

# The “Seven Dwarfs” of Symbolic Computation\*

Erich L. Kaltofen

Dept. of Mathematics, North Carolina State University,  
Raleigh, North Carolina 27695-8205, USA  
kaltofen@math.ncsu.edu; <http://www.kaltofen.us>

August 24, 2014

## Abstract

We present the Seven Dwarfs of Symbolic Computation, which are sequential and parallel algorithmic methods that today carry a great majority of all exact and hybrid symbolic compute cycles.

- SymDwf 1. Exact linear algebra, integer lattices
- SymDwf 2. Exact polynomial and differential algebra, Gröbner bases
- SymDwf 3. Inverse symbolic problems, e.g., interpolation and parameterization
- SymDwf 4. Tarski’s algebraic theory of real geometry
- SymDwf 5. Hybrid symbolic-numeric computation
- SymDwf 6. Computation of closed form solutions
- SymDwf 7. Rewrite rule systems and computational group theory

We will elaborate on each dwarf and compare with Colella’s seven and the Berkeley team’s thirteen dwarfs of scientific computing.

## Introduction

Phillip Colella [2004] in his 2004 presentation “Defining Software Requirements for Scientific Computing” about DARPA’s High Productivity Computing Systems (HPCS) program gave his list of the now-famous “*Seven Dwarfs*” of algorithms for high-end simulation in the physical sciences.

- |                                |                               |                     |
|--------------------------------|-------------------------------|---------------------|
| HPCS 1. Structured Grids       | HPCS 4. Dense Linear Algebra  | HPCS 7. Monte Carlo |
| HPCS 2. Unstructured Grids     | HPCS 5. Sparse Linear Algebra |                     |
| HPCS 3. Fast Fourier Transform | HPCS 6. Particles             |                     |

The dwarfs, in allusion to the fairy tale, mine compute cycles for golden results. Recently, the term “killer kernels” has been used to replace the notion of dwarf, but the dwarfs seem

---

\*This material is based on work supported in part by the National Science Foundation under Grants CCF-0830347, CCF-0514585 and DMS-0532140.

more like library procedures than operating system kernels. Following Colella, researchers in parallel computation at the University of California at Berkeley, who include David Patterson and Katherine Yelick, have modified and upgraded to 13 dwarfs, where “A dwarf is an algorithmic method that captures a pattern of computation and communication [URL [http://view.eecs.berkeley.edu/wiki/Dwarf\\_Mine](http://view.eecs.berkeley.edu/wiki/Dwarf_Mine)]:”

- |                                   |                                             |
|-----------------------------------|---------------------------------------------|
| Berkeley 1. Dense Linear Algebra  | Berkeley 8. Combinational Logic             |
| Berkeley 2. Sparse Linear Algebra | Berkeley 9. Graph Traversal                 |
| Berkeley 3. Spectral Methods      | Berkeley 10. Dynamic Programming            |
| Berkeley 4. N-Body Methods        | Berkeley 11. Backtrack and Branch-and-Bound |
| Berkeley 5. Structured Grids      | Berkeley 12. Graphical Models               |
| Berkeley 6. Unstructured Grids    | Berkeley 13. Finite State Machines          |
| Berkeley 7. MapReduce             |                                             |

Both lists are notably numerical computing oriented. They exclude symbolic computation, i.e., methods with exact arithmetic, or logic programming, say rewriting via rules, altogether. However, they inspire to make a corresponding list, and here we will do so for symbolic computation. Bruno Buchberger [1985] in his 1985 editorial in the first issue of *the Journal of Symbolic Computation* makes an attempt to define the discipline of symbolic computation. We adopt his breadth and view symbolic computation to include all of computer algebra [Kaltofen 1987; Grabmeier et al. 2003] and also algebraic methods for analysis, statistics and combinatorics, logic programming, computational geometry and program synthesis. The report [Boyle and Caviness 1989] offers a then glimpse into the future of symbolic computation and has made several accurate predictions (see, e.g., Section 5 below).

Here we add to this taxonomy via our seven dwarfs of symbolic computation. Our methods are oriented to mid-level and high performance computation tasks, and should not be considered comprehensive. A subject on the boundary not included is computational number theory. The important application of symbolic computation to mathematics education is not discussed. Education tasks can be compute intensive. For example, the automatic grading of the Maple homework worksheets of our calculus classes by NCSU’s egrader software consumes an entire night. On the low performance side, micro symbolic computation systems for compact devices such as cell phones constitute an important educational application of the discipline: vastly more people world-wide own cell phones than computers.

We presented the list in the talk “The Seven Dwarfs of Symbolic Computation and the Discovery of Reduced Symbolic Models” [URL <http://www.math.ncsu.edu/~kaltofen/bibliography/07/SNSC07.pdf>] at 4th International Conference on Symbolic and Numerical Scientific Computing SNSC ’08 at RISC Linz, Hagenberg, Austria, on July 24, 2008. In the following, we briefly discuss each dwarf and give selected references, which are meant to highlight some past and current results and not as a complete survey as other important work could not be included.

## 1. Exact linear algebra, integer lattices

Important breakthroughs in exact linear algebra actually happened later than those in polynomial algebra, notably after Buchberger’s Gröbner basis algorithm. One is the discovery of exact sparse iterative algorithms based on the numeric Krylov and Lanczos algorithms

[Wiedemann 1986; Kaltofen and Saunders 1991] and their block versions [Coppersmith 1994; Kaltofen 1995; Villard 1997] whose probabilistic analysis for small coefficient fields is being completed today [Eberly 2010]. The algorithms are available in the open source LinBox library [URL [www.linalg.org](http://www.linalg.org)], callable from the SAGE and Maple platforms, and put to important use. A second breakthrough are the lattice basis reduction algorithms [Ferguson and Forcade 1982; Lenstra et al. 1982] that today have greatly improved implementations [Novocin et al. 2011] and are used extensively for discovery of exact identities from numeric approximations ([Håstad et al. 1989], “the inverse symbolic calculator” [URL <http://oldweb.ccm.sfu.ca/projects/ISC/ISCmain.html>]).

We observe additional trends today: Strassen’s fast matrix multiplication algorithm and cache-efficient BLAS libraries improve performance of exact linear algebra [Dumas et al. 2008]; characteristic polynomials and integer Smith normal forms of sparse integer matrices [Dumas et al. 2001; Giesbrecht 2001] are important invariants, for instance in computing the so-called bar code of a persistent topology of data; and structured exact linear problem solvers such as the matrix Berlekamp/Massey algorithm [Kaltofen and Yuhasz 2013] form a fundamental ingredient in sparse solvers.

Exact linear algebra algorithms are easily underestimated. Great progress has been made in the past ten years, and the software has a wide range of applications. Exact solutions are not only needed for finite field entries, but also for diophantine problems and when the exact input forms an ill-conditioned matrix.

## 2. Exact polynomial and differential algebra, Gröbner bases

Polynomial arithmetic including the computation of multivariate polynomial greatest common divisors, factorizations, and triangular and other canonical forms for polynomial systems constitute the heart of computer algebra. Classical tools include resultant computation and Hensel lifting and modern tools Buchberger’s Gröbner basis algorithm. Truncated power series are represented by polynomials and thus included in this dwarf.

The calculus of differentiation and differential ideals allows manipulation of differential equations as polynomials with a derivative operator. In addition, one can interpret the derivative (or difference) operator as a new symbol and construct composed operators as polynomials with variables and derivative (difference) symbols. Those operator rings are generalized to Ore extensions and have an additional, special, non-commutative multiplication. Two references are [Rosenkranz and Regensburger 2008; Gao et al. 2009]. See also Section 6.

Efficient implementations of polynomial factoring and Gröbner basis algorithms, for instance Jean-Charles Faugere’s FGB which is also callable from within Maple, make a serious use of the methods as easy as, say, Matlab gives access to numerical linear algebra. Today’s applications are abundant, e.g., cryptosystems have been broken with them.

Basic polynomial arithmetic of multivariate polynomials forms the core infrastructure of any symbolic manipulation system, and efficiency improvements can still be made: any speedup will speed many application algorithms. This is the more true with the arrival of multicore and multiprocessor workstations.

### 3. Inverse symbolic problems, e.g., interpolation and parameterization

Interpolation and curve fitting are basic and important operations to build mathematical models from data. Zippel’s [1990] and Ben-Or and Tiwari’s [1988] sparse multivariate polynomial algorithms are a fundamental contribution from symbolic computation to the task of function/model recovery. The paradigm of early termination via randomization has successfully been exploited [Kaltofen and Lee 2003]. In Section 5 we point to new numerical methods that were derived from the exact symbolic algorithms. More recently polynomial and rational function recovery with very high degree terms have been achieved [Garg and Schost 2009; Giesbrecht and Roche 2010; Kaltofen and Nehring 2011]. There the values are determined at roots of unity to prevent size explosion. Beyond polynomial and rational function recovery is, for instance, recovery of algebraic functions and differential equations from series solutions.

The circle as an implicitly represented curve  $x^2 + y^2 = 1$  can be rationally parameterized as  $x = \cos(\alpha) = (1 - t^2)/(t^2 + 1)$ ,  $y = \sin(\alpha) = 2t/(t^2 + 1)$  with  $-\infty \leq t = \tan(\alpha/2) \leq \infty$ . Not all real curves can be so parameterized, for instance elliptic curves. A reference is the book [Sendra et al. 2007]. Parametric curves form basic objects in geometric rendering.

Interpolation and Chinese remaindering forms the recovery step in computing with homomorphic images, where a computation is split by first computing the solution for various values of a symbolic parameter and then the symbolic solution is interpolated from those values. Because each value can be processed separately and no intermediate degree/size growth occurs, the paradigm constitutes a powerful and parallel/distributed approach.

### 4. Tarski’s algebraic theory of real geometry

Tarski’s algorithm for eliminating quantifiers in sentences formed on semi-algebraic sets makes most of Euclidean geometry and real polynomial optimization decidable. Unfortunately, the general method solves problems in a high complexity class (super-exponential). Nonetheless, George Collins’s cylindrical algebraic decomposition algorithm is implemented and has solved non-trivial problems. References are the collection [Caviness and Johnson 1998] and [Brown 2009], which has references to newer work.

A fundamental quantifier elimination problem is to determine whether a multivariate polynomial  $f(x_1, \dots, x_n)$  has a real root, which we shall call Seidenberg’s problem. For instance, deciding if a polynomial can attain negative values, i.e., is not positive semidefinite, is equivalent to deciding if  $f(x_1, \dots, x_n)x_{n+1}^2 + 1$  has a real root. Thus all (unconstrained) polynomial inequalities are reduced to Seidenberg’s problem. A more general fundamental problem is to compute a sample point in each connected component of the real solution set of a system of polynomial equations.

Modern software, such as RAGlib [Safey El Din 2008], analyzes the real critical values via Gröbner basis computation. A variant of Tarski’s quantifier elimination problem that weakens the pre- and post conditions and thus lowers the intrinsic complexity can be based on such real polynomial software [Hong and Safey El Din 2009].

Hilbert’s Problem on polynomial sums-of-squares and Artin’s Theorem offers an additional approach to real polynomial optimization, which is made possible by numerical non-linear optimization and discussed in Section 5.

## 5. Hybrid symbolic-numeric computation

The use of approximate, floating point, arithmetic and approximations of irrational functions by polynomials and rational functions is as old as logarithm tables and Taylor series and Padé fractions. Section 2.12.3 in [Grabmeier et al. 2003] describes what constitutes hybrid symbolic-numeric computation. Our description already contains the fundamental concept of computing a nearest polynomial, measured in some distance norm, that satisfies a property which the input polynomial does not. Classical properties are having non-trivial polynomial greatest common divisors and factors, or common solutions (the nearest consistent system) or solutions that have real components (the nearest polynomial with a real root) or higher multiplicities (contracting clusters of zeros to a single common point). The inputs are not exact, because of physical measurement or because the scalars come from a floating point computation, and therefore lack the needed property. The sought property may have to be avoided, and a lower bound on the distance yields a condition number. New work and references are found in the proceedings [Wang and Zhi 2007; Verschelde and Watt 2007; Kai and Sekigawa 2009].

Because there is a gradual transition to mostly numerical solution of, say, algebraic geometry problems, e.g., via programs like Bertini [URL <http://www.nd.edu/~sommese/bertini/>] and PHCpack [URL <http://www.math.uic.edu/~jan/PHCpack/phcpack.html>], the symbolic computation component in the hybrid approach is sometimes dismissed. Clearly, the algorithms for sparse approximate interpolation [Giesbrecht et al. 2009; Kaltofen et al. 2007] are based on the exact sparse polynomial interpolation algorithms by Zippel and by Ben-Or and Tiwari. Those hybrid algorithms have applications to sparse signal processing and compressive sensing. The approximate Buchberger-Möller algorithm has found an application in analyzing data from oil wells [URL <http://www.algebraic-oil.uni-passau.de/>].

Any positive semidefinite polynomial  $f$  with real (rational) coefficients (see Section 4) can be written as a finite sum

$$f(x_1, \dots, x_n) = \frac{1}{g_0(x_1, \dots, x_n)^2} \sum_{i=1}^k g_i(x_1, \dots, x_n)^2, \quad (1)$$

where  $g_i$  are polynomials with real (rational) coefficients. If there exist  $g_i$  with  $g_0 = 1$ ,  $f$  is said to be SOS, but not all  $f$  are, e.g., Motzkin’s polynomial. Any polynomial inequality  $f \geq h$  is equivalent to  $f - h$  being positive semidefinite;  $h$  in global optimization is the real infimum (or a rational lower bound) of all values of  $f$ . Therefore, any  $g_i$  satisfying  $f - h = 1/g_0^2 \sum_i g_i^2$  constitute a proof (exact certificate) for the inequality/optimum. Two recent developments have made it possible to compute such certificates. The first are the numerical optimization algorithms for semidefinite programming. The second is a symbolic technique for converting an imprecise SOS with floating point coefficients to an exact identity over the rational numbers [Peyrl and Parrilo 2008; Kaltofen et al. 2008, 2012]. Among the recent successes are the proof of the Monotone Column Permanent Conjecture for  $n = 4$  [Kaltofen

et al. 2009], which was completed shortly before the general conjecture could be established, the Bessis-Moussa-Villani (BMV) conjecture for  $m \leq 13$  [Klep and Schweighofer 2008], new SOS proofs for many known inequalities, and a deformation analysis approach to Seidenberg’s problem of Section 4 [Hutton et al. 2010]. Optimization with additional polynomial inequality constraints are handled by various so-called Positivstellensätze [Marshall 2008].

## 6. Computation of closed form solutions

Robert Risch’s 1970 solution of Hardy’s problem to determine if an indefinite integral can be expressed in closed form as an expression in elementary functions is a hallmark of early symbolic computation. Closed form solutions to differential equations and the inclusion of special functions, possibly defined by lower order differential equations constitutes an active area of research. References are the book [van der Put and Singer 2003] and [Yuan and van Hoeij 2010], which has references to newer work. A connection to differential elimination theory of Section 2 should be noted.

Algorithms for closed form solutions for discrete summations, difference equations, and combinatorial counts form an active subarea of symbolic computation which could be named “symbolic combinatorics” (Michael F. Singer). The members of Peter Paule’s research group, some of who are part of the Austrian DK research grant “Numerical and Symbolic Scientific Computing,” have made significant recent contributions to the area of symbolic combinatorics: URL <http://www.risc.uni-linz.ac.at/research/combinat/risc/publications/>. An example is the closed form solution for the generating function for counting so-called Gessel walks, which turned out to be an algebraic function in three variables [Bostan and Kauers 2010], which was discovered in collaboration with the Algorithms Project at INRIA [URL <http://algo.inria.fr/index.html>].

## 7. Rewrite rule systems and computational group theory

Computational group and representation theory is a traditional subject lying in the intersection of symbolic computation and combinatorics. Famous popular examples are to compute the minimum number of moves necessary for solving Rubik’s cube puzzle from any configuration [Kunkle and Cooperman 2009], which was recently completed on a Google data center <http://cube20.org>. Group decomposition plays a major role in the synthesis of high performance FFT library [Püschel et al. 2005].

Bruno Buchberger included rewrite rule systems as a subject of symbolic computation, motivated perhaps by the interpretation of his Gröbner basis algorithm as a critical-pair/completion method (Knuth-Bendix completion). Rewrite techniques are often deployed for expression simplification in symbolic computation. The RTA conference series [URL <http://rewriting.loria.fr/rta/>] covers the many applications beyond symbolic computation (see also the Coq proof assistant <http://www.lix.polytechnique.fr/coq/>). Algebraic techniques are also be applied to algorithm synthesis, such as automatic differentiation [Griewank

2008] and the transposition principle for matrix-times-vector products or elimination of divisions from algebraic algorithms.

**Acknowledgments:** I thank Bruno Salvy for his thoughtful comments.

## References

- Ben-Or, M. and Tiwari, P. A deterministic algorithm for sparse multivariate polynomial interpolation. In *Proc. Twentieth Annual ACM Symp. Theory Comput.*, pages 301–309, New York, N.Y., 1988. ACM Press.
- Bostan, Alin and Kauers, Manuel. The complete generating function for Gessel walks is algebraic. *Proceedings of the AMS*, 2010. with an Appendix by Mark van Hoeij. To appear. URL <http://www.risc.uni-linz.ac.at/people/mkauers/publications/bostan10.pdf>.
- Boyle, Ann and Caviness, B. F., editors. *Future Directions for Research in Symbolic Computation*. SIAM, Philadelphia, 1989. Report of a Workshop on Symbolic and Algebraic Computation April 29–30, 1988 Washington DC. Anthony C. Hearn Workshop Chairperson. URL <http://www.cis.udel.edu/~caviness/wsreport.pdf>.
- Brown, Christopher W. Fast simplification of Tarski formulas. In [May 2009], pages 63–70, 2009.
- Buchberger, Bruno. Symbolic computation (an editorial). *J. Symbolic Comput.*, 1(1):1–6, 1985.
- Caviness, B. F. and Johnson, J. R., editors. *Quantifier Elimination and Cylindrical Algebraic Decomposition*. Springer Verlag, Berlin/Heidelberg, 1998.
- Colella, Phillip. Defining software requirements for scientific computing. Slide of 2004 presentation included in David Patterson’s 2005 talk, 2004. URL <http://www.lanl.gov/orgs/hpc/salishan/salishan2005/davidpatterson.pdf>.
- Coppersmith, D. Solving homogeneous linear equations over  $GF(2)$  via block Wiedemann algorithm. *Math. Comput.*, 62(205):333–350, 1994.
- Dumas, Jean-Guillaume, Giorgi, Pascal, and Pernet, Clément. Dense linear algebra over finite fields: the FFLAS and FFPACK packages. *ACM Trans. Math. Software*, 35(3):1–42, November 2008.
- Dumas, Jean-Guillaume, Saunders, B. David, and Villard, Gilles. On efficient sparse integer matrix Smith normal form computation. *J. Symbolic Comput.*, 32(1/2):71–99, 2001. Special issue on Computer Algebra and Mechanized Reasoning: Selected St. Andrews’ ISSAC/Calculus Contributions. Guest editors: T. Recio and M. Kerber.
- Eberly, Wayne. Yet another block Lanczos algorithm: How to simplify the computation and reduce reliance on preconditioners in the small field case. In [Watt 2010], page to appear, July 2010.

- Ferguson, H. R. P. and Forcade, R. W. Multidimensional Euclidean algorithms. *J. reine angew. Math.*, 334:171–181, 1982.
- Gao, Xiao-Shan, der Hoeven, J. Van, Yuan, C. M., and Zhang, Gui-Lin. Characteristic set method for differential-difference polynomial systems. *J. Symb. Comput.*, 44(9):1137–1163, 2009.
- Garg, Sanchit and Schost, Éric. Interpolation of polynomials given by straight-line programs. *Theoretical Computer Science*, 410(27-29):2659 – 2662, 2009. ISSN 0304-3975. URL <http://www.csd.uwo.ca/~eschost/publications/interp.pdf>.
- Giesbrecht, M. Fast computation of the Smith form of a sparse integer matrix. *Computational Complexity*, 10:41–69, 2001.
- Giesbrecht, Mark, Labahn, George, and shin Lee, Wen. Symbolic-numeric sparse interpolation of multivariate polynomials. *J. Symbolic Comput.*, 44:943–959, 2009.
- Giesbrecht, Mark and Roche, Daniel S. Interpolation of shifted-lacunary polynomials. *Computational Complexity*, 19(3):333–354, September 2010.
- Grabmeier, J., Kaltofen, E., and Weispfenning, V., editors. *Computer Algebra Handbook*. Springer Verlag, Heidelberg, Germany, 2003. ISBN 3-540-65466-6. 637 + xx pages + CD-ROM.
- Griewank, Andreas. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. SIAM Publications, Philadelphia, 2008.
- Håstad, Johan, Just, B., Lagarias, J. C., and Schnorr, C. P. Polynomial time algorithms for finding integer relations among real numbers. *SIAM J. Comput.*, 18(5):859–881, 1989.
- Hong, Hoon and Safey El Din, Mohab. Variant real quantifier elimination: Algorithm and application. In [May 2009], pages 183–190, 2009.
- Hutton, Sharon E., Kaltofen, Erich L., and Zhi, Lihong. Computing the radius of positive semidefiniteness of a multivariate real polynomial via a dual of Seidenberg’s method. In [Watt 2010], pages 227–234, July 2010. URL: <http://www.math.ncsu.edu/~kaltofen/bibliography/10/HKZ10.pdf>.
- Jeffrey, David, editor. *ISSAC 2008*, 2008. ACM Press. ISBN 978-1-59593-904-3.
- Kai, Hiroshi and Sekigawa, Hiroshi, editors. *SNC’09 Proc. 2009 Internat. Workshop on Symbolic-Numeric Comput.*, 2009. ACM Press. ISBN 978-1-60558-664-9.
- Kaltofen, E. Computer algebra algorithms. In Traub, J. F., editor, *Annual Review in Computer Science*, volume 2, pages 91–118. Annual Reviews Inc., Palo Alto, California, 1987. URL: [http://www.math.ncsu.edu/~kaltofen/bibliography/87/Ka87\\_annrev.pdf](http://www.math.ncsu.edu/~kaltofen/bibliography/87/Ka87_annrev.pdf).
- Kaltofen, E. Analysis of Coppersmith’s block Wiedemann algorithm for the parallel solution of sparse linear systems. *Math. Comput.*, 64(210):777–806, 1995. URL: [http://www.math.ncsu.edu/~kaltofen/bibliography/95/Ka95\\_mathcomp.pdf](http://www.math.ncsu.edu/~kaltofen/bibliography/95/Ka95_mathcomp.pdf).



- Kaltofen, Erich and Lee, Wen-shin. Early termination in sparse interpolation algorithms. *J. Symbolic Comput.*, 36(3–4):365–400, 2003. Special issue Internat. Symp. Symbolic Algebraic Comput. (ISSAC 2002). Guest editors: M. Giusti & L. M. Pardo. URL: <http://www.math.ncsu.edu/~kaltofen/bibliography/03/KL03.pdf>.
- Kaltofen, Erich, Li, Bin, Yang, Zhengfeng, and Zhi, Lihong. Exact certification of global optimality of approximate factorizations via rationalizing sums-of-squares with floating point scalars. In [Jeffrey 2008], pages 155–163, 2008. URL: <http://www.math.ncsu.edu/~kaltofen/bibliography/08/KLYZ08.pdf>.
- Kaltofen, E. and Saunders, B. D. On Wiedemann’s method of solving sparse linear systems. In Mattson, H. F., Mora, T., and Rao, T. R. N., editors, *Proc. AAECC-9*, volume 539 of *Lect. Notes Comput. Sci.*, pages 29–38, Heidelberg, Germany, 1991. Springer Verlag. URL: <http://www.math.ncsu.edu/~kaltofen/bibliography/91/KaSa91.pdf>.
- Kaltofen, Erich, Yang, Zhengfeng, and Zhi, Lihong. On probabilistic analysis of randomization in hybrid symbolic-numeric algorithms. In [Vershelde and Watt 2007], pages 11–17, 2007. URL: <http://www.math.ncsu.edu/~kaltofen/bibliography/07/KYZ07.pdf>.
- Kaltofen, Erich, Yang, Zhengfeng, and Zhi, Lihong. A proof of the Monotone Column Permanent (MCP) Conjecture for dimension 4 via sums-of-squares of rational functions. In [Kai and Sekigawa 2009], pages 65–69, 2009. URL: <http://www.math.ncsu.edu/~kaltofen/bibliography/09/KYZ09.pdf>.
- Kaltofen, Erich and Yuhasz, George. On the matrix Berlekamp-Massey algorithm. *ACM Trans. Algorithms*, 9(4), September 2013. URL: <http://www.math.ncsu.edu/~kaltofen/bibliography/06/KaYu06.pdf>.
- Kaltofen, Erich L., Li, Bin, Yang, Zhengfeng, and Zhi, Lihong. Exact certification in global polynomial optimization via sums-of-squares of rational functions with rational coefficients. *J. Symbolic Comput.*, 47(1):1–15, January 2012. In memory of Wenda Wu (1929–2009). URL: <http://www.math.ncsu.edu/~kaltofen/bibliography/09/KLYZ09.pdf>.
- Kaltofen, Erich L. and Nehring, Michael. Supersparse black box rational function interpolation. In Leykin, Anton, editor, *Proc. 2011 Internat. Symp. Symbolic Algebraic Comput. ISSAC 2011*, pages 177–185, New York, N. Y., June 2011. Association for Computing Machinery. ISBN 978-1-4503-0675-1. URL: <http://www.math.ncsu.edu/~kaltofen/bibliography/11/KaNe11.pdf>.
- Klep, Igor and Schweighofer, Markus. Sums of Hermitian squares and the BMV conjecture. *J. Statistical Physics*, 133:739–760, 2008.
- Kunkle, Daniel and Cooperman, Gene. Harnessing parallel disks to solve Rubik’s cube. *J. Symbolic Comput.*, 44(7):872–890, 2009. URL <http://www.ccs.neu.edu/home/gene/papers/jsc09.pdf>.
- Lenstra, A. K., Lenstra, Jr., H. W., and Lovász, L. Factoring polynomials with rational coefficients. *Math. Ann.*, 261:515–534, 1982.

- Marshall, Murray. *Positive Polynomials and Sums of Squares*. American Math. Soc., 2008. 187 pp.
- May, John P., editor. *ISSAC 2009 Proc. 2009 Internat. Symp. Symbolic Algebraic Comput.*, 2009. ACM. ISBN 978-1-60558-609-0.
- Novocin, Andrew, Stehlé, Damien, and Villard, Gilles. An LLL-reduction algorithm with quasi-linear time complexity: extended abstract. In *Proc. 43rd Annual ACM Symp. Theory Comput.*, pages 403–412, New York, N.Y., 2011. ACM.
- Peyrl, Helfried and Parrilo, Pablo A. Computing sum of squares decompositions with rational coefficients. *Theoretical Comput. Sci.*, 409:269–281, 2008.
- Püschel, Markus, Moura, José M. F., Johnson, Jeremy, Padua, David, Veloso, Manuela, Singer, Bryan, Xiong, Jianxin, Franchetti, Franz, Gacic, Aca, Voronenko, Yevgen, Chen, Kang, Johnson, Robert W., and Rizzolo, Nicholas. SPIRAL: Code generation for DSP transforms. *Proceedings of the IEEE*, 93(2):232–275, June 2005. special issue on “Program Generation, Optimization, and Adaptation”, URL [http://spiral.ece.cmu.edu:8080/pub-spiral/pubfile/paper\\_1.pdf](http://spiral.ece.cmu.edu:8080/pub-spiral/pubfile/paper_1.pdf).
- Rosenkranz, Markus and Regensburger, Georg. Solving and factoring boundary problems for linear ordinary differential equations in differential algebras. *J. Symbolic Comput.*, 43(8): 515–544, 2008. ISSN 0747-7171.
- Safey El Din, Mohab. Computing the global optimum of a multivariate polynomial over the reals. In [Jeffrey 2008], 2008.
- Sendra, J. R., Winkler, F., and Pérez-Díaz, S. *Rational Algebraic Curves A Computer Algebra Approach*, volume 22 of *Algorithms and Computation in Mathematics*. Springer Verlag, Heidelberg, Germany, 2007. ISBN ISSN 1431-1550, ISBN 978-3-540-73724-7.
- van der Put, M. and Singer, M. F. *Galois Theory of Linear Differential Equations*, volume 328 of *Grundlehren der mathematischen Wissenschaften*. Springer Verlag, 2003. URL <http://www4.ncsu.edu/~singer/papers/dbook.ps>.
- Verschelde, Jan and Watt, Stephen M., editors. *SNC’07 Proc. 2007 Internat. Workshop on Symbolic-Numeric Comput.*, 2007. ACM Press. ISBN 978-1-59593-744-5.
- Villard, G. Further analysis of Coppersmith’s block Wiedemann algorithm for the solution of sparse linear systems. In Küchlin, W., editor, *ISSAC 97 Proc. 1997 Internat. Symp. Symbolic Algebraic Comput.*, pages 32–39, New York, N. Y., 1997. ACM Press. ISBN 0-89791-875-4.
- Wang, Dongming and Zhi, Lihong, editors. *Symbolic-Numeric Computation*. Trends in Mathematics. Birkhäuser Verlag, Basel, Switzerland, 2007. ISBN 978-3-7643-7983-4.
- Watt, Stephen M., editor. *Proc. 2010 Internat. Symp. Symbolic Algebraic Comput. ISSAC 2010*, 2010. Association for Computing Machinery. ISBN 978-1-4503-0150-3.

- Wiedemann, D. Solving sparse linear equations over finite fields. *IEEE Trans. Inf. Theory*, IT-32:54–62, 1986.
- Yuan, Quan and van Hoeij, Mark. Finding all Bessel type solutions for linear differential equations with rational function coefficients. In [Watt 2010], page to appear, July 2010.
- Zippel, R. Interpolating polynomials from their values. *J. Symbolic Comput.*, 9(3):375–403, 1990.