Electronic Transactions on Numerical Analysis.
Volume 60, pp. 123–135, 2024.

ISSN 1068–9613.
DOI: 10.1553/etna_vol60s123

ETNA
Kent State University and
Johann Radon Institute (RICAM)

DIMENSIONAL REDUCTION FOR MULTIVARIATE LAGRANGE POLYNOMIAL INTERPOLATION PROBLEMS*

M. ERRACHID†, A. ESSANHAJI†, AND A. MESSAOUDI‡

Abstract. In this work we propose a theoretical and practical method to transform the multivariate Lagrange polynomial interpolation problem into a univariate problem. This transformation allows a wide exploitation of all one-variable polynomial Lagrange interpolation schemes such as Newton's scheme or split differences, etc. Numerical comparison with other existing methods will be studied.

Key words. polynomial interpolation, multivariate Lagrange polynomial interpolation problem

AMS subject classifications. 65D05, 41A05, 41A63, 41A10, 97N50

1. Introduction. Multivariate polynomial interpolation is a very rich area of applied mathematics. It is used to estimate unknown values in a multidimensional data set and finds applications in various fields such as geostatistics [10] or cryptography [12].

In this work, $\mathbb{N}^*$ is the set of non-null integers, and $\mathbb{K}$ is a commutative field, either infinite or of large cardinality, and $d, n \in \mathbb{N}^*$. By $\Pi^d = \mathbb{K}[x_1, \ldots, x_d]$ we denote the algebra of polynomials in d variables, and Π_k^d is the $\binom{k+d}{d}$-dimensional subspace of polynomials of total degree less than or equal to k, where $k \in \mathbb{N}$.

Given a finite interpolation set $Z = \{z_1, \ldots, z_n\} \subset \mathbb{K}^d$ of n pairwise distinct nodes, the Lagrange polynomial interpolation problem consists in finding, for a given data vector $R = (r_1, \ldots, r_n) \in \mathbb{K}^n$, a polynomial $P \in \Pi^d$ such that

$$P(Z) = R, \quad \text{that is,} \quad P(z_k) = r_k, \quad k = 1, \ldots, n. \tag{1.1}$$

We will then say that P is an interpolating polynomial for (Z, R). More precisely, Z is called poised or correct or unisolvent [2, 8, 14] for a subspace $\mathcal{P}$ of Π^d if the Lagrange interpolation problem (1.1) has a unique interpolating polynomial in $\mathcal{P}$ for each given data vector $R \in \mathbb{K}^n$. Which means, in other words, that the function

$$P \in \mathcal{P} \mapsto (P(z_1), \ldots, P(z_n)) \in \mathbb{K}^n$$

is a linear isomorphism. It is then necessary that $\dim \mathcal{P} = n$. Such a subspace is called an interpolation space with respect to the interpolation set Z.

It is well known that in the univariate case ($d = 1$) the Lagrange polynomial interpolation problem with respect to n distinct points is always uniquely solvable if we take $\mathcal{P}$ the space of polynomials of degree less than or equal to $n - 1$. In several variables, however, the situation is much more difficult [1, 3, 5, 6, 8, 9, 11, 13, 15]. In this paper we show that by using linear functionals, we can transform the multivariate Lagrange polynomial interpolation problem (1.1) into a univariate one.

The rest of the paper is organized as follows. In Section 2 we theoretically justify the existence of a linear functional that transforms the multivariate Lagrange polynomial interpolation problem into a univariate problem. In Section 3 we present a deterministic

*Received May 28, 2023. Accepted January 9, 2024. Published online on March 18, 2024. Recommended by F. Marcellan.

†LabMIA-SI, Mohammed V University in Rabat, Centre Régional des Métiers de l'Enseignement et de la Formation (CRMEF) de Rabat, 1 Avenue Allal Alfassi, Madinat Al Irfane, B.P. 6210, 10 000 Rabat, Maroc (errachid.m@gmail.com).

‡LabMIA-SI, Mohammed V University in Rabat, Ecole Normale Supérieure, Av. Mohammed Belhassan El Ouazzani, B.P. 5118, Takaddoum, Rabat, Maroc.

numerical method valid in the real case, and in Section 4 we propose a probabilistic approach. In Section 5 some algorithms are given. A comparison of our methods with other existing ones is discussed in Section 6 before concluding.

2. Main theoretical results. We start this section with the following classical linear algebra result.

LEMMA 2.1. *Let* $m \in \mathbb{N}^*$*, and suppose that* $\mathrm{Card}(\mathbb{K}) \geq m$*. Let* E *be a* $\mathbb{K}$*-vector space, and let* $E_1, \ldots, E_m$ *be* m *strict subspaces, i.e.,* $E_k \neq E$*,* $k = 1, \ldots, m$*. Then, it holds that* $\bigcup_{k=1}^{m} E_k \neq E$*.*

Proof. Let us reason by the absurd and assume the existence of m strict subspaces $E_1, \ldots, E_m$ of E such that $\bigcup_{k=1}^{m} E_k = E$. We can, in addition, suppose that $E_m \not\subset \bigcup_{k=1}^{m-1} E_k$, because otherwise we reduce the number of considered subspaces. On the other hand, we have also $\bigcup_{k=1}^{m-1} E_k \not\subset E_m$. Therefore, there are two vectors x and y such that

$$x \in E_m \backslash \bigcup_{k=1}^{m-1} E_k \qquad \text{and} \qquad y \in \bigcup_{k=1}^{m-1} E_k \backslash\, E_m.$$

Then, for all $\lambda \in \mathbb{K}$, $\lambda x + y \in \bigcup_{k=1}^{m-1} E_k$. This means that for each $\lambda \in \mathbb{K}$, there exists $i_\lambda \in \{1, \ldots, m-1\}$ such that $\lambda x + y \in E_{i_\lambda}$. As then $\mathrm{Card}(\mathbb{K}) > m-1$, there exist two distinct scalars λ, μ such that $i_\lambda = i_\mu$. We deduce the existence of an index $i \in \{1, \ldots, m-1\}$ such that $(\lambda - \mu)x \in E_i$, thus $x \in E_i$, which is absurd. Hence the result follows. ∎

In the remainder of this section we assume that the field $\mathbb{K}$ is either infinite or of cardinality greater than $\frac{1}{2}n(n-1)$.

THEOREM 2.2. *There exists a linear functional* f *on* $\mathbb{K}^d$ *satisfying the following condition*

$$i \neq j \Longrightarrow f(z_i) \neq f(z_j), \qquad \forall i, j = 1, \ldots, n,$$

which means that f *separates the nodes of the interpolation set* Z*, and therefore* $f(Z)$ *is a subset of* $\mathbb{K}$ *with* n *distinct elements.*

Proof. We introduce the dual algebraic space of $\mathbb{K}^d$ and denote $E = \left(\mathbb{K}^d\right)^* := \mathcal{L}(\mathbb{K}^d, \mathbb{K})$. For each $i, j = 1, \ldots, n$ such that $i < j$, we take $a_{i,j} = z_j - z_i$ and consider the annihilator

$$A_{i,j} := \{\varphi \in E : \varphi(a_{i,j}) = 0\}.$$

Since $a_{i,j} \neq 0$, $A_{i,j}$ is a hyperplane of E, and therefore it is a strict subspace of E. Thus, taking into account the considered assumption on the cardinality of the field $\mathbb{K}$, we can apply the above lemma and deduce that $\bigcup_{1 \leq i < j \leq n} A_{i,j} \neq E$. So, there exists a linear functional f on $\mathbb{K}^d$ satisfying

$$\forall i, j = 1, \ldots, n,\ i < j : \quad f(a_{i,j}) \neq 0.$$

Hence the theorem follows. ∎

REMARK 2.3. In the case where the field is the reals, $\mathbb{K} = \mathbb{R}$, the previous result can be deduced from the Hahn-Banach theorem.

For the so determined functional f, we set

$$t_k = f(z_k), \qquad k = 1, \ldots, n.$$

We consider f a one-to-one mapping between the sets $Z = \{z_1, \dots, z_n\} \subset \mathbb{K}^d$ and $T = \{t_1, \dots, t_n\} \subset \mathbb{K}$. Thus, given the vector data

$$R = (r_1, \dots, r_n) \in \mathbb{K}^n,$$

the resolution of the multivariate Lagrange polynomial interpolation problem (1.1) with respect to (Z, R) can be reduced to the univariate Lagrange polynomial interpolation problem with respect to (T, R) that consists in searching for polynomials $q \in \Pi^1$ satisfying

$$q(t_k) = r_k, \qquad k = 1, \dots, n. \tag{2.1}$$

It is well-known that the problem (2.1) has a unique solution of degree $< n$ and that all other solutions are obtained by the addition of a multiple of the polynomial $\prod_{k=1}^{n}(x - t_k)$.

PROPOSITION 2.4. *Let*

$$R = (r_1, \dots, r_n) \in \mathbb{K}^n.$$

Let q_R be the unique Lagrange interpolating polynomial of degree $< n$ with respect to (T, R). Then, the polynomial

$$p_R = q_R \circ f \in \Pi^d_{n-1},$$

is an interpolating polynomial with respect to (Z, R).

Proof. f is a linear functional on $\mathbb{K}^d$, so there exist n scalars $\alpha_1, \dots, \alpha_d$ such that

$$f(x_1, \dots, x_d) = \sum_{k=1}^{d} \alpha_k x_k, \qquad \forall (x_1, \dots, x_d) \in \mathbb{K}^d.$$

On the other hand, $q_R \in \Pi^1_{n-1}$, hence, there exist n scalars $\lambda_0, \dots, \lambda_{n-1}$ such that

$$q_R(t) = \sum_{i=0}^{n-1} \lambda_i t^i.$$

It follows that

$$p_R(x_1, \dots, x_d) = \sum_{i=0}^{n-1} \lambda_i \left(\sum_{k=1}^{d} \alpha_k x_k \right)^i.$$

Thus, we observe that p_R is a d-variate polynomial of total degree less than or equal to $n - 1$. In addition, it is easy to see that

$$p_R(z_k) = q_R(f(z_k)) = q_R(t_k) = r_k, \qquad k = 1, \dots, n,$$

which shows that p_R is an interpolating polynomial with respect to (Z, R). $\square$

To obtain an interpolation space with respect to Z, we just use the following obvious result:

LEMMA 2.5. *The functions $(f^s(x_1, \dots, x_d))_{0 \le s \le n-1}$ span an n-dimensional subspace of Π^d_{n-1}.*

Proof. We need to show that the family is linearly independent in the vector space Π^d_{n-1}. We employ the same notation from the previous proof. Let $\lambda_0, \dots, \lambda_{n-1}$ be n scalars such that

$$\sum_{s=0}^{n-1} \lambda_s f^s = 0.$$

Then,

$$\sum_{s=0}^{n-1} \lambda_s t_k^s = \sum_{s=0}^{n-1} \lambda_s f^s(z_k) = 0, \qquad k = 1, \dots, n,$$

and, as the values $t_k = f(z_k)$, $k = 1, \dots, n$, are n pairwise distinct scalars, the Vandermonde matrix

$$(t_k^s)_{\substack{1\le k\le n\\ 0\le s\le n-1}} = \begin{bmatrix} 1 & t_1 & \dots & t_1^{n-1} \\ 1 & t_2 & \dots & t_2^{n-1} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & t_n & \dots & t_n^{n-1} \end{bmatrix}$$

is invertible, and so the scalars $\lambda_0, \dots, \lambda_{n-1}$ are all null, and the result follows. □

We can now deduce the following main result of this section:

THEOREM 2.6. *The subspace*

$$\Pi^d(f, n) = \operatorname{span}(1, f, \dots, f^{n-1}) \quad \textit{of} \quad \Pi^d_{n-1}$$

is an interpolating space with respect to Z.

REMARK 2.7.
1. If f is a linear functional which separates the nodes of Z, then, for every nonzero scalar σ, the linear functional σf also separates the nodes of Z, and we have $\Pi^d(f, n) = \Pi^d(\sigma f, n)$.
2. In general, the process described above allows us to construct several interpolation spaces with respect to Z, considering linearly independent functionals separating the nodes of Z.

In the following section, we give a numerical and efficient method to construct a linear functional that separates the nodes of the interpolation set Z.

3. Node separation by a linear functional: a deterministic approach. In this section, we consider the special case $\mathbb{K} = \mathbb{R}$. We consider the real coordinates of the nodes of Z,

$$z_k = (x_{1,k}, \dots, x_{d,k}), \qquad k = 1, \dots, n.$$

We describe a numerical method to determine a linear functional separating the nodes of Z. Let us start with the bivariate case, i.e., $d = 2$. Thus, we look for two reals α_1 and α_2 such that the linear functional

$$f(x_1, x_2) = \alpha_1 x_1 + \alpha_2 x_2,$$

separates the nodes of Z. Let

$$X = \{|x_{1,i} - x_{1,j}| : 1 \le i < j \le n \mid x_{1,i} \ne x_{1,j}\},$$
$$X' = \{|x_{2,i} - x_{2,j}| : 1 \le i < j \le n \mid x_{2,i} \ne x_{2,j}\}.$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

Since $z_k = (x_{1,k}, x_{2,k})$, $k = 1, \ldots, n$, are n pairwise distinct nodes, X or X' is not empty. Thus, three cases arise.

The first case: $X = \emptyset$, so that $x_{2,k}$, $k = 1, \ldots, n$, are n pairwise distinct scalars. In this case we take $f(x_1, x_2) = x_2$, i.e., $\alpha_1 = 0$ and $\alpha_2 = 1$.

The second case: $X' = \emptyset$. We choose $f(x_1, x_2) = x_1$, i.e., $\alpha_1 = 1$ and $\alpha_2 = 0$.

The last case: $X \neq \emptyset$ and $X' \neq \emptyset$. Let $m = \min X$ and $M = \max X'$; these are non-negative numbers. Then we take

$$\alpha_1 = 1 \qquad \text{and} \qquad \alpha_2 \in \left]0, \frac{m}{M}\right[.$$

Indeed, if $i, j \in \{1, \ldots, n\}$ such that $f(z_i) = f(z_j)$, then

$$|x_{1,i} - x_{1,j}| = \alpha_2 \, |x_{2,i} - x_{2,j}| \, .$$

So, if $x_{1,i} \neq x_{1,j}$, then $x_{2,i} \neq x_{2,j}$ and also $|x_{1,i} - x_{1,j}| \in X$ and $|x_{2,i} - x_{2,j}| \in X'$. We obtain

$$m \leq |x_{1,i} - x_{1,j}| = \alpha_2 \, |x_{2,i} - x_{2,j}| \leq \alpha_2 M < m,$$

which is absurd. Thus, $x_{1,i} = x_{1,j}$, and as $\alpha_2 \neq 0$, it follows that $x_{2,i} = x_{2,j}$. We deduce that $z_i = z_j$, which justifies our choice.

Now suppose that $d \geq 3$. Let us reason by recurrence under the assumption that we can separate each finite non-empty subset of $\mathbb{R}^{d-1}$ by a linear functional on this space. We simply set

$$\widetilde{z}_k = (x_{1,k}, \ldots, x_{d-1,k}) \in \mathbb{R}^{d-1}, \qquad k = 1, \ldots, n,$$

and we consider

$$\widetilde{Z} = \{\widetilde{z}_k : k = 1, \ldots, n\}.$$

We notice that $1 \leq n' = \mathrm{Card}(\widetilde{Z}) \leq n$. According to the induction hypothesis, there exists a linear functional g on $\mathbb{R}^{d-1}$ satisfying

$$\widetilde{z}_i \neq \widetilde{z}_j \Longrightarrow g(\widetilde{z}_i) \neq g(\widetilde{z}_j), \qquad i, j = 1, \ldots, n,$$

with the convention that $g = 0$ if $n' = 1$. Now we show that there exists a real α such that the linear functional f defined on $\mathbb{R}^d$ by

$$f(x_1, \ldots, x_d) = g(x_1, \ldots, x_{d-1}) + \alpha x_d$$

separates the nodes of Z. To show this, we proceed as in the bivariate case, and we will distinguish three cases.

The first case: suppose that $n' = n$, i.e., $\widetilde{z}_k$, for $k = 1, \ldots, n$, are n pairwise distinct points in $\mathbb{R}^{d-1}$. In this case we take $\alpha = 0$. Indeed, if $f(z_k) = g(\widetilde{z}_k)$, for all $k = 1, \ldots, n$, then

$$i \neq j \Longrightarrow g(\widetilde{z}_i) \neq g(\widetilde{z}_j) \Longrightarrow f(z_i) \neq f(z_j), \qquad i, j = 1, \ldots, n.$$

The second case: $n' = 1$, i.e., $\widetilde{z}_1 = \widetilde{z}_2 = \cdots = \widetilde{z}_n$. Then, the n scalars $x_{d,k}$, $k = 1, \ldots, n$, are distinct. In this case, with the convention $g = 0$, we can just choose $\alpha = 1$, i.e.,

$$f(x_1, \ldots, x_d) = x_d.$$

The last case: $1 < n' < n$. Let

$$X = \{|g(\widetilde{z}_i) - g(\widetilde{z}_j)| : \ 1 \le i < j \le n \text{ and } \widetilde{z}_i \neq \widetilde{z}_j\},$$
$$X' = \{|x_{d,i} - x_{d,j}| : \ 1 \le i < j \le n \text{ and } x_{d,i} \neq x_{d,i}\}.$$

X and X' are two finite non-empty subsets of $\mathbb{R}_+^*$. We set

$$m = \min X, \qquad M = \max X',$$

which are two non-negative reals. We show that any choice of

$$\alpha \in \left]0, \frac{m}{M}\right[$$

is convenient. Indeed, if $i, j \in \{1, \dots, n\}$ satisfy $f(z_i) = f(z_j)$, i.e.,

$$g(\widetilde{z}_i) - g(\widetilde{z}_j) = \alpha(x_{d,i} - x_{d,j}),$$

then, if $\widetilde{z}_i \neq \widetilde{z}_j$ and according to the definition of g, we get $g(\widetilde{z}_i) \neq g(\widetilde{z}_j)$, hence, $x_{d,i} \neq x_{d,j}$. Therefore,

$$m \le |g(\widetilde{z}_i) - g(\widetilde{z}_j)| = \alpha\, |x_{d,i} - x_{d,j}| \le \alpha M < m,$$

which is absurd. Thus, $\widetilde{z}_i = \widetilde{z}_j$, and as $\alpha \neq 0$, $x_{d,i} = x_{d,j}$, so it follows that $z_i = z_j$. Hence, the linear functional f on $\mathbb{R}^d$ separates the nodes of Z.

By the following example we show that the previous study supports and generalizes the idea developed by Dharm and Amit in their paper [4].

Let $d = 2$, n_1 and n_2 be two integers > 1, and let

$$Z = \{1, \dots, n_1\} \times \{1, \dots, n_2\}$$

be the particular grid of $\mathbb{R}^2$ considered in [4]. Then, using the above notation, we obtain

$$m = \min\{|i - j| : 1 \le i < j \le n_1\} = 1,$$

and

$$M = \max\{|i - j| : 1 \le i < j \le n_2\} = n_2 - 1.$$

We then take

$$\alpha = \frac{1}{n_2} \in \left]0, \frac{m}{M}\right[= \left]0, \frac{1}{n_2 - 1}\right[.$$

So, using Remark 2.7, we deduce that the linear functional $f(x_1, x_2) = n_2 x_1 + x_2$ separates the nodes of the grid Z. This allows us to find the same interpolation space determined in [4].

4. Node separation: a probabilistic approach. In this section, we present another approach to construct an algorithm based on probabilistic aspects. As proved in Section 3, in order to construct an interpolation space with respect to the set of nodes Z using the interpolation algorithms available in the one-dimensional case, it is sufficient to find a linear functional that separates the nodes of Z. Our new approach relies on the fact that hyperplanes are subsets of $\mathbb{R}^d$ of Lebesgue measure zero.

THEOREM 4.1. *A hyperplane of $\mathbb{R}^d$ (equipped with the Lebesgue measure) is of measure zero. In particular if ϕ is a non-zero linear functional of $\mathbb{R}^d$, then its kernel is of measure zero.*

This result allows us to state that if we arbitrarily choose a linear functional ϕ on $\mathbb{R}^d$, then it is almost certain that the nodes satisfy

$$z_j - z_i \notin \ker \phi, \qquad 1 \le i < j \le n,$$

which means that ϕ separates the nodes of Z.

In fact, and in order to optimize the calculations, we can choose the linear functional ϕ in the form

$$\phi(x_1, \ldots, x_d) = \alpha_1 x_1 + \ldots + \alpha_d x_d,$$

where the coefficients $\alpha_1, \ldots, \alpha_d$ are randomly chosen integers.

In the concrete construction of the algorithm, the only change to be made to the previous algorithms consists in writing an algorithm that randomly chooses a linear functional by imposing that the choice of the coefficients be made with integers. Then we make sure that this yields a correct choice by verifying that the images of

$$z_j - z_i, \qquad 1 \le i < j \le n,$$

are not null. If this is not the case, then we repeat the process until we find such a linear functional. In general, one try is enough to find a good linear functional.

5. Some algorithms. In this section, we provide some algorithms to construct a linear functional separating the nodes of the interpolation set Z in both deterministic and probabilistic forms. Then we adopt Newton's scheme with divided differences in the implementation of our multivariate Lagrange polynomial interpolation algorithm.

5.1. Construction of a linear functional of separation: a deterministic method. We follow the scheme of the theoretical construction; we start by treating the case $d = 2$ and using python notation; cf. Algorithm 1.

Algorithm 1 Separation nodes in the case $d = 2$.

Compute
  $X = \{|x_{1,i} - x_{1,j}| : 1 \le i < j \le n \mid x_{1,i} \ne x_{1,j}\}$
  $X' = \{|x_{2,i} - x_{2,j}| : 1 \le i < j \le n \mid x_{2,i} \ne x_{2,j}\}$
if $X = \emptyset$ **then**
  Return $f : (x, y) :\to y$
else if $X^{'} = \emptyset$ **then**
  Return $f : (x, y) :\to x$
else
  Compute
  $m = \min X,\ M = \max X'$
  $\alpha = \frac{m}{2M}$ (such that $\alpha \in \left]0, \frac{m}{M}\right[$)
end if
Return $f : (x, y) \to x + \alpha y$

Now we employ the recursive construction presented in Section 3 for the case $d > 2$. The algorithm presented below, Algorithm 2, is recursive; the terminal case is the case $d = 2$.

5.2. Linear functional of separation with random approach. In Algorithm 3, we present a way to construct the linear functional that separates the nodes of Z, based on

ETNA
Kent State University and
Johann Radon Institute (RICAM)

Algorithm 2 Separation nodes for general case $d > 2$

$n = \text{length}(Z)$
if $d = 2$ **then**
  Return separation case $d = 2$
end if
Compute
  $\widetilde{Z} = \{\widetilde{z}_k : k = 1, \ldots, n\}$, where $\widetilde{z}_k = (x_{1,k}, \ldots, x_{d-1,k}) \in \mathbb{R}^{d-1}$,
  $p = \text{length}(\widetilde{Z})$
if $n = 1$ **then**
  Return $f : (x_1, \ldots, x_d) \to x_d$
else
  g = separation general case $d - 1$ (with $\widetilde{Z}$)
  if $n = p$ **then**
   Return g
  else
   Compute
   $X = \{|g(\widetilde{z}_i) - g(\widetilde{z}_j)| : 1 \le i < j \le n \text{ and } \widetilde{z}_i \neq \widetilde{z}_j\}$
   $X' = \{|x_{d,i} - x_{d,j}| : 1 \le i < j \le n \text{ and } x_{d,i} \neq x_{d,j}\}$
   $m = \min X, \ M = \max X'$
   $\alpha = \frac{m}{2M}$ (such that $\alpha \in \left]0, \frac{m}{M}\right[$)
  end if
end if
Return $f : (x_1, \ldots, x_d) \to g(x_1, \ldots, x_{d-1}) + \alpha x_d$

Algorithm 3 Random separation linear functional.

Input: Z: data vectors of nodes
Output: f: linear functional, $f(Z)$
$d = \text{length}(Z[0])$
Choose d integers in a random way: $a_1, \ldots, a_d$
Define $f : (x_1, \ldots, x_d) \to a_1 x_1 + \cdots + a_d x_d$
Check that f is a separation linear functional, otherwise repeat the process.
Return f, $f(Z)$

Theorem 4.1, and we impose, in addition, that the coefficients of the chosen linear functional are natural numbers.

REMARK 5.1. In Algorithm 3, in order to gain computational performance in the sequel, when testing the separation constraint, we also store $f(Z)$ since we will need this set during the one-variable interpolation phase.

5.3. The DRMVLPIA. Once we have a linear functional separation, as illustrated in Theorem 2.6, we are able to provide an interpolation space where Z is well-poised. In this section, we give concrete constructions of interpolation polynomials, using both the deterministic and the random approaches for node separations. To do this, we couple Horner's algorithm with a divided differences algorithm to compute an interpolating polynomial for (Z, R).

Firstly, we present the divided differences algorithm in Algorithm 4. Next, Algorithm 5 represents Newton's one-dimensional algorithm for computing interpolation polynomials using the Horner scheme.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

Algorithm 4 Divided differences.

Input:: x, y: two data vectors
Output: Polynomial interpolation in Newton Basis
$d =$ copy of y data
$n =$ length of x
for $j = 1$ to $n - 1$: **do**
  for $i = n - 1$ to j: **do**
   $d[i] = (d[i] - d[i-1])/(x[i] - x[i-j])$
  end for
end for
Return d

Algorithm 5 Newton interpolation.

Input: t, y: two data vectors.
Output: Polynomial interpolation in Newton Basis.
$d =$ divided difference algorithm (t, y)
$P = 0$
for $i = n - 1$ to 0: **do**
  $P = d[i] + (x - t[i]) * P$
end for
Return P

Algorithm 6 DRMVLPIA deterministic.

input Z, R
Output Interpolation polynomial associated with (Z, R)
$f =$ Separation linear functional of Z determined by Algorithm 2
$T = f(Z)$
$P =$ Newton interpolation algorithm (T, R)
Return $P \circ f$

Algorithm 7 DRMVLPIA probabilistic.

input Z, R
Output Interpolation polynomial associated with (Z, R)
$(f, T) =$ (Separation linear functional of $Z, f(Z)$) determined by Algorithm 3
$P =$ Newton interpolation algorithm (T, R)
Return $P \circ f$

Finally, we give the Dimensional Reduction for MultiVariate Lagrange Polynomial Interpolation Algorithm (DRMVLPIA), which computes the multivariate interpolation polynomial associated with (Z, R), both in its deterministic and probabilistic form. The first one, Algorithm 6, uses Algorithm 2 to determine a linear functional f that separates the nodes of Z. The second one, Algorithm 7, employs Algorithm 3, which randomly returns the linear functional f as well as the new real nodes $t = f(Z)$. In both versions, we use the same scheme. The only difference is that in the probabilistic form, we will not need to recompute $f(Z)$.

REMARK 5.2. Let us note here that the main benefit of this algorithm is that it takes advantage of all the algorithms developed in the context of one-variable interpolation. At this

ETNA
Kent State University and
Johann Radon Institute (RICAM)

level, it would be advantageous not to expand the expression $P \circ f$, in order to take advantage of Horner's scheme during eventual evaluations and also to minimize the cost of operations.

6. Examples. In this section, we will give examples to test our algorithms.

Example 1. Here, we consider examples taken from [4]. These examples illustrate the special case where the interpolation set Z is an $\mathbb{R}^2$-grid of the form $\{1,\dots,n_1\}\times\{1,\dots,n_2\}$, where $(n_1,n_2)\in\mathbb{N}^{*2}$.

If we choose $Z=\{(1,1),(1,2),(1,3)\}$ and $R=(1,-1,-2)$, then using the deterministic Algorithm 6, we obtain the separation linear functional

$$f:(x,y)\longmapsto y,$$

and the corresponding interpolating polynomial

$$P=\frac{1}{2}y^2-\frac{7}{2}y+4.$$

We also applied, two times, the probabilistic Algorithm 7, which gave us the following two solutions

$$P_1=\left(\frac{9x}{8}-\frac{y}{4}+\frac{3}{8}\right)(9x-2y-7)+1=\frac{81x^2}{8}-\frac{9xy}{2}-\frac{9x}{2}+\frac{y^2}{2}+y-\frac{13}{8},$$

$$P_2=(-8x+y+7)\left(-4x+\frac{y}{2}+1\right)+1=32x^2-8xy-36x+\frac{y^2}{2}+\frac{9y}{2}+8.$$

In [4], the authors have used $f_1:(x,y)\longmapsto 3x+y$ and $f_2:(x,y)\longmapsto x+y$ as possible separation linear functionals. Then the corresponding interpolating polynomials stated there are, respectively,

$$P_3=\frac{1}{2}(3x+y)^2-\frac{13}{2}(3x+y)+19=\frac{9}{2}x^2+3xy-\frac{39}{2}x+\frac{1}{2}y^2-\frac{13}{2}y+19,$$
$$P_4=\frac{1}{2}(x+y)^2-\frac{9}{2}(x+y)+8=\frac{1}{2}x^2+xy-\frac{9}{2}x+\frac{1}{2}y^2-\frac{9}{2}y+8.$$

We can notice that in this particular case (the nodes have the same abscissa), the solution provided by the deterministic algorithm is better.

If we choose $Z=\{(1,1),(1,2),(2,1),(2,2)\}$ and $R=(-15,36,-1,96)$, then using Algorithm 6, we get the interpolating polynomial

$$P=\left((296x+148y-916)\left(x+\frac{1}{2}y-2\right)+102\right)\left(x+\frac{1}{2}y-\frac{3}{2}\right)-15.$$

Applying the Algorithm 7 two times, we obtain the following other solutions:

$$P_1=\left(\left(\frac{407x}{40}-\frac{407y}{24}-\frac{23}{15}\right)(3x-5y+7)-\frac{51}{5}\right)(3x-5y+2)-15,$$
$$P_2=\left(\left(-\frac{222x}{5}+\frac{111y}{10}+\frac{443}{5}\right)(-4x+y+2)+51\right)(-4x+y+3)-15.$$

In [4], the authors have used $f_1:(x,y)\longmapsto x+2y$ and $f_2:(x,y)\longmapsto 2x+y$ as possible separation functionals. Then the corresponding interpolating polynomials stated there are, respectively,

$$\begin{aligned}P_3&=\frac{23}{2}x^2+46xy-\frac{133}{2}x+46y^2-133y+81,\\ P_4&=296x^3+444x^2y-1952x^2+222xy^2-1952xy+4196x\\ &\qquad+37y^3-488y^2+2098y-2916.\end{aligned}$$

Example 2. For this example we take $d = 3$, and we choose

$$Z = \{(0,0,0), (0,-1,1), (0,3,2), (2,0,3)\} \qquad \text{and} \qquad R = \left(0, 4, \frac{1}{3}, 1\right).$$

Using the deterministic Algorithm 6 we obtain

$$P = \left(\left(-\frac{3707712}{461125}x - \frac{926928}{461125}y - \frac{154488}{461125}z + \frac{11766288}{461125} \right) \times \left(x + \frac{1}{4}y + \frac{1}{24}z + \frac{5}{24} \right) - \frac{96}{5} \right) \left(x + \frac{1}{4}y + \frac{1}{24}z \right),$$

or after expansion,

$$\begin{aligned} P = &-\frac{3707712}{461125}x^3 - \frac{2780784}{461125}x^2y - \frac{463464}{461125}x^2z + \frac{10993848}{461125}x^2 - \frac{695196}{461125}xy^2 \\ &- \frac{231732}{461125}xyz + \frac{5496924}{461125}xy - \frac{19311}{461125}xz^2 + \frac{916154}{461125}xz - \frac{1280458}{92225}x \\ &- \frac{57933}{461125}y^3 - \frac{57933}{922250}y^2z + \frac{1374231}{922250}y^2 - \frac{19311}{1844500}yz^2 + \frac{458077}{922250}yz \\ &- \frac{640229}{184450}y - \frac{6437}{11067000}z^3 + \frac{458077}{11067000}z^2 - \frac{640229}{1106700}z. \end{aligned}$$

There are two solutions given by Algorithm 7. We just provide the factorized expression that follows from the Horner scheme.

$$P_1 = \left(\left(-7x - 7y + 5z - 12 \right) \left(\frac{203}{4554}x + \frac{203}{4554}y - \frac{145}{4554}z - \frac{247}{4554} \right) + \frac{1}{3} \right) \times \left(-7x - 7y + 5z \right),$$

$$P_2 = \left(\left(-\frac{184}{33495}x + \frac{46}{11165}y + \frac{92}{6699}z - \frac{4373}{66990} \right) \left(4x - 3y - 10z + 7 \right) - \frac{4}{7} \right) \times \left(4x - 3y - 10z \right).$$

Example 3. For this example we consider $d = 3$, and for Z we take the full grid of $\mathbb{R}^3$ considered in [6, 7]

$$\begin{aligned} Z = \Big\{ &(0,2,1), (1,2,1), (0,0,1), (1,0,1), (0,2,-\tfrac{1}{2}), \\ &(1,2,-\tfrac{1}{2}), (0,0,-\tfrac{1}{2}), (1,0,-\tfrac{1}{2}), \\ &(0,2,\tfrac{7}{3}), (1,2,\tfrac{7}{3}), (0,0,\tfrac{7}{3}), (1,0,\tfrac{7}{3}) \Big\} \end{aligned}$$

and the interpolation values

$$R = \left(1, 0, -2, -1, 1, \frac{1}{2}, -1, 1, \frac{22}{7}, 0, \frac{9}{2}, -3\right).$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

We just give the factorized form that follows from the Horner scheme:

$$
\begin{aligned}
P = \Bigg[\Bigg(\Bigg\{\Bigg[\Bigg(\Bigg\{\Bigg[\Bigg(\Bigg\{&\big(-8x+8y+9z-29\big)\times\\
&\Big[-\frac{1386508920647}{16501926339986605056 00}x+\frac{1386508920647}{1650192633998660505600}y+\\
&\qquad\frac{1386508920647}{1466837896887698227200}z-\frac{10529416653769}{4400513690663094681600}\Big]\\
&+\frac{1414713997}{904710873902774400}\Bigg\}\times\\
&\big(-8x+8y+9z-37\big)-\frac{5767739129}{180942174780554880}\Bigg)\times\\
&\big(-8x+8y+z+\frac{25}{2}\big)+\frac{28103}{30380676480}\Bigg]\times\\
&\big(-8x+8y+9z+\frac{9}{2}\big)+\frac{15709}{843907680}\Bigg\}\times\\
&\big(-8x+8y+9z-\frac{7}{2}\big)+\frac{6733}{28607040}\Bigg)\times\\
&\big(-8x+8y+9z-\frac{23}{2}\big)+\frac{571}{443520}\Bigg]\times\\
&\big(-8x+8y+9z-1\big)-\frac{1}{768}\Bigg\}\Big)\times\\
&\big(-8x+8y+9z-9\big)-\frac{1}{128}\Bigg)\times\\
&\big(-8x+8y+9z-17\big)+\frac{1}{8}\Bigg]\times\\
&\big(-8x+8y+9z-25\big)+1\,.
\end{aligned}
$$

REMARK 6.1. As illustrated in the theoretical section, the interpolation polynomial associated to a set of nodes of size n is of total degree less than or equal to $n-1$. However, the algorithm DRMVPIA, deterministic or probabilistic, has the advantage that its construction is based on one-dimensional interpolation algorithms, and in order to optimize the computation it is better to keep the form that follows from the Horner scheme.

7. Conclusion. In the present work, a theoretical and practical method has been proposed to transform the multivariate polynomial interpolation problem into a univariate problem. This transformation will allow a judicious exploitation of all univariate Lagrange interpolation schemes such as Newton's scheme or divided differences, and also to take advantage of Horner's scheme in the evaluation outside the interpolation nodes, which presents advantages in terms of numerical complexity and stability. On the other hand, the fact that the deterministic method is also applicable in the context of finite fields will eventually allow applications particularly in cryptography.

REFERENCES

[1] I. S. Berezin and N. P. Shidkhov, *Computing Methods*, Addison-Wesley, Reading, 1965.
[2] J. Carnicer and T. Sauer, *Leibniz rules for multivariate divided differences*, J. Approx. Theory, 181 (2014), pp. 43–53.
[3] C. de Boor and A. Ron, *Computational aspects of polynomial interpolation in several variables*, Math. Comp., 58 (1992), pp. 705–727.
[4] D. P. Singh and A. Ujlayan, *A bivariate polynomial interpolation problem for matrices*, Preprint on arXiv, 2018. https://arxiv.org/abs/1712.08165
[5] N. Dyn and M. S. Floater, *Multivariate polynomial interpolation on lower sets*, J. Approx. Theory, 177 (2014), pp. 34–42.
[6] M. Errachid, A. Essanhaji, and A. Messaoudi, *RMVPIA: a new algorithm for computing the Lagrange multivariate polynomial interpolation*, Numer. Algorithms, 84 (2020), pp. 1507–1534.
[7] A. Essanhaji and M. Errachid, *Lagrange multivariate polynomial interpolation: a random algorithmic approach*, J. Appl. Math., (2022), Art. ID 8227086, 8 pages.
[8] M. Gasca and T. Sauer, *On the history of multivariate polynomial interpolation*, J. Comput. Appl. Math., 122 (2000), pp. 23–35.
[9] ———, *Polynomial interpolation in several variables*, Adv. Comput. Math., 12 (2000), pp. 377–410.
[10] I. B. Gundogdu, *Usage of multivariate geostatistics in interpolation processes for meteorological precipitation maps*, Theor. Appl. Climatology, 127 (2017), pp. 81–86.
[11] E. Isaacson and H. Keller, *Analysis of Numerical Methods*, Wiley, New York, 1966.
[12] G. C. Meletiou, D. K. Tasoulis, and M. N. Vrahatis, *Cryptography through interpolation, approximation and computational intelligence methods*, Bull. Greek Math. Soc., 48 (2003), pp. 61–76.
[13] R. D. Neidinger, *Multivariate polynomial interpolation in Newton forms*, SIAM Rev., 61 (2019), pp. 361–381.
[14] T. Sauer, *Polynomial interpolation of minimal degree*, Numer. Math., 78 (1997), pp. 59–85.
[15] ———, *Lagrange interpolation on subgrids of tensor product grids*, Math. Comp., 73 (2004), pp. 181–190.