Interactive certificate for the verification of Wiedemann's Krylov sequence: application to the certification of the determinant, the minimal and the characteristic polynomials of sparse matrices

Jean-Guillaume Dumas, Erich Kaltofen, Emmanuel Thomé

Interactive certificate for the verification of Wiedemann's Krylov sequence: application to the certification of the determinant, the minimal and the characteristic polynomials of sparse matrices

Jean-Guillaume Dumas* Erich Kaltofen†
Emmanuel Thomé‡

July 3, 2015

Abstract

Certificates to a linear algebra computation are additional data structures for each output, which can be used by a—possibly randomized—verification algorithm that proves the correctness of each output. Wiedemann's algorithm projects the Krylov sequence obtained by repeatedly multiplying a vector by a matrix to obtain a linearly recurrent sequence. The minimal polynomial of this sequence divides the minimal polynomial of the matrix. For instance, if the $n \times n$ input matrix is sparse with $n^{1+o(1)}$ non-zero entries, the computation of the sequence is quadratic in the dimension of the matrix while the computation of the minimal polynomial is $n^{1+o(1)}$, once that projected Krylov sequence is obtained.

In this paper we give algorithms that compute certificates for the Krylov sequence of sparse or structured $n \times n$ matrices over an abstract field, whose Monte Carlo verification complexity can be made essentially linear. As an application this gives certificates for the determinant, the minimal and characteristic polynomials of sparse or structured matrices at the same cost.

1 Introduction

We consider a square sparse or structured matrix $A \in \mathbb{F}^{n \times n}$. By sparse or structured we mean that the multiplication of a vector by A requires less opera-

*Laboratoire J. Kuntzmann, Université de Grenoble. 51, rue des Mathématiques, umr CNRS 5224, bp 53X, F38041 Grenoble, France, Jean-Guillaume.Dumas@imag.fr, ljk.imag.fr/membres/Jean-Guillaume.Dumas.

†Department of Mathematics. North Carolina State University. Raleigh, NC 27695-8205, USA. kaltofen@math.ncsu.edu, www.kaltofen.us.

‡CARAMEL Project – INRIA Nancy Grand Est. 615 rue du Jardin Botanique–54602 Villiers-les-Nancy – France. Emmanuel.Thome@gmail.com, www.loria.fr/~thome/.

tions than that of a dense matrix-vector multiplication. The arithmetic cost to apply A is denoted by μ which thus satisfies $\mu \leq n(2n-1)$ (n^2 multiplications and $n(n-1)$ additions). In the following we also need to perform row-vector-times-matrix multiplications, which, by the transposition principle, cost $O(\mu)$ operations [3]. In the following we will simply consider that both operations (left or right multiplication by a row or column vector) cost less than μ arithmetic operations.

The main idea of this paper is to use a Baby-step/Giant-step verification of Wiedemann's Krylov sequence generation. Once the sequence is verified, the remaining operations, of lower cost, can be replayed by the Verifier.

The verification procedure used throughout this paper is that of *essentially optimal interactive certificates* with the taxonomy of [8]. Indeed, in the following, we consider a *Prover*, nicknamed *Peggy*, who will perform a computation, potentially together with additional data structures. We also consider a *Verifier*, nicknamed *Victor*, who will check the validity of the computation, faster that just by recomputing it.

By *certificates* for a problem that is given by input/output specifications, we mean, as in [15, 16], an input-dependent data structure and an algorithm that computes from that input and its certificate the specified output, and that has lower computational complexity than any known algorithm that does the same when only receiving the input. Correctness of the data structure is not assumed but validated by the algorithm.

By *interactive certificate*, we mean certificates modeled as $\sum$-protocols (as defined in [7]) were the Prover submits a *Commitment*, that is some result of a computation; the Verifier answers by a *Challenge*, usually some uniformly sampled random values; the Prover then answers with a *Response*, that the Verifier can use to convince himself of the validity of the commitment. To be useful, such proof systems is said to be *complete* if the probability that a true statement is rejected by the Verifier can be made arbitrarily small. Similarly, the protocol is *sound* if the probability that a false statement is accepted by the verifier can be made arbitrarily small. In the following we will actually only consider *perfectly complete* certificates, that is were a true statement is never rejected by the Verifier.

There two may ways to design such certificates. On the one hand, efficient protocols can be designed for delegating computational tasks. In recent years, generic protocols have been designed for circuits with polylogarithmic depth [13, 18]. The resulting protocols are interactive and their cost for the Verifier is usually only roughly proportional to the input size. They however can produce a non negligible overhead for the Prover and are restricted to certain classes of circuits. Variants with an amortized cost for the Verifier can also be designed, see for instance [17], quite often using relatively costly homomorphic routines. Moreover, we want the Verifier to run faster than the Prover, so we discard amortized models where the Verifier is allowed to do a large amount of precomputations, that can be amortized if, say, the same matrix is repeatedly used [5, 12].

On the other hand, dedicated certificates (data structures and algorithms

that are verifiable a posteriori, without interaction) have also been developed in the last few years, e.g., for dense exact linear algebra [11, 16, 10], even for problems that have no good circuit representation. There the certificate constitute a proof of correctness of a result, not of a computation, and can thus also stand a direct public verification. The obtained certificates are ad-hoc, but try to reduce as much as possible the overhead for the Prover, while preserving a fast verification procedure.

In the current paper we follow the later line of research, that is ad-hoc certificate with fast verification and negligible overhead for the Prover.

In exact linear algebra, the most simple problem to have an optimal certificate is the linear system solution, LinSolve: for a matrix A and a vector b, checking that x is actually a solution is done by one multiplication of x by A. The cost of this check similar to that of just enumerating all the non-zero coefficients of A. Thus certifying a linear system is reduced to multiplying a matrix by a vector: LinSolve$\prec$MatVecMult. In [8], two essentially optimal reductions have been made, that the rank can be certified via certificates for linear systems, and that the characteristic polynomial can be certified via certificates for the determinant: CharPoly$\prec$Det and Rank$\prec$LinSolve. But no reduction was given for the determinant. We bridge this gap in this paper. We first use Wiedemann's reduction of the determinant to the minimal polynomial of a sequence, Det$\prec$MinPoly$\prec$Sequence, [21], and then show that the computation of a sequence generated by projections of matrix-vector iterations can be checked by a small number of matrix-vector multiplications: Sequence$\prec$MatVecMult.

The complexity model we consider here is the algebraic complexity model: we count field operations, but tests (even such as checking the equality of whole vectors) are free and uniform sampling of random elements in a field is also free. This is justified by the fact that for all our proposed certificates, the number of equality tests is always lower than that of field operations and that the number of random samples is always lower than that of the communications, itself lower than that of the Verifier's work.

The paper is organized as follows. We define Wiedemann's Krylov sequence formally in Section 2. Then we use a check-pointing technique to propose a first non-quadratic certificate in Section 3. Then we derive from this technique a recursive process that can yield a method of decreasing complexities for the Verifier in Section 4. The same general idea is modified in Section 5 to get a certificate verifiable in essentially optimal time. Finally, we show in Section 6 how to derive certificates for the determinant, the minimal and the characteristic polynomial from these certificates for the Krylov sequence.

2 Wiedemann's Krylov sequence

We consider here the simple Wiedemann's sequence S (no blocks), defined for two given vectors.

Definition 1. *For $A \in \mathbb{F}^{n\times n}$, $V_0 \in \mathbb{F}^n$ and $U \in \mathbb{F}^n$,* Wiedemann's Krylov space *is defined for $i \geq 0$ as:*

$$K_{V_0} = (V_i)_i = (A^i V_0)_i$$

Wiedemann's Krylov sequence *is also defined as:*

$$S = (s[i])_i = (U^T A^i V_0)_i = (U^T V_i)_i$$

In the following, the Prover will compute this sequence, potentially together with additional data structures, and the Verifier will check the validity of the sequence, once computed.

Now, for a matrix A whose matrix-vector multiplication costs μ arithmetic operations, the original cost for the computation of $2n$ elements of Wiedemann's Krylov sequence is trivially:

$$W(n) = 2n\mu + 4n^2 = \mathcal{O}(n\mu).$$

We summarize in table 1, the complexity bounds for certificates of Wiedemann's Krylov sequence, presented in this paper .

Certificate	Verifier	Extra Communication	Prover
§ 3	$\mathcal{O}\left(n\sqrt{\mu}\right)$	$\mathcal{O}\left(n\sqrt{\mu}\right)$	$W(n)$
§ 4.1	$2\mu + \mathcal{O}\left(n\sqrt{n}\right)$	$\mathcal{O}\left(n\sqrt{n}\right)$	$W(n) + \mathcal{O}\left(\mu\sqrt{n}\right)$
§ 4.2	$4\mu + \mathcal{O}\left(n\sqrt[3]{n}\right)$	$\mathcal{O}\left(n\sqrt[3]{n}\right)$	$W(n) + \mathcal{O}\left(\mu n^{2/3}\right)$
§ 4.3	$2^k\mu + \mathcal{O}\left(n\sqrt[k]{n}\right)$	$\mathcal{O}\left(n\sqrt[k]{n}\right)$	$W(n) + o(W(n))$
§ 5	$\mathcal{O}\left(\mu\log^2(n)\right)$	$\mathcal{O}\left(n\log^2(n)\right)$	$5W(n)$
§ 5	$\mathcal{O}\left(\mu\log(n) + n\log^2(n)\right)$	$\mathcal{O}\left(n\log^2(n)\right)$	$7W(n)$

Table 1: Summary of the complexity bounds of the certificates presented in this paper for Wiedemann's Krylov sequence

3 An $n^{1+1/2}$ certificate

3.1 A four steps Baby-step/Giant-step interactive protocol

The protocol has four steps: Victor first selects the vectors for the sequence that are sent to Peggy. Peggy then computes the sequence and keeps some of the intermediate vectors, called checkpoints. She then sends the sequence to

Victor together with the additional intermediate vectors which Victor will use to certify the received sequence:

1. Communications from Victor to Peggy
   (a) Uniformly sample $V_0 \in \mathbb{F}^n$, $U \in \mathbb{F}^n$;
   (b) Sends A, U, V_0.
   (c) Asks for a sequence of $\delta + 1$ elements.
   (d) Asks for a checkpoint every $K < \min\{n, \delta\}$ matrix-vector products.

   Communication is $|A| + 2n \leq \mu + 2n$.

2. Computations of Peggy:
   (a) $V_i = AV_{i-1}$ for $i = 0..\delta$;
   (b) $s[i] = U^T V_i$ for $i = 0..\delta$.

   Complexity is exactly that of Wiedemann's sequence; that is $\mathcal{O}\left(n\mu + n^2\right)$ if $\delta = 2n$.

3. Communications from Peggy to Victor
   (a) Sends $W_j = V_{jK} = A^{jK}V_0$ for $j = 0..\lceil \frac{\delta}{K} \rceil$;
   (b) Sends $s[i]$ for $i = 0..\delta$.

   Communication is $n\lceil \frac{\delta}{K} \rceil + \delta + 1 = \mathcal{O}\left(\delta \frac{n}{K}\right)$.

4. Verifications of Victor.
   (a) Uniformly sample $R = (r[i]) \in \mathbb{F}^K$ and $X \in \mathbb{F}^n$, with $X \neq U$.

   Then first compute some baby steps:

   (b) Compute $Z = X^T A^K$, in $K\mu$ operations;
   (c) Compute $T = \sum_{i=0}^{K-1} r[i]U^T A^i$ in $(K-1)\mu + 2Kn$ operations.

   For each $j = 1..\lceil \frac{\delta}{K} \rceil$;

   (e) $X^T W_j \stackrel{?}{=} ZW_{j-1}$ in $2 \times 2n + n$ operations; // Checks the W_j with X
   (f) $\sum_{i=0}^{K-1} r[i]s[jK + i] \stackrel{?}{=} TW_j$; in $2K + 2n + 1$ operations. // Checks the $s[i]$ with R once W_j is certified

The overall complexity of the verification step is bounded by:

$$2K(\mu + n) + \left\lceil \frac{\delta}{K} \right\rceil (2K + 6n). \tag{1}$$

Lemma 1. *The above protocol is perfectly complete.*

Proof. 4e: $X^TW_j = X^TV_{jK} = X^TA^{jK}V_0 = X^TA^KA^{(j-1)K}V_0$, so that we also have $X^TW_j = X^TA^KV_{(j-1)K} = Z^TW_{j-1}$.

4f: $r[i]s[jK+i] = r[i]U^TV_{jK+i} = r[i]U^TA^iV_{jK} = r[i]U^TA^iW_j$;

□

3.2 Optimal Verifier complexity

Theorem 1. *Let $A \in \mathbb{F}^{n\times n}$ whose matrix-vector product can be computed in less than $\mu > n$ arithmetic operations and a vector $V_0 \in \mathbb{F}^n$. There exists a certificate of size:*

$$\frac{1}{\sqrt{3}}\sqrt{\delta n(\mu+n)}$$

for the $\delta+1$ first elements of Wiedemann's Krylov sequence associated to A and V_0. This certificate is verifiable in time:

$$4\sqrt{3}\sqrt{\delta n(\mu+n)}.$$

With $\mu = n^{1+o(1)}$, and $\delta = 2n$, this is a Verifier in $n^{1.5+o(1)}$ time and communications.

Proof. The optimal value of K minimizes Equation (1) and is therefore close to:

$$K \approx \sqrt{3}\sqrt{\frac{n\delta}{\mu+n}}$$

Substituting the latter into Equation (1) gives the announced time complexity. For the size of the certificate, apart from the matrix A itself, the additional communications are the initial vectors sent by Victor and the intermediate checkpointing vectors sent by Peggy. Once again substituting the value for K gives the announced complexity. □

We ran this choice on a very sparse matrix with 3 non zero elements per row. Results are shown in Table 2: computing the sequence took two hours, the thousand W checkpoints required about two giga bytes of data, and were checked in a little more than half a minute.

Prover	Communications	Verifier			
		Z	Check Z	T	Check T
1.8 hours	1.9 GB	5.6 s	14.5 s	7.0 s	6.0 s

Table 2: Verification for a matrix with $m = n = 253008$, 759022 non-zeroes and of compressed size of 3.8MB. This is 506046 iterations, and $K = 503$ was chosen on one core of an i5-4690 @3.50GHz

3.3 Soundness

For the soundness, we need to sample from a finite subset $\mathbb{S}$ of $\mathbb{F}$.

Theorem 2. *If the Verifier samples R and X uniformly and independently from a finite subset $\mathbb{S} \subseteq \mathbb{F}$, then the Verifier mistakenly misses any error in the sequence or in the check-pointing vectors with probability $\leq 1/|\mathbb{S}|$.*

Proof.

1. $W_0 = V_0$ is given. Thus, inductively, Peggy must find W_j for each $j \geq 1$ such that $M_j = W_j - A^K W_{j-1}$ satisfies $X^T M_j = 0$, for a random secret X unknown to her. If M_j is non zero then there is $1/|\mathbb{S}|$ chances that the dot-product is zero.

2. Afterwards, let Θ_j be the vector of $\Theta_j[i] = U^T A^i W_j = U^T A^{jK+i} V_0$. W_j being correct, Peggy must find a vector Δ_j with $\Delta_j[i] = s[jK + i]$ such that $N_j = \Delta_j - \Theta_j$ satisfies $R^T N_j = 0$, for a random secret R unknown to her. If N_j is non zero there is $1/|\mathbb{S}|$ chances that the dot-product is zero.

□

To improve probability, as usual, it is possible to rerun the protocol with some other vectors X and R, ...

3.4 Public verifiability

The protocol is publicly verifiable. Indeed, no response from the Prover is requested after the selection of the challenge X and R. Therefore, any external participant can also generate its own X and R and re-check the Krylov space vectors and Wiedemann's sequence, at the cost given in Theorem 1.

3.5 Constants for block Wiedemann's algorithm

It is possible to use the same protocol to check the matrix sequence produced in the block Wiedemann's algorithm [6] with a projection of s_1 vectors on the left and s_2 vectors on the right. The following modifications have to be made, mainly replacing some vectors by blocks of vectors:

- $U \in \mathbb{F}^{n \times s_1}$, $V_i \in \mathbb{F}^{n \times s_2}$, $W_j \in \mathbb{F}^{n \times s_2}$, $S[i] \in \mathbb{F}^{s_1 \times s_2}$;
- X and Z remain in $\mathbb{F}^n$ while $R \in \mathbb{F}^{K \times s_1}$ and $r[i]$ is in fact the transpose of a vector in $\mathbb{F}^{s_1}$;
- and $T \in \mathbb{F}^{s_1 \times n}$.

The length of the sequence is now $\ell = \frac{n}{s_1} + \frac{n}{s_2} + \mathcal{O}(1)$ [14, 19].

1. Communications become: $\lceil \frac{\ell}{K} \rceil (n s_2) + \ell s_1 s_2$.
2. Verifications become: $(K\mu) + K(s_1\mu + 2s_1 n + n) + \lceil \frac{\ell}{K} \rceil (4ns_2 + K(2s_1 s_2 + s_2) + 2ns_1 s_2)$

Now the optimal K becomes:

$$K \approx \sqrt{1+\frac{2}{s_1}}\sqrt{\frac{s_2}{s_1}}\sqrt{\frac{\ell n}{\mu(\frac{1}{2}+\frac{1}{2s_1})+n}}$$

As $(\frac{1}{2}+\frac{1}{2s_1}) \leq 1$, this is a Verifier in time bounded by:

$$2\sqrt{s_2}\sqrt{s_1+2}(s_1+1)\sqrt{\ell n(\mu+n)}+2\ell s_1 s_2$$

With $\mu = n^{1+o(1)}$ and $s_1 = s_2 = s$, the length of the sequence is $\ell \approx 2\frac{n}{s}$ so that the Verifier time becomes $(sn)^{1+1/2+o(1)}$.

4 Recursive verification

In fact, in the verification steps of Victor, in the protocol of Section 3, it is possible to also delegate the computation of Z and T.

4.1 Denser matrices, Verifier in time $2\mu + n^{1+1/2+o(1)}$

Next, we propose to delegate just the matrix-vector operations, so that we get a good complexity also on matrices with more than $n^{1+o(1)}$ entries. The idea is that the Verifier can delegate his computations of several successive matrix vector product and check the whole list of computed vectors. Therefore he replaces matrix-vector products by checks of validity of vectors. The trick is that verifying a vector can be done with a single dot-product of cost $2n$, while multiplying a matrix by a vector costs μ operations.

This way, correctness of a full Krylov space can be checked as given in Algorithm 1.

Algorithm 1 Checking the Krylov Space

Require: a matrix $A \in \mathbb{F}^{n\times n}$ and a vector $V_0 \in \mathbb{F}^n$;
Require: a list of d vectors $[V_0, V_1, \ldots, V_{d-1}]$;
Ensure: $[V_0, V_1, \ldots, V_{d-1}] = [V_0, AV_0, \ldots, A^{d-1}V_0]$.
1: For $\mathbb{S} \subseteq \mathbb{F}$, uniformly sample $Y \in \mathbb{S}^n$;
2: Compute $H = Y^T A$;
3: **return** $HV_{i-1} \stackrel{?}{=\!=} Y^T V_i$, for $i = 1..d$.

Lemma 2. *Algorithm 1 is sound, perfectly complete and requires $\mu + 4dn$ arithmetic operations.*

Proof. Perfect completeness is granted inductively because V_0 is known and then since $HV_{i-1} = Y^T AA^{i-1}V_0 = Y^T A^i V_0 = Y^T V_i$. Soundness is granted because whenever $AV_{i-1} - Vi \neq 0$, its dot-product with a uniformly selected $Y \in \mathbb{S}^n$ will be zero only with probability $|\mathbb{S}|^{-1}$. Complexity for the Verifier is μ operations to compute H and then d checks performed by two dot-products of size n. □

The idea is to delegate the computation of both Z (Point 4b of the protocol of section 3) and T (Point 4c of the protocol of section 3). Then to only check both resulting Krylov spaces. Note that it is mandatory that this delegation of the computation of Z and T takes place *after* the commitment of the W_j and the $s[i]$ by the Prover.

In the complexity of Theorem 1, this replaces two $K\mu$ factors (now an additional, but neglectible, cost to the Prover), each by a $\mu + 4Kn$ factor. This gives a new complexity of $2\mu + 10Kn + \left\lceil \frac{\delta}{K} \right\rceil (2K + 6n)$ for the Verifier. There are some extra communications, the vectors used for the computation of Z and T, namely $2n(K-1)$ field elements. We have proven:

Corollary 1. *Let $A \in \mathbb{F}^{n\times n}$ whose matrix-vector product can be computed in less than $\mu > n$ arithmetic operations and a vector $V_0 \in \mathbb{F}^n$. For any $1 \leq K \leq min\{n, \delta\}$, there exists a sound and perfectly complete protocol verifying the first $\delta+1$ elements of Wiedemann's Krylov sequence associated to A and V_0, in time $2\mu + 10Kn + \left\lceil \frac{\delta}{K} \right\rceil (2K + 6n)$. The associated certificate has size $n \left\lceil \frac{\delta}{K} \right\rceil + 2nK$.*

The extra work for the Prover is that of the computation of Z and T, both bounded by $\mathcal{O}(\mu K) = \mathcal{O}\left(\mu n^{2/3}\right)$, negligible with respect to the computation of the sequence, $\mathcal{O}(\mu n)$.

In terms of computational time for the verifier, the associated optimal K factor becomes $K = \sqrt{\frac{3}{5}}\sqrt{\delta}$ and the Verifier complexity is transformed into:

$$4n + 2\mu + 4n\sqrt{15\delta}.$$

With $\delta = 2n$, this gives a Verifier complexity bounded by $2\mu + 21.91n^{1.5}$, with a certificate of size bounded by $4.02n^{1.5}$.

4.2 Optimal 2-levels of recursion and an $n^{1+1/3}$ certificate for Wiedemann's algorithm

Now, instead of just delegating the matrix-vector products, we delegate the whole computation of Z and T:

1. For Z, it is actually sufficient to reuse the scheme of Section 4.1 with $\delta = K$, choosing a $K_2 < K$, and Z will be certified as the last W_j vector. The time for the Verifier for this step is thus bounded by $2\mu + 10nK_2 + \frac{K}{K_2}(2K_2 + 6n)$.
2. For T, the protocol is twofold:
   (a) Send the $r[i]$, U and A, and ask just for T in return;
   (b) Only now, send a uniformly sampled vector Ψ and ask for a certificate of the sequence $\Gamma = \gamma[i] = U^T A^i \Psi$;
   (c) Then one can check that $\sum r[i]\gamma[i] \stackrel{?}{=} T\Psi$.

Theorem 3. *For $A \in \mathbb{F}^{n\times n}$ whose matrix-vector product can be computed in less than $\mu > n$ arithmetic operations and a vector $V_0 \in \mathbb{F}^n$, there exists a sound and perfectly complete interactive certificate for the associated Wiedemann's Krylov sequence of size $\mathcal{O}\left(n^{1+1/3}\right)$. This certificate is verifiable in time*

$$4\mu + \mathcal{O}\left(n^{1+1/3}\right).$$

Proof. We still use the protocol of Section 3, but replace the computation of Z and T by the above delegated scheme.

The protocol is perfectly complete, since $\sum r[i]\gamma[i] = \sum r[i](U^T A^i \Psi) = \sum (r[i]U^T A^i)\Psi = T\Psi$.

The protocol is sound because the $\gamma[i]$ are correctly verified by a sound protocol. Then Ψ being unknown when asking for T, T cannot be engineered to satisfy the last check: if $G = T - \sum r[i](U^T A^i)$ is non zero then there is $1/|\mathbb{S}|$ chances that its dot-product with Ψ is zero.

Verifier time and space for T are that of Corollary 1 for the sequence Γ, and a supplementary dot-product. Verifier time and space for Z are also that of Corollary 1. Therefore, since $6n + 2K \leq 8n$, overall, the Verifier runs now in time bounded by:

$$2\left(2\mu + 10nK_2 + 8n\frac{K}{K_2}\right) + 2n + 8n\left\lceil\frac{\delta}{K}\right\rceil = 4\mu + \mathcal{O}\left(nK_2 + n\frac{K}{K_2} + n\frac{\delta}{K}\right) \quad (2)$$

with a certificate of size bounded by:

$$\mathcal{O}\left(n\frac{\delta}{K} + n\frac{K}{K_2} + nK_2\right).$$

With $\delta = 2n$, optimal values for K and K_2 are now respectively $n^{2/3}$ and $n^{1/3}$ for a Verifier in time $4\mu + \mathcal{O}\left(n^{1+1/3}\right)$ with a certificate of size $\mathcal{O}\left(n^{1+1/3}\right)$. □

The extra work for the Prover is that of the computation of Z and T both bounded by $\mathcal{O}(\mu K) = \mathcal{O}\left(\mu n^{2/3}\right)$, of Γ (if done together with that of T, this requires only K dot-products), and of the Zs and Ts for the verifications of Z, T and Γ. Those are bounded by $\mathcal{O}(\mu K_2) = \mathcal{O}\left(\mu n^{1/3}\right)$. All this is negligible with respect to the computation of the sequence, $\mathcal{O}(\mu n)$.

4.3 More levels and a Verifier in time $n^{1+1/k+o(1)}$

Once it is proven that the computation of Z and T can be delegated, then the computation of Z_2 and T_2 in their verification can also be delegated. The idea, is thus to use the protocol of section 4.2, also for Z and T, but with two parameters K_1 and K_2 to set and $\delta = K$ in equation (2). The verification time for Z and T becomes $4\mu + \mathcal{O}\left(nK_2 + n\frac{K_1}{K_2} + n\frac{K}{K_1}\right)$ for each and, overall, the Verifier thus runs now in time bounded by:

$$8\mu + \mathcal{O}\left(\left(nK_2 + n\frac{K_1}{K_2} + n\frac{K}{K_1}\right) + n\frac{\delta}{K}\right). \quad (3)$$

With $K_2 = n^{\alpha_2}$, $K_1 = n^{\alpha_1}$, $K = n^{\beta}$, the optimal values should equal $1 + \alpha_2 = 1 + \alpha_1 - \alpha_2 = 1 + \beta - \alpha_1 = 2 - \beta$, or differently written, $2\alpha_2 - \alpha_1 = 0; -\alpha_2 + 2\alpha_1 - \beta = 0; -\alpha_1 + 2\beta = 1$. This yields $[\alpha_2 = 1/4, \alpha_1 = 1/2, \beta = 3/4]$, so that $K_2 = n^{1/4}$, $K_1 = n^{1/2}$, $K = n^{3/4}$ and the complexity is bounded:

$$8\mu + \mathcal{O}\left(n^{1+1/4}\right).$$

As previously, the size of the certificate is also reduced to $\mathcal{O}\left(n^{1+1/4}\right)$ and the extra work for the Prover is increased to $\mathcal{O}\left(\mu n^{3/4}\right)$, still negligible with respect to $\mathcal{O}\left(\mu n\right)$.

More generally, for any k, we have

$$\begin{bmatrix} 2 & -1 & 0 & 0 & \dots & 0 \\ -1 & 2 & -1 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & -1 & 2 & -1 \\ 0 & \dots & \dots & 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} \alpha_{k-2} \\ \alpha_{k-3} \\ \vdots \\ \alpha_2 \\ \alpha_1 \\ \beta \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ \vdots \\ \vdots \\ 0 \\ 1 \end{bmatrix} \tag{4}$$

For L a unit lower triangular matrix, the latter gives, via Gaussian elimination without pivoting:

$$\begin{bmatrix} 2 & -1 & 0 & 0 & \dots & 0 \\ 0 & \frac{3}{2} & -1 & 0 & \dots & 0 \\ 0 & \ddots & \frac{4}{3} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & 0 & \frac{k-1}{k-2} & -1 \\ 0 & \dots & \dots & 0 & 0 & \frac{k}{k-1} \end{bmatrix} \begin{bmatrix} \alpha_{k-2} \\ \alpha_{k-3} \\ \vdots \\ \alpha_2 \\ \alpha_1 \\ \beta \end{bmatrix} = L^{-1} \begin{bmatrix} 0 \\ \vdots \\ \vdots \\ \vdots \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ \vdots \\ \vdots \\ 0 \\ 1 \end{bmatrix} \tag{5}$$

So that the solution is:

$$\begin{bmatrix} \alpha_{k-2} & \alpha_{k-3} & \dots & \alpha_2 & \alpha_1 & \beta \end{bmatrix} = \begin{bmatrix} \frac{1}{k} & \frac{2}{k} & \dots & \frac{k-2}{k} & \frac{k-1}{k} \end{bmatrix} \tag{6}$$

Thus $n^{1+\alpha_{k-2}} = n^{1+\alpha_{k-3}-\alpha_{k-2}} = \ldots = n^{1+\beta-\alpha_1} = n^{2-\beta} = n^{1+1/k}$.

The size of the certificate is thus $\mathcal{O}\left(n^{1+1/k}\right)$, the time for the Verifier is $2^k\mu + \mathcal{O}\left(n^{1+1/k}\right)$ and the extra work for the Prover becomes $\sum_{t=1}^{k} 2^t \mu n^{1-t/k} = \mathcal{O}\left(\mu n^{1-1/k}\right)$, still negligible with $\mathcal{O}\left(\mu n\right)$.

5 $\mu \log(n) + n \log^2(n)$ certificate

The same idea actually gives rise to a certificate verifiable with only $\log_2(n)$ matrix-vector products: use a recursive certificate with $K = \delta/2$.

We first need to separate the interactive protocol of Section 4.2 into atomic parts: a recursive interactive protocol for certifying a single vector corresponding to a large power of A times an initial vector and a combination of mutually recursive protocols for the sequence.

5.1 Certificate for the large powers

We want here to certify that $Z \stackrel{?}{=} A^d V$. For this we will need to check successive powers of two.

5.1.1 Certificate for the large powers with a logarithmic number of matrix-vector products

We define the certificate $\texttt{PowerCertificate}(A, V, d)$ to be two vectors $Z, Z_{/2}$ that satisfy $Z \stackrel{?}{=} A^d V$ and $Z_{/2} \stackrel{?}{=} A^{\lfloor d/2 \rfloor} V$. Then checking this certificate is shown in algorithm 2.

Algorithm 2 Logarithmic Interactive recursive check of $\texttt{PowerCertificate}(A, V, d)$

Require: Matrix $A \in \mathbb{F}^{n \times n}$, vector $V \in \mathbb{F}^n$, exponent d;
Require: A pair of vectors $Z, Z_{/2} = \texttt{PowerCertificate}(A, V, d)$.
Ensure: $Z \stackrel{?}{=} A^d V$ **and** $Z_{/2} \stackrel{?}{=} A^{\lfloor d/2 \rfloor} V$.

```
if d == 1 then
    return Z_{/2} ?= V and Z ?= AV.
else
    Uniformly sample W ∈ F^n;
    Request (Y, Y_{/2}) = PowerCertificate(A^T, W, ⌊d/2⌋) and recursively check it;
    if d is even then
        return W^T Z_{/2} ?= Y^T V and W^T Z ?= Y^T Z_{/2}
    else
        return W^T Z_{/2} ?= Y^T V and W^T Z ?= Y^T (A Z_{/2}).
    end if
end if
```

Lemma 3. *Algorithm 2 is sound and perfectly complete. It requires* $\log_2(d)$ *rounds,* $3\log_2(d)n$ *communications,* $2d\mu$ *arithmetic operations for the Prover, and less than* $(\mu + 8n)\log_2(d) + \mu$ *arithmetic operations for the Verifier.*

Proof. The protocol is perfectly complete by induction: the basis of the induction is given by the case $d == 1$; then by induction $Y = (A^T)^{\lfloor d/2 \rfloor} W$, so that:

$$W^T Z_{/2} = W^T A^{\lfloor d/2 \rfloor} V = (W^T A^{\lfloor d/2 \rfloor}) V = Y^T V$$

and, if d is even:

$$W^T Z = W^T A^d V = (W^T A^{\lfloor d/2 \rfloor})(A^{\lfloor d/2 \rfloor} V) = Y^T Z_{/2}$$

or, if d is odd:

$$W^T Z = W^T A^d V = (W^T A^{\lfloor d/2 \rfloor}) A (A^{\lfloor d/2 \rfloor} V) = Y^T A Z_{/2}.$$

The protocol is sound: the Prover produces the commitments Z and $Z_{/2}$, then the Verifier sends a challenge W and the Prover responds with Y. There, the Prover has two possibilities, either he returns a correct Y or not. In the first case, as W was chosen uniformly at random, there are two sub case, either $Z_{/2}$ is wrong or not. if $Z_{/2}$ was incorrectly chosen so that $Z_{/2} - A^{\lfloor d/2 \rfloor} V$ is non zero, there is $1/|\mathbb{S}|$ chances that its dot-product with W^T is zero and thus that it can pass the $W^T Z_{/2} \stackrel{?}{=} Y^T V$ check. Conversely, if $Z_{/2}$ is correct, if Z was incorrectly chosen so that $Z - A^{\lceil d/2 \rceil} Z_{/2}$ is non zero, there is $1/|\mathbb{S}|$ chances that its dot-product with W^T is zero. Both tests are not independent but overall there are less than $1/|\mathbb{S}|$ chances to pass both of them. In the second case Y, Y is incorrect but can very well be made to make both latter dot-products zero, for any values of Z, $Z_{/2}$ and W. But if Y is incorrect, it will not pass the recursive test if $\lfloor d/2 \rfloor = 1$, and will pass it only with probability $|\mathbb{S}|^{-1}$ for other values of d. Therefore, if Peggy's commitment was incorrect, the probability that it passes all the subsequent tests of Algorithm 2 is less than $|\mathbb{S}|^{-1}$.

Now, Communication is that of the certificate, the 3 vectors W, Y and $Y_{/2}$, per recursive call, that is $3\log_2(d)n$. Time complexity for the Verifier satisfies $\{T(d) \leq T(d/2)+2*4n+\mu, T(1) = \mu\}$, that is less than $(\mu+8n)\log_2(d)+\mu$. Now the cost has been transferred to the prover, who has to compute the sequence plus half a sequence, plus a fourth of a sequence, ..., recursively the overall cost for the Prover is doubled to $2d\mu$. □

5.1.2 Public verifiability of the large power

Another view of the verification of Algorithm 2 can be given as an interactive certificate in Figure 1.

As the challenge is only random samples selected after the commitment (and this is true also recursively), Fiat-Shamir heuristic can be used at each step [9, 1, 2]: W can be just the result of a cryptographically strong hash function on A, V, d, *and* Z, $Z_{/2}$. Then any external verifier can simulate the whole protocol by recomputing also the hashes.

5.1.3 Certificate for the large powers with a single matrix-vector product

Actually, algorithm 2 can be made to require a *single* matrix-vector product. The speed up for the verifier is obtained by recursively asking for a little more: some arithmetic cost for the Verifier is traded-off with an extra cost for the Prover and some extra communications.

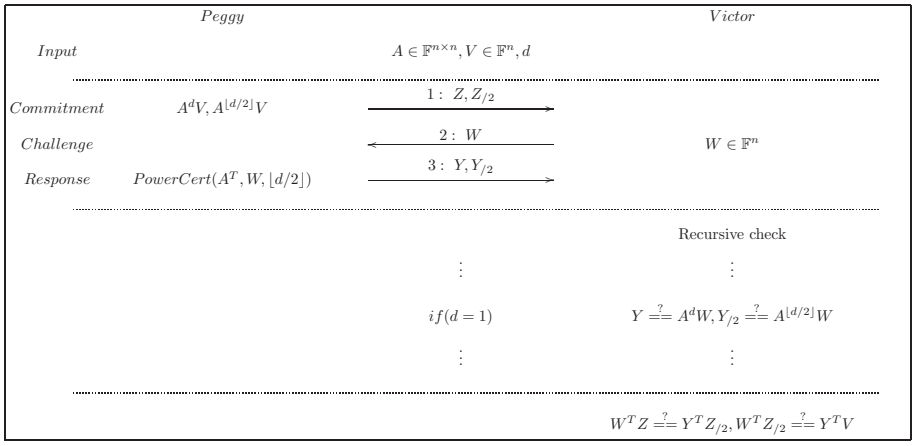

Figure 1: Interactive certificate for A^dV

The certificate $\texttt{PowerCertificate}(A, V, d)$ is modified to be *three* vectors: for any t such that $2^t \geq d$, we check A^dV, together with $A^{2^t}V$ and $A^{2^{t-1}}V$.

Algorithm 3 Interactive recursive check of $\texttt{PowerCertificate}(A, V, d, 2^t)$

Require: Matrix $A \in \mathbb{F}^{n\times n}$, vector $V \in \mathbb{F}^n$, exponent $d \geq 2$, t such that $2^t \geq d$;
Require: A triple of vectors $Z_t, Z, Z_{t-1} = \texttt{PowerCertificate}(A, V, d, 2^t)$.
Ensure: $Z_{t-1} \stackrel{?}{=} A^{2^{t-1}}V$ **and** $Z \stackrel{?}{=} A^dV$ **and** $Z_t \stackrel{?}{=} A^{2^t}V$.

1: Uniformly sample $W \in \mathbb{F}^n$;
2: **if** $d == 2$ **then**
3: Compute $Y = A^TW$;
4: **return** $W^TZ_0 \stackrel{?}{=} Y^TV$ **and** $W^TZ \stackrel{?}{=} Y^TZ_0$ **and** $Z_1 \stackrel{?}{=} Z$.
5: **else**
6: Request $(Y_{t-1}, Y, Y_{t-2}) = \texttt{PowerCertificate}(A^T, W, d - 2^{t-1}, 2^{t-1})$ and recursively check it;
7: **return** $W^TZ_{t-1} \stackrel{?}{=} Y^TV$ **and** $W^TZ \stackrel{?}{=} Y^TZ_{t-1}$ **and** $W^TZ_t \stackrel{?}{=} Y_{t-1}^TZ_{t-1}$.
8: **end if**

Lemma 4. *Algorithm 3 is sound and perfectly complete. It requires* $\log_2(d)$ *rounds,* $4\log_2(d)n$ *communications,* $2^{t+1}\mu$*, less than* $4d\mu$ *arithmetic operations for the Prover, and less than* $\mu + 8n + 12n\log_2(d)$ *arithmetic operations for the Verifier.*

The proof is similar to that of Lemma 3.

5.2 Certificate for the sequence

Now the idea is to use the protocol of Section 3, with $K = \delta/2$, but with the computations of Z and T completely delegated. The computation of Z can be verified, using either one of the `PowerCertificate`$(\ldots)$ protocols of Section 5.1. Wiedemann's Krylov sequence and T will then be verified with two distinct protocols, mutually recursive:

- For the sequence, with $K = \delta/2$, the verification loop of point 4e is reduced to the verification of two checkpoint vectors $(W, W_{/2})$ and of two parts of the sequence $S = (s[i]) = (s_H, s_L)$. Thus the data structure `SequenceCertificate`(U, A, V, d) is a combination of two vectors $(W, W_{/2})$, a sequence $S = (s[i])$ and two other certificates, one for Z: `PowerCertificate`$(A^T, X, d/2)$ and the second one for the linear combination T: `CombinationCertificate`$(R, U, A, d/2)$, for uniformly sampled X and R. The checkpoint vectors satisfy $W \stackrel{?}{=} A^d V$ and $W_{/2} \stackrel{?}{=} A^{\lfloor d/2 \rfloor} V$, and the output sequence satisfies the expected $S = (s[i]) = (s_H, s_L) \stackrel{?}{=} U^T A^i V$ for $i = 0..d$.
- For the delegation of T, it is sufficient to generate a certified sequence with another right projection. Thus, `CombinationCertificate`(R, U, A, d) is a combination of the vector T, that must satisfy as expected $T \stackrel{?}{=} \sum r[i] U^T A^i$ and of another certificate, `SequenceCertificate`(U, A, Ψ, d), for a uniformly sampled Ψ.

Checking these two certificates is done by using the following two mutually recursive procedures, shown in algorithms 4 and 5.

Theorem 4. *Let $A \in \mathbb{F}^{n \times n}$ whose matrix-vector product can be computed in less than $\mu > n$ arithmetic operations and a vector $V_0 \in \mathbb{F}^n$. There exists a certificate of size $\mathcal{O}(n \log(n))$ for the $\delta + 1$ first elements of Wiedemann's Krylov sequence associated to A and V_0. This certificate can be checked using the protocol of Algorithm 4. Depending on the `PowerCertificate`$(\ldots)$ routine chosen, the constant factor of this size and the Prover and Verifier arithmetic complexity bounds for this protocol are given in table 3.*

Power Certificate	Verifier	Extra Communication	Prover
§ 5.1.1	$\frac{1}{2}\mu \log_2^2(n) + 4n \log_2^2(n)$	$\frac{3}{2} n \log_2^2(n)$	$5W(n)$
§ 5.1.3	$\mu \log_2(n) + 6n \log_2^2(n)$	$2n \log_2^2(n)$	$7W(n)$

Table 3: Dominant terms of the complexity bounds for the verification of Wiedemann's Krylov sequence depending on the certification of $Z \stackrel{?}{=} A^d V$.

Proof. The protocol is sound and perfectly complete by induction on the size of sequence: the case $d == 1$ in Algorithm 4 gives the base of the induction;

Algorithm 4 Interactive check of $\texttt{SequenceCertificate}(U, A, V, d)$

Require: Matrix $A \in \mathbb{F}^{n\times n}$, two vectors $U, V \in \mathbb{F}^n$, sequence length $d+1$ with $d \geq 2$;
Require: A pair of vectors $W, W_{/2} \in \mathbb{F}^n$;
Require: A sequence $(s[i]) \in \mathbb{F}^{d+1}$.
Ensure: $W = A^{2\lceil \frac{d}{2} \rceil} V$ **and** $W_{/2} = A^{\lceil \frac{d}{2} \rceil} V$;
Ensure: $s[i] \stackrel{?}{=} U^T A^i V$ for $i = 0..d$.

1: **if** d==2 **then**
2: **return** $s[0] \stackrel{?}{=} U^T V$ **and** $W_{/2} \stackrel{?}{=} AV$ **and** $s[1] \stackrel{?}{=} U^T W_{/2}$ **and** $W \stackrel{?}{=} AW_{/2}$ **and** $s[2] \stackrel{?}{=} U^T W$.
3: **else**
4: Uniformly sample $X \in \mathbb{F}^n$;
5: Ask for $(Z, \ldots) = \texttt{PowerCertificate}(A^T, X, \lceil d/2 \rceil)$ and check it;
6: Let $first \leftarrow X^T W_{/2} \stackrel{?}{=} Z^T V$;
7: Let $second \leftarrow X^T W \stackrel{?}{=} Z^T W_{/2}$;
8: Uniformly sample $R \in \mathbb{F}^{\lceil \frac{d}{2} \rceil + 1}$;
9: Ask for $(T, \ldots) = \texttt{CombinationCertificate}(R, U, A, \lceil \frac{d}{2} \rceil)$ and check it;
10: Let $s_L = (s[0], \ldots, s\left[\lceil \frac{d}{2} \rceil\right])$ and $third \leftarrow R^T s_L \stackrel{?}{=} T^T W_{/2}$;
11: Let $s_H = (s\left[\lfloor \frac{d}{2} \rfloor\right], \ldots, s[d])$ and $fourth \leftarrow R^T s_H \stackrel{?}{=} T^T W$;
12: **return** $first$ **and** $second$ **and** $third$ **and** $fourth$.
13: **end if**

then the four explicit checks are correct thanks to Lemma 1 and sound thanks to Theorem 2; $\texttt{PowerCertificate}(\ldots)$ is correct and sound by Lemma 3 or Lemma 4; and $\texttt{CombinationCertificate}(\ldots)$ is correct and sound, first by induction on $\texttt{SequenceCertificate}(\ldots)$ with half the initial size, and second, since the explicit check is correct and sound by Theorem 3.

Complexity for the Verifier of the $\texttt{SequenceCertificate}(\ldots)$ sequence satisfies

$$\{\texttt{SequenceCertificate}(d) = \texttt{PowerCertificate}(d/2) + \texttt{CombinationCertificate}(d/2) + 12n + 2d,$$
$$\texttt{SequenceCertificate}(2)31 = 2\mu + 6n\}.$$

Complexity for the Verifier of T satisfies $\{\texttt{CombinationCertificate}(x) = \texttt{SequenceCertificate}(x) + 2n + 2x\}$.

With $\texttt{PowerCertificate}(x) = (\mu + 8n)\log_2(x) + \mu$ (see Lemma 3), the dominant terms of the complexity bound for the Verifier is thus:

$$\texttt{SequenceCertificate}(d) = \frac{1}{2}\mu \log_2^2(d) + 4n \log_2^2(d)$$

Similarly, with $\texttt{PowerCertificate}(x) = \mu + 8n + 12n\log_2(x) + \mu$ (see Lemma 4)

Algorithm 5 Interactive check of $\texttt{CombinationCertificate}(R, U, A, d)$

Require: Matrix $A \in \mathbb{F}^{n\times n}$, two vectors $R \in \mathbb{F}^{d+1}$ and $U \in \mathbb{F}^n$, sequence length $d+1$;
Require: A vector $T \in \mathbb{F}^n$.
Ensure: $T \stackrel{?}{=} \sum_{i=0}^{d} r[i]U^T A^i$.
1: Uniformly sample $\Psi \in \mathbb{F}^n$;
2: Ask for $(\Gamma, \ldots) = \texttt{SequenceCertificate}(U, A, \Psi, d)$ and check it;
3: **return** $R^T\Gamma \stackrel{?}{=} T\Psi$.

we get:

$$\texttt{SequenceCertificate}(d) = \mu \log_2(d) + 6n \log_2^2(d)$$

With $d = 2n$ we obtain the Verifier column of Table 3.

Similarly, communication is dominated either by $\frac{3}{2}n\log_2^2(d)$ or $2n\log_2^2(d)$.

The Prover has to compute the Krylov space and the Krylov sequence plus the work for Z, the work for T and the recursive calls: $P(d) = (d\mu + 2dn) + \texttt{PowerCertificate}(d/2) + ((d/2)\mu + 2(d/2)n + P(d/2))$, so that the overall extra cost for the Prover is dominated by either $5d\mu + 6dn$ or $7d\mu + 6dn$. For $d = 2n$, the cost for the Prover without verification is $W(()n) = 2n\mu + 4n^2$, which induces the last column of Table 3. □

6 Certificate for the determinant, the minimal and the characteristic polynomials

We denote by SeqCert a certificate for Wiedemann's Krylov sequence. This can be for instance any of the subquadratic certificate of Sections 3, 4 or 5.

This induces directly a certificate for the minimal polynomial of a sequence: the Prover just produces the sequence, and the Verifier computes by himself the minimal polynomial of the sequence via the fast extended Euclidean algorithm (EEA). In a sufficiently large field, Wiedemann has shown that this in turn induces a certificate for the minimal polynomial of a matrix, MinPoly. In smaller fields one would need to use a certificate for a Block Wiedemann sequence, and maybe some variants of the certificate of Section 3.5. Then a certificate for the determinant, Det, is obtained via Wiedemann's preconditioning, PreCondCyc, insuring the square-freeness of the characteristic polynomial. Finally, to get a certificate for the characteristic polynomial of a matrix, CharPoly, first ask for the characteristic polynomial, and then it is sufficient to certify the determinant at a random point.

We propose in Table 4 a summary of the reductions presented in this section. The details of these reductions and the proofs of the complexity claims shown in Table 4 are given in Theorems 5, 6 and 7.

MinPoly	
Verifier	Verify(SeqCert)+EEA
Communication	Communicate(SeqCert)+$2n$
Prover	Compute(SeqCert)
Det	
Verifier	Verifier(MinPoly)
Communication	Communicate(MinPoly+PreCondCyc)
Prover	Compute(MinPoly+PreCondCyc)
CharPoly	
Verifier	Verify(Det)+$2n$
Communication	Communicate(Det)+n
Prover	Compute(CharPoly)+Compute(Det)

Table 4: Summary of the complexity reductions for the certification of the determinant, the minimal and the characteristic polynomials of sparse matrices

6.1 MinPoly

Theorem 5 ([21])**.** *Certifying the minimal polynomial can be reduced to the certification of Wiedemann's Krylov sequence.*

Proof. The minimal polynomial of a linearly recurrent sequence can be computed by the fast Euclidean algorithm, see, e.g., [20, Theorem 12.10]. Then Wiedemann's analysis shows that in a sufficiently large field the minimal polynomial of a matrix can be recovered by computing the lowest common multiple of the minimal polynomial of sequences obtained by random projections [21, Proposition 4].

Therefore, the work of the Prover is just that of computing minimal polynomials of sequences at given vector projections. Communication is that of the two vector projections, $2n$. Finally the work of the Verifier is to verify the certificate for the sequence and then to apply the fast Euclidean algorithm, at cost $n^{1+o(1)}$, to recover the minimal polynomial by himself. □

6.2 Det

Theorem 6 ([21])**.** *Certifying the determinant can be reduced to the certification of the minimal polynomial.*

Proof. We use the idea of [21, Theorem 2]: precondition the initial matrix A into a modified matrix B whose characteristic polynomial is square-free, and whose determinant is an easily computable modification of that of A. For instance, such a PreCondCyc preconditioner can be a diagonal matrix if the field is sufficiently large [4, Theorem 4.2] Precondition to get a square-free charpoly [21, Theorem 2] and then certify the associated minpoly. □

6.3 CharPoly

Theorem 7 ([8]). *Certifying the characteristic polynomial can be reduced to the certification of the determinant.*

Proof. The reduction is that of [8, Figure 1]: the Prover computes the characteristic polynomial and sends it as a commitment to the Verifier; then the Verifier gives a point λ as challenge to the Prover which responds with the determinant of $\lambda I_d - A$, and a certificate for that determinant ($\lambda I_d - A$ remains sparse and costs no more than $\mu + n$ to be applied to a vector). Finally, the verifier simplify evaluates the commitment at λ and checks the equality with the certified determinant. □

6.4 Det over $\mathbb{Z}$

Here the strategy is that of [8, §4.4]: ask for MinPoly, Det, CharPoly over $\mathbb{Z}$. After the commitment, the Verifier chooses a not so large prime, and ask for a certificate of that same problem modulo the prime. Then the Verifier checks the certificate, and checks coherency with the integral counterpart. On the one hand, the minimal and characteristic polynomial over $\mathbb{Z}$ already occupy a quadratic space, so that taking modular images is already quadratic. On the other hand, for the determinant, this gives a linear time Verifier.

References

[1] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In Victoria Ashby, editor, *Proceedings of the 1st ACM Conference on Computer and Communications Security*, pages 62–73, Fairfax, Virginia, November 1993. ACM Press. URL: http://www-cse.ucsd.edu/users/mihir/papers/ro.pdf.

[2] David Bernhard, Olivier Pereira, and Bogdan Warinschi. How not to prove yourself: Pitfalls of the Fiat-Shamir heuristic and applications to helios. In Xiaoyun Wang and Kazue Sako, editors, *Advances in Cryptology - ASIACRYPT'12*, volume 7658 of *Lecture Notes in Computer Science*, pages 626–643. Springer, 2012. URL: http://www.uclouvain.be/crypto/services/download/publications.pdf.87e67d05ee05000b.6d616

[3] A. Bostan, G. Lecerf, and É. Schost. Tellegen's principle into practice. In *Proceedings of the 2003 International Symposium on Symbolic and Algebraic Computation*, ISSAC '03, pages 37–44, New York, NY, USA, 2003. ACM. doi:10.1145/860854.860870.

[4] Li Chen, Wayne Eberly, Erich Kaltofen, B. David Saunders, William J. Turner, and Gilles Villard. Efficient matrix preconditioners for black box linear algebra. *Linear Algebra and its Applications*, 343-344:119–146, 2002. doi:10.1016/S0024-3795(01)00472-4.

[5] Kai-Min Chung, Yael Tauman Kalai, and Salil P. Vadhan. Improved delegation of computation using fully homomorphic encryption. In Tal Rabin, editor, *Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings*, volume 6223 of *Lecture Notes in Computer Science*, pages 483–501. Springer, 2010. doi:10.1007/978-3-642-14623-7_26.

[6] Don Coppersmith. Solving homogeneous linear equations over $GF(2)$ via block Wiedemann algorithm. *Mathematics of Computation*, 62(205):333–350, January 1994. doi:10.2307/2153413.

[7] Ronald John Fitzgerald Cramer. *Modular design of secure yet practical cryptographic protocols*. PhD thesis, University of Amsterdam, 1996.

[8] Jean-Guillaume Dumas and Erich Kaltofen. Essentially optimal interactive certificates in linear algebra. In Katsusuke Nabeshima, editor, *ISSAC'2014, Proceedings of the 2014 ACM International Symposium on Symbolic and Algebraic Computation, Kobe, Japan*, pages 146–153. ACM Press, New York, July 2014. doi:10.1145/2608628.2608644.

[9] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In A. M. Odlyzko, editor, *Advances in Cryptology - CRYPTO'86*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer-Verlag, 1987, 11–15 August 1986. URL: http://www.cs.rit.edu/~jjk8346/FiatShamir.pdf.

[10] Dario Fiore and Rosario Gennaro. Publicly verifiable delegation of large polynomials and matrix computations, with applications. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, CCS '12, pages 501–512, New York, NY, USA, 2012. ACM. doi:10.1145/2382196.2382250.

[11] Rūsiņš Freivalds. Fast probabilistic algorithms. In J. Bečvář, editor, *Mathematical Foundations of Computer Science 1979*, volume 74 of *Lecture Notes in Computer Science*, pages 57–69, Olomouc, Czechoslovakia, September 1979. Springer-Verlag. doi:10.1007/3-540-09526-8_5.

[12] Craig Gentry, Jens Groth, Yuval Ishai, Chris Peikert, Amit Sahai, and Adam Smith. Using fully homomorphic hybrid encryption to minimize non-interative zero-knowledge proofs. *Journal of Cryptology*, pages 1–24, 2014. doi:10.1007/s00145-014-9184-y.

[13] Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. Delegating computation: interactive proofs for muggles. In Cynthia Dwork, editor, *STOC'2008, Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada*, pages 113–122. ACM Press, May 2008. doi:10.1145/1374376.1374396.

[14] Erich Kaltofen. Analysis of Coppersmith's block Wiedemann algorithm for the parallel solution of sparse linear systems. *Mathematics of Computation*, 64(210):777–806, April 1995. doi:10.2307/2153451.

[15] Erich L. Kaltofen, Bin Li, Zhengfeng Yang, and Lihong Zhi. Exact certification in global polynomial optimization via sums-of-squares of rational functions with rational coefficients. *Journal of Symbolic Computation*, 47(1):1–15, January 2012. URL: http://www.math.ncsu.edu/~kaltofen/bibliography/09/KLYZ09.pdf, doi:10.1016/j.jsc.2011.08.002.

[16] Erich L. Kaltofen, Michael Nehring, and B. David Saunders. Quadratic-time certificates in linear algebra. In Anton Leykin, editor, *ISSAC'2011, Proceedings of the 2011 ACM International Symposium on Symbolic and Algebraic Computation, San Jose, California, USA*, pages 171–176. ACM Press, New York, June 2011. doi:10.1145/1993886.1993915.

[17] Bryan Parno, Jon Howell, Craig Gentry, and Mariana Raykova. Pinocchio: Nearly practical verifiable computation. In *Proceedings of the 2013 IEEE Symposium on Security and Privacy*, SP '13, pages 238–252, Washington, DC, USA, 2013. IEEE Computer Society. doi:10.1109/SP.2013.47.

[18] Justin Thaler. Time-optimal interactive proofs for circuit evaluation. In Ran Canetti and JuanA. Garay, editors, *Advances in Cryptology - CRYPTO'13*, volume 8043 of *Lecture Notes in Computer Science*, pages 71–89. Springer Berlin Heidelberg, 2013. URL: http://arxiv.org/abs/1304.3812, doi:10.1007/978-3-642-40084-1_5.

[19] Gilles Villard. Further analysis of Coppersmith's block Wiedemann algorithm for the solution of sparse linear systems. In Wolfgang W. Küchlin, editor, *ISSAC'97, Proceedings of the 1997 ACM International Symposium on Symbolic and Algebraic Computation, Maui, Hawaii*, pages 32–39. ACM Press, New York, July 1997. doi:10.1145/258726.258742.

[20] Joachim von zur Gathen and Jürgen Gerhard. *Modern Computer Algebra (3. ed.)*. Cambridge University Press, 2013. doi:10.1017/CBO9781139856065.

[21] Douglas H. Wiedemann. Solving sparse linear equations over finite fields. *IEEE Transactions on Information Theory*, 32(1):54–62, January 1986. doi:10.1109/TIT.1986.1057137.