Interactive Certificates for Polynomial Matrices with Sub-Linear Communication

David Lucas, Vincent Neiger, Clement Pernet, Daniel Roche, Johan Rosenkilde

Interactive Certificates for Polynomial Matrices with Sub-Linear Communication

David Lucas[1], Vincent Neiger[2], Clément Pernet[1],
Daniel S. Roche[1,3], and Johan Rosenkilde[4]

[1]Laboratoire Jean Kuntzmann, Université Grenoble Alpes, France
[2]Univ. Limoges, CNRS, XLIM, UMR 7252, F-87000 Limoges, France
[3]United States Naval Academy, Annapolis, Maryland, U.S.A.
[4]Technical University of Denmark, Kgs. Lyngby, Denmark

July 2, 2018

Abstract

We develop and analyze new protocols to verify the correctness of various computations on matrices over $\mathsf{F}[x]$, where F is a field. The properties we verify concern an $\mathsf{F}[x]$-module and therefore cannot simply rely on previously-developed linear algebra certificates which work only for vector spaces. Our protocols are *interactive certificates*, often randomized, and feature a constant number of rounds of communication between the Prover and Verifier. We seek to minimize the communication cost so that the amount of data sent during the protocol is significantly smaller than the size of the result being verified, which can be useful when combining protocols or in some multi-party settings. The main tools we use are reductions to existing linear algebra certificates and a new protocol to verify that a given vector is in the $\mathsf{F}[x]$-linear span of a given matrix.

1 Introduction

Increasingly, users or institutions with large computational needs are relying on untrusted sources of computational results, which could be remote ("cloud") servers, unreliable hardware, or even just Monte Carlo randomized algorithms. The rising area of *verifiable computing* seeks to maintain the benefits in cost or speed of using such untrusted sources, without sacrificing accuracy. Generally speaking, the goal is to develop *certificates* for the correctness of some result, which can be verified much more efficiently than re-computing the result itself.

1.1 Interactive certificates

In this paper, we propose new *interactive certificates* for computations performed on univariate polynomial matrices; we refer to (Dumas and Kaltofen,

2014; Kaltofen, Nehring, and Saunders, 2011; Kaltofen, Li, Yang, and Zhi, 2012) for definitions related to such certificates. Generically, we consider protocols where a Prover performs computations and provides additional data structures or exchanges to a Verifier, who will use these to check the validity of a result, at a cheaper cost than by recomputing it.

The general flow of an interactive certificate is as follows.

1. The Prover first publishes a Commitment, which is the result of some computation.
2. The Verifier then answers with a Challenge, usually consisting of some uniformly sampled random values.
3. The Prover replies with a Response, used by the Verifier to ensure the validity of the commitment
4. In some cases, several additional rounds of Challenge/Response might be necessary for the Verifier to accept an answer.

These certificates can be simulated non-interactively in a single round following the Fiat-Shamir heuristic derandomization (Fiat and Shamir, 1987): random values produced by the Verifier are replaced by cryptographic hashes of the input and previous messages, and the Prover publishes once both the Commitment and Response to the derandomized Challenge.

There are several metrics to assess the efficiency of an interactive certificate, namely

Communication: the volume of data exchanged throughout the protocol;

Verifier cost: the worst-case number of arithmetic operations performed by the Verifier in the protocol, no matter what data was sent by the Prover;

Prover cost: the number of arithmetic operations performed by an *honest Prover* that is trying to prove a statement which is actually true without fooling the Verifier.

Note that some data, namely the input and output to the original problem, are considered as *public data* and do not count towards the communication cost. This is to remove those parts which are somehow inherent in the problem itself, as well as to separate the functions of computing and verifying a result, which can be quite useful when verification protocols are combined, as we will see.

Such protocols are said complete if the probability that a true statement is rejected by a Verifier can be made arbitrarily small; they are said perfectly complete if true statements are never rejected. For simplicity's sake, as all the protocols in this paper are perfectly complete, we will sometimes just describe them as complete. Similarly, a protocol is sound if the probability that a false statement is accepted by the Verifier can be made arbitrarily small. Note that all our protocols are probabilistically sound, which means the Verifier may be tricked into accepting a wrong answer. This is not an issue, as in practice

this probability can be reduced by simply repeating the protocol with new randomness, or by computing over a larger field. As our protocols are perfectly complete, any single failure means that the Prover did something wrong; the Verifier is never to blame.

Several approaches to verified computation exist: generic approaches based on protocol check circuits (Goldwasser, Kalai, and Rothblum, 2008) or on homomorphic encryption (Costello, Fournet, Howell, Kohlweiss, Kreuter, Naehrig, Parno, and Zahur, 2015); approaches working for any protocol where the Prover uses specific operations, as (Kaltofen et al., 2011, Section 5) which certifies any protocol where matrix multiplications are performed. Another approach consists in designing problem-specific certificates, as for instance (Freivalds, 1979; Kaltofen et al., 2011; Dumas, Lucas, and Pernet, 2017) on dense linear algebra and (Dumas and Kaltofen, 2014; Dumas, Kaltofen, Thomé, and Villard, 2016) on sparse linear algebra.

1.2 Polynomial matrices

This paper concerns computations on matrices whose entries are univariate polynomials. While certification for matrices over fields and over integer rings have been studied over the past twenty years, there are only few results on polynomial matrices (Giorgi and Neiger, 2018), and to the best of our knowledge, there are no certificates on most classical results for polynomial matrices.

Formally, a *polynomial matrix* is a matrix $\boldsymbol{M} \in \mathsf{F}[x]^{m\times n}$ whose entries are univariate polynomials over a field F. There is an isomorphism with *matrix polynomials* (univariate polynomials with matrices as coefficients) which we will sometimes use implicitly, such as when considering the evaluation $\boldsymbol{M}(\alpha) \in \mathsf{F}^{m\times n}$ of $\boldsymbol{M}$ at a point $\alpha \in \mathsf{F}$.

Computations with polynomial matrices are of central importance in computer algebra and symbolic computation, and many efficient algorithms for polynomial matrix computations have been developed.

One general approach for computing with polynomial matrices is based on evaluation and interpolation. The basic idea is to first evaluate the polynomial matrix, say $\boldsymbol{M} \in \mathsf{F}[x]^{n\times n}$ at a set of points $\alpha_1, \alpha_2, \ldots \in \mathsf{F}$ in the ground field, then to separately perform the desired computation on each $\boldsymbol{M}(\alpha_i)$ over $\mathsf{F}^{n\times n}$, and finally reconstruct the entries of the result using fast polynomial interpolation. This kind of approach works well for computations such as (nonsingular) system solving (Dixon, 1982), matrix multiplication (Bostan and Schost, 2005, Section 5.4), or determinant computation. These computations essentially concern the *vector space* in the sense that $\boldsymbol{M}$ may as well be seen as a matrix over the fractions $\mathsf{F}(x)$ without impact on the results of the computations.

Other computational problems with polynomial matrices intrinsically concern $\mathsf{F}[x]$-*modules* and thus cannot merely rely on evaluation and interpolation. Classic and important such examples are that of computing normal forms such as the Popov form and the Hermite form (Popov, 1972; Villard, 1996; Neiger, Rosenkilde, and Solomatov, 2018) and that of computing modules of relations such as approximant bases (Beckermann and Labahn, 1994; Giorgi, Jeannerod,

and Villard, 2003; Neiger and Vu, 2017). The algorithms in this case must preserve the module structure attached to the matrix and thus deal with the actual polynomials in some way; in particular, an algorithm which works with evaluations of the matrix at points $\alpha \in \mathsf{F}$ is oblivious of this module structure.

1.3 Our contributions

In this paper, after giving some preliminary material in Section 2, we propose certificates for classical properties on polynomial matrices — singularity, rank, determinant and matrix product — with sub-linear communication space with respect to the input size (Section 3). Those certificates are based on evaluating considered matrices at random points, which allows us to reduce the communication space and to use existing certificates for matrices over fields. Then, in Section 4 we give the main result of this paper, which is certifying that a given polynomial row vector is in the row space of a given polynomial matrix, which can either have full rank or be rank-deficient. Section 5 shows how to use this result to certify that for two given polynomial matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, the row space of $\boldsymbol{A}$ is contained in the row space of $\boldsymbol{B}$; and then gives certificates for some classical normal forms of polynomial matrices. In Section 6, we present certificates related to saturations and kernels of polynomial matrices. Finally, Section 7 gives a conclusion and comments on a few perspectives.

A summary of our contributions is given in Table 1, based on the following notations: the input matrix has rank r and size $n \times n$ if it is square or $m \times n$ if it can be rectangular; if there are several input matrices, then r stands for the maximum of their ranks, m for the maximum of their row dimensions, and n for the maximum of their column dimensions. Where appropriate, r is the maximum of the actual ranks of the matrices and the claimed rank by the prover. We write d for the maximum degree of any input matrix or vector. Finally, $\#\mathsf{S}$ stands for the cardinality of the finite subset $\mathsf{S} \subseteq \mathsf{F}$ from which we choose random evaluation points. The last column of the table specifies a lower bound on $\#\mathsf{S}$ which is needed to ensure both perfect completeness of the protocol and soundness with probability at least $\frac{1}{2}$. (Iterating any protocol improves the soundness probability exponentially.)

The Prover and Verifier costs are in arithmetic operations over the base field F. We use $\widetilde{O}(\cdot)$ for asymptotic cost bounds with hidden logarithmic factors, and $\omega \leqslant 3$ is the exponent of matrix multiplication, so that the multiplication of two $n \times n$ matrices over F uses $O(n^\omega)$ operations in F; see Section 2 for more details and references.

2 Preliminaries

Fields and rings. We use F to indicate an arbitrary field, $\mathsf{F}[x]$ for the ring of polynomials in one variable x with coefficients in F, and $\mathsf{F}(x)$ for the field of rational fractions, i.e., the fraction field of $\mathsf{F}[x]$. The ring of $m \times n$ matrices, for example over $\mathsf{F}[x]$, is denoted by $\mathsf{F}[x]^{m\times n}$.

	Prover		Comm.	Verifier	#S
	Deter.	Cost		Cost	
Singularity	Yes	$O(n^2d + nr^{\omega-1})$	$O(n)$	$O(n^2d)$	$2nd$
NonSingularity	No	$O(n^2d + n^\omega)$	$O(n)$	$O(n^2d)$	$nd + 1$
RankLowerBound	No	$\tilde{O}(mnr^{\omega-2}d)$	$O(r)$	$O(r^2d)$	$rd + 1$
RankUpperBound	No	$\tilde{O}(mnr^{\omega-2} + mnd)$	$O(n)$	$O(mnd)$	$2rd + 2$
Rank	No	$\tilde{O}(mnr^{\omega-2}d)$	$O(n)$	$O(mnd)$	$2rd + 2$
Determinant	Yes	$\tilde{O}(n^2d + n^\omega)$	$O(n)$	$O(n^2d)$	$2nd + 2$
SystemSolve	N/A	N/A	0	$O(n^2d)$	$4d$
MatMul	N/A	N/A	0	$O(n^2d)$	$4d + 2$
FullRankRowSpaceMembership	Yes	$\tilde{O}(nm^{\omega-1}d)$	$O(md)$	$O(mnd)$	$6md + 2d + 2$
RowSpaceMembership	No	$\tilde{O}(mnr^{\omega-2}d)$	$\tilde{O}(md + n)$	$\tilde{O}(mnd)$	$8rd + 2d + 2$
RowSpaceSubset	No	$\tilde{O}(mnr^{\omega-2}d)$	$\tilde{O}(md + n)$	$\tilde{O}(mnd)$	$8rd + 2d + 4$
RowSpaceEquality	No	$\tilde{O}(mnr^{\omega-2}d)$	$\tilde{O}(md + n)$	$\tilde{O}(mnd)$	$8rd + 2d + 4$
RowBasis	No	$\tilde{O}(mnr^{\omega-2}d)$	$\tilde{O}(md + n)$	$\tilde{O}(mnd)$	$8rd + 2d + 6$
HermiteForm	No	$\tilde{O}(mnr^{\omega-2}d)$	$\tilde{O}(md + n)$	$\tilde{O}(mnd)$	$8rd + 2d + 4$
ShiftedPopovForm	No	$\tilde{O}(mnr^{\omega-2}d)$	$\tilde{O}(md + n)$	$\tilde{O}(mnd)$	$8rd + 2d + 4$
Saturated $(m \leqslant n)$	No	$\tilde{O}(nm^{\omega-1}d)$	$\tilde{O}(nd)$	$\tilde{O}(mnd)$	$8md + 4$
Saturated $(m > n)$	No	$\tilde{O}(mn^{\omega-1}d)$	$\tilde{O}(md)$	$\tilde{O}(mnd)$	$8nd + 4$
SaturationBasis	No	$\tilde{O}(mnr^{\omega-2}d)$	$\tilde{O}(nd)$	$\tilde{O}(mnd)$	$8nd + 2d + 4$
UnimodularCompletable	No	$\tilde{O}(nm^{\omega-1}d)$	$\tilde{O}(nd)$	$\tilde{O}(mnd)$	$8md + 4$
KernelBasis	No	$\tilde{O}((m+n)m^{\omega-1}d)$	$\tilde{O}(md)$	$\tilde{O}(m(m+n)d)$	$8md + 4$

Table 1: This paper's contributions

Asymptotic complexity bounds. We use the "soft-oh" notation $\widetilde{O}(\cdot)$ to represent big-oh hiding logarithmic factors. Specifically, for two cost functions f, g, we say that $f \in \widetilde{O}(g)$ if and only if $f \in O(g\log(g)^c)$ for some constant $c \geqslant 1$.

We write ω for the exponent of matrix multiplication over F, so that any two matrices $\boldsymbol{A}, \boldsymbol{B} \in \mathsf{F}^{n\times n}$ can be multiplied using $O(n^\omega)$ field operations; we have $2 \leqslant \omega \leqslant 3$ and one may take $\omega < 2.373$ (Coppersmith and Winograd, 1990; Le Gall, 2014).

Cantor and Kaltofen (1991) have showed that multiplying two univariate polynomials of degree $\leqslant d$ over any algebra uses $\widetilde{O}(d)$ additions, multiplications, and divisions in that algebra. In particular, multiplying two matrices in $\mathsf{F}[x]^{n\times n}$ of degree at most d uses $\widetilde{O}(n^\omega d)$ operations in F.

Schwartz-Zippel lemma. Many of our protocols rely on the fact that when picking an element uniformly at random from a sufficiently large finite subset of the field, this element is unlikely to be a root of some given polynomial. This was stated formally in (Schwartz, 1980; Zippel, 1979; DeMillo and Lipton, 1978) and is customarily referred to as the *Schwartz-Zippel lemma*.

Specifically, it states that for any nonzero k-variate polynomial $f(x_1, \ldots, x_k)$ with coefficients in a field F, and any finite subset $\mathsf{S} \subseteq \mathsf{F}$, if an evaluation point $(\alpha_1, \ldots, \alpha_k) \in \mathsf{F}^k$ has entries chosen at random uniformly and independently from S, then the probability that $f(\alpha_1, \ldots, \alpha_k) = 0$ is at most $d/\#\mathsf{S}$ where d is the *total degree* of f.

Rational fractions. For a rational fraction $f \in \mathsf{F}(x)$, define its *denominator* $\mathrm{denom}(f)$ to be the unique monic polynomial $g \in \mathsf{F}[x]$ of minimal degree such that $gf \in \mathsf{F}[x]$. Correspondingly, define its *numerator* $\mathrm{numer}(f) = f\cdot\mathrm{denom}(f)$. Note that $\mathrm{denom}(a) = 1$ if and only if $a \in \mathsf{F}[x]$. More generally, for a matrix of rational fractions $\boldsymbol{A} \in \mathsf{F}(x)^{m\times n}$, define $\mathrm{denom}(\boldsymbol{A})$ to be the unique monic polynomial $g \in \mathsf{F}[x]$ of minimal degree such that $g\boldsymbol{A} \in \mathsf{F}[x]^{m\times n}$, and again write this polynomial matrix $g\boldsymbol{A}$ as $\mathrm{numer}(\boldsymbol{A})$. Note that we have the identity $\mathrm{denom}(\boldsymbol{A}) = \mathrm{lcm}_{i,j}(\mathrm{denom}(\boldsymbol{A}_{i,j}))$.

Row space, kernel, and row basis. For a given matrix $\boldsymbol{A} \in \mathsf{F}[x]^{m\times n}$, two basic sets associated to it are its row space

$$\mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A}) = \{\boldsymbol{p}\boldsymbol{A},\ \boldsymbol{p} \in \mathsf{F}[x]^{1\times m}\},$$

and its left kernel

$$\{\boldsymbol{p} \in \mathsf{F}[x]^{1\times m} \mid \boldsymbol{p}\boldsymbol{A} = \boldsymbol{0}\}.$$

Accordingly, a *row basis* of $\boldsymbol{A}$ is a matrix in $\mathsf{F}[x]^{r\times n}$ whose rows form a basis of the former set, where r is the rank of $\boldsymbol{A}$, while a *left kernel basis* of $\boldsymbol{A}$ is a matrix in $\mathsf{F}[x]^{(m-r)\times n}$ whose rows form a basis of the latter set. We use similar notions and notations for column spaces and column bases, and for right kernels and right kernel bases. We will also often consider the $\mathsf{F}(x)$-row space of $\boldsymbol{A}$, denoted by $\mathrm{RowSp}_{\mathsf{F}(x)}(\boldsymbol{A})$, which is an $\mathsf{F}(x)$-vector space.

Matrices which preserve the row space under left-multiplication, that is, $\boldsymbol{U} \in \mathsf{F}[x]^{m\times m}$ such that the $\mathsf{F}[x]$-row space of $\boldsymbol{UA}$ is the same as that of $\boldsymbol{A}$, are said to be *unimodular*. They are characterized by the fact that their determinant is a nonzero constant; or equivalently that their inverse has polynomial entries.

Protocols. In protocols, S is always a finite subset of the base field F, which we use to sample field elements uniformly and independently at random. One may use $\mathsf{S} = \mathsf{F}$ if the field F is finite. We denote by

$$\alpha \xleftarrow{\$} \mathsf{S} \quad \text{and} \quad \boldsymbol{v} \xleftarrow{\$} \mathsf{S}^{n\times 1}$$

the actions of drawing a field element uniformly at random from S and of drawing a vector of n field elements uniformly and independently at random from S.

To ensure that they are perfectly complete, our protocols require lower bounds on the cardinality $\#\mathsf{S}$ of this subset; when this bound exceeds the cardinality of F then one may use a field extension, possibly causing an increase by a logarithmic factor in the Prover/Verifier/communication costs.

Besides, many of our analyzes of protocols use the notation

$$d_{\boldsymbol{A}} = \max(1, \deg(\boldsymbol{A})) \quad \text{and} \quad r_{\boldsymbol{A}} = \operatorname{rank}(\boldsymbol{A})$$

for any polynomial matrix $\boldsymbol{A}$ that appears in this protocol.

3 Vector space computations

In this section, we give some certificates to compute classical linear algebra properties on polynomial matrices. The certificates we present here all rely on the same general idea, which consists in picking a random point and evaluating the input polynomial matrix (or matrices) at that point. This allows us to achieve sub-linear communication space. Note that this technique has been used before by Kaltofen et al. (2011) to certify the same properties for integer matrices: in that setup, computations were performed modulo some prime number, while, in our context, this translates into evaluating polynomials at some element of the base field.

In several of our certificates, the Prover has to solve a linear system over the base field. For a linear system whose matrix is in $\mathsf{F}^{m\times n}$ and has rank r, this can be done in $O(mnr^{\omega-2})$ operations in F, see (Jeannerod, Pernet, and Storjohann, 2013, Algorithm 6).

The following lemma will be frequently used when analyzing protocols: it bounds the probability of picking a "bad" evaluation point.

Lemma 3.1. *Let $\boldsymbol{A} \in \mathsf{F}[x]^{m\times n}$ with rank at least r. For any finite subset $\mathsf{S} \subseteq \mathsf{F}$ and for a point $\alpha \in \mathsf{S}$ chosen uniformly at random, the probability that $\operatorname{rank}(\boldsymbol{A}(\alpha)) < r$ is at most $r\deg(\boldsymbol{A})/\#\mathsf{S}$.*

Proof. Any $r \times r$ minor of $\boldsymbol{A}$ has degree at most $r\deg(\boldsymbol{A})$, and at least one must be nonzero since $\operatorname{rank}(\boldsymbol{A}) \geqslant r$. On the other hand, $\operatorname{rank}(\boldsymbol{A}(\alpha)) < r$ if and only if α is a root of every such determinant. □

3.1 Certificates for the singularity of polynomial matrices

We start by certifying the singularity of a matrix. Here, the Verifier picks a random evaluation point and sends it to the Prover, who evaluates the input matrix at that point and sends back a nontrivial kernel vector, which the Prover will always be able to compute since a singular polynomial matrix is still singular when evaluated at any point. Then, all the Verifier needs to do is to check that the vector received is indeed a kernel vector. Note that the evaluation trick here is really what allows us to have a sub-linear — with respect to the input size — communication space, as the answer the Prover provides to the challenge is a vector over the base field, and not over the polynomials.

Protocol 1: Singularity

Public: $\boldsymbol{A} \in \mathsf{F}[x]^{n \times n}$
Certifies: $\boldsymbol{A}$ is singular

	Prover		Verifier
1.			$\alpha \xleftarrow{\$} \mathsf{S}$
		$\xleftarrow{\alpha}$	
2.	Find $\boldsymbol{v} \in \mathsf{F}^{1\times n} \setminus \{\boldsymbol{0}\}$ s.t. $\boldsymbol{v}\boldsymbol{A}(\alpha) = \boldsymbol{0}$		
		$\xrightarrow{\boldsymbol{v}}$	
3.			$\boldsymbol{v} \stackrel{?}{\neq} \boldsymbol{0}$ $\boldsymbol{v}\boldsymbol{A}(\alpha) \stackrel{?}{=} \boldsymbol{0}$

In the next theorem, and for the remainder of the section, for convenience we write $d = \max(1, \deg(\boldsymbol{A}))$.

Theorem 3.2. *Protocol 1 is a complete and probabilistically sound interactive protocol which requires $O(n)$ communication and Verifier cost $O(n^2 d)$. The probability that the Verifier incorrectly accepts is at most $nd/\#\mathsf{S}$. If $\boldsymbol{A}$ is singular, there is an algorithm for the Prover which costs $O(n^2 d + nr^{\omega-1})$.*

Proof. If $\boldsymbol{A}$ is singular, $\boldsymbol{A}(\alpha)$ must also be singular and there exists a nontrivial nullspace vector that the Verifier will accept.

If $\boldsymbol{A}$ is nonsingular, then the Prover will be able to cheat if the Verifier picked an α such that $\boldsymbol{A}(\alpha)$ is singular which happens only with probability $nd/\#\mathsf{S}$ according to Lemma 3.1.

Now, for the complexities: the Prover will have to evaluate $\boldsymbol{A}$ at α, which costs $O(n^2 d)$ and to find a nullspace vector over the base field, which costs $O(nr^{\omega-1})$, hence the Prover cost. The Verifier computes the evaluation and a vector-matrix product over F, for a total cost of $O(n^2 d)$ operations. Finally, a

vector over F^n and a scalar are communicated, which yields a communication cost of $O(n)$ □

We now present a certificate for nonsingularity. This relies on the same evaluation-based approach, with one variation: here, we let the Prover provide the evaluation point. Indeed, if the Verifier picked a random point, they could choose an "unlucky" point for which a nonsingular matrix evaluates to a singular one, and in that case, the protocol would be incomplete as the Prover will not be able to convince the Verifier of nonsingularity. Instead, we let the Prover pick a point as they have the computational power to find a suitable point (Step 1 in NonSingularity). Once this value is committed to the Verifier, in Steps 2 to 4 we use the certificate for nonsingularity over a field due to Dumas and Kaltofen (2014, Theorem 3).

Protocol 2: NonSingularity

Public: $\boldsymbol{A} \in \mathsf{F}[x]^{n\times n}$
Certifies: $\boldsymbol{A}$ is nonsingular

	Prover		**Verifier**
1.	Find $\alpha \in \mathsf{S}$ s.t. $\det(\boldsymbol{A}(\alpha)) \neq 0$		
		$\xrightarrow{\quad\alpha\quad}$	
2.			$\boldsymbol{b} \xleftarrow{\$} \mathsf{S}^{n\times 1}$
		$\xleftarrow{\quad\boldsymbol{b}\quad}$	
3.	Find $\boldsymbol{w} \in \mathsf{F}^{n\times 1}$ s.t. $\boldsymbol{A}(\alpha)\boldsymbol{w} = \boldsymbol{b}$		
		$\xrightarrow{\quad\boldsymbol{w}\quad}$	
4.			$\boldsymbol{A}(\alpha)\boldsymbol{w} \stackrel{?}{=} \boldsymbol{b}$

Theorem 3.3. *Protocol 2 is a probabilistically sound interactive protocol and is complete assuming that $\#\mathsf{S} \geqslant nd + 1$. It requires $O(n)$ communication and Verifier cost $O(n^2 d)$. The probability that the Verifier incorrectly accepts is at most $1/\#\mathsf{S}$. There is a deterministic algorithm for the Prover with cost $\widetilde{O}(n^\omega d)$.*

Proof. If $\boldsymbol{A}$ is nonsingular, then, as the field is large enough, there exists an α for which the rank of $\boldsymbol{A}(\alpha)$ does not drop, and as Steps 2 to 4 form a complete certificate, NonSingularity is complete.

If $\boldsymbol{A}$ is singular, it is not possible to find an α such that $\boldsymbol{A}(\alpha)$ is nonsingular. This means the Prover successfully cheats if they manage to convince the Verifier that $\boldsymbol{A}(\alpha)$ is nonsingular, which only happens with probability $1/\#\mathsf{S}$ (Dumas and Kaltofen, 2014, Theorem 3), hence the soundness of NonSingularity.

Now, for the complexities: the Prover needs to find a suitable α. The Prover first computes the $\det(\boldsymbol{A}) \in \mathsf{F}[x]$ using the deterministic algorithm of Labahn, Neiger, and Zhou (2017, Theorem 1.1) in $\widetilde{O}(n^\omega d)$ time. Then, using fast multipoint evaluation, the determinant is evaluated at $nd+1$ points from S in time $\widetilde{O}(nd)$ (von zur Gathen and Gerhard, 2003, Corollary 10.8); since $\deg(\det(\boldsymbol{A})) \leqslant nd$, at least one evaluation will be nonzero. Computing this determinant dominates the later cost for the Prover to evaluate $\boldsymbol{A}(\alpha)$ and solve a linear system over the base field, hence a total cost of $\widetilde{O}(n^\omega d)$.

The Verifier needs to evaluate $\boldsymbol{A}$ at α and to perform a matrix-vector multiplication over the base field, hence a cost of $O\big(n^2 d\big)$. Finally, total communications are two vectors of size n over the base field and a scalar, hence the cost of $O(n)$. □

3.2 Certificates for the rank of polynomial matrices

From the certificate for nonsingularity, we can immediately infer one for a lower bound ρ on the rank: the Prover commits a set of indices which locate a $\rho \times \rho$ submatrix which is nonsingular, and then the certificate for nonsingularity is run on this submatrix.

Protocol 3: RankLowerBound

Public: $\boldsymbol{A} \in \mathsf{F}[x]^{m\times n}$, $\rho \in \mathbb{N}$

Certifies: $\operatorname{rank}(\boldsymbol{A}) \stackrel{?}{\geqslant} \rho$

	Prover		**Verifier**
1.	Find I, J two sets of size ρ such that $\boldsymbol{A}_{I,J}$ is nonsingular		
		$\xrightarrow{I,J}$	
2.			$\#I \stackrel{?}{=} \rho, \#J \stackrel{?}{=} \rho,$ $I \stackrel{?}{\subseteq} \{1,\dots,m\}$ $J \stackrel{?}{\subseteq} \{1,\dots,n\}$
3.	Use NonSingularity$(\boldsymbol{A}_{I,J})$		

Theorem 3.4. *Let r be the actual rank of $\boldsymbol{A}$. Protocol 3 is a probabilistically sound interactive protocol and is complete assuming $\#\mathsf{S} \geqslant \rho d + 1$ in its subprotocol. It requires $O(\rho)$ communication and Verifier cost $O\big(\rho^2 d\big)$. If ρ is indeed a lower bound on the rank of $\boldsymbol{A}$, then there is a Las Vegas randomized algorithm*

for the Prover with expected cost $\widetilde{O}(mnr^{\omega-2}d)$. *Otherwise, the probability that the Verifier incorrectly accepts is at most* $1/\#\mathsf{S}$.

Proof. If ρ is indeed a lower bound on the rank of $\boldsymbol{A}$, there exist two sets $I \subseteq \{1,\dots,m\}$ and $J \subseteq \{1,\dots,n\}$ of size ρ such that $\boldsymbol{A}_{I,J}$ is nonsingular, and since NonSingularity is complete, so is this certificate. Note that the completeness of the subcertificate is ensured only if $\#\mathsf{S} \geqslant \rho d + 1$.

If ρ is not a lower bound on the rank of $\boldsymbol{A}$, meaning $\operatorname{rank}(\boldsymbol{A}) \geqslant \rho$, then the Prover will not be able to find suitable I and J and hence the sets provided by a cheating Prover yield a singular submatrix $\boldsymbol{A}_{I,J}$. Now, if the Prover provided sets which do not contain ρ elements or which contain elements outside the allowed dimension bounds, this will always be detected by the Verifier. If the Prover provided sets with enough elements, the Verifier incorrectly accepts with the same probability as in NonSingularity, which is $1/\#\mathsf{S}$.

Regarding the complexities, the Prover has to find a $\rho \times \rho$ nonsingular submatrix of an $m \times n$ degree d matrix. This can be achieved, by first computing the rank using a Las Vegas randomized algorithm (Storjohann and Villard, 2005) which runs in $\widetilde{O}(mnr^{\omega-2}d)$, with r the *actual* rank of $\boldsymbol{A}$ and then picking a random evaluation point, random sets I and J and checking that those sets are still made of linearly independent elements over the base field by using Jeannerod et al. (2013). Because $\rho \leqslant r$, running the subprotocol NonSingularity on a $\rho \times \rho$ matrix does not dominate the complexity, and the total Prover cost is $\widetilde{O}(mnr^{\omega-2}d)$. From Theorem 3.3, the Verifier cost is $O(\rho^2 d)$. Finally, here two sets of ρ integers are transmitted, which with the communications in NonSingularity adds up to a communication cost of $O(\rho)$. □

Now, we give a certificate for an upper bound on the rank. Note that Steps 2 and 3 come from the certificate for an upper bound on the rank for matrices over a field (see Dumas and Kaltofen, 2014, Theorem 4). In this protocol, we use the notation $|\cdot|_H$ to refer to the Hamming weight: $|\boldsymbol{\gamma}|_H \leqslant \rho$ means that the vector $\boldsymbol{\gamma}$ as at most ρ nonzero entries.

Theorem 3.5. *Let* r *be the actual rank of* $\boldsymbol{A}$. *Then, Protocol 4 is a complete and probabilistically sound interactive protocol which requires* $O(n)$ *communication and Verifier cost* $O(mnd)$. *If* ρ *is indeed an upper bound on the rank of* $\boldsymbol{A}$, *then there is a Las Vegas randomized algorithm for the Prover with expected cost* $\widetilde{O}(mnr^{\omega-2} + mnd)$. *Otherwise, the probability that the Verifier incorrectly accepts is at most* $(rd+1)/\#\mathsf{S}$.

Proof. If ρ is indeed an upper bound on the rank of $\boldsymbol{A}$, then, whichever evaluation point the Verifier picked, ρ will be an upper bound on the rank of $\boldsymbol{A}(\alpha)$ and, as the certificate from (Dumas and Kaltofen, 2014, Theorem 4) is complete, this certificate is complete.

If ρ is not an upper bound on the rank of $\boldsymbol{A}$, there are two possibilities of failure. Either the Verifier picked an evaluation point for which the rank of $\boldsymbol{A}$ drops, which happens with probability at most $rd/\#\mathsf{S}$ by Lemma 3.1; or the Prover managed to cheat during the execution of Steps 2 to 3 which happens

Protocol 4: RankUpperBound

Public: $\boldsymbol{A} \in \mathsf{F}[x]^{m\times n}$, $\rho \in \mathbb{N}$
Certifies: $\operatorname{rank}(\boldsymbol{A}) \overset{?}{\leqslant} \rho$

	Prover		Verifier
1.			$\alpha \xleftarrow{\$} \mathsf{S}$ $\boldsymbol{v} \xleftarrow{\$} \mathsf{S}^{n\times 1}$
		$\alpha, \boldsymbol{v}$ ⟵	
2.	Find $\boldsymbol{\gamma} \in \mathsf{F}^{n\times 1}$ such that $\boldsymbol{A}(\alpha)\boldsymbol{\gamma} = \boldsymbol{A}(\alpha)\boldsymbol{v}$ and $\lvert\boldsymbol{\gamma}\rvert_H \leqslant \rho$		
		⟶ $\boldsymbol{\gamma}$	
3.			$\lvert\boldsymbol{\gamma}\rvert_H \overset{?}{\leqslant} \rho$ $\boldsymbol{A}(\alpha)\boldsymbol{\gamma} \overset{?}{=} \boldsymbol{A}(\alpha)\boldsymbol{v}$

with probability at most $1/\#\mathsf{S}$ (Dumas and Kaltofen, 2014, Theorem 4). Then, the union bound gives a total probability of $(rd+1)/\#\mathsf{S}$ for the Verifier to accept a wrong answer.

The Prover has to evaluate the matrix at α for a cost of $O(mnd)$, and to find at most ρ linearly independent rows of the matrix over the base field, which costs $\widetilde{O}\left(mnr^{\omega-2}\right)$, hence a total cost of $\widetilde{O}\left(mnr^{\omega-2} + mnd\right)$. The Verifier has to evaluate the matrix at α and to perform two matrix-vector products over the base field, which yields a cost of $O(mnd)$. The communication cost is the one of sending a scalar and two vectors of size n over the base field, that is, $O(n)$. □

From those two certificates, one can immediately infer a certificate for the rank.

Corollary 3.6. *Let r be the actual rank of $\boldsymbol{A}$. Protocol 5 is a probabilistically sound interactive protocol and is complete assuming $\#\mathsf{S} \geqslant rd+1$ in its subprotocol. It requires $O(n)$ communication and Verifier cost $O(mnd)$. If ρ is indeed the rank of $\boldsymbol{A}$, then there is a Las Vegas randomized algorithm for the Prover with expected cost $\widetilde{O}\left(mnr^{\omega-2}d\right)$. Otherwise, the probability that the Verifier incorrectly accepts is at most $(rd+1)/\#\mathsf{S}$.*

3.3 Determinant of polynomial matrices

We follow with a certificate for the determinant of a polynomial matrix. The trick is still the same: the Verifier checks the degree of the provided determinant in order to ensure it is suitable, and then a random evaluation point is sampled

Protocol 5: Rank

Public: $\boldsymbol{A} \in \mathsf{F}[x]^{m \times n}$, $\rho \in \mathbb{N}$
Certifies: $\operatorname{rank}(\boldsymbol{A}) \stackrel{?}{=} \rho$

	Prover	Verifier
1.	Use RankLowerBound$(\boldsymbol{A}, \rho)$	
2.	Use RankUpperBound$(\boldsymbol{A}, \rho)$	

and the actual verification occurs on evaluated input. There are two choices available for the certificate to use over the base field: either Dumas et al. (2016, Section 2), which runs in a constant number of rounds, but requires a minimum field size of n^2, or Dumas et al. (2017, Section 4.1) which runs in n rounds but only requires a minimum field size of 2. Whichever certificate is chosen here, this has no impact on the asymptotic complexities, which are the same for both or on the completeness, as both are complete.

Protocol 6: Determinant

Public: $\boldsymbol{A} \in \mathsf{F}[x]^{n \times n}, \delta \in \mathsf{F}[x]$
Certifies: $\det \boldsymbol{A} = \delta$

	Prover		Verifier
1.			$\deg(\delta) \stackrel{?}{\leqslant} nd$ $\alpha \xleftarrow{\$} \mathsf{S}$ $\beta \leftarrow \delta(\alpha)$
		$\xleftarrow{\alpha}$	
2.	Use FieldDeterminant$(\boldsymbol{A}(\alpha), \beta)$		

Theorem 3.7. *Protocol 6 is a complete and probabilistically sound interactive protocol which requires $O(n)$ communication and Verifier cost $O(n^2 d)$. If δ is indeed the determinant of $\boldsymbol{A}$, there is an algorithm for the Prover which costs $O(n^2 d + n^\omega)$. Otherwise, the probability that the Verifier incorrectly accepts is at most $(nd+1)/\#\mathsf{S}$.*

Proof. For the sake of the proof, write $g \in \mathsf{F}[x]$ as the *actual* determinant of $\boldsymbol{A}$.

If $\delta = g$ and therefore $\delta = \det \boldsymbol{A}$, then it must be the case that $\deg(\delta) \leqslant nd$. Then, as FieldDeterminant is complete, the final check will hold.

If $\delta \neq g$ and therefore $\delta \neq \det \boldsymbol{A}$, there are two possibilities of failure. Either the Verifier picked an α such that $\delta(\alpha) = g(\alpha)$, and in that case the checks from FieldDeterminant will always pass. This is the case if α is a root of $\delta - g$, which, by Schwartz-Zippel lemma, happens with probability $nd/\#\mathsf{S}$; or the Verifier picked an α which is not a root of $\delta - g$ which means they will accept δ as the determinant with the probability of failure of FieldDeterminant, $1/\#\mathsf{S}$. Overall, the probability that the Verifier accepts a wrong statement is at most $(nd+1)/\#\mathsf{S}$ by union bound.

The Prover has to evaluate the matrix at α and to compute a determinant over the base field, hence the cost of $O\left(n^2 d + n^\omega\right)$, the Verifier has to evaluate $\boldsymbol{A}$ at α, which yields a cost of $O\left(n^2 d\right)$ and the communication cost is the one of FieldDeterminant, $O(n)$. □

3.4 Certificates based on matrix multiplication

Finally, we propose some certificates related to matrix multiplication. While they are once again based on evaluation techniques, unlike the previous certificates, the ones given here are non-interactive and thus have no Prover or communication cost. We first give a certificate for linear system solving:

Protocol 7: SystemSolve

Public: $\boldsymbol{A} \in \mathsf{F}[x]^{m\times n}, \boldsymbol{b} \in \mathsf{F}[x]^{m\times 1}, \boldsymbol{v} \in \mathsf{F}[x]^{n\times 1}, \delta \in \mathsf{F}[x]$
Certifies: $\boldsymbol{A}\boldsymbol{v} = \delta\boldsymbol{b}$

Prover	Verifier
1.	$\alpha \xleftarrow{\$} \mathsf{S}$
	$\boldsymbol{A}(\alpha)\boldsymbol{v}(\alpha) - \delta(\alpha)\boldsymbol{b}(\alpha) \stackrel{?}{=} \boldsymbol{0}$

Theorem 3.8. *Let d be an upper bound on the degree of $\boldsymbol{A}$, $\boldsymbol{v}$, $\boldsymbol{b}$, and δ. Then, Protocol 7 is a complete and probabilistically sound non-interactive protocol which has Verifier cost $O(mnd)$. The probability that the Verifier incorrectly accepts is at most $2d/\#\mathsf{S}$.*

Proof. If $\boldsymbol{A}\boldsymbol{v} = \delta\boldsymbol{b}$, then the same holds when evaluating at α which leads to the completeness of this certificate.

Otherwise, we have $\boldsymbol{A}\boldsymbol{v} - \delta\boldsymbol{b} = \Delta$ for some nonzero polynomial vector Δ. If the Prover manages to cheat, it means the Verifier picked an α which is a root of every entry of Δ. The probability of this event is at most the probability of α being a root of one nonzero entry of Δ. Now, let f be a nonzero element of Δ. Its degree must be at least one, for the Verifier to be tricked, and can be at most $2d$. Then, by the Schwartz-Zippel lemma, the Verifier picked an α such that $f(\alpha) = 0$ with probability at most $2d/\#\mathsf{S}$.

The dominating step in the Verifier's checks is evaluating $\boldsymbol{A}$ at α, which costs $O(mnd)$ using Horner's rule. □

Similarly, we propose a certificate for matrix multiplication following an approach from (Freivalds, 1979).

Protocol 8: MatMul

Public: $\boldsymbol{A} \in \mathsf{F}[x]^{m\times n}, \boldsymbol{B} \in \mathsf{F}[x]^{n\times \ell}, \boldsymbol{C} \in \mathsf{F}[x]^{m\times \ell}$
Certifies: $\boldsymbol{C} \stackrel{?}{=} \boldsymbol{AB}$

Prover	Verifier
	$\deg(\boldsymbol{C}) \stackrel{?}{\leqslant} \deg(\boldsymbol{A}) + \deg(\boldsymbol{B})$
1.	$\alpha \xleftarrow{\$} \mathsf{S}$
	$\boldsymbol{v} \xleftarrow{\$} \mathsf{S}^{\ell\times 1}$
	$\boldsymbol{A}(\alpha)(\boldsymbol{B}(\alpha)\boldsymbol{v}) - \boldsymbol{C}(\alpha)\boldsymbol{v} \stackrel{?}{=} \boldsymbol{0}$

Theorem 3.9. *Let $d_{\boldsymbol{A}} = \max(1, \deg(\boldsymbol{A}))$ and similarly for $d_{\boldsymbol{B}}, d_{\boldsymbol{C}}$. Protocol 8 is a complete and probabilistically sound non-interactive protocol which has Verifier cost $O(mnd_{\boldsymbol{A}} + n\ell d_{\boldsymbol{B}} + m\ell d_{\boldsymbol{C}})$. The probability that the Verifier incorrectly accepts is at most $(d_{\boldsymbol{A}} + d_{\boldsymbol{B}} + 1)/\#\mathsf{S}$.*

Proof. Let $\boldsymbol{D}$ be the actual result of $\boldsymbol{A} \times \boldsymbol{B}$, and denote by $\boldsymbol{\Delta}$ the matrix $\boldsymbol{AB} - \boldsymbol{C}$. Note that the value computed on the left hand side on the final check by the Verifier is exactly $\boldsymbol{\Delta}(\alpha)\boldsymbol{v}$.

If $\boldsymbol{C} = \boldsymbol{D}$, then $\boldsymbol{\Delta} = \boldsymbol{0}$ and whichever evaluation point α the Verifier picks, $\boldsymbol{\Delta}(\alpha)$ will always be $\boldsymbol{0}$. The degree bound checked initially by the Verifier is also valid whenever $\boldsymbol{AB} = \boldsymbol{C}$, hence this certificate is complete.

If $\boldsymbol{C} \neq \boldsymbol{D}$, then $\boldsymbol{\Delta}$ is a nonzero matrix with degree at most $d_{\boldsymbol{A}} + d_{\boldsymbol{B}}$. There are two events that would lead to the Verifier accepting a wrong answer: either the Verifier picked an evaluation point which cancels each coefficient in $\boldsymbol{\Delta}$, which is at most the probability that α is a root of a single entry, namely $(d_{\boldsymbol{A}}+d_{\boldsymbol{B}})/\#\mathsf{S}$ by the Schwartz-Zippel lemma and in that case, whichever verification vector $\boldsymbol{v}$ is picked afterwards, the Verifier will always accept; or the Verifier picked an evaluation point for which $\boldsymbol{\Delta}(\alpha) \neq \boldsymbol{0}$ but they picked a unlucky verification vector $\boldsymbol{v}$ in the right kernel of $\boldsymbol{\Delta}(\alpha)$, which happens with probability $1/\#\mathsf{S}$ by Freivalds (1979). The union bound of these two events gives the stated probability that the Verifier incorrectly accepts.

The cost for the Verifier comes from evaluating all three matrices at α using Horner's rule and then performing three matrix-vector products over the base field. □

Verifying a matrix inverse is a straightforward application of the previous protocol.

Corollary 3.10. *For $\boldsymbol{A} \in \mathsf{F}[x]^{n\times n}$ and $\boldsymbol{B} \in \mathsf{F}[x]^{n\times n}$, there exists a non-interactive protocol which certifies that $\boldsymbol{B}$ is the inverse of $\boldsymbol{A}$ in Verifier cost $O\big(n^2 d\big)$, where $d = \max(1, \deg(\boldsymbol{A}), \deg(\boldsymbol{B}))$. If $\boldsymbol{B} \neq \boldsymbol{A}^{-1}$, the probability that the Verifier incorrectly accepts is at most $(2d+1)/\#S$.*

4 Row space membership

In this section we present the main tool for verification problems that are essentially about $\mathsf{F}[x]$-modules, which is to determine whether a given row vector $\boldsymbol{v} \in \mathsf{F}[x]^{1\times n}$ is in the $\mathsf{F}[x]$-linear row span of a given matrix $\boldsymbol{A} \in \mathsf{F}[x]^{m\times n}$.

The approach is in two steps. First, FullRankRowSpaceMembership shows how to solve the problem in case $\boldsymbol{A}$ has full row rank. Then, in RowSpaceMembership, we extend this to the general setting by means of two calls to the full row rank case.

4.1 Full row rank case

In order to prove the soundness of this protocol, we start with a few simple lemmas. The first is a standard extension of the soundness proof of Freivalds' algorithm (1979).

Lemma 4.1. *Let $\boldsymbol{A} \in \mathsf{F}^{m\times n}$ be an arbitrary matrix with at least one nonzero entry. If $S \subseteq \mathsf{F}$ and $\boldsymbol{w} \in S^n$ has its entries chosen uniformly at random from S, then $\Pr[\boldsymbol{A}\boldsymbol{w} = \boldsymbol{0}] \leqslant 1/\#S$.*

Proof. Consider each of the n entries of $\boldsymbol{w}$ as an indeterminate. Because $\boldsymbol{A}$ is not zero, $\boldsymbol{A}\boldsymbol{w}$ has at least one nonzero entry, which is a nonzero polynomial in n variables with total degree 1. Then a trivial application of the Schwartz-Zippel lemma gives the stated result. □

Lemma 4.2. *Let $\boldsymbol{v} \in \mathsf{F}(x)^{1\times n}$ be a rational function vector with $\operatorname{denom}(\boldsymbol{v}) \neq 1$, and $S \subseteq \mathsf{F}$. For a vector of scalars $\boldsymbol{w} \in S^{n\times 1}$ chosen uniformly at random, the probability that their inner product is a polynomial, i.e., that $\operatorname{denom}(\boldsymbol{v}\boldsymbol{w}) = 1$, is at most $1/\#S$.*

Proof. Write $g = \operatorname{denom}(\boldsymbol{v})$ and $\hat{\boldsymbol{v}} = \operatorname{numer}(\boldsymbol{v})$. By the condition of the lemma we know that $\deg(g) \geqslant 1$. We see that the inner product of $\boldsymbol{v}$ and $\boldsymbol{w}$ is a polynomial if and only if the inner product of $\hat{\boldsymbol{v}}$ and $\boldsymbol{w}$ is divisible by g.

Now let h be any irreducible factor of g, and consider the inner product over the extension field $\mathsf{F}[x]/\langle h\rangle$. Because $h \mid \operatorname{denom}(\boldsymbol{v})$, we know that $\hat{\boldsymbol{v}} \bmod h$ is not zero; otherwise the degree of the denominator g is not mininal.

Then, since $\mathsf{F} \subseteq \mathsf{F}[x]/\langle h\rangle$, the stated bound follows from Lemma 4.1. □

Protocol 9: FullRankRowSpaceMembership

Public: $\boldsymbol{A} \in \mathsf{F}[x]^{m \times n}$ with full row rank, $\boldsymbol{v} \in \mathsf{F}[x]^{1 \times n}$
Certifies: $\operatorname{rank}(\boldsymbol{A}) \geqslant m \Rightarrow \boldsymbol{v} \in \operatorname{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A})$

	Prover		Verifier
1.			$\boldsymbol{c} \xleftarrow{\$} \mathsf{S}^{m \times 1}$
		$\xleftarrow{\boldsymbol{c}}$	
2.	$\boldsymbol{u} \in \mathsf{F}[x]^{1 \times m}$ s.t. $\boldsymbol{u}\boldsymbol{A} = \boldsymbol{v}$ $g \leftarrow \boldsymbol{u}\boldsymbol{c}$		
		$\xrightarrow{g}$	
3.			$\deg(g) \overset{?}{\leqslant} m \deg(\boldsymbol{A}) + \deg(\boldsymbol{v})$ $\alpha \xleftarrow{\$} \mathsf{S}$
		$\xleftarrow{\alpha}$	
4.	$\boldsymbol{w} \leftarrow \boldsymbol{u}(\alpha)$		
		$\xrightarrow{\boldsymbol{w}}$	
5.			$\boldsymbol{w}\boldsymbol{A}(\alpha) \overset{?}{=} \boldsymbol{v}(\alpha)$ $\boldsymbol{w}\boldsymbol{c} \overset{?}{=} g(\alpha)$

The final ingredient in our full row space membership algorithm is a subroutine the Prover may use to actually compute the solution to the linear system, shown in Algorithm 1. It will also used in the non-full-rank protocol presented in the next section.

Algorithm 1: Rational linear solving with full row rank

Input: $\boldsymbol{A} \in \mathsf{F}[x]^{m \times n}, \boldsymbol{v} \in \mathsf{F}[x]^{1 \times n}$
Output: Either `LOW_RANK`, `NO_SOLUTION`, or a vector $\boldsymbol{u} \in \mathsf{F}(x)^{1 \times m}$ such that $\boldsymbol{uA} = \boldsymbol{v}$

1 $r, i_1, \ldots, i_r \leftarrow$ column rank profile of $\boldsymbol{A}$
2 **if** $r < m$ **then return** `LOW_RANK`
3 $\boldsymbol{B} \leftarrow$ columns $i_1, \ldots, i_r$ from $\boldsymbol{A}$
4 $\boldsymbol{y} \leftarrow$ columns $i_1, \ldots, i_r$ from $\boldsymbol{v}$
5 $\boldsymbol{u} \leftarrow \boldsymbol{y}\boldsymbol{B}^{-1}$
6 **if** $\boldsymbol{uA} \neq \boldsymbol{v}$ **then return** `NO_SOLUTION`
7 **return** $\boldsymbol{u}$

To simplify the cost bounds, for the remainder of this section we write $d_{\boldsymbol{A}} = \max(1, \deg(\boldsymbol{A}))$ and $d_{\boldsymbol{v}} = \max(1, \deg(\boldsymbol{v}))$.

Lemma 4.3. *Algorithm 1 has worst-case cost bound* $\widetilde{O}\big(m^{\omega-1} n d_{\boldsymbol{A}} + m^{\omega-1} d_{\boldsymbol{v}}\big)$. *If* $\operatorname{rank}(\boldsymbol{A}) < m$, *then* `LOW_RANK` *is returned. Otherwise, if* $\boldsymbol{v} \in \operatorname{RowSp}_{\mathsf{F}(x)}(\boldsymbol{A})$, *then the unique rational solution* $\boldsymbol{u}$ *to* $\boldsymbol{uA} = \boldsymbol{v}$ *is returned.*

Proof. (Zhou, 2012, Chapter 11) presents a deterministic algorithm to compute the column rank profile on Line 1 using $\widetilde{O}\big(m^{\omega-1} n d_{\boldsymbol{A}}\big)$ field operations. This guarantees that `LOW_RANK` is returned whenever $\boldsymbol{A}$ does not have full row rank.

Now assume that $\operatorname{rank}(\boldsymbol{A}) = m$. Then $\boldsymbol{B}$ is nonsingular, (Gupta, Sarkar, Storjohann, and Valeriote, 2012) showed how to de-randomize the high-order lifting technique in order to solve the rational linear system on Line 5 deterministically using $\widetilde{O}\big(m^{\omega} d_{\boldsymbol{A}} + m^{\omega-1} d_{\boldsymbol{v}}\big)$ operations. Let $\boldsymbol{u}$ be the rational solution to $\boldsymbol{uB} = \boldsymbol{y}$ computed on Line 5.

Assume that there exists some rational solution $\boldsymbol{w} \in \mathsf{F}(x)^{1 \times m}$ such that $\boldsymbol{wA} = \boldsymbol{v}$. Then $\boldsymbol{wB} = \boldsymbol{y}$ also. But because $\boldsymbol{B}$ is nonsingular, the solution $\boldsymbol{u}$ is unique; hence $\boldsymbol{w} = \boldsymbol{u}$ and $\boldsymbol{uA} = \boldsymbol{v}$. □

Finally, we present the main result of this subsection.

Theorem 4.4. *Protocol 9 is a complete and probabilistically sound interactive protocol which requires* $O(m d_{\boldsymbol{A}} + d_{\boldsymbol{v}})$ *communication and with Verifier cost* $O(m n d_{\boldsymbol{A}} + n d_{\boldsymbol{v}})$. *If* $\operatorname{rank}(\boldsymbol{A}) = m$ *and* $\boldsymbol{v} \in \operatorname{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A})$, *there is a deterministic algorithm for the Prover with cost* $\widetilde{O}\big(n m^{\omega-1} d_{\boldsymbol{A}} + m^{\omega-1} d_{\boldsymbol{v}}\big)$. *Otherwise, the probability that the Verifier incorrectly accepts is at most* $(3 m d_{\boldsymbol{A}} + d_{\boldsymbol{v}} + 1)/\#S$.

Proof. If $\boldsymbol{v}$ is the zero vector, then the protocol easily succeeds when the Prover sends all zeros for g and $\boldsymbol{w}$. And if $\operatorname{rank}(\boldsymbol{A}) < m$, the implication being verified

is vacuously true. So assume for the remainder of the proof that $\boldsymbol{v}$ is nonzero and $\boldsymbol{A}$ has full row rank m.

The degree check by the Verifier assures that g contains at most $md_{\boldsymbol{A}}+d_{\boldsymbol{v}}+1$ field elements, bringing the total communication over Steps 2 to 5 to at most $(d_{\boldsymbol{A}}+2)m+d_{\boldsymbol{v}}+2$ field elements.

The work of the verifier is dominated by computing the evaluations $\boldsymbol{A}(\alpha)$ and $\boldsymbol{v}(\alpha)$ on the last step. Using Horner's rule the total cost for these is $O(mnd_{\boldsymbol{A}}+nd_{\boldsymbol{v}})$, as claimed.

We now divide the proof into three cases, depending on whether $\boldsymbol{v}$ is in the *polynomial* row span of $\boldsymbol{A}$ (as checked by the protocol), the *rational* row span of $\boldsymbol{A}$, or neither.

Case 1: $\boldsymbol{v} \in \mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A})$. Here we want to prove that an honest Prover and Verifier succeed with costs as stated in the theorem.

The vector $\boldsymbol{u}$ as defined in Step 2 must exist by the definition of $\mathrm{RowSp}_{\mathsf{F}[x]}$, and computing $\boldsymbol{u}$ can be completed by the Verifier according to Lemma 4.3 in the stated cost bound.

If the computations of $\boldsymbol{u}$ and g are performed correctly by the Prover on Step 2, then the Verifier's checks on Step 5 will succeed for any choice of α.

This proves the completeness of the protocol.

Case 2: $\boldsymbol{v} \in \mathrm{RowSp}_{\mathsf{F}(x)}(\boldsymbol{A}) \setminus \mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A})$. In this case, the assertion of the protocol is false, and we want to show probabilistic *soundness*.

Let $\boldsymbol{c} \in \mathsf{F}^{m\times 1}$ be the random vector chosen by the Verifier on Step 1. By the assumption of this case, there is a unique rational solution $\boldsymbol{u} \in \mathsf{F}(x)^{1\times m}$ with $\boldsymbol{uA} = \boldsymbol{v}$, and Lemma 4.2 tells us the probability that $\boldsymbol{uc}$ is a polynomial is at most $1/\#\mathsf{F}$. If $\boldsymbol{uc}$ is not a polynomial, then $\boldsymbol{uc}-g$ is a nonzero rational function with numerator degree at most

$$\deg(g) + \deg(\mathrm{denom}(\boldsymbol{u})) \leqslant 2md_{\boldsymbol{A}} + d_{\boldsymbol{v}}. \tag{4.1}$$

From Lemma 3.1, the probability that $\boldsymbol{A}(\alpha)$ is singular is at most $md_{\boldsymbol{A}}/\#\mathsf{F}$. Otherwise, the vector $\boldsymbol{w} = \boldsymbol{u}(\alpha)$ is the unique solution to $\boldsymbol{w}\boldsymbol{A}(\alpha) = \boldsymbol{v}(\alpha)$, so the Prover is obliged to send this $\boldsymbol{w}$ on Step 4.

If the Verifier incorrectly accepts, we must have $\boldsymbol{wc} = g(\alpha)$, which means that $\boldsymbol{u}(\alpha)\boldsymbol{c} = g(\alpha)$. The degree bound in (4.1) gives an upper bound on the number of $\alpha \in \mathsf{F}$ which could satisfy this equation.

Therefore the Verifier accepts only when either $\boldsymbol{uc} \in \mathsf{F}[x]$, or $\boldsymbol{A}(\alpha)$ is singular, or α is a root of $\boldsymbol{uc}-g$, which by the union bound has probability at most $(3md_{\boldsymbol{A}}+d_{\boldsymbol{v}}+1)/\#F$, as stated.

Case 3: $\boldsymbol{v} \notin \mathrm{RowSp}_{\mathsf{F}(x)}(\boldsymbol{A})$. Again, the assertion of the protocol is false, and our goal is to prove probabilistic soundness. As with the last case, assume by way of contradiction that the Verifier accepts.

Consider the augmented system

$$\tilde{\boldsymbol{A}} = \begin{pmatrix} \boldsymbol{A} \\ \boldsymbol{v} \end{pmatrix}.$$

By the assumption of this case, $\operatorname{rank}(\tilde{\boldsymbol{A}}) = \operatorname{rank}(\boldsymbol{A}) + 1 = m + 1$. But the solution vector $\boldsymbol{w}$ provided to solve $\boldsymbol{w}\boldsymbol{A}(\alpha) = \boldsymbol{v}(\alpha)$ on the last step corresponds to a nonzero vector in the left kernel of $\tilde{\boldsymbol{A}}(\alpha)$, which therefore has rank at most m.

The proof of Lemma 3.1 shows that the probability that $\operatorname{rank}(\tilde{\boldsymbol{A}}(\alpha)) \leqslant m$ is at most $(md_{\boldsymbol{A}} + d_{\boldsymbol{v}})/\#F$. □

4.2 Arbitrary rank case

Now we move to the general case that $\operatorname{rank}(\boldsymbol{A}) \leqslant m$.

The idea of the protocol is inspired by Mulders and Storjohann (2004). Consider a matrix $\boldsymbol{C} \in \mathsf{F}[x]^{r\times m}$ such that $\boldsymbol{C}\boldsymbol{A}$ has full row rank and therefore the same *rational* row span as $\boldsymbol{A}$. Then there is a unique rational vector $\boldsymbol{w} \in \mathsf{F}(x)^{1\times r}$ such that $\boldsymbol{w}\boldsymbol{C}\boldsymbol{A} = \boldsymbol{v}$. If $\operatorname{denom} \boldsymbol{w} = 1$, the verification is already complete.

But even when $\boldsymbol{w}$ has nontrivial denominator, this approach can still be used for verification by considering *multiple* such matrices $\boldsymbol{C}$ and rational solutions $\boldsymbol{w}$. In fact, the greatest common divisor of all such rational solutions is 1 if and only if $\boldsymbol{v} \in \operatorname{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A})$, as we show in the next lemma.

Lemma 4.5. *Let $\boldsymbol{A} \in \mathsf{F}[x]^{m\times n}$ with $r = \operatorname{rank}(\boldsymbol{A})$, $\boldsymbol{v} \in \mathsf{F}[x]^{1\times n}$, $\boldsymbol{C}_1, \ldots, \boldsymbol{C}_t \in \mathsf{F}[x]^{r\times n}$, and $d_1, \ldots, d_t \in \mathsf{F}[x]$, such that, for every $i = 1, 2, \ldots, t$, we have:*

- $\operatorname{rank}(\boldsymbol{C}_i\boldsymbol{A}) = \operatorname{rank}(\boldsymbol{A}) = r$; *and*
- $d_i\boldsymbol{v} \in \operatorname{RowSp}_{\mathsf{F}[x]}(\boldsymbol{C}_i\boldsymbol{A})$.

If $\gcd(d_1, \ldots, d_t) = 1$, then $\boldsymbol{v} \in \operatorname{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A})$.

Proof. Let $\mathfrak{I}$ be the set $\{d \in \mathsf{F}[x] \mid d\boldsymbol{v} \in \operatorname{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A})\}$. We see that $\mathfrak{I}$ is an ideal in $\mathsf{F}[x]$.

For each $i \in \{1, \ldots, t\}$, there exists a polynomial vector $\boldsymbol{w} \in \mathsf{F}[x]^{1\times r}$ such that $\boldsymbol{w}\boldsymbol{C}_i\boldsymbol{A} = d_i\boldsymbol{v}$. Then $\boldsymbol{w}\boldsymbol{C}_i \in \mathsf{F}[x]^{1\times m}$ is also a polynomial vector, which shows that each $d_i \in \mathfrak{I}$.

Because $\mathsf{F}[x]$ is a principal ideal domain, $\gcd(d_1, \ldots, d_t) \in \mathfrak{I}$ also, and therefore $1 \in \mathfrak{I}$. By the definition of $\mathfrak{I}$, this means that $\boldsymbol{v} \in \operatorname{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A})$. □

Before giving the full protocol for row membership, we first present a subprotocol in CoPrime to confirm that the greatest common divisor of a set of polynomials is 1.

Lemma 4.6. *Let $d = \max_i \deg(f_i)$, and suppose $\#S \geqslant 2d$. Then Protocol 10 is a complete and probabilistically sound interactive protocol which requires $O(d + t)$ communication and Verifier cost $O(dt)$. If $\gcd(f_1, \ldots, f_t) \neq 1$,*

Protocol 10: CoPrime

Public: $f_1, \ldots, f_t \in \mathsf{F}[x]$
Certifies: $\gcd(f_1, \ldots, f_t) = 1$

	Prover		Verifier
1.	Compute polynomials $s_1, s_2 \in \mathsf{F}[x]$ and scalars $\beta_3, \ldots, \beta_t \in \mathsf{F}$ s.t. $f_1 s_1 + h s_2 = 1$, where $h = f_2 + \sum_{i=3}^{t} \beta_i f_i$	$\xrightarrow{s_1, s_2, \beta_3, \ldots, \beta_t}$	
2.			$\deg(s_1) \overset{?}{<} \max_{i \geqslant 2} \deg(f_i)$ $\deg(s_2) \overset{?}{<} \deg(f_1)$ $\alpha \xleftarrow{\$} \mathsf{S}$ $f_1(\alpha) s_1(\alpha) + h(\alpha) s_2(\alpha) \overset{?}{=} 1$

the probability that the Verifier incorrectly accepts is at most $(2d-1)/\#\mathsf{S}$. *Otherwise, there is a Las Vegas randomized algorithm for the Prover with expected cost bound* $\widetilde{O}(dt)$ *which will cause the Verifier to accept.*

Proof. The communication and Verifier costs are clear.

Write $g = \gcd(f_1, \ldots, f_t)$, and suppose first that $g \neq 1$. Then $g \mid (f_1 s_1 + h s_2)$, so the polynomial $f_1 s_1 + h s_2 - 1$ is nonzero and has degree at most $2d-1$. If the Verifier incorrectly accepts, then α must be a root of this polynomial, which justifies the probability claim.

If $g = 1$, then a well-known argument (von zur Gathen and Gerhard, 2003, Theorem 6.46) says that, for $\beta_3, \ldots, \beta_t$ chosen randomly from a subset $\mathsf{S} \subseteq \mathsf{F}$, the probability that $\gcd(f_1, h) \neq \gcd(f_1, \ldots, f_t)$ is at most $d/\#\mathsf{S}$. Based on the assumption that $\#\mathsf{S} \geqslant 2d$, the Prover can find such a tuple $\beta_3, \ldots, \beta_t$ after expected $O(1)$ iterations. Then computing the Bézout coefficients s_1, s_2 is a matter of computing the extended Euclidean algorithm on f_1 and h, which has the stated cost. □

Protocol RowSpaceMembership shows an interactive certificate for row space membership which relies on Lemma 4.5. The Verifier first selects t matrices $\boldsymbol{C}_i$ so that the corresponding denominators d_i of the rational solutions to $\boldsymbol{w}\boldsymbol{C}_i\boldsymbol{A} = \boldsymbol{v}$ have no common factor. As we will see, it suffices to choose the $\boldsymbol{C}_i$'s to be Toeplitz matrices; then sending these as well as the denominators d_i only requires $O(m + r \deg(\boldsymbol{A}))$ communication. The Verifier then confirms that the gcd of all denominators is 1 using CoPrime. Finally, Protocols RankLowerBound and FullRankRowSpaceMembership are used for each $i = 1, \ldots, t$ to confirm that

the conditions of Lemma 4.5 hold.

Protocol 11: RowSpaceMembership

Public: $\boldsymbol{A} \in \mathsf{F}[x]^{m \times n}$, $\boldsymbol{v} \in \mathsf{F}[x]^{1 \times n}$
Certifies: $\boldsymbol{v} \in \mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A})$

	Prover		Verifier
	$\rho \leftarrow \mathrm{rank}(\boldsymbol{A})$ $t \leftarrow 1 + \lceil \log_{\#\mathsf{S}/\rho}(2r \deg(\boldsymbol{A})) \rceil$ Compute Toeplitz $\boldsymbol{C}_i, \ldots, \boldsymbol{C}_t \in \mathsf{F}^{\rho \times m}$		
1.	and polynomials $d_1, \ldots, d_t, s_1, s_2 \in \mathsf{F}[x]$ s.t. $\forall i, \mathrm{rank}(\boldsymbol{C}_i \boldsymbol{A}) = \rho$, and $\forall i, d_i \boldsymbol{v} \in \mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{C}_i \boldsymbol{A})$, and $\gcd(d_1, \ldots, d_t) = 1$	$\xrightarrow{\rho, \boldsymbol{C}_1, \ldots, \boldsymbol{C}_t, d_1, \ldots, d_t}$	
2.			$\rho \stackrel{?}{\leqslant} \min(m, n)$ $\forall i, \deg(d_i) \stackrel{?}{\leqslant} \rho \deg(\boldsymbol{A})$
3.		Check $\mathrm{rank}(\boldsymbol{A}) \stackrel{?}{\leqslant} \rho$ using RankUpperBound	
4.		**for** $i = 1, \ldots, t$ **do** check $\mathrm{rank}(\boldsymbol{C}_i \boldsymbol{A}) \stackrel{?}{\geqslant} \rho$ using RankLowerBound	
5.		Check $\gcd(d_1, \ldots, d_t) \stackrel{?}{=} 1$ using CoPrime	
6.		**for** $i = 1, \ldots, t$ **do** check $d_i \boldsymbol{v} \stackrel{?}{\in} \mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{C}_i \boldsymbol{A})$ using FullRankRowSpaceMembership	

We now proceed to show how the Prover can actually find the values required on Step 1. We write $r = \mathrm{rank}(\boldsymbol{A})$; if the Prover is honest, then in fact $r = \rho$.

For the purposes of the proof, we need a factorization $\boldsymbol{AP} = \boldsymbol{UB}$, where $\boldsymbol{AP}$ is a subset of r linearly independent *pivot* columns from $\boldsymbol{A}$, and $\boldsymbol{B}$ is a *reduced row basis* for $\boldsymbol{AP}$.

Definition 4.7. *For a matrix $\boldsymbol{A} \in \mathsf{F}[x]^{m \times n}$ with $\mathrm{rank}(\boldsymbol{A}) = r$, a* pivot-only row basis *for $\boldsymbol{A}$ is a triple of matrices $\boldsymbol{P}, \boldsymbol{U}, \boldsymbol{B}$ such that $\boldsymbol{AP} = \boldsymbol{UB}$ and we have:*

- *$\boldsymbol{P} \in \{0, 1\}^{n \times r}$ selects r linearly independent columns of $\boldsymbol{A}$;*
- *$\boldsymbol{B} \in \mathsf{F}[x]^{r \times r}$ is nonsingular and has the same $\mathsf{F}[x]$-linear span as $\boldsymbol{AP}$; and*

- $\boldsymbol{U} \in \mathsf{F}[x]^{m\times r}$ *can be* completed *to a square unimodular matrix, meaning there exists some matrix* $\boldsymbol{V} \in \mathsf{F}[x]^{m\times(m-r)}$ *such that* $\det(\boldsymbol{U}|\boldsymbol{V}) \in \mathsf{F}\backslash\{0\}$.

Such a factorization always exists, for example by computing the Hermite or Popov form of $\boldsymbol{A}$ and discarding the information from the non-pivot columns. See Neiger et al. (2018) for the currently-best algorithms to compute such row bases efficiently.

The next lemma is inspired by Mulders and Storjohann (2004, Lemmas 19 & 20).

Lemma 4.8. *Let* $\boldsymbol{A} \in \mathsf{F}[x]^{m\times n}$ *and* $\boldsymbol{v} \in \mathsf{F}[x]^{1\times n}$ *such that* $\operatorname{rank}(\boldsymbol{A}) = r$ *and* $\boldsymbol{v} \in \operatorname{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A})$*, and let* $p \in \mathsf{F}[x]$ *be an irreducible polynomial. If* $\boldsymbol{C} \in \mathsf{F}^{r\times m}$ *is a Toeplitz matrix with entries chosen uniformly at random from a subset* S *of* F*, and* $\boldsymbol{w} \in \mathsf{F}(x)^{1\times n}$ *is any rational solution to* $\boldsymbol{wCA} = \boldsymbol{v}$*, then the probability that* $\operatorname{rank}(\boldsymbol{CA}) < r$ *or that* p *divides* $\operatorname{denom}\boldsymbol{w}$ *is at most* $r/\#\mathsf{S}$.

Proof. Let $\boldsymbol{P}, \boldsymbol{U}, \boldsymbol{B}$ be a pivot-only row basis of $\boldsymbol{A}$.

First we bound the probability that $p \mid \det(\boldsymbol{CU})$, by working over the quotient field $\mathsf{F}[x]/\langle p\rangle$.

Recall from Definition 4.7 that $\boldsymbol{U}$ completes to a unimodular matrix over $\mathsf{F}[x]$, which must also be unimodular (hence nonsingular) over $\mathsf{F}[x]/\langle p\rangle$. Therefore $\operatorname{rank}(\boldsymbol{U}) = r$ over $\mathsf{F}[x]/\langle p\rangle$. Treating the $r+m-1$ distinct entries of $\boldsymbol{C}$ as new indeterminates $z_1, z_2, \ldots$, then $\det(\boldsymbol{CU}) \bmod p$ is a nonzero polynomial in $\mathsf{F}[x]/\langle p\rangle[z_1, z_2, \ldots]$ with total degree r. All entries of S are distinct modulo p, so by the Schwartz-Zippel lemma, the probability that $\det(\boldsymbol{CU}) \bmod p = 0$ is at most $r/\#\mathsf{S}$.

If $p \nmid \det(\boldsymbol{CU})$, then the matrix $\boldsymbol{CU}$ must have full rank r, which also implies that $\operatorname{rank}(\boldsymbol{CA}) = r$ since

$$\operatorname{rank}(\boldsymbol{CU}) = \operatorname{rank}(\boldsymbol{CUB}) = \operatorname{rank}(\boldsymbol{CAP}) \leqslant \operatorname{rank}(\boldsymbol{CA}).$$

Next we show that whenever $\boldsymbol{CA}$ has rank r, $\operatorname{denom}\boldsymbol{w}$ divides $\det(\boldsymbol{CU})$.

By Definition 4.7 we have $\boldsymbol{AP} = \boldsymbol{UB}$, and $\boldsymbol{B}$ is nonsingular with the same polynomial row space as $\boldsymbol{AP}$. Because $\boldsymbol{v} \in \operatorname{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A})$, then $\boldsymbol{vP} \in \operatorname{RowSp}_{\mathsf{F}[x]}(\boldsymbol{B})$, so there exists $\boldsymbol{y} \in \mathsf{F}[x]^{1\times r}$ such that $\boldsymbol{yB} = \boldsymbol{vP}$. Finally, assuming $\boldsymbol{CU}$ is nonsingular, there exists an *adjugate* matrix $\boldsymbol{D} \in \mathsf{F}[x]^{r\times r}$ such that $\det(\boldsymbol{CU})(\boldsymbol{CU})^{-1} = \boldsymbol{D}$. Putting these facts together, we have

$$\begin{aligned}
\boldsymbol{wCA} &= \boldsymbol{v} \\
\boldsymbol{wCAP} &= \boldsymbol{vP} \\
\boldsymbol{wCUB} &= \boldsymbol{yB} \\
\boldsymbol{wCU} &= \boldsymbol{y} \\
\boldsymbol{w}\det(\boldsymbol{CU}) &= \boldsymbol{yD}.
\end{aligned}$$

Because the right-hand side of the last equation has entries in $\mathsf{F}[x]$, then so does the left-land side, which means that $\det(\boldsymbol{CU})$ must be a multiple of $\operatorname{denom}\boldsymbol{w}$.

In summary, we see that $p \nmid \operatorname{denom}(\boldsymbol{CU})$ with probability at least $1 - r/\#\mathsf{S}$, which in turn implies that $\boldsymbol{CA}$ has full rank r and that $p \nmid \operatorname{denom}\boldsymbol{w}$. □

Repeatedly applying the previous lemma leads to a Las Vegas randomized algorithm for an honest Prover, based on repeated calls to the rational linear solver of Algorithm 1.

Algorithm 2: Honest Prover for RowSpaceMembership

Input: $\boldsymbol{A} \in \mathsf{F}[x]^{m\times n}, \boldsymbol{v} \in \mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A}), \mathsf{S} \subseteq \mathsf{F}$
Output: $r, \boldsymbol{C}_1, \ldots, \boldsymbol{C}_t, d_1, \ldots, d_t$ satisfying the conditions of Step 1 from Protocol 11.

1 $r \leftarrow \mathrm{rank}(\boldsymbol{A})$
2 $t \leftarrow 1 + \lceil \log_{\#\mathsf{S}/r}(2r \deg(\boldsymbol{A})) \rceil$
3 **repeat**
4 $i \leftarrow 1$
5 **while** $i \leqslant t$ **do**
6 $\boldsymbol{C}_i \leftarrow$ random $(r \times m)$ Toeplitz matrix with entries from S
7 $\boldsymbol{w_i} \leftarrow$ solution to $\boldsymbol{w_i}\boldsymbol{C}_i\boldsymbol{A} = \boldsymbol{v}$ from Algorithm 1
8 **if** $\boldsymbol{w_i}$ *is not* `LOW_RANK` **then**
9 $d_i \leftarrow \mathrm{denom}\, \boldsymbol{w_i}$
10 $i \leftarrow i + 1$
11 **until** $\gcd(d_1, \ldots, d_t) = 1$
12 **return** r, $\boldsymbol{C}_1, \ldots, \boldsymbol{C}_t$, *and* $d_1, \ldots, d_t$

Lemma 4.9. *If* $\boldsymbol{v} \in \mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A})$ *and* $\#\mathsf{S} \geqslant 2\,\mathrm{rank}(\boldsymbol{A})$*, then Algorithm 2 is a correct Las Vegas randomized algorithm with expected cost bound*

$$\widetilde{O}\big(mnr^{\omega-2}d_{\boldsymbol{A}} + r^{\omega-1}d_{\boldsymbol{v}}\big).$$

Proof. To compute the rank deterministically in the stated cost bound, we may use the column rank profile algorithm from Section 11 of (Zhou, 2012), just as was used in Algorithm 1.

Each matrix product $\boldsymbol{C}_i\boldsymbol{A}$ can be explicitly computed in $\widetilde{O}(mnd_{\boldsymbol{A}})$ operations using fast Toeplitz-vector products and fast polynomial multiplication (Bini and Pan, 1994, Problem 5.1).

If the algorithm returns, correctness is clear from the correctness of Algorithm 1.

What remains is to prove the expected number of iterations of each nested loop.

The inner while loop on Lines 5 to 10 iterates until t random Toeplitz matrices $\boldsymbol{C}_i$ are found all with denominators not divisible by x. Lemma 4.8 tells us that the probability of finding such a matrix $\boldsymbol{C}_i$ and incrementing i at each iteration is at least $1 - r/\#\mathsf{S}$, and therefore the expected number of iterations of the while loop until i reaches t is at most $2t$.

Finally, to discover the expected number of iterations of the outer loop on Lines 3 to 11, we need the probability that $\gcd(d_1, \ldots, d_t) = 1$ given that each $\boldsymbol{C}_i\boldsymbol{B}$ has full row rank r.

If $\gcd(d_1, \ldots, d_t) \neq 1$, then there must be some irreducible factor p of d_1 which is also a factor of every other denominator $d_2, \ldots, d_t$. For a single factor p, according to Lemma 4.8 and because the d_i's are chosen independently of each other, the probability that this happens is at most $(r/\#\mathsf{S})^{t-1}$.

The degree of d_1 is at most $\det(\boldsymbol{C}_i\boldsymbol{B})$, which is at most $rd_{\boldsymbol{A}}$; this also gives an upper bound on the number of distinct irreducible factors p of d_1. Taking the union bound we see that the probability of *any* factor being shared by all other denominators is at most

$$\frac{r^t d_{\boldsymbol{A}}}{(\#\mathsf{S})^{t-1}},$$

which is at most $\frac{1}{2}$ from the definition of t. Therefore the expected number of iterations of the outer loop on Line 3 is $O(1)$.

The stated cost bound follows from Lemma 4.3. It does not depend explicitly on t because we can see that $t \in O(\log(rd_{\boldsymbol{A}}))$, which is subsumed by the soft-oh notation. □

For the sake of simplicity in presentation, and because they do not affect the asymptotic cost bound, we have omitted a few optimizations to the Prover's algorithm that would be useful in practice, namely:

- The Prover can reduce to the full column rank case by computing a column rank profile of $\boldsymbol{A}$ once at the beginning (using Zhou (2012, Chapter 11)), and then removing corresponding non-pivot columns from $\boldsymbol{A}$ and $\boldsymbol{v}$. This does not change the correctness, but means that each matrix $\boldsymbol{C}_i\boldsymbol{A}$ is square.

- When each $\boldsymbol{C}_i\boldsymbol{A}$ is square, instead of calling Algorithm 1, we may instead first check that $\boldsymbol{C}_i\boldsymbol{A} \bmod x$ is nonsingular to confirm the rank, and then use the high-order lifting algorithm of Storjohann (2003) directly to compute $\boldsymbol{w_i}$.

- The column rank profile and rank-preserving evaluation point $\alpha = 0$ may be re-used in the sub-protocols RankLowerBound confirming that each $\operatorname{rank}(\boldsymbol{C}_i\boldsymbol{A}) \geqslant r$.

- The solution vectors $\boldsymbol{w_i}$ may be re-used in the sub-protocols FullRankRowSpaceMembership confirming that each $d_i\boldsymbol{v} \in \operatorname{RowSp}_{\mathsf{F}[x]}(\boldsymbol{C}_i\boldsymbol{A})$.

We conclude the section by proving RowSpaceMembership is complete, sound, and efficient. As in the previous section, write $d_{\boldsymbol{A}} = \max(1, \deg(\boldsymbol{A}))$ and $d_{\boldsymbol{v}} = \max(1, \deg(\boldsymbol{v}))$, and let $r = \operatorname{rank}(\boldsymbol{A})$.

Theorem 4.10. *Whenever $\#S \geqslant 2\min(m, n)d_{\boldsymbol{A}}$, then Protocol 11 is a complete and probabilistically sound interactive protocol which requires $O(n + md_{\boldsymbol{A}}t + d_{\boldsymbol{v}}t)$ communication and Verifier cost $O(mnd_{\boldsymbol{A}}t + nd_{\boldsymbol{v}}t)$. If $\boldsymbol{v} \in \operatorname{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A})$, there is a Las Vegas randomized algorithm for the Prover with expected cost*

$\widetilde{O}\big(mnr^{\omega-2}d_{\boldsymbol{A}} + r^{\omega-1}d_{\boldsymbol{v}}\big)$. *Otherwise, the probability that the Verifier incorrectly accepts is at most*

$$\frac{4rd_{\boldsymbol{A}} + d_{\boldsymbol{v}} + 1}{\#S}.$$

Proof. For the communication, note that because each $\boldsymbol{C}_i$ is a Toeplitz matrix over the ground field, sending each $\boldsymbol{C}_i$ requires only $\rho + m - 1$ field elements, which is $O(m)$. Furthermore, the Verifier does not actually compute the products $\boldsymbol{C}_i\boldsymbol{A}$, but rather uses these as a *black box* for matrix-vector products in the two sub-protocols. For any scalar $\alpha \in \mathsf{F}$, the complexity of computing $\boldsymbol{C}_i\boldsymbol{A}(\alpha)$ times any vector of scalars on the left or right-hand side is $O(mnd_{\boldsymbol{A}})$.

Along with the degree conditions on each d_i and Theorems 3.4, 3.5 and 4.4 and Lemma 4.6, this proves the communication and Verifier cost claims.

The Prover's cost comes from Lemma 4.9, which dominates the cost for the Prover in any of the sub-protocols.

If all the statements being verified on Steps 3 to 6 are true, then the conditions of Lemma 4.5 are satisfied, which proves that $\boldsymbol{v} \in \mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A})$.

Finally, when this statement being proven is not true, we want to know an upper bound on the probability that the Verifier incorrectly accepts. For the remainder of the proof, assume that $\boldsymbol{v} \notin \mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A})$, and divide into cases depending on which sub-protocol incorrectly accepted:

Case 1: $\rho < \mathrm{rank}(\boldsymbol{A})$**.** According to Theorem 3.5, the probability that the Verifier incorrectly accepts in RankUpperBound on Step 3 is at most $(rd_{\boldsymbol{A}} + 1)/\#\mathsf{S}$.

Case 2: $\mathrm{rank}(\boldsymbol{A}) \leqslant \rho$ **and** $\exists i \in \{1, \ldots, t\}$ **s.t.** $\mathrm{rank}(\boldsymbol{C}_i\boldsymbol{A}) < \rho$**.** In this case, the statement being checked in the ith iteration of Step 4 is false, and the Verifier will only accept in sub-protocol RankLowerBound with probability at most $\frac{1}{\#\mathsf{S}}$ according to Theorem 3.4.

Now observe that if $\mathrm{rank}(\boldsymbol{A}) \leqslant \rho$ and for any i, $\mathrm{rank}(\boldsymbol{C}_i\boldsymbol{A}) \geqslant \rho$, then it must be the case that $\mathrm{rank}(\boldsymbol{A}) = \mathrm{rank}(\boldsymbol{C}_i\boldsymbol{A}) = \rho$. Further cases make this assumption.

Case 3: $\forall i \in \{1, \ldots, t\}, \mathrm{rank}(\boldsymbol{A}) = \mathrm{rank}(\boldsymbol{C}_i\boldsymbol{A}) = \rho$ **and** $\gcd(d_1, \ldots, d_t) \neq 1$**.** In this case, we know that each $\deg(d_i) \leqslant rd_{\boldsymbol{A}}$, where r is the true rank of $\boldsymbol{A}$. By Lemma 4.6, the probability that the Verifier incorrectly accepts in sub-protocol CoPrime is at most $(\max_i \deg(d_i))/\#\mathsf{S}$, which is at most $rd_{\boldsymbol{A}}/\#\mathsf{S}$.

Case 4: $\forall i \in \{1, \ldots, t\}, \mathrm{rank}(\boldsymbol{A}) = \mathrm{rank}(\boldsymbol{C}_i\boldsymbol{A}) = \rho$ **and** $\gcd(d_1, \ldots, d_t) = 1$**.** In this sub-case, according to Lemma 4.5, there must exist $i \in \{1, \ldots, t\}$ such that $d_i\boldsymbol{v} \notin \mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{C}_i\boldsymbol{A})$. By the assumption of this case, we know that $\boldsymbol{C}_i\boldsymbol{A}$ has full row rank $\delta \leqslant r$, and also that $\deg(d_i\boldsymbol{v}) \leqslant rd_{\boldsymbol{A}} + d_{\boldsymbol{v}}$. Therefore from Theorem 4.4, the probability that the Verifier incorrectly accepts in FullRankRowSpaceMembership on the ith iteration of Step 6 is at most $(4rd_{\boldsymbol{A}} + d_{\boldsymbol{v}} + 1)/\#\mathsf{S}$.

Observe that the four cases are disjoint and cover all possibilities. In every case, the probability that the Verifier incorrectly accepts is at most that in Case 4, which confirms the last part of the Theorem statement. □

We note that it is always possible to conduct the checks on Step 4 and Step 6

of RowSpaceMembership in parallel, so that the total *rounds* of communication in the protocol is $O(1)$.

A crucial factor in the communication and Verifier costs as seen in Theorem 4.10 is the value of t, which in any case satisfies $t \in O(\log(\min(m,n)))$ due to the condition on the size of S, so this adds only a logarithmic factor to the cost. Indeed, when the set S of field elements is large enough, t can be as small as 2. For clarity, we state as a corollary the conditions in which this logarithmic factor can be eliminated.

Corollary 4.11. *If $\boldsymbol{A}$ has full row rank or $\#\mathsf{S} \geqslant 2mnd_{\boldsymbol{A}}$, then Protocol 11 requires only $O(n + md_{\boldsymbol{A}} + d_{\boldsymbol{v}})$ communication and Verifier cost $O(mnd_{\boldsymbol{A}} + nd_{\boldsymbol{v}})$.*

5 Row spaces and normal forms

In this section, we use the row space membership protocol from the previous section in order to certify the equality of the row spaces of two matrices. Along with additional non-interactive checks by the Verifier, this can also be applied to prove the correctness of certain important normal forms of polynomial matrices.

5.1 Row space subset and row basis

We will use row space membership to give a protocol for the certification of *row space subset*; by this we mean the problem of deciding whether the row space of $\boldsymbol{A}$ is contained in the row space of $\boldsymbol{B}$, for two given matrices $\boldsymbol{A}$ and $\boldsymbol{B}$.

Our approach is the following: take a random vector $\boldsymbol{\lambda}$ and certify that the row space element $\boldsymbol{\lambda A}$ is in the row space of $\boldsymbol{B}$, the latter being done via row space membership (Section 4). We will see that taking $\boldsymbol{\lambda}$ with coefficients from F is enough to ensure good probability of success.

Lemma 5.1. *Let $\boldsymbol{A} \in \mathsf{F}[x]^{m\times n}$ and $\boldsymbol{B} \in \mathsf{F}[x]^{\ell\times n}$. Assuming that*

$$\mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A}) \not\subseteq \mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{B}),$$

then the F-vector space

$$R = \left\{ \boldsymbol{\lambda} \in \mathsf{F}^{1\times m} \mid \boldsymbol{\lambda A} \in \mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{B}) \right\}$$

has dimension at most $m-1$. If the entries of $\boldsymbol{\lambda}$ are chosen uniformly at random from a finite subset $\mathcal{S} \subseteq \mathsf{F}$ then $\boldsymbol{\lambda A} \notin \mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{B})$ with probability at least $1 - \frac{1}{\#\mathcal{S}}$.

Proof. Suppose that the vector space R has dimension at least m. Then R is the entire space $\mathsf{F}^{1\times m}$, and every row of $\boldsymbol{A}$ is in $\mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{B})$; hence $\mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A}) \subseteq \mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{B})$, a contradiction.

Then the probability that a uniformly random vector belongs to a proper subspace of $\mathsf{F}^{1\times m}$ comes from Lemma 4.1. □

Protocol 12: RowSpaceSubset

Public: $\boldsymbol{A} \in \mathsf{F}[x]^{m\times n}$, $\boldsymbol{B} \in \mathsf{F}[x]^{\ell\times n}$
Certifies: $\mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A}) \subseteq \mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{B})$

Prover		Verifier
1.		$\boldsymbol{\lambda} \xleftarrow{\$} \mathsf{S}^{1\times m}$
2.		$\boldsymbol{v} \leftarrow \boldsymbol{\lambda}\boldsymbol{A}$
	$\xleftarrow{\boldsymbol{\lambda}}$	
3.	Check $\boldsymbol{v} \stackrel{?}{\in} \mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{B})$ using RowSpaceMembership	

In the following, let $r_{\boldsymbol{A}}$ and $r_{\boldsymbol{B}}$ denote respectively the rank of $\boldsymbol{A}$ and $\boldsymbol{B}$ and $d_{\boldsymbol{A}} = \max(1, \deg(\boldsymbol{A}))$, $d_{\boldsymbol{B}} = \max(1, \deg(\boldsymbol{B}))$.

Theorem 5.2. *Protocol 12 is a probabilistically sound interactive protocol, and is complete assuming $\#S \geqslant 2\ell d_{\boldsymbol{B}}$ in its subprotocols. It requires $O(n + (\ell d_{\boldsymbol{B}} + d_{\boldsymbol{A}})\log(\ell))$ communication and Verifier cost*

$$O((\ell n d_{\boldsymbol{B}} + n d_{\boldsymbol{A}})\log(\ell) + mn d_{\boldsymbol{A}}).$$

If $\mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A}) \subseteq \mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{B})$, there is a Las Vegas randomized algorithm for the Prover with expected cost

$$\widetilde{O}\big(\ell n r_{\boldsymbol{B}}^{\omega-2} d_{\boldsymbol{B}} + r_{\boldsymbol{B}}^{\omega-1} d_{\boldsymbol{A}} + mn d_{\boldsymbol{A}}\big).$$

Otherwise, the probability that the Verifier incorrectly accepts is at most

$$\frac{4 r_{\boldsymbol{B}} d_{\boldsymbol{B}} + d_{\boldsymbol{A}} + 2}{\#S}.$$

Proof. The verifier may incorrectly accept if either

$$\boldsymbol{\lambda} \in R = \left\{ \boldsymbol{\lambda} \in \mathsf{F}^{1\times r} \mid \boldsymbol{\lambda}\boldsymbol{A} \in \mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{B}) \right\}$$

which happens with probability $\leqslant \frac{1}{\#\mathsf{S}}$ by Lemma 5.1, or the sub-protocol RowSpaceMembership has incorrectly accepted. From Theorem 4.10, and the union bound, we obtain the claimed probability bound. □

Repeating this check in both directions proves that two matrices have the same row space.

Protocol 13: RowSpaceEquality

Public: $\boldsymbol{A} \in \mathsf{F}[x]^{m\times n}$, $\boldsymbol{B} \in \mathsf{F}[x]^{\ell\times n}$
Certifies: $\mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A}) = \mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{B})$

	Prover	Verifier
1.	Check $\mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A}) \overset{?}{\subseteq} \mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{B})$ using RowSpaceSubset	
2.	Check $\mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{B}) \overset{?}{\subseteq} \mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A})$ using RowSpaceSubset	

Theorem 5.3. *Let $\#S \geqslant 2\max(md_{\boldsymbol{A}}, \ell d_{\boldsymbol{B}})$. Let $r = \max(r_{\boldsymbol{A}}, r_{\boldsymbol{B}})$ and $d = \max(d_{\boldsymbol{A}}, d_{\boldsymbol{B}})$. Then, Protocol 13 is a complete and probabilistically sound interactive protocol which requires*

$$O((m\log(m) + \ell\log(\ell))d + n) \subset \widetilde{O}(md + \ell d + n)$$

communication and Verifier cost

$$O((m\log(m) + \ell\log(\ell))nd) \subset \widetilde{O}(mnd + \ell nd).$$

If $\mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A}) = \mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{B})$, there is a Las Vegas randomized algorithm for the Prover with expected cost $\widetilde{O}\big((m+\ell)nr^{\omega-2}d\big)$. Otherwise, the probability that the Verifier incorrectly accepts is at most $(4rd + d + 2)/\#S$.

Protocol 14: RowBasis

Public: $\boldsymbol{A} \in \mathsf{F}[x]^{m\times n}$, $\boldsymbol{B} \in \mathsf{F}[x]^{\ell\times n}$
Certifies: $\boldsymbol{B}$ is a basis of the row space of $\boldsymbol{A}$

	Prover	Verifier
1.	Check $\mathrm{rank}(\boldsymbol{B}) \overset{?}{\geqslant} \ell$ using RankLowerBound	
2.	Check $\mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{B}) \overset{?}{=} \mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A})$ using RowSpaceEquality	

Corollary 5.4. *Let $\#S \geqslant 2\max(r_{\boldsymbol{A}} d_{\boldsymbol{A}}, md_{\boldsymbol{B}})$. Let $r = \max(r_{\boldsymbol{A}}, r_{\boldsymbol{B}})$ and $d = \max(d_{\boldsymbol{A}}, d_{\boldsymbol{B}})$. Then, Protocol 14 is a complete and probabilistically sound interactive protocol which requires*

$$O((m\log(m) + \ell\log(\ell))d + n) \subset \widetilde{O}(md + \ell d + n)$$

communication and Verifier cost

$$O((m\log(m)+\ell\log(\ell))nd) \subset \widetilde{O}(mnd+\ell nd).$$

If $\boldsymbol{B}$ *is a row basis of* $\mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A})$*, there is a Las Vegas randomized algorithm for the Prover with expected cost* $\widetilde{O}\big(mn\ell^{\omega-2}d\big)$*. Otherwise, the probability that the Verifier incorrectly accepts is at most* $(4rd+d+3)/\#S$*.*

5.2 Normal forms

Here, we give protocols for certifying *normal forms* of polynomial matrices, including the Hermite form (Hermite, 1851; MacDuffee, 1933; Newman, 1972) and the Popov form (Popov, 1972; Kailath, 1980). These forms are specific row bases with useful properties such as being triangular for the former or having minimal degrees for the latter, and being unique in the sense that a given matrix in $\mathsf{F}[x]^{m\times n}$ has exactly one row basis in Hermite (resp. Popov) form.

Roughly speaking, the Hermite form is a row echelon form that stays within the underlying ring.

Definition 5.5. *A matrix* $\boldsymbol{B}=[b_{i,j}]\in\mathsf{F}[x]^{r\times n}$ *with* $r\leqslant n$ *is in* Hermite form *if there are* pivot indices $1\leqslant k_1<\cdots<k_r\leqslant n$ *such that:*

(i) (Pivots are monic, hence nonzero)
b_{i,k_i} *is monic for all* $1\leqslant i\leqslant r$*,*

(ii) (Entries right of pivots are zero)
$b_{i,j}=0$ *for all* $i\leqslant i\leqslant r$ *and* $k_i<j\leqslant n$*,*

(iii) (Entries below pivots have smaller degree)
$\deg(b_{i',k_i})<\deg(b_{i,k_i})$ *for all* $1\leqslant i<i'\leqslant r$*.*

Each entry at row i and column k_i is called a *pivot*. Observe that these conditions guarantee $\boldsymbol{B}$ has full row rank, hence the use of the notation r for the row dimension. For a matrix $\boldsymbol{A}\in\mathsf{F}[x]^{m\times n}$, its Hermite form $\boldsymbol{B}\in\mathsf{F}[x]^{r\times n}$ is the unique row basis of $\boldsymbol{A}$ which is in Hermite form.

Protocol HermiteForm certifies that a matrix $\boldsymbol{B}\in\mathsf{F}[x]^{\ell\times n}$ is the Hermite form of $\boldsymbol{A}$. It first checks that $\boldsymbol{B}$ is in Hermite form, and then it checks that $\boldsymbol{B}$ and $\boldsymbol{A}$ have the same row space using RowSpaceEquality from Section 5.1.

Theorem 5.6. *Let* $r=\max(r_{\boldsymbol{A}},r_{\boldsymbol{B}})$ *and* $d=\max(d_{\boldsymbol{A}},d_{\boldsymbol{B}})$*. Protocol 15 is a probabilistically sound interactive protocol and is complete assuming* $\#S\geqslant 2\max(md_{\boldsymbol{A}},\ell d_{\boldsymbol{B}})$ *in its subprotocol. It requires* $O(md\log(m)+n)$ *communication and Verifier cost* $O(mnd\log(m))$*. If* $\boldsymbol{B}$ *is the Hermite form of* $\boldsymbol{A}$*, there is a Las Vegas randomized algorithm for the Prover with expected cost* $\widetilde{O}\big(mnr^{\omega-2}d\big)$*. Otherwise, the probability that the Verifier incorrectly accepts is at most* $(4rd+d+2)/\#S$*.*

Protocol 15: HermiteForm

Public: $\boldsymbol{A} \in \mathsf{F}[x]^{m\times n}$, $\boldsymbol{B} \in \mathsf{F}[x]^{\ell\times n}$
Certifies: $\boldsymbol{B}$ is the Hermite form of $\boldsymbol{A}$

	Prover	Verifier
1.		Check $\ell \overset{?}{\leqslant} m$
2.		Check that $\boldsymbol{B}$ satisfies Definition 5.5
3.	Check $\mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A}) \overset{?}{=} \mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{B})$ using RowSpaceEquality	

Proof. To check that $\boldsymbol{B}$ is in Hermite form at Step 2, the Verifier first computes the pivot indices as the index of the first nonzero on each row, then checks the degree conditions specified in Definition 5.5. (If any row is zero, $\boldsymbol{B}$ is not in Hermite form.) This is a deterministic check with complexity only $O(\ell n)$.

As discussed previously, the fact that $\boldsymbol{B}$ is in Hermite form immediately implies that it has full row rank ℓ, and hence checking the row space equality is sufficient to confirm that $\boldsymbol{B}$ is a row basis for $\boldsymbol{A}$.

The subprotocol RowSpaceEquality dominates the complexity and is also the only possibility for the Verifier to incorrectly accept when the statement is false; hence the stated costs follow directly from Theorem 5.3. □

While the Hermite form has an echelon shape, it is also common in polynomial matrix computations to resort to the Popov form, for which the pivot of a row is no longer the rightmost nonzero entry but rather the rightmost entry whose degree is maximal among the entries of that row. This form loses the echelon shape, but has the advantage of having smaller-degree entries than the Hermite form.

Here we consider the more general *shifted* forms (Van Barel and Bultheel, 1992; Beckermann, Labahn, and Villard, 2006), which encompass Hermite forms and Popov forms via the use of the following degree measure. For a given tuple $\boldsymbol{s} = (s_1, \ldots, s_n) \in \mathbb{Z}^n$, the $\boldsymbol{s}$-degree of the row vector $\boldsymbol{v} = [v_1 \ \cdots \ v_n] \in \mathsf{F}[x]^{1\times n}$ is

$$\deg_{\boldsymbol{s}}(\boldsymbol{v}) = \max(\deg(v_1) + s_1, \ldots, \deg(v_n) + s_n).$$

We use the notation $\boldsymbol{B}_{i,*}$ to denote the ith row of the matrix $\boldsymbol{B}$.

Definition 5.7. *Let $\boldsymbol{s} = (s_1, \ldots, s_n) \in \mathbb{Z}^n$. A matrix $\boldsymbol{B} = [b_{i,j}] \in \mathsf{F}[x]^{r\times n}$ with $r \leqslant n$ is in $\boldsymbol{s}$-*Popov form *if there are indices $1 \leqslant k_1 < \cdots < k_r \leqslant n$ such that,*

(i) (Pivots are monic and determine the row degree)

b_{i,k_i} *is monic and* $\deg(b_{i,k_i}) + s_{k_i} = \deg_{\boldsymbol{s}}(\boldsymbol{B}_{i,*})$ *for all* $1 \leqslant i \leqslant r$,

(ii) (Entries right of pivots do not reach the row degree)
$\deg(b_{i,j}) + s_j < \deg_{\boldsymbol{s}}(\boldsymbol{B}_{i,*})$ *for all* $1 \leqslant i \leqslant r$ *and* $k_i < j \leqslant n$,

(iii) (Entries above and below pivots have lower degree)
$\deg(b_{i',k_i}) < \deg(b_{i,k_i})$ $1 \leqslant i' \neq i \leqslant r$.

The usual Popov form corresponds to the uniform shift $\boldsymbol{s} = (0, \ldots, 0)$. Furthermore, one can verify that, specifying the shift as $\boldsymbol{s} = (nt, \ldots, 2t, t)$ for any given $t > \deg(\boldsymbol{B})$, then the Hermite form is the same as the $\boldsymbol{s}$-Popov form (Beckermann et al., 2006, Lem. 2.6).

For a matrix $\boldsymbol{A} \in \mathsf{F}[x]^{m \times n}$, there exists a unique row basis $\boldsymbol{B} \in \mathsf{F}[x]^{r \times n}$ of $\boldsymbol{A}$ which is in $\boldsymbol{s}$-Popov form (Beckermann et al., 2006, Thm. 2.7); $\boldsymbol{B}$ is called the *$\boldsymbol{s}$-Popov form of $\boldsymbol{A}$*. Generalizing Protocol 15 to this more general normal form yields Protocol 16 (although the former could be derived as a particular case of the latter for a specific shift $\boldsymbol{s}$, we preferred to write both explicitly for the sake of clarity).

Protocol 16: ShiftedPopovForm

Public: $\boldsymbol{A} \in \mathsf{F}[x]^{m \times n}$, $\boldsymbol{s} = (s_1, \ldots, s_n) \in \mathbb{Z}^n$, $\boldsymbol{B} \in \mathsf{F}[x]^{\ell \times n}$
Certifies: $\boldsymbol{B}$ is the $\boldsymbol{s}$-Popov form of $\boldsymbol{A}$

	Prover	Verifier
1.		Check $\ell \stackrel{?}{\leqslant} m$
2.		Check that $(\boldsymbol{s}, \boldsymbol{B})$ satisfies Definition 5.7
3.	Check $\mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A}) \stackrel{?}{=} \mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{B})$ using RowSpaceEquality	

The next result is identical to Theorem 5.6, in both statement and proof. The only difference in the protocol is determining the indices of each pivot column in order to confirm the conditions of $\boldsymbol{s}$-Popov form; this can be accomplished in linear time by first computing the $\boldsymbol{s}$-degree of the row and then finding the rightmost column which determines this shifted row degree.

Theorem 5.8. *Let* $r = \max(r_{\boldsymbol{A}}, r_{\boldsymbol{B}})$ *and* $d = \max(d_{\boldsymbol{A}}, d_{\boldsymbol{B}})$. *Protocol 16 is a probabilistically sound interactive protocol and is complete assuming* $\#S \geqslant 2\max(md_{\boldsymbol{A}}, \ell d_{\boldsymbol{B}})$ *in its subprotocol. It requires* $O(md\log(m) + n)$ *communication and Verifier cost* $O(mnd\log(m))$. *If* $\boldsymbol{B}$ *is the* $\boldsymbol{s}$*-Popov form of* $\boldsymbol{A}$, *there is a Las Vegas randomized algorithm for the Prover with expected cost* $\widetilde{O}\big(mnr^{\omega-2}d\big)$. *Otherwise, the probability that the Verifier incorrectly accepts is at most* $(4rd + d + 2)/\#S$.

6 Saturation and kernel bases

In this section, we use the protocols described in previous sections to design certificates for computations related to saturations and kernels of polynomial matrices.

6.1 Saturation and saturated matrices

The saturation of a matrix over a principal ideal domain is a useful tool in computations; we refer to (Bourbaki, 1972, Section II.§2.4) for a general definition of saturation. It was exploited for example in (Zhou and Labahn, 2013) where a matrix is factorized as the product of a column basis times some saturation basis, and in (Neiger et al., 2018) in order to find the location of pivots in the context of the computation of normal forms. The saturation can be computed from the Hermite form, as described in (Pernet and Stein, 2010, Section 8) for integer matrices, and alternatively it can be obtained as a left kernel basis of a right kernel basis of the matrix as we prove below (Lemma 6.3).

Definition 6.1. *The* saturation *of a matrix* $\boldsymbol{A} \in \mathsf{F}[x]^{m\times n}$ *is the* $\mathsf{F}[x]$*-module*

$$\text{Saturation}(\boldsymbol{A}) = \mathsf{F}[x]^{1\times n} \cap \text{RowSp}_{\mathsf{F}(x)}(\boldsymbol{A});$$

it contains $\text{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A})$ *and has rank* $r = \text{rank}(\boldsymbol{A})$*. A* saturation basis of $\boldsymbol{A}$ *is a matrix in* $\mathsf{F}[x]^{r\times n}$ *whose rows form a basis of the saturation of* $\boldsymbol{A}$*. A matrix is said to be* saturated *if its saturation is equal to its* $\mathsf{F}[x]$*-row space.*

Two matrices with the same saturation may have different $\mathsf{F}[x]$-row spaces. For example, the matrices

$$\begin{bmatrix} 1 & 1 \\ x^2 & x^2+x \\ x & x \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 1 & 1+x^2 \\ 0 & x^2 \end{bmatrix}$$

have the same saturation $\mathsf{F}[x]^{1\times 2}$, but the $\mathsf{F}[x]$-row space of the former matrix contains $[0\ x]$ which is not in the $\mathsf{F}[x]$-row space of the latter matrix. Remark also that all nonsingular matrices in $\mathsf{F}[x]^{n\times n}$ have saturation equal to $\mathsf{F}[x]^{1\times n}$.

The saturation is defined in terms of the $\mathsf{F}(x)$-row space of the matrix: two matrices have the same saturation if and only if they have the same $\mathsf{F}(x)$-row space. In particular, $\boldsymbol{A}$ is saturated if and only if any row basis of $\boldsymbol{A}$ is saturated. This yields the following characterization for matrices having full column rank.

Lemma 6.2. *Let* $\boldsymbol{A} \in \mathsf{F}[x]^{m\times n}$ *have full column rank. Then* $\boldsymbol{A}$ *is saturated if and only if* $\text{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A}) = \mathsf{F}[x]^{1\times n}$.

Proof. Since $\boldsymbol{A}$ has full column rank, its row bases are nonsingular $n \times n$ matrices, or equivalently, $\text{RowSp}_{\mathsf{F}(x)}(\boldsymbol{A}) = \mathsf{F}[x]^{1\times n}$. Hence the saturation of $\boldsymbol{A}$ is $\mathsf{F}[x]^{1\times n}$, and the equivalence follows by definition of being saturated. □

Thus, in this case, verifying that $\boldsymbol{A}$ is saturated boils down to verifying that $\mathsf{F}[x]^{1\times n}$ is a subset of $\mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A})$, which can be done using RowSpaceSubset.

To obtain a similar result in the case of matrices with full row rank, we will rely on the following characterization of the saturation using kernel bases.

Lemma 6.3. *Let $\boldsymbol{A} \in \mathsf{F}[x]^{m\times n}$ have rank r, and let $\boldsymbol{K} \in \mathsf{F}[x]^{n\times(n-r)}$ be a basis for the right kernel of $\boldsymbol{A}$. Then,* $\mathrm{Saturation}(\boldsymbol{A})$ *is the left kernel of $\boldsymbol{K}$. In particular, the saturation bases of $\boldsymbol{A}$ are precisely the left kernel bases of $\boldsymbol{K}$.*

Proof. Each row of $\boldsymbol{A}$ is in the left kernel of $\boldsymbol{K}$, hence so is any polynomial vector $\boldsymbol{v} \in \mathsf{F}[x]^{1\times n}$ which is an $\mathsf{F}(x)$-linear combination of rows of $\boldsymbol{A}$, that is, any $\boldsymbol{v} \in \mathrm{Saturation}(\boldsymbol{A})$.

For the other direction, it is enough to prove that each row of a given left kernel basis $\boldsymbol{B} \in \mathsf{F}[x]^{r\times n}$ of $\boldsymbol{K}$ is in $\mathrm{Saturation}(\boldsymbol{A})$. Let $\hat{\boldsymbol{A}} \in \mathsf{F}[x]^{r\times n}$ be a set of r linearly independent rows of $\boldsymbol{A}$; since these rows are in the left kernel of $\boldsymbol{K}$, we have $\hat{\boldsymbol{A}} = \boldsymbol{U}\boldsymbol{B}$ for some nonsingular $\boldsymbol{U} \in \mathsf{F}[x]^{r\times r}$. Thus each row of $\boldsymbol{B} = \boldsymbol{U}^{-1}\hat{\boldsymbol{A}}$ is an $\mathsf{F}(x)$-linear combination of rows of $\boldsymbol{A}$. □

Combining this with (Zhou and Labahn, 2013, Lemma 3.3), it follows that for any saturation basis $\boldsymbol{B} \in \mathsf{F}[x]^{r\times n}$ of $\boldsymbol{A}$ and any factorization $\boldsymbol{A} = \boldsymbol{C}\boldsymbol{B}$ with $\boldsymbol{C} \in \mathsf{F}[x]^{m\times r}$, then $\boldsymbol{C}$ is a column basis of $\boldsymbol{A}$. If $\boldsymbol{A}$ has full row rank we obtain that $\boldsymbol{C}$ is nonsingular, and that $\boldsymbol{A}$ is saturated if and only if $\boldsymbol{C}$ is unimodular, or equivalently $\mathrm{ColSp}_{\mathsf{F}[x]}(\boldsymbol{A}) = \mathsf{F}[x]^{m\times 1}$. For the sake of completeness, we now present a concise proof of this characterization (Lemma 6.5); we will need the following standard result which essentially says that any kernel basis is saturated (see for example (Giorgi and Neiger, 2018, Lemma 2.2) for a proof).

Fact 6.4. *Let $\boldsymbol{K} \in \mathsf{F}[x]^{n\times \ell}$. For any left kernel basis $\boldsymbol{B} \in \mathsf{F}[x]^{r\times n}$ of $\boldsymbol{K}$, we have* $\mathrm{ColSp}_{\mathsf{F}[x]}(\boldsymbol{B}) = \mathsf{F}[x]^{r\times 1}$.

Lemma 6.5. *Let $\boldsymbol{A} \in \mathsf{F}[x]^{m\times n}$ have full row rank. Then, $\boldsymbol{A}$ is saturated if and only if* $\mathrm{ColSp}_{\mathsf{F}[x]}(\boldsymbol{A}) = \mathsf{F}[x]^{m\times 1}$.

Proof. If $\boldsymbol{A}$ is saturated, it is a basis of its own saturation since it has full row rank. Then writing $\boldsymbol{K}$ for a right kernel basis of $\boldsymbol{A}$, by Lemma 6.3, $\boldsymbol{A}$ is a left kernel basis of $\boldsymbol{K}$. Then Fact 6.4 gives $\mathrm{ColSp}_{\mathsf{F}[x]}(\boldsymbol{A}) = \mathsf{F}[x]^{m\times 1}$.

Conversely, assume $\mathrm{ColSp}_{\mathsf{F}[x]}(\boldsymbol{A}) = \mathsf{F}[x]^{m\times 1}$. Since the row space of $\boldsymbol{A}$ is a submodule of its saturation, we have $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{B}$ where $\boldsymbol{B} \in \mathsf{F}[x]^{m\times n}$ is a saturation basis of $\boldsymbol{A}$ and $\boldsymbol{U} \in \mathsf{F}[x]^{m\times m}$ is nonsingular. By assumption, we have $\boldsymbol{A}\boldsymbol{V} = \boldsymbol{I}_m$ for some $\boldsymbol{V} \in \mathsf{F}[x]^{n\times m}$, hence $\boldsymbol{U}(\boldsymbol{B}\boldsymbol{V}) = \boldsymbol{I}_m$. Because these are all polynomial matrices, this means that $\boldsymbol{U}$ is unimodular, and $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{B}$ implies that $\boldsymbol{A}$ is saturated. □

We are now ready to state Protocol 17 for the certification that a matrix is saturated, assuming it has either full row rank or full column rank. The latter restriction is satisfied in all the applications we have in mind, including the two we present below (Section 6.2): unimodular completability and kernel basis certification. We note that, if one accepts a communication cost similar to the

size of the public matrix $\boldsymbol{A}$, then removing this assumption is easily done by making use of a row basis of $\boldsymbol{A}$.

Protocol 17: Saturated

Public: $\boldsymbol{A} \in \mathsf{F}[x]^{m\times n}$ with full rank
Certifies: $\boldsymbol{A}$ has full rank $\Rightarrow$ $\boldsymbol{A}$ is saturated

	Prover	**Verifier**
1.	**if** $m \leqslant n$: // full row rank Check $\mathsf{F}[x]^{m\times 1} \subseteq \mathrm{ColSp}_{\mathsf{F}[x]}(\boldsymbol{A})$ using RowSpaceSubset with public matrices $\boldsymbol{I}_m$ and $\boldsymbol{A}^\mathsf{T}$ **else**: // full column rank Check $\mathsf{F}[x]^{1\times n} \subseteq \mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A})$ using RowSpaceSubset with public matrices $\boldsymbol{I}_n$ and $\boldsymbol{A}$	

Theorem 6.6. *Let $d = \max(1, \deg(\boldsymbol{A}))$, $\mu = \max(m,n)$, and $\nu = \min(m,n)$. Protocol 17 is a probabilistically sound interactive protocol and is complete assuming $\#S \geqslant 2\mu d$ in its subprotocol. It requires $O(\mu d \log \mu)$ communication and Verifier cost $O(mnd \log \mu)$. Assuming that $\boldsymbol{A}$ has full rank and is saturated, there is a Las Vegas randomized algorithm for the Prover with expected cost $\widetilde{O}(\mu\nu^{\omega-1}d)$, and otherwise the probability that the Verifier incorrectly accepts is at most $(4\nu d + 2)/\#S$.*

Proof. This directly follows from Lemmas 6.2 and 6.5 and Theorem 5.2. Remark that in both cases $m \leqslant n$ and $m > n$, the protocol RowSpaceSubset is applied with public matrices $\boldsymbol{I}_\nu$ and a $\mu \times \nu$ matrix of rank at most ν and degree at most d. □

Concerning the certification of a saturation basis of $\boldsymbol{A}$, our protocol will rely on the following characterization.

Lemma 6.7. *Let $\boldsymbol{A} \in \mathsf{F}[x]^{m\times n}$. Then, a matrix $\boldsymbol{B} \in \mathsf{F}[x]^{\ell\times n}$ is a saturation basis of $\boldsymbol{A}$ if and only if the following conditions are satisfied:*

(i) $\mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A}) \subseteq \mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{B})$,

(ii) $\mathrm{rank}(\boldsymbol{B}) = \ell$ *and* $\mathrm{rank}(\boldsymbol{A}) \geqslant \ell$,

(iii) $\boldsymbol{B}$ *is saturated.*

Proof. If $\boldsymbol{B}$ is a saturation basis of $\boldsymbol{A}$, then by definition $\boldsymbol{B}$ is saturated; $\boldsymbol{B}$ has full row rank with $\ell = \mathrm{rank}(\boldsymbol{B}) = \mathrm{rank}(\boldsymbol{A})$; and $\mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{B})$ is the saturation of $\boldsymbol{A}$ and therefore contains $\mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A})$.

Conversely, assume that the three items hold. The first two items together imply both $\ell = \mathrm{rank}(\boldsymbol{B}) = \mathrm{rank}(\boldsymbol{A})$ and $\mathrm{RowSp}_{\mathsf{F}(x)}(\boldsymbol{A}) \subseteq \mathrm{RowSp}_{\mathsf{F}(x)}(\boldsymbol{B})$; hence

the latter inclusion of $\mathsf{F}(x)$-vector spaces of same dimension is an equality. Thus we have Saturation($\boldsymbol{A}$) = Saturation($\boldsymbol{B}$), and the latter saturation is equal to $\mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{B})$ since $\boldsymbol{B}$ is saturated by the third item. This concludes the proof that $\boldsymbol{B}$ has full row rank and $\mathsf{F}[x]$-row space equal to the saturation of $\boldsymbol{A}$. □

Protocol 18: SaturationBasis

Public: $\boldsymbol{A} \in \mathsf{F}[x]^{m\times n}$, $\boldsymbol{B} \in \mathsf{F}[x]^{\ell\times n}$
Certifies: $\boldsymbol{B}$ is a saturation basis of $\boldsymbol{A}$

	Prover	Verifier
1.		Check $\ell \stackrel{?}{\leqslant} \min(m,n)$
2.	Check $\mathrm{rank}(\boldsymbol{A}) \stackrel{?}{\geqslant} \ell$ using RankLowerBound	
3.	Check $\mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A}) \stackrel{?}{\subseteq} \mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{B})$ using RowSpaceSubset	
4.	Check that $\boldsymbol{B}$ is saturated using Saturated	

Theorem 6.8. *Protocol 18 is a probabilistically sound interactive protocol and is complete assuming* $\#S \geqslant \max(\ell d_{\boldsymbol{A}} + 1, 2nd_{\boldsymbol{B}})$ *in its subprotocols. It requires* $O(nd_{\boldsymbol{B}}\log(n) + d_{\boldsymbol{A}}\log(\ell))$ *communication and Verifier cost* $O(mnd_{\boldsymbol{A}} + \ell nd_{\boldsymbol{B}}\log(n))$. *If* $\boldsymbol{B}$ *is a saturation basis of* $\boldsymbol{A}$*, then there is a Las Vegas randomized algorithm for the Prover with expected cost*

$$\widetilde{O}\big(mn\ell^{\omega-2}d_{\boldsymbol{A}} + n\ell^{\omega-1}d_{\boldsymbol{B}}\big);$$

otherwise the probability that the Verifier incorrectly accepts is at most

$$\frac{4\ell d_{\boldsymbol{B}} + d_{\boldsymbol{A}} + 2}{\#S}.$$

Proof. The check on Step 1 has no arithmetic cost, but ensures that ℓ is less than or equal to m, n, rank($\boldsymbol{A}$), and rank($\boldsymbol{B}$). Then the complexities follow from Lemma 6.7 and Theorems 3.4, 5.2 and 6.6.

Note that $\mathrm{rank}(\boldsymbol{A}) \geqslant \ell$ and $\mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A}) \subseteq \mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{B})$ imply that $\mathrm{rank}(\boldsymbol{B}) = \ell$, so that the precondition for the Saturated protocol on Step 4 is valid unless one of the previous checks failed.

For the probability bound, the worst case for the Verifier is when $\mathrm{rank}(\boldsymbol{A}) \leqslant \ell$ and $\boldsymbol{B}$ is saturated, but $\mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A}) \not\subseteq \mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{B})$. Then only the statement being checked on Step 3 is false, and this sub-protocol has the greatest probability of the verifier incorrectly accepting, which matches the one in the theorem statement. □

6.2 Kernel bases and unimodular completability

Here, we derive two protocols which follow from the ones concerning the saturation. The second protocol is for the certification of kernel bases, while the first protocol is about matrices that can be completed into unimodular matrices.

The fast computation of such completions was studied by Zhou and Labahn (2014). We say that $\boldsymbol{A} \in \mathsf{F}[x]^{m\times n}$ is *unimodular completable* if $m < n$ and there exists a matrix $\boldsymbol{B} \in \mathsf{F}[x]^{m\times(m-n)}$ such that $[\boldsymbol{A}^\mathsf{T} \;\; \boldsymbol{B}^\mathsf{T}]^\mathsf{T}$ is unimodular. Note that if $\boldsymbol{A}$ does not have full row rank, then it is not unimodular completable. Otherwise, Zhou and Labahn (2014, Lemma 2.10) showed that $\boldsymbol{A}$ is unimodular completable if and only if $\boldsymbol{A}$ has unimodular column bases; by Lemma 6.5, this holds if and only if $\boldsymbol{A}$ is saturated. This readily leads us to Protocol 19.

Protocol 19: UnimodularCompletable

Public: $\boldsymbol{A} \in \mathsf{F}[x]^{m\times n}$
Certifies: $\boldsymbol{A}$ is unimodular completable

	Prover	Verifier
1.		Check $m \overset{?}{<} n$
2.	Check $\operatorname{rank}(\boldsymbol{A}) \overset{?}{\geqslant} m$ using RankLowerBound	
3.	Check that $\boldsymbol{A}$ is saturated using Saturated	

Theorem 6.9. *Protocol 19 is a probabilistically sound interactive protocol and is complete assuming $\#S \geqslant 2md$ in its subprotocols. It requires $O(nd\log(n))$ communication and Verifier cost $O(mnd\log(n))$. If $\boldsymbol{A}$ is unimodular completable, then there is a Las Vegas randomized algorithm for the Prover with expected cost $\widetilde{O}(nm^{\omega-1}d)$; otherwise the probability that the Verifier incorrectly accepts is at most $(4md+2)/\#S$.*

Proof. The costs follow from Theorems 3.4 and 6.6, noting that the protocol aborts early if $m \geqslant n$, and therefore m is an upper bound on the rank in both sub-protocols. The worst case for the Verifier is that $\operatorname{rank}(\boldsymbol{A}) \geqslant m$ but $\boldsymbol{A}$ is not saturated; then only the second statement is incorrect, which has a higher probability of the Verifier incorrectly accepting in that subprotocol. Therefore the probability of the Verifier incorrectly accepting here is the same as in Saturated from Theorem 6.6. □

Finally, Protocol 20 for the certification of kernel bases will follow from the characterization in the next lemma.

Lemma 6.10. *Let $\boldsymbol{A} \in \mathsf{F}[x]^{m\times n}$ and let $\boldsymbol{B} \in \mathsf{F}[x]^{\ell\times m}$. Then, $\boldsymbol{B}$ is a left kernel basis of $\boldsymbol{A}$ if and only if*

(i) $\operatorname{rank}(\boldsymbol{B}) = \ell$ *and* $\operatorname{rank}(\boldsymbol{A}) \geqslant m - \ell$,

(ii) $\boldsymbol{B}\boldsymbol{A} = \boldsymbol{0}$,

(iii) $\boldsymbol{B}$ *is saturated.*

Proof. If $\boldsymbol{B}$ is a left kernel basis of $\boldsymbol{A}$, then we have $\operatorname{rank}(\boldsymbol{B}) = \ell = m - \operatorname{rank}(\boldsymbol{A})$ as well as $\boldsymbol{B}\boldsymbol{A} = \boldsymbol{0}$; the third item follows from Fact 6.4 and Lemma 6.5.

Now assume that the three items hold. Consider some left kernel basis $\boldsymbol{K}$ of $\boldsymbol{A}$. Then, $\operatorname{rank}(\boldsymbol{K}) = m - \operatorname{rank}(\boldsymbol{A}) \leqslant \ell$ by the first item, while the second item implies that the row space of $\boldsymbol{B}$ is contained in the row space of $\boldsymbol{K}$, hence $\ell = \operatorname{rank}(\boldsymbol{B}) \leqslant \operatorname{rank}(\boldsymbol{K})$; therefore $\operatorname{rank}(\boldsymbol{K}) = \ell$. As a result, $\boldsymbol{B} = \boldsymbol{U}\boldsymbol{K}$ for some nonsingular $\boldsymbol{U} \in \mathsf{F}[x]^{\ell \times \ell}$. Item *(iii)* implies $\operatorname{ColSp}_{\mathsf{F}[x]}(\boldsymbol{B}) = \mathsf{F}[x]^{\ell \times 1}$ according to Lemma 6.5, hence $\boldsymbol{I}_\ell = \boldsymbol{B}\boldsymbol{V} = \boldsymbol{U}\boldsymbol{K}\boldsymbol{V}$ for some $\boldsymbol{V} \in \mathsf{F}[x]^{m \times \ell}$. Then, $\boldsymbol{U}$ must be unimodular, and thus $\boldsymbol{B} = \boldsymbol{U}\boldsymbol{K}$ is a left kernel basis of $\boldsymbol{A}$. □

Protocol 20: KernelBasis

Public: $\boldsymbol{A} \in \mathsf{F}[x]^{m \times n}$, $\boldsymbol{B} \in \mathsf{F}[x]^{\ell \times m}$
Certifies: $\boldsymbol{B}$ is a left kernel basis of $\boldsymbol{A}$

	Prover	Verifier
1.		Check $\ell \stackrel{?}{\leqslant} m$
2.	Check $\operatorname{rank}(\boldsymbol{B}) \stackrel{?}{\geqslant} \ell$ using RankLowerBound	
3.	Check $\operatorname{rank}(\boldsymbol{A}) \stackrel{?}{\geqslant} m - \ell$ using RankLowerBound	
4.	Check $\boldsymbol{B}\boldsymbol{A} \stackrel{?}{=} \boldsymbol{0}$ using MatMul	
5.	Check that $\boldsymbol{B}$ is saturated using Saturated	

Theorem 6.11. *Protocol 20 is a probabilistically sound interactive protocol and is complete assuming* $\#S \geqslant \max((m-\ell)d_{\boldsymbol{A}} + 1, 2md_{\boldsymbol{B}})$ *in its subprotocols. It requires* $O(md_{\boldsymbol{B}} \log(m))$ *communication and Verifier cost*

$$O(\ell m d_{\boldsymbol{B}} \log(m) + mnd_{\boldsymbol{A}}).$$

If $\boldsymbol{B}$ *is a left kernel basis of* $\boldsymbol{A}$*, then there is a Las Vegas randomized algorithm for the Prover with expected cost*

$$\widetilde{O}\big(m\ell^{\omega-1}d_{\boldsymbol{B}} + mn(m-\ell)^{\omega-2}d_{\boldsymbol{A}}\big);$$

otherwise the probability that the Verifier incorrectly accepts is at most

$$\frac{\max(d_{\boldsymbol{A}} + d_{\boldsymbol{B}} + 1, 4\ell d_{\boldsymbol{B}} + 2)}{\#S}.$$

Proof. The costs follow from Lemma 6.10 and Theorems 3.4, 3.9 and 6.6. As before, the worst case for the Verifier is that only one of the four checked statements is wrong, and the resulting maximum of probabilities comes either from Step 3 or Step 5. □

7 Conclusion and perspectives

We have developed interactive certificates for a variety of problems concerning polynomial matrices. For rank, determinant, system solving, and matrix multiplication (Section 3), these amount to evaluating at some random point(s) and reducing to field-based verifications. For row bases, saturation, normal forms, and kernel basis computations (Sections 5 and 6), the verifications essentially reduce to testing row space membership of a single vector (Section 4) and testing that ranks are the expected ones.

Our protocols are efficient. The volume of data exchanged in communications is roughly the size of a single row of the matrix. The time complexity for the Verifier is linear (or nearly-linear) in the size of the object being checked, and the time for the Prover is roughly the same as it would take to perform the computation being verified.

Still, there is some room for improvement in these costs. It would be nice to remove the logarithmic factors in the complexities of most later protocols for the Verifier time and communication cost; these come from the number of repetitions t required in the RowSpaceMembership protocol.

Another possibility for improvement in our complexities would be to have the same costs where d is the *average* matrix-vector degree, rather than the maximum degree. Such complexity refinements have appeared for related computational algorithms, frequently by "partial linearization" of the rows or columns with highest degree (Gupta et al., 2012, Section 6), and it would be interesting to see if similar techniques could work here. This would be especially helpful in more efficiently verifying an unbalanced shifted Popov form, and the Hermite form in particular, of a nonsingular matrix.

While we have presented protocols for a variety of basic problems on polynomial matrices, there are still more for which we do not know yet whether any efficient verification exists. These include:

- matrix division with remainder (see (Gantmacher, 1959, Section IV.§2) and (Kailath, 1980, Theorem 6.3-15));
- matrix inversion (the current fastest algorithm is by Zhou, Labahn, and Storjohann (2015));
- high-order terms in expansion of the inverse (see the high-order lifting algorithm of Storjohann (2003));
- univariate relations, generalizing Hermite-Padé approximation (Beckermann and Labahn, 2000; Neiger and Vu, 2017); and

- Smith form (see (Storjohann, 2003) for the fastest known algorithm).

Perhaps the most interesting direction for future work would be to adapt our protocols to the case of Euclidean lattices, i.e., integer matrices and vectors. It seems that most of our protocols in Section 3 should translate when we replace evaluation at a point α with reduction modulo a sufficiently-large prime p, but the analysis in terms of bit complexity rather than field operations will likely be more delicate. Another seeming hurdle is in our central protocols in Section 4 on deciding row membership: while the general ideas of these protocols *might* translate to integer lattices, the proof techniques we have used are particular for polynomials.

Acknowledgements

This work was performed while the fourth author was generously hosted by the Laboratoire Jean Kuntzmann in Grenoble.

This work was partially supported by the U.S. National Science Foundation under award #1618269, by the OpenDreamKit Horizon 2020 European Research Infrastructures project under award #676541 and by the IFD-Science 2017 research program of the Institut Français du Danemark.

References

B. Beckermann and G. Labahn. A uniform approach for the fast computation of matrix-type Padé approximants. *SIAM J. Matrix Anal. Appl.*, 15(3):804–823, July 1994. ISSN 0895-4798. doi: 10.1137/S0895479892230031.

B. Beckermann and G. Labahn. Fraction-free computation of matrix rational interpolants and matrix gcds. *SIAM J. Matrix Anal. Appl.*, 22(1):114–144, 2000. doi: 10.1137/S0895479897326912.

B. Beckermann, G. Labahn, and G. Villard. Normal forms for general polynomial matrices. *J. Symbolic Comput.*, 41(6):708–737, 2006. ISSN 0747-7171. doi: 10.1016/j.jsc.2006.02.001.

D. A. Bini and V. Pan. *Polynomial and Matrix Computations*. Progress in Theoretical Computer Science. Birkhäuser Basel, 1994. doi: 10.1007/978-1-4612-0265-3.

A. Bostan and É. Schost. Polynomial evaluation and interpolation on special sets of points. *J. Complexity*, 21(4):420–446, 2005. ISSN 0885-064X. doi: 10.1016/j.jco.2004.09.009.

N. Bourbaki. *Commutative Algebra*. Elements of Mathematics. Addison-Wesley, 1972.

D. G. Cantor and E. Kaltofen. On fast multiplication of polynomials over arbitrary algebras. *Acta Inform.*, 28(7):693–701, 1991. ISSN 0001-5903. doi: 10.1007/BF01178683.

D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions. *J. Symbolic Comput.*, 9(3):251–280, 1990. ISSN 0747-7171. doi: 10.1016/S0747-7171(08)80013-2.

C. Costello, C. Fournet, J. Howell, M. Kohlweiss, B. Kreuter, M. Naehrig, B. Parno, and S. Zahur. Geppetto: Versatile verifiable computation. In *2015 IEEE Symposium on Security and Privacy*, pages 253–270, 2015. doi: 10.1109/SP.2015.23.

R. A. DeMillo and R. J. Lipton. A probabilistic remark on algebraic program testing. *Inform. Process. Lett.*, 7(4):193–195, 1978. doi: 10.1016/0020-0190(78)90067-4.

J. D. Dixon. Exact solution of linear equations usingp-adic expansions. *Numerische Mathematik*, 40(1):137–141, Feb 1982. ISSN 0945-3245. doi: 10.1007/BF01459082.

J.-G. Dumas and E. Kaltofen. Essentially optimal interactive certificates in linear algebra. In *ISSAC '14*, pages 146–153, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2501-1. doi: 10.1145/2608628.2608644.

J.-G. Dumas, E. Kaltofen, E. Thomé, and G. Villard. Linear time interactive certificates for the minimal polynomial and the determinant of a sparse matrix. In *ISSAC '16*, pages 199–206, New York, NY, USA, 2016. ACM. doi: 10.1145/2930889.2930908.

J.-G. Dumas, D. Lucas, and C. Pernet. Certificates for triangular equivalence and rank profiles. In *ISSAC '17*, pages 133–140. ACM, 2017. doi: 10.1145/3087604.3087609.

A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO '86*, pages 186–194. Springer Berlin Heidelberg, 1987. doi: 10.1007/3-540-47721-7_12.

Rūsiņš Freivalds. Fast probabilistic algorithms. In *Mathematical Foundations of Computer Science 1979*, pages 57–69. Springer Berlin Heidelberg, 1979. doi: 10.1007/3-540-09526-8_5.

F. R. Gantmacher. *The Theory of Matrices*. Chelsea, 1959.

J. von zur Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge University Press, Cambridge, second edition, 2003.

P. Giorgi and V. Neiger. Certification of minimal approximant bases. In *ISSAC'18*, 2018. doi: 10.1145/3208976.3208991. URL https://hal-unilim.archives-ouvertes.fr/hal-01701861. (In press).

P. Giorgi, C.-P. Jeannerod, and G. Villard. On the complexity of polynomial matrix computations. In *ISSAC'03*, pages 135–142. ACM, 2003. doi: 10.1145/860854.860889.

S. Goldwasser, Y. T. Kalai, and G. N. Rothblum. Delegating computation: Interactive proofs for muggles. In *STOC '08*, pages 113–122, New York, NY, USA, 2008. ACM. doi: 10.1145/1374376.1374396.

S. Gupta, S. Sarkar, A. Storjohann, and J. Valeriote. Triangular x-basis decompositions and derandomization of linear algebra algorithms over $K[x]$. *J. Symbolic Comput.*, 47(4):422–453, 2012. doi: 10.1016/j.jsc.2011.09.006.

C. Hermite. Sur l'introduction des variables continues dans la théorie des nombres. *Journal für die reine und angewandte Mathematik*, 41:191–216, 1851.

C.-P. Jeannerod, C. Pernet, and A. Storjohann. Rank-profile revealing gaussian elimination and the cup matrix decomposition. *J. Symbolic Comput.*, 56: 46–68, 2013. doi: 10.1016/j.jsc.2013.04.004.

T. Kailath. *Linear Systems*. Prentice-Hall, 1980.

E. L. Kaltofen, M. Nehring, and B. D. Saunders. Quadratic-time certificates in linear algebra. In *ISSAC '11*, pages 171–176, New York, NY, USA, 2011. ACM. doi: 10.1145/1993886.1993915.

E. L. Kaltofen, B. Li, Z. Yang, and L. Zhi. Exact certification in global polynomial optimization via sums-of-squares of rational functions with rational coefficients. *J. Symbolic Comput.*, 47(1):1–15, 2012. doi: 10.1016/j.jsc.2011.08.002.

G. Labahn, V. Neiger, and W. Zhou. Fast, deterministic computation of the Hermite normal form and determinant of a polynomial matrix. *J. Complexity*, 42:44–71, 2017. doi: 10.1016/j.jco.2017.03.003.

F. Le Gall. Powers of tensors and fast matrix multiplication. In *ISSAC'14*, pages 296–303. ACM, 2014. doi: 10.1145/2608628.2608664.

C. C. MacDuffee. *The Theory of Matrices*. Springer-Verlag Berlin Heidelberg, 1933. doi: 10.1007/978-3-642-99234-6.

T. Mulders and A. Storjohann. Certified dense linear system solving. *Journal of Symbolic Computation*, 37(4):485–510, 2004. doi: 10.1016/j.jsc.2003.07.004.

V. Neiger and T. X. Vu. Computing canonical bases of modules of univariate relations. In *ISSAC'17*, pages 357–364. ACM, 2017. doi: 10.1145/3087604.3087656.

V. Neiger, J. Rosenkilde, and G. Solomatov. Computing Popov and Hermite forms of rectangular polynomial matrices. In *ISSAC '18*. ACM, 2018. doi: 10.1145/3208976.3208988.

M. Newman. *Integral Matrices*. Number v. 45 in Integral matrices. Academic Press, 1972.

C. Pernet and W. Stein. Fast computation of Hermite normal forms of random integer matrices. *Journal of Number Theory*, 130(7):1675–1683, 2010. doi: 10.1016/j.jnt.2010.01.017.

V. M. Popov. Invariant description of linear, time-invariant controllable systems. *SIAM Journal on Control*, 10(2):252–264, 1972. doi: 10.1137/0310020.

J. T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. ACM*, 27(4):701–717, 1980. doi: 10.1145/322217.322225.

A. Storjohann. High-order lifting and integrality certification. *J. Symbolic Comput.*, 36(3-4):613–648, 2003. doi: 10.1016/S0747-7171(03)00097-X.

A. Storjohann and G. Villard. Computing the rank and a small nullspace basis of a polynomial matrix. In *ISSAC '05*, pages 309–316, New York, NY, USA, 2005. ACM. doi: 10.1145/1073884.1073927.

M. Van Barel and A. Bultheel. A general module theoretic framework for vector M-Padé and matrix rational interpolation. *Numer. Algorithms*, 3:451–462, 1992. doi: 10.1007/BF02141952.

G. Villard. Computing Popov and Hermite forms of polynomial matrices. In *ISSAC'96*, pages 250–258. ACM, 1996. doi: 10.1145/236869.237082.

W. Zhou. *Fast Order Basis and Kernel Basis Computation and Related Problems*. PhD thesis, University of Waterloo, 2012. URL http://hdl.handle.net/10012/7326.

W. Zhou and G. Labahn. Computing column bases of polynomial matrices. In *ISSAC'13*, pages 379–386, New York, NY, USA, 2013. ACM. doi: 10.1145/2465506.2465947.

W. Zhou and G. Labahn. Unimodular completion of polynomial matrices. In *ISSAC'14*, pages 413–420. ACM, 2014. doi: 10.1145/2608628.2608640.

W. Zhou, G. Labahn, and A. Storjohann. A deterministic algorithm for inverting a polynomial matrix. *J. Complexity*, 31(2):162–173, 2015. doi: 10.1016/j.jco.2014.09.004.

R. Zippel. Probabilistic algorithms for sparse polynomials. In *EUROSAM'79*, volume 72 of *LNCS*, pages 216–226. Springer, 1979. doi: 10.1007/3-540-09519-5_73.