Faster Inversion and Other Black Box Matrix Computations Using Efficient Block Projections

Wayne Eberly[1], Mark Giesbrecht[2], Pascal Giorgi[2,4], Arne Storjohann[2], Gilles Villard[3]

(1) Department of Computer Science, University of Calgary, Calgary, Alberta, Canada
eberly@cpsc.ucalgary.ca

(2) David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada
mwg@cs.uwaterloo.ca, pgiorgi@cs.uwaterloo.ca, astorjoh@cs.uwaterloo.ca

(3) CNRS, LIP, École Normale Supérieure de Lyon, Lyon, France
Gilles.Villard@ens-lyon.fr

(4) IUT — Université de Perpignan, Perpignan, France
pascal.giorgi@univ-perp.fr

ABSTRACT

Efficient block projections of non-singular matrices have recently been used by the authors in [10] to obtain an efficient algorithm to find rational solutions for sparse systems of linear equations. In particular a bound of $O^{\sim}(n^{2.5})$ machine operations is presented for this computation assuming that the input matrix can be multiplied by a vector with constant-sized entries using $O^{\sim}(n)$ machine operations. Somewhat more general bounds for black-box matrix computations are also derived. Unfortunately, the correctness of this algorithm depends on the existence of efficient block projections of non-singular matrices, and this was only conjectured.

In this paper we establish the correctness of the algorithm from [10] by proving the existence of efficient block projections for arbitrary non-singular matrices over sufficiently large fields. We further demonstrate the usefulness of these projections by incorporating them into existing black-box matrix algorithms to derive improved bounds for the cost of several matrix problems. We consider, in particular, matrices that can be multiplied by a vector using $O^{\sim}(n)$ field operations: We show how to compute the inverse of any such non-singular matrix over any field using an expected number of $O^{\sim}(n^{2.27})$ operations in that field. A basis for the null space of such a matrix, and a certification of its rank, are obtained at the same cost. An application of this technique to Kaltofen and Villard's Baby-Steps/Giant-Steps algorithms for the determinant and Smith Form of an integer matrix is also sketched, yielding algorithms requiring $O^{\sim}(n^{2.66})$ machine operations. More general bounds involving the number of black-box matrix operations to be used are also obtained.

The derived algorithms are all probabilistic of the Las Vegas type. They are assumed to be able to generate random elements — bits or field elements — at unit cost, and always output the correct answer in the expected time given.

Categories and Subject Descriptors

I.1.2 [**Symbolic and Algebraic Manipulation**]: Algorithms—*Algebraic algorithms, analysis of algorithms*

General Terms

Algorithms

Keywords

Sparse integer matrix, structured integer matrix, linear system solving, black box linear algebra

1. INTRODUCTION

In our paper [10] we presented an algorithm which purportedly solved a sparse system of rational equations considerably more efficiently than standard linear equations solving. Unfortunately, its effectiveness in all cases was conjectural, even as its complexity and actual performance were very appealing. This effectiveness relied on a conjecture regarding the existence of so-called efficient block projections. Given a matrix $A \in \mathsf{F}^{n \times n}$ over any field F, these projections should be block vectors $u \in \mathsf{F}^{n \times s}$ (where s is a blocking factor dividing n, so $n = ms$) such that we can compute uv or $v^t u$ quickly for any $v \in \mathsf{F}^{n \times s}$, and such that the sequence of vectors u, Au, ..., $A^{m-1}u$ has rank n. In this paper, we

*This material is based on work supported in part by the French National Research Agency (ANR Gecko, Villard), and by the Natural Sciences and Engineering Research Council (NSERC) of Canada (Eberly, Giesbrecht, Storjohann).

ISSAC'07, July 29–August 1, 2007, Waterloo, Ontario, Canada.

prove the existence of a class of such efficient block projections for non-singular $n \times n$ matrices over sufficiently large fields; we require that the size of the field F exceed $n(n+1)$.

This can be used to establish a variety of results concerning matrices $A \in \mathbb{Z}^{n\times n}$ with efficient matrix-vector products — in particular, such that a matrix-vector product Ax mod p can be computed for a given integer vector x and a small (word-sized) prime p using $O\tilde{}(n)$ bit operations. Such matrices include all "sparse" matrices having $O(n)$ nonzero entries, assuming these are appropriately represented. They also include a variety of "structured" matrices, having constant "displacement rank" (for one definition of displacement rank or another) studied in the recent literature.

In particular, our existence result implies that if $A \in \mathbb{Z}^{n\times n}$ is non-singular and has an efficient matrix-vector product then the Las Vegas algorithm for system solving given in [10] can be used to solve a system $Ax = b$ for a given integer vector b using an expected number of matrix-vector products modulo a word-sized prime that is $O\tilde{}(n^{1.5}\log(\|A\| + \|b\|))$ together with an expected number of additional bit operations that is $O\tilde{}(n^{2.5}\log(\|A\| + \|b\|))$. If A has an efficient matrix-vector product then the total expected number of bit operations used by this algorithm is less than that used by any previously known algorithm, at least when "standard" (i.e., cubic) matrix arithmetic is used.

Consider, for example, the case when the cost of a matrix-vector product by A modulo a word-sized prime is $O\tilde{}(n)$ operations, and the entries in A are constant size. The cost of our algorithm will be $O\tilde{}(n^{2.5})$ bit operations. This improves upon the p-adic lifting method of Dixon [6], which requires $O\tilde{}(n^3)$ bit operations for sparse or dense matrices. This theoretical efficiency was reflected in practice in [10] at least for large matrices.

We present several other rather surprising applications of this technique. Each incorporates the technique into an existing algorithm in order to reduce the asymptotic complexity for the matrix problem to be solved. In particular, given a matrix $A \in \mathsf{F}^{n\times n}$ over an arbitrary field F, we are able to compute the complete inverse of A with $O\tilde{}(n^{3-1/(\omega-1)})$ operations in F plus $O\tilde{}(n^{2-1/(\omega-1)})$ matrix-vector products by A. Here ω is such that we can multiply two $n \times n$ matrices with $O(n^\omega)$ operations in F. Standard matrix multiplication gives $\omega = 3$, while the best known matrix multiplication of Coppersmith and Winograd [5] has $\omega = 2.376$. If again we can compute $v \mapsto Av$ with $O\tilde{}(n)$ operations in F, this implies an algorithm to compute the inverse with $O\tilde{}(n^{3-1/(\omega-1)})$ operations in F. This is always in $O\tilde{}(n^\omega)$, and in particular equals $O\tilde{}(n^{2.27})$ operations in F for the best known ω of [5]. Other relatively straightforward applications of these techniques yield algorithms for the full nullspace and (certified) rank with this same cost. Finally, we sketch how these methods can be employed in the algorithms of Kaltofen and Villard [18] and Giesbrecht [13] to computing the determinant and Smith form of sparse matrices more efficiently.

There has certainly been much important work done on finding exact solutions to sparse rational systems prior to [10]. Dixon's p-adic lifting algorithm [6] performs extremely well in practice for dense and sparse linear systems, and is implemented efficiently in LinBox [7] and Magma (see [10] for a comparison). Kaltofen and Saunders [17] are the first to propose to use Krylov-type algorithms for these problems. Krylov-type methods are used to find Smith forms of sparse matrices and to solve Diophantine systems in parallel in [12, 13], and this is further developed in [8, 18]. See the references in these papers for a more complete history. For sparse systems over a field, the seminal work is that of Wiedemann [22] who shows how to solve sparse $n \times n$ systems over a field with $O(n)$ matrix-vector products and $O(n^2)$ other operations. This research is further developed in [4, 16, 17] and many other works. The bit complexity of similar operations for various families of structured matrices is examined by Emiris and Pan [11].

2. EFFICIENT BLOCK PROJECTIONS

For now we will consider an arbitrary invertible matrix $A \in \mathsf{F}^{n\times n}$ over a field F, and s an integer, the blocking factor, that divides n exactly. Let $m = n/s$. For a so-called *block projection* $u \in \mathsf{F}^{n\times s}$ and $1 \le k \le m$, we denote by $\mathcal{K}_k(A, u)$ the *block Krylov matrix* $[u, Au, \ldots, A^{k-1}u] \in \mathsf{F}^{n\times ks}$. We wish to show that $\mathcal{K}_m(A, u) \in \mathsf{F}^{n\times n}$ is non-singular for a particularly simple and sparse u, assuming some properties of A.

Our factorization uses the special projection (which we will refer to as an *efficient block projection*)

$$u = \begin{bmatrix} I_s \\ \vdots \\ I_s \end{bmatrix} \in \mathsf{F}^{n\times s}, \tag{2.1}$$

which is comprised of m copies of I_s and thus has exactly n non-zero entries. We suggest a similar projection in [10] without proof of its reliability (i.e., that the corresponding block Krylov matrix is non-singular). We establish here that it does yield a block Krylov matrix of full rank, and hence can be used for an efficient inverse of a sparse A.

Let $\mathcal{D} = \mathrm{diag}(\delta_1, \ldots, \delta_1, \delta_2, \ldots, \delta_2, \ldots, \delta_m, \ldots, \delta_m)$ be an $n \times n$ diagonal matrix whose entries consist of m distinct indeterminates δ_i, each δ_i occurring s times.

THEOREM 2.1. *If the leading $ks \times ks$ minor of A is non-zero, for $1 \le k \le m$, then $\mathcal{K}_m(\mathcal{D}A\mathcal{D}, u) \in \mathsf{F}[\delta_1, \ldots, \delta_m]^{n\times n}$ is non-singular.*

PROOF. Let $\mathcal{B} = \mathcal{D}A\mathcal{D}$. For $1 \le k \le m$, define $\mathcal{B}_k$ as the specialization of $\mathcal{B}$ obtained by setting $\delta_{k+1}, \delta_{k+2}, \ldots, \delta_m$ to zero. Thus $\mathcal{B}_k$ is the matrix constructed by setting to zero the last $n - ks$ rows and columns of $\mathcal{B}$. Similarly, for $1 \le k \le m$ we define $u_k \in \mathsf{F}^{n\times s}$ to be the matrix constructed from u by setting to zero the last $n - ks$ rows. In particular we have $\mathcal{B}_m = \mathcal{B}$ and $u_m = u$. This specialization will allow us to argue incrementally about how the rank is increased as k increases.

We proceed by induction on k and show that

$$\mathrm{rank}\, \mathcal{K}_k(\mathcal{B}_k, u_k) = ks, \tag{2.2}$$

for $1 \le k \le m$. For the base case $k = 1$ we have $\mathcal{K}_1(\mathcal{B}_1, u_1) = u_1$ and thus $\mathrm{rank}\, \mathcal{K}_1(\mathcal{B}_1, u_1) = \mathrm{rank}\, u_1 = s$.

Now, assume that (2.2) holds for some k with $1 \le k < m$. By the definition of $\mathcal{B}_k$ and u_k, only the first ks rows of $\mathcal{B}_k$ and u_k will be involved in the left hand side of (2.2). Similarly, only the first ks columns of $\mathcal{B}_k$ will be involved. Since by assumption on $\mathcal{B}$ the leading $ks \times ks$ minor is non-zero, we have $\mathrm{rank}\, \mathcal{B}_k\mathcal{K}_k(\mathcal{B}_k, u_k) = ks$, which is equivalent to $\mathrm{rank}\, \mathcal{K}_k(\mathcal{B}_k, \mathcal{B}_k u_k) = ks$. By the fact that the first ks rows of $u_{k+1} - u_k$ are zero, we have $\mathcal{B}_k(u_{k+1} - u_k) = 0$, or equivalently $\mathcal{B}_k u_{k+1} = \mathcal{B}_k u_k$, and hence

$$\mathrm{rank}\, \mathcal{K}_k(\mathcal{B}_k, \mathcal{B}_k u_{k+1}) = ks. \tag{2.3}$$

The matrix in (2.3) can be written as

$$\mathcal{K}_k(\mathcal{B}_k, \mathcal{B}_k u_{k+1}) = \begin{bmatrix} M_k \\ 0 \end{bmatrix} \in \mathsf{F}^{n \times ks},$$

where $M_k \in \mathsf{F}^{ks \times ks}$ is non-singular. Introducing the block u_{k+1} we obtain the matrix

$$[u_{k+1}, \mathcal{K}_k(\mathcal{B}_k, \mathcal{B}_k u_{k+1})] = \begin{bmatrix} * & M_k \\ I_s & 0 \\ 0 & 0 \end{bmatrix}. \qquad (2.4)$$

whose rank is $(k+1)s$. Noticing that

$$\left[u_{k+1}, \mathcal{K}_k(\mathcal{B}_k, \mathcal{B}_k u_{k+1})\right] = \mathcal{K}_{k+1}\left(\mathcal{B}_k, u_{k+1}\right),$$

we are led to

$$\operatorname{rank} \mathcal{K}_{k+1}(\mathcal{B}_k, u_{k+1}) = (k+1)s.$$

Finally, using the fact that $\mathcal{B}_k$ is the specialization of $\mathcal{B}_{k+1}$ obtained by setting δ_{k+1} to zero, we obtain

$$\operatorname{rank} \mathcal{K}_{k+1}(\mathcal{B}_{k+1}, u_{k+1}) = (k+1)s,$$

which is (2.2) for $k+1$ and thus establishes the theorem by induction. □

If the leading $ks \times ks$ minor of A is non-zero, then the leading $ks \times ks$ minor of A^T is non-zero as well, for any integer k. This gives us the following corollary.

COROLLARY 2.2. *If the leading $ks \times ks$ minor of A is non-zero for $1 \le k \le m$, and $\mathcal{B} = \mathcal{D}A\mathcal{D}$, then $\mathcal{K}_m(\mathcal{B}^T, u)$ is non-singular.*

Suppose now that $A \in \mathsf{F}^{n \times n}$ is an arbitrary non-singular matrix and the size of F exceeds $n(n+1)$. It follows by Theorem 2 of Kaltofen and Saunders [17] that there exists a lower triangular Toeplitz matrix $L \in \mathsf{F}^{n \times n}$ and an upper triangular Toeplitz matrix $U \in \mathsf{F}^{n \times n}$ such that each of the leading minors of $\widehat{A} = UAL$ is non-zero. Let $\mathcal{B} = \mathcal{D}\widehat{A}\mathcal{D}$; the product of the determinants of the matrices $\mathcal{K}_m(\mathcal{B}, u)$ and $\mathcal{K}_m(\mathcal{B}^T, u)$ (mentioned in the above theorem and corollary) is a polynomial with total degree less than $2n(m-1) < n(n+1)$ (if $m \neq 1$). In this case it follows that there is also a non-singular diagonal matrix $D \in \mathsf{F}^{n \times n}$ such that $\mathcal{K}_m(B, u)$ and $\mathcal{K}_m(B^T, u)$ are non-singular, for

$$B = D\widehat{A}D = DUALD.$$

Now let $R = LD^2U \in \mathsf{F}^{n \times n}$, $\widehat{u} \in \mathsf{F}^{s \times n}$ and $\widehat{v} \in \mathsf{F}^{n \times s}$ such that

$$\widehat{u}^T = (L^T)^{-1} D^{-1} u \quad \text{and} \quad \widehat{v} = LDu.$$

Then

$$\mathcal{K}_m(RA, \widehat{v}) = LD\mathcal{K}_m(B, u)$$

and

$$L^T D \mathcal{K}_m((RA)^T, \widehat{u}^T) = \mathcal{K}_m(B^T, u),$$

so that $\mathcal{K}_m(RA, \widehat{v})$ and $\mathcal{K}_m((RA)^T, \widehat{u}^T)$ are each non-singular as well. Because D is diagonal and U and L are triangular Toeplitz matrices, it is now easily established that $(R, \widehat{u}, \widehat{v})$ is an *efficient block projection* for the given matrix A, where such projections are as defined in [10].

This proves Conjecture 2.1 of [10] for the case that the size of F exceeds $n(n+1)$:

COROLLARY 2.3. *For any non-singular $A \in \mathsf{F}^{n \times n}$ and $s \mid n$ (over a field of size greater than $n(n+1)$) there exists an efficient block projection $(R, u, v) \in \mathsf{F}^{n \times n} \times \mathsf{F}^{s \times n} \times \mathsf{F}^{n \times s}$.*

3. FACTORIZATION OF THE MATRIX INVERSE

The existence of the efficient block projection established in the previous section allows us to define a useful factorization of the inverse of a matrix. This was used to obtain faster heuristics for solving integer systems in [10]. The basis is the following factorization of the matrix inverse.

Let $\mathcal{B} = \mathcal{D}A\mathcal{D}$, where $\mathcal{D}$ is an $n \times n$ diagonal matrix whose diagonal entries consist of m distinct indeterminates, each occurring s times contiguously, as previously defined. Define $\mathcal{K}_u^{(r)} = \mathcal{K}_m(\mathcal{B}, u)$ with u as in (2.1) and $\mathcal{K}_u^{(\ell)} = \mathcal{K}_m(\mathcal{B}^T, u)^T$, where (r) and (ℓ) refer to projection on the right and left respectively. For any $0 \le k \le m-1$ and any two indices l and r such than $l + r = k$ we have $u^T \mathcal{B}^l \cdot \mathcal{B}^r u = u^T \mathcal{B}^k u$. Hence the matrix $\mathcal{H}_u = \mathcal{K}_u^{(\ell)} \cdot \mathcal{B} \cdot \mathcal{K}_u^{(r)}$ is block-Hankel with blocks of dimension $s \times s$:

$$\mathcal{H}_u = \begin{bmatrix} u^T\mathcal{B}u & u^T\mathcal{B}^2u & \cdots & u^T\mathcal{B}^m u \\ u^T\mathcal{B}2u & u^T\mathcal{B}^3u & \iddots & \vdots \\ \vdots & \iddots & & u^T\mathcal{B}^{2m-2}u \\ u^T\mathcal{B}^m u & \cdots & u^T\mathcal{B}^{2m-2}u & u^T\mathcal{B}^{2m-1}u \end{bmatrix}$$

Notice that $\mathcal{H}_u = \mathcal{K}_u^{(\ell)} \cdot \mathcal{B} \cdot \mathcal{K}_u^{(r)} = \mathcal{K}_u^{(\ell)} \cdot \mathcal{D}A\mathcal{D} \cdot \mathcal{K}_u^{(r)}$. Theorem 2.1 and Corollary 2.2 imply that if all leading $ks \times ks$ minors of A are non-singular then $K_u^{(\ell)}$ and $K_u^{(r)}$ are each non-singular as well. This establishes the following.

THEOREM 3.1. *If $A \in \mathsf{F}^{n \times n}$ is such that all leading $ks \times ks$ minors are non-singular, $\mathcal{D}$ is a diagonal matrix of indeterminates, and $\mathcal{B} = \mathcal{D}A\mathcal{D}$, then $\mathcal{B}^{-1}$ and A^{-1} may be factored as*

$$\begin{aligned} \mathcal{B}^{-1} &= \mathcal{K}_u^{(r)} \mathcal{H}_u^{-1} \mathcal{K}_u^{(\ell)}, \\ A^{-1} &= \mathcal{D}\mathcal{K}_u^{(r)} \mathcal{H}_u^{-1} \mathcal{K}_u^{(\ell)} \mathcal{D}, \end{aligned} \qquad (3.1)$$

where $\mathcal{K}_u^{(\ell)}$ and $\mathcal{K}_u^{(r)}$ are as defined above, and $\mathcal{H}_u \in \mathsf{F}^{n \times n}$ is block-Hankel (and invertible) with $s \times s$ blocks, as above.

Note that for any specialization of the indeterminates in $\mathcal{D}$ to field elements in F such that $\det \mathcal{H}_u \neq 0$ we obtain a similar formula to (3.1) completely over F. A similar factorization in the non-blocked case is used in [9, (4.5)] for fast parallel matrix inversion.

4. BLACK-BOX MATRIX INVERSION OVER A FIELD

Suppose again that $A \in \mathsf{F}^{n \times n}$ is invertible, and that for any $v \in \mathsf{F}^{n \times 1}$ the products Av and $A^T v$ can be computed in $\phi(n)$ operations in F (where $\phi(n) \ge n$). Following Kaltofen, we call such matrix-vector and vector-matrix products *black-box* evaluations of A. In this section we will show how to compute A^{-1} with $O^{\sim}(n^{2-1/(\omega-1)})$ black box evaluations and $O^{\sim}(n^{3-1/(\omega-1)})$ additional operations in F. Note that when $\phi(n) = O^{\sim}(n)$ the exponent in n of this cost is smaller than ω, and is $O^{\sim}(n^{2.273})$ with the currently best-known matrix multiplication.

Again assume that $n = ms$, where s is a *blocking factor* and m the number of blocks. Assume for the moment that all principal $ks \times ks$ minors of A are non-zero, $1 \le k \le m$.

Let $\delta_1, \delta_2, \ldots, \delta_m$ be the indeterminates that form the diagonal entries of $\mathcal{D}$ and let $\mathcal{B} = \mathcal{D}A\mathcal{D}$. By Theorem 2.1 and Corollary 2.2, the matrices $\mathcal{K}_m(\mathcal{B}, u)$ and $\mathcal{K}_m(\mathcal{B}^T, u)$ are each invertible. If $m \ge 2$ then the product of the determinants of these matrices is a non-zero polynomial $\Delta \in \mathsf{F}[\delta_1, \ldots, \delta_m]$ with total degree at most $2n(m-1)$.

Suppose that F has at least $2n(m-1)$ elements. Then Δ cannot be zero at all points in $(\mathsf{F} \setminus \{0\})^n$. Let $d_1, d_2, \ldots, d_m$ be non-zero elements of F such that $\Delta(d_1, d_2, \ldots, d_m) \neq 0$, let $D = \mathrm{diag}(d_1, \ldots, d_1, \ldots, d_m, \ldots, d_m)$, and let $B = DAD$. Then $K_u^{(r)} = \mathcal{K}_m(B, u) \in \mathsf{F}^{n\times n}$ and $K_u^{(\ell)} = \mathcal{K}_m(B^T, u)^T \in \mathsf{F}^{n\times n}$ are each invertible because $\Delta(d_1, d_2, \ldots, d_m) \neq 0$, B is invertible because A is and $d_1, d_2, \ldots, d_m$ are all non-zero, and thus $H_u = K_u^{(\ell)} B K_u^{(r)} \in \mathsf{F}^{n\times n}$ is invertible as well. Correspondingly, (3.1) suggests

$$B^{-1} = K_u^{(r)} H_u^{-1} K_u^{(\ell)}, \quad \text{and} \quad A^{-1} = D K_u^{(r)} H_u^{-1} K_u^{(\ell)} D$$

for computing the matrix inverse.

1. **Computation of u^T, $u^T B$, …, $u^T B^{2m-1}$ and $K_u^{(\ell)}$.**
   We can compute this sequence, hence $K_u^{(r)}$, with $m-1$ applications of B to vectors using $O(n\phi(n))$ operations in F.

2. **Computation of H_u.**
   Due to the special form (2.1) of u, one may then compute wu for any $w \in \mathsf{F}^{s\times n}$ with $O(sn)$ operations. Hence we can now compute $u^T B^i u$ for $0 \le i \le 2m-1$ with $O(n^2)$ operations in F.

3. **Computation of H_u^{-1}.**
   The off-diagonal inverse representation of H_u^{-1} as in (A.4) in the Appendix can be found with $O^{\sim}(s^\omega m)$ operations by Proposition A.1.

4. **Computation of $H_u^{-1} K_u^{(\ell)}$.**
   From Corollary A.2 in the Appendix, we can compute the product $H_u^{-1} M$ for *any* matrix $M \in \mathsf{F}^{n\times n}$ with $O^{\sim}(s^\omega m^2)$ operations.

5. **Computation of $K_u^{(r)} \cdot (H_u^{-1} K_u^{(\ell)})$.**
   We can compute $K_u^{(r)} M = [u, Bu, \ldots, B^{m-1}u]M$, for *any* $M \in \mathsf{F}^{n\times n}$ by splitting M into m blocks of s consecutive rows M_i, for $0 \le i \le m-1$:

$$\begin{aligned} K_u M &= \sum_{i=0}^{m-1} B^i (uM_i) \\ &= uM_0 + B(uM_1 + B(uM_2 + \cdots \\ &\qquad \cdots + B(uM_{m-2} + BuM_{m-1}) \cdots). \end{aligned} \tag{4.1}$$

   Because of the special form (2.1) of u, each product $uM_i \in \mathsf{F}^{n\times n}$ requires $O(n^2)$ operations, and hence all such products involved in (4.1) can be computed in $O(mn^2)$ operations. Because applying B to an $n \times n$ matrix costs $n\phi(n)$ operations, $K_u^{(r)} M$ is computed in $O(mn\phi(n) + mn^2)$ operations using the iterative form of (4.1)

In total, the above process requires $O(mn)$ applications of A to a vector (the same as for B), and $O(s^\omega m^2 + mn^2)$ additional operations. If $\phi(n) = O^{\sim}(n)$, the overall number of field operations is minimized with the blocking factor $s = n^{1/(\omega-1)}$.

THEOREM 4.1. *Let $A \in \mathsf{F}^{n\times n}$, where $n = ms$ and $s = n^{1/(\omega-1)}$, be such that all leading $ks \times ks$ minors are non-singular for $1 \le k \le m$. Let $B = DAD$, for $D = \mathrm{diag}(d_1, \ldots, d_1, \ldots, d_m, \ldots, d_m)$, such that $d_1, \ldots, d_m$ are non-zero and each of the matrices $\mathcal{K}_m(DAD, u)$ and $\mathcal{K}_m((DAD)^T, u)$ is invertible. Then the inverse matrix A^{-1} can be computed using $O(n^{2-1/(\omega-1)})$ black box operations and an additional $O^{\sim}(n^{3-1/(\omega-1)})$ operations in F.*

The above discussion makes a number of assumptions.

First, it assumes that the blocking factor s exactly divides n. This is easily accommodated by simply extending n to the nearest multiple of s, placing A in the top left corner of the augmented matrix, and adding diagonal ones in the bottom right corner.

Theorem 4.1 also makes the assumptions that all the leading $ks \times ks$ minors of A are non-singular and that the determinants of $\mathcal{K}_m(DAD, u)$ and $\mathcal{K}_m((DAD)^T, u)$ are each non-zero. Although we know of no way to ensure this deterministically in the times given, standard techniques can be used to obtain these properties probabilistically if F is sufficiently large.

Suppose, in particular, that $n \ge 16$ and that $\#\mathsf{F} > 2(m+1)n\lceil \log_2 n \rceil$. Fix a set $\mathcal{S}$ of at least $2(m+1)n\lceil \log_2 n \rceil$ non-zero elements of F. We can ensure that the leading $ks \times ks$ minors of A are non-zero by pre- and post-multiplying by butterfly network preconditioners X and Y respectively, with parameters chosen uniformly and randomly from $\mathcal{S}$. If X and Y are constructed using the generic exchange matrix of [4, §6.2], then it will use at most $n\lceil \log_2 n \rceil/2$ random elements from S, and from [4, Theorem 6.3] it follows that all leading $ks \times ks$ minors of $\widetilde{A} = XAY$ will be non-zero simultaneously with probability at least 3/4. This probability of success can be made arbitrarily close to 1 with a choice from a larger $\mathcal{S}$. We note that $A^{-1} = Y\widetilde{A}^{-1}X$. Thus, once we have computed $\widetilde{A}^{-1}$ we can compute A^{-1} with an additional $O^{\sim}(n^2)$ operations in F, using the fact that multiplication of an arbitrary $n \times n$ matrix by an $n \times n$ butterfly preconditioner can be done with $O^{\sim}(n^2)$ operations.

Once again let Δ be the products of the determinants of the matrices $\mathcal{K}_m(\mathcal{D}A\mathcal{D}, u)$ and $\mathcal{K}_m((\mathcal{D}A\mathcal{D})^T, u)$, so that Δ is non-zero with total degree at most $2n(m-1)$. If we choose randomly selected values from $\mathcal{S}$ for $\delta_1, \ldots, \delta_m$, because $\#\mathcal{S} \ge 2(m+1)n\lceil \log_2 n \rceil > 4 \deg \Delta$ the probability that Δ is zero at this point is at most 1/4 by the Schwartz-Zippel Lemma [21, 23].

In summary, for randomly selected butterfly preconditioners X, Y as above, and independently and randomly chosen values $d_1, d_2, \ldots, d_m$ the probability that $\widetilde{A} = XAY$ has non-singular leading $ks \times ks$ minors for $1 \le k \le m$ *and* $\Delta(d_1, d_2, \ldots, d_m)$ is non-zero is at least $9/16 > 1/2$ when random choices are made uniformly and independently from a finite subset $\mathcal{S}$ of $\mathsf{F}\setminus\{0\}$ with size at least $2(m+1)n\lceil \log_2 n \rceil$.

When $\#\mathsf{F} \le 2(m+1)n\lceil \log_2 n \rceil$, we can easily construct a field extension E of F that has size greater than $2(m+1)n\lceil \log_2 n \rceil$ and perform the computation in that extension. Because this extension will have degree $O(\log_{\#\mathsf{F}} n)$ over F, it will add only a logarithmic factor to the final cost. While we certainly do not claim that this is not of practical concern, it does not affect the asymptotic complexity.

This algorithm is Las Vegas (or trivially modified to be so): For if either $\mathcal{K}_m(\mathcal{DAD}, u)$ or $\mathcal{K}_m((\mathcal{DAD})^T, u)$ is singular then so is H_u and this is detected at step 3. On the other hand, if $\mathcal{K}_m(\mathcal{DAD}, u)$ and $\mathcal{K}_m((\mathcal{DAD})^T, u)$ are both non-singular then the algorithm's output is correct.

THEOREM 4.2. *Let $A \in \mathsf{F}^{n\times n}$ be non-singular. Then the inverse matrix A^{-1} can be computed by a Las Vegas algorithm whose expected cost is $O^{\sim}(n^{2-1/(\omega-1)})$ black box operations and $O^{\sim}(n^{3-1/(\omega-1)})$ additional operations in F.*

Table 4.1 (below) states the expected costs to compute the inverse using various values of ω when $\phi(n) = O^{\sim}(n)$.

ω	*Black-box applications*	*Blocking factor*	*Inversion cost*
3 (Standard)	1.5	$n^{1/2}$	$O^{\sim}(n^{2.5})$
2.807 (Strassen)	1.446	$n^{0.553}$	$O^{\sim}(n^{2.446})$
2.3755 (Cop/Win)	1.273	$n^{0.728}$	$O^{\sim}(n^{2.273})$

Table 4.1: Exponents of matrix inversion with a matrix $\times$ vector cost $\phi(n) = O^{\sim}(n)$.

REMARK 4.3. *The structure (2.1) of the projection u plays a central role in computing the product of the block Krylov matrix by a $n\times n$ matrix. For a general projection $u \in \mathsf{F}^{n\times s}$, how to do better than a general matrix multiplication, i.e., how to take advantage of the Krylov structure for computing $K_u M$, appears to be unknown.*

Multiplying a Black-Box Matrix Inverse By Any Matrix

The above method can also be used to compute $A^{-1}M$ for *any* matrix $M \in \mathsf{F}^{n\times n}$ with the same cost as in Theorem 4.2. Consider the new step 1.5:

1.5. Computation of $K_u^{(\ell)} \cdot M$.
Split M into m blocks of s columns, so that $M = [M_0, \ldots, M_{m-1}]$ where $M_k \in \mathsf{F}^{n\times s}$. Now consider computing $K_u^{(\ell)} \cdot M_k$ for some $k \in \{0, \ldots, m-1\}$. This can be accomplished by computing $B^i M_k$ for $0 \le i \le m-1$ in sequence, and then multiplying on the left by u^T to compute $u^T B^i M_k$ for each iterate.

The cost for computing $K_u^{(\ell)} M_k$ for a single k by the above process is $n - s$ multiplication of A to vectors and $O(ns)$ additional operations in F. The cost of doing this for all k such that $0 \le k \le m-1$ is thus $m(n-s) < nm$ multiplications of A to vectors and $O(n^2)$ additional operations. Since applying A (and hence B) to an $n \times n$ matrix is assumed to cost $n\phi(n)$ operations in F, $K_u^{(\ell)} \cdot M$ is computed in $O(mn\phi(n) + mn^2)$ operations in F by the process described here.

Note that this is the same as the cost of Step 5, so the overall cost estimate is not affected. Because Step 4 does not rely on any special form for $K_u^{(\ell)}$, we can replace it with a computation of $H_u^{-1} \cdot (K_u^{(\ell)} M)$ with the same cost. The output is again easily certified with n additional black-box evaluations. We obtain the following corollary.

COROLLARY 4.4. *Let $A \in \mathsf{F}^{n\times n}$ be non-singular and let $M \in \mathsf{F}^{n\times n}$. We can compute $A^{-1}M$ with a Las Vegas algorithm whose expected cost is $O^{\sim}(n^{2-1/(\omega-1)})$ black box operations and $O^{\sim}(n^{3-1/(\omega-1)})$ additional operations in F.*

The estimates in Table 4.1 apply to this computation as well.

5. APPLICATIONS TO BLACK-BOX MATRICES OVER A FIELD

The algorithms of the previous section have applications in some important computations with black-box matrices over an arbitrary field F. In particular, we consider the problems of computing the nullspace and rank of a black-box matrix. Each of these algorithms is probabilistic of the Las Vegas type; the output is certified to be correct.

Kaltofen and Saunders [17] present algorithms for computing the rank of a matrix and for randomly sampling the nullspace, building upon the work of Wiedemann [22]. In particular, they show for random lower upper and lower triangular Toeplitz matrices $U, L \in \mathsf{F}^{n\times n}$, and random diagonal D, that all leading $k \times k$ minors of $\widetilde{A} = UALD$ are non-singular for $1 \le k \le r = \operatorname{rank} A$, and that if $f^{\widetilde{A}} \in \mathsf{F}[x]$ is the minimal polynomial of $\widetilde{A}$, then it has degree $r+1$ if A is singular (and degree n if A is non-singular). This is proved to be true for any input $A \in \mathsf{F}^{n\times n}$, and for random choice of U, L and D, with high probability. The cost of computing $f^{\widetilde{A}}$ (and hence $\operatorname{rank} A$) is shown to be $O(n)$ applications of the black-box for A and $O(n^2)$ additional operations in F. However, no certificate is provided that the rank is correct within this cost (and we do not know of one or provide one here). Kaltofen and Saunders [17] also show how to generate a vector uniformly and randomly from the nullspace of A with this cost (and, of course, this is certifiable with a single evaluation of the black box for A). We also note that the algorithms of Wiedemann and Kaltofen and Saunders require only a linear amount of extra space, which will not be the case for our algorithms.

We first employ the random preconditioning of [17] and let $\widetilde{A} = UALD$ as above. *We will thus assume in what follows that A has all leading $i\times i$ minors non-singular for $1 \le i \le r$.* Although an unlucky choice may make this statement false, this case will be identified in our method. Also assume that we have computed the rank r of A with high probability. Again, this will be certified in what follows.

1. Inverting the leading minor.
Let A_0 be the leading $r \times r$ minor of A and partition A as

$$A = \begin{pmatrix} A_0 & A_1 \\ A_2 & A_3 \end{pmatrix}.$$

Using the algorithm of the previous section, compute A_0^{-1}. If this fails, and the leading $r \times r$ minor is singular, then either the randomized conditioning or the rank estimate has failed and we either report this failure or try again with a different randomized preconditioning. If we can compute A_0^{-1}, then the rank of A is at least the estimated r.

2. Applying the inverted leading minor.
Compute $A_0^{-1}A_1 \in \mathsf{F}^{r\times(n-r)}$ using the algorithm of the previous section (this could in fact be merged into the first step).

3. Confirming the nullspace.
Note that

$$\begin{pmatrix} A_0 & A_1 \\ A_2 & A_3 \end{pmatrix} \underbrace{\begin{pmatrix} A_0^{-1}A_1 \\ -I \end{pmatrix}}_{N} = \begin{pmatrix} 0 \\ A_2A_0^{-1}A_1 - A_3 \end{pmatrix} = 0,$$

and the Schur complement $A_2A_0^{-1}A_1 - A_3$ must be zero if the rank r is correct. This can be checked with $n-r$ evaluations of the black box for A. We note that because of its structure, $N = \begin{pmatrix} A_0^{-1}A_1 \\ -I \end{pmatrix}$ has rank $n-r$.

4. Output rank and nullspace basis.
If the Schur complement is zero, then output the rank r and N, whose columns give a basis for the nullspace of A. Otherwise, output "fail" (and possibly retry with a different randomized pre-conditioning).

THEOREM 5.1. *Let $A \in \mathsf{F}^{n\times n}$ have rank r. Then a basis for the nullspace of A and rank r of A can be computed with an expected number of $O^{\sim}(n^{2-1/(\omega-1)})$ applications of A to a vector, plus an additional expected number of $O^{\sim}(n^{3-1/(\omega-1)})$ operations in F. The algorithm is probabilistic of the Las Vegas type.*

6. APPLICATIONS TO SPARSE RATIONAL LINEAR SYSTEMS

Given a non-singular $A \in \mathbb{Z}^{n\times n}$ and $b \in \mathbb{Z}^{n\times 1}$, in [10] we presented an algorithm and implementation to compute $A^{-1}b$ with $O^{\sim}(n^{1.5}(\log(\|A\| + \|b\|)))$ matrix-vector products $v \mapsto A \bmod p$ for a machine-word sized prime p and any $v \in \mathbb{Z}_p^{n\times 1}$ plus $O^{\sim}(n^{2.5}(\log(\|A\|+\|b\|)))$ additional bit-operations. Assuming that A and b had constant sized entries, and that a matrix-vector product by A mod p could be performed with $O^{\sim}(n)$ operations modulo p, the algorithm presented could solve a system with $O^{\sim}(n^{2.5})$ bit operations. Unfortunately, this result was conditional upon the unproven Conjecture 2.1 of [10]: the existence of an efficient block projection. This conjecture was established in Corollary 2.3 of the current paper. We can now unconditionally state the following theorem.

THEOREM 6.1. *Given any invertible $A \in \mathbb{Z}^{n\times n}$ and $b \in \mathbb{Z}^{n\times 1}$, we can compute $A^{-1}b$ using a Las Vegas algorithm. The expected number of matrix-vector products $v \mapsto Av \bmod p$ is in $O^{\sim}(n^{1.5}(\log(\|A\| + \|b\|)))$, and the expected number of additional bit-operations used by this algorithm is in $O^{\sim}(n^{2.5}(\log(\|A\| + \|b\|)))$.*

Sparse Integer Determinant and Smith Form

The efficient block projection of Theorem 2.1 can also be employed relatively directly into the block baby-steps/giant-steps methods of [18] for computing the determinant of an integer matrix. This will yield improved algorithms for the determinant and Smith form of a sparse integer matrix. Unfortunately, the new techniques do not obviously improve the asymptotic cost of their algorithms in the case for which they were designed, namely, for computations of the determinants of dense integer matrices.

We only sketch the method for computing the determinant here following the algorithm in Section 4 of [18], and estimate its complexity. Throughout we assume that $A \in \mathbb{Z}^{n\times n}$ is non-singular and assume that we can compute $v \mapsto Av$ with $\phi(n)$ *integer* operations, where the bit-lengths of these integers are bounded by $O^{\sim}(\log(n + \|v\| + \|A\|))$.

1. Preconditioning and setup.
Precondition $A \leftarrow B = D_1UAD_2$, where D_1, D_2 are random diagonal matrices, and U is a unimodular preconditioner from [22, §5]. While we will not provide the detailed analysis here, selecting coefficients for these randomly from a set S_1 of size n^3 is sufficient to ensure a high probability of success. This preconditioning will ensure that all leading minors are non-singular and that the characteristic polynomial is squarefree with high probability (see [4] Theorem 4.3 for a proof of the latter condition). From Theorem 2.1, we also see that $\mathcal{K}_m(B, u)$ has full rank with high probability.

Let p be a prime that is larger than the a priori bound on the coefficients of the characteristic polynomial of A; this is easily determined to be $(n \log \|A\|)^{n+o(1)}$. Fix a blocking factor s to be optimized later, and assume $n = ms$.

2. Choosing projections.
Let $u \in \mathbb{Z}^{n\times s}$ be an efficient block projection as in (2.1) and $v \in \mathbb{Z}^{n\times s}$ a random (dense) block projection with coefficients chosen from a set S_2 of size at least $2n^2$.

3. Forming the sequence $\alpha_i = uA^iv \in \mathbb{Z}^{s\times s}$.
Compute this sequence for $i = 0 \dots 2m$. Computing all the A^iv takes $O^{\sim}(n\phi(n) \cdot m \log \|A\|)$ bit operations. Computing all the uA^iv takes $O^{\sim}(n^2 \cdot m \log \|A\|)$ bit operations.

4. Computing the minimal matrix generator.
The minimal matrix generator $F(\lambda)$ modulo p can be computed from the initial sequence segment $\alpha_0, \dots, \alpha_{2m-1}$. See [18, §4]. This can be accomplished with $O^{\sim}(ms^\omega \cdot n \log \|A\|)$ bit operations.

5. Extracting the determinant.
Following the algorithm in [18, §4], we first check if its degree is less than n and if so, return "failure". Otherwise, we know $\det F^A(\lambda) = \det(\lambda I - A)$. Return $\det A = \det F(0) \bmod p$.

The correctness of the algorithm, and specifically the block projections, follows from fact that $[u, Au, \dots, A^{m-1}u]$ is of full rank with high probability by Theorem 2.1. Because the projection v is dense, the analysis of [18, (2.6)] is applicable, and the minimal generating polynomial will have full degree m with high probability, and hence its determinant at $\lambda = 0$ will be the determinant of A.

The total cost of this algorithm is $O^{\sim}((n\phi(n)m + n^2m + nms^\omega) \log \|A\|)$ bit operations, which is minimized when $s = n^{1/\omega}$. This yields an algorithm for the determinant which requires $O^{\sim}((n^{2-1/\omega}\phi(n) + n^{3-1/\omega}) \log \|A\|)$ bit operations. This is probably most interesting when $\omega = 3$, where it yields an algorithm for determinant that requires $O^{\sim}(n^{2.66} \log \|A\|)$ bit operations on a matrix with pseudo-linear cost matrix-vector product.

We also note that a similar approach allows us to use the Monte Carlo Smith form algorithm of [13], which is computed by means of computing the characteristic polynomial of random preconditionings of a matrix. This reduction is

explored in [18] in the dense matrix setting. The upshot is that we obtain the Smith form with the same order of complexity, to within a poly-logarithmic factor, as we have obtained the determinant using the above techniques. See [18, §7.1] and [13] for details. We make no claim that this is practical in its present form.

Note: A referee has indicated that a "lifting" algorithm of Pan et al [20] can also be used to solve integer systems when efficient matrix-vector products (modulo small primes) are supported for both the coefficient matrix and its inverse. This would provide an alternate application of our central results to solve integer systems. We wish to thank the referee for this information.

APPENDIX

A. APPLYING THE INVERSE OF A BLOCK-HANKEL MATRIX

In this appendix we address asymptotically fast techniques for computing a representation of the inverse of a block Hankel matrix, for applying this inverse to an arbitrary matrix. The fundamental technique we will employ is to use the off-diagonal inversion formula of Beckermann & Labahn [1] and its fast variants [14]. An alternative to using the inversion formula would be to use the generalization of the Levinson-Durbin algorithm in [16].

Again assume $n = ms$ for integers m and s, and let

$$H = \begin{bmatrix} \alpha_0 & \alpha_1 & \cdots & \alpha_{m-1} \\ \alpha_1 & \alpha_2 & \iddots & \vdots \\ \vdots & \iddots & & \alpha_{2m-2} \\ \alpha_{m-1} & \cdots & \alpha_{2m-2} & \alpha_{2m-1} \end{bmatrix} \in \mathsf{F}^{n\times n} \tag{A.1}$$

be a non-singular block-Hankel matrix whose blocks are $s \times s$ matrices over F, and let α_{2m} be arbitrary in $\mathsf{F}^{s\times s}$. We follow the approach of [19] for computing the inverse matrix H^{-1}. Since H is invertible, the following four linear systems (see [19, (3.8)-(3.11)])

$$\begin{aligned} &H\,[q_{m-1}, \cdots, q_0]^t = [0, \cdots, 0, I] \in \mathsf{F}^{n\times s}, \\ &H\,[v_m, \cdots, v_1]^t = -\,[\alpha_m, \cdots \alpha_{2m-1}\alpha_{2m}] \in \mathsf{F}^{n\times s}, \end{aligned} \tag{A.2}$$

and

$$\begin{aligned} &[q_{m-1}^* \;\; \cdots \;\; q_0^*]\,H = [0 \;\; \cdots \;\; 0 \;\; I] \in \mathsf{F}^{s\times n}, \\ &[v_m^* \;\; \cdots \;\; v_1^*]\,H = -\,[\alpha_m \;\; \cdots \;\; \alpha_{2m-1} \;\; \alpha_{2m}] \in \mathsf{F}^{s\times n}, \end{aligned} \tag{A.3}$$

have unique solutions given by the $q_k, q_k^* \in \mathsf{F}^{s\times s}$, (for $0 \le k \le m-1$), and the $v_k, v_k^* \in \mathsf{F}^{s\times s}$ (for $1 \le k \le m$). We then obtain the following equation (see [19, Theorem 3.1]):

$$\begin{aligned} H^{-1} = &\begin{bmatrix} v_{m-1} & \cdots & v_1 & I \\ \vdots & \iddots & \iddots & \\ v_1 & \iddots & & \\ I & & & \end{bmatrix} \begin{bmatrix} q_{m-1}^* & \cdots & q_0^* \\ & \ddots & \vdots \\ & & q_{m-1}^* \end{bmatrix} \\ &- \begin{bmatrix} q_{m-2} & \cdots & q_0 & 0 \\ \vdots & \iddots & \iddots & \\ q_0 & \iddots & & \\ 0 & & & \end{bmatrix} \begin{bmatrix} v_m^* & \cdots & v_1^* \\ & \ddots & \vdots \\ & & v_m^* \end{bmatrix}. \end{aligned} \tag{A.4}$$

The linear systems (A.2) and (A.3) may also be formulated in terms of matrix Padé approximation problems. We associate to H the matrix polynomial $A = \sum_{i=0}^{2m} \alpha_i x^i \in \mathsf{F}^{s\times s}[x]$. The $s \times s$ matrix polynomials Q, P, Q^*, P^* in $\mathsf{F}^{s\times s}[x]$ that satisfy

$$\begin{aligned} &A(x)Q(x) \equiv P(x) + x^{2m-1} \bmod x^{2m}, \\ &\quad \text{where } \deg Q \le m-1 \text{ and } \deg P \le m-2, \\ &Q^*(x)A(x) \equiv P^*(x) + x^{2m-1} \bmod x^{2m}, \\ &\quad \text{where } \deg Q^* \le m-1 \text{ and } \deg P^* \le m-2 \end{aligned} \tag{A.5}$$

are unique and provide the coefficients $Q = \sum_{i=0}^{m-1} q_i x^i$ and $Q^* = \sum_{i=0}^{m-1} q_i^* x^i$ for constructing H^{-1} using (A.4) (see [19, Theorem 3.1]). The notation "mod x^i" for $i \ge 0$ indicates that the terms of degree i or higher are ignored. The $s \times s$ matrix polynomials V, U, V^*, U^* in $\mathsf{F}^{s\times s}[x]$ that satisfy

$$\begin{aligned} &A(x)V(x) \equiv U(x) \bmod x^{2m+1},\ V(0) = I, \\ &\quad \text{where } \deg V \le m \text{ and } \deg U \le m-1, \\ &V^*(x)A(x) \equiv U^*(x) \bmod x^{2m+1},\ V^*(0) = I, \\ &\quad \text{where } \deg Q^* \le m-1 \text{ and } \deg P^* \le m-2, \end{aligned} \tag{A.6}$$

are unique and provide the coefficients $V = 1 + \sum_{i=1}^{m} v_i x^i$ and $Q^* = 1 + \sum_{i=1}^{m} v_i^* x^i$ for (A.4).

Using the matrix Padé formulation, the matrices Q, Q^*, V, and V^* may be computed using the σ-basis algorithm in [1], or its fast counterpart in [14, §2.2] that uses fast matrix multiplication. For solving (A.5), the σ-basis algorithm with $\sigma = s(2m-1)$ solves

$$\begin{aligned} &[A \;\; -I] \begin{bmatrix} \overline{Q} \\ \overline{P} \end{bmatrix} \equiv R x^{2m-1} \bmod x^{2m}, \\ &[\overline{Q}^* \;\; \overline{P}^*] \begin{bmatrix} A \\ -I \end{bmatrix} \equiv R^* x^{2m-1} \bmod x^{2m}, \end{aligned}$$

with $\overline{Q}, \overline{P}, \overline{Q}^*, \overline{P}^* \in \mathsf{F}^{s\times s}[x]$ that satisfy the degree constraints $\deg \overline{Q} \le m-1, \deg \overline{Q}^* \le m-1$, and $\deg \overline{P} \le m-2, \deg \overline{P}^* \le m-2$. The residue matrices R and R^* in $\mathsf{F}^{s\times s}$ are non-singular, hence $Q = \overline{Q}R^{-1}$ and $Q^* = (R^*)\overline{Q}^*$ are solutions $\overline{Q}$ and $\overline{Q}^*$ for applying the inversion formula (A.4). For (A.6), the σ-basis algorithm with $\sigma = s(2m+1)$ leads to

$$\begin{aligned} &[A \;\; -I] \begin{bmatrix} \overline{V} \\ \overline{U} \end{bmatrix} \equiv 0 \bmod x^{2m+1}, \\ &[\overline{V}^* \;\; \overline{U}^*] \begin{bmatrix} A \\ -I \end{bmatrix} \equiv 0 \bmod x^{2m+1} \end{aligned}$$

with $\deg \overline{V} \le m$, $\deg \overline{V}^* \le m$, and $\deg \overline{U} \le m-1$, $\deg \overline{U}^* \le m-1$. The constant terms $\overline{V}(0)$ and $\overline{V}^*(0)$ in $\mathsf{F}^{s\times s}$ are non-singular, hence $V = \overline{V}(\overline{V}(0))^{-1}$ and $V^* = (\overline{V}^*(0))^{-1}\overline{V}^*$ are solutions for applying (A.4). Using Theorem 2.4 in [14] together with the above material we get the following cost estimate.

PROPOSITION A.1. *Computing the expression (A.4) of the inverse of the block-Hankel matrix (A.1) reduces to multiplying matrix polynomials of degree $O(m)$ in $\mathsf{F}^{s\times s}$, and can be done with $O^{\sim}(s^{\omega}m)$ operations in F.*

Multiplying a block triangular Toeplitz or Hankel matrix in $\mathsf{F}^{n\times n}$ with blocks of size $s \times s$ by a matrix in $\mathsf{F}^{n\times n}$ reduces

to the product of two matrix polynomials of degree $O(m)$, and of dimensions $s \times s$ and $s \times n$. Using the fast algorithms in [3] or [2], such a $s \times s$ product can be done in $O^{\sim}(s^{\omega}m)$ operations. By splitting the $s \times n$ matrix into $s \times s$ blocks, the $s \times s$ by $s \times n$ product can thus be done in $O^{\sim}(m \times s^{\omega}m) = O^{\sim}(s^{\omega}m^2)$ operations.

For $n = s^{\nu}$ let $\omega(1,1,\nu)$ be the exponent of the problem of $s \times s$ by $s \times n$ matrix multiplication over F. The splitting considered just above of the $s \times n$ matrix into $s \times s$ blocks, corresponds to taking $\omega(1,1,\nu) = \omega + \nu - 1 < \nu + 1.376$ ($\omega < 2.376$ due to [5]), with the total cost $O^{\sim}(s^{\omega(1,1,\nu)}m) = O^{\sim}(s^{\omega}m^2)$. Depending on $\sigma \geq 1$, a slightly smaller bound than $\nu + 1.376$ for $\omega(1,1,\nu)$ may be used due the matrix multiplication techniques specifically designed for rectangular matrices in [15]. This is true as soon as $\nu \geq 1.171$, and gives for example $\omega(1,1,\nu) < \nu + 1.334$ for $\nu = 2$, i.e., for $s = \sqrt{n}$.

COROLLARY A.2. *Let H be the block-Hankel matrix of (A.1). If the representation* (A.4) *of H^{-1} is given, then computing $H^{-1}M$ for an arbitrary $M \in \mathsf{F}^{n \times n}$ reduces to four $s \times s$ by $s \times n$ products of polynomial matrices of degree $O(m)$. This can be done with $O^{\sim}(s^{\omega(1,1,\nu)}m)$ or $O^{\sim}(s^{\omega}m^2)$ operations in F ($n = s^{\nu} = ms$).*

B. REFERENCES

[1] B. Beckermann and G. Labahn. A uniform approach for the fast computation of matrix-type Padé approximants. *SIAM J. Matrix Anal. Appl.*, 15(3):804–823, July 1994.

[2] A. Bostan and E. Schost. Polynomial evaluation and interpolation on special sets of points. *J. Complex.*, 21(4):420–446, 2005.

[3] D. Cantor and E. Kaltofen. Fast multiplication of polynomials over arbitrary algebras. *Acta Informatica*, 28:693–701, 1991.

[4] L. Chen, W. Eberly, E. Kaltofen, B. D. Saunders, W. J. Turner, and G. Villard. Efficient matrix preconditioners for black box linear algebra. *Linear Algebra and its Applications*, 343–344:119–146, 2002.

[5] D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions. *J. Symb. Comp.*, 9:251–280, 1990.

[6] John D. Dixon. Exact solution of linear equations using p-adic expansions. *Numerische Mathematik*, 40:137–141, 1982.

[7] J.-G. Dumas, T. Gautier, M. Giesbrecht, P. Giorgi, B. Hovinen, E. Kaltofen, B. D. Saunders, W. J. Turner, and G. Villard. LinBox: A generic library for exact linear algebra. In Arjeh M. Cohen, Xiao-Shan Gao, and Nobuki Takayama, editors, *Proceedings of the 2002 International Congress of Mathematical Software, Beijing, China*, pages 40–50. World Scientific, August 2002.

[8] J.-G. Dumas, B. D. Saunders, and G. Villard. Integer Smith form via the valence: experience with large sparse matrices from homology. In *ISSAC '00: Proceedings of the 2000 International Symposium on Symbolic and Algebraic Computation*, pages 95–105, New York, NY, USA, 2000. ACM Press.

[9] W. Eberly. Processor-efficient parallel matrix inversion over abstract fields: two extensions. In *Proceedings, PASCO '97*, pages 38–45, New York, NY, USA, 1997. ACM Press.

[10] W. Eberly, M. Giesbrecht, P. Giorgi, A. Storjohann, and G. Villard. Solving sparse rational linear systems. In *ISSAC '06: Proceedings of the 2006 International Symposium on Symbolic and algebraic computation*, pages 63–70, New York, NY, USA, 2006. ACM Press.

[11] I. Z. Emiris and V. Y. Pan. Improved algorithms for computing determinants and resultants. *J. Complex.*, 21(1):43–71, 2005.

[12] M. Giesbrecht. Efficient parallel solution of sparse systems of linear diophantine equations. In *Proceediings, PASCO'97*, pages 1–10, 1997.

[13] M. Giesbrecht. Fast computation of the Smith form of a sparse integer matrix. *Computational Complexity*, 10(1):41–69, 2004.

[14] P. Giorgi, C.-P. Jeannerod, and G. Villard. On the complexity of polynomial matrix computations. In Rafael Sendra, editor, *Proceedings of the 2003 International Symposium on Symbolic and Algebraic Computation, Philadelphia, Pennsylvania, USA*, pages 135–142. ACM Press, New York, August 2003.

[15] X. Huang and V. Y. Pan. Fast rectangular matrix multiplication and applications. *J. Complex.*, 14(2):257–299, 1998.

[16] E. Kaltofen. Analysis of Coppersmith's block Wiedemann algorithm for the parallel solution of sparse linear systems. *Mathematics of Computation*, 64(210):777–806, April 1995.

[17] E. Kaltofen and B. D. Saunders. On Wiedemann's method of solving sparse linear systems. In *Proc. AAECC-9*, volume 539 of *Springer Lecture Notes in Comp. Sci.*, 1991. 29–38.

[18] E. Kaltofen and G. Villard. On the complexity of computing determinants. *Computational Complexity*, 13(3-4):91–130, 2004.

[19] G. Labahn, D. K. Chio, and S. Cabay. The inverses of block Hankel and block Toeplitz matrices. *SIAM J. Comput.*, 19(1):98–123, 1990.

[20] V. Y. Pan, B. Murphy, R. E. Rosholt, and X. Wang. Toeplitz and Hankel meet Hensel and Newton: Nearly optimal algorithms and their practical acceleration with saturated initialization. Technical Report 2004 013, The Graduate Center, CUNY, New York, 2004.

[21] J. T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. Assoc. Computing Machinery*, 27:701–717, 1980.

[22] D. H. Wiedemann. Solving sparse linear equations over finite fields. *IEEE Transactions on Information Theory*, 32(1):54–62, January 1986.

[23] R. Zippel. Probabilistic algorithms for sparse polynomials. In *Proc. EUROSAM 79*, pages 216–226, Marseille, 1979.