



Small errors imply large evaluation instabilities

Robert Schaback¹

Received: 18 March 2022 / Accepted: 23 February 2023
© The Author(s) 2023

Abstract

Numerical analysts and scientists working in applications often observe that once they improve their techniques to get a better accuracy, some instability of the evaluation creeps in through the back door. This paper shows for a large class of numerical methods that such a *Trade-off Principle* between error and evaluation stability is unavoidable. It is an instance of a *no free lunch theorem*. Here, *evaluation* is the mathematical map that takes input data to output data. This is independent from the numerical routine that calculates the output. Therefore, *evaluation* stability is different from *computational* stability. The setting is confined to recovery of functions from data, but it includes solving differential equations by writing such methods as a recovery of functions under constraints imposed by differential operators and boundary values. The trade-off principle bounds the product of two terms from below. The first is related to errors, and the second turns out to be related to evaluation instability. Under certain conditions satisfied for splines and kernel-based interpolation, both can be minimized. Then the lower bound is attained, and the error term is the inverse of the instability term. As a byproduct, it is shown that Kansa's Unsymmetric Collocation Method sacrifices accuracy for improved evaluation stability, when compared to symmetric collocation.

Keywords Recovery of functions · Trade-off principle · Kansa method · No free lunch theorem · Collocation · Interpolation

Mathematics Subject Classification (2010) 41Axx · 65D05 · 65D12 · 65D15

Communicated by: Tobin Driscoll

✉ Robert Schaback
schaback@math.uni-goettingen.de

¹ Institut für Numerische und Angewandte Mathematik, Universität Göttingen, Lotzestraße 16–18, Göttingen, 37083, Germany

1 Introduction

After quite some efforts to find kernels that allow small recovery errors and well-conditioned kernel matrices at the same time, the paper [1] proved that this does not work. The result was called ‘‘Uncertainty Relation’’ or ‘‘Trade-off Principle’’ (see, e.g. Holger Wendland’s book [2] of 2005) and received quite some attention in the literature. It is a special case of the ‘‘No free lunch’’ principle. As correctly mentioned by Greg Fasshauer and Michael McCourt in their 2015 book [3], it had quite some negative influence on the development of the field, because it kept users from looking at better bases than those spanned by kernel translates. However, it will turn out here that changes of bases will be helpful in a different way, but not allow an escape from the Trade-off Principle.

Sparked by a question of C.S. Chen of the University of Southern Mississippi in an e-mail dated Dec. 28th, 2021, this paper extends the result of [1] to much more general situations. To avoid the misconceptions implied by [1], the effect of basis changes will be discussed at various places. But most of the results here are independent of choices of bases.

Since the scope of the paper will be quite wide, a good deal of abstraction will be necessary later, and therefore a classical case should be served as starters. Consider interpolation of functions $f \in C^{n+1}[-1, +1]$ on a set X_n of points $-1 \leq x_0 < x_1, \dots < x_n \leq 1$ by polynomials $I_{X_n}(f)$ of degree at most n . The well-known error bound is

$$|f(x) - I_{X_n}(f)(x)| \leq \frac{\|f^{(n+1)}\|_\infty}{(n + 1)!} \prod_{j=0}^n |x - x_j|$$

based on Newton’s formula. We can recast this as an error bound

$$|f(x) - I_{X_n}(f)(x)| \leq P_{X_n}(x) \|f^{(n+1)}\|_\infty \tag{1}$$

in terms of a *Power Function*

$$\begin{aligned} P_{X_n}(x) &:= \sup_{\|f^{(n+1)}\|_\infty \leq 1} |f(x) - I_{X_n}(f)(x)| \\ &= \frac{1}{(n + 1)!} \prod_{j=0}^n |x - x_j|. \end{aligned}$$

If we add a point x to the set X_n , the *Lagrangian* of degree $n + 1$, vanishing on X_n and being one on x is

$$u_{x, X_n}(z) = \prod_{j=0}^n \frac{z - x_j}{x - x_j}$$

with seminorm

$$\|u_{x, X_n}^{(n+1)}\|_\infty = (n + 1)! \prod_{j=0}^n |x - x_j|^{-1},$$

leading to

$$1 = P_{X_n}(x) \cdot \|u_{x, X_n}^{(n+1)}\|_\infty. \tag{2}$$

In a somewhat sloppy formulation, this is a first *Trade-off Principle*:

Small Power Functions lead to large norms of Lagrangians.

To avoid misunderstandings, it should be pointed out that (2) concerns two successive interpolation steps and lets x occur in two different meanings. In the first factor, the error for an interpolation in $n + 1$ points is evaluated at x using a specific norm, but for the second factor the interpolation takes place on $n + 2$ points with x added, and then the corresponding norm of the Lagrangian at x is taken. The error at the new step is not estimated. At some new point y , it would be bounded indirectly by $P_{X_n \cup \{x\}}(y)$ via (1) and connected to $\|u_{y, X_n \cup \{x\}}^{(n+2)}\|_\infty$ by (2) in the next step.

Users will often see interpolation as a sequence of many steps, and if they try to make the first factor go to zero for large n , they will have to face that the second goes to infinity. This goes away from the stepwise viewpoint.

By (2), the point x as the new x_{n+1} for the next step should be placed at the maximum of the Power Function. This *greedy technique* leads to Leja points and will be treated in more detail in Section 12.

The value of the interpolant I_{X_n} at some x is given by

$$I_{X_n}(x) = \sum_{j=0}^n f(x_j) u_{x_j, X_n \setminus \{x_j\}}(x) \tag{3}$$

no matter how numerical calculations work. Absolute errors in the input data will always be multiplied by the Lagrangians to get absolute errors in the values. For any norm chosen,

$$\|I_{X_n}\| \leq \sum_{j=0}^n |f(x_j)| \|u_{x_j, X_n \setminus \{x_j\}}\|$$

lets absolute errors in the input data be multiplied by

$$\sum_{j=0}^n \|u_{x_j, X_n \setminus \{x_j\}}\|$$

to deviations in norms of values, in the worst case. This is why large norms of Lagrangians lead to large evaluation instabilities measured in these norms. Therefore, the Trade-off Principle may be reformulated as

Small errors are connected to evaluation instabilities taken in the same norm.

The paper gives a rigid underpinning of this Trade-off Principle. Of course, users may prefer other formulas for computation of values, in particular barycentric ones by J.P. Berrut and L.N. Trefethen [4]. This is mandatory for univariate polynomial interpolation with large degrees. But no computation can escape from the validity of (3) that shows how absolute input errors will enter into absolute output errors in the worst case, even if a numerical algorithm avoids Lagrangians.

However, the numerical computation of the Lagrangians, if attempted, induces additional instabilities that are ignored here. For polynomials, we already mentioned *barycentric* formulas [4]. For kernel-based recoveries, various methods were invented to cope with instabilities, see the references given in Section 13.

2 Overview

Section 3 sets the stage for general recovery methods including solving differential equations. *Recovery processes* reconstruct functions from *data* given as prescribed values of *linear functionals*, and the evaluation of the result will again be an application of a functional. *Data* can include values of arbitrary linear operators acting on functions, thus rewriting methods for PDE solving as function recoveries.

Section 4 introduces the form of the recoveries considered. Nearest-neighbour methods, optimal recoveries in Hilbert spaces, and regression in machine learning are special cases described in Section 5.

The basic technique used for trade-off principles is outlined in Section 6, still in rather abstract form. Then Section 7 contains the central results, namely trade-off principles that bound the product of norms of errors and norms of certain “bump” functions from below. The lower bounds turn into equalities in case of optimal recoveries later in Section 9. The central notion of *evaluation instability* is explained at length in Section 8, defining it as the worst possible propagation factor of absolute errors in the input data to absolute errors in the function values of the recovery. It is shown that the norms of “bump” functions from Section 7 are closely connected to evaluation instability.

Examples are given in Section 10, including splines and recoveries via expansions like Fourier or Taylor series. The connection to the older trade-off principle from [1] is provided in Section 10.8, followed by extensions to unsymmetric methods like Kansa’s collocation technique. The trade-off principle holds for these as well, but they sacrifice accuracy for evaluation stability. Finally, the implications for greedy adaptive methods are sketched.

3 Data as functionals

A fairly general and useful viewpoint on Numerical Analysis or Computational Mathematics when working on functions is to see *data* of a function as *values of linear functionals*. In particular, differential equations, ordinary or partial, just impose infinitely many restrictions on a function u from some function space U by applying linear functionals. This can be conveniently written as

$$\lambda(u) = f_\lambda \in \mathbb{R} \quad (4)$$

for all functionals λ from a subset Λ of the dual U^* of U , the space of continuous linear functionals on U . The problem is to recover u from the given data f_λ . The specifics of certain differential equation problems involving differential or boundary evaluation operators disappear. And if users have only limited information in the sense of just finitely many data $f_{\lambda_j} \in \mathbb{R}$ for a finite subset $\Lambda_M = \{\lambda_1, \dots, \lambda_M\} \subset \Lambda$, one has to use computational techniques that get along with the available data. This viewpoint is behind the scenes for this paper. Readers should always be aware that differential operators may lurk behind the functionals appearing here.

For illustration, consider a standard Poisson problem

$$\begin{aligned} \Delta u &= f \quad \text{in } \Omega \\ u &= g \quad \text{in } \partial\Omega \end{aligned} \tag{5}$$

on a bounded domain $\Omega \subset \mathbb{R}^2$ for simplicity. The data functionals come in two variations:

$$\begin{aligned} \Delta_i(f) &:= \Delta f(x_i), \quad x_i \in \overline{\Omega}, \quad 1 \leq i \leq M_\Delta \\ \beta_j(f) &= f(y_j), \quad y_j \in \partial\Omega, \quad 1 \leq j \leq M_\beta \end{aligned} \tag{6}$$

caring for the PDE in the domain and for the boundary values. They are finite selections from the obvious infinite sets of functionals that define the true solution analytically. If the analytic problem is well-posed and if the function recovery from the above data is carried out with enough oversampling, this technique produces accurate and convergent approximations to the true solution of the PDE problem [5]. This reference also fits algorithms that solve problems in weak form into this framework, including the Meshless Local Petrov Galerkin approach by S.N. Atluri and T.-L. Zhu [6, 7] and Generalized Finite Element Methods, see the survey by Babuška et al. [8]. The general practical observation is that going for more accuracy causes more instabilities, in a way that will be clarified here.

Also, *evaluation* of functions is the application of a functional $\mu \in U^*$ to some function f . In particular, evaluation of a multivariate derivative D^α at a point x is the application of the functional $\delta_x(D^\alpha f) = f^{(\alpha)}(x)$ in case that $\delta_x D^\alpha$ is continuous on U . If point evaluation is not defined, as in L_2 spaces, but if local integration is feasible, one can evaluate local integral means, as substitutes for point evaluation. This is the standard way to handle problems in weak form in the references cited above.

Summarizing, everything boils down to a matter of functionals. What can we say about $\mu(f)$ if we know all $\lambda(f)$ for all $\lambda \in \Lambda$? In particular, what can we say about $f(x)$ when we know plenty of data $f(x_j)$? Note that this problem is *regression* in a probabilistic context, and it arises in Machine Learning on a large scale, with Big Data given in high-dimensional spaces.

4 Recovery of functions

We now postulate that the recovery of functions f from their data $\Lambda(f) = (\lambda_1(f), \dots, \lambda_n(f))^T \in \mathbb{R}^n$ is a linear map from \mathbb{R}^n into a space U of functions. No matter how it is actually implemented numerically, we assume that it can be mathematically written as

$$f \mapsto R_{a_\Lambda}(f) := a_\Lambda^T \Lambda(f) = \sum_{\lambda_j \in \Lambda} a_{\lambda_j} \lambda_j(f) \quad \text{for all } f \in U \tag{7}$$

via certain functions $a_{\lambda_j} \in U$ that we call *quasi-Lagrangians* because they produce the recovery like Lagrangians, but possibly without exact reproduction of the data. This is connected to the notion of *quasi-interpolation*.

To avoid certain complications, the map $f \rightarrow \Lambda(f) \in \mathbb{R}^n$ is assumed to be surjective, but we allow nonuniqueness of the a_{λ_j} . Note that the recovery map is an abstract

notion that does not imply that the a_{λ_j} are actually calculated. They just describe the way the given input data enter mathematically into the values of the recovery, and they are assumed to exist independent of how the recovery is numerically implemented.

We call the recovery process *interpolatory* if the recovery preserves the data, i.e.

$$\Lambda(R_{a_\Lambda}(f)) = \Lambda(a_\Lambda^T \Lambda(f)) = \Lambda(f) \text{ for all } f \in U,$$

and then $\lambda_k(a_{\lambda_j}) = \delta_{jk}$, $1 \leq j, k \leq n$ holds, i.e. the a_{λ_j} have the Lagrange interpolation property. We shall use the notation $u_{\mu, \Lambda}$ for a *Lagrangian* that satisfies $\Lambda(u_{\mu, \Lambda}) = 0$, $\mu(u_{\mu, \Lambda}) = 1$, and then $a_{\lambda_j} = u_{\lambda_j, \Lambda \setminus \{\lambda_j\}}$ holds in this notation.

Evaluation of the recovery via a functional μ now is

$$\mu(a_\Lambda^T \Lambda(f)) = a_\Lambda^T(\mu) \Lambda(f) = \sum_{\lambda_j \in \Lambda} \mu(a_{\lambda_j}) \lambda_j(f) \tag{8}$$

by defining a vector $a_\Lambda^T(\mu) := \mu(a_\Lambda^T)$ that is a bilinear form in Λ and μ .

One may restrict these maps to sums over neighbours λ of μ , to get more *locality*, and this is what generalized *Moving Least Squares* [9–12] or Finite Elements do. But if theoretically done for all $\mu \in U^*$, this still fits into the above framework. As a prominent example, *Barycentric formulas* by J.P. Berrut and L.N. Trefethen [4] change the way the above formula is calculated, with a significant gain in numerical stability, focusing on neighbours.

For recovery of a single value $\mu(f)$ from single values $\lambda_j(f)$, one can construct a vector of single values $a_{\lambda_j}(\mu)$ such that

$$\mu(f) \approx \sum_{\lambda_j \in \Lambda} a_{\lambda_j}(\mu) \lambda_j(f) \tag{9}$$

without necessarily calculating the a_{λ_j} as functions and taking values $\mu(a_{\lambda_j}) = a_{\lambda_j}(\mu)$ afterwards. In meshless methods (see the early survey by T. Belytschko et al. [13]), the functions a_{λ_j} are called *shape functions*. In the standard approach, they are calculated in many points, and if derivatives are needed for dealing with PDEs, these are taken afterwards or obtained by taking derivatives of the local construction process. In contrast to this, *Direct Moving Least Squares* by D. Mirzaei et al. [14, 15] use an error minimization of (9) for derivative functionals μ without the detour via shape functions.

This presentation looks unduly abstract, but it is not. It considers recovery without any fixed assumptions about how functions are represented, how norms of errors and functions are defined, and how bases are chosen. Therefore, it allows to compare actual numerical strategies on a higher level. It goes back to the input data and considers the output data, as functionals, the actual determination of the recovery map being in a black box. The final goal in this paper is to see whether going for a small error implies some sort of instability whatsoever, and this may be independent of what happens in the black box.

5 Special recovery strategies

When avoiding full functions, the recovery of a single value $\mu(f)$ from given values $\Lambda(f)$ is trivial if $\mu = \lambda$ for some $\lambda \in \Lambda$. In more generality, one would pick the functional $\lambda \in \Lambda$ that is “closest” to μ , and then take $\lambda(f)$ as an approximation of $\mu(f)$. This is the *nearest neighbour* strategy, but it needs distances between functionals, and requires to find the closest neighbour. Cases involving point geometry like nearest neighbours or triangulations will be covered by the theory developed here, but we do not include examples.

If a norm on U^* is available, one can consider the approximation problem to minimize

$$\left\| \mu - \sum_{\lambda \in \Lambda} a_\lambda \lambda \right\|_{U^*}$$

over all coefficients a_λ , denote a solution by $a_\lambda^*(\mu)$ and to approximate μ by

$$\mu_\Lambda^* := \sum_{\lambda \in \Lambda} a_\lambda^*(\mu) \lambda.$$

This avoids functions as well, but it requires norms in the dual space that users can work with.

Special cases are Reproducing Kernel Hilbert spaces. They have a kernel $K : \Omega \times \Omega$ on an abstract set Ω and define an inner product

$$(\lambda, \mu)_{U^*} := \lambda^x \mu^y K(x, y)$$

where the application on x arises as a superscript. Furthermore, each functional λ defines a function

$$f_\lambda(x) := \lambda^y K(y, x) \text{ for all } x \in \Omega$$

and these functions have the inner product

$$(f_\lambda, f_\mu)_U = (\lambda, \mu)_{U^*} = \lambda^x \mu^y K(x, y)$$

making f_λ a Riesz representer of λ . It is then easy to prove that an optimal recovery consists of the vector $a_\lambda^*(\mu)$ that solves the system

$$\begin{pmatrix} (\lambda_1, \lambda_1)_{U^*} & (\lambda_1, \lambda_2)_{U^*} & \dots & (\lambda_1, \lambda_M)_{U^*} \\ (\lambda_2, \lambda_1)_{U^*} & (\lambda_2, \lambda_2)_{U^*} & \dots & (\lambda_2, \lambda_M)_{U^*} \\ \vdots & \vdots & \ddots & \vdots \\ (\lambda_M, \lambda_1)_{U^*} & (\lambda_M, \lambda_2)_{U^*} & \dots & (\lambda_M, \lambda_M)_{U^*} \end{pmatrix} \begin{pmatrix} a_{\lambda_1}^*(\mu) \\ a_{\lambda_2}^*(\mu) \\ \vdots \\ a_{\lambda_M}^*(\mu) \end{pmatrix} = \begin{pmatrix} (\lambda_1, \mu)_{U^*} \\ (\lambda_2, \mu)_{U^*} \\ \vdots \\ (\lambda_M, \mu)_{U^*} \end{pmatrix}$$

with a *kernel matrix* that usually is positive definite.

This looks theoretical again, but it applies to Sobolev spaces, having Whittle-Matérn kernels, and therefore it is useful for solving PDE problems by recovery of functions from PDE data. This recovery strategy has various optimality properties [16] that we skip over here. See details on kernel-based methods in books by M.D. Buhmann [17], H. Wendland [2], and G. Fasshauer/M. McCourt [3].

It also applies to Machine Learning [18]. On a general set Ω , one has *feature maps* φ_n that map the abstract objects x to a real value like cost or weight or area. The

kernel then is

$$K(x, y) = \sum_n \rho_n \varphi_n(x) \varphi_n(y)$$

with positive weights ρ_n and the inner product

$$(\lambda, \mu) := \sum_n \frac{\lambda^x \varphi_n(x) \mu^y \varphi_n(y)}{\rho_n}$$

lets the above machinery work for regression, but details are omitted. Combining the cases above, Machine Learning can “learn” the solution of a PDE using this framework.

6 Dual trade-off principles

Throughout we shall assume that norms in U and U^* are defined and connected via the suprema

$$\|f\|_U = \sup_{0 \neq \mu \in U^*} \frac{\mu(f)}{\|\mu\|_{U^*}}, \quad \|\mu\|_{U^*} = \sup_{0 \neq f \in U} \frac{\mu(f)}{\|f\|_U}. \tag{10}$$

Take a functional $\mu \in U^*$ and imagine that it is evaluating an error of a recovery process. Then

$$1 \leq \|\mu\|_{U^*} \cdot \|f_\mu\|_U$$

holds for all functions $f_\mu \in U$ with $\mu(f_\mu) = 1$. If the error $\|\mu\|_{U^*}$ of the recovery process is small, the norms of the functions f_μ must be large. In Section 8, we shall connect these norms to the instability of the evaluation of function values of the recovery.

Of course, there also is a dual version

$$1 \leq \|f\|_U \cdot \|\mu_f\|_{U^*}$$

for all functions $f \in U$ and all functionals $\mu_f \in U^*$ with $\mu_f(f) = 1$. In Hilbert spaces, one can minimize the second factors under the given constraint, and the minimum is realized by Riesz representers.

Note that the above inequalities turn into equalities if the suprema in (10) are attained for the functions or functionals in the second factors. Details and applications will follow below.

It is essential that the two norms in the above inequalities are dual to each other. If $\|\cdot\|_U$ allows and penalizes high derivatives, the functionals in U^* will allow high derivatives as well. Users might want the functional factor and the function factor to use non-dual norms, but this is a quite different story.

These trade-off principles differ from certain standard algebraic ones involving inversions like $1 \leq \|A\| \|A^{-1}\|$ for square nonsingular matrices A . But

$$|(x, y)_2| \leq \|x\|_2 \|y\|_2 \tag{11}$$

for vectors is a special case, because one of the vectors can be seen as a functional acting on the other. If generalized to variances and a covariance or commutator,

the latter case is behind the Heisenberg Uncertainty Principle after a few steps of generalization.

7 Power functions and bump functions

We assume to have a finite set $\Lambda \subset U^*$ of functionals to recover functions $f \in U$ from their data $\Lambda(f)$ via a recovery (7), and we evaluate the result by applying a functional $\mu \notin \Lambda$ to f .

Definition 1 The norm

$$P_{a_\Lambda}(\mu) := \|\mu - \mu(R_{a_\Lambda})\Lambda\|_{U^*} \tag{12}$$

is called the *Generalized Power Function*.

It leads to an error bound

$$|\mu(f) - \mu(R_{a_\Lambda})(f)| \leq P_{a_\Lambda}(\mu)\|f\|_U \text{ for all } f \in U, \mu \in U^* \tag{13}$$

and this is why we use it to deal with the recovery error.

Definition 2 A bump function $f_{\mu,\Lambda} \in U$ satisfies $\mu(f_{\mu,\Lambda}) = 1$ and $\Lambda(f_{\mu,\Lambda}) = 0$.

Theorem 1 For any any functional $\mu \in U^*$ that has a bump function $f_{\mu,\Lambda}$, the trade-off principle

$$1 \leq P_{a_\Lambda}(\mu)\|f_{\mu,\Lambda}\|_U \tag{14}$$

holds, and

$$\frac{1}{\inf_{a_\lambda} P_{a_\Lambda}(\mu)} \leq \inf_{\mu(f)=1, \Lambda(f)=0} \|f\|_U \tag{15}$$

relates the minimal possible recovery error to the minimal possible norm of a bump function.

Proof Just insert a bump function into (13). □

Remark Power Functions, bump functions, and Lagrangians are independent of bases. There is no escape from the Trade-off Principle by any change of basis, as long as the recovery map or the space U are kept fixed.

Remark Furthermore, recoveries, bump functions, and Lagrangians can be defined without using norms or spaces. These come up when going over to a Power Function and a norm of a bump function. Therefore, Theorem 1 does not only cover all possible recoveries, but also all ways to handle errors and evaluation instability for these by defining norms afterwards. Furthermore, (14) is *local* in the sense that it holds for each specific μ . The right-hand side will vary considerably with μ , up to the limit $1 \leq 0 \cdot \infty$ in the excluded case $\mu \in \Lambda$.

While (14) is an *add-one-in* version, a special *leave-one-out* version is

$$1 \leq P_{a_{\Lambda \setminus \{\lambda\}}}(\lambda) \|f_{\lambda, \Lambda \setminus \{\lambda\}}\|_U \tag{16}$$

if a bump function $f_{\lambda, \Lambda \setminus \{\lambda\}}$ is available. And if the recovery is interpolatory, using a Lagrangian $u_{\lambda, \Lambda \setminus \{\lambda\}}$, we get

$$1 \leq P_{\Lambda \setminus \{\lambda\}}(\lambda) \|u_{\lambda, \Lambda \setminus \{\lambda\}}\|_U. \tag{17}$$

When using the leave-one-out version, it is a pitfall to assume that the recovery $a_{\Lambda \setminus \{\lambda\}}$ arises from deleting the component a_λ from a_Λ . The other components will still depend on all functionals in Λ .

8 Evaluation instability

Consider the evaluation of a linear formula like (8) when users have to take the data $\lambda_j(f)$ as they are. Even if the numerical calculations are performed differently and without roundoff, any absolute error in the input data will be multiplied by

$$\sum_{\lambda_j \in \Lambda} |\mu(a_{\lambda_j})| \tag{18}$$

in the worst case. If the result is as large as the recovery error, the recovery may be too polluted to be useful.

Definition 3 Evaluation stability is the worst-case propagation factor for absolute errors in the input data to absolute errors in the output values of the recovery.

To avoid misunderstandings, it should be pointed out again that evaluation stability is independent of the recovery error. And it is independent of how the recovery is numerically implemented. Evaluation instability may cause absolute worst-case errors in recovery values that are much larger than machine precision, even in the order of the attainable recovery error. Then this looks like preventing further convergence. This effect is well-known from Runge phenomenon examples that may be exponentially convergent in theory, but useless numerically.

If the above functional is a point evaluation $\mu(f) = f(x)$, one can call (18) a *pointwise evaluation instability factor*. If the functional μ varies in the dual U^* of a function space U , the left-hand side of

$$\sup_{\mu \in U^*, \|\mu\|_{U^*} \leq 1} \sum_{\lambda_j \in \Lambda} |\mu(a_{\lambda_j})| \leq \sum_{\lambda_j \in \Lambda} \|a_{\lambda_j}\|_U \tag{19}$$

is the *evaluation norm instability* for the recovery measured in U^* . If U contains functions with high derivatives, this also accounts for the instability of derivatives.

Assume that the numerical recovery is not using the formula (8) that is directly based on the data. If some basis v_1, \dots, v_n is chosen for n data $f(x_1), \dots, f(x_n)$ and the linear system

$$\sum_{j=1}^n c_j v_j(x_i) = f(x_i), \quad 1 \leq i \leq n \tag{20}$$

is solved for coefficients c_1, \dots, c_n , the evaluation formula now is

$$x \mapsto \sum_{j=1}^n c_j v_j(x). \tag{21}$$

The worst-case amplification factor for absolute errors in this evaluation formula will be

$$\sum_{j=1}^n |c_j| \|v_j\|_U$$

in the norm chosen on U . This does not account for the possible stability problems when solving the system (20) for the coefficients c_1, \dots, c_n .

No matter how exact the c_j are, the above sum will be intolerably large if the $|c_j|$ are intolerably large. This frequently happens for badly chosen bases, e.g. for monomials and translates of kernels. But the evaluation stability of (8) often comes out much smaller, e.g. in L_∞ for Chebyshev polynomials on Chebyshev points, or for kernel-based interpolation [19]. At this point, the chosen norm comes into play. If (19) is evaluated for kernel-based interpolation using the native space U , the results of [1] and Section 10.8 usually give very large results, while [19] deals with the L_∞ norm and proves a good evaluation stability via boundedness of Lagrangians on asymptotically uniformly placed data locations.

We add two other ways to see how norms of Lagrangians enter into the evaluation stability. The norm of the interpolation as a map from data to functions in U is blown up for large U -norms of Lagrangians due to

$$\begin{aligned} \max_{\lambda \in \Lambda} \|u_{\lambda, \Lambda \setminus \{\lambda\}}\|_U &\leq \sup_{0 \neq f \in U} \frac{\|R_{a_\Lambda}(f)\|_U}{\|\Lambda(f)\|} \\ &= \|R_{a_\Lambda}\|_{\Lambda(U), U} \\ &\leq \sup_{0 \neq f \in U} \frac{\|\sum_{\lambda \in \Lambda} \lambda(f) u_{\lambda, \Lambda \setminus \{\lambda\}}\|_U}{\|\Lambda(f)\|} \\ &\leq (\max_{\lambda \in \Lambda} \|u_{\lambda, \Lambda \setminus \{\lambda\}}\|_U) (\sum_{\lambda \in \Lambda} \|\lambda\|_{U^*}). \end{aligned}$$

Summing up (17) in the interpolatory case, we get a trade-off principle

$$|\Delta| \leq \|P_{\Lambda \setminus \{\lambda\}}(\lambda)\|_{p, \mathbb{R}^{|\Lambda|}} \| \|u_{\lambda, \Lambda \setminus \{\lambda\}}\|_U \|_{q, \mathbb{R}^{|\Lambda|}}$$

that lets the final factor grow when the error is small. Here, the norms in $\mathbb{R}^{|\Lambda|}$ are running over the $\lambda \in \Lambda$, and we allow $1/p + 1/q = 1$.

If we repeat the above argument in more generality using bump functions,

$$\|R_{a_\Lambda}(f_{\lambda, \Lambda \setminus \{\lambda\}})\|_U = \|a_\lambda\|_U \leq \|R_{a_\Lambda}\| \|f_{\lambda, \Lambda \setminus \{\lambda\}}\|_U$$

shows that control over norms of bump functions implies control over the $\|a_\lambda\|_U$, and we saw above that these blow up absolute errors in the input data. Going on like above, the evaluation is bounded above by

$$\begin{aligned}
 |\mu(R_{a_\Lambda})(f)| &= \left| \sum_{\lambda \in \Lambda} \mu(a_\lambda) \lambda(f) \right| \\
 &\leq \|\lambda(f)\|_{p, \mathbb{R}^{|\Lambda|}} \|\mu(a_\lambda)\|_{q, \mathbb{R}^{|\Lambda|}} \\
 &\leq \|\lambda(f)\|_{p, \mathbb{R}^{|\Lambda|}} \|\mu\|_{U^*} \|\|a_\lambda\|_U\|_{q, \mathbb{R}^{|\Lambda|}}
 \end{aligned}
 \tag{22}$$

with $1/p + 1/q = 1$ and where the norms on $\mathbb{R}^{|\Lambda|}$ run over the λ values. The bound factors into the linear influence of f and μ and keeps the final factor as something like a Lebesgue constant.

In a somewhat sloppy formulation, we summarize the above arguments as a trade-off principle

Small errors imply large evaluation instabilities

that tacitly assumes that errors are measured via norms in U^* while evaluation instabilities are measured via norms in U , the two being dual in the sense of Section 6.

Dealing with a non-dual situation, in particular with a weaker notion of evaluation instability, requires much more machinery. A typical case is L_∞ evaluation stability governed by L_∞ norms of Lagrangians for point evaluation data $\lambda_j(f) = f(x_j)$. This is the standard path along Lebesgue functions and Lebesgue constants. For univariate polynomial interpolation, this approach reveals that equidistant points have an exponential L_∞ instability, while Chebyshev-distributed points only have a logarithmic L_∞ instability. For kernel-based interpolation of function values, [19] proved uniform boundedness of Lagrangians if point sets are asymptotically uniformly distributed and if kernels have finite smoothness.

Here is a caveat. As long as there is no reasonable L_∞ error bound yielding a Power Function, Theorem 1 will be useless. There is no strict Trade-off Principle under these circumstances. However, looking at evaluation stability without a bound on the L_∞ error still gives very useful information.

Remark Certain numerical techniques factorize the map $R_{a_\Lambda} = a_\Lambda^T \Lambda$ from (7) like $R_{a_\Lambda} = S_\Lambda \circ T_\Lambda$. This occurs for changes of bases, e.g. the transition to Newton bases for polynomials or kernel-based recovery [30]. The norm-based evaluation instability will only change by

$$\|R_{a_\Lambda}\| \leq \|S_\lambda\| \cdot \|T_\Lambda\|$$

and not produce any improvement. The advantages lie in different numerical algorithms that may have a better numerical stability.

9 Optimal recovery of functions

Like in the introductory example, there are interpolatory cases where the Lagrangian satisfies (14) and (17) with equality. Then, by (15), the norms of the Lagrangians are minimal under the norms of all bump functions, and the recovery process has minimal

error. Though simple to prove, this seems to be a novel insight. Under certain conditions satisfied for splines and kernel-based interpolation, this holds systematically and in both directions:

Theorem 2 *Assume that an interpolatory recovery process R_{a_Λ} satisfies minimum-norm and bounded-error properties, i.e.*

$$\|R_{a_\Lambda}(f)\|_U = \inf_{g \in U, \Lambda(f) = \Lambda(g)} \|g\|_U \leq \|f\|_U \text{ and } \|f - R_{a_\Lambda}(f)\|_U \leq \|f\|_U \text{ for all } f \in U.$$

Then (14) and (15) hold with equality. The Lagrangian is the minimum-norm bump function, and the recovery process has a minimal Power Function.

Proof We reformulate the Power Function via

$$\begin{aligned} P_{a_\Lambda}(\mu) &= \sup_{\|f\|_U \leq 1} |\mu(f) - \mu(R_{a_\Lambda}(f))| \\ &= \sup_{\|f - R_{a_\Lambda}(f)\|_U \leq 1} |\mu(f) - \mu(R_{a_\Lambda}(f))| \end{aligned}$$

where \geq follows from replacing f by $f - R_{a_\Lambda}(f)$, and \leq follows from the minimum error property, because the set of $\|f - R_{a_\Lambda}(f)\|_U \leq 1$ contains the set of $\|f\|_U \leq 1$. Then we go on by

$$P_{a_\Lambda}(\mu) = \sup_{\|g\|_U \leq 1, \Lambda(g) = 0} |\mu(g)|$$

and note that g can be written as $\mu(g)f_{\mu,\Lambda}$ for an arbitrary bump function $f_{\mu,\Lambda}$. Now

$$\begin{aligned} P_{a_\Lambda}(\mu) &= \sup_{\|f_{\mu,\Lambda}\|_U \leq 1/|\mu(g)|} |\mu(g)| \\ &= \frac{1}{\min_{\|f_{\mu,\Lambda}\|_U} 1} \\ &= \frac{1}{\|u_{\mu,\Lambda}\|_U} \end{aligned}$$

where the final line follows from the minimum norm property. □

10 Examples of interpolatory recoveries

This section illustrates how the Trade-off Principle works under various circumstances. For the univariate cases in this section, we treat interpolation in $[-1, +1]$ of values $f(x_j)$ of functions f on points x_0, \dots, x_n and evaluation at some point x . The functionals are δ functionals, and everything can be expressed via the points. Other cases stay with the original formulation via general functionals λ_j .

10.1 Connect-the-dots

The simplest univariate case is *connect-the-dots* piecewise linear interpolation, but it has no choice of a space U yet. The simplest is $U = C[-1, +1]$ under the sup norm, the Lagrangians being hat functions, with constant extensions to the boundary, if

boundary points are not given. Then there is no proper error bound, and (17) consists of all ones.

If we keep the interpolation method as is, we can go over to zero boundary values and $U = C_0^1[-1, +1]$ under the sup norm of the first derivative. An add-one-in Lagrangian $u_{x, x_k, x_{k+1}}$ based on three adjacent increasingly ordered points $x_k < x < x_{k+1}$ will have the norm $(\min(x_{k+1} - x, x - x_k))^{-1}$. The add-one-in Power Function on some $x \in [x_k, x_{k+1}]$ is

$$\begin{aligned} P_{a_\Delta}(x) &= \sup_{\|f'\|_\infty \leq 1} |f(x) - R_{a_\Delta}(f)(x)| \\ &= \sup_{\|f'\|_\infty \leq 1} \left| \frac{x_{k+1}-x}{x_{k+1}-x_k} (f(x) - f(x_k)) + \frac{x-x_k}{x_{k+1}-x_k} (f(x) - f(x_{k+1})) \right| \\ &= 2 \frac{(x_{k+1}-x)(x-x_k)}{x_{k+1}-x_k} \end{aligned}$$

and for the Trade-off Principle we get

$$\frac{P_{a_\Delta}(x)}{\|u_{x, x_k, x_{k+1}}\|_{C_0^1[-1, +1]}} = 2 \frac{(x_{k+1}-x)(x-x_k)}{(x_{k+1}-x_k) \min(x_{k+1}-x, x-x_k)} = 2 \frac{\max(x_{k+1}-x, x-x_k)}{x_{k+1}-x_k}$$

being between 1 and 2.

If U takes the L_2 norm of first derivatives, we are in a standard spline situation [20] and Theorem 2 applies. This illustrates that the Trade-off Principle works locally and for all possible norms when the recovery problem is fixed to be connect-the-dots.

10.2 Taylor data

Here is a rather academic but mathematically interesting case. Take a space U of univariate real-valued functions on $(-1, +1)$ that have complex extensions being analytic in the unit disc, and consider Taylor data functionals $\lambda_j(f) := f^{(j)}(0)/j!$ for $j \geq 0$. Then write the functions by their Taylor series

$$f(z) = \sum_{n \geq 0} \lambda_n(f) z^n$$

and define a norm in U by

$$\|f\|_U^2 = \sum_{n \geq 0} \frac{\lambda_n(f)^2 (n!)^2}{\rho_n}$$

where the positive weights ρ_j satisfy the constraint

$$\sum_{n \geq 0} \frac{\rho_n}{(n!)^2} < \infty. \tag{23}$$

This generates a Hilbert space of functions whose reproduction formula is the Taylor series, see [21] for plenty of examples, including Hardy and Bergman spaces. The interpolation here is just a partial sum of the Taylor series, while it works by kernel translates in [21].

Now the monomials z^j are the Lagrangians for the λ_j , with norms $\|z^j\|_U^2 = (j!)^2/\rho_j$. And Theorem 2 holds because we just chop the Taylor series. Consequently,

inequalities (14) and (17) are satisfied by equality, and we also know that the Power Function in the leave-last-out form is

$$P_{\Lambda_{k-1}}(\lambda_k) = \sqrt{\rho_k}/k!$$

We treat the add-one-in case in the next section.

Note that all cases will behave like that if they take expansions into series of Lagrangians as their underlying space U , with weights for the expansion coefficients.

10.3 Orthogonal series

Now assume that a space U carries an inner product and allows an orthonormal basis u_0, u_1, \dots , while the data functionals are $\lambda_j(u) = (u, u_j)_U$. Again, the Lagrangians are the expansion basis, and we have Fourier series as a prominent example. If only such functionals are considered, this is a trivial case, because all Lagrangians and Power Functions have norm one.

Now let μ be a different functional, and we construct the norm-minimal bump function of the form

$$f_{\mu, \Lambda_n} = \sum_{j>n} c_j u_j, \quad c_j = (f_{\mu, \Lambda_n}, u_j)_U.$$

Under the constraint $\mu(f_{\mu, \Lambda_n}) = 1$, we have to minimize

$$\|f_{\mu, \Lambda_n}\|_U^2 = \sum_{j>n} c_j^2,$$

and by standard optimization arguments this results in

$$c_j = \frac{\mu(u_j)}{\sum_{k>n} \mu(u_k)^2}, \quad \|u_{\mu, \Lambda_n}\|_U^2 = \left(\sum_{j>n} \mu(u_j)^2 \right)^{-1},$$

yielding the byproduct

$$P_{\Lambda_n}(\mu)^2 = \sum_{j>n} \mu(u_j)^2$$

because Theorem 2 holds.

10.4 Recovery in Hilbert spaces

Assume that a set Λ of linearly independent data functionals $\lambda_1, \dots, \lambda_n$ from the dual U^* of some Hilbert space U is given. This approach contains generalized finite elements using $U = H_0^1(\Omega)$ and $\lambda_j(f) = (\nabla f, \nabla v_j)_2 = (f, v_j)_U$ for test functions v_j . Symmetric kernel-based collocation methods for solving PDEs are also covered [16]. It satisfies Theorems 1 and 2 after minimization in (15). The optimal recovery by optimizing the left-hand side of (15) necessarily uses the subspace spanned by the Riesz representers of the λ_j . The optimized bump functions from the right-hand side

are the Lagrangians in such spaces. These are classical optimality properties of the Rayleigh-Ritz technique. Like in Section 10.3, the optimal situation has

$$P_{\Lambda}^2(\mu) = \|\mu\|_{U^*}^2 - \sum_{j=1}^M (\mu, \lambda_j)_{U^*}^2 = \|u_{\mu, \Lambda}\|_U^{-2}$$

if the λ_j are assumed to be orthonormalized. Details are left to the reader.

10.5 Splines

These are cases where Theorem 2 applies, if they are written in their Hilbert space context. Power Functions can be calculated via reciprocals of norms of Lagrangians. But the theory of this paper allows plenty of nonstandard approaches to splines as well, using different norms.

10.6 Polynomial interpolation

Here, the choice of the space U needs special treatment, but we keep the data being values at points x_0, \dots, x_n forming a set X_n to enable exact interpolation by polynomials of degree n or order $n + 1$.

The classical way to deal with this is to take $U = C^{n+1}[-1, +1]$ and to concentrate on the $(n + 1)$ -st derivative only, i.e. taking the seminorm $\|f\|_U := \|f^{(n+1)}\|_{\infty}$. This brings us back to the introductory example in Section 1. See how (2) works locally, up to the limit $1 = 0 \cdot \infty$ in case $x \in X_n$. Choosing other norms will lead to different results. When restricted to polynomials, the next section treats a special case.

10.7 Norms via expansions

Our univariate model case here is dealing with functions in `chebfun` style (T. Driscoll et al. [22]), where U is a space of functions on $[-1, +1]$ having expansions

$$f_c(x) = \sum_{j=0}^{\infty} c_j T_j(x)$$

into Chebyshev polynomials, and where the norm takes nonnegative weights w_j of the coefficients, e.g.

$$\|f_c\|_U^2 = \sum_{j=0}^{\infty} c_j^2 w_j.$$

But it should be clear that one can use other expansions as well, including multivariate cases.

This is a reproducing kernel Hilbert space setting in disguise by

$$(f_c, f_b)_U := \sum_{j=0}^{\infty} c_j b_j w_j$$

and

$$(\lambda, \mu)_{U^*} := \sum_{j=0}^{\infty} \frac{\lambda(T_j)\mu(T_j)}{w_j}$$

and the kernel is

$$K(x, y) := \sum_{j=0}^{\infty} \frac{T_j(x)T_j(y)}{w_j}.$$

This holds in general, but for the Chebyshev case this is a periodic setting in disguise, because of

$$K(\cos \varphi, \cos \psi) = \sum_{j=0}^{\infty} \frac{\cos(j\varphi) \cos(j\psi)}{w_j}.$$

If we would treat this like in kernel-based spaces, interpolation would be done by linear combinations of non-polynomial functions $K(x, x_k)$, and it would be norm-minimal and error-minimal. Taking $w_j = 0$ for $j > n$ in the Chebyshev case, this falls back to polynomial interpolation of degree n using a basis of

$$K(x, x_k) := \sum_{j=0}^n \frac{T_j(x_k)}{w_j} T_j(x), \quad 0 \leq k \leq n.$$

The Lagrangians and the Power Functions are invariant to basis changes, and the kernel-based viewpoint shows that Theorem 2 holds. This opens an easy access to the Power Function via the reciprocal of the U -norm of the Lagrangian.

In general, by solving

$$\delta_{ij} = \sum_{k=0}^n a_{ik} \lambda_j(T_k) = \lambda_j(u_i), \quad 0 \leq i, j \leq n$$

one gets the expansion coefficients a_{ik} of the Lagrangians u_i , and then the reciprocal of

$$\|u_i\|_U^2 = \sum_{j=0}^n a_{ij}^2 w_j$$

gives the square of the leave-one-out Power Function on the left-out point x_i . To get a add-one-in Power function, start with $n - 1$ points and add another point x .

In the Chebyshev situation handling only point evaluations, the Chebyshev-Vandermonde matrix $T_k(x_j)$ is particularly well-behaving if the points are Chebyshev-distributed, as extrema of T_n or zeros of T_{n+1} .

But note that the above approach applies to all expansion-based spaces where the generating functions T_k are not the Lagrangians of the data functionals λ_j . The crucial matrix has entries $\lambda_j(T_k)$ and is a generalized Vandermonde matrix with possibly awful behaviour.

We want to check the add-one-in Power function

$$\begin{aligned}
 P_{a_\Lambda}(\mu) &= \sup_{\|f\|_U \leq 1} \left| \mu(f) - \sum_{j=0}^n \lambda_j(f) \mu(u_j) \right| \\
 &= \sup_{\|f\|_U \leq 1} \left| \sum_{k=0}^{\infty} c_k \left(\mu(T_k) - \sum_{j=0}^n \lambda_j(T_k) \mu(u_j) \right) \right| \\
 &= \sup_{\|f\|_U \leq 1} \left| \sum_{k=0}^{\infty} c_k \mu(\epsilon_k) \right|
 \end{aligned}$$

with

$$\epsilon_k = T_k - \sum_{j=0}^n \lambda_j(T_k) u_j, \quad k \geq 0$$

being the error on T_k . We see that $\epsilon_k = 0$ for $k \leq n$ and can proceed to optimize under the constraint

$$\|f\|_U^2 = \sum_{k=0}^{\infty} c_k^2 w_k \leq 1$$

to get

$$P_{\Lambda}^2(\mu) = \sum_{k=n+1}^{\infty} \frac{\mu(\epsilon_k)^2}{w_k} \geq \frac{\mu(\epsilon_{n+1})^2}{w_{n+1}}.$$

The add-one-in Lagrangian $u_{\mu, \Lambda}$ arises as

$$\epsilon_{n+1} = u_{\mu, \Lambda} \mu(\epsilon_{n+1})$$

and if rewritten as

$$\epsilon_{n+1} = T_{n+1} - \sum_{k=0}^n T_k \underbrace{\sum_{j=0}^n \lambda_j(T_{n+1}) a_{jk}}_{c_{k,n+1}}$$

the norm is

$$\|\epsilon_{n+1}\|^2 = w_{n+1} + \sum_{k=0}^n c_{k,n+1}^2 w_k \geq w_{n+1}$$

such that

$$\|u_{\mu, \Lambda}\|^2 = \frac{\|\epsilon_{n+1}\|^2}{\mu(\epsilon_{n+1})^2} \geq \frac{w_{n+1}}{\mu(\epsilon_{n+1})^2}$$

satisfies (14), but not necessarily with equality, except when only w_{n+1} is nonzero.

If one uses 11 interpolation points and an additional point at -0.9056 , Fig. 1 shows norm-minimal bump functions and Lagrangians. The bump functions used Chebyshev polynomials up to order 121 to get leeway for norm minimization, and the weights on the T_j were $w_j = (j + 1)^2$. The left plot is for Chebyshev points, the right for equidistant points. The norms of Lagrangians versus bump functions were 4.43 versus 2.43 for equidistant points, and 19.47 versus 3.3 for Chebyshev points. The product of the Power Function with the norm of the bump functions came out as 2.62 and 1.30 instead of one.

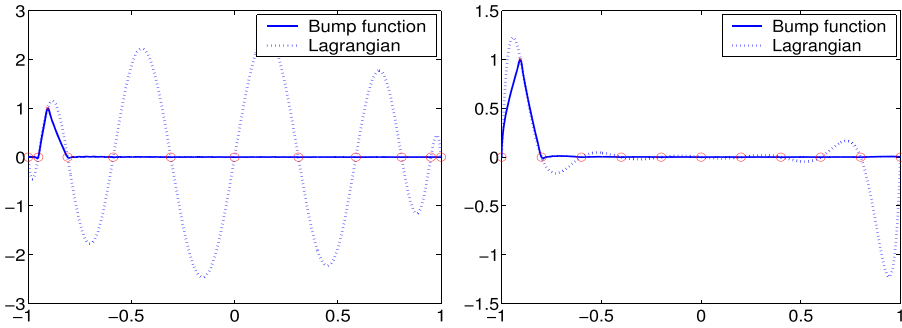


Fig. 1 Bump functions and low-order Lagrangians for Chebyshev(left) and equidistant points

10.8 Kernel-based recovery problems

The goal of this section is to include the Trade-off Principle of [1] as a special case, though it looks different, considering eigenvalues of kernel matrices there.

Assume a generalized interpolation using a set $\Lambda := \{\lambda_1, \dots, \lambda_N\}$ of linearly independent functionals and the trial space $H_{\Lambda, K} := \{\lambda^x K(x, \cdot) : \lambda \in \Lambda\}$ for a positive definite kernel K on a set Ω . The kernel matrix $A_{\Lambda, K}$ has entries

$$\lambda_j^x \lambda_k^y K(x, y) = (\lambda_j, \lambda_k)_{H_K^*}, \quad 1 \leq j, k \leq N$$

in the inner product of the dual H_K^* of the native space H_K for K and is positive definite. By standard arguments from Optimal Recovery in Reproducing Kernel Hilbert Spaces, Theorem 2 holds, and we have a Trade-off Principle in the form (17) for Lagrangians with equality.

But the result of [1] looks different. To see the connection, recall that the squared Power Function for the add-one-in situation is the quadratic form

$$\begin{pmatrix} 1 \\ -\mu(u_1) \\ \vdots \\ -\mu(u_N) \end{pmatrix}^T \begin{pmatrix} (\mu, \mu) & (\mu, \lambda_1) & \dots & (\mu, \lambda_N) \\ (\lambda_1, \mu) & (\lambda_1, \lambda_1) & \dots & (\lambda_1, \lambda_N) \\ \vdots & \vdots & \ddots & \vdots \\ (\lambda_N, \mu) & (\lambda_N, \lambda_1) & \dots & (\lambda_N, \lambda_N) \end{pmatrix} \begin{pmatrix} 1 \\ -\mu(u_1) \\ \vdots \\ -\mu(u_N) \end{pmatrix}$$

where the u_j are the Lagrangians. The proof of the trade-off principle in [1] proceeds via the smallest eigenvalue of the matrix and ignores norms of Lagrangians. Furthermore, it is just an inequality in its original form, while Theorem 2 yields an equation and is much more general.

To see how [1] could have proven equality in (17) more than 25 years earlier, we look at the connection now. This requires to identify the quadratic form with the reciprocal of $\|u_{\mu, \Lambda}\|_U^2$. Consider the function u based on the extended set of functionals and with coefficients $(1, -\mu(u_1), \dots, -\mu(u_N))^T$. Then the matrix-vector product above gives the data, and the quadratic form above is the U -norm squared. The function is

$$u(x) = \mu^y K(y, x) - \sum_j \mu(u_j) \lambda_j^y K(y, x),$$

and it is the function where the alternative form

$$P_{a_\Lambda}(\mu) = \sup_{\|u\|_U \leq 1, \Lambda(u)=0} \mu(f)$$

of the Power function attains its supremum, up to a factor [23]. Thus, $\Lambda(u) = 0$ and

$$P_{a_\Lambda}(\mu) = \frac{\mu(u)}{\|u\|_U}.$$

But the above discussion shows that $P_{a_\Lambda}^2(\mu) = \|u\|_U^2$, proving $\mu(u) = P_{a_\Lambda}^2(\mu)$. This implies $u = P_{a_\Lambda}^2(\mu)u_{\mu,\Lambda}$ and

$$P_{a_\Lambda}(\mu)^2 = \|u\|_U^2 = P_{a_\Lambda}^4(\mu)\|u_{\mu,\Lambda}\|_U^2$$

to arrive finally at the Trade-off Principle in the form

$$1 = P_{a_\Lambda}^2(\mu)\|u_{\mu,\Lambda}\|_U^2.$$

Remark As long as the recovery map a_Λ , the evaluation functionals and the chosen space U are fixed, there is no escape from the Trade-off Principle in the above form by changes of bases, because both ingredients are basis-independent. Of course, basis changes help, but in a different way. For instance, choosing a special basis and proceeding via (21) may give a much better L_∞ evaluation stability than choosing simpler bases. The Trade-off Principle for strong norms is unaffected. In general, one can sacrifice small errors for better evaluation stability by changing the recovery strategy or the chosen norms.

Remark Once the functionals are fixed, one can vary the kernel, with respect to smoothness and scale. The Trade-off Principle will hold as an equality in all cases if the norm is defined via the native space for the kernel and the scale.

11 Unsymmetric case

The previous two sections still used interpolation and Lagrangians. But there are much more general cases, e.g. for PDE solving by unsymmetric meshless methods. In the latter case, users have no freedom to choose the data functionals, because they are prescribed by the PDE to be solved. The functionals will generate boundary values or values of the differential operator in the interior. We further assume that the user prefers a certain sort of trial functions that should finally approximate the true PDE solution very well. In cases with well-posedness in the sense of Real Analysis, it suffices to come up with such a solution even if there is no uniqueness of the recovery procedure [5].

Before we look at the trial space, recall that optimal Power Functions are purely dual objects,

$$P_\Lambda(\mu) = \min_{a \in \mathbb{R}^{|\Lambda|}} \|\mu - \sum_{\lambda \in \Lambda} a_\lambda \lambda\|_{U^*},$$

not depending on trial spaces, and will always outperform other solutions, error-wise. Norm-minimal bump functions will also not be dependent on trial spaces. and if

restricted to some trial space, their norm will not be minimal. In view of a Trade-off Principle, this means that non-optimal recovery methods will sacrifice smaller errors for larger stability.

Anyway, we now consider a set Λ of data functionals $\lambda_1, \dots, \lambda_n$ and set of trial functions v_1, \dots, v_N from some normed space U of functions, spanning a subspace V . These two ingredients determine a generalized Vandermonde matrix $A_{\Lambda, V}$ of size $M \times N$ with entries $\lambda_j(v_k)$ that is in the theoretical background, though certain algorithms will never generate it as a whole. We also assume that there may be an unknown numerical rank that limits the practical use of the matrix as is. This occurs in plenty of kernel-based methods, and even in square cases $M = N$ there may be a rank loss that occurs while the matrix condition in the sense of MATLAB's `cond` is still tolerable.

There are many ways to deal with this situation, and here we assume that the practically applied technique uses an $N \times M$ matrix C that calculates coefficients for the trial space basis for a given data vector $\Lambda(f)$. By an N -vector v of the N basis functions, the result is a function $v^T C \Lambda(f)$, and evaluation of a functional μ has the error

$$\mu(f) - \mu(v)^T C \Lambda(f) = \mu(f) - \sum_{j=1}^N \mu(v_j) \sum_{k=1}^M C_{jk} \lambda_k(f) = \mu(f) - \sum_{k=1}^M \mu(a_k) \lambda_k(f) \tag{24}$$

for quasi-Lagrangians

$$a_k = \sum_{j=1}^N v_j C_{jk}, \quad 1 \leq k \leq M \tag{25}$$

leading to the Power Function being the dual norm

$$P_{\Lambda, C}(\mu) = \left\| \mu - \sum_{j=1}^N \mu(v_j) \sum_{k=1}^M C_{jk} \lambda_k \right\|_{U^*}. \tag{26}$$

Bump functions are not necessarily connected to the trial space chosen. If there exists a bump function $f_{\mu, \Lambda}$, the Trade-off Principle (14) applies for the above Power Function. The next section will treat a special case in more detail, because it has a huge background literature in applications.

11.1 Unsymmetric collocation

An important example for solving PDEs via a recovery of functions is unsymmetric collocation, named after Edward Kansa [24]. Here, we confine ourselves to a standard Poisson problem (5) discretized as (6) for simplicity. One chooses a reproducing kernel Hilbert space \mathcal{H} of functions on Ω that matches the expectable smoothness of the solution, and implements the PDE via test functionals, as sketched in Section 3. The functionals in (6) may be renamed as $\lambda_m, 1 \leq m \leq M := M_\Delta + M_\beta$ to match the notations used above. But note that M will usually exceed N .

Symmetric collocation takes a space of trial functions where these test functionals act on the kernel K , and this is an optimal recovery strategy [16] in the space \mathcal{H} , with

good convergence properties [25, 26]. The trade-off principle for this was treated in Section 10.8.

The unsymmetric approach takes a set of trial functionals $\tau_k = \delta_{z_k}$, $1 \leq k \leq N$ to generate trial functions

$$v_k(x) = \tau_k^y K(y, x) = K(z_k, x), \quad 1 \leq k \leq N, \quad x \in \overline{\Omega}.$$

The notation is now like in Sections 10.7 and 11, but we have not yet specified how we choose the matrix C of (24).

For calculation of the Power Function, we use (24), define the quasi-Lagrangians a_k from (25) and get

$$\begin{aligned} P_{\Lambda, C}^2(\mu) &= (\mu - \sum_k \mu(a_k)\lambda_k, \mu - \sum_k \mu(a_k)\lambda_k)_{U^*} \\ &= K_{\mu, \mu} - 2b^T K_{\Lambda, \mu} + b^T K_{\Lambda, \Lambda} b \end{aligned}$$

in self-evident kernel matrix notation and $b = \mu(a) = \mu(C^T v)$. The Power Function for symmetric collocation replaces b by the solution b^* of the system $K_{\Lambda, \Lambda} b = K_{\Lambda, \mu}$ and therefore realizes the minimum of the quadratic form over all possible vectors b .

Figure 2 shows squares of Power Functions for unsymmetric collocation of a Poisson problem with Dirichlet data on the unit square. The setting has 121 regular interior points, 16 regular boundary points, 121 regular trial points and uses a Matern-Sobolev kernel of order 5 at scale 1. The matrix C was the pseudoinverse of the generalized Vandermonde matrix $A_{\Lambda, V}$.

The corresponding squares of optimal Power Functions from symmetric collocation are in Fig. 3. They are not substantially smaller, just by a factor of about 1/2.

Norm-minimal bump functions exist and have norms that are related to the optimal Power Function via Theorem 2 by equality in (16) and (17). They are Lagrangians for the symmetric setting. Therefore, the right-hand sides of these inequalities get larger when the Power Function $P_{\Lambda, C}$ is inserted. The reciprocals of squared norms of the optimal bump functions are visualized in Fig. 3, because they coincide with the square of the optimal Power Function. Special bump functions in the trial space of the unsymmetric case will usually not exist if $M > N$.

But it may be an advantage of the unsymmetric technique that its evaluation is based on quasi-Lagrangians instead of Lagrangians.

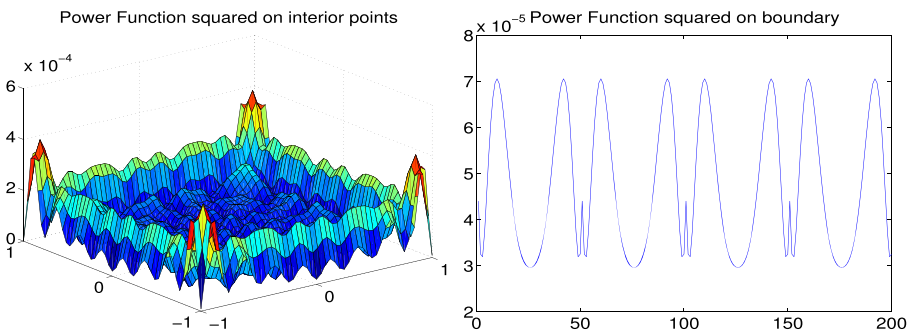


Fig. 2 Squares of Power Functions in interior and on boundary, for unsymmetric collocation

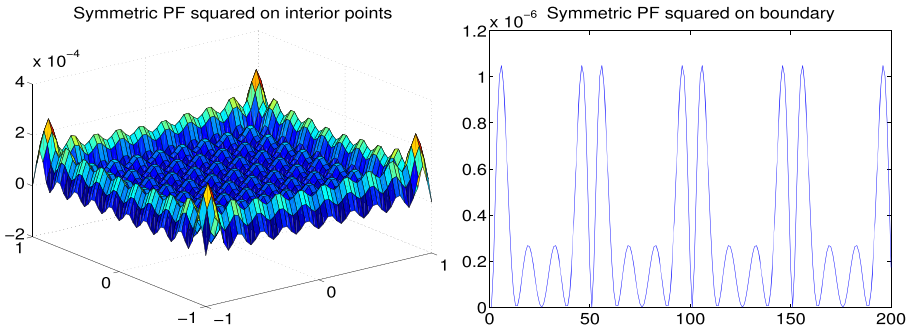


Fig. 3 Squares of optimal Power Functions in interior and on boundary, for symmetric collocation

The columns of C have the coefficients of the quasi-Lagrangians a_k in the trial basis, and therefore their squared norms $\|a_k\|_U^2$ are in the diagonal of the matrix $C^T K_{T,T} C$ where $K_{T,T}$ is the standard kernel matrix for evaluations in trial points [23]. Figure 4 shows the reciprocals of these, in order to compare with the squared optimal Power Functions in Fig. 3. These results come out only on the data locations, without any plotting refinement. The values are larger by about a factor of 18 than those of the square of the optimal Power Function, indicating that the squared norms of the quasi-Lagrangians are smaller than those of the Lagrangians for the symmetric case by a factor of 1/18.

Summarizing, unsymmetric collocation works at a larger error level than symmetric collocation, but gets better evaluation stability by using quasi-Lagrangians. The norms are kept fixed.

11.2 Regularization of linear systems

Assume an overdetermined linear system $Ax \approx b$ with an $M \times N$ matrix A . The spaces U and U^* then are \mathbb{R}^M . After a Singular Value Decomposition, the new $M \times N$

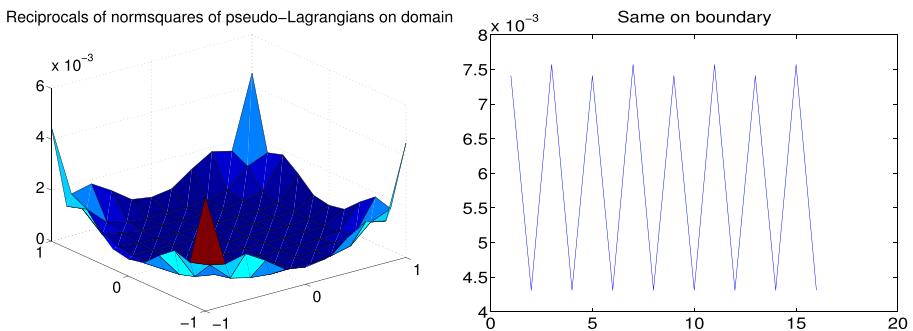


Fig. 4 Reciprocals of squares of norms of Quasi-Lagrangians in interior and on boundary

system is $\Sigma y \approx c$ where the diagonal of Σ carries the $N \leq M$ nonnegative singular values σ_j , $1 \leq j \leq N$. Formally, we set the σ_j , $N < j \leq M$ to zero. Then

$$\begin{aligned} \lambda_k(y) &= \sigma_k y_k, \text{ for } \sigma_k > 0, \text{ else } = 0 \\ c_{jk} &= \frac{\delta_{jk}}{\sigma_k}, \text{ for } \sigma_k > 0, \text{ else } = 0 \end{aligned}$$

if there is no regularization for small σ_k . By some elementary Linear Algebra,

$$P_{\Lambda, C}^2(\mu) = \sum_{\sigma_k=0} \mu_k^2.$$

All bump vectors $f_{\mu, \Lambda}$ satisfy $f_k = 0$ for $\sigma_k > 0$ and

$$1 = \mu(f) = \sum_{\sigma_k=0} \mu_k f_k$$

leading to the Trade-off Principle

$$1 \leq P_{\Lambda, C}^2(\mu) \|f_{\mu, \Lambda}\|_2^2$$

that takes the classical form (11) here. If a Tikhonov-type regularization uses

$$c_{jk}(\tau) = \frac{\delta_{jk}}{\sigma_k + \tau} \text{ for all } k,$$

the bump functions stay the same, but the Power Function increases to

$$P_{\Lambda, C(\tau)}^2(\mu) = \sum_k \mu_k^2 \frac{\tau^2}{(\sigma_k + \tau)^2}.$$

In the square nonsingular case, the Power Function is zero and there are no bump functions, leading back to the excluded $1 \leq 0 \cdot \infty$ situation.

12 Greedy methods

Assume the add-one-in situation, and consider an optimal μ to be added to Λ for an extended problem. In view of the Trade-off Principle, one should either take μ to maximize $P_{a_\Lambda}(\mu)$ or to minimize $f_{\mu, \Lambda}$. In cases satisfying Theorem 2, these strategies coincide. This aims at good stability and uses new functionals that cope with the current maximal error to make it zero in the next step. During a greedy method like this, the product in (14) stays above 1 and the first factor goes to zero, while the second increases. Again, this shows that large evaluation instabilities do not contradict convergence.

For interpolation of function values by polynomials, this leads to Leja points ([27], see also the survey by St. De Marchi [28]), while for kernel-based interpolation this is the P -greedy method of [23]. Under certain additional assumptions, these strategies are approximately optimal in the sense that they realize N -widths (G. Santin and B. Haasdonk [29]), i.e. they generate trial spaces that are asymptotically optimal under all other trial spaces of the same dimension. They can be combined with the construction of Newton bases on-the-fly [30], but we omit further details.

13 Outlook and open problems

The technique used in this paper is very elementary, and it is possible that there are earlier results on Trade-off Principles. One instance is [31] by Platte et al. proving that exponential convergence of polynomial interpolants implies exponential instability in case of equidistant points. Though the instability notion is different there, this can be seen as a case where a convergence rate is connected to a comparable rate of instability. This paper proves in general that all convergence rates have at least their exact counterpart in rates of evaluation instability when using dual norms. Such rates will change, e.g. when point locations are changed.

There are many more cases that fit into this paper, e.g. h/p Finite Elements or spaces of multivariate splines. If errors are decreased by extended smoothness properties, there always will be an increasing evaluation instability. The connection between smoothness properties and convergence rates is a well-known Trade-off Principle in Approximation Theory, holding for several important cases, but a general theory seems to be lacking.

The same holds for a hypothetical Trade-off Principle suggesting that strongly localized methods cannot have small errors and/or large smoothness.

Handling the non-dual case is an open problem as well, in particular for L_∞ evaluation stability.

If users have strong reasons to insist on very good accuracy and on evaluations of high derivatives, they have to face serious evaluation instabilities. Then it is a challenge to cope with these, including regularizations and other changes to the recovery map a_Λ . The literature on kernel-based methods provides several of such techniques, e.g. Contour-Padé [32], RBF-QR [33], and RBF-GA [34] by the group around Bengt Fornberg, and Hilbert-Schmidt-SVD by Fasshauer/McCourt [3, Chapter 13]. Greedy methods from Section 12 fight evaluation instability by choosing functionals or evaluation points adaptively.

Acknowledgements The author is grateful to C.S. Chen of the University of Southern Mississippi and Amir Noorizadegan of the National Taiwan University for an e-mail conversation that put evaluation instability into the focus for the Trade-off Principle. Furthermore, two referee reports were very helpful for avoiding certain misinterpretations and over-simplified perceptions. They led to some explicitly formulated clarifications, in particular with respect to the notion of evaluation instability.

Funding Open Access funding enabled and organized by Projekt DEAL.

Declarations

Conflict of interest The author has no conflicts of interest to declare that are relevant to the content of this article.

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

- Schaback, R.: Error estimates and condition numbers for radial basis function interpolation. *Adv. Comput. Math.*, 251–264. <https://doi.org/10.1007/BF02432002> (1995)
- Wendland, H.: *Scattered Data Approximation*, p. 394. Cambridge University Press. <https://doi.org/10.1017/CBO9780511617539> (2005)
- Fasshauer, G., McCourt, M.: *Kernel-based Approximation Methods Using MATLAB*. Interdisciplinary Mathematical Sciences, vol. 19 World Scientific. <https://doi.org/10.1142/9335> (2015)
- Berrut, J.P., Trefethen, L.N.: Barycentric Lagrange interpolation. *SIAM Rev.* **46**, 501–517 (2004). <https://doi.org/10.1137/S0036144502417715>
- Schaback, R.: All well-posed problems have uniformly stable and convergent discretizations. *Numer. Math.* **132**, 597–630 (2016). <https://doi.org/10.1007/s00211-015-0731-8>
- Atluri, S.N., Zhu, T.-L.: A new meshless local Petrov-Galerkin (MLPG) approach to nonlinear problems in Computer modeling and simulation. *Comput. Model. Simul. Eng.* **3**, 187–196 (1998)
- Atluri, S.N., Zhu, T.-L.: A new meshless local Petrov-Galerkin (MLPG) approach in Computational Mechanics *Computational Mechanics*. <https://doi.org/10.1007/s004660050346> (1998)
- Babuška, I., Banerjee, U., Osborn, J.E.: Survey of meshless and generalized finite element methods: a unified approach. *Acta Numer.* **12**, 1–125 (2003). <https://doi.org/10.1017/S0962492902000090>
- Farwig, R.: Multivariate interpolation of arbitrarily spaced data by moving least squares methods. *J. Comp. Appl. Math.* **16**, 79–93 (1986). [https://doi.org/10.1016/0377-0427\(86\)90175-5](https://doi.org/10.1016/0377-0427(86)90175-5)
- Levin, D.: The approximation power of moving least-squares. *Math. Comput.* **67**, 1517–1531 (1998). <https://doi.org/10.1090/S0025-5718-98-00974-0>
- Wendland, H.: Local polynomial reproduction and moving least squares approximation. *IMA J. Numer. Anal.* **21**, 285–300 (2001). <https://doi.org/10.1093/imanum/21.1.285>
- Armentano, M.G.: Error estimates in Sobolev spaces for moving least square approximations. *SIAM J. Numer. Anal.* **39**(1), 38–51 (2001). <https://doi.org/10.1137/S0036142999361608>
- Belytschko, T., Krongauz, Y., Organ, D.J., Fleming, M., Krysl, P.: Meshless methods: an overview and recent developments. *Computer Methods in Applied Mechanics and Engineering*, special issue **139**, 3–47 (1996). [https://doi.org/10.1016/S0045-7825\(96\)01078-X](https://doi.org/10.1016/S0045-7825(96)01078-X)
- Mirzaei, D., Schaback, R., Dehghan, M.: On generalized moving least squares and diffuse derivatives. *IMA J. Numer. Anal.* **32**(3), 983–1000 (2012). <https://doi.org/10.1093/imanum/drr030>
- Mirzaei, D., Schaback, R.: Direct Meshless Local Petrov-Galerkin (DMLPG) method: a generalized MLS approximation. *Appl. Numer. Math.* **68**, 73–82 (2013). <https://doi.org/10.1016/j.apnum.2013.01.002>
- Schaback, R.: A computational tool for comparing all linear PDE solvers. *Adv. Comput. Math.* **41**, 333–355 (2015). <https://doi.org/10.1007/s10444-014-9360-5>
- Buhmann, M.D.: *Radial Basis Functions, Theory and Implementations*, p. 270. Cambridge University Press. <https://doi.org/10.1017/CBO9780511543241> (2003)
- Schaback, R., Wendland, H.: Kernel techniques: from machine learning to meshless methods. *Acta Numer.* **15**, 543–639 (2006). <https://doi.org/10.1017/S0962492906270016>
- De Marchi, S., Schaback, R.: Stability of kernel-based interpolation. *Adv. Comp. Math.* **32**, 155–161 (2010). <https://doi.org/10.1007/s10444-008-9093-4>
- Ahlberg, J.H., Nilson, E.N., Walsh, J.L.: *The Theory of Splines and Their Applications*. Mathematics in Science and Engineering, vol. 38. Academic Press. ISBN 0120447509, 9780120447503 (1967)
- Zwicknagl, B., Schaback, R.: Interpolation and approximation in Taylor spaces. *J. Approx. Theory* **171**, 65–83 (2013). <https://doi.org/10.1016/j.jat.2013.03.006>
- Driscoll, T.A., Hale, N., Trefethen, L.N.: *Chebfun Guide*. Pafnuty Publications (2014)
- De Marchi, S., Schaback, R., Wendland, H.: Near-optimal data-independent point locations for radial basis function interpolation. *Adv. Comput. Math.* **23**(3), 317–330 (2005). <https://doi.org/10.1007/s10444-004-1829-1>
- Kansa, E.J.: Application of Hardy’s multiquadric interpolation to hydrodynamics. In: *Proc. 1986 Simul. Conf.*, vol. 4, pp. 111–117 (1986)
- Franke, C., Schaback, R.: Solving partial differential equations by collocation using radial basis functions. *Appl. Math. Comp.* **93**, 73–82 (1998). [https://doi.org/10.1016/S0096-3003\(97\)10104-7](https://doi.org/10.1016/S0096-3003(97)10104-7)
- Franke, C., Schaback, R.: Convergence order estimates of meshless collocation methods using radial basis functions. *Adv. Comput. Math.* **8**, 381–399 (1998). <https://doi.org/10.1023/A:1018916902176>

27. Leja, F.: Sur certaines suites liées aux ensemble plan et leur application á la representation conforme. *Ann. Polon. Math.* **4**, 8–13 (1957)
28. De Marchi, S.: On Leja sequences: some results and applications. *Appl. Math. Comput.* **152**, 621–647 (2004). [https://doi.org/10.1016/S0096-3003\(03\)00580-0](https://doi.org/10.1016/S0096-3003(03)00580-0)
29. Santin, G., Haasdonk, B.: Convergence rate of the data-independent P -greedy algorithm in kernel-based approximation. *Dolomites Res. Notes Approx.* **10**(Special Issue), 68–78 (2017). https://doi.org/10.14658/pupj-drna-2017-Special_Issue-9
30. Müller, S., Schaback, R.: A Newton basis for kernel spaces. *J. Approx. Theory* **161**, 645–655 (2009). <https://doi.org/10.1016/j.jat.2008.10.014>
31. Platte, R., Trefethen, L.N., Kuijlaars, B.J.: Impossibility of fast stable approximation of analytic functions from equispaced samples, 308–313. <https://doi.org/10.1137/090774707>. *SIAM Review* (2012)
32. Fornberg, B., Wright, G.: Stable computation of multiquadric interpolants for all values of the shape parameter. *Comput. Math. Appl.* **48**(5–6), 853–867 (2004). <https://doi.org/10.1016/j.camwa.2003.08.010>
33. Fornberg, B., Larsson, E., Flyer, N.: Stable computations with Gaussian radial basis functions. *SIAM J. Sci. Comput.* **33**(2), 869–892 (2011). <https://doi.org/10.1137/09076756X>
34. Fornberg, B., Lehto, E., Powell, C.: Stable calculation of Gaussian-based RBF-FD stencils. *Comput. Math. Applic.* **65**, 627–637 (2013). <https://doi.org/10.1016/j.camwa.2012.11.006>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.