Faster Inversion and Other Black Box Matrix Computations Using Efficient Block Projections
Wayne Eberly, Mark Giesbrecht, Pascal Giorgi, Arne Storjohann, Gilles Villard

To cite this version:
Wayne Eberly, Mark Giesbrecht, Pascal Giorgi, Arne Storjohann, Gilles Villard. Faster Inversion and Other Black Box Matrix Computations Using Efficient Block Projections. 2007. hal-00127807

HAL Id: hal-00127807
https://hal.archives-ouvertes.fr/hal-00127807
Submitted on 29 Jan 2007

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Faster Inversion and Other Black Box Matrix Computations
Using Efficient Block Projections

Wayne Eberly1, Mark Giesbrecht2, Pascal Giorgi3,4, Arne Storjohann2, Gilles Villard4

(1) Department of Computer Science, U. Calgary
http://pages.cpsc.ucalgary.ca/~eberly

(2) David R. Cheriton School of Computer Science, U. Waterloo
http://www.uwaterloo.ca/~{mg,pgiorgi,astorjoh}

(3) CNRS, LIP, École Normale Supérieure de Lyon
http://perso.ens-lyon.fr/gilles.villard

(4) IUT - Université de Perpignan
http://webdali.univ-perp.fr/~pgiorgi

ABSTRACT

Efficient block projections of non-singular matrices have recently been used, in [1], to obtain an efficient algorithm to find rational solutions for sparse systems of linear equations. In particular a bound of \( O(n^{2.5}) \) machine operations is obtained for this computation assuming that the input matrix can be multiplied by a vector with constant-sized entries using \( O(n) \) machine operations. Somewhat more general bounds for black-box matrix computations are also derived. Unfortunately, the correctness of this algorithm depends on a conjecture regarding the existence of so-called efficient block projections. In this paper, we prove the existence of a class of such efficient block projections, given a matrix \( A \in F^{n \times n} \) over any field \( F \), these projections should be block vectors \( u \in F^{n \times s} \) (where \( s \) is a blocking factor dividing \( n \), so \( n = ms \)) such that we can compute \( uv \) quickly for any \( v \in F^{n \times s} \), and such that the sequence of vectors \( u, Av, \ldots, A^{m-1}u \) has rank \( n \).

In this paper, we 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 implies our algorithm from [1] for finding the solution to \( A^{-1}b \) for a “sparse” system of equations \( A \in Z^{n \times n} \) and \( b \in Z^{n \times 1} \) works as stated using these projections, and requires fewer bit operations than any previously known when \( A \) is sparse, at least using “standard” (i.e., cubic) matrix arithmetic. Here, by \( A \) being sparse, we mean that it has a fast matrix-vector product modulo any small (machine-word size) prime \( p \). In particular, our algorithm requires an expected \( O(n^{1.5}(\log(\|A\| + \|b\|)) \) matrix-vector products \( v \mapsto Av \mod p \) for \( v \in Z_p^{n \times 1} \) plus additional \( O(n^{1.5}(\log(\|A\| + \log(\|b\|))) \) bit operations. The algorithm is probabilistic of the Las Vegas type. That is, it assumes the ability to generate random bits at unit cost, and always returns the correct answer with controllably high probability. When \( \phi(n) = O(n) \), the implied cost of \( O(n^{2.5}) \) bit operations improves upon the \( p \)-adic lift-
ing method of \[1\] which requires \(O(n^3)\) bit operations for sparse or dense matrices. This theoretical efficiency was reflected in practice in \[1\] 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 sparse matrix problem to be solved. In particular, given a matrix \(A \in \mathbb{F}^{m \times n}\) over an arbitrary field \(\mathbb{F}\), we are able to compute the complete inverse of \(A\) with \(O(n^{3-1/(\omega-1)})\) operations in \(\mathbb{F}\) plus \(O(n^{2-1/(\omega-1)})\) matrix-vector products by \(A\). Here \(\omega\) is such that we can multiply two \(n \times n\) matrices with \(O(n^\omega)\) operations in \(\mathbb{F}\). Standard matrix multiplication gives \(\omega = 3\), while the best known matrix multiplication of Coppersmith & Winograd \[6\] has \(\omega = 2.376\). If again we can compute \(v \mapsto Av\) with \(O(n)\) operations in \(\mathbb{F}\), this implies an algorithm to compute the inverse with \(O(n^{3-1/(\omega-1)})\) operations in \(\mathbb{F}\). This is always less than \(O(n^\omega)\), and in particular equals \(O(n^{2.377})\) operations in \(\mathbb{F}\) for the best known \(\omega\) of \[2\]. 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 \[2\] and \[8\] to computing the determinant and Smith form of sparse matrices more efficiently.

There has certainly been much more work done on finding exact solutions to sparse rational systems prior to \[1\]. Dixon’s \(p\)-adic lifting algorithm \[1\] performs extremely well in practice for dense and sparse linear systems, and is implemented efficiently in LinBox \[3\], Maple and Magma (see \[2\] for a comparison). Kaltofen & Saunders \[3\] 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 \[13, 14\], and this is further developed in \[23, 24\]. See the references in these papers for a more complete history. For sparse systems over a field, the seminal work is that of Wiedemann \[23\] who shows how to solve sparse \(n \times n\) systems over a field with \(O(n)\) matrix-vector products and \(O(n^n)\) other operations. This research is further developed in \[13, 23, 24\] and many other works. The bit complexity of similar operations for various families of structure matrices is examined in \[13\].

2. SPARSE BLOCK GENERATORS FOR THE KRYLOV SPACE

For now we will consider an arbitrary invertible matrix \(A \in \mathbb{F}^{m \times n}\) over a field \(\mathbb{F}\), and an integer — the blocking factor — which divides \(n\) exactly. Let \(m = n/s\). For a so-called block projection \(u \in \mathbb{F}^{m \times s}\) and \(1 \leq k \leq m\), we denote by \(K_k(A, u)\) the block Krylov matrix \([u, Au, \ldots, A^{k-1}u] \in \mathbb{F}^{n \times ks}\). Our goal is to show that \(K_m(A, u) \in \mathbb{F}^{n \times n}\) is nonsingular 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)

\[
\begin{bmatrix}
I_s \\
\vdots \\
I_s
\end{bmatrix} \in \mathbb{F}^{n \times s}
\]

(2.1)

which is comprised of \(m\) copies of \(I_s\) and thus has exactly \(n\) non-zero entries. A similar projection has been suggested in \[1\] 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 \(D = \text{diag}(\delta_1, \delta_2, \ldots, \delta_2, \ldots, \delta_m)\) be an \(n \times n\) diagonal matrix whose entries consist of \(m\) distinct indeterminates \(\delta_i\), each \(\delta_i\) occurring \(s\) times.

**Theorem 2.1.** If the leading \(ks \times ks\) minor of \(A\) is nonzero for \(1 \leq k \leq m\), then \(K_m(DA, u) \in \mathbb{F}^{n \times n}\) is invertible.

**Proof.** Let \(B = DA\). For \(1 \leq k \leq m\), define \(B_k\) as the specialization of \(B\) obtained by setting \(\delta_{k+1}, \delta_{k+2}, \ldots, \delta_m\) to zero. Then \(B_k\) is the matrix constructed by setting to zero the last \(n - ks\) rows and columns of \(B\). Similarly, for \(1 \leq k \leq m\) we define \(u_k \in \mathbb{F}^{n \times ks}\) to be the matrix constructed from \(u\) by setting to zero the last \(n - ks\) rows. In particular we have \(B_m = B\) and \(u_m = u\).

We proceed by induction on \(k\) and show that

\[
\text{rank} K_k(B_k, u_k) = ks,
\]

(2.2)

for \(1 \leq k \leq m\). For the base case \(k = 1\) we have \(K_1(B_1, u_1) = u_1\) and thus \(\text{rank} K_1(B_1, u_1) = \text{rank} u_1 = s\).

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

\[
\text{rank} K_k(B_k, B_k u_{k+1}) = ks.
\]

(2.3)

The matrix in (2.3) can be written as

\[
K_k(B_k, B_k u_{k+1}) = \begin{bmatrix} M_k & 0 \\ 0 & \end{bmatrix} \in \mathbb{F}^{n \times ks},
\]

where \(M_k \in \mathbb{F}^{ks \times ks}\) is nonsingular. Introducing the block \(u_{k+1}\) we obtain the following matrix:

\[
[u_{k+1}, K_k(B_k, B_k u_{k+1})] = \begin{bmatrix} * & M_k \\ I_s & 0 \\ 0 & 0 \end{bmatrix}.
\]

(2.4)

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

\[
[u_{k+1}, K_k(B_k, B_k u_{k+1})] = [K_{k+1}(B_k, u_{k+1})],
\]

we are led to

\[
\text{rank} K_{k+1}(B_k, u_{k+1}) = (k+1)s.
\]

Finally, using the fact that \(B_k\) is the specialization of \(B_{k+1}\) obtained by setting \(\delta_{k+1}\) to zero, we obtain

\[
\text{rank} K_{k+1}(B_{k+1}, u_{k+1}) = (k+1)s,
\]

which is (2.4) for \(k+1\) and thus establishes the theorem by induction.

If the leading \(ks \times ks\) minor of \(A\) is nonzero then the leading \(ks \times ks\) minor of \(A^T\) is nonzero as well, for any integer \(k\).

**Corollary 2.2.** If the leading \(ks \times ks\) minor of \(A\) is nonzero for \(1 \leq k \leq m\) and \(B = DA\), then \(K_m(B^T, u) \in \mathbb{F}^{n \times n}\) is invertible.
Suppose now that $A \in F^{n \times n}$ is an arbitrary non-singular matrix and the size of $F$ exceeds $n(n+1)$. It follows by Theorem 2 of Kaltofen and Saunders [3] that there exists a lower triangular Toeplitz matrix $L \in F^{n \times n}$ and an upper triangular Toeplitz matrix $U \in F^{n \times n}$ such that each of the leading minors of $A = ULU^T$ is nonzero. Let $B = DAD$; then the products of the determinants of the matrices $K_m(B, u)$ and $K_m(B^T, u)$ (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 $D \in F^{n \times n}$ such that $K_m(B, u)$ and $K_m(B^T, u)$ are non-singular, for

$$B = DAD = DUALD.$$  

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

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

Then

$$K_m(RA, \tilde{v}) = LDK_m(B, u)$$

and

$$L^TDK_m((RA)^T, \tilde{u}^T) = K_m(B^T, u),$$

so that $K_m(RA, \tilde{v})$ and $K_m((RA)^T, \tilde{u}^T)$ are each non-singular as well. Since $D$ is diagonal and $U$ and $L$ are triangular Toeplitz matrices it is now easily established that $(R, \tilde{u}, \tilde{v})$ is an “efficient block projection” for the given matrix $A$, where this is defined as in [1].

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

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

### 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 sparse integer matrices in [1].

The basis is the following factorization of the matrix inverse.

Let $B = DAD$, where $D$ is an $n \times n$ diagonal matrix whose diagonal entries consist of $m$ distinct indeterminates, each occurring $s$ times continguously, as previously. Define $K^{(r)}_u = K_m(B, u)$ with $u$ as in (2) and $K^{(r)}_u = K_m(B^T, u)^T$ (where $(r)$ and $(l)$ refer to projection on the right and left respectively). For any $0 \leq k \leq m-1$ and any two indices $l$ and $r$ such that $l + r = k$ we have $u^T \cdot B^k \cdot u = u^T \cdot B^k \cdot u$. Hence the matrix $H_u = K^{(r)}_u \cdot B \cdot K^{(l)}_u$ is block-Hankel with blocks of dimension $s \times s$:

$$H_u = \begin{bmatrix} u^TBu & u^TB^2u & \ldots & u^TB^mu \\ u^TB^2u & u^TB^3u & \ldots & u^TB^{m+1}u \\ \vdots & \vdots & \ddots & \vdots \\ u^TB^mu & u^TB^{m+1}u & \ldots & u^TB^{2m-2}u \\ & & & u^TB^{2m-1}u \end{bmatrix} \in F^{n \times n}$$

From $H_u = K^{(r)}_u \cdot B \cdot K^{(l)}_u = K^{(r)}_u \cdot DAD \cdot K^{(l)}_u$ (mentioned in the above theorem and corollary), we have following corollary to Theorem 2.3, which together with Corollary 2.3 implies that $K^{(r)}_u$ and $K^{(l)}_u$, and thus $H_u$ are invertible.

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

$$B^{-1} = K^{(r)}_u H^{-1}_u K^{(l)}_u,$$

$$A^{-1} = DK^{(r)}_u H^{-1}_u K^{(l)}_u D,$$

(3.1)

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

We note that for any specialization of the indeterminates in $D$ to field elements in $F$ such that det $H_u \neq 0$ we get a similar formula to (3.1) completely over $F$. A similar factorization in the non-blocked case is used in [10] (4.5) for fast parallel matrix inversion.

### 4. Black-Box Matrix Inversion over a Field

Let $A \in F^{n \times n}$ be invertible and such that for any $v \in F^{n \times 1}$ the matrix times vector product $Av$ or $A^Tv$ can be computed in $O(n)$ operations in $F$ (where $O(n)$ means $n^k$ operations in $F$). 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 $O((n^{2-1/k}(n^{-1}))$ black box evaluations and additional $O((n^{3-1/k}(n^{-1}))$ operations in $F$. Note that $O(n) = O(n)$ — a common characterization of “sparse” matrices — the exponent in $n$ of this cost is smaller than $\omega$, and is $O(n^{2.273})$ with the currently best-known matrix multiplication.

We assume for the moment that all principal $ks \times ks$ minors of $A$ are non-zero, $1 \leq k \leq m$.

Let $d_1, d_2, \ldots, d_m$ be the indeterminates which form the diagonal entries of $D$ and let $B = DAD$. By Theorem 2.3 and Corollary 2.2 the matrices $K_m(B, u)$ and $K_m(B^T, u)$ are each invertible. If $m \geq 2$ then the product of the determinants of these matrices is a non-zero polynomial $\Delta \in F[d_1, \ldots, d_m]$, with total degree less than $2n(m-1) - 1$.

Suppose that $F$ has at least $2n(m-1)$ elements. Then $\Delta$ cannot zero at all points in $\{F \setminus \{0\} \}^n$. Let $d_1, d_2, \ldots, d_m$ be nonzero elements of $F$ such that $\Delta(d_1, d_2, \ldots, d_m) \neq 0$, let $D = \text{diag}(d_1, d_2, \ldots, d_m)$, and let $B = DAD$. Then $K^{(r)}_u = K_m(B, u) \in F^{n \times n}$ and $K^{(l)}_u = K_m(B^T, u)^T \in F^{n \times n}$ are each invertible since $\Delta(d_1, d_2, \ldots, d_m) \neq 0$. $B$ is invertible since $A$ is and $d_1, d_2, \ldots, d_m$ are all nonzero, and thus $H_u = K^{(l)}_u BK^{(r)}_u \in F^{n \times n}$ is invertible as well. Correspondingly, (3.1) suggests

$$B^{-1} = K^{(r)}_u H^{-1}_u K^{(l)}_u,$$

and $A^{-1} = DK^{(r)}_u H^{-1}_u K^{(l)}_u D$

for computing the matrix inverse.

1. **Computation of $u^T$ and $u^TBu$**

We can compute this sequence, and hence $K^{(r)}_u$ with $m - 1$ applications of $B$ to vectors using $O(mO(n))$ operations in $F$.

2. **Computation of $H_u$**

Due to the special form of $H_u$ of $u$, one may then compute $wu$ for any $w \in F^{n \times 1}$ with $O(sn)$ operations.
Hence we can now compute $u^T B' u$ for $0 \leq i \leq 2m - 1$ with $O(n^3)$ operations in $F$.

3. Computation of $H^{-1}$.

The off-diagonal inverse representation of $H^{-1}$ as in $(A.4)$ in the Appendix can be found with $O(s^2m)$ operations by Proposition $A.3$.

4. Computation of $H^{-1} K_u^{(r)}$.

From Corollary $A.2$ in the Appendix, we can compute the product $H^{-1} M$ for any matrix $M \in F^{n \times n}$ with $O(s^2m^2)$ operations.

5. Computation of $K_u^{(r)} \cdot (H^{-1} K_u^{(r)})$.

We can compute $K_u^{(r)} M = [u, B u, \ldots, B^{m-1} u] M$, for any $M \in F^{n \times n}$ by splitting $M$ into $m$ blocks of $s$ consecutive rows $M_i$, for $0 \leq i \leq m - 1$:

$$K_u M = \sum_{i=0}^{m-1} B'(u M_i)$$

$$= u m_0 + B(u M_1) + B(u M_2 + \cdots + B(u m_{m-2} + B u M_{m-1}) \cdots) \quad (4.1)$$

Because of the special form $(2.1)$ of $u$, each product $u M_i \in F^{n \times n}$ requires $O(n)$ operations, and hence all such products involved in $(4.1)$ can be computed in $O(n^3)$ operations. Since applying $B$ to an $n \times n$ matrix costs $n \omega(n)$ operations, $K_u^{(r)} M$ is computed in $O(n \omega(n) + mn^2)$ 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 $O(s^2m^2 + mn^2)$ additional operations. If $\phi(n) = O(n)$ the overall number of field operations is minimized with the blocking factor $s = n^{1/2(\omega - 1)}$.

**Theorem 4.1.** Let $A \in F^{n \times n}$, where $n = ms$, be such that all leading $ks \times ks$ minors are non-singular for $1 \leq k \leq m$. Let $B = D A D$ for $D = \text{diag}(d_1, \ldots, d_1, \ldots, d_m)$ such that $d_1, d_2, \ldots, d_m$ are nonzero and each of the matrices $K_u(D AD, u)$ and $K_u((DAD)^T, u)$ is invertible. Then the inverse matrix $A^{-1}$ can be computed using $O(n^{\omega - 1/3(\omega - 1)})$ applications of $A$ to vectors and an additional $O(n^{3/2(\omega - 1)})$ 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 $K_u(D AD, u)$ and $K_u((DAD)^T, u)$ are each nonzero. While 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 $n \geq \frac{16}{\omega}$ and that $\#F > 2(m + 1)n |\log_2 n|$. Fix a set $S$ of at least $2(m + 1)n |\log_2 n|$ nonzero elements of $F$. We can ensure that the leading $ks \times ks$ minors of $A$ are non-zero by pre-multiplying by a butterfly network preconditioner $U$ with parameters chosen uniformly and randomly from $S$. If $U$ is constructed using the generic exchange matrix of $\left[ \begin{array}{c|c} \mathbb{F} & \mathbb{F} \\ \hline \mathbb{F} & \mathbb{F} \end{array} \right]$ [6, 2], then it will use at most $n |\log_2 n| / 2$ random elements from $S$, and from $\left[ \begin{array}{c|c} \mathbb{F} & \mathbb{F} \\ \hline \mathbb{F} & \mathbb{F} \end{array} \right]$. Then $\#F \geq 4n |\log_2 n| > 4 \omega \Delta$, the probability that $\Delta$ is zero at this point is at most $1/4$ by the Schwartz-Zippel Lemma [2, 2].

In summary, for randomly selected butterfly preconditioner $B$, and independently and randomly chosen values $d_1, d_2, \ldots, d_m$, the probability that $A = U A$ has non-singular leading $ks \times ks$ minors for $1 \leq k \leq m$ 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 $S$ of $F \setminus \{0\}$ with size at least $2(m + 1)n |\log_2 n|$.

When $\#F \leq 2(m + 1)n |\log_2 n|$ we can easily construct a field extension $E$ of $F$ which has size greater than $2(m + 1)n |\log_2 n|$ and perform the computation in that extension. Since this extension will have degree $O(\log_2 \#F)$ 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.

Finally, we note that it is easily checked whether the matrix returned by this algorithm is the inverse of the input by using $n$ multiplications by $A$ of the columns of the output matrix and corresponding each to the corresponding column of the identity matrix. This results in a Las Vegas algorithm for computation of the inverse of a black-box matrix with the cost as given above.

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

<table>
<thead>
<tr>
<th>$\omega$</th>
<th>Black-box operations</th>
<th>Blocking factor $s$</th>
<th>Inversion cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 (Standard)</td>
<td>1.5</td>
<td>1/2</td>
<td>$O(n^{n/5})$</td>
</tr>
<tr>
<td>2.807 (Strassen)</td>
<td>1.446</td>
<td>0.553</td>
<td>$O(n^{2.446})$</td>
</tr>
<tr>
<td>2.3755 (Cop/Win)</td>
<td>1.273</td>
<td>0.728</td>
<td>$O(n^{2.273})$</td>
</tr>
</tbody>
</table>

**Remark 4.3.** The structure $(2.3)$ 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 $u \in F^{n \times s}$, 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.

**Applying a Black-Box Matrix Inverse to a Matrix**

The above method can also be used to compute $A^{-1} M$ for any matrix $M \in \mathbb{F}^{n \times n}$ with the same cost as in Theorem 4.3. Consider the new step 1.5:

1.5. Computation of $K_u^{(t)} \cdot M$.

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

The cost for computing $K_u^{(t)} \cdot M_k$ for a single $k$ by the above process is $n - s$ 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 $nO(n)$ operations in $\mathbb{F}$, $K_u^{(t)} \cdot M$ is computed in $O(mn^2 + mn^2)$ operations in $\mathbb{F}$ 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. Since Step 4 does not rely on any special form for $K_u^{(t)}$, we can replace it with a computation of $H_u^{-1} \cdot (K_u^{(t)} \cdot M)$ with the same cost. We obtain the following:

**Corollary 4.4.** Let $A \in \mathbb{F}^{n \times n}$ be non-singular and let $M \in \mathbb{F}^{n \times n}$ be any matrix. We can compute $A^{-1} M$ using a Las Vegas algorithm whose expected cost is $O(n^{2-1/2\omega^{-1}})$ multiplications of $A$ to vectors and $O(n^{3-1/2\omega^{-1}})$ additional operations in $\mathbb{F}$.

The estimates in Table 3 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 $\mathbb{F}$. 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. That is, the output is certified to be correct.

Kaltofen & Saunders [3] present algorithms for computing the rank of a matrix and for randomly sampling the nullspace, building upon work of Wiedemann [4]. In particular, they show that for random lower upper and lower triangular Toeplitz matrices $U, L \in \mathbb{F}^{n \times n}$, and random diagonal $D$, that all leading $k \times k$ minors of $ALD$ are non-singular for $1 \leq k \leq r = \text{rank } A$, and that if $f \hat{A} \in \mathbb{F}[x]$ is the minimal polynomial of $\hat{A}$, then it has degree $r+1$ if $A$ is singular (and degree $n$ if $A$ is non-singular). This is proven to be true for any input $A \in \mathbb{F}^{n \times n}$, and for random choice of $U, L$ and $D$, with high probability. The cost of computing $f \hat{A}$ (and hence rank $A$) is shown to be $O(n)$ applications of the black-box for $A$ and $O(n^2)$ additional operations in $\mathbb{F}$. 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 & Saunders [4] 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 & 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 $A \leftarrow U ALD$ as above. We will thus assume in what follows that $A$ has all leading $i \times i$ minors non-singular for $1 \leq i \leq r$. While 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, then 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 $A_0^{-1} A_1 \in \mathbb{F}^{r \times (n-r)}$ using the algorithm of the previous section (this could in fact be merged into the first step).

3. **Confirming the nullspace.**

Note that

$$
\begin{pmatrix} A_0 & A_1 \\
A_2 & A_3 \end{pmatrix} \begin{pmatrix} A_0^{-1} A_1 \\
-I \end{pmatrix} \begin{pmatrix} 0 \\
A_2 A_0^{-1} A_1 - A_3 \end{pmatrix} = 0
$$

and the Schur complement $A_2 A_0^{-1} A_1 - A_3$ 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, $N$ 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 $A \in \mathbb{F}^{n \times n}$ have rank $r$. Then a basis for the nullspace of $A$ and rank $r$ of $A$ can be computed with an expected number of $O(n^{2-1/2\omega^{-1}})$ applications of $A$ to a vector, plus an additional expected number of $O(n^{3-1/2\omega^{-1}})$ operations in $\mathbb{F}$. 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 [3] we presented an algorithm and implementation to compute
$A^{-1}b$ with $O((n^{1.5}\log(|A|) + \|b\|))$ matrix-vector products $v \mapsto A \mod p$ for a machine-word sized prime $p$ and any $v \in \mathbb{Z}_p^n$ plus $O(n^{1.5}\log(|A|) + \|b\|))$ 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 $O(n)$ operations modulo $p$, the algorithm presented could solve a system with $O(n^2)$ bit operations. Unfortunately, this result was conditional upon the unproven Conjecture 2.1: the existence of an efficient block projection. This conjecture was established in Corollary 2.3 of [1]:

**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 $v \mapsto Av \mod p$ is in $O(n^{1.5}\log(|A|) + \|b\|))$, and the expected number of additional bit-operations used by this algorithm is in $O(n^2\log(|A|) + \|b\|))$.

**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 [21] 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 [21], 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 $O(\log(n + \|v\| + \|A\|))$.

**Preconditioning and setup.**

Precondition $A \leftarrow B = D_1 U A D_2$, where $D_1, D_2$ are random diagonal matrices, and $U$ is a unimodular pre-conditioner from [21, §5]. While we will not do the detailed analysis here, selecting coefficients for these randomly from $\mathbb{Z}^n$ 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 [1] Theorem 4.3 for a proof of the latter condition). From Theorem 2.1, we also see that $K_m(B, u)$ has full rank with high probability.

Let $p$ be a prime which 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 $n = ms$.

**Choosing projections.** Let $u \in \mathbb{Z}^{n \times s}$ be an efficient block projection as in [21] 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 $2n^2$.

**Forming the sequence $\alpha_i = u A^i v \in \mathbb{Z}^{s \times s}$.**

Compute this sequence for $i = 0 \ldots 2m$. Computing all the $A^i v$ takes $O(\phi(n) - m \log \|A\|)$ bit operations. Computing all the $u A^i v$ takes $O(n^2 - m \log \|A\|)$ bit operations.

**Computing the minimal matrix generator.**

The minimal matrix generator $F(\lambda)$ modulo $p$ can be computed from the initial sequence segment $\alpha_0, \ldots, \alpha_{2m-1}$. See [21, §4]. This can be accomplished with $O(m s^3 \cdot n \log \|A\|)$ bit operations.

**Extracting the determinant.**

Following the algorithm in [21, §4], we first check if its degree is less than $n$ and if so, return “failure”. Otherwise, we know $\det F(\lambda) = \det(A - I)$. Return $\det A = \det F(0) \mod p$.

The correctness of the algorithm, and specifically the block projections, follows from fact that $[u, Au, \ldots, A^{m-1}u]$ is of full rank with high probability by Theorem 2.1. Since the projection $v$ is dense, the analysis of [21 (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 is $O((\phi(n) m + n^2 m + mn s^3) \log \|A\|)$ bit operations, which is minimized when $s = n^{1/\omega}$. This yields an algorithm for the determinant which requires $O((n^{2.5 - 1/\omega} \log \|A\|)$ bit operations. This is probably most interesting when $\omega = 3$, where it yields an algorithm for determining which requires $O(n^{2.66} \log \|A\|)$ 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 [14], which is computed by means of computing the characteristic polynomial of random preconditionings of a matrix. This reduction is explored in [21] 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 [21, §7.1] and [3] for details. We make no claim that this is practical in its present form.

**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 [15]. An alternative to using the inversion formula would be to use the generalization of the Levinson-Durbin algorithm in [10].

For an integer $m$ that divides $n$ exactly with $s = n/m$, let

$$H = \begin{bmatrix}
\alpha_0 & \alpha_1 & \cdots & \alpha_{m-1} \\
\alpha_1 & \alpha_2 & \cdots & \alpha_{m-2} \\
\vdots & \vdots & \ddots & \vdots \\
\alpha_{m-1} & \alpha_{m-2} & \cdots & \alpha_{2m-2} \\
\end{bmatrix} \in \mathbb{F}^{s \times s}$$

be a non-singular block-Hankel matrix whose blocks are $s \times s$ matrices over $\mathbb{F}$, and let $\alpha_{m+1}$ be arbitrary in $\mathbb{F}^{s \times s}$. We follow the lines of [24] for computing the inverse matrix $H^{-1}$. Since $H$ is invertible, the following four linear systems (see [21] for details)

\begin{align*}
\alpha_0 u + \alpha_1 v & = w_1 \\
\alpha_1 u + \alpha_2 v & = w_2 \\
\vdots & \vdots \\
\alpha_{m-1} u + \alpha_{m-2} v & = w_{2m-1}
\end{align*}
with $Q, P, Q', P' \in F^{n \times n}$ that satisfy the degree constraints $\deg Q \leq m - 1$, $\deg Q' \leq m - 1$, and $\deg P \leq m - 2$, $\deg P' \leq m - 2$. The residue matrices $R$ and $R'$ in $F^{s \times s}$ are non-singular, hence $QR^{-1}$ and $(R')^{-1}Q'$ are solutions $\tilde{Q}$ and $\tilde{Q}'$ for applying the inversion formula \((\ref{4})\). For \((\ref{3}))\), the $\sigma$-basis algorithm with $\sigma = s(2m + 1)$ leads to

$$
\begin{bmatrix}
A - I & \tilde{Q}
\end{bmatrix} = \text{mod} x^{2m+1},
$$

$$
\begin{bmatrix}
\tilde{Q}' & P'
\end{bmatrix} = \text{mod} x^{2m+1},
$$

deg \tilde{V} \leq m, \deg \tilde{V}' \leq m, and $\deg \tilde{U} \leq m - 1, \deg \tilde{U}' \leq m - 1$. The constant terms $\tilde{V}(0)$ and $\tilde{V}'(0)$ in $F^{s \times s}$ are non-singular, hence $\tilde{V} = \tilde{V}(0)^{-1}$ and $\tilde{V}' = (\tilde{V}'(0))^{-1}$ are solutions for applying \((\ref{4}))\). Using Theorem 2.4 in \([10]\) together with the above material we get the following cost estimate.

**PROPOSITION A.1.** Computing the expression \((\ref{3}))\) of the inverse of the block-Hankel matrix \((\ref{4}))\) reduces to multiplying matrix polynomials of degree $O(m)$ in $F^{s \times s}$, and can be done with $O((s^m)^2)$ operations in $F$.

Multiplying a block triangular Toeplitz or Hankel matrix in $F^{s \times s}$ with blocks of size $s \times s$ by a matrix in $F^{n \times n}$ 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 \([8]\) or \([9]\), such a $s \times s$ product can be done in $O((s^m)^2)$ 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 $O((m \times s^m) = O(s^m^2)$ operations.

For $n = s^m$ 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 $\omega(1,1,\nu) = \omega + \nu - 1 < \nu + 1.376$ (for $\nu = 2.376$ due to \([11]\) with the total cost $O((s^m)^2)$), 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 \([11]\). 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 $\sqrt{n} = \sqrt{m}$.

**COROLLARY A.2.** Let $H$ be the block-Hankel matrix of \((\ref{4}))\). If the representation \((\ref{4}))\) of $H^{-1}$ is given then computing $H^{-1}M$ for an arbitrary $M \in F^{n \times n}$ reduces to four $s \times s$ by $s \times n$ products of polynomial matrices of degree $O(m)$. This can be done with $O((s^{2(1,1,\nu)})$ or $O((s^{m2})$ operations in $F$ (for $n = s^m = ms$).

**REFERENCES**


