Multivariate approximation at fake nodes

Multivariate approximation at fake nodes

S. De Marchi*†, F. Marchetti**, E. Perracchione+, D. Poggiali†
*Dipartimento di Matematica "Tullio Levi-Civita", Università di Padova, Italy;
**Dipartimento di Salute della Donna e del Bambino, Università di Padova, Italy
+Dipartimento di Matematica DIMA, Università di Genova, Italy
†PNC - Padova Neuroscience Center, Università di Padova, Italy

April 9, 2024

Abstract

The main goal of the present paper is to extend the interpolation via the so-called *mapped bases without resampling* to *any* basis and dimension. So far indeed, we investigated the univariate case, its extension to rational polynomial interpolation and its natural application to numerical integration.

The concept of mapped bases has been widely studied, but all the proposed methods show convergence provided that the function is resampled at the mapped nodes. In applications, this is often physically unfeasible. Thus, we propose an effective method for interpolating via mapped bases in the multivariate setting. We might refer to the method as *Fake Nodes Approach* (FNA). Our theoretical results are confirmed by various numerical experiments devoted to point out the robustness of the proposed scheme.

1 Introduction

Multivariate (scattered) data approximation problems come out in various applications and, in this sense, are one of the most attractive research topics in applied mathematics, numerical analysis, engineering, data science and in all those fields that need to treat (big) data. Many methods have already been proven to be effective numerical tools, such as multivariate splines, Radial Basis Functions (RBFs) and in some sense also finite elements [11, 14, 18, 43, 45]. More precisely, multivariate splines are popular tools in computer aided geometric design and geometric modelling, indeed they also offer the possibility of approximating closed surfaces defined by scattered

data and they can be seen as the limit of subdivision schemes, see e.g. [37, 38]. Concerning finite elements, the Finite Element Method (FEM) and all its recent generalizations, such as virtual elements [8], can be considered as the reference tools for people working on numerical methods for PDEs and for all the researchers that need to discretize real-life models. The main drawback of FEM lies in the construction and refinement of usually costly meshes. Alternatives schemes that in the last decades attracted people working on PDEs and in approximation theory are *meshless methods*, see e.g. [5, 44]. They are based on RBF approximation and lead to intuitive discretization schemes which are also *easy-to-implement*. The RBF methods only depend on the distances among the scattered nodes and therefore the so-constructed basis is data-dependent, see e.g. [27]. This is one of the key feature that makes the difference with respect to multivariate polynomial approximation of total degree (see e.g. [17]). The drawbacks of not being data-dependent, give rise to the problem of finding *unisolvent* sets of points for polynomial approximants and, for stability reasons, to the problem of looking for *nearly-optimal* interpolation. Both problems are not trivial and not completely solved in the multivariate setting. As a confirm, many recent papers focus on identifying data sets that make the interpolation problem *well-posed*, refer e.g. to [9, 24, 25].

In this work we investigate the so-called *Fake Nodes Approach* (FNA). Firstly introduced for univariate polynomial and rational polynomial interpolation (cf. [6, 23]), it is here extended to the multivariate setting. The FNA is based on the idea of mapping bases, which corresponds to mapping both the interpolation and the evaluation points. Such concept is not new and has been, also recently, studied by various authors, see e.g. [1, 4, 34]. In those papers the mapping approach leads to resampling the function on different node sets, ensuring faster convergence and/or higher stability in the reconstruction process. Practically, the so-constructed mapped points take advantage of mitigating oscillations that might appear in the reconstruction process, such as the well-known Runge and Gibbs phenomena (cf. [30, 31, 42]). The main difference with our approach is that there is no need to resample the function and, at the same time, we are able to take into account the behaviour of the function itself. This is meaningful especially for applications, where usually the measurements are provided at grid data. Moreover, obtaining measurements at other locations is usually costly or not even possible in real-life scenarios.

Concerning the problem of mitigating the Gibbs phenomenon (see e.g. [31] for a general overview), we point out that several schemes have already been studied. For instance, the application of subdivision schemes

is discussed in [2] while, in the context of meshfree methods, the so-called Variably Scaled Discontinuous Kernels (VSDKs) are a well-established and recent tool for mitigating the Gibbs effect (see [10, 21, 22, 40, 41]). One of the results of this paper is that the FNA shows many similarities with the VSDKs, with the flexibility of working with any generic basis.

We now need to fix some notation. We denote by $\mathbb{P}_n$ the space of the univariate polynomials (spanned by any basis), by $\mathbb{P}_n^d$ the space of the d-variate polynomials of total degree $\leq n$ and by $\bigotimes_{i=1}^{d} \mathbb{P}_n$ the space of d-tensor product polynomials of degree $\leq n$. As already mentioned, aside the Gibbs phenomenon, in multivariate polynomial approximation we also face two other important aspects: finding unisolvent sets of points and stabilizing the approximation process. Both are challenging problems. Indeed, concerning unisolvent sets for polynomial approximation of total degree, the recent research orbits around the properties of the family of *Lissajous points* (cf. [24, 25]), which unfortunately are not unisolvent for polynomial interpolation of total degree. A special family of *degenerate Lissajous points* in the square $[-1, 1]^2$ are the *Padua points* which form a set of explicitly known unisolvent quasi-optimal points for polynomial interpolation of total degree, given as the union of Chebyshev grids (cf. [9, 29]). The stability indicator represented by the *Lebesgue constant* characterizes them by a logarithmic growth [13, 19, 36]. Therefore, in the two dimensional case, it seems plausible that if we are able to map the given nodes to the Padua points, as a kind of benefit, aside unisolvency, we may also mitigate the Runge effect. A different idea could be the one of mapping the interpolation points on Chebyshev grids, which can be defined in any dimension.

To summarize, in the present work, theoretical studies, with a particular focus on the *Lebesgue constant*, and various numerical experiments are devoted to show the efficacy, the easy implementation and hence the applicability of the FNA to many bases.

The guidelines of the paper are as follows. In Section 2 we review the basics of multivariate interpolation and approximation schemes. The fake nodes and their theoretical properties are presented in Section 3. Sections 4 and 5 focus on two specific maps and on the expansion of the approximant for kernel and polynomial bases. Numerical tests are presented in Section 6. The last section deals with conclusions and work in progress.

2 Preliminaries

As observed in the Introduction, there are many real-life applications where we face the scattered data interpolation problem defined below.

Problem 2.1 (Multivariate scattered data interpolation) *Let us consider a set of pairwise distinct data points* $X_N = \{\boldsymbol{x}_i,\ i = 1, \dots, N\} \subseteq \Omega$, $\Omega \subseteq \mathbb{R}^d$. *Let* $F_N = \{f(\boldsymbol{x}_i),\ i = 1, \dots, N\}$ *be the associated function values, sampled from a function* $f : \Omega \longrightarrow \mathbb{R}$ *at* X_N. *Then, the problem consists in finding a function* $P_f \in \mathcal{B}_N$, *with* $\mathcal{B}_N$ *a given finite dimensional function space, so that*

$$P_f(\boldsymbol{x}_i) = f(\boldsymbol{x}_i), \quad i = 1, \dots, N.$$

A common choice is to assume

$$P_f \in \mathcal{B}_N \coloneqq \operatorname{span}\{B_1, \dots, B_N\},$$

where $B_i : \Omega \longrightarrow \mathbb{R}$, $i = 1, \dots, N$, are the *basis functions*. We may also consider the so-called *cardinal* functions $u_i \in \mathcal{B}_N$, which are obtained by solving the linear system

$$\mathsf{A}\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{b}(\boldsymbol{x}), \tag{1}$$

where $\mathsf{A}_{ij} = B_i(\boldsymbol{x}_j)$, $i, j = 1, \dots, N$, $\boldsymbol{u}(\boldsymbol{x}) = (u_1(\boldsymbol{x}), \dots, u_N(\boldsymbol{x}))^\intercal$ and the right hand side is given by $\boldsymbol{b}(\boldsymbol{x}) = (B_1(\boldsymbol{x}), \dots, B_N(\boldsymbol{x}))^\intercal$. It then turns out, e.g. by Cramer's rule, that $u_i(\boldsymbol{x}_j) = \delta_{ij}$, $i, j = 1, \dots, N$. Hence, for $\boldsymbol{x} \in \Omega$, given the column vector of the cardinal functions $\boldsymbol{u}(\boldsymbol{x})$ and the one of the function values $\boldsymbol{f} = (f(\boldsymbol{x}_1), \dots, f(\boldsymbol{x}_N))^\intercal$, we can write the interpolant in cardinal form as

$$P_f(\boldsymbol{x}) = \boldsymbol{f}^\intercal\, \boldsymbol{u}(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega. \tag{2}$$

The cardinal form allows to introduce the *Lebesgue constant* $\Lambda_N(\Omega)$, which depends on the domain where we pick the points and on the function space (cf. [7, 13])

$$\Lambda_N(\Omega) = \sup_{\boldsymbol{x} \in \Omega} \sum_{i=1}^{N} |u_i(\boldsymbol{x})|. \tag{3}$$

As a note, we remind that the Lebesgue constant is the key ingredient for the stability analysis of the interpolation process. The well-posedness of the scattered data problem can be analysed by resorting to the linear system

$$\mathsf{A}\boldsymbol{\alpha} = \boldsymbol{f}, \tag{4}$$

where A and $\boldsymbol{f}$ are defined above and $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_N)^\intercal$ is the vector of the unknown coefficients. By solving it we recover the classical form of the interpolant:

$$P_f(\boldsymbol{x}) = \boldsymbol{\alpha}^\intercal \boldsymbol{b}(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega. \tag{5}$$

Definition 2.1 (Well-posedness) *The problem defined in* (4) *is well-posed if and only if the matrix* A *is non-singular.*

Unfortunately, this is not always trivial, as for polynomial approximants. Indeed, while in the univariate setting we can interpolate N distinct data with a polynomial of degree $\leq N-1$, in the multivariate setting the existence and uniqueness of the solution of the scattered data interpolation problem is not always verified. To understand this, we need two more ingredients: *Haar systems* and *unisolvent sets* (for more details see the following books [17, 43]).

Definition 2.2 (Haar system) *The finite-dimensional linear space* $\mathcal{B}_N \subseteq C(\Omega)$*, with basis* $\{B_i\}_{i=1}^N$*, is a Haar space on* Ω *if*

$$\det \mathsf{A} \neq 0,$$

for any set of distinct data points $X_N = \{\boldsymbol{x}_i, i = 1, \ldots, N\} \subseteq \Omega$*. The set* $\{B_i\}_{i=1}^N$ *is called a Haar system.*

In the multivariate case, there exist only trivial Haar spaces. This is a consequence of the following Haar-Mairhuber-Curtis theorem that can be found in [28, Theorem 1.2, p.10]. It was proved originally by Haar in 1909 [32] and later independently by Mairhuber in 1956 [35] and Curtis in 1959 [16].

Theorem 2.1 (Haar-Mairhuber-Curtis) *Suppose that* $\Omega \subseteq \mathbb{R}^d$*,* $d \geq 2$*, contains an interior point. Then there exist no Haar spaces of continuous functions except for trivial ones, i.e. spaces spanned by a single function.*

According to Definition 2.2, assuming that $\{B_i\}_{i=1}^N$ is a Haar system on Ω is equivalent to state that it is a unisolvent set of functions on Ω. The notion of unisolvency for sets of points is different and sounds as follows.

Definition 2.3 (Unisolvency) *A finite set of point* $X_N \subseteq \Omega$ *is unisolvent for* $\mathcal{B}_N$ *if all the elements of* $\mathcal{B}_N$ *are completely determined by their values at* X_N*.*

While in general we do not dispose of non-trivial unisolvent bases in the multivariate setting, it is possible to consider multidimensional unisolvent sets of points. For example, the *Padua points* form a unisolvent set for bivariate polynomial interpolation of total degree.

Moreover, as a consequence of Theorem 2.1, any set of distinct nodes is unisolvent for *data-dependent kernel bases*. Therefore, to guarantee the uniqueness of the interpolant in $\mathcal{B}_N$, when possible, we consider sets of nodes that are unisolvent with respect to the chosen basis $\{B_i\}_{i=1}^N$.

Remark 2.1 (Least squares) *If the set of nodes is not unisolvent, we may relax the interpolation conditions and thus approximate the function f in the least squares sense, i.e. f is approximated in the space $\mathcal{B}_M \coloneqq span\{B_1, \dots, B_M\}$, with $M \leq N$.*

Unless otherwise noted, in what follows we refer to the interpolation problem.

3 The Fake Node Approach (FNA)

Let $S : \Omega \longrightarrow \mathbb{R}^d$ be an injective map. The main idea behind the FNA, is to construct an interpolant $R_f \in \mathcal{B}_N^S \coloneqq \text{span}\{B_1^S, \dots, B_N^S\}$ of the function f, so that

$$R_f(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i^S B_i^S(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i^S B_i(S(\boldsymbol{x})) = P_g(S(\boldsymbol{x})), \quad \boldsymbol{x} \in \Omega\,. \tag{6}$$

The function g has the property that $g_{|S(X_N)} = f_{|X_N}$, that is it assumes the same values of f at the mapped interpolation points $S(X_N)$, cf. [6, 23]. Thus, the construction of the interpolant $R_f \in \mathcal{B}_N^S$ is equivalent to building a classical interpolant $P_g \in \mathcal{B}_N$ at the *fake* or mapped nodes $S(X_N)$. In what follows we will use the term *fake nodes*, thinking to this mapping process.

Provided that we have a unisolvent set of points for the given basis, R_f can be constructed by solving the linear system

$$\mathsf{A}^S \boldsymbol{\alpha}^S = \boldsymbol{f}, \tag{7}$$

where $\boldsymbol{\alpha}^S = (\alpha_1^S, \dots, \alpha_N^S)^\intercal$, $\boldsymbol{f}$ as previously defined, and

$$\mathsf{A}^S = \begin{pmatrix} B_1^S(\boldsymbol{x}_1) & \dots & B_1^S(\boldsymbol{x}_N) \\ \vdots & \ddots & \vdots \\ B_N^S(\boldsymbol{x}_1) & \dots & B_N^S(\boldsymbol{x}_N) \end{pmatrix}.$$

Concerning the cardinal form of the mapped interpolant we state the following proposition.

Proposition 3.1 (Cardinal form) *Let $X_N = \{\boldsymbol{x}_i,\ i = 1, \dots, N\} \subseteq \Omega$ be a set of pairwise distinct data points and let $u_i \in \mathcal{B}_N$, $i = 1, \dots, N$ be the basis functions. Let $S : \Omega \longrightarrow \mathbb{R}^d$ be an injective map. The functions $\{u_1, \dots, u_N\}$ are cardinal on $S(\Omega)$ for the fake nodes $S(X_N)$ if and only if the mapped functions $\{u_1 \circ S, \dots, u_N \circ S\}$ are cardinal for the original set of nodes X_N.*

The proof is trivial and comes out by imposing the cardinality conditions to the functions u_i^S.

Hence we can write the interpolant at the fake nodes in *cardinal* form as:

$$R_f^S(\boldsymbol{x}) = \boldsymbol{f}^\intercal \boldsymbol{u}^S(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega\,, \tag{8}$$

where $\boldsymbol{u}^S(\boldsymbol{x}) = (u_1^S(\boldsymbol{x}), \dots, u_N^S(\boldsymbol{x}))^\intercal$.

A practical way for computing the cardinal form of the interpolant based on the fake nodes easily follows from its determinant form (cf. [33], [12, Theorem 3, p. 3] and [15, Exercise 15, p.64]).

Proposition 3.2 (Determinantal form) *Given a (finite) unisolvent set of nodes $X_N \subseteq \Omega$ for the space $\mathcal{B}_N$ and the associated function values F_N. Let $\boldsymbol{f}$ and $\boldsymbol{b}(\boldsymbol{x})$ be as in (2) and (5), respectively. For $\boldsymbol{x} \in \Omega$, the interpolant P_f is given by*

$$P_f(\boldsymbol{x}) = -\frac{1}{\det(\mathsf{A})} \det \left(\begin{array}{c|c} 0 & \boldsymbol{b}^\intercal(\boldsymbol{x}) \\ \hline \boldsymbol{f} & \mathsf{A} \end{array} \right),$$

where the matrix A is such that $\mathsf{A}_{ij} = B_i(\boldsymbol{x}_j)$.

Proof: Similarly to [33], we can consider the function

$$q(\boldsymbol{x}) = \det \left(\begin{array}{c|c} 0 & \boldsymbol{b}^\intercal(\boldsymbol{x}) \\ \hline \boldsymbol{f} & \mathsf{A} \end{array} \right) \in \mathcal{B}_N.$$

so that

$$q(\boldsymbol{x}_i) = \det \left(\begin{array}{c|c} -f(\boldsymbol{x}_i) & \mathbf{0} \\ \hline \boldsymbol{f} & \mathsf{A} \end{array} \right) = -f(\boldsymbol{x}_i)\det(\mathsf{A}),\ i = 1, \dots N.$$

If follows that the function

$$Q(\boldsymbol{x}) = -\frac{1}{\det(\mathsf{A})} q(\boldsymbol{x}),$$

belongs as well to $\mathcal{B}_N$ and satisfies $Q(\boldsymbol{x}_i) = f(\boldsymbol{x}_i)$, $i = 1, \dots, N$. The thesis follows from the uniqueness of the interpolant.

∎

For the computation of the interpolant at the fake nodes in its cardinal form, by using the previous result we get the following proposition.

Proposition 3.3 (Determinantal form for cardinal basis) *Let $X_N \subseteq \Omega$ be a unisolvent set of nodes for the space $\mathcal{B}_N$ and let F_N be the associated function values. Let $S : \Omega \longrightarrow \mathbb{R}^d$ be an injective map and $\boldsymbol{u}^S(\boldsymbol{x})$ be the vector of the mapped cardinal functions on $\boldsymbol{x} \in \Omega$. The interpolant R_f^S becomes*

$$R_f^S(\boldsymbol{x}) = -\det \mathsf{U}_N^S, \quad \forall\, \boldsymbol{x} \in \Omega, \tag{9}$$

with

$$\mathsf{U}_N^S = \left(\begin{array}{c|c} 0 & (\boldsymbol{u}^S(\boldsymbol{x}))^\intercal \\ \hline \boldsymbol{f} & \mathsf{I}_N \end{array} \right),$$

and where I_N denotes the $N \times N$ identity matrix.

The proof of this statement directly follows from Proposition 3.2 by observing that A reduces to I_N using the cardinal functions. Here we provide an alternative proof by induction.

Proof: For $N = 1$ and $\boldsymbol{x} \in \Omega$, we have that

$$R_f^S(\boldsymbol{x}) = -\det \mathsf{U}_1^S = -\det \begin{pmatrix} 0 & u_1^S(\boldsymbol{x}) \\ f(\boldsymbol{x}_1) & 1 \end{pmatrix} = f(\boldsymbol{x}_1) u_1^S(\boldsymbol{x}).$$

We suppose that the assertion holds true for $N - 1$. Since

$$R_f^S(\boldsymbol{x}) = -\det \mathsf{U}_N^S = -\left[(-1)^{N+2} (-1)^{N+1} f(\boldsymbol{x}_N) u_N^S(\boldsymbol{x}) \det \mathsf{I}_N + (-1)^{2N+2} \det \mathsf{U}_{N-1}^S \right],$$

by induction we get

$$\begin{aligned} R_f^S(\boldsymbol{x}) &= -\left(f(\boldsymbol{x}_N)u_N^S(\boldsymbol{x}) - \det \mathsf{U}_{N-1}\right) \\ &= \left(f(\boldsymbol{x}_N)u_N^S(\boldsymbol{x}) + f(\boldsymbol{x}_{N-1})u_{N-1}^S(\boldsymbol{x}) + \cdots + f(\boldsymbol{x}_1)u_1^S(\boldsymbol{x})\right), \end{aligned}$$

which also shows the equivalence between (8) and (9). ∎

Concerning the Lebesgue constant associated to R_f^S, we state the following result.

Proposition 3.4 (Equivalence of the Lebesgue constant) *Let $S : \Omega \longrightarrow \mathbb{R}^d$ be an injective map. Let $X_N \subseteq \Omega$ be a unisolvent set of nodes for the space $\mathcal{B}_N$, and $u_i^S \in \mathcal{B}_N^S$, $i = 1, \ldots, N$, be the associated cardinal functions. Then, the Lebesgue constant $\Lambda^S(\Omega)$ associated to the mapped nodes is*

$$\Lambda^S(\Omega) = \Lambda(S(\Omega)).$$

Proof: By definition of $\Lambda^S(\Omega)$, for $\boldsymbol{x} \in \Omega$ we trivially have that:

$$\Lambda^S(\Omega) = \sup_{\boldsymbol{x}\in\Omega} \sum_{i=1}^{N} |u_i^S(\boldsymbol{x})| = \sup_{\boldsymbol{x}\in\Omega} \sum_{i=1}^{N} |u_i(S(\boldsymbol{x}))| = \sup_{\boldsymbol{y}\in S(\Omega)} \sum_{i=1}^{N} |u_i(\boldsymbol{y})| = \Lambda(S(\Omega)).$$

∎

Remark 3.1 *The proposition states that the interpolation at the mapped basis $\mathcal{B}_N^S$ inherits the Lebesgue constant of the fake nodes $S(X_N)$ over the 'standard' basis $\mathcal{B}_N$.*

The Lebesgue constant, as well-known, represents the stability constant of the interpolation process. For analyzing the stability, we thus study an interpolant of perturbed data $\tilde{f}(\boldsymbol{x}_i)$ sampled at $\boldsymbol{x}_i$, $i = 1, \ldots, N$, i.e. data affected by measurement errors.

Proposition 3.5 (Stability) *Let $S : \Omega \longrightarrow \mathbb{R}^d$ be an injective map and $X_N \subseteq \Omega$ be a unisolvent set of nodes for the space $\mathcal{B}_N$. Let $\boldsymbol{f}$ be the associated vector of function values and $\tilde{\boldsymbol{f}}$ be the vector of perturbed values. Let R_f^S and $R_{\tilde{f}}^S$ be the interpolant of the function values $\boldsymbol{f}$ and $\tilde{\boldsymbol{f}}$ respectively. Then,*

$$||R_f^S - R_{\tilde{f}}^S||_{\infty,\Omega} \leq \Lambda^S(\Omega)\ \|\boldsymbol{f} - \tilde{\boldsymbol{f}}\|_{\infty,X_N}.$$

Proof: Taking into account that $g_{|S(X_N)} = f_{|X_N}$ and thus also $\tilde{g}_{|S(X_N)} = \tilde{f}_{|X_N}$, we deduce that

$$
\begin{aligned}
||R_f^S - R_{\tilde{f}}^S||_{\infty,\Omega} = ||P_g - P_{\tilde{g}}||_{\infty,S(\Omega)} = \sup_{x\in S(\Omega)} \left| \sum_{i=1}^{N} (g_i(\boldsymbol{x}_i) - \tilde{g}_i(\boldsymbol{x}_i))\ u_i(\boldsymbol{x}) \right| = \\
= \sup_{x\in\Omega} \left| \sum_{i=1}^{N} (g_i(S(\boldsymbol{x}_i)) - \tilde{g}_i(S(\boldsymbol{x}_i)))\ u_i(S(\boldsymbol{x})) \right| \leq \\
\leq \sup_{x\in\Omega} \sum_{i=1}^{N} |u_i(S(\boldsymbol{x}))|\ \ |g_i(S(\boldsymbol{x_i})) - \tilde{g}_i(S(\boldsymbol{x_i}))| \leq \\
\leq \sup_{x\in\Omega} \sum_{i=1}^{N} |u_i(S(\boldsymbol{x}))|\ \ \max_{i=1,\dots,N} |g_i(S(\boldsymbol{x_i})) - \tilde{g}_i(S(\boldsymbol{x_i}))| = \\
= \Lambda(S(\Omega))\ \max_{i=1,\dots,N} \left| f(\boldsymbol{x}_i) - \tilde{f}_i(\boldsymbol{x}_i) \right| \\
= \Lambda^S(\Omega)\ \|\boldsymbol{f} - \tilde{\boldsymbol{f}}\|_{\infty,X_N}.
\end{aligned}
$$

∎

Consistently with Remark 3.1, the FNA approach also inherits the error of the classical approach, as shown in the following proposition.

Proposition 3.6 (Error bound inheritance) *Letting S, X_N, f and R_f^S, as above. Then, for any given function norm, we have*

$$||R_f^S - f||_\Omega = ||P_g - g||_{S(\Omega)},$$

where $g_{|S(X_N)} = f_{|X_N}$.

Proof: From (6) we know that $R_f^S = P_g \circ S$. Choosing g such that $g \circ S = f$ on Ω (this function exists since the map S is injective), we get

$$||R_f^S - f||_\Omega = ||P_g \circ S - g \circ S||_\Omega = ||P_g - g||_{S(\Omega)}.$$

∎

We conclude the section by showing a pseudo-code for the FNA, which works for every choice of the map S and of the chosen basis (see Algorithm 1).

Algorithm 1 Fake nodes interpolation.

Inputs:
$X_N = \{\boldsymbol{x}_i,\ i = 1, \dots, N\} \subseteq \Omega \subset \mathbb{R}^d$: interpolation nodes;
$\boldsymbol{f} = (f(\boldsymbol{x}_1), \dots, f(\boldsymbol{x}_N))^\intercal$: column vector of the function values at X_N;
$\mathcal{B}_N \ = \{B_i,\ i = 1, \dots, N\}$: basis functions;
$X_M^e = \{\boldsymbol{x}_i^e,\ i = 1, \dots, M\} \subseteq \Omega \subset \mathbb{R}^d$: evaluation nodes;
$S : \Omega \longrightarrow \mathbb{R}^d$: the injective map.
Core

1. Set the interpolation matrix $\mathsf{A}_{ij}^S = B_i(S(\boldsymbol{x}_j))$, $i, j = 1, \dots, N$.
2. Set the evaluation matrix $\mathsf{E}_{ij}^S = B_i(S(\boldsymbol{x}_j^e))$, $i = 1, \dots, N$, $j = 1, \dots, M$.
3. Compute the coefficient vector $\boldsymbol{\alpha}^S$ by solving $\mathsf{A}^S \boldsymbol{\alpha}^S = \boldsymbol{f}$.
4. Compute the interpolant at X_M^e: $\boldsymbol{R}_f^S = \mathsf{E}^S \boldsymbol{\alpha}^S$.

Output: $\boldsymbol{R}_f^S = (R_f^S(\boldsymbol{x}_1^e), \dots, R_f^S(\boldsymbol{x}_M^e))^\intercal$.

Remark 3.2 *The computational cost of the Algorithm 1 is basically $\mathcal{O}(N^3)$ floating point operations, which is the same computational cost of the classical interpolation algorithm obtainable from Algorithm 1 by using as S the identity map. Indeed, the additional cost is linear and comes from the function evaluations, giving a cost of $\mathcal{O}(N + M)$ floating point operations.*

In what follows, we focus on two specific maps, by extending to the multidimensional setting the S-Runge and S-Gibbs algorithms introduced in [23].

4 S-Gibbs

Assumption 4.1 (For S-Gibbs) *We assume that Ω is the union of p pairwise disjoint sets Ω_k, $k = 1, \dots, p$, and that f is piecewise continuous. In particular, the discontinuities of f appear only at the boundaries of the subsets $\partial\Omega_k$, $k = 1, \dots, p$.*

As an example see the domain in Figure 1 (left).

Under such assumptions, we consider the function S

$$S(\boldsymbol{x}) = \boldsymbol{x} + \boldsymbol{a}_i, \tag{10}$$

for $\boldsymbol{x} \in \Omega_i$ and $\boldsymbol{a}_i = a_i \ \mathrm{diag}(\mathsf{l}_d)$, $a_i \in \mathbb{R}$, $i = 1, \dots, p$. The latter should be chosen so that we obtain p disjoint sets, according to the discontinuities

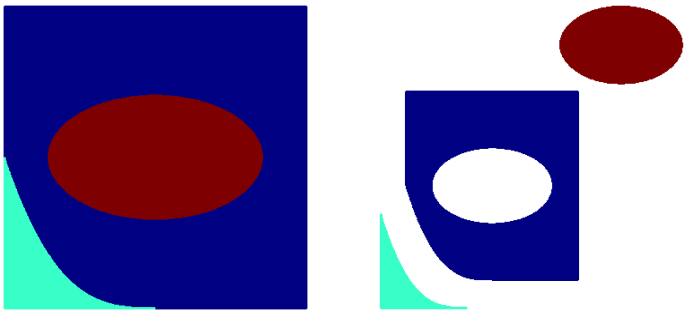

Figure 1: Left: example of possible domain Ω under Assumption 4.1. Right: the mapped domain via the S-Gibbs map S.

of f. We call this map the *S-Gibbs* map. For the previous example, the corresponding S-Gibbs map provides the result shows in Figure 1 (right).

Before discussing some numerical tests, we wish to briefly recall the basics of kernel-based approximation and more specifically of the so-called Variably Scaled Discontinuous Kernels (VSDKs), which offer an alternative to multivariate polynomial approximation.

4.1 On the approximation by kernels: a brief introduction

The method described in the previous section works with any basis. In particular, for kernels (refer e.g. to [14]), it shows strong similarities with the so-called VSDKs [10, 21, 22]. Indeed, using kernel-based interpolants, we consider $P_f \in \text{span}\{\kappa(\cdot, \boldsymbol{x}_1), \ldots, \kappa(\cdot, \boldsymbol{x}_N)\}$, where $\kappa : \Omega \times \Omega \longrightarrow \mathbb{R}$ consists of a (strictly) positive definite radial kernel. Since it is radial with respect to the Euclidean norm $\|\cdot\|$, we can associate to the kernel a univariate function $\phi : [0, \infty) \longrightarrow \mathbb{R}$ so that:

$$\kappa(\boldsymbol{x}, \boldsymbol{y}) = \phi(r), \quad \text{where} \quad r = \|\boldsymbol{x} - \boldsymbol{y}\|.$$

The distance r can be rescaled via the so-called *shape parameter* $\gamma \in \mathbb{R}, \gamma > 0$, so that we may consider $\phi(\gamma r)$ as the *radial basis function*.

Some well-known examples of RBFs, which include functions of different regularities both globally defined and compactly supported, are displayed in the Table 1.

To determine the kernel-based interpolant, we then reduce to solving a system of the form (4), with $\mathsf{A} = \phi(\mathsf{D})$, where the *distance matrix* D is given by

$$\mathsf{D}_{ij} = ||\boldsymbol{x}_i - \boldsymbol{x}_j||, \quad i, j = 1, \ldots, N.$$

$\phi_G(\gamma r) = e^{-(\gamma r)^2},$	Gaussian	C^∞
$\phi_{M,0}(\gamma r) = e^{-\gamma r},$	Matérn	C^0
$\phi_{M,2}(\gamma r) = e^{-\gamma r}(1+\gamma r),$	Matérn	C^2
$\phi_{W,0}(\gamma r) = (1-\gamma r)_+^2,$	Wendland	C^0
$\phi_{W,2}(\gamma r) = (1-\gamma r)_+^4 (4\gamma r+1)$	Wendland	C^2

Table 1: Most popular radial basis functions; $(\cdot)_+$ denotes the truncated power function.

When using the S-Gibbs map (10), the kernel matrix is denoted by $\mathsf{A}^S = \phi(\mathsf{D}^S)$, where the entries of D^S are defined as

$$\mathsf{D}^S_{ij} = ||S(\boldsymbol{x}_i) - S(\boldsymbol{x}_j)||, \quad i,j = 1,\dots,N.$$

Hence, the kernel-based interpolant based on the fake nodes $R_f^S : \Omega \longrightarrow \mathbb{R}$ is defined by solving (4) with kernel matrix $\mathsf{A}^S = \phi(\mathsf{D}^S)$.

4.2 VSDKs for the Gibbs phenomenon

Variably Scaled Kernels, shortly VSKs, have been introduced in [10]. More precisely, given $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$ and a *scale function* $\psi : \mathbb{R}^d \longrightarrow \mathbb{R}$, the idea is to consider the kernel

$$\kappa^\psi(\boldsymbol{x},\boldsymbol{y}) \coloneqq \kappa\left((\boldsymbol{x},\psi(\boldsymbol{x})),(\boldsymbol{y},\psi(\boldsymbol{y}))\right),$$

where κ is a kernel on $\mathbb{R}^{d+1} \times \mathbb{R}^{d+1}$. This process produces an *augmented* set of nodes $\tilde{X}_N = \{\tilde{\boldsymbol{x}}_i = (\boldsymbol{x}_i, \psi(\boldsymbol{x}_i)),\ i = 1,\dots,N\} \subseteq \Omega \times \mathbb{R}$. Hence, the function is interpolated by means of standard kernels on the space $\mathbb{R}^{d+1} \times \mathbb{R}^{d+1}$. In other words, the VSDK approximant $V_f : \Omega \longrightarrow \mathbb{R}$ is constructed by solving a system of the form (4) whose kernel matrix is defined as $\mathsf{A}^\psi = \phi(\mathsf{D}^\psi)$, where

$$\mathsf{D}^\psi_{ij} = ||\tilde{\boldsymbol{x}}_i - \tilde{\boldsymbol{x}}_j||, \quad i,j = 1,\dots,N.$$

To mitigate the Gibbs phenomenon, the VSDKs have been introduced in [22] and their convergence properties discussed in [21]. Under the Assumption 4.1, in [22], the idea consists in selecting the scale function ψ as

piecewise constant, i.e. as $\psi(\boldsymbol{x}) = b_i$, $b_i \in \mathbb{R}$, for $\boldsymbol{x} \in \Omega_i$. Being the RBF interpolation methods dependent on the distance among the nodes, we observe that the *"VSDKs are similar to the FNA"*, in the sense that $\mathsf{A}^{\psi} \approx \mathsf{A}^{S}$. Indeed, we observe that both VSDKs and FNA preserve the distances between points that lie on the same subdomain $\Omega_k \subset \Omega$, while they increase the distances between points lying on different subregions in Ω. To point out this this, in Figure 2 we show the effect of applying the fake nodes and the VSDKs on 12 nodes, assuming a discontinuity at the point $x = 0.5$.

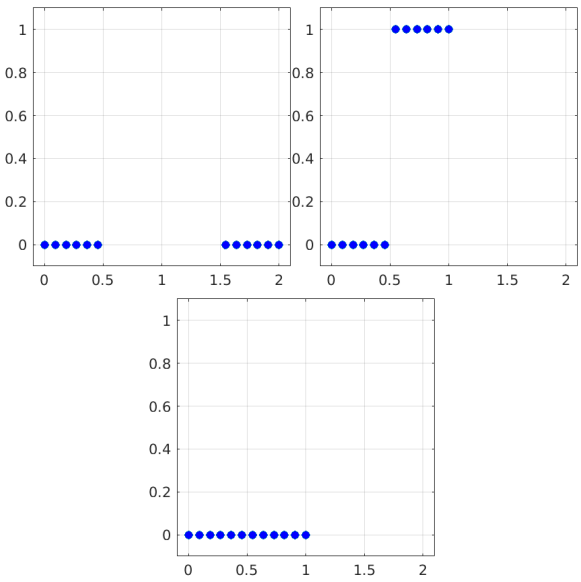

Figure 2: Top left: fake nodes mapped via S. Top right: nodes for the VSDK setting in $\mathbb{R}^2$. Bottom: original set of equispaced nodes.

To further point out the similarity between VSDKs and FNA, we recall some fundamental results. We denote by $\mathcal{N}_\kappa$ the native space associate to a given kernel κ (cf. e.g. [45]). An error bound for kernel-based interpolants in terms of the so-called fill distance

$$h = \sup_{\boldsymbol{x}\in\Omega} \left(\min_{\boldsymbol{x}_k \in X_N} \|\boldsymbol{x} - \boldsymbol{x}_k\| \right),$$

is given by

$$|f(\boldsymbol{x}) - P_f(\boldsymbol{x})| \leq C h^{\beta} ||f||_{\mathcal{N}_\kappa}, \quad \boldsymbol{x} \in \Omega.$$

This upper bound holds for $C^{2\beta}(\Omega \times \Omega)$ kernels. Concerning the constant C, we refer the reader to [27, Theorem 14.4, p. 120] for further details. For

our purposes, we need to focus on the fill-distance defined by the FNA and VSDKs. Thanks to [21, Theorem 3.4], for the VSDKs the following error bound holds

$$|f(\boldsymbol{x}) - V_f(\boldsymbol{x})| \leq C_V h_V^{\beta} ||f||_{\mathcal{N}_{\kappa^\psi}}, \quad \boldsymbol{x} \in \Omega,$$

where h_V is the fill-distance of the data set under the effect of the VSK setting, i.e. the fill-distance of the set $\{(\boldsymbol{x}, \psi(\boldsymbol{x})),\ \boldsymbol{x} \in \Omega\}$. Moreover, concerning the FNA approach, thanks to Proposition 3.6 we have that

$$|f(\boldsymbol{x}) - R_f^S(\boldsymbol{x})| \leq C_S h_S^{\beta} ||f||_{\mathcal{N}_\kappa}, \quad \boldsymbol{x} \in \Omega,$$

where h_S is the fill-distance of the *fake* data set. Indeed, under our assumptions on the maps S and ψ, we obtain $h_S \approx h_V$ (see also Figure 2), we deduce that the kernel-based FNA and the VSDKs provide almost analogous approximations.

Remark 4.1 (Trade-off) *Aside the relation between accuracy and fill-distance, another important aspect of the FNA is that, as the VSDKs, it can be eventually used to overcome the usual instability of interpolation procedures. Such instability is typically due to the ill-conditioning of the interpolation matrix. Indeed, for a given set of nodes X_N, the stability is related to the separation distance*

$$q = \frac{1}{2} \min_{\substack{x_i, x_j \in X_N \\ i \neq j}} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2.$$

In particular, one can numerically observe that the ill-conditioning increases as the separation distance and the smallest singular value of the interpolation matrix decrease; refer e.g. to [27]. This suggests to select the map S so that the separation distance on the mapped set of nodes q^S is so that $q^S \gg q$. Unfortunately, this may not be effective enough. Indeed, the separation distance grows as the fill distance and therefore, being the latter an accuracy indicator, we face a conflict between accuracy and numerical stability [10].

5 S-Runge

In this section, we focus on polynomial interpolation, since it is known to be heavily affected by the Runge phenomenon. The main idea relies on mapping nodes via a function S such that $\Lambda^S(\Omega) \leq \Lambda(\Omega)$. This is possible thanks to the fact that we inherit the Lebesgue constant on the mapped nodes; see Proposition 3.4. Moreover, since the Lebesgue constant is known

to be a stability indicator and referring to Remark 4.1, a *smaller* $\Lambda^S(\Omega)$ lead to a possible more stable, i.e. *safe*, approximations.

In what follows, we present two different methods to deal with the Runge phenomenon in the multidimensional setting. While the former is built upon recursive one-dimensional reconstructions and it can be extended to any dimensions, with the latter we restrict to the two-dimensional case and we take advantage of the *quasi-optimality* of the *Padua points*.

5.1 The lines approach

In case of univariate polynomial interpolation, we know that the Chebyshev-Lobatto (CL) nodes represent an optimal choice for stability, meaning that their Lebesgue constant grows logarithmically. Therefore, given a set of equispaced nodes on $[a, b] \subseteq \mathbb{R}$, the "fake CL nodes" are obtained by mapping them to $[-1, 1]$ using the S-map

$$S(x) = -\cos\left(\frac{x-a}{b-a}\pi\right), \quad x \in [a, b]. \tag{11}$$

When dealing with multidimensional equispaced grid data, we can extend this idea by considering tensor-product CL grids. Let

$$\Omega = \bigotimes_{j=1}^{d} [a_j, b_j] \subseteq \mathbb{R}^d,$$

with $a_j < b_j$. On $[a_j, b_j]$ we take the vector of $N_j \in \mathbb{N}$ equispaced nodes

$$\boldsymbol{t}^{(j)} \coloneqq (t_1^{(j)}, \ldots, t_{N_j}^{(j)}),$$

where $t_1^{(j)} = a_j$, $t_{N_j}^{(j)} = b_j$ and

$$t_i^{(j)} = a_j + \frac{i-1}{N_j - 1}(b_j - a_j), \quad i = 2, \ldots, N_j - 1.$$

Let $N = \prod_{j=1}^{d} N_j$. For the FNA, we associate to

$$X_N = \bigotimes_{j=1}^{d} \boldsymbol{t}^{(j)} \subset \Omega,$$

the usual set of function values F_N. By using (11), we obtain the vectors of nodes

$$S(\boldsymbol{t}^{(j)}) = \left(S(t_1^{(j)}), \ldots, S(t_{N_j}^{(j)})\right), \ j = 1, \ldots, d.$$

Then, we simply construct the tensor product interpolant at the corresponding Chebyshev grid (see e.g. [3, 18])

$$S(X_N) = \bigotimes_{j=1}^{d} S(\boldsymbol{t}^{(j)}).$$

In Figure 3, we show an example of a two-dimensional equispaced grid mapped on a tensor-product CL grid.

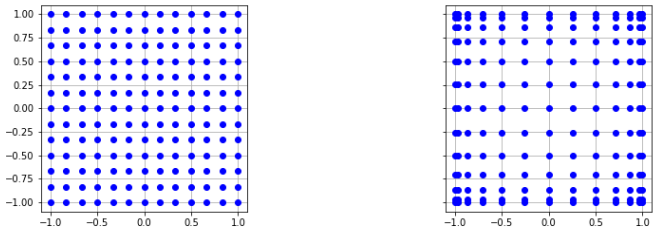

Figure 3: From a 13×13 equispaced grid (left) we obtain a tensor-product CL grid (right) by applying the univariate map S in all directions.

5.2 The fake Padua approach

Here, we consider polynomial interpolation of total degree on $[-1, 1]^2$ (any two-dimensional finite rectangular domain could be considered as well). We recall that the basis for bivariate polynomials of total degree n has cardinality

$$N = \binom{n+2}{2} = \frac{(n+1)(n+2)}{2},$$

and consequently $P_f \in \mathbb{P}_n^d$.

Let us consider the sets of nodes

$$X_N = \left\{ \left(\frac{2(i-1)}{n} - 1, \frac{2(j-1)}{n+1} - 1 \right), \begin{array}{c} i = 1, \ldots, n+1 \\ j = 1, \ldots, n+2 \end{array}, \begin{array}{c} i + j \equiv 0 \\ (\mathrm{mod}\ 2) \end{array} \right\}, \tag{12}$$

extracted from an equispaced $(n+1) \times (n+2)$ grid on $[-1, 1]^2$. The set of the so-called *Padua points* is known to be quasi-optimal for bivariate interpolation of total degree on the square. Therefore the FNA should map the given nodes eventually to the Padua points. There are four families of Padua points (obtained by counterclockwise rotations of degrees $k\pi/2$, $k =$

$1, 2, 3$ of the first family). For the sake of simplicity, we consider the first family, which is defined as follows:

$$\mathrm{Pad}_N = \left\{ \varphi\left(\frac{k\pi}{n(n+1)}\right), k = 0, \ldots, n(n+1) \right\},$$

where

$$\varphi(t) = (-\cos((n+1)t), -\cos(nt)), \quad t \in [0, \pi].$$

is the *generating curve*, a closed parametric curve in Ω, that is a special case of *Lissajous curves*, see [9, 20, 25]. We observe that the cardinality of Pad_N is exactly

$$\binom{n+1}{2}.$$

To our aims, the fundamental property of the set Pad_N is that its Lebesgue constant is of minimal growth, since it has been proven to be $\mathcal{O}((\log n)^2)$ (cf. [9]). We propose to use the FNA with the map $S : [-1, 1]^2 \longrightarrow [-1, 1]^2$ defined as

$$S(\boldsymbol{x}) = \left(-\cos\left(\pi \frac{\boldsymbol{e}_1^\intercal \boldsymbol{x} + 1}{2}\right), -\cos\left(\pi \frac{\boldsymbol{e}_2^\intercal \boldsymbol{x} + 1}{2}\right) \right).$$

where $\boldsymbol{x} = (x_1, x_2)$ and $\boldsymbol{e}_i$, $i = 1, 2$, are the unit vectors of $\mathbb{R}^2$.

In fact, it is easy to prove that the fake nodes $S(X_N)$ are exactly the Padua points of the first kind, Pad_N. Having the map S, we can construct the interpolant at the fake nodes $S(X_N)$ as described in (6). In Figure 4, we display the sets of interpolation nodes that are involved in the fake Padua approach.

Finally, we observe that if we use the condition $i + j \equiv 1 \pmod 2$ in (12), the Padua points of the second family will result. To obtain the $\pi/2$-counterclockwise rotation of the Padua points it is sufficient to swap the two coordinates in (12).

6 Numerical Experiments

In the following experiments we point out three important aspects of the fake nodes method. Precisely:

1. The versatility of the FNA with respect to different basis functions. In doing this, we focus on discontinuous test functions and therefore we test the S-Gibbs map.

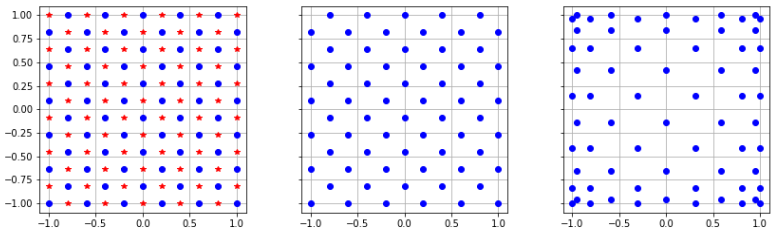

Figure 4: Here $n = 10$ (so that the cardinality is $N = 66$). On the left the set X_{66} (represented by blue dots) is extracted from a 11×12 equispaced grid (represented by both blue dots and red stars). The set X_{66} (centre) is then mapped on the set of Padua points Pad_{66} via the mapping S (right).

2. The applicability of the FNA to medical imaging. We test this via polynomial least squares.

3. The ability to mitigate the Runge effect. Also in this case we drive our attention towards polynomial bases.

The implementation of the following experiments is provided as free and open-source PYTHON software available at `https://github.com/pog87/FakeNodes2D`.

6.1 Versatility in reducing the Gibbs effect

In this subsection, our main scope consists in numerically showing the flexibility of the FNA, meaning that it can be applied to *all* basis functions, in the treatment of the Gibbs phenomenon by means of the S-Gibbs map. Indeed, this resampling-free method can be applied to any interpolation or approximation procedure.

We test three approximation techniques, i.e. three different basis functions:

- Nearest-Neighbor (NN) interpolation: for each evaluation point, the algorithm returns the function value associated to the nearest node, yielding a piecewise-constant interpolant. It is discontinuous for its nature and thus it is not affected by the Gibbs phenomenon. As a consequence, it is a challenging test for the fake nodes tool. For its implementation, we use the PYTHON function `griddata` of the package `scipy`.

- Polynomials: since for a general set of scattered data we might not have unisolvent sets, we focus on polynomial least squares and we fix the degree equal to 4. For the implementation (see e.g. [25, 26, 39]) we refer the reader to the MATLAB packages available at the CAA research group homepage `https://www.math.unipd.it/~marcov/CAA.html` and to the the GitHub repositories by Wolfgang Erb and Marco Vianello available at `https://github.com/WolfgangErb` and `https://github.com/marcovianello`, respectively.

- Kernels: in what follows, for computing the kernel-based interpolant, we fix the Matérn C^0 kernel. The shape parameter is set as $\gamma = 0.5$.

To test the efficacy of the S-Gibbs map, we take the following test function

$$f(\boldsymbol{x}) = \begin{cases} \sin(x_1 + x_2^2), & \text{if} \quad \|\boldsymbol{x}\|_2 < 0.4, \\ 1, & \text{if} \quad \|\boldsymbol{x}\|_2 \geq 0.4, \end{cases}$$

with $\boldsymbol{x} = (x_1, x_2)$, we sample it at $N = \{9, 81, 289, 1089, 4225\}$ grid data on $[-1, 1]^2$. The accuracy of the three interpolation techniques listed above is tested by evaluating the Mean Square Error (MSE) on a grid of 40^2 evaluation points. The results are reported in Figure 5. We can note that the FNA sensibly reduces the error with respect to the standard approach.

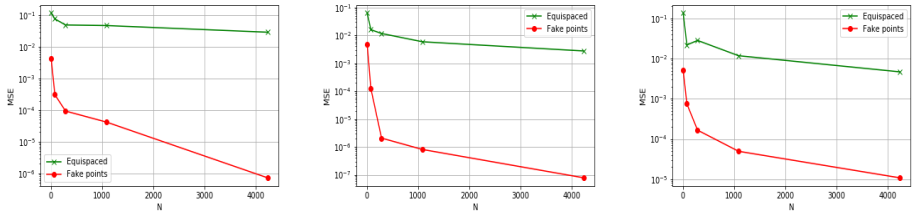

Figure 5: The MSE by varying N with polynomials (left), kernels (middle) and NN (right). The green stars represent the standard bases, while the results for mapped bases are plotted with the dotted red line.

6.2 Applicability to medical imaging

As a second example for testing the S-Gibbs map, we take the Shepp-Logan phantom, defined as the weighted sum of the characteristic functions of a chosen number of ellipses, in order to mimic an X-ray Computed Axial Tomography image of a human brain.

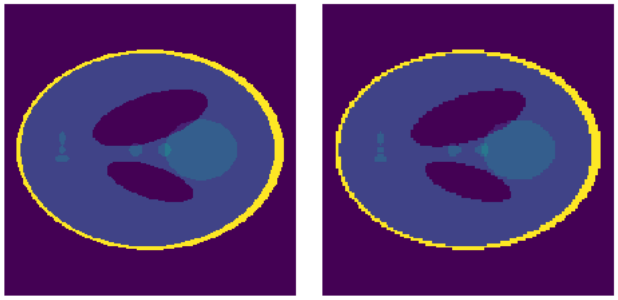

Figure 6: Original Shepp-Logan (left), subsampled Shepp-Logan (128×128) pixels (right).

This example stresses the importance of the FNA in the field of medical imaging. The Shepp-Logan phantom is plotted in Figure 6 (left) and its size is 256×256. We then subsample it on $N = \{32^2, 48^2, 64^2, 96^2, 128^2\}$ pixels (see e.g. Figure 6 right) and we evaluate the performances of the methods in reconstructing the original phantom. A graphical example is plotted in Figure 7, where we use the least squares polynomials of degree 4 with and without the use of fake nodes. The MSE is depicted in Figure 8. Once more, we can note the robustness of the presented approach in reducing the Gibbs effect.

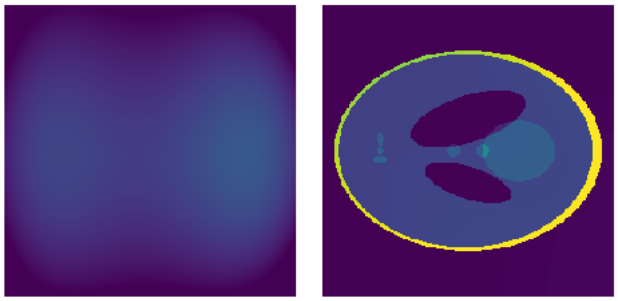

Figure 7: Left to right: standard polynomial reconstruction, fake polynomial reconstruction on 128×128 pixels.

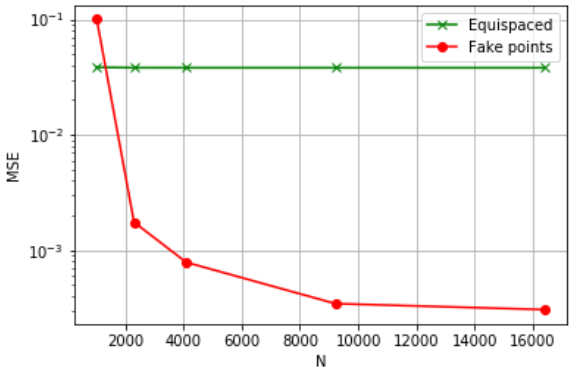

Figure 8: The MSE by varying N with polynomials for the Shepp-Logan phantom. The green stars represent the standard bases, while the results for mapped bases are plotted with the dotted red line.

6.3 Mitigating the Runge effect

In this subsection, we take the Runge-like function $f : [-1, 1]^d \longrightarrow \mathbb{R}_+$,

$$f(\boldsymbol{x}) = \frac{1}{1 + 5\|\boldsymbol{x}\|_2^2}.$$

- Consider $d = 2$. To test the lines approach, presented in Section 5.1, we compare it with the standard tensor-product polynomial reconstruction at equispaced grids and with the reconstruction obtained by resampling at Chebyshev-Lobatto grids. In Figure 9, we display the results considering a 13×15 starting interpolation grid, while in Figure 10 we observe the asymptotic behavior of the considered methods by means of an increasing sequence of $n \times n$ interpolation grids on $[-1, 1]^2$. In both cases, the reconstructions are evaluated on a 90×90 equispaced grid.

  ]],

- Considering now the fake Padua approach ($d = 2$) of Section 5.2, we test it with the bivariate polynomial reconstruction at $n \times (n+1)$ equispaced grids and with the reconstruction at Padua points with resampling. In Figure 11, we display the results considering a 10×11 starting interpolation grid, while in Figure 12 we observe the asymptotic behavior of the considered methods by means of an increasing sequence

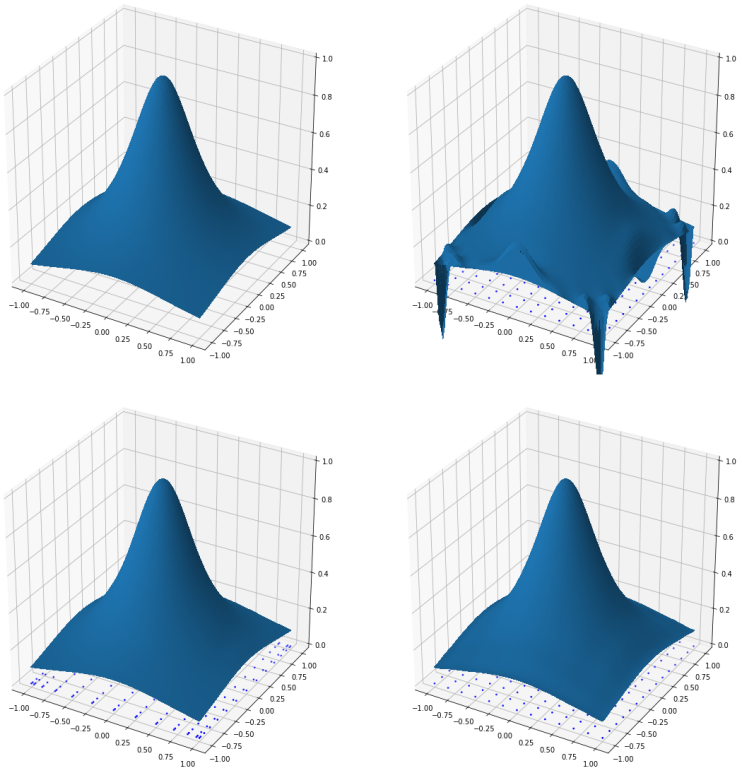

Figure 9: Results of different interpolation schemes considering a 13×15 grid: the original function (top left), the tensor-product reconstruction at the equispaced grid (top right), the reconstruction with resampling at the CL grid (bottom left) and the fake lines approach at the equispaced grid without resampling (bottom right).

of $n \times (n+1)$ interpolation grids. The equispaced $(n+10) \times (n+10)$ grid was chosen as evaluation set.

- Our last test is with $d = 3$. Being the Padua points known only on the square $[-1, 1]^2$ while the lines approach is working in any dimension, we compare the latter with the tensor-product polynomial approximation. For both the lines approach and the tensor-product approximation, we reconstruct the function on a grid of evaluation points of size $90 \times 90 \times 90$. Precisely, in Figure 13 we show the asymptotic behavior

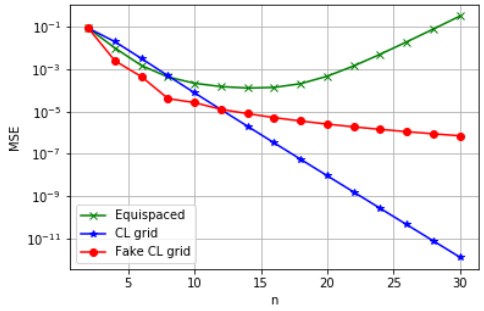

Figure 10: The MSE by varying n in dimension $d = 2$. The green crosses indicate the tensor-product reconstruction at the equispaced grid, the blue stars the reconstruction with resampling at the CL grid, while the dotted red line represents the FNA.

of both methods at increasing n^3 interpolation grids. Once more, we verify the efficacy of the FNA which is able to reduce oscillations of the Runge effect, also in higher dimensions.

As we notice in Figures 10, 12 and 13, in all the three cases the FNA leads to a smaller error than the corresponding one using classical approximation over the same equispaced nodes. On the other hand, based on our experiments, as the number of nodes increases, it is still preferable to consider samples at Chebyshev-like grids, as the approximation error is smaller than FNA approximation. This observation does not pull down the importance of using the FNA since, in most applications, Chebyshev-sampled data are not easy to get or not available at all.

7 Conclusions

We have introduced and discussed a new numerical scheme, termed Fake Node Approach (FNA), that can be used for multivariate scattered data interpolation problems. Its main advantages are: the capability to work with different approximation spaces and its easy implementation. Moreover, it allows to overcome some typical drawbacks of the reconstruction process, like the appearance of non-physical oscillations when the considered functions are characterized by steep gradients or discontinuities. All the numerical tests provided in the paper and many others that we have not included for the sake of brevity show the versatility, applicability and efficiency of the

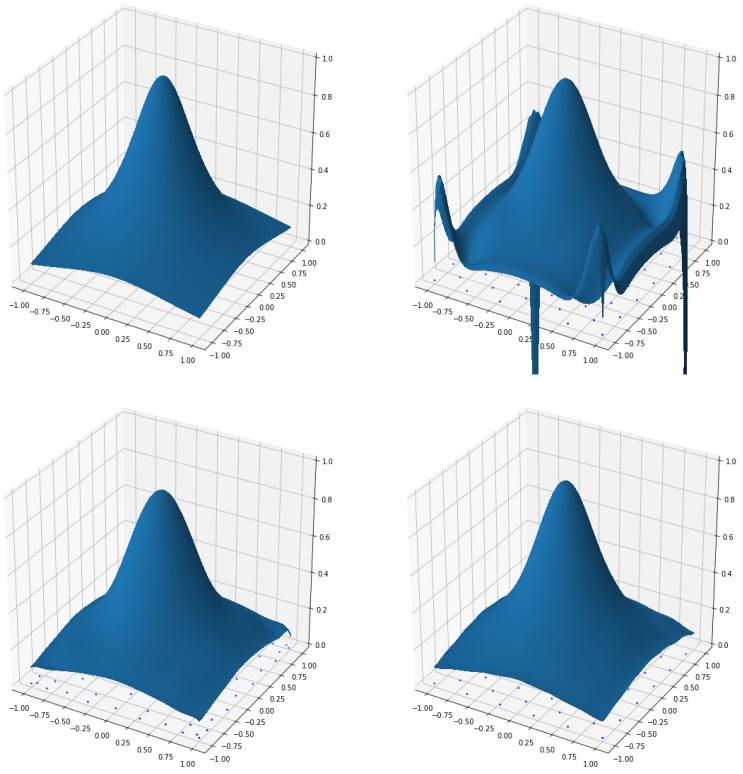

Figure 11: Results of different interpolation schemes for $n = 10$: the original function (top left), the multivariate polynomial reconstruction at the equispaced grid (top right), the reconstruction with resampling at the Padua points (bottom left) and the fake Padua approach at the equispaced grid without resampling (bottom right).

new numerical scheme.

Work in progress consists in studying its efficacy in the context of collocation schemes for solving PDEs and in investigating its applicability in the framework of machine learning.

Acknowledgments

This research has been accomplished within Rete ITaliana di Approssimazione (RITA), partially funded by GNCS-INδAM and through the European Union's Horizon 2020 research and innovation programme ERA-

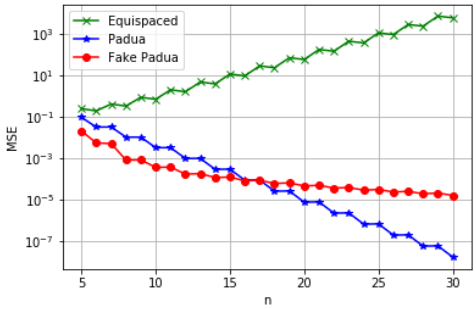

Figure 12: The MSE by varying n in dimension $d = 2$. The green crosses indicate the multivariate polynomial reconstruction at the equispaced grids, the blue stars the reconstruction with resampling at the Padua points, while the dotted red line represents the FNA.

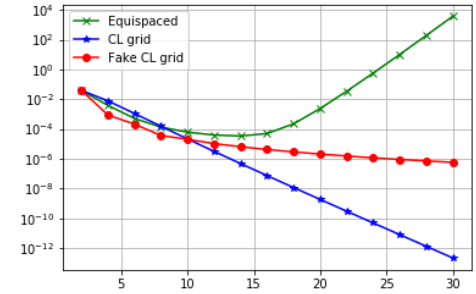

Figure 13: The MSE by varying n in dimension $d = 3$. The green crosses indicate the tensor-product reconstruction at the equispaced grid, the blue stars the reconstruction with resampling at the CL grid, while the dotted red line represents the FNA.

PLANET, grant agreement no. 689443, via the GEOEssential project.

References

[1] B. Adcock, R.B. Platte, *A mapped polynomial method for high-accuracy approximations on arbitrary grids*, SIAM J. Numer. Anal. **54** (2016), 2256–2281.

[2] S. Amat, J. Ruiz, J.C. Trillo, D.F. Yáñez, *Analysis of the Gibbs phenomenon in stationary subdivision schemes*, Appl. Math. Lett. **76** (2018), 157–163.

[3] M. Azaïez, T. Chácon Rebollo, E. Perracchione, J.M. Vega, *Recursive POD expansion for advection-diffusion-reaction equation*, Comm. Comput. Physics **24** (2018), pp. 1556–1578.

[4] A. Bayliss, E. Turkel, *Mappings and accuracy for Chebyshev pseudo-spectral approximations*, J. Comput. Phys. **101** (1992), 349–359.

[5] V. Bayona, M. Moscoso, M. Carretero, M. Kindelan, *RBD-FD formulas and convergence properties*, J. Comput. Phys. 229 (2010), 8281–8295.

[6] J.P. Berrut, S. De Marchi, G. Elefante, F. Marchetti, *Treating the Gibbs phenomenon in barycentric rational interpolation via the S-Gibbs algorithm*, Appl. Math. Letters **103** (2020), 106196.

[7] L. Bos, D. De Marchi, K. Hormann, *On the Lebesgue constant of Berrut's rational interpolant at equidistant nodes*, J. Comput. Appl. Math. **236** (2011), 504–510.

[8] F. Brezzi, S.R. Falk, L.D. Marini, *Basic principles of mixed virtual element methods*, ESAIM: Mathematical Modelling and Numerical Analysis **48** (2014), 1227–1240

[9] L. Bos, S. De Marchi, M. Vianello, *Polynomial approximation on Lissajous curves in the d−cube*. Appl. Numer. Math. **116** (2017), pp. 47–56.

[10] M. Bozzini, L. Lenarduzzi, M. Rossini, R. Schaback, *Interpolation with variably scaled kernels*, IMA J. Numer. Anal. **35** (2015), 199–219.

[11] S.C. Brenner, L.R. Scott, *The Mathematical Theory of Finite Element Methods*, Springer New York, 1994.

[12] C. Brezinski, *Biorthogonality and its Applications to Numerical Analysis*, Marcel Dekker Inc., New York, 1992.

[13] L. Brutman, *On the Lebesgue function for polynomial interpolation*, SIAM J. Numer. Anal. **15** (1978), 694–704.

[14] M.D. Buhmann, *Radial Basis Functions: Theory and Implementation*, Cambridge Monogr. Appl. Comput. Math., vol. 12, Cambridge Univ. Press, Cambridge, 2003.

[15] E.W. Cheney, *Introduction to Approximation Theory*, AMS Chelsea Pub, New York, 2000.

[16] P.C. Curtis Jr., *N-parameter families and best approximation*, Pacific J. Math. **9** (1959), pp. 1013–1027.

[17] P.J. Davis, *Interpolation and Approximation*, Dover Publications, New York, 1975.

[18] C. De Boor, *A Practical Guide to Splines*, Springer-Verlag, New York, 1978.

[19] S. De Marchi, *Polynomials arising in factoring generalized Vandermonde determinants: an algorithm for computing their coefficients*, Math. Comput. Modelling **34** (2001), 271–281.

[20] S. De Marchi, W. Erb, F. Marchetti, *Spectral filtering for the reduction of the Gibbs phenomenon for polynomial approximation methods on Lissajous curves with applications in MPI*, Dolomites Res. Notes Approx. **10** (2017), pp. 128–137.

[21] S. De Marchi, W. Erb, F. Marchetti, E. Perracchione, M. Rossini, *Shape-Driven Interpolation with Discontinuous Kernels: Error Analysis, Edge Extraction and Applications in MPI*, to appear in SIAM J. Sci. Comput. (2019), https://doi.org/10.1137/19M1248777.

[22] S. De Marchi, F. Marchetti, E. Perracchione, *Jumping with Variably Scaled Discontinuous Kernels (VSDKs)*, Bit Numer. Math. (2019), https://doi.org/10.1007/s10543-019-00786-z

[23] S. De Marchi, F. Marchetti, E. Perracchione, D. Poggiali, *Polynomial interpolation via mapped bases without resampling*, J. Comput. Appl. Math., **364** (2020), 112347–12.

[24] P. Dencker, W. Erb, *A unifying theory for multivariate polynomial interpolation on general Lissajous-Chebyshev nodes*, preprint 2019.

[25] W. Erb, *Bivariate Lagrange interpolation at the node points of Lissajous curves - the degenerate case*, Appl. Math. Comput. **289** (2016), 409–425.

[26] W. Erb, C. Kaethner, M. Ahlborg, T.M. Buzug, *Bivariate Lagrange interpolation at the node points of non-degenerate Lissajous curves*, Numer. Math. **133** (2016), 685–705.

[27] G.E. Fasshauer, *Meshfree Approximations Methods with* Matlab, World Scientific, Singapore, 2007.

[28] G.E. Fasshauer, M.J. McCourt, *Kernel-based Approximation Methods Using* Matlab, World Scientific, Singapore, 2015.

[29] M.S. Floater, *Polynomial interpolation on interlacing rectangular grids*, J. Approx. Theory **222** (2017), pp. 64–73.

[30] J.W. Gibbs, *Fourier's Series*, Nature **59** (1898).

[31] D. Gottlieb, C.W. Shu, *On the Gibbs phenomenon and its resolution*, SIAM Review **39** (1997), pp. 644–668.

[32] A. Haar, *Die Minkowskische Geometrie und die Annäherung an stetige Funktionen*, Math. Ann. **18** (1918), pp. 294–311.

[33] D. Kalman, *The Generalized Vandermonde Matrix.* Mathematics Magazine, 57(1), 15-21 (1984). doi:10.2307/2690290

[34] D. Kosloff, H. Tal-Ezer, *A modified Chebyshev pseudospectral method with an $O(N^{-1})$ time step restriction*, J. Comput. Phys. **104** (1993), 457–469.

[35] J.C. Mairhuber, *On Haar's theorem concerning Chebychev approximation problems having unique solutions*, Proc. Amer. Math. Soc. **7** (1956), 609–615.

[36] G. Mastroianni, D. Occorsio, *Optimal systems of nodes for Lagrange interpolation on bounded intervals. A survey*, J. Comput. Appl. Math. **134** (2001) pp. 325–341.

[37] J. Peters, U. Reif, *Subdivision Surfaces*, Springer series Geometry and Computing monograph 3, 2008.

[38] J. Peters, U. Reif, *Analysis of generalized B-spline subdivision algorithms*, SIAM J. of Numer. Anal. **32** (1998), 728–748,

[39] F. Piazzon, A. Sommariva, M. Vianello, *Caratheodory-Tchakaloff Least Squares*, Sampling Theory and Applications 2017, IEEE Xplore Digital Library, 672–676.

[40] M. Rossini, *Interpolating functions with gradient discontinuities via variably scaled kernels*, Dolom. Res. Notes Approx. **11** (2018), 3–14.

[41] L. Romani, M. Rossini, D. Schenone, *Edge detection methods based on RBF interpolation*, J. Comput. Appl. Math. **349** (2019), 532–547.

[42] C. Runge, *Über empirische Funktionen und die Interpolation zwischen äquidistanten Ordinaten*, Zeit. Math. Phys. **46** (1901), 224–243.

[43] L.L. Schumaker, *Spline Functions: Basic Theory*, John Wiley & Sons, New York, 1981.

[44] A. Tolstykh, D. Shirobokov, *On using radial basis functions in a finite difference mode with applications to elasticity problems*, Comput. Mech. **33** (2003), 68–79.

[45] H. Wendland, *Scattered Data Approximation*, Cambridge Monogr. Appl. Comput. Math., vol. 17, Cambridge Univ. Press, Cambridge, 2005.