Certifying Inconsistency of Sparse Linear Systems

M. Giesbrecht*
Dept. of Computer Science
University of Manitoba
Winnipeg, Manitoba
R3T 2N2, Canada
email: mwg@cs.umanitoba.ca

A. Lobo
Dept. of Mathematics and Computer Science
Washington College, 300 Washington Ave.
Chestertown, MD, 21620, USA
Email: Austin.Lobo@washcoll.edu

B. D. Saunders†
Dept. of Computer and Information Sciences
University of Delaware
Newark, DE 19716, USA
Email: saunders@cis.udel.edu

Randomized black box algorithms provide a very efficient means for solving sparse linear systems over arbitrary fields. However, when these probabilistic algorithms fail, it is not revealed whether no solution exists or whether the algorithm simply made unlucky random choices. Here we give an efficient algorithm to compute a certificate of inconsistency for a black box linear system over a field. Our method requires a black box for the transpose of the matrix. The cost of producing the certificate is shown to be about the same as that of solving the system in the black box model, while the cost of applying a given certificate to prove inconsistency is much smaller. We also give an efficient algorithm for certifying that a sparse Diophantine linear system of integer equations has no integer solutions, even when it may have rational solutions.

1 Introduction

Given a sparse matrix $A \in \mathsf{F}^{n\times n}$ over some ring F and a vector $b \in \mathsf{F}^{n\times 1}$, it is a fundamental problem to solve the linear system $Ax = b$ for $x \in \mathsf{F}^{1\times n}$. The system has a unique solution, many solutions, or no solutions. That all three cases in this tautology can occur (and that we know the structure of the solution set) is of course the core of elementary linear algebra.

Fast probabilistic algorithms which exploit sparsity in A have been developed to solve the problem of finding solutions x if one exists (over fields by Wiedemann 1986, Kaltofen & Saunders 1991, Coppersmith 1993, Coppersmith 1994, Kaltofen 1995, Lambert 1996, Villard 1997, and over the integers by Giesbrecht 1997). Furthermore a purported solution is easily checked, so that these algorithms are of Las Vegas type when the hypothesis is made a priori that the system is consistent. However, less attention has been paid to the case when no solutions exists, and these algorithms do not certify this case. There is no way to determine whether a failure to find a solution was due to inconsistency or simply to unlucky random choices.

Over a field F, we show how to compute a vector $u \in \mathsf{F}^{1\times n}$ such that $uA = 0$ and $ub \neq 0$, if no solution to $Ay = b$ exists. This u clearly demonstrates that $Ax = b$ has no solution $x \in \mathsf{F}^{n\times 1}$ since $Ax = b$ implies $uAx = ub$. Intuitively such certificates are dense in the left nullspace of A. If b is not in the column span of A, then b is orthogonal to at most a proper subspace of the left nullspace of A. Thus, with high probability, a random element of this left nullspace is a certificate.

The cost of our algorithm is measured in terms of black-box vector-times-matrix evaluations $w \mapsto wA$, for $w \in \mathsf{F}^{1\times n}$. The algorithm requires an expected $O(r)$ such black-box evaluations plus an additional $O\tilde{}(r\,\mathrm{M}(n))$ operations in F, where r is the rank of A and $O(\mathrm{M}(n))$ operations in F are sufficient to multiply two polynomials of degree n over F ($\mathrm{M}(n) = n\log n\log\log n$ using FFT-based polynomial arithmetic). Additional space for $O(n)$ values from F is required. This is approximately the cost of solving a consistent system having the same black-box matrix, using any of the known algorithms for solving sparse systems over a field. It should be noted that the black-box evaluations used here compute vector-times-matrix products rather than the matrix-times-vector products used in Wiedemann's algorithm. This does represent a weakening of the black-box model, both theoretically and practically (for a complete solver we must have efficient algorithms for both matrix-vector and vector-matrix products, and a potentially substantial extra cost in memory usage). We do note that algorithms such as Lanczos, which require a symmetric input matrix A, generally already require preconditioning involving the transpose black-box to solve non-symmetric systems; see, e.g., Eberly & Kaltofen 1997.

We extend our techniques to provide certificates of inconsistency for sparse Diophantine systems $Ax = b$, where $A \in \mathbb{Z}^{n\times n}$, $b \in \mathbb{Z}^{n\times 1}$ and where *integer* solutions $x \in \mathbb{Z}^{n\times 1}$ are sought. The interesting case here is when rational solutions to the system *do* exist. In this case, a certificate of Diophantine inconsistency is a vector $u \in \mathbb{Z}^{1\times n}$ and an integer d such that $uA \equiv 0 \bmod d$ while $ub \not\equiv 0 \bmod d$. The

*Research was supported in part by Natural Sciences and Engineering Research Council of Canada research grant OGP0155376 and University of Manitoba research grant 431-1725-80.

†Research supported by National Science Foundation grant CCR-9712362.

integer is generally a small factor or multiple of the largest determinantal divisor of A. A certificate can be found with about the same cost as that of solving a consistent rational system having the same black-box matrix, using, say, Wiedemann's algorithm. Again, the cost of using a certificate to prove inconsistency is much smaller.

2 Certifying inconsistency over a field

Let F be a field, $A \in \mathsf{F}^{n\times n}$ and $b \in \mathsf{F}^{n\times 1}$, and the linear system $Ax = b$ to be solved for $x \in \mathsf{F}^{n\times 1}$. We offer an algorithm to solve the system in the sense that either (1) a solution x which is a random sample of the solution space in a suitable sense is returned or (2) a certificate of the fact that no solution exists is given. The latter is the new feature offered in this paer. The main idea is that a random element of the left nullspace of A will serve as the certificate. Hence, choosing a random solution to a linear system in a suitable sense is involved in both cases (1) and (2). If the field F is finite and it is feasible to select uniformly at random from all of F, then we can have a uniformly random element of the nullspace of A as in Theorem 4 of Kaltofen & Saunders 1991. Here we also work out more specifically than in Kaltofen & Saunders 1991 the details of random sampling when uniform selection from a subset of F is the available tool.

We begin with the observation that drives the inconsistency certification.

THEOREM 2.1. *Let $A \in \mathsf{F}^{n\times n}$ and $b \in \mathsf{F}^{n\times 1}$. There is no $x \in \mathsf{F}^{n\times 1}$ such that $Ax = b$ if and only if there exists a $u \in \mathsf{F}^{1\times n}$ such that $uA = (0,\ldots,0) \in \mathsf{F}^{1\times n}$ and $ub \neq 0$.*

PROOF. Assume there is no $x \in \mathsf{F}^{n\times 1}$ such that $Ax = b$. Then $\mathrm{rank}[A|b] = \mathrm{rank}[A] + 1$. Thus the dimension of the left nullspace of $[A|b]$ is one less than the dimension of the left nullspace of A and there exists a $u \in \mathsf{F}^{1\times n}$ in the left nullspace of A which is not in the left nullspace of $[A|b]$. For this u, $uA = (0,\ldots,0) \in \mathsf{F}^{1\times n}$ and $ub \neq 0$.

Conversely, assume there exists a u with the described properties and an $x \in \mathsf{F}^{n\times 1}$ with $Ax = b$. Then $0 = uAx = ub \neq 0$, a contradiction. □

The algorithm to follow will generate efficiently a vector which is a random linear combination of a certain spanning set of the nullspace of A, however, this spanning set will not itself be explicitly constructed.

The next theorem shows that a vector so generated is not likely to be orthogonal to b, i.e., not likely to be in the left nullspace of $[A|b]$, a hyperplane in the left nullspace of A.

THEOREM 2.2. *Let $A \in \mathsf{F}^{n\times n}$ and suppose $v_1,\ldots,v_s \in \mathsf{F}^{1\times s}$ span N_A, the left nullspace of A. Let $\mathcal{L}$ be a finite subset of F and $\delta_1,\ldots,\delta_s$ uniformly and randomly chosen from $\mathcal{L}$. Let H be a hyperplane in N_A. Then*

$$\mathrm{Prob}\left\{\left(\sum_{1\le i\le s}\delta_i v_i\right) \notin H\right\} \ge 1 - 1/\#\mathcal{L}.$$

PROOF. The hyperplane H extends to a hyperplane H' of $\mathsf{F}^{1\times n}$ such that $H = H' \cap N_A$. Let $b \in \mathsf{F}^{n\times 1}$ define H', i.e., $H' = \{v : v\cdot b = 0\}$. Assume $v_i = (v_{i1},\ldots,v_{is}) \in \mathsf{F}^{1\times n}$ and $b = (b_1,\ldots,b_n)^t \in \mathsf{F}^{n\times 1}$. Let $z_1,\ldots,z_s$ be indeterminates and let

$$\begin{aligned} f(z_1,\ldots,z_s) &= \sum_{1\le i\le s} z_i v_i \cdot b \\ &= \sum_{1\le i\le s} z_i \sum_{1\le j\le n} v_{ij} b_j \in \mathsf{F}[z_1,\ldots,z_s]. \end{aligned}$$

The polynomial f is either identically zero or has degree 1. However, by Theorem 2.1 there exists a vector $u = \sum_{1\le i\le s} u_i v_i$ such that $f(u_1,\ldots,u_s) \neq 0$. By the Schwartz (1980)/Zippel (1979) Lemma, $\mathrm{Prob}\{f(\delta_1,\ldots,\delta_s) \neq 0\} \ge 1 - 1/\#\mathcal{L}$. for $\delta_1,\ldots,\delta_s$ chosen randomly and uniformly from $\mathcal{L}$. □

We next give the algorithm of Kaltofen & Saunders (1991) to produce a random solution to a singular linear system.

Algorithm: `RandomSol`
Input: – $A \in \mathsf{F}^{n\times n}$;
– $b \in \mathsf{F}^{n\times 1}$;
– $f(z)$, a polynomial in $\mathsf{F}[z]$ such that $f(0) \neq 0$;
– $\mathcal{L}$, a subset of F.
This algorithm is meant to be called with arguments for which you have reason to believe that $r := \deg(f) = \mathrm{rank}(A)$ and that the leading $r\times r$ principal minor of A has minimum polynomial f.
Output: – "False", meaning no solution was obtained or ("True", $x \in \mathsf{F}^{n\times 1}$) such that
(i) $Ax = b$, and
(ii) x is a random sample of the solution space to $Ay = b$ as in Theorem 2.3;

(1) Choose $w = (w_1,\ldots,w_n)^t$, with entries chosen randomly from $\mathcal{L}$;
Let $r := \deg(f)$;
(2) Let b' be the vector consisting of the first r entries of $b + Aw$;
Let A_r be the leading $r\times r$ submatrix of A;
// The black box for A_r consists of padding the vector
// with zeros, applying the black box for A and
// considering only the first r entries of the output;
Let $x = -\sum_{i=1}^{n} f_i/f_0 A_r^i b'$;
(3) If $Ax = b$ then return ("True", x) else return "False".

THEOREM 2.3.
(a) If RandomSol returns ("True",x), then x is a correct solution to $Ax = b$.
(b) If the inputs $A, b, f, \mathcal{L}$ to RandomSol satisfy $r := \deg(f) = \mathrm{rank}(A)$, $f = \mathrm{minpoly}(A_r)$, where A_r is the leading principal $r\times r$ sumbatrix of A, then the ("True",x) output is returned and for any hyperplane H in $\{y : Ay = b\}$ the probability that $x \in H$ is less than $1/\#\mathcal{L}$.
(c) The algorithm requires $O(r)$ evaluations of the black box for A, $O(nr)$ additional operations in F, and space for $O(n)$ elements of F.

PROOF. The proof is a reasonably straightforward embellishment of the proofs of Lemma 4 and Theorem 4 in Kaltofen & Saunders (1991). □

Algorithm: `CompleteSparseLinearSystemSolver`

Input:
- a black box, $x \mapsto Ax$ for $A \in \mathsf{F}^{n\times n}$;
- a black box for A^t, $u \mapsto uA$;
- $b \in \mathsf{F}^{n\times 1}$;
- $\mathcal{L}$, a subset of F containing more than $2n(n-1)$ elements, from which to choose at random;

Output: Accordingly as the system $Ay = b$ is nonsingular, singular and consistent, or singular and inconsistent, respectively, we get:
("nonsingular", x) with $x = A^{-1}b$, or
("singular-consistent", x) such that x is a random element of the solution space to $Ay = b$, or
("singular-inconsistent", u) such that $u^tA = 0$ and $u^tb \neq 0$, certifying the inconsistency.

(1) *// Try the non-singular case.*
Apply Wiedemann's algorithm to $A, b, \mathcal{L}$. It returns either a solution $x \in \mathsf{F}^{n\times 1}$ or a factor $\hat{f}(z)$ of minpoly$(A) \in \mathsf{F}[z]$ which is divisible by z.
If the former case,
then return ("nonsingular", x);
else continue to step 2;
// alternatively one could try steps 3 and 4 once on A
// without preconditioning. This alternative could
// save considerable time, however we make no claim
// about the randomness of a solution obtained in step 4
// if this is done. For this alternative:
// set $f := \hat{f}/z$;
// If z divides $f(z)$
// then go to step 2 anyway;
// else {set $B := A; c := b$; go to step 3;}

The matrix is singular.
Repeat steps 2–4 until done.

(2) *// Random preconditioning and minpoly.*
Choose random $\alpha_2, \ldots, \alpha_n, \beta_2, \ldots, \beta_n$, and $\gamma_1, \ldots, \gamma_n$ from $\mathcal{L}$ and construct black boxes for

$$U = \begin{pmatrix} 1 & \alpha_2 & \alpha_3 & \cdots & \alpha_n \\ & 1 & \alpha_2 & \ddots & \vdots \\ & & \ddots & \ddots & \vdots \\ & & & 1 & \alpha_2 \\ & & & & 1 \end{pmatrix}$$

$$L = \begin{pmatrix} 1 & & & & \\ \beta_2 & 1 & & & \\ \beta_3 & \beta_2 & \ddots & & \\ \vdots & \ddots & \ddots & \ddots & \\ \beta_n & \cdots & \beta_3 & \beta_2 & 1 \end{pmatrix}$$

and $B = UAL$.
// RandomSol can also use the black box for B to
// obtain a black box for B', the leading
// $r \times r$ submatrix of B.
Compute $\hat{f}(z) :=$ minpoly$(B) \in \mathsf{F}[z]$ with Wiedemann's algorithm;
If z divides $f(z) := \hat{f}(z)/z$,
then repeat step (4)
else set r:= deg(f); c := Ub; continue to step (3).

(3) *// Try inconsistency.*
Call RandomSol with $B^t, 0, f, \mathcal{L}$.
if it returns ("True", u) and $u^tc \neq 0$
then return ("singular-inconsistent", u);
else continue to step (4);

(4) *// Try solving.*
Call RandomSol with $B, c, f, \mathcal{L}$.
if RandomSol returns ("True", x)
then return ("singular-consistent", x);
else go to step (2);

Remarks.

Kaltofen & Saunders's algorithm (RandomSol applied after preconditioning) requires that the field F contain at least $2n(n-1)$ elements. If this is not the case, simply work in an extension field K of F containing sufficiently many elements (in particular, we can choose K so that $[\mathsf{K} : \mathsf{F}] = \lceil \log_q(2n(n-1)) \rceil$, where $q = \#\mathsf{F}$). Since if a solution exists, it lies in F (independent of any field in which F is embedded), the produced $u \in \mathsf{K}^{1\times n}$ will be a valid certificate.

One could possibly succeed in finding a solution to a consistent singular system by doing calls to RandomSol as in step 4, but doing them repeatedly with the same matrix, without fresh preconditioning, until (possible) success. However we don't see a reasonable claim to be made about the randomness of the solution in the solution space if this is done.

On each iteration, steps 3 and 4 can be done in either order. In the absence of prior knowledge of the inputs we choose the given order. For a random singular system, inconsistency is more likely than consistency.

THEOREM 2.4.

(a) Algorithm CompleteSparseLinearSystemSolver is correct.

(b) If H is a proper subspace of the solution space to $Ay = b$, and the algorithm returns ("singular-consistent", x), then $\text{Prob}\{x \notin H\} \geq (1 - (2n(n-1)/\#\mathcal{L})(1 - 1/\#\mathcal{L})$.

(c) The algorithm requires $O(n \cdot \beta)$ evaluations of the black boxes for A, A^t, an additional $O(n\,\mathrm{M}(n) \cdot \beta)$ operations in F, and space for $O(n \cdot \beta)$ elements of F. Here $\beta = \lceil \log_q(2n(n-1)) \rceil$ which is 1 if F has more than $2n(n-1)$ elements. Also take $\beta = 1$ if F is infinite. The expected value of the number of iterations of steps 2,3,4 is $O(1)$.

The expected number of iterations of steps 2,3,4 is $1/p$ if the probability of success in one iteration is p. Suppose one knows only that the probability of success in computing the minimum polynomial in step 2 is at least $1/2$, that for an inconsistent system the conditional probability is at least $1/2$ of success in step 3 given the correct minimum polynomial, and similarly that for a consistent system the conditional probability of success in step 4 is at least $1/2$. Then the probability of success in a single iteration is at least $1/4$ and the expected number of iterations is no more than 4. Often one can drive down the probability of failure in one iteration by choosing a large set $\mathcal{L}$ from which the random choices are made, thus bringing the expected number of iterations arbitrarily close to 1.

3 Certifying inconsistency over $\mathbb{Z}$

We now consider the problem of certifying inconsistency in the case when $A \in \mathbb{Z}^{n\times n}$. We are careful to take into account

expression swell and count bit (machine-word) operations in our analyses. For $X \in \mathbb{Z}^{m\times n}$, define $\|X\| = \|X\|_\Delta = \max\{X_{ij}\}$, the maximum absolute value in the matrix or vector. For a polynomial $g = \sum_{0\le i\le n} a_i z^i \in \mathbb{Z}[z]$, define $\|g\| = \max_i |a_i|$.

In Giesbrecht (1997) it is shown how to find solutions to such systems in the black box model.

FACT 3.1 *(Giesbrecht 1997). Let $A \in \mathbb{Z}^{n\times n}$ with rank r and $b \in \mathbb{Z}^{n\times 1}$, and assume a solution $x \in \mathbb{Z}^{n\times 1}$ to $Ax = b$ exists. Let $\varrho = r\log\|A\| + \log\|b\|$.*

- *We can find a $x \in \mathbb{Z}^{n\times 1}$ such that $Ax = b$ with an expected number of $O\tilde{}(r\varrho)$ matrix-vector products by A modulo primes with $O(\log n + \log\log(\|A\| + \|b\|))$ bits.*
- *The output x satisfies $\log\|x\| = O\tilde{}(r\log n + r\log\|A\| + \log\|b\|)$.*
- *An additional $O\tilde{}(r^2 + rn\varrho + n\,\mathrm{M}(\varrho))$ bit operations and additional storage for $O\tilde{}(n\varrho)$ words are required.*

The algorithm is probabilistic: solutions produced are guaranteed correct and on any invocation the algorithm produces a solution with probability at least $1/2$.

We use soft-Oh notation here to hide some rather messy logarithmic factors incurred from homomorphic imaging: For functions $f, g : \mathbb{R}^k_{>0} \to \mathbb{R}$ we have the equivalence $g = O\tilde{}(f) \iff g = O(f\cdot\log(f)^k)$ for some $k \ge 0$.

By way of comparison, if the input matrix has $O(n^{1+\epsilon})$ non-zero entries (for some $\epsilon > 0$), this algorithm requires $O\tilde{}(nr^{2+\epsilon}\log^2(\|A\| + \|b\|))$ bit operations, vs. $O\tilde{}(n^3 r\log^2\|A\|)$ bit operations for Gaussian elimination. Like Wiedemann's algorithm, if a Diophantine solution does not exist, we cannot distinguish this from unlucky random choices. Our goal in this paper is to remove this possibility of error with about the same cost.

The principle new idea required here is that if no solution exists over the integers, then no solution exists modulo d, for some integer $d \in \mathbb{Z}$ (for example, whenever d is a multiple of the rth determinantal divisor of A). Clearly, if there exists a $d \in \mathbb{Z}$ and a $u \in \mathbb{Z}^{1\times n}$ such that $uA \equiv 0 \bmod d$ and $ub \not\equiv 0 \bmod d$, then the system has no Diophantine solution $x \in \mathbb{Z}^{n\times 1}$ (otherwise $0 \equiv uAx \equiv ub \not\equiv 0 \bmod d$). The existence of such certificates is established by the following Lemma.

LEMMA 3.2. *Let $A \in \mathbb{Z}^{n\times n}$ and $b \in \mathbb{Z}^{n\times 1}$ be such that there exists no $x \in \mathbb{Z}^{n\times 1}$ with $Ax = b$. Assume A has rank r and rth determinantal divisor $\nabla_r \ne 0$. There exists a prime p such that there is no solution $x \in \mathbb{Z}^{n\times 1}$ with $Ax \equiv b \bmod p^e$, where $e \ge \mathrm{ord}_p(\nabla_r)$. As well, there exists a $u \in \mathbb{Z}^{1\times n}$ with $uA \equiv 0 \bmod p^e$ and $ub \not\equiv 0 \bmod p^e$.*

PROOF. By a unimodular change of basis we may assume that A is in Smith form. Since no solution $x \in \mathbb{Z}^{n\times 1}$ to $Ax = b$ exists, there exists a prime p and $s \le e$ such that $p^s \mid A_{ii}$ and $p^s \nmid b_i$. Clearly there can be no $x \in \mathbb{Z}^{n\times 1}$ such that $Ax \equiv b \bmod p^e$. Moreover, if $u \in \mathbb{Z}^{1\times n}$ is zero except for a p^{e-s} in the ith location, then $uA \equiv 0 \bmod p^e$ and $ub \not\equiv 0 \bmod p^e$. □

We will assume here that $Ax = b$ has rational solutions $x \in \mathbb{Q}^{n\times 1}$ such that $Ax = b$. If this is not the case, inconsistency can be certified by the techniques of the previous section. In fact, for a random single-word prime q, if $Ax = b$ has no rational solution then $Ax \equiv b \bmod q$ has no solution in $\mathbb{Z}_q$ with high probability and we can work very efficiently in $\mathbb{Z}_q$. When rational solutions do exist, the certificates $u \in \mathbb{Z}^{1\times n}$ we are looking for (such that $uA \equiv 0 \bmod d$ and $ub \not\equiv 0 \bmod d$) will *not* be in the rational nullspace of A – these will only certify rational inconsistency. Thus we need a u such that $uA \equiv 0 \bmod d$ but $uA \ne 0$. The following lemma describes how a solution randomly sampled from the modular nullspace of A will provide the desired certificate with high probability.

We will generally require some stronger conditions on A which can be obtained by random preconditioning (see Lemma 3.4). For convenience we prove these lemmas for an arbitrary principal ideal domain R. For the initial algorithm $\mathsf{R} = \mathbb{Z}$, but R will be a localization of an order of a number field at a prime later on (recall that an *order* of a number field is a subring of the ring of algebraic integers in that number field). For a prime $p \in \mathsf{R}$ and $e > 0$, define $\mathsf{R}_{p^e} = \mathsf{R}/p^e\mathsf{R}$.

LEMMA 3.3. *Let R be any principal ideal domain and $T \in \mathsf{R}^{r\times n}$ with rank r and rth determinantal divisor ∇_r. Let $p \in \mathsf{R}$ be prime in R and $e = \mathrm{ord}_p(\nabla_r)$. Let $v \in \mathsf{R}^{r\times 1}$ such that there exists no $y \in \mathsf{R}^{n\times 1}$ with $Ty \equiv v \bmod p^e$. Suppose $w_1, \ldots, w_r \in \mathsf{R}^{1\times r}$ generate the left nullspace module of T modulo p^e. For randomly selected $\delta_1, \ldots, \delta_r \in \{0, 1\}$ and $w = \sum_{1\le i\le r} \delta_i w_i \in \mathsf{R}^{1\times r}$, we have $\mathrm{Prob}\{w\cdot v \not\equiv 0 \bmod p^e\} \ge 1/2$.*

PROOF. Let $S = \mathrm{diag}(p^{m_1}, \ldots, p^{m_r}) \in \mathsf{R}^{r\times n}_{p^e}$ with $0 \le m_1 \le \cdots \le m_r \le e$ be the Smith normal form of T modulo p^e, that is $S = PTQ$ with $P \in \mathsf{R}^{r\times r}_{p^e}$, $Q \in \mathsf{R}^{n\times n}_{p^e}$ and $\det P, \det Q \in \mathsf{R}^*_{p^e}$. For $u \in \mathsf{R}^{1\times r}_{p^e}$,

$$\begin{aligned} uT \equiv 0 \bmod p^e &\iff uTQ \equiv 0 \bmod p^e \\ &\iff \underbrace{uP^{-1}}_{\bar u}\cdot S \equiv 0 \bmod p^e. \end{aligned}$$

Let $\bar v = Pv = (\bar v_1, \ldots, \bar v_r) \in \mathsf{R}^{r\times 1}$ and $\bar w_i = w_i P^{-1}$ for $1 \le i \le r$.

Define $W(z_1, \ldots, z_r) = \sum_{1\le i\le r} z_i w_i$ for indeterminates $z_1, \ldots, z_r$. Consider the linear form $W(z_1, \ldots, z_r)\cdot v = \sum_{1\le i\le r} z_i w_i \cdot v = \sum_{1\le i\le r} z_i \bar w_i \cdot \bar v$. If we can show that this form is not identically zero modulo p^e, then for randomly chosen $\delta_1, \ldots, \delta_r \in \{0, 1\}$ we have $W(\delta_1, \ldots, \delta_r)\cdot v \not\equiv 0 \bmod p^e$ with probability at least $1/2$ by the Lemma of Zippel (1979)/Schwartz (1980). Hence $w = W(\delta_1, \ldots, \delta_r)$ has the desired properties with probability at least $1/2$.

To show $W(z_1, \ldots, z_r)\cdot v$ is not identically zero, we need only show the existence of a single $\bar u \in \mathsf{R}^{1\times r}_{p^e}$ such that $\bar u S \equiv 0 \bmod p^e$ and $\bar u\cdot\bar v \not\equiv 0 \bmod p^e$. This follows since $w_1, \ldots, w_r$ are assumed to generate the left nullspace module of A mod p^e, so $\bar w_1, \ldots, \bar w_r$ must generate the left nullspace module of S mod p^e. Note that $\bar u S \equiv 0 \bmod p^e$ if and only if $\bar u = (\beta_1 p^{e-m_1}, \ldots, \beta_r p^{e-m_r}) \in \mathsf{R}^{1\times r}_{p^e}$, where $\beta_i \in \mathsf{R}_{p^e}$. Since there is no solution $y \in \mathsf{R}^{n\times 1}_{p^e}$ to $Ty = v$ there must exist an index k $(1 \le k \le r)$ such that $p^{m_k} \nmid \bar v_k$ (this is a *diagonal* system). Let $\bar u \in \mathsf{R}^{1\times r}$ be zero except for the kth component which is p^{e-m_k}. Then $\bar u S \equiv 0 \bmod p^e$ but $\bar u\cdot\bar v = p^{e-m_k}\cdot \bar v_k \not\equiv \bmod p^e$. □

Next, we present the properties which may be assume with high probability about a random preconditioning of the input matrix.

LEMMA 3.4. *Let* R *be any principal ideal domain and* $A \in \mathsf{R}^{n\times n}$ *of rank* r *with* r*th determinantal divisor* ∇_r. *Let* $\mathcal{L}$ *be a subset of* R *with at least* $2r(r+1)$ *elements and* $\alpha_2,\ldots,\alpha_n,\beta_2,\ldots,\beta_n$ *randomly and uniformly chosen from* $\mathcal{L}$. *Let*

$$U = \begin{pmatrix} 1 & \alpha_2 & \alpha_3 & \cdots & \alpha_n \\ & 1 & \alpha_2 & \ddots & \vdots \\ & & \ddots & \ddots & \vdots \\ & & & 1 & \alpha_2 \\ & & & & 1 \end{pmatrix} \quad L = \begin{pmatrix} 1 & & & & \\ \beta_2 & 1 & & & \\ \beta_3 & \beta_2 & \ddots & & \\ \vdots & \ddots & \ddots & \ddots & \\ \beta_n & \cdots & \beta_3 & \beta_2 & 1 \end{pmatrix}$$

$$D = \begin{pmatrix} \gamma_1 & & \\ & \ddots & \\ & & \gamma_n \end{pmatrix} \tag{3.1}$$

and form $B = UALD$ *with* $f(z) = \text{minpoly}(B) \in \mathsf{R}[z]$. *Let* $p \in \mathsf{R}$ *be any prime such that* $\#(\mathsf{R}/p\mathsf{R}) > 2r(r+1)$. *Then with probability at least* $1/2$:

(i) $f(z) = z\cdot\bar{f}(z)$ *with* $\deg\bar{f} = r$, $d := \bar{f}(0) \neq 0$, *and* $\text{charpoly}(B) = z^{n-r}\cdot\bar{f}(z)$;

(ii) $\nabla_r \mid d$ *and* $\text{ord}_p(d) = \text{ord}_p(\nabla_r)$;

(iii) the first r *rows of* $B \bmod p^e$ *generate the* R_{p^e}*-module spanned by all the rows of* $B \bmod p^e$ *and the first* r *columns of* $B \bmod p^e$ *generate the* R_{p^e}*-module spanned by all the columns of* $B \bmod p^e$.

PROOF. Part (i) follows from Kaltofen & Saunders (1991), Lemma 2 (by considering Wiedemann's algorithm over the quotient field of R). Parts (ii) and (iii) follow from Giesbrecht (1997), Theorem 2.5 and Theorem 2.1 respectively. □

Algorithm: `CertifyZInconsistency`

Input: – $A \in \mathbb{Z}^{n\times n}$ and $b \in \mathbb{Z}^{n\times 1}$ such that there exist rational solutions $x \in \mathbb{Q}^{n\times 1}$ to $Ax = b$ but no integer solution $x \in \mathbb{Z}^{n\times 1}$; Assume also that there is a prime $p > 2r(r+1)$ and $e > 0$ such that $Ax \equiv b \bmod p^e$ has no solution for $x \in \mathbb{Z}_{p^e}^{n\times 1}$;

Output: – $u \in \mathbb{Z}^{1\times n}$ and $d \in \mathbb{Z}$ such that $uA \equiv 0 \bmod d$ and $ub \not\equiv 0 \bmod d$;

Repeat

(1) Choose random $\alpha_2,\ldots,\alpha_n,\ \beta_2,\ldots,\beta_n,\ \gamma_1,\ldots,\gamma_n$ from $\mathcal{L} := \{0,\ldots,2n(n+1)\}$ and construct black boxes for U, L, D and $B = UALD$ as in (3.1);
Let $B_r \in \mathbb{Z}^{r\times r}$ be the leading $r\times r$ submatrix of B (for which we also have a black box);

(2) Find $\text{minpoly}(B) = x\cdot\bar{f}(x) \in \mathbb{Z}[x]$ and set $r := \deg\bar{f}$ and $d := \bar{f}(0)$;
Use Wiedemann's algorithm over $\mathbb{Q}$;

(3) Choose random $\delta_1,\ldots,\delta_r \in \{0,1\}$;
Let $v_r := (\delta_1 d,\ldots,\delta_r d) \in \mathbb{Z}^{1\times r}$;

(4) Solve $y_r B_r = v_r$ for the unique $y_r = (\xi_1,\ldots,\xi_r) \in \mathbb{Z}^{1\times r}$ using Wiedemann's algorithm over $\mathbb{Q}$;
Let $y := (\xi_1,\ldots,\xi_r,0,\ldots,0)$;

(5) Let $u := yU \in \mathbb{Z}^{1\times r}$;

(6) **Until** $uA \equiv 0 \bmod d$ and $ub \not\equiv 0 \bmod d$;

Return $u \in \mathbb{Z}^{1\times n}$ and d.

THEOREM 3.5. `CertifyZInconsistency` *works as specified when there exists a prime* $p > 2r(r+1)$, $t \geq 1$ *and a* $u \in \mathbb{Z}^{1\times n}$ *such that* $uA \equiv 0 \bmod p^t$ *and* $ub \not\equiv 0 \bmod p^t$. *In this situation* $O(1)$ *iterations of the main loop are required.*

PROOF. Clearly, if there is an output, it satisfies the required properties. We need only show that on each iteration the algorithm produces an output with probability at least $1/4$.

By Lemma 3.2 there exists a prime p and $e > 1$ such that $Ax \equiv b \bmod p^e$ has no solution $x \in \mathbb{Z}_{p^e}^{n\times 1}$, and a $u \in \mathbb{Z}^{1\times n}$ such that $uA \equiv 0 \bmod p^e$ and $ub \not\equiv 0 \bmod p^e$. By Lemma 3.4, we may assume that after steps (2)-(3), with probability at least $1/2$, the constructed B has its first r columns generate the $\mathbb{Z}_{p^e}$-module spanned by all the columns of B and whose first r rows generate the $\mathbb{Z}_{p^e}$-module spanned by all the rows of B. We may also assume that $e = \text{ord}_p(d) = \text{ord}_p(\nabla_r)$.

Let B_0 be the $r\times n$ matrix consisting of the first r rows of B and $\bar{b}_0$ the first r rows of $\bar{b} = Ub$. There can be no solution $\bar{x} \in \mathbb{Z}_{p^e}^{n\times 1}$ such that $B_0\bar{x} \equiv \bar{b}_0 \bmod p^e$, since the remaining rows of B are assumed to be linear combinations of the first r rows.

The algorithm generates random vectors $y_r \in \mathbb{Z}^{1\times r}$ which sample the left nullspace of $B_0 \bmod p^e$ as required in Lemma 3.3 in steps (4) and (5). First recall that we have assumed the first r columns of B generate the column space of $B \bmod p^e$, so the first r columns of B_0 generate the column space of $B_0 \bmod p^e$. Thus, we need only find vectors y_r such that $y_r B_r \equiv 0 \bmod p^e$, and it will follow that $y_r B_0 \equiv 0 \bmod p^e$. To do this we note that if $yB_r \equiv 0 \bmod d$, then $yB_r = dw$ for some $w \in \mathbb{Z}^{1\times r}$. Let $\mu_i \in \mathbb{Z}^{1\times r}$ be the ith unit vector and $\psi_i \in \mathbb{Z}^{1\times r}$ the unique solution to $\psi_i B_r = d\cdot\mu_i$ for $1 \leq i \leq r$ (ψ_i is integral since $\det B_r = d$). Since $\text{ord}_p(d) = \text{ord}_p(\nabla_r)$, we know $\psi_1,\ldots,\psi_r$ generate the left-nullspace module of $B_r \bmod p^e$. In step (4) we sample this as required in Lemma 3.3.

Thus, with probability at least $1/2$, $y_r B_r \equiv 0 \bmod p^e$ and $y_r\cdot\bar{b}_0 \not\equiv 0 \bmod p^e$. It follows, assuming that our pre-conditioning was correct (which is true with probability at least $1/2$), that $(y_r|0,\ldots,0)B \equiv 0 \bmod p^e$ and $(y_r|0,\ldots,0)Ub \not\equiv 0 \bmod p^e$, or equivalently that $uA \equiv 0 \bmod p^e$ and $ub \not\equiv 0 \bmod p^e$ with $u = yU$ as in step (6). Since $p^e \mid d$, $ub \not\equiv 0 \bmod p^e$ as well.

Thus, with probability at least $1/4$ on each iteration, the algorithm finds a certificate of inconsistency. □

We summarize the cost of the algorithm as follows.

THEOREM 3.6. *Let* $A \in \mathbb{Z}^{n\times n}$ *with rank* r *and* $b \in \mathbb{Z}^{n\times 1}$, *and assume no solution* $x \in \mathbb{Z}^{n\times 1}$ *to* $Ax = b$ *exists. Let* $\varrho = r\log\|A\| + \log\|b\|$. *Assume that there exists a prime* $p > 2r(r+1)$ *and integer* $e \geq 1$ *such that there is also no* $x \in \mathbb{Z}^{n\times 1}$ *such that* $Ax \equiv b \bmod p^e$.

- *We can find a certificate* $u \in \mathbb{Z}^{1\times n}$ *and integer* d *such that* $uA \equiv 0 \bmod d$ *and* $ub \not\equiv 0 \bmod d$ *with an expected number of* $O^{\sim}(r\varrho)$ *matrix-vector products by* A^t *modulo primes with* $O(\log n + \log\log(\|A\| + \|b\|))$ *bits.*
- *The output* u *satisfies* $\log\|u\| = O^{\sim}(r\log n + r\log\|A\|)$.
- *An additional* $O^{\sim}(r^2 + rn\varrho + n\,\mathrm{M}(\varrho))$ *bit operations and additional storage for* $O^{\sim}(n\varrho)$ *words are required.*

PROOF. The dominant cost is the execution of Wiedemann's algorithm to find the minimal polynomial of an integer matrix in step (2) and to solve the pre-conditioned

$r \times r$ non-singular system in step (4) an expected constant number of times. These costs are summarized in Giesbrecht (1996), Theorem 1.5 and Giesbrecht (1997), Fact 3.2. □

Inconsistency with small primes dividing determinantal divisors

The case when the only witnesses to Diophantine inconsistency are modulo powers of a small prime p (i.e., $p < 2r(r+1)$) is not covered by the above algorithm. In this case we employ the ring extension techniques of Giesbrecht (1997). The algorithm remains essentially the same, but we work in a sequence of ring extensions of $\mathbb{Z}$ such that each small prime p dividing ∇_r is guaranteed to remain inert (i.e., does not factor) in at least one of these extensions. The asymptotic cost is an extra poly-logarithmic factor.

We will compute in extension rings of $\mathbb{Z}$ as follows. Let p be a prime and $\Gamma \in \mathbb{Z}[\lambda]$ a monic polynomial in indeterminate λ such that $\Gamma \bmod p$ is irreducible in $\mathbb{Z}_p[\lambda]$. Define $\theta = \lambda \bmod \Gamma$, so $\mathbb{Z}[\theta]$ is an order in the number field $\mathbb{Q}(\theta)$. By our choice of Γ, p remains inert (i.e., does not factor) in $\mathbb{Z}[\theta]$.

A difficulty in working with $\mathbb{Z}[\theta]$ is that this is not a principal ideal domain. We can recover this property in part by considering the localization of $\mathbb{Z}[\theta]$ at p. Define $\mathbb{Z}_{p^*} = \{a/b : a, b \in \mathbb{Z}, b \not\equiv 0 \bmod p\}$, the *localization* of $\mathbb{Z}$ at p. This is a principal ideal domain where the only prime is p and all ideals have the form $p^i\mathbb{Z}_{p^*}$ for some $i \geq 0$. Since p remains inert in $\mathbb{Z}[\theta]$, the localization of $\mathbb{Z}[\theta]$ is $\mathbb{Z}_{p^*}[\theta]$ (see, e.g., Giesbrecht 1997, Section 4). In particular, $\mathbb{Z}_{p^*}[\theta]$ is also a principal ideal domain in which the only prime is p and all ideals have the form $p^i\mathbb{Z}_{p^*}[\theta]$ for some $i \geq 0$. Clearly we have the inclusions $\mathbb{Z} \subseteq \mathbb{Z}_{p^*} \subseteq \mathbb{Q}$ and $\mathbb{Z}[\theta] \subseteq \mathbb{Z}_{p^*}[\theta] \subseteq \mathbb{Q}[\theta]$.

Algorithm: `CertifyZInconsistencySmallPrimes`

Input: – $A \in \mathbb{Z}^{n\times n}$ and $b \in \mathbb{Z}^{n\times 1}$ such that there exist rational solutions $x \in \mathbb{Q}^{n\times 1}$ to $Ax = b$ but no integer solution $x \in \mathbb{Z}^{n\times 1}$; Assume also that there is a prime $p < 2r(r+1)$ and $e > 0$ such that $Ax \equiv b \bmod p^e$ has no solution for $x \in \mathbb{Z}_{p^e}^{n\times 1}$.

Output: – $u \in \mathbb{Z}^{1\times n}$ and $d \in \mathbb{Z}$ such that $uA \equiv 0 \bmod d$ and $ub \not\equiv 0 \bmod d$;

While true Do

(1) Choose random $\alpha_2, \ldots, \alpha_n,\ \beta_2, \ldots, \beta_n,\ \gamma_1, \ldots, \gamma_n$ from $\mathcal{L} := \{0, \ldots, 2n(n+1)\}$ and construct black boxes for U, L, D and $B = UALD$ as in (3.1);

(2) Find $\text{minpoly}(B) = z \cdot \bar{f}(z) \in \mathbb{Z}[z]$ and set $r := \deg \bar{f}$ and $d := \bar{f}(0)$;
Use Wiedemann's algorithm over $\mathbb{Q}$;

(3) Let $p_1, \ldots, p_\kappa \in \mathbb{Z}$ be the primes dividing d which are less than $2r(r+1)$;

(4) Using the algorithm `BuildOrders` from Giesbrecht (1997), on inputs $\eta = \lceil \log_2(2r(r+1)) \rceil$ and $p_1, \ldots, p_\kappa$, find a set $G = \{\Gamma_1, \ldots, \Gamma_l\} \subseteq \mathbb{Z}[\lambda]$ of monic polynomials of degree η such that for each p_i there exists a $\Gamma_j \in G$ such that $\Gamma_j \bmod p_i$ is irreducible in $\mathbb{Z}_p[\lambda]$;

(5) Let $\theta_j := (\lambda \bmod \Gamma_j)$ and $\mathcal{L}_j := \{\sum_{0\leq k<\eta} a_k\theta_j^k : a_k \in \{0,1\}\} \subseteq \mathbb{Z}[\theta_j]$ for $1 \leq j \leq l$;
For $j := 1$ **to** l **Do**

(6) Choose random $\alpha_2, \ldots, \alpha_n,\ \beta_2, \ldots, \beta_n,\ \gamma_1, \ldots, \gamma_n$ from $\mathcal{L}_j$ and construct black boxes for U, L, D and $B = UALD \in \mathbb{Z}[\theta_j]^{n\times n}$ as in (3.1);
Let $B_r \in \mathbb{Z}[\theta_j]^{r\times r}$ be the leading $r \times r$ submatrix of B (for which we also have a black box);

(7) Find $\text{minpoly}(B) = z \cdot \bar{f}(z) \in \mathbb{Z}[\theta_j][z]$ and set $r := \deg \bar{f}$ and $d := \bar{f}(0)$;
Use Wiedemann's algorithm over $\mathbb{Q}(\theta_j)$;

(8) Choose random $\delta_1, \ldots, \delta_r \in \{0,1\}$; Let $v_r := (\delta_1 d, \ldots, \delta_r d) \in \mathbb{Z}^{1\times r}$;

(9) Solve $y_r B_r = v_r$ for the unique $y_r := (\xi_1, \ldots, \xi_r) \in \mathbb{Z}[\theta_j]^{1\times r}$ using Wiedemann's algorithm over $\mathbb{Q}(\theta_j)$;
Let $y := (\xi_1, \ldots, \xi_r, 0, \ldots, 0)$;

(10) Let $u := yU \in \mathbb{Z}[\theta_j]^{1\times r}$;

(11) **For** all $p \in \{p_1, \ldots, p_\kappa\}$ such that $\Gamma_j \bmod p$ is irreducible in $\mathbb{Z}_p[\lambda]$ **Do**

(12) Let $e = \text{ord}_p(d)$;
If $uA \equiv 0 \bmod p^e$ and $ub \not\equiv 0 \bmod p^e$ goto (13);
End For
End For
End While

(13) Assume $u = \sum_{0\leq i<\eta} u_i\theta_j^i$ for $u_i \in \mathbb{Z}^{1\times n}$; Find a k such that $u_k b \not\equiv 0 \bmod p^e$;

(14) **Return** u_k and p^e.

THEOREM 3.7. `CertifyZInconsistencySmallPrimes` *works as specified when there exists a prime $p < 2r(r+1)$, $t \geq 1$ and a $u \in \mathbb{Z}^{1\times n}$ such that $uA \equiv 0 \bmod p^t$ and $ub \not\equiv 0 \bmod p^t$. In this situation $O(1)$ iterations of the main loop are required.*

PROOF. We first show that the output is always correct. The algorithm reaches step (13) only if it finds a $u \in \mathbb{Z}[\theta_j]$ and prime power $p^e \in \mathbb{Z}$ such that $uA \equiv 0 \bmod p^e$ and $ub \not\equiv 0 \bmod p^e$. Thus, $u \not\equiv 0 \bmod p^e$ and so there must exist a u_i such that $u_i b \not\equiv 0 \bmod p^e$. Since $u_i A \equiv 0 \bmod p^e$ as well, this is a certificate of Diophantine inconsistency.

Next we must show that the algorithm returns a certificate of inconsistency with an expected $O(1)$ iterations of the outer loop. In steps (1) and (2) we compute the rank r of B (and A) and a multiple d of ∇_r correctly with probability $1/2$ by Lemma 3.4. Assume that $p < 2r(r+1)$ is a prime such that there exists an e such that $Ax \equiv b \bmod p^e$ has no solution $x \in \mathbb{Z}^{n\times 1}$. Then there exists a $u \in \mathbb{Z}^{1\times n}$ such that $uA \equiv 0 \bmod p^e$ and $ub \not\equiv 0 \bmod p^e$.

In steps (3)-(5) we construct a set $G \subseteq \mathbb{Z}[\lambda]$ of monic polynomials, such that there exists a $\Gamma_j \in G$ with Γ_j irreducible modulo p. Thus, the localization of $\mathbb{Z}[\theta_j]$ at p is $\mathbb{Z}_{p^*}[\theta_j]$, and this is a principal ideal domain whose only prime is p. Moreover, the residue class field $\mathbb{Z}_{p^*}[\theta_j] \bmod p \cong \mathbb{Z}[\theta_j] \bmod p$ is a finite field with p^η elements.

The main loop in this algorithm from steps (6)-(12) is similar to the main loop of `CertifyZInconsistency`, as is the proof that it finds a certificate with probability $1/2$ with each iteration of the outer loop. Consider only the iteration j of the inner loop where $\Gamma_j \bmod p$ is irreducible in $\mathbb{Z}_p[\lambda]$ as above. The proof that each iteration finds a certificate u with probability $1/2$ follows similarly to Theorem 3.5. To apply Lemma 3.3 we consider this as a computation over $\mathbb{Z}_{p^*}[\theta_j]$ under the standard embedding. As in the integral case, $y_r \in \mathbb{Z}[\theta_j]^{1\times r}$ because B_r has determinant d.

One problem which arises is that the d computed in step (7) is not necessarily an integer. However, since we have a small number of potential primes to construct our certificate, we can simply take the order of these primes in d. Find the order e of p in d in step (12) simply by considering d as a polynomial in $\mathbb{Z}[\lambda]$ modulo Γ_j and taking the order of p in

the content of this polynomial (the GCD of the coefficients). This works since p is inert in $\mathbb{Z}[\theta_j]$, and $p^e \mid d$ in $\mathbb{Z}[\theta_j]$. □

THEOREM 3.8. *Let $A \in \mathbb{Z}^{n\times n}$ with rank r and $b \in \mathbb{Z}^{n\times 1}$. Assume no solution $x \in \mathbb{Z}^{n\times 1}$ to $Ax = b$ exists. Let $\varrho = r\log\|A\| + \log\|b\|$. Assume also that there is a prime $p < 2r(r+1)$ and an integer $e \geq 1$ such that $Ax \equiv b \bmod p^e$ has no solution.*

- *We can find a certificate $u \in \mathbb{Z}^{1\times n}$ and integer d such that $uA \equiv 0 \bmod d$ and $ub \not\equiv 0 \bmod d$ with an expected number of $\tilde{O}(r\varrho)$ matrix-vector products by A^t modulo primes with $O(\log n + \log\log(\|A\| + \|b\|))$ bits.*
- *The output u satisfies $\log\|u\| = \tilde{O}(r\log n + r\log\|A\|)$.*
- *An additional $\tilde{O}(r^2 + rn\varrho + n\,\mathrm{M}(\varrho))$ bit operations and additional storage for $\tilde{O}(n\varrho)$ words are required.*

PROOF. The dominant cost is the execution of Wiedemann's algorithm to find the minimal polynomial of an integer matrix in step (2) and to solve the pre-conditioned $r \times r$ non-singular system over $\mathbb{Q}(\theta_j)$ in step (7). Step (2) is performed a constant number of times. The set $\#G$ contains $O(\log^2 r)$ polynomials $\Gamma_j \in \mathbb{Z}[\lambda]$, each of degree $\eta = O(\log r)$ and such that $\log\|\Gamma_j\| = O(\log^2 r)$. The cost of solving the system $y_r B_r = v_r$ over $\mathbb{Q}[\theta_j]$ in step (9) follows from a straight-forward analysis of Wiedemann's algorithm using a homomorphic imaging scheme. Such an analysis is done in Giesbrecht (1997), Theorem 5.3. □

To summarize, we do not know a priori whether a particular Diophantine system is (a) consistent, (b) rationally inconsistent, (c) rationally consistent and has a certificate of Diophantine inconsistency modulo a power of a prime $p > 2n(n+1)$, or (d) rationally consistent and has no certificate of Diophantine inconsistency modulo a prime $p > 2n(n+1)$. A complete solution to this problem is provided in the following theorem.

THEOREM 3.9. *Let $A \in \mathbb{Z}^{n\times n}$ with rank r and $b \in \mathbb{Z}^{n\times 1}$. Let $\varrho = r\log\|A\| + \log\|b\|$.*

- *We can find a $x \in \mathbb{Z}^{n\times 1}$ such that $Ax = b$ or provide a certificate $u \in \mathbb{Z}^{1\times n}$, $d \in \mathbb{Z}$ that no such solution exists (where $uA \equiv 0 \bmod d$ and $ub \not\equiv 0 \bmod d$. The algorithm requires an expected number of $\tilde{O}(r\varrho)$ matrix-vector products by A modulo primes with $O(\log n + \log\log(\|A\| + \|b\|))$ bits.*
- *The output x (if produced) satisfies $\log\|x\| = \tilde{O}(r\log n + r\log\|A\| + \log\|b\|)$.*
- *The output u (if produced) satisfies $\log\|u\| = \tilde{O}(r\log n + r\log\|A\|)$.*
- *An expected additional $\tilde{O}(r^2 + rn\varrho + n\,\mathrm{M}(\varrho))$ bit operations and additional storage for $\tilde{O}(n\varrho)$ words are required.*

PROOF. First attempt to find a Diophantine solution with a small constant number of iterations of the solver of Fact 3.1. If this fails, attempt to determine if the system is rationally consistent by attempting to solve or prove inconsistency of the system modulo randomly chosen word-sized primes as described above and in Section 2. If the system is rationally consistent, attempt to prove Diophantine inconsistency using `CertifyZInconsistency`. If this fails after a small constant number of iterations, attempt to prove Diophantine inconsistency using `CertifyZInconsistencySmallPrimes` for a few iterations. This sequence should be repeated until either a Diophantine solution or certificate of Diophantine inconsistency has been found. The expected number of repetitions is constant. □

Acknowledgement

The authors would like to thank Erich Kaltofen for his discussions about this problem.

References

D. Coppersmith. Solving linear equations over GF(2); block Lanczos algorithm. *Linear algebra and its applications* **192**, pp. 333–350, 1993.

D. Coppersmith. Solving homogeneous linear equations over GF(2) via block Wiedemann algorithm. *Mathematics of Computation* **62**(205), pp. 333–350, 1994.

W. Eberly and E. Kaltofen. On randomized Lanczos algorithms. In *Proceedings of ISSAC'97*, pp. 176–183. ACM Press, 1997.

M. Giesbrecht. Probabilistic computation of the Smith normal form of a sparse integer matrix. In *Algorithmic Number Theory: Second International Symposium*, ed. H. Cohen, pp. 175–188, 1996. Proceedings to appear in Springer's Lecture Notes in Computer Science.

M. Giesbrecht. Efficient parallel solution of sparse systems of linear diophantine equations. In *Proceediings of PASCO'97*, 1997. To appear. 10pp.

E. Kaltofen. Analysis of Coppersmith's block Wiedemann algorithm for the parallel solution of sparse linear systems. *Mathematics of Computation* **64**(210), pp. 777–806, 1995.

E. Kaltofen and B. D. Saunders. On Wiedemann's method of solving sparse linear systems. In *Proc. AAECC-9*, vol. 539 of *Springer Lecture Notes in Comp. Sci.*, 1991. 29-38.

R. Lambert. *Computational aspects of discrete logarithms.* PhD thesis, University of Waterloo, Waterloo, ON, 1996.

J. T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. Assoc. Computing Machinery* **27**, pp. 701–717, 1980.

G. Villard. Further analysis of Coppersmith's block Wiedemann algorithm for the solution of sparse linear systems. In *Proceedings of ISSAC'97*, 1997. To appear.

D. Wiedemann. Solving sparse linear equations over finite fields. *IEEE Transactions on Information Theory* **IT-32**, pp. 54–62, 1986.

R. Zippel. Probabilistic algorithms for sparse polynomials. In *Proc. EUROSAM 79*, pp. 216–226, Marseille, 1979.