Analysis of Coppersmith's Block Wiedemann Algorithm for the Parallel Solution of Sparse Linear Systems*

Erich Kaltofen

Department of Computer Science, Rensselaer Polytechnic Institute
Troy, NY 12189-3590, USA
Inter-net: kaltofen@cs.rpi.edu

1 Introduction

Douglas Wiedemann's (1986) algorithm for computing the N-dimensional solution vector of a system of N linear equations over a finite field $\mathbb{K}$ is efficient for sparse unstructured inputs because its running time is bounded by $3N$ multiplications of the coefficient matrix B by vectors and $O(N^2 \log N)$ additional arithmetic operations in the coefficient field. It only needs $O(N)$ additional storage for field elements. Wiedemann's algorithm can be generalized to arbitrary fields, p-adic lifting, and to the problem of computing the rank of a sparse matrix (Kaltofen and Saunders 1991). The method is randomized and computes first the sequence of field elements

$$a^{(i)} = u^{\mathrm{tr}} B^i v \in \mathbb{K} \quad \text{for } 0 \leq i \leq 2N-1,$$

where u and possibly v are vectors with random entries from $\mathbb{K}$. The key property is that this sequence is generated by a linear recursion that with high probability corresponds to the minimum polynomial of B.

There are several implementations of Wiedemann's algorithm. We have observed that for large systems such as the ones arising in the sieve-based integer factoring algorithms (A. K. Lenstra et al. 1990), where N can be as large as 100,000 and B can have as many as 5 million non-zero entries, the running time is dominated by the $3N$ multiplications of B by vectors. In order to speed this bottleneck by use of parallelism, Don Coppersmith (1991) has proposed to use simultaneously m random vectors for u and n random vectors for v. The sequence now is a sequence of $m \times n$ matrices

$$a^{(i)} = x^{\mathrm{tr}} B^i y \in \mathbb{K}^{m \times n} \quad \text{where } x^{\mathrm{tr}} \in \mathbb{K}^{m \times N} \text{ and } y \in \mathbb{K}^{N \times n}.$$

Clearly, the individual entries in $a^{(i)}$ can be computed independently and in parallel. Coppersmith then has cleverly generalized the Berlekamp/Massey algorithm needed to compute a linear recurrence that generates this sequence and

*This material is based on work supported in part by the National Science Foundation under Grant No. CCR-90-06077 and under Grant No. CDA-88-05910.

observed experimentally that over the Galois field with 2 elements $\mathbb{K} = \mathbb{F}_2$ that linear recurrence is determined by the first

$$\frac{N}{m} + \frac{N}{n} + O(1)$$

matrices $a^{(i)}$. Thus the algorithm, when executed in a parallel/distributed setting performs much faster.

Unfortunately, the blocking of the projections introduces substantial difficulties in the analysis of the method. Aside from the experimental evidence, Coppersmith gives a heuristic mathematical argument on the expected running time of the algorithm. It is these difficulties that this paper further clarifies. We can prove that a certain variant of the block algorithm with high probability runs as conjectured provided that the finite field $\mathbb{K}$ is of sufficient cardinality.

Our results also impact the sequential complexity of sparse linear system solving. Suppose, e.g., that B is non-singular, and that $\epsilon > 0$. Using blocking, one can find a solution vector $x = B^{-1}b$, where $b \in \mathbb{K}^N$, by

$$(1+\epsilon)N + O(1) \text{ multiplications of } B \text{ times vectors,}$$

and $O(N^2 \log N \ \mathrm{loglog}\, N)$ arithmetic operations in $\mathbb{K}$, needing $O(N)$ additional storage for field elements. The algorithm chooses $O(N)$ random field elements and is successful with probability $1 - (N+2)^2/\mathrm{card}(\mathbb{K})$. Here the constants implied by the big-O notation grow with $1/\epsilon$.

Our analysis is based on the observation that generically, i.e., when the projection blocks x and y are symbolic, the block method can be specialized to the original Wiedemann algorithm. From that specialization one then can prove that certain necessary rank conditions must hold generically. By the commonly used Zippel/Schwarz lemma those rank conditions will thus hold with high probability for random blocks. A further problem is the generalization of the Berlekamp/Massey algorithm to sequences of matrices. Instead, we use the equivalent problem of solving a block Toeplitz homogeneous linear system, which we can accomplish efficiently by the standard Wiedemann method.

Notation: We write $\mathbb{K}^N$ for the set of column vectors over $\mathbb{K}$, and 0^N for the N-dimensional zero vector; $0^{N\times M}$ is the $N \times M$ zero matrix. Vector and matrix transposition is indicated by superscript $^{\mathrm{tr}}$. We indicate a block matrix whose entries are matrices or vectors by vertical and horizontal strokes, such as $\left[\begin{array}{c|c} a & a'' \\ \hline a' & a''' \end{array}\right]$. Note that lower case symbols, such as x and y, may also denote matrices.

2 Linearly generated sequences

We now discuss some basic facts about linearly generated sequences of elements in a vector space V over the field $\mathbb{K}$. The sequence

$$\{a_i\}_{i=0}^{\infty}, \quad \text{where } a_i \in \mathsf{V},$$

is *linearly generated* over $\mathbb{K}$ if there exist $c_0, c_1, \ldots, c_N \in \mathbb{K}$, $N \geq 0$, $c_L \neq 0$ for some L with $0 \leq L \leq N$, such that

$$\forall j \geq 0 : c_0 a_j + \cdots + c_N a_{j+N} = 0.$$

The polynomial $c_0 + c_1\lambda + \cdots + c_N\lambda^N$ is called a *generating polynomial* for $\{a_i\}_{i=0}^{\infty}$. The set of all generating polynomials for $\{a_i\}_{i=0}^{\infty}$ together with the zero polynomial forms an ideal in $\mathbb{K}[\lambda]$. The unique polynomial generating that ideal, normalized to have leading coefficient 1, is called the *minimum polynomial* of a linearly generated sequence $\{a_i\}_{i=0}^{\infty}$. Every generating polynomial is a multiple of the minimum polynomial.

Let W be also a vector space over $\mathbb{K}$, and let $L : \mathsf{V} \longrightarrow \mathsf{W}$ be a linear map from V to W. Then the sequence $\{L(a_i)\}_{i=0}^{\infty}$ is also linearly generated by a minimum polynomial that divides the minimum generating polynomial of $\{a_i\}_{i=0}^{\infty}$. Let $B \in \mathbb{K}^{N\times N}$ be a square matrix over a field. The sequence of $N \times N$ matrices $\{B^i\}_{i=0}^{\infty}$ is linearly generated, and its minimum polynomial is the minimum polynomial of B, which will be denoted by f^B. For any column vector $b \in \mathbb{K}^N$ the sequence $\{B^i b\}_{i=0}^{\infty}$, where $B^i b \in \mathbb{K}^N$, is also linearly generated by f^B. However, its minimum polynomial, denoted by $f^{B,b}$, can be a proper divisor of f^B. For any row vector $u^{\mathrm{tr}} \in \mathbb{K}^{1\times N}$ the sequence $\{u^{\mathrm{tr}} B^i b\}_{i=0}^{\infty}$, where $u^{\mathrm{tr}} B^i b \in \mathbb{K}$, is linearly generated as well, and its minimum polynomial, denoted by $f_u^{B,b}$, is again a divisor of $f^{B,b}$. Wiedemann's method is based on the fact that for random vectors u and b with high probability $f_u^{B,b} = f^B$ (c.f. Proposition 1 in §4).

The minimum generator for a sequence $\{a_i\}_{i=0}^{\infty}$ of field elements $a_i \in \mathbb{K}$ can be computed by the Berlekamp/Massey algorithm (Massey 1969). This algorithm will determine the minimum polynomial $f^{(\min)}$ of such a sequence from the first $2M$ elements, where $M = \deg(f^{(\min)})$. If more elements are given, the computed minimum polynomial cannot change. Therefore, we have the following lemma.

Lemma 1. *Suppose $\{a_i\}_{i=0}^{\infty}$, where $a_i \in \mathbb{K}$, is linearly generated by the minimum polynomial $f^{(\min)}$. Let $M = \deg(f^{(\min)})$ and let $M' \geq M$. Suppose a polynomial g with $\deg(g) \leq M'$ linearly generates a sequence*

$$\{a_0, a_1, \ldots, a_{2M'-1}, a'_{2M'}, a'_{2M'+1}, \ldots\}$$

whose first $2M'$ elements agree with $\{a_i\}_{i=0}^{\infty}$. Then $a'_i = a_i$ for all $i \geq 2M'$ and g is a polynomial multiple of $f^{(\min)}$.

3 Coppersmith's block Wiedemann algorithm

In order to prepare for later discussion, we first give a particular variant of Wiedemann's (1986, §III, first paragraph) coordinate recurrence method for solving a homogeneous linear system. This variant already accounts for some changes necessitated by the later block version. Let $B \in \mathbb{K}^{N\times N}$ be a singular matrix, where $\mathbb{K}$ is a finite field; we seek a non-zero vector $w \in \mathbb{K}^N$ such that $Bw = 0$.

Step W1: Pick random vectors $u^{\mathrm{tr}} \in \mathbb{K}^N$ and $v \in \mathbb{K}^N$. For any integers $M' \geq M \geq N$, compute

$$b = Bv, \quad a^{(i)} = u^{\mathrm{tr}} B^i b, \quad 0 \leq i \leq M + M' - 1.$$

(The letters u and b now agree with the ones in Wiedemann's paper.) This requires at least $2M + 1$ multiplications of B by vectors.

Step W2: Compute a non-zero solution to the linear homogeneous $M' \times (M+1)$ Toeplitz system

$$\begin{bmatrix} a^{(M)} & \cdots & a^{(1)} & a^{(0)} \\ a^{(M+1)} & a^{(M)} & a^{(2)} & a^{(1)} \\ \vdots & & \ddots & \vdots \\ a^{(M+M'-1)} & \cdots & & a^{(M'-1)} \end{bmatrix} \begin{bmatrix} c^{(M)} \\ c^{(M-1)} \\ \vdots \\ c^{(0)} \end{bmatrix} = 0^{M'},$$

Define the generating polynomial

$$f(\lambda) = c^{(L)}\lambda^L + c^{(L+1)}\lambda^{L+1} + \cdots + c^{(M)}\lambda^{(M)}, \quad c^{(\ell)} = 0 \text{ for } 0 \leq \ell < L,\ c^{(L)} \neq 0.$$

Such a polynomial can be determined, e.g., by the Berlekamp/Massey algorithm, which then requires, for $M' = M = N$, $O(N^2)$ arithmetic operations in $\mathbb{K}$. Here we introduce unnecessary generality for the later analysis of the block Wiedemann method. Note that

$$u^{\mathrm{tr}} B^j f(B) b = 0 \quad \text{for all } 0 \leq j \leq M - 1,$$

which implies that $f(\lambda)$ is a polynomial multiple of $f_u^{B,b}(\lambda)$. With probability no less than $1 - N/\mathrm{card}(\mathbb{K})$, $f(\lambda)$ is a polynomial multiple of the polynomial $f^{B,b}(\lambda)$, i.e.,

$$c^{(L)} B^L b + c^{(L+1)} B^{L+1} b + \cdots + c^{(M)} B^M b = 0. \tag{1}$$

Step W3: Compute

$$\widehat{w} = c^{(L)} v + c^{(L+1)} Bv + \cdots + c^{(M)} B^{M-L} v.$$

This requires at most $M - L$ additional multiplications of B times a vector. One may argue as follows that $\widehat{w} \neq 0^N$ with probability at least $1 - 1/\mathrm{card}(\mathbb{K})$ (Coppersmith 1992): for $v' = v + w_0$, where $w_0 \in \mathrm{kernel}(B)$, the vector $b = Bv'$ and hence the sequence $a^{(i)}$ does not change. However,

$$\begin{aligned} \widehat{w}' &= c^{(L)} v' + c^{(L+1)} Bv' + \cdots + c^{(M)} B^{M-L} v' \\ &= \widehat{w} + c^{(L)} w_0. \end{aligned}$$

Therefore in the set of vectors $v + \mathrm{kernel}(B)$, at most one vector can produce $\widehat{w}' = 0$. Note that the solution $c^{(0)}, \ldots, c^{(M)}$ is computed without any information on w_0.

Suppose now that $\widehat{w} \neq 0^N$. Finally, determine the first integer i such that $B^i\widehat{w} = 0^N$ and return $w = B^{i-1}\widehat{w}$. By (1), this should happen, with high probability, for an integer $i \leq L+1$. At most $L+1$ more multiplications of B by a vector are required.

Let $m, n < N$. Coppersmith's (1991) block version essentially uses

$$\begin{array}{lll} x^{\mathrm{tr}} \in \mathbb{K}^{m\times N} & \text{in place of} & u^{\mathrm{tr}}, \\ z \in \mathbb{K}^{N\times n} & \text{in place of} & v, \quad \text{and} \\ y = Bz \in \mathbb{K}^{N\times n} & \text{in place of} & b = Bv. \end{array}$$

(The letters B, x, y, and z, agree with the ones in Coppersmith's paper.) Thus the sequence is one of the $m \times n$ matrices

$$a^{(i)} = x^{\mathrm{tr}} B^i y \in \mathbb{K}^{m\times n}, \quad 0 \leq i.$$

(Coppersmith further transposes these matrices.) The main point is that a non-trivial linear dependence of the type (1) can be found from roughly $N/m + N/n$ sequence elements $a^{(i)}$. A brief description of a variant of the block Wiedemann algorithm follows:

Step C1: Pick random vectors $x_1, \ldots, x_m, z_1, \ldots, z_n \in \mathbb{K}^N$. Let

$$x^{\mathrm{tr}} = \left[\begin{array}{c} x_1^{\mathrm{tr}} \\ \hline \vdots \\ \hline x_m^{\mathrm{tr}} \end{array}\right], \quad y = B \cdot [\, z_1 \mid \ldots \mid z_n \,].$$

Compute

$$a^{(i)} = x^{\mathrm{tr}} B^i y, \quad \text{for all } 0 \leq i < \frac{N}{m} + \frac{N}{n} + \frac{2n}{m} + 1.$$

This requires less than

$$\left(1 + \frac{n}{m}\right) N + \frac{2n^2}{m} + 2n \tag{2}$$

multiplications of B times a vector. However, for every y_ν, the ν^{th} columns of the sequence matrices $a^{(i)}$, namely $x^{\mathrm{tr}} B^i y_\nu$, can be computed simultaneously, yielding a coarse-grain parallelization. Alternately, one may for each i perform the products $B \cdot B^{i-1} y_\nu$ in parallel, as Coppersmith does, which is finer grain and requires synchronization for each i. Note that the products $x^{\mathrm{tr}} \cdot (B^i y_\nu)$ additionally require for all ν some

$$O(\,(m+n)\,N^2\,)$$

arithmetic operations in $\mathbb{K}$, if done sequentially.

Step C2: Let $D = \lceil N/n \rceil$, $S = n(D+1)$, which is bounded as $N + n \le S < N + 2n$, and let $E = \lceil S/m \rceil$, $R = mE$, which are bounded as $S \le R$ and $E < N/m + 2n/m + 1$. Compute a non-zero solution to the linear homogeneous $R \times S$ linear system (of block Toeplitz structure)

$$\left[\begin{array}{c|c|c|c|c} a^{(D)} & \dots & & a^{(1)} & a^{(0)} \\ \hline a^{(D+1)} & a^{(D)} & & a^{(2)} & a^{(1)} \\ \hline \vdots & & \ddots & & \vdots \\ \hline a^{(D+E-1)} & \dots & & & a^{(E-1)} \end{array}\right] \left[\begin{array}{c} c^{(D)} \\ \hline c^{(D-1)} \\ \hline \vdots \\ \hline c^{(0)} \end{array}\right] = 0^R, \quad c^{(i)} = \left[\begin{array}{c} c_1^{(i)} \\ \vdots \\ c_n^{(i)} \end{array}\right] \in \mathbb{K}^n. \tag{3}$$

Note that

$$D + E < \frac{N}{n} + 1 + \frac{N}{m} + \frac{2n}{m} + 1,$$

which bounds the length of the sequence $a^{(i)}$. Define the generating polynomial with (right-hand-side) vector coefficients

$$f(\lambda) = \lambda^L y c^{(L)} + \lambda^{L+1} y c^{(L+1)} + \cdots + \lambda^D y c^{(D)},\ c^{(\ell)} = 0^n \text{ for } 0 \le \ell < L,\ c^{(L)} \ne 0^n.$$

Coppersmith in his paper computes such a non-zero vector polynomial by his generalization of the Berlekamp/Massey algorithm to polynomials with matrix coefficients. In any case, we need to have

$$x^{\mathrm{tr}} B^j f(B) = 0^m \quad \text{for all } 0 \le j \le E - 1.$$

As we will argue later, with high probability the projections by x^{tr} do not introduce any additional linear dependence, so that

$$f(B) = B^L y c^{(L)} + B^{L+1} y c^{(L+1)} + \cdots + B^D y c^{(D)} = 0^N. \tag{4}$$

Step C3: Compute

$$\widehat{w} = z c^{(L)} + B z c^{(L+1)} + \cdots + B^{D-L} z c^{(D)}.$$

This requires at most $D - L$ additional multiplications of B times a vector (using a Horner evaluation scheme). One may argue as above that $\widehat{w} \ne 0^N$ with probability at least $1 - 1/\mathrm{card}(\mathbb{K})$ (see also proof of Theorem 1 in §4). Suppose now that $\widehat{w} \ne 0^N$. Finally, determine the first integer i such that $B^i \widehat{w} = 0^N$ and return $w = B^{i-1}\widehat{w}$. By (4), this should happen, with high probability, for an integer $i \le L + 1$. At most $L + 1$ more multiplications of B by a vector are required. Altogether, this step performs

$$D + 1 < \frac{N}{n} + 2 \tag{5}$$

multiplications of B by a vector, and additionally $O(N^2)$ arithmetic operations in $\mathbb{K}$ are required to compute $z c^{(i)}$ for $L \le i \le D$ and add the $D - L + 1$ vectors in the Horner scheme.

Coppersmith's paper raises two distinct problems:

1. The efficient computation of a non-trivial solution to (3). He proposes a clever generalization of the Berlekamp/Massey algorithm to linearly generated sequences of matrices. Although one can define the notion of a minimum generator, a proof that the algorithm produces it has so far eluded us. However, we may proceed directly by computing a non-trivial solution of our system by either a method for Toeplitz-like matrices or by the Wiedemann algorithm itself and by using a fast polynomial (over $\mathbb{K}$) multiplication algorithm (see §5 and §6).

2. The probabilistic analysis, in particular the fact that with high probability the found polynomial $f(\lambda)$ satisfies (4). We will show this to be true at least in the case that the minimum polynomial f^B of the coefficient matrix B has degree $\deg(f^B) = \operatorname{rank}(B) + 1$. Fortunately, by certain randomizations this condition can be enforced for any matrix B (Kaltofen and Saunders 1991; see also the proof of Theorem 2 in §5). Let us consider, e.g., solving a non-singular system $x = A^{-1}b$. We then can randomize $\widetilde{A} = AVG$ where V is random unit triangular Toeplitz matrix and G is random diagonal matrix, and execute the block Wiedemann method on the $(N+1) \times (N+1)$ matrix

$$B = \left[\begin{array}{c|c} \widetilde{A} & b \\ \hline 0^{1\times N} & 0 \end{array}\right].$$

   Note that $\widetilde{A}$ has with high probability n distinct eigenvalues.

4 Probabilistic analysis

We now justify Coppersmith's block version of the Wiedemann We will prove the following theorem.

Theorem 1. *Let $\mathbb{K}$ be a finite field, and let $B \in \mathbb{K}^{N\times N}$ be a singular matrix whose minimal polynomial f^B has degree*

$$\deg(f^B) = \operatorname{rank}(B) + 1.$$

Suppose the vector blocks $x^{\mathrm{tr}} \in \mathbb{K}^{m\times N}$ and $z \in \mathbb{K}^{N\times n}$ are chosen at random. Suppose $\widehat{w} \in \mathbb{K}^N$ is computed by steps (C1)–(C3) of §3. Then with probability no less than

$$1 - \frac{2\operatorname{rank}(B) + 1}{\operatorname{card}(\mathbb{K})} \geq 1 - \frac{2N-1}{\operatorname{card}(\mathbb{K})}$$

we have

$$\widehat{w} \neq 0^N, \quad B^{L+1}\widehat{w} = 0^N \text{ for some integer } L \leq \lceil N/n \rceil.$$

The key property for the algorithm to succeed is equation (4). We will prove (4) first if the entries in x and z are indeterminates $\xi_{\iota,\mu}$ and $\zeta_{\iota,\nu}$, where $1 \le \mu \le m$, $1 \le \nu \le n$, and $1 \le \iota \le N$. In this case, the algorithm is performed over the multivariate rational function field over $\mathbb{K}$,

$$\mathbb{L} = \mathbb{K}(\xi_{1,1}, \ldots, \xi_{N,m}, \zeta_{1,1}, \ldots, \zeta_{N,n}).$$

In order to distinguish when the algorithm is performed over $\mathbb{K}$ and when over $\mathbb{L}$, we will write $\mathcal{X}$ and $\mathcal{Z}$ for the undetermined x and z and

$$\mathcal{Y} = B\mathcal{Z} \in \mathbb{L}^{N\times n}, \quad \mathcal{A}^{(i)} = \mathcal{X}^{\mathrm{tr}} B^i \mathcal{Y} \in \mathbb{L}^{m\times n}.$$

The equation (4) is equivalent to the solution vector c of (3) satisfying the following block Krylov system:

$$[\, B^{D+1}z \mid \ldots \mid B^2 z \mid Bz \,] \begin{bmatrix} c^{(D)} \\ \hline c^{(D-1)} \\ \hline \vdots \\ \hline c^{(0)} \end{bmatrix} = 0. \tag{6}$$

Clearly, any solution of (6) also solves (3). We first state that generically, i.e., over $\mathbb{L}$, no other solutions to (3) exists. We will prove this fact later using Proposition 1 stated below.

Proposition 2. *Suppose that the minimum polynomial f^B of B has the degree*

$$\deg(f^B) = \min\{N, \mathrm{rank}(B) + 1\}.$$

Then for $D = \lceil N/n \rceil$ and $E = \lceil n(D+1)/m \rceil$ we have the rank equalities

$$\begin{aligned} \mathrm{rank}\Bigg(& \left[\begin{array}{c|c|c|c|c} \mathcal{A}^{(D)} & \ldots & & \mathcal{A}^{(1)} & \mathcal{A}^{(0)} \\ \hline \mathcal{A}^{(D+1)} & \mathcal{A}^{(D)} & & \mathcal{A}^{(2)} & \mathcal{A}^{(1)} \\ \hline \vdots & & \ddots & & \vdots \\ \hline \mathcal{A}^{(D+E-1)} & \ldots & & & \mathcal{A}^{(E-1)} \end{array}\right] \Bigg) \\ & = \mathrm{rank}([\, B^D\mathcal{Y} \mid B^{D-1}\mathcal{Y} \mid \ldots \mid B\mathcal{Y} \mid \mathcal{Y} \,]) \\ & = \mathrm{rank}(B) = \begin{cases} N & \text{if } B \text{ is non-singular,} \\ \deg(f^B) - 1 & \text{if } B \text{ is singular.} \end{cases} \end{aligned}$$

The proof of this proposition is based on its validity for $m = n = 1$, which we shall formulate as our first proposition. We will denote our generic sequence by

$$\alpha^{(i)} = \mathcal{A}^{(i)}_{1,1} = \mathcal{X}_1^{\mathrm{tr}} B^{i+1} \mathcal{Z}_1 \in \mathbb{L}, \quad \text{for } i \ge 0.$$

Proposition 1. *Let $M' \geq M \geq N$. Define*

$$\mathcal{T} = \begin{bmatrix} \alpha^{(M)} & \dots & \alpha^{(1)} & \alpha^{(0)} \\ \alpha^{(M+1)} & \alpha^{(M)} & \alpha^{(2)} & \alpha^{(1)} \\ \vdots & & \ddots & \vdots \\ \alpha^{(M+M'-1)} & \dots & & \alpha^{(M'-1)} \end{bmatrix} \in \mathbb{L}^{M' \times (M+1)}$$

and

$$\mathcal{K} = [\, B^M \mathcal{Z}_1 \mid \dots \mid B^2 \mathcal{Z}_1 \mid B\mathcal{Z}_1 \,] \in \mathbb{L}^{N \times M}.$$

Then

$$\operatorname{rank}(\mathcal{T}) = \operatorname{rank}(\mathcal{K}) = \begin{cases} \deg(f^B) & \text{if } B \text{ is non-singular,} \\ \deg(f^B) - 1 & \text{if } B \text{ is singular.} \end{cases}$$

Proof. The argument can be deduced from the probabilistic analysis of Wiedemann (1986, §V and §VI). Since $f^B(\lambda)$ linearly generates the sequence

$$\{B^i \mathcal{Z}_1\}_{i=0}^{\infty}, \quad \text{where } B^i \mathcal{Z}_1 \in \mathbb{L}^N,$$

we must have the rank inequality

$$\operatorname{rank}([\, B^M \mathcal{Z}_1 \mid \dots \mid B\mathcal{Z}_1 \mid \mathcal{Z}_1 \,]) \leq \deg(f^B)$$

for any vector $\mathcal{Z}_1$ and any integer $M \geq N$. Moreover, there exists a specialization of $\mathcal{Z}_1$ to a vector $z_1 \in \mathbb{K}^N$ such $f^B(\lambda)$ is the minimum linear generating polynomial of the sequence

$$\{B^i z_1\}_{i=0}^{\infty}, \quad \text{where } B^i z_1 \in \mathbb{K}^N,$$

hence

$$\operatorname{rank}([\, B^N z_1 \mid \dots \mid B z_1 \mid z_1 \,]) = \deg(f^B),$$

and therefore generically the rank cannot be lower. The existence of such a vector z_1 follows, e.g., by considering the rational canonical form (Frobenius form) of B. Certainly, for B in companion form

$$B = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & & 1 \\ b_{N,1} & b_{N,2} & \dots & & b_{N,N} \end{bmatrix} \text{ we may choose } z_1 = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix},$$

and this argument extends to the block diagonal shape with companion blocks of the rational canonical form. If B is non-singular, the minimum generating polynomial of

$$\{B^i \mathcal{Y}_1\}_{i=0}^{\infty} = \{B^{i+1} \mathcal{Z}_1\}_{i=0}^{\infty}$$

does not change, while for singular B the minimum generating polynomial is $f^B(\lambda)/\lambda$, thus the rank drops by 1. We define this polynomial as

$$f_-^B(\lambda) = \begin{cases} f^B(\lambda) & \text{if } B \text{ is non-singular,} \\ f^B(\lambda)/\lambda & \text{if } B \text{ is singular.} \end{cases}$$

So far, we have shown that $\mathrm{rank}(\mathcal{K}) = \deg(f_-^B)$.

Second, we need to prove that $\mathrm{rank}(\mathcal{T}) = \mathrm{rank}(\mathcal{K})$. The proof of this is very similar. It follows from Wiedemann (1986, §VI) that there exists a vector $x_1 \in \mathbb{K}^N$ such that the sequence

$$\{x_1^{\mathrm{tr}} B^{i+1} z_1\}_{i=0}^{\infty}$$

has the same minimum generator as

$$\{B^{i+1} z_1\}_{i=0}^{\infty},$$

which is $f_-^B(\lambda)$. Furthermore, $f_-^B(\lambda)$ is already a generating polynomial of the generic sequence

$$\{\alpha^{(i)}\}_{i=0}^{\infty}, \tag{7}$$

and must therefore also be minimal for that sequence, because a specialization is.

We finally argue that the rank of $\mathcal{T}$ is the degree of the minimum generator of (7), namely that $\mathrm{rank}(\mathcal{T}) = \deg(f_-^B)$. Consider any non-zero solution in $\mathbb{L}^{M+1}$ of

$$\mathcal{T}\gamma = \begin{bmatrix} \alpha^{(M)} & \cdots & \alpha^{(1)} & \alpha^{(0)} \\ \alpha^{(M+1)} & \alpha^{(M)} & \alpha^{(2)} & \alpha^{(1)} \\ \vdots & & \ddots & \vdots \\ \alpha^{(M+M'-1)} & \cdots & & \alpha^{(M'-1)} \end{bmatrix} \begin{bmatrix} \gamma^{(M)} \\ \gamma^{(M-1)} \\ \vdots \\ \gamma^{(0)} \end{bmatrix} = 0^{M'}. \tag{8}$$

Then for all $j = 0, \ldots, M-1 \le M'-1$

$$\alpha^{(M+j)}\gamma^{(M)} + \cdots + \alpha^{(j)}\gamma^{(0)} = 0,$$

hence the polynomial

$$\varphi(\lambda) = \gamma^{(M)}\lambda^M + \cdots + \gamma^{(1)}\lambda + \gamma^{(0)} \in \mathbb{L}[\lambda]$$

generates the entire sequence (7) (Lemma 1 in §2). This implies that f_-^B divides φ, so φ is in the linear span over $\mathbb{L}$ of

$$f_-^B(\lambda), \lambda f_-^B(\lambda), \lambda^2 f_-^B(\lambda), \ldots, \lambda^\delta f_-^B(\lambda), \quad \text{where } \delta = M - \deg(f_-^B).$$

Also, any coefficient vector of a polynomial in that linear span generates the sequence, thus solves (8). Therefore the rank of $\mathcal{T}$,

$$M + 1 - \text{the dimension of kernel}(\mathcal{T}),$$

is equal to $\deg(f_-^B)$. $\boxtimes$

Proof of Proposition 2. Consider the specialization

$$\mathcal{Z}' = [\, \mathcal{Z}_1 \mid \underbrace{B^{D+1}\mathcal{Z}_1}_{\mathcal{Z}'_2} \mid \underbrace{B^{2(D+1)}\mathcal{Z}_1}_{\mathcal{Z}'_3} \mid \ldots \mid \underbrace{B^{(n-1)(D+1)}\mathcal{Z}_1}_{\mathcal{Z}'_n} \,].$$

Then the set of columns in

$$[\, B^D\mathcal{Z}' \mid B^{D-1}\mathcal{Z}' \mid \ldots \mid \mathcal{Z}' \,]$$

is equal to

$$\{\mathcal{Z}_1, B\mathcal{Z}_1, B^2\mathcal{Z}_1, \ldots, B^{n(D+1)-1}\mathcal{Z}_1\}.$$

Since $n(D+1) > N$, this set has rank equal $\deg(f^B)$, as is argued in the proof of Propostition 1. Therefore the "more generic" matrix

$$[\, B^D\mathcal{Z} \mid B^{D-1}\mathcal{Z} \mid \ldots \mid \mathcal{Z} \,]$$

also has rank greater equal $\deg(f^B)$. Now define

$$\mathcal{K}^{\boxplus} = [\, B^D\mathcal{Y} \mid B^{D-1}\mathcal{Y} \mid \ldots \mid \mathcal{Y} \,] = B \cdot [\, B^D\mathcal{Z} \mid B^{D-1}\mathcal{Z} \mid \ldots \mid \mathcal{Z} \,],$$

which thus satisfies $\mathrm{rank}(\mathcal{K}^{\boxplus}) \leq \mathrm{rank}(B)$. If B is non-singular, the matrix $\mathcal{K}^{\boxplus}$ actually has full rank N, since by assumption $\deg(f^B) = N$. From Proposition 1 we further get for a singular B that

$$\deg(f^B) - 1 = \mathrm{rank}([\, B^{D+1}\mathcal{Z}' \mid B^D\mathcal{Z}' \mid \ldots \mid B\mathcal{Z}' \,]) \leq \mathrm{rank}(\mathcal{K}^{\boxplus}),$$

hence

$$\deg(f^B) - 1 \leq \mathrm{rank}(\mathcal{K}^{\boxplus}) \leq \mathrm{rank}(B),$$

which implies by the assumption of the theorem that $\mathrm{rank}(\mathcal{K}^{\boxplus}) = \mathrm{rank}(B)$. Furthermore, if B is singular and $\deg(f^B) = N$ it follows that $\mathrm{rank}(B) = N-1 = \deg(f^B) - 1$.

We will use a similar specialization for the columns of $\mathcal{X}$ to establish that the rank of

$$\mathcal{T}^{\boxplus} = \left[\begin{array}{c|c|c|c|c} \mathcal{A}^{(D)} & \ldots & & \mathcal{A}^{(1)} & \mathcal{A}^{(0)} \\ \hline \mathcal{A}^{(D+1)} & \mathcal{A}^{(D)} & & \mathcal{A}^{(2)} & \mathcal{A}^{(1)} \\ \hline \vdots & & \ddots & & \vdots \\ \hline \mathcal{A}^{(D+E-1)} & \ldots & & & \mathcal{A}^{(E-1)} \end{array}\right] \tag{9}$$

agrees with the rank of $\mathcal{T} \in \mathbb{L}^{M' \times (M+1)}$ of Proposition 1 with the dimensions

$$M = n(D+1) - 1 = S - 1 \geq N$$

and

$$M' = mE = R > S - 1.$$

Consider the specialization $\mathcal{Z}'$ given above, and

$$\mathcal{X}' = [\mathcal{X}_1 \mid \underbrace{B^E \mathcal{X}_1}_{\mathcal{X}'_2} \mid \underbrace{B^{2E} \mathcal{X}_1}_{\mathcal{X}'_3} \mid \ldots \mid \underbrace{B^{(m-1)E} \mathcal{X}_1}_{\mathcal{X}'_n}].$$

Then with

$$\mathcal{A}'^{(i)} = \mathcal{X}'^{\mathrm{tr}} B^{i+1} \mathcal{Z}'$$

there exist permutation matrices $P \in \{0,1\}^{R \times R}$ and $Q \in \{0,1\}^{S \times S}$ such that

$$P \mathcal{T} Q = \left[\begin{array}{c|c|c|c|c} \mathcal{A}'^{(D)} & \ldots & & \mathcal{A}'^{(1)} & \mathcal{A}'^{(0)} \\ \hline \mathcal{A}'^{(D+1)} & \mathcal{A}'^{(D)} & & \mathcal{A}'^{(2)} & \mathcal{A}'^{(1)} \\ \hline \vdots & & \ddots & & \vdots \\ \hline \mathcal{A}'^{(D+E-1)} & \ldots & & & \mathcal{A}'^{(E-1)} \end{array}\right].$$

The row and column permutations move the entry

$$\begin{aligned} \mathcal{A}'^{(D+I-J)}_{i,j} &= \mathcal{X}'^{\mathrm{tr}}_i B^{D+I-J} B \mathcal{Z}'_j \\ &= \mathcal{X}_1^{\mathrm{tr}} B^{(i-1)E} B^{D+1+I-J} B^{(j-1)(D+1)} \mathcal{Z}_1 \end{aligned}$$

in the right hand side block Toeplitz matrix, which is in row $mI + i$, where $0 \le I < E$ and $1 \le i \le m$, and column $nJ + j$, where $0 \le J < D + 1$ and $1 \le j \le n$, to row $E(i-1) + I + 1$ and column $(D+1)(n-j) + J + 1$ in $\mathcal{T}$, namely

$$\begin{aligned} \mathcal{T}_{E(i-1)+I+1,(D+1)(n-j)+J+1} &= \alpha^{(\,M\ +\ E(i-1)+I+1\ -\ ((D+1)(n-j)+J+1)\,)} \\ &= \mathcal{X}_1^{\mathrm{tr}} B^{n(D+1)+E(i-1)+I+(D+1)(j-n)-J} \mathcal{Z}_1. \end{aligned}$$

Therefore, the rank of $\mathcal{T}^{\boxplus}$ is no less than the rank of $\mathcal{T}$ with the given dimensions, which by Proposition 1 and the assumptions is equal to $\deg(f^B) = N$ for non-singular B, and $\deg(f^B) - 1$ for singular B. Since the kernel of $\mathcal{K}^{\boxplus}$ is contained in the kernel of $\mathcal{T}^{\boxplus}$, the rank cannot be more. $\boxtimes$

Proof of Theorem 1. Let

$$\Delta(\xi_{1,1}, \ldots, \xi_{N,m}, \zeta_{1,1}, \ldots, \zeta_{N,n})$$

be a non-zero maximal minor of $\mathcal{T}^{\boxplus}$ in (9). Then for all matrices x and z with

$$\Delta(x_{1,1}, \ldots, x_{N,m}, z_{1,1}, \ldots, z_{N,n}) \neq 0$$

any solution to (3) must also solve (6), because the ranks of both coefficient matrices will be $\deg(f^B) - 1$. Hence

$$Bzc^{(0)} + B^2zc^{(1)} + \cdots + B^{D+1}zc^{(D)} = B^{L+1}\widehat{w} = 0^N$$

for $0 \le L \le D$ such that $c^{(L)} \neq 0$ and $c^{(\ell)} = 0$ for $0 \le \ell < L$. By a lemma of Zippel (1979)/Schwartz (1980) the probability of hitting a zero of Δ is no more than $\deg(\Delta)/\mathrm{card}(\mathbb{K}) \le 2\,\mathrm{rank}(B)/\mathrm{card}(\mathbb{K})$.

It remains to estimate the probability that $\widehat{w} \neq 0$. The argument, by Coppersmith, is as that for step W3. For a matrix $y = Bz \in \mathbb{K}^{N\times n}$ consider the equivalence class

$$\{z' \in \mathbb{K}^{N\times n} \mid y = Bz' = Bz\} \tag{10}$$

of $\mathbb{K}^{N\times n}$. Then for each member in that class

$$\begin{aligned}\widehat{w}' &= z'c^{(L)} + Bz'c^{(L+1)} + \cdots + B^{(D-L)}z'c^{(D)} \\ &= \underbrace{zc^{(L)} + Bzc^{(L+1)} + \cdots + B^{(D-L)}zc^{(D)}}_{\widehat{w}} + (z'-z)c^{(L)},\end{aligned}$$

where

$$z' - z = [\,w_1 \mid w_2 \mid \ldots \mid w_n\,] \quad \text{with } w_\nu \in \mathrm{kernel}(B) \text{ for all } 1 \le \nu \le n.$$

Since, given $c^{(L)} \in \mathbb{K}^n \setminus \{0^n\}$, the linear span

$$c_1^{(L)}w_1 + \cdots + c_n^{(L)}w_n$$

uniformly samples kernel(B) for randomly chosen $w_\nu \in \mathrm{kernel}(B)$, at most a fraction of $1/\mathrm{card}(\mathbb{K})$ matrices in the set (10) can give $-\widehat{w}$ as that linear combination and thus lead to $\widehat{w}' = 0$. Therefore, the probability that $\widehat{w} = 0$ is no more than $1/\mathrm{card}(\mathbb{K})$.

Summing both estimates bounds the probability of failure. ⊠

5 Algorithms and their running times

The block Wiedemann method of §3 is used to solve both non-singular and singular sparse linear systems, i.e., linear systems with an efficient way to multiply the coefficient matrix by any vector. The method is randomized and can be executed sequentially or in parallel. Especially in the latter form the method becomes very efficient. We now present several algorithms that are based on the block Wiedemann algorithm of §3. One main point is that we are able to give both explicit expected running times and estimates on the success probability of the randomizations. We have the following theorem, which focuses on the sequential performance of the blocking. A corollary considering the parallel costs is given below.

Theorem 2. *Let $B \in \mathbb{K}^{N\times N}$ be a singular matrix and $1 \le m, n \le N$. Then one can compute a solution vector $w \in \mathbb{K}^N \setminus \{0^N\}$ with $Bw = 0^N$ in no more than*

$$\left\lfloor \left(1 + \frac{n}{m} + \frac{1}{n}\right) N + \frac{2n^2}{m} + 2n + 2 \right\rfloor$$

multiplications of B times a vector in $\mathbb{K}^N$, and an additional

$$O((m+n)N^2 \log N \operatorname{loglog} N)$$

arithmetic operations in $\mathbb{K}$. The algorithm selects no more than $(m+n+5)N$ random elements in $\mathbb{K}$ and succeeds to produce a solution with probability no less than

$$1 - \frac{3/2\,(N^2+N)}{\operatorname{card}(\mathbb{K})}.$$

The algorithm requires an additional $O((m+n)N)$ amount of storage for field elements in $\mathbb{K}$.

Proof. Consider the perturbed matrix

$$\widetilde{B} = UBVG$$

where $U \in \mathbb{K}^{N\times N}$ is a random unit upper triangular Toeplitz matrix, $V \in \mathbb{K}^{N\times N}$ is a random unit lower triangular Toeplitz matrix, and $G \in \mathbb{K}^{N\times N}$ is a random non-singular diagonal matrix. Then with probability of at least

$$1 - \frac{3(N-1)N/2}{\operatorname{card}(\mathbb{K})}$$

for the minimum polynomial $f^{\widetilde{B}}$ of $\widetilde{B}$ we have

$$\deg(f^{\widetilde{B}}) = \operatorname{rank}(\widetilde{B}) + 1$$

(Kaltofen and Saunders 1991, Theorem 2 and Lemma 2). Also, for a vector $b \in \mathbb{K}^N$ the product $\widetilde{B}b$ can be computed by one multiplication of B by a vector, and an additional $O(N \log N \operatorname{loglog} N)$ arithmetic operations in $\mathbb{K}$. We remark that the use of Beneš networks (Wiedemann 1986, §V) can reduce the latter complexity by the $\operatorname{loglog} N$ factor at the cost of requiring $O(N \log N)$ random field elements.

Now, the matrix $\widetilde{B}$ satisfies the assumptions of Theorem 1, and we can find a non-zero solution $\widetilde{w} \in \mathbb{K}^N \setminus \{0^N\}$ to $\widetilde{B}\,\widetilde{w} = 0^N$. Thus $w = VG\widetilde{w} \neq 0^N$ solves $Bw = 0^N$. By (2) and (5) the method multiplies $\widetilde{B}$ by a vector no more than

$$\left(1 + \frac{n}{m} + \frac{1}{n}\right) N + \frac{2n^2}{m} + 2n + 2$$

many times. The extra work in terms of arithmetic operations in $\mathbb{K}$ is $O((m+n)N^2)$ plus the work it takes to solve (3). Suppose then that we compute a non-zero solution to (3) by the standard Wiedemann method, e.g., by the algorithm described in §3.

The coefficient matrix

$$A = \left[\begin{array}{c|c|c|c|c} a^{(D)} & \cdots & & a^{(1)} & a^{(0)} \\ \hline a^{(D+1)} & a^{(D)} & & a^{(2)} & a^{(1)} \\ \hline \vdots & & \ddots & & \vdots \\ \hline a^{(D+E-1)} & \cdots & & & a^{(E-1)} \end{array}\right]$$

of (3) is not square, which requires some modification. One could, e.g., pre-multiply A by a random unit upper triangular $S \times R$ Toeplitz matrix, which is a rank preserving operation with high probability. However, by inspection of the proof of Propostion 2 we see that we may drop the rows in position R, $R-m, \ldots, R-(R-S-1)m$ without affecting the probabilistic rank estimates. The Wiedemann algorithm requires $\leq 3R$ multiplications of A times a vector and $O(R^2)$ arithmetic operations. We are left with the problem of efficiently performing the matrix times vector multiplication

$$A \cdot \left[\begin{array}{c} b^{(0)} \\ \hline \vdots \\ \hline b^{(D)} \end{array}\right] = \left[\begin{array}{c} b'^{(0)} \\ \hline \vdots \\ \hline b'^{(D)} \end{array}\right], \quad \text{where } b^{(i)}, b'^{(i)} \in \mathbb{K}^n. \tag{11}$$

Consider the polynomial multiplication

$$\begin{aligned}(a^{(0)} + a^{(1)}\lambda + \cdots + a^{(D+E-1)}\lambda^{D+E-1}) \cdot (b^{(0)} + b^{(1)}\lambda + \cdots + b^{(D)}\lambda^D) \\ = \cdots + b'^{(0)}\lambda^D + b'^{(1)}\lambda^{D+1} + \cdots + b'^{(D)}\lambda^{D+E-1} + \cdots\end{aligned}$$

with non-commuting coefficients in the algebras $\mathbb{K}^{m\times n}$ and $\mathbb{K}^n$. By the results in (Kaltofen and Cantor 1991) the product can be found in

$$O(D' \log D' \operatorname{loglog} D'), \quad \text{where } D' = D + E,$$

algebra operations, i.e., additions and subtractions in $\mathbb{K}^{m\times n}$ and $\mathbb{K}^n$, and $O(D' \times \log D')$ multiplications of $m \times n$ matrices by vectors in $\mathbb{K}^n$. These are

$$O((m+n)N \log N \operatorname{loglog} N)$$

arithmetic operations in $\mathbb{K}$ for computing Ab. Alternately, we could have rearranged the rows and columns of A to obtain an $m \times n$ block matrix with $(D+1) \times E$ Toeplitz blocks. $\boxtimes$

Theorem 2 can be employed to solve non-singular systems as outlined in the last paragraph of §3. We shall formulate the result not in terms of the block sizes m and n, but in terms of the quantity

$$\epsilon = \frac{n}{m} + \frac{1}{n}.$$

For suitable constant block sizes ϵ can be made arbitrarily close to 0. Thus we have the following sequential complexity result.

Corollary 1. *Let $B \in \mathbb{K}^{N\times N}$ be a non-singular matrix and let $\epsilon > 0$ be fixed. Then one can compute the solution vector $w = B^{-1}b$ with $b \in \mathbb{K}^N$ in no more than*

$$(1+\epsilon)N + O(1)$$

multiplications of B times a vector in $\mathbb{K}^N$, and an additional

$$O(N^2 \log N \operatorname{loglog} N)$$

arithmetic operations in $\mathbb{K}$. The algorithm selects $O(N)$ random elements in $\mathbb{K}$ and succeeds to produce the solution with probability no less than

$$1 - \frac{N^2 + 4N + 3}{\operatorname{card}(\mathbb{K})}.$$

The algorithm requires an additional $O(N)$ amount of storage for field elements in $\mathbb{K}$. Note that here all big-O estimates depend on ϵ.

Of course, the main application of blocking is to compute the sequence of matrices $a^{(i)}$ in parallel. In order to make the statement of the next corollary simpler, we suppose that $m = n \approx \sqrt{N}$ and that we have n (loosely linked) parallel processors.

Corollary 2. *On $\lceil\sqrt{N}\rceil$ processors one may compute using $O(N\sqrt{N})$ random elements in $\mathbb{K}$ a solution to the linear system $Bw = b$, where $B \in \mathbb{K}^{N\times N}$ and $b \in \mathbb{K}^N$, in $O(\sqrt{N})$ (parallel) multiplications of B times vectors, $O(N^2)$ (parallel) arithmetic operations in $\mathbb{K}$, and an additional $O(N^2\sqrt{N}\log N \operatorname{loglog} N)$ sequential arithmetic operations in $\mathbb{K}$.*

Note that we may also solve the arising block-Toeplitz system (3) in parallel, which improves on the sequential operation count $O(N^2\sqrt{N}\log N \operatorname{loglog} N)$ in the above corollary by a factor of $\sqrt{N}$ as follows. When using the adaption of Wiedemann's method proposed in the proof of Theorem 2, we must first parallelize the computation of (11). For this we employ a parallel version of the algorithm by Cantor and Kaltofen (1991), which with $n = m \approx \sqrt{N}$ processors can compute the matrix times vector product (11) in $O(N \log N \operatorname{loglog} N)$ arithmetic operations in $\mathbb{K}$; this is because the Cantor/Kaltofen algorithm for polynomial multiplication constructs a parallel circuit that is actually of depth $O(\log N)$. Second, we must also parallelize the Berlekamp/Massey step in Wiedemann's algorithm (Step W2 in §3). Again, we can appeal to the parallel implementation of the extended Euclidean algorithm on a systolic array (Brent and Kung 1983), which with $\lceil\sqrt{N}\rceil$ processors finds the needed linear recurrence in $O(N\sqrt{N})$ arithmetic steps (see also Dornstetter 1987). The operations necessary for the evaluation of the generating polynomial at the matrix (see Step W3 in §3) are again parallelizable by using the parallel method for computing products such as (11) discussed before. Altogether, we require with $\lceil\sqrt{N}\rceil$ processors $O(N^2 \log N \operatorname{loglog} N)$ (parallel) arithmetic operations in $\mathbb{K}$ for the solution of (3). However, all these substeps utilize a much more fine grain parallelism than does the parallel computation of the sequence $a^{(i)}$ of Step C1 in §2.

6 Conclusion

Our main contribution in this paper is to give a theoretical basis for the block generalization of the Wiedemann method. We could prove our algorithm for sufficiently large fields and by using certain perturbations of the input matrix. The algorithm may still be valid without the assumptions on the degree of the minimum polynomial. However, Coppersmith also notes that for certain "pathological" cases the straight-forward algorithm might fail to compute a solution. Also, Coppersmith's application to integer factoring has the smallest coefficient field $\mathbb{K} = \mathbb{F}_2$. However, in that situation, Propostion 2 could be relaxed. If the rank of (3) were one or two less than the rank of (6), with probability 1/2 or 1/4 we still would find a solution to (6). For very large finite fields such a rank deficiency would make the problem quite infeasible.

Our algorithms are formulated for finite fields only, but it is not difficult to extend them to fields such as the rational numbers and functions by the use of Chinese remaindering, interpolation, and p-adic lifting (McClellan 1973, Moenck and Carter 1979).

We have used the standard Wiedemann method for analyzing the complexity of computing a solution to (3). There is a slightly faster way of deriving a solution, based on the theory of Toeplitz-like matrices, i.e., matrices with small displacement rank (Kailath et al. 1979). Then it is possible to compute a solution to (3) in $O((m+n)N^2)$ arithmetic operations in $\mathbb{K}$, thus saving the $\log N \operatorname{loglog} N$ factor for Step C2, using the generalized Levinson-Trench algorithm for inverting a matrix of small displacement rank (the condition that all possible leading principal minors are non-zero can be enforced by multiplying with random triangular Toeplitz matrices). Incidentally, Coppersmith's generalized Berlekamp/Massey method has the same asymptotic complexity.

Lobo has implemented several versions of the block Wiedemann algorithm in the programming language C for $\mathbb{K} = \mathbb{F}_p$ and executed it using simultaneously 4 Sun Sparc 2 processors, each rated 28.5MIPS. For $p = 32749$ he can solve a 20K×20K system with 1.32M non-zero entries in about 60 CPU hours. The details of this experiment will be published in a forthcoming paper.

Acknowledgement: Thanks to Austin Lobo for discussions on the theory and implementation of the block Wiedemann method, and to the referee for his comments.

Literature Cited

Brent, R. P. and Kung, H. T., "Systolic VLSI arrays for linear-time GCD computation," *Proc. VLSI '83*, pp. 145–154 (1983).

Cantor, D. G. and Kaltofen, E., "On fast multiplication of polynomials over arbitrary algebras," *Acta Inform.* **28**/7, pp. 693–701 (1991).

Coppersmith, D., "Solving linear equations over GF(2) via block Wiedemann algorithm," *Math. Comput.*, p. to appear (1992).

Dornstetter, J. L., "On the equivalence between Berlekamp's and Euclid's algorithms," *IEEE Trans. Inf. Theory* IT-**33**/3, pp. 428–431 (1987).

Kailath, T., Kung, S.-Y., and Morf, M., "Displacement ranks of matrices and linear equations," *J. Math. Analysis Applications* **68**, pp. 395–407 (1979).

Kaltofen, E. and Saunders, B. D., "On Wiedemann's method of solving sparse linear systems," in *Proc. AAECC-9*, Springer Lect. Notes Comput. Sci. **539**; pp. 29–38, 1991.

Lenstra, A. K., Lenstra, H. W., Manasse, M. S., and Pollard, J. M., "The number field sieve," *Proc. 22nd Annual ACM Symp. Theory Comp.*, pp. 564–572 (1990).

Massey, J. L., "Shift-register synthesis and BCH decoding," *IEEE Trans. Inf. Theory* **IT-15**, pp. 122–127 (1969).

McClellan, M. T., "The exact solution of systems of linear equations with polynomial coefficients," *J. ACM* **20**, pp. 563–588 (1973).

Moenck, R. T. and Carter, J. H., "Approximate algorithms to derive exact solutions to systems of linear equations," *Proc. EUROSAM '79, Springer Lec. Notes Comp. Sci.* **72**, pp. 65–73 (1979).

Schwartz, J. T., "Fast probabilistic algorithms for verification of polynomial identities," *J. ACM* **27**, pp. 701–717 (1980).

Wiedemann, D., "Solving sparse linear equations over finite fields," *IEEE Trans. Inf. Theory* **IT-32**, pp. 54–62 (1986).

Zippel, R., "Probabilistic algorithms for sparse polynomials," *Proc. EUROSAM '79, Springer Lec. Notes Comp. Sci.* **72**, pp. 216–226 (1979).