A fast algorithm for computing the Smith normal form with multipliers for a nonsingular integer matrix

Stavros Birmpilis

Cheriton School of Computer Science, University of Waterloo, Waterloo ON, Canada N2L 3G1

George Labahn

Cheriton School of Computer Science, University of Waterloo, Waterloo ON, Canada N2L 3G1

Arne Storjohann

Cheriton School of Computer Science, University of Waterloo, Waterloo ON, Canada N2L 3G1

Abstract

A Las Vegas randomized algorithm is given to compute the Smith multipliers for a nonsingular integer matrix A, that is, unimodular matrices U and V such that $AV = US$, with S the Smith normal form of A. The expected running time of the algorithm is about the same as required to multiply together two matrices of the same dimension and size of entries as A. Explicit bounds are given for the size of the entries in both unimodular multipliers. The main tool used by the algorithm is the Smith massager, a relaxed version of V, the unimodular matrix specifying the column operations of the Smith computation. From the perspective of efficiency, the main tools used are fast linear system solving and partial linearization of integer matrices. As an application of the Smith with multipliers algorithm, a fast algorithm is given to find the fractional part of the inverse of the input matrix.

Keywords: Smith normal form; Unimodular matrices; Integer matrices

1. Introduction

Let $A \in \mathbb{Z}^{n\times n}$ be a nonsingular integer matrix with

$$S := \operatorname{diag}(s_1, s_2, \ldots, s_n) = \begin{bmatrix} s_1 & & & \\ & s_2 & & \\ & & \ddots & \\ & & & s_n \end{bmatrix} \in \mathbb{Z}^{n\times n}$$

its Smith normal form. There are unimodular matrices $U, V \in \mathbb{Z}^{n\times n}$ which describe the set of invertible integer row and column operations which transform A into its Smith form S or vice

Email addresses: sbirmpil@uwaterloo.ca (Stavros Birmpilis), glabahn@uwaterloo.ca (George Labahn), astorjoh@uwaterloo.ca (Arne Storjohann)

versa. These row and column operations are typically defined as satisfying matrix equations in the form $UAV = S$ or $A = USV$. In our case, it will be convenient to specify these Smith form multipliers as unimodular matrices satisfying

$$AV = US. \tag{1}$$

Motivation. In some cases, just knowing the Smith form is all that is needed in applications. For example, to determine whether two integer matrices are equivalent up to unimodular row and column operations, it is enough to see if they have the same Smith form. Similarly, if A is the relation matrix for a finite abelian group G, then knowing its Smith form is enough to classify the group into a direct sum of cyclic groups (Cohen, 1996; Newman, 1972). Such a classification in turn is used, for example, to efficiently compute Gröbner bases of ideals invariant under the action of an abelian group (Faugère and Svartz, 2013).

However there are applications where both the Smith form and its unimodular multipliers are needed. Consider for example the linear system solving problem

$$xA = b, \tag{2}$$

that is, given a row vector $b \in \mathbb{Z}^{1\times n}$, find the unique row vector $x \in \mathbb{Q}^{1\times n}$ such that $xA = b$. Using the representation (1), we can transform the linear system in (2) to

$$\bar{x}S = \bar{b},$$

with $\bar{x} = xU$ and $\bar{b} = bV$. Since S is in Smith form, the new system allows for easier determination of possible properties of the solution. For example, the denominator of x, the smallest integer $d \in \mathbb{Z}_{>0}$ such that dx is integral, will be the same as the denominator of $\bar{x} = \bar{b}S^{-1}$.

The above example gives one application where both the Smith form and its unimodular multipliers are needed. Smith multipliers are also needed in a number of other settings. For example, when one not only wants the classification of a finite abelian group into the direct sum of its cyclic components, but also the isomorphism which takes the group to the direct sum of cyclic factors. If x is a row vector whose entries are generators of an abelian group and matrix A represents the relations among the entries of x such that $xA = 0$, then $\bar{x} = xU$ is a new set of generators with relations simply given by $\bar{x}S = 0$. Both the Smith form and its multipliers are needed when one looks for possible rational symmetry by a finite abelian group action for a set of polynomial equations along with determining the rational invariants and rewrite rules of such an action (Hubert and Labahn, 2016). Other applications which make use of the Smith multipliers include determining lattice rules for quadrature formulas over the unit cube (Lyness and Keast, 1995), its use in chip-firing for finite connected graphs in combinatorics (Stanley, 2016), and many more.

Computation. Initial algorithms for Smith form computation such as Smith (1861); Bradley (1970) were modelled on Gaussian elimination where greatest common divisors and the associated solutions of linear diophantine equations replaced division. These early algorithms encountered rapid growth of intermediate computations. However, efficient computation of the Smith form could make use of the fact that the diagonal elements are the invariant factors of the matrix, factors which can be represented as ratios of greatest common divisors of minors of the matrix. As the Smith form is unique one can for example use homomorphic imaging techniques (Geddes et al., 1992) for these computations. The first algorithm to compute the Smith form with

multipliers in polynomial time originated with Kannan and Bachem (1979). The multipliers are not unique with Storjohann (2000) being the first to consider the problem of small unimodular multipliers for Smith computation.

Let ω be a valid exponent of matrix multiplication: two $n \times n$ matrices can be multiplied in $O(n^\omega)$ operations from the domain of entries. Furthermore, let $\|A\|$ denote the largest entry of A in absolute value. Recent fast methods include that of Kaltofen and Villard (2005) which combines a Las Vegas algorithm for computing the characteristic polynomial with ideas of Giesbrecht (2001), to obtain a Monte Carlo algorithm for the Smith form in time $(n^{3.2} \log \|A\|)^{1+o(1)}$ assuming $\omega = 3$, and in time $(n^{2.695591} \log \|A\|)^{1+o(1)}$ assuming the currently best known upper bound $\omega < 2.37286$ for ω by Alman and Williams (2021) and the best known bound for rectangular matrix multiplication by Le Gall and Urrutia (2018).

Our main contribution. An important long-term program in exact linear algebra with polynomial or integer matrices is to obtain algorithms whose cost is about the same as for multiplying two matrices of corresponding dimension and entry sizes. In the case of Smith form this was solved in (Birmpilis et al., 2020) which gave a Las Vegas algorithm for the Smith form in time $(n^\omega \log \|A\|)^{1+o(1)}$. However it was not yet known how one can obtain both the Smith form and its multipliers in a similar complexity. A major difficulty is that the bitlength of the entries in U and V can be asymptotically larger than those in A. The previously fastest algorithm given in Storjohann (2000) recovers U and V in the form $UAV = S$ in time $(n^{\omega+1} \log \|A\|)^{1+o(1)}$.

The main contribution in this paper is a new Las Vegas algorithm which allows us to compute S, U and V satisfying (1) with approximately the same number of bit operations as required to multiply two matrices of the same dimension and size of entries as the input matrix. As we already have a fast way to compute the Smith form S, our goal in this paper is an efficient algorithm that also returns the unimodular matrices U and V. Previously, determining the Smith form alone had been considered easier than determining the Smith form and it's multipliers. In this paper, we show that finding the multipliers can be done in the same time as computing the Smith form, at least in terms of asymptotic complexity. However, finding the multipliers requires some new, novel ideas.

Our approach. The Las Vegas algorithm in Birmpilis et al. (2020) computes not only the Smith form S but also returns a *massager* matrix M. This matrix satisfies the property that

$$AM \equiv 0 \,\mathbf{c}\mathrm{mod}\, S \text{ and } WM \equiv I_n \,\mathbf{c}\mathrm{mod}\, S$$

for some integer matrix W. Here, **c**mod denotes working modulo columns: $B \equiv C \,\mathbf{c}\mathrm{mod}\, S$ if column j of B is congruent modulo s_j to column j of C, $1 \leq j \leq n$. On the one hand, a massager M is in general not unimodular and thus is a relaxed version of V in the equations

$$AV = US \text{ and } V^{-1}V = I_n,$$

where V^{-1} is integral since V is unimodular. On the other hand, a Smith multiplier V is precisely a massager that *is* unimodular. Massagers were introduced by Birmpilis et al. (2019) and are the main tool used in this paper to efficiently compute the Smith multipliers. Our approach is to perturb a massager M by a random matrix R scaled by the Smith form, that is, a matrix of the form $\bar{M} := M + RS$. We show that the perturbed matrix $\bar{M}$ remains a massager. Moreover, we prove that with high probability the perturbation has the effect that the submatrix comprised of

the last $n-1$ columns of $\bar{M}$ will be primitive, that is, $\bar{M}$ will be left equivalent to a nonsingular lower triangular matrix $\bar{H}$ that has the shape

$$\bar{H} = \begin{bmatrix} |\det \bar{M}| & & & & \\ * & 1 & & & \\ * & & 1 & & \\ \vdots & & & \ddots & \\ * & & & & 1 \end{bmatrix}, \tag{3}$$

with all $*$ entries nonnegative and reduced modulo $|\det \bar{M}|$. We remark that $\bar{H}$ is the unique lower triangular row Hermite form of $\bar{M}$. In case the perturbation is successful and $\bar{H}$ is trivial, that is, has the shape shown in (3) with all off-diagonal entries except for possibly the first equal to one, then we give an algorithm to compute it quickly (or determine that it is not trivial and report Fail). Since $\bar{H}$ is left equivalent to $\bar{M}$, the matrix $V := \bar{M}\bar{H}^{-1}$ will not only be integral but also unimodular. Based on the structure $\bar{H}$ we can show that V is also a massager. The matrix V is then one of our Smith multipliers. Exploiting again the fact that $\bar{H}$ is trivial, we show how to compute the product $\bar{M}\bar{H}^{-1}$ efficiently. The other multiplier U is constructed using (1).

Our approach allows us to establish explicit bounds on the size of the two unimodular multipliers. For example, if we define the bitlength of an integer column vector to be bitlength of the maximum magnitude entry, then we can show that the average bitlength of the columns of either unimodular multiplier matrix is bounded by $O(n(\log n + \log \|A\|))$. The overall size (the sum of the bitlengths of all of the entries) of either multiplier matrix is then bounded by $O(n^2(\log n + \log \|A\|))$.

Additional contributions. In order to obtain the desired running time for our algorithm we need to extend a some previously known algorithms to a slightly more general setting.

Our first additional contribution is to give extensions of subroutines for *linear system solving* and *integrality certification*. We briefly recall what these two problems are. Given an integer matrix B with the same number of rows as A, together with an integer lifting modulus $X \in \mathbb{Z}_{>0}$ that is relatively prime to $\det A$, the linear system solving problem is to compute $\mathrm{Rem}(A^{-1}B, X^d)$ for a given precision d. Here, $\mathrm{Rem}(a, X)$ for an integer a and positive integer X denotes the unique integer in the range $[0, X-1]$ that is congruent to a modulo X. If the first argument of Rem is a matrix or vector, the function applies element-wise. The integrality certification problem is to determine if $A^{-1}B$ is integral. Birmpilis et al. (2019) use the double-plus-one lifting approach of Pauderis and Storjohann (2012) to obtain a fast algorithm for the linear system solving problem. Birmpilis et al. (2020) follows this up with a fast algorithm for integrality certification. Both of the algorithms mentioned above were analyzed only in the special case when X is a power of 2, thus requiring the hypothesis that $\det A$ is an odd integer. In Section 3 we extend the linear system solving and integrality certification algorithms in (Birmpilis et al., 2019, 2020) to the case where X is the power of a small prime, thus allowing to handle the case of input matrices A with arbitrary determinant.

Our second additional contribution is to extend partial linearization techniques previously developed for polynomial matrices to the integer setting. The cost of algorithms on an integer matrix A are typically sensitive to $\log \|A\|$, the maximum bitlength of the entries. If only some entries have large bitlength, for example the average bitlength of the rows or columns is small, then for many problems partial linearization can be used to transform to a new problem on an input matrix that has maximum bitlength of entries the average bitlength of the rows or columns of

the original. Section 4 extends the partial linearization technique of Gupta et al. (2012, Section 6) for polynomial matrices to the integer setting, and gives applications to a number of problems. In particular this includes the linear system solving and integrality certification problems discussed above.

Our final contribution is to resolve an open question from Storjohann (2015), which asks if one can compute the proper fractional part of A^{-1} while avoiding any dependence on $\log \|A^{-1}\|$. Note that $\log \|A^{-1}\|$ is a measure of how much larger the bitlength of numerators in $A^{-1} \in \mathbb{Q}^{n\times n}$ are compared to their respective denominators. (If $\log \|A^{-1}\| < 0$ then all entries in A^{-1} are proper fractions, but it is possible that $\log \|A^{-1}\| \in \Omega(n(\log n + \log \|A\|))$, for example if A is unimodular.) Recall the notion of the proper fractional part of A^{-1}. Let $s \in \mathbb{Z}_{>0}$ be the largest entry in the Smith form of A. Then s is the minimal integer such that sA^{-1} is integral. The proper fractional part of A^{-1} is then $\mathrm{Rem}(sA^{-1}, s)/s$. To computing the proper fractional part of A it is thus sufficient to compute $\mathrm{Rem}(sA^{-1}, s)$.

Storjohann (2015) computes $\mathrm{Rem}(sA^{-1}, s)$ by first computing an *outer product adjoint formula* for A: a triple of matrices $(\bar{V}, S, \bar{U})$ such that

$$\mathrm{Rem}(sA^{-1}, s) = \mathrm{Rem}(\bar{V}(sS^{-1})\bar{U}, s).$$

There is a direct relationship between an outer product adjoint formula and the unimodular Smith multipliers U and V. Using this relationship, and as an application of our work, we show in Section 9 that an outer product formula can be computed in time $(n^\omega \log \|A\|)^{1+o(1)}$ bit operations. This improves on the algorithm of (Storjohann, 2015) by incorporating fast matrix multiplication and removing any dependence of the complexity on $\log \|A^{-1}\|$ in case $\|A^{-1}\| > 1$.

Organization of the paper. The remainder of this paper is organized as follows. Section 2 defines our main tool, the Smith massager of a nonsingular integer matrix, and gives several important properties. Section 3 gathers together a collection of computational tools related to linear system solving which we will require for our main algorithm. Section 4 presents a partial linearization technique which, in many algorithms, helps us replace the dependency of the cost estimates on the bit length of the largest entry of the input with the average bit length. Section 5 gives a high-level description of our main algorithm for computing Smith multipliers using an example. Section 6 proves the main probabilistic argument of our process, namely, the fact that a randomly perturbed Smith massager has an almost trivial Hermite form. Sections 7 and 8 present the main algorithm and rigorously prove the claimed time complexity along with bounds on the sizes of the multipliers. Section 9 shows how we can apply the Smith multiplier matrices in order to obtain an outer adjoint formula along with its complexity. The paper ends with a conclusion and topics for future research.

Cost model. Following (von zur Gathen and Gerhard, 2013, Section 8.3), cost estimates are given using a function $\mathsf{M}(d)$ that bounds the number of bit operations required to multiply two integers bounded in magnitude by 2^d. We use $\mathsf{B}(d)$ to bound the cost of integer gcd-related computations such as the extended euclidean algorithm. We can always take $\mathsf{B}(d) = O(\mathsf{M}(d)\log d)$. If $\mathsf{M}(d) \in \Omega(d^{1+\epsilon})$ for some $\epsilon > 0$ then $\mathsf{B}(d) \in O(\mathsf{M}(d))$.

As usual, we assume that M is superlinear and subquadratic. We also assume that $\mathsf{M}(ab) \in O(\mathsf{M}(a)\,\mathsf{M}(b))$ for $a, b \geq 1$. We assume that $\omega > 2$, and to simplify cost estimates we make the assumption that $\mathsf{M}(d) \in O(d^{\omega-1})$. This assumption simply stipulates that if fast matrix multiplication techniques are used, then fast integer multiplication techniques should also be used. The assumptions stated in this paragraph apply also to B.

2. Smith massagers

In this section we introduce our main tool, the *Smith massager* of a nonsingular integer matrix $A \in \mathbb{Z}^{n\times n}$. We provide the definition and basic features and identify some matrix operations that keep the massager properties intact. In Subsection 2.1, we show how the Smith massager gives an alternative, compact representation of the lattice $\{vA \mid v \in \mathbb{Z}^{n\times n}\}$, the set of all $\mathbb{Z}$-linear combinations of the rows of A. Finally, in Subsection 2.2, we present additional properties of massagers which will help us to compute Smith multipliers.

Definition 1. *Let $A \in \mathbb{Z}^{n\times n}$ be a nonsingular integer matrix with Smith form S. A matrix $M \in \mathbb{Z}^{n\times n}$ is a* Smith massager *for A if*

(i) it satisfies that

$$AM \equiv 0 \bmod S, \text{ and} \tag{4}$$

(ii) there exists a matrix $W \in \mathbb{Z}^{n\times n}$ such that

$$WM \equiv I_n \bmod S. \tag{5}$$

Property (i) of a Smith massager M implies that the matrix AMS^{-1} is integral, while property (ii) implies that M is unimodular up to modulo the columns of S. Thus, matrix M acts like the multiplier matrix V in $AV = US$ except that it relaxes the unimodularity property. Our objective will be to transform M to a new Smith massager that is in fact unimodular over the integers. Note that any Smith massager reduced column modulo S is still a Smith massager. If $M = (M \bmod S)$, then M is called a *reduced Smith massager*. We remark that a reduced massager can be be represented with only $O(n^2(\log n + \log \|A\|))$ bits.

Example 2. *The Smith form of*

$$A = \begin{bmatrix} -6 & 3 & -13 & -15 \\ -4 & 19 & 12 & -1 \\ -4 & 10 & -6 & 17 \\ -26 & -13 & 1 & -2 \end{bmatrix}$$

is $S = \operatorname{diag}(1, 1, 9, 29088)$. *For*

$$M = \begin{bmatrix} 0 & 0 & 7 & 805 \\ 0 & 0 & 5 & 23668 \\ 0 & 0 & 3 & 6 \\ 0 & 0 & 4 & 10224 \end{bmatrix},$$

we have $AM \equiv 0 \bmod S$, *while setting*

$$W = \begin{bmatrix} 4 & -19 & -12 & 1 \\ -306 & 3 & 133 & 0 \\ 5156 & 805 & 6332 & 0 \\ 12017 & -403 & 11356 & 0 \end{bmatrix}$$

gives

$$WM = I_4 + \begin{bmatrix} -1 & 0 & -99 & -436320 \\ 0 & -1 & -1728 & -174528 \\ 0 & 0 & 59112 & 23241312 \\ 0 & 0 & 116172 & 203616 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 9 & \\ & & & 29088 \end{bmatrix},$$

implying that $WM \equiv I_4 \bmod S$. *It follows that M is a Smith massager for A.*

It will be useful to notice that a Smith massager M for some matrix A remains a valid Smith massager under some specific columns operations.

Lemma 3. *Assume $M \in \mathbb{Z}^{n\times n}$ is a Smith massager for A. Then the matrix obtained from M by*

(i) adding any integer column vector multiplied by s_i to column i,

(ii) adding any multiple of a latter to a former column, or

(iii) multiplying (or dividing exactly) the i^{th} column by an integer relatively prime to s_i

is also a Smith massager for A.

Proof. For each one of these operations, we need to show that the modified matrix M still satisfies properties (i) and (ii) of Definition 1.

Let $\bar{M}$ be the matrix obtained from M by performing operation (i). Then $\bar{M} \equiv M \bmod S$ and thus $A\bar{M} \equiv 0 \bmod S$ and $W\bar{M} \equiv I_n \bmod S$ still hold.

For operation (ii), let $1 \le i_1 < i_2 \le n$ and $c \in \mathbb{Z}$. Let $\bar{M}$ be the matrix obtained from M by adding c times column i_2 to column i_1. Because $s_{i_1} \mid s_{i_2}$, $A\bar{M} \equiv 0 \bmod S$ still holds. Let $\bar{W}$ be the matrix obtained from W by adding $-c$ times row i_1 to row i_2. Then $\bar{W}\bar{M} \equiv I_n \bmod S$.

For operations (iii), let $c \in \mathbb{Z}$ be relatively prime to s_i. Let $\bar{M}$ be the matrix obtained from M by multiplying column i by c. Then $A\bar{M} \equiv 0 \bmod S$ still holds. Let $\bar{W}$ be the matrix obtained from M by multiplying row i by $\mathrm{Rem}(1/c, s_n) \in \mathbb{Z}$. Then $\bar{W}\bar{M} \equiv I_n \bmod S$. The case for $1/c$ is similar. □

2.1. Alternate characterizations of the lattice $\{vA \mid v \in \mathbb{Z}^{1\times n}\}$

Let $A \in \mathbb{Z}^{n\times n}$ be nonsingular. The set of all $\mathbb{Z}$-linear combinations of the rows of A generates the integer lattice $\{vA \mid v \in \mathbb{Z}^{1\times n}\}$. The following theorem gives alternate characterizations of the same lattice which will be useful in Section 7 to give an compact description of the Hermite form of A in terms of a Smith massager for A.

Theorem 4. *Let $A \in \mathbb{Z}^{n\times n}$ be nonsingular with Smith form S and Smith massager M. Let s be the largest invariant factor of S. The following lattices are all identical:*

- $L_1 = \{vA \mid v \in \mathbb{Z}^{1xn}\}$
- $L_2 = \{v \mid vA^{-1} \in \mathbb{Z}^{1\times n}\}$
- $L_3 = \{v \mid vMS^{-1} \in \mathbb{Z}^{1\times n}\}$
- $L_4 = \{v \mid vM(sS^{-1}) \equiv 0_{1\times n} \bmod s\}$
- $L_5 = \{v \mid vM \equiv 0_{1\times n} \bmod S\}$

Proof. It is straightforward to show that $L_1 = L_2$, $L_3 = L_4$ and $L_4 = L_5$ by verifying that each of these pairs of sets are subsets of each other. To complete the proof it will be sufficient to show that $L_2 = L_3$.

Let

$$B = \begin{bmatrix} A & \\ & I_n \end{bmatrix}\begin{bmatrix} I_n & \\ -W & I_n \end{bmatrix}\begin{bmatrix} I_n & M \\ & I_n \end{bmatrix}\begin{bmatrix} & I_n \\ S^{-1} & \end{bmatrix} = \begin{bmatrix} AMS^{-1} & A \\ (I_n - WM)S^{-1} & -W \end{bmatrix}.$$

By Definition 1 B is integral. Furthermore, since $|\det A| = \det S \neq 0$, B is unimodular. If we premultiply B by $\mathrm{diag}(A^{-1}, I_n)$ and then restrict to the first n rows, we obtain

$$\begin{bmatrix} A^{-1} & \end{bmatrix} B = \begin{bmatrix} MS^{-1} & I_n \end{bmatrix}. \tag{6}$$

Since both B and B^{-1} are integral, we conclude that for any $v \in \mathbb{Z}^{1\times n}$, vA^{-1} is integral if and only if vMS^{-1} is integral. It follows that $L_2 = L_3$. □

The following corollary follows from the equality of L_2 and L_3 in Theorem 4.

Corollary 5. *Let $A \in \mathbb{Z}^{n\times n}$ be nonsingular with Smith form S and Smith massager M. For any row vector $v \in \mathbb{Z}^{1\times n}$, the denominator of vA^{-1} equals the denominator of vMS^{-1}.*

As remarked earlier, if M is a reduced massager, then MS^{-1} can be represented with only $O(n^2(\log n + \log \|A\|))$ bits. This compares to $O(n^3(\log n + \log \|A\|))$ bits required for A^{-1}.

Example 6. *Matrix*

$$A = \begin{bmatrix} -6 & 3 & -13 & -15 \\ -4 & 19 & 12 & -1 \\ -4 & 10 & -6 & 17 \\ -26 & -13 & 1 & -2 \end{bmatrix},$$

from Example 2, has Smith form $S = \mathrm{diag}(1, 1, 9, 29088)$ and Smith massager

$$M = \begin{bmatrix} 0 & 0 & 7 & 805 \\ 0 & 0 & 5 & 23668 \\ 0 & 0 & 3 & 6 \\ 0 & 0 & 4 & 10224 \end{bmatrix}.$$

In this case,

$$A^{-1} = \frac{1}{29088} \begin{bmatrix} -271 & -402 & -373 & -937 \\ 580 & 920 & 524 & -356 \\ -1074 & 804 & -870 & 258 \\ -784 & -352 & 1008 & 80 \end{bmatrix},$$

and from Corollary 5, for any row vector $v \in \mathbb{Z}^{1\times n}$, the denominator of vA^{-1} equals the denominator of

$$v \begin{bmatrix} 7 & 805 \\ 5 & 23668 \\ 3 & 6 \\ 4 & 10224 \end{bmatrix} \begin{bmatrix} 1/9 & \\ & 1/29088 \end{bmatrix},$$

where the first two columns can be omitted because the corresponding invariant factors are 1. *Equivalently, from the equality of L_3 and L_5 in Theorem 4, we have that*

$$\begin{bmatrix} -271 & -402 & -373 & -937 \\ 580 & 920 & 524 & -356 \\ -1074 & 804 & -870 & 258 \\ -784 & -352 & 1008 & 80 \end{bmatrix} \equiv_R \begin{bmatrix} 7 & 805 \\ 5 & 23668 \\ 3 & 6 \\ 4 & 10224 \end{bmatrix} \begin{bmatrix} 3232 & \\ & 1 \end{bmatrix} \bmod 29088.$$

Recall that a basis for the lattice L_1 in Theorem 4 is any matrix that is left equivalent to A, for example A itself. The following theorem follows from the equality of L_1 and L_5 in Theorem 4.

Theorem 7. *Let $A \in \mathbb{Z}^{n\times n}$ be nonsingular with Smith form S and a Smith massager M. A matrix $H \in \mathbb{Z}^{n\times n}$ is left equivalent to A if and only if $|\det H| = \det S$ and $HM \equiv 0 \bmod S$.*

In other words, the Smith form S and a Smith massager M can be used to describe a left equivalent canonical form of a matrix A in a compact and fraction-free way. We will use Theorem 7 later in Section 7.

2.2. Creating a unimodular Smith massager

Let $A \in \mathbb{Z}^{n\times n}$ be nonsingular. In this subsection we give a high level overview of our algorithm to produce a Smith multiplier V such that $AV = US$. Recall that a Smith multiplier V is precisely a Smith massager that is unimodular. Once V has been found we recover U as $U := AVS^{-1}$. Our approach to computing a unimodular V has four steps:

1. Compute the Smith form S and a reduced Smith massager M for $2A$.
2. Choose a random perturbation matrix $R \in \mathbb{Z}^{n\times n}$ and let $\bar{M} := M + 2RS$.
3. Compute the lower triangular row Hermite form H of $\bar{M}$.
4. Return $V := \bar{M}H^{-1}$.

The reason, in step 1, for computing a Smith massager M for $2A$ instead of A is that matrix $\bar{M}$ produced in step 2 will be a nonsingular, independent of the choice of R. The purpose of the perturbation in step 2 is to ensure, with high probability, that $\bar{M}$ has a trivial lower triangular Hermite form, that is, with all but possibly the first diagonal entry equal to 1. Knowing *a priori* that $\bar{M}$ is nonsingular simplifies our deriviation of a lower bound on the probability the Hermite form H of $\bar{M}$ has at most one non-trivial column. Having H be trivial is important for the efficiency of steps 3 and 4, and also to obtain good bounds on the size of entries of V.

Filling in the details of how to choose R in step 2 and how to do each of the steps efficiently is the main topic of the rest of this article. Section 3 gathers together required subroutines related to linear system solving, and in particular shows that step 1 can be done efficiently. Section 4 develops a partial linearization technique which allows to efficiently compute with matrices with entries of skewed bitlength, for example the matrix $\bar{M}$ in step 2 which has columns of skewed bitlength. Section 5 then gives a worked example of the above four step algorithm and points to Sections 6–8 for algorithms to perform steps 3–4 efficiently.

In the remainder of this subsection, our goal is only to establish that the above recipe is correct, namely, that the matrix V returned in step 4 will be a unimodular Smith massager, independent of the choice of R in step 2. To do this, we need to establish that: (a) M in step 1 is a nonsingular Smith massager of A even though it is computed to be a Smith massager for $2A$; (b) $\bar{M}$ in step 2 remains a nonsingular Smith massager for A, despite the additive perturbation $+2RS$, and independent of choice of R; (c) the matrix V produced in step 4 is a Smith massager for A. On the on hand, the fact that V produced in step 4 is unimodular is straightforward: H is left equivalent to $\bar{M}$ and so $\bar{M}H^{-1}$ will be integral with determinant ± 1. On the other hand, what we need to prove in step 4 is that the column operations effected by the postmultiplication of H^{-1} in $V := \bar{M}H^{-1}$ always produces a V that is a Smith massager of A.

Proposition 8. *Let $c \in \mathbb{Z}_{>0}$ and $A \in \mathbb{Z}^{n\times n}$. If $M \in \mathbb{Z}^{n\times n}$ is a Smith massager for cA, then for any matrix $R \in \mathbb{Z}^{n\times n}$:*

(i) $M + R(cS)$ is a Smith massager for A.

(ii) The last i columns of $M + R(cS)$ have full rank over $\mathbb{Z}/(p)$ for any prime p that divides (cs_{n-i+1}).

An immediate corollary of Proposition 8 is that a Smith massager for $2A$ will be a nonsingular Smith massager of A. The proof of Proposition 8 follows directly from the next two lemmas and Definition 1.

Lemma 9. *Let $c \in \mathbb{Z}_{>0}$ and $A \in \mathbb{Z}^{n\times n}$. If $M \in \mathbb{Z}^{n\times n}$ is a Smith massager for cA, then M is also a Smith massager for A.*

Proof. First note that if $S \in \mathbb{Z}^{n\times n}$ is the Smith form of A, then cS is the Smith form of cA. Since M is a Smith massager for cA, Definition 1 states that

$$cAM \equiv 0 \,\mathbf{c}\mathrm{mod}\, cS, \tag{7}$$

and that there exists a $W \in \mathbb{Z}^{n\times n}$ such that

$$WM \equiv I_n \,\mathbf{c}\mathrm{mod}\, cS. \tag{8}$$

It follows from (7) that $AM \equiv 0 \,\mathbf{c}\mathrm{mod}\, S$ and from (8) that $WM \equiv I_n \,\mathbf{c}\mathrm{mod}\, S$, and thus by Definition 1, M is a Smith massager for A. □

Lemma 10. *For any prime p that divides s_{n-i+1}, the last i columns of a Smith massager M have full rank over $\mathbb{Z}/(p)$.*

Proof. The claim follows from Definition 1 of the Smith massager since

$$\begin{aligned} WM &\equiv I_n \,\mathbf{c}\mathrm{mod} \begin{bmatrix} s_1 & & \\ & \ddots & \\ & & s_n \end{bmatrix} \\ &\equiv I_n \,\mathbf{c}\mathrm{mod} \begin{bmatrix} s_1 & & & & & \\ & \ddots & & & & \\ & & s_{n-i} & & & \\ & & & p & & \\ & & & & \ddots & \\ & & & & & p \end{bmatrix}. \end{aligned}$$

If the last i columns of WM mod p have full rank, then the last i columns of M mod p also have full rank. □

Now consider steps 3 and 4 of the recipe. The *lower triangular row Hermite form* of a nonsingular matrix $A \in \mathbb{Z}^{n\times n}$ is the unique matrix

$$H := \begin{bmatrix} h_1 & & & \\ * & h_2 & & \\ \vdots & \vdots & \ddots & \\ * & * & \cdots & h_n \end{bmatrix} \in \mathbb{Z}^{n\times n}$$

that is left equivalent to A, has positive diagonal entries, and has off-diagonal entries in each column reduced by the diagonal entry in the same column. Lemma 12 provides the final ingredient to establish the correctness of our recipe by proving that a nonsingular Smith massager for A, post-multiplied by the inverse of its lower triangular row Hermite form, is still a Smith massager for A. Lemma 11 is an intermediate result.

Lemma 11. *Let $M \in \mathbb{Z}^{n\times n}$ be a nonsingular Smith massager and S the corresponding Smith form. If h_i is the i^{th} diagonal entry of the lower row Hermite form H of M, then $gcd(h_i, s_i) = 1$.*

Proof. The lemma follows from the fact that a matrix and its row Hermite form share the same column rank profile. Therefore, since, by Lemma 10, the last i columns of M have full rank over $\mathbb{Z}/(p)$ for any $p \mid s_{n-i+1}$, then the last i columns of H have full rank over $\mathbb{Z}/(p)$, and thus, $p \nmid h_{n-i+1}$. □

Lemma 12. *Let $M \in \mathbb{Z}^{n\times n}$ be a nonsingular Smith massager for a matrix A, and let $H \in \mathbb{Z}^{n\times n}$ be the lower triangular row Hermite form of M. Then, MH^{-1} is a unimodular Smith massager for A.*

Proof. Since H is unimodularly left equivalent to M, we have that matrix MH^{-1} is integral with $\det MH^{-1} = \pm 1$. It follows that MH^{-1} is unimodular. It remains to establish that MH^{-1} is a Smith massager for A. To this end, note that the inverse of any lower triangular matrix can be decomposed as the product of n pairs of matrices as follows.

$$H^{-1} = \prod_{i=0}^{n-1} \begin{bmatrix} I & & & \\ & 1 & & \\ & -h_{n-i+1,n-i} & 1 & \\ & \vdots & & \ddots \\ & -h_{n,n-i} & & 1 \end{bmatrix} \begin{bmatrix} I & & & \\ & 1/h_{n-i} & & \\ & & 1 & \\ & & & \ddots \\ & & & & 1 \end{bmatrix} \tag{9}$$

Thus multiplying M with H^{-1} can be represented as a series of n products, where each multiplication first applies an operation of the type as described in Lemma 3(ii), and second applies one of the type as in Lemma 3(iii) as certified by Lemma 11. Therefore, MH^{-1} is a Smith massager for A. □

3. Computational tools

An efficient algorithm for computing a Smith massager is given by Birmpilis et al. (2020). However, this relied on some subroutines for linear system solving that were restricted to input matrices A with $2 \perp \det A$. In this section, we give simple extensions of these subroutines, enabling us to extend the Smith massager algorithm of Birmpilis et al. (2020) to input matrices with arbitrary nonzero determinant.

The first subroutine we need is for nonsingular system solving. Given a nonsingular $A \in \mathbb{Z}^{n\times n}$ and matrix $B \in \mathbb{Z}^{n\times m}$, together with a lifting modulus $X \in \mathbb{Z}_{>0}$ that satisfies $X \perp \det A$ and $\log X \in O(\log n + \log \|A\|)$, the linear system solving problem is to compute $\mathrm{Rem}(A^{-1}B, X^d)$ for a given precision d. The second problem is integrality certification. Given an $s \in \mathbb{Z}_{>0}$ in addition to B, determine whether $sA^{-1}B$ is integral, and, if so, return the matrix $\mathrm{Rem}(sA^{-1}B, s)$. Provided the "dimension × precision ≤ invariant" compromises $m \times d \in O(n)$ and $m \times (\log \|B\| + \log s) \in O(n \log X)$ hold, our target complexity for solving these problems is

$$O(n^\omega \mathsf{M}(\log n + \log \|A\|) \log n) \tag{10}$$

bit operations. The algorithm supporting (Birmpilis et al., 2019, Corollary 7) solves the first problem in time (10) but was analyzed only when X is a power of 2. The algorithm for integrality certification by (Birmpilis et al., 2020, Section 2.2) has the same constraint since it relies on the

algorithm supporting (Birmpilis et al., 2019, Corollary 7). The analysis in (Birmpilis et al., 2019, Corollary 7) exploited the fact that radix conversions to go between the X-adic and binary representation of intermediate integers were not required since X was a power of 2. Here, we extend the the linear system solving algorithm of Birmpilis et al. (2019) by showing how to choose X to be the power of a small prime. Even though radix conversions are now required, we show how to maintain the cost (10) by keeping intermediate results in X-adic form and only doing radix conversions at the beginning and end of the process.

Subsection 3.1 shows how to choose X as the power of a small random prime. Subsection 3.2 recalls the double-plus-one lifting algorithm of Pauderis and Storjohann (2012) which forms the basis of the linear system solving and integrality certification algorithms. Subsections 3.3 and 3.4 extend the linear system solving and integrality certification algorithms, respectively, to work with an X as chosen in Subsection 3.1. Subsection 3.5 uses the results developed in the previous subsections to extend the Smith massager algorithm of (Birmpilis et al., 2020) to arbitrary nonsingular matrices.

3.1. Lifting initialization

Let C be an upper bound for $|\det A|$. von zur Gathen and Gerhard (2013, Theorem 18.10) show how to produce an integer p the range $6 \log C < p < 12 \log C$ that is both prime and satisfies $p \perp \det A$ with probability at least $1/2$. If p is prime, we can check if $p \perp \det A$ by trying to compute an LUP decomposition of A mod p over $\mathbb{Z}/(p)$. If $p \perp \det A$, then we can choose our lifting modulus X to be a power of p. In the following lemma, conditions (iii) and (iv) are included because they are preconditions of the double-plus-one lifting algorithm described in the next subsection.

Lemma 13. *There exists a Las Vegas algorithm that takes as input a nonsingular $A \in \mathbb{Z}^{n \times n}$, and returns as output an odd integer X that satisfies*

(i) *X is the power of a prime p with $\log p \in \Theta(\log n + \operatorname{loglog} \|A\|)$,*

(ii) *$X \perp \det A$,*

(iii) *$X \geq \max(10000, 3.61 n^2 \|A\|)$, and*

(iv) *$\log X \in O(\log n + \log \|A\|)$.*

The cost of the algorithm is $O(n^\omega \mathsf{M}(\log n + \log \|A\|))$ bit operations. The algorithm returns FAIL *with probability at most $1/2$.*

Proof. By Hadamard's bound we have $C := n^{n/2} \|A\|^n \geq |\det A|$. By von zur Gathen and Gerhard (2013, Theorem 18.10), producing an integer p in the range $6 \log C < p < 12 \log C$ that is both prime and does not divide $\det A$ with probability at least $1/2$ can be done within the allotted time. Proving that p is prime can be done within the allotted time using the algorithm of Agrawal et al. (2004). If it is determined that p is not prime, then report Fail. Working over $\mathbb{Z}/(p)$, we use $O(n^\omega \mathsf{M}(\log p) + n \mathsf{B}(\log p))$ bit operations to compute an LUP decomposition (Aho et al., 1974, §6.4) of $\operatorname{Rem}(A, p)$. The $n \mathsf{B}(\log p)$ term in this cost estimate is for inverting the n nonzero pivots arising during the elimination. Computing $\operatorname{Rem}(A, p)$ and then its LUP decomposition is within our target cost since $\log p \in O(\log n + \operatorname{loglog} \|A\|)$ and $\mathsf{B}(\log p) \in O(\mathsf{M}(\log p)(\operatorname{loglog} p))$. If, during the course of the LUP decomposition, it is determined that A is singular modulo p, then return Fail. Otherwise, let X be the smallest power of p which satisfies the third requirement of the lemma. Then, X also satisfies the fourth requirement. □

Corollary 14. *If X is a lifting modulus as in Lemma 13, then* $\mathrm{Rem}(A^{-1}, X)$ *can be computed in time* $O(n^\omega \,\mathsf{M}(\log n + \log \|A\|))$.

Proof. Let p and LUP be as in the proof of Lemma 13. Compute $\mathrm{Rem}(A^{-1}, p) = \mathrm{Rem}(P^T U^{-1} L^{-1}, p)$, and use $O(\mathrm{loglog}\, X)$ steps of algebraic Newton iteration (von zur Gathen and Gerhard, 2013, Algorithm 9.3) to lift $\mathrm{Rem}(A^{-1}, p)$ to $\mathrm{Rem}(A^{-1}, X)$. The running time is dominated by the last step of the lifting, which is within the claimed cost. □

3.2. Double-plus-one lifting

Let X be a lifting modulus as in Lemma 13. Given a $k \in \mathbb{Z}_{>0}$, the double-plus-one lifting of Pauderis and Storjohann (2012, Section 3) computes a straight line formula that is congruent modulo X^k to the X-adic expansion

$$A^{-1} \equiv * + *X + *X^2 + \cdots + *X^{k-1} \bmod X^k. \tag{11}$$

The straight line formula consists of only $O(\log k)$ matrices instead of k as in (11). More precisely, given a $k \in \mathbb{Z}_{>0}$ that is one less than a power of 2, double-plus-one lifting computes a residue $R \in \mathbb{Z}^{n\times n}$ such that

$$A^{-1} = D + A^{-1} R X^k, \tag{12}$$

where $D \in \mathbb{Z}^{n\times n}$ satisfies $\|D\| \le 0.6X^k$. Note that $D \equiv A^{-1} \bmod X^k$. Instead of computing D explicitly, double-plus-one lifting computes a formula

$$D = (\cdots((*(I + *X) + *X^2)(I + *X^3) + *X^6)(I + *X^7) + *X^{14})\cdots), \tag{13}$$

where each $*$ is an $n \times n$ integer matrix with $\|*\| < X$. The following result is (Pauderis and Storjohann, 2012, Corollary 6) except that we use Corollary 14 to compute $\mathrm{Rem}(A^{-1}, X)$ in the allotted time.

Lemma 15. (Pauderis and Storjohann, 2012, Corollary 6) *Assume we have a lifting modulus X as in Lemma 13. Let $k \in \mathbb{Z}_{>0}$ be one less than a power of two. If* $\log k \in O(\log n)$*, then a residue R as in (12) and a straight line formula for D as shown in (13) can be computed in time* $O(n^\omega \,\mathsf{M}(\log n + \log \|A\|) \log n)$.

3.3. System solving

Let X be a lifting modulus as in Lemma 13. Consider equations (12) and (13). If $k \ge d$, then given a $B \in \mathbb{Z}^{n\times m}$, we can compute $\mathrm{Rem}(A^{-1}B, X^d)$ by premultiplying B by the straight line formula for $D \equiv A^{-1} \bmod X^k$ on the right hand side of (13), keeping intermediate expressions reduced modulo X^d. Applying the formula requires doing the following operation $O(\log k)$ times: premultiplying an $n\times m$ matrix with entries reduced modulo X^d by an $n\times n$ matrix $*$ with $\|*\| < X$. When X is a power of 2, and $m \times d \in O(n)$, Birmpilis et al. (2019, Corollary 7) show that this can be done within our target cost (10).

When X is not a power of 2, we need to use radix conversion to go between the binary and X-adic representation of integers. To avoid unnecessary radix conversions, we can compute the X-adic expansion of B once at the beginning, and then keep intermediate results in X-adic form. The following result is a corollary of Storjohann (2005, Theorem 33).

Lemma 16. *Let $X \in \mathbb{Z}_{>0}$ satisfy $\log X \in O(\log n + \log \|A\|)$. Let $C \in \mathbb{Z}^{n\times n}$ with $\|C\| < X$ and $B \in \mathbb{Z}^{n\times m}$ with $B = \operatorname{Rem}(B, X^d)$. If $m \times d \in O(n)$, then $\operatorname{Rem}(CB, X^d)$ can be computed in time $O(n^\omega \mathsf{M}(\log n + \log \|A\|))$, assuming the input parameter B and output $\operatorname{Rem}(CB, X^d)$ are given as X-adic expansions.*

The following extends (Birmpilis et al., 2019, Corollary 7) using Lemmas 15 and 16.

Theorem 17. *Assume we have a lifting modulus X as in Lemma 13. If entries in $B \in \mathbb{Z}^{n\times m}$ are reduced modulo X^d and $m \times d \in O(n)$, then $\operatorname{Rem}(A^{-1}B, X^d)$ can be computed in time $O(n^\omega \mathsf{M}(\log n + \log \|A\|) \log n)$.*

Proof. Using the radix conversion of (von zur Gathen and Gerhard, 2013, Theorem 9.17), compute the X-adic expansion of B in time $O(nm\,\mathsf{M}(d \log X) \log d)$. Simplifying this cost estimate using $\mathsf{M}(d \log X) \in O(d^{\omega-1}\mathsf{M}(\log X))$ and $d \in O(n/m)$ shows that this is within the allotted time. Compute a straight line formula congruent to $A^{-1} \bmod x^d$ using Lemma 15. Applying the straight line formula to $B \bmod X^d$ to compute the X-adic expansion of $\operatorname{Rem}(A^{-1}B, X^d)$ now requires $O(\log n)$ applications of Lemma 16, plus some matrix additions which do not dominate the cost. Note that the multiplications with powers of X are free since we are working with X-adic expansions throughout. Finally, compute $\operatorname{Rem}(A^{-1}B, X^d)$ from its X-adic expansion using another radix conversion. □

3.4. Integrality certification

Any rational number can be written as an integer and a proper fraction. For example,

$$\frac{9622976468279041913}{21341} = 450914974381661 + \frac{14512}{21341},$$

where 450914974381661 is the quotient and 14512 is the remainder of the numerator with respect to the denominator. Similarly, a rational system solution $A^{-1}B$ can have entries with large numerators compared to denominators. In some situations only the information containing the proper fractional part of the system solutions is required. Given an $s \in \mathbb{Z}_{>0}$, integrality certification can be used to determine whether $sA^{-1}B$ is integral in a cost that depends on $\log \|A\| + \log s + \log \|B\|$ instead of $\log \|A^{-1}\| + \log s + \log \|B\|$. If $sA^{-1}B$ is integral, the version of integrality certification developed by Birmpilis et al. (2020, Section 2.2) also returns the proper fractional part $\operatorname{Rem}(sA^{-1}B, s)/s$ of $A^{-1}B$, but required that $2 \perp \det A$. Using the tools developed in the previous subsections the algorithm extends easily to handle the case of an A with arbitrary nonzero determinant. For completeness, we give the recipe here.

1. Using Lemma 15 compute a high-order residue $R \in \mathbb{Z}^{n\times n}$ such that $A^{-1} = D + A^{-1}R \times X^h$ for an $h \in \mathbb{Z}_{>0}$ such that $X^h > 2sn^{n/2}\|A\|^{n-1}\|B\|$.
2. Using Theorem 17, compute the system solution $\operatorname{Rem}(A^{-1}(sRB), X^\ell)$ for some $\ell \in \mathbb{Z}_{>0}$ such that $X^\ell > 2n\|A\|(0.6sn\|B\|)$.
3. Let C be the matrix that is congruent to $\operatorname{Rem}(A^{-1}(sRB), X^\ell)$ but with entries reduced in the symmetric range modulo X^ℓ.
   if $\|C\| < 0.6sn\|B\|$ **then**
      return $\operatorname{Rem}(C \times X^h, s)$
   else
      return NotIntegral

Theorem 18. *Assume we have a lifting modulus X as in Lemma 13. Let $s \in \mathbb{Z}_{>0}$ and $B \in \mathbb{Z}^{n\times m}$ be given. There exists an algorithm that determines whether $sA^{-1}B$ is integral, and, if so, returns $\mathrm{Rem}(sA^{-1}B, s)$. If $m \times (\log s + \log \|B\|) \in O(n \log X)$ and $m \in O(n)$, then the running time is $O(n^{\omega}\,\mathsf{M}(\log n + \log \|A\|)\log n)$.*

3.5. Computing a Smith massager for any A

Finally, we show how to generalize the Smith massager algorithm of Birmpilis et al. (2020) to arbitrary nonsingular input matrices by using the results developed in the previous subsections. We remark that the cost estimate of the following theorem uses B instead of M because the algorithm for computing a massager makes extensive use of gcd computations to compute intermediate Smith forms.

Theorem 19. *There exists a Las Vegas algorithm that takes as input a nonsingular $A \in \mathbb{Z}^{n\times n}$, and returns as output the Smith form $S \in \mathbb{Z}^{n\times n}$ of A together with a reduced Smith massager $M \in \mathbb{Z}^{n\times n}$. The cost of the algorithm is $O(n^{\omega}\,\mathsf{B}(\log n + \log \|A\|)(\log n)^2)$ bit operations. The algorithm returns* FAIL *with probability at most* $1/2$.

Proof. Birmpilis et al. (2020, Algorithm `SmithMassager`) returns a so-called *index*-$(0, n)$ *Smith massager*. This is a 4-tuple (U, M, T, S) of matrices from $\mathbb{Z}^{n\times n}$, such that T is unit upper triangular, S is the Smith form, and the matrix

$$B = \begin{bmatrix} A & AMS^{-1} \\ U & (UM + T)S^{-1} \end{bmatrix} \in \mathbb{Z}^{2n\times 2n} \tag{14}$$

is unimodular. From (14) and the fact that B is integral, we have that

$$AM \equiv 0 \bmod S \text{ and } UM + T \equiv 0 \bmod S. \tag{15}$$

The second equation in (15) is equivalent to

$$(-T^{-1}U)M \equiv I_n \bmod S, \tag{16}$$

implying that the matrix M is a Smith massager for A.

To apply (Birmpilis et al., 2020, Algorithm `SmithMassager`) in the case where A may not satisfy $2 \perp \det A$, we first use the Las Vegas algorithm of Lemma 13 (at most twice) to compute a lifting modulus X with probability at least $1/4$. Then we can directly use (Birmpilis et al., 2020, Algorithm `SmithMassager`) but with the following changes: in the proof of (Birmpilis et al., 2020, Theorem 12) we appeal to Theorem 18 instead of (Birmpilis et al., 2020, Theorem 2); in the proof of (Birmpilis et al., 2020, Theorem 21) we appeal to Theorem 17 instead of (Birmpilis et al., 2019, Corollary 7). By running this generalization of (Birmpilis et al., 2020, Algorithm `SmithMassager`) just described (at most twice) we can compute S and M with probability at least $1/4$. □

By running the Las Vegas algorithm of Theorem 19 at most three times, we obtain the following result, which will be useful in subsequent sections.

Corollary 20. *There exists a Las Vegas algorithm* `SmithMassager`(A) *with the input/output specification and the running time stated in Theorem 19. The algorithm returns* `FAIL` *with probability at most* $1/8$.

4. Partial linearization

The cost of algorithms that take as input an integer matrix $A \in \mathbb{Z}^{n\times m}$ are typically expressed in terms of the dimensions n and m, and $\log \|A\|$, which is proportional to the bitlength of the largest entry of A in absolute value. More precisely, let us define length(a) for an integer a to be the number of bits in its binary representation, that is,

$$\text{length}(a) := \begin{cases} 1 & \text{if } a = 0 \\ 1 + \lfloor \log_2 |a| \rfloor & \text{otherwise} \end{cases}.$$

By extension, for a matrix we define length(A) := length($\|A\|$), so length(A) is the length of the largest entry of A in absolute value.

But consider decomposing A into columns as

$$A = \left[\begin{array}{c|c|c} v_1 & \cdots & v_m \end{array}\right] \in \mathbb{Z}^{n\times m}.$$

For some inputs, the lengths of the columns v_i can be skewed, that is, the *average* column length

$$d = \left\lceil \sum_{i=1}^{m} \text{length}(v_i)/m \right\rceil$$

can be asymptomatically smaller than length(A) = $\max_i$ length(v_i). Even length(A) $\approx md$ is possible in the case of one column of large length. For such inputs, being able to replace the term length(A) with the average length d can give significantly improved cost estimates.

Example 21. *For the identity matrix I_m, we have* length(A) = 1 *and the average column length is also $d = 1$. Now let I'_m be equal to I_m but with the last column multiplied by $2^{m+1} - 1$. Then* length(I') = $m + 1$ *but the average column length is only $d = 2$.*

In this section, we adapt the partial linearization technique for polynomial matrices given by Gupta et al. (2012, Section 6) to the case of integer matrices. The main motivation is to extend the algorithms from Section 3 so that their cost estimates depend on the average length d and not length(A).

The technique transforms the input matrix A into a new matrix D which can be used in place of A for all of the algorithms presented in Section 3, and many more (see below and also the remarks at the end of Subsection 4.2). Matrix D will satisfy that length(D) $\leq d + 1$, at the cost of D having at most m more rows and columns than $A \in \mathbb{Z}^{n\times m}$.

More importantly, the constructed matrix D will "imitate" A in a way such that the output of the routines with D as input includes the original output in a direct way. Specifically, matrix D will satisfy the following two fundamental properties with respect to A:

(i) D can be obtained from diag(A, I) using unimodular row and column operations.

(ii) The principal $n \times n$ submatrix of the adjoint of D equals the adjoint of A (for square matrices).

Property (i) establishes that the rank, the determinant (for square matrices) and the Smith form of matrix A can be trivially deduced from the same objects for matrix D. In Subsection 4.3 we show that computing the Smith massager of a nonsingular A can also be directly reduced to computing the Smith massager of D.

Property (ii) provides us with a direct extension of system solving. If $A \in \mathbb{Z}^{n\times n}$ is nonsingular, then for any matrix $B \in \mathbb{Z}^{n\times *}$, we have that the first n rows of

$$D^{-1}\begin{bmatrix} B \\ 0 \end{bmatrix}$$

are equal to $A^{-1}B$. Finally, because $\det D = \det A$ and using property (ii), it follows that the principal $n\times n$ submatrix of the lower row Hermite form of D equals the lower row Hermite form of A.

Example 22. *Let*

$$A = \begin{bmatrix} 2 & 4 & 44199 & 3061969404 \\ 4 & 8 & 19644 & 765492351 \\ 7 & 8 & 44199 & 5358446457 \\ 7 & 5 & 9822 & 765492351 \end{bmatrix} \in \mathbb{Z}^{4\times 4},$$

a matrix with skewed column lengths. In this case $\text{length}(A) = 33$ *and average column length is* $d = 14$*. The partial linearization of A constructed later in this section will be*

$$D = \begin{bmatrix} 2 & 4 & 11431 & 12796 & 2 & 6663 & 11 \\ 4 & 8 & 3260 & 15487 & 1 & 13953 & 2 \\ 7 & 8 & 11431 & 10105 & 2 & 15757 & 19 \\ 7 & 5 & 9822 & 15487 & 0 & 13953 & 2 \\ & & -16384 & & 1 & & \\ & & & -16384 & & 1 & \\ & & & & & -16384 & 1 \end{bmatrix} \in \mathbb{Z}^{7\times 7}.$$

Notice that $\|D\| \leq 2^d = 16384$.

4.1. The partial linearization construction

Let $e \in \mathbb{Z}_{\geq 0}$ and $d \in \mathbb{Z}_{\geq 1}$ be given and assume for the moment that a column vector $v \in \mathbb{Z}_{\geq 0}^{n\times 1}$ contains only nonnegative entries. Then, we define $C_{e,d}(v)$ to be the unique $n \times e$ matrix over $\mathbb{Z}_{\geq 0}$ that satisfies

$$\text{Quo}(v, 2^d) = C_{e,d}(v)\begin{bmatrix} 1 \\ 2^d \\ \vdots \\ 2^{(e-1)d} \end{bmatrix}, \tag{17}$$

with all but possibly the last column (if $e > 0$) of magnitude strictly less than 2^d. If $e = 0$ then $C_{e,d}(v)$ is the $n \times 0$ matrix, while for $e \geq 1$,

$$v = \text{Rem}(v, 2^d) + \text{Col}(C_{e,d}(v), 1)2^d + \cdots + \text{Col}(C_{e,d}(v), e)2^{ed} \tag{18}$$

is the 2^d-adic series expansion of v, except that the coefficient $\text{Col}(C_{e,d}(v), e)$ of 2^{ed} may have magnitude greater than or equal to 2^d.

Example 23. *For* $v = \begin{bmatrix} 29821 \end{bmatrix}$, $\text{Rem}(v, 2^3) = 5$ *and* $C_{3,3}(v) = \begin{bmatrix} 7 & 1 & 58 \end{bmatrix}$.

We can extend the definition of $C_{e,d}$ to an arbitrary vector $v \in \mathbb{Z}^{n\times 1}$ in the following way. Let $v^{(+)}$ denote the vector v but with all negative entries zeroed out, and $v^{(-)} := v - v^{(+)}$ denote the vector v but with all but the positive entries zeroed out. Then, $v^{(+)}$ and $-v^{(-)}$ are over $\mathbb{Z}_{\geq 0}$, and $v = v^{(+)} - (-v^{(-)})$. Finally we let

$$C^*_{e,d}(v) := C_{e,d}(v^{(+)}) - C_{e,d}(-v^{(-)}),$$

which still satisfies equations (17) and (18) if we replace Rem and Quo by

$$\mathrm{Rem}^*(v, 2^d) := \mathrm{Rem}(v^{(+)}, 2^d) - \mathrm{Rem}(-v^{(-)}, 2^d),$$

$$\mathrm{Quo}^*(v, 2^d) := \mathrm{Quo}(v^{(+)}, 2^d) - \mathrm{Quo}(-v^{(-)}, 2^d).$$

We define structured matrices E_d and F_d by

$$E_d := -2^d \,\mathrm{Col}(I, 1) = \begin{bmatrix} -2^d \\ \\ \\ \end{bmatrix} \quad \text{and} \quad F_d := \begin{bmatrix} 1 & & & & \\ -2^d & 1 & & & \\ & -2^d & \ddots & & \\ & & \ddots & 1 & \\ & & & -2^d & 1 \end{bmatrix},$$

with the dimensions of E_d and F_d to be determined by the context. We remark that F_d^{-1} will be the unit lower triangular Toeplitz matrix with 2^{id} on the ith subdiagonal. The next lemma follows from the definition of E_d and F_d and equations (17) and (18).

Lemma 24. *Given $v \in \mathbb{Z}^{n\times 1}$, $e \in \mathbb{Z}_{\geq 0}$ and $d \in \mathbb{Z}_{\geq 1}$, let*

$$c := \begin{cases} v & \text{if } e = 0 \\ \mathrm{Rem}^*(v, 2^d) & \text{if } e > 0 \end{cases},$$

and

$$Q_{e,d}(v) = \left[\begin{array}{c|c|c} \mathrm{Quo}^*(v, 2^d) & \cdots & \mathrm{Quo}^*(v, 2^{ed}) \end{array}\right].$$

Then,

$$\left[\begin{array}{c|c} c & C^*_{e,d}(v) \\ \hline E_d & F_d \end{array}\right] = \left[\begin{array}{c|c} I_n & Q_{e,d}(v) \\ \hline & I_e \end{array}\right] \left[\begin{array}{c|c} v & \\ \hline & I_e \end{array}\right] \left[\begin{array}{c|c} 1 & \\ \hline E_d & F_d \end{array}\right]. \tag{19}$$

By replacing the single column vector v with a matrix $A = \left[\begin{array}{c|c|c} v_1 & \cdots & v_m \end{array}\right]$ of m column vectors v_i, we obtain:

Corollary 25. *Given $A = \left[\begin{array}{c|c|c} v_1 & \cdots & v_m \end{array}\right] \in \mathbb{Z}^{n\times m}$, $\bar{e} = (e_1, \ldots, e_m) \in \mathbb{Z}^m_{\geq 0}$ and $d \in \mathbb{Z}_{\geq 1}$. Let*

$$c_i := \begin{cases} v_i & \text{if } e_i = 0 \\ \mathrm{Rem}^*(v_i, 2^d) & \text{if } e_i > 0 \end{cases},$$

for $1 \leq i \leq m$, and define the matrix

$$D = D_{\bar{e},d}(A) := \left[\begin{array}{c|c|c||c|c|c} c_1 & \cdots & c_m & C^*_{e_1,d}(v_1) & \cdots & C^*_{e_m,d}(v_m) \\ \hline\hline E_d & & & F_d & & \\ \hline & \ddots & & & \ddots & \\ \hline & & E_d & & & F_d \end{array}\right] \in \mathbb{Z}^{\bar{n}\times\bar{m}},$$

with $\bar{n} = n + e_{[m]}$ and $\bar{m} = m + e_{[m]}$, where $e_{[m]} = e_1 + \cdots + e_m$. Then, matrix D satisfies

$$D = \begin{bmatrix} I_n & Q \\ & I_{e_{[m]}} \end{bmatrix} \begin{bmatrix} A & \\ & I_{e_{[m]}} \end{bmatrix} \begin{bmatrix} I_m & \\ E & F \end{bmatrix}, \tag{20}$$

where $Q = \begin{bmatrix} Q_{e_1,d}(v_1) \mid \cdots \mid Q_{e_m,d}(v_m) \end{bmatrix} \in \mathbb{Z}^{n \times e_{[m]}}$, $E = \operatorname{diag}(E_d, \ldots, E_d) \in \mathbb{Z}^{e_{[m]} \times m}$ and $F = \operatorname{diag}(F_d, \ldots, F_d) \in \mathbb{Z}^{e_{[m]} \times e_{[m]}}$.

From equation (20), it is apparent that D enjoys the following properties:

Corollary 26. *Given $A \in \mathbb{Z}^{n \times m}$, $\bar{e} = (e_1, \ldots, e_m) \in \mathbb{Z}^m_{\geq 0}$ and $d \in \mathbb{Z}_{\geq 1}$. Let $D = D_{\bar{e},d}(A)$ as in Corollary 25. Then*

(i) $\operatorname{rank}(D) = \operatorname{rank}(A) + e_{[m]}$.

(ii) D has the same Smith form as A up to additional trivial invariant factors.

Furthermore, if $n = m$, then:

(iii) $\det D = \det A$.

(iv) The principal $n \times n$ submatrix of the adjoint of D equals the adjoint of A.

Notice that Corollary 25 does not make any assumptions on the parameters $\bar{e}$ and d. The properties of matrix $D = D_{\bar{e},d}(A)$ corresponding to the original matrix A are true for any $\bar{e}$ and d. However, the partial linearization technique is particularly useful if we pick $\bar{e}$ and d in a way such that $\bar{m} \in O(m)$ and $\log \|D\|$ corresponds to the the average length of the columns of A. The following is the main result of this section.

Theorem 27. *Given matrix $A = \begin{bmatrix} v_1 \mid \cdots \mid v_m \end{bmatrix} \in \mathbb{Z}^{n \times m}$, let*

$$d := \left\lceil \sum_{i=1}^{m} \operatorname{length}(v_i)/m \right\rceil,$$

$\bar{e} = (e_1, \ldots, e_m) \in \mathbb{Z}^m_{\geq 0}$ where each $e_i \in \mathbb{Z}_{\geq 0}$ is chosen minimal such that $\operatorname{length}(v_i) \leq (e_i + 1)d$, and $D = D_{\bar{e},d}(A)$. Then:

- $\|D\| \leq 2^d$,
- $\bar{n} < n + m$ *and* $\bar{m} < 2m$.

Proof. The choice of e_i ensures that, for each v_i, the expansion in (18) is the 2^d-adic expansion of v. This shows that the length of all entries in the first n rows of D are bounded by d. Since the entries in the last $\bar{n} - n$ rows of D are bounded in magnitude by 2^d, the claimed bound for $\|D\|$ follows.

To prove our upper bounds for $\bar{n}$ and $\bar{m}$ we show that $\sum_{i=1}^m e_i < m$. Note that e_i is precisely defined as

$$e_i = \left\lceil \frac{\operatorname{length}(v_i)}{d} - 1 \right\rceil < \frac{\operatorname{length}(v_i)}{d},$$

and so

$$\sum_{i=1}^{m} e_i < \sum_{i=1}^{m} \frac{\operatorname{length}(v_i)}{d} \leq m.$$

□

Example 28. *Let*

$$A = \begin{bmatrix} 2 & 4 & 44199 & 3061969404 \\ 4 & 8 & 19644 & 765492351 \\ 7 & 8 & 44199 & 5358446457 \\ 7 & 5 & 9822 & 765492351 \end{bmatrix},$$

be the matrix from Example 22. Then, with the average (column) length $d = 14$ *and* $\bar{e} = (0, 0, 1, 2)$ *we get*

$$D = \left[\begin{array}{cccc||ccc} 2 & 4 & 11431 & 12796 & 2 & 6663 & 11 \\ 4 & 8 & 3260 & 15487 & 1 & 13953 & 2 \\ 7 & 8 & 11431 & 10105 & 2 & 15757 & 19 \\ 7 & 5 & 9822 & 15487 & 0 & 13953 & 2 \\ \hline\hline & & -16384 & & 1 & & \\ & & & -16384 & & 1 & \\ & & & & & -16384 & 1 \end{array}\right].$$

One can easily verify that the adjoint of A lies in the principal 4×4 *sub-matrix of the adjoint of D, and that the Smith form of A lies in the trailing* 4×4 *sub-matrix of the Smith form of D.*

The approach of Corollary 25 can also be used to partially linearize the rows of a matrix A. If we transpose a matrix A with skewed row lengths, then it has skewed column lengths. Then, by transposing the linearization of A^T, it satisfies all the properties given in Corollary 26. We can see that from the row linearization equivalent of equation (20), which is

$$D_{\bar{e},d}(A^T)^T = \begin{bmatrix} I & E^T \\ & B^T \end{bmatrix} \begin{bmatrix} A & \\ & I \end{bmatrix} \begin{bmatrix} I & \\ Q^T & I \end{bmatrix}. \tag{21}$$

Corollary 29. *Let* $A \in \mathbb{Z}^{m \times n}$*, and consider the matrix* $D = D_{\bar{e},d}(A^T)^T$*. The magnitude of the entries in D will then be bounded by* 2^d *where d is the average* length *over the rows of A, and D will enjoy all the properties following from Corollary 25 and Theorem 27.*

4.2. The permutation bound

Our approach so far is particularly effective for matrices $A \in \mathbb{Z}^{n \times n}$ where the average of the sum of the lengths of the columns (or rows) is small compared to length(A). However, the technique is not useful for input matrices that have, simultaneously, some columns and rows of large length. For this reason, as in the case of polynomial matrices (Gupta et al., 2012, Section 6), we develop an approach to handle such inputs based on the following *a priori* upper bound for $|\det A|$.

By definition, $\det A = \sum_{\sigma \in S_n} \operatorname{sign}(\sigma) \prod_{i=1}^{n} A_{i,\sigma_i}$, where S_n is the set of all permutations of $(1, 2, \ldots, n)$. Therefore,

$$\det A \leq n! \max_{\sigma \in S_n} \prod_{i=1}^{n} |A_{i,\sigma_i}|,$$

and so, we define

$$\text{PermutBnd}(A) := \max_{\sigma \in S_n} \sum_{i=1}^{n} \text{length}(A_{i,\sigma_i}).$$

As in the polynomial case, up to a row and column permutation, we may assume that $d_i := \text{length}(A_{i,i})$ bounds the length of the submatrix $A_{i\ldots n,i\ldots n}$, for $1 \le i \le n$. Such a row and column permutation can be found by sorting the set of triples $\{(i, j, |A_{i,j}|)\}_{1\le i,j\le n}$ into nonincreasing order according to their third component. Then, by definition, $d_1 + \cdots + d_n \le \text{PermutBnd}(A)$.

Let $d := \left\lceil \sum_{i=1}^{n} d_i/n \right\rceil$ and $\bar{e} = (e_1, \ldots, e_n)$ with $e_i \in \mathbb{Z}_{\ge 0}$ minimal such that $d_i \le (e_i + 1)d$. Then, due to the choice of d_i, row i of matrix $D_{\bar{e},d}(A)$ will have length bounded by $d_i + 1$ for $1 \le i \le n$, and all the extra rows will have length bounded by $d + 1$. Furthermore, let $\bar{e}'$ contain $\bar{e}$ augmented with $\sum_{i=1}^{n} e_i$ zeros. We have the following corollary for matrix $D := D_{\bar{e}',d}(D_{\bar{e},d}(A)^T)^T$.

Corollary 30. *Let $A \in \mathbb{Z}^{n\times n}$ be given. Using the choices for d, $\bar{e}$ and $\bar{e}'$ as specified above, the matrix $D := D_{\bar{e}',d}(D_{\bar{e},d}(A)^T)^T \in \mathbb{Z}^{\bar{n}'\times\bar{n}'}$ satisfies*

(i) $\|D\| \le 2^d$ *with* $d \le \lceil \text{PermutBnd}(A)/n \rceil$, *and*

(ii) $\bar{n}' < 3n$,

along with all the properties from Corollary 26.

Remark 31 (Application to system solving)**.** *The fact that the principal $n \times n$ submatrix of the adjoint of the partially linearized matrix D is equal to the adjoint of the original matrix A provides us with a direct extension to system solving. For any matrix $B \in \mathbb{Z}^{n\times m}$, we have that the first n rows of*

$$D^{-1}\begin{bmatrix} B \\ 0 \end{bmatrix}$$

are equal to $A^{-1}B$. Therefore, Theorem 17 can have cost which depends on the average bitlength d of A and not the bitlength of the largest entry. The average bitlength d can assume any of the three definitions given by Theorem 27, Corollary 29 and Corollary 30.

Remark 32 (Application to integrality certification)**.** *Suppose D is a partial linearization of A. For any $s \in \mathbb{Z}_{>0}$ and $B \in \mathbb{Z}^{n\times m}$, it follows from equations (20) and (21) that*

$$sD^{-1}\begin{bmatrix} B \\ 0 \end{bmatrix}$$

will be integral if and only if $sA^{-1}B$ is integral. Therefore, Theorem 18 can have cost which depends on the average bitlength d of A and not the bitlength of the largest entry. The average bitlength d can assume any of the three definitions given by Theorem 27, Corollary 29 and Corollary 30.

Remark 33 (Application to inverting unimodular matrices)**.** *Suppose D is a partial linearization of a unimodular matrix A. A straight line formula for A^{-1} is given by*

$$\left[\begin{array}{c|c} I_n & 0 \end{array}\right] T \left[\begin{array}{c} I_n \\ \hline 0 \end{array}\right]$$

where T is a straight line formula for the inverse of a partial linearization of A. Such a straight line formula for A^{-1} can thus be computed deterministically in $O(n^\omega \mathsf{M}(\log n + d)\log n)$ bit operations by (Pauderis and Storjohann, 2012, Section 3), where d is the average bitlength of A according to any of the three definitions given by Theorem 27, Corollary 29 and Corollary 30.

Remark 34 (Application to computing the Hermite form)**.** *If $A \in \mathbb{Z}^{n\times n}$ is nonsingular, then the lower triangular row Hermite form of A shows up as the principal $n \times n$ submatrix of the Hermite form of the partially linearized matrix D.*

Example 35. *The lower triangular row Hermite form of the matrix D from Example 28 is*

$$\left[\begin{array}{cccc||cc} 777 & & & & & \\ 401 & 1 & & & & \\ 174 & 0 & 4911 & & & \\ 762 & 0 & 0 & 765492351 & & \\ \hline\hline 696 & 0 & 3260 & 0 & 1 & \\ 762 & 0 & 0 & 765475967 & & 1 \\ 762 & 0 & 0 & 497056895 & & \end{array}\right]$$

with the 4×4 principal sub-matrix being the corresponding lower triangular row Hermite form of A.

4.3. Smith massagers and partial linearization

We can also employ the partial linearization technique to replace the $\log \|A\|$ term in Theorem 19 with the average bitlength d of the columns (or rows) in A.

Theorem 36. *Let $A \in \mathbb{Z}^{n\times n}$ and $D \in \mathbb{Z}^{\bar{n}\times\bar{n}}$ be the partially linearized version of A from Theorem 27. If*

$$\left(\begin{bmatrix} I_{\bar{n}-n} & \\ & S \end{bmatrix}, \begin{bmatrix} 0 & M_1 \\ 0 & M_2 \end{bmatrix}\right) \tag{22}$$

is a Smith massager for D, where $S \in \mathbb{Z}^{n\times n}$, $M_1 \in \mathbb{Z}^{n\times n}$ and $M_2 \in \mathbb{Z}^{(\bar{n}-n)\times n}$, then (S, M_1) is a Smith massager for A.

Proof. We will show that $S, M_1 \in \mathbb{Z}^{n\times n}$ satisfy Definition 1 for A.

From Theorem 27, we have that

$$\begin{aligned} D\begin{bmatrix} 0 & M_1 \\ 0 & M_2 \end{bmatrix} &= \begin{bmatrix} I_n & Q \\ & I_{\bar{n}-n} \end{bmatrix}\begin{bmatrix} A & \\ & I_{\bar{n}-n} \end{bmatrix}\begin{bmatrix} I_n & \\ E & F \end{bmatrix}\begin{bmatrix} 0 & M_1 \\ 0 & M_2 \end{bmatrix} \\ &= \begin{bmatrix} I_n & Q \\ & I_{\bar{n}-n} \end{bmatrix}\begin{bmatrix} 0 & AM_1 \\ 0 & EM_1 + FM_2 \end{bmatrix}. \end{aligned}$$

Since (22) is a Smith massager for D, it follows from Definition 1.(i) that

$$D\begin{bmatrix} 0 & M_1 \\ 0 & M_2 \end{bmatrix} \equiv 0 \bmod \begin{bmatrix} I_{\bar{n}-n} & \\ & S \end{bmatrix},$$

it follows that

$$\begin{bmatrix} 0 & AM_1 \\ 0 & EM_1 + FM_2 \end{bmatrix} \equiv 0 \bmod \begin{bmatrix} I_{\bar{n}-n} & \\ & S \end{bmatrix},$$

and that

$$AM_1 \equiv 0 \bmod S.$$

Moreover, since B is unit lower triangular, we see that

$$M_2 \equiv -F^{-1}EM_1 \bmod S.$$

Finally, by Definition 1.(ii), there exist a matrix $W_D \in \mathbb{Z}^{\bar{n}\times\bar{n}}$ such that

$$W_D \begin{bmatrix} 0 & M_1 \\ 0 & M_2 \end{bmatrix} \equiv \begin{bmatrix} I_{\bar{n}-n} & \\ & I_n \end{bmatrix} \mathbf{c}\mathrm{mod} \begin{bmatrix} I_{\bar{n}-n} & \\ & S \end{bmatrix}.$$

The last equation can be transformed to

$$\left(W_D \begin{bmatrix} I_n & \\ -F^{-1}E & I_{\bar{n}-n} \end{bmatrix}\right) \begin{bmatrix} 0 & M_1 \\ 0 & 0 \end{bmatrix} \equiv \begin{bmatrix} I_{\bar{n}-n} & \\ & I_n \end{bmatrix} \mathbf{c}\mathrm{mod} \begin{bmatrix} I_{\bar{n}-n} & \\ & S \end{bmatrix},$$

from which it directly follows that there exists a matrix $W \in \mathbb{Z}^{n\times n}$ such that

$$WM_1 \equiv I_n \,\mathbf{c}\mathrm{mod}\, S.$$

□

Furthermore, by equation (21) and by following the same steps as in the proof Theorem 36, we obtain the following corollary.

Corollary 37. *Let $A \in \mathbb{Z}^{n\times n}$ and $D \in \mathbb{Z}^{\bar{n}\times\bar{n}}$ be the partially linearized version of A from Corollary 29 or Corollary 30. If*

$$\left(\begin{bmatrix} I_{\bar{n}-n} & \\ & S \end{bmatrix}, \begin{bmatrix} 0 & M_1 \\ 0 & M_2 \end{bmatrix}\right)$$

is a Smith massager for D, where $S \in \mathbb{Z}^{n\times n}$, $M_1 \in \mathbb{Z}^{n\times n}$ and $M_2 \in \mathbb{Z}^{(\bar{n}-n)\times n}$, then (S, M_1) is a Smith massager for A.

5. Example

In this section, we illustrate our Smith form with multipliers algorithm using the following example. We have already discussed the algorithm in Section 2.2, and we will rigorously establish it in Sections 6–8.

Example 38. *Let our input matrix be*

$$A := \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 1 & 2 & 4 & 1 \\ 1 & 3 & 2 & 6 & 4 & 5 & 1 \\ 1 & 4 & 2 & 1 & 4 & 2 & 1 \\ 1 & 5 & 4 & 6 & 2 & 3 & 1 \\ 1 & 6 & 1 & 6 & 1 & 6 & 1 \end{bmatrix}.$$

Given as input $2A$, the algorithm supporting Theorem 19 returns the Smith form $2S$ and a Smith massager M for $2A$:

$$2S := \begin{bmatrix} 2 & & & & & & \\ & 2 & & & & & \\ & & 2 & & & & \\ & & & 2 & & & \\ & & & & 2 & & \\ & & & & & 16 & \\ & & & & & & 160 \end{bmatrix}, \quad M := \begin{bmatrix} 1 & 0 & 1 & 1 & 2 & 8 & 0 \\ 0 & 1 & 1 & 0 & 2 & 11 & 65 \\ 1 & 0 & 1 & 1 & 1 & 12 & 15 \\ 0 & 1 & 1 & 1 & 3 & 6 & 98 \\ 0 & 0 & 0 & 0 & 0 & 12 & 155 \\ 1 & 1 & 1 & 1 & 1 & 7 & 125 \\ 1 & 1 & 1 & 1 & 1 & 0 & 2 \end{bmatrix}.$$

We always take M to be reduced column modulo $2S$, that is, it should be a reduced Smith massager.

The next step is to pick a random matrix

$$R := \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 \end{bmatrix},$$

where each entry is chosen independently and uniformly from a set $[0, \lambda - 1]$ of $\lambda \in O(n\|A\|)$ consecutive integers. (For the example, we let $\lambda := 2$.)

By perturbing M by the random choice of R post-multiplied with $2S$, we obtain

$$B := M + 2RS = \begin{bmatrix} 1 & 0 & 3 & 3 & 6 & 8 & 160 \\ 2 & 3 & 1 & 0 & 6 & 11 & 225 \\ 1 & 0 & 1 & 1 & 1 & 12 & 15 \\ 2 & 3 & 3 & 3 & 3 & 6 & 98 \\ 0 & 2 & 0 & 2 & 0 & 28 & 155 \\ 1 & 3 & 3 & 3 & 5 & 23 & 125 \\ 3 & 3 & 1 & 1 & 5 & 16 & 22 \end{bmatrix},$$

which, by Proposition 8, is a Smith massager for A.

Computing the lower triangular row Hermite form of the random matrix B, gives

$$H := \begin{bmatrix} 830295 & & & & & & \\ 547348 & 1 & & & & & \\ 602711 & & 1 & & & & \\ 592450 & & & 1 & & & \\ 540934 & & & & 1 & & \\ 350043 & & & & & 1 & \\ 323815 & & & & & & 1 \end{bmatrix}.$$

Our aim is for H to have only the first diagonal entry non-trivial. If B is not left equivalent to such a matrix H, then the algorithm fails. This happens, for example, if the random R has the entry in row 1 and column 6 equal to 1 rather than 0. Showing that the Hermite form of B is almost trivial with high probability is the main focus of Section 6. Then, in Section 7, we give an algorithm to assay if the Hermite form of B has the desired structure, and if so, to compute the Hermite form itself.

To obtain a unimodular Smith massager, we simply extract H from B by post-multiplying with H^{-1}.

$$V := BH^{-1} = \begin{bmatrix} -74 & 0 & 3 & 3 & 6 & 8 & 160 \\ -99 & 3 & 1 & 0 & 6 & 11 & 225 \\ -13 & 0 & 1 & 1 & 1 & 12 & 15 \\ -49 & 3 & 3 & 3 & 3 & 6 & 98 \\ -75 & 2 & 0 & 2 & 0 & 28 & 155 \\ -68 & 3 & 3 & 3 & 5 & 23 & 125 \\ -22 & 3 & 1 & 1 & 5 & 16 & 22 \end{bmatrix}.$$

By construction, the matrix V is integral and unimodular. In addition, and as proven by Lemma 12, V is a Smith massager for A.

The fact that H has only one non-trivial column allows us to easily establish a nice bound on the size of matrix V. Notice that the columns of V have the same bitlength as the columns of B except for only the first column. In addition, the bitlength of the columns of B equals the bitlength of the columns of the Smith massager M plus the bitlength of λ. In Section 8, we give the overall algorithm for computing the Smith multipliers and establish explicit bounds on the size of their entries.

Finally, since V is a unimodular Smith massager for A, this makes the matrix

$$U := AVS^{-1} = \begin{bmatrix} -74 & 0 & 3 & 3 & 3 & 1 & 2 \\ -400 & 14 & 12 & 13 & 13 & 13 & 10 \\ -817 & 28 & 25 & 27 & 25 & 31 & 20 \\ -1353 & 53 & 42 & 47 & 37 & 43 & 34 \\ -1003 & 32 & 19 & 23 & 25 & 32 & 26 \\ -1291 & 49 & 40 & 39 & 39 & 36 & 33 \\ -1480 & 59 & 47 & 43 & 48 & 38 & 38 \end{bmatrix}$$

also integral and unimodular. By construction, the two unimodular matrices $V, U \in \mathbb{Z}^{n\times n}$ satisfy $AV = US$.

6. Random perturbations of Smith massagers

Let $A \in \mathbb{Z}^{n\times n}$ be nonsingular with Smith form S. In this section, we show how to perturb a Smith massager M for A into a unimodular Smith massager V. The first step will be to obtain a Smith massager $B := M + RS$ that is left equivalent (over $\mathbb{Z}$) to a lower triangular row Hermite form with the shape

$$\begin{bmatrix} |\det B| & & & & \\ * & 1 & & & \\ * & & 1 & & \\ \vdots & & & \ddots & \\ * & & & & 1 \end{bmatrix} \in \mathbb{Z}^{n\times n}. \tag{23}$$

The property that the last $n-1$ diagonal entries of B are equal to 1 coincides with the property that the last $n-1$ columns of B mod p are linearly independent over $\mathbb{Z}/(p)$ for all primes p.

Our approach is inspired by and follows that of Eberly et al. (2000, Section 6), where the following general result is established: for $\lambda \geq 2$, a matrix $R \in \mathbb{Z}^{n\times n}$ with entries chosen uniformly and randomly from $[0, \lambda - 1]$ will have an expected number of $O(\log_\lambda n)$ nontrivial invariant factors.

Theorem 39. *Let $A \in \mathbb{Z}^{n\times n}$ be nonsingular with Smith form S. Let M be a reduced Smith massager for $2A$. For any $R \in \mathbb{Z}^{n\times n}$,*

(i) the matrix $B := M + 2RS$ is a Smith massager for A, and

(ii) if entries in R are chosen chosen uniformly and randomly from $[0, \lambda - 1]$, where

$$\lambda = 105 \max(n, \lceil (\det 2S)^{1/n} \rceil),$$

then the probability that there exists a prime p such that the last $n-1$ columns of B mod p are linearly dependent over $\mathbb{Z}/(p)$ is less than $1/2$.

Part (i) of Theorem 39 follows directly from Proposition 8, so it remains only to prove part (ii). This will be done using a sequence of lemmas. For the rest of this section, we let A, S, M, R, λ and $B = M + 2RS$ be as defined in Theorem 39.

We start by defining a set of probabilistic events that will facilitate the proofs in this section. For a prime p and $1 \le m \le n-1$, let Dep_m^p denote the event that the last m columns of B are linearly dependent modulo p. To complete the proof of Theorem 39 we show that $\Pr[\vee_p \mathsf{Dep}_{n-1}^p] < 0.5$, where $\vee_p$ means ranging over all primes. We begin with Lemmas 40 and 41 that hold for all primes p. Then, following Eberly et al. (2000, Section 6), we will separately consider the small primes $p < \lambda$ in Subsection 6.1, and the large primes $p \ge \lambda$ in Subsection 6.2.

Lemma 40. *For any prime p we have*

$$\Pr[\mathsf{Dep}_1^p] \le \left(\frac{1}{\lambda}\left\lceil\frac{\lambda}{p}\right\rceil\right)^n, \tag{24}$$

and for any $2 \le m \le n-1$,

$$\Pr[\mathsf{Dep}_m^p \mid \neg\mathsf{Dep}_{m-1}^p] \le \left(\frac{1}{\lambda}\left\lceil\frac{\lambda}{p}\right\rceil\right)^{n-m+1}. \tag{25}$$

Proof. We have Dep_1^p precisely when the last column of B is zero modulo p. By Lemma 10, for any prime p that divides $2s_n$ we have $\Pr[\mathsf{Dep}_1^p] = 0$. For a prime p that does not divide $2s_n$, Dep_1^p is equivalent to the vector

$$\underbrace{(2s_n)^{-1}M_{1..n,n}}_{\text{fixed}} + R_{1..n,n} \bmod p \in \mathbb{Z}/(p)^{n\times 1} \tag{26}$$

being zero modulo p. Each random entry $R_{i,n}$ is equal to $-(2s_n)^{-1}M_{i,n}$ modulo p with probability at most

$$\frac{1}{\lambda}\left\lceil\frac{\lambda}{p}\right\rceil.$$

The bound (24) now follows by noting that vector in (26) has n entries.

Now consider the case $2 \le m \le n-1$. By Lemma 10, we have that $\Pr[\mathsf{Dep}_m^p] = 0$ for any prime p that divides $2s_{n-m+1}$. Assume henceforth that p does not divide $2s_{n-m+1}$. Given $\neg\mathsf{Dep}_{m-1}^p$, there is an $(m-1)\times(m-1)$ submatrix D in the last $m-1$ columns of B that is nonsingular modulo p. Assume, without loss of generality, up to a row permutation of B, that D is the trailing $(m-1)\times(m-1)$ submatrix of B. Decompose the last m columns of B as follows:

$$\left[\begin{array}{c|c} v & C \\ \hline w & D \end{array}\right] \in \mathbb{Z}^{n\times m}.$$

Then C and D are fixed at this point and vectors v and w still depend on the random choice of column $n-m+1$ of R. Fix the choice of w also. Note that

$$\left[\begin{array}{c|c} I_{n-m+1} & -CD^{-1} \\ \hline & D^{-1} \end{array}\right]\left[\begin{array}{c|c} v & C \\ \hline w & D \end{array}\right] = \left[\begin{array}{c|c} a & \\ \hline * & I_{m-1} \end{array}\right] \bmod p \in \mathbb{Z}/(p)^{n\times m}.$$

Then Dep_m^p is equivalent to the vector

$$(2s_{n-m+1})^{-1}a = \underbrace{(2s_{n-m+1})^{-1}M_{1..n-m+1,n-m+1} - CD^{-1}w}_{\text{fixed}} + R_{1..n-m+1,n-m+1} \bmod p \in \mathbb{Z}/(p)^{(n-m+1)\times 1}$$

being zero modulo p. By a similar argument as before, the probability of this happening is bounded by (25). □

The next lemma follows simply from the union bound on the set of events for $1 \le i \le n-1$ that happen when the ith column from the end is the first that is linearly dependent.

Lemma 41. *For any prime p we have*

$$\Pr[\mathsf{Dep}^p_{n-1}] \le \Pr[\mathsf{Dep}^p_1] + \sum_{i=2}^{n-1} \Pr[\mathsf{Dep}^p_i \mid \neg\mathsf{Dep}^p_{i-1}].$$

6.1. Small primes

We first deal with the specific small primes $\{3, 5, 7\}$. Notice that from Proposition 8, we know that $\Pr[\mathsf{Dep}^2_{n-1}] = 0$.

Lemma 42. $\Pr[\vee_{p\in\{3,5,7\}}\mathsf{Dep}^p_{n-1}] < 0.23$.

Proof. We exploit the fact that λ is a multiple of $105 = 3 \times 5 \times 7$. Let $p \in \{3, 5, 7\}$. Since $p \mid \lambda$, the bound of Lemma 40 simplifies to

$$\Pr[\mathsf{Dep}^p_m \mid \neg\mathsf{Dep}^p_{m-1}] \le \left(\frac{1}{p}\right)^{n-m+1},$$

and Lemma 41 gives

$$\Pr[\mathsf{Dep}^p_{n-1}] \le \sum_{i=1}^{n-1}\left(\frac{1}{p}\right)^{i+1} < \frac{1}{p}\sum_{i=1}^{\infty}\left(\frac{1}{p}\right)^{i} = \frac{1}{p(p-1)}. \tag{27}$$

Since the events Dep^3_{n-1}, Dep^5_{n-1} and Dep^7_{n-1} are independent,

$$\Pr[\vee_{p\in\{3,5,7\}}\mathsf{Dep}^p_{n-1}] = 1 - \prod_{p\in\{3,5,7\}}(1 - \Pr[\mathsf{Dep}^p_{n-1}]). \tag{28}$$

The result now follows by bounding from above the probabilities on the right hand size of (28) using (27). □

Next we handle the small primes in the range $7 < p < \lambda$.

Lemma 43. $\Pr[\vee_{7<p<\lambda}\mathsf{Dep}^p_{n-1}] < 0.23$

Proof. Let $7 < p < \lambda$. Since $p < \lambda$,

$$\frac{1}{\lambda}\left\lceil\frac{\lambda}{p}\right\rceil < \frac{1}{\lambda}\left(\frac{\lambda}{p}+1\right) = \frac{1}{p} + \frac{1}{\lambda} < \frac{2}{p} = \frac{1}{p/2},$$

and the bound of Lemma 40 simplifies to

$$\Pr[\mathsf{Dep}^p_m \mid \neg\mathsf{Dep}^p_{m-1}] \le \left(\frac{1}{p/2}\right)^{n-m+1}. \tag{29}$$

Lemma 41 together with (29) gives

$$\Pr[\mathsf{Dep}^p_{n-1}] \leq \frac{1}{(p/2)(p/2-1)} < \frac{1}{((p-1)/2)^2}. \tag{30}$$

Using the union bound and then (30) gives

$$\begin{aligned}
\Pr[\vee_{7<p<\lambda}\mathsf{Dep}^p_{n-1}] &\leq \sum_{7<p<\lambda} \Pr[\mathsf{Dep}^p_{n-1}] \\
&< \sum_{7<p<\lambda} \frac{1}{((p-1)/2)^2} \\
&< \sum_{x\geq 11,\ \text{odd}} \frac{1}{((x-1)/2)^2} \\
&= \sum_{x\geq 5} \frac{1}{x^2} \\
&= \zeta(2) - \sum_{x=1}^{4} \frac{1}{x^2} \\
&= \frac{\pi^2}{6} - \frac{205}{144} \\
&< 0.23.
\end{aligned}$$

□

6.2. Large primes

Consider now the large primes $p \geq \lambda$. Although it follows from Lemmas 40 and 41 that $\Pr[\mathsf{Dep}^p_{n-1}] \leq (1/(\lambda(\lambda-1))$ for any particular prime $p \geq \lambda$, this doesn't help us to bound $\Pr[\vee_{p\geq\lambda}\mathsf{Dep}^p_{n-1}]$ using the union bound since there exist an infinite number of such primes. Instead, we follow the approach of Eberly et al. (2000, Section 6) and show that we only need to consider those primes which divide some necessarily nonzero minors of B.

Lemma 44. *Any minor of B is bounded in magnitude by $\lambda^{2.5n}$.*

Proof. It will suffice to bound $|\det B|$ using Hadamard's inequality, which states that $|\det B|$ is bounded by the product of the Euclidean norms of the columns of B. Recall that $B = M + 2RS$ where $M = M \bmod 2S$ and entries in R are chosen from $[0, \lambda-1]$, with $\lambda \geq \max((\det 2S)^{1/n}, n)$. Then

$$\begin{aligned}
|\det B| &\leq \prod_{j=1}^{n} \left\| B_{1\ldots n,j} \right\|_2 \\
&= \prod_{j=1}^{n} \left\| M_{1\ldots n,j} + 2s_j R_{1\ldots n,j} \right\|_2 \\
&\leq \prod_{j=1}^{n} n^{1/2}(2s_j - 1 + 2s_j(\lambda-1)) \\
&< (\det 2S) n^{n/2} \lambda^n \\
&\leq \lambda^{2.5n}.
\end{aligned}$$

□

Next we develop the following analogue of Lemma 40.

Lemma 45. *We have*

$$\Pr[\vee_{p\geq\lambda}\mathsf{Dep}_1^p] \leq 2.53n\left(\frac{1}{\lambda}\right)^{n-1}$$

and for any $2 \leq m \leq n-1$,

$$\Pr[\vee_{p\geq\lambda}\mathsf{Dep}_m^p \mid \neg \vee_{p\geq\lambda} \mathsf{Dep}_{m-1}^p] \leq 2.53n\left(\frac{1}{\lambda}\right)^{n-m}.$$

Proof. By Proposition 8, $B = M + R(2S)$ is nonsingular modulo 2, independent of the choice of R. Thus, up to an initial row permutation of M, we may assume that the trailing $j \times j$ submatrix of B mod 2 is nonsingular over $\mathbb{Z}/(2)$ for every $1 \leq j \leq n$.

First consider the case for $m = 1$. Decompose the last column of B as

$$\left[\begin{array}{c} v \\ w \end{array}\right] \in \mathbb{Z}^{n\times 1},$$

where $v \in \mathbb{Z}^{(n-1)\times 1}$ and $w \in \mathbb{Z}$. Fix the choice of w, that is, fix the last entry in the last column of R. By assumption, $w \neq 0 \bmod 2$ and thus $w \neq 0$ over $\mathbb{Z}$. For every prime $p \nmid w$ we have $\Pr[\mathsf{Dep}_1^p] = 0$, and since there are $n-1$ entries in v that are still free to be chosen, the union bound gives

$$\begin{aligned}\Pr[\vee_{p\geq\lambda}\mathsf{Dep}_1^p] &= \Pr[\vee_{p\geq\lambda,p|w}\mathsf{Dep}_1^p] \\ &\leq (\log_\lambda |w|)\left(\frac{1}{\lambda}\right)^{n-1}.\end{aligned}$$

Lemma 44 gives $\log_\lambda |w| \leq 2.5n < 2.53n$, establishing the first part of the lemma.

Now consider $2 \leq m \leq n-1$. Decompose the last m columns of B as follows:

$$\left[\begin{array}{c|c} v & C \\ \hline w & D \end{array}\right] \in \mathbb{Z}^{n\times m},$$

where $D \in \mathbb{Z}^{(m-1)\times(m-1)}$. Then C and D are fixed at this point and vectors v and w still depend on the random choice of column $n-m+1$ of R. Let $d = \det D$, which we know to be nonzero. There are at most $\log_\lambda |d|$ primes $p \geq \lambda$ that divide d. Using Lemma 40 with the union bound gives

$$\sum_{p\geq\lambda,p|d} \Pr[\mathsf{Dep}_m^p \mid \neg\mathsf{Dep}_{m-1}^p] \leq (\log_\lambda |d|)\left(\frac{1}{\lambda}\right)^{n-m+1}. \tag{31}$$

Next we consider the primes $p \nmid d$. Note that

$$\left[\begin{array}{c|c} dI_{n-m+1} & -dCD^{-1} \\ \hline & dD^{-1} \end{array}\right]\left[\begin{array}{c|c} v & C \\ \hline w & D \end{array}\right] = \left[\begin{array}{c|c} \begin{array}{c} a_1 \\ \vdots \\ a_{n-m} \\ \hline a_{n-m+1} \end{array} & \\ \hline * & dI_{m-1} \end{array}\right] \in \mathbb{Z}^{n\times m},$$

where, by Cramer's rule, a_{n-m+1} is the determinant of the trailing $m \times m$ submatrix of B. Since $p \nmid d$, event Dep_m^p holds if and only if the vector

$$\begin{bmatrix} a_1 \\ \vdots \\ a_{n-m} \\ a_{n-m+1} \end{bmatrix} = d \begin{bmatrix} v_1 \\ \vdots \\ v_{n-m} \\ v_{n-m+1} \end{bmatrix} - dCD^{-1}w. \tag{32}$$

is zero modulo p. Fix the choice of w and v_{n-m+1}. Then $a_{n-m+1} \neq 0$ is also fixed, and for every prime $p \nmid a_{n-m+1}$ we have $\Pr[\mathsf{Dep}_m^p \mid \neg\mathsf{Dep}_{m-1}^p] = 0$. Since there can be at most $\log_\lambda |a_{n-m+1}|$ primes $p \geq \lambda$ that divide a_{n-m+1}, and since $v_1, \ldots, v_{n-m}$ are still free to be chosen, we have

$$\sum_{p \geq \lambda, p \nmid d} \Pr[\mathsf{Dep}_m^p \mid \neg\mathsf{Dep}_{m-1}^p] \leq (\log_\lambda |a_{n-m+1}|) \left(\frac{1}{\lambda}\right)^{n-m}. \tag{33}$$

Combining the bounds (31) and (33) and using the estimate of Lemma 44 for $|d|$ and $|a_{n-m+1}|$, we obtain

$$\begin{aligned} \Pr[\vee_{p\geq\lambda}\mathsf{Dep}_m^p \mid \neg \vee_{p\geq\lambda} \mathsf{Dep}_{m-1}^p] &\leq 2.5n\left(\left(\frac{1}{\lambda}\right)^{n-m+1} + \left(\frac{1}{\lambda}\right)^{n-m}\right) \\ &= 2.5n\left(\frac{1}{\lambda}\right)^{n-m}\left(\frac{1}{\lambda} + 1\right) \\ &< 2.53n\left(\frac{1}{\lambda}\right)^{n-m}. \end{aligned} \tag{34}$$

Here, (34) follows using $\lambda \geq 105$. □

Lemma 46. $\Pr[\vee_{p\geq\lambda}\mathsf{Dep}_{n-1}^p] < 0.03$.

Proof. Analogous to Lemma 41, we have

$$\Pr[\vee_{p\geq\lambda}\mathsf{Dep}_{n-1}^p] \leq \Pr[\vee_{p\geq\lambda}\mathsf{Dep}_1^p] + \sum_{i=2}^{n-1} \Pr[\vee_{p\geq\lambda}\mathsf{Dep}_i^p \mid \neg \vee_{p\geq\lambda} \mathsf{Dep}_{i-1}^p].$$

Using the estimates of Lemma 45 now gives

$$\begin{aligned} \Pr[\vee_{p\geq\lambda}\mathsf{Dep}_{n-1}^p] &\leq 2.53n\left(\frac{1}{\lambda}\right)^{n-1} + 2.53n\sum_{i=2}^{n-1}\left(\frac{1}{\lambda}\right)^{n-i} \\ &< 2.53n\left(\frac{1}{\lambda - 1}\right). \end{aligned}$$

Simplifying the last bound using the assumption $\lambda \geq 105n$ gives the result. □

Proof of Theorem 39. The probability defined by Theorem 39 is bounded by the sum of probabilities in Lemmas 42, 43 and 46, that is,

$$\begin{aligned} \Pr[\mathsf{Dep}_{n-1}] &\leq \Pr[\vee_{p\in\{3,5,7\}}\mathsf{Dep}_{n-1}^p] + \Pr[\vee_{7<p<\lambda}\mathsf{Dep}_{n-1}^p] + \Pr[\vee_{p\geq\lambda}\mathsf{Dep}_{n-1}^p] \\ &< 0.23 + 0.23 + 0.03 \\ &< 0.5. \end{aligned}$$

□

7. Almost trivial Hermite form certification

In this section, we show how to verify whether the last $n-1$ columns of the matrix $B \in \mathbb{Z}^{n\times n}$ from Theorem 39 are linearly independent for any prime $p \in \mathbb{Z}$. As we have already mentioned, this means that B is left equivalent to a lower triangular row Hermite form with the shape

$$H = \begin{bmatrix} |\det B| & & & \\ * & 1 & & \\ \vdots & & \ddots & \\ * & & & 1 \end{bmatrix} \in \mathbb{Z}^{n\times n}. \tag{35}$$

Our main tool will once more be the Smith form and a Smith massager for B.

Theorem 47. *Let $A \in \mathbb{Z}^{n\times n}$ be nonsingular with Smith form S and a Smith massager M. If $H \in \mathbb{Z}^{n\times n}$ is a matrix in Hermite form which satisfies that $\det H = \det S$ and $HM \equiv 0 \bmod S$, then H is the row Hermite form of A.*

Proof. The statement follows from Theorem 7 and the uniqueness of the Hermite form of A. □

We plan to use the description of Theorem 47 here in order to check whether the lower triangular row Hermite form H of the matrix B has $n-1$ trailing trivial columns, and, if yes, then also compute the first non-trivial column. For this section, matrices S and M refer to the Smith form and Smith massager of matrix B.

First of all, we need to ensure that the Smith form $S := \operatorname{diag}(s_1, \ldots, s_n)$ of B also has only one non-trivial invariant factor. If otherwise, then H does not have the desired structure. Let $h_1, h_2, \ldots, h_n$ be the diagonal entries of H. The product $h_2 \cdots h_n$ equals the gcd of all the $(n-1)\times(n-1)$ minors in the last $n-1$ columns of B. On the other hand, the product $s_1 \cdots s_{n-1}$ equals the gcd of all the $(n-1)\times(n-1)$ minors of B, which means that $(s_1 \cdots s_{n-1}) \mid (h_2 \cdots h_n)$. So, if $s_1 \cdots s_{n-1} \neq 1$, then $h_2 \cdots h_n \neq 1$.

Now, assuming that $S := \operatorname{diag}(1, \ldots, 1, s_n)$, we are looking to see whether there exists a vector $\bar{h} \in \mathbb{Z}^{(n-1)\times 1}$ such that

$$\begin{bmatrix} s_n & \\ \bar{h} & I_{n-1} \end{bmatrix} M_{1..n,n} \equiv 0 \bmod s_n,$$

which is equivalent to

$$M_{1,n}\bar{h} + M_{2..n,n} \equiv 0 \bmod s_n. \tag{36}$$

Since the Hermite form H must be unique, equation (36) must have exactly one solution, which is true if and only if $\gcd(M_{1,n}, s_n) = 1$.

The algorithm follows.

`TrivialLowerHermiteForm`(B)
Input: A nonsingular matrix $B \in \mathbb{Z}^{n\times n}$.
Output: The lower triangular Hermite form $H \in \mathbb{Z}^{n\times n}$ of B if only the first column is non-trivial, otherwise NotTrivial.
Note: Fail might be returned with probability less than $1/8$.

1. [Compute a Smith massager for B.]
   (If `SmithMassager` fails, return Fail)
   $S, M :=$ `SmithMassager`(B)

2. [Certify that B is left equivalent to a matrix H as in (35).]
   if $S_{n-1,n-1} \neq 1$ **then return** NotTrivial
   if $\gcd(S_{n,n}, M_{1,n}) \neq 1$ **then return** NotTrivial

3. [Compute matrix H and return.]
   $H := \begin{bmatrix} h_1 & \\ \bar{h} & I_{n-1} \end{bmatrix}$
   where $h_1 := S_{n,n}$ and $\bar{h} := \mathrm{Rem}(-M_{1,n}^{-1} M_{2..n,n}, S_{n,n})$.
   return H

Figure 1: Algorithm `TrivialLowerHermiteForm`

Theorem 48. *Algorithm* `TrivialLowerHermiteForm` *is correct and runs in time*

$$O(n^\omega \mathsf{B}(d + \log n)(\log n)^2),$$

where d is the average bitlength of the columns of $B \in \mathbb{Z}^{n\times n}$.

Proof. The correctness follows from the preceding discussion.

Regarding the time complexity, the computation of the Smith form $S \in \mathbb{Z}^{n\times n}$ of B along with a Smith massager $M \in \mathbb{Z}^{n\times n}$ dominates the rest of the operations. Let D_B be the partially linearized version of matrix B as specified by Theorem 27. Then, by Theorem 36, we can obtain S and M from the Smith form and a Smith massager for D_B without any extra computation. Therefore, the complexity of step 1 is bounded by the complexity of computing a Smith massager for D_B, which is $O(n^\omega \mathsf{B}(d + \log n)(\log n)^2)$ by Theorem 19.

The probability of the algorithm failing follows from Corollary 20. □

8. A Las Vegas algorithm for Smith form and multipliers

In this section, we combine all of the previous results established so far in order to develop our multiplier algorithm. In particular, we show that there exists a Las Vegas probabilistic algorithm that computes the Smith form $S \in \mathbb{Z}^{n\times n}$ of a nonsingular $A \in \mathbb{Z}^{n\times n}$ along with two unimodular matrices $V, U \in \mathbb{Z}^{n\times n}$ such that

$$AV = US,$$

using $O(n^\omega \mathsf{B}(\log n + \log \|A\|)(\log n)^2)$ bit operations. The algorithm will return the correct output with probability at least $1/4$ or Fail otherwise.

`SmithFormMultipliers`(A)
Input: A nonsingular matrix $A \in \mathbb{Z}^{n\times n}$.
Output: The Smith form $S \in \mathbb{Z}^{n\times n}$ of A and two unimodular matrices $U, V \in \mathbb{Z}^{n\times n}$ such that $AV = US$.
Note: FAIL will be returned with probability less than $3/4$.

1. [Compute the Smith form and a Smith massager for $2A$.]
   (If `SmithMassager` fails, return FAIL)
   $(2S, M) :=$ `SmithMassager`$(2A)$
2. [Perturb the Smith massager M by a random matrix.]
   Pick a uniformly **random** matrix $R \in \mathbb{Z}/(\lambda)^{n\times n}$ for
   $\lambda := 105 \max(n, \left\lceil (\det 2S)^{1/n} \right\rceil)$ as in Theorem 39.
   $B := M + R(2S)$
3. [Certify that B is left equivalent to a matrix H as in (35) and return it.]
   (If `TrivialLowerHermiteForm` fails, return FAIL)
   $H :=$ `TrivialLowerHermiteForm`(B)
   if H is NOTTRIVIAL **then return** FAIL
4. [Compute a unimodular Smith massager.]
   $V := BH^{-1}$
5. [Compute matrix U and return.]
   $U := AVS^{-1}$
   return (S, V, U)

Figure 2: Algorithm `SmithFormMultipliers`

Theorem 49. *Algorithm* `SmithFormMultipliers` *is correct and runs in time*

$$O(n^\omega \mathsf{B}(\log n + \log \|A\|)(\log n)^2).$$

Proof. Step 1 of the algorithm computes the Smith form and a Smith massager for matrix $2A$. From the Smith form of matrix $2A$ we can trivially obtain the Smith form S of A. Furthermore, a Smith massager M for $2A$ is also a Smith massager for A by Lemma 9. Step 1 runs in $O(n^\omega \mathsf{B}(\log n + \log \|A\|)(\log n)^2)$ by Theorem 19, and it will return FAIL with probability at most $1/8$ as stated in Corollary 20.

In step 2, we are perturbing the Smith massager M by a random matrix $R \in \mathbb{Z}^{n\times n}$ multiplied with the Smith form $2S$. By Proposition 8, matrix $B = M + R(2S)$ is also a Smith massager for A, and it is nonsingular. Moreover, by Theorem 39, the last $n-1$ columns of B are linearly independent over $\mathbb{Z}/(p)$ for every prime p with probability greater than $1/2$. As we already mentioned in Section 6, this is equivalent to B being left equivalent to a matrix

$$H = \begin{bmatrix} h_1 & \\ \bar{h} & I_{n-1} \end{bmatrix}, \tag{37}$$

where $h_1 = |\det B|$. The runtime of step 2 is dominated by the claimed complexity.

Algorithm `TrivialLowerHermiteForm` called in step 3 then certifies that B has the desired structure and returns matrix H. The complexity of the subroutine depends on the average length of the columns of B, for which

$$\frac{1}{n}\sum_{j=1}^{n}\text{length}(B_{1..n,j}) \le \frac{1}{n}\left(\log\left(\prod_{j=1}^{n}\|B_{1..n,j}\|\right)+n\right) \le 2.5\log\lambda+1,$$

as per Lemma 44. Since $\lambda \in O(n\|A\|)$, the complexity of step 3 is also $O(n^\omega\,\mathsf{B}(\log n+\log\|A\|)\,(\log n)^2)$.

Algorithm `TrivialLowerHermiteForm` itself might return Fail with probability at most $1/8$. In addition, if it does not fail, the output of the subroutine will be NotTrivial with probability at most $1/2$. This makes the probability of success of Algorithm `SmithFormMultipliers` to be at least $1-(1/8+1/2+1/8)=1/4$ as claimed.

Now, since we know that $B \equiv_L H$, the matrix $V := BH^{-1}$ in step 4 must be integral and unimodular. The evaluation of the product

$$BH^{-1} = B\begin{bmatrix} 1 & \\ -\bar{h} & I_{n-1}\end{bmatrix}\begin{bmatrix} 1/h_1 & \\ & I_{n-1}\end{bmatrix}$$

is covered exactly under Lemma 52 and can be computed, for $d = n(2.5\log\lambda+1)$, in time $O(n^\omega\,\mathsf{M}(\log n+\log\|A\|))$. Furthermore, by Lemma 12, V is a unimodular Smith massager for A.

Finally, by the properties of the Smith massager, matrix $U := AVS^{-1}$ is integral, and unimodular since V is unimodular. By Lemma 53, matrix U can be computed in $O(n^\omega\mathsf{M}(\log n+\log\|A\|))$ bit operations. □

8.1. Sizes of V and U

It will be important to have good bounds on the magnitude of entries in matrices V and U, in order to facilitate the complexity analysis of operations that may use V and U in general.

Lemma 50. *The Smith multiplier matrices $V, U \in \mathbb{Z}^{n\times n}$ returned by Algorithm* `SmithFormMultipliers` *satisfy that:*

(i) $\|V_{1..n,j}\| \le cn\|A\| \cdot \begin{cases} |\det A|+n & \text{if } j=1 \\ s_j & \text{otherwise}\end{cases}$,

(ii) $\|U_{1..n,j}\| \le cn^2\|A\|^2 \cdot \begin{cases} |\det A|+n & \text{if } j=1 \\ 1 & \text{otherwise}\end{cases}$.

for $c = 420$.

Proof. First of all, for $\lambda := 105\max(n, \lceil(\det 2S)^{1/n}\rceil)$, we have, by Hadamard's bound, that $\lambda \le 210n\|A\|$.

By construction, we know that $\|B_{1..n,j}\| \le 2\lambda s_j$ for every $j = 1,\dots,n$. Then, multiplying B with H^{-1} alters only the first column of B. The magnitude of the first column of $V = BH^{-1}$ satisfies that

$$\|V_{1..n,1}\| \le \left(2\lambda h_1\sum_{j=1}^{n}s_j\right)/h_1 \le 2\lambda(|\det A|+n).$$

Furthermore, since $U = AVS^{-1}$, the magnitude of every column of U is bounded by

$$\|U_{1..n,j}\| \le n\|A\|\|V_{1..n,j}\|/s_j.$$

By replacing λ with $210n\|A\|$, the claimed bounds follow. □

Corollary 51. *The average bitlength of the columns of both V and U is bounded by $O(\log n + \log \|A\|)$.*

8.2. Unbalanced multiplication reduced to balanced

The remaining tools needed for our algorithm involves reducing unbalanced matrix multiplications to balanced multiplications. The two lemmas given in this section are used in the proof of Theorem 49. The following lemma is based on Birmpilis et al. (2019, Theorem 20).

Lemma 52. *Let $M \in \mathbb{Z}^{n\times n}$ and $w \in \mathbb{Z}^{n\times 1}$. If $\sum_{j=1}^{n} \operatorname{length}(M_{1..n,j}) \leq d$ and $\operatorname{length}(w) \leq d$ for some $d \in \mathbb{Z}_{\geq 0}$, then the product Mw can be computed in time $O(n^\omega \mathsf{M}(d/n + \log n))$.*

Proof. Choose $X := 2^{\lceil d/n \rceil}$ and let

$$M = M_0 + M_1 X + \cdots + M_{n-1} X^{n-1}$$

$$w = w_0 + w_1 X + \cdots + w_{n-1} X^{n-1}$$

be the X-adic expansions of M and w, respectively. (The coefficients are computed in the symmetric range modulo X.) Our approach is to compute the product

$$\overbrace{\begin{bmatrix} M_0 & M_1 & \cdots & M_{n-1} \end{bmatrix}}^{\bar{M}} \overbrace{\begin{bmatrix} w_0 & w_1 & \cdots & w_{n-1} & & & \\ & w_0 & \cdots & w_{n-2} & w_{n-1} & & \\ & & \ddots & \vdots & \vdots & \ddots & \\ & & & w_0 & w_1 & \cdots & w_{n-1} \end{bmatrix}}^{\bar{W}},$$

from which Mw can be recovered fast. (Notice that the operations to compute the X-adic expansion from a matrix or the matrix from an X-adic expansion take linear time on the number of entries when X is a power of 2.)

Now, the column dimension of $\bar{M}$ and row dimension of $\bar{W}$ is n^2 which is too large to fit within our target complexity. However, because of the assumption that $\sum_{j=1}^{n} \operatorname{length}(M_{1..n,j}) \leq d$ and the fact that $\log(X) = \lceil d/n \rceil$, matrix $\bar{M}$ must contain many zero columns. More specifically, the number of non-zero columns in $\bar{M}$ cannot exceed

$$\sum_{j=1}^{n} \left\lceil \frac{\operatorname{length}(M_{1..n,j})}{\lceil d/n \rceil} \right\rceil \leq \sum_{j=1}^{n} \left(n \frac{\operatorname{length}(M_{1..n,j})}{d} + 1 \right) \leq 2n.$$

Therefore, let $\tilde{M} \in \mathbb{Z}^{n\times 2n}$ be the matrix obtained from $\bar{M}$ by omitting $n^2 - 2n$ zero columns, and let $\tilde{W} \in \mathbb{Z}^{2n\times 2n-1}$ be the matrix obtained from $\bar{W}$ by omitting $n^2 - 2n$ rows corresponding to the columns that were omitted in $\bar{M}$. This transformation reduces the multiplication of $\bar{M}\bar{W}$ to the multiplication of $\tilde{M}\tilde{W}$ which can be done in time $O(n^\omega \mathsf{M}(d/n + \log n))$ since $\log \|\tilde{M}\tilde{W}\| \in O(d/n + \log n)$. □

Moreover, the following lemma uses a similar proof technique and is based on Birmpilis et al. (2020, Lemma 19).

Lemma 53. *Let $A, M \in \mathbb{Z}^{n\times n}$. If $\operatorname{length}(A) \leq d$ and $\sum_{j=1}^{n} \operatorname{length}(M_{1..n,j}) \leq nd$ for some $d \in \mathbb{Z}_{\geq 0}$, then we can compute the product AM in time $O(n^\omega \mathsf{M}(d + \log n))$.*

Proof. Choose $X := 2^d$ and let

$$M = M_0 + M_1 X + \cdots + M_{n-1} X^{n-1}$$

be the X-adic expansion of M. (The coefficients are computed in the symmetric range modulo X.) Our approach is to compute the product

$$A\overbrace{\begin{bmatrix} M_0 & M_1 & \cdots & M_{n-1} \end{bmatrix}}^{\bar{M}},$$

from which AM can be recovered fast. (Notice that the operations to compute the X-adic expansion from a matrix or the matrix from an X-adic expansion take linear time on the number of entries when X is a power of 2.)

Now, the column dimension of $\bar{M}$ is n^2 which is too large to fit within our target complexity. However, because of the assumption that $\sum_{j=1}^{n} \text{length}(M_{1..n,j}) \leq nd$ and the fact that $\log(X) = d$, matrix $\bar{M}$ must contain many zero columns. More specifically, the number of non-zero columns in $\bar{M}$ cannot exceed

$$\sum_{j=1}^{n} \left\lceil \frac{\text{length}(M_{1..n,j})}{d} \right\rceil \leq \sum_{j=1}^{n} \left(\frac{\text{length}(M_{1..n,j})}{d} + 1 \right) \leq 2n.$$

Therefore, let $\tilde{M} \in \mathbb{Z}^{n \times 2n}$ be the matrix obtained from $\bar{M}$ by omitting $n^2 - 2n$ zero columns. This transformation reduces the multiplication of $A\bar{M}$ to the multiplication of $A\tilde{M}$ which can be done in time $O(n^\omega \mathsf{M}(d + \log n))$ since $\log \|A\tilde{M}\| \in O(d + \log n)$. □

Remark 54. *Lemma 53 can be also stated with matrix $A \in \mathbb{Z}^{n \times n}$ replaced by a matrix $U \in \mathbb{Z}^{n \times n}$ that satisfies $\sum_{j=1}^{m} \text{length}(U_{1..n,j}) \leq nd$.*

9. Application: Computing an outer product adjoint formula for A

In this section, we mention an application of the Smith form with the multiplier matrices. Let $A \in \mathbb{Z}^{n \times n}$ be nonsingular and assume that we have precomputed the Smith form S of A, together with unimodular matrices U and V such that $AV = US$.

Let $s := S_{n,n}$ be the largest invariant factor of A. Recall that s is the minimal positive integer that clears the denominators in $A^{-1} \in \mathbb{Q}^{n \times n}$, that is, if entries in A^{-1} are expressed as reduced fractions, then s is the least common multiple of the denominators of the entries. The inverse of A can thus be recovered by computing the integer matrix sA^{-1} and dividing by s. As a tool to compute A^{-1}, Storjohann (2015) developed an algorithm to compute an *outer product adjoint formula* for A: a triple of matrices $(\bar{V}, S, \bar{U})$ such that

$$\text{Rem}(sA^{-1}, s) = \text{Rem}(\bar{V}(sS^{-1})\bar{U}, s).$$

Moreover, $\bar{V} = (\bar{V} \,\mathbf{c}\!\bmod S)$ and $\bar{U} = (\bar{U} \,\mathbf{r}\!\bmod S)$, where $\bar{U} \,\mathbf{r}\!\bmod S$ means reduction of the rows modulo the corresponding diagonal entries of S. While a tight upper bound for the number of bits required to represent $\text{Rem}(sA^{-1}, s)$ explicitly in the worst case is $O(n^3(\log n + \log \|A\|))$, an outer product adjoint formula $(\bar{V}, S, \bar{U})$ requires only $O(n^2(\log n + \log \|A\|))$ bits. Note that $\text{Rem}(sA^{-1}, s)/s$ corresponds to only the fractional part of A^{-1}, that is, if C is the matrix obtained from $\text{Rem}(sA^{-1}, s)$ by reducing entries in the symmetric range modulo s, then $A^{-1} - C/s \in \mathbb{Z}^{n \times n}$ may be nonzero. However, if $\|A^{-1}\| < 1/2$, then C will be identically equal to sA^{-1}.

Example 55. *Matrix*

$$A = \begin{bmatrix} -6 & 3 & -13 & -15 \\ -4 & 19 & 12 & -1 \\ -4 & 10 & -6 & 17 \\ -26 & -13 & 1 & -2 \end{bmatrix}$$

has Smith form $S := \mathrm{Diag}(s_1, s_2, s_3, s_4) = \mathrm{Diag}(1, 1, 9, 29088)$ *and*

$$s_4 A^{-1} = \begin{bmatrix} -271 & -402 & -373 & -937 \\ 580 & 920 & 524 & -356 \\ -1074 & 804 & -870 & 258 \\ -784 & -352 & 1008 & 80 \end{bmatrix}.$$

An outer product adjoint formula for A is given by $(\bar{V}, S, \bar{U})$ *where*

$$\bar{V} = \begin{bmatrix} 0 & 0 & 7 & 805 \\ 0 & 0 & 5 & 23668 \\ 0 & 0 & 3 & 6 \\ 0 & 0 & 4 & 10224 \end{bmatrix} \text{ and } \bar{U} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 2 & 2 & 0 & 2 \\ 20829 & 1750 & 28943 & 16203 \end{bmatrix}.$$

For this particular A, which satisfies $\|A^{-1}\| < 1/2$*, multiplying out* $\bar{V}(s_4 S^{-1})\bar{U}$ *and reducing entries in the symmetric range modulo* s_4 *gives* $s_4 A^{-1}$*. Because* $s_1 = s_2 = 1$ *the first two columns of V and first two rows of U can be omitted, giving*

$$\begin{bmatrix} 7 & 805 \\ 5 & 23668 \\ 3 & 6 \\ 4 & 10224 \end{bmatrix} \begin{bmatrix} 3232 & \\ & 1 \end{bmatrix} \begin{bmatrix} 2 & 2 & 0 & 2 \\ 20829 & 1750 & 28943 & 16203 \end{bmatrix} \equiv s_4 A^{-1} \bmod s_4.$$

There is a direct relationship between an outer product adjoint formula and the unimodular Smith multipliers U and V.

Lemma 56. *Let* $U, V \in \mathbb{Z}^{n \times n}$ *be unimodular matrices such that* $AV = US$*. Then, the triple* $(V \bmod^{\mathbf{c}} S, S, U^{-1} \bmod^{\mathbf{r}} S)$ *gives an outer product adjoint formula for A.*

Proof. We have that $sA^{-1} = V(sS^{-1})U^{-1}$. Furthermore, $V(sS^{-1}) = (V \bmod^{\mathbf{c}} S)(sS^{-1}) \bmod s$ and $(sS^{-1})U^{-1} = (sS^{-1})(U^{-1} \bmod^{\mathbf{r}} S) \bmod s$, and so

$$\mathrm{Rem}(sA^{-1}, s) = \mathrm{Rem}((V \bmod^{\mathbf{c}} S)(sS^{-1})(U^{-1} \bmod^{\mathbf{r}} S), s).$$

□

Storjohann (2015) gives a randomized algorithm to compute an outer product adjoint formula in

$$O(n^2(\log n)\mathsf{B}(n(\log n + \log \|A\|)) \tag{38}$$

plus

$$O(n^3 \max(\log n, \log \|A^{-1}\|)\, \mathsf{B}(\log n + \log \|A\|)) \tag{39}$$

bit operations. Note that (38) implies that fast (pseudo-linear) integer arithmetic needs to be used to achieve a cost that is softly cubic in n, while (39) reveals a sensitivity to $\|A^{-1}\|$. Indeed, we

may have $\log \|A^{-1}\| \in \Omega(n(\log n + \log \|A\|))$, in which case the upper bound in (39) becomes quartic in n. It was left as an open question if an outer product adjoint formula can be computed in time $(n^\omega \log \|A\|)^{1+o(1)}$ bit operations. Here, we can resolve this question by using the approach of Lemma 56.

Theorem 57. *Assume we have the output (S, V, U) of Algorithm* `SmithFormMultipliers`(A). *Then, an outer product adjoint formula for A can be computed in time $O(n^\omega \,\mathsf{M}(\log n + \log \|A\|) \log n)$.*

Proof. First compute $\bar{V} := V \bmod S$. This can be done in time $O(n \sum_{i=1}^{n} \mathsf{M}(\text{length}(V_{1\ldots n,i}))$. By Corollary 51, $\sum_{i=1}^{n} \text{length}(V_{1\ldots n,i}) \in O(n(\log n + \log \|A\|))$, which shows that the matrix $\bar{V}$ can be computed in time $O(n\,\mathsf{M}(n(\log n + \log \|A\|))$.

It remains to compute $\bar{U} := U^{-1} \bmod S$. Let $D \in \mathbb{Z}^{m \times m}$ be the partial column linearization of U as in Theorem 27. It will be that $m \in O(n)$, and again by Corollary 51, $\log \|D\| \in O(\log n + \log \|A\|)$. Therefore, by Lemma 15, we can compute a straight line formula for D^{-1} in time $O(n^\omega \,\mathsf{M}(\log n + \log \|A\|) \log n)$. The formula consists of $O(\log n)$ integer matrices of dimension m and bitlength bounded by $O(\log n + \log \|A\|)$.

Finally, we can compute $U^{-1} \bmod S$ by evaluating $D^{-1} \bmod \text{diag}(S, I_{m-n})$ using the straight line formula. The evaluation of the formula requires $O(\log n)$ matrix multiplications where the first operand is an $m \times m$ integer matrix reduced $\bmod \text{diag}(S, I)$ and the second operand is an $m \times m$ integer matrix with bitlength bounded by $O(\log n + \log \|A\|)$. This type of matrix multiplication falls exactly under Lemma 53 by simply transposing the operation. Therefore, we can compute $U^{-1} \bmod S$ in time $O(n^\omega \,\mathsf{M}(\log n + \log \|A\|) \log n)$. □

An application of the outer product adjoint formula is to compute the proper fractional part of a linear system solution. Let $b \in \mathbb{Z}^{n \times 1}$ satisfy $\log \|b\| \in (n \log \|A\|)^{1+o(1)}$. Then

$$A^{-1}b = \overbrace{A^{-1}b - \text{Rem}(sA^{-1}b, s)/s}^{\in \mathbb{Z}^n} + \text{Rem}(sA^{-1}b, s)/s,$$

where $\text{Rem}(sA^{-1}b, s)/s$ is a vector of proper fractions. By Lemma 17, $A^{-1}b \in \mathbb{Q}^{n \times 1}$ can be computed in a Las Vegas fashion in $(n^\omega \log \|A\|)^{1+o(1)}$ bit operations, or $(n^3 \log \|A\|)^{1+o(1)}$ bit operations if $\omega = 3$. If an outer product adjoint formula for A is known, then the proper fractional part of $A^{-1}b$ can be computed in only $(n^2 \log \|A\|)^{1+o(1)}$ bit operations. The following result is a corollary of (Storjohann, 2015, Lemma 4.11).

Lemma 58. *Assume we have an outer product adjoint formula $(\bar{V}, S, \bar{U})$ for a nonsingular $A \in \mathbb{Z}^{n \times n}$, and let $s = S_{n,n}$. Given a vector $b \in \mathbb{Z}^{n \times 1}$ with $\log \|b\| \in O(\log s)$, we can compute $\text{Rem}(sA^{-1}b, s)$ in time $O(n\,\mathsf{M}(\log s))$.*

Example 59. *Let $A \in \mathbb{Z}^{n \times n}$ be the matrix of Example 55 and*

$$b = \begin{bmatrix} 25 \\ 94 \\ 12 \\ -2 \end{bmatrix}.$$

Then

$$\bar{V}(29088S^{-1})\bar{U}b \equiv \begin{bmatrix} 11011 \\ 20716 \\ 8682 \\ 17424 \end{bmatrix} \bmod 29088.$$

Indeed, we have

$$A^{-1}b = \begin{bmatrix} -2 \\ 3 \\ 1 \\ -2 \end{bmatrix} + \begin{bmatrix} 11011 \\ 20716 \\ 8682 \\ 17424 \end{bmatrix} \frac{1}{29088}.$$

Applying Lemma 58 with $b = I_n$ gives the following corollary of Theorems 49 and 57.

Corollary 60. *Given a nonsingular integer input matrix $A \in \mathbb{Z}^{n\times n}$, the largest invariant factor s of A, together with* $\mathrm{Rem}(sA^{-1}, s)$*, can be computed in a Las Vegas fashion in*

$$O(n^{\omega}\,\mathsf{B}(\log n + \log \|A\|)(\log n)^2 + n^2\,\mathsf{M}(\log s))$$

bit operations. This is bounded by $(n^3 \log \|A\|)^{1+o(1)}$ bit operations.

10. Conclusion and topics for future research

In this paper we have presented a new, Las Vegas probabilistic algorithm for determining the unimodular Smith multipliers for a nonsingular integer matrix. Combining this with our previous results in (Birmpilis et al., 2020), implies that we can determine the Smith form and a pair of unimodular multipliers in time $(n^{\omega} \log \|A\|)^{1+o(1)}$, approximately about same number of bit operations as required to multiply two matrices of the same dimension and size of entries as the input matrix. We have also given explicit bounds on the sizes of our multipliers and made use of such bounds to efficiently determine an outer adjoint formula for an integer matrix. We also include computational tools and partial linearization sections which should be of independent interest.

In terms of future directions, a natural direction is to find a *deterministic* algorithm for both the Smith form and the Smith form with multipliers problems. In the case of integer matrices we have already seen that linear system solving can be derandomized within the desired cost. An easier problem than to derandomize Smith form computation would be to first find a deterministic algorithm for finding only the largest invariant factor s_n, a problem that has a solution in the case of polynomial matrices (Zhou et al., 2014).

Another problem which arises naturally is that of finding algorithms for the computation of other integer matrix forms, in particular the Hermite normal form, with the target complexity being the number of bit operations required to multiply two matrices of the same dimension and size of entries as the input matrix. We expect that our primary tool, the Smith massager can also play an important intermediate role here.

Finally, our algorithms and tools all assume that the input matrix is nonsingular, unlike for example the procedure from Kaltofen and Villard (2005). It is of interest to extend the present work to singular integer matrices, likely through compression techniques to reduce the problem to a smaller nonsingular matrix.

11. Acknowledgements

We would like to thank the anonymous referees for their suggestions on making the paper more readable. This research was partly supported by the Natural Sciences and Engineering Research Council (NSERC) Canada.

References

M. Agrawal, N. Kayal, and N. Saxena. PRIMES is in P. *Annals of Mathematics*, 160:781–793, 2004.

A. V. Aho, J. E. Hopcroft, and J. D. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, 1974.

J. Alman and V. V. Williams. A refined laser method and faster matrix multiplication. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 522–539, 2021. doi: 10.1137/1.9781611976465.32.

S. Birmpilis, G. Labahn, and A. Storjohann. Deterministic reduction of integer nonsingular linear system solving to matrix multiplication. In *Proc. Int'l. Symp. on Symbolic and Algebraic Computation: ISSAC'19*, page 58–65, New York, NY, USA, 2019. ACM. ISBN 9781450360845. doi: 10.1145/3326229.3326263.

S. Birmpilis, G. Labahn, and A. Storjohann. A Las Vegas algorithm for computing the Smith form of a nonsingular integer matrix. In *Proc. Int'l. Symp. on Symbolic and Algebraic Computation: ISSAC'20*, page 38–45, New York, NY, USA, 2020. ACM. ISBN 9781450371001. doi: 10.1145/3373207.3404022.

G. H. Bradley. Algorithm and bound for the greatest common divisor of *n* integers. *Communications of the ACM*, 13(7): 433–436, July 1970.

H. Cohen. *A Course in Computational Algebraic Number Theory*. Springer-Verlag, 1996.

W. Eberly, M. Giesbrecht, and G. Villard. Computing the determinant and Smith form of an integer matrix. In *Proc. 31st Ann. IEEE Symp. Foundations of Computer Science*, pages 675–685, 2000.

J.-C. Faugère and J. Svartz. Gröbner bases of ideals invariant under a commutative group: The non-modular case. In *Proc. Int'l. Symp. on Symbolic and Algebraic Computation: ISSAC'13*, pages 347–354. ACM Press, New York, 2013.

J. von zur Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge University Press, 3rd edition, 2013.

K. O. Geddes, S. R. Czapor, and G. Labahn. *Algorithms for Computer Algebra*. Kluwer, Boston, MA, 1992.

M. Giesbrecht. Fast computation of the Smith form of a sparse integer matrix. *Computational Complexity*, 10(1):41–69, 11 2001.

S. Gupta, S. Sarkar, A. Storjohann, and J. Valeriote. Triangular x-basis decompositions and derandomization of linear algebra algorithms over $\mathsf{K}[x]$. *Journal of Symbolic Computation*, 47(4), 2012. doi: 10.1016/j.jsc.2011.09.006. Festschrift for the 60th Birthday of Joachim von zur Gathen.

E. Hubert and G. Labahn. Computation of invariants of finite abelian groups. *Mathematics of Computation*, 85:3029–3050, 2016.

E. Kaltofen and G. Villard. On the complexity of computing determinants. *Computational Complexity*, 13(3–4):91–130, 2005.

R. Kannan and A. Bachem. Polynomial algorithms for computing the Smith and Hermite normal forms of an integer matrix. *SIAM Journal of Computing*, 8(4):499–507, November 1979.

F. Le Gall and F. Urrutia. Improved rectangular matrix multiplication using powers of the Coppersmith-Winograd tensor. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 1029–1046, 2018. doi: 10.1137/1.9781611975031.67.

J. N. Lyness and P. Keast. Application of the Smith Normal Form to the structure of lattice rules. *SIAM J. Matrix Anal. Appl.*, 16(1):218–231, 1995.

M. Newman. *Integral Matrices*. Academic Press, 1972.

C. Pauderis and A. Storjohann. Deterministic unimodularity certification. In *Proc. Int'l. Symp. on Symbolic and Algebraic Computation: ISSAC'12*, page 281–288. ACM Press, New York, 2012. ISBN 9781450312691. doi: 10.1145/2442829.2442870.

H. J. S. Smith. On systems of linear indeterminate equations and congruences. *Phil. Trans. Roy. Soc. London*, 151: 293–326, 1861.

R. Stanley. Smith normal form in combinatorics. *Journal of Combinatorial Theory, Series A*, pages 476–495, 2016.

A. Storjohann. *Algorithms for Matrix Canonical Forms*. PhD thesis, Swiss Federal Institute of Technology, ETH–Zurich, 2000.

A. Storjohann. The shifted number system for fast linear algebra on integer matrices. *Journal of Complexity*, 21(4): 609–650, 2005. Festschrift for the 70th Birthday of Arnold Schönhage.

A. Storjohann. On the complexity of inverting integer and polynomial matrices. *Computational Complexity*, 24:777–821, 2015. doi: http://dx.doi.org/10.1007/s00037-015-0106-7.

W. Zhou, G. Labahn, and A. Storjohann. A deterministic algorithm for inverting a polynomial matrix. *Journal of Complexity*, 2014.