2005

TR-2005008: Toeplitz and Hankel Meet Hensel and Newton Modulo a Power of Two

Victor Y. Pan

Brian Murphy

Rhys E. Rosholt

Xinmao Wang

Toeplitz and Hankel Meet Hensel and Newton Modulo a Power of Two *†

Victor Y. Pan, Brian Murphy, Rhys E. Rosholt
Department of Mathematics and Computer Science
Lehman College of CUNY, Bronx, NY 10468, USA
vpan@lehman.cuny.edu
bmurphy@lehman.cuny.edu
rosholt@lehman.cuny.edu
and
Xinmao Wang
Ph.D. Program in Mathematics
Graduate School of CUNY, New York, NY 10016, USA
xwang2@gc.cuny.edu

June 16, 2005

Abstract

We extend Hensel lifting for solving general and structured linear systems of equations to the rings of integers modulo nonprimes, e.g. modulo a power of two. This enables significant saving of word operations. We elaborate upon this approach in the case of Toeplitz linear systems. In this case, we initialize lifting with the MBA superfast algorithm, estimate that the overall bit operation (Boolean) cost of the solution is optimal up to roughly a logarithmic factor, and prove that the degeneration is unlikely even where the basic prime is fixed but the input matrix is random. We also comment on the extension of our algorithm to some other fundamental computations with (possibly singular) general and structured matrices and univariate polynomials as well as to the computation of the sign and the value of the determinant of an integer matrix.

2000 Math. Subject Classification: 68W30, 68W20, 65F05, 68Q25

*Some results of this paper have been presented at the Annual International Conference on Application of Computer Algebra, Volos, Greece, June 2002; ACM International Symposium on Sympolic and Algebraic Computation, Lille, France, July 2002; and the 5th Annual Conference on Computer Algebra in Scientific Computing, Yalta, Crimea, Ukraine, September 2002.

†Supported by NSF Grant CCR 9732206 and PSC CUNY Awards 65393–0034 and 66437–0035

Key Words: Solving linear systems, Hensel's lifting, Computations modulo a power of two, Toeplitz matrices.

1 Introduction

1.1 Lifting for structured linear systems (some motivation)

Hensel lifting in the field of integers modulo a prime is a well and long known tool for the solution of general linear systems of equations with integer coefficients (see Moenck and Carter 1979 [MC79], Dixon 1982 [D82]). Lifting computations are performed with a lower precision, which gives them advantage over numerical approach where the systems are ill-conditioned. This is frequently the case where the input coefficent matrix is structured, e.g., this is the case for every positive definite Hankel matrix (see Tyrtyshnikov 1994 [T94]).

For structured input matrices the power of lifting is the greatest. In particular, lifting supports a nearly optimal randomized upper bound on the overall bit operation complexity of the solution of a Toeplitz or Hankel linear system of n equations with n unknowns. For integer input values in $n^{O(1)}$, this upper bound is within a roughly logarithmic factor from the information lower bound of $n^2 \log n$ bit operations (see Theorem 9.1 and Table 9.1). Lifting also enables low cost solution of a block Hankel linear system, and this immediately implies a substantial improvement of a recent advanced and widely acclaimed algorithm for computing the determinant of an integer matrix (see Section 12.3).

The lifting algorithm remains highly effective for various other classes of structured linear systems such as Hankel, Toeplitz/Hankel-like, block and polynomial Hankel/Toeplitz, and banded systems. More precisely, the algorithm is effective as long as the integer, rational, or polynomial input matrix and its preconditioned inverse (or the inverse of its largest nonsingular submatrix) can be multiplied by a vector fast in the ring of integers modulo a fixed integer m. We list some important extensions and applications in Section 12, and we refer the reader to the bibliography on structured matrices in Kailath and Sayed 1999 [KS99] and Pan 2000, 2001, and 2004 [P00], [P01], [P04].

1.2 Lifting in the rings of integers modulo nonprimes

Our main technical contribution, however, is the extension of lifting and its initialization to the ring of integers modulo a prime power, e.g., modulo 2^w. This extension seems to be completely missing from the literature, although it enables binary computations and the saving of lifting steps and word operations, whereas in terms of the bit operation complexity, our solution cost remains nearly optimal for a Toeplitz input. To yield saving, we begin lifting with M^{-1} modulo m where $\log_2 m$ is slightly less than the length λ of a computer word. We call this policy *saturated initialization*. In practice 2^λ is huge, and

it is inconvenient to deal with primes m that large. The most attractive choice seems to be $m = 2^w$, a power of two, allowing binary computations.

Lifting modulo nonprimes, however, leads to some technical challenges, particularly regarding degeneration and initialization. To meet these challenges, we first introduce a simple concept of factor nonsingularity of an integer matrix; then we modify the algorithm to perform it modulo a prime power. For the initialization modulo a prime power we adjust the MBA divide-and-conquer algorithm by Morf 1974 and 1980 and Bitmead and Anderson [M74], [M80], and [BA80] and also propose two alternative algorithms.

We prove that our initialization rarely degenerates even where we fix the basic prime but choose the input matrix at random; furthermore, we propose some heuristic recipes to counter the unlikely degeneration if it still occurs.

1.3 Some extensions and applications

Our algorithm can be extended to some fundamental computations with possibly singular Toeplitz matrices such as computing their determinants, their ranks, and the vectors in their null spaces. Further applications include computing the gcd, lcm and resultant of a univariate polynomial, as well as Padé approximations and interpolating rational functions. We still nearly optimize the word and bit operation complexity in all these computations.

Our lifting algorithm is effective for any sparse and/or structured integer or rational matrix if its preconditioned inverse can be multiplied by a vector fast. Our analysis can be readily extended from the case of a Toeplitz input matrix except for the estimates for the probability of degeneration in the reduction modulo a power of a fixed prime. We only have such estimates in the cases of general and Toeplitz random input matrices, and we present some experimental results for tridiagonal and five-diagonal matrices.

1.4 Organization of our paper

We state some basic definitions and auxiliary results in the next section. We cover Hensel's and Newton's lifting algorithms for linear equations and matrix inversion in the rings of integers modulo an integer q in Sections 3 and 4. We initialize lifting in Section 5 and Appendix A. We recall the techniques for the recovery of the rational solution from its truncated q-adic extension in Sections 6, 7, and 8. We estimate the computational complexity of our algorithm in Section 9. In Section 10, we study the degeneration problem and include the results of our experiments. In Section 11, we demonstrate our algorithms with some simple examples. In Section 12 we comment on the extensions of our study. Section 10.4 is due to the fourth author, the implementation of the algorithms to the second and third authors, and all other parts of the work and the paper to the first author.

Acknowledgements. Our thanks go to Mark Giesbrecht and Arne Storjohann for the (p)reprints of the papers Eberly et al. 2000 [EGV00], Mulders and

Storjohann 2004 [MS04], and Storjohann 2003 [S03], and to Richard Isaac for suggesting a format for the statistical tests reported in Section 10.4.

2 Definitions and basic facts

Hereafter $z \bmod q$ for $z, q \in \mathbb{Z}$, $q > 1$ is a unique integer z_q such that q divides $z - z_q$ and $0 \le z_q < q$. We write log for $\log_2$, $\mathbb{Z}$ for the ring of integers, $\mathbb{Z}_q$ for the ring of integers modulo an integer q, and $\mathbb{Q}$ for the field of rational numbers.

2.1 General matrices

Definition 2.1. *$M = (m_{i,j})_{i,j=1}^{k,l} \in \mathbb{R}^{n\times n}$ is a $k \times l$ matrix with entries $m_{i,j}$ in a ring $\mathbb{R}$. $\mathbf{v} = (v_i)_{i=1}^{k}$ is a vector. I is the identity matrix of a proper size. I_l is the $l \times l$ identity matrix. M^T is the transpose of M; $M^{(h)}$ is the $h \times h$* leading principal*, that is, northwestern, submatrix of M. A matrix M of rank r has* generic rank profile *if its submatrices $M^{(k)}$ are nonsingular for $k = 1, \ldots, r$, that is, up to the rank size $r \times r$. M is* strongly nonsingular *if it is nonsingular and has generic rank profile.*

Definition 2.2. *$\det M$ and $\operatorname{adj} M$ are the determinant and the adjoint of a matrix M, respectively. ($\operatorname{adj} M = M^{-1} \det M$ if M is nonsingular.)*

Definition 2.3. *$|M| = ||M||_1 = \max_j \sum_i |m_{i,j}|$ is the column norm of a matrix $M = (m_{i,j})_{i,j}$; $|\mathbf{v}| = \sum_i |v_i|$ is the ℓ_1-norm of a vector $\mathbf{v} = (v_i)_i$; $\alpha(M) = \max_{i,j} |m_{i,j}|$, $\beta(\mathbf{v}) = \max_i |v_i|$.*

Definition 2.4. *$v_S \le 2n^2 - n$ and i_S are the minimum numbers of arithmetic operations sufficient to multiply a given $n \times n$ matrix S by a vector and to invert it, respectively.*

Definition 2.5. *$d_k = d_k(M)$ is the k-th determinantal divisor of a matrix $M \in \mathbb{Z}^{n\times n}$ for $k = 1, \ldots, n$, that is, the greatest common divisor (gcd) of all its $k \times k$ minors (subdeterminants). $s_0 = d_0 = 1$, $s_k = s_k(M) = d_k/d_{k-1}$ are the k-th* Smith invariant factors *of M for $k = 1, \ldots, n$.*

Hadamard's estimate below is known to be sharp in the worst case but is an over-estimate on the average according to Abbott et al. 1999 [ABM99].

Fact 2.1. *$|\det M| \le \prod_j \left(\Sigma_i m_{i,j}^2\right)^{1/2} \le (\alpha(M)\sqrt{n})^n$, $|\det M| \le |M|^n$, $|\operatorname{adj} M| \le n\alpha(\operatorname{adj} M)$, and so $|\operatorname{adj} M| \le (\alpha(M)\sqrt{n-1})^{n-1} n$, $|\operatorname{adj} M| \le n|M|^{n-1}$ for an $n \times n$ matrix $M = (m_{i,j})_{i,j}$.*

It is easily deduced (see Newman 1972 [N72]) that $s_1, \ldots, s_n \in \mathbb{Z}$ and $|\det M| = s_1 \cdots s_n$. Therefore

$$s_n \le |\det M| \le |M|^n. \tag{2.1}$$

Hereafter $\mathbf{b} \ne \mathbf{0}$, $n > 2$, $|M| > 2$, and so $\log n > 1$, $\log |M| > 1$.

Definition 2.6. *For two integers $q > 0$ and $s > 1$, a matrix M in $\mathbb{Z}_{qs}^{n\times n}$ is* factor-q nonsingular *modulo qs if there exists a matrix Q in $\mathbb{Z}_{s}^{n\times n}$ such that*

$$MQ \bmod (qs) = qI \tag{2.2}$$

or equivalently if there exists the s-adic expansion $qM^{-1} = q\sum_{i=0}^{\infty} Q_i s^i$, $Q_0 = Q$, $Q_i \in \mathbb{Z}_{s}^{n\times n}$ for all i.

2.2 Polynomial and integer multiplication

Let $m(n)$ field operations be required to multiply two polynomials of degree $n-1$ or less. We have $m(n) \geq 2n-1$ (this is an information lower bound), $m(n) = O(n \log n)$ over the fields or rings that support FFT, and

$$m(n) \leq c_{class} n^2, \quad m(n) \leq c_k n^{\log 3}, \quad m(n) \leq (c_{ck} n \log n) \log\log n \tag{2.3}$$

over any field, ring with unity, or algebra. Here and hereafter log stands for $\log_2$ unless we specify otherwise, so that $\log 3 = 1.58496\ldots$; c_{class}, c_k, and c_{ck} are three constants, $0 < c_{class} < c_k < c_{ck}$, and the above bounds are supported by the classical, Karatsuba's, and Cantor and Kaltofen's algorithms; the practical choice among them depends on the degree n (see Bernstein 2003 [B03], [GG03]).

Each arithmetic operation over integers modulo q, represented with the d-bit precision for $d = \lceil \log q \rceil$, can be performed by using $O(\mu(d))$ bit operations, where $\mu(d)$ denotes the bit operation complexity of multiplication of two integers modulo q, $\mu(d) \geq 2d-2$ (an information lower bound),

$$\mu(d) \leq C_{class} d^2, \quad \mu(d) \leq C_k d^{\log 3}, \quad \mu(d) \leq (C_{ss} d \log d) \log\log d, \tag{2.4}$$

C_{class}, C_k, and C_{ss} are three constants, $0 < C_{class} < C_k < C_{ss}$, and the above bounds are supported by the classical algorithm and those of Karutsuba 1963 and Schönhage and Strassen 1971 (see Knuth 1998 [K98], [B03], [GG03]).

2.3 Toeplitz and Hankel matrices

Definition 2.7. *$T = (t_{i,j})_{i,j}$ is a Toeplitz matrix if $t_{i,j} = t_{i+1,j+1}$ for every pair of its entries $t_{i,j}$ and $t_{i+1,j+1}$. $Z(\mathbf{v})$ is the lower triangular Toeplitz matrix defined by its first column $\mathbf{v}$. $H = (h_{i,j})_{i,j}$ is a Hankel matrix if $h_{i,j} = h_{i-1,j+1}$ for every pair of its entries $h_{i,j}$ and $h_{i-1,j+1}$. $J = (j_{g,h})_{g,h=0}^{n-1,n-1}$ is the unit Hankel (reflection) matrix where $j_{g,n-1-g} = 1$ for $g = 0,\ldots,n-1$, $j_{g,h} = 0$ for $h+g \neq n-1$. ($J\ (v_i)_{i=0}^{n-1} = (v_{n-i-1})_{i=0}^{n-1}$, $J^2 = I$.)*

TJ and JT are Hankel matrices if T is a Toeplitz matrix, and HJ and JH are Toeplitz matrices if H is a Hankel matrix. Therefore, Toeplitz and Hankel linear systems are immediately reduced to each other, and *we just study the Toeplitz case.*

Theorem 2.1. *Multiplication of an $n \times n$ Toeplitz matrix T by a vector is a subproblem of multiplication of two polynomials of degrees $2n-2$ and $n-1$ whose coefficients are given by the entries of the input matrix and vector, respectively, that is, $v_T \le m(3n-3)$ for v_T and $m(n)$ in Sections 2.1 and 2.2. If the Toeplitz matrix T is triangular and $m = n$, then both of these polynomials have degree $n-1$, that is, in this case $v_T \le m(2n-2)$.*

Proof. See, e.g., [P01, pages 27–28]. □

The next theorem of Heinig 1979 [H79] (cf. Heinig and Rost 1984 [HR84]) extends the Gohberg–Semencul formula of 1972.

Theorem 2.2. *Let $T = (t_{i,j})_{i,j=0}^{n-1}$ be a nonsingular Toeplitz matrix, let t_{-n} be any scalar (e.g., $t_{-n} = 0$), and write $t_{i-j} = t_{i,j}$ for $i, j = 0, \dots, n-1$; $p_n = -1$, $\mathbf{t} = (t_{i-n})_{i=0}^{n-1}$, $\mathbf{p} = (p_i)_{i=0}^{n-1} = T^{-1}\mathbf{t}$, $\mathbf{q} = (p_{n-i})_{i=0}^{n-1}$, $\mathbf{v} = T^{-1}\mathbf{e}_1$, $\mathbf{e_1}^T = (1, 0, \dots, 0)^T$, $\mathbf{u} = ZJ\mathbf{v}$. Then $T^{-1} = Z(\mathbf{p})Z^T(\mathbf{u}) - Z(\mathbf{v})Z^T(\mathbf{q})$.*

Hereafter the $n \times 2$ matrix $(\mathbf{v}, \mathbf{p})$ for the above vectors $\mathbf{v}$ and $\mathbf{p} = \mathbf{p}(T, t_{-n})$ is called a *generator* for T^{-1}. The next theorem is a corollary of Theorems 2.1 and 2.2.

Theorem 2.3. *$v_{T^{-1}} \le 4m(2n-2) + n$ for v_S in Definition 2.4, $m(n)$ in (2.3), and a nonsingular Toeplitz matrix T provided the matrix T^{-1} is given with its generator, that is, the vectors $\mathbf{v}$ and $\mathbf{p}$ in Theorem 2.2.*

2.4 Rational number reconstruction

Definition 2.8. *$\mathrm{ord}_q(m)$, the order of q in m, is the maximal integer l such that q^l divides m. $\nu(y)$ is the numerator, and $\delta(y)$ is the denominator in the ratio $y = \nu(y)/\delta(y)$ of two coprime integers $\nu(y)$ and $\delta(y)$.*

Modular rational roundoff is the recovery of a rational number x/y from three integers k, l, and $r = (x/y) \bmod l$ provided x and y are coprime unless $r = 0$, l and y are coprime, $|x| < k \le l$, and $0 < y \le l/k$. $\rho(\log l)$ is the bit-operation complexity of this recovery. Clearly, we may write $x = r, y = 1$ if $k > |r|$. The pair (x, y) is unique under the additional assumption that $2|x| < k$ [GG03].

Theorem 2.4. *We have*

$$\rho(d) \le cd^2, \rho(d) \le C\mu(d) \log d \tag{2.5}$$

for $\mu(d)$ in (2.4) and two positive constants C and c, $c < C$.

Proof. To support the theorem, it is sufficient to apply the algorithms in any of the papers by Pan and Wang 2002 [PW02], 2003 [WP03], 2004 [PW04], or Monahan 2004 [M04]. □

3 The generalized Hensel's lifting for linear systems

Let us generalize Hensel's lifting algorithm in [MC79], [D82] to perform it in the rings $\mathbb{Z}_{qs}$ for two integers $q > 0$ and $s > 1$. Actually we only need the case where they are the powers of two or another fixed integer $m > 1$, possibly a prime. We assume that M is a factor-q nonsingular matrix in $\mathbb{Z}_{qs}^{n\times n}$ (see Definition 2.6) and that we are given the matrix Q satisfying (2.2). In fact it is sufficient if this matrix is given with a block box for its multiplication by a vector or with its generator $(\mathbf{v},\mathbf{p})$ in the case of a Toeplitz matrix T (see Theorem 2.2). Then we compute the first h terms in the s-adic expansion of the vector $qM^{-1}\mathbf{b} = q\sum_{i=0}^{\infty}\mathbf{u}^{(i)}s^i$, $\mathbf{u}^{(i)} \in \mathbb{Z}_s^n$, $i = 0, 1, \ldots$.

Algorithm 3.1. *The generalized lifting (see Examples 11.1–11.3).*

INPUT: *a matrix $M \in \mathbb{Z}^{n\times n}$, a vector $\mathbf{b} \in \mathbb{Z}^n$, three positive integers h, q, and s, and a matrix $Q = (qM^{-1}) \bmod (qs)$ satisfying (2.2).*

OUTPUT: *the vector $\mathbf{x}^{(h)} \in \mathbb{Z}^n$ such that $\mathbf{x}^{(h)} = (qM^{-1}\mathbf{b}) \bmod (qs^h)$, that is, such that $M\mathbf{x}^{(h)} = (q\mathbf{b}) \bmod (qs^h)$.*

INITIALIZATION: $\mathbf{r}^{(0)} = \mathbf{b}$.

COMPUTATIONS: *for $i = 0, 1, \ldots, h-1$, compute the vectors*

$$\mathbf{u}^{(i)} = Q\mathbf{r}^{(i)} \bmod (qs), \quad \mathbf{r}^{(i+1)} = (q\mathbf{r}^{(i)} - M\mathbf{u}^{(i)})/(qs).$$

Output the vector $\mathbf{x}^{(h)} = \sum_{i=0}^{h-1}\mathbf{u}^{(i)}s^i$.

The following theorem shows correctness of the algorithm (see part b) and bounds the precision of its computations. For $q = 1$ and a prime s, Algorithm 3.1 and the theorem have appeared in [D82].

Theorem 3.1. *For $\mathbf{r}^{(i)}$ and $\mathbf{x}^{(h)}$ in Algorithm 3.1, we have*

a) $\mathbf{r}^{(i)} \in \mathbb{Z}^n$ for all i;

b) $M\mathbf{x}^{(h)} = q\mathbf{b} \bmod (qs^h)$;

c) all components $r_j^{(i)}$ of all vectors $\mathbf{r}^{(i)} = (r_j^{(i)})_j$ satisfy the bounds $|r_j^{(i)}| \le |b_j|/s^i + \alpha n\frac{qs-1}{q}\sum_{k=1}^{i}s^{-k} < \beta/s^i + \alpha n(qs-1)/(qs-q) < \gamma$ where $M = (m_{i,j})_{i,j=1}^n$, $\mathbf{b} = (b_j)_{j=1}^n$,

$$\beta = \beta(\mathbf{b}) = \max_j |b_j|, \quad \alpha = \alpha(M) = \max_{i,j} |m_{i,j}|, \quad \gamma = 2\alpha n + \beta. \tag{3.1}$$

Proof.

a) $(q\mathbf{r}^{(i)} - M\mathbf{u}^{(i)}) \bmod (qs) = (qI - MQ)\mathbf{r}^{(i)} \bmod (qs)$, and the claim follows because $MQ = qI \bmod (qs)$.

b) $M\mathbf{x}^{(h)} = \sum_{i=0}^{h-1} M\mathbf{u}^{(i)}s^i = \sum_{i=0}^{h-1}(q\mathbf{r}^{(i)} - qs\mathbf{r}^{(i+1)})s^i = q\mathbf{b} - qs^h\mathbf{r}^{(h)} = q\mathbf{b} \bmod (qs^h)$.

c) By definition, all components $u_j^{(i)}$ of all vectors $\mathbf{u}^{(i)}$ satisfy $|u_j^{(i)}| \le qs-1$, and so $qs|r_j^{(i+1)}| \le q|r_j^{(i)}| + \alpha n \max_k |u_k^{(i)}| \le q|r_j^{(i)}| + (qs-1)\alpha n$. The claim now follows by induction on i.

□

Clearly the arithmetic computational cost of a lifting step is in $v_M + v_Q + O(n)$. Here is a coarse bound on the presision of computing.

Lemma 3.1. *Algorithm 3.1 operates with integers in the range* $[-2^{d_1}, 2^{d_1})$ *where*

$$d_1 = \lceil \log(2qs\gamma) \rceil \tag{3.2}$$

for γ *in (3.1).*

Proof. The lemma follows from Theorem 3.1 a) and c) since the vectors $\mathbf{u}^{(i)}$ are computed in $\mathbb{Z}_{qs}$. □

The bit precision of computing in Algorithm 3.1 is at most d_1 and is only $\lceil \log(qs) \rceil$ at the stages of computing the vectors $\mathbf{u}^{(i)}$. Therefore, each lifting step requires $(v_M + O(n))\mu(d_1) + v_Q\mu(\log(qs))$ bit operations. If λ is the length of a computer word and $d_1 \le \lambda$, then all arithmetic operations in the algorithm are word operations. To save lifting steps and word operations, we apply the policy of *saturated initialization*, that is, choose q and s to maximize $d_1 \le \lambda$.

4 Matrix inversion via the generalized Newton's lifting

Let us extend the generalized Hensel's lifting to matrix inversion. Recursively compute the matrices

$$X_0 = qM^{-1} \bmod (qs), X_i = X_{i-1}(2qI - MX_{i-1}) \bmod (qs^{2^i}), \tag{4.1}$$

$i = 1, 2, \ldots, h$. Assuming the reduction modulo qs^{2^i}, deduce that $qI - MX_i = (qI - MX_{i-1})^2 = (qI - MX_0)^{2^i} = 0$, that is, $qI = MX_i \bmod (qs^{2^i})$. For $q = 1$, this is Newton's lifting for matrix inversion [MC79] whose i-th step squares the residual matrix $I - MX_{i-1}$, thus implying guadratic convergence of the approximations X_i to M^{-1}.

Every Newton's step (4.1) is essentially reduced to performing $n \times n$ matrix multiplication twice. For Toeplitz matrices, however, we simplify the iteration. Indeed, for a Toeplitz matrix $T = (t_{k-j})_{k,j} = M/q$, $\mathbf{e}_1 = (1, 0, \ldots, 0)^T$ and

$\mathbf{t}$ defined in Theorem 2.2, the inverses $X_i = qM^{-1} \bmod (qs^{2^i})$ in $\mathbb{Z}_{qs^{2^i}}$, $i = 0, 1, \ldots$, can be represented with their $n \times 2$ generators $X_i(\mathbf{e}_1, \mathbf{t}) = (X_i\mathbf{e}_1, X_i\mathbf{t})$. In this case our iteration (4.1) takes the following form,

$$\begin{aligned} X_0(\mathbf{e}_1, \mathbf{t}) &= qM^{-1}(\mathbf{e}_1, \mathbf{t}) \bmod (qs), \\ X_i(\mathbf{e}_1, \mathbf{t}) &= X_{i-1}(2qI - MX_{i-1})(\mathbf{e}_1, \mathbf{t}) \bmod (qs^{2^i}), \end{aligned} \tag{4.2}$$

$i = 1, 2, \ldots$. Its every step is reduced essentially to the multiplication of the matrix M by the $n \times 2$ matrix $X_{i-1}(\mathbf{e}_1, \mathbf{t})$ and of the matrix X_{i-1} by the resulting $n \times 2$ matrix. This is still $O(m(n))$ arithmetic operations (see Theorems 2.1 and 2.2), which is much less than the complexity of $n \times n$ matrix multiplication.

For $q = 1$ the iteration processes (4.1) and (4.2) are similar to Newton's iteration for numerical inversion of a matrix M [P01, Chapter 6], which takes the forms

$$X_i = X_{i-1}(2I - MX_{i-1}), \quad i = 1, 2, \ldots, \tag{4.3}$$

for a general matrix M and

$$X_i(\mathbf{e}_1, \mathbf{t}) = X_{i-1}(2I - MX_{i-1})(\mathbf{e}_1, \mathbf{t}), i = 1, 2, \ldots, \tag{4.4}$$

for a Toeplitz matrix M.

Remark 4.1. *Striking similarity can be also observed between the algebraic Algorithm 3.1 for Hensel's lifting and the celebrated algorithm for iterative improvement of numerical solution to a linear system of equations. (Compare Golub and Van Loan 1996 [GL96, Section 3.5.3], Skeel 1980 [S80], Higham 1996 [H96], and our Algorithm 3.1.) This similarity was exploited in Pan 1992 [P92a] and Emiris et al. 1998 [EPY98] to improve the solution algorithm. The improvement relies on extending modular arithmetic to binary rational numbers to avoid computations with the vanishing leading bits of the residuals.*

In h steps, the generalized Newton lifting achieves as much as the generalized Hensel's in 2^h steps, but the precision of computing is roughly doubled in every Newton's step, reaching the level of $(2^h \log s + \log q)$-bit precision in h steps. As is easily verified, the overall asymptotic bit operation cost estimate slightly increases versus the generalized Hensel's lifting even where $\mu(d) = O((d \log d) \log\log d)$, but the Newton's approach enables some saving of the word operations in the case where initially the ratio $\lceil \log(2qs\gamma) \rceil / \lambda$ is small. In this case we may apply the generalized Newton's steps as long as the precision of computing stays within a bound w which we fix a little below the length of a computer word. If the precision at the next step would exceed w, we switch to the generalized Hensel's lifting, thus yielding its saturated initialization.

5 Initialization of the generalized lifting

In this section we study the following problem.

Problem 5.1. *Initialization of the generalized lifting.*

INPUT: *a nonsingular matrix $M \in \mathbb{Z}^{n\times n}$, a prime p, and two positive integers λ, the length of a computer word, and w such that*

$$\lceil \log(2p^w\gamma) \rceil = \lambda \tag{5.1}$$

for γ in (3.1).

OUTPUT: *either FAILURE or two integers $q > 0$ and $s > 1$, both the powers of p and such that $\log_p(qs) = w$, and a matrix Q satisfying (2.2).*

Due to Lemma 3.1, the subsequent lifting steps require a precision within λ. With a smaller value of w, we would have needed some extra lifting steps and word operations.

5.1 Solution in the case of general input matrix

Gaussian elimination with column pivoting enables us to solve Problem 5.1 for a general matrix M.

Algorithm 5.1. *Initialization via Gaussian elimination.*

For a fixed prime p, compute w in (5.1), and apply Gaussian elimination to invert the matrix M. Perform the computations in $\mathbb{Z}_{p^w}$. Apply column pivoting to avoid divisions by the multiples of p, that is, at every elimination step interchange the rows to minimize the order of p in the pivot entry (cf. Definition 2.8 on the order of p). If at some step the order exceeds w, output FAILURE and stop. Otherwise continue the elimination until M is diagonalized. Then choose $v = \mathrm{ord}_p(s_n)$ for $s_n = s_n(M)$ denoting the Smith leading invariant factor of the matrix M in Definition 2.5, that is, choose v equal to the maximal order in p among all pivot entries (which are the diagonal entries of the output diagonal matrix). Finally fix the positive integer $u = w - v$ and compute the matrix Q satisfying (2.2) for $q = p^v$ and $s = p^u$.

The algorithm does not fail if and only if

$$w \geq \mathrm{ord}_p(s_n(M)) \tag{5.2}$$

for w in (5.1). Due to (2.1) it is sufficient if $w \geq n \log_p |M|$. In Section 10 we show the failure probability for this computation in the case of a random choice of p or M.

Algorithm 5.1 uses the order of n^3 arithmetic operations performed with the precision of $\lceil \log(qs) \rceil$. If (5.1) holds, they are word operations. This cost bound is the same as or is dominated at the lifting stage. (We ignore the chances for theoretical asymptotic acceleration and minor practical speed up based on fast matrix multiplication, on which we refer the reader to Kaporin 2004 [K04], Dumas et al. 2004 [DGP04], and the bibliography therein.)

5.2 Solution in the case of a Toeplitz input matrix

In the case of a Toeplitz input matrix, we replace Algorithm 5.1 by adapting the MBA divide-and-conquer algorithm, which requires only $O(m(n)\log n)$ arithmetic operations.

Recall that we cannot perform division in $\mathbb{Z}_{p^w}$ if p divides the divisor. In the MBA algorithm, we can avoid such divisions if and only if the input matrix M is strongly nonsingular in the field $\mathbb{Z}_p$ (see [P01, Chapter 5]). Let us list some relevant results on strong nonsingularity from Section 10 and [P01]:

- A random $n \times n$ integer Toeplitz matrix is likely to be strongly nonsingular modulo any fixed prime $p >> n$ (Theorem 10.4).
- If M is not strongly nonsingular in $\mathbb{Z}_p$ for a random prime p sampled from a large range, then M is unlikely to be strongly nonsingular even in $\mathbb{Z}$ (in virtue of Theorem 10.1).
- The matrices M^TM and MM^T are strongly nonsingular in $\mathbb{Z}$ if M is nonsingular in $\mathbb{Z}$ [P01] and are likely to remain strongly nonsingular in $\mathbb{Z}_p$ for a larger random prime p (in virtue of Theorem 10.1).

Having the matrices M^TM or MM^T inverted, we obtain

$$M^{-1} = (M^TM)^{-1}M^T = M^T(MM^T)^{-1}.$$

M^TM and MM^T are in the class of $n \times n$ Toeplitz-like matrices [P01]. Such matrices generalize $n \times n$ Toeplitz matrices. They can be represented in a compact form with their *displacement generators* made up of $O(n)$ parameters and, like the matrix T^{-1} in Theorem 2.2, can be multiplied by vectors fast. These properties also hold for the inverses of nonsingular Toeplitz-like matrices.

If a Toeplitz-like matrix is strongly nonsingular in $\mathbb{Z}_p$, we adapt the MBA algorithm to compute in $\mathbb{Z}_p$ a linear number of parameters which define the displacement generator of its inverse; this takes $O(m(n)\log n)$ field operations. More precisely, as long as the input matrix is Toeplitz-like and strongly nonsingular, we just complement the original MBA algorithm in [M80], [BA80] with the low cost deterministic algorithm in [P01, Section 4.6.2] (traced back to Pan 1992 [P92, Proposition A.6]), which compresses the dispacement generators in $\mathbb{Z}_p$ wherever they involve extraneous parameters. This compression algorithm also works in the rings $\mathbb{Z}_{p^w}$ if we direct its pivoting to avoid degeneration in $\mathbb{Z}_{p^w}$ rather than in $\mathbb{Z}_p$.

We only need to apply the MBA algorithm to compute the $2n$ entries of a generator $(\mathbf{v}, \mathbf{p})$ for the matrix Q (cf. Theorem 2.2) rather than its n^2 entries, but as by-product the algorithm also computes the $O(n\log n)$ parameters defining recursive triangular factorization of a strongly nonsingular input matrix, whose determinant is readily available from the factorization. For a detailed description and analysis of this algorithm and further bibliography, see [M74], [M80], [BA80], [P01, Chapter 5], and [P04].

In the Appendix we propose two alternative initialization algorithms which work in $\mathbb{Z}_{qs}$ for $q = p^v, s = p^u$ (where p can be small) provided M is a factor-q nonsingular Toeplitz matrix. If a Toeplitz input matrix M is strongly nonsingular in $\mathbb{Z}_p$, we may also initialize lifting by applying the Levinson–Durbin algorithm (see Levinson 1947 [L47] and Durbin 1959 [D59]). It uses $O(n^2)$ operations in $\mathbb{Z}_p$; this is more than in the MBA algorithm but still translates into a bit cost bound dominated at the lifting stage.

6 Extension to the singular case and applications to polynomial computations

For a singular input matrix M of a rank r, we seek the inverse of its nonsingular $r \times r$ submatrix M_r. Algorithm 5.1 can be immediately extended to compute $r = \operatorname{rank} M$, M_r, and the matrix Q_r such that $Q_r M_r \mod (qs) = qI_r$ for appropriate q and s. Then by applying the generalized lifting and the customary techniques in [BP94, page 110], we may compute a solution to a consistent linear system $M\mathbf{x} = \mathbf{b}$ and vectors from the null space of M. The overall asymptotic bounds on the computational cost do not increase.

We may respectively extend the MBA algorithm as well provided the input matrix M is a Toeplitz matrix of rank r and has generic rank profile. We can ensure the generic rank profile property with a high probability by shifting to the Toeplitz-like matrix UML where U^T and L are random unit lower triangular Toeplitz matrices (see Kaltofen and Saunders 1991 [KS91]).

The transition from M to UML involves random matrices U and L but does not increase the overall asymptotic complexity bounds; these bounds can be applied also to the verification that the $r \times r$ leading principal submatrix is nonsingular, which implies that rank $M \geq r$. Our cost bounds for the rank computation for a Toeplitz matrix M are, however, Monte Carlo randomized because they do not cover the verification that rank $M = r$.

The latter Monte Carlo complexity estimates can be extended to computing the gcds, lcms, Padé approximations, and rational interpolation functions where the input is given by univariate polynomials with integer coefficients [BGY80], [BP94], [P96], [P01].

7 Deterministic recovery of the rational solution

7.1 The rational number reconstruction of every component

To recover the unique vector $\mathbf{x} = qM^{-1}\mathbf{b}$ from the output vector $\mathbf{x}^{(h)}$ of Algorithm 3.1, we need to have a sufficiently large h. Let us estimate how large.

Theorem 7.1. *Let $\mathbf{x} = qM^{-1}\mathbf{b}$ denote a unique solution to the linear system*

$M\mathbf{x} = q\mathbf{b}$. *Assume* $\rho(d)$ *in (2.5),*

$$d = \lceil \log(2(\alpha\sqrt{n})^{2n-1}n\beta q)\rceil = O(n\log\gamma + \log q), \tag{7.1}$$

and α, β *and* γ *in (3.1). Suppose that in*

$$h = 1 + \lfloor \log_s(2(\alpha\sqrt{n})^{2n-1}n\beta)\rfloor \tag{7.2}$$

steps Algorithm 3.1 computes the vector

$$\mathbf{x}^{(h)} = \sum_{i=0}^{h-1} \mathbf{u}^{(i)}p^i = \mathbf{x} \bmod (qs^h).$$

Then it is sufficient to perform

$$B = n\rho(d) \tag{7.3}$$

bit operations to recover the vector $\mathbf{x}$ *from the vector* $\mathbf{x}^{(h)}$.

Proof. Suppose that the pairs of coprimes $\nu_j = \nu(x_j)$ and $\delta_j = \delta(x_j)$ define the rational components $x_j = \nu_j/\delta_j$ of the vector $\mathbf{x} = (x_j)_j = qM^{-1}\mathbf{b}$. Fix the smallest integer $k > 2(\alpha\sqrt{n-1})^{n-1}n\beta q$. Note that $s^h > 2(\alpha\sqrt{n})^{2n-1}n\beta$ for h in (7.2). Deduce from Fact 2.1 that $l = qs^h > 2|\nu_j|\delta_j$ and $2|\nu_j| < k \le qs^h$. Then according to Section 2.4 every component x_j can be uniquely recovered from $qx_j \bmod (qs^h)$. Now Theorem 2.4 supports the claimed bit complexity bound for this recovery. □

7.2 The recovery with lifting the recursive triangular factorization

Suppose that we have initialized the lifting of a Toeplitz-like matrix M with the MBA algorithm. As a by-product it computes recursive triangular factorization of $M \bmod (qs)$, which immediately defines $(\det M) \bmod (qs)$. By combining the MBA algorithm with recursive application of h lifting steps of Algorithm 3.1, we can compute $(\det M) \bmod (qs^h)$ and the integer vector $\mathbf{y} = (\det M)\mathbf{x} = (\det M)\mathbf{x} \bmod (qs^h)$ where $M\mathbf{x} = q\mathbf{b}$ [P00], [P04]. For the integer entries y_i of $\mathbf{y}$ we have $|y_i| \le nq|M|^{n-1}|\mathbf{b}|$, and so if $qs^h > 2|M|^n, qs^h > 2nq|M|^{n-1}|\mathbf{b}|$, then we may immediately reconstruct them and $\det M$ and then output $\mathbf{x} = \mathbf{y}/\det M$.

Since $\det(MM^T) = \det(M^TM) = (\det M)^2, \det(UML) = \det M$ where U and L are unit triangular matrices, the above technique covers also the case of preconditioned matrices M.

8 Randomized recovery of the rational solution

For the values $\mu(d)$ in $O(d^{\log 3})$ or $O((d\log d)\log\log d)$ and $\rho(d)$ bounded in (2.5), we may decrease the bit complexity bound in (7.3) by the factor of $\log d$

by using randomization. This is Las Vegas randomization, that is, we allow failure with a probability of at most ϵ for a fixed positive ϵ, but otherwise within the same computational cost bound we certify that the output is correct. We assume that $\log(1/\epsilon) = O(\log n)$.

The acceleration relies on two observations:

(a) The vector $\mathbf{y} = \delta\mathbf{x}$ is filled with integers provided

$$\delta = \mathrm{lcm}_j\, \delta(x_j), \quad 1 \le j \le n \tag{8.1}$$

(for $\delta(y)$ in Definition 2.8), that is, δ is the least common multiple of the denominators in all rational coordinates x_j of the solution $\mathbf{x} = (x_j)_j$ to the system $M\mathbf{x} = q\mathbf{b}$. Due to the integrality of the vector $\mathbf{y}$, its recovery from the vector $\mathbf{y} \bmod (qs^h)$ is immediate if $qs^h > 2\delta|\mathbf{x}| = 2|\mathbf{y}|$. Since $\delta \le s_n(M) \le |\det M| \le (\alpha(M)\sqrt{n})^n$ (see (2.1) and Fact 2.1), it is sufficient to use h of (7.2). Multiplication of the vector $\mathbf{x}$ by δ requires the order of $n\mu(d)$ bit operations, thus limiting the theoretical gain versus the estimate $B = n\rho(d)$ in (7.3). The practical gain can be significant, however, because the constant bounding the ratio $\rho(d)/(\mu(d)(\log d))$ from above can be quite large.

(b) Computation of the value δ can be accelerated with randomization because this value is likely to equal the least common multiple of the denominators in a smaller number K of random linear combinations $\mathbf{c}_k^T\mathbf{x}$ of the coordinates $x_1, \dots, x_n$, $k = 1, \dots, K$. According to the tests by Victor Shoup and Jean-Guillaume Dumas, one may typically use the denominators of some selected coordinates themselves, e.g., the first, the second, etc., instead of their random linear combinations.

The approach can be traced back to Pan 1988 [P88, Section 6]. Its recent studies include [ABM99], Cooperman et al. 1999 [CFG99], Eberly et al. 2000 [EGV00], and Mulders and Storjohann 2004 [MS04]. Let us specify and briefly analyze the generalized Hensel's lifting with randomized recovery.

Algorithm 8.1. *Randomized recovery of the rational solution.*

INPUT: *The same as in Algorithm 3.1 and in addition a positive $\varepsilon < 1$ and the vector $\mathbf{x}^{(h)} = (x_i^{(h)})_{i=1}^n = qM^{-1}\mathbf{b} \bmod (qs^h)$ for h in (7.2).*

OUTPUT: *FAILURE with a probability of at most ε or a positive integer δ and an integer vector $\mathbf{y}$ such that*

$$M\mathbf{y} = \delta q\mathbf{b}. \tag{8.2}$$

INITIALIZATION: *Compute*

$$K = 2\lceil \log(1/\varepsilon) \rceil, \tag{8.3}$$

$$\eta = 6 + 2n \log(n\alpha), \tag{8.4}$$

$$h = 1 + \lfloor \log_s(2n(\alpha\sqrt{n})^{2n-1}\eta\beta) \rfloor \tag{8.5}$$

for α and β in (3.1). Then sample K pseudo random vectors

$$\mathbf{c}_k = (c_{jk})_{j=1}^n \in \mathbb{Z}_\eta^n, \quad k = 1, \ldots, K. \tag{8.6}$$

COMPUTATIONS:

1. *Compute the K integers*

$$w_k = \mathbf{c}_k^T \mathbf{x}^{(h)} = \sum_{j=1}^n c_{jk} x_j^{(h)}, \quad k = 1, \ldots, K.$$

2. *Recover a unique set of the pairs of coprime integers ν_k and δ_k such that*

$$(\nu_k/\delta_k) \bmod (qs^h) = w_k, 1 \le 2\delta_k|\nu_k| \le qs^h, \quad 2|\nu_k| < qs^h, \quad k = 1, \ldots, K. \tag{8.7}$$

3. *Compute the least common multiple of the denominators*

$$\delta_{lcd} = \operatorname{lcm}_k \delta_k, \quad 1 \le k \le K. \tag{8.8}$$

4. *Compute the integer vector $\mathbf{y} = (y_j)_{j=1}^n$ such that $\mathbf{y} \bmod (qs^h) = \delta_{lcd}\mathbf{x}^{(h)}$ and $2|y_j| < qs^h$ for all j. If $M\mathbf{y} = q\delta_{lcd}\mathbf{b}$, output $\mathbf{y}$ and $\delta = \delta_{lcd}$; otherwise output FAILURE.*

Combining equations (8.4)–(8.6) and Fact 2.1 implies (8.7). Now, correctness of Algorithm 8.1 is implied by the following simple result.

Theorem 8.1. *δ_{lcd} in (8.8) divides δ in (8.1). Furthermore,*

$$\text{Probability}(\delta_{lcd} \ne \delta) \le \varepsilon.$$

Theorem 8.1 is deduced similarly to Theorem 2.1 in [EGV00] based on (8.3)–(8.8) and the following lemma.

Lemma 8.1. *For a prime p, integers K in (8.3), k such that $1 \le k \le K$, δ in (8.1), η in (8.4), and δ_k in (8.7), we have* $\text{Probability}(\text{ord}_p(\delta_k) < \text{ord}_p(\delta)) \le \max\{\frac{1}{p}, \frac{1}{\eta}\}$.

Proof. Let $l = \text{ord}_p(\delta) = \max_j \text{ord}_p(\delta(x_j))$ for $1 \le j \le n$. W.l.o.g., let $l = \text{ord}_p(\delta(x_1))$ and let c denote the first coordinate of the vector $\mathbf{c} = \mathbf{c}_k$. Then we have

$$\mathbf{c}^T \mathbf{x} = \frac{cu}{ap^l} - \frac{v}{p^h b} = \frac{cub - avp^{l-h}}{abp^l}$$

where $\mathbf{x} = M^{-1}\mathbf{b}$, $l \ge h$, and a, b, u, and v are four integers coprime with p. Clearly, $\text{ord}_p(\delta_k)$ for δ_k in (8.7) never exceeds l; it equals l if and only if $cub - avp^{l-h}$ is coprime with p. Since ub is coprime with p and since c is random, the probability bound follows. □

Let us estimate the bit complexity of performing Algorithm 8.1 in terms of $d = O(n \log \gamma + \log q)$ in (7.1), $\mu(d)$ in (2.4), $\rho(d)$ in (2.5), and K in (8.3). We need the following auxiliary result.

Lemma 8.2. *Let j and k be positive integer parameters, $j \to \infty$. Then $O(\mu(j)k)$ bit operations are sufficient to multiply two positive integers u and v such that $u < 2^j$ and $v < 2^{j+k}$.*

Proof. Represent v as $\sum_{i=0}^{k-1} v_i 2^{ij}, 0 \le v_i < 2^j$ for all i. Compute the products $w_i = uv_i$ for $i = 0, 1, \ldots, k-1$. This takes $O(\mu(j)k)$ bit operations. Now compute the sum $uv = \sum_{i=0}^{k-1} w_i 2^{ij}$. This takes $O(jk)$ bit operations. □

Algorithm 8.1 involves $O(Kn\mu(d))$ bit operations at Stage 1; $O(K\rho(d))$ at Stage 2; $O(K\mu(d)\log d)$ at Stage 3, and $O(n\mu(d))$, $O(n\mu(\log\beta)d/\log\beta)$, and $O(m(n)\mu(\log\gamma)d/\log\gamma)$ for computing the vectors $\delta_{lcd}\mathbf{x}^{(h)}$, $q\delta_{lcd}\mathbf{b}$, and $M\mathbf{y}$ at Stage 4, respectively. (The two latter bounds are deduced based on Lemma 8.2.) Summarizing, we obtain the following estimate.

Theorem 8.2. *Algorithm 8.1 generates nK random elements in $\mathbb{Z}_\eta$ for η in (8.4) and $K = 2\lceil \log(1/\epsilon) \rceil$ in (8.3). It either fails (this occurs with a probability of at most ϵ) or computes the solution $\mathbf{y}, \delta$ to the linear system (8.2). The algorithm involves*

$$B_1 = O(Kn\mu(d) + K\rho(d) + m(n)\mu(\log\gamma)d/\log\gamma)$$

bit operations for $d = O(n\log\gamma + \log q)$ in (7.1), $\rho(d)$ in (2.5), γ in (3.1), $m(n)$ in (2.3), and $\mu(d)$ in (2.4); it involves $o(B_1)$ bit operations for generating nK pseudo random elements in $\mathbb{Z}_\eta$.

9 Computational complexity estimates

Let us summarize the bit complexity estimates for our algorithms. We first state the detailed refined estimates in a theorem and then show them in a more observable coarser form in a table.

Theorem 9.1. *Let for a prime p and its powers q and s, a matrix $M \in \mathbb{Z}^{n\times n}$ be q-factor nonsingular modulo qs. Let $Q \in \mathbb{Z}_s^{n\times n}$ satisfy (2.2). Let $\mathbf{b} \in \mathbb{Z}^n$. Then we may compute the rational solution $\mathbf{x}$ to the linear system $M\mathbf{x} = \mathbf{b}$ by applying the algorithms in Sections 3 – 8 at the bit–operation cost within the following bounds:*

a) $O(n^3\mu(\log(qs)))$ *at the initialization stage for a general matrix M;*

b) $O((m(n)\log n)\mu(\log(qs)))$ *at the initialization stage for a Toeplitz matrix M;*

c) $(v_Q\mu(\log(qs)) + (v_M + O(n))\mu(\log(\gamma qs)))h$ *at the lifting stage;*

d) $n\rho(d) = O(n\mu(d)\log d), d = O(n\log\gamma + \log q)$ *for deterministic recovery of the* n *coordinates* x_i *of the rational solution* $\mathbf{x}$ *where all* x_i *are recovered independently of each other;*

e) $O((hm(n)\log n)\mu(\log(\gamma qs))) + n\mu(d)$ *for the deterministic recovery of the* n *coordinates* x_i *based on lifting the recursive triangular factorization of the (preconditioned) input matrix;*

f) $O(\log(\frac{1}{\varepsilon})(n\mu(d) + \rho(d) + m(n)\frac{\mu(\log\gamma)}{\log\gamma}d)$ *for the (Las Vegas) recovery which involves* $n\lceil\log\frac{1}{\varepsilon}\rceil\lceil\log(6 + 2n\log(n\alpha))\rceil$ *random bits and may fail with a probability of at most* $\varepsilon > 0$ *but otherwise is certified to be correct.*

Here $m(n), \mu(d)$, *and* $\rho(d)$ *are defined in (2.3)–(2.5),* γ *and* α *in (3.1),* $d = O(n\log\gamma + \log q)$ *in (7.1),* $h = O(n\log(\gamma n))$ *in (7.2) and (8.5)*
Futhermore, randomized Toeplitz preconditioning $M \to UML$ *in Section 6 enables the extension of the above asymptotic bit operation complexity bounds to the solution of a singular consistent system* $M\mathbf{x} = \mathbf{b}$ *and to the computation of a nonzero vector* $\mathbf{v}$ *from the null space* $\mathbb{N}(M) = \{\mathbf{v} : M\mathbf{v} = \mathbf{0}\}$ *for a singular matrix* M. *The bounds also cover the certification of the correctness of the solution.*

Remark 9.1. *(Cf. Bini and Pan 1994 [BP94].) The randomized complexity estimates of Theorem 9.1 also apply to computing the rank* r *of the matrix* M *and an* $r \times r$ *nonsingular submatrix of the matrix* UML *in Section 6 but in this case they are Monte Carlo estimates, that is, they do not cover the correctness certification of the solution, thus allowing undetected errors with a probability of at most* ϵ. *For a Toeplitz input matrix* M, *the asymptotic Las Vegas estimates for the bit complexity of computing a vector from the null space can be extended to Monte Carlo estimates for computing a shortest displacement generator of a matrix whose columns form a basis for the null space of* M.

Table 9.1 summarizes the bit complexity estimates in Theorem 9.1. To make the estimates more observable, we use the notation "$\tilde{O}$" (which means "O" up to the factors in $(\log\log n)^{O(1)}$) and the following simplifying assumptions,

$$\log_s\gamma = O(1), \quad \log(qs) = O(\log n), \quad \log(1/\epsilon) = O(\log n). \tag{9.1}$$

Here ϵ is the error probability in the randomized rational reconstruction of the output.

Our bound of $\tilde{O}(n^2\log^2 n)$ bit operations on the overall randomized complexity of the initialization, lifting and rational solution reconstruction is nearly optimal (assuming a Toeplitz input matrix M and equations (9.1)), because $n^2\log n$ bits are required to represent the n rational output values $x_1, \ldots, x_n$, and therefore at least as many bit operations are required to compute these values.

Let us also estimate the number of word operations in our algorithm assuming that every arithmetic operation in $\mathbb{Z}_t$ requires $\tilde{O}(\lceil\frac{t}{\lambda}\rceil)$ word operations and that $\log(\gamma qs) = O(\gamma)^v$. The latter assumptions imply a constant bound

Table 9.1: The bit complexity of lifting (for general and Toeplitz input matrices M) and rational reconstruction (deterministic and randomized), under (9.1), (2.3)–(2.5), and (3.1).

Initialization complexity B_{-1} (for a general matrix M)	$O(n^3\mu(\log n)) = \tilde{O}(n^3 \log n)$
Initialization complexity B_{-1} (for a Toeplitz matrix M)	$O((m(n) \log n)\mu(\log n)) = \tilde{O}(n \log^3 n)$
Lifting complexity B_0 (for a general matrix M)	$O(n^3\mu(\log n)) = \tilde{O}(n^3 \log n)$
Lifting complexity B_0 (for a Toeplitz matrix M)	$O(nm(n)\mu(\log n)) = \tilde{O}(n^2 \log^2 n)$
Reconstruction complexity B_1 (deterministic)	$O(n\rho(n \log n)) = \tilde{O}(n^2 \log^3 n)$
Reconstruction complexity B_1 (randomized)	$O(n\mu(n \log n) + \rho(n)(\log n)) = \tilde{O}(n^2 \log^2 n)$
Reconstruction complexity B_1 (with lifting the factorization)	$O(nm(n)(\log n)\mu(\log n)) = \tilde{O}(n^2 \log^3 n)$

on the word complexity of an operation in $\mathbb{Z}_t$ where $\log t = O(\log(\alpha qs))$. Then the bounds $(a) - (f)$ in Theorem 9.1 turn into the following word complexity estimates:

a) $\tilde{O}(n^3)$

b) $\tilde{O}(m(n) \log n)$

c) $\tilde{O}((v_Q + v_M)h)$

d) $\tilde{O}(n\rho(\frac{d}{\lambda}))$

e) $\tilde{O}(hm(n) \log n + n\mu(\frac{d}{\lambda}))$

f) $\tilde{O}(\log(\frac{1}{\varepsilon}(n\mu(\frac{d}{\lambda}) + \rho(\frac{d}{\lambda}) + m(n)))$

In part (f), we avoid using Lemma 8.2 and operate with longer numbers, whose length is closer to λ.
The above estimate show the decrease by roughly the factor of λ versus the bit complexity estimates in Theorem 9.1. Furthermore, the bounds in parts c) and e) decrease as s increases, and we maximize s when we ensure the saturated initialization.

10 Degeneration in the rings $\mathbb{Z}_m$

10.1 The probability of degeneration in $\mathbb{Z}_{p^v}$ for a random prime p

For a fixed nonsingular matrix M, the condition (5.2) for nondegeneration depends on the prime p. Let us assume a random prime p, fix its power v, and estimate the probability that p^v divides $\det M$, recalling that $s_n(M)$ is a divisor of $\det M$.

We begin with some definitions and basic lemmas. Hereafter $\ln = \log_e$ stands for the natural logarithms (with the base $e = 2.718281\ldots$) and $\pi(y)$ denotes the number of primes not exceeding y.

Lemma 10.1. *(See also (10.4).) If $y > 114$, then $1 < \frac{\pi(y)}{y} \ln y < 1.25$.*

Proof. See Rosser and Schoenfeld 1962 [RS62]. □

Lemma 10.2. *Let $y \geq 114$, then $\pi(y) - \pi(\frac{y}{20}) > (1/\tilde{\beta})\frac{y}{\ln y}$ for*

$$\tilde{\beta} = \frac{1}{1-\tilde{\alpha}} = 1.2049303\ldots, \quad \tilde{\alpha} = \frac{\ln 114}{16 \ln 5.7} = 0.17007650\ldots. \tag{10.1}$$

Proof. By Lemma 10.1, we have $\pi(y) - \pi(\frac{y}{20}) > \frac{y}{\ln y} - \frac{1.25y}{20\ln(y/20)}$. Observe that $\frac{\ln(y/20)}{\ln y}$ is monotone increasing as y grows. So $\frac{1.25}{20\ln(y/20)} \leq \frac{\tilde{\alpha}}{\ln y}$ for $\tilde{\alpha}$ in (10.1) and $y \geq 114$. Combine the above estimates. □

Lemma 10.3. *(Cf. Corollary 7.8.2 in [P01].) Let y, v, h, and k be positive integers such that*

$$y \geq 114, \quad 0 < h^{1/k} \leq y/20. \tag{10.2}$$

Let p be a random prime selected in the range $(y/20, y]$ under the uniform probability distribution. Then Probability$(h \bmod p^v = 0) < \frac{\tilde{\beta} k \ln y}{vy}$ *for $\tilde{\beta}$ in (10.1).*

Proof. Suppose that in the above range there are exactly l distinct primes whose v-th powers divide h. Then the product of these powers also divides h, and therefore we have $h \geq (\frac{y}{20})^{vl}$ because each of the l primes lying in the range $[y/20, y]$ is at least as large as $\frac{y}{20}$. On the other hand, $h \leq (\frac{y}{20})^k$ by assumption. Therefore, $vl \leq k$, that is, $l \leq k/v$. Compare the latter upper bound on l with the lower bound in Lemma 10.2 on the overall number of primes in the range $(\frac{y}{20}, y]$. □

Theorem 10.1. *(Cf. Corollary 7.8.3 in [P01].) Suppose that ξ is a positive number, v is a positive integer, $M \in \mathbb{Z}^{n\times n}$ is nonsingular, and a prime p is randomly sampled from the range $(y/20, y]$ under the uniform probability distribution in this range where $y = \frac{n\xi \ln |M|}{v\epsilon} \geq 114$ and $\xi = \frac{16 \ln 114}{16 \ln 5.7 - \ln 114} = 16\tilde{\alpha}\tilde{\beta} = 3.278885\ldots$ for $\tilde{\alpha}$ and $\tilde{\beta}$ in (10.1). Then we have*

$$P = \text{Probability}((\det M) \bmod p^v = 0) < \epsilon. \tag{10.3}$$

Proof. Write $h = |\det M|$, $k = \frac{n \ln |M|}{\ln(y/20)}$, so that $h \le |M|^n$ and $k \ln \frac{y}{20} \ge \ln h$, which implies (10.2). Apply Lemma 10.3 and deduce that

$$P < \frac{\tilde{\beta} k \ln y}{uy} = \frac{\tilde{\beta} n \ln |M|}{v \ln(y/20)} \frac{\ln y}{y} = \frac{v\epsilon\tilde{\beta} n \ln |M|}{vn \ln |M|} \frac{\ln y}{\xi \ln(y/20)} = \frac{\epsilon\tilde{\beta} \ln y}{\delta \ln(y/20)}.$$

Note that

$$\frac{\ln y}{\ln(y/20)} \le \frac{\ln 114}{\ln 5.7}$$

for $y \ge 114$. Therefore

$$P < \epsilon \frac{\tilde{\beta} \ln 114}{\xi \ln 5.7} = \frac{16\tilde{\alpha}\tilde{\beta}}{\xi}\epsilon = \epsilon.$$

□

To extend the above results to smaller y, one may exploit the known extensions of Lemma 10.1, e.g.,

$$1 + \frac{1}{2 \ln y} < \pi(y) \frac{\ln y}{y} < 1 + \frac{3}{2 \ln y} \tag{10.4}$$

for $y \ge 59$ [GG03, Theorem 18.7]. Refined estimates for $\pi(y)$ can be found in Karatsuba 1990 [K90].

Let us extend Theorem 10.1 to any integer q instead of $q = p^v$. We rely on the following observation.

Lemma 10.4. *Let p and q be coprime and let u, v, and h be three positive integers. Then $p^u q^v$ divides h if and only if both p^u and q^v divide h.*

Corollary 10.1. *Let p_1, ..., p_h be h distinct primes sampled randomly and independently in the ranges $(y_i/20, y_i]$, $i = 1, \dots, h$, respectively, under the uniform probability distribution. Here $y_i = \frac{\xi n}{2u_i\epsilon} \ln |M| \ge 114$ for ξ in Theorem 10.1 and $i = 1, \dots, h$; the matrix $M \in \mathbb{Z}^{n \times n}$ is nonsingular; $v_1, \dots, v_h$ are positive integers, and*

$$2h - 2 \le \frac{y_i}{\tilde{\beta} \ln y_i} \tag{10.5}$$

for $\tilde{\beta}$ in Lemma 10.2 and for all i. Then we have

$$P = \text{Probability}(p_1^{v_1} \cdots p_h^{v_h} \text{ divides } \det M) \le \epsilon^h.$$

Proof. Corollary 10.1 follows from Lemma 10.4 and Theorem 10.1 for $y = y_i$ and $v = 2v_i$. The primes $p_1, \dots, p_{i-1}$ are excluded from the range $(y_i/20, y_i]$ for every i; this decreases the overall number of primes in this range but less than by twice for $i \le h$ because of (10.5) and Lemma 10.2. The effect of this decrease on the probability estimates is overweighed by the increase of v from v_i to $2v_i$. □

Remark 10.1. *A random integer matrix M is strongly nonsingular in $\mathbb{R}_q^{n \times n}$ for $q = p^v$ or $q = p_1^{v_1} \cdots p_k^{v_k}$ with a probability which is within the factor of n from the respective bounds in Theorem 10.1 and Corollary 10.1.*

10.2 The probability of degeneration for a fixed p

Suppose that for a fixed basic prime p, a random integer matrix M, and two appropriate integers $q = p^v$ and $s = p^u$, we wish to estimate the probability that our computations can be performed with a precision within the word length λ.

Studying computations with general matrices we are guided by the following analytic estimate by Brent and McKay 1987 for the proportion of singular matrices in $\mathbb{Z}_{p^u}^{n\times n}$. (They also supply similar estimates in $\mathbb{Z}_q^{n\times n}$ for any integer $q > 1$.)

Theorem 10.2. *[BMK87, Corollary 2.2]. Write $P_k(r) = (1-r)(1-r^2)\cdots(1-r^k)$, $r = 1/p$. Then the nonsingular matrices make up the fraction $\frac{P_{n+u-1}(r)}{P_{u-1}(r)}$ of all matrices in $\mathbb{Z}_{p^u}^{n\times n}$.*

Brent and McKay show specific estimates for their ratios as $n \to \infty$ and $q = 1, \ldots, 16$. Our Table 10.4 in Section 10.4 shows some statistics of nonsingularity of random integer matrices in $\mathbb{Z}_q$, for $n = 5, 10, 50, 100, q = 2^g$, and $g = 0, 1, \ldots, 20$.

They are in reasonable agreement with the analytic estimates in [BMK87].

For Toeplitz versus general matrices M, the known analytic estimates and the results of our experiments in Tables 10.1–10.3 in Section 10.4 show a little higher proportion of nonsingular matrices in $\mathbb{Z}_q^{n\times n}$. In Daykin 1960 [D60] and Kaltofen and Lobo 1996 [KL96] the case of a prime q is studied.

Theorem 10.3. *For any pair of a prime p and a positive integer n, the fraction of $1/p$ of all Toeplitz matrices in $\mathbb{Z}_p^{n\times n}$ are singular.*

We wish to point out a corollary of independent interest.

Corollary 10.2. *For any pair of a prime p and a positive integer n, consider the space of the pairs of polynomials $u(x)$ and $v(x)$ over $\mathbb{Z}_p$ such that $\deg v(x) = n, \deg u(x) < n$. Then the pairs of coprime polynomials make up a fraction of $1 - 1/p$ in this space.*

Proof. The corollary follows by combining the latter theorem with Proposition 9.1 on page 159 in the book [BP94]. This proposition defines a bijection map of all pairs (h, H) of $h \in \mathbb{Z}_p$ and nonsingular Hankel matrices H in $\mathbb{Z}_p^{n\times n}$ to all pairs of coprime polynomials $u(x)$ and $v(x)$ over $\mathbb{Z}_p$ where $v(x)$ is monic, $\deg v(x) = n$, and $\deg u(x) < n$. Combine the bijection $J : H \leftrightarrow T = HT$ with Theorem 10.3 to count the number of pairs (h, H) where H is nonsingular in $\mathbb{Z}_p^{n\times n}$ and extend this count to the number of pairs of coprime polynomials $u(x)$ and $v(x)$ over $\mathbb{Z}_p$ to obtain the corollary. □

Theorem 10.4. *[KL96, Theorem 5]. For any pair of a prime p and a natural n, the strongly nonsingular matrices (that is, nonsingular with all their leading principal submatrices) make up a fraction of $(1 - \frac{1}{p})(1 - \frac{p-1}{p^2})^{n-1}$ in the space of all Toeplitz matrices in $\mathbb{Z}_p^{n\times n}$.*

We know of no extensions of the above analytic estimates to the rings $\mathbb{Z}_q$ for any integer $q > 1$. Our next results may partly fill this void.

Theorem 10.5. *The fraction of at least* $1 - n/q$ *Toeplitz matrices in* $\mathbb{Z}_q^{n\times n}$ *are nonsingular.*

Proof. There are q^{2n} pairs of univariate polynomials u, v over $\mathbb{Z}_q$ where $\deg u < n$, $\deg v = n$, v is monic. These polynomials are not coprime if and only if their resultant vanishes in $\mathbb{Z}_q$. In virtue of the celebrated lemma in Demillo and Lipton 1978 [DL78] (also in Zippel 1979 [Z79] and Schwartz 1980 [S80]), this occurs for the fraction of at most n/q pairs because the resultant is a polynomial of degree of at most n in the coefficients of u and v. This means at least $(q-n)q^{2n-1}$ pairs of coprime polynomials u and v over $\mathbb{Z}_q$. Due to the bijection in Proposition 9.1 on page 159 in [BP94], already cited in the proof of Corollary 10.2, we have as many pairs (h, H) in $(\mathbb{Z}_q, \mathbb{Z}_q^{n\times n})$ where H is a nonsingular Hankel matrix. Therefore, there are at least $(q-n)q^{2n-2}$ nonsingular matrices among a total of q^{2n-1} Hankel matrices in $\mathbb{Z}_q^{n\times n}$. The bijection $J : H \leftrightarrow T = HJ$ extends this count to Toeplitz matrices. □

Corollary 10.3. *The fraction of at least* $1 - \frac{(n+1)n}{2q}$ *Toeplitz matrices in* $\mathbb{Z}_q^{n\times n}$ *are strongly nonsingular.*

Proof. There are at most iq^{2n-2} Toeplitz matrices in $\mathbb{Z}_q^{n\times n}$ with singular $i \times i$ leading principal submatrix for $i = 1, \dots, n$, due to Theorem 10.5. This makes up at most $\sum_{i=1}^{n} iq^{2n-2} = q^{2n-2}n(n+1)/2$ submatrices which are not strongly nonsingular in the set of all q^{2n-1} Toeplitz matrices in $\mathbb{Z}_q^{n\times n}$. □

According to the latter results as well as the results of our experimental tests for nonsingularity of random integer Toeplitz and general matrices in $\mathbb{Z}_{q^w}^{n\times n}$ for $q = 2$, $w \le 20$, and $n \le 100$ presented in Section 10.4, the transition to the rings $\mathbb{Z}_{p^w}$ for larger p^w keeps the chances for the degeneration quite remote on the average.

10.3 Additive perturbations counter degeneracies

Suppose we have the rare case where, for a fixed triple of λ, M and p, one cannot perform the generalized lifting by computing within the word size precision because $(\det M) \bmod p^v = 0$ for all $v \le \lambda$. Suppose we prefer not to change p. Should we necessarily give up lifting? Not right away, because we may usually reduce the solution of the linear system $M\mathbf{x} = \mathbf{b}$ to solving a linear system with the coefficient matrix of the form

$$M_i = M - U_i V_i. \tag{10.6}$$

Here U_i in $\mathbb{Z}_a^{n\times i}$ and V_i in $\mathbb{Z}_b^{i\times n}$ are random general (or random Toeplitz) matrices for two integers

$$b \ge 2n^2 \log(n|M|), \quad a \ge 21n^2 b, \tag{10.7}$$

and a relatively small $i = O(1)$.

Namely, we fix two positive integers i_+ and j_+ and recursively apply our lifting initialization algorithms to the matrices $M_{i,j} = M - U_{i,j}V_{i,j}$ for random matrices $U_{i,j}$ and $V_{i,j}$ for $i = 1$, $j = 1, \dots, j_+$; $i = 2$, $j = 1, \dots, j_+$; $\dots$, and so on, until we either yield the desired initialization for $M_i = M_{i,j}$ and some $i \le i_+$, $j \le j_+$ by computing with the word precision λ or reach $i = i_+ + 1$. In the latter case, the algorithm outputs FAILURE. In the former case we compute the vector $M^{-1}\mathbf{b}$ for a fixed integer vector $\mathbf{b}$ based on (10.6) and the Sherman–Morrison–Woodbury formula [GL96, page 50],

$$M^{-1} = (M_i + U_iV_i)^{-1} = M_i^{-1} - M_i^{-1}U_i(I_i + V_iM_i^{-1}U_i)^{-1}V_iM_i^{-1}. \quad (10.8)$$

The formula holds provided that the matrices M_i, M, and

$$W_i = I_i + V_iM_i^{-1}U_i$$

are nonsingular for the pair of $n \times i$ matrices U_i and V_i^T. We refer to these computations as **Algorithm 10.1**.

We first apply the generalized lifting to the vector $qM_i^{-1}\mathbf{b}$ and to every column of the $n \times i$ matrix $qM_i^{-1}U_i$ to obtain these vector and columns in $\mathbb{Z}_{qs^h}$. Then we compute the vector

$$qM^{-1}\mathbf{b} \bmod (qs^h) = ((I - qM_i^{-1}U_i(qI + qV_iM_i^{-1}U_i)^{-1}V_i)qM_i^{-1}\mathbf{b}) \bmod (qs^h),$$

and reconstruct the rational vector $M^{-1}\mathbf{b}$.

For random matrices M in $\mathbb{Z}^{n\times n}$, the algorithm is likely to succeed already for reasonably small integers i_+ and j_+ due to the two following theorems in [EGV00], which relate this likelihood to the choice of the bounds i_+ and j_+.

Theorem 10.6. *[EGV00, Theorem 3.8]. For two positive integers i and n, $i < n$, a nonsingular matrix M in $\mathbb{Z}^{n\times n}$, and sufficiently large integers a and b satisfying (10.7), let $U_i = U_{i,j}$ and $V_i^T = V_{i,j}^T$ denote the pairs of random matrices in $\mathbb{Z}_a^{n\times i}$ for $j = 1, 2, \dots, 15$, and in $\mathbb{Z}_b^{n\times i}$ for $j = 16$, and let the matrices $M_i = M_{i,j}$ be defined by (10.6). Then with a probability of at least $1/2$, we have $s_{n-i}(M) = \gcd(s_n(M), \gcd_{j=1}^{16}(s_n(M_{i,j})))$. To increase the probability bound above $1 - \varepsilon$ for a fixed positive ε, it is sufficient to include j_+ matrices $M_{i,j}$, $j = 1, \dots, j_+$, for every i and for a sufficiently large j_+ in $O(\log(1/\varepsilon))$.*

Theorem 10.7. *[EGV00, Theorem 6.2]. For a fixed pair of integers $\lambda > 0$ and η, let the entries of an $n \times n$ matrix M be independently sampled under the uniform probability distribution in a set of integers $\eta, \eta + 1, \dots, \eta + \lambda - 1$. Then* Probability$(s_{n-j}(M) > 1) \le \lambda^{-n} + 9(\frac{2}{3})^{j-1} + \frac{n^3}{\lambda^{j-1}}$.

Due to Theorems 10.6 and 10.7 (and also according to the well known statistics), we have with a high probability that

$$\gcd\left(s_n(M), \gcd_{j=1}^{j_+}(s_n(M_{i,j}))\right) = 1$$

for a random $n \times n$ integer matrix M, the matrices $M_{i,j}$ defined above, some $i \le i_+$, and reasonably small integers i_+ and j_+. In fact we just need a weaker property that the above gcd is likely to be coprime with a fixed prime p, and this property has been statistically observed in our experiments with random Toeplitz matrices for $p = 2$ (see the next subsection).

10.4 Experimental computations: how frequently is a random integer Toeplitz matrix non-singular modulo a fixed power of two?

In out tests we have randomly generated an $n \times n$ Toeplitz matrix $M = (t_{i-j})_{i,j}$. Its entries $t_{1-n}, \dots, t_{n-1}$ have been chosen independently of each other under the uniform random distribution on $\mathbb{Z}_q$ for $q = 2^w$ and for a positive integer w. The first column in each of Tables 10.1–10.3 shows how frequently in our tests a random $n \times n$ integer Toeplitz matrix M was nonsingular in $\mathbb{Z}_q$.

Whenever the test showed singularity, we repeated the test recursively (up to at most four times), each time adding the outer product of two random vectors to the input matrix. The *(1+i)*-th column of each table, for $i = 0, 1, 2, 3, 4$, shows for how many out of 100,000 samples the results were positive for the matrices $M - U_j V_j^T$, for some $j \le i$ where $U_j, V_j^T \in \mathbb{Z}_q^{n \times l}$, $M \in \mathbb{Z}_q^{n \times n}$, $q = 2^w$. These data should motivate using Algorithm 10.1 for smaller i_+ and j_+. They should be compared with similar statistics for general, tridiagonal, and five-diagonal matrices. Table 10.4 shows such statistics, although without small rank perturbations. According to our tests in the case where $q = 2^w$, the degeneration is more likely for five-diagonal than general matrices, and is even more likely for tridiagonal matrices, but even for the latter matrices it is quite rare for larger w. We have also observed for tridiagonal matrices that the degeneration is substantially less likely where we shift from $q = 2^w$ to $q = 5^w$.

Table 10.1: Number of times the matrix $M + A_i$ for a random 20×20 Toeplitz matrix M and a random 50×50 matrix A of rank at most i is nonsingular in the ring $\mathbb{Z}_q$ for $q = 2^w$ out of 100,000 samples

$w\backslash i$	0	1	2	3	4
1	50173	59450	66672	72514	77452
2	68814	80808	87785	92256	95133
3	82971	92311	96197	98164	99136
4	90559	96899	98862	99567	99852
5	95079	98809	99671	99907	99973
6	97333	99557	99907	99981	99997
7	98643	99859	99973	99998	100000
8	99302	99948	99993	99999	100000
9	99639	99983	100000	100000	100000
10	99816	99997	100000	100000	100000
11	99903	99999	100000	100000	100000
12	99955	100000	100000	100000	100000

Table 10.2: Number of times the matrix $M + A_i$ for a random 50×50 Toeplitz matrix M and a random 50×50 matrix A of rank at most i is nonsingular in the ring $\mathbb{Z}_q$ for $q = 2^w$ out of 100,000 samples

$w\backslash i$	0	1	2	3	4
1	50054	59383	66661	72665	77581
2	68781	80792	87812	92341	95151
3	82842	92263	96282	98203	99139
4	90507	96868	98877	99589	99844
5	95132	98846	99695	99915	99976
6	97440	99597	99912	99981	99994
7	98667	99857	99972	99994	99998
8	99315	99953	99989	99997	99999
9	99653	99985	100000	100000	100000
10	99829	99997	100000	100000	100000
11	99917	99999	100000	100000	100000
12	99967	100000	100000	100000	100000

Table 10.3: Number of times the matrix $M + A_i$ for a random 100×100 Toeplitz matrix M and a random 100×100 matrix A of rank at most i is nonsingular in $\mathbb{Z}_q$ for $q = 2^w$ out of 100,000 samples

	0	1	2	3	4
1	50170	59672	66652	72460	77368
2	68969	80960	87833	92188	95130
3	82799	92261	96240	98128	99122
4	90498	96935	98884	99570	99845
5	94975	98837	99662	99893	99971
6	97255	99547	99898	99970	99991
7	98591	99827	99966	99994	99998
8	99249	99931	99989	99998	99998
9	99616	99976	99997	100000	100000
10	99804	99994	100000	100000	100000
11	99898	99998	100000	100000	100000
12	99948	100000	100000	100000	100000

Table 10.4: Number of times a random $n \times n$ general matrix M is nonsingular in the ring $\mathbb{Z}_q$ out of $100,000$ samples for $q = 2^w$

w	$n = 5$	$n = 10$	$n = 50$	$n = 100$
$w = 0$	29,986	28,781	28,940	28,781
$w = 1$	58,637	57,679	57,884	57,782
$w = 2$	77,650	76,817	77,047	77,104
$w = 3$	88,399	87,916	88,000	88,080
$w = 4$	94,102	93,888	93,943	93,921
$w = 5$	97,046	96,911	96,963	96,937
$w = 6$	98,519	98,414	98,483	98,452
$w = 7$	99,245	99,180	99,212	99,235
$w = 8$	99,634	99,598	99,590	99,620
$w = 9$	99,820	99,791	99,783	99,806
$w = 10$	99,911	99,894	99,892	99,899
$w = 11$	99,956	99,957	99,950	99,953
$w = 12$	99,977	99,977	99,978	99,980
$w = 13$	99,985	99,992	99,991	99,992
$w = 14$	99,992	99,996	99,993	99,995
$w = 15$	99,993	99,997	99,996	99,998
$w = 16$	99,995	99,999	99,999	99,998
$w = 17$	99,998	99,999	99,999	99,998
$w = 18$	99,999	100,000	99,999	99,999
$w = 19$	99,999	100,000	100,000	100,000
$w = 20$	99,999	100,000	100,000	100,000

Table 10.5: Number of times a random $n \times n$ tridiagonal matrix M is nonsingular in the ring $\mathbb{Z}_q$ out of $1,000,000$ samples for $q = 2^w$

w	$n = 5$	$n = 10$	$n = 50$	$n = 100$
$w = 1$	117356	17514	0	0
$w = 2$	320625	75599	0	0
$w = 3$	531878	182052	1	0
$w = 4$	703335	324629	4	0
$w = 5$	823672	478421	17	0
$w = 6$	899773	620734	64	0
$w = 7$	945210	738216	188	0
$w = 8$	970459	828122	494	0
$w = 9$	984437	891854	1324	0
$w = 10$	991892	934290	3043	0
$w = 11$	995862	961334	6210	0
$w = 12$	997903	978026	11855	0
$w = 13$	998940	987761	20951	0
$w = 14$	999455	993236	34980	1
$w = 15$	999719	996395	54784	1
$w = 16$	999859	998089	82128	7
$w = 17$	999920	999012	117742	13
$w = 18$	999962	999515	161178	22
$w = 19$	999980	999743	213241	37
$w = 20$	999988	999866	271703	72
$w = 21$	999996	999947	336451	138
$w = 22$	999999	999973	404624	289
$w = 23$	999999	999986	474595	520
$w = 24$	1000000	999992	543974	941
$w = 25$		999996	610648	1601
$w = 26$		999996	673129	2629
$w = 27$		999999	730268	4193
$w = 28$		1000000	780932	6542
$w = 29$			825245	9787
$w = 30$			862955	14404
$w = 31$			893896	20884
$w = 32$			919407	29409

Table 10.6: Number of times a random $n \times n$ tridiagonal matrix M is nonsingular in the ring $\mathbb{Z}_q$ out of $1,000,000$ samples for $q = 5^w$

w	$n = 5$	$n = 10$	$n = 50$	$n = 100$
$w = 1$	629193	453573	33036	1319
$w = 2$	902825	795859	142767	9536
$w = 3$	978126	939323	326520	36124
$w = 4$	995314	984670	537288	94676
$w = 5$	999063	996494	720272	192946
$w = 6$	999808	999263	849531	324563
$w = 7$	999964	999840	927780	473054
$w = 8$	999992	999968	968268	618013
$w = 9$	999997	999991	987267	741988
$w = 10$	1000000	999999	995305	837468
$w = 11$	1000000	1000000	998419	904365
$w = 12$	1000000	1000000	999479	947171
$w = 13$	1000000	1000000	999829	972681

Table 10.7: Number of times a random $n \times n$ tridiagonal matrix M is nonsingular in the ring $\mathbb{Z}_q$ out of $1,000,000$ samples for $q = 10^w$

w	$n = 5$	$n = 10$	$n = 50$	$n = 100$
$w = 1$	673132	462691	33063	1286
$w = 2$	934076	811362	142940	9469
$w = 3$	989846	950586	326682	36115
$w = 4$	998658	989599	537027	95083
$w = 5$	999834	998176	720163	193223
$w = 6$	999981	999689	850071	325041
$w = 7$	999997	999950	927627	473759
$w = 8$	1000000	999991	968218	618106
$w = 9$	1000000	999998	987183	742050

Table 10.8: Number of times a random $n \times n$ five-diagonal matrix M is nonsingular in the ring $\mathbb{Z}_q$ out of $1,000,000$ samples for $q = 2^w$

w	$n = 5$	$n = 10$	$n = 50$	$n = 100$
$w = 1$	291205	138605	407	0
$w = 2$	554299	353025	3189	1
$w = 3$	744030	560876	12666	28
$w = 4$	860894	725098	35279	131
$w = 5$	926953	837646	77277	534
$w = 6$	962364	908618	141802	1617
$w = 7$	980846	950327	227489	4233
$w = 8$	990280	973763	327674	9452
$w = 9$	995111	986273	433370	19223
$w = 10$	997602	992932	537207	35864
$w = 11$	998787	996416	631600	61342
$w = 12$	999400	998230	713344	97265
$w = 13$	999694	999130	781426	144238
$w = 14$	999861	999545	836186	201805
$w = 15$	999936	999768	878746	268132
$w = 16$	999969	999880	911666	340560
$w = 17$	999984	999938	936426	415820
$w = 18$	999993	999968	955325	490841
$w = 19$	999996	999986	969282	562743
$w = 20$	999998	999994	979171	628840
$w = 21$	999999	999999	986186	687759
$w = 22$	1000000	1000000	991041	738768
$w = 23$			994296	782062
$w = 24$			996454	818466
$w = 25$			997802	849199
$w = 26$			998675	874827
$w = 27$			999193	896088
$w = 28$			999547	914020
$w = 29$			999746	929000
$w = 30$			999855	942334
$w = 31$			999914	953521
$w = 32$			999957	962893

Analysis of the results of the experiments

For fixed q and n, we assume that M is singular over $\mathbb{Z}_q$ with a probability p. Next we estimate p. Let x be a random variable such that

$$x = \begin{cases} 1, & \det M = 0 \bmod q; \\ 0, & \det M \neq 0 \bmod q. \end{cases}$$

Let $x_1, \ldots, x_m$ be the observed values of x. By the Central Limit Theorem,

$$\lim_{m\to\infty} \frac{(x_1 + \ldots + x_m) - mp}{\sqrt{mp(1-p)}} = N(0,1)$$

where $N(0,1)$ is the standard normal probability distribution. Therefore, a confidence interval of probability $1-\alpha$ for p is

$$\left(\bar{x} - Z_{\alpha/2}\sqrt{\bar{x}(1-\bar{x})/m},\ \bar{x} + Z_{\alpha/2}\sqrt{\bar{x}(1-\bar{x})/m}\right)$$

where $\bar{x} = \frac{1}{m}(x_1 + \ldots + x_m)$, Z_α is defined by Probability$(N(0,1) > Z_\alpha) = \alpha$.

Example 10.1. *For $g = 8, n = 50$, we are "99.9%" sure that*

- *Probability(Toeplitz matrix M is non-singular)* $= 0.993 \pm 0.001$*;*
- *Probability(Toeplitz matrix M is strongly non-singular)* $= 0.731 \pm 0.005$*;*
- *Probability(general matrix M is non-singular)* $= 0.992 \pm 0.001$*;*
- *Probability(general matrix M is strongly non-singular)* $= 0.688 \pm 0.005$*.*

11 Demonstration of algorithms with examples

Let us demonstrate the work of Algorithms 3.1 and 10.1 with simple examples.

Example 11.1. $M = \left(\begin{smallmatrix} 2 & 1 \\ 3 & 2 \end{smallmatrix}\right)$, $\mathbf{b} = \binom{3}{4}$. *So* $\mathbf{x} = \binom{2}{-1}$. *By applying Algorithm 3.1 for $q = 1$, $s = 2$, $\mathbf{r}^{(0)} = \mathbf{b}$, we successively compute* $Q = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$, $\mathbf{u}^{(0)} = \binom{0}{1}$, $\mathbf{r}^{(1)} = \binom{1}{1}$, $\mathbf{u}^{(1)} = \binom{1}{1}$, $\mathbf{r}^{(2)} = \binom{-1}{-2}$, $\mathbf{u}^{(2)} = \binom{0}{1}, \ldots$ *to define* $\mathbf{x}^{(3)} = 2\mathbf{x} \bmod 8 = \binom{0}{1} + 2\binom{1}{1} + 4\binom{0}{1}$.

Example 11.2. $M = \left(\begin{smallmatrix} 4 & 1 \\ 6 & 2 \end{smallmatrix}\right)$, $\mathbf{b} = \binom{3}{4}$, *so* $\mathbf{x} = \binom{1}{-1}$.

a) By applying Algorithm 3.1 for $q = s = 2$, $\mathbf{r}^{(0)} = \mathbf{b}$, we successively compute $Q = \left(\begin{smallmatrix} 0 & 1 \\ 2 & 0 \end{smallmatrix}\right)$, $\mathbf{u}^{(0)} = \binom{0}{2}$, $\mathbf{r}^{(1)} = \binom{1}{1}$, $\mathbf{u}^{(1)} = \binom{1}{2}$, $\mathbf{r}^{(2)} = \binom{-1}{-2}$, $\mathbf{u}^{(2)} = \binom{2}{2}, \ldots$. *So, we have* $\mathbf{x}^{(3)} = 2\mathbf{x} \bmod 8 = \binom{0}{2} + 2\binom{1}{2} + 4\binom{2}{2}$, $(M\mathbf{x}^{(h)} - 2\mathbf{b}) \bmod 2^{h+1} = 0$ *for $h = 1, 2, 3$.*

b) Alternatively, we observe that $s_2(M) = 2$, $s_1(M) = 1$ *and apply Algorithm 10.1 to* $M_1 = M - U_1V_1$, $U_1 = V_1^T = \binom{1}{1}$, *and* $\mathbf{b} = \binom{3}{4}$, *so that* $M_1 = \left(\begin{smallmatrix} 3 & 0 \\ 5 & 1 \end{smallmatrix}\right)$, $M_1^{-1} = \left(\begin{smallmatrix} 13 & 0 \\ -53 & 1 \end{smallmatrix}\right)$. *Due to the Sherman–Morrison–Woodbury formula (10.8), this reduces the computation of* $\mathbf{x}$ *to the triple application of Algorithm 3.1 for* $q = 1$, $s = 2$ *with the right-hand-side vectors* $\mathbf{b} = \binom{3}{4}$, $\mathbf{b}^{(1)} = \binom{1}{1}$, *and* $\mathbf{b}^{(2)} = (1/3)\left(\begin{smallmatrix} 1 & 1 \\ 1 & 1 \end{smallmatrix}\right) M_1^{-1}\binom{3}{4}$, *respectively. We obtain* $M_1^{-1}\binom{3}{4} = \binom{1}{-1}$, $M_1^{-1}\binom{1}{1} = \binom{1/3}{-2/3}$. *Therefore,* $\mathbf{b}^{(2)} = \mathbf{0}$, $M_1^{-1}\mathbf{b}^{(2)} = \mathbf{0}$, $M^{-1}\mathbf{b} = M_1^{-1}\mathbf{b} = \binom{1}{-1}$.

Example 11.3. $M = \left(\begin{smallmatrix} 32 & 2 \\ 48 & 4 \end{smallmatrix}\right)$, $\mathbf{b} = \binom{24}{32}$. *So,* $\mathbf{x} = \binom{1}{-4}$, $s_2(M) = 32$, $s_1(M) = 2$. *We may*

a) apply Algorithm 3.1 to M *and* $\mathbf{b}$ *for* $q = 3, s = 2$, *or*

b) apply Algorithm 10.1 to $M_1 = M - U_1V_1$, $U_1 = \binom{1}{1}$, $V_1^T = \binom{2}{2}$, $M_1 = \left(\begin{smallmatrix} 30 & 0 \\ 46 & 2 \end{smallmatrix}\right)$. *For solving the equations* $M_1^{-1}\mathbf{b}^{(i)}, i = 1, 2, 3$ *(cf. Example 11.2 b), apply Algorithm 3.1 for* $q = s = 2$.

12 Further extensions and a discussion

12.1 Extension of the class of structured matrices

Our lifting and rational reconstruction algorithms can be applied effectively to any integer or rational input matrix provided it is defined and nonsingular in $\mathbb{Z}_q$ and its precomputed inverse and itself can be multiplied by vectors fast. Toeplitz/Hankel-like, Vandermonde-like, Pick and Cauchy-like, and sparse and structured (particularly banded) matrices seem to be obvious examples. The same initialization and rational reconstruction algorithms as well as the probability estimates in Section 10.1 for degeneration in the case of using a random prime can be applied to all these matrices as well. See [P01, Chapter 5] and [PSA95] on the extension of the MBA algorithm to these matrices.

To apply Algorithm 3.1 to banded matrices M we just need to precompute (a generator for) the matrix Q satisfying (2.2) or just to define a black box for its fast multiplication by a vector. Besides the MBA algorithm, both block cyclic reduction (see Heller 1976 [H76]) and Gaussian elimination enables us to compute $M^{-1}\mathbf{v}$ fast even with no preconditioning if M is a banded and strongly nonsingular. Here, Gaussian elimination is slightly faster than the other algorithms; pre-computing the LU factors of M yields a further small acceleration. If M is nonsingular in $\mathbb{Z}_p$ but not strongly non-singular, we may alternatively apply the well-known inversion formulae in Asplund 1959 [A59], Ikebe 1979 [I79], Barrett 1979 [B79], and Barrett and Feinsilver 1981 [BF81], which enable preprocessing for fast multiplication by a vector of the inverse matrix Q satisfying (2.2). For matrices M having a bandwidth in $O(1)$, multiplication of the matrix Q by a vector in linear arithmetic time using linear memory space can be performed based on the compressed representation of this matrix. For a large and important class of sparse linear systems (arising from discretization of the

PDE's), appropriate triangular factorization into sparse factors is defined with the techniques of generalized nested dissection in Gilbert and Hafsteinsson 1990 [GH90] and Gilbert and Schreiber 1992 [GS92]. This factorization provides the preconditioning for the subsequent rapid multiplication of the matrix M^{-1} by a vector.

Our lifting algorithm can also be applied to linear systems of equations with the semi-separable plus diagonal matrices, which generalize banded matrices. The preconditioning can rely on the inversion formulae in Eidelman and Gohberg 1997 [EG97].

Our probability estimates for the degeneration of general and Toeplitz matrices in the case of a fixed prime are not easily extended to the sparse, banded, and other structured matrices. This is an open research area. Here are our initial comments.

Recall that $\det V = \prod_{i>j}(x_i - x_j)$ for a Vandermonde matrix $V = (x_i^j)_{i,j=0}^{n-1}$ and $\det C = \prod_{i<j}(s_i - s_j)(t_i - t_j)/\prod_{i,j}(s_i - t_j)$ for a Cauchy matrix $C = (\frac{1}{s_i - t_j})_{i,j=0}^{n-1}$. Therefore Vandermonde (resp. Cauchy) matrices are nonsingular (resp. defined and nonsingular) if and only if $x_0, \dots, x_{n-1}$ (resp. $s_0, \dots, s_{n-1}$, $t_0, \dots, t_{n-1}$) are distinct, and if so, they are also strongly nonsingular. For random choice of the parameters x_k, s_k, t_k, the degeneration is quite likely for these matrices in $\mathbb{Z}_{p^w}$ for smaller p, e.g., $p = 2$, and $w = o(n)$. We may try to avoid the problem by applying the displacement transformations to the Toeplitz/Hankel-like matrices [P90], [P01], [CP04]. These transformations produce matrices with real or complex entries, and so we should truncate these entries and scale the matrices to arrive at an integral input. The same recipe can be applied to Vandermonde-like, Pick, and other Cauchy-like matrices.

The results of our experimental tests of the singularity of random tridiagonal and five-diagonal integer matrices reported in our Tables 10.5 – 10.8 show that singularity modulo p^w is a little more likely than for the Toeplitz and general matrices, particularly in the case of $p = 2$ and tridiagonal input matrices.

12.2 Computing the determinant and Smith's factors of structured and general matrices

We have already cited the application of fast algorithms for solving a block Hankel linear system to the computation of the determinant of a general matrix. The MBA algorithm outputs the determinant of an input matrix M as by-product and also certifies its correctness at a low cost. Since the resultant of two univariate polynomials is a Toeplitz-like matrix, we yield the resultant as a special case. The application of the MBA algorithm implies the increase of the overall Las Vegas complexity bounds by the factor of $\log n$ versus our solution of a linear system with lifting. To avoid this increase and also to compute the Smith factors of M, let us recall the randomized Monte Carlo approach proposed in [EGV00] for general input matrices and adapt it to the Toeplitz/Hankel-like case.

In [EGV00], computing Smith's factors and the determinant of a general

integer matrix M is reduced to solving a small number of linear systems $M\mathbf{x}_k = \mathbf{b}_k$ for random vectors $\mathbf{b}_k$. The reduction is immediately extended to a Toeplitz/ Hankel-like matrix M. Here are the resulting bit cost estimates.

Theorem 12.1. *Allow output errors with a probability of at most $\nu > 0$, and also allow an additional factor of $\log(1/\nu)$ in all asymptotic estimates in Theorem 9.1 for the numbers of random bits and bit operations. Then the resulting (increased) estimates apply to the computation of Smith's leading factor $s_n(M)$ of an $n \times n$ integer Toeplitz/Hankel-like matrix M; the estimates do not include the correctness verification cost. Up to increasing the bit operation complexity bounds by the factor of k and sampling $O(kn \log n)$ additional random bits, the same bounds cover the computation of the next k distinct leading Smith's factors of M; with the l-fold increase, the bit operation cost bounds of Theorem 9.1 cover the computation of all Smith's factors of M and $\det M$ (without correction verification) where l is the overall number of distinct Smith's factors, $l \le \sqrt{\log\det |M|} \le \sqrt{n \log |M|}$ for every matrix $M \in \mathbb{Z}^{n\times n}$.*

The theorem is supported by the algorithm in [EGV00] complemented by the smaller complexity bounds for solving Toeplitz (rather than general) linear systems, given by Theorem 9.1. We recall a basic lemma in [EGV00].

Lemma 12.1. *Let $\mathbf{b}$ be a random vector in $\mathbb{Z}^n$. Then δ in (5.2) divides $s_n = s_n(M)$, and furthermore, for any prime p, we have*

$$\text{Probability}(\text{ord}_p(\delta) < \text{ord}_p(s_n)) \le \max\{1/\eta, 1/p\}.$$

Proof. The lemma follows from Theorem 2 in [ABM99], but here is a simple direct proof. We have $x_i = \sum_j (-1)^{i+j} d_{i,j} b_j / \det M$, $s_n = |(\det M)/d|$, $d = \gcd(d_{i,j})_{i,j}$ for $d_{i,j}$ in Definition 2.1, and $\mathbf{b} = (b_j)_{j=1}^n$. Write $h_{i,j} = \text{ord}_p(d_{i,j})$, $h = \text{ord}_p(d) = \min_{i,j} d_{i,j}$. We have $h = \text{ord}_p(d_{u,v})$ for some u, v; w.l.o.g., let $u = v = 0$. Furthermore, write $\bar{d}_{i,j} = d_{i,j}/d$ for all i and j. Then it follows that $s_n x_0^{(k)} = \bar{d}_{0,0} b_0^{(k)} + r$, where $r = \sum_{j=1}^{n-1} (-1)^j \bar{d}_{0,j} b_j^{(k)} \in \mathbb{Z}$. It remains to recall that $\text{ord}_p(\bar{d}_{0,0}) = 0$ and $b_0^{(k)}$ is randomly sampled from $\mathbb{Z}_\eta$, independently of $\bar{d}_{0,0}$. □

Finally, our accelerated solution of block Toeplitz/Hankel linear system can be incorporated into the recent algorithm in [KV01], [KV04] for computing the value and sign of the determinant and Smith's factors of a general matrix. Moreover, this stage is the bottleneck for practical application of the algorithm, and the lifting removes this obstable. We may slightly accelerate the solution in [P04a] by applying our algorithm directly to the block Hankel input, rather than shifting to Hankel–like matrices.

12.3 Further topics

The list of the remaining open problems includes the extension of our theorems on and statistics of degeneration in $\mathbb{Z}_{p^w}$ for a fixed p, in particular the extension

of the results in Section 10.2 from Toeplitz to Toeplitz-like, banded, and other structured and/or sparse matrices and theoretical support for our data in Section 10.4.

In the case where the integral input values are obtained via truncation of real and complex values followed by scaling, it is interesting to find out how variations of the truncation policy and small input perturbations affect the nonsingularity in $\mathbb{Z}_q$.

It is also important to refine our codes for our algorithms in the rings $\mathbb{Z}_{qs}$ for $s = 2^u$ and $q = 2^v$ and to experiment with the parameters involved, e.g., a and m in our initialization Algorithms A.1 and A.2 in the Appendix. Another important direction is the extension of these codes to computing polynomial gcd, lcm, etc. and their experimental comparison with the alternative computations in $\mathbb{Z}_p$ for larger random primes p.

Should we expect to see a further asymptotic decrease of our bit complexity estimates? The factor of $m(n)$ in them comes from our basic operation of Toeplitz matrix-by-vector multiplication or equivalently polynomial multiplication. It is unlikely that any efficient algebraic computation scheme for our tasks could dispense with this operation. (Try to imagine such a scheme, e.g., for polynomial gcd.) This informal argument suggests that improvement of our bit complexity bounds by the factor of $m(n)/n$ is unlikely. Our basic operation can be viewed as multiplication of polynomials with bounded integer coefficients, and therefore the binary segmentation technique of Fischer and Paterson 1974 [FP74] (cf. [BP94, Section 3.9]) could yield theoretical acceleration by the factor of $(\log\log n)\log\log\log n$. The resulting complexity bound in $O(n\mu(n\log n))$, however, is not practically attractive unless n is huge. Indeed the overhead constant C_{ss} is large, whereas with C_{class} and C_k in (2.4) the overall bit complexity bounds become as large as n^α for $\alpha > 2.5$.

Appendix

A Alternative initializations of the generalized Toeplitz–Hensel's lifting modulo the power of a prime

Let us specify and analyze two alternative algorithms for the initialization of the generalized lifting for factor-q nonsingular Toeplitz linear systems. We first show these algorithms for solving modulo qs a linear system $M\mathbf{x} = q\mathbf{f}$. The integers q and s, both the powers of a fixed prime p, are computed in the process of performing the algorithms. We estimate the bit complexity of these algorithms and extend them to inverting the matrix Q in (2.2). Finally, we compare these algorithms with the MBA initialization in Section 5.2.

Given a prime p, its power $m = p^b$, a matrix M, and a vector $\mathbf{f} = (f_i)_i$, both of our algorithms first compute the rational vector $M_0^{-1}\mathbf{f}$ for the matrix $M_0 = aM + mI$ and a fixed integer a coprime with m ($a = 1$ in our second algorithm). At the final stage of the algorithms, we extend this to computing the vector $qM^{-1}\mathbf{f} \bmod (qs)$ for appropriate q and s, both the powers of p.

A.1 Step 1: solving a linear system with modular continuation

Algorithm A.1. *Initialization of Toeplitz–Hensel's lifting with modular continuation.*

INPUT: *A nonsingular matrix $M \in \mathbb{Z}^{n\times n}$, a vector $\mathbf{f} \in \mathbb{Z}^n$, a prime p, and two integers $m = p^b$ for a positive integer b and $\lambda > 0$, the length of a computer word. (If this length is not bounded, write $\lambda = \infty$.)*

OUTPUT: *FAILURE if ($\det M$) $\bmod (qs) = 0$ or two positive integers, q and s, both being the powers of p such that $qs < 2^\lambda$, and the vector $\mathbf{y} = (qM^{-1}\mathbf{f}) \bmod (qs)$.*

INITIALIZATION: *Choose an integer $a > 1$ coprime with p and such that*

$$\gamma^+ = \beta(\mathbf{f}) + 2(m + a\alpha(M))n < 2^\lambda. \tag{A.1}$$

(We assume that the values of m and a are sufficiently small to have this bound.)

COMPUTATIONS:

1. *Compute the integer $r = m^{-1} \bmod a$ and the matrix $M_0 = mI + aM$; note that $Q = M_0^{-1} \bmod a = rI$.*

2. *Let* $\alpha = \alpha(M_0), \beta = \beta(\mathbf{f})$ *and choose* h *in (7.2) or (8.5) to support deterministic or randomized recovery of the vector* $M_0^{-1}\mathbf{f}$ *according to Sections 7 or 8, respectively. Specifically, in the deterministic case we write*

$$h = 1 + \lfloor \log_a(2((a|M| + m)\sqrt{n})^{2n-1} n\beta(\mathbf{f}) \rfloor. \tag{A.2}$$

Apply Algorithm 3.1 (for $q = 1$*,* M *replaced by* M_0*,* $\mathbf{b}$ *by* $\mathbf{f}$*, and* s *by* a*) to compute the vector* $M_0^{-1}\mathbf{f} \bmod a^h$*; recover the rational vector* $M_0^{-1}\mathbf{f}$*.*

3. *Compute* $d = \max_j \operatorname{ord}_m(\delta((M_0^{-1}\mathbf{f})_j))$*. If* $2d \le b$*, output the integers* $q = p^d$ *and* $s = p^{b-2d} = m/q^2$*; compute and output the vector*

$$\mathbf{y} = (aqM_0^{-1}\mathbf{f}) \bmod (qs) = (qM^{-1}\mathbf{f}) \bmod (qs).$$

Otherwise output FAILURE.

Correctness of the algorithm follows because, as soon as we yield the equation $q^2 s = m$ at Stage 3, we have $M_0/q = (m/q)I + (a/q)M = qsI + (a/q)M = (a/q)M \bmod (qs)$, which implies the desired equations

$$M\mathbf{y} = aqMM_0^{-1}\mathbf{f} = q\mathbf{f} \bmod (qs).$$

The bit operation complexity of performing the algorithm is clearly dominated at its Stage 2. The estimates in Theorem 9.1 can be applied for

$$q = 1, s = a, \gamma = 2\alpha n + \beta, \beta = \beta(\mathbf{f}), \alpha = \alpha(M_0), \tag{A.3}$$

so that $\alpha \le \alpha^+ = m + a\alpha(M), \gamma \le \gamma^+$ for γ^+ in (A.1).

Theorem A.1. *The bit operation complexity of Algorithm A.1 applied to a Toeplitz matrix* M *is bounded according to Theorem 9.1 where* $q, s, \alpha, \beta,$ *and* γ *are defined in (A.3).*

The following properties should guide us in choosing the integers a and b.

(a) The larger a, the fewer lifting steps at Stage 2 of Algorithm A.1.

(b) The larger b, the more bit operations in Algorithm A.1.

(c) The larger a and/or b, the longer the precision of the computations at Stage 2, but the bound (A.1) is sufficient to keep the precision below $\lambda + 1$.

(d) (A.1) holds for a positive integer b if

$$b \le b^+ = \lceil \log_p \Delta \rceil, \Delta = \frac{2^\lambda - 1 - \beta(\mathbf{f})}{2n} - a\alpha(M) > 1. \tag{A.4}$$

(e) If the integer b^+ is fixed and we wish to minimize the word complexity, we should apply Algorithm A.1 for $b = b^+$. If $b^+ \geq 2d$ for d in Stage 2, the algorithm produces the desired output integers q and s and vector $\mathbf{y}$. Otherwise, the algorithm fails, but we may repeat the computations for distinct a and/or p.

We can also see the two following adverse results of increasing the integer d :

(f) If $2d$ exceeds b^+, then Algorithm A.1 fails.

(g) The number h of lifting steps defined in (7.2) and (8.5) for $\alpha = \alpha(M_0)$, $\beta = \beta(\mathbf{f})$ (cf. (A.2)) is roughly proportional to $\log_a \alpha(M_0)$ and $\log_a(a\alpha(M) + m)$. Therefore h is roughly proportional to $d/\log a$ if $m = p^b = p^{2d}$ dominates $a\alpha(M)$.

Let us estimate d.

Theorem A.2. *$d = \max_j \operatorname{ord}_p((\delta(M_0^{-1})_j) = \operatorname{ord}_p(s_n(M_0)) = \operatorname{ord}_p(s_n(M)) \leq \operatorname{ord}_p(\det M)$, and so $b \geq 2d$ at Stage 2 of Algorithm A.1 if $b \geq 2\operatorname{ord}_p(\det M)$. The latter bound holds if $b \geq 2\log_p |\det M|$.*

Proof. The theorem is easily deduced from the definitions of M_0, $\delta(x/y)$, and $s_n(M)$ and from the bounds (2.1) since a and p are coprime. □

Now, in addition to (2.1), recall that for a larger random prime p and/or a random integer matrix M, $\operatorname{ord}_p(s_n(M))$ tends to be within a small factor from $\operatorname{ord}_p(\det M)$ (see Theorems 10.1 and 10.7) and therefore within a small factor from $n\log_p(\sqrt{n}\alpha(M))$. Then, in virtue of Theorem A.2, the integers b and d should be of at most the order of $n\log_p(\sqrt{n}\alpha(M))$. This means that for a moderate bound λ and a larger integer n, Algorithm A.1 should fail, whereas for a larger λ, that is, for computations with the extended precision, the number h of lifting steps at Stage 2 of this algorithm should grow by roughly the factor of $n/\log a$ versus the estimates in (7.2) and (8.5). Due to this growth caused by the term $mI = p^bI$ in M_0, the arithmetic, word, and bit complexity estimates for the initialization with Algorithm A.1 should exceed by roughly the factor of n the respective estimates in Theorem 9.1 for the complexity of the subsequent solution of a Toeplitz linear system. We avoid decreasing h by means of increasing the value $\log a$ because of the high price for increasing $\alpha(M_0)$ and Δ in (A.4).

The above comments apply to the worst case input p and M. For a larger random prime p and/or a random Toeplitz matrix M, however, the chances for the failure of Algorithm A.1 dramatically decrease because the integers $\operatorname{ord}_p(\det M)$ and d tend to be in $O(\log_p n)$ according to the estimates in Sections 10.1 and 10.2. In this case we have $\log p^b = O(\log n), \log\alpha(M_0) = O(\log(a\alpha(M) + n))$, and adding the complexity estimates for the initialization with Algorithm A.1 would not affect our overall asymptotic estimates for solving Toeplitz linear systems.

A.2 Step 1: solving a linear system with variable diagonal

With Algorithm A.1 we cannot keep the computation of the vector $\mathbf{x} = M^{-1}\mathbf{f}$ in binary form because a and s are coprime and thus cannot both equal the powers of two. Our next algorithm does not have this deficiency and still uses about as many lifting steps and bit operations as Algorithm A.1. The lifting stage of our second algorithm can be performed numerically with bounded precision. We specify only deterministic recovery at Stage 2, but one may immediately extend the recipes of Section 8 for randomized or heuristic acceleration. At Stage 2 of this algorithm we apply *numerical rational roundoff*, that is, we recover a unique rational number x/y from three integers ν, δ, and k provided $1 \le y \le k$, $|x| < k$, $|x|$ and y are coprime unless $x = 0$; $|x/y - \nu/\delta| < 1/(2k^2)$, and $|\nu| < \delta$. We can apply the bound (2.5) for $d = \delta$ to the bit-operation complexity of this recovery as well [WP03].

Algorithm A.2. *Initialization of Toeplitz–Hensel's lifting by using the variable diagonal technique (cf. [P00]).*

INPUT: *as in Algorithm A.1 and $c > 1$ such that $m \ge c|M|$.*

OUTPUT: *as in Algorithm A.1.*

INITIALIZATION: *Write $\mathbf{z}_0 = \mathbf{0}$, $\mathbf{r}_0 = \mathbf{f}$.*

COMPUTATIONS *(cf. Definition 2.3):*

1. *Compute the matrices $M_0 = M + mI$ and $Q = m^{-1}I$.*
2. *Recursively compute the vectors $\mathbf{z}_{i+1} - \mathbf{z}_i = Q\mathbf{r}_i = m^{-1}\mathbf{r}_i$, $\mathbf{r}_{i+1} = \mathbf{f} - M_0\mathbf{z}_{i+1} = \mathbf{r}_i - M_0Q\mathbf{r}_i = -m^{-1}M\mathbf{r}_i$ for $i = 0, 1, \dots, h-1$ and*

$$h = \lfloor (2n-1)\log_c(|M| + m) + \log_c(2|\mathbf{f}|^2/(c-1)) \rfloor \tag{A.5}$$

   (cf. (A.2)).
3. *Recover the vector $\mathbf{z} = M_0^{-1}\mathbf{f}$ from $\mathbf{z}_h$ deterministically, by using the numerical rational roundoff algorithms.*
4. *Proceed as in Stage 3 of Algorithm A.1 for $a = 1$ and $\mathbf{y} = \mathbf{z}$.*

Stage 2 can be implemented numerically as the customary residual correction algorithm for iterative improvement of the computed approximations to $\mathbf{z}$ where the initial approximation is given by the scaled identity matrix $Q = m^{-1}I$ (see [S80a], [GL96, Section 3.5.3], [H96]). We employ this algorithm in lieu of Hensel's auxiliary lifting.

We have

$$\begin{gathered} M_0Q - I = QM_0 - I = m^{-1}M, \\ \mathbf{z} - \mathbf{z}_h = M_0^{-1}(\mathbf{f} - M_0\mathbf{z}_h) \\ = M_0^{-1}\mathbf{r}_h, \\ \mathbf{r}_h = -m^{-1}M\mathbf{r}_{h-1} = (-m^{-1}M)^h\mathbf{r}_0 \\ = (-m^{-1}M)^h\mathbf{f}. \end{gathered}$$

Furthermore,

$$|M_0^{-1}| = m^{-1}|(I + M/m)^{-1}| \le m^{-1}\sum_{i=0}^{\infty}(|M|/m)^i,$$

and so

$$|M_0^{-1}| \le \frac{1}{(c-1)m}$$

since $m \ge c|M|$.

Therefore

$$|\mathbf{z} - \mathbf{z}_h| \le (m^{-1}|M|)^h|\mathbf{f}||M_0^{-1}| \le (m^{-1}|M|)^h|\mathbf{f}|/((c-1)m) \le c^{-h}|\mathbf{f}|/((c-1)m) \tag{A.6}$$

for $m \ge 2|M|$.

To ensure correct recovery of the vector $\mathbf{z}$ from $\mathbf{z}_h$ with using the cited numerical rational roundoff algorithms, which extend the algorithms in Section 2.4, it is sufficient to approximate $\mathbf{z}$ by $\mathbf{z}_h$ within the error norm less than $1/(2|M_0|^{2n-1}|\mathbf{f}|)$. This bound is achieved in Algorithm A.2 due to (A.5)–(A.6) and the inequality $|M_0| \le |M| + m$.

The analysis in the previous subsection (for $a = 1$) (including Theorems A.1 and A.2) is immediately extended. b^+ in (A.6) increases since $a = 1$, and the parameter c (rather than a) plays the role of the lifting and logarithmic base (cf. (A.6)).

A.3 Step 2: extension from system solving to matrix inversion and Newton's acceleration

To initialize lifting, we seek the matrix $Q = (qM^{-1}) \mod (qs)$. For general matrix M, this requires the solution of n linear systems of equations with the coefficient matrix M. In the Toeplitz case, we only solve the two linear systems $M\mathbf{x} = q_0\mathbf{t} \mod (q_0s_0)$ and $M\mathbf{y} = q_1\mathbf{e}_1 \mod (q_1s_1)$ where q_0, q_1, s_0 and s_1 denote the respective values of the integer parameters q and s for these two systems and where the two vectors $\mathbf{e}_1 = (1, 0, \dots, 0)^T$ and $\mathbf{t}$ define the generator of the matrix M^{-1} (see Theorem 2.2).

We choose the same basic prime p for both systems and reconcile the choice of $q_0 = q_1$ and $s_0 = s_1$ by computing

$$q = q_0 = q_1 = p^\sigma, \sigma = \max_j \mathrm{ord}_p(\delta(M_0^{-1}(\mathbf{t}, \mathbf{e}_1))_j))$$

and $s = s_0 = s_1 = \frac{m}{q}$ at Stage 3, which is common in Algorithm A.1 (or A.2) for both linear systems with $q\mathbf{t}$ and $q\mathbf{e}_1$ on the right-hand sides.

If the precision at the lifting steps in Stage 2 in Algorithms A.1 or A.2 is substantially less than λ, we may accelerate lifting by applying Newton's steps (4.2) or (4.4), respectively.

A.4 Extension to computations with singular matrices

As by-product, both Algorithm A.1 and A.2 determine whether the matrix M is singular in $\mathbb{Z}$. Therefore combined with the binary search, they can replace the MBA algorithm for computing the rank r and $r \times r$ nonsingular submatrix of a preconditioned matrix, UML, having generic rank profile (cf. [KS91] and the end of our Section 6).

A.5 Comparison with the application of the MBA approach

Recall that Theorem A.1 covers the bit complexity of performing both Algorithms A.1 and A.2 and implies that the estimated overall cost of Toeplitz solving increases versus Theorem 9.1 by a factor ranging from a moderate constant for the random average input matrix M to roughly n in the worst case.

Versus the MBA algorithm, Algorithms A.1 and A.2 have the advantage of avoiding divisions, so that they can be performed in the rings for any input matrix M which can be multiplied by a vector fast and do not fail in $\mathbb{Z}_{qs}$ unless $\frac{M}{q}$ is singular.

The initialization with the algorithm of the MBA type has lower asymptotic bit complexity than the subsequent stages of Toeplitz solving, but Algorithms A.1 and A.2 require a little simpler codes than the MBA algorithm and in particular involve no auxiliary matrices of smaller sizes.

References

[A59] E. Asplund, Inverses of Matrices $(a_{i,j})$ Which Satisfy $a_{i,j} = 0$ for $j > i + p$, *Mathematica Scandinavica*, **7**, 57–60, 1959.

[ABM99] J. Abbott, M. Bronstein, T. Mulders. Fast Deterministic Computation of the Determinants of Dense Matrices, *Proc. of International Symposium on Symbolic and Algebraic Computation (ISSAC'99)*, 197–204, ACM Press, New York, 1999.

[B68] E. H. Bareiss, Sylvester Identity and Multistep Integer–Preserving Caussian Elimination, *Math. of Computations*, **22**, 565–578, 1968.

[B79] W. W. Barrett, A Theorem on Inverses of Tridiagonal Matrices, *Linear Algebra and Its Applications*, **27**, 211–217, 1979.

[B85] J. R. Bunch, Stability of Methods for Solving Toeplitz Systems of Equations, *SIAM J. of Scientific and Statistical Computing*, **6(2)**, 349–364, 1985.

[B03] D. J. Bernstein, Fast Multiplication and Its Applications, preprint, 2003. Availabe from http://cr.yp.to/papers.html

[BA80] R. R. Bitmead, B. D. O. Anderson, Asymptotically Fast Solution of Toeplitz and Related Systems of Linear Equations, *Linear Algebra and Its Applications*, **34**, 103–116, 1980.

[BF81] W. W. Barrett, P. I. Feinsilver, Inverses of Banded Matrices, *Linear Algebra and Its Applications*, **41**, 111–130, 1981.

[BGY80] R. P. Brent, F. G. Gustavson, D. Y. Y. Yun, Fast Solution of Toeplitz Systems of Equations and Computation of Padé Approximations, *J. Algorithms*, **1**, 259–295, 1980.

[BMK87] R. P. Brent, B. D. McKay, Determinants and Ranks of Random Matrices over $\mathbb{Z}_m$, *Discrete Math.*, **66**, 35–49, 1987.

[BP94] D. Bini, V. Y. Pan, *Polynomial and Matrix Computations, Volume. 1: Fundamental Algorithms*, Birkhäuser, Boston, 1994.

[CFG99] G. Cooperman, S. Feisel, J. von zur Gathen, G. Havas, GCD of Many Integers, *Computing and Combinatorics, Lecture Notes in Computer Science*, **1627**, 310–317, Springer, Berlin, 1999.

[CK91] D. G. Cantor, E. Kaltofen, On Fast Multiplication of Polynomials over Arbitrary Rings, *Acta Informatica*, **28(7)**, 697–701, 1991.

[CP04] Z. Chen, V. Pan, An Efficient Solution for Cauchy-like Systems of Linear Equations, *Computers and Mathematics with Applications*, **48**, 529–537, 2004.

[CW90] D. Coppersmith, S. Winograd, Matrix Multiplicaton via Arithmetic Progressions. *J. of Symbolic Computation*, **9(3)**, 251–280, 1990.

[D59] J. Durbin, The Fitting of Time-Series Models, *Review of International Statistical Institute,* **28**, 229–249, 1959.

[D60] D. E. Daykin, Distribution of Bordered Persymmetric Matrices in a Finite Field. *J. Reine und Angewandte Math.,* **203**, 47–54, 1960.

[D82] J. D. Dixon, Exact Solution of Linear Equations Using p-adic Expansions, *Numerische Math.*, **40**, 137–141, 1982.

[DGP04] J.-G. Dumas, P. Giorgi, C. Pernet, FFPack: Finite Field Linear Algebra Package, *Proc. International Symp. on Algebraic and Symbolic Computation (ISSAC'04)*, 119–126, ACM Press, New York, 2004.

[DL78] R. A. Demillo, R. J. Lipton, A Probabilistic Remark on Algebraic Program Testing, *Information Processing Letters*, **7(4)**, 193–195, 1978.

[DSV01] J.-G. Dumas, B. D. Saunders, G. Villard, On Efficient Sparse Integer Matrix Smith Form Computations, *J. of Symbolic Computation*, **32**, 71–99, 2001.

[EG97] Y. Eidelman, I. Gohberg, Inversion Formulas and Linear Complexity Algorithm for Diagonal Plus Semiseparable Matrices, *Computers and Mathematics with Applications*, **33(4)**, 69–79, 1997.

[EG01] Y. Eidelman, I. Gohberg, Fast Inversion Algorithms for a Class of Block Structured Matrices, *Contemporary Mathematics*, **281**, 17–38, 2001.

[EG02] Y. Eidelman, I. Gohberg, A Modification of the Dewilde–Van der Veen Method for Inversion Finite Structured Matrices, *Linear Algebra and Its Applications*, **343–344**, 419–450, 2002.

[EG03] Y. Eidelman, I. Gohberg, Fast Inversion Algorithms for a Class of Structured Operator Matrices, *Linear Algebra and Its Applications*, **371**, 153-190, 2003.

[EGV00] W. Eberly, M. Giesbrecht, G. Villard, On Computing the Determinant and Smith Form of an Integer Matrix, *Proc. 41st Annual Symposium on Foundations of Computer Science (FOCS'2000)*, 675–685, IEEE Computer Society Press, Los Alamitos, California, 2000.

[EPY98] I. Z. Emiris, V. Y. Pan, Y. Yu, Modular Arithmetic for Linear Algebra Computations in the Real Field, *J. of Symbolic Computation*, **26**, 71–87, 1998.

[FP74] M. J. Fischer, M. S. Paterson, String Matching and Other Problems, *SIAM-AMS Proceedings*, **7**, 113–125, 1974.

[GG03] J. von zur Gathen, J. Gerhard, *Modern Computer Algebra*, Cambridge University Press, Cambridge, UK, 2003 (second edition).

[GH90] J. R. Gilbert, H. Hafsteinsson, Parallel Symbolic factorization of Sparse Linear Systems, *Parallel Computing*, **14**, 151–162, 1990.

[GS92] J. R. Gilbert, R. Schreiber, Highly Parallel Sparse Cholesky Factorization, *SIAM J. Scientific Computing*, **13**, 1151–1172, 1992.

[GL96] G. H. Golub, C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, Maryland, 1996 (third addition).

[H76] D. E. Heller, Some Aspects of the Cyclic Reduction Algorithm for Block Tridiagonal Linear Systems, *SIAM J. on Numerical Analysis*, **13**, 484–496, 1976.

[H79] G. Heinig, Beitrage zur spektraltheorie von Operatorbuschen und zur algebraischen Theorie von Toeplitzmatrizen, Dissertation **B**, *TH Karl-Marx-Stadt*, 1979.

[H96] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM Publications, Philadelphia, PA, 1996.

[HR84] G. Heinig, K. Rost, *Algebraic Methods for Toeplitz-like Matrices and Operators, Operator Theory*, **13**, Birkhäuser, 1984.

[I79] Y. Ikebe, On Inverses of Hessenberg Matrices, *Linear Algebra and Its Applications*, **24**, 93–97, 1979.

[K90] A. A. Karatsuba, The Distribution of Prime Numbers, *Russian Math. Surveys*, **45**, 99–171, 1990.

[K98] D. E. Knuth, *The Art of Computer Programming. Vol. 2: Seminumerical Algorithms*, Addison-Wesley, Reading, Massachusetts, 1998.

[K04] I. Kaporin, The Aggregation and Cancellation Techniques as a Practical Tool For Faster Matrix Multiplication, *Theoretical Computer Science*, **315**, 469–510, 2004.

[KL96] E. Kaltofen, A. Lobo, On Rank Properties of Toeplitz Matrices over Finite Fields, *Proceedings of International Symposium on Symbolic and Algebraic Computation (ISSAC'96)*, 241–249, ACM Press, New York, 1996.

[KS91] E. Kaltofen, B. D. Saunders, On Wiedemann's Method for Solving Sparse Linear Systems, *Proceedings of AAECC–5, Lecture Notes in Computer Science*, **536**, 29–38, Springer, Berlin, 1991.

[KS99] T. Kailath, A. H. Sayed (editors), *Fast Reliable Algorithms for Matrices with Structure*, SIAM Publications, Philadelphia, PA, 1999.

[KV01] E. Kaltofen, G. Villard. On the Complexity of Computing Determinants. *Proc. Fifth Asian Symposium on Computer Mathematics (ASCM 2001)*, (Shirayanagi, Kiyoshi and Yokoyama, Kazuhiro, editors), *Lecture Notes Series on Computing*, **9**, 13–27, World Scientific, Singapore, 2001.

[KV04] E. Kaltofen, G. Villard. Computing the Sign or the Value of the Determinant of an Integer Matrix, a Complexity Survey. *J. Computational Applied Math.*, **162** (**1**), 133–146, 2004.

[L47] N. Levinson, The Wiener RMS (Root-Mean-Square) Error Criterion in the Filter Design and Prediction, *Journal of Mathematical Physics*, **25**, 261–278, 1947.

[M74] M. Morf, Fast Algorithms for Multivariable Systems, Ph.D. Thesis, *Department of Electrical Engineering, Stanford University,* Stanford, CA, 1974.

[M80] M. Morf, Doubling Algorithms for Toeplitz and Related Equations, *Proceedings of IEEE International Conference on ASSP*, 954–959, IEEE Press, Piscataway, New Jersey, 1980.

[M04] M. Monahan, Maximal Quotient Rational Reconstruction: an Almost Optimal Algorithm for Rational Reconstruction, *Proc. International Symp. on Algebraic and Symbolic Computation (ISSAC'04)*, 243–249, ACM Press, New York, 2004.

[MC79] R. T. Moenck, J. H. Carter, Approximate Algorithms to Derive Exact Solutions to Systems of Linear Equations, *Proceedings of EUROSAM, Lecture Notes in Computer Science*, **72**, 63–73, Springer, Berlin, 1979.

[MS04] T. Mulders, A. Storjohann, Certified Dense Linear System Solving, *J. of Symbolic Computation*, **37(4)**, 485-510, 2004.

[N72] M. Newman., *Integral Matrices*, Academic Press, New York, 1972.

[P87] V. Y. Pan, Complexity of Parallel Matrix Computations, *Theoretical Computer Science*, **54**, 65–85, 1987.

[P88] V. Y. Pan, Computing the Determinant and the Characteristic Polynomials of a Matrix via Solving Linear Systems of Equations, *Information Processing Letters*, **28**, 71–75, 1988.

[P90] V. Y. Pan, On Computations with Dense Structured Matrices, *Mathematics of Computation*, **55(191)**, 179–190, 1990.

[P92] V. Y. Pan, Parametrization of Newton's Iteration for Computations with Structured Matrices and Applications, *Computers and Mathematics (with Applications)*, **24(3)**, 61–75, 1992.

[P92a] V. Y. Pan, Can We Utilize the Cancellation of the Most Significant Digits?, Tech. Report TR–92–061, *The International Computer Science Institute*, Berkeley, California, 1992.

[P96] V. Y. Pan, Parallel Computation of Polynomial GCD and Some Related Parallel Computations over Abstract Fields, *Theoretical Computer Science*, **162(2)**, 173–223, 1996.

[P00] V. Y. Pan, Parallel Complexity of Computations with General and Toeplitz-like Matrices Filled with Integers and Extensions, *SIAM J. Comput.*, **30(4)**, 1080–1125, 2000.

[P01] V. Y. Pan, *Structured Matrices and Polynomials: Unified Superfast Algorithms*, Birkhäuser/Springer, Boston/New York, 2001.

[P02] V. Y. Pan, Can We Optimize Toeplitz/Hankel Computations? *Proc. of the Fifth International Workshop on Computer Algebra in Scientific Computing (CASC'02)*, Yalta, Crimea, Sept. 2002 (E. W. Mayr, V. G. Ganzha, E. V. Vorozhtzov, Editors), 253–264, *Technische Universität München*, Germany, 2002.

[P02a] V. Y. Pan, Nearly Optimal Toeplitz/Hankel Computations, Technical Reports TR 2002 001 and 2002 017, *Ph.D. Program in Computer Science, The Graduate Center of the City University of New York*, New York, 2002.

[P04] V. Y. Pan, Superfast Algorithms for Singular Integer Toeplitz/ Hankel-like Matrices, Technical Reports 2002 002, 2003 004 and 2004 015, *Ph.D. Program in Computer Science, The Graduate Center of the City University of New York*, New York, 2002/2003/2004.

[P04a] V. Y. Pan, On Theoretical and Practical Acceleration of Randomized Computation of the Determinant of an Integer Matrix, preprint, 2004.

[PSA95] V. Y. Pan, I. Sobze, A. Atinkpahoun, On Parallel Computations with Banded Matrices, *Information and Computation*, **120 (2)**, 237–250, 1995.

[PW02] V. Y. Pan, X. Wang, Acceleration of Euclidean Algorithm and Extensions, *Proceedings of the International Symposium on Symbolic and Algebraic Computation (ISSAC'02)*, (Teo Mora editor), 207–213, ACM Press, New York, 2002.

[PW04] V. Y. Pan, X. Wang, On Rational Number Reconstruction and Approximation, *SIAM J. on Computing*, **33(2)**, 502–503, 2004.

[RS62] J. B. Rosser, L. Schoenfeld, Approximate Formulas of Some Functions of Prime Numbers, *Illinois J. of Math.*, **6**, 64–94, 1962.

[S80] R. D. Skeel, Iterative Refinement Implies Numerical Stability for Gaussian Elimination, *Math. of Computation*, **35**, 817–832, 1980.

[S80a] J. T. Schwartz, Fast Probabilistic Algorithms for Verification of Polynomial Identities, *Journal of ACM*, **27(4)**, 701–717, 1980.

[S86] A. Schrijver, *Theory of Linear and Integer Programming*, Wiley, New York, 1986.

[S03] A. Storjohann, High Order Lifting and Integrality Certificaiton, *J. of Symbolic Computation*, **36(3–4)**, 613–648, 2003.

[T94] E. E. Tyrtyshnikov, How Bad Are Hankel Matrices? *Numerische Mathematik*, **67(2)**, 261–269, 1994.

[V04] R. Vandebril, Semiseparable Matrices and the Symmetric Eigenvalue Problem, PhD Thesis, *Katholieke Universiteit Leuven, Departement Computerwetenschappen*, Leuven, Belgium, May 2004.

[W86] D. Wiedemann., Solving Sparse Linear Equations over Finite Fields, *IEEE Trans. Inf. Theory*, IT-**32**, 54–62, 1986.

[WP03] X. Wang, V. Y. Pan, Acceleration of Euclidean Algorithm and Rational Number Reconstruction, *SIAM J. on Computing*, **32(2)**, 548-556, 2003.

[Z79] R. E. Zippel, Probabilistic Algorithms for Sparse Polynomials, *Proceedings of EUROSAM'79, Lecture Notes in Computer Science*, **72**, 216–226, Springer, Berlin, 1979.

[Z93] R. Zippel, *Effective Polynomial Computation*, Kluwer, Boston, 1993.