

Data-Driven Extrapolation Via Feature Augmentation Based on Variably Scaled Thin Plate Splines

*Original*

Data-Driven Extrapolation Via Feature Augmentation Based on Variably Scaled Thin Plate Splines / Campagna, R.; Perracchione, E.. - In: JOURNAL OF SCIENTIFIC COMPUTING. - ISSN 0885-7474. - ELETTRONICO. - 88:1(2021). [10.1007/s10915-021-01526-8]

*Availability:*

This version is available at: 11583/2956979 since: 2023-06-04T10:44:46Z

*Publisher:*

Springer

*Published*

DOI:10.1007/s10915-021-01526-8

*Terms of use:*

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

*Publisher copyright*

(Article begins on next page)

# Data-driven extrapolation via feature augmentation based on variably scaled thin plate splines

Rosanna Campagna<sup>+</sup>, Emma Perracchione\*

<sup>+</sup>Dept. of Mathematics and Physics, University of Campania “L. Vanvitelli”, Italy

\*Dept. of Mathematics DIMA, University of Genova, Italy

rosanna.campagna@unicampania.it; perracchione@dima.unige.it

**Keywords.** Data-driven extrapolation, Feature augmentation, Radial basis functions, Variably scaled kernels, Thin plate splines, Ridge regression.  
65D05 41A05 65D10 65D15

## Abstract

The data driven extrapolation requires the definition of a functional model depending on the available data and has the application scope of providing reliable predictions on the unknown dynamics. Since data might be scattered, we drive our attention towards kernel models that have the advantage of being meshfree. Precisely, the proposed numerical method makes use of the so-called Variably Scaled Kernels (VSKs), which are introduced to implement a feature augmentation-like strategy based on discrete data. Due to the possible uncertainty on the data and since we are interested in modelling the behaviour of the considered dynamics, we seek for a regularized solution by ridge regression. Focusing on polyharmonic splines, we investigate their implementation in the VSK setting and we provide error bounds in Beppo-Levi spaces. The performances of the method are then tested on functions which are common in the framework of the Laplace transform inversion. Comparisons with Support Vector Regression (SVR) are also carried out and show that the proposed method is effective particularly since it does not require to train complex architecture constructions.

# 1 Introduction

The extrapolation of functional data is an attractive issue for both approximation and information theory, due to the ever-growing need to predict states from noisy data. This is a challenging problem, indeed recovering information on the dynamics of data through extrapolation is a so-defined *hopelessly ill-conditioned problem* [1].

While some authors refer to the extrapolation as super-resolution, i.e. the extrapolation of fine-scale details from low-resolution data [2], we here consider the extrapolation *out of samples*. For polynomial and rational analytic functions, an *extrapolant* can be defined as a least-squares polynomial approximant [3]. Usually, the extrapolation from univariate data is defined by spline models [4], which might give unpredictable results. Indeed, the behaviour of the solution out of the reconstruction interval is strongly forced by the model gradient constraints at the boundaries. This implies that one usually recovers reliable approximations only locally, i.e. for short extensions outside the domain.

In this work we instead consider kernel-based *meshfree* models constructed via polyharmonic splines whose definition includes in particular the cubic RBF and the Thin Plate Spline (TPS); for further details see [5]. For the extrapolation issue, we take advantage of the use of the so-called Variably Scaled Kernels (VSKs) [6], which might lead to more stable and accurate schemes [7]. Here the VSKs are introduced to define a feature augmentation strategy (see e.g. [8, 9]) with the aim to realize *accurate* extrapolation in larger neighbouring subsets. For strictly positive definite kernels, error bounds for VSK interpolants can be found in [10, 11]. Here we extend the VSKs to work with strictly *conditionally* positive kernels and we provide error bounds for the TPS-VSK interpolant in Beppo-Levi spaces. Furthermore, since data might be affected by measurement errors and uncertainty, we look for a regularized solution by ridge regression (see e.g. [12]). In doing so we focus on samples that decay rationally or exponentially.

The need to define data-driven rational or exponential models arises in several contexts, such as radioactive decay, atmospheric pressure changes, epidemic growth patterns from infectious disease outbreak data. To model situations in which the decay begins rapidly and then slows down to get closer and closer to zero, generalized spline models have already been investigated [13, 14], so as the exponential regression [15]. Here the extrapolation is driven by data through the definition of a scaling function which

is computed via a preliminary non-linear fitting of the available data. Our experiments show that the so-constructed method can be effectively used for the extrapolation problem and point out that the proposed tool might be seen as an alternative to the sometimes computationally demanding Support Vector Regression (SVR), that might be considered as the *state of the art* for regression purposes; refer e.g. to [16].

The paper is organized as follows. In Section 2 we briefly review the basics of kernel-based interpolants. In Section 3, we investigate the VSKs based on polyharmonic splines and we provide error bounds. In Section 4 we further discuss our feature augmentation strategy and the related definition of a scaling function. Section 5 deals with numerical experiments, while conclusions and future work are outlined in Section 6.

## 2 Kernel framework

Let  $X = \{\mathbf{x}_i, i = 1, \dots, n\} \subseteq \Omega$ ,  $\Omega \subseteq \mathbb{R}^d$ , be a set of distinct scattered points and  $F = \{f_i = f(\mathbf{x}_i), i = 1, \dots, n\} \subseteq \mathbb{R}$  be the associated data values, sampled from an unknown function  $f : \Omega \rightarrow \mathbb{R}$ , we define an approximating model for such samples and empirically study its effectiveness for the extrapolation outside the domain  $\Omega$ . A kernel-based interpolant, defined on a set  $\Lambda \subseteq \mathbb{R}^d$ , is a function  $P_f : \Lambda \supseteq \Omega \rightarrow \mathbb{R}$  of the form [5]:

$$P_f(\mathbf{x}) = \sum_{k=1}^n \alpha_k \kappa(\mathbf{x}, \mathbf{x}_k) + \sum_{j=1}^m \beta_j p_j(\mathbf{x}), \quad \mathbf{x} \in \Lambda, \quad (1)$$

with conditionally positive definite radial kernels  $\kappa : \Lambda \times \Lambda \rightarrow \mathbb{R}$  of order  $l$  on  $\mathbb{R}^d$  and  $\{p_1, \dots, p_m\}$  a basis for the  $m$ -dimensional linear space  $\Pi_{l-1}^d$  of polynomials of total degree less than or equal to  $l - 1$  in  $d$  variables, where

$$m = \binom{l-1+d}{l-1}.$$

The coefficients  $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n)^\top$ , and  $\boldsymbol{\beta} = (\beta_1, \dots, \beta_m)^\top$ , are uniquely identified by imposing the  $n$  interpolation conditions  $P_f(\mathbf{x}_i) = f_i$ ,  $i = 1, \dots, n$ , and the  $m$  constraints for polynomial reproduction [17]. In other words, they are determined by solving

$$\underbrace{\begin{pmatrix} \mathbf{A} & \mathbf{P} \\ \mathbf{P}^\top & \mathbf{O} \end{pmatrix}}_{\mathbf{K}} \underbrace{\begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{pmatrix}}_{\boldsymbol{\gamma}} = \underbrace{\begin{pmatrix} \mathbf{f} \\ \mathbf{0} \end{pmatrix}}_{\mathbf{g}}, \quad (2)$$

where

$$\begin{aligned} \mathbf{A}_{ik} &= \kappa(\mathbf{x}_i, \mathbf{x}_k), \quad i, k = 1, \dots, n, \\ \mathbf{P}_{ij} &= p_j(\mathbf{x}_i), \quad i = 1, \dots, n, \quad j = 1, \dots, m. \end{aligned}$$

Moreover,  $\mathbf{f} = (f_1, \dots, f_n)^\top$ ,  $\mathbf{0}$  is a zero vector of length  $m$  and  $\mathbf{O}$  is a zero matrix  $m \times m$ . To assess the conditions under which such system admits a unique solution, we introduce the definition of *unisolvent set*.

**Definition 2.1.** *A set of points  $X = \{\mathbf{x}_i, i = 1, \dots, n\} \subseteq \Omega$  is called  $(l - 1)$ -unisolvent if the only polynomial of total degree at most  $l - 1$  interpolating zero data on  $X$  is the zero polynomial.*

We remark that if  $\kappa$  is a strictly conditionally positive definite function of order  $l$  on  $\mathbb{R}^d$  and the set  $X$  of data points forms a  $(l - 1)$ -unisolvent set, the system (2) admits a unique solution; refer e.g. to [5, Theorem 7.2].

Under the hypothesis of radial kernels, there exists a function  $\varphi : [0, \infty) \rightarrow \mathbb{R}$ , known as Radial Basis Function (RBF), such that for all  $\mathbf{x}, \mathbf{y} \in \Lambda$

$$\kappa(\mathbf{x}, \mathbf{y}) = \varphi(\|\mathbf{x} - \mathbf{y}\|_2) = \varphi(r), \quad r := \|\mathbf{x} - \mathbf{y}\|_2.$$

Principally to overcome instability issues due to the ill-conditioning of the kernel matrix (see e.g. [18]), the so-called Variably Scaled Kernels (VSKs) [6] have been introduced. In particular, given a scaling function  $\psi : \mathbb{R}^d \rightarrow \Sigma \subseteq \mathbb{R}$  we denote by

$$G_\Psi(\Lambda) = \{(\mathbf{x}, \psi(\mathbf{x})) \mid \mathbf{x} \in \Lambda\} \subset \Lambda \times \Sigma, \quad (3)$$

the graph of the function  $\psi$  with respect to  $\Lambda$ . Then, the VSK  $\kappa^\Psi : \Lambda \times \Lambda \rightarrow \mathbb{R}$  is defined as a standard kernel  $\kappa : G_\Psi(\Lambda) \times G_\Psi(\Lambda) \rightarrow \mathbb{R}$ , i.e.

$$\kappa^\Psi(\mathbf{x}, \mathbf{y}) := \kappa((\mathbf{x}, \psi(\mathbf{x})), (\mathbf{y}, \psi(\mathbf{y}))), \quad (4)$$

for  $\mathbf{x}, \mathbf{y} \in \Lambda$ .

Suitable choices of the scaling function  $\psi$  might improve stability and preserve shape properties of the original function, see [10, 11, 19, 20]. In the next section, we will focus on specific kernels, and will use the VSKs as feature augmentation strategy for the extrapolation issue.

### 3 Feature augmentation for polyharmonic splines

The selected kernel bases are the so-called *polyharmonic splines*, defined as

$$\kappa_{d,l}(\mathbf{x}, \mathbf{y}) = \varphi_{d,l}(r) = \begin{cases} r^{2l-d}, & \text{for } d \text{ odd,} \\ r^{2l-d} \log r, & \text{for } d \text{ even,} \end{cases} \quad (5)$$

for  $\mathbf{x}, \mathbf{y} \in \Lambda$  and  $2l > d$ , with  $d$  the space dimension. They have been introduced by J. Duchon, R. Harder and R.N. Desmarais the 1970s; refer to [21, 22]. Such functions are strictly conditionally positive definite of order  $l$ . Moreover, the condition number of the approximation problem induced by polyharmonic splines is invariant under rotations, translations and uniform scalings; see [23, Theorem 4.5].

Concerning the VSKs seen in the context of feature augmentation tools, if we suppose to have some a priori information on the samples, e.g. the knowledge about the asymptotic behaviour of the function or about the steep gradients, we can directly encode such information into the kernel. Indeed, the scaling function  $\psi$  might be selected so that it *mimics* the samples. In the following, we propose to use a non-linear fitting of  $\psi : \mathbb{R}^d \rightarrow \Sigma \subseteq \mathbb{R}$  acting as a feature augmentation rule (see e.g. [8, 24]). So the VSK encodes the information through the new feature introduced by  $\psi$ . At this end, we firstly define a function  $\Psi : \Lambda \rightarrow G_\Psi(\Lambda)$  as

$$\Psi(\mathbf{x}) := (\mathbf{x}, \psi(\mathbf{x})),$$

that extends the data vector  $\mathbf{x} \in \Omega$ , including one more *feature* depending on the original ones. Then, equivalently to (4), in the VSK setting, we define the kernel  $\kappa^\Psi : \Lambda \times \Lambda \rightarrow \mathbb{R}$ , given by

$$\kappa^\Psi(\mathbf{x}, \mathbf{y}) = \kappa(\Psi(\mathbf{x}), \Psi(\mathbf{y})),$$

where  $\kappa : G_\Psi(\Lambda) \times G_\Psi(\Lambda) \rightarrow \mathbb{R}$ .

Being  $G_\Psi(\Lambda) \subseteq \mathbb{R}^{d+1}$ , we now investigate the VSK interpolant for polyharmonic splines, whose definition depends on  $d$  (see [6]).

**Definition 3.1.** *Given  $\Lambda \subseteq \mathbb{R}^d$ , and let  $\psi : \mathbb{R}^d \rightarrow \Sigma$  be the scaling function, the VSK interpolant  $P_f^\Psi : \Lambda \rightarrow \mathbb{R}$  is defined as*

$$P_f^\Psi(\mathbf{x}) = P_f((\mathbf{x}, \psi(\mathbf{x}))) \quad \mathbf{x} \in \Lambda,$$

where  $(\mathbf{x}, \psi(\mathbf{x})) \in G_\Psi(\Lambda)$  and  $P_f : G_\Psi(\Lambda) \rightarrow \mathbb{R}$  and  $G_\Psi$  are defined as in (1) and (3), respectively.

Because of the equivalence in Definition 3.1 we will refer to both  $P_f^\Psi : \Lambda \rightarrow \mathbb{R}$  and  $P_f : G_\Psi(\Lambda) \rightarrow \mathbb{R}$  as VSK interpolants. We better formalize this concept via the following proposition.

**Proposition 3.1.** *Let  $\kappa_{d,l}$  be a polyharmonic spline with  $2l > d + 1$ . Let  $\psi : \mathbb{R}^d \rightarrow \Sigma \subseteq \mathbb{R}$  be the scaling function for the VSK setting and  $G_\Psi(\Lambda)$  its associated graph on  $\Lambda$ . Given a set of scattered data  $X = \{\mathbf{x}_1, \dots, \mathbf{x}_n\} \subseteq \Omega$ ,  $\Omega \subseteq \mathbb{R}^d$ , and the associated function values  $F = \{f_1, \dots, f_n\} \subseteq \mathbb{R}$ , the coefficients  $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n)^\top$  and  $\boldsymbol{\beta} = (\beta_1, \dots, \beta_m)^\top$  for the VSK interpolant  $P_f^\Psi : \Lambda \rightarrow \mathbb{R}$  are given by the solution of a system of the form (2), with*

$$\mathbf{A}_{ik} = \kappa_{d+1,l}((\mathbf{x}_i, \psi(\mathbf{x}_i)), (\mathbf{x}_k, \psi(\mathbf{x}_k))), \quad i, k = 1, \dots, n,$$

$$P_{ij} = p_j((\mathbf{x}_i, \psi(\mathbf{x}_i))), \quad i = 1, \dots, n, \quad j = 1, \dots, m,$$

$p_j \in \Pi_{l-1}^{d+1}$ ,  $j = 1, \dots, m$ , and

$$m = \binom{l+d}{l-1}. \quad (6)$$

*Proof.* From (4) and (5), we obtain that for  $\mathbf{x}, \mathbf{y} \in \Lambda$

$$\kappa_{d,l}^\Psi(\mathbf{x}, \mathbf{y}) = \kappa_{d+1,l}((\mathbf{x}, \psi(\mathbf{x})), (\mathbf{y}, \psi(\mathbf{y}))).$$

Moreover, since for  $\mathbf{x} \in \Lambda$ ,  $P_f^\Psi(\mathbf{x}) = P_f((\mathbf{x}, \psi(\mathbf{x})))$ , where  $(\mathbf{x}, \psi(\mathbf{x})) \in G_\Psi(\Lambda)$ , we have that:

$$P_f^\Psi(\mathbf{x}) = \sum_{k=1}^n \alpha_k \kappa_{d+1,l}((\mathbf{x}, \psi(\mathbf{x})), (\mathbf{x}_k, \psi(\mathbf{x}_k))) + \sum_{j=1}^m \beta_j p_j((\mathbf{x}, \psi(\mathbf{x}))), \quad (7)$$

where  $p_1, \dots, p_m$ , are a basis for  $\Pi_{l-1}^{d+1}$  with  $m$  defined as in (6). By imposing the interpolation conditions  $P_f^\Psi(\mathbf{x}_i) = f_i$  the thesis follows.  $\blacksquare$

To provide error bounds for the VSK interpolant constructed via polyharmonic splines, we first need to recall that to any kernel  $\kappa$  we can associate the so-called native space, refer e.g. to [17, 18] for its definition. For our scopes, we only need to point out that the error analysis for polyharmonic splines is carried out in particular native spaces, precisely in the Beppo–Levi spaces, see [25, p. 366]. They are also referred to as homogeneous Sobolev spaces. Letting  $h \in \mathbb{Z}^+$ , the Beppo–Levi space  $\text{BL}_h(\mathbb{R}^d)$  is defined as the space of all

tempered distributions  $f$  on  $\mathbb{R}^d$  such that  $D^\xi f \in L^2(\mathbb{R}^d)$  for all  $\xi \in \mathbb{R}^d$  so that  $|\xi| = h$ . The associated seminorm is

$$|f|_{\text{BL}_h(\mathbb{R}^d)}^2 = \sum_{|\xi|=h} \frac{h!}{\xi_1! \cdots \xi_d!} \|D^\xi f\|_2^2.$$

Set  $l = 2$  in (5), for  $d = 1$  and  $2$  (which are the cases of interest for our numerical experiments),  $\varphi_{d,l}$  is usually referred to as cubic kernel and Thin Plate Spline (TPS), respectively. Both kernels are strictly conditionally positive definite of order 2, which implies that we have to impose the polynomial reproduction.

Here we extend to VSKs an existing result true for TPSs. The following proposition theoretically grants the effectiveness of the proposed model, giving local upper bounds for the VSK interpolant.

**Proposition 3.2.** *Let  $X = \{x_1, \dots, x_n\} \subseteq \Omega$ ,  $\Omega \subseteq \mathbb{R}$ , be a set of scattered data and  $F = \{f_1, \dots, f_n\} \subseteq \mathbb{R}$  the associated function values. Let  $\kappa_{1,2}$  be the cubic kernel,  $\psi : \mathbb{R} \rightarrow \Sigma \subseteq \mathbb{R}$  the scaling function for the VSK setting and  $G_\Psi(\Lambda)$  its associated graph on  $\Lambda$ . Let  $P_f : G_\Psi(\Lambda) \rightarrow \mathbb{R}$  be the VSK interpolant. There exists an absolute constant  $C$  such that, on any closed triangle  $T$  corresponding to three nodes on  $G_\psi(X) = \{(x_i, \psi(x_i)) \mid x_i \in X\}$ , we obtain*

$$\|f - P_f\|_{\infty, T} \leq C\rho |f|_{\text{BL}_2(\mathbb{R}^2)}, \quad f \in \text{BL}_2(\mathbb{R}^2),$$

where  $\rho$  is the longest edge of  $T$ .

*Proof.* Given  $X$  and the cubic kernel, for  $x \in \Lambda$  the classical interpolant  $P_f : \Lambda \rightarrow \mathbb{R}$  assumes the form

$$P_f(x) = \sum_{k=1}^n \alpha_k \kappa_{1,2}(x, x_k) + \sum_{j=1}^2 \beta_j p_j(x) = \sum_{k=1}^n \alpha_k |x - x_k|^3 + \sum_{j=1}^2 \beta_j p_j(x),$$

where  $p_1$  and  $p_2$  are a basis for  $\Pi_1^1$ . Following Proposition 3.1, from (7), for



$x \in \Lambda$  we have that the VSK interpolant  $P_f^\Psi : \Lambda \rightarrow \mathbb{R}$  is

$$\begin{aligned} P_f^\Psi(x) &= P_f((x, \psi(x))) = \sum_{k=1}^n \alpha_k \kappa_{2,2}((x, \psi(x)), (x_k, \psi(x))) + \sum_{j=1}^3 \beta_j p_j((x, \psi(x))) = \\ &= \sum_{k=1}^n \alpha_k \|(x, \psi(x)) - (x_k, \psi(x_k))\|_2^2 \log(\|(x, \psi(x)) - (x_k, \psi(x_k))\|_2) + \\ &+ \sum_{j=1}^3 \beta_j p_j((x, \psi(x))), \end{aligned}$$

where  $p_1, p_2, p_3$ , are a basis for  $\Pi_1^2$ . This is the standard setting for the TPS computed on scattered data and thanks to [26, p. 176] the thesis follows; refer also to [27].  $\blacksquare$

Even if we provided error bounds for the interpolation setting, we have to point out that the interpolation conditions might be relaxed in some cases, for instance when the measurements are affected by noise and errors or when the main issue is to capture the trend of data. To accomplish this, we use a method that is generally referred to as ridge regression; see e.g. [12, 16]. Precisely, to smooth out the noise for both standard and VSK interpolants, we introduce a penalty term weighted by a non-negative regularization parameter  $\lambda \in \mathbb{R}^+$  and, starting from the system (2), we compute the coefficients by solving:

$$(\mathbf{K} + \lambda \mathbf{I})\boldsymbol{\gamma} = \mathbf{g}, \quad (8)$$

where  $\mathbf{I} \in \mathbb{R}^{(n+m) \times (n+m)}$  is the identity matrix. Note that in the VSK setting the matrix  $\mathbf{K}$  is defined by  $\mathbf{A}$  and  $\mathbf{P}$  both set as in Proposition 3.1.

We conclude this section by observing that the procedure for constructing a VSK approximant requires the definition of the scaling function. In the following, we thus derive this setting from assumptions on data.

## 4 Practical VSK setting

Here we point out the empirical framework that we will use for the numerical tests. Precisely, we first discuss the selection of the scaling function and then we also report some basics of SVR, introduced for comparisons.

## 4.1 The scaling function

As in the previous section we set  $d = 1$ , and we assume that the model is defined on a set of samples that decay with an exponential or rational trend. The choice is justified by the interest towards applications in which the acquired data show this trend, as already pointed out. Given  $X = \{x_1, \dots, x_n\} \subseteq \Omega$ ,  $\Omega \subseteq \mathbb{R}$ , and  $F = \{f_1, \dots, f_n\} \subseteq \mathbb{R}$ , we consider two classes of rational and exponential parametric real functions:

$$\mathcal{R} = \left\{ g : \Lambda \times \mathbb{R}^3 \longrightarrow \mathbb{R} : g(x, \boldsymbol{\mu}) = \frac{x^{-\mu_1}}{x^{\mu_2} + \mu_3} \right\}, \quad (9)$$

$$\mathcal{E} = \{g : \Lambda \times \mathbb{R}^3 \longrightarrow \mathbb{R} : g(x, \boldsymbol{\mu}) = \mu_1 x e^{-\mu_2 x} + \mu_3 e^{-\mu_2 x}\}, \quad (10)$$

where  $\boldsymbol{\mu} = (\mu_1, \mu_2, \mu_3)^\top$ .

The estimation of the parameters  $\mu_1, \mu_2, \mu_3$  is carried out with an iterative reweighted least squares algorithm for non-linear fit [28, 29] of samples  $(x_i, f_i)_{i=1, \dots, n}$ . At each iteration, the robust weights downweight outliers, so that their influence becomes neglectable. Computationally speaking we use the MATLAB<sup>®</sup> function `nlinfit.m`. Once the *optimal* parameters  $\mu_i^*$ ,  $i = 1, \dots, 3$ , are estimated from the samples for both classes  $\mathcal{R}$  and  $\mathcal{E}$  we define  $\psi_1(x) = g(x, \boldsymbol{\mu}^*)$  for  $g \in \mathcal{R}$  and  $\psi_2(x) = g(x, \boldsymbol{\mu}^*)$  for  $g \in \mathcal{E}$ . Then, letting  $\boldsymbol{\psi}_j = (\psi_j(x_1), \dots, \psi_j(x_n))^\top$ ,  $j = 1, 2$ , we set  $\boldsymbol{\psi} \equiv \boldsymbol{\psi}_s$ , where

$$s = \operatorname{argmin}_{j=1,2} \|\boldsymbol{f} - \boldsymbol{\psi}_j\|_2.$$

Finally we define the VSK approximant on the set  $\{(x_1, \psi(x_1)), \dots, (x_n, \psi(x_n))\}$  by fixing  $l = 2$  in (5), i.e. via the TPS.

In the numerical experiments we compare our model with SVR that we briefly describe below for clarity and for making the paper self-contained.

## 4.2 Support Vector Regression

Given  $X = \{x_1, \dots, x_n\} \subseteq \Omega$ ,  $\Omega \subseteq \mathbb{R}$ , and  $F = \{f_1, \dots, f_n\} \subseteq \mathbb{R}$ , the SVR model  $P_f : \Lambda \longrightarrow \mathbb{R}$  is constructed via a minimization problem. Precisely, it reduces to solving [16]

$$\min_{\boldsymbol{\alpha}, \boldsymbol{\alpha}^* \in \mathbb{R}^n} \left[ \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n (\alpha_i - \alpha_i^*)(\alpha_j^* - \alpha_j) \kappa(x_i, x_j) + \epsilon \sum_{i=1}^n (\alpha_i^* + \alpha_i) - \sum_{i=1}^n f_i (\alpha_i^* - \alpha_i) \right],$$

subject to:

$$0 \leq \alpha_i, \alpha_i^* \leq \zeta, \quad i = 1, \dots, n,$$

$$\sum_{i=1}^n (\alpha_i^* - \alpha_i) = 0.$$

where  $\zeta \geq 0$  represents the so-called *trade-off parameter* and it is indeed a smoothing parameter. The *hyper-parameter*  $\epsilon \geq 0$  indicates the width of the *tube* in which the samples can fall into without being counted as errors. From the Karush Kuhn Tucker conditions (see e.g. [30]), we have [9]:

$$P_f(\mathbf{x}) = \sum_{i=1}^n (\alpha_i^* - \alpha_i) \kappa(x, x_i) + b,$$

where

$$b = \begin{cases} f_i - \sum_{j=1}^n (\alpha_j^* - \alpha_j) \kappa(x_i, x_j) - \epsilon, & \text{for } \alpha_i \in (0, \zeta), \\ f_i - \sum_{j=1}^n (\alpha_j^* - \alpha_j) \kappa(x_i, x_j) + \epsilon, & \text{for } \alpha_i^* \in (0, \zeta), \end{cases}$$

is defined via an average over all candidates.

We now have all the ingredients for testing in the next section the proposed technique.

## 5 Numerical experiments

Based on the assumed data trend, we present results on data sets generated by rational/exponential functions. Particularly, in accordance with previous studies [13,31], we consider the following test functions taken from a database of Laplace transforms [32]:

$$f_1(x) = \frac{1}{x(x+1)^2}, \quad f_2(x) = \frac{1}{x+1},$$

$$f_3(x) = \frac{x}{(x^2+1)^2}, \quad f_4(x) = e^{-2x},$$

$$f_5(x) = \arctan \frac{20}{x}, \quad f_6(x) = \frac{x}{x^2+1}.$$

and we sample them on  $\Omega = [a, b]$ , set  $a = 0.1$  and  $b = 2$ , with 30 nodes which follow four different distributions, precisely quasi-uniform Halton points,

Chebyshev nodes, random and uniform points. For the extrapolation issue, we need to define the interval  $\Lambda = [\Lambda_1, \Lambda_2]$ . While  $\Lambda_1 = a$ ,  $\Lambda_2 = b + 0.1i$ ,  $i = 0, \dots, 10$ . In this way we can numerically verify the robustness of our method as  $\Lambda_2$  increases. Of course, dealing with *asymptotic* extrapolation, we expect that to larger  $\Lambda_2$  correspond larger errors. Specifically, we evaluate the Root Mean Square Error (RMSE) on  $s = 40$  equispaced nodes  $\bar{x}_i$ ,  $i = 1, \dots, s$ , on  $\Lambda$ , i.e. we compute

$$\text{RMSE} = \left( \frac{\sum_{i=1}^s (f(\bar{x}_i) - A(\bar{x}_i))^2}{s} \right)^{1/2},$$

where  $A$  is the approximant constructed via either  $P_f : \Lambda \rightarrow \mathbb{R}$  or  $P_f^\Psi : \Lambda \rightarrow \mathbb{R}$ . Moreover, the regularization parameter in (8) is set as  $\lambda = 1e-06$ .

Tests have been carried out on a Intel(R) Core(TM) i7 CPU 4712MQ 2.13 GHz processor. The RMSEs obtained by considering both the standard approximation via the cubic kernel and the TPS-VSKs are reported in Tables 1–4 for the test functions  $f_i$ ,  $i = 1, \dots, 4$ , respectively. The augmented feature for the VSK setting is dynamically selected by fitting  $f_1$  and  $f_2$  with a *rational decay*, while  $f_3$  and  $f_4$  are modelled by an *exponential decay*. The reader should note that closer is the fitting of  $\psi$  to the behaviour of the true function  $f$ , smaller is the RMSE. Indeed, the method almost reaches the machine precision when the test function  $f$  belongs to one of the classes  $\mathcal{R}$  or  $\mathcal{E}$ , defined in (9) and (10), respectively. This holds true for  $f_2$  and  $f_4$ .

Moreover, we note that both methods behave similarly for each of the selected node distributions. This is more in general a peculiarity of kernel methods that are robust for different data distributions. However, in our experiments, the TPS-VSK method usually outperforms the standard cubic extrapolation. For a graphical feedback on the absolute errors refer to Figure 1.

As last example, we take the functions  $f_5$  and  $f_6$  and we compare our varying scale setting with SVR. In this experiments, we also introduce noise on the measurements, i.e.  $f_i = f(x_i) + \delta_i$ ,  $i = 1, \dots, n$ . Precisely, we assume Gaussian white noise, i.e.  $\boldsymbol{\delta} = (\delta_1, \dots, \delta_n)^\top \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$ , where  $\mathbf{I}$  is the  $n \times n$  identity matrix and we fix  $\sigma = 1e-04$ . We compare the VSK setting with the SVR trained with the cubic kernel and a standard 3-fold validation for optimizing the hyperparameters (for this scope we use the MATLAB<sup>®</sup> function `fitrsvm.m`). Since we do not use any data-filling strategy for SVR,

Table 1: The RMSE for different values of  $\Lambda_2$  and different node distributions obtained via the cubic kernel and the TPS-VSKs for  $f_1$ .

	$\Lambda_2$	Halton	Chebyshev	Random	Uniform
Cubic	2.00	1.64e-02	3.94e-03	1.16e-02	2.69e-02
	2.10	1.24e-02	3.93e-03	8.71e-03	2.30e-02
	2.20	8.31e-03	3.94e-03	6.11e-03	1.88e-02
	2.30	4.53e-03	4.00e-03	3.95e-03	1.46e-02
	2.40	2.00e-03	4.19e-03	2.60e-03	1.04e-02
	2.50	3.45e-03	4.57e-03	2.69e-03	6.62e-03
	2.60	5.97e-03	5.20e-03	3.75e-03	4.31e-03
	2.70	8.34e-03	6.09e-03	5.11e-03	5.08e-03
	2.80	1.05e-02	7.23e-03	6.61e-03	7.53e-03
	2.90	1.25e-02	8.58e-03	8.23e-03	1.02e-02
3.00	1.44e-02	1.01e-02	9.97e-03	1.28e-02	
TPS-VSK	2.00	1.70e-03	1.40e-03	1.39e-03	2.88e-03
	2.10	1.21e-03	1.40e-03	1.02e-03	2.32e-03
	2.20	1.02e-03	1.48e-03	7.41e-04	1.91e-03
	2.30	1.45e-03	1.70e-03	6.25e-04	1.95e-03
	2.40	2.31e-03	2.13e-03	7.06e-04	2.57e-03
	2.50	3.39e-03	2.77e-03	8.94e-04	3.61e-03
	2.60	4.65e-03	3.59e-03	1.11e-03	4.93e-03
	2.70	6.06e-03	4.55e-03	1.34e-03	6.43e-03
	2.80	7.62e-03	5.65e-03	1.58e-03	8.10e-03
	2.90	9.31e-03	6.87e-03	1.83e-03	9.91e-03
3.00	1.11e-02	8.20e-03	2.07e-03	1.19e-02	

Table 2: The RMSE for different values of  $\Lambda_2$  and different node distributions obtained via the cubic kernel and the TPS-VSKs for  $f_2$ .

	$\Lambda_2$	Halton	Chebyshev	Random	Uniform
Cubic	2.00	1.85e-05	3.56e-06	1.31e-05	2.89e-05
	2.10	1.01e-04	6.67e-05	8.52e-05	8.63e-05
	2.20	3.55e-04	2.74e-04	3.18e-04	3.13e-04
	2.30	7.87e-04	6.50e-04	7.24e-04	7.16e-04
	2.40	1.41e-03	1.21e-03	1.31e-03	1.30e-03
	2.50	2.22e-03	1.95e-03	2.09e-03	2.08e-03
	2.60	3.22e-03	2.87e-03	3.06e-03	3.04e-03
	2.70	4.40e-03	3.98e-03	4.20e-03	4.18e-03
	2.80	5.76e-03	5.26e-03	5.53e-03	5.50e-03
	2.90	7.30e-03	6.71e-03	7.03e-03	7.00e-03
	3.00	9.01e-03	8.34e-03	8.70e-03	8.66e-03
TPS-VSK	2.00	3.14e-16	2.10e-16	3.03e-16	3.34e-16
	2.10	3.08e-16	4.26e-16	2.83e-16	3.17e-16
	2.20	2.91e-16	1.25e-15	2.75e-16	3.08e-16
	2.30	2.95e-16	2.49e-15	3.00e-16	3.05e-16
	2.40	2.88e-16	4.12e-15	2.70e-16	3.16e-16
	2.50	2.81e-16	6.12e-15	2.79e-16	3.00e-16
	2.60	2.76e-16	8.43e-15	2.55e-16	3.05e-16
	2.70	2.71e-16	1.11e-14	2.59e-16	3.18e-16
	2.80	2.82e-16	1.39e-14	2.59e-16	3.27e-16
	2.90	2.66e-16	1.71e-14	2.87e-16	3.37e-16
	3.00	2.89e-16	2.05e-14	2.69e-16	3.32e-16

Table 3: The RMSE for different values of  $\Lambda_2$  and different node distributions obtained via the cubic kernel and the TPS-VSKs for  $f_3$ .

	$\Lambda_2$	Halton	Chebyshev	Random	Uniform
Cubic	2.00	1.76e-05	3.05e-06	2.16e-05	2.24e-05
	2.10	1.56e-04	1.02e-04	1.32e-04	1.29e-04
	2.20	5.43e-04	4.17e-04	4.85e-04	4.78e-04
	2.30	1.19e-03	9.78e-04	1.09e-03	1.08e-03
	2.40	2.11e-03	1.79e-03	1.96e-03	1.94e-03
	2.50	3.29e-03	2.86e-03	3.09e-03	3.07e-03
	2.60	4.72e-03	4.18e-03	4.47e-03	4.44e-03
	2.70	6.39e-03	5.73e-03	6.08e-03	6.05e-03
	2.80	8.29e-03	7.50e-03	7.92e-03	7.88e-03
	2.90	1.04e-02	9.48e-03	9.97e-03	9.92e-03
	3.00	1.27e-02	1.16e-02	1.22e-02	1.22e-02
TPS-VSK	2.00	4.85e-05	1.38e-05	1.09e-04	8.30e-05
	2.10	7.44e-05	5.39e-05	1.16e-04	9.25e-05
	2.20	1.88e-04	1.52e-04	1.88e-04	1.87e-04
	2.30	3.47e-04	2.80e-04	3.16e-04	3.37e-04
	2.40	5.33e-04	4.22e-04	4.71e-04	5.18e-04
	2.50	7.35e-04	5.66e-04	6.39e-04	7.16e-04
	2.60	9.45e-04	7.04e-04	8.13e-04	9.21e-04
	2.70	1.16e-03	8.31e-04	9.85e-04	1.13e-03
	2.80	1.37e-03	9.42e-04	1.15e-03	1.34e-03
	2.90	1.58e-03	1.04e-03	1.31e-03	1.54e-03
	3.00	1.79e-03	1.12e-03	1.47e-03	1.74e-03

Table 4: The RMSE for different values of  $\Lambda_2$  and different node distributions obtained via the cubic kernel and the TPS-VSKs for  $f_4$ .

	$\Lambda_2$	Halton	Chebyshev	Random	Uniform
Cubic	2.00	3.75e-05	7.79e-06	2.90e-05	6.31e-05
	2.10	1.01e-04	6.45e-05	8.49e-05	9.64e-05
	2.20	3.37e-04	2.57e-04	3.00e-04	2.98e-04
	2.30	7.29e-04	5.93e-04	6.66e-04	6.59e-04
	2.40	1.27e-03	1.07e-03	1.18e-03	1.17e-03
	2.50	1.96e-03	1.69e-03	1.84e-03	1.82e-03
	2.60	2.78e-03	2.44e-03	2.62e-03	2.60e-03
	2.70	3.72e-03	3.30e-03	3.53e-03	3.51e-03
	2.80	4.78e-03	4.28e-03	4.55e-03	4.52e-03
	2.90	5.93e-03	5.35e-03	5.67e-03	5.63e-03
	3.00	7.18e-03	6.51e-03	6.87e-03	6.84e-03
TPS-VSK	2.00	5.64e-16	2.03e-16	8.27e-16	6.69e-16
	2.10	4.32e-15	2.68e-15	4.86e-15	3.67e-15
	2.20	1.28e-14	8.38e-15	1.46e-14	1.11e-14
	2.30	2.50e-14	1.67e-14	2.87e-14	2.19e-14
	2.40	4.02e-14	2.73e-14	4.66e-14	3.55e-14
	2.50	5.82e-14	3.97e-14	6.76e-14	5.15e-14
	2.60	7.85e-14	5.38e-14	9.14e-14	6.97e-14
	2.70	1.01e-13	6.93e-14	1.18e-13	8.96e-14
	2.80	1.25e-13	8.60e-14	1.46e-13	1.11e-13
	2.90	1.50e-13	1.04e-13	1.75e-13	1.34e-13
	3.00	1.76e-13	1.22e-13	2.07e-13	1.58e-13





Figure 1: Left to right, top to bottom: the absolute errors in logarithmic scale, for  $\Lambda_2 = 3$  and Halton node distributions obtained via the cubic kernel (' $\square$ ') and the TPS-VSKs (' $\triangle$ ') for  $f_1$ ,  $f_2$ ,  $f_3$  and  $f_4$ .

Table 5: The RMSE for different values of  $\Lambda_2$  and uniform node distribution obtained via SVR and the TPS-VSKs for  $f_5$  and  $f_6$ .

$\Lambda_2$	$f_5$		$f_6$	
	SVR	TPS-VSK	SVR	TPS-VSK
2.00	2.32e-05	1.29e-04	3.79e-03	1.29e-04
2.10	2.35e-05	1.32e-04	6.09e-03	1.31e-04
2.20	2.54e-05	1.46e-04	1.09e-02	1.38e-04
2.30	2.93e-05	1.73e-04	1.84e-02	1.49e-04
2.40	3.56e-05	2.14e-04	2.88e-02	1.67e-04
2.50	4.42e-05	2.64e-04	4.24e-02	1.83e-04
2.60	5.50e-05	3.26e-04	5.97e-02	2.04e-04
2.70	6.78e-05	3.98e-04	8.11e-02	2.26e-04
2.80	8.26e-05	4.76e-04	1.07e-01	2.44e-04
2.90	9.94e-05	5.68e-04	1.38e-01	2.68e-04
3.00	1.18e-04	6.67e-04	1.74e-01	2.88e-04

we only take into account equispaced data. The results are reported in Table 5. For a graphical feedback, refer to Figure 2. We note that, for the function  $f_5$ , which is approximately *linear* on  $\Lambda$ , our results are comparable with SVR; on the opposite, when *learning* the function is not trivial, e.g. for  $f_6$ , our model outperforms the standard SVR. However, we have to point out that the SVR performances could be improved by data assimilation procedures, i.e. one could construct a SVR model for each values of  $\Lambda_2$ . Nevertheless, this procedure would be too expensive if compared to the VSK strategy that only requires the computation of an additional feature.

## 6 Conclusion and work in progress

We investigated a novel procedure for the extrapolation issue based on the use of VSKs that serve as feature augmentation strategy. After extending them to strictly conditionally positive definite kernels and providing error bounds for the TPS in Beppo-Levi spaces, we tested the tool on several models. The results and in particular the comparison with SVR stress the benefits coming from the use of VSKs for the extrapolation issue.

Future work consists in extending the VSK setting to the context of SVR.

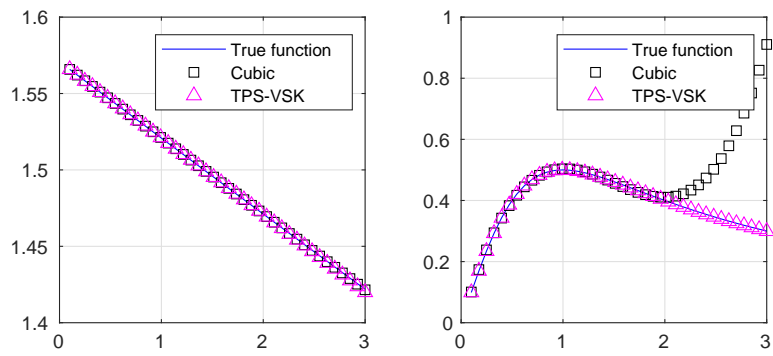


Figure 2: The graphical results for  $\Lambda_2 = 3$  and equispaced node distributions obtained via the cubic kernel (' $\square$ ') and the TPS-VSKs (' $\triangle$ ') for  $f_5$  and  $f_6$ , left and right respectively.

The use of a scaling function indeed would introduce *new feature maps* and *spaces* which need further investigations.

## References

- [1] Nikolaos P. Bakas. Numerical solution for the extrapolation problem of analytic functions. *Research*, 2019, 2019.
- [2] Emmanuel J Candès and Carlos Fernandez-Granda. Towards a mathematical theory of super-resolution. *Communications on pure and applied Mathematics*, 67(6):906–956, 2014.
- [3] Laurent Demanet and Alex Townsend. Stable extrapolation of analytic functions. *CoRR*, abs/1605.09601, 2016.
- [4] S. Shetty and P.R. White. Curvature-continuous extensions for rational B-spline curves and surfaces. *Computer-Aided Design*, 23(7):484 – 491, 1991.
- [5] G. E. Fasshauer. *Meshfree Approximation Methods with MATLAB*. Interdisciplinary mathematical sciences. World Scientific, Singapore, 2007.
- [6] M. Bozzini, L. Lenarduzzi, M. Rossini, and R. Schaback. Interpolation with variably scaled kernels. *IMA Journal of Numerical Analysis*, 35(1):199–219, 2015.
- [7] R. Campagna, S. Cuomo, S. De Marchi, E. Perracchione, and G. Severino. A stable meshfree pde solver for source-type flows in porous media. *Applied Numerical Mathematics*, 149:30–42, 2020.
- [8] W. Li, L. Duan, D. Xu, and I. W. Tsang. Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(6):1134–1148, 2014.
- [9] B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, USA, December 2002.

- [10] S. De Marchi, W. Erb, F. Marchetti, E. Perracchione, and M. Rossini. Shape-driven interpolation with discontinuous kernels: Error analysis, edge extraction, and applications in magnetic particle imaging. *SIAM Journal on Scientific Computing*, 42(2):B472–B491, 2020.
- [11] S. De Marchi, F. Marchetti, and E. Perracchione. Jumping with variably scaled discontinuous kernels (VSDKs). *BIT Numerical Mathematics*, 60:441–463, 2020.
- [12] G. Wahba. *Spline Models for Observational Data*. Society for Industrial and Applied Mathematics, Philadelphia, 1990.
- [13] R. Campagna, C. Conti, and S. Cuomo. Smoothing exponential-polynomial splines for multiexponential decay data. *Dolomites Research Notes on Approximation*, 12:86–100, 2019.
- [14] L. D’Amore, R. Campagna, A. Galletti, L. Marcellino, and A. Murli. A smoothing spline that approximates Laplace transform functions only known on measurements on the real axis. *Inverse Problems*, 28(2):025007, jan 2012.
- [15] Annalisa Romano, Rosanna Campagna, Paolo Masi, and Gerardo Toraldo. Nmr data analysis of water mobility in wheat flour dough: A computational approach. In Yaroslav D. Sergeyev and Dmitri E. Kvasov, editors, *Numerical Computations: Theory and Algorithms*, pages 146–157, Cham, 2020. Springer International Publishing.
- [16] G. E. Fasshauer and M. McCourt. *Kernel-based Approximation Methods using MATLAB*. World scientific, Singapore, 2015.
- [17] H. Wendland. *Scattered Data Approximation*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2004.
- [18] R. Schaback. Error estimates and condition numbers for radial basis function interpolation. *Advances in Computational Mathematics*, 3:251–264, Apr 1995.
- [19] L. Romani, M. Rossini, and D. Schenone. Edge detection methods based on RBF interpolation. *Journal of Computational and Applied Mathematics*, 349:532 – 547, 2019.

- [20] M. Rossini. Interpolating functions with gradient discontinuities via variable scaled kernels. *Dolomites Research Notes on Approximation*, 11:3–14, 2018.
- [21] Jean Duchon. Interpolation des fonctions de deux variables suivant le principe de la flexion des plaques minces. *ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique*, 10(R3):5–12, 1976.
- [22] R.L. Harder and R.N. Desmarais. Interpolation using surface splines. *Journal of Aircraft*, 9(2):189–191, 1972.
- [23] Armin Iske. On the approximation order and numerical stability of local lagrange interpolation by polyharmonic splines. In Werner Haussmann, Kurt Jetter, Manfred Reimer, and Joachim Stöckler, editors, *Modern Developments in Multivariate Approximation*, pages 153–165, Basel, 2003. Birkhäuser Basel.
- [24] John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, illustrated edition edition, 2004.
- [25] J. Deny and J.L. Lions. Les espaces du type de Beppo Levi. *Ann. Inst. Fourier, Grenoble*, 5:302–370, 1954.
- [26] R.K. Beatson, H. Q. Bui, and J. Levesley. Embeddings of Beppo-Levi spaces in Hölder-Zygmund spaces, and a new method for radial basis function interpolation error estimates. *Journal of Approximation Theory*, 137(2):166–178, 2005.
- [27] M.J.D. Powell. The uniform convergence of thin plate spline interpolation in two dimensions. *Numerische Mathematik*, 68(1):107–128, 1994.
- [28] P. W. Holland and R. E. Welsch. Robust regression using iteratively reweighted least-squares. *Communications in Statistics - Theory and Methods*, 6(9):813–827, 1977.
- [29] G.A.F. Seber and C.J. Wild. *Nonlinear Regression*. Wiley-Interscience, Hoboken, NJ, USA, 2003.
- [30] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, New York, NY, USA, second edition, 2006.

- [31] Rosanna Campagna, Costanza Conti, and Salvatore Cuomo. A procedure for Laplace transform inversion based on smoothing exponential-polynomial splines. In Yaroslav D. Sergeyev and Dmitri E. Kvasov, editors, *Numerical Computations: Theory and Algorithms*, pages 11–18, Cham, 2020. Springer International Publishing.
- [32] Rosanna Campagna, Costanza Conti, and Salvatore Cuomo. Computational error bounds for Laplace transform inversion based on smoothing splines. *Applied Mathematics and Computation*, 383:125376, 2020.