Diophantine Linear System Solving

Thom Mulders Arne Storjohann

Institute of Scientific Computing
ETH Zurich, Switzerland
{mulders,storjoha}@inf.ethz.ch

Abstract

A simple randomized algorithm is given for finding an integer solution to a system of linear Diophantine equations. Given as input a system which admits an integer solution, the algorithm can be used to find such a solution with probability at least 1/2. The running time (number of bit operations) is essentially cubic in the dimension of the system. The analogous result is presented for linear systems over the ring of polynomials with coefficients from a field.

1 Introduction

Solving a system of linear Diophantine equations is a classical mathematical problem: given an integer matrix A and vector b, the goal is to find an integer vector x that satisfies $Ax = b$. We present a simple randomized algorithm for solving this problem. We also show how to adapt the algorithm to solve linear systems over the ring of univariate polynomials with coefficients from a field. The algorithm is easy to implement, memory efficient, and faster than previous methods. The algorithm is probabilistic in the following sense. If a solution vector is returned it is guaranteed to be correct. On the other hand, the algorithm does not currently certify inconsistency and may return NIL for a system which does admit a solution; the chance of this happening is bounded by a user specified error tolerance $\epsilon > 0$.

For the moment we restrict the discussion to integer systems; the case for polynomial systems is analogous. In this paper we analyse our algorithms under the assumption of standard, quadratic integer arithmetic. Let $O(M(t))$ bit operations be sufficient to multiply two t-bit integers. Then standard arithmetic has $M(t) = t^2$. FFT-based methods allow $M(t) = t \log t \log\log t$.

Let n be the larger of the row or column dimension of a linear Diophantine system $Ax = b$. Let β be a bound on the magnitudes of entries in A and b. If $Ax = b$ admits an integer solution for x, then it admits an integer solution with entries having length $O(n(\log n + \log \beta))$ bits — this bound is tight in the worst case. The algorithm we present here uses an expected number of $O^{\sim}\left(n^3(\log \beta)^2\right)$ bit operations to return a solution with this size bound. For input systems with $\log \beta$ small compared to n, this improves on the previously fastest algorithm which uses $O^{\sim}\left(n^4 \log \beta + nM(n \log \beta)\right)$ bit operations in the worst case [7]. Moreover, the algorithm we present here, like that in [7], requires additional storage for only $O\left(n^2(\log n + \log \beta)\right)$ bits. This space complexity, which is linear in the size of the output vector and essentially linear in the size of a dense input system, is an important feature of the algorithm.

The global technique of our Diophantine solver is similar to the randomized algorithm in [7]. The key idea is to compute a small number of rational solutions of perturbations of the input system. In [7], the perturbed systems have the form $UALy = Ub$, where U and L are randomly chosen Toeplitz matrices. In our algorithm, the perturbed systems have the form $APy = b$, where P is a random dense matrix. Note that if y is a rational solution to $APy = b$, then Py is a rational solution to the original system $Ax = b$. The idea is to compute a sequence of these perturbed rational solutions which can then hopefully be combined to obtain a Diophantine solution. This suggests an iterative algorithm which is supposed to "converge" to a Diophantine solution. A worked example demonstrating this technique is given in Section 2. The challenge is twofold. First, show how to efficiently solve the perturbed rational systems. Second, specify how to choose the entries in the perturbation matrices, and prove that fast convergence can be expected with these choices.

In [7], the perturbed rational systems are solved using the Wiedemann algorithm together with a homomorphic imaging scheme; this leads to an algorithm which admits an easy coarse grain parallelization and exploits the possible structure or sparsity of A. Choosing the U and L to be Toeplitz is important so that matrix-vector products involving U and L do not cost more than matrix-vector products involving A. For a very sparse matrix A, with number of nonzero entries essentially linear instead of quadratic in n, this approach leads to an algorithm with running time $O^{\sim}\left(n^3 \log \beta + nM(n \log \beta)\right)$ bit operations. For a dense input system, though, the running time is $O^{\sim}(n^4 \log \beta + nM(n \log \beta))$ bit operations. In order to prove that a small (i.e. logarithmic) number of perturbed rational solutions will converge to a Diophantine solution, entries in U and L are chosen from a small extension ring of the integers.

Our approach is to solve the perturbed rational systems using linear p-adic lifting as described in [4]. In Section 5 we offer pseudo-code for the p-adic rational solver together with

a running time analysis under the assumption of standard arithmetic.

This leaves the question of how to choose entries in the preconditioning matrices P. For practical reasons, we would like to avoid requiring to work over an extension ring. Idealy, entries in P should be very small integers so as to keep growth of numbers to a minimum. To achieve the claimed running time bound for our iterative Diophantine solver, we require convergence to a Diophantine solution in an expected number of $O(\log n + \log\log\beta)$ iterations. In this paper we prove that such a rate of convergence can be expected even if the entries in P are chosen from the smallest possible subset of the ring, namely $\{0,1\}$. This proof is based on two results. First, we give in Section 3 a new characterization of when a preconditioning matrix P will be successful with respect to a given prime p, that is, when p does not appear in the denominator of the perturbed rational solution. Second, in Section 4 we use a linear algebra based counting argument, similar to that in [14], to prove that a randomly chosen $\{0,1\}$-matrix P will be effective with respect to p with probability at least 1/4. The arguments in Section 3 and 4 use only elementary facts about lattices and vector spaces and are applicable over an arbitrary principal ideal domain.

Many systems arising in practice are rectangular and/or singular and it is desirable to take this into account when designing algorithms. We analyse our algorithms in terms of the five paramaters n, m, r, $||A||$, and $||b||$. Here, r is the rank of an $n \times m$ input matrix A and $||A||$ is a bound on the magnitudes of entries in A (similarly for $||b||$). Then the cost of our algorithm is $O^{\sim}\left(nmr(\log||A||)^2 + m(\log||b||)^2\right)$ bit operations. If a solution vector is produced, it will have entries bounded in length by $O(r\log(m||A||)+\log||b||)$ bits. We give the analogous result for solving a system over the ring $F[X]$, F a field. In this case the algorithm uses $O^{\sim}\left(nmr(\deg A)^2 + m(\deg b)^2\right)$ field operations from F. Entries in solution vectors will have degrees bounded by $O(r\deg A + \deg b)$.

2 Diophantine linear system solving

In this section we give an example making clear how our randomized algorithm for solving Diophantine linear systems works. We only give an example of a full row rank system, leaving the details of the general case to Section 6. The idea of the algorithm described here is not only valid for solving integer Diophantine linear systems. Therefore, we present some results in this paper in the more general setting of principal ideal domains R. In Section 6 we specialize to the case when $R = \mathbb{Z}$ and $R = F[X]$, F a field.

Henceforth let R be a principal ideal domain and K its quotient field.

Let $x \in K^m$ be given. It is easy to see that the set of all $u \in R$ such that $ux \in R^m$ is an ideal in R.

Definition 1 *For $x \in K^m$, a generator of the ideal of all $u \in R$ such that $ux \in R^n$ is called a* denominator *of x.*

Let $A \in R^{n\times m}$ of rank n and $b \in R^n$. When A happens to be a square matrix, the system $Ax = b$ has a Diophantine solution if and only if the unique rational solution $A^{-1}b$ of the system has denominator a unit from R. If, however, A is not a square matrix, the system $Ax = b$ has infinitely many rational solutions. The idea is now to combine finitely many rational solutions to find a possible Diophantine solution of the system.

To keep all expressions integral (i.e. in R), we represent $x \in K^m$ by a pair (z, w), where $z \in R^m$ and $w \in R$ such that $x = z/w$. If $Az = wb$ then we call the pair (z, w) a rational solution of the linear system $Ax = b$. If in addition w is a unit in R then we call the pair (z, w) a Diophantine solution of the system.

The idea of our algorithm, also used in [7], is to take a linear combination of several rational solutions of the system in order to obtain a Diophantine solution. For this we use the following lemma.

Lemma 2 *Let (z, w) and (y, v) be rational solutions of $Ax = b$. Let $d, s, t \in R$ such that $d = \gcd(w, v) = sw + tv$. Then $(sz + ty, d)$ is a rational solution of $Ax = b$.*

Proof: $A(sz+ty) = sAz+tAy = swb+tvb = (sw+tv)b = db$.•

Iterative application of Lemma 2 to combine more than two solution vectors may result in expression swell. To avoid this we also use the following modified version of the lemma.

Lemma 3 *Let (z, w), (y, v) and (q, u) be rational solutions of $Ax = b$. Let $t \in R$ such that $\gcd(v+tu, w) = \gcd(v, u, w)$. Then $(y + tq, v + tu)$ is a rational solution of $Ax = b$.*

Proof: $A(y + tq) = Ay + tAq = vb + tub = (v + tu)b.$ •

Let (z, w) and $(z_1, w_1), \ldots, (z_s, w_s)$ be rational solutions of $Ax = b$. We can then use Lemma 3 to compute rational solutions $(y_1, v_1), \ldots, (y_s, v_s)$ of $Ax = b$, such that for all i we have $\gcd(v_i, w) = \gcd(w_1, \ldots, w_i, w)$. Next we can use Lemma 2 to compute a rational solution $(\hat{z}, \hat{w})$ of $Ax = b$ such that $\hat{w} = \gcd(w, v_s) = \gcd(w, w_1, \ldots, w_s)$.

To get another rational solution of $Ax = b$, we apply the following result for different choices of P.

Lemma 4 *Let $P \in R^{m\times n}$. If (z, w) is a rational solution of $APx = b$, then (Pz, w) is a rational solution of $Ax = b$.*

Our approach is to choose the matrices P randomly from $\{0,1\}^{m\times n}$. The following example will demonstrate the method that we use.

Example Let

$$A = \begin{bmatrix} 1 & 2 & 1 & 3 \\ 2 & 5 & 3 & 2 \end{bmatrix} \text{ and } b = \begin{bmatrix} -1 \\ 2 \end{bmatrix}.$$

Taking several choices for P we get the following solutions.

P	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}$
AP	$\begin{bmatrix} 3 & 4 \\ 8 & 4 \end{bmatrix}$	$\begin{bmatrix} 3 & 3 \\ 7 & 2 \end{bmatrix}$	$\begin{bmatrix} 6 & 6 \\ 9 & 10 \end{bmatrix}$
q	$\begin{bmatrix} 6 \\ -7 \end{bmatrix}$	$\begin{bmatrix} 8 \\ -13 \end{bmatrix}$	$\begin{bmatrix} -22 \\ 21 \end{bmatrix}$
w	10	15	6
Pq	$\begin{bmatrix} -7 \\ 6 \\ 6 \\ -7 \end{bmatrix}$	$\begin{bmatrix} 8 \\ 8 \\ 0 \\ -13 \end{bmatrix}$	$\begin{bmatrix} -22 \\ -1 \\ 21 \\ -1 \end{bmatrix}$

We now use Lemma 3 to combine the second and the third rational solution. Since $\gcd(15+2\cdot 6, 10) = \gcd(15, 6, 10)$, we get the new rational solution

$$\left(\begin{bmatrix} -36 \\ 6 \\ 42 \\ -15 \end{bmatrix}, 27 \right).$$

Then we use Lemma 2 to combine this new rational solution with the first rational solution. Since $1 = \gcd(10, 27) = -8\cdot 10 + 3\cdot 27$, we get the new rational solution

$$\left(\begin{bmatrix} -52 \\ -30 \\ 78 \\ 11 \end{bmatrix}, 1 \right),$$

which is a Diophantine solution of $Ax = b$.

3 Preconditioning

In the algorithm demonstrated in Section 2 we multiply the matrix A by a matrix P in order to "randomize" the primes that divide the denominator of the computed rational solution x of the system $Ax = b$. We call this *preconditioning* the system $Ax = b$. In this section we study when a prime p will be present in the denominator of x. In fact, we consider the slightly more general situation, when $Ax = b$ does not necessarily have a Diophantine solution.

Let R be a principal ideal domain. Let $A \in R^{n\times m}$ be of rank n and $b \in R^n$. Recall that to keep expressions integral we represent rational solutions of $Ax = b$ by ordered pairs (z, w) with $z \in R^m$, $w \in R$ and $Az = wb$. It is easy to see that the set of all $w \in R$ such that there exists a $z \in R^m$ with (z, w) a rational solution of $Ax = b$ is an ideal in R.

Definition 5 *$d(A, b)$ is a generator of the ideal*

$$\{w \in R \mid \exists z \in R^m : Az = wb\}.$$

$d(A, b)$ is the *minimal denominator* that a rational solution of $Ax = b$ can have in the sense that $d(A, b)$ divides the denominator of any rational solution of $Ax = b$. Clearly, $Ax = b$ has a Diophantine solution if and only if $d(A, b)$ is a unit in R. Note that $d(A, b)$ is defined only up to multiplication by a unit in R, but for our purpose this suits well.

Definition 6 *A rational solution (z, w) of $Ax = b$ is called a solution with* minimal denominator *if w is an associate of $d(A, b)$, that is, if $w/d(A, b)$ is a unit in R.*

Definition 7 *Let $p \in R$ be prime. For $a \in R$ we define $\mathrm{ord}_p(a)$ as the maximum integer n such that p^n divides a.*

To get a sequence of rational solutions of the system $Ax = b$ we apply the following result for different choices of P.

Lemma 8 *Let $P \in R^{m\times n}$ be such that AP is nonsingular, and let w be a denominator of $(AP)^{-1}b$. Then $(P(w(AP)^{-1}b), w)$ is a rational solution of $Ax = b$.*

Proof: $AP(w(AP)^{-1}b) = w(AP)(AP)^{-1}b = wb$ and $P(w(AP)^{-1}b) \in R^n$ since $w(AP)^{-1}b \in R^m$. •

The hope is that a rational solution of $Ax = b$ constructed as in Lemma 8 has no extra prime divisor p in its denominator, that is, that $\mathrm{ord}_p(w) = \mathrm{ord}_p(d(A, b))$. This will in general not be true for all primes at once.

Definition 9 *Let $p \in R$ be prime. $P \in R^{m\times n}$ is a* good preconditioner with respect to p for the system $Ax = b$ *if:*

- *AP is nonsingular, and*
- *$\mathrm{ord}_p(w) = \mathrm{ord}_p(d(A, b))$, w a denominator of $(AP)^{-1}b$.*

The following fact is used in the proof of the next lemma. Recall that a square matrix V over R is said to be unimodular if V is invertible over R, that is, if V^{-1} is over R. The unimodular matrices over R are precisely those with determinant a unit from R.

Fact 10 *For any matrix $A \in R^{n\times m}$ of rank n, there exists a unimodular matrix $V \in R^{m\times m}$ such that $AV = H = \begin{bmatrix} H_1 & 0 \end{bmatrix}$, where H_1 is $n\times n$ and nonsingular. For example, we can take H to be the column Hermite normal form of A (see [11]).*

Lemma 11 *Let $A \in R^{n\times m}$ of rank n and $b \in R^n$. There exists a $W \in R^{n\times m}$ such that for every prime $p \in R$:*

- *W has rank n modulo p.*
- *For $P \in R^{m\times n}$:*

$$p \nmid \det(WP) \Rightarrow \begin{array}{c} P \text{ is a good preconditioner for} \\ Ax = b \text{ with respect to } p. \end{array}$$

Proof: Let V and H be as in Fact 10. We claim that we can take for W the first n rows of V^{-1}. Let $p \in R$ be prime. Since V^{-1} is also over R and unimodular it is clear that W has rank n modulo p.

This shows that W satisfies the first property of the lemma. We now show that W satisfies the second property of the lemma. Let $P \in R^{m\times n}$ and assume $p \nmid \det(WP)$. Since $A = HV^{-1}$ and $H = \begin{bmatrix} H_1 & 0 \end{bmatrix}$ we have

$$A = H_1 W. \tag{1}$$

It follows that $AP = H_1 WP$ is nonsingular since H_1 is nonsingular and WP is nonsingular modulo p. It remains to show the second property of Definition 9. Let (y, d) be a rational solution of $Ax = b$ with minimal denominator, that is, $Ay = db$ with d an associate of $d(A, b)$. Substituting (1) into $Ay = db$ yields $H_1^{-1}b = 1/d \cdot Wy$. Then

$$\begin{aligned} (AP)^{-1}b &= (H_1 WP)^{-1}b \\ &= (WP)^{-1}H_1^{-1}b \\ &= \frac{1}{d\det(WP)} \cdot \det(WP)(WP)^{-1}Wy \end{aligned} \tag{2}$$

Let w be a denominator of $(AP)^{-1}b$. From (2) we see that $w | (d\det(WP))$ since $\det(WP)(WP)^{-1}Wy$ is over R. It follows that $\mathrm{ord}_p(w) \le \mathrm{ord}_p(d)$ since by assumption $p \nmid \det(WP)$. On the other hand, we must have $\mathrm{ord}_p(d) \le \mathrm{ord}_p(w)$ since $(P(w(AP)^{-1}b), w)$ is a rational solution of $Ax = b$ (by Lemma 8). It follows that $\mathrm{ord}_p(w) = \mathrm{ord}_p(d)$. Thus, P satisfies the two properties of Definition 9 and is a good conditioner with respect to p. •

Let $p \in R$ be prime and let $(\hat{z}, \hat{w})$ be a rational solution of $Ax = b$ resulting (as in Section 2) from combining several rational solutions obtained by using several preconditioners P. In order that there be no extra factor p in $\hat{w}$ — i.e. $\mathrm{ord}_p(\hat{w}) = \mathrm{ord}_p(d(A, b))$ — it is required that at least one of these matrices P is a good preconditioner of $Ax = b$ with respect to p. For this it suffices that $p \nmid \det(WP)$. In the next section we will study the probability that a prime divides $\det(WP)$ when $P \in \{0, 1\}^{m\times n}$ is chosen at random.

4 Probability of good preconditioning

In this section we will prove the following theorem.

Theorem 12 *Let $A \in R^{n\times m}$ of rank n and $b \in R^n$. Let $p \in R$ be prime. The probability that a randomly chosen $P \in \{0,1\}^{m\times n}$ is a good preconditioner for $Ax = b$ with respect to p is at least $1/4$.*

For a matrix A over a field F we will denote by $C(A)$ the column-span of A over F, that is, $C(A)$ is the vector space generated by the columns of A. When A has zero columns, $C(A) = \{0\}$. Recall that a matrix $A \in F^{m\times n}$ of rank s is in reduced column echelon form if: the last $n-s$ columns are zero; $i_1 < i_2 \ldots < i_s$, where for $1 \le j \le s$, the first nonzero entry in column j occurs in row i_j; $A_{i_j,j} = 1$ and $A_{i_j,k} = 0$ for $1 \le j \le s$, $k \ne j$. For example,

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ * & 0 & 0 & 0 \\ * & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ * & * & * & 0 \end{bmatrix}$$

is in reduced column echelon form with $s = 3$ and $[i_1, i_2, i_3] = [1, 4, 5]$. Any matrix can be transformed to reduced column echelon form using only elementary column operations; this leaves the rank and column-span unchanged.

Lemma 13 *Let F be a field and $N \in F^{m\times n}$ of rank s. Then*

$$\#\{u \in \{0,1\}^m \mid u \in C(N)\} \le 2^s.$$

Proof: Without loss of generality we may assume that N is in reduced column echelon form and has full column rank. For $x \in F^s$ we have

$$Nx \in \{0,1\}^m \Rightarrow x \in \{0,1\}^s.$$

Since $C(N) = \{Nx \mid x \in F^s\}$, the lemma follows easily. •

Definition 14 *A matrix K is a* right-kernel *for a matrix A if $AK = 0$ and for all x such that $Ax = 0$, we have $x \in C(K)$.*

Lemma 15 *Let F be a field, $W \in F^{n\times m}$ and $P \in F^{m\times t}$ such that WP has full column rank. Let K be a right-kernel for W. Then for $u \in F^m$:*

$$W\begin{bmatrix} P & u \end{bmatrix} \text{ has full column rank } \Leftrightarrow u \notin C(\begin{bmatrix} K & P \end{bmatrix}).$$

Proof: When $t = 0$, the lemma holds. So assume $t > 0$. $W\begin{bmatrix} P & u \end{bmatrix}$ has full column rank if and only if $Wu \notin C(WP)$. Suppose $Wu \in C(WP)$, say $Wu = WPx$ for $x \in F^t$. Then $u - Px \in C(K)$ and thus $u \in C(\begin{bmatrix} K & P \end{bmatrix})$. When $u \in C(\begin{bmatrix} K & P \end{bmatrix})$, then $Wu \in C(W\begin{bmatrix} K & P \end{bmatrix}) = C(WP)$. •

Lemma 16 *Let F be a field and $W \in F^{n\times m}$ of rank n. Then for $0 \le t \le n$:*

$$\#\{P \in \{0,1\}^{m\times t} \mid WP \text{ has rank } t\}$$
$$\ge 2^{mt}\left(1 - \left(\frac{1}{2}\right)^{n-t+1}\right)\cdots\left(1 - \left(\frac{1}{2}\right)^{n}\right).$$

Moreover, when K is a right kernel for W and $P \in \{0,1\}^{m\times t}$ is such that $\mathrm{rank}(WP) = t$, *then* $\mathrm{rank}(\begin{bmatrix} K & P \end{bmatrix}) = m - n + t$.

Proof: By induction to t. The lemma trivially holds for $t = 0$. Now assume that the lemma holds for some $0 \le t < n$.

Write $P \in \{0,1\}^{m\times(t+1)}$ as $\begin{bmatrix} Q & u \end{bmatrix}$, where $Q \in \{0,1\}^{m\times t}$ and $u \in \{0,1\}^m$. From Lemma 15 it follows that

$$\mathrm{rank}(WP) = t+1 \Leftrightarrow \mathrm{rank}(WQ) = t \text{ and } u \notin C(\begin{bmatrix} K & Q \end{bmatrix}).$$

From this and the induction hypothesis it follows that when $\mathrm{rank}(WP) = t+1$, then $\mathrm{rank}(\begin{bmatrix} K & Q \end{bmatrix}) = m - n + t$ and $u \notin C(\begin{bmatrix} K & Q \end{bmatrix})$, and thus $\mathrm{rank}(\begin{bmatrix} K & Q & u \end{bmatrix}) = m - n + t + 1$. By the induction hypothesis there are at least $2^{mt}\left(1 - \left(\frac{1}{2}\right)^{n-t+1}\right)\cdots\left(1 - \left(\frac{1}{2}\right)^{n}\right)$ matrices $Q \in \{0,1\}^{m\times t}$ such that $\mathrm{rank}(WQ) = t$. From Lemma 13 we see that for each $Q \in \{0,1\}^{m\times t}$ such that $\mathrm{rank}(WQ) = t$, there are at least $2^m - 2^{m-n+t}$ vectors $u \in \{0,1\}^m$ such that $u \notin C(\begin{bmatrix} K & Q \end{bmatrix})$. The inequality follows for $t+1$. •

Corollary 17 *Let F be a field and $W \in F^{n\times m}$ of rank n. When we choose a random $P \in \{0,1\}^{m\times n}$, then the probability that WP is nonsingular is at least*

$$\left(1 - \frac{1}{2}\right)\cdots\left(1 - \left(\frac{1}{2}\right)^{n}\right) > 1/4.$$

Proof: The inequality is well known (see for example [5]). The result follows from Lemma 16 (take $t = n$). •

Now we can prove Theorem 12.

Proof: (of Theorem 12) Let W be a matrix as in Lemma 11. Then P is a good preconditioner for $Ax = b$ with respect to p, when $p \nmid \det(WP)$. Now apply Corollary 17 with $F = R/pR$. •

5 Rational linear system solving

Let R be a principal ideal domain and K its quotient field. Every linear system

$$Ax = b, \quad A \in R^{n\times n}, \quad \det A \ne 0, \quad b \in R^n \tag{3}$$

admits a unique rational solution $x \in K^n$. In this section we recall how to recover x using p-adic lifting and rational reconstruction. The algorithm is applicable when $R = \mathbb{Z}$ or $R = F[X]$. We present the algorithm first for the case $R = \mathbb{Z}$. The result for $R = F[X]$ is analogous and will be given in the sequel. For Z a vector or matrix over $\mathbb{Z}$, we denote by $||Z||$ the maximal magnitude of all entries.

The method we present here is similar to that described in [4, 13]. One of our contributions here is to give an analysis in terms of the parameters $||A||$ and $||b||$ instead of a common parameter $\beta = \max(||A||, ||b||)$. In particular, our analysis shows that even if entries in b have length $O(n(\log n + \log ||A||))$ bits, the asymptotic running time of the algorithm remains unchanged (see Theorem 20.) This feature is exploited in [1].

Before presenting the algorithm we first bound the size of the rational solution x to (3). The following well known bounds follow from Hadamard's inequality and Cramer's rule [6].

Fact 18 $|\det A| \le n^{n/2}||A||^n$. *Moreover,* $(\det A)x$ *is over* $\mathbb{Z}$ *and satisfies* $||(\det A)x|| \le n^{n/2}||A||^{n-1}||b||$.

Algorithm RationalSolver(A, b, p)
Input: $A \in R^{n\times n}$, $b \in R^n$, $p \in R$.
Output: Either NIL or $x \in K^n$ such that $Ax = b$.

(1) [Initialize:]
$N :=$ NumeratorBound(A, b);
$D :=$ DenominatorBound(A);
$L :=$ LiftingBound(N, D);
if $p \perp \det A$ **then**
$B := \mathrm{mod}(A^{-1}, p)$
else
return NIL
fi;

(2) [Lift:]
$z := 0^{n\times 1}$;
$c := b$;
$M := 1$;
while Size$(M) \leq L$ **do**
$\bar{c} := \mathrm{mod}(c, p)$;
$\bar{z} := \mathrm{mod}(B\bar{c}, p)$;
$c := (c - A\bar{z})/p$;
$z := z + M\bar{z}$;
$M := Mp$
od;

(3) [Reconstruct:]
$x :=$ RationalReconstruction(z, M, N, D);
return x

Figure 1: Algorithm RationalSolver

Next we define the mod function. Let $a \in \mathbb{Q}$ have denominator relatively prime to $p \in \mathbb{N}$. Then $\mathrm{mod}(a, p)$ returns the unique integer c, $0 \leq c < p$, which is congruent to a modulo p. If X is a matrix or vector, then $\mathrm{mod}(X, p)$ returns X with all entries reduced modulo p. The algorithm for solving (3) is now easy to describe. We assume we have a $p \in \mathbb{N}$, $p > 1$, $p \perp \det A$. (We write $p \perp \det A$ to mean p is relatively prime to $\det A$.) Then x has a unique p-adic expansion $x = z_0 + z_1 p + z_2 p^2 + \cdots$ where each $z_i \in \mathbb{Z}^n$ satisfies $z_i = \mathrm{mod}(z_i, p)$. We use linear p-adic lifting to compute the first k terms of this expansion for large enough k, that is, find $z = z_0 + z_1 p + z_2 p^2 + \cdots + z_{k-1} p^{k-1}$ so that $z = \mathrm{mod}(x, M)$ with $M = p^k$, and then recover x from z using rational reconstruction (see [3]).

Fact 19 *Let $z \in \mathbb{Z}$ and $N, D, M \in \mathbb{N}$ be given. If $M > 2ND$, then there exists at most one $x \in \mathbb{Q}$ with $z = \mathrm{mod}(x, M)$ and with numerator and denominator bounded in magnitude by N and D respectively. If $|z| \leq M$ then such an x can be recovered in $O\left((\log M)^2\right)$ bit operations.*

The algorithm for solving (3) is shown in Figure 1. The reconstruction phase is performed using the algorithm of Fact 19. The remaining functions used in the algorithm can be defined as follows:

- NumeratorBound$(A, b) \to \lfloor n^{n/2} ||A||^{n-1} ||b|| \rfloor$;
- DenominatorBound$(A) \to \lfloor n^{n/2} ||A||^n \rfloor$;
- LiftingBound$(N, D) \to 2ND$;
- Size$(M) \to M$.

Theorem 20 $(R = \mathbb{Z})$ *Algorithm* RationalSolver *is correct. The cost of the algorithm is bounded by*

$$O\left(n^3(\log n + \log ||A|| + \log p)^2 + n(\log ||b||)^2\right)$$

bit operations.

Proof: To compute $\det A$ in the initialization phase - and determine B at the same time in case $p \perp \det A$ - first reduce all entries of A modulo p and apply Gaussian elimination. This requires $O\left(n^2(\log ||A||)(\log p) + n^3(\log p)^2\right)$ bit operations. The cost bound for the reconstruction phase follows from Fact 19. It remains to bound the cost of the lifting phase. The while loop implements linear p-adic lifting: after the i-th iteration we have $z = \mathrm{mod}(A^{-1}b, p^i)$ and $c = (b - Az)/p^i$. Let $E = n \log n + 2n \log ||A|| + \log ||b|| + \log p + 1$. Then $\log ||z||, \log ||c||, \log M \leq E$ holds throughout the computation. For a single pass through the loop, the cost of computing $\bar{z}$ is bounded by $O\left(n^2(\log p)^2\right)$ bit operations; the cost of the remaining computations are bounded by $O(nE \log p)$ bit operations. Noting that the number of iterations of the while loop is at most $\lfloor E/\log p \rfloor$, we arrive at a total cost bound for the while loop of $O\left(n^2 E \log p + nE^2\right)$ bit operations. This simplifies to $O(T_1 + T_2 + T_3)$ bit operations where $T_1 = n^3(\log n + \log ||A|| + \log p)^2$, $T_2 = n(\log ||b||)^2$ and $T_3 = n^2(\log p)(\log ||b||)$. If $\log ||b|| < n \log p$ then $T_3 < T_1$. Otherwise, $\log ||b|| \geq n \log p$ and $T_3 < T_2$. The result follows. •

Now consider the case when $R = F[X]$, F a field. Let $a \in F(X)$ have denominator relatively prime to $p \in F[X]$. Then $\mathrm{mod}(a, p)$ returns the unique polynomial c, $0 \leq \deg c < \deg p$, which is congruent to a modulo p. For Z a vector or matrix over $F[X]$, we denote by $\deg Z$ the maximal degree of all entries. Analogous to Fact 18 we have the following degree bounds for the solution vector x to (3).

Fact 21 $\deg \det A \leq n \deg A$. *Moreover,* $(\det A)x$ *is over* $F[X]$ *and satisfies* $\deg((\det A)x) \leq (n-1)\deg A + \deg b$.

Thus, degrees of numerators and denominators appearing in x will be bounded by $N = n \deg A$ and $D = (n-1)\deg A + \deg b$ respectively. Analagous to Fact 19, recovering these entries using rational reconstruction requires lifting up to degree bound $M > N + D$ and costs $O\left(M^2\right)$ field operations. The functions used in algorithm RationalSolver can be defined as follows:

- NumeratorBound$(A, b) \to (n-1)\deg A + \deg b$;
- DenominatorBound$(A) \to n \deg A$;
- LiftingBound$(N, D) \to N + D$;
- Size$(M) \to \deg M$.

Theorem 22 $(R = F[X])$ *Algorithm* RationalSolver *is correct. The cost of algorithm is bounded by*

$$O\left(n^3(\deg A + \deg p)^2 + n(\deg b)^2\right)$$

field operations from F.

Proof: The proof is analogous to that for Theorem 20. Let $E = 2n \deg A + \deg b + \deg p$. Then $\deg z, \deg c, \deg M < E$ holds throughout the lifting phase and the number of iterations of the while loop is bounded by $\lfloor E/\deg p \rfloor$. For a single pass through the loop, the cost of computing $\bar{z}$ is bounded by $O\left(n^2(\deg p)^2\right)$ and the remaining computation by $O(nE \deg p)$ field operations. The result follows. •

6 Diophantine linear system solving (general case)

In this section we will describe a probabilistic algorithm to solve a Diophantine linear system in the general, not necessarily full row rank case. In fact, the algorithm will compute, with prescribed probability of success, a solution with minimal denominator, if one exists. When the system has a Diophantine solution, then a solution with minimal denominator is a Diophantine solution.

We present the algorithm first for the case $R = \mathbb{Z}$. The result for $R = F[X]$ is analogous and will be given in the sequel.

The algorithm is presented in Figure 3. The idea is to first extract a full row rank subsystem which has the same solution space. This subsystem is then solved using the algorithm indicated in Section 2. Finally, it will be checked if the solution found for the subsystem is also a solution of the original system.

In algorithm DiophantineSolver we have used the following functions:

- NumberOfPrimes(A) $\rightarrow 2\lfloor \min(n,m)\log_2(\min(n,m)^{1/2}m\|A\|)\rfloor$;
- NumberOfIterations(u,ϵ) $\rightarrow \lceil \log_{(8/7)}(2\log_2(u)/\epsilon)\rceil$;
- ExtendedGCD(u,v) $\rightarrow (d,s,t)$ with $d = \gcd(u,v) = su+tv$, $|s| \le |v|$, $|t| \le |u|$.

We also use algorithm Split, described in Figure 2.

Algorithm Split(v,u)
Input: $v,u \in R$.
Output: $t \in R$ such that $t|u$ and for all primes p we have:

$$\begin{cases} p|t & \Rightarrow & p\not|\, v \\ p|u/t & \Rightarrow & p|v \end{cases}$$

$x := v$;
$t := u$;
while $x \neq 1$ **do**
  $x := \gcd(x,t)$;
  $t := t/x$
od;
return t

Figure 2: Algorithm Split

Theorem 23 *($R = \mathbb{Z}$) Algorithm* Split *is correct. When $|v| < |u|$, its cost is bounded by $O\left((\log|u|)^2\right)$ bit operations.*

Proof: The correctness of the algorithm is evident. Let $x_0,\ldots,x_s$ be the sequence of subsequent values of x. Since always $|t| \le |u|$, the cost of the algorithm is bounded by $O(\sum_{i=0}^{s-1}(\log|u|)(\log|x_i|)) = O((\log|u|)(\log|x_0x_1\cdots x_{s-1}|))$ bit operations. Since $x_0x_1 1\cdots x_{s-1}|u$, the result follows. •

Lemma 24 *For $u,v,w \in R$:*

$$\gcd(v + \text{Split}(v,u)w, u) = \gcd(v,w,u).$$

Proof: Since for $d|u,v$ we have $\text{Split}(v/d,u/d) = \text{Split}(v,u)$ it suffices to prove the lemma when $\gcd(v,w,u) = 1$. Let $p|u$ be prime. When $p|v$, then $p\not|\,\text{Split}(v,u),w$ and when $p\not|\,v$, then $p|\text{Split}(v,u)$. From this the lemma follows. •

Algorithm DiophantineSolver(A,b,ϵ)
Input: $A \in R^{n\times m}$, $b \in R^n$, $\epsilon > 0$.
Output: Either NIL or a rational solution of $Ax = b$.

(1) [Initialization:]
  $s :=$ NumberOfPrimes(A);
  $S :=$ set of s primes;

(2) [Find rank and determine subsystem:]
  $M := \lceil \log_2(2/\epsilon)\rceil$;
  $r := -1$;
  for i **to** M **do**
    $\hat{p} :=$ a random element from S;
    $\hat{r} :=$ the rank of A modulo $\hat{p}$;
    if $\hat{r} > r$ **then**
      $p := \hat{p}$;
      $r := \hat{r}$
    fi
  od;
  $Q,P :=$ submatrices of permutation matrices such that QAP is an $r \times r$ submatrix of A which is nonsingular modulo p;
  $B := QA$;
  $c := Qb$;

(3) [Get initial solution:]
  $q :=$ RationalSolver(BP,c,p);
  $u :=$ least common multiple of denominators in q;
  $x := P(uq)$;

(4) [Refine solution:]
  $N :=$ NumberOfIterations(u,ϵ);
  $y := 0$;
  $v := 0$;
  for i **to** N **while** $\gcd(u,v) \neq 1$ **do**
    Comment Now $Bx = uc$, $By = vc$
    $P :=$ a random matrix from $\{0,1\}^{m\times r}$;
    $p :=$ a random element from S;
    $q :=$ RationalSolver(BP,c,p);
    if $q \neq$ NIL **then**
      $w :=$ least common multiple of denominators in q;
      if $\gcd(v,w,u) \neq \gcd(v,u)$ **then**
        $t :=$ Split(v,u);
        $z := P(wq)$;
        $y := y + tz$;
        $v := v + tw$
      fi
    fi
  od;
  $(d,s,t) :=$ ExtendedGCD(u,v);
  $x := sx + ty$;

(5) [Check solution:]
  if $Ax = db$ **then**
    return (x,d)
  else
    return NIL
  fi

Figure 3: Algorithm DiophantineSolver

Theorem 25 *($R = \mathbb{Z}$) Suppose $Ax = b$ has a rational solution. Then* DiophantineSolver(A, b, ϵ) *will return a rational solution $(\hat{x}, \hat{u})$ of $Ax = b$ with minimal denominator, with probability at least $1 - \epsilon$.*

Proof: Assume that $Ax = b$ has a rational solution and let t be the rank of A. Then A has a $t \times t$ nonsingular submatrix $\hat{A}$. From Hadamard's bound we see that the $\det(\hat{A})$ has at most $s/2$ different prime divisors. When $\hat{p}$ does not divide $\det(\hat{A})$, then the $\hat{r}$ computed for $\hat{p}$ is equal to t. So the probability that $\hat{r} < t$, for a random $\hat{p}$ from S, is at most $1/2$. So the probability that the rank r computed in step (2) equals t is at least $1 - (1/2)^M \geq 1 - \epsilon/2$.

Since, in step (3), BP is nonsingular modulo p, RationalSolver(BP, c, p) will return a solution. So after step (3), u is well defined and (x, u) is a rational solution of $Bx = c$.

Let $\hat{p} \in \mathbb{Z}$ be prime. Suppose $P \in \{0,1\}^{m \times n}$ is a good preconditioner for $Bx = c$ with respect to $\hat{p}$, $p \in \mathbb{Z}$ is a prime such that $p \nmid \det(BP)$, $q =$ RationalSolver(BP, c, p) and w is the denominator of q. Then $(P(wq), w)$ is a rational solution of $Bx = c$ and $\mathrm{ord}_{\hat{p}}(w) = \mathrm{ord}_{\hat{p}}(d(B, c))$.

For a random P from $\{0,1\}^{m \times n}$ we have that P is a good preconditioner for $Bx = c$ with respect to $\hat{p}$ with probability $> 1/4$. When BP is nonsingular, we see from Hadamard's bound that $\det(BP)$ has at most $s/2$ different prime divisors and so, for a random p from S, the probability that $p | \det(BP)$ is at most $1/2$. We see that, in one iteration of the loop in step (4), we get with probabiltiy at least $1/8$, a rational solution (z, w) of $Bx = c$ such that $\mathrm{ord}_{\hat{p}}(w) = \mathrm{ord}_{\hat{p}}(d(B, c))$. When at the end of step (4), we still have $\mathrm{ord}_{\hat{p}}(d) > \mathrm{ord}_{\hat{p}}(d(B, c))$, then we never had $\mathrm{ord}_{\hat{p}}(w) = \mathrm{ord}_{\hat{p}}(d(B, c))$ in the loop. This happens with probability at most $(7/8)^N$. So the probability that at the end of step (4), we still have $\mathrm{ord}_{\hat{p}}(d) > \mathrm{ord}_{\hat{p}}(d(B, c))$ for some prime divisor $\hat{p}$ of u is at most $\log_2(u)(7/8)^N \leq \epsilon/2$, since u has at most $\log_2(u)$ prime divisors. So the probability that $\mathrm{ord}_{\hat{p}}(d) = \mathrm{ord}_{\hat{p}}(d(B, c))$ for all primes $\hat{p}$ is at least $1 - \epsilon/2$.

When the computed rank r of A equals t, then a solution with minimal denominator of $Bx = c$ is also a solution with minimal denominator of $Ax = b$. So the probability that DiophantineSolver(A, b, ϵ) will return a solution of $Ax = b$ with minimal denominator is at least $(1 - \epsilon/2)^2 > 1 - \epsilon$. •

Theorem 26 *($R = \mathbb{Z}$) The cost of algorithm* DiophantineSolver *is bounded by*

$$O\left((nmr(\log m + \log ||A||)^2 + m(\log ||b||)^2) \cdot N\right)$$

bit operations where $N = O(\log(1/\epsilon) + \log r + \log\log ||A||)$ and r is the rank of A. The solution vector will satisfy $\log ||x|| = O(r \log(m||A||) + \log ||b||)$.

Proof: For S we take the set of first s primes. From [12, Corollary 1] it follows that for $l \geq 6$ the number of primes $\leq l(\log l)^2$ is at least l. From this we can derive that primes in S have length bounded by $O(\log m + \log\log ||A||)$ bits; we use this length bound implicitly throughout the rest of the proof. The set S can be found in the alloted time using standard sieving techniques [2, Theorem 9.8.1].

To compute the rank of A modulo $\hat{p}$ — and determine Q and P — first reduce all entries of A modulo $\hat{p}$ and then perform Gaussian elimination modulo $\hat{p}$. This requires $O(nm(\log ||A||)(\log \hat{p}) + nmr(\log \hat{p})^2)$ bit operations. It follows that step (2) can be completed in the alloted time.

Note that step (3) can be considered as one iteration of the loop in step (4). Since $||BP|| \leq m||A||$ and $||c|| \leq ||b||$, the cost of one call to RationalSolver(BP, c, p) is bounded by $O(r^3(\log(m||A||))^2 + r(\log ||b||)^2)$ bit operations. By Fact 18, $w \leq r^{r/2}(m||A||)^r$ and $||z|| \leq r r^{r/2}(m||A||)^{r-1}||b||$. In step (3), P is a permutation matrix, so $||BP|| \leq ||A||$ which leads to slightly better bounds for u and $||x||$. By Fact 18, $u \leq r^{r/2}||A||^r$ and $||x|| \leq r^{r/2}||A||^{r-1}||b||$; the claimed bound for N follows from this bound for u. So far, we have that the bit length of u, $||x||$ and each w, $||z||$ is bounded by $O(r \log(m||A||) + \log ||b||)$. Now we show that the size of v and $||y||$ remains small. Note that v and $||y||$ will be modified only if $\gcd(v, w, u) < \gcd(v, u)$; the number of times this can happen is bounded by $\log_2 u$. From this, and the bounds previously established for u, w, $||z||$, and the fact that $0 \leq t < u$, it follows easily that the bit length of v and of entries in y remains bounded by $O(r \log(m||A||) + \log ||b||)$. From these bounds on the size of numbers, the claimed size bound for the output vector $||x||$, as well as the running time for steps (4) and (5), follows easily. •

Now we consider the case when $R = F[X]$, F a field. The fuctions used in DiophantineSolver can be defined as follows:

- NumberOfPrimes$(A) \to 2\min(n, m)\deg(A)$;
- NumberOfIterations$(u, \epsilon) \to \lceil \log_{(8/7)}(2\deg(u)/\epsilon) \rceil$;
- ExtendedGCD(u, v)
  $\to (d, s, t)$ with $d = \gcd(u, v) = su + tv$,
  $\deg(s) < \deg(v)$, $\deg(t) < \deg(u)$.

Theorem 27 *($R = F[X]$) Algorithm* Split *is correct. When $\deg v < \deg u$, its cost is bounded by $O\left((\deg u)^2\right)$ field operations.*

Proof: Analogous to the proof of Theorem 23. •

Theorem 28 *($R = F[X]$) Suppose $Ax = b$ has a rational solution. Then* DiophantineSolver(A, b, ϵ) *will return a rational solution $(\hat{x}, \hat{u})$ of $Ax = b$ with minimal denominator, with probability at least $1 - \epsilon$.*

Proof: The proof is similar to the proof of Theorem 25. •

Theorem 29 *($R = F[X]$) The cost of algorithm* DiophantineSolver *is bounded by*

$$O\left((nmr(\deg A)^2 + m(\deg b)^2) \cdot N\right)$$

field operations where $N = O(\log 1/\epsilon + \log r + \log\deg A)$ and r is the rank of A. The solution vector will satisfy $\deg x = O(r \deg A + \deg b)$.

Proof: The analysis is very similar to the proof of Theorem 26. We only have to say a few words about the construction of S. Let $q = \#F$.

When $s \leq q$, we can take for S the polynomials $X - a$, where a belongs to some subset of F of size s.

Otherwise compute l such that $q^{l-1}/(2(l-1)) < s$ and $q^l/(2l) \geq s$. It is easy to see that $l = O(\log_q s)$. From [10, Exercise 3.27] it follows that there are at least s monic irreducible polynomials of degree l over F.

When $l = 2$, we can compute S by eliminating $\{(X - a)(X - b) | a, b \in F\}$ from the set of all monic polynomials of degree 2. Since $\#F < s$ this can be done in the allotted time.

When $l > 2$, we compute the set of all monic irreducible polynomials of degree l. To check whether a polynomial is irreducible we use part of Berlekamp's factorization algorithm (see [6]). The cost of this check is bounded by $O(l^2 \log q + l^3)$ field operations. So the cost for generating S is in this case bounded by $O\left(q^l(l^2 \log q + l^3)\right)$. Since $q < (2s(l-1))^{1/(l-1)}$ and $l/(l-1) < 2$ we can bound the cost of generating S by $O(s^{l/(l-1)} f(\log(s))) = O(s^2)$, where f is a polynomial. From this it follows that we can construct S in the allotted time. •

7 Conclusions, extensions and future work

We have presented a randomized algorithm for computing particular solutions to systems of linear Diophantine equations. The running time is essentially cubic in the dimension of the input system and the space requirement essentially linear in the output size. In this paper we offer detailed pseudo-code; implementation should be straightforward in any system which offers support for long integer arithmetic. We also show how to extend the algorithm to solve linear systems over the ring of polynomials with coefficients from a field.

The algorithm is iterative and randomized. Each iteration involves post-multiplication by a preconditioning matrix which has entries chosen randomly from $\{0, 1\}$. We use a key idea from [7], which is to combine several "random" rational solutions to get a Diophantine solution.

The transfer from structured (Toeplitz) preconditioners in [7] to unstructured (dense) preconditioners in this paper is reminiscent of the transfer from [9] to [5]. In both cases the possible need for ring or field extensions has been eliminated.

A feature of our algorithm is that entries in the preconditioning matrices are chosen randomly from $\{0, 1\}$. We prove that the algorithm will converge in an expected number of $O(\log n + \log\log ||A||)$ iterations using these random $\{0, 1\}$-matrices. A possible modification is to choose the elements of the random matrices from a slightly larger set. Doing this, we can show that the algorithm will converge in an expected *constant* number of iterations, independent of the size of the input system. In this way, the complexity of the algorithm is improved by a factor $O(\log n + \log\log ||A||)$. Another modification we have explored is to choose part of the random preconditioning matrices fixed; this will improve the probability that the preconditioning is effective. A thorough analysis, in the same spirit as in this paper, of the effectiveness of random matrices as described above will be given, together with applications, in the near future.

The following questions are still open and need some further study. Are the Diophantine solutions generated by our algorithm in some sense randomly distributed? Can one make an analysis, similar to the one in Section 4, for structured matrices? This might be useful to improve the results in [7].

The running time estimate for our Diophantine solver is dominanted by the cost of solving rational systems. We have recalled how to accomplish this in $O^{\sim}(n^3(\log \beta)^2)$ bit operations using linear p-adic lifting. Our analysis assumes standard, quadratic integer arithmetic. We remark that this analysis can be improved using subquadratic integer arithmetic. The idea is to use quadratic lifting to first construct the inverse of A modulo p^k for $k = \Theta(\log_p \beta)$, and then solve the system using linear p^k-adic lifting. Using this approach we can prove a running time of $O^{\sim}(n^3 M(\log \beta))$ bit operations. The details of the analysis will be presented in a future paper.

Currently, the algorithm presented here does not certify inconsistency of an input system which does not admit a solution. In [8] a randomized algorithm is described which solves this problem. Another possibility for future work is to combine the techniques developed in this paper with those in [8] to get an algorithm — with running time essentially cubic in the dimension of the system — that will either compute a Diophantine solution or will certify that such a solution does not exist.

References

[1] Abbott, J., Bronstein, M., and Mulders, T. Fast deterministic computation of determinants of dense matrices. These proceedings.

[2] Bach, E., and Shallit, J. *Algorithmic Number Theory*, vol. 1 : Efficient Algorithms. MIT Press, 1996.

[3] Collins, G. E., and Encarnación, M. J. Efficient rational number reconstruction. *Journal of Symbolic Computation 20* (1995), 287—297.

[4] Dixon, J. Exact solution of linear equations using p-adic expansions. *Numer. Math. 40* (1982), 137–141.

[5] Eberly, W. Processor-efficient parallel matrix inversion over abstract fields: Two extensions. In *Second Int'l Symp. on Parallel Symbolic Computation: PASCO '97* (1997), M. Hitz and E. Kaltofen, Eds., ACM Press, pp. 38–45.

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

[7] Giesbrecht, M. Efficient parallel solution of sparse systems of linear diophantine equations. In *Second Int'l Symp. on Parallel Symbolic Computation: PASCO '97* (1997), M. Hitz and E. Kaltofen, Eds., ACM Press, pp. 1–10.

[8] Giesbrecht, M., Lobo, A., and Saunders, B. D. Certifying inconsistency of sparse linear systems. In *Proc. Int'l. Symp. on Symbolic and Algebraic Computation: ISSAC '98* (1998), O. Gloor, Ed., pp. 113—119.

[9] Kaltofen, E., and Pan, V. Processor-efficient parallel solution of linear systems II: The general case. In *Proc. 33rd IEEE Symposium on Foundations of Computer Science* (Pittsburgh, USA, 1992), pp. 714–723.

[10] Lidl, R., and Niederreiter, H. *Finite Fields.* Addison-Wesley, 1983.

[11] Newman, M. *Integral Matrices.* Academic Press, 1972.

[12] Rosser, J. B., and Schoenfeld, L. Approximate formulas for some functions of prime numbers. *Ill. J. Math. 6* (1962), 64—94.

[13] Villard, G. *Calcul Formel et Parallélisme: Resolution de Systemes Lineaires.* PhD thesis, L'Institut National Polytechnique de Grenoble, 1988.

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