Report

Certified dense linear system solving

Author(s):
Mulders, Thom; Storjohann, Arne

Publication Date:
2000

Permanent Link:
https://doi.org/10.3929/ethz-a-006654225

Rights / License:
In Copyright - Non-Commercial Use Permitted
Certified Dense Linear System Solving

T. Mulders and A. Storjohann

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

Abstract

The following problems related to linear systems are studied: finding a diophantine solution; finding a rational solution; proving no diophantine solution exists; proving no rational solution exists. These problems are reduced, via randomization, to that of computing an expected constant number of rational solutions of square nonsingular systems using adic lifting. The bit complexity of the latter problem is improved by incorporating fast arithmetic and fast matrix multiplication. The resulting randomized algorithm for certified dense linear system solving has substantially better asymptotic complexity than previous algorithms for either rational or diophantine linear system solving.

1. Introduction

Finding a particular solution to a system of linear equations is a classical mathematical problem. In the literature we typically find separate treatments for two versions of the problem. The first version — *rational system solving* — can be stated as follows: given an integer matrix $A$ and vector $b$, find a rational vector $x$ that satisfies $Ax = b$. The second version — *diophantine system solving* — asks for an integer vector $x$ that satisfies $Ax = b$. Recently, Giesbrecht (1997) has shown that the latter problem is computationally not essentially more difficult than the first; finding a diophantine solution of a linear system can be reduced, up to logarithmic factors and via randomization, to finding an expected logarithmic number of rational solutions of the same system.

Of course, a given system might admit a rational but not diophantine solution for $x$, or even be inconsistent over the rationals. In Giesbrecht et al. (1998), an extension of the algorithm from Giesbrecht (1997) is described which not only computes a possible diophantine solution but eventually also certifies the non-existence of such or even the non-existence of a rational solution. The studies in Giesbrecht (1997) and Giesbrecht et al. (1998) focus on the case of sparse
structured linear systems, with an emphasis also on algorithms which admit a
good coarse grain parallelization.

This paper considers the problems mentioned above for the case of dense and
unstructured linear systems, with an emphasis on sequential complexity. This
change of focus leads us to a different approach as compared to Giesbrecht (1997)
— both for achieving the randomized reduction* and for solving the rational
linear systems.† Also, we propose a natural generalization of linear system solving
that encompasses all of the situations summarized above (i.e. existence or non–
existence of solutions). We define this generalization now.

Let \( x \) satisfy \( Ax = b \). If \( d \) is the smallest positive integer such that \( dx \) is
integral, and \( d \) is minimal among all solutions to the system, then we call \( x \) a
minimal denominator solution of the system. The problem we solve is

- to find a minimal denominator solution of the system together with a cer-
tificate which proves minimality, or
- to prove that no rational solution exists and to produce also a certificate of
inconsistency.

We call this certified linear system solving. Note that a given system admits a
diophantine solution precisely when the system is consistent and has minimal
denominator one.

Many systems arising in practice are rectangular and/or singular. The algo-
rithm for certified linear system solving that we present here is designed espe-
cially to handle these cases and will be analyzed in terms of the five parameters
\( n, m, r, ||A||, \text{ and } ||b||. \) 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 \( ||b|| \)). To summarize
results here in the introduction, let \( n \) be the larger of the row or column dimen-
sion of a linear system \( Ax = b \) and let \( \beta = \max(||A||, ||b||) \). If \( Ax = b \) admits a
solution for \( x \), it admits a solution whose entries have both numerator and
denominator bounded in length by \( O(n(\log n + \log \beta)) \) bits—this bound is tight
in the worst case. Under the assumption of standard, quadratic integer arithmetic
the algorithm uses an expected number of \( O(n^2(\log n + \log \beta)^2) \) bit operations to
return a minimal denominator solution together with this size bound. Moreover,
the algorithm requires additional storage for only \( O(n^2(\log n + \log \beta)) \) bits.

The main contribution of this paper is to reduce, via randomization, the cer-
tified linear system solving problem described above for the case of a dense
input system with arbitrary shape and rank, to an expected constant number
of instances of the conceptually simpler problem of finding the unique rational
solution of a square nonsingular system. The global technique of the reduction
is similar to that given for sparse linear diophantine systems presented in Gies-
brecht (1997). The key idea is to compute a small number of rational solutions

*Dense preconditioners and algebraic counting arguments here vs. Toeplitz conditioners
over a ring extensions and use of the Schwartz-Zippel lemma.

†adic lifting here vs. Wiedemann's algorithm combined with homomorphic imaging.
of perturbations of the input system which can then hopefully be combined to obtain a diophantine solution. For example, in our case, the perturbed systems are chosen of the form $APx = b$, where entries in $P$ are chosen uniformly and randomly from subset of $\mathbb{Z}$.

In Section 2 we give a high level description of the algorithm for certified linear system solving. The algorithm is described in the general setting of principal ideal domains and we assume we have a subroutine for solving nonsingular rational systems over the domain. This section gives the mathematical background needed to prove the correctness of the algorithm. In Section 3 we develop the necessary mathematical background to analyze the performance of the algorithm; this section contains some general probability results concerned with randomly chosen matrices. In Section 4 we apply the results from Section 3 in order to get bounds on the expected number of perturbed systems needed during the course of the algorithm.

In Section 5 we specialize to the case of integer and polynomial matrices and estimate the cost of the algorithm in these cases. In order to avoid intermediate expression swell we have to adjust the algorithm from Section 2. The dominant cost is to solve nonsingular rational systems; this can be accomplished in $O(n^3(\log n + \log \beta)^2)$ bit operations using $p$-adic lifting as described in Dixon (1982). This assumes standard matrix and integer arithmetic.

In Section 5 we also show how to introduce fast arithmetic and matrix multiplication into the lifting algorithm, thus improving the asymptotic running time to $O(n^t \beta)$, where $t$ depends on the algorithm for matrix multiplication. For example, if multiplying two $n \times n$ matrices costs $O(n^\theta)$ operations, and we assume the current record $\theta \approx 2.376$ by Coppersmith and Winograd (1990), then $t = 2.761$. Incorporating the currently best rectangular matrix multiplication achieves $t = 2.69$ (see Huang and Pan (1997)).

Preliminary version of some of the results here are presented in Mulders and Storjohann (1999). There we gave a Monte Carlo algorithm for the computation of a solution with minimal denominator. That algorithm converged to a solution with high probability using $O(\log n + \log \log \beta)$ perturbed rational systems. New in this paper is the computation of certificates, a thorough study of the effectiveness of the dense preconditioners, and the incorporation of fast arithmetic and matrix multiplication into the adic lifting algorithm.

2. The algorithm

We begin this section by explaining the key idea of the algorithm using an integer matrix example. Then we give the basic results on which correctness is based. Finally, we give a detailed description of the algorithm in the general case; in Section 5 this algorithm will be specialized to the case of integer and polynomial case.

Let $A \in \mathbb{Z}^{n \times m}$ be an integer matrix and $b \in \mathbb{Z}^n$ an integer vector having the
same number of rows. For the linear system of equations $Ax = b$ there are three possibilities:

1. The system has no rational solution, that is, there is no $x \in \mathbb{Q}^m$ such that $Ax = b$.

2. The system has a rational solution but no diophantine solution, that is, there is an $x \in \mathbb{Q}^m$ such that $Ax = b$ but there exists no $x \in \mathbb{Z}^m$ such that $Ax = b$.

3. The system has a diophantine solution, that is, there is an $x \in \mathbb{Z}^m$ such that $Ax = b$.

**Example 1:** Let

$$A = \begin{bmatrix} 2 & 7 & 5 & 1 \\ 1 & 11 & 10 & 8 \\ 3 & 18 & 15 & 9 \end{bmatrix} \in \mathbb{Z}^{3 \times 4}.$$ 

Then $Ax = b$ has

1. no rational solution for $b = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^t$;

2. a rational (e.g. $x = y = \begin{bmatrix} 7/10 & -1/30 & -1/30 & 0 \end{bmatrix}^t$) but no diophantine solution for $b = \begin{bmatrix} 1 & 0 & 1 \end{bmatrix}^t$;

3. a diophantine solution (e.g. $x = \begin{bmatrix} 0 & -1 & 1 & 1 \end{bmatrix}^t$) for $b = \begin{bmatrix} -1 & 7 & 6 \end{bmatrix}^t$.

Suppose that in the first case of Example 1 the system would have a rational solution $x$. Since for $q = \begin{bmatrix} -1 & -1 & 1 \end{bmatrix}$ we have $qA = 0$, we would then have $qb = q(Ax) = (qA)x = 0x = 0$. However, $qb = -1$, so $q$ certifies that the system has no solution.

In the second case of Example 1 there are many more rational solutions, e.g. $x = \tilde{y} = \begin{bmatrix} -11/45 & 19/45 & -11/45 & -11/45 \end{bmatrix}^t$. We see that the solutions $y$ and $\tilde{y}$ have denominators 30 and 15. By taking the linear combination $\tilde{y} = -2y + 3\tilde{y} = \begin{bmatrix} -32/15 & 4/3 & -2/3 & -11/15 \end{bmatrix}^t$ we get another solution of the system. In fact, $\tilde{y}$ has denominator 15, which is smaller than the two previous denominators.

Now suppose that $x$ is a rational solution of the system with denominator $d$. For $z = \begin{bmatrix} 0 & -3 & 11/3 \end{bmatrix}$ we have $zA \in \mathbb{Z}^3$ and thus $d(zb) = d(zAx) = (zA)(dx) \in \mathbb{Z}$. So $d$ is a multiple of the denominator of $zb = 11/3$. We see that 3 divides the denominator of any rational solution of the system and thus also the denominator of a rational solution with minimal denominator. In the same way we have for $\tilde{z} = \begin{bmatrix} 0 & -19/5 & 18/5 \end{bmatrix}$ that $\tilde{z}A \in \mathbb{Z}^3$ and $\tilde{z}b = 18/5$. So 5, and thus also 15, divides the denominator of a rational solution with minimal denominator. Since $\tilde{y}$ has denominator 15, we see that $\tilde{y}$ is a rational solution with minimal denominator. By taking the linear combination $\tilde{z} = z - \tilde{z} = \begin{bmatrix} 0 & 4/5 & 1/15 \end{bmatrix}$ we get $\tilde{z}A \in \mathbb{Z}^3$ and $\tilde{z}b = 1/15$, which certifies that indeed 15 must divide the denominator of a rational solution with minimal denominator.
In Mulders and Storjohann (1999) we give a Monte Carlo probabilistic algorithm that computes, in case the system has a rational solution, a rational solution with minimal denominator. In this section we will turn this algorithm into a Las Vegas probabilistic algorithm and moreover the algorithm will produce certificates that enables one to easily verify the outcome of the algorithm. More specific, the algorithm will return one of

1. ("no solution", q), where
   - \( q \in \mathbb{Q}^{1 \times n} \);
   - \( qA = 0 \);
   - \( qb \neq 0 \).

2. \((y, z)\), where
   - \( y \in \mathbb{Q}^m \);
   - \( Ay = b \);
   - \( z \in \mathbb{Q}^{1 \times n} \);
   - \( zA \in \mathbb{Z}^m \);
   - \( zb \) and \( y \) have the same denominator.

In the first case the fact that the system has no solution can be certified by verifying that \( qA = 0 \) and \( qb \neq 0 \). In the second case the fact that \( y \) is a solution with minimal denominator can be certified by verifying that \( Ay = b \), \( zA \in \mathbb{Z}^m \) and that \( zb \) and \( y \) have the same denominator. When the minimal denominator happens to be 1, the computed solution is in fact a diophantine solution.

Above, we obtained a rational solution with minimal denominator as a linear combination of other rational solutions and a vector certifying the minimality of the solutions denominator as a linear combination of vectors that only certify the presence of particular divisors in the minimal denominator. Taking a linear combination of several rational solutions and of several certificates is the key idea used in computing a rational solution with minimal denominator and its certificate.

In what follows we will use the framework of principal ideal domains so that our results are applicable in domains different from the integers.

Let \( R \) be a principal ideal domain and \( F \) its quotient field. For \( v, w \in F \) we say that \( v \) and \( w \) are associates (notation: \( v \sim w \)) if there is a unit \( u \) in \( R \) such that \( v = uw \). We assume that for every equivalence class of associate elements we have a unique representative and that this representative is 1 for the class of units in \( R \). In this way we get a unique generator \( d(I) \in R \) for every ideal \( I \) of \( R \) and this allows us to use greatest common divisors and least common multiples without ambiguity. Let \( A \in \mathbb{R}^{n \times m} \) of rank \( r \), \( b \in \mathbb{R}^n \) and \( \delta = \min(n, m) \).

**Definition:** Let \( x \in F^m \). It is easy to see that the set of all \( v \in R \) such that \( vx \in \mathbb{R}^m \) is an ideal \( I \) of \( R \). We denote \( d(I) \) by \( d(x) \) and call it the denominator of \( x \). By \( n(x) \) we denote \( d(x)x \) and call it the numerator of \( x \).
Definition: $y \in F^m$ such that $Ay = b$ is called a rational solution of the linear system $Ax = b$. If in addition $d(y) = 1$, then we call $y$ a diophantine solution of the system.

Definition: If the system $Ax = b$ has a rational solution, let $I$ be the ideal of $R$ generated by the set of denominators of all rational solutions of $Ax = b$. We denote $d(I)$ by $d(A, b)$.

d$(A, b)$ is the minimal denominator that a rational solution of $Ax = b$ can have in the sense that $d(A, b)$ divides $d(y)$ for any rational solution $y$ of $Ax = b$. Clearly, if $Ax = b$ has a diophantine solution, then $d(A, b) = 1$.

The following lemma shows how we can take a linear combination of two rational solutions in order to get a new rational solution with a smaller denominator. This idea is also used in Giesbrecht (1997).

Lemma 2.1: Let $y, \hat{y} \in F^m$ be rational solutions of $Ax = b$. Let $d, s, t \in R$ such that $d = \gcd(d(y), d(\hat{y})) = sd(y) + td(\hat{y})$ and

$$\tilde{y} = \frac{sd(y)y + td(\hat{y})\hat{y}}{d}.$$  

Then $\tilde{y}$ is a rational solution of $Ax = b$.

Proof:

$$A\tilde{y} = A(sd(y)y + td(\hat{y})\hat{y})/d = (sd(y)Ay + td(\hat{y})A\hat{y})/d = (sd(y)b + td(\hat{y})b)/d = ((sd(y) + td(\hat{y}))/d)b = b,$$

so $\tilde{y}$ is a rational solution. □

Note that in Lemma 2.1 the denominator of $\tilde{y}$ divides $d = \gcd(d(y), d(\hat{y}))$. From Lemma 2.1 it follows that there exists a rational solution whose denominator is $d(A, b)$ and thus if $d(A, b) = 1$, the system has a diophantine solution.

Definition: A rational solution $y$ of $Ax = b$ with $d(y) = d(A, b)$ is called a solution with minimal denominator.

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

Lemma 2.2: Let $P \in R^{m \times n}$. If $y$ is a rational solution of $APx = b$, then $Py$ is a rational solution of $Ax = b$. 
By taking linear combinations of several rational solutions as in Lemma 2.1 we get rational solutions with decreasing denominators. If we find in that way a diophantine solution we know that we have a solution with minimal denominator and we can stop. If, however, the system admits no diophantine solutions we can never be sure that the computed solution whose denominator is minimal so far is in fact a solution with minimal denominator. In order to solve this problem we need a way to certify that a solutions denominator is really minimal. This can be accomplished by the following lemma.

Lemma 2.3: Suppose $Ax = b$ has a rational solution and let $z \in F^{1 \times n}$ such that $zA \in R^{1 \times m}$. Then $d(zb)$ divides $d(A, b)$.

Proof: Let $y$ be a rational solution of $Ax = b$ with minimal denominator. Then
\[ d(A, b)(zb) = d(A, b)zAy = (zA)(d(A, b)y) \in R, \]
since $zA \in R^{1 \times m}$ and $d(A, b)y \in R^n$. \qed

Lemma 2.3 states that $z$ certifies the factor $d(zb)$ of $d(A, b)$.

The following lemma shows how we can take a linear combination of two certifying vectors in order to get a new vector certifying a larger factor of $d(A, b)$.

Lemma 2.4: Let $z, \hat{z} \in F^{1 \times n}$ such that $zA, \hat{z}A \in R^{1 \times m}$. Write $zb = n/d$ and $\hat{z}b = \hat{n}/\hat{d}$, where $\gcd(n, d) = \gcd(\hat{n}, \hat{d}) = 1$. Let $g = \gcd(d, \hat{d}), l = \text{lcm}(d, \hat{d}), e, s, t \in R$ such that
\[ e = \gcd\left(\frac{\hat{d}}{g}, \frac{n}{g}\right) = sn\frac{\hat{d}}{g} + t\hat{n}\frac{d}{g} \]
and $\hat{z} = sz + t\hat{z}$. Then $\hat{z}A \in R^{1 \times m}$ and $d(\hat{z}b) = l$.

Proof: $\hat{z}A = (sz + t\hat{z})A = s(zA) + t(\hat{z}A) \in R^{1 \times m}$ and
\[
\begin{align*}
\hat{z}b &= (sz + t\hat{z})b \\
&= s(zb) + t(\hat{z}b) \\
&= \frac{s}{d}n + t\hat{n}\frac{d}{\hat{d}} \\
&= \frac{sn\hat{d} + t\hat{n}d}{dd} \\
&\sim \frac{sn\hat{d} + t\hat{n}d}{gl} \\
&= e/l.
\end{align*}
\]
Let $p \in R$ be prime. If $p$ divides $d$ but not $\hat{d}$, then $p$ does not divide $n\hat{d}$ and thus $p$ does not divide $e$. If $p$ divides $\hat{d}$ but not $d$, then $p$ does not divide $\hat{n}d$ and thus $p$ does not divide $e$. If $p$ divides both $d$ and $\hat{d}$, then $p$ does not divide $n$ and $\hat{n}$. Since also $\gcd(d/g, \hat{d}/g) = 1$, $p$ does not divide $e$. So $\gcd(e, l) = 1$ and thus $d(e/l) = l$. \qed
To get another $z \in F^{1 \times n}$ such that $zA \in R^{1 \times m}$, we apply the following lemma for different random choices of $P$ and $t$.

**Lemma 2.5:** Let $P \in R^{m \times n}$ and $t \in R^{1 \times n}$. If $z \in F^{1 \times n}$ is such that $zAP = t$, then $(d(zA)z)A \in R^{1 \times m}$.

The idea of our algorithm is to take linear combinations of several rational solutions in order to get rational solutions with decreasing denominator. We also take linear combinations of several certifying vectors in order to get vectors certifying increasing factors of $d(A, b)$. When the denominator and certified factor found so far coincide, we know that we have found a solution with minimal denominator and a vector certifying this minimal denominator. Meanwhile, when we find that the system is inconsistent, we try to find a certificate for this using the following lemma (see also Giesbrecht et al. (1998)).

**Lemma 2.6:** Let $q \in F^{1 \times n}$ such that $qA = 0$ and $qb \neq 0$. Then the system $Ax = b$ has no solution.

**Proof:** If $y \in F^m$ is such that $Ay = b$, then $qb = q(Ay) = (qA)y = 0y = 0$. □

In Figure 1 we give a detailed description of the algorithm. The randomly chosen matrices and vectors in Algorithm *MinimalSolution* are chosen uniformly, that is, every element from the set of possible values under consideration has the same probability of being chosen. Determining the matrices $B$ and $C$ and solving the various linear systems in the algorithm can be done by any preferred method, e.g. (fraction-free) Gaussian elimination. When a system to be solved is singular, then NIL will be returned. In the specialized version of the algorithm in the case of integer and polynomial matrices (see Section 5), we will determine $B$ and $C$ using probabilistic modular methods and solve the various linear systems using Hensel-lifting.

**Lemma 2.7:** When Algorithm *MinimalSolution* stops, it returns one of

1. ("no solution", $q$), where $q \in F^{1 \times n}$ such that $qA = 0$ and $qb \neq 0$.
2. $(y, z)$, where $y \in F^m$ and $z \in F^{1 \times n}$ such that $Ay = b$, $zA \in R^{1 \times m}$ and $d(y) = d(zb)$.

In the first case the system $Ax = b$ has no solution and $q$ certifies this. In the second case $y$ is a rational solution with minimal denominator and $z$ certifies this.

**Proof:** First note that since $\text{rank}(B) = \text{rank}(A)$, the system $Ax = b$ is consistent if and only if $Ay_0 = b$. The lemma now follows immediately from Lemmas 2.1,2.3, 2.4 and 2.6. □
**Algorithm** \( \text{MinimalSolution} \)

**Input:** \( A \in R^{n \times m}, b \in R^n \).

**Output:** Either ("no solution", \( q \)) or \( (y, z) \) as described before.

**Comment:** Determine equivalent subsystem
Determine submatrix \( B \in R^{s \times m} \) of \( A \) and corresponding subvector \( c \in R^s \) of \( b \) such that \( s = \text{rank}(B) = \text{rank}(A) \);

**Comment:** Get first solution
Determine nonsingular submatrix \( C \in R^{s \times s} \) of \( B \);
Solve \( C\zeta = c \);
\( y_0 := \text{extension of } \zeta \text{ to } F^m \) by inserting 0’s;

\( U := \text{finite subset of } R; \)

if \( Ay_0 \neq b \) then

**Comment:** Certify inconsistency

do

Choose \( P \in U^{m \times s} \) and \( t \in U^{1 \times n} \);
Solve \( \nu BP = tAP \);

if \( \nu \neq \text{NIL} \) then

\( q := \text{extension of } \nu \text{ to } F^{1 \times n} \) by inserting 0’s;

if \( (q-t)A = 0 \) and \( (q-t)b \neq 0 \) then

return ("no solution", \( q-t \))

fi

fi

do

else

**Comment:** Find minimal solution and certificate

\( y := y_0 \);
\( z := 0 \);

do

Choose \( P \in U^{m \times s} \);
Solve \( BP\xi = c \);

if \( \xi \neq \text{NIL} \) then

**Comment:** Decrease denominator of solution

\( \hat{y} := P\xi \);
\( y := \text{combine } y \text{ with } \hat{y} \) using Lemma 2.1;

**Comment:** Increase known factor of \( d(A, b) \)

Choose \( l \in U^{1 \times s} \);
Solve \( \eta BP = l \);
\( \hat{z} := d(\eta B)\eta \);
\( z := \text{combine } z \text{ with } \hat{z} \) using Lemma 2.4;

fi

until \( d(y) = d(\xi) \);

\( z := \text{extension of } z \text{ to } F^{1 \times n} \) by inserting 0’s;

return \( (y, z) \)

fi

**Figure 1:** Algorithm \( \text{MinimalSolution} \)
In Algorithm MinimalSolution we multiply the matrix $B$ on the right with a random matrix $P$ in order to get a nonsingular matrix $BP$. We call this preconditioning. Solving linear systems involving such matrices $BP$ will result in solutions whose denominators have “random” prime divisors. We also choose random vectors $t$ in order to compute the certificates. In the next section we will develop the necessary background to study the probability that these random choices are successful.

3. Rank properties of random matrices

In the literature one can find a lot of facts concerning rank properties of random matrices of many kinds (Cooper (2000),Gerth III (1986)). The results that we will prove here are in some sense generalizations of some of those facts and specialize to them in particular situations.

We state the results in this section in a general setting so that they can be used in several situations. The coefficients in the matrices we consider are from a field $K$. We also use a finite set $U$ and a map $\phi: U \to K$. In this way we cover several possible applications of our results, e.g.

1. $U \subseteq K$, $\phi$ the inclusion map.
2. $R$ a principal ideal domain, $U$ a finite subset of $R$, $K = R/pR$, where $p$ is a prime in $R$ and $\phi$ the projection map.

The map $\phi$ is supposed to be a non-constant map.

**Notation 1**: Let $K$ be a field and $A$ a matrix over $K$. By $R(A)$ we denote the row span of $A$ over $K$, that is, $R(A)$ is the vector space over $K$ generated by the rows of $A$. Similar, we define the column span $C(A)$ of $A$.

**Proposition 3.1**: Let $K$ be a field, $A \in K^{n \times m_1}$, $B \in K^{n \times m_2}$ and $v \in K^{1 \times m_1}$. Let $t = \text{rank}(A)$ and $s = \text{rank} \left[ \begin{array}{c} A \\ B \end{array} \right]$. Let $U$ be a finite set and $\phi: U \to K$ a map. Let $g$ be the maximum number of elements in the preimage of any element of $K$ under $\phi$. Then

a) if $v \notin R(A)$, then

$$\# \{ u \in U^{1 \times m_2} \mid \left[ \begin{array}{c} v \\ \phi(u) \end{array} \right] \in R \left[ \begin{array}{c} A \\ B \end{array} \right] \} = 0.$$

b) if $v \in R(A)$, then

$$\# \{ u \in U^{1 \times m_2} \mid \left[ \begin{array}{c} v \\ \phi(u) \end{array} \right] \in R \left[ \begin{array}{c} A \\ B \end{array} \right] \} \leq (\#U)^{s-t}g^{m_2-(s-t)},$$

with equality when the preimages of all elements of $F$ have the same size.

**Proof**: The only non-trivial statement of the proposition is b. Deleting a row from $\left[ \begin{array}{c} A \\ B \end{array} \right]$ that is in the row span of the other rows of $\left[ \begin{array}{c} A \\ B \end{array} \right]$ does not change any essential data in the proposition. Neither does any elementary row
operation on \( \begin{bmatrix} A & B \end{bmatrix} \). So we may assume that \( \begin{bmatrix} A & B \end{bmatrix} \) has full row rank, i.e. \( s = n \), and that \( \begin{bmatrix} A & B \end{bmatrix} \) is in reduced row echelon form. Let \((j_1, \ldots, j_n)\) be the rank profile of \( \begin{bmatrix} A & B \end{bmatrix} \). Then \( j_t \leq m_1, j_{t+1} > m_1 \), the first non-zero entry in row \( i \) is on the \( j_t \)-th position and the \( j_t \)-th column is the 0-column, except for a 1 in the \( i \)-th row. A possible configuration for \( \begin{bmatrix} A & B \end{bmatrix} \) could look as follows:

\[
\begin{bmatrix}
1 & * & 0 & * & * & 0 & * & 0 & * & 0 & * \\
0 & 0 & 1 & * & 0 & * & 0 & * & 0 & * \\
0 & 0 & 0 & 0 & 1 & * & 0 & * & 0 & * \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
\]

Suppose \( v \in R(A) \). For \( u \in U^{1 \times m_2} \) we then have:

\[
\begin{bmatrix} v & \phi(u) \end{bmatrix} \in R \begin{bmatrix} A & B \end{bmatrix} \iff \phi(u_j) \text{ equals the } j \text{th coordinate of } (v_{j_1}, \ldots, v_{j_t}, \phi(u_{j_{t+1}-m_1}), \ldots, \phi(u_{j_n-m_1}))B
\]

So, in order that \( \begin{bmatrix} v & \phi(u) \end{bmatrix} \in R \begin{bmatrix} A & B \end{bmatrix} \), \( u_j \in U \) can be anything for \( j \in \{j_{t+1}-m_1, \ldots, j_n-m_1\} \) and they uniquely determine \( \phi(u_j) \) for \( j \in \{1, \ldots, m_2\} \setminus \{j_{t+1}-m_1, \ldots, j_n-m_1\} \). From this \( b \) follows easily.

Note that the bound in part b of Proposition 3.1 is in general not sharp. In many cases the bound is even very pessimistic. This is because for some choices of the \( u_j \) with \( j \in \{j_{t+1}-m_1, \ldots, j_n-m_1\} \) there may exist \( k \in \{1, \ldots, m_2\} \setminus \{j_{t+1}-m_1, \ldots, j_n-m_1\} \) such that there are less than \( g \) different (or even no) \( u \in U \) with \( \phi(u) \) equal to the \( k \)th coordinate of \( (v_{j_1}, \ldots, v_{j_t}, \phi(u_{j_{t+1}-m_1}), \ldots, \phi(u_{j_n-m_1}))B \).

The sharpness of the bound will in general depend on the particular values of \( A, B \) and \( v \). We will now give examples of the two most extreme cases.

**Example 2:** Let \( U = \{0, 1\} \), \( K = \mathbb{Q} \) and \( \phi \) the inclusion map. For

\[
A = \begin{bmatrix} 1 & 0 \\
1 & 1 \\
1 & 1 \\
1 & 1 
\end{bmatrix}, B = \begin{bmatrix} 1 & 0 & 1 & 1 \\
0 & 1 & 2 & 1 \\
0 & 0 & 2 & 1 \\
1 & 1 & 2 & 1 
\end{bmatrix}, \text{ and } v = \begin{bmatrix} 1 & -1 \end{bmatrix},
\]

we have \( \text{rank}(A) = 2, \text{rank} \begin{bmatrix} A & B \end{bmatrix} = 4 \) and \( \# \{u \in U^{1 \times 4} \mid \begin{bmatrix} v & u \end{bmatrix} \in R \begin{bmatrix} A & B \end{bmatrix} \} = 4 \), so the bound is sharp in this case. For

\[
A = \begin{bmatrix} 1 & 2 \\
2 & 1 \\
2 & 2 \\
1 & 0 
\end{bmatrix}, B = \begin{bmatrix} 1 & 1 & 1 & 2 \\
2 & 1 & 2 & 1 \\
1 & 2 & 2 & 1 \\
2 & 2 & 2 & 1 
\end{bmatrix}, \text{ and } v = \begin{bmatrix} 1 & 1 \end{bmatrix},
\]

we have \( \text{rank}(A) = 2, \text{rank} \begin{bmatrix} A & B \end{bmatrix} = 4 \) and \( \# \{u \in U^{1 \times 4} \mid \begin{bmatrix} v & u \end{bmatrix} \in R \begin{bmatrix} A & B \end{bmatrix} \} = 0 \), so the bound is extremely bad in this case.
Corollary 3.1: When we choose in Proposition 3.1 the entries in \( u \) uniformly from \( U \), then the probability that \( \begin{bmatrix} v & \phi(u) \end{bmatrix} \notin R \begin{bmatrix} A & B \end{bmatrix} \) is

\[
\begin{cases} 
1, & \text{if } v \notin R(A); \\
\geq 1 - \left( \frac{g}{\#U} \right)^{m_2 - (s-t)}, & \text{if } v \in R(A),
\end{cases}
\]

with equality when the preimage of all elements from \( F \) have the same size.

As a consequence of the unsharpness of the bound in Proposition 3.1 that we mentioned before, the bound for the probability in Corollary 3.1 will in general not be sharp. This implies that the probabilities that we will encounter in the sequel are in general pessimistic and thus the performance of our algorithm will in practice be better than our analysis predicts.

We will now successively augment rows to a matrix in order to increase its rank. Applying Corollary 3.1 a number of times gives us a bound for the probability of success.

Lemma 3.1: Let \( K \) be a field. Let \( A \in K^{n_1 \times m_1}, B \in K^{n_1 \times m_2} \) and \( C \in K^{n_2 \times m_1} \). Let \( t = \text{rank}(A), s = \text{rank} \begin{bmatrix} A & B \end{bmatrix} \) and \( r = \text{rank} \begin{bmatrix} A \\ C \end{bmatrix} \). Let \( U \) be a finite set and \( \phi: U \rightarrow K \) a map. Let \( g \) be the maximum number of elements in the preimage of any element from \( K \) under \( \phi \). Let \( P \) be the probability that

\[ \text{rank} \begin{bmatrix} A & B \\ C & \phi(D) \end{bmatrix} = s + n_2, \]

when the entries of \( D \in U^{n_2 \times m_2} \) are chosen uniformly from \( U \). Then

\[ P \geq \prod_{i=m_2-n_2+r-s+1}^{m_2-s-t} \left( 1 - \left( \frac{g}{\#U} \right)^i \right), \]

with equality when the preimage of all elements from \( F \) have the same size.

Proof: We choose the rows of \( D \) one after the other. Let \( C_i \) be the first \( i \) rows of \( C \) and \( D_i \) the first \( i \) rows of \( D \). Let \( A_i = \begin{bmatrix} A \\ C \end{bmatrix} \) and \( B_i = \begin{bmatrix} B \\ \phi(D_i) \end{bmatrix} \).

Then \( \text{rank} \begin{bmatrix} A & B \\ C & \phi(D) \end{bmatrix} = s + n_2 \) if and only if \( \text{rank} \begin{bmatrix} A_i & B_i \end{bmatrix} = s + i \) for all \( i \), i.e. every row we add must increase the rank by one. Let \( t_i = \text{rank}(A_i) \) and \( s_i = \text{rank} \begin{bmatrix} A_i & B_i \end{bmatrix} \). Suppose we have chosen \( D_i \) such that \( s_i = s + i \). Let \( v_{i+1} \) be the \( (i+1) \)-th row of \( C \). We want to choose \( u \in U^{1 \times m_2} \) such that \( \text{rank} \begin{bmatrix} A_i & B_i \\ v_{i+1} & \phi(u) \end{bmatrix} = s + i + 1 \), i.e. such that \( \begin{bmatrix} v_{i+1} & \phi(u) \end{bmatrix} \notin R \begin{bmatrix} A_i & B_i \end{bmatrix} \). Let
$P_i$ be the probability that $\begin{bmatrix} v_{i+1} & \phi(u) \end{bmatrix} \notin R \begin{bmatrix} A_i & B_i \end{bmatrix}$. From Corollary 3.1 we get

$$\begin{cases} P_i = 1 & \text{if } v_{i+1} \notin R(A_i); \\
 P_i \geq 1 - \left( \frac{q}{\#U} \right)^{m_2-(s_i-t_i)} & \text{if } v_{i+1} \in R(A_i), \end{cases} \quad (1)$$

with equality when the preimage of all elements from $F$ have the same size.

Since

a) $t_{i+1} = t_i + 1$, if $v_{i+1} \notin R(A_i)$;

b) $t_{i+1} = t_i$, if $v_{i+1} \in R(A_i)$,

we see that case a applies $r-t$ times and that case b applies $n_2-(r-t)$ times.

If we have chosen $u$ such that $\begin{bmatrix} v_{i+1} & \phi(u) \end{bmatrix} \notin R \begin{bmatrix} A_i & B_i \end{bmatrix}$, then $s_{i+1} = s_i + 1$, and so if case a applies, then $s_i - t_i$ does not change and if case b applies, then $s_i - t_i$ is incremented. Since $P = P_1 P_2 \cdots P_{n_2}$ and $s_0 - t_0 = s - t$, the lemma now follows from Equation (1). \hfill \square

**Definition:** Let $K$ be a field and $A \in K^{n \times m}$. We call the set $\{ x \in K^m \mid Ax = 0 \}$ the right–kernel of $A$. $N \in K^{m \times k}$ is called a right–kernel for $A$ if $C(N)$ is the right–kernel of $A$.

In a similar way we define left–kernel.

**Lemma 3.2:** Let $K$ be a field, $A \in K^{n \times m}$ and $B \in K^{m \times k}$. Let $N$ be a right–kernel for $A$. Then

$$\text{rank}(AB) = \text{rank} \begin{bmatrix} N & B \end{bmatrix} - \text{rank}(N).$$

**Proof:** Note that for a matrix $M$, $\text{rank}(M) = \dim(C(M))$. Since $C(AB) = C \begin{bmatrix} A & N & B \end{bmatrix}$, we get:

$$\dim(C(AB)) = \dim \left( C \begin{bmatrix} A & N & B \end{bmatrix} \right)$$

$$= \dim \left( C \begin{bmatrix} N & B \end{bmatrix} \right) - \dim \left( C \begin{bmatrix} N & B \end{bmatrix} \cap C(N) \right)$$

$$= \dim \left( C \begin{bmatrix} N & B \end{bmatrix} \right) - \dim(C(N)).$$

\hfill \square

**Corollary 3.2:** Let $K$ be a field, $W_1 \in K^{n \times m_1}$ and $W_2 \in K^{n \times m_2}$ such that $\begin{bmatrix} W_1 & W_2 \end{bmatrix}$ has full row rank, and $M \in K^{m_1 \times n}$. Let $\begin{bmatrix} N_1 & N_2 \end{bmatrix}$ be a right–kernel for $\begin{bmatrix} W_1 & W_2 \end{bmatrix}$. Let $r_1 = \text{rank}(N_1)$ and $r_2 = \text{rank} \begin{bmatrix} N_1 & M \end{bmatrix}$. Let $U$ be a finite set and $\phi: U \to K$ a map. Let $g$ be the maximum number of elements in the preimage of any element from $F$ under $\phi$. When the entries of $P \in U^{m_2 \times n}$ are chosen uniformly from $U$, then the probability that

$$\begin{bmatrix} W_1 & W_2 \end{bmatrix} \begin{bmatrix} M \\ \phi(P) \end{bmatrix}$$

has rank $n$,
is at least
\[ \prod_{i = r_2 - m_1 + 1}^{n + r_1 - m_1} \left( 1 - \left( \frac{g}{\#U} \right)^i \right), \]
with equality when the preimage of all elements from \( F \) have the same size.

Proof: From Lemma 3.2 it follows that
\[
\text{rank} \left( \begin{bmatrix} W_1 & W_2 \\ \phi(P) \end{bmatrix} \right) = \text{rank} \left( \begin{bmatrix} N_1 & M \\ N_2 & \phi(P) \end{bmatrix} \right) - \text{rank} \left( \begin{bmatrix} N_1 \\ N_2 \end{bmatrix} \right).
\]
Using \( \text{rank} \left( \begin{bmatrix} N_1 \\ N_2 \end{bmatrix} \right) = m_1 + m_2 - n \), the lemma now follows by applying Lemma 3.1 with \( A = N_1^t \), \( B = N_2^t \), \( C = M^t \) and \( D = P^t \). \( \square \)

4. Performance bounds

In this section we will bound the expected number of iterations that Algorithm MinimalSolution has to perform. This bound will depend on whether the system has a solution or not and on the size of the set \( U \). We will use the notation of Algorithm MinimalSolution, that is, if not explicitly stated otherwise all names represent the variables in the algorithm. Remember that \( P, t \) and \( l \) are chosen uniformly from their respective value sets. We will frequently use the following propositions.

**Proposition 4.1:** For \( 0 < x < 1 \) we have
\[
\prod_{i=1}^{\infty} (1 - x^i) \geq 1 - x - x^2.
\]

Proof: From Hardy and Wright (1979, Theorem 358) we get for \( 0 < x < 1 \):
\[
\prod_{i=1}^{\infty} (1 - x^i) = 1 + \sum_{k=0}^{\infty} (-1)^{k+1} \left( x^{(k+1)(3k+2)/2} + x^{(k+1)(3k+4)/2} \right).
\]
Since for odd \( k \) the sum of the \( k \)th and \((k+1)\)th term in the sum is positive, the proposition follows. \( \square \)

**Proposition 4.2:** If the probability of success of an experiment is at least \( p \), then the expected number of experiments one has to perform in order to have success is at most \( 1/p \).

Proof: Let \( N \) be the number of experiments one has to perform in order to
have success and let \( P[E] \) denote the probability that event \( E \) occurs. Then the expected value \( E(N) \) of \( N \) is

\[
E(N) = \sum_{k=1}^{\infty} k P[N = k] = \sum_{k=1}^{\infty} P[N \geq k] \leq \sum_{k=1}^{\infty} (1 - p)^{k-1} = \frac{1}{p}.
\]

\[\square\]

4.1. The inconsistent case

In this section we will consider the case when the system has no rational solution. Then the algorithm has to find a certificate for the systems inconsistency.

**Lemma 4.1:** The probability that \( BP \) is nonsingular is at least

\[
1 - \frac{1}{\#U} - \frac{1}{(\#U)^2}.
\]

**Proof:** Applying Corollary 3.2 with \( K = F \), \( \phi: U \to F \) the inclusion map, \( m_1 = 0 \) and \( W_2 = B \), we see that the probability that \( BP \) is nonsingular is at least

\[
\prod_{i=1}^{s} \left(1 - \left(\frac{1}{\#U}\right)^i\right).
\]

The lemma now follows from Proposition 4.1. \[\square\]

**Lemma 4.2:** When the system \( Ax = b \) has no solution and \( P \) is chosen such that \( BP \) is nonsingular, then the probability that a chosen \( t \) gives rise to a certificate for the inconsistency of the system is at least \( 1 - 1/\#U \).

**Proof:** Let \( N_A \) (resp. \( N_{AP} \)) denote the left–kernel of \( A \) (resp. \( AP \)). Since \( \text{rank}(A) \geq \text{rank}(AP) \geq \text{rank}(BP) = s = \text{rank}(A) \), we have \( \text{rank}(A) = \text{rank}(AP) \) and thus \( \dim(N_A) = n - \text{rank}(A) = n - \text{rank}(AP) = \dim(N_{AP}) \). Since \( N_A \subseteq N_{AP} \) it follows that \( N_A = N_{AP} \).

Let

\[
\psi: \mathbb{F}^{l \times n} \to \mathbb{F}^{l \times n}, \quad t \mapsto q - t,
\]

where \( q \) is as in the algorithm. Then \( \psi \) is a linear map and \( \psi(t) = 0 \) if and only if the \( i \)th coordinate of \( t \) is zero whenever the \( i \)th row of \( A \) is not contained in
B. So \( \dim(\ker(\psi)) = \text{rank}(B) = \text{rank}(A) \) and thus \( \dim(\text{Im}(\psi)) = n - \text{rank}(A) = \dim(N_A) \). Since \( \text{Im}(\psi) \subseteq N_{AP} = N_A \) we get \( \text{Im}(\psi) = N_A \).

Let \( N_{[A \ b]} \) denote the left-kernel of \( \begin{bmatrix} A & b \end{bmatrix} \). Since the system is inconsistent we have rank \( \begin{bmatrix} A & b \end{bmatrix} = \text{rank}(A) + 1 \) and thus \( \dim(N_{[A \ b]}) = \dim(N_A) - 1 \). Since \( N_{[A \ b]} \subseteq N_A \) it follows now from \( \text{Im}(\psi) = N_A \) that \( \dim(\psi^{-1}(N_{[A \ b]})) = n - 1 \).

\( t \in U^{1 \times n} \) gives a certificate for the inconsistency of the system if and only if \( t \not\in \psi^{-1}(N_{[A \ b]}) \). Applying Corollary 3.1 with \( K = F, \phi: U \to F \) the inclusion map, \( m_1 = 0 \) and \( B \in F^{(n-1) \times n} \) a matrix with row span equal to \( \psi^{-1}(N_{[A \ b]}) \) the lemma follows.

Algorithm \textit{MinimalSolution} will find in a particular iteration of the loop a certificate for the inconsistency of the system if \( P \) is chosen such that \( BP \) is nonsingular and \( t \) is chosen such that the corresponding \( q - t \) is a certificate for the inconsistency. We get the following corollary.

**Corollary 4.1:** When the system \( Ax = b \) has no solution, then the probability that a certificate for the inconsistency of the system is found in one iteration of the loop in Algorithm \textit{MinimalSolution} is at least

\[
1 - \frac{2}{\#U} + \frac{1}{(\#U)^3}.
\]

**Proof:** Combining Lemma 4.1 and Lemma 4.2 we see that the probability is at least

\[
\left(1 - \frac{1}{\#U}\right) \left(1 - \frac{1}{\#U} - \frac{1}{(\#U)^2}\right).
\]

\qed

We will now bound the expected number of iterations Algorithm \textit{MinimalSolution} has to perform in order to find a certificate for the systems inconsistency.

**Lemma 4.3:** When the system \( Ax = b \) has no solution, then the expected number of iterations the loop in Algorithm \textit{MinimalSolution} has to perform in order to find a certificate for the inconsistency of the system is at most \( (\#U)^3/((\#U)^3 - 2(\#U)^2 + 1) \).

**Proof:** This follows from Corollary 4.1 and Proposition 4.2. \qed

The function \( \#U \mapsto (\#U)^3/((\#U)^3 - 2(\#U)^2 + 1) \) is a decreasing function with limit 1 for \( \#U \) approaching \( \infty \). Figure 2 gives some evaluations for some values of \( \#U \). We see that for \( \#U \geq 4 \) the expected number of iterations of Algorithm \textit{MinimalSolution} is less than two.
<table>
<thead>
<tr>
<th>$#U$</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\frac{#U^3}{((#U^3 - 2#U^2 + 1)}$</td>
<td>8.0</td>
<td>2.7</td>
<td>1.9</td>
<td>1.6</td>
<td>1.5</td>
<td>1.4</td>
<td>1.3</td>
</tr>
</tbody>
</table>

**Figure 2:** Expected number of iterations of Algorithm *MinimalSolution* (inconsistent case)

### 4.2. The consistent case

In this section we will consider the case when the system does have a rational solution. Then the algorithm has to find a rational solution with minimal denominator and a certificate certifying the minimality of the solutions denominator. Note that since the systems $Bx = c$ and $Ax = b$ are equivalent, they have the same solutions with minimal denominator and $d(B, c) = d(A, b)$.

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

In general, $\hat{y}$ in the algorithm will not be a solution of $Bx = c$ with minimal denominator. However, it follows from Lemma 2.1 that if for a prime $p \in R$ we have $\text{ord}_p(d(\hat{y})) = \text{ord}_p(d(B, c))$ for at least one $\hat{y}$, then the returned solution $y$ will satisfy $\text{ord}_p(d(y)) = \text{ord}_p(d(B, c))$. Similar, $\hat{z}$ will in general not certify all of $d(B, c)$. However, it follows from Lemma 2.4 that if for a prime $p \in R$ we have $\text{ord}_p(d(\hat{z}c)) = \text{ord}_p(d(B, c))$ for at least one $\hat{z}$, then the returned certificate $z$ will satisfy $\text{ord}_p(d(zc)) = \text{ord}_p(d(B, c))$.

The following fact is used in 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 1:** There exists a unimodular matrix $V \in R^{m \times m}$ such that $BV = H = \begin{bmatrix} H_1 & 0 \end{bmatrix}$, where $H_1$ is $s \times s$ and nonsingular. For example, we can take $H$ to be the column Hermite normal form of $B$ Newman (1972).

**Lemma 4.4:** Let $\alpha \in R^m$ such that $B\alpha = d(B, c)c$, that is, $\alpha/d(B, c)$ is a solution of $Bx = c$ with minimal denominator. Let $V$ be as in Fact 1 and $W$ the first $s$ rows of $V^{-1}$. Let $p \in R$ be prime. Let $P$ such that $p \not| \det(WP)$. Then $BP$ is nonsingular and $\text{ord}_p(d(\hat{y})) = \text{ord}_p(d(B, c))$. If moreover $l$ is such that $p \not| l\det(WP)(WP)^{-1}W\alpha$, then $\text{ord}_p(d(\hat{z}c)) = \text{ord}_p(d(B, c))$.

**Proof:** Since $B = HV^{-1}$ and $H = \begin{bmatrix} H_1 & 0 \end{bmatrix}$ we have

$$B = H_1W.$$  \hspace{1cm} (2)

It follows that $BP = H_1WP$ is nonsingular since $H_1$ is nonsingular and $WP$ is nonsingular modulo $p$. 


Substituting (2) into \(B\alpha/d(B, c) = c\) yields \(H_1^{-1}c = W\alpha/d(B, c)\). Then

\[
\hat{y} = P(BP)^{-1}c \\
= P(H_1WP)^{-1}c \\
= P(WP)^{-1}H_1^{-1}c \\
= \frac{1}{\det(WP)d(B, c)} \cdot P \det(WP)(WP)^{-1}W\alpha. \tag{3}
\]

From (3) we see that \(d(\hat{y})|(\det(WP)d(B, c))\) since \(P \det(WP)(WP)^{-1}W\alpha\) is over \(R\). It follows that \(\text{ord}_p(d(\hat{y})) \leq \text{ord}_p(d(B, c))\) since by assumption \(p\nmid \det(WP)\). On the other hand, we must have \(\text{ord}_p(d(B, c)) \leq \text{ord}_p(d(\hat{y}))\) since \(\hat{y}\) is a rational solution of \(Bx = c\). It follows that \(\text{ord}_p(d(\hat{y})) = \text{ord}_p(d(B, c))\).

Since \(\eta = l(BP)^{-1} = l(H_1WP)^{-1} = l(WP)^{-1}H_1^{-1}\) we have

\[
\hat{z}c = d(\eta B)\eta c \\
= d(\eta B)l(WP)^{-1}H_1^{-1}H_1W\alpha/d(B, c) \\
= \frac{1}{\det(WP)d(B, c)}d(\eta B)l \det(WP)(WP)^{-1}W\alpha. \tag{4}
\]

Since \(V\) is unimodular we have \(d(\eta B) = d(\eta BV) = d(\eta H) = d(\eta H_1)\). Since \(p\nmid \det(WP)\), \(p\nmid d(\eta H_1)\) would imply that \(p|d(\eta H_1(WP)) = d(\eta BP) = l(B) = 1\); a contradiction, so \(p\nmid d(\eta H_1) = d(\eta B)\). Since \(p\nmid l \det(WP)(WP)^{-1}W\alpha\) we see from (4) that \(\text{ord}_p(d(\hat{z}c)) \geq \text{ord}_p(d(B, c))\). Since always \(\text{ord}_p(d(\hat{z}c)) \leq \text{ord}_p(d(B, c))\) it follows that \(\text{ord}_p(d(\hat{z}c)) = \text{ord}_p(d(B, c))\).

\text{Definition:} We say that the pair \((P, l)\) is a \textit{good pair} with respect to the prime \(p\) if

1. \(BP\) is nonsingular;
2. \(\text{ord}_p(d(\hat{y})) = \text{ord}_p(d(B, c))\);
3. \(\text{ord}_p(d(\hat{z}c)) = \text{ord}_p(d(B, c))\).

So if we choose in Algorithm \textit{MinimalSolution} a good pair \((P, l)\) with respect to the prime \(p\), \(y\) and \(z\) will satisfy from that moment on \(\text{ord}_p(d(y)) = \text{ord}_p(d(zc))\).

\text{Lemma 4.5:} Let \(p \in R\) be prime, \(\phi: U \rightarrow R/pR\) the projection map and \(g\) the maximum number of elements in the preimage of any element from \(R/pR\) under \(\phi\). Then the probability that in a particular iteration of the loop in Algorithm \textit{MinimalSolution} a good pair \((P, l)\) with respect to \(p\) is chosen is at least

\[
\left(1 - \frac{g}{\#U}\right) \left(1 - \frac{g}{\#U} - \left(\frac{g}{\#U}\right)^2\right).
\]
Proof: Let $V, W$ and $\alpha$ be as in Lemma 4.4. If $p \mid d(B, c)$ and $p \mid \det(WP)$ we have for all $l \in U^{1 \times s}$ that $\operatorname{ord}_p(d(\hat{c}l)) = \operatorname{ord}_p(d(B, c)) = 0$. So in that case it follows from Lemma 4.4 that in order for $(P, l)$ to be a good pair with respect to $p$ it suffices that $p \mid \det(WP)$.

Since $V^{-1}$ is also over $R$ and unimodular, it is clear that $W$ modulo $p$ has rank $s$. Applying Corollary 3.2 with $F = R/pR, m_1 = 0$ and $W_2 = W$, we see that the probability that $p \mid \det(WP)$ is at least

$$\prod_{i=1}^{s} \left(1 - \left(\frac{g}{\#U}\right)^i\right).$$

By Proposition 4.1 this is at least $1 - g/\#U - (g/\#U)^2$. The lemma follows when $p \nmid d(B, c)$.

Now assume that $p \mid d(B, c)\text{ and } p \mid \det(WP)$. Suppose $p \mid W\alpha$. Since the columns of $W$ span all of $R^s$ we then have $(W\alpha)/p = W\beta$ for some $\beta \in R^m$ and thus $B\beta = H_1 W\beta = H_1 W\alpha/p = B\alpha/p = (d(B, c)/p)\alpha$, contradicting the minimality of $d(B, c)$. So $p \nmid W\alpha$ and thus $p \nmid \det(WP)(WP)^{-1}W\alpha$. Applying 3.2 with $F = R/pR, m_1 = 0, W_2 = (\det(WP)(WP)^{-1}W\alpha)^t$ and $P = l^t$, we see that the probability that $p \nmid \det(WP)(WP)^{-1}W\alpha$ is at least $1 - g/\#U$. The lemma follows from Lemma 4.4.

We want the numbers of elements in the preimage of all elements from $R/pR$ under $\phi: U \to R/pR$ to differ as little as possible.

Definition: Let $U \subseteq R$ finite and $p \in R$ prime. We say that $U$ is evenly distributed with respect to $p$, if

1. $\#(R/pR) < \infty$: for all $w \in R$

$$\left\lfloor \frac{\#U}{\#(R/pR)} \right\rfloor \leq \# \{u \in U \mid u \equiv w \pmod{p} \} \leq \left\lceil \frac{\#U}{\#(R/pR)} \right\rceil;$$

2. $\#(R/pR) = \infty$: for all $w \in R$

$$\# \{u \in U \mid u \equiv w \pmod{p} \} \leq 1.$$

Corollary 4.2: Let $p \in R$ be prime and $U$ evenly distributed with respect to $p$. Then the probability that $(P, l)$ is not a good pair with respect to $p$ is at most

$$\begin{cases} \frac{\#U}{\#(R/pR)} & \text{if } \#U = 2 \text{ or } (\#U \geq 25 \text{ and } \#(R/pR) = 2); \\ \frac{2}{\#U} & \text{if } \#U < \#(R/pR); \\ \frac{\#(R/pR)}{\#U} & \text{if } \#(R/pR) \mid \#U; \\ \frac{2}{\#U} + \frac{\#(R/pR)}{\#U} & \text{if } \#(R/pR) \nmid \#U. \end{cases}$$
Proof: Since \((1 - x)(1 - x - x^2) = 1 - 2x + x^3\) it follows from Lemma 4.5 that the wanted probability is at most \(2g/\#U - (g/\#U)^3 \leq 2g/\#U\). The lemma now follows by noting that

\[
g = \begin{cases} 
  1 & \text{if } \#U < \#(R/pR); \\
  \frac{\#U}{\#(R/pR)} & \text{if } \#(R/pR) \mid \#U; \\
  \frac{\#U}{\#(R/pR)} + 1 & \text{if } \#(R/pR) \nmid \#U.
\end{cases}
\]

One can give sharper bounds for the probability bounded in Corollary 4.2. However, the bounds in Corollary 4.2 are easy to use and suffice for our purposes, so we will not give a more detailed analysis of the probability.

**Lemma 4.6:** Let \(S\) be a finite set of primes of \(R\). Let \(U \subseteq R\) be evenly distributed with respect to all primes in \(S\). For \(q \in \mathbb{Z}_{\geq 2}\) and \(q = \infty\) let \(S_q = \{p \in S \mid \#(R/pR) = q\}\). Then the probability that after \(N\) iterations of the loop in Algorithm MinimalSolution there is still a prime \(p \in S\) such that no good pair \((P, l)\) with respect to \(p\) was chosen is at most

\[
\begin{cases} 
  \#S \left(\frac{q}{10}\right)^N & \text{if } \#U = 2; \\
  \#S_2 \left(\frac{q}{10}\right)^N + \sum_{q \mid \#U} \#S_q \left(\frac{2}{\#U}\right)^N + \sum_{q \mid \#U, q > 2} \#S_q \left(\frac{2}{q}\right)^N & \text{if } \#U \geq 25.
\end{cases}
\]

Proof: The wanted probability is at most the sum over all primes \(p \in S\) of the probability that no good pair with respect to \(p\) was chosen. The probability that \(N\) independent experiments, each with a probability of failure less than \(f\), all fail is less than \(f^N\). The Lemma now follows from Corollary 4.2.

We will now apply the foregoing on two particular instances of \(R\), that is, the integers and polynomial rings. In both cases we will consider \(U\) to be a minimal possible set, i.e. \(U = \{0, 1\}\) and \(U\) of bigger size. For an integer matrix \(A\) we denote by \(||A||\) the maximum magnitude of an entry in \(A\). For a polynomial matrix \(A\) we denote by \(||A||\) the maximum degree of an entry in \(A\). The following well known bounds follow from Cramer’s rule and Hadamard’s inequality Horn and Johnson (1985).

**Fact 2:** Let \(A \in \mathbb{Z}^{n \times m}\) nonsingular, \(b \in \mathbb{Z}^n\) and \(x \in \mathbb{Q}^n\) such that \(Ax = b\). Then \(d(x) \leq n^{n/2}||A||^n\) and \(||n(x)|| \leq n^{n/2}||A||^{n-1}||b||\). If \(A \in K[x]^{n \times m}\) nonsingular, \(b \in K[x]^n\) and \(x \in K[x]^n\) such that \(Ax = b\) then \(\deg(d(x)) \leq n||A||\) and \(||n(x)|| \leq (n-1)||A|| + ||b||\).

**Corollary 4.3:** Let \(A \in \mathbb{Z}^{n \times m}\) of rank \(r\) and \(b \in \mathbb{Z}^n\) such that the system \(Ax = b\) is consistent. Let \(U = \{0, 1\}\). Then the expected number of iterations the loop in Algorithm MinimalSolution has to perform is \(O(\log(r) + \log \log(||A||))\).
Proof: Let $S$ be the set of prime divisors of the denominator of $y_0$. By Lemma 4.6 the probability that after $N$ iterations there is still a prime $p \in S$ such that $\text{ord}_p(d(y)) \neq \text{ord}_p(d(zc))$ is at most $\#S(9/10)^N$. From Proposition 4.2 it then follows that the expected number of iterations in order that $\text{ord}_p(d(y)) = \text{ord}_p(d(zc))$ for all $p \in S$ is at most

$$\frac{N}{1 - \#S \left(\frac{9}{10}\right)^N}.$$  

(5)

Taking $N = \left\lceil \log_{10,9}(2 \#S) \right\rceil$ we see that (5) is at most $2N$. By Fact 2 $\#S \leq r(\log(r) + \log(||A||))$ and the lemma follows. \hfill \Box

**Corollary 4.4:** Let $A \in \mathbb{Z}^{n \times m}$ and $b \in \mathbb{Z}^n$ such that the system $Ax = b$ is consistent. Let $\delta = \min(n, m)$, $M = \max(12, \left\lceil \delta \log(\delta) + \log(||A||))/2 + 1 \right\rceil)$ and $U = \{-M, -M + 1, \ldots, M - 1, M\}$. Then the expected number of iterations the loop in Algorithm MinimalSolution has to perform is $O(1)$.

*Proof:* The proof is similar to the one of Corollary 4.3. Note that $\#U \geq \#S + 2$ and $\#U \geq 25$. Now, the probability that after $N$ iterations there is still a prime $p \in S$ such that $\text{ord}_p(d(y)) \neq \text{ord}_p(d(zc))$ is at most

$$\rho = \left(\frac{9}{10}\right)^N + \sum_{p \in S, p > 2} \left(\frac{2}{q} + \frac{2}{\#U}\right)^N$$

$$\leq \left(\frac{9}{10}\right)^N + \sum_{k=3}^{\#S+2} \left(\frac{2}{k} + \frac{2}{\#U}\right)^N$$

$$\leq \left(\frac{9}{10}\right)^N + \sum_{k=3}^{A} \left(\frac{2}{k} + \frac{2}{25}\right)^N + \sum_{k=A+1}^{\#S+2} \left(\frac{2}{k} + \frac{2}{\#S+2}\right)^N$$

$$\leq \left(\frac{9}{10}\right)^N + \sum_{k=3}^{A} \left(\frac{2}{k} + \frac{2}{25}\right)^N + \sum_{k=A+1}^{\#S+2} \left(\frac{4}{k}\right)^N$$

and then the expected number of iterations in order that $\text{ord}_p(d(y)) = \text{ord}_p(d(zc))$ for all $p \in S$ is at most $N/(1 - \rho)$. Taking $N = 10$ and $A = 10$ we see that this is less than 17. \hfill \Box

**Corollary 4.5:** Let $K$ be a field, $A \in K[x]^{n \times m}$ of rank $r$ and $b \in K[x]^n$ such that the system $Ax = b$ is consistent. Let $U = \{0, 1\}$. Then the expected number of iterations the loop in Algorithm MinimalSolution has to perform is $O(\log(r) + \log(||A||))$. 
Proof: The proof is similar to the proof of Corollary 4.3. Now \( \#S \leq r||A|| \). □

**Corollary 4.6:** Let \( K \) be a field, \( A \in K[x]^{n \times m} \) and \( b \in K[x]^n \) such that the system \( Ax = b \) is consistent. Let \( \delta = \min(n, m) \). If \( K \) is not finite, let \( U \subseteq K \) of size \( \max(25, 3\delta||A||) \); if \( K \) is finite let \( t \) such that \( t(#K)^t \geq 3\delta||A|| \) and \( U = \{ f \in K[x] : \deg(f) < t \} \). Then the expected number of iterations the loop in Algorithm MinimalSolution has to perform is \( O(1) \).

**Proof:** Suppose \( K \) is not finite. Then \( \#U \geq 3(#S) \) and the probability that after one iteration there is still a prime \( p \in S \) such that \( \text{ord}_p(d(y)) \neq \text{ord}_p(d(zc)) \) is at most \( #S(2/#U) \leq 2/3 \). Thus the expected number of iterations in order that \( \text{ord}_p(d(y)) = \text{ord}_p(d(zc)) \) for all \( p \in S \) is at most 3.

Now suppose that \( K \) is finite. There are at most \( \delta||A||/(t + 1) \) primes in \( S \) of degree \( > t \) and at most \( (#K)^k \) primes of degree \( k \). If \( #K > 2 \) the probability that after \( N \) iterations there is still a prime \( p \in S \) such that \( \text{ord}_p(d(y)) \neq \text{ord}_p(d(zc)) \) is at most

\[
\rho = \frac{\delta||A||}{t + 1} \left( \frac{2}{(#K)^t} \right)^N + \sum_{k=1}^{t} (#K)^k \left( \frac{2}{(#K)^k} \right)^N
\]

\[
\leq \left( \frac{2}{3} \right)^N + \sum_{k=1}^{t} 2 \left( \frac{2}{3^k} \right)^{N-1}
\]

\[
\leq \left( \frac{2}{3} \right)^N + \sum_{k=1}^{\infty} 2 \left( \frac{2}{3^k} \right)^{N-1},
\]

and the expected number of iterations in order that \( \text{ord}_p(d(y)) = \text{ord}_p(d(zc)) \) for all \( p \in S \) is at most \( N/(1 - \rho) \). Taking \( N = 8 \) this is at most 10.

If \( #K = 2 \) there are at most two primes \( p \) such that \( #(R/pR) = 2 \) and we get

\[
\rho = 2 \left( \frac{9}{10} \right)^N + \frac{\delta||A||}{t + 1} \left( \frac{2}{2^t} \right)^N + \sum_{k=2}^{t} (#K)^k \left( \frac{2}{(#K)^k} \right)^N
\]

\[
\leq 2 \left( \frac{9}{10} \right)^N + \left( \frac{2}{3} \right)^N + \sum_{k=2}^{t} 2 \left( \frac{2}{2^k} \right)^{N-1}
\]

\[
\leq 2 \left( \frac{9}{10} \right)^N + \left( \frac{2}{3} \right)^N + \sum_{k=1}^{\infty} 2 \left( \frac{2}{2^k} \right)^{N-1}.
\]

The expected number of iterations in order that \( \text{ord}_p(d(y)) = \text{ord}_p(d(zc)) \) for all \( p \in S \) is at most \( N/(1 - \rho) \). Taking \( N = 15 \) this is at most 26. □

The bounds in the proofs of Corollaries 4.3 to 4.6 are very coarse. In practice algorithm MinimalSolution will perform much less iterations.
\[
\begin{array}{c|c}
  d(y) & d(zc) \\
  \hline
  787165917582480 & 5983322572080 \\
  3935829587914240 & 19679147939571120 \\
  3935829587914240 & 19679147939571120 \\
  3935829587914240 & 19679147939571120 \\
  19679147939571120 & 19679147939571120 \\
  19679147939571120 & 9839573969785560 \\
  19679147939571120 & 19679147939571120 \\
  157433183516568960 & 16819784567360 \\
  19679147939571120 & 19679147939571120 \\
  19679147939571120 & 19679147939571120 \\
  19679147939571120 & 37844515268406 \\
  19679147939571120 & 9839573969785560 \\
  19679147939571120 & 19679147939571120 \\
\end{array}
\]

**Figure 3:** \(d(y)\) and \(d(zc)\) during several runs of algorithm \textit{MinimalSolution}

**Example 3:** \(A \in \mathbb{Z}^{15 \times 20}\) and \(b \in \mathbb{Z}^{15}\) with 6-digit entries such that \(Ax = b\) is consistent. We take \(U = \{0, 1\}\). The denominator of \(y_0\) is

\[
10400362865580705336166417648555205138303022080.
\]

Figure 3 gives the consecutive denominators of \(y\) and \(zc\) during five independent runs of Algorithm \textit{MinimalSolution} on input \(A\) and \(b\). We see that the denominator of \(y_0\) is much too big. This is because this denominator contains some “random” extraneous factors. The denominator of the first value for \(y\) is already very close to the minimal denominator. This is because most of the extraneous factors of the denominator of \(y_0\) are filtered out by the second rational solution, which has other extraneous factors.

Similar, the denominator of \(zc\) for the first non-zero value of \(z\) is very close to the minimal denominator. It only lacks some small factors.

5. **Complexity**

In this section we will compute the complexity of Algorithm \textit{MinimalSolution} when \(R\) is either the ring of integers or a polynomial ring. Since Algorithm \textit{MinimalSolution} is not explicit about the methods used to compute \(B\) and \(C\) and to solve the various nonsingular linear systems we will have to make a choice at this place. Also, in order to avoid expression swell, we will have to modify the way in which we combine the various rational solutions and certificates.

**Lemma 5.1:** Let \(y_0, y, \hat{y} \in F^m\) be rational solutions of \(Ax = b\). Let \(a \in R\) such that \(\gcd(d(y_0), d(y) + ad(\hat{y})) = \gcd(d(y_0), d(y), d(\hat{y}))\) and

\[
\hat{y} = \frac{d(y)y + ad(\hat{y})\hat{y}}{d(y) + ad(\hat{y})}.
\]
Then $\tilde{y}$ is a rational solution of $Ax = b$ and $\gcd((d(y_0), d(\tilde{y})))|\gcd(d(y_0), d(y), d(\tilde{y}))$.

Proof:

\[
A\tilde{y} = A(d(y)y + ad(\tilde{y})\tilde{y})/(d(y) + ad(\tilde{y})) \\
= (d(y)A y + ad(\tilde{y})A\tilde{y})/(d(y) + ad(\tilde{y})) \\
= (d(y)b + ad(\tilde{y})b)/(d(y) + ad(\tilde{y})) \\
= ((d(y) + ad(\tilde{y}))/(d(y) + ad(\tilde{y}))b \\
= b,
\]

so $\tilde{y}$ is a rational solution. Since $d(\tilde{y})|(d(y)+ad(\tilde{y}))$ we have $\gcd(d(y_0), d(\tilde{y}))|\gcd(d(y_0), d(y) + ad(\tilde{y})) = \gcd(d(y_0), d(y), d(\tilde{y}))$. \qed

Let $y_0$ and $\hat{y}_1, \ldots, \hat{y}_l$ be rational solutions. We can use Lemma 5.1 to compute rational solutions $y_2, \ldots, y_l$ such that $\gcd(d(y_0), d(y_i))|\gcd(d(y_0), d(\hat{y}_1), \ldots, d(\hat{y}_i))$. Next we can use Lemma 2.1 to compute a rational solution $y$ such that $d(y)|\gcd(d(y_0), d(y_i))|\gcd(d(y_0), d(\hat{y}_1), \ldots, d(\hat{y}_i))$.

Lemma 5.2: Let $z, \hat{z} \in F^{1 \times n}$ such that $zA, \hat{z}A \in R^{1 \times m}$. Write $zh = n/d$ and $\hat{z}b = \hat{n}/\hat{d}$, where $\gcd(n, d) = \gcd(\hat{n}, \hat{d}) = 1$. Let $g = \gcd(d, \hat{d})$ and $l = \text{lcm}(d, \hat{d})$.

Then $\gcd \left( \frac{n}{g}, \frac{\hat{n}}{g}, l \right) = 1$. Let $a \in R$ such that

\[
\gcd \left( \frac{\hat{n} a}{g}, \frac{\hat{n} a}{g}, l \right) = 1
\]

and $\tilde{z} = z + a\hat{z}$. Then $\tilde{z}A \in R^{1 \times m}$ and $d(\tilde{z}b) = l$.

Proof: That $\gcd \left( \frac{n}{g}, \frac{\hat{n} a}{g}, l \right) = 1$ is proven in the proof of Lemma 2.4. Moreover, $\tilde{z}A = (z + a\hat{z})A = zA + a(\hat{z}A) \in R^{1 \times m}$ and

\[
\tilde{z}b = (z + a\hat{z})b \\
= zb + a(\hat{z}b) \\
= \frac{n}{d} + \frac{\hat{n}}{\hat{d}} \\
= \frac{n\hat{d} + a\hat{n}d}{d\hat{d}} \\
\approx \frac{n\hat{d} + a\hat{n}d}{gl} \\
= \frac{n\hat{d} + a\hat{n}d}{l}.
\]

Since $\gcd \left( \frac{n}{g} + a\frac{n}{g}, l \right) = 1$, we have $d(\tilde{z}b) = l$. \qed
algorithm Split
input: \( v, u \in R \).
output: \( t \in R \) such that \( t|u \) and for all primes \( p \) we have:

\[
\begin{align*}
    p|t & \Rightarrow p|v \\
    p|u/t & \Rightarrow p|v 
\end{align*}
\]

\( x := v; \)
\( t := u; \)

while \( x \neq 1 \) do

\( x := \text{gcd}(x, t); \)
\( t := t/x \)

od;
return \( t \)

Figure 4: Algorithm Split

To compute \( a \in R \) as needed in Lemmas 5.1 and 5.2 we can use the algorithm described in Figure 4.

Lemma 5.3: For \( u, v, w \in R \):

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

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 \( \text{gcd}(v, w, u) = 1 \). Let \( p|u \) be prime. When \( p|v \), then \( p|\text{Split}(v, u), w \) and when \( p|v \), then \( p|\text{Split}(v, u) \). From this the lemma follows. \( \square \)

We will now give the modified algorithm MinimalSolution for \( R \) the ring of integers or a polynomial ring. We determine the matrices \( B \) and \( C \) probabilistically using a random prime \( q \). By repeating this for several \( q \) we avoid the possibility of always using \( B \) and \( C \) of wrong size. In the consistent case we compute a solution \( y \) with minimal denominator and certificate \( z \) for the subsystem \( Bx = c \) and then transform \( z \) into a certificate for the whole system. If we find out that \( B \) and \( A \) are not equivalent, i.e. when \( A_{gy} \neq b \) or \((AP)\xi \neq b \), this might be because \( q \) is not well-chosen. Therefore we only try once to find a certificate for the inconsistency. If we do not succeed, we choose another prime \( q \).

We solve the linear systems using Hensel-lifting. For this we need a prime \( \hat{q} \in R \) such that \( BP \) is nonsingular modulo \( \hat{q} \). We choose \( \hat{q} \) randomly from a precomputed set of primes. When \( BP \) happens to be singular modulo \( \hat{q} \) we return NIL instead of a solution.

In order to keep \( y \) and \( z \) small, we only combine them with new solutions or certificates when this will lead to some progress in the computation, i.e. \( d(y) \) gets smaller or \( d(zc) \) gets bigger.

For \( T \) we take a set of primes such that for at most half of the primes \( q \in T \) we have \( \text{rank}(A \pmod q) < \text{rank}(A) \), and such that for nonsingular \( BP, BP \)
algorithm SpecialMinimalSolution
input: $A \in \mathbb{R}^{n \times m}, b \in \mathbb{R}^n$
output: Either ("no solution", $q$) or $(y, z)$ as described before.
$U :=$ finite subset of $R$;
$T :=$ SetOfPrimes($A, U$);
do
Choose $q \in T$;
Determine submatrix $B \in \mathbb{R}^{s \times m}$ of $A$, submatrix $C \in \mathbb{R}^{s \times s}$ of $B$ and corresponding subvector $c \in \mathbb{R}^s$ of $b$ such that $s = \text{rank}(B \mod q) = \text{rank}(A \mod q)$ and
$C$ is nonsingular modulo $q$;
Solve $C\zeta = c$ using $q$;
$y_0 :=$ extension of $\zeta$ to $\mathbb{F}^m$ by inserting 0's;
if $Ay_0 \neq b$ then
(*): Choose $P \in \mathbb{U}^{m \times s}$, $t \in \mathbb{U}^{1 \times n}$ and $\hat{q} \in T$;
Solve $\nu(BP) = t(AP)$ using $\hat{q}$;
if $\nu \neq \text{NIL}$ then
\hspace{1em} $q :=$ extension of $\nu$ to $\mathbb{F}^{1 \times n}$ by inserting 0's;
\hspace{1em} if $qA = tA$ and $qB \neq tB$ then
\hspace{2em} return ("no solution", $q - t$)
fi
else
$y := y_0$;
$z := 0$;
do
Choose $P \in \mathbb{U}^{m \times s}$ and $\hat{q} \in T$;
Solve $(BP)\xi = c$ using $\hat{q}$;
if $\xi \neq \text{NIL}$ then
\hspace{1em} if $(AP)\xi \neq b$ then goto (*)
else
\hspace{2em} $\hat{y} := P\xi$;
\hspace{2em} if $\text{gcd}(d(y_0), d(y), d(\hat{y})) \neq \text{gcd}(d(y_0), d(y))$ then
\hspace{3em} $y :=$ combine $y$ with $\hat{y}$ using Lemma 5.1
fi
Choose $l \in \mathbb{U}^{1 \times s}$;
Solve $\eta(BP) = l$ using $\hat{q}$;
$\hat{z} := d(\eta B)\eta$;
if $\text{lcm}(d(zc), d(\hat{z}c)) \neq d(zc)$ then
\hspace{1em} $z :=$ combine $z$ with $\hat{z}$ using Lemma 5.2
fi
fi
until $\text{gcd}(d(y_0), d(y)) = d(zc)$;
$y :=$ combine $y$ with $y_0$ using Lemma 2.1;
$z :=$ extension of $y$ to $\mathbb{F}^{1 \times n}$ by inserting 0's;
return $(y, z)$
fi
od

Figure 5: Algorithm SpecialMinimalSolution
(mod \(\delta\)) is singular for at most half of the primes \(\delta \in T\). Figure 5 gives a detailed description of the modified algorithm.

When \(q\) is badly chosen, i.e. when \(\text{rank}(A \pmod q) < \text{rank}(A)\), the algorithm may do some useless work or may still do some useful work. When the system is inconsistent we may still find a certificate for this. When the system is consistent it may happen that we try to find a certificate for inconsistency or we may still find a rational solution with minimal denominator and a certificate for this. Important to note is that whatever the algorithm does, it will never give wrong results.

**Lemma 5.4:** The expected number of iterations Algorithm \textit{SpecialMinimalSolution} has to perform is at most four times the expected number of iterations Algorithm \textit{MinimalSolution} has to perform.

**Proof:** The probability that \(q\) is well-chosen, i.e. that \(\text{rank}(A \pmod q) = \text{rank}(A)\), is at least 1/2. When \(q\) is well-chosen and \(P\) is such that \(BP\) is nonsingular, the probability that \(\delta\) is well-chosen, i.e. that \(BP\) is nonsingular modulo \(\delta\), is at least 1/2.

When \(q\) and \(\delta\) are well-chosen, one iteration in Algorithm \textit{SpecialMinimalSolution} is similar to one iteration in Algorithm \textit{MinimalSolution}. The lemma follows. \(\square\)

### 5.1. \(R = \mathbb{Z}\)

Now we will compute the complexity of Algorithm \textit{SpecialMinimalSolution} when \(R = \mathbb{Z}\). Remember that \(r = \text{rank}(A)\) and \(\delta = \min(n, m)\). When the elements in \(U\) are bounded in magnitude by \(M\), the entries in \(BP\) are bounded in magnitude by \(m M ||A||\) and thus a maximal minor of \(BP\) is bounded in magnitude by \(N = (\delta m M ||A||)^\delta\). We can take for \(T\) a set of \(2\left\lceil \frac{\log_2(N)}{(l-1)} \right\rceil\) primes \(p\) such that \(2^{\delta-1} < p < 2^l\), where \(l = 6 + \log \log N\), and \(T\) can be computed in \(O(\log N \log \log N)\) bit operations (see Giesbrecht (1993)). In what follows we will either take \(U = \{0, 1\}\) or take \(U = \{-M, -M + 1, \ldots, M - 1, M\}\), where \(M = \max(12, \lceil \delta (\log(\delta) + \log(||A||))/2 + 1 \rceil)\). It follows that the primes in \(T\) have bitlength bounded by \(O(\log \delta + \log \log m + \log \log ||A||)\); we use this length bound implicitly in what follows.

By Fact 2 we get the following length bounds:

| \(n(y_0)\), \(d(y_0)\) | \(O(r \log(r||A||) + \log ||b||)\) |
| \(d(q), d(\xi), d(\eta), n(\eta), n(\hat{z})\) | \(O(r \log(m||A||))\) |
| \(n(q)\) | \(O(r \log(m||A||) + \log n)\) |
| \(n(\xi), n(\hat{g})\) | \(O(r \log(m||A||) + \log ||b||)\) |

Let \(V, H\) and \(H_1\) be as in Fact 1, where \(H\) is the Hermite normal form of \(B\). Since \(\hat{z} B \in R^{1 \times m}\) we also have \(\hat{z} \begin{bmatrix} H_1 & 0 \end{bmatrix} = \hat{z} B V \in R^{1 \times m}\) and thus \(d(\hat{z}) | \det(H_1)\). In
the same way we find \( d(z) | \det(H_1) \). Since \( \gcd(d(y_0), d(y)) \) and \( d(zc) \) are always bounded by \( d(y_0) \) it follows that \( y \) and \( z \) will be modified at most \( O(r \log(r||A||)) \) times. We get the following length bounds:

\[
\begin{array}{c|c}
\text{bound} & \text{length} \\
\hline
d(\hat{z}), d(z) & O(r \log(r||A||)) \\
n(y) & O(r \log(m||A||) + \log ||b||)) \\
d(y), n(z) & O(r \log(m||A||)) \\
\end{array}
\]

We get the following lemma.

**Lemma 5.5:** The numerator and denominator of the certificate \( q \) or solution \( y \) and certificate \( z \) computed by Algorithm SpecialMinimalSolution all have length \( O(r \log(m||A||) + \log(m||b||)) \).

### 5.1.1. Standard arithmetic

First we will compute the complexity when standard arithmetic is used for handling rational numbers and no fast matrix multiplication is used.

**Proposition 5.1:** The costs of all single steps in Algorithm SpecialMinimalSolution are bounded by \( O(nmr(\log(nm||A||))^2 + (m + n)(\log||b||)^2) \) bit operations.

**Proof:** We use the fact that \( y_0, q \) and \( \xi \) have at most \( r \) nonzero entries. Moreover we use the fact that

\[
O(A(A + B)) = \begin{cases} 
O(A^2) & \text{if } B \leq A; \\
O(B^2) & \text{if } A \leq B.
\end{cases}
\]

For most steps (e.g. computing denominators, matrix/vector arithmetic etc.) the lemma follows from standard complexity considerations. Computing \( T \) can be accomplished in the allotted time (see before). Computing \( B \) and \( C \) can be accomplished in the allotted time by reducing all entries of \( A \) modulo \( p \) and performing Gaussian elimination modulo \( p \). By Mulders and Storjohann (1999, Theorem 23) Algorithm Split runs in the allotted time. Finally, from Mulders and Storjohann (1999, Theorem 20) it follows that solving \( C\zeta = c, \nu(BP) = t(AP), (BP)\xi = c \) and \( \eta(BP) = l \) can be accomplished in the allotted time by using Hensel-lifting.

**Lemma 5.6:** Taking \( U = \{0, 1\} \), the expected cost of Algorithm SpecialMinimalSolution is bounded by

\[
O\left((nmr(\log(nm||A||))^2 + (m + n)(\log||b||)^2)(\log r + \log \log||A||)\right)
\]

bit operations.

**Proof:** This follows immediately from Proposition 5.1, Lemma 5.4 and Corollaries 4.3 and 4.4.
**Algorithm** \( \text{FastRationalSolver} \)

**Input:** \( A \in R^{n \times n}, b \in R^n, q \in R \).

**Output:** Either NIL or \( x \in K^n \) such that \( Ax = b \).

**Comment:** Initialize

\[ N := \text{NumeratorBound}(A, b); \]
\[ D := \text{DenominatorBound}(A); \]
\[ L := \text{LiftingBound}(N, D); \]

if \( A \) is nonsingular modulo \( q \) then

\[ B := \text{mod}(A^{-1}, q) \]

else

return NIL

fi;

**Comment:** Lift

\[ z := 0; \]
\[ c := Bb; \]
\[ C := (BA - I)/q; \]
\[ M := 1; \]

while \( ||M|| \leq L \) do

\[ \bar{c} := \text{mod}(c, q); \]
\[ z := z + M\bar{c}; \]
\[ c := (c - \bar{c})/q - C\bar{c}; \]
\[ M := Mq \]

od;

**Comment:** Reconstruct

\( x := \text{RationalReconstruction}(z, M, N, D); \)

return \( x \)

---

5.1.2. Fast arithmetic

Now we will compute the complexity of Algorithm \( \text{SpecialMinimalSolution} \) when we use asymptotically fast arithmetic and matrix multiplication. For simplicity we will summarize the bounds \( ||A|| \) and \( ||b|| \) to one bound \( \beta \), i.e. \( ||A||, ||b|| \leq \beta \). Moreover we will only compute the approximate complexity by using the soft-Oh (\( O^* \)) notation. We denote the cost of multiplying two \( n \times n \) matrices by \( n^\theta \) arithmetic operations. The current record is \( \theta = 2.376 \) Coppersmith and Winograd (1990). We will assume that \( 2 \leq \theta \leq 3 \). The cost for all elementary operations on two \( n \)-bit numbers, i.e. sum, product, division and gcd, is bounded by \( O^*(n) \).

Before we can analyze the asymptotic complexity of Algorithm \( \text{SpecialMinimalSolution} \) we first have to introduce fast arithmetic and matrix multiplication in the algorithm that solves linear systems via Hensel-lifting. Figure 6 describes the fast rational solver.

When \( R = \mathbb{Z} \) we take
• NumeratorBound \((A, b) \rightarrow |n^{n/2}| |A||r^{n-1}| |b|||\); 
• DenominatorBound \((A) \rightarrow |n^{n/2}| |A||n|\); 
• LiftingBound \((N, D) \rightarrow 2ND\).

In fact Algorithm FastRationalSolver is Algorithm RationalSolver from Mulders and Storjohann (1999), adjusted to solve the system \((BA)x = Bb\) so that it can use the fact that \(BA = I + Cq\), where \(C\) has small entries.

**Lemma 5.7:** When \(R = \mathbb{Z}, ||A||, ||b|| \leq \gamma, \tau \geq 1, l = n^{1+\theta}r^2\) and \((n\gamma)^l \leq q < \tau(n\gamma)^l\), then the cost of Algorithm RationalSolver can be bounded by \(O^-(n^l \log(\tau\gamma))\) bit operations, where \(t = \frac{3+3t_0 - \theta^2}{4-\theta}\).

**Proof:** First note that \(t \geq \theta\). Computing \(N, D\) and \(L\) can be accomplished in \(O^-(n^2 \log \gamma)\) bit operations. \(B\) can be computed by computing the Howell form of \(A\) (mod \(q\)) and the corresponding transformation matrix. The cost of this is bounded by \(O^-(n^l \log q)\) (Storjohann and Mulders (1998, Corollary 2)). Also computing \(Bb\) and \(C\) can be accomplished in \(O^-(n^l \log q)\). Note that \(O^-(n^l \log q) = O^-(n^l \log(\tau\gamma))\).

From \(BA = I + Cq\) it follows that \(||C|| < n\gamma + 1\). \(C\tilde{c}\) can be computed as follows:

1. Divide the entries in \(\tilde{c}\) in chunks of size \([\log(n\tau\gamma)]\) and consider \(\tilde{c}\) as a matrix \(\tilde{c}\) with \(\lambda = O(l)\) columns.
2. Consider \(C\) and \(\tilde{c}\) as a matrices \(\hat{C}\) and \(\hat{c}\) whose entries are \(l \times l\) matrices. Then \(\hat{C}\) is \([n/l] \times [n/l]\) and \(\hat{c}\) is \([n/l] \times 1\).
3. Multiply \(\hat{C}\) and \(\hat{c}\), where each multiplication of two \(l \times l\) matrices is performed by fast matrix multiplication.
4. Compute \(C\tilde{c}\) from \(\hat{C}\hat{c}\) by shifts and additions.

The cost of computing \(C\tilde{c}\) in this way can be bounded by

\[
O^-(n/l)^2 \theta \log(n\tau\gamma)) = O^-(n^{1+\frac{3+3t_0 - \theta^2}{4-\theta} \log(\tau\gamma)}). \tag{6}
\]

\(\log ||z||, \log ||c||, \log M = O(n \log(n\gamma))\) holds throughout the computation, thus the cost of all remaining computations in one pass of the loop can also be bounded by (6). Since the number of iterations of the loop is bounded by \(O(n \log(n\gamma) / \log q) = O(n/l)\), it follows that the cost of the loop is bounded by \(O^-(n^{1+\frac{3+3t_0 - \theta^2}{4-\theta} \log(\tau\gamma)}) = O^-((n^l \log(\tau\gamma)).\) Finally, the cost of the reconstruction is bounded by \(O^-(n \log q)\) (von zur Gathen and Gerhard (1999)). \(\square\)

**Corollary 5.1:** Let \(A \in \mathbb{Z}^{n \times n}\) nonsingular and \(b \in \mathbb{Z}^n\). Suppose \(||A||, ||b|| \leq \gamma\). Let \(p \in \mathbb{Z}\) such that \(\gcd(p, \det(A)) = 1\). Then \(Ax = b\) can be solved in \(O^-(n^l \log(p\gamma))\) bit operations, where \(t = \frac{3+3t_0 - \theta^2}{4-\theta}\).
Proof: Apply Lemma 5.7 with \( q = p^s \). \( p^s \) can be computed in the allotted time by repeated squaring, multiplying and comparison, in effect finding the binary representation for \( s \). \( \square \)

Taking \( \theta = 2.376 \) and for \( p \) a word size prime not dividing \( \det(A) \), we can solve \( Ax = b \) in \( O(n^2.761 \log \gamma) \) bit operations.

**Lemma 5.8:** The expected cost of Algorithm SpecialMinimalSolution can be bounded by \( O(nmr^{t-2} \log \beta) \) bit operations, where \( t = \frac{3 + 30 - \theta^2}{4 - \theta} \).

*Proof:* We take \( U = \{-M, -M+1, \ldots, M-1, M\} \), where \( M = \max(12, \lfloor \delta \log(\delta) + \log(\|A\|)/2 + 1 \rfloor) \). Computing \( T \) can be accomplished in the allotted time. \( B \) can be determined by computing the column Howell form of \( A \) (mod \( q \)). \( C \) can be determined by computing the row Howell form of \( B \) (mod \( q \)). Both \( B \) and \( C \) can thus be determined in \( O(nmr^{\theta-2} \log \beta) \) bit operations Storjohann and Mulders (1998). Algorithm Split has to be replaced by the asymptotically fast implementation from Storjohann and Mulders (1998), which has running time bounded by \( O(t \log(m\beta)) \). Computing \( BP \) and \( AP \) can be accomplished in time \( O(nmr^{t-2} \log \beta) \).

The systems \( C\zeta = c, \nu(BP) = t(AP), (BP)\xi = c \) and \( \eta(BP) = \ell \) all involve entries with magnitude bounded by \( \gamma = nm\beta M^2 \). They are solved by Algorithm FastRationalSolver with \( q = \hat{q}^s \) such that \( (s\gamma)^u \leq q < \hat{q}(s\gamma)^u \), where \( u = s^{\frac{3-\theta}{3-\theta}} \). From Lemma 5.7 it follows that solving one such system costs \( O((s^4 \log(\hat{q} \gamma)) = O(nmr^{t-2} \log \beta) \) bit operations. As described in the proof of Corollary 5.1, computing \( q \) can be accomplished in the same time.

Recall that all numerators and denominators involved in \( y_0, q, \xi, \hat{y}, \eta, \hat{z}, \ell \) and \( y \) have bitlength bounded by \( O(t \log(nm\beta)) \). \( Ay_0 \) and \( (AP)\xi \) can be computed (after clearing the denominator) as in the proof of Lemma 5.7 by dividing \( y_0, q \) and \( \xi \) into chunks of size \( \lfloor \log(nm\beta) \rfloor \) and considering them as matrices with \( O(r) \) columns. In a similar way \( qA \) can be computed. The cost of this is \( O(nmr^{t-2} \log \beta) \).

From the bounds on numerators and denominators it follows easily that all other single steps in the algorithm can be accomplished in \( O(nmr^{t-2} \log \beta) \) bit operations. The lemma now follows from Lemma 5.4 and Corollary 4.4. \( \square \)

Taking \( \theta = 2.376 \) in Lemma 5.8, the expected cost of Algorithm SpecialSolution can be bounded by \( O(nmr^{0.761} \log \beta) \) bit operations.

### 5.2. \( R = K[x] \)

In this section we will compute the complexity of Algorithm SpecialMinimalSolution when \( R = K[x] \), where \( K \) is a field. Remember that \( r = \text{rank}(A) \) and \( \delta = \min(n,m) \). When the elements in \( U \) have degree bounded by \( t \), the entries in \( BP \) have degree bounded by \( \|A\| + t \) and thus the degree of a minor of \( BP \) is bounded by \( N = \delta(\|A\| + t) \) and thus we can take for \( T \) a set of \( 2N \) primes. Let
$q = \# K$. If $2N \leq q$, take for $T$ a set of $2N$ polynomials of the form $X - a$, with $a \in K$. If $2N > q$, let $l \in \mathbb{Z}_{2^2}$ such that $q^{l-1}/(2(l-1)) < 2N$ and $q^l/(2l) \geq 2N$. Then $l = O(\log_q N)$. From Lidl and Niederreiter (1983, Exercise 3.27) it follows that there are at least $2N$ monic irreducible polynomials of degree $l$ over $K$. Take for $T$ a set of $2N$ of these. Since a monic irreducible polynomial of degree $l$ divides $x^{q^l} - 1$, we can find such a set $T$ by factoring $x^{q^l} - 1$ and by von zur Gathen and Gerhard (1999, Exercise 14.47) this can be accomplished in $O^-(q^l) = O^-(N)$ field operations.

In what follows we will either take $U = \{0, 1\}$ or $U$ as in Corollary 4.6. Then always $t = O(\log_q (\delta ||A||))$. It follows that the degree of a prime in $T$ is bounded by $O(\log_q (\delta ||A||))$. In a similar way as in the integer case we get the following degree bounds:

\[
\begin{align*}
n(y_0) & \quad O(r ||A|| + ||b||) \\
d(y_0), d(z) & \quad O(r ||A||) \\
d(q), n(q), d(\xi), d(\eta), n(\xi), n(\eta), d(\tilde{y}), n(\tilde{y}) & \quad O(r (||A|| + \log_q \delta) + ||b||) \\
n(\tilde{z}), n(\tilde{z}), n(y) & \quad O(r (||A|| + \log_q \delta) + ||b||)
\end{align*}
\]

**Lemma 5.9:** The numerator and denominator of the certificate $q$ or solution $y$ and certificate $z$ computed by Algorithm SpecialMinimalSolution all have degree $O(r (||A|| + \log_q \delta) + ||b||)$.

5.2.1. Standard arithmetic

First we will give the complexity when standard polynomial arithmetic and no fast matrix multiplication is used.

**Proposition 5.2:** The cost of every single step in Algorithm SpecialMinimalSolution is bounded by $O(n m r ||A||^2 + (m + n) ||b||^2)$ field operations.

*Proof:* The proof is very similar to the proof of Proposition 5.1. Now we use Mulders and Storjohann (1999, Theorem 27) and Mulders and Storjohann (1999, Theorem 22). \qed

**Lemma 5.10:** Taking $U = \{0, 1\}$, the expected cost of Algorithm SpecialMinimalSolution is bounded by

\[
O(n m r ||A||^2 + (m + n) ||b||^2) (\log r + \log ||A||)
\]

field operations. Taking $U$ as in Corollary 4.6, the expected cost of Algorithm SpecialMinimalSolution is bounded by

\[
O(n m r ||A||^2 + (m + n) ||b||^2)
\]

field operations.

*Proof:* This follows immediately from Proposition 5.2, Lemma 5.4 and Corollaries 4.5 and 4.6. \qed
5.2.2. Fast arithmetic

Now we will compute the complexity of Algorithm \textit{SpecialMinimalSolution} when we use asymptotically fast polynomial arithmetic and matrix multiplication. This is very similar to the analysis in Section 5.1.2. Again we will assume that \( \|A\|, \|b\| \leq \beta \) and we will use soft-Oh notation. The cost for all elementary operations on two degree \( n \) polynomials can be bounded by \( O^n(n) \) field operations.

Now we take in Algorithm \textit{FastRationalSolver}

- \textbf{NumeratorBound}(\( A, b \)) \( \rightarrow (n - 1)\|A\| + \|b\|; \)
- \textbf{DenominatorBound}(\( A, b \)) \( \rightarrow n\|A\|; \)
- \textbf{LiftingBound}(\( N, D \)) \( \rightarrow N + D. \)

**Lemma 5.11:** When \( R = K[x] \), where \( K \) is a field, \( \|A\|, \|b\| \leq \gamma, \tau \geq 1, \)
\( l = \frac{3\theta}{4-\theta} \) and \( l\gamma \leq \|q\| < l\gamma + \tau, \) then the cost of Algorithm \textit{RationalSolver} can be bounded by \( O^n(n^t(\gamma + \tau)) \) field operations, where \( t = \frac{3 + 3\theta - \theta^2}{4-\theta}. \)

**Proof:** The proof is very similar to the proof of Lemma 5.7. Now we use the polynomial version of the Howell form algorithm from Storjohann and Mulders (1998), which has complexity \( O^n(n^t|q|) \). In this case we have \( \|C\| < \gamma \) and we compute \( C \tau \) similar as in the integer case, now dividing the entries of \( \bar{c} \) into chunks of size \( \gamma + \tau. \) The cost of this is \( O^n(n^{2+3\theta-\theta^2}(\gamma + \tau)). \) The number of iterations of the loop is bounded by \( O(n/\|q\|) = O(n/l). \) The cost of the reconstruction is bounded by \( O^n(n\gamma) \) (von zur Gathen and Gerhard (1999)). \( \Box \)

**Corollary 5.2:** Let \( A \in K[x]^{n \times n} \) nonsingular and \( b \in K[x], \) where \( K \) is a field. Suppose \( \|A\|, \|b\| \leq \gamma. \) Let \( p \in K[x] \) such that \( \gcd(p, \det(A)) = 1. \) Then \( Ax = b \) can be solved in \( O^n(n^t(\gamma + \|p\|)) \) field operations, where \( t = \frac{3 + 3\theta - \theta^2}{4-\theta}. \)

**Proof:** Similar to the proof of Corollary 5.1. \( \Box \)

Taking \( \theta = 2.376 \) and \( p \) an irreducible polynomial of small degree, we can solve \( Ax = b \) in \( O^n(n^{2.761\gamma}) \) field operations.

**Lemma 5.12:** The expected cost of Algorithm \textit{SpecialMinimalSolution} can be bounded by \( O^n(nmn^t-2\beta) \) field operations, where \( t = \frac{3 + 3\theta - \theta^2}{4-\theta}. \)

**Proof:** Take \( U \) as in Corollary 4.6. The proof is very similar to the proof in the integer case, now using the polynomial versions of the various results used in the proof of Lemma 5.8. \( \Box \)

Taking \( \theta = 2.376 \) in Lemma 5.12, the expected cost of Algorithm \textit{SpecialMinimalSolution} can be bounded by \( O^n(nmn^{0.761\beta}) \) field operations.
6. Minimal vectors

In this section we will extend the notion of minimal denominator to minimal factor. The algorithms discussed in this paper can then be used to compute such a minimal factor.

The minimal denominator $d(A, b)$ is the smallest integer $d$ such that $Ax = db$ has a diophantine solution. One could say that $d$ is the missing factor in order for $b$ to belong to the integer lattice spanned by the columns of $A$. However, it may also happen that $b$ has an abundant factor, i.e. even $Ax = b/d$ has a diophantine solution for some $d \in \mathbb{Z}_{>1}$.

**Example 4:** Let

$$A = \begin{bmatrix} 2 & 5 & 1 \\ 2 & 6 & 2 \end{bmatrix} \in \mathbb{Z}^{2 \times 3}.$$  

For $b = \begin{bmatrix} 20 \\ 30 \end{bmatrix}^t$ the system $Ax = b$ has a diophantine solution, e.g. $x = \begin{bmatrix} 3 \\ 1 \\ 9 \end{bmatrix}^t$. However, also the system $Ax = b/5$ has a diophantine solution, e.g. $x = \begin{bmatrix} -1 \\ 1 \\ 1 \end{bmatrix}^t$. Although $b$ is also divisible by 2, $Ax = b/2$ has no diophantine solution.

Let $R$ and $F$ be as in Section 2. The set of all $f \in F$ for which $Ax = fb$ admits a diophantine solution is a fractional ideal of $R$ in $F$, i.e. an $R$-module $I \subseteq F$ such that $cI \subseteq R$ for some $c \in R \setminus \{0\}$ Lang (1986). As in Section 2 we get a unique generator $f(A, b)$ for this fractional ideal, i.e. the set equals $f(A, b)R$. We call $f(A, b)$ the minimal factor of the system $ax = b$.

**Lemma 6.1:** Let $g$ be the greatest common divisor of all entries in $b$. Then

$$f(A, b) = \frac{d(A, b/g)}{g}.$$  

**Proof:** It is clear that $Ax = (d(A, b/g)/g)b$ has a diophantine solution. Suppose $f \in F$ and $y \in R^n$ such that $Ay = fb$. Let $f/(d(A, b/g)/g) = n/d$, with $n, d \in R$. Then

$$d(A, b/g) \frac{n b}{d g} = Ay \in R^n$$

and since $d$ and $b/g$ have no common factors it follows that $d(A, b/g)n/d \in R$. Moreover, it follows that $d(A, b/g)|(d(A, b/g)n/d)$ and thus $d|n$. So $f \in (d(A, b/g)/g)R$ and the lemma follows. □

Lemma 6.1 indicates how we can compute the minimal factor of a system. First we compute the greatest common divisor $g$ of the entries of $b$. Then we compute the minimal denominator $d(A, b/g)$. For this we can use one of the algorithms we have seen before. Finally we get the minimal factor as $d(A, b/g)/g$.

**Example 5:** With $A$ and $b$ as in example 4 we have $g = 10$ and $d(A, b/10) = 2$, so $f(A, b) = 1/5$. With the same $A$ but $b = \begin{bmatrix} 10 \\ 15 \end{bmatrix}^t$ we get $f(A, b) = 2/5$. 
The vector $f(A, b)b$ can be viewed as the minimal vector in the direction of $b$ that is contained in the $R$-lattice $\mathcal{L}(A)$ spanned by the columns of $A$. In other words, every vector in the direction of $b$ that is in the lattice $\mathcal{L}(A)$ is an $R$-multiple of $f(A, b)b$.

7. Conclusions

We have improved the Monte Carlo probabilistic algorithm from Mulders and Storjohann (1999) that computes for a linear system of equations a rational solution with minimal denominator. The improved algorithm not only computes a solution with minimal denominator, but also produces a certificate, certifying the minimality of the solutions denominator. In this way the algorithm has become a Las Vegas probabilistic algorithm.

Moreover, we have improved the running time of the algorithm by allowing more freedom in the random choices made during the course of the algorithm.

Finally, we have introduced fast arithmetic and matrix multiplication, thus leading to new algorithms, solving rational and diophantine linear systems, that run faster than any previous known algorithms.

The analysis in this paper always assumes the worst case behaviour of the algorithms, i.e. worst case bounds for probabilities are used. In practice the algorithms will perform much better, i.e. will perform less iterations than predicted.

Probably, the fast matrix multiplication technique introduced in Sections 5.1.2 and 5.2.2 are not yet of any practical value. Using fast arithmetic as described in these sections can however be beneficial, when one uses practical fast multiplication algorithms. Further research on real implementations would be interesting. This should reveal what steps in the algorithms are worthwhile using fast arithmetic.

References


