

Stable interpolation with exponential-polynomial splines and node selection via greedy algorithms

*Original*

Stable interpolation with exponential-polynomial splines and node selection via greedy algorithms / Campagna, R.; De Marchi, S.; Perracchione, E.; Santin, G.. - In: ADVANCES IN COMPUTATIONAL MATHEMATICS. - ISSN 1019-7168. - ELETTRONICO. - 48:6(2022). [10.1007/s10444-022-09986-8]

*Availability:*

This version is available at: 11583/2973116 since: 2023-06-04T10:41:49Z

*Publisher:*

SPRINGER

*Published*

DOI:10.1007/s10444-022-09986-8

*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)



# Stable interpolation with exponential-polynomial splines and node selection via greedy algorithms

R. Campagna<sup>1</sup> · S. De Marchi<sup>2</sup> · E. Perracchione<sup>3</sup>  · G. Santin<sup>4</sup>

Received: 28 September 2021 / Accepted: 26 September 2022  
© The Author(s) 2022

## Abstract

In this work we extend some ideas about greedy algorithms, which are well-established tools for, e.g., kernel bases, and exponential-polynomial splines whose main drawback consists in possible *overfitting* and consequent oscillations of the approximant. To partially overcome this issue, we develop some results on theoretically optimal interpolation points. Moreover, we introduce two algorithms which perform an adaptive selection of the spline interpolation points based on the minimization either of the sample residuals ( $f$ -greedy), or of an upper bound for the approximation error based on the spline Lebesgue function ( $\lambda$ -greedy). Both methods allow us to obtain an adaptive selection of the sampling points, i.e., the spline nodes. While the  $f$ -greedy selection is tailored to one specific target function, the  $\lambda$ -greedy algorithm enables us to define target-data-independent interpolation nodes.

**Keywords** Greedy methods · Lebesgue function · Exponential-polynomial splines · Node selection

---

Communicated by Anthony Patera.

✉ E. Perracchione  
emma.perracchione@polito.it

R. Campagna  
rosanna.campagna@unicampania.it

S. De Marchi  
stefano.demarchi@unipd.it

G. Santin  
gsantin@fbk.eu

<sup>1</sup> Department of Mathematics and Physics, University of Campania “L. Vanvitelli”, Caserta, Italy

<sup>2</sup> Department of Mathematics “Tullio Levi-Civita”, University of Padova, Padua, Italy

<sup>3</sup> Dipartimento di Scienze Matematiche “Giuseppe Luigi Lagrange”, Politecnico di Torino, Corso Duca degli Abruzzi, 24, 10129 Torino, Italy

<sup>4</sup> Digital Society Center (DIGIS), Bruno Kessler Foundation, Trento, Italy

**Mathematics Subject Classification** 65D15 · 41A05

## 1 Introduction

Scattered data interpolation is one of the most investigated topics in the field of numerical analysis, and it is successfully used in many applications. As a consequence, many methods have been developed, including interpolation with polynomials of total degree (see, e.g., [1]), exponential splines [2], and kernel-based methods (refer, e.g., to [3, 4]), with their recent developments in the context of image processing (e.g., [5]) and machine learning [6, 7].

More recently, the so-called Exponential-Polynomial Splines (EPS) have been introduced with the main purpose of studying approximations in spaces which generalize classical polynomial splines. The latter find many applications, ranging from geometric modelling to image analysis, passing through isogeometric analysis and system theory (see, for example, [8–10]). In other words they are able to deal with more complex tasks, in that they reproduce functions which are linear combinations of exponentials, and this is quite a common setting in the applications. For instance, several papers deal with the approximation of univariate multi-exponentially decaying functions with a smoothing effect [11–13]. Such a smoothing strategy is implemented by considering a regularization parameter.

To understand the EPS-based process, we study pointwise interpolation error bounds. We are able to obtain these bounds thanks to the definition of the cardinal form of the EPS interpolant that then allows us to introduce the Lebesgue function and constant [14, 15]. The latter are known to be stability indicators for polynomial bases; see, e.g., [1, 16–20]. In this paper we are interested in the design of appropriate sampling strategies for EPS interpolation. Namely, assuming to be given either only an input space discretization, or a dataset of input points and corresponding function evaluations, we aim at selecting a small subset of approximation points to be used to construct the EPS interpolant. We first address the problem from a theoretical point of view, and show that the Lebesgue constant associated to equally spaced sampling points is uniformly bounded by a (small) constant. This fact indicates that these points are quasi-optimal, given the well known relation between interpolation and best approximation. However, since these uniform sampling locations may not be available in practical applications, we further consider incremental methods that, given an initial set of samples, construct an EPS interpolant by iteratively selecting a new point at each iteration. The iterative rule is dictated by greedy methods (see [21]), which have been investigated, e.g., for kernel interpolation (refer, e.g., to [22–27]) and lead to sparse models which turn out to be helpful in many applications, see, e.g., [28]. This iterative selection is a convenient proxy for the optimal selection of the sampling points from a fixed set, which is in turn usually an extremely computationally demanding procedure.

We first consider the greedy method associated to the iterative minimization of the Lebesgue constant. Based on this error indicator, we define an algorithm for selecting data-independent points for EPS. Then, we propose a second extraction strategy

that takes into account also the function values. This kind of approach is usually more expensive, but it allows to select points that are tailored to one specific target function, and thus are usually able to better resolve local features such as steep gradients or oscillations. In both cases, we numerically explore the behavior of the node distribution for the spline basis, and we test our findings under different perspectives.

The paper is organized as follows. In Section 2 we briefly review the basics of greedy methods and EPS interpolation, and in Section 3 we recall the definition of the associated Lebesgue constant. In particular, we use it to show that equally spaced points are quasi-optimal since the corresponding Lebesgue constant is uniformly bounded independently of the number of interpolation points. Motivated by the fact that this kind of points may not be available in practical applications, we introduce two greedy selection strategies in Section 4. The numerical experiments are presented and discussed in Section 5, while Section 6 deals with an application to real data from a Nuclear Magnetic Resonance (NMR) experiment. Conclusions with an outline of future works are provided in Section 7.

## 2 Exponential-polynomial splines and greedy schemes

In this section we present the main features of EPS and provide some generalities on greedy methods.

### 2.1 Exponential-polynomial splines

The EPS are a particular spline model that has been introduced in [12], and which are a particular instantiation of a general technique described in [29]. This *exponential natural smoothing L-spline* is the solution of the minimization of a cost functional defined in [12], which comprises a weighted least square loss and a penalization term depending on a suitable differential operator. It can be proven that a unique solution of this optimization problem exists, and numerical evidence suggests that this model is well-suited for the approximation of a certain class of functions, as outlined in the Introduction (Section 1).

In this paper we focus on the non-regularized and unweighted version of this model, which can be formulated explicitly via a simple interpolation problem. We refer to [12, 29] for a definition of the most general version of the spline model, and we directly use this simpler approach in the following.

Namely, we consider a real valued continuous function  $f \in C([a, b]) := C([a, b], \mathbb{R})$ ,  $[a, b] \subset \mathbb{R}$ , and an associated set of function values  $F := \{y_i := f(x_i)\}_{i=1}^n$  sampled on an input data set  $X := \{x_i\}_{i=1}^n$ , that constitutes a partition of  $[a, b]$ , i.e.,  $a = x_1 < x_2 < \dots < x_n = b$ . For a given parameter  $\alpha \in \mathbb{R}$ , we define the interpolant of  $f$  as

$$I_{X,\alpha}(f)(x) = \sum_{i=1}^n c_i \varphi_i(x), \quad (1)$$

where  $\{\varphi_j\}_{j=1}^n$  is a basis of *exponential B-splines*, also referred to as Generalized B-splines (GB-splines), and the coefficients  $\mathbf{c} := [c_1, \dots, c_n]^T \in \mathbb{R}^n$  are obtained by imposing the interpolation conditions  $I_{X,\alpha}(f)(x_i) = f(x_i)$ ,  $1 \leq i \leq n$  and  $C^2$  regularity at knots. These coefficients exist and are unique because the corresponding interpolation matrix is invertible, as discussed below.

Since the interpolant of any continuous function is now well defined, we denote here and in the following by  $I_{X,\alpha} : C([a, b]) \rightarrow C([a, b])$  the linear interpolation operator, so that  $I_{X,\alpha}f : \mathbb{R} \rightarrow \mathbb{R}$  is the function interpolating  $f \in C([a, b])$  at  $X$ .

The GS-splines are defined so that  $\varphi_j|_{[x_i, x_{i+1}]} \in \mathbb{E}_{4,\alpha}$ , where  $|_A$  denotes the restriction on a set  $A \subset \mathbb{R}$  and

$$\mathbb{E}_{4,\alpha} := \text{span}\{e^{\alpha x}, xe^{\alpha x}, e^{-\alpha x}, xe^{-\alpha x}\},$$

and they have the following properties (see Fig. 1): they are bell-shaped with compact support, identified by 5 nodes, with the blending segments belonging to  $\mathbb{E}_{4,\alpha}$ , and have global  $C^2$ -smoothness. The generic basis function  $\varphi$  can then be expressed as

$$\varphi(x)|_{[x_i, x_{i+1}]} = \sum_{k=1}^4 b_{i,k} g_k(x), \quad g_k \in \mathbb{E}_{4,\alpha}, \tag{2}$$

where  $x_i$ ,  $i = 1, \dots, n - 1$ , denotes the left point of the partition element, and  $k = 1, \dots, 4$ , denotes the index of the local basis element. Indeed, each function  $\varphi$  in the form (2) has  $4(n - 1)$  degrees of freedom, given by the coefficients  $\{b_{i,k}\}$  with  $1 \leq i \leq n - 1$ ,  $1 \leq k \leq 4$ . Then, to define such a basis of dimension  $n$ , the nodes vector has to be augmented with two extra nodes before  $x_1$  and two others after  $x_n$ , i.e., an augmented node set as  $x_{-1} \leq x_0 \leq x_1 = a < \dots < x_n = b \leq x_{n+1} \leq x_{n+2}$  has to be considered. Such extra nodes affect the construction of the so-called *boundary* basis functions:  $\varphi_1, \varphi_2$  and  $\varphi_{n-1}, \varphi_n$ .

The global space of the exponential-polynomial splines  $\mathcal{E}_{X,\alpha}(\mathbb{E}_{4,\alpha})$  is then defined by gluing local patches defined over each interval such that  $\mathcal{E}_{X,\alpha}(\mathbb{E}_{4,\alpha}) \subset C^2([a, b])$ , and any element in this space can be expressed using coefficients  $b_{i,k}$  as in (2). In particular, this implies via (1) that  $I_{X,\alpha} \in C^2([a, b])$ .

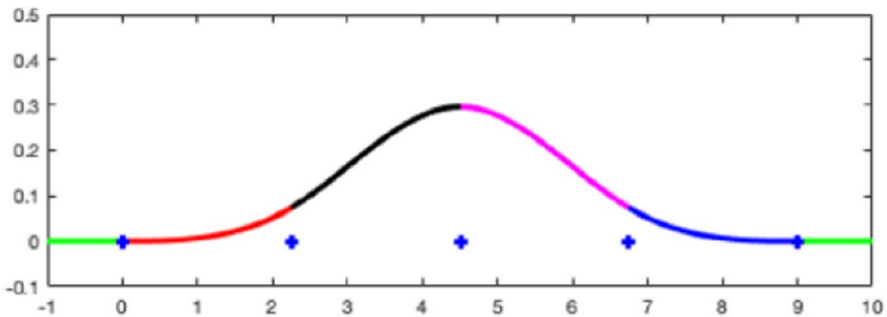


Fig. 1 An example of a GB-spline with segments in the spaces  $\mathbb{E}_{4,\alpha}$

**Inputs:**  $X \subseteq \Omega, \mathbf{y}, \alpha$ .

**Outputs:** The interpolant coefficients  $\mathbf{c} := [c_1, \dots, c_n]^T$ .

- 1: Define the function space  $\mathbb{E}_{4,\alpha}$ .
- 2: Define the augmented nodes for the boundary basis functions.
- 3: Compute the GB-spline basis functions  $\{\varphi_j\}_{j=1}^n$  in Bernstein-like basis.
- 4: Compute the collocation matrix and solve the linear system as in (4).

**Algorithm 1** Pseudo-code for EPS

**Remark 1** We remark that in [12] the authors assume that  $\alpha > 0$  (instead of  $\alpha \in \mathbb{R}$ ) in order to enforce a certain exponential behavior outside of the interpolation interval. Moreover, in the same paper it is additionally assumed that the boundary pieces in (2) are contained in a different space  $\mathbb{E}_{2,-\alpha} := \text{span}\{e^{-\alpha x}, x e^{-\alpha x}\}$ .

Following [12], we furthermore use a Bernstein-like basis to represent each segment of the GB-splines  $\{\varphi_j\}_{j=1}^n$  (see [12, Appendix] for an explicit construction), which are defined as

$$\varphi_j(x)|_{[x_i, x_{i+1}]} = \sum_{k=1}^4 \gamma_{i,j,k} B_k(x - x_j), \tag{3}$$

where  $B_k, k = 1, \dots, 4$ , are Bernstein-like functions,  $1 \leq i \leq n - 1$  and  $1 \leq j \leq n$ . The existence and uniqueness of such a functional space is proved in [12, Theorem 2.1]. The advantage of the GB-spline basis is that the computations can be performed locally in the support of each  $\varphi_j$ , where  $\text{supp}(\varphi_j) \subset [x_{j-2}, x_{j+2}]$ ,  $1 \leq j \leq n$ . In particular, the global interpolation matrix  $\Phi$  with entries given by  $\Phi_{ij} := \varphi_j(x_i)$  is tridiagonal, non-singular, and the vector of the coefficients  $\mathbf{c} := [c_1, \dots, c_n]^T \in \mathbb{R}^n$  in (1) is the solution of

$$\Phi \mathbf{c} = \mathbf{y}, \tag{4}$$

where  $\mathbf{y} := [y_1, \dots, y_n]^T \in \mathbb{R}^n$ . Observe that we have  $I_{X,\alpha}(f) = f$  for all  $f \in \mathcal{E}_{X,\alpha}$ , i.e., every function in  $\mathcal{E}_{X,\alpha}$  is uniquely determined by its values on  $X$ . We summarize in Algorithm 1 the steps for computing the EPS interpolant.

### 2.2 Greedy schemes

We briefly recall the main ideas behind greedy techniques.

Given  $X$  and  $F$ , the main goal of a greedy algorithm consists in selecting a suitable subset  $\tilde{X} \subset X$  so that the *greedy interpolant* is constructed on a smaller number of data, hence producing an approximation of the interpolation operator  $I_X$ , meaning that  $I_{\tilde{X}}f$  is close to  $I_Xf$  in some suitable norm. Such iterative algorithms belong essentially to two classes:

- Target-data-dependent greedy schemes: the set  $\tilde{X}$  is constructed taking into account the function values  $F$ .

- Target-data-independent greedy methods: the set  $\tilde{X}$  is built independently of the function values  $F$ . Observe that in this case it is reasonable to expect that  $I_{\tilde{X}}g$  is close to  $I_Xg$  also for a larger class of functions  $g \neq f$ .

The general iterative rules for these two algorithms are summarized in Table 1, where  $\lambda$  denotes a pointwise error indicator independent of the function values. We remark that the same notation will be used from the next section also to denote the Lebesgue function, but this should create no confusion since it is the actual error indicator that we will use in practice. Both methods will be investigated in Section 4 for the special case of EPS.

**Remark 2** We would like to point out that the notation and terminology used in this section is taken from the literature on greedy kernel methods (see, e.g., [27]), where a similar distinction has been introduced for the target-data-independent  $P$ -greedy algorithm, and the target-data-dependent  $f$ -,  $f|P$ -, and  $f \cdot P$ -greedy algorithms.

We will use the same language in the rest of this paper to describe and classify greedy algorithms.

### 3 Quasi-optimal point locations

Interpolatory approximation schemes can be analyzed in terms of their relation to best approximation. To this end, we review the construction of a cardinal basis and the associated definition of the Lebesgue constant, and use it to prove the quasi-optimality of equally spaced points in this case.

#### 3.1 Lagrange functions and Lebesgue constant

In this section, for simplicity of notation, we sometimes omit the dependency on  $\alpha$  when no confusion arises.

Given  $\{\varphi_j\}_{j=1}^n$  as in (2), since the associated matrix  $\Phi$  is invertible we may write  $d_{j\ell} := (\Phi^{-1})_{j\ell}$ . In this way we have that the functions

$$\psi_\ell(x) := \sum_{j=1}^n d_{j\ell} \varphi_j(x), \quad 1 \leq \ell \leq n,$$

satisfy the cardinal conditions

**Table 1** Point selection rules for the target-data-dependent and independent greedy strategies

Greedy method	Selection rule
Target-data-dependent	$x^* = \operatorname{argmax}_{x \in X \setminus \tilde{X}}  f(x) - I_{X,\alpha}(f)(x) $
Target-data-independent	$x^* = \operatorname{argmax}_{x \in X \setminus \tilde{X}} \lambda(x)$

$$\psi_\ell(x_i) = \delta_{i\ell}, \quad 1 \leq i, \ell \leq n, \tag{5}$$

i.e., they are a global Lagrange (or cardinal) basis. To see this, just observe that for  $1 \leq i, \ell \leq n$ , it holds true that

$$\psi_\ell(x_i) = \sum_{j=1}^n \varphi_j(x_i) d_{j\ell} = \sum_{j=1}^n \Phi_{ij}(\Phi^{-1})_{j\ell} = (\Phi \cdot \Phi^{-1})_{i\ell} = \delta_{i\ell}, \quad 1 \leq i, \ell \leq n.$$

Using the cardinal basis, the interpolant (1) may be written as

$$I_{X,\alpha}(f)(x) = \sum_{j=1}^n f(x_j) \psi_j(x), \quad x \in [a, b]. \tag{6}$$

Some examples of cardinal bases for the EPS are plotted in Fig. 2. In this illustrative example, the cardinal functions are computed for  $n = 8$  equispaced, Halton and Chebyshev data locations, and are evaluated on 400 equispaced points.

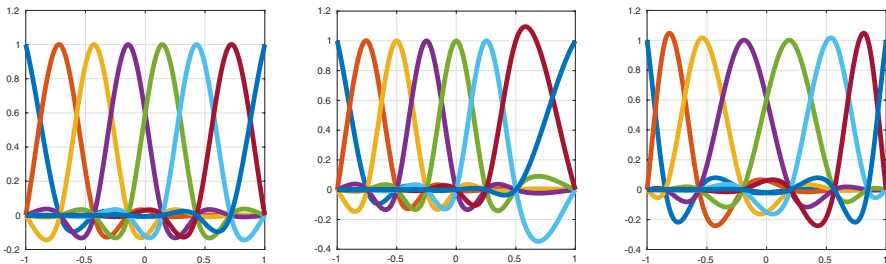
Once the cardinal basis is computed, the Lebesgue function is defined in the usual way as

$$\lambda(x) := \lambda_{X,\alpha}(x) = \sum_{j=1}^n |\psi_j(x)|, \quad x \in [a, b],$$

and its maximum value is called the Lebesgue constant, defined by

$$\Lambda := \Lambda_{X,\alpha} = \sup_{a \leq x \leq b} \lambda_{X,\alpha}(x).$$

Both  $\lambda$  and  $\Lambda$  depend on the location of the interpolation points and on their number  $n$ , but not on the function values and, as it is well known, they are stability indicators.



**Fig. 2** From left to right: cardinal functions computed on  $n = 8$  equispaced, Halton and Chebyshev points, respectively



### 3.1.1 Lebesgue function and error estimation

The Lebesgue constant allows to simply relate the best approximation and interpolation error. For completeness we review the details of this fact in the following result, which is a simple instance of the classical Lebesgue Lemma (see, e.g., [30, 31]).

**Theorem 1** (Approximation error) *Let  $f \in C([a, b])$  and let  $f_{X,\alpha}^* \in \mathcal{E}_{X,\alpha}$  be its best approximation in  $\mathcal{E}_{X,\alpha}$  with respect to the norm  $\|\cdot\|_\infty$ . Then it holds that*

$$\left| (f - I_{X,\alpha}(f))(x) \right| \leq (1 + \lambda_{X,\alpha}(x)) \left\| f - f_{X,\alpha}^* \right\|_\infty, \quad x \in [a, b]. \tag{7}$$

**Proof** Since  $I_{X,\alpha}(g) = g$  for all  $g \in \mathcal{E}_{X,\alpha}$ , and in particular for  $g = f_{X,\alpha}^*$ , and using the arguments provided by, e.g., [30], we have that

$$\begin{aligned} |f(x) - I_{X,\alpha}(f)(x)| &= \left| f(x) - f_{X,\alpha}^*(x) + f_{X,\alpha}^*(x) - I_{X,\alpha}(f)(x) \right| \\ &\leq \left\| f - f_{X,\alpha}^* \right\|_\infty + \left| I_{X,\alpha}(f_{X,\alpha}^* - f)(x) \right|. \end{aligned} \tag{8}$$

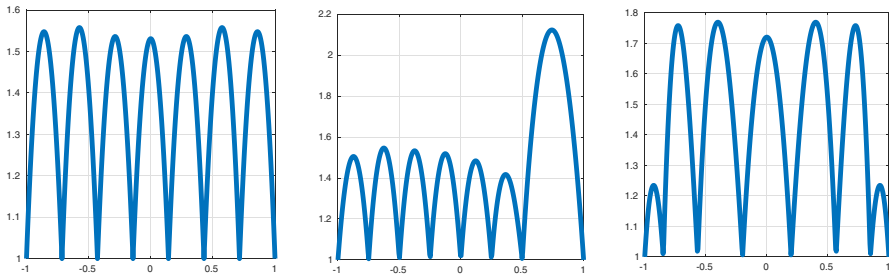
To bound the second term we use (6) and thus:

$$\begin{aligned} \left| I_{X,\alpha}(f_{X,\alpha}^* - f)(x) \right| &= \left| \sum_{j=1}^n (f_{X,\alpha}^* - f)(x_j) \psi_j(x) \right| \\ &\leq \max_{1 \leq j \leq n} \left| (f_{X,\alpha}^* - f)(x_j) \right| \sum_{j=1}^n |\psi_j(x)| \\ &\leq \left\| f_{X,\alpha}^* - f \right\|_\infty \lambda(x). \end{aligned}$$

Then, the thesis follows from (8).

**Remark 3 (Related results)** Observe that the error bound in (7) is analogous but not equivalent to similar statements in other methods (e.g., polynomial or kernel-based interpolation). Indeed, the splitting of the error on the right hand side is only partially separating the  $f$ -dependent and the  $f$ -independent terms, since the best approximant  $f_{X,\alpha}^*$  is depending on the interpolation points. In other words, one may try to minimize the first term to find *good*, i.e., sub-optimal, interpolation points, but this may spoil the second term.

As an illustrative example in Fig. 3, in the same setting as in Fig. 2 we plot the Lebesgue functions corresponding to  $n = 8$  equispaced, Halton and Chebyshev points, and evaluated on a grid of 400 points. Observe that in this case the Chebyshev points seem to not provide the smallest Lebesgue constant, in contrast with interpolation with global polynomials. We further explore this fact in the following section.



**Fig. 3** From left to right: Lebesgue functions computed on  $n = 8$  equispaced, Halton and Chebyshev points, respectively

### 3.2 Equally spaced points

In the case of equally spaced points, a more explicit characterization of the exponential-spline basis is known. We use it to derive precise bounds on the Lebesgue constant.

To this end we set

$$p_\alpha(t) := \exp(\alpha t) + \exp(-\alpha t), \quad m_\alpha(t) := \exp(\alpha t) - \exp(-\alpha t),$$

and we have from [32] that the generating exponential  $B$ -spline  $B^{(\alpha)} : \mathbb{R} \rightarrow \mathbb{R}$ , defined on the integer points  $\mathbb{Z}$ , can be computed as

$$B^{(\alpha)}(t) = \begin{cases} f_\alpha^{(k)}(t), & k - 1 < t \leq k, 1 \leq k \leq 4, \\ 0, & t \notin (0, 4), \end{cases}$$

with

$$\begin{aligned} f_\alpha^{(1)}(t) &:= \frac{1}{4\alpha^2} \left( t p_\alpha(t) - \frac{1}{\alpha} m_\alpha(t) \right), \\ f_\alpha^{(2)}(t) &:= \frac{1}{4\alpha^2} \left( -2(t-1)p_\alpha(t-2) - (t-2)p_\alpha(t) + \frac{2}{\alpha} m_\alpha(t-2) + \frac{1}{\alpha} m_\alpha(t) \right), \\ f_\alpha^{(3)}(t) &:= \frac{1}{4\alpha^2} \left( (t-2)p_\alpha(t-4) + 2(t-3)p_\alpha(t-2) - \frac{1}{\alpha} m_\alpha(t-4) - \frac{2}{\alpha} m_\alpha(t-2) \right), \\ f_\alpha^{(4)}(t) &:= \frac{1}{4\alpha^2} \left( -(t-4)p_\alpha(t-4) + \frac{1}{\alpha} m_\alpha(t-4) \right). \end{aligned}$$

It is easily verified that  $B^{(\alpha)}$  is symmetric around the point 2, where it has a global maximum, and that  $B^{(\alpha)} \in C^2(\mathbb{R})$ . In particular it holds that  $B^{(\alpha)}(0) = B^{(\alpha)}(4) = 0$ . This base spline can be applied to the regular grid  $h\mathbb{Z}$  with  $h > 0$  simply by scaling the input, i.e., defining  $B^{(\alpha,h)}(t) := B^{(\alpha)}(t/h)$ .

One may now consider an interval  $[a, b] \subset \mathbb{R}$ , a number  $n \in \mathbb{N}$ , a grid size  $h := (b - a)/(n - 1)$ , and define a set of  $n$  equally spaced interpolation points  $x_j \in [a, b]$ ,  $1 \leq j \leq n$ . Without loss of generality (i.e., up to a translation), we assume in this section that  $a = 0$ , so that  $x_j = (j - 1) \cdot h \in [a, b]$ ,  $1 \leq j \leq n$ . It has

been proven in [32] that for all  $\alpha \in \mathbb{R}$  a basis for the space  $\mathcal{E}_{x,\alpha}(\mathbb{E}_{4,\alpha})$  is given by the exponential  $B$ -splines  $B_j, 1 \leq j \leq n$  defined by

$$B_j(x) := \frac{1}{h^2} B^{(\alpha,h)}(x - x_j + 2h) = \frac{1}{h^2} B^{(\alpha,h)}(x - (j - 1)h + 2h), \tag{9}$$

i.e., each  $B_j$  is given by the translation of  $B^{(\alpha,h)}$  so that it has a maximum in  $x_j$ , and by a scaling by  $h^2$ . It follows that

$$\begin{aligned} B_j(x_i) &= \frac{1}{h^2} B^{(\alpha,h)}((i - 1)h - (j - 1)h + 2h) = \frac{1}{h^2} B^{(\alpha,h)}(h(i - j + 2)) \\ &= \frac{1}{h^2} \begin{cases} b_{-1} := B^{(\alpha,h)}(h), & i - j + 2 = 1 \\ b_0 := B^{(\alpha,h)}(2h), & i - j + 2 = 2 \\ b_1 := B^{(\alpha,h)}(3h), & i - j + 2 = 3 \\ 0, & \text{otherwise} \end{cases} = \frac{1}{h^2} \begin{cases} b_1, & i = j - 1 \\ b_0, & i = j \\ b_1, & i = j + 1 \\ 0, & \text{otherwise,} \end{cases} \end{aligned} \tag{10}$$

where we used the fact that  $B^{(\alpha,h)}(h) = B^{(\alpha,h)}(3h)$  since  $B^{(\alpha,h)}$  is symmetric around  $2h$ .

We assume that the extended points are also equally spaced, i.e.,  $x_{-1} := a - 2h, x_0 := a - h, x_{n+1} := b + h, x_{n+2} := b + 2h$ . Using (9), it follows that the interpolation matrix  $\Phi_h := (B_j(x_i))_{i,j=1}^n \in \mathbb{R}^{n \times n}$  with respect to this basis is indeed a tridiagonal symmetric Toeplitz matrix, i.e.,

$$\Phi_h = \frac{1}{h^2} \begin{bmatrix} b_0 & b_1 & \dots & 0 \\ b_1 & b_0 & \dots & 0 \\ \vdots & \ddots & \ddots & b_1 \\ 0 & \dots & b_1 & b_0 \end{bmatrix} =: \frac{1}{h^2} \Phi,$$

where we set  $\Phi := \text{Toep}(b_1, b_0, b_1)$ , i.e., the  $h$ -independent symmetric Toeplitz matrix that appears in the last equation.

In the following we will derive some bounds involving various combinations of the values  $b_0$  and  $b_1$ , for which we will need the following lemma.

**Lemma 1** *For all  $\alpha \in \mathbb{R}$  we have*

$$b_0 = \frac{1}{\alpha^2} \left( -1 + \frac{1}{2\alpha} \sinh(2\alpha) \right), \quad b_1 = \frac{1}{2\alpha^2} \left( \cosh(\alpha) - \frac{1}{\alpha} \sinh(\alpha) \right). \tag{11}$$

Moreover

$$b_0 - 2b_1 = \frac{2}{\alpha^3} \cosh\left(\frac{\alpha}{2}\right)^2 (\sinh(\alpha) - \alpha) \geq \frac{1}{3} \left( 1 + \frac{1}{20} \alpha^2 \right), \tag{12}$$

and

$$\kappa(\alpha) := \frac{b_0 + 2b_1}{b_0 - 2b_1} = \tanh\left(\frac{\alpha}{2}\right)^2 \frac{\sinh(\alpha) + \alpha}{\sinh(\alpha) - \alpha}, \tag{13}$$

with  $1 \leq \kappa(\alpha) \leq 3, \lim_{\alpha \rightarrow \pm\infty} \kappa(\alpha) = 1, \lim_{\alpha \rightarrow 0} \kappa(\alpha) = 3$ .

**Proof** See Appendix A.

We can thus use the known expression for the eigenvalues of an  $n \times n$  tridiagonal Toeplitz matrix (see [33]), obtaining

$$\lambda_k(\Phi) = b_0 + 2b_1 \cos\left(\pi \frac{k}{n+1}\right), \quad 1 \leq k \leq n,$$

and thus

$$\lambda_k(\Phi_h) = \frac{1}{h^2} \left( b_0 + 2b_1 \cos\left(\pi \frac{k}{n+1}\right) \right), \quad 1 \leq k \leq n. \tag{14}$$

We have in particular that  $\lambda_1(\Phi_h) \geq \lambda_2(\Phi_h) \geq \dots \geq \lambda_n(\Phi_h)$ , since  $\cos(\pi x)$  is decreasing for  $x := k/(n+1) \in (0, 1)$ .

Moreover the bound (12) together with (14) implies that

$$\lambda_k(\Phi_h) \geq \lambda_N(\Phi_h) \geq \frac{1}{h^2} (b_0 - 2b_1) \geq \frac{1}{3h^2} \left( 1 + \frac{\alpha^2}{20} \right) > 0, \tag{15}$$

and in particular this implies that  $\Phi_h$  is positive definite for all  $\alpha \in \mathbb{R}$ , since is symmetric by definition. It is thus also a normal matrix, and in particular the singular values  $\sigma_k(\Phi_h)$  of  $\Phi_h$  coincide with its eigenvalues.

We combine these facts with the bounds of Lemma 1 to obtain the following estimates regarding the matrix  $\Phi_h$  and its inverse.

**Proposition 1** *For all  $\alpha \in \mathbb{R}$ , if the interpolation points are equally spaced it holds*

$$\begin{aligned} \|\Phi_h^{-1}\|_2 &\leq \frac{h^2}{b_0 - 2b_1} \leq \frac{3h^2}{1 + \alpha^2/20}, \\ \|\Phi_h^{-1}\|_\infty &\leq \frac{h^2}{b_0 - 2b_1} \leq \frac{3h^2}{1 + \alpha^2/20}, \\ \text{cond}_2(\Phi_h) &\leq \frac{b_0 + 2b_1}{b_0 - 2b_1} \leq \tanh\left(\frac{\alpha}{2}\right)^2 \frac{\sinh(\alpha) + \alpha}{\sinh(\alpha) - \alpha} \leq 3. \end{aligned}$$

*In particular for all  $h > 0$  we have that  $\lim_{\alpha \rightarrow \pm\infty} \text{cond}_2(\Phi_h) = 1$  and  $\text{cond}_2(\Phi_h) = 3 - \frac{2}{5}\alpha^2 + \mathcal{O}(\alpha^4)$  for  $\alpha \rightarrow 0$ .*

**Proof** Even if it would be sufficient to use that fact that  $\|\Phi_h^{-1}\|_2 \leq \|\Phi_h^{-1}\|_\infty$ , we simply use (15) to obtain

$$\|\Phi_h^{-1}\|_2 = \sigma_1(\Phi_h^{-1}) = \frac{1}{\lambda_n(\Phi_h)} \leq \frac{h^2}{b_0 - 2b_1} \leq \frac{3h^2}{1 + \alpha^2/20}.$$

For the condition number instead it holds that

$$\text{cond}_2(\Phi) = \frac{\sigma_1(\Phi)}{\sigma_n(\Phi)} = \frac{\lambda_1(\Phi)}{\lambda_n(\Phi)} = \frac{b_0 + 2b_1 \cos(\pi/(n + 1))}{b_0 + 2b_1 \cos(\pi n/(n + 1))} \leq \frac{b_0 + 2b_1}{b_0 - 2b_1},$$

and we may use Lemma 1 to bound the last term.

For the  $\infty$ -norm we have

$$\|\Phi^{-1}\|_\infty = \max_{0 \neq v \in \mathbb{R}^n} \frac{\|\Phi^{-1}v\|_\infty}{\|v\|_\infty} = \max_{0 \neq w \in \mathbb{R}^n} \frac{\|w\|_\infty}{\|\Phi w\|_\infty} = \left( \min_{0 \neq w \in \mathbb{R}^n} \frac{\|\Phi w\|_\infty}{\|w\|_\infty} \right)^{-1},$$

where we used the fact that  $\Phi$  is invertible and thus we can change variable from  $v \neq 0$  to  $w := \Phi^{-1}v \neq 0$ . Moreover, up to rescaling the numerator and denominator by the same term we may assume  $\|w\|_\infty = 1$ , i.e.,

$$\min_{0 \neq w \in \mathbb{R}^n} \frac{\|\Phi w\|_\infty}{\|w\|_\infty} = \min_{\|w\|_\infty \leq 1} \|\Phi w\|_\infty.$$

Now, setting  $w := [w_1, \dots, w_n]^T \in \mathbb{R}^n$ , by definition of  $\Phi$  we have

$$\Phi w = \begin{bmatrix} b_0 w_1 + b_1 w_2 \\ b_1 w_1 + b_0 w_2 + b_1 w_3 \\ \vdots \\ b_1 w_{n-2} + b_0 w_{n-1} + b_1 w_n \\ b_1 w_{n-1} + b_0 w_n \end{bmatrix},$$

and thus

$$\min_{\|w\|_\infty \leq 1} \|\Phi w\|_\infty = \min_{-1 \leq w_i \leq 1} \left\{ b_0 w_1 + b_1 w_2, b_1 w_{n-1} + b_0 w_n, \min_{2 \leq i \leq n-1} |b_1(w_{i-1} + w_{i+1}) + b_0 w_i| \right\}.$$

Since in the last term the objective function and the constraints are all linear, the minimum is necessarily reached when the constraints are met with equality, i.e.,  $w_i \in \{-1, 1\}$ . Moreover, all the terms in the minimum are minimized by the same value independently of  $i$ . We thus have

$$\min_{\|w\|_\infty \leq 1} \|\Phi w\|_\infty = \min \left\{ \min_{w_1, w_2 \in \{-1, 1\}} |b_0 w_1 + b_1 w_2|, \min_{w_1, w_2, w_3 \in \{-1, 1\}} |b_1(w_1 + w_3) + b_0 w_2| \right\}.$$

Checking all the possible values by enumeration, and remembering that  $b_0 - 2b_1 > 0$ , we have that the argument of the first minimum can take the values  $b_0 - b_1$  or  $b_0 + b_1$ , while the second the values  $b_0, b_0 + 2b_1, b_0 - 2b_1$ . This last value is the smallest one, and thus

$$\|\Phi^{-1}\|_\infty \leq \frac{1}{b_0 - 2b_1},$$

which gives the desired bound.

Observe that the bound on the condition number guarantees that computing the interpolant with equally spaced points is a stable numerical operation, and this stability does not deteriorate as the number of points increases. Moreover, we may use the bound on the  $\infty$ -norm of the interpolation matrix to derive the following.

**Theorem 2** *The Lebesgue constant for equally spaced points satisfies the bound*

$$\Lambda_{X,\alpha} \leq \frac{b_0 + 2b_1}{b_0 - 2b_1} \leq \tanh\left(\frac{\alpha}{2}\right)^2 \frac{\sinh(\alpha) + \alpha}{\sinh(\alpha) - \alpha} \in [1, 3],$$

with  $\lim_{\alpha \rightarrow \pm\infty} \Lambda_{X,\alpha} = 1$  and  $\lim_{\alpha \rightarrow 0} \Lambda_{X,\alpha} = 3$ .

**Proof** For any  $x \in [a, b]$  we define the set of indices  $I(x) := \{i \in \{1, \dots, n\} : B_i(x) \neq 0\}$ . The definition of the Lebesgue function thus gives

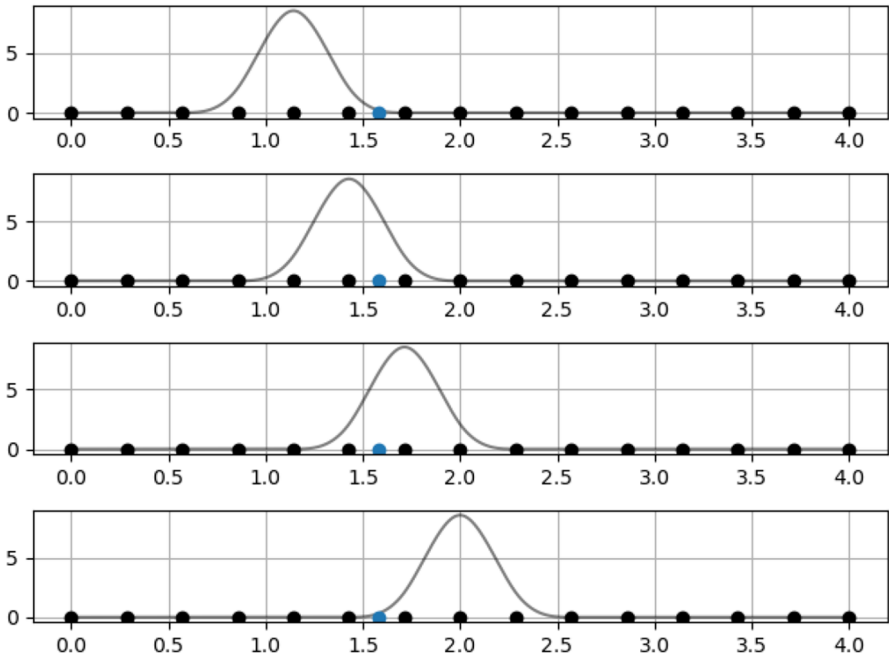
$$\begin{aligned} \lambda_{X,\alpha}(x) &= \sup_{0 \neq f \in C([a,b])} \frac{|I_x f(x)|}{\|f\|_\infty} = \max_{0 \neq v \in \mathbb{R}^n} \frac{\left| \sum_{i=1}^N (\Phi_h^{-1} v)_i B_i(x) \right|}{\|v\|_\infty} \\ &= \max_{0 \neq v \in \mathbb{R}^n} \frac{\left| \sum_{i \in I(x)} (\Phi_h^{-1} v)_i B_i(x) \right|}{\|v\|_\infty} \\ &\leq \max_{0 \neq v \in \mathbb{R}^n} \frac{\|\Phi_h^{-1} v\|_\infty}{\|v\|_\infty} \sum_{i \in I(x)} |B_i(x)| = \|\Phi_h^{-1}\|_\infty \sum_{i \in I(x)} |B_i(x)|. \end{aligned}$$

To bound the last term, we assume that  $x \in [x_k, x_{k+1}]$  for some  $k \in \{1, \dots, n-1\}$ , so that  $I(x) = \{k-1, k, k+1, k+2\} \cap \{1, \dots, n\}$  (see Fig. 4).

We set  $h_x := x - x_k \in [0, h]$ , so that  $x - x_{k-1} = h + h_x$ ,  $x - x_{k+1} = -h + h_x$ ,  $x - x_{k+2} = -2h + h_x$ , and thus using (8) we have

$$\begin{aligned} h^2 B_{k-1}(x) &= B^{(\alpha,h)}(x - x_{k-1} + 2h) = B^{(\alpha,h)}(h_x + 3h) = B^{(\alpha)}(h_x/h + 3) \\ &= f_\alpha^{(4)}(3 + h_x/h), \\ h^2 B_k(x) &= B^{(\alpha,h)}(x - x_k + 2h) = B^{(\alpha,h)}(h_x + 2h) = B^{(\alpha)}(h_x/h + 2) \\ &= f_\alpha^{(3)}(2 + h_x/h), \\ h^2 B_{k+1}(x) &= B^{(\alpha,h)}(x - x_{k+1} + 2h) = B^{(\alpha,h)}(h_x + h) = B^{(\alpha)}(h_x/h + 1) \\ &= f_\alpha^{(2)}(1 + h_x/h), \\ h^2 B_{k+2}(x) &= B^{(\alpha,h)}(x - x_{k+2} + 2h) = B^{(\alpha,h)}(h_x) = B^{(\alpha)}(h_x/h) \\ &= f_\alpha^{(1)}(h_x/h), \end{aligned}$$

and hence



**Fig. 4** Example of positions of the basis elements  $B_j$ . The figure shows as black dots a set of  $n = 15$  equally spaced points in  $[0, 4]$ , and in blue an arbitrary point  $x \in [0, 4]$ . For this setting and  $\alpha = 2$ , the four panels show the basis elements  $B_j$  which are non-zero in  $x$

$$\sum_{i \in I(x)} |B_i(x)| \leq \frac{1}{h^2} \sum_{i=1}^4 f_\alpha^{(i)}((i-1) + h_x/h),$$

where the inequality is an equality when  $I(x)$  contain the four elements  $\{k-1, \dots, k+2\}$ , and a strict inequality when some of these indices are outside the set  $\{1, \dots, N\}$ . Direct computation gives furthermore

$$\begin{aligned} F_\alpha(t) &:= \sum_{i=1}^4 f_\alpha^{(i)}((i-1) + t) \\ &= \frac{2}{\alpha^3} \sinh\left(\frac{\alpha}{2}\right)^2 ((1-t)\alpha \cosh(t\alpha) + t\alpha \cosh(\alpha - t\alpha) + \sinh(t\alpha) + \sinh(\alpha(1-t))). \end{aligned}$$

The function  $F_\alpha : [0, 1] \rightarrow \mathbb{R}$  is continuously differentiable, with

$$F'_\alpha(t) = \frac{2}{\alpha} \sinh\left(\frac{\alpha}{2}\right)^2 ((1-t) \sinh(t\alpha) + t \sinh(\alpha(1-t))),$$

and thus  $F'_\alpha$  vanishes in  $t = 1/2$ , with  $F'_\alpha(t) < 0$  for  $t \in [0, 1/2]$  and  $F'_\alpha(t) > 0$  for  $t \in [1/2, 1]$ . Moreover

$$F_\alpha(0) = F_\alpha(1) = \frac{2}{\alpha^3} \sinh\left(\frac{\alpha}{2}\right)^2 (\sinh(\alpha) + \alpha) = b_0 + 2b_1$$

and it follows that  $F_\alpha(t) \leq F_\alpha(0) = F_\alpha(1)$  for all  $t \in [0, 1]$ , and thus

$$\sum_{i \in I(x)} |B_i(x)| \leq \frac{1}{h^2} F_\alpha(h_x/h) \leq \frac{1}{h^2} F_\alpha(0) = \frac{1}{h^2} (b_0 + 2b_1).$$

Using now the bound on the  $\infty$ -norm of  $\Phi_h^{-1}$ , it follows that

$$\lambda_{X,\alpha}(x) \leq \left\| \Phi_h^{-1} \right\|_\infty \sum_{i \in I(x)} |B_i(x)| \leq \frac{h^2}{b_0 - 2b_1} \frac{b_0 + 2b_1}{h^2},$$

and the result follows by applying Lemma 1.

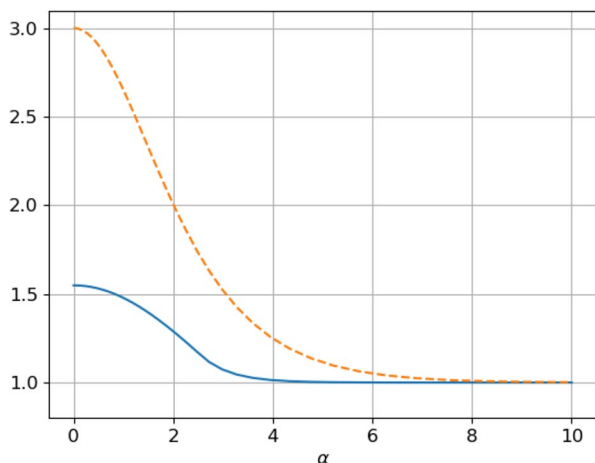
This theorem, in conjunction with Theorem 1, proves that interpolation on equally spaced points is quasi-optimal. Indeed, for all  $f \in C([a, b])$  we have in this case

$$\left| (f - I_{X,\alpha}(f))(x) \right| \leq 4 \left\| f - f_{X,\alpha}^* \right\|_\infty, \quad x \in [a, b],$$

which indeed proves that interpolation with equally spaced points provides the same asymptotic error of best approximation, even if with a different constant, which is however rather small. In addition, this constant converges for  $\alpha \rightarrow \pm\infty$  to the optimal value that can be attained with the estimate of Theorem 1, i.e.,  $1 + \Lambda_{X,\alpha} = 2$ , as proven again by Theorem 2.

It should be furthermore noted that the estimate of Theorem 2 seems to be not sharp for small  $\alpha$ , since numerical evidence suggests that  $\Lambda_{X,\alpha} \approx 1.6$  as  $\alpha \rightarrow 0$ . To give a glance at this fact, we show in Fig. 5 the behavior of the computed Lebesgue constant and of the upper bound of Theorem 2 for  $n = 100$  equally spaced

**Fig. 5** Example of the behavior of the Lebesgue constant as a function of  $\alpha$ . The figure shows for  $n = 100$  and  $\alpha \in [0, 10]$  the value of the Lebesgue constant (blue solid line) and the bound of Theorem 2 (orange dashed line)





---

**Inputs:**  $X \subseteq \Omega$ , the augmented nodes,  $F$ ,  $\tau$ ,  $\alpha$ .

**Outputs:**  $\tilde{X} \subset X$ ,  $I_{\tilde{X},\alpha}(f)$ .

- 1: Take an initial set of sorted points  $\tilde{X} \subset X$ , with  $a, b \in \tilde{X}$ .
  - 2: Compute an initial interpolant  $I_{\tilde{X},\alpha}(f)$  as in (1).
  - 3: While  $\max_{x_i \in X \setminus \tilde{X}} |f(x_i) - I_{\tilde{X},\alpha}(f)(x_i)| > \tau$ :
    - i. Define  $x^* = \operatorname{argmax}_{x_i \in X \setminus \tilde{X}} |f(x_i) - I_{\tilde{X},\alpha}(f)(x_i)|$ .
    - ii. Set  $\tilde{X} = \tilde{X} \cup \{x^*\}$  and sort  $\tilde{X}$ .
    - iii. Compute  $I_{\tilde{X},\alpha}(f)$  as in (1).
- 

**Algorithm 2** Pseudo-code for the  $f$ -greedy algorithm

points in  $[0, 2]$ , and  $\alpha \in [0, 10]$ . Observe that, due to the symmetry of the basis functions, the same values are observed for negative  $\alpha$ .

Despite these desirable quasi-optimality properties, equally spaced interpolation points may not be available in practice, either because of some obstructions to the sampling of certain values, or because the data points are provided as a given, pre-recorded dataset. In these cases, it is thus of interest to have methods that are able to select a small subset of the data so that the resulting interpolant is fast and stable, while providing a sufficient accuracy. To address this aspect, we investigate in the next section greedy point selection strategies.

## 4 Greedy schemes for EPS

In this section we first recall a simple target-data-dependent greedy scheme, that is known as  $f$ -greedy in literature (see [34]), and that can be easily used with any interpolation basis. On the other hand, target-data-independent greedy schemes need to be tailored for the considered basis, and we will discuss their implementation for EPS.

### 4.1 Target-data-dependent greedy selection

As already mentioned,  $f$ -greedy schemes are quite straightforward to extend to any kind of basis. Precisely, we consider an initial (training) set of sorted points  $\tilde{X} \subset X$ , with  $a, b \in \tilde{X}$  and we also keep the augmented nodes fixed. Then, given  $F$  and a fixed tolerance  $\tau$ , the target-data-dependent greedy scheme for exponential-polynomial splines is summarized in Algorithm 2.

The result of the  $f$ -greedy scheme is thus a set of data locations  $\tilde{X}$  and the corresponding interpolant  $I_{\tilde{X},\alpha}(f)$ . Let  $\tilde{n}$  be the cardinality of  $\tilde{X}$ . Since we usually have that  $\tilde{n} \ll n$ , the greedy interpolant  $I_{\tilde{X},\alpha}(f)$  can be understood as a sparse approximation of  $I_{X,\alpha}(f)$ .

This scheme is very easy to implement, and additionally the interpolation points are selected adaptively in order to be suited for the particular target function  $f$ , and they are thus expected to provide an accurate approximation.

**Inputs:**  $X \subseteq \Omega$ , the augmented nodes,  $F$ ,  $\tau$ ,  $\alpha$ .

**Outputs:**  $\tilde{X} \subset X$ ,  $I_{\tilde{X},\alpha}(f)$ .

- 1: Take an initial set of sorted points  $\tilde{X} \subset X$ , with  $a, b \in \tilde{X}$ .
- 2: Compute  $\lambda(x)$  with the initial set  $\tilde{X}$ .
- 3: While  $\operatorname{argmax}_{x_i \in X \setminus \tilde{X}} \lambda(x) > \tau$ :
  - i. Define  $x^* = \operatorname{argmax}_{x_i \in X \setminus \tilde{X}} \lambda(x)$ .
  - ii. Set  $\tilde{X} = \tilde{X} \cup \{x^*\}$  and sort  $\tilde{X}$ .
  - iii. Compute  $\lambda(x)$  with the set  $\tilde{X}$ .

**Algorithm 3** Pseudo-code for the  $\lambda$ -greedy algorithm

Moreover, the algorithm can be used to approximate vector-valued functions (i.e., with values in  $\mathbb{R}^p$  for some  $p \in \mathbb{N}$ , see [25]), or equivalently to simultaneously approximate  $p \in \mathbb{N}$  different scalar-valued functions. In this case, the cost of the search for the next point to be selected scales linearly with  $p$ .

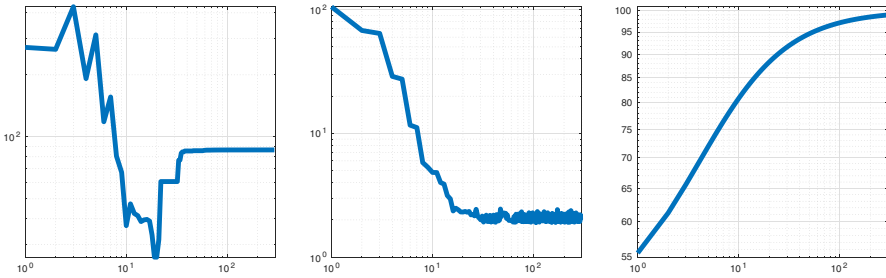
Despite the effectiveness of this method, there are cases where a set of target data values may be missing, or may be expensive to collect, such as in Uncertainty Quantification (see, e.g., [35]). In this case, it is of interest to develop target-data-independent greedy schemes, to which we drive our attention in the next section.

### 4.2 Target-data-independent greedy selection

Given the error bounds of Section 3, we introduce a new greedy selection scheme that we call  $\lambda$ -greedy. Given  $X$ ,  $F$  and  $\tau$ , a fixed tolerance, the  $\lambda$ -greedy algorithm for exponential-polynomial splines can be summarized as in Algorithm 3.

**Remark 4** (Computational aspects) Observe that the efficient execution of the  $\lambda$ -greedy algorithm depends on the efficient computation of  $\lambda$  and of  $I_{\tilde{X},\alpha}(f)$ . Both of them can be computed rather efficiently by means of the local basis. Indeed, in this case for all  $x \in X$  one needs to locate the index  $i$  such that  $x \in [x_i, x_{i+1}]$ , and then only perform local computations inside this interval.

**Remark 5** In the  $\lambda$ -greedy selection, we fix a tolerance for the Lebesgue function. However, we are able to prove the efficacy, i.e., the convergence of the  $\lambda$ -greedy scheme, only numerically. As an illustrative example, in Fig. 6, we take  $n = 300$  equispaced nodes and we apply the  $\lambda$ -greedy scheme without any stopping rule, i.e., we extract  $\tilde{n} = 300$  nodes. This didactic example aims at understanding the behavior of the Lebesgue constant when the number of nodes grows and how it relates with the conditioning of the problem. Precisely, from the first and second panel, we observe that the Lebesgue constant initially decreases and then it saturates coherently with the condition number of the interpolation matrix. In the last panel we further show the sparsity (the percentage of zero elements) of the collocation matrix that increases as the number of nodes increases. This empirically explains the behavior of the condition number and of Lebesgue functions. In other words, our  $\lambda$ -greedy is effective until both the condition number and the Lebesgue constant do



**Fig. 6** Illustrative example of the  $\lambda$ -greedy extraction of 300 nodes. At each step of the algorithm we compute the condition of the collocation matrix (left), the Lebesgue constant (middle) and the sparsity of the collocation matrix (right). Plots are in logarithmic scale and the horizontal axes denote the iteration number

not saturate. Then, as an alternative stopping rule, one may look at the difference between the Lebesgue constant or the condition number at two consecutive iterations of the  $\lambda$ -greedy scheme. We remark that this behavior is due to the fact that the interpolation matrix is sparse and it has a band structure, since each basis function  $\varphi_i$  is supported on up to four subintervals of the partition (see (1) and the subsequent discussion).

**Remark 6** As already outlined in Remark 2, in kernel interpolation the  $P$ -greedy algorithm is a well-established greedy interpolation algorithm, and it is target-data-independent similarly to the new  $\lambda$ -greedy method introduced in this section.

The two algorithms are similar since they rely on the greedy minimization of a worst-case error indicator, but they are different because of the indicator that they minimize. In more details, in kernel-based interpolation one may carry out an error analysis for the interpolation of functions  $f \in H$ , where  $(H, \langle \cdot, \cdot \rangle)$  is a certain Hilbert space associated with the kernel (the native space of the kernel or the reproducing kernel Hilbert space). Formulating the problem in a single real variable, as the case considered in the paper, we can prove that there is a continuous function  $P_X : [a, b] \rightarrow \mathbb{R}$ , named power function, such that for all  $f \in H$  the kernel interpolant  $I_X f$  provides an error bounded as follows:

$$\left| (f - I_X(f))(x) \right| \leq P_X(x) \|f - I_X f\|_H, \quad \forall x \in [a, b]. \tag{16}$$

The structure of this equation recalls (7), where the term  $1 + \lambda(x)$  plays the role of  $P_X(x)$ . A similar error bound could possibly be investigated in the case of EPS, provided we can define a RKHS structure on  $\mathcal{E}_{X,\alpha}$ . However, in our context the  $\lambda$ -greedy algorithm and the bound (7) used to derive it, have the advantage to be applicable to general continuous functions.

## 5 Numerical experiments

In the following experiments, we test both the target-data-dependent and independent schemes with different node distributions. Precisely, we consider equispaced, Halton and Chebyshev points. Moreover, for all data sets, without loss of generality, we take as initial set for the greedy strategy the first and last two nodes. Tests have been carried out on a Intel(R) Core(TM) i7 CPU 4712MQ 2.13 GHz processor.

### 5.1 Testing the $f$ -greedy

Throughout this subsection, we consider the following test function

$$f_1(x) = \alpha \tan(55x), \quad x \in [-1, 1],$$

and we further fix  $\alpha = 2$ .

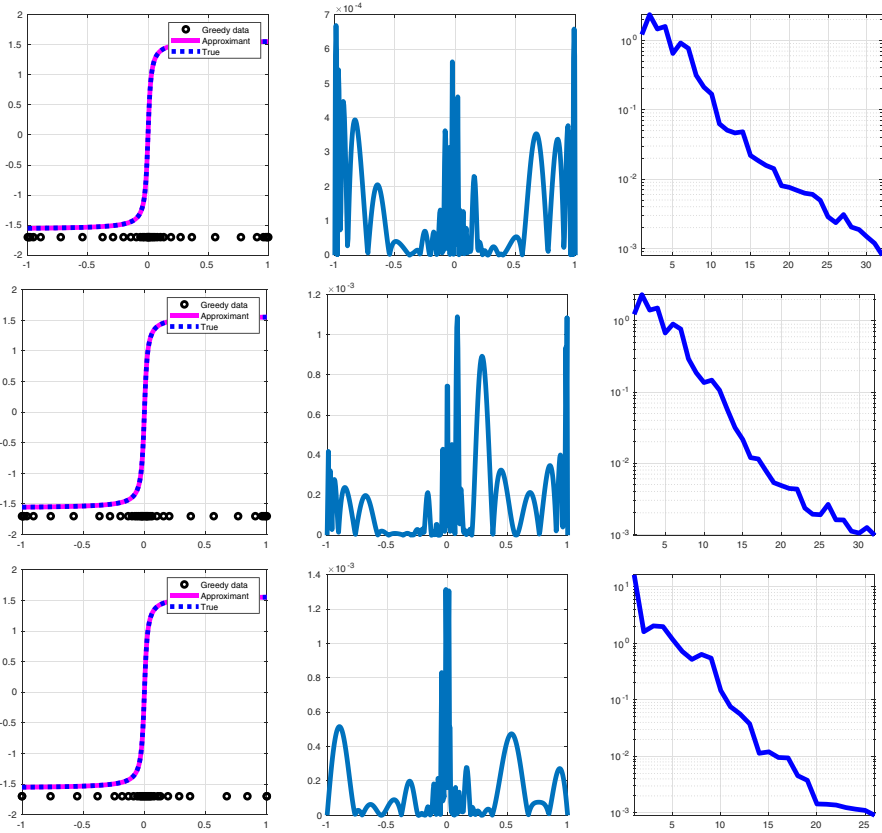
As far as the  $f$ -greedy method which makes use of exponential-polynomial splines is concerned, we fix the tolerance  $\tau = 10^{-3}$ . In Fig. 7, we plot the results obtained by selecting the points from 300 equispaced, Halton and Chebyshev points. The number of extracted greedy points are respectively  $\tilde{n} = 36, 36$  and 30 that, as expected, cluster where the test function  $f_1$  has steep gradients. To stress the importance of using the  $f$ -greedy strategy when approximating functions characterized by steep gradients or singularities, we report in Table 2 the maximum absolute error obtained by taking  $\tilde{n}$  (non-greedy) equispaced, Halton and Chebyshev points.

### 5.2 Testing the $\lambda$ -greedy

One interesting feature of the  $\lambda$ -greedy scheme is that it is able to construct node sets without specifying interpolation values, and could thus be expected to be good for any possible target function. If the initial search set is large enough, and since the Lebesgue function is optimized iteratively, it is reasonable to expect that the final set of points may be close to the optimal distribution that one would obtain by a global minimization of the Lebesgue function. This claim should of course be proven, but since these globally optimal points are not known in the case of the exponential splines considered in this work, it is of interest to study the geometrical distribution of these  $\lambda$ -greedy points to have at least a first insight.

To this end, we take an initial set of 300 equispaced points and we apply the  $\lambda$ -greedy scheme with  $\tau = 2$ . The result is depicted in Fig. 8. It is interesting to notice that for the exponential-polynomial splines, the greedy points tend to cluster close to the boundary, showing some similarities with Chebyshev nodes.

As second experiment, in Fig. 9, we plot the results of the  $\lambda$ -greedy scheme starting with 300 equispaced, Halton and Chebyshev points. In this case, we fix the tolerance as  $\tau = 3$ . The algorithm selects  $\tilde{n} = 18, 19$  and 36 equispaced, Halton and Chebyshev points, respectively. In all cases they cluster on the boundary. We point out that, consistently with what observed in Fig. 6, for small tolerances



**Fig. 7** Results for the  $f$ -greedy algorithm. First column: the extracted greedy points (black dots), the true function  $f_1$  (blue dotted line) and the reconstructed function taking the greedy points (magenta solid line). Second column: the absolute error evaluated on 400 equispaced points. Third column: the maximum of the residuals at each iteration of the greedy scheme. The experiment is carried out for equispaced, Halton and Chebyshev nodes, first, second and third row, respectively

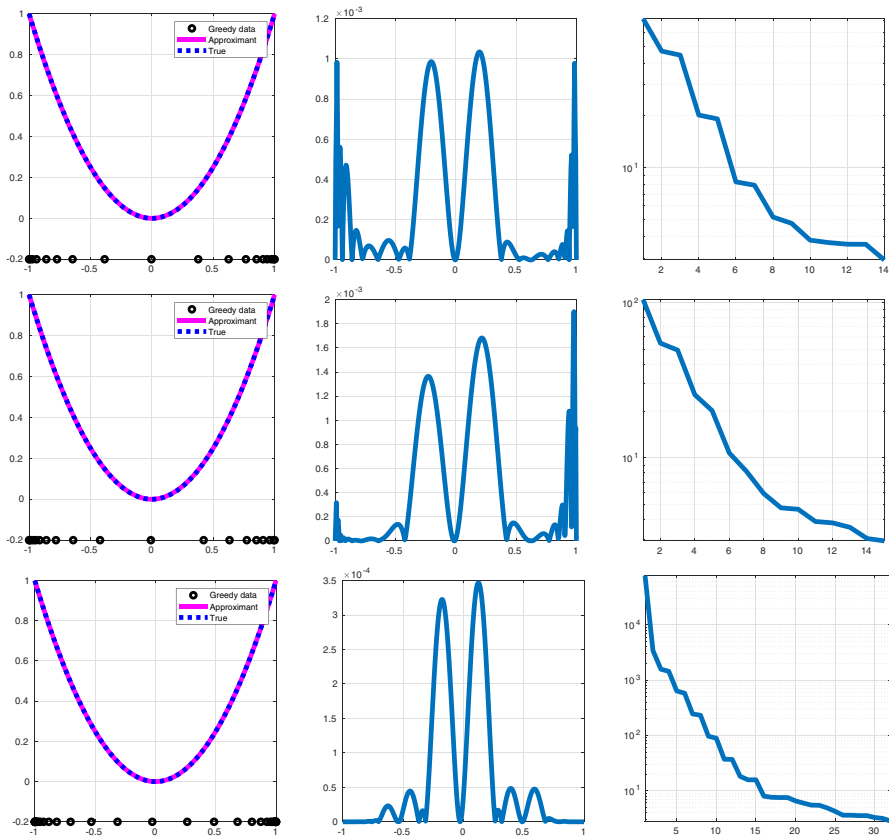
**Table 2** The three columns contain respectively the maximum absolute errors for  $f_1$  obtained via  $\tilde{n} = 36, 36$  and  $30$  greedy and non-greedy equispaced, Halton and Chebyshev points

Greedy	$6.68e - 04$	$1.08e - 03$	$1.31e - 03$
Non-Greedy	$1.11e - 01$	$3.33e - 01$	$2.13e - 01$



**Fig. 8** Node distributions obtained via the target-data-independent greedy approach for EPS

$\tau$  the algorithm may not terminate. To get a feedback on the accuracy, with the selected points, we reconstruct the function  $f_2(x) = x^2$ . The associated absolute error is depicted in the second column of Fig. 9. Furthermore, in the last column of Fig. 9, we report the Lebesgue constant at each iteration of the greedy



**Fig. 9** Results for the  $\lambda$ -greedy algorithm. First column: the extracted greedy points (black dots), the true function  $f_2$  (blue dotted line) and the reconstructed function taking the greedy points (magenta solid line). Second column: the absolute error evaluated on 400 equispaced points. Third column: the Lebesgue constant at each iteration of the greedy scheme. The experiment is carried out for equispaced, Halton and Chebyshev nodes, first, second and third row, respectively

scheme. To better analyze its asymptotic behavior, we extended the experiment to  $\tau = 2$  (i.e., selecting more points) and we obtained terminating values  $\Lambda_X = 1.94$  (for search over equally spaced points),  $\Lambda_X = 1.97$  (Halton points),  $\Lambda_X = 1.98$  (Chebyshev points). These values are very close to the numerical observed value for equally spaced interpolation points, and smaller than the corresponding upper bound proven in Section 3. Moreover, in all cases the values are bounded as a function of the number of points. This observation suggests that the points selected by the  $\lambda$ -greedy algorithm, although not being uniform, are still quasi-optimal.

Finally, to underline the importance of the  $\lambda$ -greedy strategy, we show in Table 3 the maximum absolute error obtained by taking  $\tilde{n}$  (non-greedy) equispaced, Halton points and Chebyshev points. We observe that, even if the function is smooth, a greedy selection of the points allows us to achieve a good accuracy with a relatively low number of data.

**Table 3** The three columns contain respectively the maximum absolute errors for  $f_2$  obtained via  $\tilde{n} = 18, 19$  and  $36$  greedy and non-greedy equispaced, Halton and Chebyshev points

Greedy	1.03e - 03	1.90e - 03	3.46e - 04
Non-Greedy	1.21e - 01	1.35e - 01	7.70e - 04

### 5.3 $f$ -greedy VS $\lambda$ -greedy

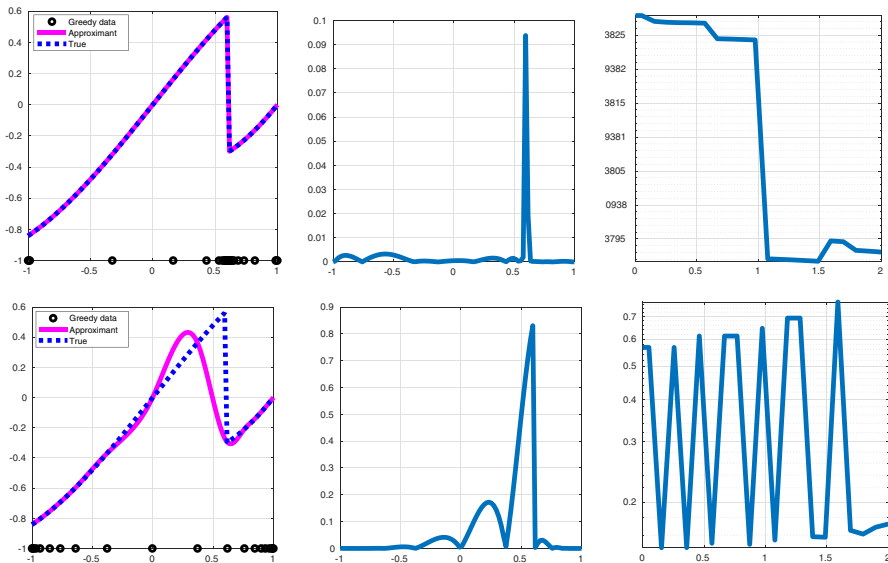
In this subsection we focus on comparing the two proposed greedy schemes. In doing so, we take a function with singularities and one belonging to the Runge family, precisely:

$$f_3(x) = \begin{cases} \sin x, & \text{if } x \leq 0.6, \\ x \log x & \text{if } x > 0.6, \end{cases}$$

and

$$f_4(x) = \frac{1}{1 + 6x^2}, \quad x \in [-1, 1].$$

In the following, particular attention is devoted to empirically observe how the non-physical oscillations known as Runge and Gibbs phenomena are mitigated via the greedy selection and how the absolute error depends on the parameter  $\alpha$ . In Figs. 10

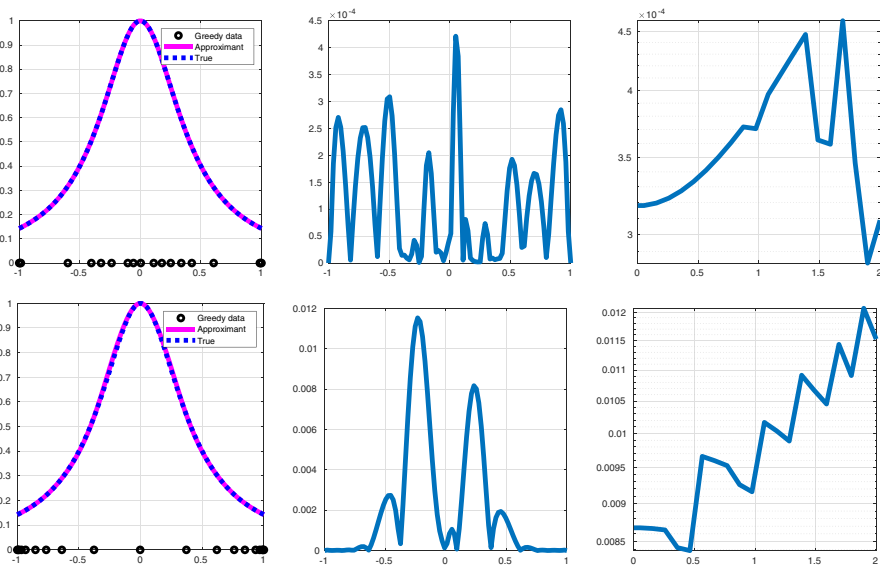


**Fig. 10** Results obtained via the  $f$ -greedy (first row) and  $\lambda$ -greedy (second row) algorithms for  $f_3$ . First column: the  $\tilde{n} = 26$  (first row) and  $20$  (second row) extracted  $f$  and  $\lambda$ -greedy points (black dots), the true function  $f_3$  (blue dotted line) and the reconstructed function taking the greedy points (magenta solid line). Second column: the absolute error evaluated on 400 equispaced points. Third column: the maximum absolute error by varying  $\alpha$  for the  $f$  and  $\lambda$ -greedy strategies

and 11 (first and second columns), we display the results obtained by taking the parameter  $\alpha = 2$  for both the  $f$ -greedy ( $\tau = 1e - 02$  for  $f_3$  and  $5e - 04$  for  $f_4$ ) and  $\lambda$ -greedy ( $\tau = 3.5$  for  $f_3$  and  $4$  for  $f_4$ ) approaches. In the last columns of Figs. 10 and 11 we show how the maximum absolute error varies according to  $\alpha$ . Precisely, the maximum absolute error associated with 30 equispaced values of  $\alpha \in (0, 2]$  are reported. In all cases, we take an initial set of 300 equispaced data. We observe that for  $f_4$  both the  $\lambda$  and  $f$ -greedy schemes return suitable approximants. On the opposite, as expected, only the  $f$ -greedy algorithm is able to capture the singularity of the function  $f_3$ . As far as the selection of  $\alpha$  is concerned, we note that the greedy algorithms are not so sensitive with respect to its selection (except the case of the  $\lambda$ -greedy algorithm for  $f_3$ , which however provides quite poor approximations). Indeed, a greedy algorithm optimally selects the nodes for the given basis (defined by  $\alpha$ ) and hence it naturally adapts to such parameter. For further details on safe ways to select  $\alpha$ , we refer the reader to [32].

### 6 $\lambda$ -greedy for nuclear magnetic resonance

The aim of this section is to investigate potential applications of our analysis. The dataset used in this experiment comes from a concrete problem where the effects of the NMR on the changes in water molecule mobility during the mixing phase of the bread making process are studied (for further details see, e.g., [36, 37]). The data



**Fig. 11** Results obtained via the  $f$ -greedy (first row) and  $\lambda$ -greedy (second row) algorithms for  $f_4$ . First column: the  $\tilde{n} = 23$  (first row) and 19 (second row) extracted  $f$  and  $\lambda$ -greedy points (black dots), the true function  $f_4$  (blue dotted line) and the reconstructed function taking the greedy points (magenta solid line). Second column: the absolute error evaluated on 400 equispaced points. Third column: the maximum absolute error by varying  $\alpha$  for the  $f$  and  $\lambda$ -greedy strategies



rapidly decay and the signal consists of  $n = 200$  values of three different acquisitions of transverse relaxation times for water protons in flour doughs, at mixing time of 3 minutes long. The tree sets of data are plotted in Fig. 12 (first panel). Being real and noisy data, we introduce a regression Tikhonov parameter  $\mu = 1e - 06$ ; refer to [3, §15, p. 276]. In the same setting of the previous experiments ( $\alpha = 2$ ), we run both the  $f$  and  $\lambda$ -greedy algorithms (we fix  $\tau$  as  $1e - 02$  and  $3.5$ , respectively). For a visual feedback on the results, refer to Fig. 12 (second and third panel). The function to reconstruct is smooth and hence the results of the two algorithms are similar.

## 7 Conclusions and work in progress

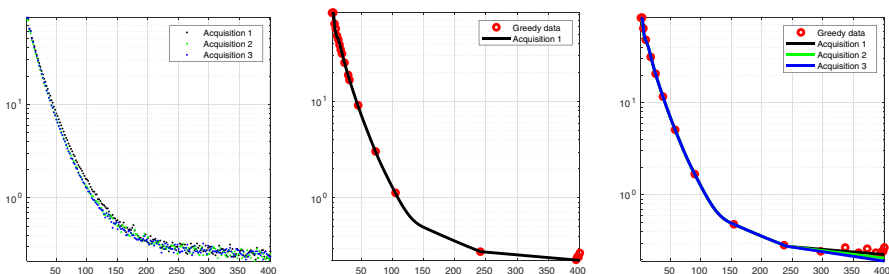
We have investigated the use of greedy strategies for Exponential-Polynomial Spline (EPS) interpolation. To this end we have studied the cardinal form of the EPS interpolant and then we have provided error bounds based on the Lebesgue function. The results show that the target-data-independent greedy points for EPS tend to cluster at the boundary of the approximation interval, despite the fact that Chebyshev points are not necessarily optimal in this case.

Work in progress consists in investigating the proposed tool in applications, as in the context of Laplace transform inversion based on smoothing splines [13], as well as for interpolation/extrapolation algorithms for the inversion of the Fourier transform [38].

Moreover, an interesting extension would be the maximization of the determinant of the interpolation matrix, instead of the Lebesgue function, to selected target-data-independent interpolation points.

## Appendix. Proof of Lemma 1

*Proof* We use the values



**Fig. 12** First panel: the NMR data at three acquisition times. Second panel: the result of the  $f$ -greedy algorithm for the first acquisition time. Third panel: the result of the  $\lambda$ -greedy scheme for the three acquisition times. Plots are in logarithmic scale

$$\begin{aligned} m_\alpha(0) &:= 0, & p_\alpha(0) &:= 2, \\ m_\alpha(1) &:= \exp(\alpha) - \exp(-\alpha), & p_\alpha(1) &:= \exp(\alpha) + \exp(-\alpha), \\ m_\alpha(2) &:= \exp(2\alpha) - \exp(-2\alpha), \end{aligned}$$

which give

$$\begin{aligned} b_0 &= B^{(\alpha)}(2) = f_\alpha^{(2)}(2) = \frac{1}{4\alpha^2} \left( -2p_\alpha(0) + \frac{2}{\alpha}m_\alpha(0) + \frac{1}{\alpha}m_\alpha(2) \right) \\ &= \frac{1}{4\alpha^2} \left( -4 + \frac{1}{\alpha}(\exp(2\alpha) - \exp(-2\alpha)) \right) = \frac{1}{\alpha^2} \left( -1 + \frac{1}{2\alpha} \sinh(2\alpha) \right), \\ b_1 &= B^{(\alpha)}(1) = f_\alpha^{(1)}(1) = \frac{1}{4\alpha^2} \left( p_\alpha(1) - \frac{1}{\alpha}m_\alpha(1) \right) \\ &= \frac{1}{4\alpha^2} \left( \exp(\alpha) + \exp(-\alpha) - \frac{1}{\alpha}(\exp(\alpha) - \exp(-\alpha)) \right) = \frac{1}{2\alpha^2} \left( \cosh(\alpha) - \frac{1}{\alpha} \sinh(\alpha) \right). \end{aligned}$$

It follows that

$$\begin{aligned} b_0 - 2b_1 &= \frac{1}{\alpha^2} \left( -1 + \frac{1}{2\alpha} \sinh(2\alpha) - \cosh(\alpha) + \frac{1}{\alpha} \sinh(\alpha) \right) = \frac{2}{\alpha^3} \cosh\left(\frac{\alpha}{2}\right)^2 (\sinh(\alpha) - \alpha), \\ b_0 + 2b_1 &= \frac{1}{\alpha^2} \left( -1 + \frac{1}{2\alpha} \sinh(2\alpha) + \cosh(\alpha) - \frac{1}{\alpha} \sinh(\alpha) \right) = \frac{2}{\alpha^3} \sinh\left(\frac{\alpha}{2}\right)^2 (\sinh(\alpha) + \alpha). \end{aligned}$$

Now using the fact that  $\cosh(x) \geq 1$  for all  $x \in \mathbb{R}$ , and the Taylor expansion  $\sinh(\alpha) = \sum_{n=0}^\infty \frac{\alpha^{2n+1}}{(2n+1)!}$ , we have

$$\begin{aligned} b_0 - 2b_1 &\geq \frac{2}{\alpha^2} \left( \frac{\sinh(\alpha)}{\alpha} - 1 \right) = \frac{2}{\alpha^2} \left( \sum_{n=0}^\infty \frac{\alpha^{2n}}{(2n+1)!} - 1 \right) = 2 \sum_{n=1}^\infty \frac{\alpha^{2(n-1)}}{(2n+1)!} \\ &\geq 2 \left( \frac{1}{3!} + \frac{\alpha^2}{5!} \right) = \frac{1}{3} \left( 1 + \frac{1}{20} \alpha^2 \right), \end{aligned}$$

where we used the fact that only even powers occur in the sum.

Moreover,

$$\kappa(\alpha) = \frac{b_0 + 2b_1}{b_0 - 2b_1} = \frac{\sinh(\alpha/2)^2(\sinh(\alpha) + \alpha)}{\cosh(\alpha/2)^2(\sinh(\alpha) - \alpha)} = \tanh(\alpha/2)^2 \frac{\sinh(\alpha) + \alpha}{\sinh(\alpha) - \alpha},$$

and in particular  $\kappa(-\alpha) = \kappa(\alpha)$  since  $\tanh(\alpha/2)^2$  is even and  $\sinh$  is odd. To study the behavior of  $\kappa(\alpha)$  we can thus restrict to non-negative values of  $\alpha$ . It clearly holds that  $\kappa(\alpha) \geq 1$  by definition. Moreover  $\lim_{\alpha \rightarrow \infty} \tanh(\alpha) = 1$ , and  $\lim_{\alpha \rightarrow \infty} \frac{\sinh(\alpha)+\alpha}{\sinh(\alpha)-\alpha} = 1$  since  $\sinh$  has a super-linear growth, and thus  $\lim_{\alpha \rightarrow \infty} \kappa(\alpha) = 1$ . The Taylor expansion of  $\kappa(\alpha)$  around zero gives  $\kappa(\alpha) = 3 - \frac{2}{5}\alpha^5 + \mathcal{O}(\alpha^4)$  for small  $\alpha$ , which in turn gives the desired asymptotic in zero.

Finally, we have

$$\kappa'(\alpha) = -\frac{\tanh(\alpha/2)}{\cosh(\alpha/2)^2} \frac{(2 + 2\alpha^2 - 2 \cosh(2\alpha) + \alpha \sinh(2\alpha))}{2(\alpha - \sinh(\alpha))^2}.$$

The denominator of  $\kappa'(\alpha)$  is non-negative. Moreover

$$2 + 2\alpha^2 - 2 \cosh(2\alpha) + \alpha \sinh(2\alpha) \geq 0,$$

and thus  $\kappa'(\alpha)$  has the same sign of  $-\tanh(\alpha/2)$ , i.e.,  $\kappa'(\alpha) \leq 0$  if  $\alpha \geq 0$ . Since  $\lim_{\alpha \rightarrow \infty} \kappa(\alpha) = 1$ , and  $\lim_{\alpha \rightarrow 0} \kappa(\alpha) = 3$ , we have  $1 \leq \kappa(\alpha) \leq 3$ .

**Acknowledgements** We thank the support of the GNCS-INdAM project “Interpolazione e smoothing: aspetti teorici, computazionali e applicativi”. This research has been carried out within the Italian Network on Approximation (RITA) and the thematic group on “Approximation Theory and Applications” of the Italian Mathematical Union (UMI). We sincerely thank the reviewers for helping us to significantly improve the paper.

**Funding** Open access funding provided by Politecnico di Torino within the CRUI-CARE Agreement.

## Declarations

**Conflict of interest** The authors declare no competing interests.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Bos, L., De Marchi, S., Vianello, M.: Polynomial approximation on Lissajous curves in the  $d$ -cube. *Appl. Num. Math.* **116**, 47–56 (2017)
2. Campagna, R., Conti, C.: Penalized hyperbolic-polynomial splines. *Appl. Math. Lett.* **118** (2021). <https://doi.org/10.1016/j.aml.2021.107159>
3. Fasshauer, G.E., McCourt, M.: *Kernel-based Approximation Methods Using MATLAB*. World scientific, Singapore (2015)
4. Wendland, H.: *Scattered Data Approximation*. Cambridge Monographs on Applied and Computational Mathematics, vol. 17. Cambridge University Press, Cambridge (2005)
5. Conti, C., Romani, L., Schenone, D.: Semi-automatic spline fitting of planar curvilinear profiles in digital images using the Hough transform. *Pattern Recogn.* **74**, 64–76 (2018)
6. Bohra, P., Campos, J., Gupta, H., Aziznejad, S., Unser, M.: Learning activation functions in deep (spline) neural networks. *IEEE Open J. Signal Process.* **1**, 295–309 (2020)
7. Unser, M.: A representer theorem for deep neural networks. *J. Machine Learning Res.* **20**, 1–30 (2019)
8. Cohen, E., Riesenfeld, R., Elber, G.: *Geometric Modeling with Splines*. CRC Press, New York (2001)
9. Uhlmann, V., Delgado-Gonzalo, R., Conti, C., Romani, L., Unser, M.: Exponential Hermite splines for the analysis of biomedical images. In: *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2014, Florence, Italy, May 4-9, 2014*, pp. 1631–1634. IEEE, (2014). <https://doi.org/10.1109/ICASSP.2014.6853874>
10. Unser, M., Blu, T.: Cardinal exponential splines: part I - theory and filtering algorithms. *IEEE Trans. Signal Process.* **53**(4), 1425–1438 (2005). <https://doi.org/10.1109/TSP.2005.843700>
11. Campagna, R., Bayona, V., Cuomo, S.: Using local PHS+poly approximations for Laplace transform inversion by Gaver-Stehfest algorithm. *Dolomites Res. Notes Approx.* **13**, 55–64 (2020)
12. Campagna, R., Conti, C., Cuomo, S.: Smoothing exponential-polynomial splines for multiexponential decay data. *Dolomites Res. Notes Approx.* **12**(1), 86–100 (2019)
13. Campagna, R., Conti, C., Cuomo, S.: Computational error bounds for Laplace transform inversion based on smoothing splines. *Appl. Math. Comput.* **383**, 125376 (2020)

14. Brutman, L.: On the Lebesgue function for polynomial interpolation. *SIAM J. Numer. Anal.* **15**, 694–704 (1978)
15. Brutman, L.: Lebesgue functions for polynomial interpolation - a survey. *Ann. Numer. Math.* **4**, 111–127 (1997)
16. Bayliss, A., Turkel, E.: Mappings and accuracy for Chebyshev pseudo-spectral approximations. *J. Comput. Phys.* **101**, 349–359 (1992)
17. Berrut, J.P., Mittelmann, H.D.: Lebesgue constant minimizing linear rational interpolation of continuous functions over the interval. *Comput. Math. Appl.* **33**(6), 77–86 (1997)
18. Bos, L., Caliari, M., De Marchi, S., Vianello, M., Xu, Y.: Bivariate Lagrange interpolation at the Padua points: The generating curve approach. *J. Approx. Theory* **143**(1), 15–25 (2006)
19. Bos, L., De Marchi, S., Hormann, K.: On the Lebesgue constant of Berrut's rational interpolant at equidistant nodes. *J. Comput. Appl. Math.* **236**(4), 504–510 (2011)
20. De Marchi, S., Marchetti, F., Perracchione, E., Poggiali, D.: Multivariate approximation at fake nodes. *Appl. Math. Comput.* **391**, 125628 (2021)
21. Temlyakov, V.N.: Greedy approximation. *Acta Numer* **17**, 235–409 (2008). <https://doi.org/10.1017/S0962492906380014>
22. Haasdonk, B., Santin, G.: Greedy kernel approximation for sparse surrogate modeling. In: Keiper, W., Milde, A., Volkwein, S. (eds.) *Reduced-Order Modeling (ROM) for Simulation and Optimization: Powerful Algorithms as Key Enablers for Scientific Computing*, pp. 21–45. Springer, Cham (2018)
23. Marchi, S.D., Schaback, R., Wendland, H.: Near-optimal data-independent point locations for radial basis function interpolation. *Adv. Comput. Math.* **23**, 317–330 (2005)
24. Santin, G., Haasdonk, B.: Convergence rate of the data-independent  $P$ -greedy algorithm in kernel-based approximation. *Dolomites Res. Notes Approx.* **10**(2), 68–78 (2017)
25. Wirtz, D., Haasdonk, B.: A Vectorial Kernel Orthogonal Greedy Algorithm. *Dolomites Res. Notes Approx.* **6**, 83–100 (2013). [https://doi.org/10.14658/pupj-drna-2013-Special\\_Issue](https://doi.org/10.14658/pupj-drna-2013-Special_Issue)
26. Wenzel, T., Santin, G., Haasdonk, B.: A novel class of stabilized greedy kernel approximation algorithms: Convergence, stability and uniform point distribution. *Journal of Approximation Theory* **262**, 105508 (2021). <https://doi.org/10.1016/j.jat.2020.105508>
27. Wenzel, T., Santin, G., Haasdonk, B.: Analysis of target data-dependent greedy kernel algorithms: Convergence rates for  $f$ -,  $f$ - $P$ - and  $f$ - $P$ -greedy. <https://arxiv.org/abs/2105.07411>, Accepted for publication in *Constructive Approximation* (2021)
28. Dutta, S., Farthing, M.W., Perracchione, E., Savant, G., Putti, M.: A greedy non-intrusive reduced order model for shallow water equations. *J. Comput. Phys.* **439**, 110378 (2021)
29. Seatzu, S.: Un metodo per la costruzione di smoothing splines naturali mono e bidimensionali. *Calcolo* **12**, 259–273 (1975)
30. Stoer, J., Bulirsch, R.: *Introduction to Numerical Analysis vol. 3*. Springer (2002)
31. Atkinson, K.E.: *An Introduction to Numerical Analysis*. John Wiley & Sons (2008)
32. Campagna, R., Conti, C., Cuomo, S.: Data-driven selection of HP-splines frequency parameter. *Manuscript* (2022)
33. Yueh, W.-C.: Eigenvalues of several tridiagonal matrices. *Applied Mathematics E-Notes [electronic only]* **5**, 66–74 (2005)
34. Schaback, R., Wendland, H.: Adaptive greedy techniques for approximate solution of large RBF systems. *Numer. Algorithms* **24**(3), 239–254 (2000). <https://doi.org/10.1023/A:1019105612985>
35. Köppel, M., Franzelin, F., Kröker, I., Oladyshkin, S., Santin, G., Wittwar, D., Barth, A., Haasdonk, B., Nowak, W., Pflüger, D., Rohde, C.: Comparison of data-driven uncertainty quantification methods for a carbon dioxide storage benchmark scenario. *Comput. Geosci.* **23**(2), 339–354 (2019). <https://doi.org/10.1007/s10596-018-9785-x>
36. Romano, A., Campagna, R., Masi, P., Toraldo, G.: NMR data analysis of water mobility in wheat flour dough: A computational approach. In: Sergeev, Y.D., Kvasov, D.E. (eds.) *Numerical Computations: Theory and Algorithms*, pp. 146–157. Springer, Cham (2020)
37. Campagna, R., Perracchione, E.: Feature augmentation for numerical inversion of multi-exponential decay curves. *AIP Conference Proceedings* **2425**(1), 050004 (2022) <https://aip.scitation.org/doi/pdf/10.1063/5.0081505>. <https://doi.org/10.1063/5.0081505>
38. Perracchione, E., Massone, A.M., Piana, M.: Feature augmentation for the inversion of the Fourier transform with limited data. *Inverse Problems* (2021). <https://doi.org/10.1088/1361-6420/ac1ad7>