quad (3.1)$$

where $l_{i,j} \in \{x_1, \bar{x}_1, x_2, \bar{x}_2, \dots\}$, is it satisfiable? In other words, is there an assignment of $\{T, F\}$ to the x_i such that all the clauses are *simultaneously* true.

3-SAT is the quintessential NP-complete problem [24]. 2-SAT is polynomial, and k -SAT for $k > 3$ is polynomial-transformable into 3-SAT. In practice we deal with SAT — i.e. no limitations on the length of the clauses and no requirement that all clauses have the same length.

Let n be the number of i such that x_i (and/or \bar{x}_i) actually occur. Typically n is of a similar size to N .

Despite the problem class being NP-complete, nearly all examples are easy (e.g. SAT-solving has been routinely used in the German car industry for over twenty years [38]): either easily solved (SAT) or easily proved insoluble (UNSAT). For random problems there seems to be a distinct phase transition between the two: [2, 3, 30], with the hard problems typically lying on the boundary.

This means that constructing difficult examples is itself difficult, and a topical research area: [5, 53].

SAT solving has many applications, so we want effective solvers for “real” problems, not just “random” ones. This gives us the fundamental question: what does this mean?

3.2 SAT Contests

These are described at <http://www.satcompetition.org>. They have been run since 2002. In the early years, there were distinct tracks for Industrial/Handmade/Random problems: this has been abandoned.

The methodology is that the organisers accept submissions (from contestants² and others), then produce a list of problems (in DIMACS, a standard format) and set a time (and memory) limit, and see how many of the problems the submitted systems can solve on the contest hardware.

SAT is easy to certify (the solver just produces a list of values of the x_i). Verifying UNSAT is much harder, but since 2013 the contest has required proofs of UNSAT for the UNSAT track, and since 2020 in all tracks, in DRAT: a specified format (some of these proofs have been $> 100\text{GB}$).

The general feeling is that these contests have really pushed the development of SAT solvers, roughly speaking $\times 2/\text{year}$. For comparison, Linear Programming has done $\times 1.8$ over a greater timeline and with more rigorous documentation [11].

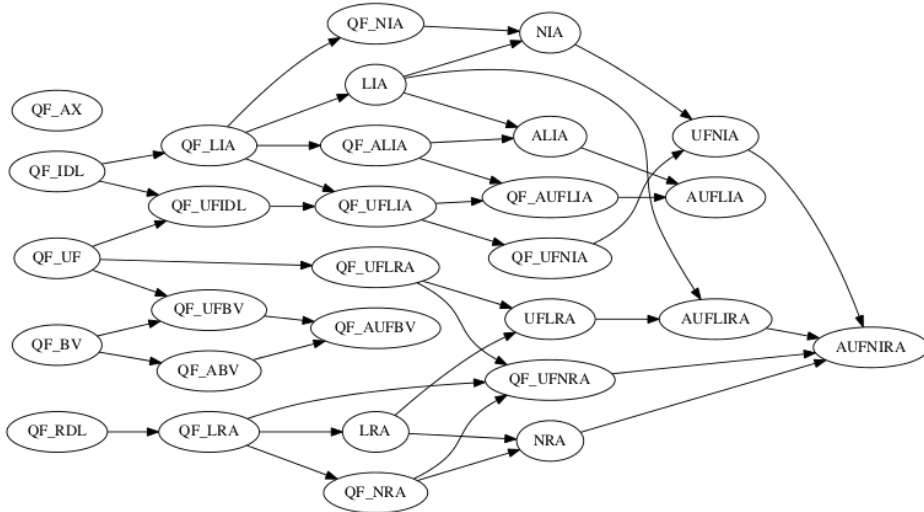
3.3 SMT: Life Beyond SAT

Consider a theory T , with variables y_j , and various Boolean-valued statements in T of the form $F_i(y_1, \dots, y_n)$, and a CNF \mathcal{L} in the form of (3.1) with $F_i(y_1, \dots, y_n)$ rather than just x_i . In principle T can be anything: those currently supported³ are given in Figure 1.2.

²In 2020, contestants were required to submit at least 20 problems, as well as a solver.

³By the SMT-LIB standard: [6], which also says “New logics are added to the standard opportunistically, once enough benchmarks are available”.

Figure 1.2. Available logics (March 2022) <https://smtlib.cs.uiowa.edu/logics.shtml>



For example QF_NRA is the Quantifier-Free theory of Nonlinear Real Arithmetic, and QF_LRA (Linear Real Arithmetic) is included in this. Both QF_NRA and QF_UFLRA (Uninterpreted Functions and Linear Real Arithmetic) are included in QF_UFNRA.

Then the SAT/UNSAT question is similar: do there exist values of y_i such that \mathcal{L} is true (SAT), or can we state that no such exist (UNSAT), and the community runs SMT Competitions (<https://smt-comp.github.io/2022/>). There is a separate track for each theory T , as the problems will be different. Within each, the problems are subdivided as industrial/crafted/random.

The SMT-LIB format [6] provides a standard input format. The question of proving UNSAT is in general unsolved (but see [37] for one particular theory T).

There has been substantial progress in SMT-solving over the years, possibly similar to SAT, and probably also spurred by the contests.

4 Computer Algebra: Where are we?

Obviously, Group Theory and others are parts of computer algebra: what about the rest of computer algebra?

In general the problems of computer algebra have a bad worst-case complexity, and we want effective solvers for “real” problems, not just “random” ones. The question, as in SAT and SMT, is “what does this mean?”.

But there are also various logistical challenges.

(1) **Format:** there is no widely accepted common standard. We do have OpenMath [15], but it’s not as widely supported as we would like.

- (2) **Contests:** There are currently none. Could SIGSAM organise them?
- (3) **Problem Sets:** There are essentially no independent ones. Each author chooses his own.
- (4) **Archive:** Not really.

We now consider various specific problems.

4.1 Polynomial GCD

This problem is NP-hard (for sparse polynomials, even univariate) [27, 48]. Even for dense polynomials, it can be challenging for multivariate. There is no standard database: one has to trawl previous papers (and often need to ask the authors, as the polynomials were too big to print in the paper). Verification is a challenge: one can check that the result is a *common divisor*, but verifying *greatest* is still NP-hard [48].

4.2 Polynomial Factorisation

This is known to be polynomial-time for dense encodings [40], even though their exponent is large, and much work has gone into better algorithms, e.g. [1]. Presumably it is NP-hard for sparse encodings, though the author does not know of an explicit proof. There is no standard database: one has to trawl previous papers (and often need to ask the authors, as the polynomials were too big to print in the paper).

Verification is a challenge: one can check that the result is a *factorisation*, but checking completeness (i.e. that these factors are irreducible) seems to be as hard as the original problem in the worst cases.

It is worth noting that, with probability 1, a random dense polynomial is irreducible (and easily proved so by the Musser test [47]), so the question “what are the *interesting* problems?” is vital.

4.3 Gröbner Bases

The computation of Gröbner bases has many applications, from engineering to cryptography. But this has doubly exponential (w.r.t. n , the number of variables) worst-case complexity [45], even for a prime ideal [20]. If we take n “random” equations in n variables, they will satisfy the conditions for the Shape Lemma [7] and have $D \leq n^n$ solutions, so a Gröbner base in a purely lexicographical order will look like

$$\{p_1(x_1), x_2 - p_2(x_1), x_3 - p_3(x_1), \dots, x_n - p_n(x_1)\}, \quad (4.1)$$

where p_1 is a polynomial of degree D in x_1 and the other p_i are polynomials of degree at most $D - 1$ in x_1 . Experience shows that the coefficients of the p_i will generally be large (theoretically, they can be D times as long as the input coefficients). Conversely, if we have $n + 1$ equations, there are generally no solutions and the Gröbner base is $\{1\}$: much shorter than (4.1).

The good news from the point of view of this paper is that there is a collection [8], but it's very old (1996), so most of the examples are trivial with today's hardware and software, and completely static. Worse, some of the examples are only available in PDF.

There always is a Gröbner base (no concept of UNSAT as such) but it's not clear what a useful certificate of “ G is a Gröbner base for input L ” might mean in general (but see [4]). If $G = \{g_1, \dots, g_M\}$ is a Gröbner base of $F = \{f_1, \dots, f_N\}$ then a general certificate would consist of three components:

- (1) A proof that G is a Gröbner base, which would mean that every S -polynomial $S(g_i, g_j)$ reduces to 0 under G , which is easily checked;
 - (2) A proof that $(F) \subseteq (G)$, which could be a set of $\lambda_{i,j}$ such that every $f_i = \sum \lambda_{i,j} g_j$;
 - (3) A proof that $(G) \subseteq (F)$, which could be a set of $\mu_{i,j}$ such that every $g_i = \sum \mu_{i,j} f_j$.
- However, the $\lambda_{i,j}$ and $\mu_{i,j}$ might be (and generally are) extremely large.

4.4 Real Algebraic Geometry

Again, the problem of describing the decomposition of \mathbf{R}^n sign-invariant for a set S of polynomials f_i in n variables has doubly exponential (w.r.t. n) worst-case complexity [14]. However, unlike Gröbner bases, it seems that this is the “typical” complexity, though the author knows no formal statement of this. For a given problem, the complexity can vary greatly: [14, Theorem 7] is an example of a polynomial p in $3n + 4$ variables such that *any* Cylindrical Algebraic Decomposition (CAD), w.r.t. one order, of \mathbf{R}^{3n+4} sign-invariant for p has $O(2^{2^n})$ cells, but w.r.t. another order has 3 cells.

$$\begin{aligned}
 p := & x^{n+1} \left(\left(y_{n-1} - \frac{1}{2} \right)^2 + (x_{n-1} - z_n)^2 \right) \left((y_{n-1} - z_n)^2 + (x_{n-1} - x_n)^2 \right) \\
 & + \sum_{i=1}^{n-1} x^{i+1} \left((y_{i-1} - y_i)^2 + (x_{i-1} - z_i)^2 \right) \left((y_{i-1} - z_i)^2 + (x_{i-1} - x_i)^2 \right) \\
 & + x \left((y_0 - 2x_0)^2 + \left(\alpha^2 + \left(x_0 - \frac{1}{2} \right) \right)^2 \right) \times \\
 & \left((y_0 - 2 + 2x_0)^2 + \left(\alpha^2 + \left(x_0 - \frac{1}{2} \right) \right)^2 \right) + a.
 \end{aligned}$$

The bad order (eliminating x , then $y_0, \alpha, x_0, z_1, y_1, z_1, \dots, x_n, a$) needs $O(2^{2^n})$ (Maple: 141 when $n = 0$) cells. Any order eliminating a first says that R^{3n+3} is undecomposed, and the only question is $p = 0$, which is linear in a , and we get three cells: $p < 0$, $p = 0$ and $p > 0$.

However, if we replace a by a^3 , the topology is essentially the same, but the discriminant is no longer trivial, and the “good” order now generates 213 cells in Maple, rather than three.

There is a collection [57], not quite as old as [8] (2014 was the last update), but still completely static. The DEWCAD project [12] might update this, but there are still

issues of long-term conservation. The format has learned from [8] and each example is available in text, Maple input and QEPCAD.

If we are just looking at computing a CAD, which we might wish to do for motion planning purposes [58], there is no concept of UNSAT, and the question of certificates of correctness is essentially unsolved. Attempts to produce a formally verified CAD algorithm have also so far been unsuccessful [21].

However, CAD was invented [22] for the purpose of quantifier elimination, i.e. converting $Q_k x_k Q_{k+1} x_{k+1} \cdots Q_n x_n \Phi(f_i)$, where $Q_i \in \{\exists, \forall\}$ and Φ is a Boolean combination of equalities and inequalities in the f_i , into $\Psi(g_1, \dots, g_{n'})$, where Ψ is a Boolean combination of equalities and inequalities in the g_i , polynomials in x_1, \dots, x_{k-1} , and if the statement is fully quantified, the result is a Boolean. A common case, particularly in program verification, is the fully existential case (all Q_i are \exists), where Φ is “something has gone wrong”, and we want to show this can’t happen. Then SAT is easy (exhibit values of x_i such that Φ is true, but UNSAT is much harder to certify. See [37] for some steps in this direction.

4.5 Integration

The computational complexity of integration, i.e. given a formula f in a class \mathcal{L} , is there a formula $g \in \mathcal{L}$, or in an agreed extension of \mathcal{L} , such that $g' = f$, is essentially unknown (but integration certainly involves GCD, factorisation etc.). When \mathcal{L} includes algebraic functions, difficult questions of algebraic geometry arise (see [26, as corrected in [44]]), and there is no known bound on the complexity of these.

“Paper” mathematics produced large databases of integrals, e.g. [31], but these are (at best) in PDF, and the way they are commonly printed makes it extremely hard to recover semantics from the layout. Probably the best current database is described in [35]. But these databases are almost entirely of successful (SAT in our notation) examples, and there is almost no collection of UNSAT ($\nexists g \in \mathcal{L} : g' = f$) examples. Algorithm-based software (e.g. [26]) has an internal proof of UNSAT, but I know of no software that can exhibit it. That proof is typically very reliant on the underlying mathematics.

A new question here is the “niceness” of the output in the SAT case. Jeffrey and Rich [35] give the example of

$$\int \frac{5x^4}{(1+x)^6} dx = \frac{x^5}{(1+x)^5}, \quad (4.2)$$

where Maple’s answer is

$$\frac{-10}{(1+x)^3} + \frac{5}{(1+x)^4} - \frac{5}{(1+x)} - \frac{1}{(1+x)^5} + \frac{10}{(1+x)^2}. \quad (4.3)$$

Note that (4.3) is not just an ugly form of the right-hand side of (4.2): the two differ by 1, which is a legitimate constant of integration.

While some element of “niceness” is probably beyond automation, “simplicity” in the sense of [19], essentially minimal Kolmogorov complexity, is probably a good proxy, and could be automatically judged (at least in principle: there are probably some messy system-dependent issues in practice).

5 Conclusions

- (1) The field of computer algebra really ought to invest in the sort of contests that have stimulated the SAT and SMT worlds.
- (2) This requires much larger databases of “relevant” problems than we currently have, and they need to be properly curated.
 - + The technology of collaborative working, e.g. wikis, or GitHub, has greatly advanced since the days of [8], which should make collaborative construction of example sets easier, and would also help with the preservation challenge.
 - Although OpenMath is in principle a suitable system-neutral notation that could be the standard input (and output) format, such a use would challenge OpenMath implementations. This would be a good development, though.
- (3) This would allow much better benchmarking practices: see the description in [13].
- (4) There are significant challenges in providing “certificates”, not just of UNSAT in the case of integration, but elsewhere in algebra. For example, asserting $g = \gcd(f_1, f_2)$ involves, not just the claim that g divides f_1 and f_2 , but also that $f_1/g, f_2/g$ are relatively prime, which may be much harder to demonstrate.

Acknowledgements. The author is grateful to Dr. Uncu for his comments on drafts, and to the organisers of the MIDAS session at the 8th European Congress of Mathematicians for prompting these reflections.

Funding. This work was partially supported by EPSRC Grant EP/T015713/1.

Bibliography

- [1] J. Abbott, V. Shoup and P. Zimmermann, Factorization in $\mathbf{Z}[x]$: The Searching Phase. In *Proceedings ISSAC 2000*, edited by C. Traverso, 1–7 (2000)
- [2] D. Achlioptas and Y. Peres, The threshold for random k -SAT is $2^k \log 2 - O(k)$. *J. Amer. Math. Soc.* **17**, 947–973 (2004)
- [3] D. Achlioptas and Y. Peres, Random k -SAT: Two Moments Suffice to Cross a Sharp Threshold. *SIAM J. Comput.* **36**, 740–762 (2006)
- [4] E. Arnold, Modular algorithms for computing Gröbner bases. *J. Symbolic Comp.* **35**, 403–419 (2003)

-
- [5] T. Balyo and L. Chrpa, Using algorithm configuration tools to generate hard SAT benchmarks. In *Eleventh Annual Symposium on Combinatorial Search*, 133–137 (2018)
- [6] C. Barrett, P. Fontaine and C. Tinelli, The SMT-LIB Standard: Version 2.6. <http://smtlib.cs.uiowa.edu/papers/smt-lib-reference-v2.6-r2021-05-12.pdf> (2021)
- [7] E. Becker, M. Marinari, T. Mora and C. Traverso, The shape of the shape lemma. In *Proceedings ISSAC 1994*, 129–133 (1994)
- [8] D. Bini and B. Mourrain, Polynomial test suite. <http://www-sop.inria.fr/saga/POL/> (1996)
- [9] B. Birch and H. Swinnerton-Dyer, Notes on Elliptic Curves I. *J. für reine und angew. Math.* **212**, 7–23 (1963)
- [10] B. Birch and H. Swinnerton-Dyer, Notes on Elliptic Curves II. *J. für reine und angew. Math.* **218**, 79–108 (1965)
- [11] R. Bixby, Computational Progress in Linear and Mixed Integer Programming. *Presentation at ICIAM 2015* (2015)
- [12] R. Bradford, J. Davenport, M. England, A. Sadeghimanesh and A. Uncu, The DEWCAD Project: Pushing Back the Doubly Exponential Wall of Cylindrical Algebraic Decomposition. *ACM Comm. Computer Algebra* **55**, 107–111 (2021)
- [13] M. Brain, J. Davenport and A. Griggio, Benchmarking Solvers, SAT-style. *SC² 2017 Satisfiability Checking and Symbolic Computation CEUR Workshop* **1974**, 1–15 (2017)
- [14] C. Brown and J. Davenport, The Complexity of Quantifier Elimination and Cylindrical Algebraic Decomposition. In *Proceedings ISSAC 2007*, edited by C. Brown, 54–60 (2007)
- [15] S. Buswell, O. Caprotti, D. Carlisle, M. Dewar, M. Gaëtano, M. Kohlhase, J. Davenport and P. Ion, The OpenMath Standard 2.0 Revision 1. <http://www.openmath.org> (2017)
- [16] G. Butler, The transitive groups of degree fourteen and fifteen. *J. Symbolic Comp.* **16**, 413–422 (1993)
- [17] G. Butler and J. McKay, The transitive groups of degree up to 11. *Comm. Algebra* **11**, 863–911 (1983)
- [18] J. Cannon and D. Holt, The transitive permutation groups of degree 32. *Experiment. Math.* **17**, 307–314 (2008)
- [19] J. Carette, Understanding Expression Simplification. In *Proceedings ISSAC 2004*, edited by J. Gutierrez, 72–79 (2004)
- [20] A. Chistov, Double-exponential lower bound for the degree of any system of generators of a polynomial prime ideal. *St. Petersburg Math. J.* **20**, 983–1001 (2009)
- [21] C. Cohen and A. Mahboubi, A Formal Quantifier Elimination for Algebraically Closed Fields. In *Proceedings CICM 2010*, edited by S. Autexier *et al.*, 189–203 (2010)
- [22] G. Collins, Quantifier Elimination for Real Closed Fields by Cylindrical Algebraic Decomposition. In *Proceedings 2nd. GI Conference Automata Theory & Formal Languages*, 134–183 (1975)
- [23] J. Conway and R. Guy, *The Book of Numbers*. Copernicus (1996)
- [24] S. Cook, On the minimum computation time of functions. Ph.D. thesis, Department of Mathematics Harvard University (1966)
-

- [25] M. Daquino, S. Peroni, D. Shotton, G. Colavizza, B. Ghavimi, A. Lauscher, P. Mayr, M. Romanello and P. Zumstein, The OpenCitations Data Model. *International Semantic Web Conference 2020* 447–463 (2020)
- [26] J. Davenport, *On the Integration of Algebraic Functions*. Springer Lecture Notes in Computer Science 102, Springer Berlin–Heidelberg–New York (Russian ed. MIR Moscow 1985) (1981)
- [27] J. Davenport and J. Carette, The Sparsity Challenges. In *Proceedings SYNASC 2009*, edited by S. Watt *et al.*, 3–7 (2010)
- [28] J. Davenport, B. Poonen, J. Maynard, H. Helfgott, P. Huu Tiep and L. Cruz-Filipe, Machine-Assisted Proofs (ICM 2018 Panel). In *Proceedings ICM 2018*, 1085–1110 (2018)
- [29] E. Garfield, The evolution of the Science Citation Index. *International Microbiology* **10**, 65–69 (2007)
- [30] I. Gent and T. Walsh, The SAT phase transition. *ECAI* **94**, 105–109 (1994)
- [31] I. Gradshteyn and I. Ryzhik, *Table of Integrals, Series and Products* 7th edition (ed. A. Jeffrey and D. Zwillinger). Academic Press (2007)
- [32] A. Hulpke, Konstruktion transitiver Permutationsgruppen. Ph.D. thesis, RWTH Aachen (1996)
- [33] A. Hulpke, Constructing transitive permutation groups. *J. Symbolic Comput.* **39**, 1–30 (2005)
- [34] C. Jacobi, *Canon arithmeticus, sive tabulae quibus exhibentur pro singulis numeris primis vel primorum potestatibus infra 1000 numeri ad datos indices et indices ad datos numeros pertinentes*. Berolini, Typis Academicis (1839)
- [35] D. Jeffrey and A. Rich, Reducing Expression Size Using Rule-Based Integration. In *Proceedings CICM 2010*, edited by S. Autexier *et al.*, 234–246 (2010)
- [36] J. Kratz and C. Strasser, Data publication consensus and controversies (version 3). *F1000Research Article 94* **3** (2014)
- [37] G. Kremer, E. Ábrahám, M. England and J. Davenport, Cylindrical Algebraic Coverings for Satisfiability Modulo Theories Solving. *In preparation* (2021)
- [38] W. Küchlin and C. Sinz, Proving Consistency Assertions for Automotive Product Data Management. *J. Automated Reasoning* **24**, 145–163 (2000)
- [39] V. Larivière, Y. Gingras and E. Archambault, The decline in the concentration of citations, 1900–2007. *J. Amer. Soc. Info. Science and Tech.* **60**, 858–862 (2009)
- [40] A. Lenstra, H. Lenstra Jun. and L. Lovász, Factoring Polynomials with Rational Coefficients. *Math. Ann.* **261**, 515–534 (1982)
- [41] G. Leonelli, *Supplément Logarithmique*. Bordeaux (1803)
- [42] D. Mansfield, Plimpton 322: A Study of Rectangles. *To appear in Foundations of Science* (2021)
- [43] E. Mark, Maturation, fecundation, and segmentation of *Limax campestris*, Binney. *Bulletin of the Museum of Comparative Zoology at Harvard College* **6** (1881)

- [44] D. Masser and U. Zannier, Torsion points, Pell's equation and integration in elementary terms. *Acta Mathematica* **225**, 227–313 (2020)
- [45] E. Mayr and S. Ritscher, Dimension-dependent bounds for Gröbner bases of polynomial ideals. *J. Symbolic Comp.* **49**, 78–94 (2013)
- [46] H. Mooney and M. Newton, The Anatomy of a Data Citation: Discovery, Reuse, and Credit. *Journal of Librarianship and Scholarly Communication Article p.eP1035* **1** (2012)
- [47] D. Musser, On the efficiency of a polynomial irreducibility test. *J. ACM* **25**, 271–282 (1978)
- [48] D. Plaisted, New NP-Hard and NP-Complete Polynomial and Integer Divisibility Problems. *Theor. Comp. Sci.* **31**, 125–138 (1984)
- [49] D. Remler, Are 90% of academic papers really never cited? Reviewing the literature on academic citations. <http://blogs.lse.ac.uk/impactofsocialsciences/2014/04/23/academic-papers-citation-rates-remler/> (2014)
- [50] E. Robson, Neither Sherlock Holmes nor Babylon: a reassessment of Plimpton 322. *Historia Mathematica* **28**, 167–206 (2001)
- [51] G. Royle, The Transitive Groups of Degree Twelve. *J. Symbolic Comp.* **4**, 255–268 (1987)
- [52] N. Sloane, The Online Encyclopedia of Integer Sequences. *Notices A.M.S.* **50**, 912–915 (2003)
- [53] I. Spence, Weakening Cardinality Constraints Creates Harder Satisfiability Benchmarks. *J. Exp. Algorithmics Article 1.4* **20** (2015)
- [54] H. Swinnerton-Dyer, N. Stephens, J. Davenport, J. Vélu, F. Coghlan, A. Atkin and D. Tingley, Numerical Tables on Elliptic Curves. *Modular Functions of One Variable IV (Proceedings Antwerp 1972)* 75–114 (1975)
- [55] The GAP Group, GAP — Groups, Algorithms, and Programming, Version 4.11.1. <https://www.gap-system.org> (2021)
- [56] S. van de Sandt, L. Nielsen, A. Ioannidis, A. Muench, E. Henneken, A. Accomazzi, C. Bigarella, J. Lopez and S. Dallmeier-Tiessen, Practice meets Principle: Tracking Software and Data Citations to Zenodo DOIs. <https://arxiv.org/abs/1911.00295> (2019)
- [57] D. Wilson, R. Bradford and J. Davenport, A Repository for CAD Examples. *ACM Communications in Computer Algebra* **46**, 67–69 (2012)
- [58] D. Wilson, J. Davenport, M. England and R. Bradford, A “Piano Movers” Problem Reformulated. In *Proceedings SYNASC 2013*, 53–60 (2013)
- [59] J. Zech, *Tafeln des Additions- und Subtractions- Logarithmen*. Weidmann (1849)

James H. Davenport

Department of Computer Science, University of Bath, Bath U.K.

J.H.Davenport@bath.ac.uk