Efficient Parallel Solution of Sparse Systems of Linear Diophantine Equations[†]

Mark Giesbrecht

Department of Computer Science
University of Manitoba
Winnipeg, Manitoba, Canada, R3T 2N2
Email: mwg@cs.umanitoba.ca

Abstract

We present a new iterative algorithm for solving large sparse systems of linear Diophantine equations which is fast, provably exploits sparsity, and allows an efficient parallel implementation. This is accomplished by reducing the problem of finding an *integer* solution to that of finding a very small number of *rational* solutions of random Toeplitz preconditionings of the original system. We then employ the Block-Wiedemann algorithm to solve these preconditioned systems efficiently in parallel. Solutions produced are small and space required is essentially linear in the output size.

1 Introduction

Computing integer solutions to systems of linear Diophantine equations is a classical mathematical problem with many interesting applications in number theory (see, e.g., Cohen 1993), group theory (see, e.g., Newman 1972) and combinatorics (see, e.g., Gibbons 1996). Given an input matrix $A \in \mathbb{Z}^{n\times n}$ and vector $w \in \mathbb{Z}^{n\times 1}$, the problem is to find *integer* vectors $v \in \mathbb{Z}^{n\times 1}$ such that $Av = w$. It appears to be considerably harder to compute integer solutions than solutions over $\mathbb{Q}$ or more general fields, the main difficulty being controlling (potentially exponential) intermediate expression swell. Moreover, in practice many of the matrices encountered are sparse (lots of entries are zero) and it is desirable to exploit this in our algorithms (see, e.g., Hafner & McCurley 1989). For matrices over fields this has been accomplished admirably by the algorithms of Wiedemann (1986), Coppersmith (1994), Kaltofen (1995) and Villard (1997). The latter algorithms are also extremely well-suited to a coarse-grained parallel implementation. In this paper we show how to achieve similar success with sparse integer matrices, producing integer solutions of small size while eliminating intermediate expression swell and fill-in. Our algorithm gives a substantial improvement for sparse matrices, at least asymptotically, over the best known algorithms (see below) in both sequential and coarse-grained parallel implementations. The main result we demonstrate is (summarized from Corollary 5.4):

> *Let $A \in \mathbb{Z}^{n\times n}$ with rank r and $w \in \mathbb{Z}^{n\times 1}$, and assume a solution $v \in \mathbb{Z}^{n\times 1}$ to $Av = w$ exists. Let $\varrho = r\log\|A\| + r\log r + \log\|w\|$ and suppose we are computing on a network of $N \le r\varrho$ processors.*
>
> - *We can find a $v \in \mathbb{Z}^{n\times 1}$ such that $Av = w$ with an expected number of $O^{\sim}(r\varrho/N)$ matrix-vector products by A modulo primes with $O(\log n + \log\varrho)$ bits.*
> - *The output v satisfies $\log\|v\| = O^{\sim}(r\log n + \varrho)$.*
> - *An additional $O^{\sim}(r^2 + rn\varrho/N + n\,\mathrm{M}(\varrho)/\min(n,N))$ bit operations are executed simultaneously by each processor.*
> - *Each processor requires additional storage for $O^{\sim}(n + n\varrho/\min(n,N))$ words (not including possibly shared images of A modulo single-word primes).*
>
> *Here $\|A\| = \max_{ij}|A_{ij}|$ (similarly for $\|v\|, \|w\|$).*

The algorithm is probabilistic and solutions produced are guaranteed correct; if a solution exists for a particular input, any invocation of the algorithm on that input produces a solution with probability at least 1/2. $O(\mathrm{M}(l))$ bit operations are required to multiply two integers with l bits ($\mathrm{M}(l) = l^2$ with standard arithmetic and $\mathrm{M}(l) = l\log l\log\log l$ using FFT-based methods). For convenience we occasionally use "soft-Oh" notation in our cost analyses: for any $f, g : \mathbb{R}^l \to \mathbb{R}$, $f = O^{\sim}(g)$ if and only if $f = O(g\cdot\log^c g)$ for some constant $c > 0$.

Like the algorithms of Wiedemann (1986) and Coppersmith (1994) which motivated this work, we employ the so-called "black-box" paradigm, in which a matrix is defined by its action on vectors by matrix-vector product. Individual entries of the input matrix are not manipulated directly. Clearly a matrix with lots of zero entries will have a fast black box. As in Giesbrecht (1996) we adapt this technique to integer matrices by working with matrix-vector products modulo word-sized primes. Our goal then is to demonstrate comparable results with Diophantine linear systems as have been obtained for systems over a field.

Early attempts at solving systems of linear Diophantine equations go back at least to Blankinship (1966), Borosh & Fraenkel (1966) and Bradley (1971), while the first polynomial-time solution appears in Kannan & Bachem (1979). Since then, there have been many improvements; see, e.g., Chou & Collins (1982), Iliopolous (1989), Havas *et*

[†]Research was supported in part by Natural Sciences and Engineering Research Council of Canada research grant OGP0155376.

Appears in the Proceedings of PASCO'97: ACM International Symposium on Parallel Symbolic Computation, 1997, pp. 1-10.

al. (1993), Havas & Majewski (1994), Storjohann & Labahn (1996) and Storjohann (1996). Most of these methods proceed by computing a triangular (Hermite) or diagonal (Smith) form of A with multiplier matrices, from which the space of solutions to the system is easily determined. Storjohann (1996) presents the asymptotically best solution to date:

> *On input $A \in \mathbb{Z}^{m\times n}$ and $w \in \mathbb{Z}^{m\times 1}$, with $m \le n$, a vector $v \in \mathbb{Z}^{n\times 1}$ such that $Av = w$ can be found with $\tilde{O}(nm^3 \log^2(\|A\|+\|w\|) + m^4 \log^3(\|A\|+\|w\|))$ bit operations using standard integer and matrix arithmetic. The output $v \in \mathbb{Z}^{n\times 1}$ satisfies $\log\|v\| = O(m\log(m)\cdot(\log\|A\| + \log\|w\|))$.*

This is probably close to the best possible asymptotic cost for dense matrices without resorting to non-standard matrix arithmetic, and is very close to the cost of finding a rational solution to the same system. By comparison, our new algorithm, implemented sequentially ($N = 1$), performs comparably — even marginally better — on *dense* input, and substantially better on *sparse* input:

> *On input $A \in \mathbb{Z}^{m\times n}$ with $O(nm^\xi)$ non-zero elements (for some $0 \le \xi \le 1$) and $w \in \mathbb{Z}^{m\times 1}$, with $m \le n$, a vector $v \in \mathbb{Z}^{n\times 1}$ such that $Av = w$ can be found with an expected number of $\tilde{O}(nm^{2+\xi}\log(\|A\|+\|w\|) + nm^2\log^2(\|A\|+\|w\|))$ bit operations using standard integer and matrix arithmetic. The output $v \in \mathbb{Z}^{n\times 1}$ satisfies $\log\|v\| = \tilde{O}(m\log n + m\log\|A\| + \log\|w\|)$.*

The basic idea behind our algorithm is to solve the leading $r \times r$ system (where $r = \operatorname{rank} A$) of a small set of equivalent, random Toeplitz preconditionings of the original system over $\mathbb{Q}$. Let $U, L \in \mathbb{Z}^{n\times n}$ be "random" unimodular upper and lower triangular Toeplitz matrices respectively, and consider solving the system $UALv = Uw$. Kaltofen & Saunders (1991) showed that over a field the leading $r \times r$ submatrix B_r of UAL is strongly non-singular, and by solving this system we quickly obtain as solution $\hat{v} \in \mathbb{Q}^{n\times 1}$ to $A\hat{v} = w$. In Section 2 we extend Kaltofen & Saunders' result by noting that if $d_1, \ldots, d_r \in \mathbb{Z}$ are the determinantal divisors of A (where the kth determinantal divisor of A is the GCD of all $k \times k$ minors of A), and p is a "large" prime dividing d_k, then the order of p in the leading $k \times k$ minor of B equals the order of p in d_k with high probability (where the *order* of a prime in an integer is the number of times it divides that integer). Moreover, with high probability p does not divide the denominators of any of the coefficients of the obtained solution $\hat{v}$. This is proven by examining the solution space of the preconditioned system in the p-adic closure $\mathbb{Q}_p$ of $\mathbb{Q}$. By considering a very small number ($\approx \log\log(n + \|A\|)$) of preconditioned systems we hopefully obtain a set of rational solutions whose denominators are relatively prime, from which we can construct an integer solution vector. We prove that using the above technique we can efficiently find a solution whose coefficients have "smooth" denominators, i.e., only divisible by primes less than $2r(r+1)$. This method is realized in the algorithm `SmoothSolver` in Section 3.

Unfortunately our analysis fails for small primes dividing d_r (even if the algorithm does not seem to fail often in practice). The problem stems from the failure of the inequality used to bound away from zero the probability of getting a non-zero of a multi-variate polynomial (the so called Zippel-Schwartz Lemma) in this case. To overcome this we considerably extend a technique developed in Giesbrecht (1995) and work in a very small number of orders of number fields of small degree over $\mathbb{Q}$ such that each small prime dividing d_r remains inert in at least one of these orders (the number, degree, and height of these orders is logarithmic in r). While these orders are no longer principal ideal domains (and hence much of the mathematical structure characterizing Diophantine solutions no longer exists), their localizations at these inert primes *are* PID's and we think of our algorithms as working in these p-adic closures (even when they really just compute in a small number field). We prove that rational solutions obtained by preconditioning with random Toeplitz matrices over these orders, and solving over their quotient number fields, are free of small primes dividing their denominators with high probability.

The algorithms for generating these orders with specified inert primes, and the theory for working with their localizations is presented in Section 4. Finally, in Section 5 we present an algorithm `RefineToDiophantine` which takes a smooth rational solution and produces a Diophantine solution. The structure of this algorithm is almost identical to that of `SmoothSolver` except for the computation in number fields; the cost is within a poly-logarithmic factor.

Definitions and Notation

We denote by $\mathbb{F}_p$ the finite field with p elements *(not to be confused with the p-adic integers $\mathbb{Z}_p$, to be introduced later).*

We define a height function on $\mathbb{Q}$ as follows. For $a, b \in \mathbb{Z}$ with $\gcd(a,b) = 1$, we define the *height* of $a/b \in \mathbb{Q}$ as $\mathcal{H}(a/b) = \max\{|a|, |b|\}$. The norm of a matrix $B \in \mathbb{Q}^{m\times n}$ is defined as $\|B\| = \max_{ij} \mathcal{H}(B_{ij})$ and of a polynomial $g = \sum_{0\le i\le m} b_i x^i \in \mathbb{Q}[x]$ as $\|g\| = \max_i \mathcal{H}(b_i)$.

For integers n and $k \le n$, define $\mathcal{C}_k^n = \{(c_1, \ldots, c_k) \in \mathbb{N}^k : 1 \le c_1 < \cdots < c_k \le n\}$. In a ring R, with $B \in \mathsf{R}^{m\times n}$, $\sigma = (b_1, \ldots, b_k) \in \mathcal{C}_k^m$ and $\tau = (c_1, \ldots, c_k) \in \mathcal{C}_k^n$ define the *submatrix* $B\left[\begin{smallmatrix}\sigma\\ \tau\end{smallmatrix}\right]$:

$$B\begin{bmatrix}\sigma\\ \tau\end{bmatrix} = \begin{pmatrix} B_{b_1c_1} & \cdots & B_{b_1c_k} \\ \vdots & & \vdots \\ B_{b_kc_1} & \cdots & B_{b_kc_k}\end{pmatrix} \in \mathsf{R}^{k\times k},$$

and the $(k \times k)$ *minor* $B\binom{\sigma}{\tau} = \det B\left[\begin{smallmatrix}\sigma\\ \tau\end{smallmatrix}\right] \in \mathsf{R}$.

2 Conditions for Diophantine solutions

In this section we present the necessary mathematical underpinnings to our algorithm for solving Diophantine equations. Much of this section is presented abstractly for principal ideal domains. We typically apply these theorems to localizations of $\mathbb{Z}$ and more general orders of number fields.

Smith dominant matrices over PID's

Let R be a principal ideal domain and K its field of fractions. We write $a \sim b$ if there exists a $\mu \in \mathsf{R}^*$ such that $a = \mu b$. Let $B \in \mathsf{R}^{n\times n}$ of rank r with non-zero determinantal divisors $d_1, \ldots, d_r \in \mathsf{R}$. We say that B is *Smith dominant* if $B\binom{1\ldots k}{1\ldots k} \sim d_k$ for $1 \le i \le r$. Note that if R is a field, Smith dominant matrices are exactly those which are strongly non-singular, that is, all leading minors are non-zero.

THEOREM 2.1. *Let $B \in \mathsf{R}^{n\times n}$ be Smith dominant of rank r with non-zero determinantal divisors $d_1, \ldots, d_r$ and $w \in \mathsf{R}^{n\times 1}$. There exists a solution $v \in \mathsf{R}^{n\times 1}$ such that $Bv = w$ if and only if there exist $v_1, \ldots, v_r \in \mathsf{R}$ such that $B(v_1, \ldots, v_r, 0, \ldots, 0)^t = w$.*

REMARK 2.2. *Since* $B\binom{1\dots r}{1\dots r} \sim d_r \neq 0$, $(v_1,\dots,v_r)^t$ *is the unique solution in* $\mathsf{K}^{r\times 1}$ *of*

$$B\begin{bmatrix}1\dots r\\ 1\dots r\end{bmatrix}\begin{pmatrix}v_1\\ \vdots\\ v_r\end{pmatrix} = \begin{pmatrix}w_1\\ \vdots\\ w_r\end{pmatrix}.$$

where $w = (w_1,\dots,w_n)^t$.

PROOF. Since B is Smith dominant, standard unimodular row and column elimination on B (without pivoting) yields the factorization $B = XSY$, where $X \in \mathsf{R}^{n\times n}$ is lower triangular with ones on the diagonal, Y is upper triangular with ones on the diagonal and $S = \mathrm{diag}(s_1,\dots,s_r,0,\dots,0) \in \mathsf{R}^{n\times n}$ is the Smith form of B (that is $s_1 \sim d_1$ and $s_i \sim d_i/d_{i-1}$ for $2 \le i \le r$). Then $Bv = w \iff XSYv = w \iff SYv = X^{-1}w \iff S\hat{v} = \hat{w}$, where $\hat{v} = Yv$ and $\hat{w} = X^{-1}w$. Suppose there exists a solution $v \in \mathsf{R}^{n\times 1}$ to $Bv = w$. Then there exists a $\hat{v} \in \mathsf{R}^{n\times 1}$ such that $S\hat{v} = \hat{w}$, and we can choose $\hat{v} = (\hat{v}_1,\dots,\hat{v}_r,0,\dots,0)^t \in \mathsf{R}^{n\times 1}$ (since columns $r+1\dots n$ of S are all zeros). This yields $v = Y^{-1}\hat{v}$ as a solution to $Av = w$ and $v = (y_1,\dots,y_r,0,\dots,0) \in \mathsf{R}^{n\times 1}$ since Y^{-1} is also upper triangular. The converse is trivial. □

Toeplitz preconditioning into Smith dominant form

Let R be a principal ideal domain and K its field of quotients. Define

$$\mathfrak{U} = \begin{pmatrix}1 & x_2 & x_3 & \cdots & x_n\\ & 1 & x_2 & \ddots & \vdots\\ & & \ddots & \ddots & \vdots\\ & & & 1 & x_2\\ & & & & 1\end{pmatrix},\ \mathfrak{L} = \begin{pmatrix}1 & & & & \\ y_2 & 1 & & & \\ y_3 & y_2 & \ddots & & \\ \vdots & \ddots & \ddots & \ddots & \\ y_n & \cdots & y_3 & y_2 & 1\end{pmatrix}$$

where $\Lambda = \{x_2,\dots,x_{n-1},y_2,\dots,y_{n-1}\}$ is a set of algebraically independent indeterminates over K.

THEOREM 2.3. *Let* $A \in \mathsf{R}^{n\times n}$ *have rank* r *and* $\mathfrak{B} = \mathfrak{U}A\mathfrak{L} \in \mathsf{R}[\Lambda]^{n\times n}$. *For* $1 \le k \le r$ *we have* $\mathrm{cont}(\mathfrak{B}\binom{1\dots k}{1\dots k}) \sim d_k$, *where* d_k *is the* k*th determinantal divisor of* A *and* $\mathrm{cont}(\mathfrak{B}\binom{1\dots k}{1\dots k})$ *is the content (GCD of all non-zero coefficients) of* $\mathfrak{B}\binom{1\dots k}{1\dots k}$.

PROOF. Using a Binet-Cauchy minor expansion (see Gantmacher 1990, p. 9), we have

$$\mathfrak{B}\begin{pmatrix}1\dots k\\ 1\dots k\end{pmatrix} = \sum_{\sigma,\tau\in\mathcal{C}_k^n} \mathfrak{U}\begin{pmatrix}1\dots k\\ \sigma\end{pmatrix}\mathfrak{L}\begin{pmatrix}\tau\\ 1\dots k\end{pmatrix}\cdot A\begin{pmatrix}\sigma\\ \tau\end{pmatrix}.$$

Under the variable ordering $x_2 < \cdots < x_n$ and $y_2 < \cdots < y_n$, Kaltofen & Saunders (1991) show that the lexicographically smallest term of $\mathfrak{U}\binom{1\dots k}{\sigma}$ and $\mathfrak{L}\binom{\tau}{1\dots k}$ are unique to this choice of σ,τ. Thus the polynomials $f_{\sigma,\tau} = \mathfrak{U}\binom{1\dots k}{\sigma}\mathfrak{L}\binom{\tau}{1\dots k} \in \mathsf{R}[\Lambda]$ are linearly independent over K, and in fact over any quotient field $\mathsf{R}/p\mathsf{R}$ for any prime $p \in \mathsf{R}$. Let p be a prime in R and $l = \mathrm{ord}_p(d_k)$, the order of p in d_k. Clearly, $p^l \mid \mathrm{cont}(\mathfrak{B}\binom{1\dots k}{1\dots k})$. Suppose $p^{l+1} \mid \mathrm{cont}(\mathfrak{B}\binom{1\dots k}{1\dots k})$. Then

$$\sum_{\sigma,\tau\in\mathcal{C}_k^n} \mathfrak{U}\begin{pmatrix}1\dots k\\ \sigma\end{pmatrix}\mathfrak{L}\begin{pmatrix}\tau\\ 1\dots k\end{pmatrix}\cdot A\begin{pmatrix}\sigma\\ \tau\end{pmatrix}/p^l$$

$$= \sum_{\sigma,\tau\in\mathcal{C}_k^n} f_{\sigma,\tau}\cdot A\begin{pmatrix}\sigma\\ \tau\end{pmatrix}/p^l \equiv 0 \bmod p.$$

This implies the $f_{\sigma,\tau}$'s are linearly dependent modulo p or that $A\binom{\sigma}{\tau} \equiv 0 \bmod p^{l+1}$ for all $\sigma,\tau \in \mathcal{C}_k^n$. The latter statement is false by our definition of l, and the former leads to a contradiction. Thus $\mathrm{ord}_p d_k = \mathrm{ord}_p \mathrm{cont}(\mathfrak{B}\binom{1\dots k}{1\dots k})$ for all $p \in \mathsf{R}$, whence $d_k \sim \mathrm{cont}(\mathfrak{B}\binom{1\dots k}{1\dots k})$. □

We can use the above theorem to precondition a matrix into Smith dominant form with high probability. We will employ the "Zippel-Schwartz" lemma to bound the probability of obtaining a zero of a multi-variate polynomial:

FACT 2.4 *(Zippel 1979, Schwartz 1980). Assume* $f \in \mathsf{D}[x_1,\dots,x_k]$ *is non-zero,* D *an integral domain, and* $\mathcal{V}$ *a finite subset of* D. *Suppose elements* $a_1,\dots,a_k$ *are randomly and uniformly chosen from* $\mathcal{V}$. *Then* $\mathrm{Prob}\{f(a_1,\dots,a_k) = 0 : a_1,\dots,a_k \in \mathcal{V}\} \le \deg(f)/\#\mathcal{V}$.

THEOREM 2.5. *Let* $A \in \mathsf{R}^{n\times n}$ *with rank* r *and determinantal divisors* $d_1,\dots d_r \in \mathsf{R}$. *Let* $p \in \mathsf{R}$ *a prime in* R *and* $\mathcal{V}$ *a finite subset of* R *whose elements are in distinct cosets modulo* p. *Suppose* $u_2,\dots,u_n$, $l_2,\dots,l_n$ *are chosen randomly and uniformly from* $\mathcal{V}$ *and we construct* $B = UAL$, *where*

$$U = \begin{pmatrix}1 & u_2 & u_3 & \cdots & u_n\\ & 1 & u_2 & \ddots & \vdots\\ & & \ddots & \ddots & \vdots\\ & & & 1 & u_2\\ & & & & 1\end{pmatrix},\ L = \begin{pmatrix}1 & & & & \\ l_2 & 1 & & & \\ l_3 & l_2 & \ddots & & \\ \vdots & \ddots & \ddots & \ddots & \\ l_n & \cdots & l_3 & l_2 & 1\end{pmatrix} \tag{2.1}$$

Then

$$\mathrm{Prob}\Big\{\mathrm{ord}_p B\begin{pmatrix}1\dots k\\ 1\dots k\end{pmatrix} = \mathrm{ord}_p d_k \quad \forall k : 1 \le k \le r\Big\} \ge 1 - \frac{r(r+1)}{\#\mathcal{V}}.$$

PROOF. For any k,

$$B\begin{pmatrix}1\dots k\\ 1\dots k\end{pmatrix} = d_k \cdot f_k(u_2,\dots,u_n,l_2,\dots,l_n)$$

for some $f_k \in \mathsf{R}[x_2,\dots,x_n,y_2,\dots,y_n]$ with content 1 and degree $2k$ by Theorem 2.3. Thus p has the same order in $B\binom{1\dots k}{1\dots k}$ as in d_k if and only if $f_k(u_2,\dots,u_n,l_2,\dots,l_n) \not\equiv 0 \bmod p$. Since all elements of $\mathcal{V}$ are in distinct cosets modulo p, by Fact 2.4, $f_k(u_2,\dots,u_n,l_2,\dots,l_n) \equiv 0 \bmod p$ with probability at most $2k/\#\mathcal{V}$. Thus the probability of $f_k(u_2,\dots,u_n,l_2,\dots,l_n) \equiv 0 \bmod p$ for any $1 \le k \le r$ is at most $\sum_{1\le k\le r}(2k)/\#\mathcal{V} = r(r+1)/\#\mathcal{V}$. □

The following simple lemma allows us to solve a preconditioned system to obtain a solution to the original system.

LEMMA 2.6. *Let* $A \in \mathsf{R}^{n\times n}$ *and* $w \in \mathsf{R}^{n\times 1}$. *Let* $U, L \in \mathsf{R}^{n\times n}$ *with* $\det U, \det L \in \mathsf{R}^*$ *and* $B = UAL$. *Then* $\bar{v} \in \mathsf{R}^{n\times 1}$ *is a solution to* $B\bar{v} = Uw$ *if an only if* $v = L\bar{v}$ *is a solution to* $Av = w$.

PROOF. For the forward direction, assume $\bar{v}$ is a solution to $B\bar{v} = Uw$. Then

$$B\bar{v} = Uw \Longrightarrow UAL\bar{v} = Uw \Longrightarrow AL\bar{v} = w \Longrightarrow Av = w,$$

since U is invertible in $\mathsf{R}^{n\times n}$. Conversely, if $AL\bar{v} = w$ then $UAL\bar{v} = UAL\bar{v} = B\bar{v} = Uw$. □

Localizations of $\mathbb{Z}$ and $\mathbb{Q}$

It will be convenient to consider the localizations of $\mathbb{Q}$ and algebraic number fields at a prime p. We identify the p-adic integers $\mathbb{Z}_p$ and p-adic rationals $\mathbb{Q}_p$ with the (infinite) Laurent series

$$\mathbb{Z}_p = \left\{ \sum_{0 \le i < \infty} a_i p^i : a_i \in \{0, \ldots, p-1\} \right\},$$

$$\mathbb{Q}_p = \left\{ \sum_{m \le i < \infty} a_i p^i : a_i \in \{0, \ldots, p-1\},\ m \in \mathbb{Z} \right\},$$

under the usual arithmetic. A useful reference for general localizations is Lang (1986), and for p-adic numbers and analysis is Cassels (1986). Clearly $\mathbb{Z}_p \subseteq \mathbb{Q}_p$ and $\mathbb{Z} \subseteq \mathbb{Z}_p$. Also, if $v \in \mathbb{Z}$ is relatively prime with p then $1/v \in \mathbb{Z}_p$ by Hensel's Lemma (essentially p-adic Newton iteration – see Cassels (1986), Lemma 3.1). Since any element in $\mathbb{Q}$ can be written as $p^e u/v$, where $e \in \mathbb{Z}$, $u, v \in \mathbb{Z}$ and $\gcd(v, p) = 1$, we see that $\mathbb{Q} \subseteq \mathbb{Q}_p$. If $a = \sum_{m \le i < \infty} a_i p^i \in \mathbb{Q}_p$ for $a_i \in \{0, \ldots, p-1\}$ and $a_m \neq 0$, we define the *p-adic order* of a as $\mathrm{ord}_p(a) = m$ and the *p-adic norm* of a as $|a|_p = p^{-m}$ with $|0|_p = 0$. Thus $\mathbb{Z}_p = \{a \in \mathbb{Q}_p : |a|_p \le 1\}$.

Parallel modular computation over $\mathbb{Q}$

We next summarize for convenience a standard homomorphic scheme for parallel computing over $\mathbb{Q}$ (see Wang *et al.* 1982, Collins & Encarnación 1995). Let $\Psi : \mathbb{Q}^s \to \mathbb{Q}^t$ be a function we wish to compute and suppose that we know a quickly computable ("upper bound") function $\tau : \mathbb{Q}^s \to \mathbb{R}$ such that $\tau(\bar{x}) \ge \max\{\|\bar{x}\|, \|\Psi(\bar{x})\|\}$; the cost of computing τ is assumed to be dominated by that of other computations. Suppose also that for all primes $p \in \mathbb{Z}$, except for those in a finite set $\mathcal{B} \subset \mathbb{Z}$, we can compute $\Psi(\bar{x}) \bmod p$ from input $(\bar{x} \bmod p)$ with $O(\psi(s))$ operations in $\mathbb{F}_p$; when $p \in \mathcal{B}$ we can report this fact in the same amount of time. For convenience we will assume that $\#\mathcal{B} = (\log(\tau(\bar{x}))^{O(1)}$.

Following standard practice, we first construct a set $\mathcal{P} \subseteq \mathbb{Z}$ of sufficiently many small primes. We then compute $\Psi(\bar{x}) \bmod p$ for randomly chosen $p \in \mathcal{P}$, rejecting bad primes as we encounter them. Finally, when the product of the good primes chosen is at least $2\tau(\bar{x})^2$, we recover the solution by the Chinese remainder theorem and integer Padé approximation (this is sufficiently many to recover numerator, denominator and sign). See Wang *et al.* (1982). We crudely estimate that at least $\rho \le \log_2(2\tau(\bar{x})^2)$ good primes are required, though much better estimates are easily computed at run-time.

It is also convenient to allow for an n-point FFT to be performed efficiently (so we may practically multiply polynomials of degree up to n with $O(n \log n)$ operations). To facilitate this, we choose primes p such that $2^l \mid (p-1)$, where $l \ge \lceil \log_2 n \rceil$. By Dirichlet's density theorem on primes in an arithmetic progression, it is easily derived that we can efficiently construct a set $\mathcal{P}$ with $\#\mathcal{P} \ge 2(\#\mathcal{B}) + \rho$ such that $\log p = O(\log n + \log(\#\mathcal{B}) + \log\log\tau(\bar{x}))$ for all $p \in \mathcal{P}$ (see, e.g., Giesbrecht 1996, Section 3.2). For notational convenience we assume that $n = s^{O(1)}$. From a practical point of view, primes of this size should fit into a single (32-bit or 64-bit) machine word, and operations modulo such a prime will have constant cost.

A randomly chosen prime (without replacement) will be bad with probability at most 1/2. Thus we expect to compute $\Psi(\bar{x}) \bmod p$ for $2\rho = O(\log(\tau(\bar{x})))$ primes p, and the computation in $\mathbb{F}_p$ requires $O(\psi(s) \cdot (\log n + \log(\#\mathcal{B}) + \log\log\tau(\bar{x}))^2)$ bit operations. Reduction of $\bar{x} \bmod p$ for the used primes $p \in \mathcal{P}$ requires $O\tilde{}(s \log\|\bar{x}\| \cdot \log(\tau(\bar{x})))$ bit operations and recovery of the final integer solution require $O\tilde{}(t \cdot \mathrm{M}(\log\tau(\bar{x})))$ bit operations; see Wang *et al.* (1982).

We summarize the sequential cost in the following theorem.

THEOREM 2.7. *We can construct a probabilistic algorithm which on any input $\bar{x} \in \mathbb{Q}^s$ computes $\Psi(\bar{x}) \in \mathbb{Q}^t$. The algorithm requires an expected number of $O\tilde{}(\psi(s) \cdot \log(\tau(\bar{x})) + s\log(\|\bar{x}\|)\log(\tau(\bar{x})) + t \cdot \mathrm{M}(\log(\tau(\bar{x}))))$ bit operations. We may assume in our cost function ψ the availability of a practical n-point FFT at cost $O(n \log n)$, where $n = s^{O(1)}$.*

The computation modulo individual primes is independent and hence can be parallelized in a straightforward manner. The three stages of the algorithm, (i) reduction modulo the prime base, (ii) local computation, and (iii) recovery of global solutions, are analysed separately.

THEOREM 2.8. *We can construct a probabilistic algorithm which on any input $\bar{x} \in \mathbb{Q}^s$ computes $\Psi(\bar{x}) \in \mathbb{Q}^t$ which runs in parallel on N processors:*

(i) for $N \le s\rho$, we can reduce $\bar{x} \bmod p$ for the expected number of ρ primes used from $\mathcal{P}$ with $O\tilde{}(s\rho \log\|\bar{x}\|/N)$ bit operations carried out simultaneously by each processor;

(ii) for $N \le \rho$, we can compute $\Psi(\bar{x} \bmod p)$ for the expected number of ρ primes p from $\mathcal{P}$ in an expected number of $O\tilde{}(\phi(s) \cdot \rho/N)$ bit operations carried out simultaneously by each processor;

(iii) for $N \le t$ we can recover the solutions in $\mathbb{Q}^t$ from images modulo ρ good primes in an expected number of $O\tilde{}(t \cdot \mathrm{M}(\log(\tau(\bar{x})))/N)$ bit operations carried out simultaneously by each processor;

where $\rho = \log(\tau(\bar{x}))$. We may assume in our cost function ψ the availability of a practical n-point FFT at cost $O(n \log n)$, where $n = s^{O(1)}$.

3 Finding rational solutions with smooth denominators

We present our algorithm for finding integer solutions to systems of integer equations in two parts. The first part is the basic algorithm and finds a rational solution whose denominators are λ-smooth, that is, only primes less than or equal to λ divide the denominators of the coefficients. This algorithm appears to work well even with $\lambda = 1$ (and hence obtains integer solutions), but unfortunately we can only prove it for $\lambda \ge 2r(r+1)$, where r is the rank of the input matrix. A modification is then presented in Sections 5 to deal with the remaining case in a theoretically sound way at an additional logarithmic factor in the cost.

For $v \in \mathbb{Q}^{n \times 1}$ we define the *denominator* of v to be $\mathrm{denom}(v) = \min\{d \in \mathbb{Z}_{>0} : dv \in \mathbb{Z}^{n \times 1}\}$, the least common multiple of all the denominators of the coefficients (in lowest terms) of v. For any $\lambda > 0$, we say that an integer b is λ-*smooth* if all prime factors of b are less than or equal to λ (or $b = \pm 1$ if $\lambda = 1$).

Our algorithm also has two additional parameters aside from A and w:

- $\lambda > 0$: the returned solution should have a denominator which is λ-smooth. By setting $\lambda = 1$ we achieve integer solutions.
- $\epsilon > 0$: an error tolerance. If it is reported that "No Integer Solution Exists" then this is correct with probability at least $1-\epsilon$. If a solution is returned, it is always correct. The need for such an error tolerance parameter ϵ is also present in the underlying Wiedemann and Block-Wiedemann algorithms for solving sparse singular systems over a field.

Algorithm: `SmoothSolver`
Input: – $A \in \mathbb{Z}^{n\times n}$ and $w \in \mathbb{Z}^{n\times 1}$;
– a smoothness bound $\lambda > 0$;
– an error tolerance $\epsilon > 0$;
Output: – $v \in \mathbb{Q}^{n\times 1}$ where $\mathrm{denom}(v)$ is λ-smooth, or a report "No Integer Solution Exists";
(1) Compute $r := \mathrm{rank}(A)$, correct with probability at least $1 - \epsilon/2$;
(2) $\beta := 2r(r+1)$; $\mathcal{V} := \{-\beta/2, \ldots, \beta/2\} \subseteq \mathbb{Z}$;
$g := 0$;
(3) For $b := 1$ to $\lceil 1 + \log_2(1/\epsilon)\rceil$ Do
(4) For $i := 0$ to $s := \lceil 1 + \log_2(\log_\lambda(n^2\beta^2\|A\|))\rceil$ Do
(5) Choose random $u_2, \ldots, u_n, l_2, \ldots, l_n \in \mathcal{V}$;
"Build" a black box for $B = UAL$ where U, L are as in (2.1); Let $B_r = B\left[\begin{smallmatrix}1\ldots r\\ 1\ldots r\end{smallmatrix}\right]$;
$\bar{w} := Uw = (\bar{w}_1, \ldots, \bar{w}_n)^t \in \mathbb{Z}^{n\times 1}$;
(6) Solve $B_r\bar{v} = (\bar{w}_1, \ldots, \bar{w}_r)^t$ for $\bar{v} = (\bar{v}_1, \ldots, \bar{v}_r) \in \mathbb{Q}^{r\times 1}$ with black box for B
If B_r is singular, goto (5);
(7) $v^{(i)} := L(\bar{v}_1, \ldots, \bar{v}_r, 0, \ldots, 0)^t \in \mathbb{Q}^{n\times 1}$;
$\delta_i := \mathrm{denom}(v^{(i)})$;
If $Av^{(i)} \neq w$ then report "No solution to Diophantine system exists";
(8) $g := \gcd(g, \delta_i)$;
End For;
End For;
If g is λ-smooth Then
(9) Find $\gamma_0, \ldots, \gamma_s \in \mathbb{Z}$ such that $\sum_{0\le i\le s}\gamma_i\delta_i = g$;
(10) Return $v := (1/g)\cdot\sum_{0\le i\le s}\gamma_i\delta_i \cdot v^{(i)}$;
Else Report "No solution to Diophantine system exists".
End If;

THEOREM 3.1. *The algorithm* `SmoothSolver` *works as specified. Suppose the input matrix $A \in \mathbb{Z}^{n\times 1}$ has (unknown) rank r.*
(i) If a solution $v \in \mathbb{Q}^{n\times 1}$ is returned, it is always correct;
(ii) $\log\|v\| = \tilde{O}(r\log n + r\log\|A\| + \log\|w\|)$;
(iii) if $\lambda \ge 2r(r+1)$ and a λ-smooth solution exists to the system, a λ-smooth solution is found with probability at least $1-\epsilon$.

PROOF. The rank of A is obtained with probability at least $1-\epsilon/2$ via the algorithm of Kaltofen & Saunders (1991) as generalized to integer matrices in Giesbrecht (1996).

For part (i), we note that $A(\delta_i v^{(i)}) = \delta_i w$ for $0 \le i \le s$. Thus

$$Av = A\left((1/g)\sum_{1\le i\le s}\gamma_i\delta_i v^{(i)}\right) = \left((1/g)\sum_{1\le i\le s}\gamma_i\delta_i\right)w = w$$

and $\mathrm{denom}(v) = g$, which is λ-smooth by construction.

For parts (ii) and (iii), first consider an iteration of the inner loop (4)-(8). We have

$$\|B\| \le n^2\|U\|\cdot\|A\|\cdot\|L\| \le n^2\beta^2\|A\| = O(n^2r^4\|A\|),$$
$$\|Uw\| \le n\|U\|\cdot\|w\| \le n\beta\|w\| = O(nr^2\|w\|).$$

Applying Hadamard's bound and Cramer's rule we find

$$\begin{aligned}\log_2|\delta_i| &= O(r\log n + r\log\|A\|),\\ \log\|\bar{v}\| &= O(r\log r + r\log\|B_r\| + \log\|\bar{w}\|)\\ &= O(r\log n + r\log\|A\| + \log\|w\|),\\ \log\|v^{(i)}\| &= O(r\log n + r\log\|A\| + \log\|w\|),\end{aligned}$$

for $0 \le i \le s$. Also, $\log_\lambda\delta_i \le \log_\lambda(n^2\beta^2\|A\|)$ is a (crude) upper bound on the number of primes greater than λ which can divide δ_i.

Assume that the rank r is calculated correctly in step (1). Since $B\left(\begin{smallmatrix}1\ldots r\\ 1\ldots r\end{smallmatrix}\right)$ is a non-zero polynomial in $u_2, \ldots, u_n$, $l_2, \ldots, l_n$ of degree $2r$, B_r is non-singular with probability at least $1 - 2r/(2r(r+1)) = r/(r+1)$ by Fact 2.4. Thus we expect to execute steps (5) and (6) a constant number of times for each iteration of the inner For loop. If a solution to $Av = w$ exists over $\mathbb{Z}$ it certainly exists over $\mathbb{Q}$, and by Lemma 2.6 $v^{(i)}$ will be such a solution (since $\mathbb{Q}$ is a PID). Once the GCD of the denominators is λ-smooth, we execute steps (9) and (10). Step (9) is probably best done in practice by the algorithm of Majewski & Havas (1995), but for a simpler analysis here we employ the algorithm Iliopolous (1989) which finds $\gamma_0, \ldots, \gamma_s \in \mathbb{Z}$ such that $\log|\gamma_i| = O(\log\max_{0\le i\le s}|\delta_i|\cdot\log s)$. The constructed v thus satisfies

$$\begin{aligned}\log\|v\| &= \log\left(\max\left\{|g|, \sum_{0\le i\le s}\gamma_i\|\delta_i v^{(i)}\|\right\}\right)\\ &= O((r\log n + r\log\|A\|)\\ &\qquad\cdot(\log\log\log n + \log\log\log\|A\|) + \log\|w\|)\end{aligned}$$

or $\tilde{O}(r\log n + r\log\|A\| + \log\|w\|)$, which proves (ii).

To prove (iii) assume that an λ-smooth solution does indeed exist. We show that with each iteration of the outer For loop, the algorithm finds such a solution with probability at least $1/2$. Let $p > \lambda$ be a prime dividing δ_0. Since $\#(\mathcal{V} \bmod p) \ge 2r(r+1)$, by Theorem 2.5,

$$\mathrm{Prob}\left\{\mathrm{ord}_p B\left(\begin{smallmatrix}1\ldots k\\ 1\ldots k\end{smallmatrix}\right) = \mathrm{ord}_p d_k \quad \forall k : 1 \le k \le r\right\} \ge 1/2.$$

If indeed $\mathrm{ord}_p B\left(\begin{smallmatrix}1\ldots k\\ 1\ldots k\end{smallmatrix}\right) = \mathrm{ord}_p d_k$ for all k $(1 \le k \le r)$, the image of B in $\mathbb{Z}_p^{n\times n}$ is Smith dominant. Thus by Theorem 2.1, the image of $\bar{v}$ in $\mathbb{Q}_p^{r\times 1}$ lies in $\mathbb{Z}_p^{r\times 1}$ and the image of $v^{(i)}$ in $\mathbb{Q}_p^{n\times 1}$ lies in $\mathbb{Z}_p^{n\times 1}$, whence $p \nmid \mathrm{denom}(v^{(i)})$. Thus, the probability that $p \mid \mathrm{denom}(v^{(i)})$ for all $1 \le i \le s = 1 + \log_2(\log_\lambda(n^2\beta^2\|A\|))$ is at most $(1/2)\cdot 1/\log_\lambda(n^2\beta^2\|A\|)$. The probability this is true for *any* prime $p \ge \lambda$ dividing δ_0 is thus at most $1/2$, since there are at most $\log_\lambda(n^2\beta^2\|A\|)$ such primes. By executing the outer For loop $1 + \log_2(1/\epsilon)$ times we ensure that if a solution exists (and we obtained the rank correctly), we will find a solution with probability at least $1 - \epsilon/2$. Since the rank is correct with probability $1 - \epsilon/2$, the theorem follows. □

We will employ the Wiedemann and Block-Wiedemann linear equation solvers over a finite field, as developed in Wiedemann (1986), Kaltofen & Saunders (1991) and Coppersmith (1994), and analysed in Kaltofen (1995).

FACT 3.2. *Suppose we are given a black box for a non-singular matrix $B \in \mathsf{K}^{r\times r}$ and vector $\bar{w} \in \mathsf{K}^{r\times 1}$ over a field K with at least $16r^2$ elements. On a network of $N \le r$ processors we can solve $B\bar{v} = \bar{w}$ for $\bar{v} \in \mathsf{K}^{r\times 1}$ with an expected $O(r/N)$ matrix-vector products by B and $O(r^2 \log r)$ operations in K, executed simultaneously on each processor (assuming an r-point FFT is available in K). Each processor requires additional storage for $O(r)$ elements of K (not including a possibly shared image of B).*

This algorithm can be applied to non-singular rational matrices as a direct application of the techniques of Theorem 2.8. See Kaltofen & Saunders (1991) for a different approach.

THEOREM 3.3. *Suppose we are given a black box for a non-singular matrix $B \in \mathbb{Z}^{r\times r}$ and vector $\bar{w} \in \mathbb{Z}^{r\times 1}$ and wish to solve $B\bar{v} = \bar{w}$ for $\bar{v} \in \mathbb{Q}^{r\times 1}$. Let $\varrho = r\log\|B\| + r\log r + \log\|\bar{w}\|$. On a network of $N \le r\varrho$ processors we can solve for $\bar{v}$ with an expected $O^{\sim}(r\varrho/N)$ matrix-vector products by B modulo (single-word) primes with $O(\log r + \log\log(\|B\| + \|\bar{w}\|))$ bits. An additional $O^{\sim}(r^2 + r\,\mathrm{M}(\varrho)/\min(r,N))$ bit operations is executed simultaneously by each processor. Each processor requires additional storage for $O(r\varrho/\min(r,N))$ words (not including possibly shared images of B modulo single-word primes).*

PROOF. To apply Theorem 2.8, we need only note that the only bad primes are those which divide the determinant of B, and there are at most $O(r(\log r + \log\|B\|))$. It is also generally convenient to eliminate small primes (say those less than $16r^2$) to allow the Wiedemann and Block-Wiedemann algorithms to (provably) work without the use of field extensions. □

We are parallelizing the linear solver in two different ways. First, we break the problem into an expected ϱ independent problems modulo ϱ distinct primes. Second, for each prime we use up to r processors to solve a non-singular system over a finite field via the Block-Wiedemann algorithm. Here ϱ is a crude upper bound on the logarithm of the absolute value of the coefficients in the unique solution.

A potential bottleneck is the recovery of rational solutions: each of the r entries in the solution vectors is recovered independently from its modular images on up to r processors. If $\mathrm{M}(\varrho) = \varrho^2$ then the recovery phase potentially dominates the overall cost, at least in theory.

THEOREM 3.4. *Let $A \in \mathbb{Z}^{n\times n}$ of (unknown) rank $r \le m$, $w \in \mathbb{Z}^{n\times 1}$, $\lambda > 0$ and $\epsilon > 0$ be as in the input to* `SmoothSolver`. *Let $\varrho = r\log\|A\| + r\log r + \log\|w\|$ and suppose we are computing on a network of $N \le r\varrho$ processors.*

(i) If a λ-smooth solution $v \in \mathbb{Q}^{n\times 1}$ to $Av = w$ exists, `SmoothSolver` *finds one with an expected number of $O^{\sim}(r\varrho/N)$ matrix-vector products by A modulo primes with $O(\log n + \log\varrho)$ bits. An additional $O^{\sim}(r^2 + rn\varrho/N + n\,\mathrm{M}(\varrho)/\min(n,N))$ bit operations is executed simultaneously by each processor.*

(ii) If no λ-smooth solution $v \in \mathbb{Q}^{n\times 1}$ to $Av = w$ exists, `SmoothSolver` *requires an expected number of $O^{\sim}((r\varrho/N)\cdot\log(1/\epsilon))$ matrix-vector products by A modulo primes with $O(\log n + \log\varrho)$ bits. An additional $O^{\sim}((r^2 + rn\varrho/N + n\,\mathrm{M}(\varrho)/\min(n,N))\cdot\log(1/\epsilon))$ bit operations is executed simultaneously by each processor. Each processor requires storage for an additional $O(n + n\varrho/\min(n,N))$ words (not including possibly shared images of A modulo single-word primes).*

PROOF. The inner For loop iterates $O(\log\log n + \log\log\|A\|)$ times. If a solution exists, we expect the outer loop to iterate twice. If no solution exists, the outer For loop iterates $O(\log(1/\epsilon))$ times.

Each evaluation of the black box for $y \mapsto B_r y$ where $y = (y_1,\ldots,y_r)^t \in \mathbb{Z}^{r\times 1}$ is performed by evaluating $UAL(y_1,\ldots,y_r,0,\ldots,0)^t = (z_1,\ldots,z_n)^t$, and returning $(z_1,\ldots,z_r)^t = B_r y$. Pre-multiplication by a unit triangular Toeplitz matrix takes $O(n\log n)$ operations in the ground field assuming an n-point FFT (see Kailath 1980). Thus each matrix-vector product by B_r requires one black box evaluation of A modulo primes with $O(\log r + \log\log(\|B\| + \|\bar{w}\|))$ or $O(\log n + \log\varrho)$ bits, plus $O(n\log n)$ additional operations modulo primes of this same size. The linear system $B_r\bar{v} = \bar{w}$ in step (6) is then solved using the Block-Wiedemann method described in Theorem 3.3. The algorithm Iliopolous (1989), which finds $\gamma_0,\ldots,\gamma_c \in \mathbb{Z}$, requires $O((\log\log n + \log\log\|A\|)\cdot(\log r + \log\log n + \log\log\|A\|)\cdot \mathrm{M}(r\log n + r\log\|A\|))$ or $O^{\sim}(\mathrm{M}(r\log\|A\|))$ bit operations, which we will execute on a single processor. Finally, to recover the solutions in $\mathbb{Z}^{n\times 1}$ requires $O^{\sim}(n\,\mathrm{M}(\varrho))$ for the Chinese remainder algorithm and integer Padé approximation on each coefficient (see Bach & Shallit 1996). □

4 Constructing orders of number fields with selected inert primes

The main theoretical hurdle to be overcome in finding Diophantine solutions (instead of just solutions with smooth denominators) is that the Zippel-Schwartz lemma fails us for small primes dividing the determinantal divisors. Our solution is to work in a collection of small extension rings over $\mathbb{Z}$. Recall that an *order* of a number field is a submodule of the ring of integers of a number field (see, e.g., Cassels (1986), Chapter 10) and contains $\mathbb{Z}$ as a subring. In this section we describe how to construct orders of number fields such that certain primes remain inert (i.e., the ideals they generate remain prime) of some prescribed degree. We also discuss some useful properties of the p-adic integral closures of these orders which will be important in the next section.

For any $\eta \in \mathbb{N}$, and $s \in \mathbb{R}_{>0}$ define $\mathcal{M}(\eta;s) = \{g \in \mathbb{Z}[z] : g \text{ monic}, \deg g = \eta, \|g\| \le s\}$.

Algorithm: `BuildOrders`
Input: – $\eta \in \mathbb{Z}$ and primes $p_1,\ldots,p_\kappa \in \{2,\ldots,\tau\}$;
Output: – a set $G \subseteq \mathcal{M}(\eta;\eta\tau)$ such that for each $i \in \{1,\ldots,\kappa\}$, there exists a $\Gamma_i \in G$ with $\Gamma_i \bmod p_i$ irreducible in $\mathbb{F}_{p_i}[z]$.

(1) Repeat
(2) Let $\mathcal{P} := \{1,\ldots,\kappa\}$; $G := \{\}$;
(3) Let $l := 8\eta\log(2\kappa)$;
(4) For $j := 1$ to l do
(5) Choose a random $h_j \in \mathcal{M}(\eta;\eta\tau)$;
(6) For all $i \in \mathcal{P}$ do
(7) If $h_j \bmod p_i \in \mathbb{F}_{p_i}[z]$ is irreducible in $\mathbb{F}_{p_i}[z]$ Then $\mathcal{P} := \mathcal{P} \setminus \{i\}$; $G := G \cup \{h_j\}$;

End For;
End For;
Until $\mathcal{P} = \{\}$;
(8) Return G.

THEOREM 4.1. *The algorithm* `BuildOrders` *always produces the correct results as described and requires an expected number of* $O((\eta^3 + \eta^2 \log \tau) \cdot \kappa\eta \log \kappa \cdot \log^2 \tau)$ *bit operations. Also,* $\#G = O(\eta \log \kappa)$ *and for all* $g \in G$, $\|g\| \leq \eta\tau$.

PROOF. First, for any prime p and $\eta \in \mathbb{N}$, define

$$\mathcal{M}_p(\eta) = \{g \in \mathbb{F}_p[z] : g \text{ monic}, \deg g = \eta\} = \mathcal{M}(\eta; \eta\tau) \bmod p.$$

For a randomly chosen $h \in \mathcal{M}_p(\eta)$ and $\eta \geq 3$, the probability that h is irreducible in $\mathbb{F}_p[z]$ is at least

$$\frac{1}{\eta} \sum_{d \mid \eta} \mu(d) q^{\eta/d} \geq \frac{p^\eta}{\eta} - \frac{p(p^{\eta/2} - 1)}{\eta(p-1)} \geq \frac{3p^\eta}{4\eta}$$

by Lidl & Niederreiter (1983), Exercise 3.27. If we choose h randomly and uniformly from $\mathcal{M}(\eta; \eta\tau)$, $h \bmod p$ falls into any particular residue class in $\mathcal{M}_p(\eta)$ with probability at least $(\lfloor (2\eta\tau + 1)/p \rfloor / (2\eta\tau + 1))^\eta \geq (1/p - 1/(2\eta\tau + 1))^\eta$. The probability that $h \bmod p$ is irreducible in $\mathbb{F}_p[z]$ is at least

$$\left(\frac{1}{p} - \frac{1}{2\eta\tau + 1}\right)^\eta \cdot \frac{3p^\eta}{4\eta} = \frac{3}{4\eta} \cdot \left(1 - \frac{p}{2\eta\tau + 1}\right)^\eta > \frac{3}{4\eta} \cdot \left(1 - \frac{1}{2\eta}\right)^\eta \geq \frac{3}{8\eta}.$$

For any fixed prime $p_i \in \{p_1, \ldots, p_\kappa\}$, the probability that in a single iteration of the outer loop steps (2)-(7), for all random choices in step (5), we do not choose an $h_j \in \mathcal{M}(\eta; \eta\tau)$ with $h_j \bmod p_i$ irreducible in $\mathbb{F}_{p_i}[z]$ is at most $(1 - 3/(8\eta))^l$. The probability that there exists *any* prime $p_i \in \{p_1, \ldots, p_\kappa\}$ for which we do not choose such an h_j is thus at most $\kappa \cdot (1 - 3/(8\eta))^l < 1/2$ by our choice of $l = 8\eta \log(2\kappa)$.

For each random choice of $h_i \in \mathcal{M}(\eta; \eta\tau)$ the inner loop of steps (6)–(7) can be accomplished with an expected number of $O((\eta^3 + \eta^2 \log \tau) \cdot \kappa \cdot \log^2 \tau)$ bit operations using Berlekamp's (1970) factoring algorithm, and this loop is executed $l = 8\eta \log(2\kappa)$ times per iteration of the outer loop. □

Heights and localizations of orders of number fields Let $\Gamma = \sum_{0 \leq i \leq \eta} \gamma_i z^i \in \mathbb{Z}[z]$ be monic and irreducible of degree η and $\theta = z \bmod \Gamma$, so $\mathbb{Z}[\theta] = \mathbb{Z}[z]/(\Gamma)$ is an order in $\mathbb{Q}(\theta)$ and a sub-order of the maximal order $\mathcal{O}$ (the ring of algebraic integers) in $\mathbb{Q}(\theta)$. Computationally we represent $\mathbb{Z}[\theta]$ with respect to the power basis $\{1, z, z^2, \ldots, z^{\eta-1}\}$, where elements are uniquely represented by an integer polynomial of degree less than η (under standard addition and multiplication of polynomials, reduced modulo Γ).

We define a *Height* function $\mathcal{H} : \mathbb{Q}(\theta) \to \mathbb{N}$ as follows. Let $\Theta \in \mathbb{Z}^{\eta \times \eta}$ be the companion matrix of Γ. For $a = \sum_{0 \leq i < \eta} a_i \theta^i \in \mathbb{Z}[\theta]$, define $\mathcal{H}(a) = \| \sum_{0 \leq i < \eta} a_i \Theta^i \|_\infty$. It is easily verified that

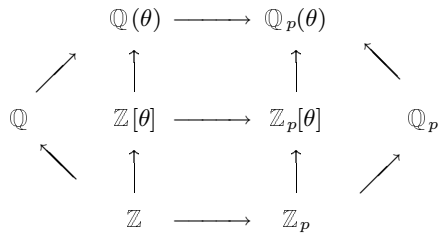
$$\mathcal{H}(a) \leq \begin{cases} |a| & \text{if } a \in \mathbb{Z}, \\ (\max_{0 \leq i < \eta} |a_i|) \cdot \eta(1 + \|\Gamma\|)^{\eta - 1} & \text{otherwise,} \end{cases}$$

and that for $a, b \in \mathbb{Z}[\theta]$, $\mathcal{H}(ab) \leq \mathcal{H}(a) \cdot \mathcal{H}(b)$ and $\mathcal{H}(a+b) \leq \mathcal{H}(a) + \mathcal{H}(b)$. Moreover, a can be represented as an integer polynomial of degree less than η with $O(\log \mathcal{H}(a))$ bits.

We represent an element $\alpha \in \mathbb{Q}(\theta)$ by $\alpha = a/b$, where $a \in \mathbb{Z}[\theta]$ as above and $b \in \mathbb{Z}$ is relatively prime to $\gcd(a_0, \ldots, a_{\eta-1})$. Define $\mathcal{H}(\alpha) = \max\{|b|, \mathcal{H}(a)\}$. It is easily verified that for $\alpha \in \mathbb{Q}$, $\mathcal{H}(1/\alpha) = \mathcal{H}(\alpha)$, while for general $\alpha \in \mathbb{Q}(\theta)$, $\mathcal{H}(1/\alpha) \leq \eta^\eta \mathcal{H}(a)^\eta$. We similarly extend $\|\cdot\|$ to matrices and polynomials over $\mathbb{Q}(\theta)$: for $B \in \mathbb{Q}(\theta)^{m \times n}$, $\|B\| = \max_{ij} \mathcal{H}(B_{ij})$ and for $g = \sum_{0 \leq i \leq m} b_i x^i \in \mathbb{Q}(\theta)[x]$, $\|g\| = \max_i \mathcal{H}(b_i)$.

Next suppose $p \in \mathbb{Z}$ is a prime such that $\Gamma \bmod p$ is irreducible in $\mathbb{F}_p[z]$. The prime p remains inert in the ring of integers $\mathcal{O}$ of $\mathbb{Q}(\theta)$, that is, the ideal $p\mathcal{O}$ is prime in $\mathcal{O}$. This also implies that $\mathbb{Z}[z]/(p, \Gamma) \cong \mathbb{F}_{p^\eta}$, the finite field with p^η elements. We can adjoin a root $\theta = (z \bmod \Gamma)$ of $\Gamma(z)$ to $\mathbb{Q}_p$ to obtain an extension field $\mathbb{Q}_p(\theta) \supseteq \mathbb{Q}_p$, called the localization of $\mathbb{Q}(\theta)$ at p. Similarly, we have $\mathbb{Z}_p[\theta]$, a ring extension of $\mathbb{Z}_p$ containing $\mathbb{Z}[\theta]$. $\mathbb{Z}_p[\theta]$ is easily shown to be a principal ideal domain (see Lang (1986), Section 2.1). It is also easily verified that $\mathbb{Z}_p[\theta]/(p, \Gamma) \cong \mathbb{F}_{p^\eta}$ (the *residue class field* of $\mathbb{Q}_p(\theta)$). Thus $[\mathbb{Z}_p[\theta] : \mathbb{Z}_p] = [\mathbb{Q}_p(\theta) : \mathbb{Q}_p] = \eta$ and $\{1, \theta, \theta^2, \ldots, \theta^{\eta-1}\}$ forms a $\mathbb{Z}_p$-basis for $\mathbb{Z}_p[\theta]$ and a $\mathbb{Q}_p$ basis for $\mathbb{Q}_p(\theta)$. We can extend the p-adic order and p-adic norm to $\mathbb{Q}_p(\theta)$ by letting $\mathrm{ord}_p(a) = \min\{\mathrm{ord}_p(a_i) : 0 \leq i < \eta\} \in \mathbb{Z}$ and $|a|_p = \max\{|a_i|_p : 0 \leq i < \eta\} \in \mathbb{R}_{\geq 0}$ for $a = \sum_{0 \leq i \leq \eta} a_i \theta^i \in \mathbb{Q}_p(\theta)$ (where $a_i \in \mathbb{Q}_p$). These definitions agree with the p-adic norm and order on $\mathbb{Q}_p$ on its embedding in $\mathbb{Q}_p(\theta)$. We then identify $\mathbb{Z}_p[\theta] = \{a \in \mathbb{Q}_p(\theta) : |a|_p \leq 1\}$.

In the language of p-adic analysis, $\mathbb{Q}_p(\theta)$ is the unique unramified extension field of degree η over $\mathbb{Q}_p$. $\mathbb{Z}_p[\theta]$ is the integral closure of $\mathbb{Z}_p$ in $\mathbb{Q}_p(\theta)$, that is, the elements of $\mathbb{Q}_p(\theta)$ which are roots of monic polynomials in $\mathbb{Z}_p[z]$. All this is in some sense made possible because p (or rather the principal ideal generated by p) remains prime in the ring of integers of $\mathbb{Q}(\theta)$. We obtain the following diagram of inclusions:



The utility in these definitions is in the following observation. Suppose we wish to evaluate a rational function $\Psi \in \mathbb{Z}(x_1, \ldots, x_n)$ (a quotient of integer polynomials) at a point $\bar{a} = (a_1, \ldots, a_n) \in \mathbb{Z}[\theta]^n$, say $b = \Psi(\bar{a}) \in \mathbb{Q}(\theta)$. Computationally b is represented by a polynomial $\sum_{0 \leq i < \eta} b_i z^i \in \mathbb{Q}[z]$. To show that a prime p does not divide any of the denominators of the b_i's, we can show that $b \in \mathbb{Z}_p[\theta]$. Since $\Psi \in \mathbb{Z}(x_1, \ldots, x_n) \subseteq \mathbb{Z}_p(x_1, \ldots, x_n)$ and $\bar{a} \in \mathbb{Z}^n \subseteq \mathbb{Z}_p^n$, we can view the computation as taking place over $\mathbb{Z}_p[\theta]$, which, unlike $\mathbb{Z}[\theta]$, is a PID. Obviously, this does not change the algorithm, only our perception of the space on which it operates.

5 Refining smooth solutions to Diophantine solutions

We can now present our algorithm `RefineToDiophantine` to refine a $\lambda = 2r(r+1)$-smooth solution into a Diophantine solution. The algorithm is very similar to `SmoothSolver`, but works in a series of orders of number fields of very small degree over $\mathbb{Z}$.

Algorithm: `RefineToDiophantine`

Input: – $A \in \mathbb{Z}^{n\times n}$, $r = \operatorname{rank} A$ and $w \in \mathbb{Z}^{n\times 1}$;
– $v^{(0)} \in \mathbb{Q}^{n\times 1}$ such that $Av^{(0)} = w$ and $\delta_0 = \operatorname{denom}(v^{(0)})$ is $2r(r+1)$-smooth;
– an error tolerance $\epsilon > 0$;

Output: – $v \in \mathbb{Z}^{n\times 1}$ such that $Av = w$ or a report "No Integer Solution Exists";

(1) Let $p_1, \ldots, p_\kappa \le 2r(r+1)$ be the primes dividing δ_0;
(2) Using `BuildOrders` on inputs $\eta = \lceil \log_2(2r(r+1)) \rceil$ and $p_1, \ldots, p_\kappa$, find a set $G = \{\Gamma_1, \ldots, \Gamma_l\} \subseteq \mathbb{Z}[z]$ 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 $\mathbb{F}_p[z]$;
Let $\theta_j = (z \bmod \Gamma_j)$;
(3) Let $g := \delta_0$;
(4) For $c := 1$ to $\lceil 1 + \log_2(1/\epsilon) \rceil$ While $g \neq 1$ Do
(5) For $i := 1$ to $s := \lceil 1 + \log_2(\kappa) \rceil$ Do
(6) For $j := 1$ to l Do
(7) Let $\mathcal{V}_j = \{\sum_{0\le k<\eta} a_k \theta_j^k : a_k \in \{0,1\}\} \subseteq \mathbb{Z}[\theta_j]$;
(8) Choose random $u_2, \ldots, u_n, l_2, \ldots, l_n \in \mathcal{V}_j$;
"Build" a black box for $B = UAL$ with U, L as in (2.1); Let $B_r = B\left[\begin{smallmatrix}1\ldots r\\ 1\ldots r\end{smallmatrix}\right]$;
Let $\bar{w} := Uw = (\bar{w}_1, \ldots, \bar{w}_n)^t \in \mathbb{Z}[\theta_j]^{n\times 1}$;
(9) Solve $B_r \bar{v} = (\bar{w}_1, \ldots, \bar{w}_r)^t$ for $\bar{v} = (\bar{v}_1, \ldots, \bar{v}_r)^t \in \mathbb{Q}(\theta_j)^{r\times 1}$;
If B_r is singular, goto (8);
(10) Let $\sum_{0\le k<\eta} v_k^{(i,j)} \theta^k := L(\bar{v}_1, \ldots, \bar{v}_r, 0, \ldots, 0)^t$, where $v_k^{(i,j)} \in \mathbb{Q}^{n\times 1}$ for $0 \le k < \eta$;
Let $v^{(i,j)} := v_0^{(i,j)} \in \mathbb{Q}^{n\times 1}$
$\delta_{i,j} := \operatorname{denom}(v^{(i,j)})$;
If $Av^{(i,j)} \neq w$ then report "No solution to Diophantine system exists";
(11) Let $g := \gcd(g, \delta_{i,j})$;
End For;
End For;
End For;
If $g = 1$ Then
(12) Find $\gamma_0, \gamma_{i,j} \in \mathbb{Z}$ for $1 \le i \le s$ and $1 \le j \le l$ such that $\gamma_0\delta_0 + \sum \gamma_{i,j}\delta_{i,j} = 1$;
(13) Return $v := \gamma_0\delta_0 v^{(0)} + \sum \gamma_{i,j}\delta_{i,j} v^{(i,j)} \in \mathbb{Z}^{n\times 1}$;
Else Report "No solution to Diophantine system exists".
End If;

THEOREM 5.1. *The algorithm* `RefineToDiophantine` *works as specified.*

(i) If a solution $v \in \mathbb{Z}^{n\times 1}$ is returned, it is always correct;
(ii) $\log\|v\| = O^{\sim}(r\log n + r\log\|A\| + \log\|w\|)$ when $\log\|v^{(0)}\|$ is of this same order of size;
(iii) If an integer solution exists to the system, a solution is found with probability at least $1-\epsilon$.

PROOF. The proof follows in much the same way as Theorem 3.1. For part (i), we note that $A(\delta_{i,j}v^{(i,j)}) = \delta_{i,j}w$ for $1 \le i \le s$ and $1 \le j \le l$. Thus

$$Av = A\left(\sum_{\substack{1\le i\le s\\ 1\le j\le l}} \gamma_{i,j}\delta_{i,j}\cdot v^{(i,j)}\right) = \left(\sum_{\substack{1\le i\le s\\ 1\le j\le l}} \gamma_{i,j}\delta_{i,j}\right)\cdot w = w$$

and $v \in \mathbb{Z}^{n\times 1}$

For parts (ii) and (iii), first consider an iteration of the inner loop (7)-(11). We have

$$\|B\| \le n^2\|U\|\cdot\|A\|\cdot\|L\| = O(n^2\cdot\|A\|\cdot(\eta(1+\|\Gamma\|)^{\eta-1})^2)$$
$$\|Uw\| \le n\|U\|\cdot\|w\| = O(n\|w\|\cdot\eta(1+\|\Gamma\|)^{\eta-1}).$$

Applying Hadamard's bound and Cramer's rule we find

$$\begin{aligned}
\log\|\det(B_r)\| &= O(r\log n + r\log\|A\| + r\eta\log\|\Gamma\|),\\
\log|\delta_{i,j}| &\le \log\|1/\det(B_r)\|\\
&= O(r\eta\log n + r\eta\log\|A\| + r\eta^2\log\|\Gamma\|),\\
\log\|\det(B_r)\bar{v}\| &= O(r\log n + r\log\|A\| + \log\|w\|\\
&\qquad + r\eta\log\|\Gamma\|),\\
\log\|v^{(i,j)}\| &= O(r\eta\log n + r\eta\log\|A\| + \log\|w\|\\
&\qquad + r\eta^2\log\|\Gamma\|),
\end{aligned}$$

for $1 \le i \le s$ and $1 \le j \le l$.

Since $B\left(\begin{smallmatrix}1\ldots r\\ 1\ldots r\end{smallmatrix}\right)$ is a non-zero polynomial in $u_2, \ldots, u_n$, $l_2, \ldots, l_n$ of degree $2r$, B_r is non-singular with probability at least $1 - 2r/(2r(r+1)) = r/(r+1)$ by Fact 2.4. Thus we expect to execute steps (8) and (9) a constant number of times for each iteration of the inner For loop. If a solution to $Av = w$ exists over $\mathbb{Z}$ it certainly exists over $\mathbb{Q}(\theta_j)$, and by Lemma 2.6 $v^{(i,j)}$ will be such a solution (since $\mathbb{Q}(\theta_j)$ is a field and PID). Once the GCD of the denominators is one, we execute steps (12) and (13), as in `SmoothSolver`. Iliopolous's (1989) algorithm finds $\gamma_{ij} \in \mathbb{Z}$ such that $\log|\gamma_{i,j}| = O(\log\max_{i,j}|\delta_{i,j}|\cdot\log(sl))$. The constructed v satisfies

$$\begin{aligned}
\log\|v\| &= \log\left(\sum_{\substack{1\le i\le s\\ 1\le j\le l}} \gamma_{i,j}\|\delta_{i,j}v^{(i,j)}\|\right)\\
&= O((r\eta\log n + r\eta\log\|A\| + r\eta^2\log\|\Gamma\|)\cdot\log(rl)\\
&\qquad + \log\|w\|).
\end{aligned}$$

Since $\eta = O(\log r)$, $l = O(\log^2 r)$ and $\log\|\Gamma\| = O(\log^2 r)$ by Theorem 4.1, $\log\|v\| = O^{\sim}(r\log n + r\log\|A\| + \log\|w\|)$ which proves (ii).

To prove (iii) assume that a Diophantine solution does indeed exist. We show that with each iteration of the outer For loop, the algorithm finds such a solution with probability at least 1/2. Let $p \in \{p_1, \ldots, p_\kappa\}$ and suppose that $\Gamma_j \in G$ is irreducible modulo p and $\theta_j = (z \bmod \Gamma_j)$. Let B_p be the image of B in $\mathbb{Z}_p(\theta_j)^{r\times r}$. Since $\#(\mathcal{V}_j \bmod p) \ge 2r(r+1)$, by Theorem 2.5,

$$\operatorname{Prob}\left\{\operatorname{ord}_p B_p\begin{pmatrix}1\ldots k\\ 1\ldots k\end{pmatrix} = \operatorname{ord}_p d_k \quad \forall k : 1 \le k \le r\right\} \ge 1/2.$$

If indeed $\operatorname{ord}_p B_p\left(\begin{smallmatrix}1\ldots k\\ 1\ldots k\end{smallmatrix}\right) = \operatorname{ord}_p d_k$ for all k $(1 \le k \le r)$, B_p is Smith dominant. Thus by Theorem 2.1, the image

of $\bar{v}$ in $\mathbb{Q}_p(\theta_j)^{r\times 1}$ lies in $\mathbb{Z}_p[\theta_j]^{r\times 1}$, and the image of $v^{(i,j)}$ in $\mathbb{Q}_p(\theta_j)^{n\times 1}$ lies in $\mathbb{Z}_p[\theta_j]^{n\times 1}$, whence $p \nmid \operatorname{denom}(v^{(i,j)})$. Since $A \in \mathbb{Z}^{n\times n}$ and $w \in \mathbb{Z}^{n\times 1}$, $Av^{(i,j)} = Av_0^{(i,j)} = w$ and $Av_k^{(i,j)} = 0$ for $1 \le k < \eta$. Thus, the probability that $p \mid \operatorname{denom}(v^{(i,j)})$ for all $1 \le i \le s$ is at most $1/(2\kappa)$ and the probability this is true for any prime $p \in \{p_1, \ldots, p_\kappa\}$ is thus at most $1/2$. By executing the outer For loop $\lceil 1 + \log_2(1/\epsilon)\rceil$ times we ensure that if a solution exists, we will find one with probability at least $1-\epsilon$. □

Like `SmoothSolver`, `RefineToDiophantine` can be applied to rational matrices as a direct application of the techniques of Theorem 2.8. We first examine the cost of solving non-singular systems over a number field using the Block-Wiedemann algorithm.

THEOREM 5.2. *Let $\Gamma \in \mathbb{Z}[z]$ be irreducible of degree $\eta = O(\log r)$ with $\log \mathcal{H}(\Gamma) = O(\log^2 r)$ and $\theta = (z \bmod \Gamma)$. Suppose we are given a black box for a non-singular matrix $B \in \mathbb{Z}[\theta]^{r\times r}$ and vector $\bar{w} \in \mathbb{Z}[\theta]^{r\times 1}$ and wish to solve $B\bar{v} = \bar{w}$ for $\bar{v} \in \mathbb{Q}(\theta)^{r\times 1}$. Let $\varrho = r\log\|B\| + r\log r + \log\|\bar{w}\|$. On a network of $N \le r\varrho$ processors we can solve for $\bar{v}$ with an expected $O\tilde{}(r\varrho/N)$ matrix-vector products by B modulo (single word) primes with $O(\log r + \log\varrho)$ bits. An additional $O\tilde{}(r^2 + r\,\mathrm{M}(\varrho)/\min(r,N))$ bit operations is executed simultaneously by each processor. Each processor requires additional storage for $O\tilde{}(r\varrho/\min(r,N))$ words (not including possibly shared images of B modulo single-word primes).*

PROOF. We will apply Theorem 2.8 in a somewhat more complicated way than in Theorem 3.3. The set $\mathcal{B}$ of bad primes will consist of those primes p which either (i) divide the discriminant of Γ (so $\Gamma \bmod p$ is not squarefree; there are $O(\eta\log\eta + \eta\log\|\Gamma\|)$ such primes), (ii) are such that $(\det B \bmod p)$ is not a unit in the finite ring $\mathbb{Z}[z]/(\Gamma, p)$ (so B is not invertible modulo p; there are $O(\eta\log\eta + \eta r\log r + \eta r\log\|B\|)$ of these), or (iii) are less than $16r^2$ (there are $O(r^2/\log(r))$ of these). Thus $\#\mathcal{B}$ is polynomial in the logarithm of the output height $O(r\log r + r\log\|B\|)$.

After constructing a set of small primes $\mathcal{P}$ as in Theorem 2.8 (immediately eliminating those bad primes falling into cases (i) and (iii) above), the computation proceeds by completely factoring $(\Gamma \bmod p) \equiv \Gamma_1^{(p)} \cdots \Gamma_k^{(p)}$, where $\Gamma_i \in \mathbb{F}_p[z]$. We then apply the block-Wiedemann algorithm over the finite fields $\Gamma[z]/(\Gamma_i^{(p)}, p)$ for $1 \le i \le k$ (see Fact 3.2). The solution is first recovered by the Chinese remainder algorithm to get a solution in $(\mathbb{F}_p[z]/(\Gamma,p))^{r\times 1}$ and finally a solution in $\mathbb{Q}(\theta)^{r\times 1}$.

The execution costs can now be estimated as in Theorem 3.3. □

THEOREM 5.3. *The algorithm* `RefineToDiophantine` *works as stated on input $A \in \mathbb{Z}^{n\times n}$ with rank r, $w \in \mathbb{Z}^{n\times 1}$, $v^{(0)} \in \mathbb{Q}^{n\times 1}$ with $\delta_0 = \operatorname{denom}(v^{(0)})$ being $2r(r+1)$-smooth, and $\epsilon > 0$. Let $\varrho = r\log\|A\| + r\log r + \log\|w\|$ and suppose we are computing on a network of $N \le r\varrho$ processors.*

(i) If a Diophantine solution $v \in \mathbb{Z}^{n\times 1}$ to $Av = w$ exists, `RefineToDiophantine` *finds one with an expected number of $O\tilde{}(r\varrho/N)$ matrix-vector products by A modulo primes with $O(\log n + \log\varrho)$ bits. An additional $O\tilde{}(r^2 + rn\varrho/N + n\,\mathrm{M}(\varrho)/\min(n,N))$ bit operations is executed simultaneously by each processor.*

(ii) If no Diophantine solution exists, `RefineToDiophantine` *requires an expected number of $O\tilde{}((r\varrho/N)\cdot\log(1/\epsilon))$ matrix-vector products by A modulo primes with $O(\log n + \log\varrho)$ bits. An additional $O\tilde{}((r^2 + rn\varrho/N + n\,\mathrm{M}(\varrho)/\min(n,N))\cdot\log(1/\epsilon))$ bit operations is executed simultaneously by each processor.*

Each processor requires additional storage for $O\tilde{}(n + n\varrho/\min(n,N))$ words (not including possibly shared images of A modulo single-word primes).

PROOF. The number of elements in G governs the number l of iterations of the innermost For loop; by Theorem 4.1 $l = O(\log^2 r)$. Thus we execute (7)-(11) an expected number $2ls = O(\log^3 r)$ times if a solution exists and $O(\log^3(r)\cdot\log(1/\epsilon))$ times otherwise.

Each evaluation of the black box for $y \mapsto B_r y$ where $y = (y_1, \ldots, y_r)^t \in \mathbb{Z}[\theta_j]^{r\times 1}$ is performed by evaluating $UAL(y_1, \ldots, y_r, 0, \ldots, 0)^t = (z_1, \ldots, z_n)^t$, and returning $(z_1, \ldots, z_r)^t = B_r y$. Thus each matrix-vector product by B_r requires one black box evaluation of A modulo primes with $O(\log r + \log\log(\|B\| + \|\bar{w}\|))$ or $O(\log n + \log\varrho)$ bits, plus $O(n\log n\cdot\eta\log\eta)$ additional operations in $\mathbb{Z}[z]/(p,\Gamma)$ for primes p of this same size. The linear system $B_r\bar{v} = \bar{w}$ in step (6) is then solved using the Block-Wiedemann method described in Theorem 5.2. The remaining cost analysis follows in the same manner as Theorem 3.4. □

Given $A \in \mathbb{Z}^{n\times n}$ and $w \in \mathbb{Z}^{n\times 1}$, a complete algorithm for finding a Diophantine solution $v \in \mathbb{Z}^{n\times 1}$ such that $Av = w$ is obtained by first applying `SmoothSolver` with smoothness bound $\lambda = 2r(r+1)$ (to get a solution with λ-smooth denominator) followed by `RefineToDiophantine` (using the λ-smooth solution as additional input $v^{(0)}$). We immediately obtain the following corollary.

COROLLARY 5.4. *Let $A \in \mathbb{Z}^{n\times n}$ with rank r, $w \in \mathbb{Z}^{n\times 1}$ and $\epsilon > 0$. Let $\varrho = r\log\|A\| + r\log r + \log\|w\|$ and suppose we are computing on a network of $N \le r\varrho$ processors.*

(i) If a Diophantine solution $v \in \mathbb{Z}^{n\times 1}$ to $Av = w$ exists, we can find one with an expected number of $O\tilde{}(r\varrho/N)$ matrix-vector products by A modulo primes with $O(\log n + \log\varrho)$ bits. An additional $O\tilde{}(r^2 + rn\varrho/N + n\,\mathrm{M}(\varrho)/\min(n,N))$ bit operations is executed simultaneously by each processor. The returned $v \in \mathbb{Z}^{n\times 1}$ satisfies $\log\|v\| = O\tilde{}(r\log n + r\log\|A\| + \log\|w\|)$.

(ii) If no Diophantine solution exists, we can determine this with an expected number of $O\tilde{}((r\varrho/N)\cdot\log(1/\epsilon))$ matrix-vector products by A modulo primes with $O(\log n + \log\varrho)$ bits; an additional $O\tilde{}((r^2 + rn\varrho/N + n\,\mathrm{M}(\varrho)/\min(n,N))\cdot\log(1/\epsilon))$ bit operations is executed simultaneously by each processor.

An incorrect solution is never returned. If any solution exists, one is found with probability at least $1-\epsilon$. Each processor requires additional storage for $O\tilde{}(n + n\rho/\min(n,N))$ words (not including possibly shared images of A modulo single-word primes).

6 Open Questions

A number of important questions remain unresolved and extensions remain unexplored.

Random generation of Diophantine solutions. While the solutions we generate are in some sense random, it has not been proven that they in any way sample uniformly from the solution space. It is generally not possible to write down a complete basis for the solution space within the amount of time and space allowed. Still, Kaltofen & Saunders (1991) showed how to randomly sample from the solution manifold for singular systems of linear equations over a field. Such a result should be obtainable in the current context.

Proving `SmoothSolver` yields Diophantine solutions. `SmoothSolver` is currently only shown to give solutions whose denominators are $2r(r+1)$-smooth. These are later refined to integer solutions by `RefineToDiophantine`. It seems quite possible that `SmoothSolver` finds Diophantine solutions quickly as well, but this appears difficult to prove. The problem seems akin to showing Coppersmith's (1994) algorithm works over $\mathbb{F}_2$; see Kaltofen (1995).

Implementation. The algorithms discussed here are currently being implemented using the LiDIA library for computational number theory.

References

E. Bach and J. Shallit. *Algorithmic Number Theory, Volume 1: Efficient Algorithms*. MIT Press (Cambridge, MA), 1996.

E. R. Berlekamp. Factoring polynomials over large finite fields. *Math. Comp.* **24**, pp. 713–735, 1970.

W. A. Blankinship. Algorithm 288, solution of simultaneous linear diophantine equations. *Comm. ACM* **9**, pp. 514, 1966.

I. Borosh and A. S. Fraenkel. Exact solutions of linear equations with rational coefficients by congruence techniques. *Mathematics of Computation* **20**, pp. 107–112, 1966.

G. Bradley. Algorithms for Hermite and Smith normal matrices and linear diophantine equations. *Math. Comp* **25**(116), pp. 897–907, 1971.

J.W.S. Cassels. *Local Fields*, vol. 3 of *London Mathematical Society Student Texts*. Cambridge University Press, 1986.

T. J. Chou and G. E. Collins. Algorithms for the solution of systems of linear Diophantine equations. *SIAM J. of Computing* **11**, pp. 687–708, 1982.

H. Cohen. *A Course in Computational Number Theory*. Springer, 1993.

G. Collins and M. Encarnación. Efficient rational number reconstructions. *Journal of Symbolic Computation* **20**, pp. 287–297, 1995.

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

F. R. Gantmacher. *The Theory of Matrices, Vol. I.* Chelsea Publishing Co. (New York NY), 1990.

P. Gibbons. Computational methods in design theory. In *The CRC Handbook of Combinatorial Designs*, ed. C. Colbourn and J. Dinitz, pp. 725–728. CRC Press, 1996.

M. Giesbrecht. Nearly optimal algorithms for canonical matrix forms. *SIAM J. Comp.* **24**, pp. 948–969, 1995.

M. Giesbrecht. Fast computation of the Smith form of a sparse integer matrix. *Computational Complexity* , 1996. Submitted.

J. L. Hafner and K. S. McCurley. A rigorous subexponential algorithm for computation of class groups. *J. Amer. Math. Soc.* **2**, pp. 837–850, 1989.

G. Havas and B.S. Majewski. Hermite normal form computation for integer matrices. *Congressus Numerantium* **105**, pp. 184–193, 1994.

G. Havas, D. Holt, and S. Rees. Recognizing badly presented Z-modules. *Linear algebra and its applications* **192**, pp. 137–163, 1993.

C. Iliopolous. Worst-case complexity bounds on algorithms for computing the canonical structure of finite abelian groups and the Hermite and Smith normal forms of an integer matrix. *SIAM J. Computing* **18**, pp. 658–669, 1989.

T. Kailath. *Linear systems*. Prentice-Hall (Englewood Cliffs, New Jersey), 1980.

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. Kannan and A. Bachem. Polynomial algorithms for computing the Smith and Hermite normal forms of an integer matrix. *SIAM J. Comp.* **8**, pp. 499–507, 1979.

S. Lang. *Algebraic Number Theory*. Springer-Verlag (New York), 1986.

R. Lidl and H. Niederreiter. *Finite Fields*, vol. 20 of *Encyclopedia of Mathematics and its Applications*. Addison-Wesley (Reading MA), 1983.

B. Majewski and G. Havas. A solution to the extended gcd problem. In *Proc. ISSAC'95*, pp. 248–253, Montreal, Canada, 1995.

M. Newman. *Integral Matrices*. Academic Press (New York), 1972.

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

A. Storjohann. A fast+practical+deterministic algorithm for triangularizing integer matrices. Preprint, 1996.

A. Storjohann and G. Labahn. Asymptotically fast computation of Hermite normal forms of integer matrices. In *Proceedings of ISSAC'96*, pp. 259–266, Zurich, Switzerland, 1996.

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.

P. Wang, M. Guy, and J. Davenport. *P*-adic reconstruction of rational numbers. *SIGSAM Bulletin* **16**(2), pp. 2–3, 1982.

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.