Verification Protocols with Sub-Linear Communication for Polynomial Matrix Operations

David Lucas, Vincent Neiger, Clément Pernet, Daniel S. Roche, Johan Rosenkilde

Verification Protocols with Sub-Linear Communication for Polynomial Matrix Operations

David Lucas[a], Vincent Neiger[b], Clément Pernet[a], Daniel S. Roche[c,*], Johan Rosenkilde[d]

[a] *Univ. Grenoble Alpes, CNRS, Grenoble INP, LJK, 38000 Grenoble, France*
[b] *Univ. Limoges, CNRS, XLIM, UMR 7252, F-87000 Limoges, France*
[c] *United States Naval Academy, Annapolis, Maryland, U.S.A.*
[d] *Technical University of Denmark, Kgs. Lyngby, Denmark*

Abstract

We design and analyze new protocols to verify the correctness of various computations on matrices over the ring $\mathsf{F}[x]$ of univariate polynomials over a field F. For the sake of efficiency, and because many of the properties we verify are specific to matrices over a principal ideal domain, we cannot simply rely on previously-developed linear algebra protocols for matrices over a field. Our protocols are *interactive*, 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 verification protocols and a new protocol to verify that a given vector is in the $\mathsf{F}[x]$-row space 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 *protocols* certifying the correctness of some result, which can be verified much more efficiently than re-computing the result itself.

1.1. Verification protocols

In this paper, we propose new *verification protocols* for computations performed on univariate polynomial matrices; we refer to (Dumas, 2018; Dumas and Kaltofen, 2014; Kaltofen et al., 2011, 2012) for definitions related to such protocols. Generically, we consider protocols where a Prover performs computations and provides additional data structures to or exchanges with a

*Corresponding author
Email addresses: David.Lucas@univ-grenoble-alpes.fr (David Lucas), vincent.neiger@unilim.fr (Vincent Neiger), Clement.Pernet@univ-grenoble-alpes.fr (Clément Pernet), roche@usna.edu (Daniel S. Roche), jsrn@dtu.dk (Johan Rosenkilde)

Verifier, who will use these to check the validity of a result, at a lower cost than by recomputing it. From the viewpoint of the theoretical computer science community, this corresponds to instances of interactive proof protocols (Goldwasser et al., 1989, 2008). In the context of certified compter algebra, the term interactive proof of work certificate was introduced by Dumas (2018). In this paper, we will simply refer to these protocols as verification protocols, as they do not necessarily verify the work of the computation but only the result (which can be obtained by any means).

The general flow of a verification protocol is as follows.

1. At the beginning, the Prover and the Verifier share the knowledge of the result of a computation, which they have to verify.
2. The Verifier sends a Challenge to the Prover, 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 protocols can be simulated non-interactively in a single round following the heuristic derandomization of 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 a verification protocol, namely

Communication cost: 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 there is a small probability 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 statement is false and/or the Prover did something wrong; the Verifier or unlucky randomness are never to blame.

Several approaches to verified computation exist: generic approaches based on protocol check circuits (Goldwasser et al., 2008) or on homomorphic encryption (Costello et al., 2015);

and approaches working for any algorithm where the Prover uses specific operations, such as that of Kaltofen et al. (2011, Section 5) which certifies any protocol where matrix multiplications are performed. Another approach consists in designing problem-specific verification protocols, as was done for instance in (Freivalds, 1979; Kaltofen et al., 2011; Dumas et al., 2017) on dense linear algebra and (Dumas and Kaltofen, 2014; Dumas et al., 2016) on sparse linear algebra.

1.2. Polynomial matrices

This paper deals with 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 (Dumas, 2018; Giorgi and Neiger, 2018).

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 operations such as 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 et al., 2018) and that of computing modules of relations such as approximant bases (Beckermann and Labahn, 1994; Giorgi et al., 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 only with evaluations of the matrix at points $\alpha \in \mathsf{F}$ is oblivious to this module structure.

1.3. Our contributions

In this paper, after giving some preliminary material in Section 2, we propose verification protocols for classical properties of polynomial matrices — singularity, rank, determinant and matrix product — with sub-linear communication cost with respect to the input size (Section 3). Those protocols are based on evaluating considered matrices at random points, which allows us to reduce the communication space and to use existing verification protocols 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 verification protocols for some classical normal forms of polynomial matrices.

In Section 6, we present verification protocols 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.

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 $\omega \leq 3$ 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.

The last column indicates the smallest size of the ground field F needed to ensure both perfect completeness of the protocol and soundness with probability at least $\frac{1}{2}$. If this lower bound is not met, an extension field may be agreed on in advance by the Prover and Verifier, for a (logarithmic) increase in arithmetic and communication costs. For all protocols, an arbitrary low probability p of failure can be achieved by simply iterating the protocol at most $\lceil \log_2(1/p) \rceil$ times.

2. Preliminaries

2.1. Notation and assumptions

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 $\mathsf{F}[x]^{m\times n}$.

Asymptotic complexity bounds. Throughout the paper, the cost bounds are worst-case deterministic unless otherwise indicated. We use the "soft-oh" notation $\widetilde{O}(\cdot)$ to give asymptotic bounds hiding logarithmic factors. Precisely, for two cost functions f, g, having $f \in \widetilde{O}(g)$ means that $f \in O(g \log(g)^c)$ for some constant $c > 0$.

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 $2 \leq \omega \leq 3$ and one may take $\omega < 2.373$ (Coppersmith and Winograd, 1990; Le Gall, 2014).

Cantor and Kaltofen (1991) have shown that multiplying two univariate polynomials of degree $\leq d$ 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.

Sampling set. In our protocols, S is always a finite subset of the base field F which the Verifier uses to sample field elements uniformly and independently at random. We denote by

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

respectively 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 and probabilistically sound, our protocols require lower bounds on the cardinality $\#\mathsf{S}$ of this subset, and therefore of the field F itself. Generally speaking, choosing S larger will increase the soundness probability, at the cost of higher randomness complexity. In particular, one may use $\mathsf{S} = \mathsf{F}$ if the field F is finite and sufficiently large. If

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

Table 1: Summary of the contributions. The first column states whether the Prover's algorithm is deterministic or not. The costs are given in number of arithmetic operations over the base field and the communication is in number of elements in the base field F. The last column reports the minimum size size of F needed to ensure perfect completeness and soundness with probability at least $\frac{1}{2}$.

#F is too small, then one may use a field extension, causing up to a logarithmic factor increase in the Prover/Verifier/communication costs.

Protocols. Many of our analyses and protocols use the notation

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

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

Following (Dumas and Kaltofen, 2014; Dumas et al., 2016, 2017) we use the notation $x \stackrel{?}{=} y$ as a placeholder for

If $x \neq y$ **then** abort and report failure

to improve the brevity and readability of the protocols. Similarly, we use $x \stackrel{?}{<} y, x \stackrel{?}{\geq} y$, $U \stackrel{?}{\subseteq} V$, etc. to check inequalities and set inclusion.

In all our protocols, we implicitly assume that the Verifier always checks whether the data received has the correct type and size, and aborts immediately if such checks fail. For instance, in Protocol 3, the Verifier implicitly checks that the subsets I and J are actually subsets of ρ distinct integers each, as specified by the protocol, but explicitly checks they are subsets of $\{1, \ldots, m\}$ and $\{1, \ldots, n\}$ respectively.

Threat model. As stated before, our protocols are perfectly complete and probabilistically sound, meaning that (1) if the statement is true and the Prover is honest, then an honest Verifier will always accept the statement; and (2) if the statement is not true, then an honest Verifier will reject the statement with probability at least $\frac{1}{2}$.

In practice, this chance of incorrectly accepting a false statement can be made arbitrarily low by iterating the protocol; namely, the probability of wrongly accepting a false statement is less than $2^{-\lambda}$ if the protocol succeeds in every one of λ iterations. Note that these iterations can always be performed in parallel, so that there is a linear scaling in the communication, Verifier, and Prover costs, but not in the number of rounds in the protocols.

The probabilistic soundness holds whenever the Verifier is honest; the Prover may be malicious and deviate from the stated protocol. Our security depends only on the random challenges chosen by the Verifier and various algebraic properties; we do not make any cryptographic assumptions or impose limits on the computational power of the Prover.

The cost of an honest Verifier depends only on the public information — that is, the parameters of the statement being verified — and even a malicious Prover cannot cause the Verifier to do more work than this. However, there is an issue here in the case of very large (or even infinite) fields F, where the Prover may essentially execute a denial of service attack by sending arbitrary large field elements in the protocol. Because our analysis is generic in terms of field operations, it is not able to capture this weakness, and we do not attempt to address it.

Rational fractions. For a rational fraction $f \in \mathsf{F}(x)$, define its *denominator* $\operatorname{denom}(f)$ 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* $\operatorname{numer}(f) = f \cdot \operatorname{denom}(f)$. Note that $\operatorname{denom}(a) = 1$ 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 $\operatorname{denom}(\boldsymbol{A})$ 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 $\operatorname{numer}(\boldsymbol{A})$. Note that we have the identity $\operatorname{denom}(\boldsymbol{A}) = \operatorname{lcm}_{i,j}(\operatorname{denom}(\boldsymbol{A}_{i,j}))$.

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

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

and its left kernel

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

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 *rational* row space or $\mathsf{F}(x)$-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 they have an inverse (with entries in $\mathsf{F}[x]$).

2.2. *Some probability bounds.*

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 often referred to as the *DeMillo-Lipton-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.

The following consequence is a standard extension of the soundness proof of Freivalds' algorithm (1979).

Lemma 2.1. *Let $\boldsymbol{A} \in \mathsf{F}^{m\times n}$ be a matrix with at least one nonzero entry and let $\mathsf{S} \subseteq \mathsf{F}$ be a finite subset. For a vector of scalars $\boldsymbol{w} \in \mathsf{S}^{n\times 1}$ chosen uniformly at random, we have* $\Pr[\boldsymbol{Aw} = \boldsymbol{0}] \leq 1/\#\mathsf{S}$.

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

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

Lemma 2.2. *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* $\mathrm{rank}(\boldsymbol{A}(\alpha)) < r$ *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 $\mathrm{rank}(\boldsymbol{A}) \geq r$. On the other hand, $\mathrm{rank}(\boldsymbol{A}(\alpha)) < r$ if and only if α is a root of all such minors. □

3. Linear algebra operations

In this section, we give some verification protocols for the computation of classical linear algebra properties on polynomial matrices: singularity, rank, and determinant of a matrix, as well as system solving and matrix multiplication.

The protocols 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 protocols, 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 et al., 2013, Algorithm 6).

3.1. Singularity and nonsingularity

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 cost, 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.			$\alpha \xleftarrow{\$} \mathsf{S}$
		$\xleftarrow{\quad\alpha\quad}$	
2.	Find $\boldsymbol{v} \in \mathsf{F}^{1\times n} \setminus \{\boldsymbol{0}\}$ s.t. $\boldsymbol{v}\boldsymbol{A}(\alpha) = \boldsymbol{0}$		
		$\xrightarrow{\quad\boldsymbol{v}\quad}$	
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 3.1. *Protocol 1 is a complete and probabilistically sound interactive protocol which requires $O(n)$ communication and has 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 with cost $O(n^2 d + nr^{\omega-1})$.*

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 2.2.

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 verification protocol 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 protocol verifying 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 $\alpha \in \mathsf{F}$ s.t. $\det(\boldsymbol{A}(\alpha)) \neq 0$		
		$\alpha \longrightarrow$	
2.			$\boldsymbol{b} \xleftarrow{\$} \mathsf{S}^{n\times 1}$
		$\longleftarrow \boldsymbol{b}$	
3.	Find $\boldsymbol{w} \in \mathsf{F}^{n\times 1}$ s.t. $\boldsymbol{A}(\alpha)\boldsymbol{w} = \boldsymbol{b}$		
		$\boldsymbol{w} \longrightarrow$	
4.			$\boldsymbol{A}(\alpha)\boldsymbol{w} \stackrel{?}{=} \boldsymbol{b}$

Theorem 3.2. *Protocol 2 is a probabilistically sound interactive protocol and is complete assuming that $\#\mathsf{S} \geq nd + 1$. It requires $O(n)$ communication and has Verifier cost $O(n^2 d)$. The*

probability that the Verifier incorrectly accepts is at most $1/\#\mathsf{S}$*. There is an 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 verification protocol, 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 $\det(\boldsymbol{A}) \in \mathsf{F}[x]$ using the deterministic algorithm of Labahn et al. (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, 2013, Corollary 10.8); since $\deg(\det(\boldsymbol{A})) \leq nd$, 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 $O(n^2 d)$. Finally, total communications are two vectors of size n over the base field and a scalar, hence the cost of $O(n)$. □

3.2. Matrix Rank

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

Protocol 3: RankLowerBound

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

	Prover		**Verifier**
1.	Find index sets I, J such that $\#I = \rho$, $\#J = \rho$, and $\boldsymbol{A}_{I,J}$ is nonsingular		
		$\xrightarrow{I, J}$	
2.			$I \overset{?}{\subseteq} \{1, \ldots, m\}$ $J \overset{?}{\subseteq} \{1, \ldots, n\}$
3.		Use NonSingularity($\boldsymbol{A}_{I,J}$)	

Theorem 3.3. *Let r be the actual rank of $\boldsymbol{A}$. Protocol 3 is a probabilistically sound interactive protocol and is complete assuming $\#\mathsf{S} \geq \rho d + 1$ in its subprotocol. It requires $O(\rho)$ communication and has Verifier cost $O(\rho^2 d)$. If we indeed have $r \geq \rho$, then there is a Las Vegas randomized algorithm for the Prover with expected cost $O(mnr^{\omega-2} + mnd)$. 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 $I \subseteq \{1, \ldots, m\}$ and $J \subseteq \{1, \ldots, n\}$ of size ρ such that $\boldsymbol{A}_{I,J}$ is nonsingular, and since NonSingularity is complete, so is this protocol. Note that the completeness of the sub-protocol is ensured only if $\#\mathsf{S} \geq \rho d + 1$.

If ρ is not a lower bound on the rank of $\boldsymbol{A}$, meaning $\text{rank}(\boldsymbol{A}) < \rho$, 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 in a Las Vegas fashion, by evaluating the matrix $\boldsymbol{A}$ at a random α in time $O(mnd)$, and computing the rank profile matrix (or a rank profile revealing PLUQ decomposition) of $\boldsymbol{A}(\alpha)$, see for instance (Dumas et al., 2015). The cost of this computation is $O(mnr^{\omega-2})$, with r the actual rank of $\boldsymbol{A}$.

As $\rho \leq r$, running Protocol 2 on a $\rho \times \rho$ matrix does not dominate the complexity, hence the total Prover cost of $O(mnr^{\omega-2} + mnd)$. From Theorem 3.2, 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 protocol verifying an upper bound on the rank. Note that Steps 2 and 3 of Protocol 4 come from the protocol verifying 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: $|\boldsymbol{\gamma}|_H \leq \rho$ means that the vector $\boldsymbol{\gamma}$ as at most ρ nonzero entries.

Theorem 3.4. *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 has Verifier cost $O(mnd)$. If we indeed have $r \leq \rho$, then there is an algorithm for the Prover with cost bound $O(mnr^{\omega-2}+mnd)$. 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 protocol from (Dumas and Kaltofen, 2014, Theorem 4) is complete, this protocol 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 2.2; or the Prover managed to cheat during the execution of Steps 2 and 3 which happens 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 $O(mnd)$. Then to find the vector $\boldsymbol{\gamma}$, the Prover can for instance first compute a PLUQ decomposition $\boldsymbol{A}(\alpha) = \boldsymbol{P}\begin{bmatrix}\boldsymbol{L}_1\\ \boldsymbol{L}_2\end{bmatrix}\begin{bmatrix}\boldsymbol{U}_1 & \boldsymbol{U}_2\end{bmatrix}\boldsymbol{Q}$ for a cost of $O(mnr^{\omega-2})$. Then, the Prover computes the vector $\begin{bmatrix}\boldsymbol{w}_1\\ \boldsymbol{w}_2\end{bmatrix} = \boldsymbol{Q}\boldsymbol{v}$, where $\boldsymbol{w}_1$ has size r, and

Protocol 4: RankUpperBound

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

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

computes $\boldsymbol{\gamma} = \boldsymbol{Q}^{\mathsf{T}}\begin{bmatrix}\boldsymbol{w}_1 + \boldsymbol{U}_1^{-1}\boldsymbol{U}_2\boldsymbol{w}_2 \\ \boldsymbol{0}\end{bmatrix}$. This vector has Hamming weight at most r (recall that $\boldsymbol{Q}$ is a permutation matrix) and satisfies $\boldsymbol{A}(\alpha)\boldsymbol{\gamma} = \boldsymbol{P}\begin{bmatrix}\boldsymbol{L}_1 \\ \boldsymbol{L}_2\end{bmatrix}\left[\boldsymbol{U}_1\boldsymbol{w}_1 + \boldsymbol{U}_2\boldsymbol{w}_2\right] = \boldsymbol{A}(\alpha)\boldsymbol{v}$. 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 these protocols verifying upper bounds and lower bounds on the rank, we directly obtain one for the rank (Protocol 5).

Protocol 5: Rank

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

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

Corollary 3.5. *Let r be the actual rank of $\boldsymbol{A}$. Protocol 5 is a probabilistically sound interactive protocol and is complete assuming $\#\mathsf{S} \geq rd + 1$ in its subprotocols. It requires $O(n)$ communica-*

tion and has Verifier cost $O(mnd)$. If we indeed have $\rho = r$, then there is a Las Vegas randomized algorithm for the Prover with expected cost $O(mnr^{\omega-2} + mnd)$. Otherwise, the probability that the Verifier incorrectly accepts is at most $(rd + 1)/\#\mathsf{S}$.

3.3. Determinant

We follow on with a protocol verifying the determinant of a polynomial matrix, using a similar evaluation-based approach: after the Verifier has checked that the degree of the provided determinant is suitable, a random evaluation point is sampled and the actual verification occurs on the evaluated input. The check on the degree of the provided determinant is to allow that the DeMillo-Lipton-Schwartz-Zippel Lemma applies and produces the claimed probability of the success. There are two choices available for the protocol to use over the base field: either the one from (Dumas et al., 2016, Section 2), which runs in a constant number of rounds but requires a minimum field size of n^2, or the one from (Dumas et al., 2017, Section 4.1) which runs in n rounds but has no requirement on the field size. Whichever protocol is chosen here, this has no impact on the asymptotic complexities which are the same for both, or on the completeness as both are perfectly 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**
		$\deg(\delta) \overset{?}{\leq} nd$
1.		$\alpha \overset{\$}{\leftarrow} \mathsf{S}$
		$\beta \leftarrow \delta(\alpha)$
	$\xleftarrow{\alpha}$	
2.	Use FieldDeterminant$(\boldsymbol{A}(\alpha), \beta)$	

Theorem 3.6. *Protocol 6 is a complete and probabilistically sound interactive protocol which requires $O(n)$ communication and has 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. Let $g = \det(\boldsymbol{A}) \in \mathsf{F}[x]$ be the actual determinant of $\boldsymbol{A}$.

If $\delta = g$, then it must be the case that $\deg(\delta) \leq nd$. Then, as FieldDeterminant is complete, the final check will be positive.

If $\delta \neq g$, there are two possibilities of failure. If the Verifier has picked an α which is a root of $\delta - g$, then $\delta(\alpha) = g(\alpha)$ and the checks from FieldDeterminant will always pass; by the DeMillo-Lipton-Schwartz-Zippel lemma, this happens with probability at most $nd/\#\mathsf{S}$. Otherwise, the Verifier has picked an α such that $\delta(\alpha) \neq g(\alpha)$ which means they will accept δ as the determinant with the probability of failure of FieldDeterminant, that is, $1/\#\mathsf{S}$. Overall, by the union bound, the probability that the Verifier accepts a wrong statement is at most $(nd + 1)/\#\mathsf{S}$.

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

3.4. Protocols based on matrix multiplication

Finally, we propose verification protocols related to matrix multiplication. While they are once again based on evaluation techniques, unlike the above protocols the ones given in this section are non-interactive and thus have no Prover or communication cost. We first consider linear system solving; recall that when working over $\mathsf{F}[x]$, given a nonsingular matrix $\boldsymbol{A}$ and a vector $\boldsymbol{b}$, this problem consists in finding a solution vector $\boldsymbol{v}$ over $\mathsf{F}[x]$ *together with* a nonzero polynomial $\delta \in \mathsf{F}[x]$ such that $\delta^{-1}\boldsymbol{v} = \boldsymbol{A}^{-1}\boldsymbol{b}$ (see for example Gupta et al., 2012).

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: $\deg(\delta) \le \min(m,n)\deg(\boldsymbol{A})$ and $\boldsymbol{A}\boldsymbol{v} = \delta\boldsymbol{b}$

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

Theorem 3.7. *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, hence the completeness of this protocol.

Otherwise, we have $\boldsymbol{A}\boldsymbol{v} - \delta\boldsymbol{b} = \boldsymbol{\Delta}$ for some nonzero vector $\boldsymbol{\Delta} \in \mathsf{F}[x]^{m\times 1}$, and the Verifier incorrectly accepts if and only if the Verifier picked an α such that $\boldsymbol{\Delta}(\alpha) = 0$. Using Lemma 2.2 with the vector $\boldsymbol{\Delta}$ of rank 1 and degree at most $2d$, it follows that the Verifier picked such an α with probability at most $2d/\#\mathsf{S}$.

The dominating step in the Verifier's work is evaluating $\boldsymbol{A}$ at α, which costs $O(mnd)$ operations in F. □

Remark that when solving linear systems over $\mathsf{F}[x]$ one is often interested in the case of a nonsingular matrix $\boldsymbol{A}$ with $m = n$. In this context, one usually seeks a solution $(\boldsymbol{v}, \delta)$ with δ of minimal degree (see (Storjohann, 2003, Section 9) and (Gupta et al., 2012, Section 7)); this implies $\deg(\delta) \le \deg(\det(\boldsymbol{A})) \le n\deg(\boldsymbol{A})$ and $\deg(\boldsymbol{v}) = \deg(\delta\boldsymbol{A}^{-1}\boldsymbol{b}) \le (n-1)\deg(\boldsymbol{A}) + \deg(\boldsymbol{b})$. In this particular case, these bounds could be checked by the Verifier at the beginning of the protocol; the probability that the Verifier incorrectly accepts becomes $(nd_{\boldsymbol{A}} + d_{\boldsymbol{b}})/\#\mathsf{S}$; and the Verifier's work costs $O(n^2d_{\boldsymbol{A}} + nd_{\boldsymbol{b}})$ operations.

Similarly, we propose a protocol verifying matrix multiplication following an approach attributed to 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: $\boldsymbol{C} = \boldsymbol{AB}$

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

Theorem 3.8. *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 product $\boldsymbol{D} = \boldsymbol{AB}$, and let $\boldsymbol{\Delta} = \boldsymbol{D} - \boldsymbol{C}$. Note that the final check of the Verifier is equivalent to $\boldsymbol{\Delta}(\alpha)\boldsymbol{v} \overset{?}{=} \boldsymbol{0}$.

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

Otherwise, $\boldsymbol{\Delta}$ 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 incorrectly. First, if the Verifier picked an evaluation point such that $\boldsymbol{\Delta}(\alpha) = \boldsymbol{0}$, which happens with probability at most $(d_{\boldsymbol{A}} + d_{\boldsymbol{B}})/\#\mathsf{S}$ by Lemma 2.2 (with rank lower bound 1), then whichever verification vector $\boldsymbol{v}$ is picked afterwards, the Verifier will always accept. Otherwise, the Verifier picked an evaluation point for which $\boldsymbol{\Delta}(\alpha) \neq \boldsymbol{0}$ but they picked a unlucky verification vector $\boldsymbol{v}$, that is, $\boldsymbol{v}$ is in the right kernel of $\boldsymbol{\Delta}(\alpha)$, which happens with probability at most $1/\#\mathsf{S}$ according to Lemma 2.1. The union bound gives the stated bound for the probability that the Verifier incorrectly accepts.

The cost for the Verifier comes from evaluating all three matrices at α and then performing three matrix-vector products over F. □

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

Corollary 3.9. *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 $O(n^2 d)$, 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 $(2d + 1)/\#\mathsf{S}$.*

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 $\mathsf{F}[x]$-row space of a given matrix $\boldsymbol{A} \in \mathsf{F}[x]^{m\times n}$.

The approach has 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 designing a reduction to several calls to the full row rank case.

4.1. Full row rank case

Protocol 9: FullRankRowSpaceMembership

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

	Prover		**Verifier**
1.		$\mathrm{rank}(\boldsymbol{A}) \overset{?}{\geq} m$ using RankLowerBound	
2.			$\boldsymbol{c} \overset{\$}{\leftarrow} \mathsf{S}^{m\times 1}$
		$\boldsymbol{c}$ ⟵	
3.	$\boldsymbol{u} \in \mathsf{F}[x]^{1\times m}$ s.t. $\boldsymbol{u}\boldsymbol{A} = \boldsymbol{v}$ $g \leftarrow \boldsymbol{u}\boldsymbol{c}$		
		g ⟶	
4.			$\deg(g) \overset{?}{\leq} m\deg(\boldsymbol{A}) + \deg(\boldsymbol{v})$ $\alpha \overset{\$}{\leftarrow} \mathsf{S}$
		α ⟵	
5.	$\boldsymbol{w} \leftarrow \boldsymbol{u}(\alpha) \in \mathsf{F}^{1\times m}$		
		$\boldsymbol{w}$ ⟶	
6.			$\boldsymbol{w}\boldsymbol{A}(\alpha) \overset{?}{=} \boldsymbol{v}(\alpha)$ $\boldsymbol{w}\boldsymbol{c} \overset{?}{=} g(\alpha)$

For this case, we propose Protocol 9. Before studying its properties, we emphasize that its soundness crucially depends on the fact that $\boldsymbol{A}$ has full row rank. To see why, let $\boldsymbol{A} = [x \quad -x]^\mathsf{T}$ and $\boldsymbol{v} = [1]$, and write $\boldsymbol{c} = [c_1 \quad c_2]^\mathsf{T}$ for the random vector chosen by the Verifier. Here, $\boldsymbol{A}$ does not have full row rank and $\boldsymbol{v}$ is not in the row space of $\boldsymbol{A}$; it is however in the rational row space of $\boldsymbol{A}$. This allows a dishonest Prover to make the Verifier accept by means of forging a *rational* vector $\boldsymbol{u}$ such that $\boldsymbol{u}\boldsymbol{A} = \boldsymbol{v}$ and $\boldsymbol{u}\boldsymbol{c} \in \mathsf{F}[x]$: the Verifier cannot detect that $\boldsymbol{u}$ was not over $\mathsf{F}[x]$, since they only receive $\boldsymbol{u}\boldsymbol{c}$ and an evaluation of $\boldsymbol{u}$. Indeed, any vector of the form $\boldsymbol{u} = [x^{-1} + f(x) \quad f(x)]$ for some $f \in \mathsf{F}(x)$ is such that $\boldsymbol{u}\boldsymbol{A} = \boldsymbol{v}$. In the likely event that $c_1 + c_2 \neq 0$, the Prover can choose any polynomial $g \in \mathsf{F}[x]$ and define $f = (c_1 + c_2)^{-1}(g - c_1 x^{-1})$; then $\boldsymbol{u}\boldsymbol{c} = g$ is a polynomial.

Remark that if $\boldsymbol{A}$ has full row rank and $\boldsymbol{v}$ belongs to the rational row space of $\boldsymbol{A}$, then we have *uniqueness* of the (rational) vector $\boldsymbol{u}$ such that $\boldsymbol{u}\boldsymbol{A} = \boldsymbol{v}$ and thus there is no flexibility for the Prover on the choice of $\boldsymbol{u}$. In this case, the following lemma plays a key role in the soundness of Protocol 9.

Lemma 4.1. *Let* $\boldsymbol{u} \in \mathsf{F}(x)^{1\times n}$ *be a rational fraction vector with* $\operatorname{denom}(\boldsymbol{u}) \neq 1$ *and let* $\mathsf{S} \subseteq \mathsf{F}$ *be a finite subset. For a vector of scalars* $\boldsymbol{c} \in \mathsf{S}^{n\times 1}$ *chosen uniformly at random, the probability that the inner product* $\boldsymbol{uc}$ *is a polynomial, i.e., that* $\operatorname{denom}(\boldsymbol{uc}) = 1$*, is at most* $1/\#\mathsf{S}$*.*

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

Now let h be any irreducible factor of f, and consider the inner product $\hat{\boldsymbol{u}}\boldsymbol{c}$ with $\hat{\boldsymbol{u}}$ seen as a vector over the extension field $\mathsf{F}[x]/\langle h\rangle$. Because $h \mid \operatorname{denom}(\boldsymbol{u})$, we know that $\hat{\boldsymbol{u}} \bmod h$ is not zero; otherwise the degree of the denominator f is not minimal. Then, since $\mathsf{S} \subseteq \mathsf{F} \subseteq \mathsf{F}[x]/\langle h\rangle$, the stated bound follows from Lemma 2.1. □

Another ingredient for our full row rank space membership protocol is a subroutine the Prover may use to actually compute the solution $\boldsymbol{u}$ to the linear system, shown in Algorithm 1. More precisely, this algorithm computes the numerator $\hat{\boldsymbol{u}}$ and the corresponding minimal denominator f. This algorithm will also be used in the protocol for arbitrary-rank matrices presented in the next section.

Algorithm 1: Linear system 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`, or `NO_SOLUTION`, or $(\hat{\boldsymbol{u}}, f) \in (\mathsf{F}[x]^{1\times m} \times \mathsf{F}[x])$ such that $\hat{\boldsymbol{u}}\boldsymbol{A} = f\boldsymbol{v}$ and f has minimal degree

1 $r, i_1, \ldots, i_r \leftarrow$ column rank profile of $\boldsymbol{A}$
2 **if** $r < m$ **then return** `LOW_RANK` `// below, r = m`
3 $\boldsymbol{B} \in \mathsf{F}[x]^{r\times r} \leftarrow$ columns $i_1, \ldots, i_r$ from $\boldsymbol{A}$
4 $\boldsymbol{y} \in \mathsf{F}[x]^{1\times r} \leftarrow$ columns $i_1, \ldots, i_r$ from $\boldsymbol{v}$
5 Compute $(\hat{\boldsymbol{u}}, f) \in (\mathsf{F}[x]^{1\times m} \times \mathsf{F}[x])$ such that $f^{-1}\hat{\boldsymbol{u}} = \boldsymbol{y}\boldsymbol{B}^{-1}$ and f has minimal degree, using (Gupta et al., 2012, Algorithm `RationalSystemSolve`)
6 **if** $\hat{\boldsymbol{u}}\boldsymbol{A} \neq f\boldsymbol{v}$ **then return** `NO_SOLUTION`
7 **return** $\hat{\boldsymbol{u}}$

As above, to simplify the cost bounds we write $d_{\boldsymbol{A}} = \max(1, \deg(\boldsymbol{A}))$ and $d_{\boldsymbol{v}} = \max(1, \deg(\boldsymbol{v}))$.

Lemma 4.2. *Algorithm 1 uses* $\widetilde{O}(m^{\omega-1}nd_{\boldsymbol{A}} + m^{\omega-1}d_{\boldsymbol{v}})$ *operations in* F*. If* $\boldsymbol{A}$ *has rank less than* m*, then* `LOW_RANK` *is returned. If* $\boldsymbol{A}$ *has rank* m *and* $\boldsymbol{v} \notin \operatorname{RowSp}_{\mathsf{F}(x)}(\boldsymbol{A})$*, then* `NO_SOLUTION` *is returned. Otherwise, the algorithm returns* $(\hat{\boldsymbol{u}}, f)$ *such that* $\hat{\boldsymbol{u}}\boldsymbol{A} = f\boldsymbol{v}$ *and* f *has minimal degree; in particular,* $\deg(f) \leq m\deg(\boldsymbol{A})$ *and* $\deg(\hat{\boldsymbol{u}}) \leq (m-1)\deg(\boldsymbol{A}) + \deg(\boldsymbol{v})$*.*

Proof. (Zhou, 2012, Chapter 11) presents a deterministic algorithm to compute the column rank profile on Line 1 using $\widetilde{O}(m^{\omega-1}nd_{\boldsymbol{A}})$ 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, and Gupta et al. (2012) showed how to solve the linear system on Line 5 deterministically using $\widetilde{O}(m^{\omega}d_{\boldsymbol{A}} + m^{\omega-1}d_{\boldsymbol{v}})$ operations; precisely, this cost bound is obtained from the results in (*ibid.*, Section 7) applied with $d = \max(d_{\boldsymbol{A}}, md_{\boldsymbol{v}})$. The degree bounds on $\hat{\boldsymbol{u}}$ and f follow from Cramer's rule.

Let $(\hat{\boldsymbol{u}}, f)$ be the system solution computed on Line 5. If $\boldsymbol{v} \notin \operatorname{RowSp}_{\mathsf{F}(x)}(\boldsymbol{A})$, then we must have $\hat{\boldsymbol{u}}\boldsymbol{A} \neq f\boldsymbol{v}$ and thus `NO_SOLUTION` is returned. Now assume that $\boldsymbol{v} \in \operatorname{RowSp}_{\mathsf{F}(x)}(\boldsymbol{A})$, that is,

there exists $\boldsymbol{w} \in \mathsf{F}(x)^{1\times m}$ such that $\boldsymbol{wA} = \boldsymbol{v}$. Then we have in particular $\boldsymbol{wB} = \boldsymbol{y}$. But because $\boldsymbol{B}$ is nonsingular, we have $\boldsymbol{w} = \boldsymbol{yB}^{-1} = f^{-1}\hat{\boldsymbol{u}}$; hence $\hat{\boldsymbol{u}}\boldsymbol{A} = f\boldsymbol{v}$. □

Finally, we present the main result of this subsection.

Theorem 4.3. *Protocol 9 is a complete and probabilistically sound interactive protocol which requires $O(md_{\boldsymbol{A}} + d_{\boldsymbol{v}})$ communication and has Verifier cost $O(mnd_{\boldsymbol{A}} + nd_{\boldsymbol{v}})$. If $\operatorname{rank}(\boldsymbol{A}) = m$ and $\boldsymbol{v} \in \operatorname{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A})$, there is an algorithm for the Prover with cost $\widetilde{O}(nm^{\omega-1}d_{\boldsymbol{A}} + m^{\omega-1}d_{\boldsymbol{v}})$. Otherwise, the probability that the Verifier incorrectly accepts is at most $(3md_{\boldsymbol{A}} + d_{\boldsymbol{v}} + 1)/\#\mathsf{S}$.*

Proof. If $\operatorname{rank}(\boldsymbol{A}) < m$, then from Theorem 3.3, the probability that the Verifier incorrectly accepts is at most $1/\#\mathsf{S}$, less than the stated bound in the theorem. And if $\boldsymbol{v}$ is the zero vector, then the protocol easily succeeds when the Prover sends all zeros for g and $\boldsymbol{w}$; remark that $\boldsymbol{u} = \boldsymbol{0}$ is the only solution to $\boldsymbol{uA} = \boldsymbol{v}$ when $\boldsymbol{A}$ has full row rank. So for the remainder of the proof, assume that $\boldsymbol{v}$ is nonzero and $\boldsymbol{A}$ has full row rank m.

The rank check entails $2m + 1$ field elements of communication, and 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 in the protocol to at most $m(d_{\boldsymbol{A}} + 4) + d_{\boldsymbol{v}} + 3$ 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 method 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 space of $\boldsymbol{A}$ (as checked by the protocol), the rational row space of $\boldsymbol{A}$, or neither.

Case 1: $\boldsymbol{v} \in \operatorname{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A})$. 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 3 must exist by the definition of $\operatorname{RowSp}_{\mathsf{F}[x]}$, and computing $\boldsymbol{u}$ can be completed by the Verifier according to Lemma 4.2 in the stated cost bound.

If the computations of $\boldsymbol{u}$ and g at Step 3 and of $\boldsymbol{w}$ at Step 5 are performed correctly by the Prover, then the Verifier's checks on Step 6 will succeed for any choice of α. Note also that in this case, $\deg(g) = \deg(\boldsymbol{uc}) \le \deg(\boldsymbol{u})$, and $\deg(\boldsymbol{u}) \le m\deg(\boldsymbol{A}) + \deg(\boldsymbol{v})$ holds (see Lemma 4.2), hence the degree check at Step 4.

This proves the completeness of the protocol.

Case 2: $\boldsymbol{v} \in \operatorname{RowSp}_{\mathsf{F}(x)}(\boldsymbol{A}) \setminus \operatorname{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A})$. In this case, the assertion being verified 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 2. Since $\boldsymbol{A}$ has full row rank, there is a unique rational solution $\boldsymbol{u} \in \mathsf{F}(x)^{1\times m}$ such that $\boldsymbol{uA} = \boldsymbol{v}$, and by the assumption of this case we have $\operatorname{denom}(\boldsymbol{u}) \ne 1$; besides, Lemma 4.2 ensures $\deg(\operatorname{denom}(\boldsymbol{u})) \le md_{\boldsymbol{A}}$ and $\deg(\operatorname{numer}(\boldsymbol{u})) \le (m-1)d_{\boldsymbol{A}} + d_{\boldsymbol{v}}$. Then, Lemma 4.1 tells us that the probability that $\boldsymbol{uc}$ is a polynomial is at most $1/\#\mathsf{S}$. Let g be the polynomial sent by the Prover at Step 3. If $\boldsymbol{uc}$ is not a polynomial, then $\boldsymbol{uc} - g$ is a nonzero rational fraction with numerator degree at most

$$\max(\deg(\operatorname{numer}(\boldsymbol{u})), \deg(g) + \deg(\operatorname{denom}(\boldsymbol{u}))) \le 2md_{\boldsymbol{A}} + d_{\boldsymbol{v}}. \tag{4.1}$$

From Lemma 2.2, the probability that $\boldsymbol{A}(\alpha)$ does not have full row rank is at most $md_{\boldsymbol{A}}/\#\mathsf{S}$. Otherwise, the vector $\boldsymbol{w} = \boldsymbol{u}(\alpha)$ is the unique solution to $\boldsymbol{wA}(\alpha) = \boldsymbol{v}(\alpha)$, so the Prover is obliged to send this $\boldsymbol{w}$ on Step 5.

Then, if the Verifier incorrectly accepts, we must have $\boldsymbol{wc} = g(\alpha)$, which means $\boldsymbol{u}(\alpha)\boldsymbol{c} = g(\alpha)$, or in other words, α is a root of $\boldsymbol{uc} - g$. The degree bound in Equation (4.1) gives an upper bound on the number of such roots $\alpha \in \mathsf{F}$.

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 $(3md_{\boldsymbol{A}} + d_{\boldsymbol{v}} + 1)/\#\mathsf{S}$, as stated.

Case 3: $\boldsymbol{v} \notin \mathrm{RowSp}_{\mathsf{F}(x)}(\boldsymbol{A})$. Again, the assertion being verified 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 matrix

$$\tilde{\boldsymbol{A}} = \begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{v} \end{bmatrix} \in \mathsf{F}[x]^{(m+1)\times n}.$$

By the assumption of this case, $\mathrm{rank}(\tilde{\boldsymbol{A}}) = \mathrm{rank}(\boldsymbol{A}) + 1 = m + 1$. But the vector $\boldsymbol{w}$ provided at Step 6 is such that $\boldsymbol{w}\boldsymbol{A}(\alpha) = \boldsymbol{v}(\alpha)$: it corresponds to a nonzero vector $[-\boldsymbol{w} \;\; 1]$ in the left kernel of $\tilde{\boldsymbol{A}}(\alpha)$, which therefore has rank at most m.

Since all $(m+1)\times(m+1)$ minors of $\tilde{\boldsymbol{A}}$ have degree at most $md_{\boldsymbol{A}} + d_{\boldsymbol{v}}$, the proof of Lemma 2.2 shows that the probability that $\mathrm{rank}(\tilde{\boldsymbol{A}}(\alpha)) \le m$ is at most $(md_{\boldsymbol{A}} + d_{\boldsymbol{v}})/\#\mathsf{S}$. □

4.2. Arbitrary rank case

Now we move to the general case of a matrix $\boldsymbol{A}$ with arbitrary rank r.

The idea behind our protocol is inspired by Mulders and Storjohann (2004). We make use of the full row rank case by considering a matrix $\boldsymbol{C} \in \mathsf{F}[x]^{r\times m}$ such that $\boldsymbol{CA}$ has full row rank. Thus $\boldsymbol{CA}$ has the same rational row space as $\boldsymbol{A}$, and if $\boldsymbol{v} \in \mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A})$, then there is a unique rational vector $\boldsymbol{w} \in \mathsf{F}(x)^{1\times r}$ such that $\boldsymbol{wCA} = \boldsymbol{v}$. In particular, for $f = \mathrm{denom}(\boldsymbol{w})$ we have $f\boldsymbol{v} \in \mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A})$, and therefore if $f = 1$ the verification is already complete.

Although it might be that $\boldsymbol{w}$ has a nontrivial denominator f, this approach can still be used for verification by considering *several* such matrices $\boldsymbol{C}_1, \ldots, \boldsymbol{C}_t$ and rational vectors $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_t$ with denominators $f_1, \ldots, f_t$. Indeed, we will see that these matrices can be chosen such that the greatest common divisor of $f_1, \ldots, f_t$ is 1; as we show in the next lemma, this implies $\boldsymbol{v} \in \mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A})$.

Lemma 4.4. *Let* $\boldsymbol{A} \in \mathsf{F}[x]^{m\times n}$ *and* $\boldsymbol{v} \in \mathsf{F}[x]^{1\times n}$. *Let* $f_1, \ldots, f_t \in \mathsf{F}[x]$ *be such that* $f_i\boldsymbol{v} \in \mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A})$ *for* $1 \le i \le t$. *If* $\gcd(f_1, \ldots, f_t) = 1$, *then* $\boldsymbol{v} \in \mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A})$.

Proof. The gcd assumption implies that there exist $u_1, \ldots, u_t \in \mathsf{F}[x]$ such that $u_1 f_1 + \cdots + u_t f_t = 1$. It directly follows that $\boldsymbol{v} = u_1(f_1\boldsymbol{v}) + \cdots + u_t(f_t\boldsymbol{v})$ belongs to $\mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A})$. □

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

Lemma 4.5. *Let* $d = \max_i \deg(f_i)$, *and suppose* $\#\mathsf{S} \ge 2d$. *Then Protocol 10 is a complete and probabilistically sound interactive protocol which requires* $O(d + t)$ *communication and has Verifier cost* $O(dt)$. *If* $\gcd(f_1, \ldots, f_t) = 1$, *then there is a Las Vegas randomized algorithm for the Prover with expected cost bound* $\widetilde{O}(dt)$. *Otherwise, the probability that the Verifier incorrectly accepts is at most* $(2d - 1)/\#\mathsf{S}$.

Protocol 10: CoPrime

Public: $t \geq 2$ polynomials $f_1, \ldots, f_t \in \mathsf{F}[x]$
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. $f_1 s_1 + h s_2 = 1$, $\deg(s_1) < \deg(h)$, and $\deg(s_2) < \deg(f_1)$, where $h = f_2 + \sum_{i=3}^{t} \beta_i f_i$	$\xrightarrow{s_1, s_2, \beta_3, \ldots, \beta_t}$	
2.			$\deg(s_1) \overset{?}{<} \max_{i \geq 2} \deg(f_i)$ $\deg(s_2) \overset{?}{<} \deg(f_1)$ $\alpha \overset{\$}{\leftarrow} \mathsf{S}$ $h_\alpha \leftarrow f_2(\alpha) + \sum_{i=3}^{t} \beta_i f_i(\alpha)$ $f_1(\alpha) s_1(\alpha) + h_\alpha s_2(\alpha) \overset{?}{=} 1$

Proof. The communication and Verifier costs are clear.

Write $g = \gcd(f_1, \ldots, f_t)$, and suppose first that $g \neq 1$. Since g divides $f_1 s_1 + h s_2$, 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, 2013, 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 $\#\mathsf{S} \geq 2d$, 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 done via the fast extended Euclidean algorithm on f_1 and h, which costs $\widetilde{O}(dt)$. □

Protocol 11 shows an interactive protocol verifying row space membership. For free (and as a necessary aspect of the protocol), the rank ρ is also verified.

The Prover first selects t matrices $\boldsymbol{C}_i \in \mathsf{F}^{r \times m}$ such that $\boldsymbol{C}_i \boldsymbol{A}$ has full row rank $r = \operatorname{rank}(\boldsymbol{A})$ and the corresponding denominators f_i of the rational solutions to $\boldsymbol{w} \boldsymbol{C}_i \boldsymbol{A} = \boldsymbol{v}$ have no common factor.

The Verifier then confirms that the gcd of all denominators is 1 using CoPrime. Using FullRankRowSpaceMembership, the Verifier also confirms that each $\boldsymbol{C}_i \boldsymbol{A}$ has full rank and each $f_i \boldsymbol{v}$ is in the row space of $\boldsymbol{C}_i \boldsymbol{A}$ and therefore in the row space of $\boldsymbol{A}$ as well; by Lemma 4.4 this ensures that $\boldsymbol{v}$ is itself in the row space of $\boldsymbol{A}$.

To save communication costs, the matrices $\boldsymbol{C}_i$ have a certain structure:

Definition 4.6. *A matrix $\boldsymbol{C} \in \mathsf{F}^{m \times n}$ is a* sub-Toeplitz *matrix if $m \leq n$ and $\boldsymbol{C}$ consists of m rows selected out of a full $n \times n$ Toeplitz matrix.*

Note that we can always write such a matrix $\boldsymbol{C}$ as a sub-permutation matrix $\boldsymbol{S} \in \{0,1\}^{m\times n}$ times the full Toeplitz matrix $\boldsymbol{T} \in \mathsf{F}^{n\times n}$, i.e., $\boldsymbol{C} = \boldsymbol{S}\boldsymbol{T}$[1]. The benefit for us is in the communication savings:

Lemma 4.7. *An $m \times n$ sub-Toeplitz matrix $\boldsymbol{C}$ can be sent with $O(n)$ communication.*

Proof. Writing $\boldsymbol{C} = \boldsymbol{S}\boldsymbol{T}$ as above, simply send the $2n-1$ entries of the full Toeplitz matrix $\boldsymbol{T}$ and the m row indices selected by $\boldsymbol{S}$. □

The number t of sub-Toeplitz matrices sent must be large enough, according to the field size, so that the Prover can actually find them with the required properties (see Algorithm 2 below). This value t is computed by the Verifier and Prover independently as shown in Step 2, where we use the slight abuse of notation that, when F is infinite, $\log_{\#\mathsf{F}} \alpha = 0$ for any positive finite value α.

Protocol 11: RowSpaceMembership

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

	Prover		Verifier
1.		$\mathrm{rank}(\boldsymbol{A}) \stackrel{?}{\leq} \rho$ using RankUpperBound	
2.			$\rho \stackrel{?}{\leq} \min(m,n)$ $t \leftarrow 1+$ $\max\left(1, \lceil \log_{\#\mathsf{F}/\rho}(2\rho \deg(\boldsymbol{A}))\rceil\right)$
3.	Compute sub-Toeplitz $\boldsymbol{C}_1,\ldots,\boldsymbol{C}_t \in \mathsf{F}^{\rho\times m}$ and polynomials $f_1,\ldots,f_t \in \mathsf{F}[x]$ s.t. $\forall i, \mathrm{rank}(\boldsymbol{C}_i\boldsymbol{A}) = \rho$, and $\forall i, f_i\boldsymbol{v} \in \mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{C}_i\boldsymbol{A})$, and $\gcd(f_1,\ldots,f_t) = 1$		
		$\boldsymbol{C}_1,\ldots,\boldsymbol{C}_t, f_1,\ldots,f_t \longrightarrow$	
4.			$\forall i, \deg(f_i) \stackrel{?}{\leq} \rho \deg(\boldsymbol{A})$
5.		$\gcd(f_1,\ldots,f_t) \stackrel{?}{=} 1$ using CoPrime	
6.	**for** $i = 1,\ldots,t$ **do** $f_i\boldsymbol{v} \stackrel{?}{\in} \mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{C}_i\boldsymbol{A})$ and $\mathrm{rank}(\boldsymbol{C}_i\boldsymbol{A}) \stackrel{?}{=} \rho$ using FullRankRowSpaceMembership		

[1]We hope that the reader will forgive us for overloading the capital letter S: a bold $\boldsymbol{S}$ always refers to this sub-permutation matrix, while a sans-serif S refers to a subset of the field F used to select random elements.

We now proceed to show how the Prover can actually find the values required on Step 3. We write $r = \text{rank}(\boldsymbol{A})$; if the Prover is honest, then in fact $r = \rho$. The next lemma is inspired from (Mulders and Storjohann, 2004).

Lemma 4.8. *Let $\boldsymbol{A} \in \mathsf{F}[x]^{m\times n}$ with rank r; $\boldsymbol{v} \in \text{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A})$; $\boldsymbol{S} \in \{0,1\}^{r\times m}$ be a selection of r out of m rows; $p \in \mathsf{F}[x]$ be an irreducible polynomial; and $\boldsymbol{T} \in \mathsf{F}^{m\times m}$ be a Toeplitz matrix with entries chosen independently and uniformly at random from a finite subset S of F. Then either $\boldsymbol{STA}$* always *has rank strictly below r, or for any rational solution $\boldsymbol{w} \in \mathsf{F}(x)^{1\times r}$ to $\boldsymbol{wSTA} = \boldsymbol{v}$, the probability that $\text{rank}(\boldsymbol{STA}) < r$ or that p divides $\text{denom}(\boldsymbol{w})$ is at most $r/\#\mathsf{S}$.*

Proof. Let $\hat{\boldsymbol{T}}$ be a generic $m \times m$ Toeplitz matrix, defined by $2m-1$ indeterminates $z_1, \ldots, z_{2m-1}$. Because $\boldsymbol{I}_m$ is an evaluation of $\hat{\boldsymbol{T}}$, then clearly $\text{rank}(\hat{\boldsymbol{T}}\boldsymbol{A}) = \text{rank}(\boldsymbol{A}) = r$, and furthermore $\text{rank}(\boldsymbol{S}\hat{\boldsymbol{T}}\boldsymbol{A}) = r$ if and only if $\boldsymbol{S}$ selects r linearly independent rows from $\hat{\boldsymbol{T}}\boldsymbol{A}$.

So for the remainder assume that $\text{rank}(\boldsymbol{S}\hat{\boldsymbol{T}}\boldsymbol{A})$ is nonsingular over $(\mathsf{F}[x])[z_1, \ldots, z_{2m-1}]$; otherwise $\text{rank}(\boldsymbol{STA}) < r$ for any choice of $\boldsymbol{T}$, and we are done.

The structure of the proof is as follows. We first show the existence of a unimodular-completable matrix $\boldsymbol{U} \in \mathsf{F}[x]^{n\times r}$ such that $\boldsymbol{STA}$ and $\boldsymbol{STU}$ are closely related: in particular, they both have full rank r if and only if the latter has non-zero determinant, and p divides $\boldsymbol{w}$ only when this determinant is divisible by p. The proof proceeds to demonstrate these properties, as well as the fact that $\boldsymbol{S}\hat{\boldsymbol{T}}\boldsymbol{U}$ has nonzero determinant generically, and therefore with high probability for a random choice of $\boldsymbol{T}$.

Let $\boldsymbol{P} \in \{0,1\}^{n\times r}$ be a sub-permutation matrix which selects r linearly independent columns from $\boldsymbol{S}\hat{\boldsymbol{T}}\boldsymbol{A}$. Then $\text{rank}(\boldsymbol{AP}) = r$ and we consider a factorization $\boldsymbol{AP} = \boldsymbol{UB}$, where

- $\boldsymbol{B} \in \mathsf{F}[x]^{r\times r}$ is a row basis of $\boldsymbol{AP}$ (and therefore $\boldsymbol{B}$ is nonsingular); 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 $\det([\boldsymbol{U} \mid \boldsymbol{V}]) \in \mathsf{F} \setminus \{0\}$.

Such a factorization always exists: if $\hat{\boldsymbol{B}} \in \mathsf{F}[x]^{r\times n}$ is any row basis of $\boldsymbol{A}$, then there is a unimodular matrix $[\boldsymbol{U} \mid \boldsymbol{V}] \in \mathsf{F}[x]^{m\times m}$ such that $[\boldsymbol{U} \mid \boldsymbol{V}][\hat{\boldsymbol{B}}^{\mathsf{T}} \mid \boldsymbol{0}]^{\mathsf{T}} = \boldsymbol{U}\hat{\boldsymbol{B}} = \boldsymbol{A}$; and then we have $\boldsymbol{UB} = \boldsymbol{AP}$ where $\boldsymbol{B} = \hat{\boldsymbol{B}}\boldsymbol{P}$. It is easily verified that $\boldsymbol{B}$ has full row rank and the same $\mathsf{F}[x]$-row space as $\boldsymbol{AP}$; that is, $\boldsymbol{B}$ is a row basis of $\boldsymbol{AP}$.

From this factorization and the fact that $\boldsymbol{B}$ is nonsingular, we know that

$$\text{rank}(\boldsymbol{AP}) = \text{rank}(\boldsymbol{S}\hat{\boldsymbol{T}}\boldsymbol{AP}) = \text{rank}(\boldsymbol{S}\hat{\boldsymbol{T}}\boldsymbol{UB}) = \text{rank}(\boldsymbol{S}\hat{\boldsymbol{T}}\boldsymbol{U}) = r$$

over the ring $(\mathsf{F}[x])[z_1, \ldots, z_{2m-1}]$.

Now because $[\boldsymbol{U} \mid \boldsymbol{V}]$ is unimodular, it is always nonsingular over the extension field $\mathsf{F}[x]/\langle p\rangle$, and therefore $\text{rank}(\boldsymbol{U}) = r$ over $\mathsf{F}[x]/\langle p\rangle$. Since the entries of $\boldsymbol{S}\hat{\boldsymbol{T}}$ do not contain x and from the rank condition above, this means that $\boldsymbol{S}\hat{\boldsymbol{T}}\boldsymbol{U}$ is nonsingular over $(\mathsf{F}[x]/\langle p\rangle)[z_1, \ldots, z_{2m-1}]$ for any choice of the polynomial p.

The determinant $\det(\boldsymbol{S}\hat{\boldsymbol{T}}\boldsymbol{U})$ is therefore a nonzero polynomial in $z_1, \ldots, z_{2m-1}$ over $\mathsf{F}[x]/\langle p\rangle$ with total degree at most r. Then, by the DeMillo-Lipton-Schwartz-Zippel lemma, the probability that $\det(\boldsymbol{STU}) \bmod p = 0$ is at most $r/\#\mathsf{S}$.

Connecting this back to $\boldsymbol{A}$, if $p \nmid \det(\boldsymbol{STU})$, then we have

$$r = \text{rank}(\boldsymbol{STU}) = \text{rank}(\boldsymbol{STUB}) = \text{rank}(\boldsymbol{STAP}) \leq \text{rank}(\boldsymbol{STA}) \leq r,$$

and hence $\boldsymbol{STA}$ has full row rank r.

Finally, we show that denom($\boldsymbol{w}$) divides det($\boldsymbol{STU}$); this implies $p \nmid \text{denom}(\boldsymbol{w})$. Recall that $\boldsymbol{B}$ is nonsingular with the same $\mathsf{F}[x]$-row space as $\boldsymbol{AP}$; then, because $\boldsymbol{v} \in \text{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A})$, we have $\boldsymbol{vP} \in \text{RowSp}_{\mathsf{F}[x]}(\boldsymbol{B})$, so there exists $\boldsymbol{y} \in \mathsf{F}[x]^{1\times r}$ such that $\boldsymbol{yB} = \boldsymbol{vP}$. In addition, since $\boldsymbol{STU}$ is nonsingular, there exists an *adjugate* matrix $\boldsymbol{D} \in \mathsf{F}[x]^{r\times r}$ such that $\det(\boldsymbol{STU})(\boldsymbol{STU})^{-1} = \boldsymbol{D}$. Putting these facts together, we have

$$\begin{aligned} \boldsymbol{wSTA} &= \boldsymbol{v} \\ \boldsymbol{wSTAP} &= \boldsymbol{vP} \\ \boldsymbol{wSTUB} &= \boldsymbol{yB} \\ \boldsymbol{wSTU} &= \boldsymbol{y} \\ \boldsymbol{w}\det(\boldsymbol{STU}) &= \boldsymbol{yD}. \end{aligned}$$

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($\boldsymbol{STU}$) is a multiple of denom($\boldsymbol{w}$). Hence denom($\boldsymbol{w}$) is divisible by p only if det($\boldsymbol{STU}$) is divisible by p, which we already established occurs with probability at most $r/\#\mathsf{S}$. □

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

Here we require the Prover to know a finite subset $\mathsf{S} \subseteq \mathsf{F}$. Because this set is never communicated nor part of the public information, it is not necessarily the same as any subset S used by the Verifier in other protocols. In order to match with Protocol 11, the Prover should choose $\mathsf{S} = \mathsf{F}$ if F is finite, and otherwise $\#\mathsf{S} \geq 2r^2 \deg(\boldsymbol{A})$.

Algorithm 2: Honest Prover for RowSpaceMembership

Input: $\boldsymbol{A} \in \mathsf{F}[x]^{m\times n}$ with rank r, $\boldsymbol{v} \in \text{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A})$, finite $\mathsf{S} \subseteq \mathsf{F}$
Output: $\boldsymbol{C}_1, \ldots, \boldsymbol{C}_t, f_1, \ldots, f_t$ satisfying the conditions of Step 3 from Protocol 11

$t \leftarrow 1 + \lceil \log_{\#\mathsf{S}/r}(2r\deg(\boldsymbol{A})) \rceil$
repeat
  $\boldsymbol{T}_1 \leftarrow$ random $m \times m$ Toeplitz matrix with entries from S
until $\text{rank}(\boldsymbol{T}_1\boldsymbol{A}) = r$
$\boldsymbol{S} \in \{0,1\}^{r\times m} \leftarrow$ selection of r linearly independent rows from $\boldsymbol{T}_1\boldsymbol{A}$
$w_1 \leftarrow$ solution to $\boldsymbol{w}_1\boldsymbol{ST}_1\boldsymbol{A} = \boldsymbol{v}$, using Algorithm 1
repeat
  $i \leftarrow 2$
  while $i \leq t$ **do**
    $\boldsymbol{T}_i \leftarrow$ random $m \times m$ Toeplitz matrix with entries from S
    $\boldsymbol{w}_i \leftarrow$ solution to $\boldsymbol{w}_i\boldsymbol{ST}_i\boldsymbol{A} = \boldsymbol{v}$, using Algorithm 1
    if $\boldsymbol{w}_i$ *is not* `LOW_RANK` **then** $i \leftarrow i + 1$
until $\gcd(\text{denom}(\boldsymbol{w}_1), \ldots, \text{denom}(\boldsymbol{w}_t)) = 1$
return $\boldsymbol{ST}_1, \ldots, \boldsymbol{ST}_t$ *and* $\text{denom}(\boldsymbol{w}_1), \ldots, \text{denom}(\boldsymbol{w}_t)$

Lemma 4.9. *If* $\boldsymbol{v} \in \text{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A})$ *and* $\#\mathsf{S} \geq 2r$ *where* $r = \text{rank}(\boldsymbol{A})$*, then Algorithm 2 is a correct Las Vegas randomized algorithm with expected cost bound* $\widetilde{O}(mnr^{\omega-2}d_{\boldsymbol{A}} + r^{\omega-1}d_{\boldsymbol{v}})$.

Proof. Computing the rank and the row rank profile (giving r independent rows) on Lines 4 and 5 can be done deterministically in the stated cost bound via the column rank profile algorithm from (Zhou, 2012, Section 11), just as was used in Algorithm 1.

Each matrix product $\boldsymbol{T}_i\boldsymbol{A}$ can be explicitly computed in $\widetilde{O}(mnd_{\boldsymbol{A}})$ operations using $O(nd_{\boldsymbol{A}})$ Toeplitz-vector products, each done in $\widetilde{O}(m)$ operations by relying on 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.

From Lemma 4.8, for each random Toeplitz matrix $\boldsymbol{T}_i$, the probability that $\boldsymbol{S}\boldsymbol{T}_i\boldsymbol{A}$ is nonsingular is at least $1 - r/\#\mathsf{S} \geq 1/2$. Therefore the expected number of iterations of the first loop is at most 2, and the expected number of iterations of the nested while loop until i reaches t is at most $2t$.

Write $f_i = \text{denom}(\boldsymbol{w}_i)$. To find the expected number of iterations of the outer loop on Lines 7 to 13, we need the probability that $\gcd(f_1, \ldots, f_t) = 1$ given that each $\boldsymbol{S}\boldsymbol{T}_i\boldsymbol{A}$ has full row rank r.

If $\gcd(f_1, \ldots, f_t) \neq 1$, then there is some irreducible polynomial p which divides every denominator $f_1, \ldots, f_t$. Because the $\boldsymbol{T}_i$'s are chosen independently of each other, the events "p divides f_i" are pairwise independent; thus, according to Lemma 4.8, the probability that any given irreducible polynomial p is such a common factor is at most $(r/\#\mathsf{S})^{t-1}$.

The degree of f_1 is at most $rd_{\boldsymbol{A}}$ since $\boldsymbol{S}\boldsymbol{T}_i\boldsymbol{A}$ is $r \times n$ with degree $d_{\boldsymbol{A}}$; this also gives an upper bound on the number of distinct irreducible factors p of f_1. Taking the union bound we see that the probability of *any* factor being shared by all 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 is $O(1)$.

The stated cost bound follows from Lemma 4.2. It does not involve 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 $\boldsymbol{S}\boldsymbol{T}_i\boldsymbol{A}$ is square.

- When each $\boldsymbol{S}\boldsymbol{T}_i\boldsymbol{A}$ is square, computing $\boldsymbol{w}_i$ can be done in a simpler way than by calling Algorithm 1, as follows: check that $\boldsymbol{S}\boldsymbol{T}_i\boldsymbol{A}$ is nonsingular to confirm the rank, and then use a fast linear system solver to obtain $\boldsymbol{w}_i$.

- The solution vectors $\boldsymbol{w}_i$ may be re-used in the subprotocols FullRankRowSpaceMembership confirming that each $f_i\boldsymbol{v} \in \text{RowSp}_{\mathsf{F}[x]}(\boldsymbol{S}\boldsymbol{T}_i\boldsymbol{A})$.

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

Theorem 4.10. *Assuming* $\#S \geq 2\min(m,n)d_{\boldsymbol{A}}$*, 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 has Verifier cost* $O(mnd_{\boldsymbol{A}}t + nd_{\boldsymbol{v}}t)$*. If* $\boldsymbol{v} \in \mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A})$*, there is a Las Vegas randomized algorithm for the Prover with expected cost* $\widetilde{O}(mnr^{\omega-2}d_{\boldsymbol{A}} + r^{\omega-1}d_{\boldsymbol{v}})$*. Otherwise, the probability that the Verifier incorrectly accepts is at most* $(3rd_{\boldsymbol{A}} + d_{\boldsymbol{v}} + 1)/\#S$*.*

Proof. For the communication, note that sending each $\boldsymbol{C}_i$ has communication cost $O(m)$ from Lemma 4.7. 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 subprotocols. 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 f_i and Theorems 3.3, 3.4 and 4.3 and Lemma 4.5, 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 subprotocols.

If the rank conditions being verified on Steps 1 and 6 are true, then all matrices $\boldsymbol{C}_i\boldsymbol{A}$ have full row rank equal to the rank of $\boldsymbol{A}$, that is, $\mathrm{rank}(\boldsymbol{C}_i\boldsymbol{A}) = \rho = r$. And if the statements verified on Steps 5 and 6 are true as well, then we have $\boldsymbol{v} \in \mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A})$ according to Lemma 4.4. Therefore the soundness of this protocol depends only on the probabilistic soundness of those subprotocols.

For the remainder of the proof, we assume that $\boldsymbol{v} \notin \mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A})$ and we want to know an upper bound on the probability that the Verifier incorrectly accepts. For this, we divide into cases depending on which subprotocol incorrectly accepted:

Case 1: $\mathrm{rank}(\boldsymbol{A}) > \rho$. According to Theorem 3.4, the probability that the Verifier incorrectly accepts in RankUpperBound on Step 1 is at most $(rd_{\boldsymbol{A}} + 1)/\#\mathsf{S}$.

Case 2: $\mathrm{rank}(\boldsymbol{A}) \leq \rho$ *and* $\gcd(f_1, \ldots, f_t) \neq 1$. We know that each $\deg(f_i) \leq rd_{\boldsymbol{A}}$, where r is the true rank of $\boldsymbol{A}$. By Lemma 4.5, the probability that the Verifier incorrectly accepts in subprotocol CoPrime is at most $(2\max_i(\deg(f_i)) - 1)/\#\mathsf{S}$, which is at most $(2rd_{\boldsymbol{A}} - 1)/\#\mathsf{S}$.

Case 3: $\mathrm{rank}(\boldsymbol{A}) \leq \rho$ *and* $\gcd(f_1, \ldots, f_t) = 1$. Then, by Lemma 4.4, there exists $i \in \{1, \ldots, t\}$ such that $f_i\boldsymbol{v} \notin \mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A})$, and thus either $\mathrm{rank}(\boldsymbol{C}_i\boldsymbol{A}) < \rho$ or $f_i\boldsymbol{v} \notin \mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{C}_i\boldsymbol{A})$. That is, the statement being verified by FullRankRowSpaceMembership on the ith iteration of Step 6 is false.

Because of the degree checks on Step 4, we know that $\deg(f_i\boldsymbol{v}) \leq rd_{\boldsymbol{A}} + d_{\boldsymbol{v}}$. Therefore from Theorem 4.3, the probability that the Verifier incorrectly accepts in FullRankRowSpaceMembership is at most $(4rd_{\boldsymbol{A}} + d_{\boldsymbol{v}} + 1)/\#\mathsf{S}$.

Observe that the three cases are disjoint and cover all possibilities. In every case, the probability that the Verifier incorrectly accepts is at most that in Case 3, which proves the last claim in the Theorem statement. □

We note that it is always possible to conduct the checks on Step 6 of RowSpaceMembership in parallel, so that the total number of *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 a condition under which this logarithmic factor can be eliminated.

Corollary 4.11. *If $\#S \geq 2mnd_A$, then Protocol 11 requires only $O(n + md_A + d_v)$ communication and has Verifier cost $O(mnd_A + nd_v)$.*

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 RowSpaceMembership 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 entries in the base field is enough to ensure good probability of success.

Lemma 5.1. *Let $\boldsymbol{A} \in \mathsf{F}[x]^{m\times n}$ and $\boldsymbol{B} \in \mathsf{F}[x]^{\ell\times n}$. Let $R \in \{\mathsf{F}[x], \mathsf{F}(x)\}$. Then the following statement holds: Assuming that*

$$\mathrm{RowSp}_R(\boldsymbol{A}) \not\subseteq \mathrm{RowSp}_R(\boldsymbol{B}),$$

then the F-vector space

$$\mathsf{V} = \left\{\boldsymbol{\lambda} \in \mathsf{F}^{1\times m} \mid \boldsymbol{\lambda A} \in \mathrm{RowSp}_R(\boldsymbol{B})\right\}$$

has dimension at most $m-1$. For $\boldsymbol{\lambda} \in \mathsf{F}^{1\times m}$ with entries chosen independently and uniformly at random from a finite subset $S \subseteq \mathsf{F}$ then $\boldsymbol{\lambda A} \in \mathrm{RowSp}_R(\boldsymbol{B})$ with probability at most $\frac{1}{\#S}$.

Proof. Suppose that the vector space V has dimension at least m. Then V is the entire space $\mathsf{F}^{1\times m}$, and every row of $\boldsymbol{A}$ is in $\mathrm{RowSp}_R(\boldsymbol{B})$; hence $\mathrm{RowSp}_R(\boldsymbol{A}) \subseteq \mathrm{RowSp}_R(\boldsymbol{B})$, a contradiction. Then the probability that the uniformly random vector $\boldsymbol{\lambda}$ belongs to the proper subspace $\mathsf{V} \subsetneq \mathsf{F}^{1\times m}$ follows from Lemma 2.1. □

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

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

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

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

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

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

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

Protocol 12: RowSpaceSubset

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

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

Proof. The Verifier may incorrectly accept if either $\boldsymbol{\lambda}$ is such that $\boldsymbol{\lambda}\boldsymbol{A} \in \mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{B})$, which happens with probability at most $1/\#\mathsf{S}$ by Lemma 5.1, or the subprotocol 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: $\boldsymbol{A} \in \mathsf{F}[x]^{m \times n}$, $\boldsymbol{B} \in \mathsf{F}[x]^{\ell \times n}$, $\rho \in \mathbb{N}$
Certifies: $\mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A}) = \mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{B})$ and $\mathrm{rank}(\boldsymbol{A}) = \mathrm{rank}(\boldsymbol{B}) = \rho$

	Prover	**Verifier**
1.	$\mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A}) \stackrel{?}{\subseteq} \mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{B})$ and $\mathrm{rank}(\boldsymbol{B}) = \rho$ using RowSpaceSubset	
2.	$\mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{B}) \stackrel{?}{\subseteq} \mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A})$ and $\mathrm{rank}(\boldsymbol{A}) = \rho$ using RowSpaceSubset	

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

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

communication and has 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* $\widetilde{O}((m+\ell)nr^{\omega-2}d)$*. Otherwise, the probability that the Verifier incorrectly accepts is at most* $(4rd+d+2)/\#\mathsf{S}$*.*

From RowSpaceEquality, we deduce a protocol verifying the property that $\boldsymbol{B}$ is a row basis of $\boldsymbol{A}$, that is, a matrix which has the same row space as $\boldsymbol{A}$ and which has full row rank.

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 row basis of $\boldsymbol{A}$

Prover		Verifier
1.	$\mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{B}) \stackrel{?}{=} \mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A})$ and $\mathrm{rank}(\boldsymbol{B}) \stackrel{?}{=} \ell$ using RowSpaceEquality	

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

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

communication and has 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* $\widetilde{O}(mn\ell^{\omega-2}d)$*, with* $\ell = r_{\boldsymbol{A}}$ *in this case. Otherwise, the probability that the Verifier incorrectly accepts is at most* $(4rd+d+3)/\#\mathsf{S}$*.*

5.2. Normal forms

Here, we give protocols for certifying *normal forms* of polynomial matrices, including the Hermite form (Hermite, 1851; MacDuffee, 1933) 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* $r \leq n$ *is in* Hermite form *if there are* pivot indices $1 \leq k_1 < \cdots < k_r \leq n$ *such that:*

(i) *(Pivots are monic, hence nonzero)*
b_{i,k_i} *is monic for all* $1 \leq i \leq r$*,*
(ii) *(Entries right of pivots are zero)*
$b_{i,j} = 0$ *for all* $i \leq i \leq r$ *and* $k_i < j \leq n$*,*

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

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.

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 that $\boldsymbol{B}$ satisfies Definition 5.5
2.	$\mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A}) \stackrel{?}{=} \mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{B})$ and $\mathrm{rank}(\boldsymbol{B}) \stackrel{?}{=} \ell$ using RowSpaceEquality	

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* $\#S \ge 2\max(md_{\boldsymbol{A}}, \ell d_{\boldsymbol{B}})$ *in its subprotocol. It requires* $O(md\log(m) + n)$ *communication and has 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* $\widetilde{O}(mnr^{\omega-2}d)$. *Otherwise, the probability that the Verifier incorrectly accepts is at most* $(4rd + d + 2)/\#S$.

Proof. To check that $\boldsymbol{B}$ is in Hermite form at Step 1, 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 et al., 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 $r \le n$ is in* $\boldsymbol{s}$-Popov form *if there are indices $1 \le k_1 < \cdots < k_r \le n$ 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* $1 \le i \le r$,

(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* $1 \le i \le r$ *and* $k_i < j \le n$,

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

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 that $(\boldsymbol{s}, \boldsymbol{B})$ satisfies Definition 5.7
2.	$\mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A}) \stackrel{?}{=} \mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{B})$ and $\mathrm{rank}(\boldsymbol{B}) \stackrel{?}{=} \ell$ 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 $\#S \ge 2\max(md_{\boldsymbol{A}}, \ell d_{\boldsymbol{B}})$ in its subprotocol.*

It requires $O(md\log(m)+n)$ *communication and has 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* $\widetilde{O}(mnr^{\omega-2}d)$. *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 protocols verifying 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. We also remark 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 $\boldsymbol{A}$.

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 $\#\mathsf{S} \geq 2\mu d$ in its subprotocol. It requires $O(\mu d \log \mu)$ communication and has 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*

Protocol 17: Saturated

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

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

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 $m \le n$ 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* $\mathrm{rank}(\boldsymbol{A}) \ge \ell$,
(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 $\mathrm{RowSp}_{\mathsf{F}(x)}(\boldsymbol{B}) = \mathrm{RowSp}_{\mathsf{F}(x)}(\boldsymbol{A})$.

Conversely, assume that the three items hold. The first two items together imply $\ell = \mathrm{rank}(\boldsymbol{B}) = \mathrm{rank}(\boldsymbol{A})$, and hence $\mathrm{RowSp}_{\mathsf{F}(x)}(\boldsymbol{A}) = \mathrm{RowSp}_{\mathsf{F}(x)}(\boldsymbol{B})$. This means $\mathrm{Saturation}(\boldsymbol{A}) = \mathrm{Saturation}(\boldsymbol{B})$, and the latter saturation is equal to $\mathrm{RowSp}_{\mathsf{F}[x]}(\boldsymbol{B})$ since $\boldsymbol{B}$ is saturated by the third item. Hence $\boldsymbol{B}$ has full row rank and $\mathsf{F}[x]$-row space equal to the saturation of $\boldsymbol{A}$. □

Theorem 6.8. *Protocol 18 is a probabilistically sound interactive protocol and is complete assuming* $\#S \ge \max(\ell d_{\boldsymbol{A}} + 1, 2nd_{\boldsymbol{B}})$ *in its subprotocols. It requires* $O(nd_{\boldsymbol{B}}\log(n))$ *communication and has 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*

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

otherwise the probability that the Verifier incorrectly accepts is at most $(4\ell d_{\boldsymbol{B}} + 2)/\#S$.

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 $\ell \overset{?}{\leq} \min(m, n)$
2.		Check $\operatorname{rank}(\boldsymbol{A}) \overset{?}{\geq} \ell$ using RankLowerBound	
3.			$\alpha \overset{\$}{\leftarrow} \mathsf{S}$ $\boldsymbol{\lambda} \overset{\$}{\leftarrow} \mathsf{S}^{1 \times n}$ $\boldsymbol{v} \leftarrow \boldsymbol{\lambda}\boldsymbol{A}(\alpha) \in \mathsf{F}^{1 \times n}$
		$\alpha, \boldsymbol{v}$ ⟵	
4.	Find $\boldsymbol{\gamma} \in \mathsf{F}^{1 \times n}$ such that $\boldsymbol{\gamma}\boldsymbol{B}(\alpha) = \boldsymbol{v}$.		
		⟶ $\boldsymbol{\gamma}$	
5.			$\boldsymbol{\gamma}\boldsymbol{B}(\alpha) \overset{?}{=} \boldsymbol{v}$
6.		Check that $\boldsymbol{B}$ is saturated using Saturated	

Proof. The check on Step 1 has no arithmetic cost, but ensures that ℓ is less than or equal to m and n. Steps 3 to 5 certify that $\operatorname{RowSp}_{\mathsf{F}(x)}(\boldsymbol{A}) \subseteq \operatorname{RowSp}_{\mathsf{F}(x)}(\boldsymbol{B})$. Note that this only communicates $O(n)$ field elements, and that the Verifier's cost at these steps amounts to the evaluation of $\boldsymbol{A}$ and $\boldsymbol{B}$ at α as well as two scalar vector-matrix products, hence a total of $O(mnd_{\boldsymbol{A}} + \ell n d_{\boldsymbol{B}})$ operations. Then the Verifier and Prover's costs follow from Theorems 3.3 and 6.6.

Note that $\operatorname{rank}(\boldsymbol{A}) \geq \ell$ and $\operatorname{RowSp}_{\mathsf{F}[x]}(\boldsymbol{A}) \subseteq \operatorname{RowSp}_{\mathsf{F}[x]}(\boldsymbol{B})$ imply that $\operatorname{rank}(\boldsymbol{B}) = \ell$, so that the precondition for the Saturated protocol on Step 6 is valid unless one of the previous checks failed.

For the probability bound, we consider each of the checks in Steps 2, 5 and 6. By Theorem 3.3, the Verifier incorrectly accepts an $\boldsymbol{A}$ with rank $< \ell$ with probability at most $1/\#\mathsf{S}$. By Theorem 6.6, the Verifier incorrectly accepts an unsaturated $\boldsymbol{B}$ with probability at most $(4\ell d_{\boldsymbol{B}} + 2)/\#\mathsf{S}$.

For Step 5, assume now that $\operatorname{rank}(\boldsymbol{A}) \geq \ell$ and $\boldsymbol{B}$ is saturated, and in particular $\operatorname{rank}(\boldsymbol{B}) = \ell$, but that $\operatorname{RowSp}_{\mathsf{F}(x)}(\boldsymbol{A}) \not\subset \operatorname{RowSp}_{\mathsf{F}(x)}(\boldsymbol{B})$. By Lemma 5.1, then for a random $\boldsymbol{\lambda} \in \mathsf{S}^{1 \times m}$ we have $\boldsymbol{\lambda}\boldsymbol{A} \in \operatorname{RowSp}_{\mathsf{F}(x)}(\boldsymbol{B})$ with probability at most $\frac{1}{\#\mathsf{S}}$. Consider now that $\boldsymbol{\lambda}\boldsymbol{A} \notin \operatorname{RowSp}_{\mathsf{F}(x)}(\boldsymbol{B})$. Let $\boldsymbol{P}$ be an $n \times n$ permutation matrix such that $\boldsymbol{B}\boldsymbol{P} = [\boldsymbol{B}_0 \mid \boldsymbol{B}_1]$ with $\boldsymbol{B}_0 \in \mathsf{F}[x]^{\ell \times \ell}$ and full rank. Let $\boldsymbol{u} \in \mathsf{F}(x)^{1 \times \ell}$ be the unique vector such that $\boldsymbol{\lambda}\boldsymbol{A}P = [\boldsymbol{u}\boldsymbol{B}_0 \mid \hat{\boldsymbol{a}}]$, for some $\hat{\boldsymbol{a}} \in \mathsf{F}(x)^{1 \times (n-\ell)}$. Then there is some index i of $\hat{\boldsymbol{a}}$ that differs from index i of $\boldsymbol{u}\boldsymbol{B}_1$. By Cramer's rule the entries of $\boldsymbol{u}$ can be written with common denominator $\det(B)$ and numerators of degree at most $(\ell - 1)d_{\boldsymbol{B}}$. Hence the entries of $\boldsymbol{u}\boldsymbol{B}_1$ have denominators and numerators of degree at most $\ell d_{\boldsymbol{B}}$ each. Hence $\boldsymbol{\lambda}\boldsymbol{A}(\alpha)$ and $\boldsymbol{u}(\alpha)\boldsymbol{B}(\alpha)$ agree at index i for at most $2\ell d_{\boldsymbol{B}}$ choices of $\alpha \in \mathsf{F}$, and we conclude that the Verifier incorrectly accepts in this case with probability at most $2\ell d_{\boldsymbol{B}}/\#\mathsf{S}$.

Summing up, the worst case is the check at Step 6, where the Verifier incorrectly accepts with probability at most $(4\ell d_{\boldsymbol{B}} + 2)/\#\mathsf{S}$. □

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 $\boldsymbol{B} \in \mathsf{F}[x]^{(n-m)\times n}$ such that $\begin{bmatrix}\boldsymbol{A}\\ \boldsymbol{B}\end{bmatrix}$ 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.		$m \overset{?}{<} n$
2.	Check that $\boldsymbol{A}$ is saturated and $\operatorname{rank}(\boldsymbol{A}) \overset{?}{=} m$ using Saturated	

Theorem 6.9. *Protocol 19 is a probabilistically sound interactive protocol and is complete assuming $\#\mathsf{S} \geq 2md$ in its subprotocols. It requires $O(nd\log(n))$ communication and has 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 $(4md + 2)/\#\mathsf{S}$.*

Proof. The costs follow from Theorems 3.3 and 6.6, noting that the protocol aborts early if $m \geq n$, and therefore m is an upper bound on the rank in the Saturated subprotocol. 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* $\operatorname{rank}(\boldsymbol{A}) \geq m - \ell$,
(ii) $\boldsymbol{BA} = \boldsymbol{0}$,
(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 $\boldsymbol{BA} = \boldsymbol{0}$; 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, $\operatorname{rank}(\boldsymbol{K}) = m - \operatorname{rank}(\boldsymbol{A}) \leq \ell$ 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 $\ell = \operatorname{rank}(\boldsymbol{B}) \leq \operatorname{rank}(\boldsymbol{K})$; therefore $\operatorname{rank}(\boldsymbol{K}) = \ell$. As a result, $\boldsymbol{B} = \boldsymbol{UK}$ 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 $\boldsymbol{I}_\ell = \boldsymbol{BV} = \boldsymbol{UKV}$ for some $\boldsymbol{V} \in \mathsf{F}[x]^{m\times\ell}$. Then, $\boldsymbol{U}$ must be unimodular, and thus $\boldsymbol{B} = \boldsymbol{UK}$ 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.		$\ell \overset{?}{\leq} m$
2.	$\operatorname{rank}(\boldsymbol{A}) \overset{?}{\geq} m - \ell$ using RankLowerBound	
3.	$\boldsymbol{BA} \overset{?}{=} \boldsymbol{0}$ using MatMul	
4.	Check that $\boldsymbol{B}$ is saturated and $\operatorname{rank}(\boldsymbol{B}) \overset{?}{=} \ell$ using Saturated	

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

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

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 3.3, 3.8 and 6.6. As before, the worst case for the Verifier is that only one of the three checked statements is wrong, and the resulting maximum of probabilities comes either from Step 2 or Step 4. □

7. Conclusion and perspectives

We have developed interactive protocols verifying 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.

Our protocols also require to work over sufficiently large fields, to ensure soundness of the randomized verification. For smaller fields, a classic workaround is to resort to a field extension, increasing the arithmetic and communication cost by a logarithmic factor. An alternative is to increase the dimension in the challenges and responses, e.g. verifying a block of vectors instead of a single vector of field elements. A further study on whether this approach is applicable and competitive here is required.

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.

The protocols presented here do not assume that the Prover has computed the result to be verified. This is however likely to be the case in many instances of verified computing, and it would then be relevant to identify which intermediate results in a Prover's computation of the solution (such as the rank profile matrix, the weak Popov form, etc), could be reused in a certificate for verifying this solution. Though more constraining on the Prover's choice of an algorithm, such information would help reducing the leading constant in the arithmetic cost of its computation.

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

We also do not know in all cases how to prove the *negation* of our statements — for example, that a vector is *not* in the row space of a polynomial matrix. It seems that some similar techniques to those we have used may work, but we have not investigated the question deeply.

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, by the IFD-Science 2017 research program of the Institut Franais du Danemark, and by the CNRS-INS2I Institute through its program for young researchers.

References

Beckermann, B., Labahn, G., 1994. A uniform approach for the fast computation of matrix-type Padé approximants. SIAM J. Matrix Anal. Appl. 15, 804–823. doi:10.1137/S0895479892230031.

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

Beckermann, B., Labahn, G., Villard, G., 2006. Normal forms for general polynomial matrices. J. Symbolic Comput. 41, 708–737. doi:10.1016/j.jsc.2006.02.001.

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

Bostan, A., Schost, É., 2005. Polynomial evaluation and interpolation on special sets of points. J. Complexity 21, 420–446. doi:10.1016/j.jco.2004.09.009.

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

Cantor, D.G., Kaltofen, E., 1991. On fast multiplication of polynomials over arbitrary algebras. Acta Inform. 28, 693–701. doi:10.1007/BF01178683.

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

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

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

Dumas, J.G., 2018. Proof-of-work certificates that can be efficiently computed in the cloud (invited talk), in: Gerdt, V.P., Koepf, W., Seiler, W.M., Vorozhtsov, E.V. (Eds.), Computer Algebra in Scientific Computing, Springer International Publishing, Cham. pp. 1–17.

Dumas, J.G., Kaltofen, E., 2014. Essentially optimal interactive certificates in linear algebra, in: ISSAC '14, ACM, New York, NY, USA. pp. 146–153. doi:10.1145/2608628.2608644.

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

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

Dumas, J.G., Pernet, C., Sultan, Z., 2015. Computing the rank profile matrix, in: ISSAC '15, ACM, New York, NY, USA. pp. 149–156. doi:10.1145/2755996.2756682.
Fiat, A., Shamir, A., 1987. How to prove yourself: Practical solutions to identification and signature problems, in: CRYPTO '86, Springer Berlin Heidelberg. pp. 186–194. doi:10.1007/3-540-47721-7_12.
Freivalds, R., 1979. Fast probabilistic algorithms, in: Mathematical Foundations of Computer Science 1979, Springer Berlin Heidelberg. pp. 57–69. doi:10.1007/3-540-09526-8_5.
Gantmacher, F.R., 1959. The Theory of Matrices. Chelsea.
von zur Gathen, J., Gerhard, J., 2013. Modern Computer Algebra. Third ed., Cambridge University Press, Cambridge.
Giorgi, P., Jeannerod, C.P., Villard, G., 2003. On the complexity of polynomial matrix computations, in: ISSAC'03, ACM. pp. 135–142. doi:10.1145/860854.860889.
Giorgi, P., Neiger, V., 2018. Certification of minimal approximant bases, in: ISSAC'18. doi:10.1145/3208976.3208991.
Goldwasser, S., Kalai, Y.T., Rothblum, G.N., 2008. Delegating computation: Interactive proofs for muggles, in: STOC '08, ACM, New York, NY, USA. pp. 113–122. doi:10.1145/1374376.1374396.
Goldwasser, S., Micali, S., Rackoff, C., 1989. The Knowledge Complexity of Interactive Proof Systems. SIAM Journal on Computing 18, 186–208. doi:10.1137/0218012.
Gupta, S., Sarkar, S., Storjohann, A., Valeriote, J., 2012. Triangular x-basis decompositions and derandomization of linear algebra algorithms over $K[x]$. J. Symbolic Comput. 47, 422–453. doi:10.1016/j.jsc.2011.09.006.
Hermite, C., 1851. Sur l'introduction des variables continues dans la théorie des nombres. Journal für die reine und angewandte Mathematik 41, 191–216.
Jeannerod, C.P., Pernet, C., Storjohann, A., 2013. Rank-profile revealing gaussian elimination and the CUP matrix decomposition. J. Symbolic Comput. 56, 46–68. doi:10.1016/j.jsc.2013.04.004.
Kailath, T., 1980. Linear Systems. Prentice-Hall.
Kaltofen, E.L., Li, B., Yang, Z., Zhi, L., 2012. Exact certification in global polynomial optimization via sums-of-squares of rational functions with rational coefficients. J. Symbolic Comput. 47, 1–15. doi:10.1016/j.jsc.2011.08.002.
Kaltofen, E.L., Nehring, M., Saunders, B.D., 2011. Quadratic-time certificates in linear algebra, in: ISSAC '11, ACM, New York, NY, USA. pp. 171–176. doi:10.1145/1993886.1993915.
Labahn, G., Neiger, V., Zhou, W., 2017. Fast, deterministic computation of the Hermite normal form and determinant of a polynomial matrix. J. Complexity 42, 44–71. doi:10.1016/j.jco.2017.03.003.
Le Gall, F., 2014. Powers of tensors and fast matrix multiplication, in: ISSAC'14, ACM. pp. 296–303. doi:10.1145/2608628.2608664.
MacDuffee, C.C., 1933. The Theory of Matrices. Springer-Verlag Berlin Heidelberg. doi:10.1007/978-3-642-99234-6.
Mulders, T., Storjohann, A., 2004. Certified dense linear system solving. Journal of Symbolic Computation 37, 485–510. doi:10.1016/j.jsc.2003.07.004.
Neiger, V., Rosenkilde, J., Solomatov, G., 2018. Computing Popov and Hermite forms of rectangular polynomial matrices, in: ISSAC '18, ACM. doi:10.1145/3208976.3208988.
Neiger, V., Vu, T.X., 2017. Computing canonical bases of modules of univariate relations, in: ISSAC'17, ACM. pp. 357–364. doi:10.1145/3087604.3087656.
Pernet, C., Stein, W., 2010. Fast computation of Hermite normal forms of random integer matrices. Journal of Number Theory 130, 1675–1683. doi:10.1016/j.jnt.2010.01.017.
Popov, V.M., 1972. Invariant description of linear, time-invariant controllable systems. SIAM Journal on Control 10, 252–264. doi:10.1137/0310020.
Schwartz, J.T., 1980. Fast probabilistic algorithms for verification of polynomial identities. J. ACM 27, 701–717. doi:10.1145/322217.322225.
Storjohann, A., 2003. High-order lifting and integrality certification. J. Symbolic Comput. 36, 613–648. doi:10.1016/S0747-7171(03)00097-X.
Van Barel, M., Bultheel, A., 1992. A general module theoretic framework for vector M-Padé and matrix rational interpolation. Numer. Algorithms 3, 451–462. doi:10.1007/BF02141952.
Villard, G., 1996. Computing Popov and Hermite forms of polynomial matrices, in: ISSAC'96, ACM. pp. 250–258. doi:10.1145/236869.237082.
Zhou, W., 2012. Fast Order Basis and Kernel Basis Computation and Related Problems. Ph.D. thesis. University of Waterloo. URL: http://hdl.handle.net/10012/7326.
Zhou, W., Labahn, G., 2013. Computing column bases of polynomial matrices, in: ISSAC'13, ACM, New York, NY, USA. pp. 379–386. doi:10.1145/2465506.2465947.
Zhou, W., Labahn, G., 2014. Unimodular completion of polynomial matrices, in: ISSAC'14, ACM. pp. 413–420. doi:10.1145/2608628.2608640.
Zippel, R., 1979. Probabilistic algorithms for sparse polynomials, in: EUROSAM'79, Springer. pp. 216–226. doi:10.1007/3-540-09519-5_73.