Encyclopedia of Applied and Computational Mathematics, Björn Enquist, ed., Mathematics of Computer Science, Discrete Mathematics, Johan Håstad, field ed., Springer, 2015, pp. 227–233.

Title: The Complexity of Computational Problems in Exact Linear Algebra

Name: Erich L. Kaltofen[1], Arne Storjohann[2]

Affil./Addr. 1: North Carolina State University
Department of Mathematics
Raleigh, NC 27695-8205, USA
Phone: (919) 515-8785
E-mail: kaltofen@math.ncsu.edu

Affil./Addr. 2: University of Waterloo
David R. Cheriton School of Computer Science
Waterloo, Ontario, Canada N2L 3G1
Phone: (519) 888-4567x36361
E-mail: astorjoh@uwaterloo.ca

The Complexity of Computational Problems in Exact Linear Algebra

Synonyms

Linear algebra algorithms, efficiency of linear algebra algorithms, bit complexity of linear algebra algorithms

Definition

Computational problems in exact linear algebra include computing an exact solution of a system of linear equations with exact scalars, which can be exact rational numbers, integers modulo a prime number, or algebraic extensions of those represented by their residues modulo a minimum polynomial. Classical linear algebra problems are computing for a matrix its rank, determinant, characteristic and minimal polynomial, and rational canonical form (= Frobenius normal form). For matrices with integer and polynomial entries one computes the Hermite and Smith normal forms. If a rational matrix is symmetric, one determines if the matrix is definite.

Algorithms For Dense Matrices

The building block of efficient algorithms for dense linear algebra is matrix multiplication. Because the complexity of this problem remains an open question, the running times of algorithms are stated in terms of a parameter ω such that two $n \times n$ matrices over a ring can be multiplied together in $O(n^\omega)$ ring operations. The standard algorithm has $\omega = 3$, and the best known estimates by [Coppersmith and Winograd 1990] allow $\omega \approx 2.376$. Most practical implementations use Strassen-Winograd's algorithm which has $\omega \approx 2.807$.

The complexity of many linear algebra problems over a field are linked to that of matrix multiplication. The following shows Winograd's 1970 reduction of multiplication to inversion:

$$\begin{bmatrix} I_n & A & \\ & I_n & B \\ & & I_n \end{bmatrix}^{-1} = \begin{bmatrix} I_n & -A & AB \\ & I_n & -B \\ & & I_n \end{bmatrix}.$$

More strikingly, techniques by Baur and Strassen from 1982 and Strassen from 1973 give a reduction of matrix multiplication to determinant in the arithmetic circuit model.

Echelon Forms over a Field

The main tool for solving linear algebra problems over a field K, including determinant and inverse for a square and nonsingular matrix, and nullspace bases for a matrix of arbitrary shape and rank, is transformation to echelon form. For an input matrix $A \in \mathsf{K}^{n\times m}$, the classical formulation of this problem asks as output a nonsingular $U \in \mathsf{K}^{n\times n}$ together with $H = UA$ in (row) echelon form — nonzero rows in H precede zero rows and the first nonzero entry in each nonzero row (a pivot entry) is to the right of the pivot entries in previous rows. The number of nonzero rows of H is the rank r of A and the set of column indices containing the pivot entries is the rank profile of A. There are many variations, including the Gauss–Jordan canonical form which has pivot entries in H equal to 1 and entries above pivots zeroed, and the LSP decomposition of Ibarra, Moran and Hui which expresses A as a product of matrices of special shape. By employing a divide and conquer approach to recursively reduce to matrix multiplication, transformation to echelon form costs $O(nmr^{\omega-2})$ arithmetic operations from K. [Dumas et al. 2008] present highly optimized algorithms and implementations for computing echelon forms and related matrix decompositions, for a variety of finite fields.

Transformation to echelon form uses $O(nmr^{\omega-2})$ field operations to solve a linear system: given a target vector b either produce a solution v such that $Av = b$ or determine that the system is inconsistent. Mulders and Storjohann [ISSAC 2000] give an algorithm for linear solving that uses $O((n+m)r^2)$ field operations, which is $o(nm)$ when $r \in o(\sqrt{\min(n,m)})$. For computing the rank, the use of essentially quadratic preconditioners (see below) to achieve generic rank profile gives a Monte Carlo randomized algorithm that uses $(nm + r^\omega)^{1+o(1)}$ field operations, which is $(nm)^{1+o(1)}$ when $r \in O((nm)^{1/\omega})$.

Frobenius Form over a Field

For an $n \times n$ matrix A over K, the block diagonal Frobenius form $F = TAT^{-1} = \text{Diag}(C_{f_1}, C_{f_2}, \ldots, C_{f_l})$ is a canonical form for the set of matrices similar to A. Each diagonal block C_{f_i} is the companion matrix of a monic $f_i \in \mathsf{K}[x]$ and f_i divides f_{i+1} for all $1 \leq i \leq l-1$. The minimal polynomial of A is f_l and the characteristic polynomial $c^A(x) \stackrel{\text{def}}{=} \det(xI - A)$ is equal to the product $f_1 f_2 \cdots f_l$ — of which the constant coefficient is the determinant of A. The problem of computing the Frobenius form and invariants have received a lot of attention. Making use of Keller-Gehrig's 1985 algorithm for the characteristic polynomial, Giesbrecht in 1993 gave a randomized reduction to matrix multiplication, for sufficiently large fields, for computing F together with a transformation matrix T such that $F = TAT^{-1}$. In 2000 Eberly gave, for the same problem, a Las Vegas algorithm with running time $O(n^\omega \log n)$ that works for any field, and Storjohann an $O(n^\omega (\log n)(\text{loglog}\, n))$ deterministic algorithm. Most recently, [Pernet and Storjohann 2007] show that if cardinality$(\mathsf{K}) \geq 2n^2$, the form itself can be computed in a Las Vegas fashion using an expected number of $O(n^\omega)$ field operations, matching the lower bound for this problem.

Division-Free Algorithms

Consider an $n \times n$ input matrix A over an abstract commutative ring R, that is, when no divisions are possible. Although the characteristic polynomial $c^A(x) \in \mathsf{R}[x]$ is well defined over R, the fastest known algorithms mentioned above to compute it use divisions, and directly applying Strassen's 1973 removal of divisions technique adds a factor of n to their cost. [Kaltofen and Villard 2004] give a division free algorithm for the characteristic polynomial with running time $O(n^{2.697263})$. The same cost estimate holds for division-free computation of the adjoint of A. An open problem is to obtain a division-free algorithm to compute $c^A(x)$ using $n^{\omega+o(1)}$ operations.

Fast Bit Complexity

For linear algebra problems over integer matrices, the sizes (numbers of digits) of integers involved in the computation and the answer affect the running time of the algorithms used. For example, the determinant of an $A \in \mathbb{Z}^{n\times n}$ can have size at most $(n \log \|A\|)^{1+o(1)}$, where $\|A\|$ denotes the largest entry in absolute value. Classical methods, such as working modulo a basis of primes and reconstructing using Chinese remaindering, require $(n^{\omega+1} \log \|A\|)^{1+o(1)}$ bit operations to compute $\det A$.

Many problems on integer matrices, such as diophantine system solving and determining the structure of finitely presented abelian groups, are solved by transforming an input matrix to Hermite and Smith canonical form. The Hermite form $H = UA$ is in echelon form and the Smith form $S = VAW = \mathrm{Diag}(s_1, s_2, \ldots, s_r, 0, \ldots, 0)$ is diagonal with s_{i-1} dividing $s_i \neq 0$ for all $1 < i \leq r$. The transformation matrices U, V and W are invertible over $\mathbb{Z}$ (that is, they have determinant ± 1). In 1983, Domich showed how to control intermediate expression swell during the computation by working modulo the determinant, and fast matrix multiplication is taken advantage of by Hafner and McCurley in 1989. Since the transformation matrices to achieve the forms are not unique, care must be taken to produce ones with good bounds on the size of entries. Storjohann's dissertation from 2000 gives a survey of work up to that date, and describes deterministic algorithms that take as input an $A \in \mathbb{Z}^{n\times m}$ of rank r, and compute the the Hermite and Smith form, together with transformation matrices, in time $(nmr^{\omega-1} \log \|A\|)^{1+o(1)}$. Note that if $n = m = r$ this cost estimate becomes $(n^{\omega+1} \log \|A\|)^{1+o(1)}$, with the exponent of n in this bit complexity estimate 1 higher than that for the corresponding algebraic cost. Much recent effort has focused on reducing or eliminating, for a variety of problems, the commonly occurring $+1$ in the exponent of bit complexity estimates. One of the initial efforts in this direction is Eberly, Giesbrecht and Villard's Monte Carlo algorithm from 2000 for computing the

determinant and Smith form of a nonsingular matrix in $(n^{2+\omega/2}(\log \|A\|)^{3/2})^{1+o(1)}$ bit operations.

Linear System Solving

Already in 1982, Dixon showed that the algebraic analogue of numerical iterative refinement, combined with rational number reconstruction, can be used to compute $A^{-1}b$ for a nonsingular $A \in \mathbb{Z}^{n\times n}$ and $b \in \mathbb{Z}^{n\times 1}$ with a cost that is softly cubic in n instead of quartic. Storjohann's high-order lifting technique incorporates matrix multiplication to compute $A^{-1}b$ in an expected number of $(n^{\omega} \log \|A\|)^{1+o(1)}$ operations.

The general case of the linear solving problem, when A has arbitrary shape and rank, is more subtle: a solution vector may not exist, and if solution vectors do exist they may not be unique and have fractional entries. The classical approach of transforming A to echelon form, or to Hermite/Smith form in case a diophantine solution is desired, solves the problem completely but currently has cost $(nmr^{\omega-1}d)^{1+o(1)}$ bit operations. Giesbrecht in 1997 introduced the technique of combining random rational solutions to produce a diophantine solution, should one exist, and in the next year Giesbrecht, Lobo and Saunders show how to compute certificates of inconsistency. Based on these ideas, Mulders and Storjohann in 2004 give Las Vegas algorithms with cost $(nmr^{\omega-2} \log \|A\|)^{1+o(1)}$ for either proving inconsistency, or producing a solution vector v that has a minimal size denominator among all solution vectors, in particular a diophantine solution when one exists.

Integer Matrix Invariants and Certificates

Let $A \in \mathbb{Z}^{n\times n}$. Extensions of the division-free algorithms of Kaltofen and Villard mentioned above compute the Frobenius form (and hence characteristic polynomial) and Smith form in a randomized Monte Carlo fashion in $n^{2.697263}(\log \|A\|)^{1+o(1)}$ bit op-

erations. A main open problem is to compute the characteristic polynomial of A in $(n^\omega \log \|A\|)^{1+o(1)}$ bit operations.

[Storjohann 2005] combines fast nonsingular rational system solving with other ideas to get an algorithm for computing $\det A$ in an expected number of $(n^\omega(\log \|A\|)^{1+o(1)}$ bit operations (Las Vegas). This is currently the fastest algorithm for the determinant. For computing the rank r in case A is singular, the fastest known Monte Carlo algorithm uses essentially quadratic preconditioning and projection modulo a random prime and completes in $(n^2 \log \|A\| + r^\omega)^{1+o(1)}$ bit operations. The fastest known Las Vegas algorithm for rank has expected running time $(n^2 r^{\omega-2} \log \|A\|)^{1+o(1)}$.

Freivald's famous 1979 quadratic time certificate for matrix product assays the equation $BC - D = 0$ in a Monte Carlo fashion by projecting with a random vector. [Kaltofen et al. 2011] use the Las Vegas algorithms mentioned above to obtain randomized algorithms that certify the rank and determinant of A in a Monte Carlo fashion in $(n^2(\log \|A\|))^{1+o(1)}$ bit operations.

Lattice Basis Reduction

The seminal 1982 paper of Arjen Lenstra, Hendrik Lenstra Jr and László Lovász introduced the famous LLL lattice basis reduction algorithm: given an $A \in \mathbb{Z}^{n\times m}$, the LLL algorithm finds a basis for the $\mathbb{Z}$-lattice generated by the rows of A that consists of nearly orthogonal (and thus relatively short) vectors. Originally applied to problems in computer algebra and algebraic number theory, many more applications have been discovered in areas such as cryptography and communications theory. The current state of the art for LLL-type reduction is an algorithm with cost that is softly linear in $\log \|A\|$ [Novocin et al. 2011].

Claus-Peter Schnorr has shown there exists a continuum of algorithms between those solving the shortest vector problem (shown to be NP-hard by Atjai) and finding

an approximation of the shortest vector as produced by LLL. A survey of recent results is given by [Hanrot et al. 2011].

Matrices with Polynomial Entries

Let $\deg A$ denote the maximal degree of entries in an $A \in \mathsf{K}[x]^{n\times m}$. Because of the natural analogy between $\mathsf{K}[x]$ and $\mathbb{Z}$, many of the algorithms supporting complexity results stated above for integer matrices have analogues over $\mathsf{K}[x]$ that support the same complexity bound, but now counting field operations from K and with $\deg A$ replacing $\log \|A\|$: these include in particular nonsingular system solving and determinant in expected time $(n^\omega \deg A)^{1+o(1)}$ field operations.

A nearly optimal Las Vegas randomized algorithm to compute A^{-1} is given by [Storjohann 2011]. As an application, given any scalar matrix $B \in \mathsf{K}^{n\times n}$, the sequence $I, B, B^2, \ldots, B^n \in \mathsf{K}^{n\times n}$ of matrix powers can be computed using an expected number of $(n^3)^{1+o(1)}$ field operations from K by computing $(xI_n - B)^{-1}$. Currently, the analogous result for integer matrix inversion has only been established for well-conditioned input.

A concept with many applications for polynomial matrices that has no natural analogue for integer matrices is minimal approximant bases. Given a matrix power series $G \in \mathsf{K}[[x]]^{n\times m}$ with $m \leq n$, and an approximation order d, these are nonsingular $n \times n$ polynomial matrices M (with minimal row degrees) such that $MG \equiv 0 \bmod x^d$. Beckermann and Labahn's algorithm from 1994 is adapted to exploit matrix multiplication in [Giorgi et al. 2003], reducing the cost of computing M to $(n^\omega \deg A)^{1+o(1)}$ field operations from K. As an application, they give a Las Vegas algorithm algorithm with same cost bound for computing, for a nonsingular $A \in \mathsf{K}[x]^{n\times n}$, a row reduced form: a matrix R and unimodular matrix U such that $A = UR$, with degrees of rows of R minimal among all matrix equivalent to A under unimodular pre-multiplication.

Parallel Algorithms

There are several theoretical and practical models of parallel computation over an abstract field. One is the arithmetic synchronous circuit model, where the parallel time is the depth of the acyclic computation digraph with the arithmetic operations performed at the bounded fan-in, bounded fan-out vertices. Equality tests can be allowed in a decision tree model. It was shown by Csanky in 1976 that many linear algebra problems, such as the determinant of an $n \times n$ integer matrix, can be computed on an arithmetic circuit of depth $O((\log n)^2)$. In [Kaltofen and Pan 1992], the total size of the circuit was reduced to $n^{\omega+o(1)}$ via Wiedemann's method (see below), thus obtaining a processor efficient solution. Those circuits are valid over any field and have random elements as inputs so that with high probability division by zero is avoided in all vertices that perform divisions. The construction fails for the characteristic polynomial. A reference is the book [Bini and Pan 1995].

Algorithms For Sparse Matrices

Exact linear algebra computations with sparse matrices, i.e., matrices that have many entries equal 0, originated from the matrix problems that arise in integer factoring algorithms based on Pollards quadratic sieve: there the entries are integers modulo 2, and initially sparsity preserving echelon form methods, which today are known as "super-LU," were deployed. Douglas Wiedemann's 1986 IEEE Transactions on Information Theory paper on Krylov space-based iterative algorithms in exact arithmetic has had a far-reaching impact, as it provides a complexity model which we describe next.

Sparse and Black Box Algorithms

Black box matrices are represented by a procedure that performs a matrix-times-vector product. One seeks algorithms that call the procedure $O(n)$ times (for a, say, $n \times$

n matrix) and with an additional $n^{2+o(1)}$ scalar operations in the field of entries to complete their tasks. In addition, one restricts to $O(n)$ additional intermediate storage locations for auxiliary scalars, excluding the storage that the black box uses. Thus, one gets an essentially-quadratic time, linear space solution for matrices whose black box procedures run in $n^{1+o(1)}$ arithmetic operations and whose scalar arithmetic suffers no expression swell, such as matrices with $n^{1+o(1)}$ non-zero entries over a finite field.

Wiedemann's approach has at least two drawbacks. One is that the use of "bi-directional Lanczos-like" iteration that over a finite field can lead to self-orthogonal vectors ("unlucky projections"). A second is that in normal situations the exact dimension of the Krylov subspace can be as large as $\Omega(n)$, thus requiring many black box calls. In contrast, numerical methods use the fact that the approximate dimension of the Krylov subspace is small and a good approximation to the solution is found early. The problem of unlucky projections is dealt with by preconditioning the black box matrix, and today we have a wealth of fast preconditioners. The dimension of the Krylov subspace is reduced by projecting simultaneously with blocks of β vectors. The latter can reduce the number of required black box calls to $(1 + \epsilon)n$, or $O(n/\beta)$ if parallelism is utilized.

The algorithms for black box matrices known today can compute a solution to a linear system, a random nullspace vector, the minimal polynomial, determinant and rank of a black box matrix within the stated complexity measures. All algorithms are probabilistic, and rank is Monte Carlo. Blocking can also improve the success probabilities. Rank certification in the black box model constitutes a major open problem. The best algorithm known for the characteristic polynomial, by Gilles Villard, has higher asymptotic complexity.

Black box algorithms apply to matrices over the rational numbers, either by Chinese remaindering the solution and subsequently using rational vector recovery,

or by other modular techniques. For example, the Smith normal form of a black box integer matrix is computed by Giesbrecht in 2001 from the characteristic polynomial after preconditioning. The length of intermediate integers in sparse rational solvers has been reduced by Eberly et al. [ISSAC 2007] by blocking, like was done for the dense algorithms by Kaltofen and Villard discussed above.

The literature on black box exact linear algebra computation is quite large. A collection of preconditioners with reference to most literature before 2002 is [Chen et al. 2002]. The most recent analysis of blocking over finite fields is [Eberly 2010].

Finally, we mention the open source LinBox library [http://www.linalg.org] which provides C++ efficient implementations for many black box algorithms that can accommodate the scalar arithmetic in a plug-and-play generic way.

Structured Matrix Algorithms

Black box algorithms apply generically to structured matrices, such as Toeplitz and Vandermonde matrices. However, the resulting complexity is quadratic in their dimensions, and essentially-linear complexity can in some cases be achieved.

Toeplitz and Hankel Matrices

In 1981 Brent, Gustavson, Yun utilized the connection between Toeplitz/Hankel solving and the extended Euclidean algorithm for deterministically computing solutions to non-singular Toeplitz systems in $O(n(\log n)^2 \operatorname{loglog} n)$ arithmetic operations. Their solution borrows ideas from Israel Gohberg's displacement rank representations, but allows for singularity in the arising submatrices. The jumps in the Padé table, which correspond to sequences of zero discrepancies in the Berlekamp/Massey algorithm, amount to drops in the polynomial remainder degrees that are overcome by embedded polynomial divisions

in the half GCD algorithm. Essentially-linear complexity is achieved by polynomial arithmetic that utilizes the fast Fourier transform.

Matrices of Small Displacement Rank

The notion of displacement rank generalizes the notion of Toeplitz matrix. A matrix of displacement rank α has a succinct representation, as a sum of LU products, where both L and U are Toeplitz. The notion is closed under inverses, meaning that the inverse has displacement rank $\leq \alpha + 2$. Gohberg's and Koltracht's $O(\alpha n^2)$ solvers could be improved in 1980 independently by Bitmead and Anderson and by Morf to arithmetic complexity $\alpha^2 n^{1+o(1)}$ by Strassen-like divide and conquer techniques. The algorithms does not allow for singular submatrices. In 1994, Kaltofen introduced randomized preconditioning with constant displacement rank to guarantee non-singularity with high probability, on which all subsequent exact algorithms over abstract fields rely. An arithmetic complexity of $\alpha^{\omega-1} n^{1+o(1)}$ is achieved in [Bostan et al. 2007].

The theory of displacement rank applies to other types of matrices, such as Cauchy and Vandermonde matrices. For block Toeplitz and Hankel matrices, the Euclidean algorithm on matrix polynomials can be used. A scalarization to blocks of polynomials is Beckermann's and Labahn's notion of σ-bases, which can be computed deterministically asymptotically fast and by which structured linear problems can be solved.

Note added in Proof February 18, 2015: Since writing the survey in September 2011, major achievements have been made: Alexander M. Davie, Andrew J. Stothers and independently Virginia Vassilevska Williams have shown that the matrix multiplication exponent $\omega < 2.3736898$ and François Le Gall that $\omega < 2.3728639$. Jean-Guillaume Dumas and Erich L. Kaltofen have given essentially optimal certificates for the characteristic polynomial of a dense integer matrix, and as a corollary for positive

semidefiniteness, and for the rank of a sparse matrix with integer entries. Le Gall's paper and Dumas-Kaltofen's paper appear in the Proceedings of ISSAC 2014, ACM. Wei Zhou, George Labahn and Arne Storjohann have given a deterministic algorithm for inverting a polynomial matrix with scalars in a field whose running time is $(n^3 s)^{1+o(1)}$ field operations, where n is the dimension of the matrix and s is the average column degree [J. Complexity, to appear].

Cross-references

References

Bini, D. and Pan, V. *Numerical and Algebraic Computations with Matrices and Polynomials Volume 1 Fundamental Algorithms*. Birkhäuser Boston, Inc., 1995. Lecture Notes in Theor. Comput. Sci., R. V. Book, series editor.

Bostan, Alin, Jeannerod, Claude-Pierre, and Schost, Éric. Solving Toeplitz- and Vandermonde-like linear systems with large displacement rank. In Brown [2007], pages 33–40, 2007.

Brown, Christopher W., editor. *ISSAC 2007 Proc. 2007 Internat. Symp. Symbolic Algebraic Comput.*, 2007. ACM Press. ISBN 978-1-59593-743-8.

Chen, L., Eberly, W., Kaltofen, E., Saunders, B. D., Turner, W. J., and Villard, G. Efficient matrix preconditioners for black box linear algebra. *Linear Algebra and Applications*, 343–344: 119–146, 2002. Special issue on *Structured and Infinite Systems of Linear Equations*, edited by P. Dewilde, V. Olshevsky and A. H. Sayed. URL: http://www.math.ncsu.edu/~kaltofen/bibliography/02/CEKSTV02.pdf.

Coppersmith, D. and Winograd, S. Matrix multiplication via arithmetic progressions. *J. Symbolic Comput.*, 9(3):251–280, 1990. Special issue on complexity theory.

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.

Eberly, Wayne. Yet another block Lanczos algorithm: How to simplify the computation and reduce reliance on preconditioners in the small field case. In Watt, Stephen M., editor, *Proc. 2010*

Internat. Symp. Symbolic Algebraic Comput. ISSAC 2010, page 289–296, New York, N. Y., July 2010. Association for Computing Machinery. ISBN 978-1-4503-0150-3.

Giorgi, Pascal, Jeannerod, Claude-Pierre, and Villard, Gilles. On the complexity of polynomial matrix computations. In Sendra, J. R., editor, *Proc. 2003 Internat. Symp. Symbolic Algebraic Comput. (ISSAC'03)*, pages 135–142, New York, N. Y., 2003. ACM Press. ISBN 1-58113-641-2.

Hanrot, Guillaume, Pujol, Xavier, and Stehlé, Damien. Algorithms for the shortest and closest lattice vector problems. In Chee, Yeow Meng, Guo, Zhenbo, Ling, San, Shao, Fengjing, Tang, Yuansheng, Wang, Huaxiong, and Xing, Chaoping, editors, *IWCC*, volume 6639 of *Lecture Notes in Computer Science*, pages 159–190. Springer, 2011. ISBN 978-3-642-20900-0.

Kaltofen, E. and Pan, V. Processor-efficient parallel solution of linear systems II: the positive characteristic and singular cases. In *Proc. 33rd Annual Symp. Foundations of Comp. Sci.*, pages 714–723, Los Alamitos, California, 1992. IEEE Computer Society Press. URL: http://www.math.ncsu.edu/~kaltofen/bibliography/92/KaPa92.pdf.

Kaltofen, Erich and Villard, Gilles. On the complexity of computing determinants. *Computational Complexity*, 13(3-4):91–130, 2004. URL: http://www.math.ncsu.edu/~kaltofen/bibliography/04/KaVi04_2697263.pdf.

Kaltofen, Erich L., Nehring, Michael, and Saunders, B. David. Quadratic-time certificates in linear algebra. In Leykin, Anton, editor, *Proc. 2011 Internat. Symp. Symbolic Algebraic Comput. ISSAC 2011*, pages 171–176, 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/KNS11.pdf.

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.

Pernet, Clément and Storjohann, Arne. Faster algorithms for the characteristic polynomial. In Brown [2007], pages 307–314, 2007.

Storjohann, Arne. The shifted number system for fast linear algebra on integer matrices. *J. Complexity*, 21(5):609–650, 2005.

Storjohann, Arne. On the complexity of inverting integer and polynomial matrices. *Computational Complexity*, 2015. to appear.