On multivariate Lagrange interpolation

by

Thomas Sauer
Mathematical Institue
University Erlangen–Nuremberg
91054 Erlangen
Germany
sauer@mi.uni-erlangen.de

Yuan Xu†
Department of Mathematics
University of Oregon
Eugene, Oregon 97403
USA
yuan@bright.uoregon.edu

Abstract

Lagrange interpolation by polynomials in several variables is studied through a finite differences approach. We establish an interpolation formula analogous to that of Newton and a remainder formula, both of them in terms of the finite differences. We prove that the finite difference admits an integral representation involving simplex spline functions. In particular, this provides a remainder formula for Lagrange interpolation of degree n of a function f, which is a sum of integrals of certain $(n+1)$–st directional derivatives of f multiplied by simplex spline functions. We also provide two algorithms for the computation of Lagrange interpolants which use only addition, scalar multiplication, and point evaluation of polynomials.

Keywords

Lagrange interpolation, finite difference, simplex spline, remainder formula, algorithm.

1991 Mathematics Subject Classification: Primary: 41A05, 41A10, 65D05, 65D10

Version: June 10, 1994, (Compiled June 10, 1994)

†Supported by National Science Fundation Grant No. 9302721 and the Alexander von Humboldt Foundation

1 Introduction

Let Π^d be the space of all polynomials in d variables, and let Π_n^d be the subspace of polynomials of total degree at most n. For a sequence of pairwise distinct points in $\mathbb{R}^d$, denoted by $\mathcal{X}$, we say that the associated Lagrange interpolation problem is poised for a subspace $\Pi_{\mathcal{X}}^d \subset \Pi_n^d$, if for any f defined on $\mathbb{R}^d$ there exists a unique polynomial $P_f \in \Pi_{\mathcal{X}}^d$ which matches f on $\mathcal{X}$.

It is well–known that there are essential difficulties in solving Lagrange interpolation by polynomials in several variables. First of all, there is the problem of choosing the right polynomial subspace, for there are many linearly independent polynomials of the same total degree. Secondly, and much more troublesome, the uniqueness of interpolation depends on the geometric configuration of the interpolating points. Thus, for example, if $|\mathcal{X}| = \dim\Pi_n^d$, then the Lagrange interpolation problem is poised if, and only if, the node sequence $\mathcal{X}$ does not lie on a hypersurface of degree n; i.e., there does not exist a polynomial in Π_n^d which vanishes on all of the nodes; equivalently, the Vandermonde determinant formed by the interpolation points does not vanish. Thirdly, even if the interpolation problem is poised, the computation of the interpolating polynomial can be difficult and there is no known formula for the remainder term in the general case.

One should mention that there have been various efforts to overcome at least some of these difficulties. One approach is to put conditions on the location of nodes to guarantee both the uniqueness of the interpolation and a simple construction of the interpolating polynomials. However, such conditions are usually too restrictive and difficult to fulfill and apply. After all, the set of nodes for which Lagrange interpolation is not unique has measure zero, and interpolation is almost always possible; for literature and an historical account we refer to the recent survey [2] and the monograph [4], the latter one also containing a particularly extensive bibliography. Recently, a very interesting approach has been given by de Boor and Ron (see [1] and the references therein). They showed that for any given $\mathcal{X}$ there always exists a polynomial subspace $\Pi_{\mathcal{X}}^d$ for which the corresponding Lagrange interpolation problem is poised. In addition to an extensive investigation of the theoretic aspect of their approach they also provide an algorithm for the computation of the interpolant.

In this paper, we shall take a different approach, which turns out to be surprisingly close to the classical univariate one. The starting point for our investigation is the realization that the multivariate problem analogous to univariate interpolation is what we will refer to as *interpolation in block*, meaning that the total number of interpolation nodes is equal to $\dim\Pi_n^d$ and the interpolation points are grouped in blocks whose cardinality is equal to the dimension of the polynomial subspaces. This viewpoint allows us to develop a finite differences approach to Lagrange interpolation that offers formulae very much comparable to the classical univariate ones. Our finite differences in several variables are defined by a recurrence relation, and lead to a Newton formula for Lagrange interpolation that allows us to compute just several additional terms for each block of interpolation points added. But perhaps even more important is the representation of an n–th order finite difference in terms of a sum of integrals of n–fold directional derivatives of f multiplied by simplex splines, which is analogous to the B-spline representation of the univariate divided difference. This representation leads to an elegant remainder formula for the Lagrange interpolation, and for $d = 1$ this formula coincides with the well-known univariate one.

The usual representation of the interpolation polynomials is given through the *Lagrange fundamental polynomials* which are one in one of the points and zero in all the other ones. Our finite differences approach, however, will use a different basis of polynomials which we will call *Newton fundamental polynomials*. The name is justified by the fact that these polynomials and the associated finite differences give a multivariate analogy of the univariate Newton formula. Both polynomial bases can be given in terms of Vandermonde determinants, but for computational purposes determinants can be difficult to handle and are known to be highly unstable. As an alternative, we provide two algorithms which seem to be of independent interest; these algorithms only use the natural operations on polynomials, i.e., addition, multiplication with scalars and point evaluation. The first algorithm computes the Lagrange fundamental polynomials and stops if the Lagrange interpolation problem is not poised; the second one determines the Newton fundamental polynomials if the interpolation problem is poised, or it gives an algebraic surface of minimal degree which vanishes on all the nodes.

The paper is organized as follows. In Section 2, we give the necessary preliminaries. The finite differences approach to interpolation is contained in Section 3, and the algorithms are given in Section 4. Finally, in Section 5, we provide an example in $\mathbb{R}^2$ which is analogous to equidistant points in one variable.

2 Preliminaries

We use standard multiindex notation. For example, for $\alpha = (\alpha_1, \ldots, \alpha_d) \in \mathbb{N}_0^d$ we write $|\alpha| = \alpha_1 + \ldots + \alpha_d$, and for $x \in \mathbb{R}^d$ we write $x = (\xi_1, \ldots, \xi_d)$ and $x^\alpha = \xi_1^{\alpha_1} \cdots \xi_d^{\alpha_d}$. For each $n \in \mathbb{N}_0$ there are $r_n^d = \binom{n+d-1}{n}$ monomials x^α which have total degree n. A natural basis for Π_n^d is formed by the monomials $\{x^\alpha : 0 \le |\alpha| \le n\}$.

Let $\mathcal{X} = \{x_0, x_1, \ldots\}$ be a sequence of pairwise distinct points in $\mathbb{R}^d$, and let $\mathcal{X}_N = \{x_0, \ldots, x_N\}$. If $N = \dim \Pi_n^d$ and if there is a unique polynomial $P \in \Pi_n^d$ such that

$$P(x_k) = f(x_k), \qquad 1 \le k \le N, \tag{2.1}$$

for any $f : \mathbb{R}^d \mapsto \mathbb{R}$, then we say that the Lagrange interpolation problem (2.1) is *poised* with respect to $\mathcal{X}_N$ in Π_n^d and we denote P by $L_n(f)$. More general, given N points, not necessarily $N = \dim \Pi_n^d$ for some n, and a subspace $\Pi_{\mathcal{X}_N}^d \subset \Pi^d$, we say that the Lagrange interpolation problem is poised with respect to $\mathcal{X}_N$ in $\Pi_{\mathcal{X}_N}^d$ if for any $f : \mathbb{R}^d \to \mathbb{R}$ there is a unique polynomial $P \in \Pi_{\mathcal{X}_N}^d$ such that (2.1) is satisfied. From Kergin interpolation we know that for any choice of pairwise disjoint points $x_1, \ldots, x_N$ there always exists (at least) one subspace $\Pi_{\mathcal{X}_N}^d \subset \Pi_{N-1}^d$ for which the interpolation problem with respect to $\mathcal{X}_N$ is poised; in other words: given any sequence of nodes we can find a subspace of Π^d for which the Lagrange interpolation problem is poised. Finally, we call the (possibly infinite) sequence $\mathcal{X}$ *poised in block*, if for any $n \in \mathbb{N}_0$ the Lagrange interpolation problem is poised with respect to $\mathcal{X}_N$ in Π_n^d, whenever $N = \dim \Pi_n^d$.

We start from the observation that multivariate monomials are naturally grouped in *blocks*; i.e., instead of a single monomial of degree n in one variable, we have a whole block of monomials of degree n in several variables, namely, the monomials x^α, $|\alpha| = n$. If we arrange multiindices $|\alpha| = n$ in lexicographical order, we can number the monomials of total degree n as $q_1^{[n]}, \ldots, q_{r_n^d}^{[n]}$, and Π^d is spanned by

$$\left\{ q_1^{[0]} \mid q_1^{[1]}, \ldots, q_d^{[1]} \mid q_1^{[2]}, \ldots q_{r_2^d}^{[2]} \mid \ldots \mid q_1^{[n]}, \ldots, q_{r_n^d}^{[n]} \mid \ldots \right\}.$$

From this blockwise viewpoint, it is only natural to group the interpolation points $\mathcal{X}$ according to the same structure and rewrite them as

$$\mathcal{X} = \left\{ x_1^{(0)} \mid x_1^{(1)}, \ldots, x_d^{(1)} \mid x_1^{(2)}, \ldots, x_{r_2^d}^{(2)} \mid \ldots \mid x_1^{(n)}, \ldots, x_{r_n^d}^{(n)} \mid \ldots \right\}.$$

We refer to Lagrange interpolation with $\mathcal{X}$ arranged in this way as *interpolation in block*. As a byproduct of Algorithm 4.4 we will show that, whenever $\mathcal{X}$ is poised in block, the points $x_1, x_2, \ldots$ can be arranged in such a way.

If $\mathcal{X}$ is poised in block, the n-th *Newton fundamental polynomials*, denoted by $p_j^{[n]} \in \Pi_n^d$, $1 \le j \le r_n^d$, are uniquely defined by the conditions

$$p_j^{[n]}(x_i^{(k)}) = 0,\ k < n, \qquad \text{and} \qquad p_j^{[n]}(x_i^{(n)}) = \delta_{ij},\ i = 1, \ldots, r_n^d. \tag{2.2}$$

This means that for each level n there are r_n^d Newton fundamental polynomials of degree exactly n which vanish on all points of lower level and all points of n-th level except the one which has the same index. Actually, these polynomials can be given explicitly as follows. We introduce the vectors

$$\boldsymbol{p}^n(x) := [x^\alpha]_{|\alpha| \le n}, \qquad \boldsymbol{p}^n(x) \in \mathbb{R}^N, \quad N = \dim \Pi_n^d,$$

and the Vandermonde determinant

$$\tau(\boldsymbol{p}^n) := \det \left[\boldsymbol{p}^n(x_1^{(0)}), \boldsymbol{p}^n(x_1^{(1)}), \boldsymbol{p}^n(x_2^{(1)}), \ldots, \boldsymbol{p}^n(x_1^{(n)}), \ldots, \boldsymbol{p}^n(x_{r_d^n}^{(n)}) \right],$$

which is nonzero if and only if the interpolation problem is poised; likewise,

$$\tau_j(\boldsymbol{p}^n|x) := \det \left[\boldsymbol{p}^n(x_1^{(0)}), \ldots, \boldsymbol{p}^n(x_1^{(n)}), \ldots, \boldsymbol{p}^n(x_{j-1}^{(n)}), \boldsymbol{p}^n(x), \boldsymbol{p}^n(x_{j+1}^{(n)}), \ldots, \boldsymbol{p}^n(x_{r_n^d}^{(n)}) \right],$$

for $1 \le j \le r_n^d$. Then it can be easily verified that the polynomials $p_j^{[n]}$ are equal to

$$p_j^{[n]}(x) = \frac{\tau_j(\boldsymbol{p}^n|x)}{\tau(\boldsymbol{p}^n)}, \quad j = 1, \ldots, r_n^d. \tag{2.3}$$

With the help of the Newton fundamental polynomials we can also deal with interpolation problems based on $\mathcal{X}_N$ for $N < \dim \Pi_n^d$. The point is that we can consider the polynomial subspace $\Pi_{\mathcal{X}} = \Pi^d_{n-1} \cup W$ where W is spanned by some of the $p_j^{[n]}$ such that $\dim \Pi_{\mathcal{X}} = N$. We only consider interpolation of this type in our first algorithm, and the choice of $p_j^{[n]}$ will become clear from the algorithm.

Next we recall the definition and some properties of simplex splines, following the fundamental paper of Micchelli [5]. Given $n+1 \geq d+1$ knots $v^0, \ldots, v^n \in \mathbb{R}^d$, the *simplex spline* $M(x|v^0, \ldots, v^n)$ is defined by the condition

$$\int_{\mathbb{R}^d} f(x)\, M(x|v^0, \ldots, v^n)\, dx = (n-d)! \int_{S_n} f(\sigma_0 v^0 + \cdots + \sigma_n v^n) d\sigma, \quad f \in C(\mathbb{R}^d), \quad (2.4)$$

where

$$S_n := \{\sigma = (\sigma_0, \ldots, \sigma_n) : \sigma_i \geq 0,\, \sigma_0 + \cdots + \sigma_n = 1\}.$$

To exclude cases of degeneration which can be handled similarly, let us assume here that the convex hull of the knots, $[v^0, \ldots, v^n]$, has dimension d. Then the simplex spline $M(\cdot|v^0, \ldots, v^n)$ is a nonnegative piecewise polynomial of degree $n-d$, supported on $[v^0, \ldots, v^n]$. The order of differentiability depends on the position of the knots; if, e.g., the knots are in general position, i.e., any subset of $d+1$ knots spans a proper simplex, then the simplex spline has maximal order of differentiability, namely $n-d-1$. The most important property for our present purposes is the formula for directional derivatives, namely

$$D_y M(x|v^0, \ldots, v^n) = \sum_{j=0}^n \mu_j M(x|v^0, \ldots, v^{j-1}, v^{j+1}, \ldots, v^n), \quad y = \sum_{j=0}^n \mu_j v^j,\ \sum_{j=0}^n \mu_j = 0. \quad (2.5)$$

In particular, for $0 \leq i, j \leq n$,

$$D_{v^i - v^j} M(x|v^0, \ldots, v^n) = M(x|v^0, \ldots, v^{i-1}, v^{i+1}, \ldots, v^n) - M(x|v^0, \ldots, v^{j-1}, v^{j+1}, \ldots, v^n). \quad (2.6)$$

3 Finite differences and the Newton formula

Let $\mathcal{X} \subset \mathbb{R}^d$ be poised in block. To simplify the notation, we introduce the vectors

$$\boldsymbol{x}^n = \left[x_1^{(n)}, \ldots, x_{r_n^d}^{(n)}\right] \quad \text{and} \quad \boldsymbol{x} = \left[\boldsymbol{x}^0, \boldsymbol{x}^1, \ldots\right].$$

Our definition of finite difference is given as follows:

Definition 3.1. The finite difference in $\mathbb{R}^d$, denoted by

$$\lambda_n[\boldsymbol{x}^0, \ldots, \boldsymbol{x}^{n-1}, x]f, \quad x \in \mathbb{R}^d,$$

is defined recursively as

$$\lambda_0[x]f := f(x) \quad (3.1)$$

$$\lambda_{n+1}[\boldsymbol{x}^0, \ldots, \boldsymbol{x}^n, x]f := \lambda_n[\boldsymbol{x}^0, \ldots, \boldsymbol{x}^{n-1}, x]f - \sum_{i=1}^{r_n^d} \lambda_n[\boldsymbol{x}^0, \ldots, \boldsymbol{x}^{n-1}, x_i^{(n)}]f \cdot p_i^{[n]}(x). \quad (3.2)$$

If $d = 1$, and if we assume that our interpolation points are ordered as $x_0 < x_1 < \ldots < x_n \in \mathbb{R}$, then for each fixed n there is only one fundamental polynomial, given by

$$p_1^{[n]}(x) = \frac{(x - x_0) \cdots (x - x_{n-1})}{(x_n - x_0) \cdots (x_n - x_{n-1})}. \tag{3.3}$$

Let $f[x_0, \ldots, x_n]$ be the classical divided difference of a function of one variable, then it can be easily verified from Definition 3.1 that our finite difference in one variable equals

$$\lambda_n[x_0, \ldots, x_n]f = f[x_0, \ldots, x_n] \cdot (x_n - x_0) \cdots (x_n - x_{n-1}), \tag{3.4}$$

which suggests the name finite difference. The definition of this new difference is justified by several nice properties that we present below.

Proposition 3.2. *For every $f : \mathbb{R}^d \mapsto \mathbb{R}$, $n \in \mathbb{N}$, and $1 \le k \le r_n^d$*

$$f(x_k^{(n)}) = \sum_{j=0}^{n} \sum_{i=1}^{r_j^d} \lambda_j[\boldsymbol{x}^0, \ldots, \boldsymbol{x}^{j-1}, x_i^{(j)}]f \cdot p_i^{[j]}(x_k^{(n)}). \tag{3.5}$$

Proof : Using the fact that $p_j^{[n]}(x_k^{(n)}) = \delta_{jk}$ and applying (3.2) repeatedly we obtain

$$\begin{aligned}
&\sum_{i=1}^{r_n^d} \lambda_n[\boldsymbol{x}^0, \ldots, \boldsymbol{x}^{n-1}, x_i^{(n)}]f \cdot p_i^{[n]}(x_k^{(n)}) = \lambda_n[\boldsymbol{x}^0, \ldots, \boldsymbol{x}^{n-1}, x_k^{(n)}]f \\
&= \lambda_{n-1}[\boldsymbol{x}^0, \ldots, \boldsymbol{x}^{n-2}, x_k^{(n)}]f - \sum_{i=1}^{r_{n-1}^d} \lambda_{n-1}[\boldsymbol{x}^0, \ldots, \boldsymbol{x}^{n-2}, x_i^{(n-1)}]f \cdot p_i^{[n-1]}(x_k^{(n)}) \\
&= \lambda_{n-2}[\boldsymbol{x}^0, \ldots, \boldsymbol{x}^{n-3}, x_k^{(n)}]f - \sum_{j=n-2}^{n-1} \sum_{i=1}^{r_j^d} \lambda_j[\boldsymbol{x}^0, \ldots, \boldsymbol{x}^{j-1}, x_i^{(j)}]f \cdot p_i^{[j]}(x_k^{(n)}) \\
&= \lambda_0[x_k^{(n)}]f - \sum_{j=0}^{n-1} \sum_{i=1}^{r_j^d} \lambda_j[\boldsymbol{x}^0, \ldots, \boldsymbol{x}^{j-1}, x_i^{(j)}]f \cdot p_i^{[j]}(x_k^{(n)}) \\
&= f(x_k^{(n)}) - \sum_{j=0}^{n-1} \sum_{i=1}^{r_j^d} \lambda_j[\boldsymbol{x}^0, \ldots, \boldsymbol{x}^{j-1}, x_i^{(j)}]f \cdot p_i^{[j]}(x_k^{(n)})
\end{aligned}$$

from which (3.5) follows readily. $\square$

As an immediate consequence of this proposition we obtain in analogy to the Newton formula for Lagrange interpolation,

Theorem 3.3. *Let the interpolation problem (2.1), based on the points $\boldsymbol{x}^0, \ldots, \boldsymbol{x}^n$, be poised. Then the Lagrange interpolation polynomial $L_n(f) \in \Pi_n^d$ is given by*

$$L_n(f, x) = \sum_{j=0}^{n} \sum_{i=1}^{r_j^d} \lambda_j[\boldsymbol{x}^0, \ldots, \boldsymbol{x}^{j-1}, x_i^{(j)}]f \cdot p_i^{[j]}(x). \tag{3.6}$$

For $d = 1$, it follows from (3.3) and (3.4) that formula (3.6) becomes

$$L_n(f, x) = \sum_{i=0}^{n} \lambda_i[x_0, \ldots, x_i]f \cdot p_1^{[i]}(x) = \sum_{i=0}^{n} f[x_0, \ldots, x_i] \cdot (x - x_0) \cdots (x - x_{i-1}),$$

which is the classical Newton formula. Moreover, this notion of the finite difference also leads to the following remainder formula for interpolating polynomials:

Theorem 3.4. *For each $f : \mathbb{R}^d \mapsto \mathbb{R}$ and $n \in \mathbb{N}$*

$$f(x) - L_n(f, x) = \lambda_{n+1}[\boldsymbol{x}^0, \ldots, \boldsymbol{x}^n, x]f \tag{3.7}$$

Proof : Starting with (3.5) and using (3.2) repeatly we obtain

$$\begin{aligned}
f(x) - L_n(f, x) &= f(x) - \sum_{j=0}^{n} \sum_{i=1}^{r_j^d} \lambda_j[\boldsymbol{x}^0, \ldots, \boldsymbol{x}^{(j-1)}, x_i^{(j)}]f \cdot p_i^{[j]}(x) \\
&= \lambda_0[x]f - \lambda_0[x_1^{(0)}]f \cdot p_1^{[0]}(x) - \sum_{j=1}^{n} \sum_{i=1}^{r_j^d} \lambda_j[\boldsymbol{x}^0, \ldots, \boldsymbol{x}^{(j-1)}, x_i^{(j)}]f \cdot p_i^{[j]}(x) \\
&= \lambda_1[\boldsymbol{x}^0, x]f - \sum_{j=1}^{n} \sum_{i=1}^{r_j^d} \lambda_j[\boldsymbol{x}^0, \ldots, \boldsymbol{x}^{(j-1)}, x_i^{(j)}]f \cdot p_i^{[j]}(x) \\
&= \lambda_1[\boldsymbol{x}^0, x]f - \sum_{i=1}^{r_1^d} \lambda_1[\boldsymbol{x}^0, x_i^{(1)}]f \cdot p_i^{[1]}(x) - \sum_{j=2}^{n} \sum_{i=1}^{r_j^d} \lambda_j[\boldsymbol{x}^0, \ldots, \boldsymbol{x}^{(j-1)}, x_i^{(j)}]f \cdot p_i^{[j]}(x) \\
&= \lambda_2[\boldsymbol{x}^0, \boldsymbol{x}^1, x]f - \sum_{j=2}^{n} \sum_{i=1}^{r_j^d} \lambda_j[\boldsymbol{x}^0, \ldots, \boldsymbol{x}^{(j-1)}, x_i^{(j)}]f \cdot p_i^{[j]}(x) \\
&\vdots \\
&= \lambda_n[\boldsymbol{x}^0, \ldots, \boldsymbol{x}^{n-1}, x]f - \sum_{i=1}^{r_n^d} \lambda_n[\boldsymbol{x}^0, \ldots, \boldsymbol{x}^{n-1}, x_i^{(n)}]f \cdot p_i^{[n]}(x) \\
&= \lambda_{n+1}[\boldsymbol{x}^0, \ldots, \boldsymbol{x}^n, x]f
\end{aligned}$$

□

Since $L_n(f, x)$ interpolates f at all points of level $\leq n$, as a consequence of this theorem we obtain

Corollary 3.5. *For every $f : \mathbb{R}^d \mapsto \mathbb{R}$ and $n \in \mathbb{N}$*

$$\lambda_{n+1}[\boldsymbol{x}^0, \ldots, \boldsymbol{x}^n, x_j^{(k)}]f = 0, \quad j = 1, \ldots, r_k^d,\ k = 0, \ldots, n$$

Next we shall establish a representation of our finite difference in terms of simplex splines which requires some additional notation. Let

$$\Lambda_n := \left\{ \mu = (\mu_0, \ldots, \mu_n) \in \mathbb{N}_0^{n+1} : 1 \leq \mu_i \leq r_i^d,\ i = 0, \ldots, n \right\},$$

be an index set. Each $\mu \in \Lambda_n$ defines a *path* among the components of $\boldsymbol{x}^0, \ldots, \boldsymbol{x}^n$, which we denoted by $\boldsymbol{x}^\mu$,

$$\boldsymbol{x}^\mu := \{x_{\mu_0}^{(0)}, \ldots, x_{\mu_n}^{(n)}\}.$$

We note that $\mu_0 = 1$ by definition, thus, the path described by $\boldsymbol{x}^\mu$ starts from $x_1^{(0)}$, passes through $x_{\mu_1}^{(1)}, \ldots, x_{\mu_{n-1}}^{(n-1)}$, and ends at $x_{\mu_n}^{(n)}$. One example of the path for $d = 2$ is depicted in Fig. 1.

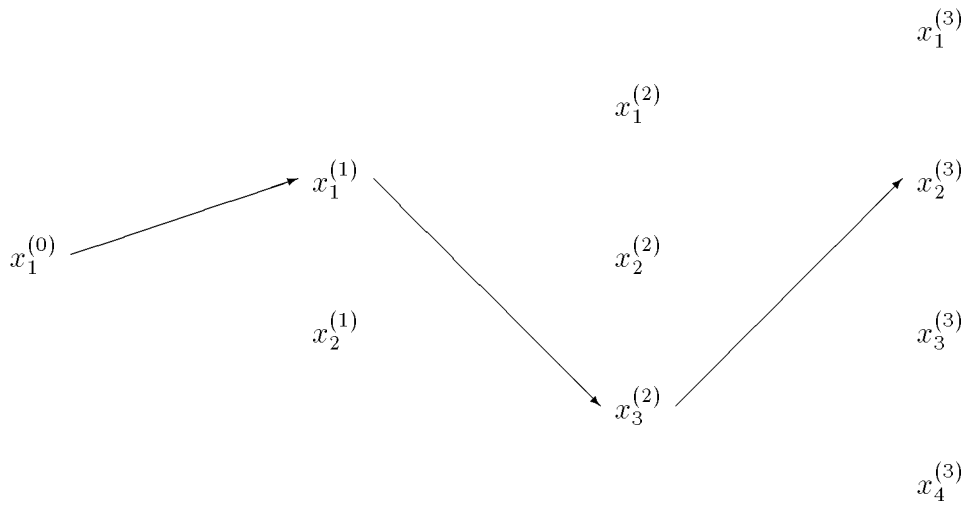

Figure 1: The path $\mu = (1, 1, 3, 2)$. Points of the same level are aligned as columns.

The collection $\{\boldsymbol{x}^\mu : \mu \in \Lambda_n\}$ contains all paths from the sole point $x_1^{(0)}$ of level 0 to all the points of level n. For any path $\boldsymbol{x}^\mu$, $\mu \in \Lambda_n$, we define the n–th directional derivative along that path as

$$D_{\boldsymbol{x}^\mu}^n := D_{x_{\mu_n}^{(n)} - x_{\mu_{n-1}}^{(n-1)}} D_{x_{\mu_{n-1}}^{(n-1)} - x_{\mu_{n-2}}^{(n-2)}} \cdots D_{x_{\mu_1}^{(1)} - x_{\mu_0}^{(0)}}, \quad \mu \in \Lambda_n.$$

In addition, we will need the values

$$\pi_\mu(\boldsymbol{x}^\mu) := \prod_{i=0}^{n-1} p_{\mu_i}^{[i]}(x_{\mu_{i+1}}^{(i+1)}), \quad \mu \in \Lambda_n.$$

We are now able to state the representation of our finite difference in terms of simplex splines and directional derivatives.

Theorem 3.6. *Let $f \in C^{n+1}(\mathbb{R}^d)$. Then*

$$\lambda_{n+1}[\boldsymbol{x}^0, \ldots, \boldsymbol{x}^n, x]f = \sum_{\mu \in \Lambda_n} p_{\mu_n}^{[n]}(x)\pi_\mu(\boldsymbol{x}^\mu) \int_{\mathbb{R}^d} D_{x - x_{\mu_n}^{(n)}} D_{\boldsymbol{x}^\mu}^n f(y) M(y|\boldsymbol{x}^\mu, x) dy. \tag{3.8}$$

In order to prove this theorem, we need to state and prove two preparatory results first:

Lemma 3.7. *Let the Lagrange interpolation problem be poised with respect to $\mathcal{X}_N$, $N = \dim \Pi_n^d$, and let i be a fixed integer such that $1 \leq i \leq r_{n-1}^d$. Then there exist d indices $k_1, \ldots, k_d$ such that the vectors*

$$p_i^{[n-1]}(x_{k_j}^{(n)}) \left(x_{k_j}^{(n)} - x_i^{(n-1)} \right), \quad j = 1, \ldots, d,$$

are nonzero and linearly independent.

Proof : Without loss of generality we can assume that $k_j = j$, $1 \leq j \leq d$ and that the points are rearranged so that

$$p_i^{[n-1]}(x_j^{(n)}) \begin{cases} \neq 0 & 1 \leq j \leq r \leq r_n^d \\ = 0 & r < j \leq r_n^d. \end{cases}$$

Assume that

$$\operatorname{rank} \left[x_j^{(n)} - x_i^{(n-1)} \right]_{j=1}^r < d;$$

i.e., the points $x_i^{(n-1)}, x_1^{(n)}, \ldots, x_r^{(n)}$ lie on an affine hyperplane of dimension $\leq d-1$. Then, there is a nonzero affine function ℓ such that

$$\ell(x_i^{(n-1)}) = \ell(x_1^{(n)}) = \ldots = \ell(x_r^{(n)}) = 0.$$

Since $p_i^{[n-1]}$ vanishes at all points of level $\leq n-1$ except at $x_i^{(n-1)}$ and at $x_j^{(n)}$, $j = r+1, \ldots, r_n^d$, it is easy to verify that the nonzero polynomial

$$p_i^{[n-1]}(x) \cdot \ell(x) \in \Pi_n^d$$

vanishes at all points of level $\leq n$. But this contradicts the assumption that the interpolation problem is poised with respect to $\mathcal{X}_N$. $\square$

As an immediate consequence of this lemma we obtain the following corollary which is of independent interests in itself.

Corollary 3.8. *Let the Lagrange interpolation problem be poised with respect to $\mathcal{X}_N$, $N = \dim \Pi_n^d$. Then every convex hull*

$$\left[x_i^{(n-1)}, x_1^{(n)}, \ldots, x_{r_n^d}^{(n)} \right], \quad i = 1, \ldots, r_{n-1}^d,$$

has dimension d.

This corollary provides a necessary condition for the uniqueness of Lagrange interpolation in block. The condition can be seen as an extension of the univariate requirement that the points have to be distinct. The main technical lemma for the proof of Theorem 3.6 is the following,

Lemma 3.9. *Let the Lagrange interpolation problem be poised with respect to $\mathcal{X}_N$, $N = \dim \Pi_n^d$. Then for $1 \le i \le r_{n-1}^d$ and $x \in \mathbb{R}^d$,*

$$p_i^{[n-1]}(x)\left(x - x_i^{(n-1)}\right) = \sum_{j=1}^{r_n^d} p_j^{[n]}(x) p_i^{[n-1]}(x_j^{(n)})\left(x_j^{(n)} - x_i^{(n-1)}\right). \tag{3.9}$$

Proof : Let i and x be fixed. We choose d indices k_j as in Lemma 3.7 and again assume that $k_j = j$. We consider the linear system of equations

$$\sum_{j=1}^{d} a_j p_i^{[n-1]}(x_j^{(n)})\left(x_j^{(n)} - x_i^{(n-1)}\right) = y, \tag{3.10}$$

where

$$y := p_i^{[n-1]}(x)\left(x - x_i^{(n-1)}\right) - \sum_{j=d+1}^{r_n^d} p_j^{[n]}(x) p_i^{[n-1]}(x_j^{(n)})\left(x_j^{(n)} - x_i^{(n-1)}\right). \tag{3.11}$$

Clearly, (3.9) holds true, if and only if $a_j = p_j^{[n]}(x)$, $j = 1, \ldots, d$, forms a solution of (3.10). From Lemma 3.7 it follows that

$$\begin{aligned} 0 \neq \Delta &= \det \left[p_i^{[n-1]}(x_j^{(n)})\left(x_j^{(n)} - x_i^{(n-1)}\right)\right]_{j=1}^{d} \\ &= \left(\prod_{j=1}^{d} p_i^{[n-1]}(x_j^{(n)})\right) \cdot \det \begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_i^{(n-1)} & x_1^{(n)} & \cdots & x_d^{(n)} \end{bmatrix}. \end{aligned}$$

To simplify the notation we define

$$\gamma_j := \prod_{l=1,\, l\neq j}^{d} p_i^{[n-1]}(x_l^{(n)}), \quad 1 \le j \le d;$$

we also introduce the notation

$$\tau(X) := \det \begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_i^{(n-1)} & x_1^{(n)} & \cdots & x_d^{(n)} \end{bmatrix}$$

and

$$\tau_j(X|x) := \det \begin{bmatrix} 1 & 1 & \cdots & 1 & 1 & 1 & \cdots & 1 \\ x_i^{(n-1)} & x_1^{(n)} & \cdots & x_{j-1}^{(n)} & x & x_{j+1}^{(n)} & \cdots & x_d^{(n)} \end{bmatrix}, \quad 1 \le j \le d.$$

In particular, we have

$$\Delta = \gamma_j \, p_i^{[n-1]}(x_j^{(n)}) \, \tau(X), \quad j = 1, \ldots, d. \tag{3.12}$$

Since $\gamma_j \neq 0$, which follows from $\Delta \neq 0$, we can apply Cramer's rule on (3.10) and, replacing y by (3.11), we obtain for $1 \leq j \leq d$

$$a_j = \frac{\gamma_j \left(p_i^{[n-1]}(x)\tau_j(X|x) - \sum_{l=d+1}^{r_n^d} p_l^{[n]}(x) p_i^{[n-1]}(x_l^{(n)}) \tau_j(X|x_l^{(n)}) \right)}{\gamma_j p_i^{[n-1]}(x_j^{(n)}) \tau(x)}.$$

That is, rearranging terms,

$$a_j p_i^{[n-1]}(x_j^{(n)})\tau(X) + \sum_{l=d+1}^{r_n^d} p_l^{[n]}(x) p_i^{[n-1]}(x_l^{(n)}) \tau_j(X|x_l^{(n)}) = p_i^{[n-1]}(x)\tau_j(X|x). \tag{3.13}$$

We now consider the two polynomials

$$\begin{aligned} q_1(x) &:= p_j^{[n]}(x) p_i^{[n-1]}(x_j^{(n)})\tau(X) + \sum_{l=d+1}^{r_n^d} p_l^{[n]}(x) p_i^{[n-1]}(x_l^{(n)}) \tau_j(X|x_l^{(n)}) \\ q_2(x) &:= p_i^{[n-1]}(x)\tau_j(X|x), \end{aligned}$$

both of them belonging to Π_n^d. Notice that q_1 is the left hand side of (3.13), and that q_2 is the right–hand side of (3.13). It can now be easily verified that these two polynomials coincide on all points of level $\leq n$, that is

$$q_1(z) = q_2(z), \quad z = x_1^{(0)}, \ldots, x_{r_n^d}^{(n)}.$$

Hence, due to the uniqueness of the interpolation polynomial, q_1 and q_2 are identical. Therefore, for any $x \in \mathbb{R}^d$, the choice $a_j = p_j^{[n]}(x)$, $j = 1, \ldots, r_n^d$, is a solution of (3.10). This concludes the proof. □

It may be of some interest to mention that the quotients $\tau_j(X|x)/\tau(X)$, $j = 1, \ldots, d$, appearing in the proof, are the barycentric coordinates of x with respect to the simplex $[x_i^{(n-1)}, x_1^{(n)}, \ldots, x_d^{(n)}]$; i.e.,

$$x = \sum_{j=1}^{d} \frac{\tau_j(X|x)}{\tau(X)} x_j^{(n)} + \left(1 - \sum_{j=1}^{d} \frac{\tau_j(X|x)}{\tau(X)}\right) x_i^{(n-1)}.$$

We are now in the position to prove Theorem 3.6.

Proof of Theorem 3.6 : We use induction on n. For $n = 0$ the equation (3.8) states

$$f(x) = \int_{\mathbb{R}^d} f(y) M(y|x) dy,$$

which is nothing but the definition of the simplex spline $M(\cdot|x)$ in the distributional sense.

For $n>0$, by the induction hypothesis, we can write

$$\lambda_n[\boldsymbol{x}^0,\ldots,\boldsymbol{x}^{n-1},x] = \sum_{\mu\in\Lambda_{n-1}} p_{\mu_{n-1}}^{[n-1]}(x)\pi_\mu(\boldsymbol{x}^\mu)\int_{\mathbb{R}^d} D_{x-x_{\mu_{n-1}}^{(n-1)}} D_{\boldsymbol{x}^\mu}^{n-1} f(y) M(y|\boldsymbol{x}^\mu, x)dy. \quad (3.14)$$

Using Lemma 3.9 and recalling the linearity of directional derivatives, we have

$$p_{\mu_{n-1}}^{[n-1]}(x) D_{x-x_{\mu_{n-1}}^{(n-1)}} = \sum_{i=1}^{r_n^d} p_i^{[n]}(x) p_{\mu_{n-1}}^{[n-1]}(x_i^{(n)}) D_{x_i^{(n)}-x_{\mu_{n-1}}^{(n-1)}}.$$

Hence, the right–hand side of (3.14) reads

$$\sum_{\mu\in\Lambda_{n-1}} \sum_{i=1}^{r_n^d} p_i^{[n]}(x) p_{\mu_{n-1}}^{[n-1]}(x_i^{(n)}) \pi_\mu(\boldsymbol{x}^\mu)\int_{\mathbb{R}^d} D_{x_i^{(n)}-x_{\mu_{n-1}}^{(n-1)}} D_{\boldsymbol{x}^\mu}^{n-1} f(y) M(y|\boldsymbol{x}^\mu, x)dy. \quad (3.15)$$

By the recurrence relation (3.2) and the induction hypothesis, equation (3.15) yields

$$\begin{aligned}
\lambda_{n+1}[\boldsymbol{x}^0,\ldots,\boldsymbol{x}^n,x]f &= \lambda_n[\boldsymbol{x}^0,\ldots,\boldsymbol{x}^{n-1},x]f - \sum_{i=1}^{r_n^d} p_i^{[n]}(x)\lambda_n[\boldsymbol{x}^0,\ldots,\boldsymbol{x}^{n-1},x_i^{(n)}]f \\
&= \sum_{\mu\in\Lambda_{n-1}} \sum_{i=1}^{r_n^d} p_i^{[n]}(x) p_{\mu_{n-1}}^{[n-1]}(x_i^{(n)}) \pi_\mu(\boldsymbol{x}^\mu)\int_{\mathbb{R}^d} D_{x_i^{(n)}-x_{\mu_{n-1}}^{(n-1)}} D_{\boldsymbol{x}^\mu}^{n-1} f(y) M(y|\boldsymbol{x}^\mu, x)dy \\
&\quad - \sum_{\mu\in\Lambda_{n-1}} \sum_{i=1}^{r_n^d} p_i^{[n]}(x) p_{\mu_{n-1}}^{[n-1]}(x_i^{(n)}) \pi_\mu(\boldsymbol{x}^\mu)\int_{\mathbb{R}^d} D_{x_i^{(n)}-x_{\mu_{n-1}}^{(n-1)}} D_{\boldsymbol{x}^\mu}^{n-1} f(y) M(y|\boldsymbol{x}^\mu, x_i^{(n)})dy \\
&= \sum_{\mu\in\Lambda_{n-1}} \sum_{i=1}^{r_n^d} p_i^{[n]}(x) p_{\mu_{n-1}}^{[n-1]}(x_i^{(n)}) \pi_\mu(\boldsymbol{x}^\mu)\int_{\mathbb{R}^d} D_{x_i^{(n)}-x_{\mu_{n-1}}^{(n-1)}} D_{\boldsymbol{x}^\mu}^{n-1} f(y) \times \\
&\qquad \times \left(M(y|\boldsymbol{x}^\mu, x) - M(y|\boldsymbol{x}^\mu, x_i^{(n)})\right) dy.
\end{aligned}$$

Applying the differentiation rule for simplex splines (2.6), we have

$$\begin{aligned}
&\lambda_{n+1}[\boldsymbol{x}^0,\ldots,\boldsymbol{x}^n,x]f \\
&\quad = \sum_{\mu\in\Lambda_{n-1}} \sum_{i=1}^{r_n^d} p_i^{[n]}(x) p_{\mu_{n-1}}^{[n-1]}(x_i^{(n)}) \pi_\mu(\boldsymbol{x}^\mu)\int_{\mathbb{R}^d} D_{x_i^{(n)}-x_{\mu_{n-1}}^{(n-1)}} D_{\boldsymbol{x}^\mu}^{n-1} f(y) D_{x_i^{(n)}-x} M(y|\boldsymbol{x}^\mu, x)dy \\
&\quad = \sum_{\mu\in\Lambda_n} p_{\mu_n}^{[n]}(x)\pi_\mu(\boldsymbol{x}^\mu)\int_{\mathbb{R}^d} D_{x-x_{\mu_n}^{(n)}} D_{\boldsymbol{x}^\mu}^{n} f(y) M(y|\boldsymbol{x}^\mu, x)dy;
\end{aligned}$$

the final step follows from replacing i by μ_n and from integration by parts, which reads

$$\begin{aligned}
\int_{\mathbb{R}^d} f(y) D_v g(y) dy &= \sum_{i=1}^{d} v_i \int_{\mathbb{R}^d} f(y)\frac{\partial}{\partial\xi_i} g(y)dy = -\sum_{i=1}^{d} v_i \int_{\mathbb{R}^d} \frac{\partial}{\partial\xi_i} f(y) g(y)dy = \\
&= -\int_{\mathbb{R}^d} D_v f(y) g(y) dy,
\end{aligned}$$

where g is a function with compact support. □

Defining $\Lambda_n(k) = \{\mu \in \Lambda_n : \mu_n = k\}$, $k = 1, \ldots, r_n^d$, we obtain a remarkable compact version of Theorem 3.6:

Corollary 3.10. *For $n \in \mathbb{N}$ and $f \in C^{n+1}(\mathbb{R}^d)$*

$$\lambda_n[\boldsymbol{x}^0, \ldots, \boldsymbol{x}^{n-1}, x_k^{(n)}]f = \sum_{\mu \in \Lambda_n(k)} \pi_\mu(\boldsymbol{x}^\mu) \int_{\mathbb{R}^d} D^n_{\boldsymbol{x}^\mu} f(y) M(y|\boldsymbol{x}^\mu) dy, \quad k = 1, \ldots, r_n^d. \tag{3.16}$$

For $\mu \in \Lambda_n(k)$, the sequence $\boldsymbol{x}^\mu$ is a path from $x_1^{(0)}$ to $x_k^{(n)}$, and the set $\{\boldsymbol{x}^\mu : \mu \in \Lambda_n(k)\}$ contains all paths from $x_1^{(0)}$ to $x_k^{(n)}$. Combining Theorem 3.4 and Theorem 3.6 gives a remainder formula for Lagrange interpolation:

Corollary 3.11. *For $n \in \mathbb{N}$ and $f \in C^{n+1}(\mathbb{R}^d)$*

$$L_n(f, x) - f(x) = \sum_{\mu \in \Lambda_n} p_{\mu_n}^{[n]}(x) \pi_\mu(\boldsymbol{x}^\mu) \int_{\mathbb{R}^d} D_{x - x_{\mu_n}^{(n)}} D^n_{\boldsymbol{x}^\mu} f(y) M(y|\boldsymbol{x}^\mu, x) dy. \tag{3.17}$$

Moreover, we have the error estimate

$$|L_n(f, x) - f(x)| \leq \sum_{\mu \in \Lambda_n} \frac{1}{(n+1)!} \|D_{x - x_{\mu_n}^{(n)}} D^n_{\boldsymbol{x}^\mu} f\|_\infty \cdot |\pi_\mu(\boldsymbol{x}^\mu) p_{\mu_n}^{[n]}(x)|, \quad x \in \mathbb{R}^d, \tag{3.18}$$

where it suffices to take the supremum norm over the convex hull of $\{\boldsymbol{x}^0, \ldots, \boldsymbol{x}^n, x\}$.

To end this section, let us take $d = 1$ in (3.18). We first compute $\pi_\mu(\boldsymbol{x}^\mu)$, using the same notation as in the univariate case considered before (for $d = 1$ there is only a single path),

$$\begin{aligned}
\pi_\mu(\boldsymbol{x}^\mu) &= \prod_{i=0}^{n-1} p_i(x_{i+1}) \\
&= \prod_{i=0}^{n-1} \frac{(x_{i+1} - x_0) \cdots (x_{i+1} - x_{i-1})}{(x_i - x_0) \cdots (x_i - x_{i-1})} \\
&= 1 \cdot \frac{x_2 - x_0}{x_1 - x_0} \cdot \frac{(x_3 - x_0)(x_3 - x_1)}{(x_2 - x_0)(x_2 - x_1)} \cdots \frac{(x_n - x_0) \cdots (x_n - x_{n-2})}{(x_{n-1} - x_0) \cdots (x_{n-1} - x_{n-2})} \\
&= \prod_{i=0}^{n-2} \frac{x_n - x_i}{x_{i+1} - x_i}.
\end{aligned}$$

On the other hand, we clearly have

$$D_{x - x_n} D_{x_n - x_{n-1}} \cdots D_{x_1 - x_0} f(y) = (x - x_n) \prod_{i=0}^{n-1} (x_{i+1} - x_i) f^{(n+1)}(y).$$

Inserting this into (3.18) we thus obtain

$$\begin{aligned}|L_n(f,x)-f(x)| &\le \frac{1}{(n+1)!}|x-x_n|\cdot\left|\prod_{i=0}^{n-1}(x_{i+1}-x_i)\right|\cdot\|f^{(n+1)}\|\cdot\prod_{i=0}^{n-2}\left|\frac{x_n-x_i}{x_{i+1}-x_i}\right|\times\\ &\qquad\times\left|\frac{(x-x_0)\cdots(x-x_{n-1})}{(x_n-x_0)(x_n-x_{n-1})}\right|\\ &= \frac{\|f^{(n+1)}\|}{(n+1)!}\left|(x-x_0)\dots(x-x_n)\right|,\end{aligned}$$

which is the well–known estimate of one variable. This shows that (3.18) is a proper extension of the univariate formula, and it offers another justification for our definition of finite difference.

4 Algorithms

In this section let $\mathcal{X}=\{x_1,\dots,x_N\}$, $N\in\mathbb{N}$, be a finite sequence of pairwise distinct points and assume that $N=\dim\Pi_n^d$ for some $n\ge 0$.

In our algorithms we will use the following notation: let $\alpha_1^n,\dots,\alpha_{r_n^d}^n$ be the multiindices $|\alpha|=n$, arranged in lexicographical order. Moreover, by $\alpha_1,\dots,\alpha_N$, we denote the ordering $\alpha_1^1,\alpha_1^2,\dots,\alpha_d^2,\dots,\alpha_1^n,\dots,\alpha_{r_n^d}^n$. The pseudocode that we will use to formulate the algorithms uses **while do** ... **done;** and **for do** ... **done;** loops (the latter one may be ascending or descending; this will become clear from the argument) as well as the **if then** ...**fi;** construction.

Our first goal is to give an algorithm for the computation of the Lagrange fundamental polynomials $P_1,\dots,P_N\in\Pi_n^d$; i.e., the polynomials which satisfy the conditions $P_i(x_j)=\delta_{ij}$, $i,j=1,\dots,N$.

The basic idea is to successively construct polynomials $P_1^{[k]},\dots,P_k^{[k]}$ for $k=1,\dots,N$, which satisfy

$$P_i^{[k]}(x_j)=\delta_{ij},\qquad i,j=1,\dots,k\le N. \tag{4.1}$$

This is trivial for $k=1$ since we can take $P_1^{[1]}(x)\equiv 1$. Thus, suppose we already constructed $P_1^{[k]},\dots,P_k^{[k]}$ for some $k\ge 1$. It is obvious from (4.1) that these polynomials are linearly independent and span a k–dimensional subspace of Π_n^d. Thus, we can find polynomials $Q_{k+1},\dots,Q_N$ such that $P_1^{[k]},\dots,P_k^{[k]},Q_{k+1},\dots,Q_N$ form a basis of Π_n^d; moreover, we can assume that

$$Q_{k+1}(x)=\dots=Q_N(x)=0,\qquad x\in\{x_1,\dots,x_k\}, \tag{4.2}$$

since otherwise we can replace $Q_j(x)$ by

$$Q_j'(x)=Q_j(x)-\sum_{i=1}^{k}Q_j(x_i)P_i(x). \tag{4.3}$$

Next, we claim that either there exists some $j \in \{k+1, \ldots, N\}$ such that $Q_j(x_{k+1}) \neq 0$, or the Lagrange interpolation problem is not poised with respect to $\mathcal{X}$. To prove this, assume that the Lagrange interpolation problem is poised with respect to $\mathcal{X}$ and let Q be a solution of the interpolation problem

$$Q(x_j) = \delta_{j,k+1}, \qquad j = 1, \ldots, N; \tag{4.4}$$

we write Q with respect to the basis defined above as

$$Q(x) = \sum_{j=1}^{k} a_j P_j^{[k]}(x) + \sum_{j=k+1}^{N} a_j Q_j(x).$$

Inserting $x = x_1, \ldots, x_k$ readily gives $a_j = 0$, $j = 1, \ldots, k$, which yields, setting $x = x_{k+1}$,

$$1 = \sum_{j=k+1}^{N} a_j Q_j(x_{k+1}).$$

Hence, not all Q_j, $j = k+1, \ldots, N$, can vanish in x_{k+1}, which proves the claim. Therefore, supposing poisedness, we can, without loss of generality, assume that $Q_{k+1}(x_{k+1}) \neq 0$. Setting

$$P_{k+1}^{[k+1]}(x) = \frac{Q_{k+1}(x)}{Q_{k+1}(x_{k+1})}$$

and

$$P_j^{[k+1]}(x) = P_j^{[k]}(x) - P_j^{[k]}(x_{k+1}) P_{k+1}^{[k+1]}(x), \qquad j = 1, \ldots, k,$$

we obtain the $k+1$ Lagrange fundamental polynomials for $x_1, \ldots, x_{k+1}$, which completes the inductional step.

In the end, for $k = N$ we either generate the Lagrange fundamental polynomials with respect to $\mathcal{X}$ in Π_n^d or we find that the Lagrange interpolation problem is not poised with respect to $\mathcal{X}$ in Π_n^d.

Instead of completing $P_1^{[k]}, \ldots, P_k^{[k]}$ to a basis of Π_n^d in each individual step, the following algorithm simultaneously computes $P_1^{[k]}, \ldots, P_k^{[k]}$, which are the Lagrange fundamental polynomials for the subproblem at $x_1, \ldots, x_k$, and the polynomials $Q_{k+1}, \ldots, Q_N$, which vanish at $x_1, \ldots, x_k$, and complete $P_1^{[k]}, \ldots, P_k^{[k]}$ to a basis of Π_n^d for $k = 1, \ldots, N$. Since for $k = 0$ there is no restriction imposed on $Q_1, \ldots, Q_N$, we can simply initialize these polynomials with the monomials. This reads as follows:

Algorithm 4.1.

```
Input: $N \in \mathbb{N}$ and $x_1, \ldots, x_N \in \mathbb{R}^d$.
Initialization:
    for $k = 1, 2, \ldots, N$ do
        $Q_k := x^{\alpha_k}$;
    done;
Computation:
    **for** $k = 1, 2, \ldots, N$ **do**
        $i := \min(\{k \le j \le N : Q_j(x_k) \ne 0\} \cup \{N+1\})$;
        **if** $i = N+1$ **then stop**: No unique interpolation; **fi**;
        $P_k(x) := \dfrac{Q_i(x)}{Q_i(x_k)}$;
        **for** $j = 1, 2, \ldots, k-1$          **do** $P_j(x) := P_j(x) - P_j(x_k) P_k(x)$; **done**;
        **for** $j = i, i-1, \ldots, k+1$        **do** $Q_j(x) := Q_{j-1}(x) - Q_{j-1}(x_k) P_k(x)$; **done**;
        **for** $j = i+1, i+2, \ldots, N$        **do** $Q_j(x) := Q_j(x) - Q_j(x_k) P_k(x)$; **done**;
    **done**;
**Output:** $P_1, \ldots, P_N \in \Pi_n^d$.
```

The algorithm may also be seen from a different point of view: the polynomials $\mathcal{P} = \{P_1, \ldots, P_k\}$, constructed in the k–th step of the algorithm, are an orthonormal basis with respect to the scalar product

$$\langle p, q \rangle_k = \sum_{j=1}^{k} p(x_j)\, q(x_j),$$

while $Q_{k+1}, \ldots, Q_N$ form a basis of the orthogonal complement of the span of $\mathcal{P}$ in Π_n^d. From this viewpoint Algorithm 4.1 is a variation of the Gram–Schmidt orthogonalization process.

If Algorithm 4.1 ends with **stop**, then the interpolation problem is not poised; otherwise, we say that the algorithm *terminates properly.* From the deduction of Algorithm 4.1 we finally formalize what the algorithm does as follows:

Proposition 4.2. *The Lagrange interpolation problem with respect to $\mathcal{X}$ is poised in Π_n^d if, and only if, Algorithm 4.1 terminates properly.*

Let us remark that Algorithm 4.1 also works if $N \le M := \dim \Pi_n^d$. Clearly, the Lagrange interpolation problem with respect to $\mathcal{X}$ cannot be poised in Π_n^d if $N < M$, but we can run the algorithm to compute Lagrange fundamental polynomials $P_1, \ldots, P_N$ as before; these polynomials then span a subspace $\Pi_{\mathcal{X}}^d \subset \Pi_n^d$ for which any interpolation problem at $\mathcal{X}$ is uniquely solvable. On the other hand, it is also easily verified that, whenever the algorithm does not terminate properly, the Lagrange interpolation problem at $\mathcal{X}$ is unsolvable in general.

Next we make some remarks on the algorithm and introduce some possible improvements:

Remark 4.3.

(1) This algorithm does not require solving linear system equations or compute Vandermonde determinants. It uses only the natural operations on polynomials; i.e., addition, multiplication by scalars and point evaluation.

(2) Due to its simplicity, the algorithm is easy to implement and, moreover, very fast. For example, interpolation with polynomials of degree 13 at 100 random points in $[0,1]^2$ takes less than 3 seconds in a C++ implementation on a SUN SparcStation 10.

(3) From Kergin interpolation we know that there always exists a subspace $\Pi^d_{\mathcal{X}} \subset \Pi^d_{N-1}$ such that the Lagrange interpolation problem with respect to $\mathcal{X}$ is poised in $\Pi^d_{\mathcal{X}}$. This suggests the following improvement of Algorithm 4.1: whenever it turns out that the Lagrange interpolation problem with respect to $\mathcal{X}$ is not poised in Π^d_n, then try the same process in Π^d_{n+1}; if in this space the Lagrange interpolation problem with respect to $\mathcal{X}$ is still not poised, then proceed to Π^d_{n+2} and so on; the above remark guarantees success after a finite number of steps. To be precise, we modify Algorithm 4.1 as follows, replacing the **stop** statement by the **while** loop below:

```
Initialization:
    $M := N$;
...
Computation:
    for $k = 1, 2, \ldots, N$ do
        $i := \min(\{k \le j \le M : Q_j(x_k) \ne 0\} \cup \{M+1\})$;
        while $i = M + 1$ do
            for $j = 1, \ldots, r^d_{n+1}$ do
                $Q_{N+j}(x) := x^{\alpha_j^{n+1}} - \sum_{l=1}^{k} x_l^{\alpha_i^{n+1}} P_l(x)$;
            done;
            $M := \dim \Pi^d_{n+1}$;
            $n := n + 1$;
            $i := \min(\{k \le j \le M : Q_i(x_j) \ne 0\} \cup \{M+1\})$;
        done;
...
```

The resulting polynomials $P_1, \ldots, P_N$ then span a subspace of Π^d_{N-1} for which the Lagrange interpolation problem is always poised with respect to $\mathcal{X}$.

(4) The algorithm works with any polynomial basis $\{Q_j : j = 1, \ldots, N\}$ of Π^d_n. In particular, this allows the use of the Bernstein–Bézier polynomial basis which is known to be very stable. If the interpolation points lie in a triangular domain, numerical experiment shows that the change to the Bernstein–Bézier basis increases stability significantly.

(5) Another way to obtain better stability is to use pivoting strategies when searching for the index i such that $Q_i(x_k) \ne 0$. In particular, two strategies are possible:

- (Polynomial pivoting): Determine i such that Q_i has maximal absolute value at x_k; this reads

```
$m := \max\{|Q_j(x_k)| : k \le j \le N\}$;
$i := \min\{k \le j \le N : |Q_j(x_k)| = m\}$;
**if** $Q_i(x_k) = 0$ **then** $i := N+1$; **fi**;
```

- (Total pivoting): Take the maximum on all polynomials $Q_k, \ldots, Q_N$ and all points $x_k, \ldots, x_N$ and switch x_k and the point where the maximum has been attained.

```
$m := \max\{|Q_j(x_l)| : k \le j, l \le N\}$;
$(i,j) := \min\{(k,k) \le (r,s) \le (N,N) : |Q_r(x_s)| = m\}$;
$x := x_k$; $x_k := x_j$; $x_j := x_k$;
**if** $Q_i(x_k) = 0$ **then** $i := N+1$; **fi**;
```

Numerical experiment shows that even polynomial pivoting can increase the stability of the algorithm significantly. In fact, the quality of interpolation; i.e., the error at the interpolation points, is usually improved by two decimals.

Our second algorithm, which computes the Newton fundamental polynomials, works for interpolation in block, but it can also be used to check whether the points $x_1, \ldots, x_N$ lie on an algebraic hypersurface of degree n; i.e., if there is some polynomial $Q \in \Pi_n^d$ such that $Q(x_1) = \cdots = Q(x_N) = 0$. It is obvious that, in this case, interpolation is not unique in Π_n^d.

A simple but quite illustrative example for this phenomenon is $d = 2$, $n = 3$ and six points $x_1, \ldots, x_6$ lying on some circle in the plane. Since a circle is the zero set of some quadratic polynomial, the Lagrange interpolation problem has no unique solution in terms of quadratic polynomials. Nevertheless, there is a subspace of Π_3^2 (i.e., of cubic polynomials) for which the Lagrange interpolation problem with respect to $x_1, \ldots, x_6$ is poised. This subspace can be given explicitly as follows: let ℓ_i be a nonzero linear function that vanishes on the edge $[x_i, x_{i+1}]$ (setting $x_7 = x_1$, $x_8 = x_2, \ldots$); then the cubic polynomial

$$Q_i(x) = \ell_{i+1}(x) \cdot \ell_{i+2}(x) \cdot \ell_{i+4}(x)$$

vanishes in all points except x_i and can thus be renormalized to obtain the Lagrange fundamental polynomial P_i.

To avoid excessive notation in stating the algorithm, let us first recall the order of multiindices $|\alpha| \le n$ as $\alpha_1, \ldots, \alpha_N$, which we introduced at the beginning of this section. We now order pairs (j,k), $1 \le j \le r_k^d$, $0 \le k \le n$, in such a way that $(j,k) \le (i,l)$ if, and only if, α_j^k appears before α_i^l in the above order of multiindices.

For the computation of the Newton fundamental polynomials we will modify Algorithm 4.1 in the following way: instead of proceeding "point by point" from x_1 to x_N and find a polynomial that does not vanish at the point x_k at k–th step, we now do it the other way around, proceeding by polynomials and searching for points where the polynomials do not vanish.

More precisely: by induction on $k = 0, \ldots, n$ and $j = 1, \ldots, r_k^d$, we construct polynomials $P_i^{[l]} \in \Pi_l^d$ and points $x_i^{(l)} \in \mathcal{X}$, $(i,l) \le (j,k)$, and complementary polynomials $Q_i^{[l]} \in \Pi_l^d$, $(j,k) < (i,l)$, such that

$$P_i^{[l]}(x_r^{(s)}) = \delta_{ls}\delta_{ir}, \qquad (r,s) \le (i,l) \le (j,k), \tag{4.5}$$

$$Q_i^{[l]}(x_r^{(s)}) = 0, \qquad (r,s) \le (j,k) < (i,l), \tag{4.6}$$

and

$$\Pi_n^d = \operatorname{span}\left(\left\{P_i^{[l]} : (i,l) \le (j,k)\right\} \cup \left\{Q_i^{[l]} : (j,k) < (i,l)\right\}\right). \tag{4.7}$$

Initializing $Q_i^{[l]} = x^{\alpha_i^l}$, we start induction at $j = k = 0$, for which (4.5), (4.6) and (4.7) are satisfied trivially. For the case $j = 1, k = 0$ any $x_1^{(0)} \in \mathcal{X}$ and $P_1^{[0]} \equiv 1$, $Q_i^{[l]} = x^{\alpha_i^l} - (x_1^{(0)})^{\alpha_i^l}$ fulfill the above requirements.

Now, suppose that for some (j,k) we have constructed polynomials of proper degree and points satisfying (4.5), (4.6) and (4.7). Moreover, let $\mathcal{Y} = \mathcal{X} \setminus \{x_i^{(l)} : (i,l) \le (j,k)\}$ denote the set of points that have not yet been put into blocks. Assume first that $j < r_k^d$. If there is no point $x_{j+1}^{(k)} \in \mathcal{Y}$ such that $Q_{j+1}^{[k]}(x_{j+1}^{(k)}) \neq 0$, then, taking (4.6) into account, $Q_{j+1}^{[k]}$ vanishes on all of $\mathcal{X}$ and, therefore, the interpolation problem cannot be poised in Π_n^d. If, on the other hand, the interpolation problem is poised, then we set

$$P_{j+1}^{[k]}(x) = \frac{Q_{j+1}^{[k]}(x)}{Q_{j+1}^{[k]}(x_{j+1}^{(k)})},$$

which satisfies $P_{j+1}^{[k]}(x_i^{(l)}) = \delta_{kl}\delta_{ij}$, whenever $(i,l) \le (j+1,k)$. Replacing $P_i^{[l]}(x)$ by

$$P_i^{[k]}(x) - P_i^{[k]}(x_{j+1}^{(k)})P_{j+1}^{[k]}(x), \qquad i = 1, \ldots, j, \tag{4.8}$$

and $Q_i^{[l]}(x)$ by

$$Q_i^{[l]}(x) - Q_i^{[l]}(x_{j+1}^{(l)})P_{j+1}^{[k]}(x), \qquad (j+1,k) < (i,l), \tag{4.9}$$

we obtain polynomials and points which can easily be seen to satisfy (4.5), (4.6), and (4.7) with (j,k) being replaced by $(j+1,k)$. This finishes the inductional step $j \to j+1$. It is important to notice that, in view of (4.8) and (4.9), we subtract a multiple of $P_{j+1}^{[k]}$ only from polynomials of level k and higher; i.e., from polynomials of degree at least k. This guarantees that at any step of the algorithm all the polynomials $P_i^{[l]}$ and $Q_i^{[l]}$ are polynomials of total degree exactly l. If $j = r_k^d$ we move on to $Q_1^{[k+1]}$, to do the inductional step $k \to k+1$.

After the final step (r_n^d, n), we have constructed polynomials $P_j^{[k]} \in \Pi_k^d$ and points $x_j^{(k)}$, $j = 1, \ldots, r_k^d$, $k = 0, \ldots, n$, such that

$$P_j^{[k]}(x_i^{(l)}) = \delta_{kl}\delta_{ij}, \qquad 1 \le j \le r_k^d,\ 1 \le i \le r_l^d,\ 0 \le l \le k \le n.$$

These polynomials are the Newton fundamental polynomials.

The algorithm can be formulated as follows:

Algorithm 4.4.

```
Input: $N \in \mathbb{N}$ and $x_1, \dots, x_N \in \mathbb{R}^d$.
Initialization:
    $\mathcal{Y} := \{x_1, \dots, x_N\}$;
    for $k = 0, 1, \dots, n$ do
        for $j = 1, 2, \dots, r_k^d$ do
            $Q_k := x^{\alpha_j^k}$;
        done;
    done;
Computation:
    for $k = 0, 1, \dots, n$ do
        for $j = 1, 2, \dots, r_k^d$ do
            $i := \min\Big(\{1 \le l \le N : x_l \in \mathcal{Y} \text{ and } Q_j^{[k]}(x_l) \neq 0\} \cup \{N+1\}\Big)$;
            if $i = N + 1$ then stop: No unique interpolation; fi;
            $x_j^{(k)} := x_i$;
            $\mathcal{Y} := \mathcal{Y} \setminus \{x_j^{(k)}\}$;
            $P_j^{[k]}(x) := \dfrac{Q_j^{[k]}(x)}{Q_j^{[k]}(x_j^{(k)})}$;
            for $i = 1, 2, \dots, j-1$          do $P_i^{[k]}(x) := P_i^{[k]}(x) - P_i^{[k]}(x_j^{(k)}) P_j^{[k]}(x)$; done;
            for $i = j+1, j+2, \dots, r_k^d$   do $Q_i^{[k]}(x) := Q_i^{[k]}(x) - Q_i^{[k]}(x_j^{(k)}) P_j^{[k]}(x)$; done;
            for $l = k+1, k+2, \dots, n$ do
                for $i = 1, 2, \dots, r_l^d$    do $Q_i^{[l]}(x) := Q_i^{[l]}(x) - Q_i^{[l]}(x_j^{(k)}) P_j^{[k]}(x)$; done;
            done;
        done;
    done;
Output: $P_1^{[k]}, \dots, P_{r_k^d}^{[k]} \in \Pi_k^d$, $k = 0, \dots, n$, and $x_1^{(0)}, \dots, x_{r_n^d}^{(n)} \in \mathbb{R}^d$.
```

We again say that the algorithm terminates properly if it does not reach the **stop**. We summarize what Algorithm 4.4 does in the following

Proposition 4.5. *The points $x_1, \dots, x_N$ do not lie on an algebraic hypersurface of degree n if, and only if, Algorithm 4.4 terminates properly. Moreover, if the algorithm terminates properly, it puts the points into blocks; in other words, $x_1, \dots, x_N$ are arranged as $x_j^{(k)}$, $j = 1, \dots, r_k^d$, $k = 0, \dots, n$.*

Remark 4.6.

(1) The polynomials $P_j^{[k]}$ produced by the algorithm are exactly the $p_j^{[k]}$ used in the previous section. If one wants to do numerical work with the finite differences, we recommend to compute the $p_j^{[k]}$ using the above algorithm instead of evaluating the determinants in (2.3).

(2) If the algorithm stops before producing the interpolating polynomial, then it not only tells that the points lie on an algebraic hypersurface of certain degree $k \le n$,

it also computes a minimal degree polynomial describing the hypersurface, namely $Q_j^{[k]}$.

(3) The algorithm can also be run with the Bernstein–Bézier basis and it can also be equipped with pivoting strategies to increase numerical stability.

(4) The polynomials

$$P_i^{[k]}(x) - \sum_{l=k+1}^{n} \sum_{j=1}^{r_l^d} P_i^{[k]}(x_j^{(l)}) P_j^{[l]}(x), \qquad k = 0, \dots, n,\ i = 1, \dots, r_k^d,$$

can easily be seen to be the Lagrange fundamental polynomials of degree n.

5 Example

In this final section we apply the theory developed in the previous sections to a simple case, namely, the extension of univariate interpolation at the knots $0, \dots, n$ to d variables. A surprising connection emerged from this study: our finite differences, restricted to the equidistant points, coincide with the well–known classical forward differences. Here we just present an example for $d = 2$ to illustrate the formulae of Section 3, omitting all the proofs.

For univariate interpolation, the equidistant points lead to the simplest divided differences and Newton interpolation formula. In Euclidean d–space, our finite differences and interpolation formula take a prticularly simple form if the interpolation nodes are the lattice points with all nonnegative components; more specifically, for each fixed n, we take as interpolation nodes the lattice points in the standard simplex

$$\mathbb{S}_n = \{x : 0 \le x_i \le n, x_1 + \cdots + x_d \le n\} \subset \mathbb{R}^d.$$

For $d = 2$ the interpolation points are given by

$$x_j^{(m)} = (m - j, j), \qquad 0 \le j \le m, \quad 0 \le m \le n.$$

They are shown in Fig. 2 for $n = 5$.

We remark here that this particular Lagrange interpolation problem has been studied before, for example in [3]
For $x \in \mathbb{R}^2$, we denote its components by u and v; i.e., $x = (u, v)$. It is not hard to verify that the fundamental polynomials are given as follows

$$p_k^{[m]}(x) = \frac{1}{(m-k)!k!} \prod_{i=1}^{m-k} (u - i + 1) \prod_{i=1}^{k} (v - i + 1), \quad 0 \le k \le m, \tag{5.1}$$

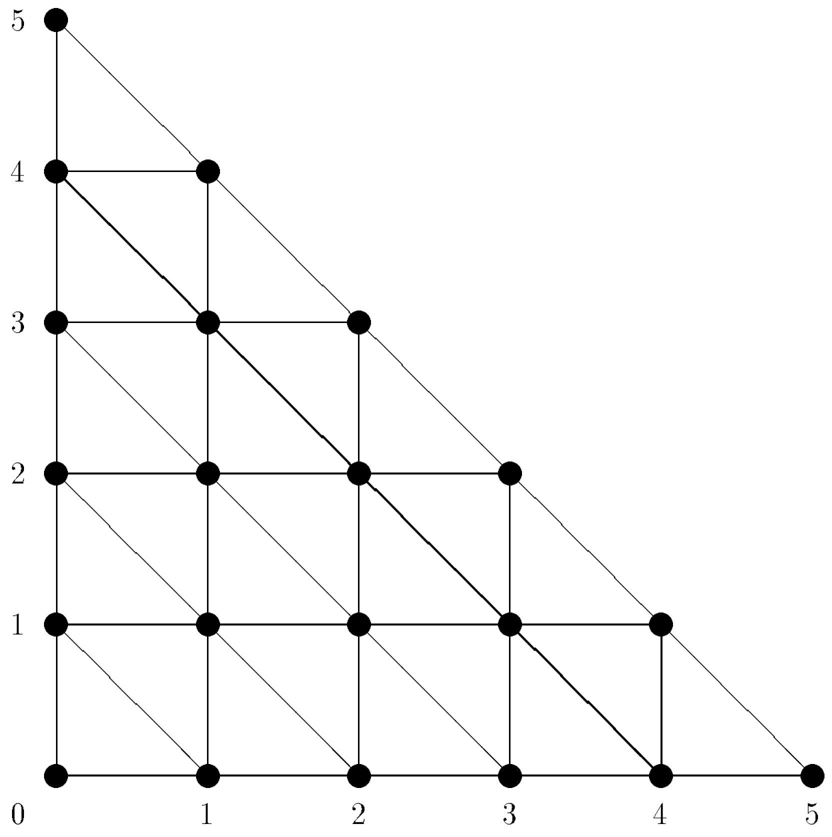

Figure 2: The points for cardinal interpolation of degree 5. Points of the same level lie on the diagonals; e.g., the bolder line connects all points of level 4.

where we define the empty product to be equal to one. From these formulae, one readily verifies that

$$p_k^{[m]}(x_k^{(m+1)}) = m+1-k, \quad p_{k-1}^{[m]}(x_k^{(m+1)}) = k \tag{5.2}$$

and

$$p_j^{[m]}(x_k^{(m+1)}) = 0, \quad j \neq k, k-1. \tag{5.3}$$

Moreover, if we use the notations Λ_m and $\Lambda_m(k)$ of Section 3, we have the following

Lemma 5.1. *Let* $\Lambda_m^*(k) = \{\mu \in \Lambda_m(k) : \mu_i \leq \mu_{i+1} \leq \mu_i + 1\}$. *Then*

$$\pi_\mu(\boldsymbol{x}^\mu) = 0, \qquad \forall \mu \in \Lambda_m(k) \setminus \Lambda_m^*(k)$$

and

$$\pi_\mu(\boldsymbol{x}^\mu) = k!(m-k)!, \qquad \forall \mu \in \Lambda_m^*(k).$$

Any $\mu \in \Lambda^*_m(k)$ corresponds to a path from $x^{(0)}_0 = (0,0)$ to $x^{(m)}_k = (m-k,k)$ which lies entirely in $\{0 \le u \le m-k, 0 \le v \le k\}$, or, geometrically, a path that represents the "shortest way" from $(0,0)$ to $(m-k,k)$. For each path in $\Lambda^*_m(k)$ the directional derivative $D^n_{\boldsymbol{x}^\mu}$ takes the form

$$D^m_{\boldsymbol{x}^\mu} = \frac{\partial^m}{\partial u^{m-k} \partial v^k}, \qquad \forall \mu \in \Lambda^*_m(k). \tag{5.4}$$

Therefore, in this case, we have

Theorem 5.2. *Let $x^{(m)}_k = (m-k,k)$, $1 \le k \le m$ and $m \le n$. If $f \in C^n(\mathbb{S}_n)$, then*

$$\lambda_n[\boldsymbol{x}^0, \dots, \boldsymbol{x}^{n-1}, x^{(n)}_k] f = (n-k)!k! \int_{\mathbb{R}^2} \frac{\partial^n}{\partial u^{n-k} \partial v^k} f(y) \sum_{\mu \in \Lambda^*_n(k)} M(y|\boldsymbol{x}^\mu) dy. \tag{5.5}$$

Moreover,

$$\lambda_n[\boldsymbol{x}^0, \dots, \boldsymbol{x}^{n-1}, x^{(n)}_k] f = \frac{\partial^n}{\partial u^{n-k} \partial v^k} f(x^*), \tag{5.6}$$

where $x^* \in \mathbb{S}_n$.

Formula (5.6) shows that for this configuration of interpolation points the finite difference $\lambda_n[\boldsymbol{x}^0, \dots, \boldsymbol{x}^{n-1}, x^{(n)}_k] f$ is closely related to the n-th partial derivative of f, one more reason to justify our definition. But ties are still closer: introducing the cardinal forward differences $\Delta^{(i,j)} f(x)$ by the well–known recurrence

$$\begin{aligned} \Delta^{(0,0)} f(u,v) &= f(u,v) \\ \Delta^{(i+1,j)} f(u,v) &= \Delta^{(i,j)} f(u+1,v) - \Delta^{(i,j)} f(u,v) \\ \Delta^{(i,j+1)} f(u,v) &= \Delta^{(i,j)} f(u,v+1) - \Delta^{(i,j)} f(u,v), \end{aligned}$$

we obtain

Theorem 5.3. *For any $0 \le k \le n$*

$$\lambda_n[\boldsymbol{x}^0, \dots, \boldsymbol{x}^{n-1}, x^{(n)}_k] f = \Delta^{(n-k,k)} f(0,0). \tag{5.7}$$

Thus, if we introduce the spline function

$$M^n_k(x) = k!(n-k)! \sum_{\mu \in \Lambda^*_n(k)} M(x|\boldsymbol{x}^\mu),$$

which can be easily seen to be a piecewise polynomial of degree $n-2$ supported on $[0, n-k] \times [0,k]$, and which is in fact a cube spline, we can combine (5.5) and (5.7) to identify M^n_k as the Peano kernel for the forward difference. Precisely, for any $n \in \mathbb{N}_0$ and $0 \le i \le n$

$$\Delta^{(n-i,i)} f(0,0) = \int_{\mathbb{R}^2} \frac{\partial^n}{\partial u^{n-i} \partial v^i} f(y) M^n_i(y) dy. \tag{5.8}$$

Finally, let us take the interpolation points inside the standard triangle $\mathbb{S} = \{(u,v) : u, v \geq 0, u+v \leq 1\}$, and give an estimate of the interpolation error.

Corollary 5.4. *Let $x_k^{(m)} = (\frac{m-k}{n}, \frac{k}{n})$, $1 \leq k \leq m$ and $m \leq n$, and $L_n(f)$ be the Lagrange interpolation polynomials based on these points. If $f \in C^n(\mathbb{S})$, then*

$$|L_n(f,x) - f(x)| \leq \sum_{k=0}^{n+1} \frac{1}{(n+1-k)!k!} \left\| \frac{\partial^{n+1}}{\partial u^{n+1-k} \partial v^k} f \right\|_\infty \prod_{i=1}^{n+1-k} \left| u - \frac{i-1}{n} \right| \prod_{i=1}^{k} \left| v - \frac{i-1}{n} \right|.$$

Acknowlegdement

We thank the referee for his kind and very careful review and for his many helpful suggestions.

References

[1] C. de Boor and A. Ron. Computational aspects of polynomial interpolation in several variables. *Math. Comp.*, **58** (1992), 705–727.

[2] M. Gasca. Multivariate polynomial interpolation. In W. Dahmen, M. Gasca, and C. A. Micchelli, editors, *Computation of Curves and Surfaces*, pages 215–236. Kluver Academic Publishers, 1990.

[3] Isaacson and Keller. *Analysis of Numerical Methods.* John Wiley & Sons, 1966.

[4] R. A. Lorentz. *Multivariate Birkhoff Interpolation.* Number 1516 in Lecture Notes in Mathematics. Springer Verlag, 1992.

[5] C. A. Micchelli. On a numerically efficient method of computing multivariate B–splines. In W. Schempp and K. Zeller, editors, *Multivariate Approximation Theory*, pages 211–248. Birkhäuser, Basel, 1979.