Solving Toeplitz- and Vandermonde-like linear systems with large displacement rank

Alin Bostan, Claude-Pierre Jeannerod, Éric Schost

Solving Toeplitz- and Vandermonde-like Linear Systems with Large Displacement Rank

Alin Bostan
Algorithms Project, INRIA Rocquencourt
78153 Le Chesnay Cedex, France
Alin.Bostan@inria.fr

Claude-Pierre Jeannerod
Arénaire Project, INRIA Rhônes-Alpes
46 allée d'Italie, 69364 Lyon Cedex 07, France
cpjeanne@ens-lyon.fr

Éric Schost
ORCCA and Computer Science Department
University of Western Ontario
London, ON, Canada
eschost@uwo.ca

ABSTRACT

Linear systems with structures such as Toeplitz-, Vandermonde- or Cauchy-likeness can be solved in $O^{\sim}(\alpha^2 n)$ operations, where n is the matrix size, α is its displacement rank, and $O^{\sim}$ denotes the omission of logarithmic factors. We show that for Toeplitz-like and Vandermonde-like matrices, this cost can be reduced to $O^{\sim}(\alpha^{\omega-1} n)$, where ω is a feasible exponent for matrix multiplication over the base field. The best known estimate for ω is $\omega < 2.38$, resulting in costs of order $O^{\sim}(\alpha^{1.38} n)$. We also present consequences for Hermite-Padé approximation and bivariate interpolation.

Categories and Subject Descriptors:
I.1.2 [**Computing Methodologies**]: Symbolic and Algebraic Manipulation – *Algebraic Algorithms*

General Terms: Algorithms, Theory

Keywords: Structured linear algebra, Dense linear algebra

1. INTRODUCTION

Structured linear algebra techniques are a versatile set of tools. They enable one to deal at once with matrices with features such as Toeplitz-, Vandermonde- or Cauchy-likeness, and that arise in various problems, from interpolation to reconstruction of rational or algebraic functions, etc.

Following [21], the usual way of measuring to what extent a matrix possesses one such structure is through its *displacement rank*, that is, the rank of its image through a suitable *displacement operator*. For P and Q in respectively $\mathbb{K}^{n\times n}$ and $\mathbb{K}^{m\times m}$, where $\mathbb{K}$ is our base field, we will use the displacement operator

$$\begin{array}{rccc} \Delta[\mathsf{P},\mathsf{Q}]: & \mathbb{K}^{n\times m} & \to & \mathbb{K}^{n\times m} \\ & \mathsf{A} & \mapsto & \mathsf{A} - \mathsf{P}\,\mathsf{A}\,\mathsf{Q}. \end{array}$$

Two matrices (Y, Z) in $\mathbb{K}^{n\times\alpha} \times \mathbb{K}^{m\times\alpha}$ will be called a P,Q-*generator of length* α for A if $\Delta[\mathsf{P},\mathsf{Q}](\mathsf{A}) = \mathsf{Y}\,\mathsf{Z}^t$. The main idea behind algorithms for structured matrices is to use such generators as a compact data structure, in cases when $\Delta[\mathsf{P},\mathsf{Q}](\mathsf{A})$ has low rank. Even though these definitions hold for rectangular A, in most of the paper, except Subsection 4.1, we have $n = m$.

Usual choices for P or Q are diagonal matrices, or cyclic down-shift matrices of size n, defined for φ in $\mathbb{K}$ by

$$\mathbb{Z}_{n,\varphi} = \begin{bmatrix} 0 & & & \varphi \\ 1 & \ddots & & \\ & \ddots & \ddots & \\ & & 1 & 0 \end{bmatrix} \in \mathbb{K}^{n\times n}.$$

The Toeplitz structure corresponds to $\mathsf{P} = \mathbb{Z}_{n,0}$ and $\mathsf{Q} = \mathbb{Z}_{m,0}^t$, so that $\Delta[\mathbb{Z}_{n,0}, \mathbb{Z}_{m,0}^t](\mathsf{A})$ equals A − (A shifted down and right by one unit). The Vandermonde structure is obtained by taking P diagonal and Q a cyclic right-shift matrix $\mathbb{Z}_{m,\varphi}^t$. For the Cauchy structure, both P and Q are diagonal.

In this paper, we consider the following task:

$\mathsf{LinearSystem}(\mathsf{P},\mathsf{Q},\alpha)$: *Given a* P, Q*-generator of length* α *for a matrix* $\mathsf{A} \in \mathbb{K}^{n\times n}$*, with* $\alpha \le n$*, and given* $\mathsf{v} \in \mathbb{K}^n$*, find a uniform random solution to the equation* $\mathsf{Au} = \mathsf{v}$*, or determine that none exists.*

This problem makes sense only when the operator $\Delta[\mathsf{P},\mathsf{Q}]$ is invertible: this will be the case in our two cases of focus, Toeplitz-like and Vandermonde-like matrices. Previous work then yielded the following kind of results: for the Toeplitz, Vandermonde and Cauchy structures, one can solve the problem $\mathsf{LinearSystem}$ using $O^{\sim}(\alpha^2 n)$ operations in $\mathbb{K}$, where the $O^{\sim}$ notation hides logarithmic factors.

When α is constant, such estimates are optimal up to logarithmic factors. However, there are several situations where α is not bounded *a priori* (see examples below). In the extreme case of very loosely structured matrices, when α goes up to $\alpha \simeq n$, the cost above becomes $O^{\sim}(n^3)$.

On the other side of the spectrum, we find *dense* linear algebra methods. Let $\omega < 3$ be such that $n\times n$ matrices over $\mathbb{K}$ can be multiplied in $O(n^\omega)$ operations (the current record estimate is $\omega < 2.38$ [10]). Then, linear systems of size n can be solved in time $O(n^\omega)$, using e.g. LSP factorization [20]; with $\omega < 3$, this is better than the above $O^{\sim}(n^3)$ estimate.

Our contribution bridges a gap between the approaches of structured and dense linear algebra, in the case of Toeplitz-like and Vandermonde-like matrices. The algorithms rely on polynomial multiplication; we will thus denote by $\mathsf{M} : \mathbb{N}_{>0} \to \mathbb{R}_{>0}$ a function such that polynomials in $\mathbb{K}[x]$ of degree less than d can be multiplied in $\mathsf{M}(d)$ operations. We make the standard super-linearity assumption that $\mathsf{M}(d + d') \ge \mathsf{M}(d) + \mathsf{M}(d')$ holds for all d, d'; see [16, Chapter 8]. Using [38, 9], one can take $\mathsf{M}(d) \in O(d \log(d) \log\log(d))$.

ISSAC'07, July 29–August 1, 2007, Waterloo, Ontario, Canada.

Using [9], polynomial matrices over $\mathbb{K}$ of degree less than d and size n can be multiplied in $O(\mathsf{M}(d)n^\omega)$ operations in $\mathbb{K}$.

The algorithms are probabilistic; to simplify the presentation, we will say that an algorithm has type $P(r,d)$ if it chooses r random elements in $\mathbb{K}$, say $\ell_1,\dots,\ell_r$, and if there exists a non-zero polynomial $\Gamma \in \mathbb{K}[L_1,\dots,L_r]$ of degree at most d such that if $\Gamma(\ell_1,\dots,\ell_r)\neq 0$, the algorithm succeeds. It follows from the Zippel-Schwartz lemma [11, 44, 39] that if $\ell_1,\dots,\ell_r$ are chosen uniformly at random in a finite subset S of $\mathbb{K}$, the probability of success is at least $1-d/|S|$.

Main results. Our first result covers matrices with Toeplitz-like structure, with $\mathsf{P}=\mathbb{Z}_{n,0}$ and $\mathsf{Q}=\mathbb{Z}^t_{n,0}$. We obtain a complexity in $O^{\tilde{}}(\alpha^{\omega-1}n)\subset O^{\tilde{}}(\alpha^{1.38}n)$, to be compared with an optimal cost of $O(\alpha n)$. For α constant, our result is quasi-linear in n; when $\alpha\simeq n$, we recover the $O(n^\omega)$ behaviour of dense methods, up to logarithmic factors.

THEOREM 1. *The problem* $\mathsf{LinearSystem}(\mathbb{Z}_{n,0},\mathbb{Z}^t_{n,0},\alpha)$ *can be solved in time* $O(\alpha^{\omega-1}\mathsf{M}(n)\log^2(n))$, *by a probabilistic algorithm of type* $P(3n-2,n^2+n)$.

A fundamental application of this result is the solution of approximation problems: given a *master polynomial* M and polynomials $f_1,\dots,f_s$, one seeks a combination of the f_i, with polynomial coefficients of prescribed degrees, which vanishes modulo M. This includes in particular Padé and Hermite-Padé approximation (taking $M=x^n$), with applications to e.g. recovering the minimal polynomial of an algebraic power series f (taking $f_i=f^{i-1}$).

COROLLARY 1. *Let* $M\in\mathbb{K}[x]$ *be of degree* n, $f_1,\dots,f_s\in\mathbb{K}[x]$ *be of degrees less than* n *and let* $\nu_1,\dots,\nu_s\in\mathbb{N}$ *be such that* $\sum_{i\le s}\nu_i=n+1$. *One can find* $g_1,\dots,g_s\in\mathbb{K}[x]$, *not all zero, of respective degrees less than* $\nu_1,\dots,\nu_s$, *such that* $g_1f_1+\cdots+g_sf_s=0 \bmod M$, *in time* $O(s^{\omega-1}\mathsf{M}(n)\log^2(n))$. *The algorithm is probabilistic of type* $P(3n-2,n^2+n)$.

As observed before, the cost is thus in $O^{\tilde{}}(s^{\omega-1}n)\subset O^{\tilde{}}(s^{1.38}n)$, to be compared with an optimal cost of $O(sn)$.

Our second result addresses the Vandermonde case, where $\mathsf{P}=\mathbb{D}(\mathsf{x})$ is the diagonal matrix with diagonal $\mathsf{x}=[x_1,\dots,x_n]$ and Q has the form $\mathbb{Z}^t_{n,\psi}$. In this article, we work under the following assumption:

$$\mathbf{A}.\ \text{For } i\le n,\ \text{one has } \psi\, x_i^n\neq 1. \tag{1}$$

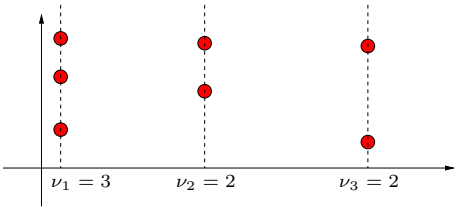
The complexity in the Vandermonde case is then similar to that of the Toeplitz case:

THEOREM 2. *Suppose that* $\mathbb{K}$ *has cardinality at least* n. *If* $\mathsf{x}\in\mathbb{K}^n$ *and* ψ *satisfy assumption* $\mathbf{A}$, *one can solve the problem* $\mathsf{LinearSystem}(\mathbb{D}(\mathsf{x}),\mathbb{Z}^t_{n,\psi},\alpha)$ *in time* $O(\alpha^{\omega-1}\mathsf{M}(n)\log^2(n))$, *by a probabilistic algorithm of type* $P(3n-2,n^2+n)$.

We conclude with an application of the latter theorem to polynomial interpolation. The approach applies to any number of variables, but we discuss only the bivariate case for simplicity. Consider n interpolation points in $\mathbb{K}^2$; without loss of generality, we assume that they are written as

$$\begin{array}{lll} p_{1,1}=(x_1,y_{1,1}) & \dots & p_{1,\nu_1}=(x_1,y_{1,\nu_1}) \\ & \dots & \\ p_{s,1}=(x_s,y_{s,1}) & \dots & p_{s,\nu_s}=(x_s,y_{s,\nu_s}), \end{array} \tag{2}$$

with $\nu_1\ge\cdots\ge\nu_s>0$ and $n=\nu_1+\cdots+\nu_s$. The following figure illustrates the case $s=3$, $n=7$, $\nu_1=3$, $\nu_2=\nu_3=2$.



In general, it is difficult to state *a priori* that a multivariate interpolation problem is well-defined. Here, however, given $[v_{i,j}]_{1\le i\le s,1\le j\le\nu_i}$ in $\mathbb{K}^n$, Theorem 1 in [29] (see also [14]) implies that there exists a unique $F\in\mathbb{K}[x,y]$ of the form

$$F=\textstyle\sum_{1\le i\le s,\,1\le j\le\nu_i} f_{i,j}x^{i-1}y^{j-1}$$

such that $F(p_{i,j})=v_{i,j}$ for all i,j. Finding the coefficients $f_{i,j}$ of F is a linear problem, with Vandermonde-like structure; we deduce the following corollary of Theorem 2.

COROLLARY 2. *If* $\mathbb{K}$ *has cardinality at least* n, *given the values* $v_{i,j}$, *the coefficients* $f_{i,j}$ *can be computed in time* $O(\min(s,\nu_1)^{\omega-1}\mathsf{M}(n)\log^2(n))$. *The algorithm is probabilistic of type* $P(3n-2,n^2+n)$.

Suppose for instance that $\nu_1=s,\nu_2=s-1,\dots,\nu_s=1$, so that we are interpolating on the simplex of monomials of degree less than s; here, $n=s(s+1)/2$. Then, our algorithm has subquadratic complexity $O^{\tilde{}}(n^{(\omega+1)/2})\subset O^{\tilde{}}(n^{1.69})$.

Few practical algorithms are currently known for matrix multiplication with complexity better than cubic (see [42, 28] and [26] for an exponent 2.77). However, even when using algorithms of cubic complexity, the re-introduction of dense matrix arithmetic in our algorithms means that we can rely on extremely optimized implementations of matrix multiplication, such as the ones relying on BLAS libraries for finite field arithmetic [12]. Hence, besides theoretical estimates, our approach may lead to practical improvements.

Previous work. The notions of displacement rank and displacement operators originate from the work of Kailath, Kung and Morf [21]. Since then, the literature has vastly developed; see [36] for a list of references.

The basis of Theorem 1 is the algorithm of Bitmead and Anderson [6] and Morf [30, 31], which requires several invertibility conditions to hold. Kaltofen [22, 23] extended this idea to arbitrary matrices (see also [5, p. 204] for some related ideas), obtaining a complexity of $O(\alpha^2\mathsf{M}(n)\log(n))$; for small α, this is better than our result in Theorem 1. We follow his approach, our main technical contribution being the fast multiplication of a Toeplitz-like matrix (given by its generators) by *several* vectors.

An important example of Toeplitz-like system solving is the approximation problem of Corollary 1. In the particular case of Hermite-Padé approximation, with $M=x^n$, a central reference is Beckermann-Labahn's algorithm [2], that has complexity $O(s^\omega\mathsf{M}(n)\log(n))$ for computing a σ-basis of order n of the input system (and thus a solution to the approximation problem); see [18]. In generic cases, an unpublished result of Lecerf reduces the cost to $O(s^{\omega-1}\mathsf{M}(n)\log(n))$ and Storjohann [41] subsequently obtained a deterministic algorithm of similar complexity, applying in all cases. However, to our knowledge, these results do not extend to an arbitrary choice of M. Following notably [1, 43], Beckermann and Labahn study that general case in [3] under the angle of fraction-free algorithms, with however a complexity more than linear in n.

Another example of Toeplitz-like system that occurs frequently is when the matrix is block-Toeplitz, a block-size equal to α giving a displacement rank in $O(\alpha)$. Although Theorem 1 applies to any such system, a deterministic cost of $O(\alpha^{\omega-1}\mathsf{M}(n)\log(n))$ can be obtained in the particular case where the matrix is invertible. As described for example in [13], this cost follows from combining an inversion formula of [27] with σ-basis computations as in [18].

To prove Theorem 2, we transform a Vandermonde-like system into a Toeplitz-like one, following Pan's idea [34]. We use a transformation from [19], generalizing it by taking into account the possibility of repetitions in the diagonal component of the operator. Again, the main technical tool is to multiply (submatrices of) a Vandermonde-like matrix, given by its generator, by *several* vectors.

Multivariate polynomial interpolation has been extensively studied (see [15] for a survey and [4, 8, 45] for algorithms relevant from sparse techniques). However, to our knowledge, previous references either do not cover the problems we deal with, or have higher complexity (typically, quadratic). Regarding the converse evaluation problem, let us mention the subquadratic complexity result of [33], which however deals with more general situations than ours.

Organization of the paper. After introducing basic notation and results in Section 2, we present in Section 3 the bases of our technical improvement, which can be stated in terms of polynomial operations only. These results are then applied, first to the Toeplitz case in Section 4, then to the Vandermonde case in Section 5.

All complexities are expressed in terms of base field operations. In several cases, we will add big-Oh estimates, with sometimes a non-constant number of summands. While such big-Oh additions are in general delicate to handle, one easily sees that all our simplifications are indeed valid.

Acknowledgments. We thank E. Kaltofen, G. Labahn, M. Morf, V. Y. Pan, B. Salvy, A. Storjohann and G. Villard for useful discussions and comments.

2. NOTATION AND PRELIMINARIES

General notation. In what follows, we consider matrices and vectors over a field $\mathbb{K}$. Matrices (resp. vectors) are written in upper-case (resp. lower-case) sans-serif font. If A is a matrix, a_i is its ith column. If x is a vector, its ith entry is written x_i. Special matrices (diagonal, Vandermonde, ...) will be written with Blackboard Bold letters $\mathbb{D}$, $\mathbb{V}$, ...

If F is a function on $\mathbb{K}$ and x is in $\mathbb{K}^n$, then $F(\mathsf{x})$ denotes the vector of values $[F(x_1),\dots,F(x_n)]^t$. We will write $\mathsf{Flip}(\mathsf{x})$ for the vector $[0,x_n,\dots,x_2]^t \in \mathbb{K}^n$. For $n \in \mathbb{N}$ and $r = r_0 + \cdots + r_{n-1}x^{n-1} \in \mathbb{K}[x]$ of degree less than n, we will write $\mathsf{Rev}_n(r) = r_{n-1} + \cdots + r_0 x^{n-1}$. For $r \in \mathbb{K}[x]$ and $s \in \mathbb{K}[x]$ nonzero, r div s and r mod s are the quotient and the remainder in the division of r by s. Finally, $\mathsf{Pol}(\mathsf{x})$ is the polynomial $\sum_{i=0}^{n-1} x_{i+1}x^i \in \mathbb{K}[x]$.

Structured matrices. We associate several matrices to a vector x in $\mathbb{K}^n$: $\mathbb{D}(\mathsf{x})$ is the diagonal matrix with diagonal x; $\mathbb{L}(\mathsf{x})$ is the lower-triangular Toeplitz matrix with first column x; $\mathbb{U}(\mathsf{x}) = \mathbb{L}(\mathsf{x})^t$ is the upper-triangular Toeplitz matrix with first row x^t. For $m \in \mathbb{N}$, $\mathbb{V}(\mathsf{x},m)$ is the $n \times m$ Vandermonde matrix $\mathbb{V}(\mathsf{x},m) = [x_i^j]_{1\le i\le n, 0\le j<m}$. For φ in $\mathbb{K}$, $\mathbb{C}(\mathsf{x},\varphi)$ is the φ-circulant matrix with first column x; that is, $\mathbb{C}(\mathsf{x},\varphi) = \mathbb{L}(\mathsf{x}) + \varphi\mathbb{U}(\mathsf{Flip}(\mathsf{x}))$.

Multiplication by lower- or upper-triangular Toeplitz matrices can be seen in terms of polynomial operations. For y and z in $\mathbb{K}^n$, letting $\mathsf{u} = \mathbb{L}(\mathsf{y})\mathsf{z}$ and $\mathsf{v} = \mathbb{U}(\mathsf{y})\mathsf{z}$, we have

$$\begin{array}{rcl} \mathsf{Pol}(\mathsf{u}) & = & \mathsf{Pol}(\mathsf{y})\mathsf{Pol}(\mathsf{z}) \bmod x^n \\ \mathsf{Pol}(\mathsf{v}) & = & \mathsf{Rev}_n(\mathsf{Pol}(\mathsf{y}))\mathsf{Pol}(\mathsf{z}) \text{ div } x^{n-1}. \end{array} \quad (3)$$

The lemma below describes a more complex operation, needed in Section 5.1 for handling Vandermonde-like matrices. (See [7] for a proof.)

LEMMA 1. *Let φ be in $\mathbb{K}$, let z be in $\mathbb{K}^n$, let x, y and f be in $\mathbb{K}^\nu$, and let $\mathsf{g} \in \mathbb{K}^n$ be defined by*

$$\mathsf{g} = \mathbb{C}(\mathsf{z},\varphi)\,\mathbb{V}(\mathsf{x},n)^t\,\mathbb{D}(\mathsf{y})\,\mathsf{f}.$$

Let $\mathsf{z}' = \mathsf{Flip}(\mathsf{z})$, $\mathsf{f}' = \mathbb{V}(\mathsf{x},n+\nu-1)^t\,\mathsf{f}$ and $F = \mathsf{Pol}(\mathsf{f}')$. Assuming that the entries of x are pairwise distinct, define G as the unique polynomial of degree less than ν such that $G(\mathsf{x}) = \mathsf{y}$. Then we have the equality

$$\begin{aligned} \mathsf{Pol}(\mathsf{g}) = \mathsf{Pol}(\mathsf{z})(F\,\mathsf{Rev}_\nu(G) \text{ div } x^{\nu-1}) \bmod x^n + \\ \varphi\,\mathsf{Rev}_n\big(\mathsf{Pol}(\mathsf{z}')(\mathsf{Rev}_{n+\nu-1}(F)\,G \text{ div } x^{\nu-1}) \bmod x^n\big). \end{aligned} \quad (4)$$

Given P,Q-generators for a matrix A, a useful tool is the determination of generators for A of minimal length. Remark 4.6.7 in [36] gives the following result.

PROPOSITION 1. *Let $\mathsf{P},\mathsf{Q} \in \mathbb{K}^{n\times n}$. Given a P,Q-generator of length α for $\mathsf{A} \in \mathbb{K}^{n\times n}$, one can compute a P,Q-generator for A of minimal length in $O(\alpha^{\omega-1}n)$ operations.*

3. POLYNOMIAL OPERATIONS

We discuss here two problems involving polynomials, that boil down to suitably using polynomial matrix multiplication to speed up the simultaneous computation of several trilinear expressions.

3.1 First problem

In the following, some integers n and $\alpha \le n$ are fixed. Let $(Y_i)_{i\le\alpha}$, $(Z_i)_{i\le\alpha}$ and $(F_j)_{j\le\alpha}$ be in $\mathbb{K}[x]$, all of degree less than n. The next proposition will be used in Section 4.

PROPOSITION 2. *One can compute the polynomials*

$$G_j = \textstyle\sum_{i=1}^{\alpha} Y_i(Z_iF_j \bmod x^n), \quad j = 1,\dots,\alpha$$

using $O(\alpha^{\omega-1}\mathsf{M}(n)\log(n))$ operations in $\mathbb{K}$.

PROOF. Up to replacing n with $\bar{n} = 2^{\lceil\log(n)\rceil}$ and F_j with $x^{\bar{n}-n}F_j$, we can (and will) suppose that n is a power of 2.

We first show how to rewrite truncated products using non-truncated ones, using ideas reminiscent of *short products* [32]. Let $k \ge 1$ be a power of 2 and let ℓ be in $\mathbb{N}$. For $P = p_0 + p_1x + \cdots$, we define $P^{(\ell,k)} \in \mathbb{K}[x]$ as follows:

$$P^{(\ell,1)} = p_\ell \quad \text{and} \quad P^{(\ell,k)} = \sum_{i=\ell k}^{\ell k + k/2 - 1} p_i x^{i-\ell k} \quad \text{for } k \ge 2.$$

In all cases, $P^{(\ell,k)}$ is a polynomial of degree less than $k/2$. Using this subdivision enables us to rewrite a truncated product $PQ \bmod x^n$ as a sum of non-truncated ones.

LEMMA 2. *For P and Q in $\mathbb{K}[x]$ and m a power of 2,*

$$PQ \bmod x^m = \sum_{k=1,2,4,\dots,m} x^{m-k} \sum_{\ell=0}^{m/k-1} P^{(\ell,k)}Q^{(m/k-1-\ell,k)},$$

where the sum is taken on all $k \le m$ that are powers of 2.

PROOF. We proceed by induction on $m \geq 1$, for m a power of 2. If $m = 1$ this is clear, so assume $m > 1$. Let us write

$$P \bmod x^m = P_0 + x^{m/2} P_1 \quad \text{and} \quad Q \bmod x^m = Q_0 + x^{m/2} Q_1,$$

with P_0, P_1, Q_0, Q_1 of degree less than $m/2$. Then we have

$$P_0^{(\ell,k)} = P^{(\ell,k)} \quad \text{and} \quad P_1^{(\ell,k)} = P^{(\ell+m/2k,k)}$$

for any $k \geq 1$ and $\ell \geq 0$ such that $\ell k + k/2 \leq m/2$. Analogous equalities hold for Q, Q_0 and Q_1. Now, by definition,

$$PQ \bmod x^m = P_0 Q_0 + x^{m/2}(P_0 Q_1 + P_1 Q_0 \bmod x^{m/2}). \quad (5)$$

Observe first that $P_0 Q_0$ equals $P^{(0,m)} Q^{(0,m)}$, which corresponds to the term $k = m$ in the right-hand side of the formula we wish to establish. Next, the induction assumption shows that $P_0 Q_1 \bmod x^{m/2}$ is given by

$$\begin{aligned} & \textstyle\sum_{k=1,2,\ldots,m/2} x^{m/2-k} \sum_{\ell=0}^{m/2k-1} P_0^{(\ell,k)} Q_1^{(m/2k-1-\ell,k)} \\ = \; & \textstyle\sum_{k=1,2,\ldots,m/2} x^{m/2-k} \sum_{\ell=0}^{m/2k-1} P^{(\ell,k)} Q^{(m/k-1-\ell,k)}. \end{aligned}$$

Similarly, $P_1 Q_0 \bmod x^{m/2}$ equals

$$\textstyle\sum_{k=1,2,\ldots,m/2} x^{m/2-k} \sum_{\ell=m/2k}^{m/k-1} P^{(\ell,k)} Q^{(m/k-1-\ell,k)}.$$

Putting these equalities in Equation (5) ends the proof. □

We can now prove the proposition. Lemma 2 shows that for all i and j, $Y_i(Z_i F_j \bmod x^n)$ equals

$$\textstyle\sum_{k=1,2,4,\ldots,n} x^{n-k} \sum_{\ell=0}^{n/k-1} Y_i \, Z_i^{(\ell,k)} F_j^{(n/k-1-\ell,k)}.$$

Thus for $j \leq \alpha$, we have $G_j = \sum_{k=1,2,4,\ldots,n} x^{n-k} G_{j,k}$, with

$$G_{j,k} = \textstyle\sum_{i=1}^{\alpha} \sum_{\ell=0}^{n/k-1} Y_i \, Z_i^{(\ell,k)} \, F_j^{(n/k-1-\ell,k)}.$$

LEMMA 3. *Let $k \leq n$ be a power of 2. Then one can compute $G_{1,k}, \ldots, G_{\alpha,k}$ in $O(\alpha^{\omega-1}\mathsf{M}(n))$ operations in $\mathbb{K}$.*

PROOF. Let $k' = n/k$, and let Z and F be the $\alpha \times k'$ and $k' \times \alpha$ polynomial matrices

$$\mathsf{Z} = \begin{bmatrix} Z_1^{(0,k)} & \cdots & Z_1^{(k'-1,k)} \\ \vdots & & \vdots \\ Z_\alpha^{(0,k)} & \cdots & Z_\alpha^{(k'-1,k)} \end{bmatrix}, \quad \mathsf{F} = \begin{bmatrix} F_1^{(k'-1,k)} & \cdots & F_\alpha^{(k'-1,k)} \\ \vdots & & \vdots \\ F_1^{(0,k)} & \cdots & F_\alpha^{(0,k)} \end{bmatrix}.$$

Then we have the equality

$$\begin{bmatrix} G_{1,k} & \cdots & G_{\alpha,k} \end{bmatrix} = \begin{bmatrix} Y_1 & \cdots & Y_\alpha \end{bmatrix} \mathsf{Z}\mathsf{F}.$$

All entries of Z and F have degree less than $k/2$. Hence, for $i \leq \alpha$, we write $Y_i = \sum_{\ell=0}^{k'-1} Y_{i,\ell} x^{k\ell}$, with $Y_{i,\ell}$ of degree less than k. We can then define the $k' \times \alpha$ matrix

$$\mathsf{Y} = \begin{bmatrix} Y_{1,0} & \cdots & Y_{\alpha,0} \\ \vdots & & \vdots \\ Y_{1,k'-1} & \cdots & Y_{\alpha,k'-1} \end{bmatrix}$$

with polynomial entries of degree less than k, such that

$$\begin{bmatrix} Y_1 & \cdots & Y_\alpha \end{bmatrix} = \begin{bmatrix} 1 & x^k & x^{2k} & \cdots & x^{(k'-1)k} \end{bmatrix} \mathsf{Y}. \quad (6)$$

We bound the cost of computing the product $\mathsf{Y}\,\mathsf{Z}\,\mathsf{F}$ by considering two cases: if $\alpha \leq k'$ then compute $\mathsf{Y}\,\mathsf{Z}\,\mathsf{F}$ as $\mathsf{Y}(\mathsf{Z}\,\mathsf{F})$ in time $O(\alpha^{\omega-1} k' \mathsf{M}(k))$; if $k' \leq \alpha$ then compute it as $(\mathsf{Y}\,\mathsf{Z})\mathsf{F}$ in time $O(\alpha^{\omega-1} k' \mathsf{M}(k))$. Both costs are in $O(\alpha^{\omega-1}\mathsf{M}(n))$ because of $k' \mathsf{M}(k) \leq \mathsf{M}(n)$. Finally, by (6), $G_{1,k}, \ldots, G_{\alpha,k}$ are deduced from $\mathsf{Y}\,\mathsf{Z}\,\mathsf{F}$ in time $O(k' \alpha\, k) \subset O(\alpha\, n)$. □

To conclude the proof of Proposition 2, we apply Lemma 3 to $k = 1, 2, 4, \ldots, n$, for a total cost of $O(\alpha^{\omega-1}\mathsf{M}(n)\log(n))$. The cost of deducing $G_1, \ldots, G_\alpha$ is $O(\alpha\, n \log(n))$. □

3.2 Second problem

As above, integers $n \in \mathbb{N}$ and $\alpha \leq n$ are fixed. Let also $s \leq \alpha$ and $\nu_1, \ldots, \nu_s \in \mathbb{N}_{>0}$ be such that $n = \nu_1 + \cdots + \nu_s$, and $(Z_i)_{i \leq \alpha}$, $(H_{i,j})_{i \leq \alpha, j \leq s}$, and $(W_j)_{j \leq s}$ be in $\mathbb{K}[x]$, with $\deg(Z_i) < n$, $\deg(H_{i,j}) < \nu_j$ and $\deg(W_j) < n + \nu_j$. The next proposition will be used in Section 5.

PROPOSITION 3. *One can compute the polynomials*

$$P_j = \textstyle\sum_{i=1}^{\alpha} Z_i \left(H_{i,j} W_j \text{ div } x^{\nu_j - 1} \right), \quad j = 1, \ldots, s$$

using $O(\alpha^{\omega-1}\mathsf{M}(n)\log(n))$ operations in $\mathbb{K}$.

PROOF. We start with a lemma.

LEMMA 4. *For $i \leq \alpha$ and $j \leq s$, let $G_{i,j} = H_{i,j} W_j \bmod x^{\nu_j - 1}$. Then one can compute the polynomials*

$$Q_j = \textstyle\sum_{i=1}^{\alpha} Z_i G_{i,j} \quad \text{and} \quad R_j = \sum_{i=1}^{\alpha} Z_i H_{i,j} \quad j = 1, \ldots, s$$

using $O(\alpha^{\omega-1}\mathsf{M}(n)\log(n))$ operations in $\mathbb{K}$.

PROOF. For given i and j, $G_{i,j}$ can be computed in $\mathsf{M}(\nu_j)$ operations, so the total cost for all $G_{i,j}$ is at most $\alpha\,\mathsf{M}(n)$. The polynomials $G_{i,j}$ and $H_{i,j}$ both have degree less than ν_j; thus, computing $Q_1, \ldots, Q_s$ and computing $R_1, \ldots, R_s$ are similar problems and we focus only on the first of them.

Let $\beta \leq n$ be a power of 2 and define $S = \{j \leq s \mid \lfloor \beta/2 \rfloor \leq \nu_j < \beta\}$. We will prove below that one can compute the polynomials $\{Q_j \mid j \in S\}$ with $O(\alpha^{\omega-1}\mathsf{M}(n))$ operations in $\mathbb{K}$. This will yield the conclusion of the lemma, since it suffices to take $\beta = 1, 2, 4, \ldots, 2^{1+\lfloor \log(n) \rfloor}$ to obtain all Q_j.

Let $L = \lceil n/\beta \rceil$, and let us write $Z_i = \sum_{\ell < L} Z_{i,\ell} x^{\beta\ell}$, with $Z_{i,\ell}$ of degree less than β. It follows that for all j, Q_j equals $\sum_{\ell < L} Q_{j,\ell} x^{\beta\ell}$, with $Q_{j,\ell} = \sum_{i=1}^{\alpha} Z_{i,\ell} G_{i,j}$. For $j \in S$, the polynomials $Q_{j,\ell}$ have degree less than 2β. Thus, once these polynomials are known, the polynomials Q_j, for $j \in S$, can be recovered in time $O(rn) \in O(\alpha n)$, with $r = |S|$.

Writing $S = \{j_1, \ldots, j_r\}$, the polynomials $Q_{j,\ell}$ are obtained through the following matrix-matrix product:

$$\begin{bmatrix} Q_{j_1,0} & \cdots & Q_{j_r,0} \\ \vdots & & \vdots \\ Q_{j_1,L-1} & \cdots & Q_{j_r,L-1} \end{bmatrix} = \begin{bmatrix} Z_{1,0} & \cdots & Z_{\alpha,0} \\ \vdots & & \vdots \\ Z_{1,L-1} & \cdots & Z_{\alpha,L-1} \end{bmatrix} \begin{bmatrix} G_{1,j_1} & \cdots & G_{1,j_r} \\ \vdots & & \vdots \\ G_{\alpha,j_1} & \cdots & G_{\alpha,j_r} \end{bmatrix}.$$

These matrices have sizes $(L \times r)$, $(L \times \alpha)$ and $(\alpha \times r)$, with entries of degree less than β. To conclude, we distinguish two cases, using the fact that $r \leq s \leq \alpha \leq n$. If $r \leq L$ then $r = \min\{L, \alpha, r\}$ and the above product can be computed in time $O(L\,\alpha\, r^{\omega-2}\mathsf{M}(\beta))$, which is in $O(\alpha^{\omega-1}\mathsf{M}(n))$. If $r > L$ then $L = \min\{L, \alpha, r\}$ and the cost is now $O(\alpha\, r\, L^{\omega-2}\,\mathsf{M}(\beta))$; to get the same bound as in the previous case, let us check that $r \leq 2L$. By definition of L, one has $r > n/\beta$, thus $\beta > 1$. Then $r\beta/2 \leq \sum_{j \in S} \nu_j \leq n$ by definition of S and thus $r \leq 2L$. □

The proof of Prop. 3 comes from $P_j = (R_j W_j - Q_j) \text{ div } x^{\nu_j - 1}$. Indeed, knowing the polynomials Q_j and R_j, we can deduce the polynomials P_j in time $O(s\,\mathsf{M}(n)) \subset O(\alpha\,\mathsf{M}(n))$. □

4. THE TOEPLITZ CASE

The operator associated with the Toeplitz structure is

$$\Delta[\mathbb{Z}_{n,0}, \mathbb{Z}_{n,0}^t](\mathsf{A}) = \mathsf{A} - \mathbb{Z}_{n,0}\,\mathsf{A}\,\mathbb{Z}_{n,0}^t, \quad \mathsf{A} \in \mathbb{K}^{n \times n}. \quad (7)$$

This operator is invertible: given (Y, Z) in $\mathbb{K}^{n \times \alpha} \times \mathbb{K}^{n \times \alpha}$, there is a unique A such that $\Delta[\mathbb{Z}_{n,0}, \mathbb{Z}_{n,0}^t](\mathsf{A}) = \mathsf{Y}\mathsf{Z}^t$. In addition one has the so-called $\Sigma\mathrm{LU}$ representation [22]

$$\mathsf{A} = \textstyle\sum_{i=1}^{\alpha} \mathbb{L}(\mathsf{y}_i)\mathbb{U}(\mathsf{z}_i).$$

Using Equation (3), it allows to compute a matrix-vector product Au in $O(\alpha\,\mathsf{M}(n))$ operations in $\mathbb{K}$. Our problem in this section is the converse one: given v in $\mathbb{K}^n$, find u such that $\mathsf{Au} = \mathsf{v}$ (or conclude that no such vector exists).

We improve former algorithms in the case α large, reducing the cost from $O(\alpha^2\mathsf{M}(n)\log(n))$ to $O(\alpha^{\omega-1}\mathsf{M}(n)\log^2(n))$. The key is an extension of the direct problem: given Y, Z and $\mathsf{u}_1, \ldots, \mathsf{u}_\alpha$ in $\mathbb{K}^n$, compute the α products $\mathsf{v}_j = \mathsf{Au}_j \in \mathbb{K}^n$.

4.1 Preliminaries

In addition to the operator in (7) we will use the operator

$$\Delta[\mathbb{Z}^t_{n,0}, \mathbb{Z}_{m,0}](\mathsf{A}) = \mathsf{A} - \mathbb{Z}^t_{n,0}\,\mathsf{A}\,\mathbb{Z}_{m,0}, \quad \mathsf{A} \in \mathbb{K}^{n\times m}.$$

Regardless of dimensions, the operators $\Delta[\mathbb{Z}^t_{n,0}, \mathbb{Z}_{m,0}]$ and $\Delta[\mathbb{Z}_{n,0}, \mathbb{Z}^t_{m,0}]$ are called respectively ϕ_- and ϕ_+ in [35, 22, 23]; their generators are ϕ_--generators and ϕ_+-generators (from now on, we use this simplifying notation.)

We conclude this subsection with some useful results on generators for submatrices, sums, products, ... Our contribution is Proposition 6 below, which is a faster version of [35, Prop. A.3] for generating matrix products; as in [22, 23] we extend the result to rectangular matrices. Proofs not given here can be found in e.g. [6, 31, 35, 22].

First, a key feature of ϕ_- is that when A is invertible, the ranks of $\phi_+(\mathsf{A})$ and $\phi_-(\mathsf{A}^{-1})$ coincide. Second, when A is square then the ranks of $\phi_+(\mathsf{A})$ and $\phi_-(\mathsf{A})$ differ by at most 2. The next lemma gives the complexity of converting from ϕ_-- to ϕ_+-generators; the same holds for converting back.

LEMMA 5. *Given a ϕ_--generator of length α for the matrix $\mathsf{A} \in \mathbb{K}^{n\times n}$, one can compute a ϕ_+-generator of length $\alpha+2$ for A in $O(\alpha\,\mathsf{M}(n))$ operations in $\mathbb{K}$.*

Assuming that $n = m$, partition A into blocks as

$$\mathsf{A} = \begin{bmatrix} \mathsf{A}_{1,1} & \mathsf{A}_{1,2} \\ \mathsf{A}_{2,1} & \mathsf{A}_{2,2} \end{bmatrix}, \tag{8}$$

with $\mathsf{A}_{i,j} \in \mathbb{K}^{n_i\times n_j}$, and $n_1 + n_2 = n$. Then the rank of $\phi_+(\mathsf{A}_{1,1})$ is at most the rank of $\phi_+(\mathsf{A})$; if $\mathsf{A}_{1,1}$ is invertible and has its upper-left entry non-zero then the same bound holds for $\mathsf{A}_{2,2} - \mathsf{A}_{2,1}\mathsf{A}^{-1}_{1,1}\mathsf{A}_{1,2}$.

PROPOSITION 4. *Given a ϕ_+-generator of length α for A, one can find ϕ_+-generators of length $O(\alpha)$ for all $\mathsf{A}_{i,j}$ in time $O(\alpha\,\mathsf{M}(n))$. Conversely, given ϕ_+-generators of length at most α for all $\mathsf{A}_{i,j}$, one can find a ϕ_+-generator of length $O(\alpha)$ for A in time $O(\alpha\,\mathsf{M}(n))$.*

PROPOSITION 5. *If (T, U) and (Y, Z) are ϕ_+-generators for the $n \times m$ matrices A and B, then $([\mathsf{T}\ \mathsf{Y}], [\mathsf{U}\ \mathsf{Z}])$ is a ϕ_+-generator for $\mathsf{A}+\mathsf{B}$.*

PROPOSITION 6. *If (T, U) and (Y, Z) are ϕ_+-generators for $\mathsf{A} \in \mathbb{K}^{n\times m}$ and $\mathsf{B} \in \mathbb{K}^{m\times p}$, of lenghts α and β, one can find a ϕ_+-generator of length $\alpha+\beta+1$ for AB in time $O(\gamma^{\omega-1}\mathsf{M}(q)\log(q))$, with $\gamma = \max(\alpha,\beta)$, $q = \max(n,m,p)$.*

PROOF. Let $\mathsf{V} = \mathsf{B}^t\mathsf{U}$ and $\mathsf{W} = \mathbb{Z}_{n,0}\,\mathsf{A}\,\mathbb{Z}^t_{m,0}\mathsf{Y}$; let also a (resp. b) be the lower shift of the last column of A (resp. B^t). Then the proof of [22, Prop. 2] shows that $[\mathsf{T}\ \mathsf{W}\ \mathsf{a}]$ and $[\mathsf{V}\ \mathsf{Z}\ -\mathsf{b}]$ form a ϕ_+-generator of length $\alpha+\beta+1$ for AB.

Let us detail the computation of V when $m \geq p$. We reduce to the square case by taking $\mathsf{B}' = [0\ \mathsf{B}] \in \mathbb{K}^{m\times m}$. Then $\phi_+(\mathsf{B}') = \mathsf{YZ}'^t$ with $\mathsf{Z}'^t = [0\ \mathsf{Z}^t] \in \mathbb{K}^{\beta\times m}$ and, V being read off $\mathsf{V}' = \mathsf{B}'^t\mathsf{U}$, we focus on computing the product V'.

Since B' is square, $\mathsf{B}' = \sum_{i=1}^{\beta} \mathbb{L}(\mathsf{y}_i)\mathbb{U}(\mathsf{z}'_i)$ with y_i (resp. z'_i) the ith column of Y (resp. Z'). Thus, its transpose is $\mathsf{B}'^t = \sum_{i=1}^{\beta} \mathbb{L}(\mathsf{z}'_i)\mathbb{U}(\mathsf{y}_i) = \sum_{i=1}^{\beta} \mathbb{J}\,\mathbb{U}(\mathsf{z}'_i)\,\mathbb{L}(\mathsf{y}_i)\,\mathbb{J}$, with $\mathbb{J}$ the reversal matrix of order m. Now let u_j (resp. v'_j) be the reverse of the jth column of U (resp. V'). The formula for B'^t thus gives $\mathsf{v}'_j = \sum_{i=1}^{\beta} \mathbb{U}(\mathsf{z}'_i)\mathbb{L}(\mathsf{y}_i)\,\mathsf{u}_j$. In polynomial terms this reads

$$V'_j = \left(\sum_{i=1}^{\beta} Z'_i\,(Y_iU_j \bmod x^m)\right) \operatorname{div} x^{m-1},$$

with $V'_j = \mathsf{Pol}(\mathsf{v}'_j)$, $Z'_i = \mathsf{Rev}_m(\mathsf{Pol}(\mathsf{z}'_i))$, $Y_i = \mathsf{Pol}(\mathsf{y}_i)$ and $U_j = \mathsf{Pol}(\mathsf{u}_j)$, all of those being in $\mathbb{K}[x]$. By Proposition 2, we can compute the polynomials $\sum_{i=1}^{\beta} Z'_i\,(Y_iU_j \bmod x^m)$ for $j = 1, \ldots, \alpha$ (and thus V') in time $O(\gamma^{\omega-1}\mathsf{M}(m)\log(m))$.

The case $p > m$ is treated similarly, padding B with $p-m$ zero rows. The computation of W is done similarly too, by multiplying A on the right by $\mathbb{Z}^t_{m,0}\mathsf{Y}$. Computing a and b is faster: it suffices to multiply A and B^t by a single vector. □

4.2 Solving Toeplitz-like linear systems

We now prove Theorem 1. Let $(\mathsf{T}, \mathsf{U}, \mathsf{w}) \in \mathbb{K}^{n\times\alpha} \times \mathbb{K}^{n\times\alpha} \times \mathbb{K}^n$ be the input of problem $\mathsf{LinearSystem}(\mathbb{Z}_{n,0}, \mathbb{Z}^t_{n,0}, \alpha)$. As in [22, 23] we reduce by randomization to the same problem but with "more regular" input $(\mathsf{Y}, \mathsf{Z}, \mathsf{v})$. Let B be given by $\phi_+(\mathsf{B}) = \mathsf{TU}^t$, let $\mathsf{A} = \mathbb{U}(\mathsf{y})\,\mathsf{B}\,\mathbb{L}(\mathsf{z})$ and $\mathsf{v} = \mathbb{U}(\mathsf{y})\,\mathsf{w}$ where y, z are random vectors in $\mathbb{K}^n$ with first entry 1. Then, $\mathsf{Bt} = \mathsf{w}$ if and only if $\mathsf{Au} = \mathsf{v}$ and $\mathsf{t} = \mathbb{L}(\mathsf{z})\,\mathsf{u}$. We focus on the latter problem, since t can be recovered from u in time $O(\mathsf{M}(n))$. Note that we can get $(\mathsf{Y}, \mathsf{Z}) \in \mathbb{K}^{n\times O(\alpha)} \times \mathbb{K}^{n\times O(\alpha)}$ such that $\phi_+(\mathsf{A}) = \mathsf{YZ}^t$ in time $O(\alpha\,\mathsf{M}(n))$.

By Theorem 2 in [25], there exists a non-zero polynomial Γ of $2n-2$ variables and degree n^2+n, such that if $\Gamma(y_2, \ldots, y_n, z_2, \ldots, z_n) \neq 0$, A has generic rank profile. Suppose that this is the case; with r the rank of A, define now $\mathsf{A}_r \in \mathbb{K}^{r\times r}$ as the largest non-singular leading principal submatrix of A. Given a ϕ_--generator of length α for A_r^{-1}, and using a third random vector of size n, Theorem 4 in [25] (see also [22, Prop. 3]) shows how to find a uniform random solution to the equation $\mathsf{Au} = \mathsf{v}$ (if one exists) in $O(\alpha\mathsf{M}(n))$ operations. The following proposition gives the cost of finding a suitable ϕ_--generator for A_r^{-1}, proving Theorem 1.

PROPOSITION 7. *Given a ϕ_+-generator of length α for $\mathsf{A} \in \mathbb{K}^{n\times n}$ with generic rank profile, one can compute its rank r as well as a ϕ_--generator of length at most α for A_r^{-1} in $O(\alpha^{\omega-1}\mathsf{M}(n)\log^2(n))$ operations in $\mathbb{K}$.*

PROOF. We use Kaltofen's *Leading Principal Inverse* algorithm [22, 23]; with Proposition 1, it becomes deterministic, as noted in [37, §7]. The proof of Theorem 3 in [22] shows that its cost is $T(\alpha, n) = O(\alpha^\omega)$ if $n \leq \alpha$ and otherwise

$$\begin{aligned} T(\alpha,n) &= T(\alpha,\lceil n/2\rceil) + T(\alpha,\lfloor n/2\rfloor) \\ &\quad + T_1(\alpha,n) + T_2(\alpha,n) + O(\alpha^{\omega-1}n + \alpha\mathsf{M}(n)). \end{aligned}$$

Here the term in $O(\alpha^{\omega-1}n+\alpha\mathsf{M}(n))$ bounds the cost of some conversions between ϕ_+- and ϕ_--generators (Lemma 5) and the cost of some length minimizations (Proposition 1); the terms $T_1(\alpha,n)$ and $T_2(\alpha,n)$ are the costs of two tasks we shall describe now, after recalling some notations from [22].

With $n_1 = \lceil n/2\rceil$, partition A as in (8) and A_r as

$$\mathsf{A}_r = \begin{bmatrix} \mathsf{A}_{1,1} & \mathsf{A}'_{1,2} \\ \mathsf{A}'_{2,1} & \mathsf{A}'_{2,2} \end{bmatrix}.$$

Assume that $\mathsf{A}_{1,1}$ is non-singular (else, the cost is smaller) and let $\Delta = \mathsf{A}_{2,2} - \mathsf{A}_{2,1}\mathsf{A}^{-1}_{1,1}\mathsf{A}_{1,2}$ and $\Delta' = \mathsf{A}'_{2,2} - \mathsf{A}'_{2,1}\mathsf{A}^{-1}_{1,1}\mathsf{A}'_{1,2}$.

Given ϕ_+-generators of length $O(\alpha)$ for A and $\mathsf{A}_{1,1}^{-1}$, the first task is to compute a ϕ_+-generator for Δ. Using Propositions 4, 5, 6, its cost is $T_1(\alpha, n) = O(\alpha^{\omega-1}\mathsf{M}(n)\log(n))$.

The second task is: *Given ϕ_+-generators of length $O(\alpha)$ for A, $\mathsf{A}_{1,1}^{-1}$, Δ'^{-1}, compute a ϕ_+-generator for A_r^{-1}.* Recall first that (see e.g. Theorem 5.2.3 in [36])

$$\mathsf{A}_r^{-1} = \begin{bmatrix} \mathsf{B}'_{1,1} & \mathsf{B}'_{1,2} \\ \mathsf{B}'_{2,1} & \Delta'^{-1} \end{bmatrix} \quad \text{with} \quad \begin{aligned} \mathsf{B}'_{1,2} &= -\mathsf{A}_{1,1}^{-1}\mathsf{A}'_{1,2}\Delta'^{-1} \\ \mathsf{B}'_{2,1} &= -\Delta'^{-1}\mathsf{A}'_{2,1}\mathsf{A}_{1,1}^{-1} \\ \mathsf{B}'_{1,1} &= \mathsf{A}_{1,1}^{-1} - \mathsf{B}'_{1,2}\mathsf{A}'_{2,1}\mathsf{A}_{1,1}^{-1}. \end{aligned}$$

Then we get as before $T_2(\alpha, n) = O(\alpha^{\omega-1}\mathsf{M}(n)\log(n))$. It follows that $T(\alpha, n) = O(\alpha^{\omega-1}\mathsf{M}(n)\log^2(n))$. □

4.3 Application: Padé-type approximation

We conclude by proving Corollary 1. Write $M = \sum_{i=0}^{n} m_i x^i$, with $m_n = 1$ and let $\mathbb{M} \in \mathbb{K}^{n\times n}$ be the matrix of multiplication by x modulo M. For $i \le s$, let $\mathsf{A}_i \in \mathbb{K}^{n\times\nu_i}$ be the matrix $[\mathsf{f}_i\ \mathbb{M}\mathsf{f}_i\ \cdots\ \mathbb{M}^{\nu_i-1}\mathsf{f}_i]$, where $\mathsf{f}_i = [f_{i,0}\ \cdots\ f_{i,n-1}]^t$ is the vector of coefficients of f_i. Let finally $\mathsf{A} = [\mathsf{A}_1\ \cdots\ \mathsf{A}_s] \in \mathbb{K}^{n\times(n+1)}$ and $\mathsf{A}' \in \mathbb{K}^{(n+1)\times(n+1)}$ be the matrix obtained by padding A with an $(n+1)$th row full of 1's.

Since A has non-trivial kernel, the system $\mathsf{A}'\mathsf{u} = [0\ \cdots\ 0\ 1]^t$ admits a solution, and any such solution solves our problem. The following lemma shows the Toeplitz-like structure of the matrix A'; combining it with Theorem 1 proves Corollary 1.

LEMMA 6. *One can compute in time $O(s\,\mathsf{M}(n))$ a ϕ_+-generator of length $s+2$ for A'.*

PROOF. One has $\mathbb{M} = \mathbb{Z}_{n,0} - \mathsf{m}\mathsf{e}_n^t$, with $\mathsf{m} = [m_j]_{0\le j<n}^t$ and $\mathsf{e}_n = [0\ \cdots\ 0\ 1]^t \in \mathbb{K}^n$. For $i \le s$, let $\mathsf{a}_i \in \mathbb{K}^{1\times\nu_i}$ be the last row of A_i, let $\mathsf{m}_i = [m_{n-j}]_{0\le j<\nu_i}^t$ and let $\mathsf{b}_i = [f_{i,n-j}]_{1\le j\le\nu_i}^t$ with $f_{i,-1} = 0$. Noticing that $\mathbb{L}(\mathsf{m}_i)\mathsf{a}_i^t = \mathsf{b}_i$, we see that the entries of a_i can be computed in time $O(\mathsf{M}(\nu_i))$. Thus the last row of A, which is $\mathsf{a} = [\mathsf{a}_1\ \ldots\ \mathsf{a}_s] \in \mathbb{K}^{1\times(n+1)}$, can be computed in time $O(\mathsf{M}(n))$.

Given a ϕ_+-generator of length α for A, one then obtains a ϕ_+-generator of length $\alpha+1$ for A' by adjoining the columns e_{n+1} and $[1\ \cdots\ 1]^t - \mathsf{b}$, with $\mathsf{b} = \mathbb{Z}_{n+1,0}\mathsf{a}^t$. We can thus focus on finding a generator for A.

One has $\phi_+(\mathsf{A}) = \mathsf{A} - \mathbb{M}\,\mathsf{A}\,\mathbb{Z}_{n+1,0}^t - \mathsf{m}\,\mathsf{b}^t$. Taking $\mathsf{f}_0 = 0$, we can write $\mathsf{A} - \mathbb{M}\,\mathsf{A}\,\mathbb{Z}_{n+1,0}^t$ as $\mathsf{Y}\,\mathsf{Z}^t = [\mathsf{y}_1\ \cdots\ \mathsf{y}_s][\mathsf{z}_1\ \cdots\ \mathsf{z}_s]^t$, where $\mathsf{y}_i = \mathsf{f}_i - \mathbb{M}^{\nu_{i-1}}\mathsf{f}_{i-1}$ and z_i is zero, except for a 1 at row $1+\nu_1+\cdots+\nu_{i-1}$. Since $\mathbb{M}^{\nu_{i-1}}\mathsf{f}_{i-1}$ is the coefficient vector of $x^{\nu_{i-1}}f_{i-1} \bmod M$, it can be computed in time $O(\mathsf{M}(n))$, so Y and Z can be computed in time $O(s\,\mathsf{M}(n))$. Using the remarks in the above paragraphs, this proves the lemma. □

5. THE VANDERMONDE CASE

In this section, $\mathsf{x} \in \mathbb{K}^n$ and $\psi \in \mathbb{K}$ are as in Equation (1). The operator associated with the Vandermonde structure is

$$\Delta[\mathbb{D}(\mathsf{x}), \mathbb{Z}_{n,\psi}^t](\mathsf{A}) = \mathsf{A} - \mathbb{D}(\mathsf{x})\,\mathsf{A}\,\mathbb{Z}_{n,\psi}^t.$$

With our choice of ψ, Theorem 4.3.2 in [36] shows that this operator is invertible. Moreover, given Y, Z in $\mathbb{K}^{n\times\alpha}$, Ex. 4.4.6 in [36] shows that the unique matrix $\mathsf{A} \in \mathbb{K}^{n\times n}$ such that $\Delta[\mathbb{D}(\mathsf{x}), \mathbb{Z}_{n,\psi}^t](\mathsf{A}) = \mathsf{Y}\mathsf{Z}^t$ is

$$\mathsf{A} = \mathbb{D}\big((1-\psi\,\mathsf{x}^n)^{-1}\big)\sum_{i=1}^{\alpha}\mathbb{D}(\mathsf{y}_i)\,\mathbb{V}(\mathsf{x}, n)\,\mathbb{C}(\mathsf{z}_i, \psi)^t. \qquad (9)$$

In this section, we prove Theorem 2. Following [34], we transform a Vandermonde-like system $\mathsf{A}\mathsf{u} = \mathsf{v}$ into a Toeplitz-like one. Our reduction follows that of [19]. However, that reference requires the entries of x to be pairwise distinct, *i.e.*, that $\mathbb{V}(\mathsf{x}, n)$ be invertible; else, the preprocessing step in [19, Section 2] fails. Similarly, the reduction in [36, Example 4.8.4] does not solve the problem when $\mathbb{V}(\mathsf{x}, n)$ is singular.

In the application of Subsection 5.4, this assumption does not hold. Hence, a new parameter will enter the discussion, the *multiplicity* of x, which is the maximal number of repetitions in x. Formally, if $\mathsf{x} = [x_1, \ldots, x_n]$, the multiplicity s of x is defined as $\max_{i\le n} \#\{1 \le j \le n \mid x_i = x_j\}$.

If σ is a permutation of $\{1, \ldots, n\}$, we have the relation

$$\Delta[\mathbb{D}(\mathsf{x}), \mathbb{Z}_{n,\psi}^t](\mathsf{A}) = \Sigma\,\Delta[\mathbb{D}(\sigma\cdot\mathsf{x}), \mathbb{Z}_{n,\psi}^t](\Sigma^{-1}\mathsf{A}), \qquad (10)$$

where Σ is the permutation matrix of σ. Knowing $\mathbb{D}(\mathsf{x}), \mathbb{Z}_{n,\psi}^t$-generators of A gives $\mathbb{D}(\sigma\cdot\mathsf{x}), \mathbb{Z}_{n,\psi}^t$-generators of $\Sigma^{-1}\mathsf{A}$ by permutation. Solving $\mathsf{A}\mathsf{u} = \mathsf{v}$ and $\Sigma^{-1}\mathsf{A}\mathsf{u} = \Sigma^{-1}\mathsf{v}$ are equivalent problems, so we can permute the entries of x if needed.

5.1 A multiplication problem

Up to permutation, we can (and will) suppose that x has the form $\mathsf{x} = [\mathsf{x}_1^t, \ldots, \mathsf{x}_s^t]^t$, with x_j a repetition-free vector of size ν_j, and that for $j < s$, all entries of x_j belong to x_{j+1}.

Let Y and Z be in $\mathbb{K}^{n\times\alpha}$, and let A be the unique $n\times n$ matrix such that $\Delta[\mathbb{D}(\mathsf{x}), \mathbb{Z}_{n,\psi}^t](\mathsf{A}) = \mathsf{Y}\,\mathsf{Z}^t$. Splitting A along its rows according to the above partition of x, we write

$$\mathsf{A} = \begin{bmatrix}\mathsf{A}_1^t & \cdots & \mathsf{A}_s^t\end{bmatrix}^t, \quad \text{with } \mathsf{A}_j \text{ in } \mathbb{K}^{\nu_j\times n}. \qquad (11)$$

Given vectors $\mathsf{w}_1, \ldots, \mathsf{w}_s$, with w_j in $\mathbb{K}^{\nu_j}$, we study in this subsection the cost of computing all products $\mathsf{A}_j^t\mathsf{w}_j \in \mathbb{K}^n$.

PROPOSITION 8. *On input x, ψ, Y, Z and $\mathsf{w}_1, \ldots, \mathsf{w}_s$ as above, and assuming $s \le \alpha$, one can compute all products $\mathsf{A}_j^t\,\mathsf{w}_j$ using $O(\alpha^{\omega-1}\mathsf{M}(n)\log(n))$ operations in $\mathbb{K}$.*

PROOF. Let y_i and z_i be the columns of Y and Z. We adapt the partition of x and A to the vectors y_i, writing $\mathsf{y}_i = [\mathsf{y}_{i,1}^t, \ldots, \mathsf{y}_{i,s}^t]^t$ with $\mathsf{y}_{i,j}$ in $\mathbb{K}^{\nu_j}$. Since A is given by (9), its submatrices A_j are given by $\mathsf{A}_j = \mathbb{D}\big((1-\psi\,\mathsf{x}_j^n)^{-1}\big)\,\mathsf{B}_j$, with

$$\mathsf{B}_j = \textstyle\sum_{i=1}^{\alpha}\mathbb{D}(\mathsf{y}_{i,j})\,\mathbb{V}(\mathsf{x}_j, n)\,\mathbb{C}(\mathsf{z}_i, \psi)^t.$$

For $j \le s$, let $\mathsf{f}_j = \mathbb{D}((1-\psi\,\mathsf{x}_j^n)^{-1})\,\mathsf{w}_j$. Deducing all the f_j from x and ψ and the w_j in $O(n\log(n))$ operations in $\mathbb{K}$, we are left with computing all the products $\mathsf{B}_j^t\mathsf{f}_j$.

For $i \le \alpha$ and $j \le s$, define first the vectors $\mathsf{g}_{i,j}$ in $\mathbb{K}^n$ by $\mathsf{g}_{i,j} = \mathbb{C}(\mathsf{z}_i, \psi)\,\mathbb{V}(\mathsf{x}_j, n)^t\,\mathbb{D}(\mathsf{y}_{i,j})\,\mathsf{f}_j$. It follows that $\mathsf{Pol}(\mathsf{B}_j^t\mathsf{f}_j) = \sum_{i=1}^{\alpha}\mathsf{Pol}(\mathsf{g}_{i,j})$. Next define the vectors $\mathsf{f}'_j = \mathbb{V}(\mathsf{x}_j, n+\nu_j-1)^t\,\mathsf{f}_j$ and their corresponding polynomials $F_j = \mathsf{Pol}(\mathsf{f}'_j)$ and $F'_j = \mathsf{Rev}_{n+\nu_j-1}(F_j)$. Define also $G_{i,j}$ as the unique polynomial of degree less than ν_j such that $G_{i,j}(\mathsf{x}_j) = \mathsf{y}_{i,j}$. Finally, let $G'_{i,j} = \mathsf{Rev}_{\nu_j}(G_{i,j})$, $Z_i = \mathsf{Pol}(\mathsf{z}_i)$ and $Z'_i = \mathsf{Pol}(\mathsf{Flip}(\mathsf{z}_i))$. The vector x_j being repetition-free, Lemma 1 then gives

$$\begin{aligned}\mathsf{Pol}\big(\mathsf{B}_j^t\mathsf{f}_j\big) = {} & \textstyle\sum_{i=1}^{\alpha} Z_i\,(F_j\,G'_{i,j} \operatorname{div} x^{\nu_j-1}) \bmod x^n + \\ & \psi\,\mathsf{Rev}_n\big(\textstyle\sum_{i=1}^{\alpha} Z'_i\,(F'_j\,G_{i,j} \operatorname{div} x^{\nu_j-1}) \bmod x^n\big).\end{aligned}$$

Applying the transpose of a rectangular Vandermonde matrix of size $\nu_j \times (n+\nu_j-1)$ to a vector can be done in time $O(\mathsf{M}(n)\log(n))$ by [17, Theorem 10.4]; since $s \le \alpha$, all f'_j, and thus all F_j and F'_j, can be computed in time $O(\alpha\,\mathsf{M}(n)\log(n))$. Using fast interpolation [16, Chapter 10], we compute each $G_{i,j}$ in time $O(\mathsf{M}(\nu_j)\log(\nu_j))$ and thus all $G_{i,j}$ and $G'_{i,j}$ in time $O(\alpha\,\mathsf{M}(n)\log(n))$. Proposition 3 shows eventually that all $\mathsf{Pol}(\mathsf{B}_j^t\mathsf{f}_j)$ can be computed in time $O(\alpha^{\omega-1}\mathsf{M}(n)\log(n))$, which concludes the proof. □

5.2 The case of low multiplicities

We reduce here the Vandermonde case to the Toeplitz one; our reduction adapts that of [19], allowing for repetitions in x.

PROPOSITION 9. *Let $\mathsf{x} \in \mathbb{K}^n$ and $\psi \in \mathbb{K}$ be as in Equation (1), and let s be the multiplicity of x. For $s \leq \alpha$, one can solve the problem $\mathsf{LinearSystem}(\mathbb{D}(\mathsf{x}), \mathbb{Z}^t_{n,\psi}, \alpha)$ using $O(\alpha^{\omega-1}\mathsf{M}(n)\log^2(n))$ operations in $\mathbb{K}$. The algorithm is probabilistic of type $P(3n-2, n^2+n)$.*

PROOF. Given Y and Z in $\mathbb{K}^{n\times\alpha}$ and v in $\mathbb{K}^n$, we are looking for solutions u to the system $\mathsf{Au} = \mathsf{v}$, where A is the $n \times n$ matrix such that $\Delta[\mathbb{D}(\mathsf{x}), \mathbb{Z}^t_{n,\psi}](\mathsf{A}) = \mathsf{Y}\,\mathsf{Z}^t$. We first reorder the entries of x so that $\mathsf{x} = [\mathsf{x}_1^t, \ldots, \mathsf{x}_s^t]^t$, where each $\mathsf{x}_j \in \mathbb{K}^{\nu_j}$ is repetition-free and, for $j < s$, with entries belonging to x_{j+1}; we reorder v and the rows of A and Y accordingly. Then for $j \leq s$,

$$\mathbb{D}(\mathsf{x}_j) = \mathbb{V}(\mathsf{x}_j, \nu_j)\mathbb{M}_j\mathbb{V}(\mathsf{x}_j, \nu_j)^{-1},$$

where $\mathbb{M}_j$ is the $\nu_j \times \nu_j$ companion matrix associated with the monic polynomial $m_j = \prod_{a\in\mathsf{x}_j}(x-a)$. It follows that

$$\mathbb{D}(\mathsf{x}) = \mathbb{W}\,\mathbb{M}\,\mathbb{W}^{-1}, \tag{12}$$

where $\mathbb{W}$ is block-diagonal, with blocks $\mathbb{V}(\mathsf{x}_j, \nu_j)$, and $\mathbb{M}$ is block-diagonal, with blocks $\mathbb{M}_j$. For $k \in \mathbb{N}_{>0}$ and $i \leq k$, let $\mathsf{e}_{k,i}$ be the ith unit vector in $\mathbb{K}^k$; for $j \leq s$, let m_j be the coefficient vector of $-m_j$. Then $\mathbb{M}_j = \mathbb{Z}_{\nu_j,0} + \mathsf{m}_j\mathsf{e}^t_{\nu_j,\nu_j}$ and, defining $\nu_j^* = \nu_1 + \cdots + \nu_j$,

$$\mathbb{M} = \mathbb{Z}_{n,0} + \textstyle\sum_{j=1}^s \mathsf{g}_j\mathsf{e}^t_{n,\nu_j^*}, \tag{13}$$

with $\mathsf{g}_1, \ldots, \mathsf{g}_s$ in $\mathbb{K}^n$. Using subproduct-tree techniques [16, Chapter 10], all polynomials m_j, and thus all vectors g_j, can be deduced from $\mathsf{x} \in \mathbb{K}^n$ in $O(\mathsf{M}(n)\log(n))$ operations.

With $\mathsf{B} = \mathbb{W}^{-1}\mathsf{A}$ and $\mathsf{v}' = \mathbb{W}^{-1}\mathsf{v}$, solving $\mathsf{Au} = \mathsf{v}$ amounts to solve $\mathsf{Bu} = \mathsf{v}'$. To do so in the claimed complexity, we exhibit the Toeplitz-like structure of B and bound the cost of computing v' and a generator for B. From (12) we get

$$\mathsf{B} - \mathbb{M}\,\mathsf{B}\,\mathbb{Z}^t_{n,\psi} = \mathsf{Y}'\,\mathsf{Z}^t, \qquad \mathsf{Y}' = \mathbb{W}^{-1}\mathsf{Y}.$$

Then, from (13) and the relation $\mathbb{Z}_{n,\psi} = \mathbb{Z}_{n,0} + \psi\,\mathsf{e}_{n,1}\mathsf{e}^t_{n,n}$, we deduce that $\mathsf{B} - \mathbb{Z}_{n,0}\,\mathsf{B}\,\mathbb{Z}^t_{n,0}$ is given by

$$\psi\,\mathbb{Z}_{n,0}\,\mathsf{B}\,\mathsf{e}_{n,n}\mathsf{e}^t_{n,1} \;+\; \Big(\textstyle\sum_{j=1}^s \mathsf{g}_j\mathsf{e}^t_{n,\nu_j^*}\Big)\mathsf{B}\,\mathbb{Z}^t_{n,\psi} \;+\; \mathsf{Y}'\,\mathsf{Z}^t.$$

Define the vectors $\mathsf{f}_1 = \psi\,\mathbb{Z}_{n,0}\mathsf{B}\,\mathsf{e}_{n,n}$ and, for $j \leq s$, $\mathsf{h}_j = \mathsf{B}^t\mathsf{e}_{n,\nu_j^*}$ and $\mathsf{h}'_j = \mathbb{Z}_{n,\psi}\mathsf{h}_j$. The above formula then becomes

$$\Delta[\mathbb{Z}_{n,0}, \mathbb{Z}^t_{n,0}](\mathsf{B}) = \mathsf{f}_1\mathsf{e}^t_{n,1} + \mathsf{G}\mathsf{H}'^t + \mathsf{Y}'\mathsf{Z}^t,$$

where G (resp. H') has columns g_j (resp. h'_j). The matrices $[\mathsf{f}_1\ \mathsf{G}\ \mathsf{Y}']$ and $[\mathsf{e}_{n,1}\ \mathsf{H}'\ \mathsf{Z}]$ thus form a $\mathbb{Z}_{n,0}, \mathbb{Z}^t_{n,0}$-generator of length $\alpha + s + 1 \leq 2\alpha + 1$ for B. Once this generator and v' are known, $\mathsf{Bu} = \mathsf{v}'$ can be solved within the prescribed complexity by Theorem 1. Hence it remains to estimate the cost of computing $\mathsf{v}', \mathsf{f}_1, \mathsf{Y}', \mathsf{H}'$ (for G, this was done above).

Recall e.g. from [19, Section 2] that in view of (9), multiplying A by a vector has cost $O(\alpha\,\mathsf{M}(n)\log(n))$. Since multiplication by $\mathbb{W}^{-1}$ has cost $O(\mathsf{M}(n)\log(n))$, we deduce that $\mathsf{B}\,\mathsf{e}_{n,n} = \mathbb{W}^{-1}(\mathsf{A}\,\mathsf{e}_{n,n})$, and thus f_1, can be computed in time $O(\alpha\,\mathsf{M}(n)\log(n))$. The same bound holds for computing the α columns of $\mathsf{Y}' = \mathbb{W}^{-1}\mathsf{Y}$, whereas computing $\mathsf{v}' = \mathbb{W}^{-1}\mathsf{v}$ costs only $O(\mathsf{M}(n)\log(n))$. We are thus left with computing the vectors h_j, as deducing the vectors h'_j takes time $O(\alpha)$.

For $j \leq s$, one has by definition $\mathsf{h}_j = \mathsf{A}^t\mathbb{W}^{-t}\mathsf{e}_{n,\nu_j^*}$. Defining w_j as the last row of the inverse of $\mathbb{V}(\mathsf{x}_j, \nu_j)$ and using (11), we see that h_j is in fact the vector $\mathsf{A}_j^t\mathsf{w}_j^t$. Now observe that w_j is obtained by multiplying $\mathbb{V}(\mathsf{x}_j, \nu_j)^{-1}$ by a vector *on the left*, which can be done in time $O(\mathsf{M}(\nu_j)\log(\nu_j))$ by the algorithm of [24, Section 5]; hence, all vectors w_j can be computed in time $O(\mathsf{M}(n)\log(n))$. Proposition 8 then shows that all vectors h_j can be computed in time $O(\alpha^{\omega-1}\mathsf{M}(n)\log(n))$, which concludes the proof. □

5.3 The case of high multiplicities

We conclude the proof of Theorem 2 by considering the case of high multiplicities ($s > \alpha$), reducing it to the case of low multiplicities ($s \leq \alpha$) seen in Subsection 5.2. Our reduction has cost $O(\alpha^{\omega-1}n)$ and fits in the requested bound.

As above, we are given Y and Z in $\mathbb{K}^{n\times\alpha}$ and v in $\mathbb{K}^n$, and look for solutions u to the system $\mathsf{Au} = \mathsf{v}$, where A is the $n \times n$ matrix such that $\Delta[\mathbb{D}(\mathsf{x}), \mathbb{Z}^t_{n,\psi}](\mathsf{A}) = \mathsf{Y}\,\mathsf{Z}^t$.

We assume that Y and Z have full rank (if this is not the case, replace them by minimal-length generators, for a cost of $O(\alpha^{\omega-1}n)$ by Proposition 1). Then we reorder x, to write it as $\mathsf{x} = [\mathsf{x}_1^t, \ldots, \mathsf{x}_r^t]^t$, where x_i is a vector consisting of μ_i repetitions of the same element ξ_i, so that $n = \mu_1 + \cdots + \mu_r$, and with $\xi_i \neq \xi_j$ for $i \neq j$ and $\mu_1 \geq \cdots \geq \mu_r$. Applying the same reordering to the rows of A and Y, we write

$$\mathsf{A} = \begin{bmatrix}\mathsf{A}_1^t & \cdots & \mathsf{A}_r^t\end{bmatrix}^t \quad\text{and}\quad \mathsf{Y} = \begin{bmatrix}\mathsf{Y}_1^t & \cdots & \mathsf{Y}_r^t\end{bmatrix}^t,$$

with A_i in $\mathbb{K}^{\mu_i\times n}$ and Y_i in $\mathbb{K}^{\mu_i\times\alpha}$. Hence, $\mathsf{A}_i - \mathbb{D}(\mathsf{x}_i)\mathsf{A}_i\mathbb{Z}^t_{n,\psi}$ equals $\mathsf{Y}_i\,\mathsf{Z}^t$. For $k \in \mathbb{N}_{>0}$ denote by $\mathbb{I}_k$ the $k \times k$ identity matrix. Then $\mathbb{D}(\mathsf{x}_i)$ equals $\xi_i\mathbb{I}_{\mu_i}$ and, since $\psi\xi_i{}^n \neq 1$ for all i, all matrices $\mathbb{I}_n - \xi_i\mathbb{Z}^t_{n,\psi}$ are invertible. We thus obtain the equalities $\mathsf{A}_i = \mathsf{Y}_i\mathsf{Z}^t(\mathbb{I}_n - \xi_i\mathbb{Z}^t_{n,\psi})^{-1}$ for $1 \leq i \leq r$.

Since the matrix $\mathsf{Z}^t(\mathbb{I}_n - \xi_i\mathbb{Z}^t_{n,\psi})^{-1}$ has full row rank, the linear dependencies between the rows of A_i are the same as those between the rows of Y_i.

Now let τ be such that $\mu_\tau > \alpha \geq \mu_{\tau+1}$. For $i \leq \tau$, let $\rho_i = \mathrm{rank}(\mathsf{Y}_i) = \mathrm{rank}(\mathsf{A}_i)$ and let $J_i \subset \{1, \ldots, \mu_i\}$ be such that the rows of Y_i indexed by J_i are linearly independent. Since Y_i has dimensions $\mu_i \times \alpha$ with $\alpha \leq \mu_i$, one can compute J_i in time $O(\alpha^{\omega-1}\mu_i)$, for example using [40, Prop. 2.15]. Since $\sum_{i=1}^\tau \mu_i \leq n$, the total cost is in $O(\alpha^{\omega-1}n)$.

For $i \leq \tau$, let $\mathsf{A}'_i \in \mathbb{K}^{\rho_i\times n}$ be the submatrix of A_i obtained by deleting the rows of index not in J_i; for $i > \tau$, let $\mathsf{A}'_i = \mathsf{A}_i$. Define now $\mathsf{A}' \in \mathbb{K}^{n\times n}$ by stacking the matrices A'_i and padding with $\sum_{i=1}^\tau(\mu_i - \rho_i)$ zero rows. From the right-hand side of $\mathsf{Au} = \mathsf{v}$, define $\mathsf{v}' \in \mathbb{K}^n$ in the same way as A', by zeroing out appropriate entries. Since the solution sets of $\mathsf{Au} = \mathsf{v}$ and $\mathsf{A}'\mathsf{u} = \mathsf{v}'$ coincide, we solve the latter problem.

Define the matrices Y'_i and Y' similarly to A'_i and A', by removing redundant rows and adding zero rows. Define also the vector $\mathsf{x}' \in \mathbb{K}^n$ by removing, for $i \leq \tau$, $\mu_i - \rho_i$ entries from x_i, and completing by $\sum_{i=1}^\tau(\mu_i - \rho_i)$ pairwise distinct values not already in x. Then by construction $\mathsf{A}' - \mathbb{D}(\mathsf{x}')\mathsf{A}'\mathbb{Z}^t_{n,\psi}$ equals $\mathsf{Y}'\mathsf{Z}^t$. Furthermore, the multiplicity of the vector x' is now at most α, since all ρ_i are at most α. Hence, we are left to solve a Vandermonde-like system with multiplicity at most α, the cost of which follows from Proposition 9.

5.4 Application: bivariate interpolation

Let $\{p_{i,j}\}$ be a set of points as in Equation (2) of Section 1, recalling that we assume $\nu_1 \geq \cdots \geq \nu_s > 0$; we also let $\nu_{s+1} = 0$. We conclude by proving Corollary 2 on the com-

plexity of interpolation at the points $\{p_{i,j}\}$.

We first order the input set of points. For $1 \leq i \leq s$, let P_i be the list $[p_{i,j} \mid 1 \leq j \leq \nu_i]$, and let $P = [P_1, \cdots, P_s]$. For $p = p_{i,j} \in P$, we also write $x(p) = x_i$, $y(p) = y_{i,j}$. Taking x-coordinates, for $i \leq s$, we let x_i be the vector $[x_i, \ldots, x_i] \in \mathbb{K}^{\nu_i}$ and write $\mathsf{x} = [\mathsf{x}_1, \ldots, \mathsf{x}_s] \in \mathbb{K}^n$.

We next order the monomial support. For $1 \leq j \leq \nu_1$, let $1 \leq \tau_j \leq s$ be such that $\nu_{\tau_j} \geq j > \nu_{\tau_j+1}$ holds, let B_j be the list $[x^{i-1}y^{j-1} \mid 1 \leq i \leq \tau_j]$ and let $B = [B_1, \cdots, B_{\nu_1}]$. Letting $\mathsf{Span}(B)$ be the vector space $\{\sum_{b\in B} f_b b \mid f_b \in \mathbb{K}\}$, we are thus interested in the evaluation map $F \in \mathsf{Span}(B) \mapsto [F(p)]_{p\in P}$ and its inverse.

Let $\mathsf{A} = [b(p)]_{p\in P, b\in B} \in \mathbb{K}^{n\times n}$ be the matrix of this map, with rows indexed by P and columns by B. Let us write $\mathsf{A} = [\mathsf{A}_1 \ \cdots \ \mathsf{A}_{\nu_1}]$, with $\mathsf{A}_j = [b(p)]_{p\in P, b\in B_j} \in \mathbb{K}^{n\times\tau_j}$. Then $\Delta[\mathbb{D}(\mathsf{x}), \mathbb{Z}^t_{n,0}](\mathsf{A})$ can be written $\mathsf{G}\mathsf{H}^t = [\mathsf{g}_1 \cdots \mathsf{g}_{\nu_1}][\mathsf{h}_1 \cdots \mathsf{h}_{\nu_1}]^t$ where h_j is zero, except for a 1 at row $1 + \sum_{k=1}^{j-1} \tau_k$, and $\mathsf{g}_1 = [1, \ldots, 1]^t$ and $\mathsf{g}_j = [y(p)^{j-1} - y(p)^{j-2}x(p)^{\tau_{j-1}}]^t_{p\in P}$ for $j > 1$.

The matrices G, H can be computed in time $O(\nu_1 n \log(n))$; Theorem 2 then shows that the system $\mathsf{A}\mathsf{f} = \mathsf{v}$ can be solved in time $O(\nu_1^{\omega-1}\mathsf{M}(n)\log^2(n))$, where f is the coefficient vector of the polynomial to interpolate and v is the value vector.

To prove Corollary 2, we prove another upper bound of the form $O(s^{\omega-1}\mathsf{M}(n)\log^2(n))$. This is done by reordering the entries of B. For $i \leq s$, write $B'_i = [x^{i-1}y^{j-1} \mid 1 \leq j \leq \nu_i]$, and let $B' = [B'_1, \cdots, B'_s]$, so that B' and B coincide up to order. We then define the matrix $\mathsf{A}' = [b(p)]_{p\in P, b\in B'}$, which equals A up to reordering the columns. Using now the y-coordinates of the points in P to describe the Vandermonde structure of A' leads as above to the claimed bound.

6. REFERENCES

[1] B. Beckermann. A reliable method for computing M-Padé approximants on arbitrary staircases. *J. Comput. Appl. Math.*, 40(1):19–42, 1992.

[2] B. Beckermann and G. Labahn. A uniform approach for the fast computation of matrix-type Padé approximants. *SIAM J. Matrix Anal. Appl.*, 15(3):804–823, 1994.

[3] B. Beckermann and G. Labahn. Fraction-free computation of matrix rational interpolants and matrix GCDs. *SIAM J. Matrix Anal. Appl.*, 22(1):114–144, 2000.

[4] M. Ben-Or and P. Tiwari. A deterministic algorithm for sparse multivariate polynomial interpolation. In *20th Annual ACM Symp. Theory Comp.*, pages 301–309, 1988.

[5] D. Bini and V. Y. Pan. *Polynomial and Matrix Computations, volume 1: Fundamental Algorithms*. Birkhäuser, 1994.

[6] R. R. Bitmead and B. D. O. Anderson. Asymptotically fast solution of Toeplitz and related systems of linear equations. *Linear Algebra Appl.*, 34:103–116, 1980.

[7] A. Bostan, C.-P. Jeannerod, and É. Schost. Solving structured linear systems with large displacement rank. Technical report.

[8] J. Canny, E. Kaltofen, and Y. Lakshman. Solving systems of non-linear polynomial equations faster. In *ISSAC'89*, pages 121–128. ACM, 1989.

[9] D. G. Cantor and E. Kaltofen. On fast multiplication of polynomials over arbitrary algebras. *Acta Informatica*, 28(7):693–701, 1991.

[10] D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions. *J. Symb. Comput.*, 9(3):251–280, 1990.

[11] R. A. DeMillo and R. J. Lipton. A probabilistic remark on algebraic program testing. *Inf. Process. Lett.*, 7(4):193–195, 1978.

[12] J.-G. Dumas, T. Gautier, and C. Pernet. Finite field linear algebra subroutines. In *ISSAC'02*, pages 63–74. ACM, 2002.

[13] W. Eberly, M. Giesbrecht, P. Giorgi, A. Storjohann, and G. Villard. Solving sparse rational linear systems. In *ISSAC'06*, pages 63–70. ACM, 2006.

[14] S. Gao, V. M. Rodrigues, and J. Stroomer. Gröbner basis structure of finite sets of points, preprint, 2003.

[15] M. Gasca and T. Sauer. Polynomial interpolation in several variables. *Adv. Comput. Math.*, 12(4):377–410, 2000.

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

[17] J. von zur Gathen and V. Shoup. Computing Frobenius maps and factoring polynomials. *C. Complex.*, 2(3):187–224, 1992.

[18] P. Giorgi, C.-P. Jeannerod, and G. Villard. On the complexity of polynomial matrix computations. In *ISSAC'03*, pages 135–142. ACM, 2003.

[19] I. Gohberg and V. Olshevsky. Complexity of multiplication with vectors for structured matrices. *Linear Algebra Appl.*, 202:163–192, 1994.

[20] O. H. Ibarra, S. Moran, and R. Hui. A generalization of the fast lup matrix decomposition algorithm and applications. *J. Algorithms*, 3(1):45–56, 1982.

[21] T. Kailath, S. Y. Kung, and M. Morf. Displacement ranks of matrices and linear equations. *J. Math. Anal. Appl.*, 68(2):395–407, 1979.

[22] E. Kaltofen. Asymptotically fast solution of Toeplitz-like singular linear systems. In *ISSAC'94*, pages 297–304. ACM, 1994.

[23] E. Kaltofen. Analysis of Coppersmith's block Wiedemann algorithm for the parallel solution of sparse linear systems. *Mathematics of Computation*, 64(210):777–806, 1995.

[24] E. Kaltofen and Y. Lakshman. Improved sparse multivariate polynomial interpolation algorithms. In *ISSAC'88*, volume 358 of *LNCS*. Springer Verlag, 467–474.

[25] E. Kaltofen and D. Saunders. On Wiedemann's method of solving sparse linear systems. In *AAECC-9*, volume 539 of *LNCS*, pages 29–38. Springer Verlag, 1991.

[26] I. Kaporin. The aggregation and cancellation techniques as a practical tool for faster matrix multiplication. *Theor. Comput. Sci.*, 315(2-3):469–510, 2004.

[27] G. Labahn, D. K. Choi, and S. Cabay. The inverses of block Hankel and block Toeplitz matrices. *SIAM J. Comput.*, 19(1):98–123, 1990.

[28] J. Laderman, V. Y. Pan, and X.-H. Sha. On practical algorithms for accelerated matrix multiplication. *Linear Algebra Appl.*, 162-164:557–588, 1992.

[29] D. Lazard. Ideal bases and primary decomposition: the case of two variables. *J. Symb. Comput.*, 1:261–270, 1985.

[30] M. Morf. *Fast algorithms for multivariable systems*. PhD thesis, Stanford University, 1974.

[31] M. Morf. Doubling algorithms for Toeplitz and related equations. In *IEEE Conference on Acoustics, Speech, and Signal Processing*, pages 954–959, 1980.

[32] T. Mulders. On short multiplications and divisions. *AAECC*, 11(1):69–88, 2000.

[33] M. Nüsken and M. Ziegler. Fast multipoint evaluation of bivariate polynomials. In *ESA 2004*, number 3222 in LNCS, pages 544–555. Springer, 2004.

[34] V. Y. Pan. On computations with dense structured matrices. *Math. Comp.*, 55(191):179–190, 1990.

[35] V. Y. Pan. Parametrization of Newton's iteration for computations with structured matrices and applications. *Computers Math. Applic.*, 24(3):61–75, 1992.

[36] V. Y. Pan. *Structured Matrices and Polynomials*. Birkhäuser Boston Inc., 2001.

[37] V. Y. Pan and A. Zheng. Superfast algorithms for Cauchy-like matrix computations and extensions. *Linear Algebra Appl.*, 310:83–108, 2000.

[38] A. Schönhage and V. Strassen. Schnelle Multiplikation großer Zahlen. *Computing*, 7:281–292, 1971.

[39] J. T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. ACM*, 27(4):701–717, October 1980.

[40] A. Storjohann. *Algorithms for matrix canonical forms*. PhD thesis, ETH, Zürich, 2000.

[41] A. Storjohann. Notes on computing minimal approximant bases. Technical report, Symbolic Computation Group, University of Waterloo, 2006.

[42] V. Strassen. Gaussian elimination is not optimal. *Numerische Mathematik*, 13:354–356, 1969.

[43] M. Van Barel and A. Bultheel. A general module theoretic framework for vector M-Padé and matrix rational interpolation. *Numer. Algorithms*, 3:451–461, 1992.

[44] R. Zippel. Probabilistic algorithms for sparse polynomials. In *EUROSAM' 79*, volume 72 of *LNCS*. Springer Verlag, 1979.

[45] R. Zippel. Interpolating polynomials from their values. *J. Symb. Comp.*, 9(3):375–403, 1990.