A Block Wiedemann Rank Algorithm

William J. Turner
Department of Mathematics & Computer Science
Wabash College
Crawfordsville, IN 47933 USA
turnerw@wabash.edu

ABSTRACT

This paper makes two contributions to block Wiedemann algorithms. We describe how to compute the minimal generating matrix polynomial using Beckermann and Labahn's Fast Power Hermite-Padé Solver, and we develop a block Monte Carlo method to compute rank of a black box matrix over a large field by extending the Kaltofen-Saunders black box matrix rank algorithm.

Categories and Subject Descriptors

I.1.2 [**Symbolic and Algebraic Manipulation**]: Algorithms—*algebraic algorithms*

General Terms

Algorithms, Performance, Reliability, Theory.

Keywords

Black box linear algebra, block Wiedemann method, power Hermite-Padé approximation, rank algorithm

1. INTRODUCTION

Coppersmith (1994) introduces blocking to the Wiedemann method to allow parallelization. This block method replaces the scalar sequence of Wiedemann (1986) with a matrix sequence that is linearly generated by not only a scalar polynomial, but also by vector and matrix polynomials. It also has both a minimal generating polynomial and a minimal generating matrix polynomial. In Section 2, we review the relevant interpretations of the block Wiedemann method and the extension of the description of the blocked sequences to general linearly generated matrix sequences and their minimal generating matrix polynomials.

To compute the minimal generating matrix polynomial of the block Wiedemann sequence, Coppersmith (1994) uses a multivariate generalization of the Berlekamp-Massey algorithm, and Kaltofen (1995) solves a homogeneous block Toeplitz system. Although block Berlekamp-Massey algorithms have been proved to correctly compute the minimal generating matrix polynomial of a well-behaved block Wiedemann sequence, no proof for arbitrary block Wiedemann sequences is known. Villard (1997b, p. 12) proposes using the Fast Power Hermite-Padé Solver (FPHPS) algorithm of Beckermann and Labahn (1994) to compute the minimal generating matrix polynomial without a full description and proof of the technique. Section 3 describes such an approach, which has potential applications that include a reliable way to incorporate early termination into a block Wiedemann algorithm.

Kaltofen and Saunders (1991, §4) describe an algorithm that is asymptotically faster than the binary search algorithm Wiedemann (1986) proposes to compute the rank of a black box matrix over large fields. They first precondition the matrix to place it into a generic rank profile and then apply a diagonal multiplier so that the rank of the original singular matrix A is one less than the degree of the minimal polynomial of the preconditioned matrix $\tilde{A}$ with high probability. Eberly (2004) discusses a block Lanczos rank algorithm; however, no block Wiedemann rank algorithm is known. In Section 4, we extend the Kaltofen-Saunders rank algorithm into a Monte Carlo block Wiedemann algorithm for computing the rank of a black box matrix, which has the advantage over a block Lanczos method of allowing the use of rectangular matrices as blocks.

2. MATRIX SEQUENCES

In this section we review the relevant interpretations of the block Wiedemann method, including the works of Villard (1997b) and Kaltofen and Villard (2001, 2004), and their extension to the general linearly generated matrix sequences (Turner, 2002).

Consider the matrix sequence

$$\{B_i\}_{i=0}^{\infty} \in (\mathbb{F}^{\beta_l \times \beta_r})^{\mathbb{Z}_{\geq 0}} \tag{2.1}$$

over a field $\mathbb{F}$ with $\beta_l, \beta_r > 0$. From standard recursion theory, we say the nonzero (scalar) polynomial $g = \sum_{i=0}^{d} g_i \lambda^i \in \mathbb{F}[\lambda]$ linearly generates the matrix sequence if for every $j \geq 0$, $\sum_{i=0}^{d} g_i B_{i+j} = 0^{\beta_l \times \beta_r}$ (Kaltofen, 1995, §3). We say the polynomial g is a generating (scalar) polynomial for the matrix sequence.

In particular, for a matrix $A \in \mathbb{F}^{n \times n}$, we are interested in the block Krylov sequence

$$\{A^i Y\}_{i=0}^{\infty} \in (\mathbb{F}^{n \times \beta_r})^{\mathbb{Z}_{\geq 0}} \tag{2.2}$$

ISSAC'06, July 9–12, 2006, Genova, Italy.

and the block Wiedemann sequence

$$\{X^T A^i Y\}_{i=0}^{\infty} \in (\mathbb{F}^{\beta_l \times \beta_r})^{\mathbb{Z}_{\geq 0}} \tag{2.3}$$

arising from the block Wiedemann method with general block projections $X \in \mathbb{F}^{n \times \beta_l}$ and $Y \in \mathbb{F}^{n \times \beta_r}$. Because the minimal polynomial f^A of the matrix A generates the matrix power sequence

$$\{A^i\}_{i=0}^{\infty} \in (\mathbb{F}^{n \times n})^{\mathbb{Z}_{\geq 0}},$$

f^A must also generate the block Krylov and Wiedemann sequences as well.

In a similar fashion, the nonzero vector polynomial

$$C = \sum_{i=0}^{d} C_i \lambda^i \in \mathbb{F}^{\beta_r}[\lambda] \tag{2.4}$$

and the nonsingular matrix polynomial

$$G = \sum_{i=0}^{d} G_i \lambda^i \in \mathbb{F}^{\beta_r \times \beta_r}[\lambda] \tag{2.5}$$

linearly generate the matrix sequence from the right if

$$\sum_{i=0}^{d} B_{i+j} C_i = 0^{\beta_l} \quad \text{and} \quad \sum_{i=0}^{d} B_{i+j} G_i = 0^{\beta_l \times \beta_r}$$

for every $j \geq 0$, respectively. We say C and G are right generating vector and matrix polynomials, respectively, for the matrix sequence. Similarly, we can define a left generating vector and matrix polynomials by multiplying the matrices by the vector and matrix coefficients on the left. In this paper, we will consider only right generating vector and matrix polynomials, and for convenience we will consider any generating vector and matrix polynomials to linearly generate the matrix sequence from the right unless otherwise specified.

The set of any right generating vector polynomials of the matrix sequence (2.1) forms a submodule of the module of vector polynomials $\mathbb{F}^{\beta_r}$ over the polynomials $\mathbb{F}[\lambda]$ (Turner, 2002, Lem. 4.1). This means that if the matrix polynomial G generates the matrix sequence from the right, then the matrix polynomial GM also generates the sequence from the right where M is any nonsingular matrix polynomial of the proper dimensions (Turner, 2002, Cor. 4.1). We can then show a matrix sequence has a generating scalar polynomial if and only if it has a generating matrix polynomial (Turner, 2002, Thm. 4.1), which means the three notions of linearly generated sequences—by scalar, vector, and matrix polynomials—are equivalent. We say such a matrix sequence is linearly generated.

This submodule of right generating vector polynomials also has a basis of β_r elements over the field of rational functions $\mathbb{F}(\lambda)$ (Turner, 2002, Lem. 4.2). The matrices corresponding to all such integral bases are right equivalent with respect to multiplication on the right by a unimodular matrix (Turner, 2002, Thm. 4.2). Popov (1970) introduces a canonical form for this right equivalence, and following the lead of Villard (1997b, Def. 2.5), we can define the (right) minimal generating matrix polynomial F for the linearly generated matrix sequence (2.1) as the Popov canonical form of the matrices whose columns are the basis elements over $\mathbb{F}[\lambda]$ of the module of right generating vector polynomials of the matrix sequence (Turner, 2002, Def. 4.5). We denote the minimal generating matrix polynomials for the block Krylov (2.2) and Wiedemann (2.3) sequences as $F^{A,Y}$ and $F_X^{A,Y}$, respectively.

Let us denote the degree of the minimal generating matrix polynomial by $\gamma_r = \deg(F)$. Because of the Popov form,

$$\deg(FC) = \max_{1 \leq i \leq \beta_r} \{\deg(F_{[i]}) + \deg(C_{[i]})\}$$

for any nonzero vector C (2.4), where $F_{[i]}$ denotes the ith column of the matrix F and $C_{[i]}$ denotes the ith entry of the vector C (Turner, 2002, Cor. 4.3). Thus, the columns of F form a minimal basis for the module of generating vector polynomials for the matrix sequence (2.1) (Villard, 1997a, Thm. 2). Given any (nonsingular) matrix polynomial G (2.5) that generates the matrix sequence (2.1), there exists some nonsingular matrix polynomial M such that $G = FM$ and both

$$\deg(\det(G)) \geq \deg(\det(F)) \text{ and } \deg(G) \geq \deg(F)$$

(Turner, 2002, Thm. 4.3). This means if the scalar polynomial $g \in \mathbb{F}[\lambda]$ linearly generates the matrix sequence (2.1), $\gamma_r = \deg(F) \leq \deg(g)$, and the minimal generating matrix polynomial has degree no greater than that of the minimal generating scalar polynomial (Turner, 2002, Cor. 4.4).

In general, we do not know any bounds on $\deg(F)$ or $\deg(\det(F))$, but because the minimal polynomial f^A of the matrix A generates the the block Krylov sequence (2.2), we have the degree bound $\deg(F^{A,Y}) \leq \deg(f^A) \leq n$. In addition, the ith largest invariant factor $s_{n-i+1}(F^{A,Y})$ of $F^{A,Y}$ divides the ith largest invariant factor $s_{n-i+1}(\lambda I - A)$ of the characteristic matrix $\lambda I - A$ (Kaltofen and Villard, 2001, Thm. 1), so $\deg(\det(F^{A,Y})) \leq \nu$ where ν is the sum of the degrees of the β_r largest invariant factors of $\lambda I - A$:

$$\nu = \sum_{i=0}^{\beta_r - 1} \deg(s_{n-i}(\lambda I - A)) \leq n. \tag{2.6}$$

Furthermore, because any vector polynomial that generates the block Krylov sequence generates the block Wiedemann sequence (2.3), $F^{A,Y}$ must generate the block Wiedemann sequence from the right. Then,

$$\deg(\det(F_X^{A,Y})) \leq \deg(\det(F^{A,Y})) \leq \nu \leq n$$

and

$$\deg(F_X^{A,Y}) \leq \deg(F^{A,Y}) \leq \deg(f^A) \leq n$$

(Turner, 2002, Thm. 4.11).

Returning again to the general case, and following the example set by Villard (1997b), let us define the block Hankel matrix

$$H(\nu_l, \nu_r) = \begin{bmatrix} B_0 & B_1 & \cdots & B_{\nu_r - 1} \\ B_1 & B_2 & \cdots & B_{\nu_r} \\ \vdots & \vdots & \ddots & \vdots \\ B_{\nu_l - 1} & B_{\nu_l} & \cdots & B_{\nu_l + \nu_r - 2} \end{bmatrix}. \tag{2.7}$$

Then, we can show $\operatorname{rank}(H(\nu_l, \nu_r)) = \operatorname{rank}(H(\nu_l, \gamma_r))$ for all $\nu_l \geq 1$ and $\nu_r \geq \gamma_r$ (Turner, 2002, Lem. 4.5). Thus, γ_r bounds the required number of columns in the block Hankel matrix (2.7).

Let γ_l be the smallest positive integer such that the block Hankel matrix $H(\gamma_l, \gamma_r + 1)$ has maximal rank. In other words,

$$\operatorname{rank}(H(\nu_l, \gamma_r + 1)) = \operatorname{rank}(H(\gamma_l, \gamma_r + 1))$$

for any $\nu_l \geq \gamma_l$. We can show $\gamma_l \leq \deg(g)$ where g is any polynomial that linearly generates the matrix sequence (2.1) (Turner, 2002, Thm. 4.4). In particular, γ_l is no larger than the degree of the minimal generating scalar polynomial. This means $\gamma_l \leq \deg(f^A) \leq n$ for both the block Krylov and block Wiedemann sequences. We cannot find a better bound for the block Wiedemann sequence with a general block left projection X, but we can find $\gamma_l = 1$ for the block Krylov sequence (Turner, 2002, Thm. 4.12).

These bounds for γ_l and γ_r hold for any block projections X and Y. As we shall see in Section 4, the bounds will be much smaller over a large field $\mathbb{F}$ with a high probability.

The definition of γ_l and the maximality of γ_r means the block Hankel matrices $H(\nu_l, \nu_r + 1)$ and $H(\gamma_l, \gamma_r + 1)$ have the same rank for any $\nu_l \geq \gamma_l$ and $\nu_r \geq \gamma_r$, which in turn means γ_l determines how much of the matrix sequence (2.1) we require to decide whether a vector polynomial generates the matrix sequence. In particular, if $\nu_l \geq \gamma_l$, the vector polynomial C (2.4) generates the matrix sequence from the right if and only if

$$\sum_{j=0}^{d} B_{i+j} C_j = 0^{\beta_l}, \qquad 0 \leq i \leq \nu_l - 1 \tag{2.8}$$

(Turner, 2002, Thm. 4.5). Because we do not require $C_d \neq 0$, we only know $\deg(C) \leq d$. Because the minimal generating matrix polynomial F of the matrix sequence has degree γ_r, we can set $d = \gamma_r$ and find a basis for the solutions of the $\beta_l\gamma_l \times \beta_r(\gamma_r + 1)$ homogeneous block Hankel system defined by $H(\gamma_l, \gamma_r + 1)$ to find a basis over $\mathbb{F}[\lambda]$ for the generating vector polynomial C (2.4) for the generating vector polynomials of the linearly generated matrix sequence (2.1) and thus the sequence's minimal generating matrix polynomial F.

In the scalar case, when $\beta_l = \beta_r = 1$, the Berlekamp-Massey algorithm finds the minimal generating polynomial f of the sequence by solving a modular equivalence involving the reversal of f. Recall the reversal of a polynomial g with respect to the degree d for $d \geq \deg(g)$ is $\mathrm{rev}_d(g) = \lambda^d g(1/\lambda)$ (von zur Gathen and Gerhard, 2003, p.254). Similarly, the reversal of a vector polynomial C with respect to the degree d for $d \geq \deg(C)$ is $\mathrm{rev}_d(C) = \lambda^d C(1/\lambda)$ and has degree at most d: $\deg(\mathrm{rev}_d(C)) \leq d$.

In general, if $\nu_l \geq \gamma_l$ and $\nu_r \geq d$, then C (2.4) is a right generating vector polynomial for the matrix sequence if and only if its vector polynomial reversal $\hat{C} = \mathrm{rev}_d(C)$ with respect to degree d satisfies the equivalence relation

$$\left(\sum_{i=0}^{\nu_l+\nu_r-1} B_i \lambda^i \right) \hat{C} \equiv C^{(\mathrm{res})} \pmod{\lambda^{\nu_l+\nu_r}} \tag{2.9}$$

for a vector polynomial $C^{(\mathrm{res})}$ of degree at most $d-1$ (Turner, 2002, Thm. 4.6). Again, this does not require $C_d \neq 0$, so we only know $\deg(C) \leq d$. Because the minimal generating matrix polynomial F of the matrix sequence has degree γ_r, the equivalence relation (2.9) with $\nu_l \geq \gamma_l$ and $\nu_r \geq \gamma_r$ must hold for every column C of F (Turner, 2002, Cor. 4.6). We will use these results in Section 3 to compute the minimal generating matrix polynomial F.

The rank of the block Hankel matrix (2.7) also gives a lower bound on the determinantal degree of the minimal generating matrix polynomial F that we will use to prove the block Wiedemann rank algorithm. Namely, the rank of the block Hankel matrix $H(\nu_l, \nu_r+1)$ is at most the determinantal degree of the minimal generating matrix polynomial F of the matrix sequence,

$$\mathrm{rank}(H(\nu_l, \nu_r + 1)) \leq \sum_{j=1}^{\beta_r} d_j = \deg(\det(F)),$$

for any $\nu_l \geq 1$ and $\nu_r \geq 0$ (Turner, 2002, Lem. 4.6).

3. BECKERMANN-LABAHN FPHPS

Beckermann and Labahn (1992) introduce power Hermite-Padé approximants as a generalization of the classical scalar Hermite-Padé approximants to provide a uniform approach for different concepts of matrix-type Padé approximants, including vector approximants. Just as von zur Gathen and Gerhard (2003) formulate the scalar Wiedemann algorithm as solving a Padé approximation problem, we can transform our problem of computing a basis for the right generating vector polynomials of the matrix sequence (2.1) into a power Hermite-Padé approximant problem. We can then use the Fast Power Hermite-Padé Solver (FPHPS) algorithm Beckermann and Labahn (1994, §3) introduce to compute a basis for these approximants by computing all solutions along a "diagonal path" to compute a basis for the generating vector polynomials. (Van Barel and Bultheel (1991) also used this computational technique to solve the Hermite-Padé approximation problem.) We can then construct the minimal generating matrix polynomial of the sequence by computing the Popov form of the matrix formed by these basis vectors as its columns.

Consider the multi-index $\mathbf{n} = (n_1, \ldots, n_m)$. The polynomial tuple $\mathbf{P} = (P_1, \ldots, P_m) \in \mathbb{F}^{1\times m}[\lambda]$ is a vector Hermite-Padé approximant (VHPA) of type $(\mathbf{n}, \sigma)$ for the vector polynomials $G_1, \ldots, G_m \in \mathbb{F}^s[\lambda]$ if there exists a vector power-series $R \in \mathbb{F}^s[[\lambda]]$ such that $\sum_{i=0}^{m} P_i\, G_i = \lambda^\sigma R$, where $\deg(P_i) \leq n_i$ for $1 \leq i \leq m$ (Beckermann and Labahn, 1992, Def. 2.1). By setting

$$f_i = (1, \lambda, \lambda^2, \ldots, \lambda^{s-1}) \cdot G_i(\lambda^s), \qquad 1 \leq i \leq m,$$

we see computing the VHPAs of type $(\mathbf{n}, \sigma)$ and dimension s is equivalent to computing the power Hermite-Padé approximants (PHPA) of type $(\mathbf{n}, \sigma s, s)$ (Beckermann and Labahn, 1994, Ex. 2.5). In other words, there exists a power-series $\hat{R} \in \mathbb{F}[[\lambda]]$ such that

$$\mathbf{P}(\lambda^s) \cdot \mathbf{f}(\lambda) = \sum_{i=0}^{m} P_i(\lambda^s)\ f_i(\lambda) = \lambda^{\sigma s} \hat{R}(\lambda)$$

and $\deg(P_i) \leq n_i$ for $1 \leq i \leq m$. We shall call the power-series $\lambda^{\sigma s}\hat{R}$ the s-residual of the PHPA (Beckermann and Labahn, 1994, Def. 1.1). The defect of the PHPA polynomial tuple $\mathbf{P}$ of type $(\mathbf{n}, \sigma, s)$ is one more than the minimum difference between a member polynomial of $\mathbf{P}$ and the corresponding degree bound:

$$\mathrm{dct}(\mathbf{P}) = \min_{1\leq i\leq \beta_l+\beta_r} \{n_i - \deg(P_i) + 1\}.$$

The modular equivalence of equation (2.9) shows the reversals of a generating vector polynomial must solve the PHPA problem in much the same way the reversals of a generating polynomial are Padé approximations (von zur Gathen and Gerhard, 2003, Lem. 12.8). In particular, the vector polynomial C (2.4) generates the matrix sequence (2.1)

from the right if and only if there exists some polynomial $R \in \mathbb{F}^{\beta_l}[\lambda]$ such that

$$\begin{bmatrix} I & -(\sum_{i=0}^{\nu_l+\nu_r-1} B_i\lambda^i) \end{bmatrix} \begin{bmatrix} C^{(\mathrm{res})} \\ \mathrm{rev}_d(C) \end{bmatrix} = \lambda^{\nu_l+\nu_r} R$$

for some vector polynomial $C^{(\mathrm{res})}$ of degree at most $d-1$. In other words,

$$\mathbf{P} = \begin{bmatrix} C^{(\mathrm{res})} \\ \mathrm{rev}_d(C) \end{bmatrix} \in \mathbb{F}^{\beta_l+\beta_r}[\lambda] \tag{3.1}$$

solves the VHPA problem of type $(\mathbf{n}, \nu_l+\nu_r)$ with the indices

$$n_i = \begin{cases} \nu_r - 1 & \text{if } 1 \le i \le \beta_l \\ \nu_r & \text{if } \beta_r + 1 \le i \le \beta_l + \beta_r \end{cases} \tag{3.2}$$

for the vector polynomials $G_1, \ldots, G_{(\nu_l+\nu_r)\beta_l}$ where G_i is the ith column of the matrix polynomial

$$G = \begin{bmatrix} I & -(\sum_{i=0}^{\nu_l+\nu_r-1} B_i\lambda^i) \end{bmatrix} \in \mathbb{F}^{\beta_l \times (\beta_l+\beta_r)}[\lambda], \tag{3.3}$$

or equivalently $\mathbf{P}$ (3.1) solves the PHPA problem of type $(\mathbf{n}, (\nu_l+\nu_r)\beta_l, \beta_l)$ with indices (3.2) for the scalar polynomials

$$f_i = (1, \lambda, \lambda^2, \ldots, \lambda^{\beta_l-1}) \cdot G_i(\lambda^{\beta_l}) \tag{3.4}$$

where the vector polynomial $G_i(\lambda)$ is the ith column of the matrix polynomial (3.3).

Conversely, suppose $\mathbf{P}$ solves the PHPA problem of type $(\mathbf{n}, (\nu_l+\nu_r)\beta_l, \beta_l)$ with indices (3.2) for the scalar polynomial (3.4). Let $d = \nu_r + 1 - \mathrm{dct}(\mathbf{P})$, and let $C^{(\mathrm{res})}$ and $\hat{C}$ be the vector polynomials

$$C^{(\mathrm{res})} = \begin{bmatrix} P_1 & \ldots & P_{\beta_l} \end{bmatrix}^{\mathrm{T}} \in \mathbb{F}^{\beta_l}[\lambda] \tag{3.5}$$

and

$$\hat{C} = \begin{bmatrix} P_{\beta_l+1} & \ldots & P_{\beta_l+\beta_r} \end{bmatrix}^{\mathrm{T}} \in \mathbb{F}^{\beta_r}[\lambda]. \tag{3.6}$$

Then $\deg(C^{(\mathrm{res})}) \le d-1$ and $\deg(\hat{C}) \le d$, and $C^{(\mathrm{res})}$ and $\hat{C}$ satisfy the equivalence relation (2.9). In other words, the vector polynomial $C = \mathrm{rev}_d(\hat{C})$ generates the matrix sequence (2.1) from the right, and we can use the FPHPS algorithm to compute generating vector polynomials for the matrix sequence.

The FPHPS algorithm computes a basis, which we call a σ-basis, for the polynomial tuples $\mathbf{P}$ satisfying the PHPA problem of type $(\mathbf{n}, \sigma, s)$ by iterating through the powers λ^σ so that at the end of the kth step the algorithm has computed polynomial tuples $\mathbf{P}_{1,k}, \ldots, \mathbf{P}_{m,k}$ and their corresponding defects. After the kth step, we can write any solution $\mathbf{P}$ to the PHPA $\mathbf{f}(\lambda)\cdot\mathbf{P}(\lambda^s) \equiv 0 \pmod{\lambda^k}$ as a unique linear combination of the polynomial tuples $\mathbf{P}_{1,k}, \ldots, \mathbf{P}_{m,k}$:

$$\mathbf{P} = \sum_{i=0}^{m} g_i \mathbf{P}_{i,k}$$

where the polynomial $g_i \in \mathbb{F}[\lambda]$ has degree bounded by $\deg(g_i) < \mathrm{dct}(\mathbf{P})$ so that only the polynomial tuples with positive defect contribute to this basis for the solutions. After $k\beta_l$ steps of the FPHPS algorithm using $s = \beta_l$, $m = \beta_l + \beta_r$, the scalar polynomials (3.4), and indices (3.2), the polynomial tuples $\mathbf{P}_{1,k\beta_l}, \ldots, \mathbf{P}_{m,k\beta_l}$ form a basis over $\mathbb{F}[\lambda]$ for the solutions to the equation

$$\begin{bmatrix} I & -(\sum_{i=0}^{\nu_l+\nu_r-1} B_i\lambda^i) \end{bmatrix} \mathbf{P} \equiv 0^{\beta_l+\beta_r} \pmod{\lambda^k}.$$

At the end of $\sigma = (\gamma_l+\gamma_r)\beta_l$ steps, the FPHPS algorithm returns the polynomial tuples $\mathbf{P}_{1,\sigma}, \ldots, \mathbf{P}_{\beta_l+\beta_r,\sigma}$ and their defects.

If the algorithm always chooses the update index π to have the smallest possible value whenever it has a choice, the leading coefficients of the polynomial tuples $\mathbf{P}_{1,\sigma}, \ldots, \mathbf{P}_{\beta_l+\beta_r,\sigma}$ are always linearly independent over $\mathbb{F}$ for $\sigma \ge 0$. We can see this by induction. The leading coefficients of the polynomial tuples $\mathbf{P}_{1,0}, \ldots, \mathbf{P}_{\beta_l+\beta_r,0}$ are linearly independent over $\mathbb{F}$ by construction. Suppose the leading coefficients of $\mathbf{P}_{1,\sigma}, \ldots, \mathbf{P}_{\beta_l+\beta_r,\sigma}$ are linearly independent over $\mathbb{F}$. If the set of indices to update Λ_σ is empty, the leading coefficients of $\mathbf{P}_{1,\sigma+1}, \ldots, \mathbf{P}_{\beta_l+\beta_r,\sigma+1}$ are also linearly independent over $\mathbb{F}$ since $\mathrm{lc}(\mathbf{P}_{i,\sigma+1}) = \mathrm{lc}(\mathbf{P}_{i,\sigma})$ for $1 \le i \le \beta_l+\beta_r$. On the other hand, if Λ_σ is not empty, the $\mathrm{lc}(\mathbf{P}_{i,\sigma+1}) = \mathrm{lc}(\mathbf{P}_{i,\sigma})$ except when $l \in \Lambda_\sigma$ and $l \ne \pi = \pi_\sigma$. In that case, choosing π to have the smallest possible value whenever we have a choice ensures $\deg(\mathbf{P}_{l,\sigma}) \ge \deg(\mathbf{P}_{\pi,\sigma})$. If $\deg(\mathbf{P}_{l,\sigma}) > \deg(\mathbf{P}_{\pi,\sigma})$, then $\mathrm{lc}(\mathbf{P}_{l,\sigma+1}) = \mathrm{lc}(\mathbf{P}_{l,\sigma})$. If $\deg(\mathbf{P}_{l,\sigma}) = \deg(\mathbf{P}_{\pi,\sigma})$, then

$$\mathrm{lc}(\mathbf{P}_{l,\sigma+1}) = \mathrm{lc}(\mathbf{P}_{l,\sigma}) - \frac{c_{l,\sigma}}{c_{\pi,\sigma}} \mathrm{lc}(\mathbf{P}_{\pi,\sigma}) \ne 0^{\beta_l+\beta_r}$$

where $c_{l,\sigma}, c_{\pi,\sigma} \in \mathbb{F} \setminus \{0\}$. Therefore the update preserves the linear independence over $\mathbb{F}$ of the leading coefficients.

Exactly β_r of the polynomial tuples $\mathbf{P}_{1,\sigma}, \ldots, \mathbf{P}_{\beta_l+\beta_r,\sigma}$ returned by the FPHPS algorithm when $\sigma = (\nu_l+\nu_r)\beta_r$ have positive defect. We can see this if we let C_i be the ith column of the minimal generating matrix polynomial F of the matrix sequence (2.1). Then, if $\hat{C}_i = \mathrm{rev}_{d_i}(C_i)$ is the vector polynomial reversal of C_i with respect to degree $d_i = \deg(C_i)$, there exists a vector polynomial $C_i^{(\mathrm{res})}$ of degree at most $d_i - 1$ such that the polynomial tuple

$$\bar{\mathbf{P}}_i = \begin{bmatrix} C_i^{(\mathrm{res})} \\ \hat{C}_i \end{bmatrix} \in \mathbb{F}^{\beta_l+\beta_r}[\lambda]$$

solves the associated PHPA problem, $\bar{\mathbf{P}}_i \ne 0$, and $\mathrm{dct}(\bar{\mathbf{P}}_i) > 0$. Furthermore, because F is in Popov form, the leading coefficients of $C_1, \ldots, C_{\beta_r}$ are linearly independent over $\mathbb{F}$. Thus, $\hat{C}_1(0), \ldots, \hat{C}_{\beta_r}(0)$ and $\bar{\mathbf{P}}_1(0), \ldots, \bar{\mathbf{P}}_{\beta_r}(0)$ are also linearly independent over $\mathbb{F}$ and $\bar{\mathbf{P}}_1, \ldots, \bar{\mathbf{P}}_{\beta_r}$ are linearly independent over $\mathbb{F}[\lambda]$. This means at least β_r of the polynomial tuples output from the FPHPS algorithm must have positive defect.

Suppose another approximant $\mathbf{P}$ is linearly independent over $\mathbb{F}$ from $\bar{\mathbf{P}}_1, \ldots, \bar{\mathbf{P}}_{\beta_r}$, and let $C^{(\mathrm{res})}$ and $\hat{C}$ be the associated vector polynomials (3.5) and (3.6), respectively, and let $C = \mathrm{rev}_d(\hat{C})$ be the vector reversal of $\hat{C}$ with respect to degree $d = \nu_r + 1 - \mathrm{dct}(\mathbf{P})$. We know C must generate the matrix sequence (2.1) from the right, which means C must be a linear combination of $C_1, \ldots, C_{\beta_r}$ over $\mathbb{F}[\lambda]$: $C = \sum_{i=1}^{\beta_r} g_i C_i$ where $g_i \in \mathbb{F}[\lambda]$ for $1 \le i \le \beta_r$. Thus,

$$\hat{C} = \mathrm{rev}_d(C) = \sum_{i=1}^{\beta_r} \lambda^{d-d_i} g_i(1/\lambda)\ \hat{C}_i(\lambda)$$

Because the leading coefficients of $C_1, \ldots, C_{\beta_r}$ are linearly independent over $\mathbb{F}$,

$$\deg(C) = \max_{1 \le i \le \beta_r} \{\deg(g_i) + \deg(C_i)\} = \max_{1 \le i \le \beta_r} \{\deg(g_i) + d_i\}$$

and

$$\deg(g_i) + d_i \le \deg(C) \le d, \qquad 1 \le i \le \beta_r.$$

Thus, $\deg(g_i) \le d - d_i$ for every $1 \le i \le \beta_r$, and

$$\lambda^{d-d_i} g_i(1/\lambda) \in \mathbb{F}[\lambda], \qquad 1 \le i \le \beta_r.$$

Therefore $\hat{C}$ is a linear combination of $\hat{C}_1, \ldots, \hat{C}_{\beta_r}$ over $\mathbb{F}[\lambda]$. Then, let $\mathbf{P}_0$ be the polynomial tuple

$$\mathbf{P}_0 = \mathbf{P} - \sum_{i=1}^{\beta_r} \lambda^{d-d_i} g_i(1/\lambda)\ \bar{\mathbf{P}}_i(\lambda),$$

which solves the PHPA problem. Furthermore, let $C_0^{(\mathrm{res})}$ and $\hat{C}_0$ be the associated vector polynomials (3.5) and (3.6), respectively, so that

$$\hat{C}_0 = \hat{C} - \sum_{i=1}^{\beta_r} \lambda^{d-d_i} g_i(1/\lambda)\ \hat{C}_i(\lambda) = 0^{\beta_r}.$$

Because $C_0^{(\mathrm{res})}$ and $\hat{C}_0$ satisfy the equivalence relation (2.9),

$$C_0^{(\mathrm{res})} \equiv 0^{\beta_l} \pmod{\lambda^{\nu_l+\nu_r}}.$$

Furthermore, because $\deg(\mathbf{P}_0) \le d < \nu_l + \nu_r$, this means $C_0^{(\mathrm{res})} = 0^{\beta_l}$ and $\mathbf{P}_0 = 0^{\beta_l+\beta_r}$ which contradicts the assumption that $\mathbf{P}$ is linearly independent from $\bar{\mathbf{P}}_1, \ldots, \bar{\mathbf{P}}_{\beta_r}$ over $\mathbb{F}[\lambda]$. Therefore, there are exactly β_r solutions to the PHPA problem that are linearly independent over $\mathbb{F}[\lambda]$, and thus only β_r of the approximants $\mathbf{P}_{1,\sigma}, \ldots, \mathbf{P}_{\beta_l+\beta_r,\sigma}$ returned by the FPHPS algorithm have positive defect (Beckermann and Labahn, 1994, Cor. 4.2).

Let $\mathbf{P}_1, \ldots, \mathbf{P}_{\beta_r}$ be the polynomial tuples returned by the FPHPS algorithm with positive defect, and let $\hat{C}_1, \ldots, \hat{C}_{\beta_r}$ be their associated vector polynomials (3.6). If the vector polynomial $C_i = \mathrm{rev}_{d_i}(\hat{C}_i)$ is the vector reversal of $\hat{C}_i$ with respect to degree $d_i = \nu_r + 1 - \mathrm{dct}(\mathbf{P}_i)$, we know $C_1, \ldots, C_{\beta_r}$ generate the matrix sequence (2.1) from the right. Furthermore, because the leading coefficients of $\mathbf{P}_1, \ldots, \mathbf{P}_{\beta_r}$ are linearly independent over $\mathbb{F}$, we know $C_1, \ldots, C_{\beta_r}$ are linearly independent over $\mathbb{F}[\lambda]$, so we can find the minimal generating matrix polynomial F of the matrix sequence (2.1) by computing the Popov form of the matrix whose columns are $C_1, \ldots, C_{\beta_r}$:

$$C = \begin{bmatrix} C_1 & \cdots & C_{\beta_r} \end{bmatrix}. \tag{3.7}$$

4. BLOCK RANK ALGORITHM

The Kaltofen-Saunders rank algorithm computes the rank of a matrix A from the minimal polynomial $f^{\tilde{A}}$ of the preconditioned matrix $\tilde{A}$ in a Monte Carlo fashion. The algorithm relies on preconditioning the matrix A so the minimal polynomial $f^{\tilde{A}}$ is, with high probability, $f^{\tilde{A}} = \lambda\ f(\lambda)$ where $\mathrm{rank}(A) = \deg(f)$ and $f(0) \neq 0$. The algorithm then returns

$$\mathrm{rank}(A) = \deg(f^{\tilde{A}}) - 1 = \deg(f^{\tilde{A}}) - \mathrm{codeg}(f^{\tilde{A}})$$

where the co-degree, $\mathrm{codeg}(f)$, of a polynomial f is the degree of the smallest term with nonzero coefficient.

To convert this algorithm to a block version, we must first know how the invariant factors of the minimal generating matrix polynomial $F_X^{A,Y}$ of the block Wiedemann sequence relate to the invariant factors of the characteristic matrix $\lambda I - A$. If $\beta_l \ge \beta_r$, the ith largest invariant factor of $F_X^{A,Y}$ divides the ith largest invariant factor of $\lambda I - A$, and in fact they may be equal (Kaltofen and Villard, 2001, Thm. 1). We can find the probability they are equal for randomly selected block projections X and Y by examining the determinantal degree of $F_X^{A,Y}$ and the rank of the block Hankel matrix $H_X^{A,Y}(\nu_l, \nu_r + 1)$. In Section 2, we bounded the determinantal degree of $F_X^{A,Y}$ between the rank of the block Hankel matrix $H_X^{A,Y}(\nu_l, \nu_r + 1)$ and ν (2.6). When these bounds are equal, we force the ith largest invariant factor of $F_X^{A,Y}$ to equal the ith largest invariant factor of $\lambda I - A$.

LEMMA 1. *Let $\mathbb{F}$ be a field, S be a finite subset of $\mathbb{F}$, $A \in \mathbb{F}^{n\times n}$, and $1 \le \beta_r \le \beta_l \le n$. Let ν be the sum of the degrees of the β_r largest invariant factors of the characteristic polynomial $\lambda I - A$ (2.6). If $X \in S^{n\times\beta_l}$ and $Y \in S^{n\times\beta_r}$ are matrices whose entries are chosen uniformly and independently from S, then $\gamma_l \le \lceil \nu/\beta_l \rceil \le \lceil n/\beta_l \rceil$, $\gamma_r \le \lceil \nu/\beta_r \rceil \le \lceil n/\beta_r \rceil$, and the ith largest invariant factor of the minimal generating matrix polynomial $F_X^{A,Y}$ of the block Wiedemann sequence (2.3) equals the ith largest invariant factor of the characteristic matrix $\lambda I - A$,*

$$s_{\beta_r - i}(F_X^{A,Y}) = s_{n-i}(\lambda I - A), \qquad 0 \le i \le \beta_r - 1,$$

with probability at least $1 - 2\nu/|S| \ge 1 - 2n/|S|$.

PROOF. The proof follows from Villard (1997a, Cor. 1) and the Schwartz-Zippel Lemma (Schwartz, 1980; Zippel, 1979, 1990).

Let $\mathcal{X}$ and $\mathcal{Y}$ be matrices whose entries consist of indeterminates $\zeta_{i,j}$ and $\xi_{i,k}$, respectively, over $\mathbb{F}$ where $1 \le i \le n$, $1 \le \beta_l$, and $1 \le k \le \beta_r$, and let $\nu_l = \lceil \nu/\beta_l \rceil$ and $\nu_r = \lceil \nu/\beta_r \rceil$. The symbolic block Hankel matrix $H_{\mathcal{X}}^{A,\mathcal{Y}}(\nu_l, \nu_r + 1)$ has rank ν (Villard, 1997a,b). This means $H_{\mathcal{X}}^{A,\mathcal{Y}}(\nu_l, \nu_r + 1)$ has a nonzero $\nu \times \nu$ minor that is a polynomial of degree at most 2ν in the indeterminates. Let us denote this minor $\det((H_{\mathcal{X}}^{A,\mathcal{Y}}(\nu_l, \nu_r + 1))_{[i_1,\ldots,i_\nu;i_1,\ldots,i_\nu]})$. If X and Y are the matrices resulting from choosing values for $\zeta_{i,j}$ and $\xi_{i,k}$ uniformly and independently from S, the corresponding minor $\det((H_X^{A,Y}(\nu_l, \nu_r + 1))_{[i_1,\ldots,i_\nu;i_1,\ldots,i_\nu]})$ is a nonzero element of $\mathbb{F}$ with probability at least $1 - 2\nu/|S|$ by the Schwartz-Zippel Lemma. Since $H_X^{A,Y}(\nu_l, \nu_r + 1)$ has rank no more than $\det(\deg(F_X^{A,Y}))$, which is in turn bounded above by ν, this means

$$\mathrm{rank}(H_X^{A,Y}(\nu_l, \nu_r + 1)) = \det(\deg(F_X^{A,Y})) = \nu$$

with the given probability. When this happens, the maximal rank of the block Hankel matrix gives the desired bounds on γ_l and γ_r, and the maximal determinantal degree of $F_X^{A,Y}$ means $s_{\beta_r - i}(F_X^{A,Y}) = s_{n-i}(\lambda I - A)$. □

The probability of equality of the invariant factors given by Lemma 1 along with any preconditioner for the Kaltofen-Saunders rank algorithm gives a block Monte Carlo method to compute the rank of a singular matrix.

THEOREM 2. *Let $\mathbb{F}$ be a field, S be a finite subset of $\mathbb{F}$, $A \in \mathbb{F}^{n\times n}$, and $1 \le \beta_r \le \beta_l \le n$. Let $X \in S^{n\times\beta_l}$ and $Y \in S^{n\times\beta_r}$ be matrices whose entries are chosen uniformly and independently from S, and let $D = \mathrm{diag}(d_1, \ldots, d_n)$ where $d_1, \ldots, d_n$ are chosen uniformly and independently from S. Then, the difference $\deg(\det(F_X^{AD,Y})) - \mathrm{codeg}(\det(F_X^{AD,Y}))$ equals the rank of A, and we can compute $F_X^{AD,Y}$ from the first $\lceil n/\beta_l \rceil + \lceil n/\beta_r \rceil$ matrices in the block Wiedemann sequence $\{X^T(AD)^iY\}_{i=0}^{\infty}$ with probability at least $1 - n(n+3)/(2|S|)$.*

PROOF. Because the preconditioned matrix AD has minimal and characteristic polynomials

$$f^{AD} = \lambda\ f \text{ and } \det(\lambda I - AD) = \lambda^{n-r}\ f,$$

respectively, where f is squarefree and not divisible by λ with probability at least $1 - r(r+1)/(2|S|)$ (Turner, 2003, Thm. 3.2), the product of the β_r largest invariant factors of $\lambda I - AD$ is

$$\prod_{i=0}^{\beta_r - 1} s_{n-i}(\lambda I - AD) = \lambda^k\ f$$

where $1 \le k \le n - r$, f is squarefree, is not divisible by λ, and has degree r with probability at least $1 - r(r+1)/(2|S|)$. At the same time,

$$\det(F_X^{AD,Y}) = \prod_{i=0}^{\beta_r - 1} s_{\beta_r - i}(F_X^{AD,Y}) = \prod_{i=0}^{\beta_r - 1} s_{n-i}(\lambda I - AD)$$

and we can compute $F_X^{AD,Y}$ from the first $\lceil n/\beta_l \rceil + \lceil n/\beta_r \rceil$ matrices in the block Wiedemann sequence with probability at least $1 - 2n/|S|$ by Lemma 1. Thus, A has rank

$$r = \deg(f) = \deg(\det(F_X^{AD,Y})) - \operatorname{codeg}(\det(F_X^{AD,Y}))$$

with the required probability. □

A similar argument holds for DA and $F_X^{DA,Y}$.

Theorem 2 gives a block Monte Carlo method to compute the rank of a singular matrix. If A is nonsingular, let S be a finite subset of $\mathbb{F} \setminus \{0\}$ so $\det(AD) \neq 0$ and AD has equal minimal and characteristic polynomials, $f^{AD} = \det(\lambda I - AD)$, with probability at least $1 - n(n-1)/(2|S|)$ (Chen *et al.*, 2002, Thm. 4.2). Then $\operatorname{codeg}(f^{AD}) = 0$ and A has rank

$$r = \deg(f^{AD}) = \deg(f^{AD}) - \operatorname{codeg}(f^{AD})$$

with the same probability, and rank

$$r = \deg(F_X^{AD,Y}) - \operatorname{codeg}(F_X^{AD,Y})$$

with probability at least $1 - n(n+3)/(2|S|)$ by Lemma 1. In other words, by excluding 0 from S, the method provides the correct rank when A is nonsingular.

We now have a complete block Monte Carlo algorithm to compute the rank of any matrix A. (See Algorithm 1.) First, precondition the matrix A to have a nonzero $r \times r$ principal minor, for example by pre- and post-multiplying by butterfly network preconditioners (Chen *et al.*, 2002). Then, construct the minimal generating polynomial $F_X^{\tilde{A},Y}$ and compute the rank of A from its determinantal degree and co-degree.

This algorithm can incorporate any method to compute the minimal generating matrix polynomial $F_X^{\tilde{A},Y}$ from the matrices $\{X^T \tilde{A}^i Y\}_{i=0}^{\nu_l + \nu_r - 1}$. In particular, one may use a σ-basis computation such as Section 3 presents. Giorgi *et al.* (2003) prove one can compute a σ-basis for the matrix polynomial (3.3) in $O^{\sim}(\beta_l{}^{\omega} \nu_r)$ field operations and then convert the resulting matrix to Popov form in an additional $O^{\sim}(\beta_l{}^{\omega} \nu_r)$ field operations. We can then compute the determinant $\det(F_X^{\tilde{A},Y})$ in an additional $O^{\sim}(\beta_l{}^{\omega} \nu_r)$ field operations (Giorgi *et al.*, 2003). Here the "soft O" notation $O^{\sim}$ indicates some missing $\log(\beta_l \nu_r)$ factors and ω is the exponent of matrix multiplication over the field $\mathbb{F}$. Eberly *et al.* (2006) describe the development of σ-basis code in the LinBox library (Dumas *et al.*, 2002).

Algorithm 1 Block Rank Algorithm

Require: $A \in \mathbb{F}^{n \times n}$, S a finite subset of $\mathbb{F} \setminus \{0\}$, and $1 \le \beta_r \le \beta_l \le n$

Ensure: $r = \operatorname{rank}(A)$ with probability at least $1 - n(n+3)/(2|S|)$

1: $B_1, B_2 \leftarrow$ butterfly network preconditioners with parameters chosen uniformly and independently from S
2: $D \leftarrow \operatorname{diag}(d_1, \ldots, d_n)$, $d_1, \ldots, d_n$ chosen uniformly and independently from S
3: $\tilde{A} \leftarrow B_1^T A B_2 D$ {Implement via black box model}
4: Choose $X \in S^{n \times \beta_l}$ and $Y \in S^{n \times \beta_r}$ uniformly and independently
5: $\nu_l \leftarrow \lceil n/\beta_l \rceil$ and $\nu_r \leftarrow \lceil n/\beta_r \rceil$
6: Compute $F_X^{\tilde{A},Y}$ from $\{X^T \tilde{A}^i Y\}_{i=0}^{\nu_l + \nu_r - 1}$ {Possibly returning failure}
7: $r \leftarrow \deg(\det(F_X^{\tilde{A},Y})) - \operatorname{codeg}(\det(F_X^{\tilde{A},Y}))$

Avoiding the determinant computation could increase the speed of the algorithm. Indeed, because the minimal generating matrix polynomial is in Popov form, its determinantal degree $\deg(\det(F_X^{\tilde{A},Y})$ is the sum of the degrees of the columns of its columns. However, its determinantal codegree $\operatorname{codeg}(\det(F_X^{\tilde{A},Y}))$ is not quite as evident. If the projections are good, we may be able to recover the codegree from the rank of the constant matrix $F_X^{\tilde{A},Y}(0)$, which would allow us to avoid the determinant computation completely. Another possible increase in speed lies in avoiding the Popov form computation and using the matrix C (3.7) created from the σ-basis directly to compute the rank. These possibilities to increase the speed of the algorithm warrant further investigation.

This algorithm is similar to the one presented by Kaltofen and Saunders (1991, Thm. 3). In this block version, if the butterfly network preconditioners B_1 and B_2 are constructed using the generic exchange matrix of Chen *et al.* (2002, §6.2), then they each use at most $n\lceil \log_2(n) \rceil/2$ random elements from S (Chen *et al.*, 2002, Thm. 6.2) and are PRECONDIND preconditioners with probability at least $1 - r\lceil \log_2(n) \rceil/|S|$ (Chen *et al.*, 2002, Thm. 6.3). Thus, the leading $r \times r$ principal minor of $B_1^T A B_2$ is nonzero with probability at least

$$\left(1 - \frac{r\lceil \log_2(n) \rceil}{|S|}\right)^2 \ge 1 - \frac{2r\lceil \log_2(n) \rceil}{|S|} \ge 1 - \frac{2n\lceil \log_2(n) \rceil}{|S|}$$

(Chen *et al.*, 2002, Thm. 3.1). Thus, the complete algorithm uses a total of at most

$$2\frac{n\lceil \log_2(n) \rceil}{2} + n + \beta_l n + \beta_r n = n\left(\beta_l + \beta_r + 1 + \lceil \log_2(n) \rceil\right)$$

random elements from S and returns the correct rank with probability at least

$$\left(1 - \frac{2r\lceil \log_2(n) \rceil}{|S|}\right)\left(1 - \frac{n(n+3)}{2|S|}\right) \ge 1 - \frac{n(n+3+4\lceil \log_2(n) \rceil)}{2|S|}.$$

For comparison, the probability that the minimal polynomial $f_u^{\tilde{A},v}$ of the Wiedemann sequence $\{u^T \tilde{A}^i y\}_{i=0}^{\infty}$ is equal to the minimal polynomial $f^{\tilde{A}}$ of the matrix $\tilde{A}$ is at least

$1 - 2\deg(f^{\tilde{A}})/|S|$ (Kaltofen and Pan, 1991, Lem. 2), which is the probability given by Lemma 1 with blocking factors $\beta_l = \beta_r = 1$. Thus, the Kaltofen-Saunders rank algorithm with the same preconditioner uses no more than

$$2\frac{n\lceil\log_2(n)\rceil}{2} + 3n = n\left(3 + \lceil\log_2(n)\rceil\right)$$

random elements from S and returns the correct rank with probability at least

$$1 - \frac{4\deg(f^{\tilde{A}}) + r(r+1) + 4r\lceil\log_2(n)\rceil)}{2|S|} \geq 1 - \frac{n(n+3+4\lceil\log_2(n)\rceil)}{2|S|}.$$

The block rank algorithm with a diagonal preconditioner and blocking factors $\beta_l = \beta_r = 1$ is the original Kaltofen-Saunders rank algorithm with the same preconditioner. Increasing the blocking factors causes an increase in the number of random field elements required and a decrease in the algorithm's probability of success.

On the other hand, the block algorithm has two advantages over the non-blocked form. It is a parallel algorithm (Coppersmith, 1994; Kaltofen, 1995; Villard, 2000), and it captures more than just the largest invariant factor of the characteristic matrix $\lambda I - A$. Our preconditioner does not take advantage of this, but other preconditioners may exist that do.

In addition, unlike Eberly's block Lanczos rank algorithm (Eberly, 2004), this algorithm allows one to use different blocking factors (*i.e.*, $\beta_r < \beta_l$). As observed in Kaltofen (1995), this can reduce the number of black box operations that various computations require. Future work should explore in more detail the advantages and disadvantages of this new algorithm.

Although a certificate exists for the rank of a matrix over a field of characteristic zero (Saunders *et al.*, 2004), no such certificate is known over an arbitrary field. However, this Monte Carlo method will always return a value no greater than the rank of the matrix.

THEOREM 3. *Let $\mathbb{F}$ be a field, $A \in \mathbb{F}^{n\times n}$ have rank r, and $1 \leq \beta_r \leq \beta_l \leq n$. Let $X \in \mathbb{F}^{n\times\beta_l}$, $Y \in \mathbb{F}^{n\times\beta_r}$, and $D = \operatorname{diag}(d_1, \ldots, d_n) \in \mathbb{F}^{n\times n}$. Then, the rank of A is bounded from below by*

$$\begin{aligned} r &\geq \deg(\det(\lambda I - AD)) - \operatorname{codeg}(\det(\lambda I - AD)) \\ &\geq \deg(\det(F_X^{AD,Y})) - \operatorname{codeg}(\det(F_X^{AD,Y})). \end{aligned}$$

PROOF. Let f_1 and f_2 be polynomials in $\mathbb{F}[\lambda]$ such that λ does not divide either and

$$\det(F_X^{AD,Y}) = \lambda^{k_1} f_1 \text{ and } \det(\lambda I - AD) = \lambda^{k_2} f_2.$$

This means

$$\deg(f_1) = \deg(\det(F_X^{AD,Y})) - \operatorname{codeg}(\det(F_X^{AD,Y}))$$

and

$$\deg(f_2) = \deg(\det(\lambda I - AD)) - \operatorname{codeg}(\det(\lambda I - AD)).$$

We know λ^{n-r} divides $\det(\lambda I - AD)$ (Turner, 2003, Thm. 3.2), which means both $\operatorname{codeg}(\det(\lambda I - AD)) \geq n - r$ and $\deg(f_2) \leq r$. The ith largest invariant factor of $F_X^{AD,Y}$ divides the ith largest invariant factor of $\lambda I - AD$ (Kaltofen and Villard, 2001, Thm. 1), so $\det(F_X^{AD,Y})$ divides $\det(\lambda I - AD)$, f_1 divides f_2, and $\deg(f_1) \leq \deg(f_2) \leq r$. □

5. ACKNOWLEDGMENTS

The author gratefully acknowledges the assistance, guidance, and support of Erich Kaltofen, under whose direction he wrote his Ph.D. dissertation (Turner, 2002), and from which much of this paper has been abstracted. He also thanks the reviewers for their many helpful comments and suggestions that have helped shape the final version of this paper. In addition, financial support of both the National Science Foundation and Wabash College, especially the college's Byron K. Trippett fund that financially supports new assistant professors in all disciplines.

References

BERNHARD BECKERMANN AND GEORGE LABAHN (1992). A Uniform Approach for Hermite Padé and Simultaneous Padé Approximants and Their Matrix-Type Generalizations. *Numerical Algorithms*, 3:45–54.

BERNHARD BECKERMANN AND GEORGE LABAHN (1994). A Uniform Approach for the Fast Computation of Matrix-Type Padé Approximants. *SIAM Journal on Matrix Analysis and Applications*, 15(3):804–823.

LI CHEN, WAYNE EBERLY, ERICH KALTOFEN, B. DAVID SAUNDERS, WILLIAM J. TURNER, AND GILLES VILLARD (2002). Efficient Matrix Preconditioners for Black Box Linear Algebra. *Linear Algebra and its Applications*, 343-344:119–146. Special issue on *Infinite Systems of Linear Equations Finitely Specified*, edited by P. Dewilde, V. Olshevsky and A. H. Sayed.

DON COPPERSMITH (1994). Solving Homogeneous Linear Equations Over GF(2) via Block Wiedemann Algorithm. *Mathematics of Compututation*, 62:333–350.

J.-G. DUMAS, T. GAUTIER, M. GIESBRECHT, P. GIORGI, B. HOVINEN, E. KALTOFEN, B. D. SAUNDERS, W. J. TURNER, AND G. VILLARD (2002). LinBox: A Generic Library for Exact Linear Algebra. In ARJEH M. COHEN, XIAO-SHAN GAO, AND NOBUKI TAKAYAMA, editors, *Proceedings of the 2006 International Congress of Mathematical Software*. World Scientific.

WAYNE EBERLY (2004). Reliable Krylov-Based Algorithms for Matrix Null Space and Rank (Extended Abstract). In JAIME GUTIERREZ, editor, *ISSAC 2004: Proceedings of the 2004 International Symposium on Symbolic and Algebraic Computation*, pages 127–134. ACM Press.

WAYNE EBERLY, MARK GIESBRECHT, PASCAL GIORGI, ARNE STORJOHANN, AND GILLES VILLARD (2006). Solving Sparse Rational Linear Systems. In JEAN-GUILLAUME DUMAS, editor, *ISSAC 2006: Proceedings of the 2006 International Symposium on Symbolic and Algebraic Computation*. ACM Press.

JOACHIM VON ZUR GATHEN AND JÜRGEN GERHARD (2003). *Modern Computer Algebra*. Cambridge University Press, Cambridge, 2nd edition.

PASCAL GIORGI, CLAUDE-PIERRE JEANNEROD, AND GILLES VILLARD (2003). On the Complexity of Plynomial Matrix Computations. In J. RAFAEL SENDRA, editor, *ISSAC 2003: Proceedings of the 2003 International Symposium on Symbolic and Algebraic Computation*, pages 135–142. ACM Press.

Erich Kaltofen (1995). Analysis of Coppersmith's Block Wiedemann Algorithm for the Parallel Solution of Sparse Linear Systems. *Mathematics of Compututation*, 64(210):777–806.

Erich Kaltofen and Victor Pan (1991). Processor Efficient Parallel Solution of Linear Systems over an Abstract Field. In *Proceedings of SPAA '91 3rd Annual ACM Symposium on Parallel Algorithms and Architectures*, pages 180–191. ACM Press.

Erich Kaltofen and B. David Saunders (1991). On Wiedemann's Method of Solving Sparse Linear Systems. In H. F. Mattson, T. Mora, and T. R. N. Rao, editors, *AAECC-9: Proceedings of the 1991 Applied Algebra, Algebraic Algorithms and Error-Correcting Codes, International Conference*, volume 539 of *Lecture Notes in Computer Science*, pages 29–38. Springer Verlag.

Erich Kaltofen and Gilles Villard (2001). On the Complexity of Computing Determinants (Extended abstract). In Kiyoshi Shirayanagi and Kazuhiro Yokoyama, editors, *Proceedings of the Fifth Asian Symposium on Computer Mathematics (ASCM 2001)*, volume 9 of *Lecture Notes Series on Computing*, pages 13–27. World Scientific. Invited contribution.

Erich Kaltofen and Gilles Villard (2004). On the Complexity of Computing Determinants. *Computational Complexity*, 13(3–4):91–130.

V. M. Popov (1970). Some Properties of Control Systems with Irreducible Matrix Transfer Functions. In J. A. Yorke, editor, *Seminar on Differential Equations and Dynamical Systems, II*, volume 144 of *Lecture Notes in Computer Science*, pages 169–180. Springer Verlag.

B. David Saunders, Arne Storjohann, and Gilles Villard (2004). Matrix Rank Certification. *Elect. J. Linear Algebra*, 11:16–23.

J. T. Schwartz (1980). Fast Probabilistic Algorithms for Verification of Polynomial Identities. *Journal of the ACM*, 27:701–717.

William J. Turner (2002). *Black Box Linear Algebra with the LinBox Library*. Ph.D. thesis, North Carolina State University, Raleigh, NC USA.

William J. Turner (2003). Determinantal Divisors and Matrix Preconditioners. Submitted to Journal of Symbolic Computation.

Marc Van Barel and Adhemar Bultheel (1991). The Computation of Non-Perfect Padé-Hermite Approximants. *Numerical Algorithms*, 1:285–304.

Gilles Villard (1997a). Further Analysis of Coppersmith's Block Wiedemann Algorithm for the Solution of Sparse Linear Systems (Extended Abstract). In Wolfgang W. Küchlin, editor, *ISSAC'96: Proceedings of the 1996 International Symposium on Symbolic and Algebraic Computation*, pages 32–39. ACM Press.

Gilles Villard (1997b). A Study of Coppersmith's Block Wiedemann Algorithm using Matrix Polynomials. Rapport de Recherche 975 IM, Institut d'Informatique et de Mathématiques Appliquées de Grenoble.

Gilles Villard (2000). Processor Efficient Parallel Solution of Linear Systems of Equations. *Journal of Algorithms*, 35(1):122–126.

Douglas H. Wiedemann (1986). Solving Sparse Linear Equations Over Finite Fields. *IEEE Transactions on Information Theory*, IT-32(1):54–62.

Richard Zippel (1979). Probabilistic Algorithms for Sparse Polynomials. In Edward W. Ng, editor, *Symbolic and Algebraic Computation, EUROSAM '79, An International Symposiumon Symbolic and Algebraic Computation, Marseille, France, June 1979, Proceedings*, volume 72 of *Lecture Notes in Computer Science*, pages 216–226. Springer Verlag.

Richard Zippel (1990). Interpolating Polynomials from Their Values. *Journal of Symbolic Computation*, 9(3):375–403.