Digital collections of examples in mathematical sciences

James H. Davenport

Abstract. 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.
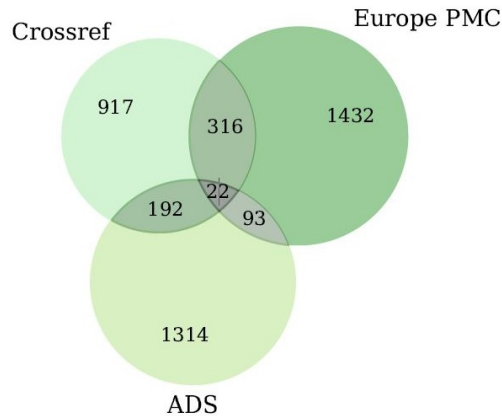
Figure 1. Overlaps between data citation harvesters [56, Figure 5].

Despite the success of article citation, data citation is a mess in practice [56]: only 1.16% of dataset DOIs in Zenodo are cited[1] (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. Assuming independence and looking at the overlap statistics, we can estimate that there are between 4,000 to 20,000 datasets waiting to be cited. In such circumstances, de facto people cite a paper if they can find one.

2. Pure mathematics

2.1. On-line 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!

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.

---

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

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) $n = 32$ is not in the default build for GAP.

These are really great resources (if that is 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 [27].

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 as follows: given a 3-literals/clause CNF satisfiability problem,

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

where $l_{i,j} \in \{x_1, \overline{x_1}, x_2, \overline{x_2}, \ldots\}$; 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 $\overline{x_i}$) 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 contestants[2] and others), then produce a list of problems (in DIMACS, a standard format), 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 $\times 2$/year. For comparison, Linear Programming has done $\times 1.8$ over a greater timeline and with more rigorous dcoumentation [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 $F_i(y_1, \ldots, y_n)$, and a CNF $\mathcal{L}$ in the form of (1) with $F_i(y_1, \ldots, y_n)$ rather than just x_i. In principle, T can be anything: those currently supported[3] are given in Figure 2.

---

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

[3]By the SMT-LIB standard [6], which also says " New logics are added to the standard opportunistically, once enough benchmarks are available".
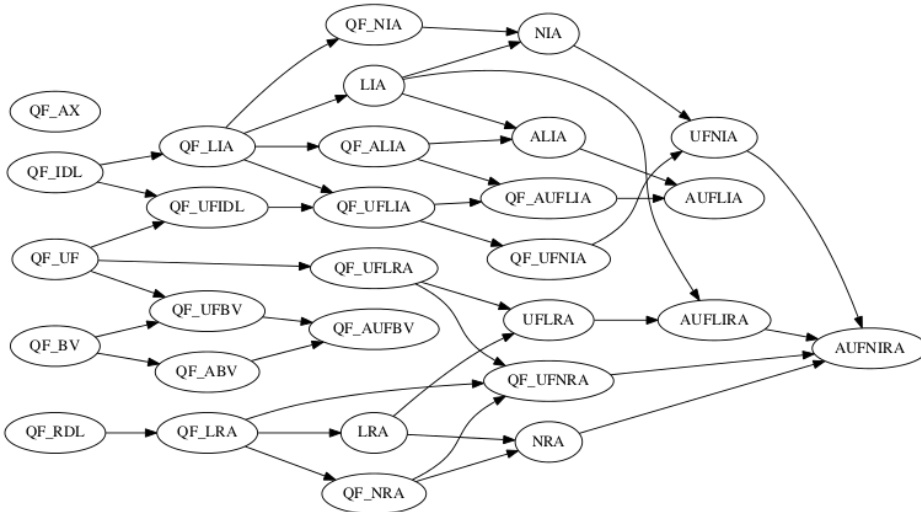
Figure 2. Available logics (March 2022) https://smtlib.cs.uiowa.edu/logics.shtml.

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 Open-Math [15], but it is 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) [26, 48]. Even for dense polynomials, it can be challenging for multivariates. 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 needs 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 (with respect to 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

$$\{p_1(x_1), x_2 - p_2(x_1), x_3 - p_3(x_1), \ldots, x_n - p_n(x_1)\}, \tag{2}$$

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 (2).

The good news from the point of view of this paper is that there is a collection [8], but it is 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 is always a Gröbner base (no concept of UNSAT as such) but it is not clear what a useful certificate of "G is a Gröbner base for input L" might mean in general (but see [4]). If $G = \{g_1, \ldots, g_M\}$ is a Gröbner base of $F = \{f_1, \ldots, f_N\}$, 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 (with respect to 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), with respect to one order, of $\mathbf{R}^{3n+4}$ sign-invariant for p has $O\left(2^{2^n}\right)$ cells, but with respect to another order has 3 cells:

$$p := x^{n+1}\left(\left(y_{n-1} - \frac{1}{2}\right)^2 + (x_{n-1} - z_n)^2\right)\left((y_{n-1} - z_n)^2 + (x_{n-1} - x_n)^2\right)$$

$$+ \sum_{i=1}^{n-1} x^{i+1}\left((y_{i-1} - y_i)^2 + (x_{i-1} - z_i)^2\right)\left((y_{i-1} - z_i)^2 + (x_{i-1} - x_i)^2\right)$$

$$+ x\left((y_0 - 2x_0)^2 + \left(\alpha^2 + \left(x_0 - \frac{1}{2}\right)\right)^2\right)$$

$$\times \left((y_0 - 2 + 2x_0)^2 + \left(\alpha^2 + \left(x_0 - \frac{1}{2}\right)\right)^2\right) + a.$$

The bad order (eliminating x, then $y_0, \alpha, x_0, z_1, y_1, z_1, \ldots, x_n, a$) 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 [58], 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 [57], 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 $\Psi(g_1, \ldots, g_{n'})$, where Ψ is a Boolean combination of equalities and inequalities in the g_i, polynomials in $x_1, \ldots, x_{k-1}$, 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 that this cannot 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 [28, 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. [28]) 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

$$\int \frac{5x^4}{(1+x)^6} \, dx = \frac{x^5}{(1+x)^5}, \tag{3}$$

where Maple's answer is

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

Note that (4) is not just an ugly form of the right-hand side of (3): 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.

References

[1] J. Abbott, V. Shoup, and P. Zimmermann, Factorization in $\mathbb{Z}[x]$: the searching phase. In *ISSAC 2000*, edited by C. Traverso, pp. 1–7, ACM, New York, 2000   Zbl 1326.68339

[2] D. Achlioptas and C. Moore, Random k-SAT: two moments suffice to cross a sharp threshold. *SIAM J. Comput.* **36** (2006), no. 3, 740–762   Zbl 1120.68096   MR 2263010

[3] D. Achlioptas and Y. Peres, The threshold for random k-SAT is $2^k \log 2 - O(k)$. *J. Amer. Math. Soc.* **17** (2004), no. 4, 947–973   Zbl 1093.68075   MR 2083472

[4] E. A. Arnold, Modular algorithms for computing Gröbner bases. *J. Symbolic Comput.* **35** (2003), no. 4, 403–419   Zbl 1046.13018   MR 1976575

[5] T. Balyo and L. Chrpa, Using algorithm configuration tools to generate hard SAT benchmarks. In *The Eleventh International Symposium on Combinatorial Search (SoCS 2018)*, pp. 133–137, 2018

[6] C. Barrett, P. Fontaine, and C. Tinelli, The SMT-LIB standard: Version 2.6. 2021, http://smtlib.cs.uiowa.edu/papers/smt-lib-reference-v2.6-r2021-05-12.pdf

[7] E. Becker, M. G. Marinari, T. Mora, and C. Traverso, The shape of the Shape Lemma. In *ISSAC 1994*, pp. 129–133, ACM, Baltimore, MD, 1994   Zbl 0925.13006

[8] D. Bini and B. Mourrain, Polynomial test suite. 1996, http://www-sop.inria.fr/saga/POL/

[9] B. J. Birch and H. P. F. Swinnerton-Dyer, Notes on elliptic curves. I. *J. Reine Angew. Math.* **212** (1963), 7–25   Zbl 0118.27601   MR 146143

[10] B. J. Birch and H. P. F. Swinnerton-Dyer, Notes on elliptic curves. II. *J. Reine Angew. Math.* **218** (1965), 79–108   Zbl 0147.02506   MR 179168

[11] R. Bixby, Computational progress in linear and mixed integer programming. 2015, presentation at ICIAM 2015

[12] R. Bradford, J. H. Davenport, M. England, A. Sadeghimanesh, and A. Uncu, The DEW-CAD Project: pushing back the Doubly Exponential Wall of Cylindrical Algebraic Decomposition. *ACM Commun. Comput. Algebra* **55** (2021), no. 3, 107–111   MR 4363371

[13] M. Brain, J. Davenport, and A. Griggio, Benchmarking solvers, SAT-style. In *SC^2 2017 Satisfiability Checking and Symbolic Computation CEUR Workshop 1974*, pp. 1–15, 2017

[14] C. W. Brown and J. H. Davenport, The complexity of quantifier elimination and cylindrical algebraic decomposition. In *ISSAC 2007*, edited by C. Brown, pp. 54–60, ACM, New York, 2007   Zbl 1190.68028   MR 2396184

[15] S. Buswell et al., The OpenMath Standard 2.0 Revision 1. 2017, http://www.openmath.org

[16] G. Butler, The transitive groups of degree fourteen and fifteen. *J. Symbolic Comput.* **16** (1993), no. 5, 413–422   Zbl 0813.20003   MR 1271082

[17] G. Butler and J. McKay, The transitive groups of degree up to eleven. *Comm. Algebra* **11** (1983), no. 8, 863–911   Zbl 0518.20003   MR 695893

[18] J. J. Cannon and D. F. Holt, The transitive permutation groups of degree 32. *Experiment. Math.* **17** (2008), no. 3, 307–314   Zbl 1175.20004   MR 2455702

[19] J. Carette, Understanding expression simplification. In *ISSAC 2004*, pp. 72–79, ACM, New York, 2004   Zbl 1134.68596   MR 2126927

[20] A. L. Chistov, Double-exponential lower bound for the degree of any system of generators of a polynomial prime ideal. *St. Petersburg Math. J.* **20** (2009), no. 6, 983–1001   Zbl 1206.13031

[21] C. Cohen and A. Mahboubi, A formal quantifier elimination for algebraically closed fields. In *CICM 2010*, edited by S. Autexier et al., pp. 189–203, Springer, Berlin, 2010   Zbl 1286.68394

[22] G. E. Collins, Quantifier elimination for real closed fields by cylindrical algebraic decomposition. In *Automata Theory and Formal Languages (Second GI Conf., Kaiserslautern, 1975)*, pp. 134–183, Lecture Notes in Comput. Sci. 33, Springer, Berlin, 1975   Zbl 0318.02051   MR 0403962

[23] J. H. Conway and R. K. Guy, *The Book of Numbers*. Copernicus, New York, 1996   Zbl 0866.00001   MR 1411676

[24] S. Cook, *On the minimum computation time of functions*. Ph.D. thesis, Department of Mathematics, Harvard University, 1966

[25] M. Daquino et al., The OpenCitations data model. In *The 19th International Semantic Web Conference (ISWC 2020)*, pp. 447–463, 2020

[26] J. Davenport and J. Carette, The sparsity challenges. In *SYNASC 2009*, edited by S. Watt et al., pp. 3–7, 2010

[27] J. Davenport, B. Poonen, J. Maynard, H. Helfgott, P. Tiep, and L. Cruz-Filipe, Machine-assisted proofs. In *Proceedings of the International Congress of Mathematicians—Rio de Janeiro 2018. Vol. I. Plenary Lectures*, pp. 1085–1110, World Sci. Publ., Hackensack, NJ, 2018   Zbl 1452.68262   MR 3966753

[28] J. H. Davenport, *On the Integration of Algebraic Functions*. Lecture Notes in Comput. Sci. 102, Springer, Berlin, 1981   Zbl 0471.14009   MR 617377

[29] E. Garfield, The evolution of the Science Citation Index. *Int. Microbiol.* **10** (2007), 65–69

[30] I. Gent and T. Walsh, The SAT phase transition. In *ECAI 1994*, edited by A. Cohn, pp. 105–109, John Wiley, New York, 1994

[31] I. S. Gradshteyn and I. M. Ryzhik, *Table of Integrals, Series, and Products*. 7th edn., Academic Press, Amsterdam, 2007   Zbl 1208.65001   MR 2360010

[32] A. Hulpke, *Konstruktion transitiver Permutationsgruppen*. Ph.D. thesis, RWTH Aachen, 1996

[33] A. Hulpke, Constructing transitive permutation groups. *J. Symbolic Comput.* **39** (2005), no. 1, 1–30   Zbl 1131.20003   MR 2168238

[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*. Typis Academicis, Berolini, 1839

[35] D. J. Jeffrey and A. D. Rich, Reducing expression size using rule-based integration. In *CICM 2010*, edited by S. Autexier et al., pp. 234–246, Springer, Berlin, 2010   Zbl 1286.68517

[36] J. Kratz and C. Strasser, Data publication consensus and controversies [version 3]. *F1000Research* **3** (2014), Article No. 94

[37] G. Kremer, E. Ábrahám, M. England, and J. H. Davenport, On the implementation of cylindrical algebraic coverings for satisfiability modulo theories solving. In *SYNASC 2021*, pp. 37–39, 2021

[38] W. Küchlin and C. Sinz, Proving consistency assertions for automotive product data management. *J. Autom. Reasoning* **24** (2000), no. 1–2, 145–163   Zbl 0968.68042

[39] V. Larivière, Y. Gingras, and E. Archambault, The decline in the concentration of citations, 1900–2007. *J. Amer. Soc. Info. Sci. Technol.* **60** (2009), no. 4, 858–862

[40] A. K. Lenstra, H. W. Lenstra Jr., and L. Lovász, Factoring polynomials with rational coefficients. *Math. Ann.* **261** (1982), no. 4, 515–534   Zbl 0488.12001   MR 682664

[41] G. Leonelli, *Supplément logarithmique. Théorie des logarithmes additionels et diductifs*. Brossier, Bordeaux, 1803

[42] D. F. Mansfield, Plimpton 322: a study of rectangles. *Found. Sci.* **26** (2021), no. 4, 977–1005   Zbl 07554371   MR 4334265

[43] E. Mark, Maturation, fecundation, and segmentation of Limax campestris, Binney. *Bulletin of the Museum of Comparative Zoology at Harvard College* **6** (1881), no. 12, 173–625

[44] D. Masser and U. Zannier, Torsion points, Pell's equation, and integration in elementary terms. *Acta Math.* **225** (2020), no. 2, 227–313   Zbl 1470.11163   MR 4205408

[45] E. W. Mayr and S. Ritscher, Dimension-dependent bounds for Gröbner bases of polynomial ideals. *J. Symbolic Comput.* **49** (2013), 78–94   Zbl 1258.13032   MR 2997841

[46] H. Mooney and M. Newton, The anatomy of a data citation: Discovery, reuse, and credit. *Journal of Librarianship and Scholarly Communication* **1** (2012), no. 1, Article No. eP1035

[47] D. R. Musser, On the efficiency of a polynomial irreducibility test. *J. Assoc. Comput. Mach.* **25** (1978), no. 2, 271–282   Zbl 0372.68014   MR 488309

[48] D. A. Plaisted, New NP-hard and NP-complete polynomial and integer divisibility problems. *Theoret. Comput. Sci.* **31** (1984), no. 1-2, 125–138   Zbl 0572.68027   MR 752098

[49] D. Remler, Are 90% of academic papers really never cited? Reviewing the literature on academic citations. 2014, http://blogs.lse.ac.uk/impactofsocialsciences/2014/04/23/academic-papers-citation-rates-remler/

[50] E. Robson, Neither Sherlock Holmes nor Babylon: a reassessment of Plimpton 322. *Historia Math.* **28** (2001), no. 3, 167–206   Zbl 0991.01001   MR 1849797

[51] G. F. Royle, The transitive groups of degree twelve. *J. Symbolic Comput.* **4** (1987), no. 2, 255–268   Zbl 0683.20002   MR 922391

[52] N. J. A. Sloane, The On-Line Encyclopedia of Integer Sequences. *Notices Amer. Math. Soc.* **50** (2003), no. 8, 912–915   Zbl 1044.11108   MR 1992789

[53] I. Spence, Weakening cardinality constraints creates harder satisfiability benchmarks. *ACM J. Exp. Algorithmics* **20** (2015), Article No. 1.4   MR 3353196

[54] H. Swinnerton-Dyer et al., Numerical tables on elliptic curves. In *Modular Functions of One Variable IV*, pp. 74–144, Lecture Notes in Math. 476, Springer, Berlin, 1975 Zbl 1214.11006

[55] The GAP Group, GAP—Groups, algorithms, and programming, version 4.11.1. 2021, https://www.gap-system.org

[56] S. van de Sandt et al., Practice meets principle: Tracking software and data citations to Zenodo DOIs. 2019, arXiv:1911.00295

[57] D. Wilson, J. Davenport, M. England, and R. Bradford, A "Piano Movers" Problem reformulated. In *SYNASC 2013*, pp. 53–60, 2013

[58] D. J. Wilson, R. J. Bradford, and J. H. Davenport, A repository for CAD examples. *ACM Commun. Comput. Algebra* **46** (2012), no. 3, 67–69   Zbl 1322.68294

[59] J. Zech, *Tafeln der Additions- und Subtractions-Logarithmen*. Weidmannsche Buchhandlung, Berlin, 1849

James H. Davenport

Department of Computer Science, University of Bath, Bath BA2 7AY, UK;
j.h.davenport@bath.ac.uk