Reliable Krylov-Based Algorithms for Matrix Null Space and Rank

Extended Abstract

Wayne Eberly *
Department of Computer Science
University of Calgary
Calgary, Alberta, Canada
eberly@cpsc.ucalgary.ca

ABSTRACT

Krylov-based algorithms have recently been used, in combination with other methods, to solve systems of linear equations and to perform related matrix computations over finite fields. For example, large and sparse systems of linear equations over F_2 are formed during the use of the number field sieve for integer factorization, and elements of the null space of these systems are sampled.

Two rather different kinds of block algorithms have recently been considered. Block Wiedemann algorithms have now been presented and fully analyzed. Block Lanczos algorithms were proposed earlier but are not yet as well understood. In particular, proofs of reliability of block Lanczos algorithms are not yet available. Nevertheless, an examination of the computational number theory literature suggests that block Lanczos algorithms continue to be preferred.

This report presents a block Lanczos algorithm that is somewhat simpler than block algorithms that are presently in use and provably reliable for computations over large fields. To my knowledge, this is the first block Lanczos algorithm for which a proof of reliability is available.

A different Krylov-based approach is considered for computations over small fields: It is shown that if Wiedemann's sparse matrix preconditioner is applied to an arbitrary matrix then the number of nontrivial invariant factors of the result is, with high probability, quite small. A Krylov-based algorithm to compute a partial Frobenius decomposition can then be used to sample from the null space of the original matrix or to compute its rank. This yields a randomized (Monte Carlo) black box algorithm for matrix rank that is asymptotically faster, in the small field case, than any other that is presently known.

*Research was supported in part by Natural Sciences and Engineering Research Council of Canada research grant OGP0089756.

Categories and Subject Descriptors

I.1.2 [**Symbolic and Algebraic Manipulation**]: Algorithms—*algebraic algorithms, analysis of algorithms*; F.2.1 [**Analysis of Algorithms and Problem Complexity**]: Numerical Algorithms and Problems—*computations in finite fields, computations on matrices*

General Terms

Algorithms, Reliability, Theory

Keywords

Black box matrix, block Lanczos algorithm, Frobenius decomposition, linear system solution, matrix rank, randomized algorithm

1. INTRODUCTION

Consider the problem of selecting a vector uniformly and randomly from the (right) null space of a given matrix. As discussed in the report of Buhler, Lenstra, and Pomerance [1], this problem arises for large, sparse matrices over the finite field $\mathsf{F} = \mathsf{F}_2$ when the number field sieve is applied.

As reported by LaMacchia and Odlyzko [13] structured Gaussian Elimination can be used for this computation. However, storage requirements may be prohibitive for large problems when this technique is applied. Krylov-based algorithms, such as the algorithm of Lanczos [14], require less storage and are reliable for computations over the real numbers, but require modification if they are applied for computations over small finite fields.

A block Lanczos algorithm was proposed for this purpose by Coppersmith [3] in the early 1990's, with the objectives of improving both reliability and coarse-grain parallelism. Variants of this algorithm, including a simpler algorithm of Montgomery [15], have been used (frequently in combination with elimination-based methods) with considerable success. Unfortunately, the reliability of these algorithms has not been adequately analyzed.

Indeed, these algorithms are not reliable in the worst case for computations over small finite fields: Krylov-based algorithms for singular matrix computations perform poorly if they are applied to matrices whose minimal polynomials (in $\mathsf{F}[z]$) are divisible by z^2 and that have a large number of invariant factors. The heuristics currently in use do not

overcome this problem. For example, they are ineffective for computations over F_2 when applied to block-diagonal matrices that include a large number of diagonal blocks

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

along with a large identity matrix as a final block. Heuristics that use symmetrization to condition the input — replacing A by A^tA or by AA^t – are defeated by block diagonal matrices with a form similar to the above, if multiple blocks

$$\begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} \qquad \text{and} \qquad \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}$$

are also used.

A related Krylov-based algorithm, namely, that of Wiedemann [17], has subsequently been developed and fully analyzed. Furthermore, a block variant (with improved parallelism, once again) has also been proposed by Coppersmith [4]. Unlike the block Lanczos algorithms, a somewhat modified version of this block Wiedemann algorithm has been shown to be reliable — see Kaltofen [11] for the analysis in the large field case and Villard [16] for the analysis over small finite fields. A subsequent paper of Kaltofen and Villard [12] includes a summary of this algorithm, the underlying theory, and some additional results concerning the behaviour of the algorithm in the large field case.

The block Wiedemann algorithm has at least one other notable advantage over any block Lanczos algorithm proposed to date: As noted by Kaltofen [11], the block Wiedemann algorithm allows the use of rectangular matrices as blocks. If block sizes are appropriately chosen, then the resulting algorithm is considerably faster than any block Lanczos algorithm that is presently known.

Nevertheless, variants of the Lanczos algorithm continue to be used instead. We are therefore lead to ask whether algorithms that resemble the currently used heuristics are provably reliable.

A partial answer is provided in this report: A block Lanczos algorithm that is provably reliable for computations over large fields is described in Section 2. I am not aware of any interesting application for which this algorithm is superior to the block Wiedemann algorithms that are already available. This material is presented, instead, in the hope that it will help to improve our understanding of Krylov-based algorithms and, perhaps, facilitate future improvements.

A rather different algorithm is described in Section 3 for computations over small fields. It is shown that if Wiedemann's sparse preconditioner is applied to an arbitrary matrix then the number of nontrivial invariant factors of the result is, with high probability, at most logarithmic in the rank of the input matrix. A Krylov-based algorithm to compute a partial Frobenius decomposition can then be applied, either to sample from the null space of the original matrix or to compute its rank.

This new algorithm has a few advantages over existing block Wiedemann algorithms, together with at least one notable disadvantage. While it is possible that the new algorithm might have practical application, I consider it to be at least as likely that this will lead to modifications of the analysis and implementation of block Wiedemann algorithms, to obtain the advantages of both approaches.

Several proofs are omitted from this abstract. These can be found in the more complete version of this paper [8].

2. COMPUTATIONS IN LARGE FIELDS

Eberly and Kaltofen [9] present a simple scalar Lanczos algorithm and show that it is reliable over arbitrary large fields. In this section, this algorithm is modified to produce a simple block algorithm that can be used to sample from the null space of a given matrix A with entries in a sufficiently large field or to compute its rank.

2.1 A Matrix Preconditioner

We begin with a diagonal matrix preconditioner described in the above paper. Additional information about this preconditioner can be found in the report of Chen et. al. [2].

LEMMA 1 (EBERLY AND KALTOFEN [9]). *Suppose F is a field and let $A \in \mathsf{F}^{m\times n}$ be a matrix with rank r. Let S be a finite subset of $\mathsf{F} \setminus \{0\}$, and suppose*

$$\alpha_1, \alpha_2, \ldots, \alpha_n, \beta_1, \beta_2, \ldots, \beta_m$$

are chosen uniformly and independently from S. Let

$$D_{\vec{\alpha}} = \begin{bmatrix} \alpha_1 & & & \\ & \alpha_2 & & \\ & & \ddots & \\ & & & \alpha_n \end{bmatrix} \in \mathsf{F}^{n\times n}$$

and let

$$D_{\vec{\beta}} = \begin{bmatrix} \beta_m & & & \\ & \beta_2 & & \\ & & \ddots & \\ & & & \beta_m \end{bmatrix} \in \mathsf{F}^{m\times m}.$$

Then, with probability at most $\frac{11n^2-n}{2|S|}$, the matrix

$$\widetilde{A} = D_{\vec{\alpha}} A^T D_{\vec{\beta}} A D_{\vec{\alpha}} \in \mathsf{F}^{n\times n}$$

is a matrix with rank r, whose characteristic polynomial is $z^{n-r}f$ for some squarefree polynomial $f \in \mathsf{F}[z]$ with degree r such that $f(0) \neq 0$.

A consideration of the rank of A confirms that if the above-mentioned matrix $\widetilde{A}$ has the properties described in the lemma, then the minimal polynomial of $\widetilde{A}$ is zf and $\widetilde{A}$ is similar to a diagonal matrix over a suitable extension of F (namely, a splitting field of f).

Eberly and Kaltofen observe that if $\widetilde{A}$ is as described above, and if a system of linear equations $\widetilde{A}x = b$ is consistent, then a solution for the system can be found within the Krylov space of b. That is, there exists a linear combination x of the vectors $b, \widetilde{A}b, \widetilde{A}^2b, \ldots$ that satisfies the above system of equations. If F is a finite field then we may select an element uniformly from the null space of $\widetilde{A}$ by uniformly selecting a vector z, choosing a vector x such that $\widetilde{A}x = b$, for $b = \widetilde{A}z$, and returning the vector $z - x$.

Since A and $\widetilde{A}$ have the same rank, $\widetilde{A} = D_{\vec{\alpha}} A^T D_{\vec{\beta}} A D_{\vec{\alpha}}$, and the diagonal matrix $D_{\vec{\alpha}}$ is nonsingular, a vector y is in the null space of $\widetilde{A}$ if and only if $D_{\vec{\alpha}}y$ is in the null space of A. We may also return the vector $D_{\vec{\alpha}}(z-x)$ as a uniformly selected element of the null space of A.

Suppose now that $k \geq 1$, and that k vectors $z_1, z_2, \ldots, z_k$ have been uniformly and independently selected from $\mathsf{F}^{n\times 1}$. Let $\vec{z}$ be the matrix in $\mathsf{F}^{n\times k}$ whose i^{th} column is z_i, for $1 \leq i \leq k$. It follows by a straightforward generalization

Input: A symmetric matrix $\widetilde{A} \in \mathsf{F}^{n \times n}$ and a matrix $\vec{b} \in \mathsf{F}^{n \times k}$

Output: A matrix $\vec{x} \in \mathsf{F}^{n \times k}$ such that $\widetilde{A}\vec{x} = \vec{b}$, or failure

1. $\vec{w}_{-1} := 0_{n \times k}$; $\vec{v}_0 := 0_{n \times k}$; $\vec{x}_{-1} := 0_{n \times k}$; $\vec{t}_{-1} := I_k$
2. $\vec{w}_0 := \vec{b}$; $\vec{v}_1 := \widetilde{A}\vec{w}_0$; $\vec{t}_0 := \vec{v}_1^t \vec{w}_0$
3. $i := 0$
4. **while** $\det \vec{t}_i \neq 0$ **do**
5. $\quad \vec{x}_i := \vec{x}_{i-1} + \vec{w}_i \vec{t}_i^{-1} \vec{w}_i^t \vec{b}$
6. $\quad \vec{w}_{i+1} := \vec{v}_{i+1} - \vec{w}_i \vec{t}_i^{-1} \vec{v}_i^t \vec{v}_{i+1} - \vec{w}_{i-1} \vec{t}_{i-1}^{-1} \vec{v}_{i-1}^t \vec{v}_{i+1}$
7. $\quad \vec{v}_{i+2} := \widetilde{A}\vec{w}_{i+1}$
8. $\quad \vec{t}_{i+1} := \vec{v}_{i+2}^t \vec{w}_{i+1}$
9. $\quad i := i + 1$

end while

10. **if** $\vec{w}_i \neq 0_{n \times k}$ **then**
11. $\quad$ Set h to the the largest integer such that the leftmost h columns of $\vec{w}_i$ are linearly independent.
12. $\quad$ Set $\vec{w}_i$ to be the matrix in $\mathsf{F}^{n \times h}$ that includes the leftmost h columns of the current $\vec{w}_i$.
13. $\quad$ Set $\vec{t}_i$ to be the top left $h \times h$ submatrix of the current $\vec{t}_i$, so that $\vec{t}_i \in \mathsf{F}^{h \times h}$.
14. $\quad$ **if** $h = 0$ **or** $\det \vec{t}_i = 0$ **then**
15. $\quad\quad$ **report** failure

$\quad$ **else**

16. $\quad\quad \vec{x} := \vec{x}_{i-1} + \vec{w}_i \vec{t}_i^{-1} \vec{w}_i^t \vec{b}$

$\quad$ **end if**

else

17. $\quad \vec{x} := \vec{x}_{i-1}$

end if

18. **if** $\widetilde{A}\vec{x} = \vec{b}$ **then**
19. $\quad$ **return** $\vec{x}$

$\quad$ **else**

20. $\quad$ **return** failure

end if

Figure 1: A Block Lanczos Algorithm

of the above process that a sequence of k vectors can be uniformly and independently sampled from the null space of $\widetilde{A}$ by finding a solution $\vec{x} \in \mathsf{F}^{n \times k}$ for the system

$$\widetilde{A}\vec{x} = \vec{b} \qquad \text{for} \quad \vec{b} = \widetilde{A}\vec{z},$$

and returning the columns of the matrix $D_{\vec{\alpha}}(\vec{z} - \vec{x})$.

2.2 A Block Lanczos Algorithm

Consider the algorithm that is shown in Figure 1. This is a straightforward generalization of the "standard Lanczos algorithm" shown in Figure 1 of the paper of Eberly and Kaltofen [9].

Suppose $\widetilde{A} \in \mathsf{F}^{n \times n}$ is symmetric with rank r. Once again, we are interested in the behaviour of the given algorithm when $\widetilde{A}$ has a minimal polynomial zf for some squarefree polynomial $f \in \mathsf{F}[z]$ with degree r such that $f(0) \neq 0$, so that $\widetilde{A}$ is similar to a diagonal matrix over an extension of F.

Let $\ell = \lceil r/k \rceil - 1$, where k is the block size used in the algorithm.

If failure is not reported, then the algorithm generates a sequence of matrices

$$\vec{w}_0, \vec{w}_1, \vec{w}_2, \ldots, \vec{w}_\ell$$

such that $\vec{w}_i \in \mathsf{F}^{n \times k}$ for $0 \leq i \leq \ell - 1$ and $\vec{w}_\ell \in \mathsf{F}^{n \times h}$ for some integer h such that $1 \leq h \leq k$. As noted in the next section, it will frequently be the case that $h = k$ if r is divisible by k, and that $h = r - k\ell$ if m is not divisible by k.

The columns of the matrices $\vec{w}_0, \vec{w}_1, \vec{w}_2, \ldots, \vec{w}_i$ are linearly independent and form a basis for the vector space spanned by the columns of the matrices

$$\vec{b}, \widetilde{A}\vec{b}, \widetilde{A}^2\vec{b}, \ldots, \widetilde{A}^i\vec{b}$$

for each integer i such that $0 \leq i \leq \ell$. Consequently, if h has its usual value (as given above), then the columns of the matrices $\vec{w}_0, \vec{w}_1, \vec{w}_2, \ldots, \vec{w}_\ell$ form a basis for the column space of $\widetilde{A}$, and the number of these columns is equal to the rank of $\widetilde{A}$.

A useful *orthogonality condition* is achieved:

$$\vec{w}_i^t \widetilde{A} \vec{w}_j = 0 \tag{1}$$

for all integers i and j such that $0 \leq i, j \leq \ell$ and $i \neq j$, and

$$\det \vec{w}_i^t \widetilde{A} \vec{w}_i \neq 0 \tag{2}$$

for $0 \leq i \leq \ell$.

Two other sequences of matrices are computed along the way, in order to minimize the number of multiplications by $\widetilde{A}$ that are used: $\vec{v}_0, \vec{v}_1, \vec{v}_2, \ldots, \vec{v}_\ell$ are matrices such that

$$\vec{v}_{i+1} = \widetilde{A}\vec{w}_i \qquad \text{for } 0 \leq i \leq \ell - 1, \tag{3}$$

and $\vec{t}_0, \vec{t}_1, \ldots, \vec{t}_\ell$ are square matrices such that

$$\vec{t}_i = \vec{w}_i^t \widetilde{A} \vec{w}_i \qquad \text{for } 0 \leq i \leq \ell. \tag{4}$$

The algorithm maintains one more sequence of matrices, in order to produce a solution for the given system:

$$\vec{x}_0, \vec{x}_1, \ldots, \vec{x}_{\ell-1}$$

are matrices in $\mathsf{F}^{n \times k}$ such that

$$\vec{w}_j^t (A\vec{x}_i - b) = 0 \tag{5}$$

for all integers i and j such that $0 \leq j \leq i \leq \ell - 1$; this is used at the end of the algorithm to generate a matrix $\vec{x}$ such that

$$\vec{w}_j^t (A\vec{x} - b) = 0 \tag{6}$$

for all j such that $0 \leq j \leq \ell$.

A comparison of this algorithm with the scalar algorithm will confirm that this is, indeed, a straightforward generalization: The two algorithms maintain the same sequences of matrices when $k = 1$, using virtually the same sets of operations. It is somewhat simpler than block Lanczos algorithms of Coppersmith [3] or Montgomery [15], due to the omission of any kind of lookahead mechanism. There is good reason to include such mechanisms for computations over small fields. However, as argued in the next section, these are not required for computations over large fields, when the coefficient matrix $\widetilde{A}$ has the properties that have been described here and the columns of $\vec{b}$ are randomly chosen from the column space of $\widetilde{A}$.

2.3 Analysis of Reliability

The following proof of reliability of the block Lanczos algorithm is, again, a modification of that of the reliability of the algorithm of Eberly and Kaltofen [9]. Suppose, once again, that $\widetilde{A} \in \mathsf{F}^{n\times n}$ is a symmetric matrix with rank r, and that $\vec{b} \in \mathsf{F}^{n\times k}$ for an integer $k \geq 1$. Let us consider the following block-Hankel matrices. For $1 \leq i \leq \lfloor r/k \rfloor$, let

$$H_i(\widetilde{A}, \vec{b}) = \begin{bmatrix} \vec{b}^t\widetilde{A}\vec{b} & \vec{b}^t\widetilde{A}^2\vec{b} & \cdots & \vec{b}^t\widetilde{A}^i\vec{b} \\ \vec{b}^t\widetilde{A}^2\vec{b} & \vec{b}^t\widetilde{A}^3\vec{b} & \cdots & \vec{b}^t\widetilde{A}^{i+1}\vec{b} \\ \vdots & \vdots & \ddots & \vdots \\ \vec{b}^t\widetilde{A}^i\vec{b} & \vec{b}^t\widetilde{A}^{i+1}\vec{b} & \cdots & \vec{b}^t\widetilde{A}^{2i-1}\vec{b} \end{bmatrix}. \quad (7)$$

Let $H(\widetilde{A}, \vec{b}) \in \mathsf{F}^{r\times r}$ be the matrix $H_{r/k}(\widetilde{A}, \vec{b})$ if r is divisible by k, and let $H(\widetilde{A}, \vec{b})$ be the top left $r \times r$ submatrix of $H_{\lceil r/k \rceil}(\widetilde{A}, \vec{b})$, otherwise.

The following result can be proved using a reasonably straightforward generalization of the proof of Lemma 3.2 in Eberly and Kaltofen [9].

LEMMA 2. *Suppose that $\widetilde{A} \in \mathsf{F}^{n\times n}$ is a symmetric matrix with rank r, whose minimal polynomial has the form zf, where $f \in \mathsf{F}[z]$ is a squarefree polynomial with degree r such that $f(0) \neq 0$. Let $\vec{b} \in \mathsf{F}^{n\times k}$ be a matrix such that the system $\widetilde{A}\vec{x} = \vec{b}$ is consistent — that is, each of the columns of $\vec{b}$ belongs to the column space of $\widetilde{A}$. Finally, suppose that $\det H_i(\widetilde{A}, \vec{b}) \neq 0$ for $1 \leq i \leq \lfloor r/k \rfloor$ and that $\det H(\widetilde{A}, \vec{b}) \neq 0$ as well.*

Then the algorithm shown in Figure 1 succeeds. In particular, it generates a sequence of matrices

$$w_0, w_1, \ldots, w_\ell$$

for $\ell = \lceil r/k \rceil - 1$ whose columns are linearly independent and form a basis for the column space of $\widetilde{A}$, and it returns a matrix $\vec{x} \in \mathsf{F}^{n\times k}$ such that $\widetilde{A}\vec{x} = \vec{b}$.

It now suffices to bound the probability that the conditions given in the above lemma are satisfied. Suppose that $z_{i,j}$ are distinct indeterminates over F, for $1 \leq i \leq n$ and $1 \leq j \leq k$. Let

$$\vec{\zeta} = \begin{bmatrix} z_{1,1} & z_{1,2} & \cdots & z_{1,k} \\ z_{2,1} & z_{2,2} & \cdots & z_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ z_{n,1} & z_{n,2} & \cdots & z_{n,k} \end{bmatrix} \in \mathsf{F}[z_{1,1} \ldots, z_{n,k}]^{n\times k}, \quad (8)$$

and suppose that

$$\vec{\beta} = \widetilde{A}\vec{\zeta}. \quad (9)$$

The determinants of the corresponding matrices $H_i(\widetilde{A}, \vec{\beta})$ and $H(\widetilde{A}, \vec{\beta})$ are polynomials in $\mathsf{F}[z_{1,1}, \ldots, z_{n,k}]$ with total degrees at most $2ki$ and $2r$, respectively.

LEMMA 3. *Suppose once again that $\widetilde{A} \in \mathsf{F}^{n\times n}$ is a symmetric matrix with rank r whose minimal polynomial has the form zf, where $f \in \mathsf{F}[z]$ is a squarefree polynomial with degree r such that $f(0) \neq 0$. Suppose as well that k is odd and k is not divisible by the characteristic of the field F.*

Then if $\vec{\zeta}$ and $\vec{\beta}$ are as given in Equations (8) and (9), then the polynomials $\det H_i(\widetilde{A}, \beta)$ are nonzero for $1 \leq i \leq \lfloor r/k \rfloor$, and the polynomial $\det H(\widetilde{A}, \vec{\beta})$ is nonzero as well.

PROOF (SKETCH). If $\widetilde{A}$ and k are as in the lemma then there exists a matrix B with entries over an extension of F such that B is similar to a diagonal matrix and

$$\widetilde{A} = B^k.$$

It follows by results of Eberly and Kaltofen [9] that there exists a vector u with entries over an extension of F such that the matrix

$$\begin{bmatrix} u^tBu & u^tB^2u & \cdots & u^tB^ru \\ u^tB^2u & u^tB^3u & \cdots & u^tB^{r+1}u \\ \vdots & \vdots & \ddots & \vdots \\ u^tB^ru & u^tB^{r+1}u & \cdots & u^tB^{2r-1}u \end{bmatrix}$$

is in generic rank profile. Furthermore, there exists a vector v with entries in an extension of F such that

$$B^{(3k-1)/2}v = u.$$

Now if one sets

$$\vec{v} = \begin{bmatrix} v & Bv & \cdots & B^{k-1}v \end{bmatrix}$$

then it can be shown that polynomials mentioned in the lemma all have nonzero values when $\vec{\zeta}$ is replaced with $\vec{v}$. □

The following can be deduced using Lemmas 2 and 3, along with an application of the Schwartz-Zippel lemma.

THEOREM 4. *Suppose that $\widetilde{A} \in \mathsf{F}^{n\times n}$ is a symmetric matrix with rank r, whose minimal polynomial has the form zf, where $f \in \mathsf{F}[z]$ is a squarefree polynomial with degree r such that $f(0) \neq 0$. Let $k \geq 1$ such that k is odd and k is not divisible by the characteristic of F. Finally, suppose that the algorithm shown in Figure 1 is applied with inputs $\widetilde{A}$ and a matrix $\vec{b} = \widetilde{A}\vec{z} \in \mathsf{F}^{n\times k}$, where the entries of the matrix $\vec{z} \in \mathsf{F}^{n\times k}$ are chosen uniformly and independently from a finite subset S of F.*

Then the algorithm returns a matrix $\vec{x} \in \mathsf{F}^{n\times k}$ such that $\widetilde{A}\vec{x} = \vec{b}$ with probability at least $1 - r(r+1)/|S|$.

Furthermore, if F is a finite field and $S = \mathsf{F}$, then the resulting matrix $\vec{x}$ is uniformly chosen from the set of solutions for the above system of equations.

It follows that the process described in this section can be used to produce a set of k elements of the null space of a given matrix A: It suffices to sample elements uniformly and independently from a finite subset S of F with size in $O(n^2/\epsilon)$ in order to bound the probability of failure by ϵ, for any given error tolerance $\epsilon > 0$. Furthermore, if F is a finite field and once chooses S to be the entire field F, then the resulting vectors are sampled uniformly and independently from the null space of the given matrix.

2.4 Computation of the Rank

If the conditions given in Theorem 4 are satisfied then the matrix $\widehat{A}$ has rank $k\ell + h$ where k is the blocking factor used and ℓ and h are as described above. It follows by Lemma 1 and the above theorem that this algorithm can be used to compute the rank of a given matrix; the likelihood that a generated value is incorrect is in $O(n^2/|S|)$.

2.5 Comparison with Block Wiedemann

The above algorithm is considerably simpler than any block Wiedemann algorithm that is currently in use. However, as previously noted, I am not aware of any interesting

application for which the above algorithm is superior to existing block Wiedemann algorithms.

A superficial examination of the literature might suggest otherwise: The bound on the number of matrix-vector products given above, for a reliable block Lanczos algorithm, depends on the rank of the input matrix instead of its order. Bounds that are given in the literature for block Wiedemann algorithms are generally stated in terms of the order of the matrix instead. However, it appears that a block Wiedemann algorithm that incorporates a reasonable form of "early termination" could be used for computations over sufficiently large fields, either to sample from the null space of a given matrix or compute its rank, using a number of matrix-vector products that is considerably lower than the number claimed above. While this is not explicitly stated, it appears that all of the results needed to establish this can be found in the work of Kaltofen and Villard [12].

3. COMPUTATIONS IN SMALL FIELDS

In this section, a different approach is used to produce a Krylov-based algorithm that is reliable for computations over small fields.

3.1 The Frobenius Form

Consider a square matrix $\widehat{A} \in \mathsf{F}^{\ell \times \ell}$ for a positive integer ℓ. It is well known (see, for example, Gantmacher [10]) that $\widehat{A}$ is similar to a unique block diagonal matrix with companion matrices of monic polynomials $f_1, f_2, \ldots, f_k$ on the diagonal, for some integer $k \leq \ell$, where f_i is divisible by f_{i+1} for $1 \leq i \leq k-1$. That is, there exists a nonsingular matrix $V \in \mathsf{F}^{\ell \times \ell}$ such that

$$V\widehat{A}V^{-1} = \begin{bmatrix} C_{f_1} & & & 0 \\ & C_{f_2} & & \\ & & \ddots & \\ 0 & & & C_{f_k} \end{bmatrix} \tag{10}$$

and where

$$C_g = \begin{bmatrix} 0 & \cdots & 0 & -g_0 \\ 1 & & 0 & -g_1 \\ & \ddots & \vdots & \vdots \\ 0 & & 1 & -g_{d-1} \end{bmatrix} \in \mathsf{F}^{d \times d} \tag{11}$$

is the companion matrix of a monic polynomial

$$g = x^d + g_{d-1}x^{d-1} + g_{d-2}x^{d-2} + \cdots + g_1 x + g_0 \in \mathsf{F}[x].$$

The block diagonal matrix shown on the right hand side of Equation (10) is commonly called the *Frobenius form* of $\widehat{A}$, and the polynomials $f_1, f_2, \ldots, f_k$ are called the *invariant factors* of $\widehat{A}$.

If the matrix $\widehat{A}$ is singular then one or more of the invariant factors of $\widehat{A}$ may be equal to the polynomial x; we will say that an invariant factor f_i is a *nontrivial invariant factor* if $f_i \neq x$.

3.2 A Sparse Matrix Preconditioner

Suppose, once again, that $A \in \mathsf{F}^{n \times m}$. Let $q = |\mathsf{F}|$, let $N = \max(n, m)$, and let h be an integer such that

$$\operatorname{rank}(A) \leq h \leq N. \tag{12}$$

Let

$$\ell = h + \lceil 2 \log_q N \rceil, \tag{13}$$

and let

$$\widehat{c} = \begin{cases} 3 & \text{if } q = 2, \\ \lceil 3 \ln q \rceil & \text{otherwise.} \end{cases}$$

Consider matrices $L \in \mathsf{F}^{\ell \times n}$ and $R \in \mathsf{F}^{m \times \ell}$ whose entries are randomly selected according to the following distribution.

- If $1 \leq i \leq h$ then each entry in row i of L or column i of R is set to be zero with probability
$$\max\left(1 - \frac{\widehat{c}\log_q N}{i}, \frac{1}{q}\right).$$
- If $1 \leq i \leq h$ then each entry in row i of L or column i of R that has not been set to be 0, above, is chosen uniformly and independently from $\mathsf{F} \setminus \{0\}$.
- Finally, if $h < i \leq \ell$, then each entry of row i of L or column of R is chosen uniformly and independently from F.

Let $\widehat{A} = LAR \in \mathsf{F}^{\ell \times \ell}$.

Matrices resembling L and R have been investigated by Wiedemann [17]; additional useful properties are discussed in the report of Chen et. al. [2]. The following can be established by a generalization of their analysis.

LEMMA 5. *Suppose that $N = \max(n, m) \geq 6$ and the matrices L and R are generated as described above.*

(a) *The expected number of nonzero entries in each of L and R is in $O(N(\log_q N)^2)$.*

(b) *The rank of $\widehat{A}$ is less than or equal to that of A. The probability that the ranks of the two matrices are different is at most $1/N$.*

(c) *The expected number of invariant factors of $\widehat{A}$ that are divisible by x^2 is less than 5.*

(d) *The expected number of nontrivial invariant factors of the matrix $\widehat{A}$ is at most $\log_q r + 10$.*

PROOF (SKETCH). Parts (a) and (b) can be established using generalizations of arguments used by Wiedemann [17], in Section III of his paper, to prove similar results for a related sparse matrix preconditioner. Part (c) can be proved using techniques used by Chen et. al. [2] to establish a similar result (in Section 7 of their report).

It is sufficient to bound the expected number of invariant factors of $\widehat{A}$ that are not powers of x in order to establish part (d). Suppose there are at least k of these. Then there must exist a nonzero element λ of an extension of F such that the rank of the matrix $\widehat{A} - \lambda I_\ell$ is at most $\ell - k$. Techniques that were used to established part (b) can be applied (with a bit of care) to bound the probability that this is the case. This can then be used to bound the expected number of invariant factors that are not powers of x as required. □

3.3 Computing the Frobenius Form

Once again, let us consider a matrix $\widehat{A} \in \mathsf{F}^{\ell \times \ell}$. Suppose that we are given $\widehat{A}$ along with a positive integer u_f that will serve as a conjectured upper bound on the number of nontrivial invariant factors of $\widehat{A}$. Let d be the sum of the degrees of the nontrivial invariant factors of this matrix.

A black box algorithm for a Frobenius decomposition of a matrix has been given by Eberly [5, 6]. In this section, modifications of the algorithm that allow it to be used to sample from the null space or compute the rank of a suitably conditioned matrix will be described. In particular, this algorithm will be modified to obtain a Monte Carlo algorithm that accepts $\widehat{A}$ and u_f as inputs and that has the following properties.

- If the number of nontrivial invariant factors is, indeed, less than or equal to u_f, then the algorithm will return the nontrivial invariant factors of $\widehat{A}$, and a basis for the intersection of the null space and the column space of $\widehat{A}$, with high probability.
- If the bound u_f is incorrect — that is, $\widehat{A}$ includes more than u_f nontrivial invariant factors — then the algorithm will report failure with high probability, instead.
- If the algorithm is successful then the expected number of matrix-vector products used by the algorithm is in $O(d)$. The expected number of additional operations required over F is in $O(\ell d u_f)$, and the amount of storage space used is in $O(\ell u_f^2 + \ell \log \ell)$.

The algorithm of Eberly [5] makes repeated use of a procedure minpolspace that is presented and analyzed in Section 3.1 of the above reports.

On its initial application, this procedure uses a sequence of uniformly and independently selected vectors from $\mathsf{F}^{\ell \times 1}$ in order to generate a pair of vectors u_1 and v_1 in $\mathsf{F}^{\ell \times 1}$, and a monic polynomial $f_1 \in \mathsf{F}[x]$, such that the following properties hold.

- f_1 is the monic polynomial of least degree such that $f_1(\widehat{A})v_1 = 0$.
- f_1 is also the monic polynomial of least degree such that $f_1(\widehat{A}^t)u_1 = 0$.
- Finally, f_1 is the minimal polynomial of the linearly recurrent sequence
$$u_1^t v_1, u_1^t \widehat{A} v_1, u_1 \widehat{A}^2 v_1 \ldots$$
- The expected number of vectors that must be selected from $\mathsf{F}^{\ell \times 1}$ to perform this computation is in $O(1)$. The expected number of matrix-vector products by $\widehat{A}$ or $\widehat{A}^t$ that is used is linear in the degree of f_1. Finally, the expected number of additional operations over the field F that are used by this procedure is linear in the product of ℓ and the degree of f_1.
- The polynomial f_1 is always a divisor of the minimal polynomial of $\widehat{A}$; it is equal to the minimal polynomial of $\widehat{A}$ with probability at least one-half.

Suppose the above polynomial f_1 has degree d_1. If the above conditions are satisfied then the Hankel matrix
$$\begin{bmatrix} u_1^t v_1 & u_1^t \widehat{A} v_1 & \cdots & u_1^t \widehat{A}^{d_1-1} v_1 \\ u_1^t \widehat{A} v_1 & u_1^t \widehat{A}^2 v_1 & \cdots & u_1^t \widehat{A}^{d_1} v_1 \\ \vdots & \vdots & \ddots & \vdots \\ u_1^t \widehat{A}^{d_1-1} v_1 & u_1^t \widehat{A}_{d_1} v_1 & \cdots & u_1^t \widehat{A}^{2d_1-2} v_1 \end{bmatrix}$$
is nonsingular. However, it is desirable to ensure that leading submatrices are likely to be nonsingular as well.

A **first modification** that will be made to the algorithm will therefore be a randomization: The vector v_1 will be replaced by $g_1(A)v_1$, where g_1 is a randomly chosen polynomial in $\mathsf{F}[x]$ that is relatively prime to f_1. Then the above conditions are still satisfied, and the above Hankel matrix is still nonsingular. Furthermore, it follows by a straightforward modification of a result of Eberly [7] that a scalar Lanczos algorithm can be used, with u_1 and v_1, in order to orthogonalize a pair of sets of k vectors with respect to
$$u_1, \widehat{A}^t u_1, \ldots, (\widehat{A}^t)^{d_1-1} v_1 \quad \text{and} \quad v_1, \widehat{A} v_1, \ldots, \widehat{A}^{d_1-1} v_1$$
respectively. In particular, this computation can be performed using the vectors u_1, v_1, and the vectors to be orthogonalized, while using storage space for $O(\ell \log \ell + k)$ field elements in the worst case.

As discussed below, vectors u_i and v_i will be generated for $i \geq 2$ using a similar process. The choice of vectors $v_2, v_3, \ldots$ will be also be changed as described above.

A **second modification** can now be made: Rather than storing all of
$$u_i, \widehat{A}^t u_i, \ldots, (\widehat{A}^t)^{d_i-1} u_i \tag{14}$$
and
$$v_i, \widehat{A} v_i, \ldots, \widehat{A}^{d_i-1} v_i, \tag{15}$$
— or a dual basis for the Krylov spaces that are generated by u_i and v_i — the algorithm will store u_i and v_i alone.

The algorithm of Eberly [5] requires a supply of vectors that have been generated by selecting $O(u_f)$ vectors uniformly and independently from $\mathsf{F}^{\ell \times 1}$, and orthogonalizing these vectors with respect to Krylov spaces corresponding to the invariant factors that have currently been generated.

A **third modification** concerns the way these vectors are produced: In the original procedure the values shown above in lines (14) and (15) are used repeatedly to generate them. Since these values are no longer being stored, the process must be changed to ensure that they are not recomputed more than a constant number of times.

The first application of the revised procedure minpolspace ends with the uniform and independent selection of $2cu_f$ vectors from $\mathsf{F}^{\ell \times 1}$, for a suitable constant c. A scalar Lanczos algorithm is applied to u_1 and v_1 once again, in order to orthogonalize these vectors, resulting in vectors
$$\alpha_{1,1}, \ldots, \alpha_{1,s_1}, \beta_{1,1}, \ldots, \beta_{1,s_1} \in \mathsf{F}^{\ell \times 1},$$
where $s_1 = cu_f$, such that
$$\alpha_{1,i}^t \widehat{A}^j v_1 = u_1^t \widehat{A}^j \beta_{1,i} = 0$$
for $1 \leq i \leq s_1$ and $0 \leq j \leq d_1 - 1$.

The amount of storage space needed to perform this computation is in $O(\ell \log \ell + \ell u_f)$. It will be useful to use the orthogonalized vectors in later steps, so these will be stored. The total amount of storage space needed for all these vectors is linear in the product of ℓu_f and the total number of applications of minpolspace that must be used. Since this number of applications is linear in u_f, the amount of storage space required for all of these orthogonalized vectors is in $O(\ell u_f^2)$.

Each subsequent application of minpolspace will take place after a sequence of vectors and polynomials
$$(u_1, v_1, f_1), (u_2, v_2, f_2), \ldots, (u_i, v_i f_i)$$

have been generated. A set of $2s_j$ vectors

$$\alpha_{j,1}, \ldots, \alpha_{j,s_j}, \beta_{j,1}, \ldots, \beta_{j,s_j} \in \mathsf{F}^{\ell \times 1}$$

will be available as well, for some integer s_j such that $1 \leq s_j \leq cu_f$ and for $1 \leq j \leq i$. These vectors will have been orthogonalized with respect to previous Krylov spaces — that is,

$$\alpha_{j,k}^t \widehat{A}^a v_b = u_b^t \widehat{A}^a \beta_{j,k} = 0$$

for all integers j, k, a, and b such that $1 \leq b \leq j$, $1 \leq k \leq s_j$, and $0 \leq a \leq d_b$, where d_b is the degree of f_b.

In order to ensure that the vectors u_{i+1} and v_{i+1} to be generated during the current application of minpolspace are orthogonal to the Krylov spaces that have been generated already, these will be generated using vectors from the sequences

$$\alpha_{i,1}, \ldots, \alpha_{i,s_i} \qquad \text{and} \qquad \beta_{i,1}, \ldots, \beta_{i,s_i} \tag{16}$$

instead of randomly selected vectors from $\mathsf{F}^{\ell \times 1}$. The vectors that are used will then be discarded (decreasing the value of s_i). A scalar Lanczos algorithm will be applied, using u_{i+1} and v_{i+1}, to orthogonalize the vectors shown at line (16) with respect to the $i+1^{\text{st}}$ Krylov spaces, in order to produce the next set of vectors

$$\alpha_{i+1,1}, \ldots, \alpha_{i+1,s_{i+1}} \qquad \text{and} \qquad \beta_{i+1,1}, \ldots, \beta_{i,s_{i+1}}$$

at the end of this application of minpolspace.

As a result of this modification it will be necessary to recompute each of the vectors shown at lines (14) and (15) at most once.

The algorithm will make repeated use of the modified procedure minpolspace, generating estimates of the invariant factors (and discarding polynomials and Krylov spaces, when estimates are discovered to incorrect) as in Eberly [5].

A **fourth modification** will needed to provide a basis for the intersection of the null space and column space of $\widehat{A}$ as part of the output: A set of vectors that is initially empty will be maintained. As soon as it has been established, with sufficiently high probability, that the i^{th} polynomial f_i being stored is, indeed, the i^{th} invariant factor, the tuple (u_i, v_i, f_i) will be used to increase this set. In particular, if f_i is divisible by x (and is a nontrivial invariant factor) then the vector $(f_i/x)(\widehat{A})v_i$ will be included in it. The set will not be changed if x does not divide f_i (or if $f_i = x$, so that f_i is not a nontrivial invariant factor at all).

A consideration of the Jordan form of $\widehat{A}$ is sufficient to confirm that the resulting set is, indeed, a basis for the intersection of the null space and column space, if the algorithm succeeds in finding all the nontrivial invariant factors.

A **fifth modification** should also be made: The computation will halt as soon as it has been established, with high probability, either that $\widehat{A}$ includes at most u_f invariant factors, or that the $u_f + 1^{\text{st}}$ invariant factor is different from x. The algorithm reports failure in the latter case.

Unfortunately, the result is a Monte Carlo algorithm instead of a Las Vegas one: Since a complete set of invariant factors (including all trivial factors, along with corresponding Krylov spaces) is not generated, it is possible that the some of polynomials returned by this algorithm are proper divisors of the corresponding invariant factors of $\widehat{A}$.

The analysis of Eberly [5] can now be modified to establish that the above algorithm computes the desired values at the costs given at the beginning of this section.

3.4 Sampling from the Null Space

Suppose, now, that the above algorithm has successfully been used to compute the values identified as its outputs at the beginning of Section 3.3. Recall that this includes a basis for the intersection of the null space and the column space of the matrix $\widehat{A} \in \mathsf{F}^{\ell \times \ell}$ to which the algorithm was applied. Suppose that κ is the dimension of the above intersection, and that

$$w_1, w_2, \ldots, w_\kappa$$

is the above basis.

Let y be a uniformly chosen vector in $\mathsf{F}^{\ell \times 1}$. Using the output of the above algorithm (effectively, applying a Lanczos process using the generated vectors $u_1, u_2, \ldots$ and $v_1, v_2, \ldots$, and orthogonalizing y along the way), it is possible to express y as

$$y = y_1 + y_2$$

where y_1 belongs to the sum of the Krylov spaces of the vectors $v_1, v_2, \ldots$ and where y_2 is orthogonal to these spaces. One can see by a consideration of the Jordan form of $\widehat{A}$ that if values $\alpha_1, \alpha_2, \ldots, \alpha_\kappa$ are uniformly and independently selected from F then the vector

$$\widehat{y} = \sum_{i=1}^{\kappa} \alpha_i w_i + y_2$$

is a uniformly chosen element of the null space of $\widehat{A}$.

Once again, consider a matrix $A \in \mathsf{F}^{n \times m}$. Let $h = m$, so that condition (12) is satisfied, and suppose the positive integer ℓ is as chosen in line (13). Suppose the matrices $L \in \mathsf{F}^{\ell \times n}$ and $R \in \mathsf{F}^{m \times \ell}$ are randomly chosen as described in Section 3.2, and let $\widehat{A} = LAR \in \mathsf{F}^{\ell \times \ell}$.

Suppose that $\text{rank}(\widehat{A}) = \text{rank}(A)$; as noted in Lemma 5, this is the case with high probability. Suppose, furthermore, that $\text{rank}(R) = m$. Then it can be argued that if a vector z is uniformly selected from $\mathsf{F}^{m \times 1}$ then the corresponding vector Rz is uniformly selected from $\mathsf{F}^{m \times 1}$. More importantly, for this application, if a vector $\widehat{y}$ is uniformly selected from the null space of $\widehat{A}$, then the corresponding vector $R\widehat{y}$ is uniformly selected from the null space of A.

It follows that the above preconditioner and algorithm can be used to sample randomly from the null space of a given matrix.

3.5 Computation of the Rank

Let $h = \min(n, m)$, so that condition (12) is satisfied, and suppose the positive integer ℓ is as chosen in line (13). Once again, let L, R, and $\widehat{A}$ be as described in Section 3.2.

Suppose that the nontrivial invariant factors $f_1, f_2, \ldots, f_k$ of the conditioned matrix $\widehat{A} \in \mathsf{F}^{\ell \times \ell}$ have been computed as described above. Let

$$e_i = \begin{cases} \deg f_i & \text{if } f_i \text{ is not divisible by } x, \\ \deg f_i - 1 & \text{otherwise.} \end{cases}$$

Then the rank of $\widehat{A}$ is $e_1 + e_2 + \cdots + e_k$. Furthermore, it follows by Lemma 5 that A and $\widehat{A}$ have the same rank with high probability. Thus the above preconditioner and algorithm can be used to compute the rank of a given matrix as well.

3.6 Summary of Results

The following has now been established. While the results concerning the cost to sample from the null space are known (as noted below) those concerning matrix rank are new.

THEOREM 6. *Let $A \in \mathsf{F}^{n\times m}$ be a matrix over a finite field F, with (unknown) rank r. Let $N = \max(n, m)$. Then it is possible to choose an element uniformly from the null space of A, or to compute the rank of A, by a Monte Carlo algorithm that uses $O(r)$ matrix-vector products by A or A^t, along with $O(rN(\log N)^2)$ additional operations over F, and using space to store $O(N(\log N)^2)$ elements of F.*

PROOF (SKETCH). See the above discussion. The number of arithmetic operations used to multiply vectors by the preconditioners, L, and R, dominates the cost of additional operations (other than the cost of matrix-vector products by A or A^t) when the algorithm of Section 3.3 is applied. □

3.7 Comparison with Block Wiedemann

Villard [16] has established that a block Wiedemann algorithm can be used, efficiently and reliably, to generate a single element of the null space of a given matrix. Chen et. al. [2] have shown that a sparse matrix preconditioner can be used to sample uniformly and randomly from the null space of a given matrix, as well. Indeed, the method that is described in the above paper uses a preconditioner that is similar to the one being used here.

While the algorithm described in this paper has yet to be implemented, it seems clear that a single application of a block Wiedemann algorithm will be less expensive. However, the results (in Section 7) of Chen at. al. [2] suggest that multiple trials of a block Wiedemann algorithm might generally be required to sample uniformly from the null space. It is therefore unclear which technique might be faster.

The difference is clearer when the problem of computing the rank of a matrix is considered: The Monte Carlo algorithm described in this paper requires a number of matrix-vector products that is at most linear in the rank of the given matrix. Previously available methods either use computations over a field extensions, or use a binary search to find the rank, increasing the required number of matrix-vector products by a logarithmic factor in either case. Thus the new algorithm is asymptotically faster than any previous one known to be reliable over small fields if, as usual, the cost of matrix-vector products dominates the cost of the computation.

On the other hand, the block Wiedemann approach has an advantage over a new one: It is easily parallelized, and might therefore be better for a multiprocessor environment.

This leads to the following question: Can the analysis or applications of block Wiedemann algorithms be modified, or (if necessary) can the algorithm itself be modified, to produce a block algorithm that combines the advantages of existing block Wiedemann algorithms and the new one discussed above?

4. REFERENCES

[1] J. P. Buhler, H. W. Lenstra, Jr, and C. Pomerance. Factoring integers with the number field sieve. In *The Development of the Number Field Sieve*, volume 1554 of *Lecture Notes in Mathematics*, pages 50–94. Springer-Verlag, 1993.

[2] 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.

[3] D. Coppersmith. Solving linear equations over GF(2): Block Lanczos algorithm. *Linear Algebra and Its Applications*, 192:33–60, 1993.

[4] D. Coppersmith. Solving homogeneous linear equations over GF(2) via block Wiedemann algorithm. *Mathematics of Computation*, 62(205):33–60, 1994.

[5] W. Eberly. Black box Frobenius decompositions over small fields. In *Proceedings, ISSAC '00*, pages 106–113, 2000.

[6] W. Eberly. Asymptotically efficient algorithms for the Frobenius form. Technical Report 2003-723-26, Department of Computer Science, University of Calgary, 2003. Available at `www.cpsc.ucalgary.ca/~eberly/Publications/`.

[7] W. Eberly. Early termination over small fields (extended abstract). In *Proceedings, ISSAC '03*, pages 80–87, 2003. Complete version available at `www.cpsc.ucalgary.ca/~eberly/Publications/`.

[8] W. Eberly. Reliable Krylov-based algorithms for matrix null space and rank. Technical Report 2004-749-14, Department of Computer Science, University of Calgary, 2004. Available at `www.cpsc.ucalgary.ca/~eberly/Publications/`.

[9] W. Eberly and E. Kaltofen. On randomized Lanczos algorithms. In *Proceedings, ISSAC '97*, pages 176–183, 1997. A more complete version is available at `www.cpsc.ucalgary.ca/~eberly/Publications/`.

[10] F. R. Gantmacher. *The Theory of Matrices*, volume one. Chelsea Publishing Company, second edition, 1959.

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

[12] E. Kaltofen and G. Villard. On the complexity of computing determinants. Research Report 36, Laboratoire de 'Informatique du Parallélisme, Ecole Normale Supérieure de Lyon, France. Available at `www.ens-lyon.fr/LIP/Pub/rr2003.html`, 2003.

[13] B. A. LaMacchia and A. M. Odlyzko. Solving large sparse linear systems over finite fields. In *Advances in Cryptology — CRYPTO '90*, volume 537 of *Lecture Notes in Computer Science*, pages 109–133. Springer-Verlag, 1990.

[14] C. Lanczos. Solution of systems of linear equations by minimized iterations. *J. Res. Nat. Bureau of Standards*, 49:33–53, 1952.

[15] P. Montgomery. A block Lanczos algorithm for finding dependencies over GF(2). In *EUROCRYPT '95*, volume 921 of *Lecture Notes in Computer Science*, pages 106–120. Springer-Verlag, 1995.

[16] G. Villard. Further analysis of Coppersmith's block Wiedemann algorithm for the solution of sparse linear systems. In *Proceedings, ISSAC '97*, pages 32–39, 1997.

[17] D. Wiedemann. Solving sparse linear systems over finite fields. *IEEE Transactions on Information Theory*, 32:54–62, 1986.