

FINETUNING GREEDY KERNEL MODELS BY EXCHANGE ALGORITHMS*

TIZIAN WENZEL[†] AND ARMIN ISKE[‡]

Abstract. Kernel based approximation offers versatile tools for high-dimensional approximation, which can especially be leveraged for surrogate modeling. For this purpose, both “knot insertion” and “knot removal” approaches aim at choosing a suitable subset of the data, in order to obtain a sparse but nevertheless accurate kernel model.

In the present work, focussing on kernel based interpolation, we aim at combining these two approaches to further improve the accuracy of kernel models, without increasing the computational complexity of the final kernel model. For this, we introduce a class of kernel exchange algorithms (KEA). The resulting KEA algorithm can be used for finetuning greedy kernel surrogate models, allowing for an reduction of the error up to 86.4% (17.2% on average) in our experiments.

Key words. Kernel based approximation, greedy algorithms, optimal data selection

AMS subject classifications. 65D15, 41A05, 46E22

1. Introduction. Kernel methods comprise versatile tools for multivariate numerical approximation, statistical learning and machine learning [13, 24, 25]. They are popular due to their easy implementation, good performance and well-established mathematical theory based on reproducing kernel Hilbert spaces. In applications, sparse kernel models are frequently used for surrogate modeling purposes [5, 6, 11], where efficient models are required that are able to accurately describe an expensive full model. This can be achieved by using only a small subset of a possibly big initial training set for computing the final model. For this, the computation of an optimal subset is frequently intractable due to the high combinatorial complexity [12]. Thus, a more practical way is provided by using greedy algorithms, which are only *locally optimal*, however very efficient to run. For the task at hand, these greedy algorithms either start with a small or even empty set and then add points, or they start with the full set and then remove points. In both approaches, a desired expansion size or a desired accuracy threshold may serve as stopping criteria.

In first approach, i.e. *inserting* points, most work focussed on greedy algorithms that iteratively add single points according to some selection criterion. For this, various criteria with different purposes and advantages have been introduced, e.g. the *f*-greedy algorithm [23], the *P*-greedy algorithm [1], or the *f/P*-greedy algorithm [17]. These algorithms were jointly analyzed in a framework of so-called β -greedy algorithms in [28]. Despite a greedy algorithm is a concatenation of local optimal steps, the overall procedure usually does not give a globally optimal solution. Nevertheless, some of these greedy insertion algorithms have been proven to be *asymptotically optimal*: For the *P*-greedy algorithm, the optimality of the resulting convergence rate

*The authors acknowledge financial support through the projects LD-SODA of the *Landesforschungsförderung Hamburg* (LFF) and support from the RTG 2583 “Modeling, Simulation and Optimization of Fluid Dynamic Applications” funded by the *Deutsche Forschungsgemeinschaft* (DFG).

[†]Department of Mathematics, Universität Hamburg, Bundesstraße 55, 20146 Hamburg, Germany, (tizian.wenzel@uni-hamburg.de).

[‡]Department of Mathematics, Universität Hamburg, Bundesstraße 55, 20146 Hamburg, Germany (armin.iske@uni-hamburg.de).

in several cases was proven in [20, 27], while the optimality of the convergence rate of the f -greedy algorithm is discussed in [22].

The second approach, i.e. *removing* points, was proposed in [8, 9] under the notion of *thinning algorithms*, for the purpose of generating quasi-uniformly distributed subsets of scattered points. Adaptive thinning algorithms were later suggested and analyzed in [2, 3]. Further work in this direction was done e.g. in [16] under the notion *knot removal* schemes, where an *efficient reduced basis algorithm* (ERBA) was introduced for removing points. Also for these removal algorithms, optimality results are available [4].

As these greedy insertion algorithms are (partly) known to be asymptotically optimal, the rate of convergence cannot be improved anymore. Nevertheless it may be possible to further minimize the absolute error for a given expansion size, without achieving a faster convergence rate, i.e. by minimizing the prefactors in front of the asymptotic decay. To the best of the authors knowledge, this subject was never considered in the literature so far, probably because it is not possible to modify the greedy selection criteria such that a smaller error is obtained for *any* expansion size. In this sense, greedy algorithms can be seen as limited when aiming for a small error given an restricted expansion size (budget).

While one might be tempted to think about a global optimization of the centers and a decoupling of centers and function values (as in unsymmetric collocation [14]), this would likely require costly gradient descent techniques while also loosing the theoretical access based on the well-known kernel representer theorem [7, 25].

Therefore, we introduce *exchange algorithms*, which solely make use of the available training data and thus stick to the framework and mathematical theory provided by the representer theorem. By using an initial set of greedily selected centers – obtained either via insertion or removal strategies – and subsequent exchange steps of these centers, we are able to finetune greedy kernel models. While this introduces a small computational overhead, the improved accuracy of the kernel model frequently pays off in subsequent evaluations, as the number of centers and thus the cost of evaluation of the kernel model stays fixed. All in all, the combination of greedy algorithms and kernel exchange algorithms is a further contribution step towards the optimal selection of centers for kernel approximation.

The paper is structured as follows: In Section 2, greedy kernel algorithms are reviewed, which serve as building blocks for the kernel exchange algorithm (KEA), which is introduced and discussed in Section 3. Section 4 showcases the use of the KEA algorithms on a variety of use cases, achieving improvements of up to 86.4%. Finally Section 5 concludes the paper.

2. Background on greedy kernel models. The following section reviews the most important terminology from kernel interpolation, which is required for the introduction and discussion of the kernel exchange algorithms (KEA) in the subsequent Section 3.

For our purposes, we consider strictly positive definite continuous kernels $k : \Omega \subset \mathbb{R}^d \rightarrow \mathbb{R}$, which are defined on some bounded subset $\Omega \subset \mathbb{R}^d$. Strictly positive definite means, that the kernel matrix $k(X, X) := (k(x_i, x_j))_{i,j=1}^n \in \mathbb{R}^{n \times n}$ is positive definite for any choice of pairwise distinct points $\{x_1, \dots, x_n\}_{i=1}^n \subset \Omega$. Given such a kernel k , there always exists a unique native space of functions associated to k , the so called reproducing kernel Hilbert space $\mathcal{H}_k(\Omega) \subset \mathcal{C}(\Omega)$. A typical example of such a kernel

is given by the basic Matérn kernel

$$k(x, z) = \exp(-\|x - z\|), \quad (2.1)$$

which is also called exponential kernel or Laplace kernel. The class of Matérn kernels will be used for the numerical experiments in Section 4.

Given a function $f \in \mathcal{H}_k(\Omega)$ and pairwise distinct interpolation nodes $X_n \subset \Omega$, the kernel representer theorem states that there exists a minimum norm interpolant

$$s_{f, X_n} = \sum_{j=1}^n \alpha_j^{(n)} k(\cdot, x_j), \quad (2.2)$$

where the coefficients $\alpha = (\alpha_j^{(n)})_{j=1}^n \in \mathbb{R}^n$ can be computed directly by solving the linear equation system $k(X, X)\alpha = (f(x_i))_{i=1}^n$. Thus accuracy of the interpolant s_{f, X_n} for approximation of the target function f crucially depends on the choice of the kernel and on the choice of interpolation points $\{x_j\}_{j=1}^n \subset \Omega$. In this work we deal with the second case, i.e. a suitable choice of interpolation points. As elaborated in Section 1, greedy algorithms provide a computationally efficient method to obtain a suitable set of interpolation points.

2.1. Greedy point insertion. Greedy insertion algorithms usually start with an empty set $X_0 := \{\}$, which is iteratively updated by inserting a bunch of points. We focus on the most popular case, which adds single points x_{n+1} and thus reads $X_{n+1} := X_n \cup \{x_{n+1}\}$. This procedure is iterated, until a suitable expansion size n is met or some accuracy or stability threshold is reached. For the choice of the new point x_{n+1} , several criteria have been established (see e.g. [28, Section 1] for a more detailed discussion), of which we focus in the following on the residual based f -greedy and the power function based P -greedy criterion:

$$\begin{aligned} x_{n+1} &= \arg \max_{x \in \Omega} |(f - s_{f, X_n})(x)| && (f - \text{greedy}), \\ x_{n+1} &= \arg \max_{x \in \Omega} P_{X_n}(x) && (P - \text{greedy}). \end{aligned} \quad (2.3)$$

Here, $P_{X_n}(x)$ is the so-called power function, defined as

$$P_{X_n}(x) = \sup_{0 \neq f \in \mathcal{H}_k(\Omega)} \frac{|(f - s_{f, X_n})(x)|}{\|f\|_{\mathcal{H}_k(\Omega)}},$$

which measures the worst case error. The power function can be computed efficiently based on the centers X_n and the kernel k . In practice, a large discrete base set $X \subset \Omega$ is used instead of the domain Ω .

In order to update the kernel model Eq. (2.2), one typically does not use the kernel basis $\{k(\cdot, x_j), j = 1, \dots, n\}$, because it would require a recomputation of the coefficients $\{\alpha_j^{(n)}\}_{j=1}^n$ for updating s_{f, X_n} to $s_{f, X_{n+1}}$. Therefore, one usually prefers to work in the Newton basis, which allows for efficient updating of s_{f, X_n} to $s_{f, X_{n+1}}$. We refer to the reference [18] for more updates on the Newton basis and the corresponding properties. These efficient update procedures due to the Newton basis will also be leveraged for an efficient implementation of the kernel exchange algorithm (KEA), to be introduced in Algorithm 1.

An implementation of such greedy insertion algorithms is provided e.g. by the VKOGA (*vectorial kernel orthogonal greedy algorithm*) package [21].

2.2. Greedy point removal. Greedy removal algorithms start with a large discrete base set $X \subset \Omega$, and are iteratively updated by removing a subset of the included points [16]. In the following we consider the case of removing single points, such that the update is given as $X_n := X_{n+1} \setminus \{x_{n+1}\}$ for $n = |X| - 1, |X| - 2, \dots$. This is iterated until a desired expansion size is reached or some accuracy threshold is met.

Analogously to the selection criteria of the greedy insertion algorithms in Eq. (2.3), there is again a residual based as well as a power function based criterion. The idea is to remove the point which results in the smallest increase of the corresponding error indicator. Thus in this case, the residual based as well as power function based selection criteria read

$$\begin{aligned} x_{n+1} &= \arg \min_{x \in X_{n+1}} |(f - s_{f, X_{n+1} \setminus \{x\}})(x)| \\ x_{n+1} &= \arg \min_{x \in X_{n+1}} P_{X_{n+1} \setminus \{x\}}(x). \end{aligned} \quad (2.4)$$

Both these selection criteria are based on leave-one-out cross validation errors. While the computation of these leave-one-out cross validation errors is computational more demanding, there are efficient implementations based on *Rippa's rule* and extensions thereof [15, 19].

An implementation of such greedy removal algorithms is provided e.g. by the ERBA (*efficient reduced basis algorithm*) package [16].

Both greedy insertion algorithms as well as greedy removal algorithms are limited in the sense, that they only increase respectively decrease the number of centers. Thus, a suboptimal step can never be reversed, which can be seen as a limitation. This limitation is lifted with the kernel exchange algorithms (KEA) introduced in Section 3, as they exchange selected centers by inserting a center and also removing a center in every step, thus performing a locally optimal update.

3. Kernel exchange algorithm: KEA. Section 1 and Section 2 discussed several aspects of the optimality of the greedy insertion and removal algorithms. Especially the convergence rates (in the number of interpolation points) of greedy insertion algorithms is known to be asymptotically optimal in several cases. Nevertheless, the globally *optimal* selection of interpolation points still remains unclear, especially due to its computational complexity. In order to narrow this gap between greedily selected points and optimal points from a practical point of view, we propose kernel exchange algorithms:

We consider an initial base set (X, Y) of $N := |X|$ input points with corresponding target values Y , as well as a kernel k for approximation of these data points. We assume a non-empty initial set $X_n \subset X$ of $n < N$ centers to be given, which can be obtained for example by a greedy insertion algorithm (see Section 2.1) or a greedy removal algorithm (see Section 2.2). Given a maximal number of m exchange steps, for every exchange step $i = 1, \dots, m$, we pick a data point of $X \setminus X_n$ to be added, as well as a data point of X_n to be removed. Like this, the set of selected centers is updated as

$$X_n^{(i+1)} := X_n^{(i)} \setminus \{x_{\text{remove}}\} \cup \{x_{\text{add}}\}. \quad (3.1)$$

For the selection of the points x_{add} and x_{remove} , we leverage the residual based f -greedy criterion as well as the power function based P -greedy criterion, see Eq. (2.3)

respective Eq. (2.4). The step of Eq. (3.1) is repeated, until the predefined number of maximal exchanges m is reached, or some predefined stopping criterion (based e.g. on the final accuracy) is met. The overall algorithm is formalized as pseudocode in Algorithm 1.

For the implementation of the kernel exchange algorithm, we combine the efficient implementations of the greedy insertion and removal algorithms, as implemented e.g. in the algorithms VKOGA [21] and ERBA [16], see Section 2.

Algorithm 1: Kernel Exchange Algorithm (KEA) for kernel model $\sum_{j=1}^n \alpha_j k(\cdot, x_j)$ of expansion size n

```

1   Input : Data  $(X, Y)$ , initial set of centers  $X_n^{(0)} \subset X$  of  $n > 0$  points,
          kernel  $k$ , number of exchange steps  $m \in \mathbb{N}$ 
2   Result: Exchanged set of centers  $X_n^{(m)} \subset X$ , final kernel model  $s_n^{(m)}(\tilde{X}_n)$ 
3
4   for  $i = 1, \dots, m$  do
5        $x_{\text{add}} \leftarrow \text{SELECT\_ADD}(s_n, X, Y)$  according to Eq. (2.3) ;
6        $x_{\text{remove}} \leftarrow \text{SELECT\_REMOVE}(s_n, X, Y)$  according to Eq. (2.4);
7        $X_n^{(i)} \leftarrow X_n^{(i-1)} \setminus \{x_{\text{remove}}\} \cup \{x_{\text{add}}\}$ ;
8
9        $s_i = s_i(X_n^{(i)})$  ;                                /* Compute updated kernel model */
10
11     EARLY\_STOPPING( $x_{\text{add}}, x_{\text{remove}}$ )
12 end
13
14 RETURN  $X_n^{(m)}, s_n = s_n(X_n^{(m)})$ 
```

4. Numerical experiments. This section provides numerical experiments on the introduced kernel exchange algorithm (KEA) of Section 3. We start in Section 4.1 with a comparison of greedy insertion and greedy removal algorithms. Subsequently, focussing on the case of greedy insertion algorithms, Section 4.2 considers low dimensional and Section 4.3 considers higher dimensional examples, where KEA is used to finetune greedy kernel insertion algorithms. In particular, Section 4.2 and Section 4.3 compare the accuracy of kernel models using a base set $X_n^{(0)}$ with the accuracy of a kernel model based on the exchanged set $X_n^{(m)}$ after using the KEA algorithm.

As kernels we consider Matérn kernels of different smoothnesses, namely $k(x, z) = \Phi(x - z)$, where the function $\Phi : \mathbb{R}^d \rightarrow \mathbb{R}$ is defined via its Fourier transform $\hat{\Phi}(\omega) = (1 + \|\omega\|_2^2)^{-\tau}$ with $\tau = \frac{d+(2p+1)}{2}$ for $p \in \{0, 1, 2, 3, 4\}$. All these kernels have an explicit form, and for $p = 0$ we obtain the kernel from Eq. (2.1).

The implementation of the KEA algorithm as well as the code to reproduce the numerical experiments can be found at:

<https://gitlab.rrz.uni-hamburg.de/bbd9097/parameter-2024-finetuning-greedy-kernel-models>

4.1. Greedy insertion vs. greedy removal. In a first numerical experiment, we briefly compare the two possible approaches of greedy insertion of centers vs the greedy removal of centers as introduced in Section 2.1 and Section 2.2. For this, we make use of the corresponding software packages VKOGA [21] and ERBA [16].

We present two exemplary numerical results, though we remark that the findings presented here also hold for other examples. We consider the domains $\Omega = [0, 1]^2$ respective $[0, 1]^3$ with target functions

$$f_2(x) = \|x\|^2 \quad \text{respective} \quad f_3(x) = |x_1 - 0.5| + \sin(x_2 + x_3). \quad (4.1)$$

As a base set, we consider each 256 low discrepancy points X within Ω . In contrast to the numerical experiments in the next sections, we use low-discrepancy points instead of randomly sampled points to avoid numerical instabilities due to too small values, which may occur as soon as nearby points are used as centers. The greedy insertion algorithm starts with an empty set of centers, and adds centers until all the 256 points are used. The removal algorithm operates in the reverse sense and starts with all the centers, and then removes centers until no center is used. The resulting maximal error $\max_{x \in X} |f(x) - s_n(x)|$ over the number n of centers is visualized for each the greedy insertion model and the greedy removal models s_n in Figure 4.1: One can observe that the maximal error is approximately equal for all the expansion sizes n . Thus, from the theoretical point of view, either applying a greedy insertion or a greedy removal yields approximately the same accuracy. However, from a practical point of view, if only a small number of centers $n \ll |X|$ is desired, it makes more sense to leverage the greedy insertion algorithm. On the contrary, if only few centers should be removed, i.e. $n \lesssim |X|$, it is more practical to use the greedy removal algorithm.

For our motivated purpose of surrogate modeling, the first case $n \ll |X|$ is more important. Thus we make use of the greedy insertion algorithm in the following, and investigate to which extent its results can be improved by applying KEA (see Algorithm 1).

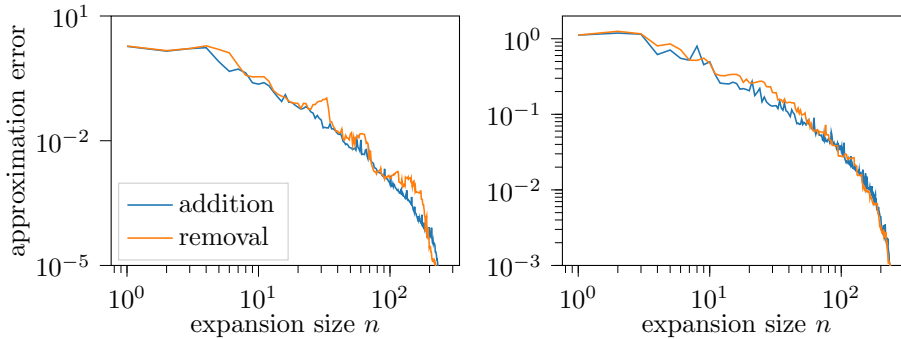


FIG. 4.1. Visualization of the training error (y-axis) over the number of used interpolation points (x-axis) for the target functions in Eq. (4.1). The insertion algorithm of Section 2.1 increases the number of points (operates from “left to right” on the x-axis), the removal algorithm of 2.2 decreases the number of points (operates from “right to left”). Both approaches yield approximately the same interpolation errors for all expansion sizes n .

4.2. Function approximation: Low dimensional example. As a first test case, we consider the domain $\Omega = [0, 1]^2 \subset \mathbb{R}^d$ as input space and target values generated by four test functions. For this we chose the classical well-known Franke

test function f_1 as well as three further Franke test function given as [10]

$$\begin{aligned} f_2(x) &= \frac{1}{9}(\tanh(9x_2 - 9x_1) + 1), \\ f_3(x) &= \frac{\frac{125}{100} + \cos(5.4x_2)}{6 + 6(3x_1 - 1)^2}, \\ f_4(x) &= \frac{1}{3} \exp\left(-\frac{81}{16}((x_1 - 1/2)^2 + (x_2 - 1/2)^2)\right). \end{aligned}$$

For all the four functions, the input domain Ω is discretized with each 10^3 uniformly randomly sampled training points X_{train} and testing points X_{test} . The greedy insertion algorithm with the f -greedy criterion (see Section 2.1) is used to select up to 150 (for f_1) respective 80 (for f_2, f_3, f_4) centers from the base set X_{train} . For 10 logarithmically equally spaced values n within 5 and 150 respective 80, the resulting greedy insertion kernel model s_n is finetuned with help of KEA (using at most $m = 100$ exchange steps) to obtain the model $s_{n,\text{KEA}}$.

In order to assess the improvement due to the finetuning by KEA, we consider the improvement ratio on the test set X_{test} , i.e.

$$\frac{\|f - s_{n,\text{KEA}}\|_{L^\infty(\Omega)}}{\|f - s_n\|_{L^\infty(\Omega)}}, \quad (4.2)$$

computed on the 10^3 test points.

The resulting improvement ratios of Eq. (4.2) over the number of centers n are visualized in Figure 4.2 for all the four test functions f_1 to f_4 and all the five considered Matérn kernels k . It can be clearly seen, that in most cases the improvement ratio is below 1, which means that the finetuning of the kernel model s_n due to KEA achieved a smaller test error. The best improvement is obtained for the Matérn kernel with smoothness $p = 2$ and an expansion size of $n = 12$: Here, the improvement ratio is 0.136, which means that the exchange due to KEA gave a tremendous improvement, without changing the size n of the kernel model. On average, an improvement of 17.2% is obtained. In general, the improvement seems to be more pronounced for smoother kernels, i.e. higher values of p . This observation can be explained in view of the convergence analysis for greedy insertion algorithms in [28]: There, a convergence bound as $C \cdot n^{-\alpha}$ (for some prefactor $C > 0$ and some convergence rate $\alpha > 0$) is proven. The prefactor C is increased (compared to the prefactor of a non-greedy algorithm) by an α -dependent factors as

$$C \propto 2^{\alpha+1/2} e^\alpha, \quad (4.3)$$

see [28, Corollary 11]. In the case of Matérn kernels considered here, it holds $\alpha = \frac{2p+1}{2d}$, such that larger values of p imply a larger increase of the prefactor C . In effect, for larger values of p there is a bigger gap, which can be narrowed with help of KEA.

Only for a few instances within Figure 4.2, no improvement or in some rare cases even a deterioration can be observed. We remark that such a deterioration can be observed despite the local optimality of the kernel exchange algorithm, because the error is evaluated on an independent test set, which was withheld from the training process.

4.3. Function approximation: High dimensional example. As a second test case, we consider the domains $\Omega = [0, 1]^d \subset \mathbb{R}^d$ for $d \in \{5, 6\}$ and the following two

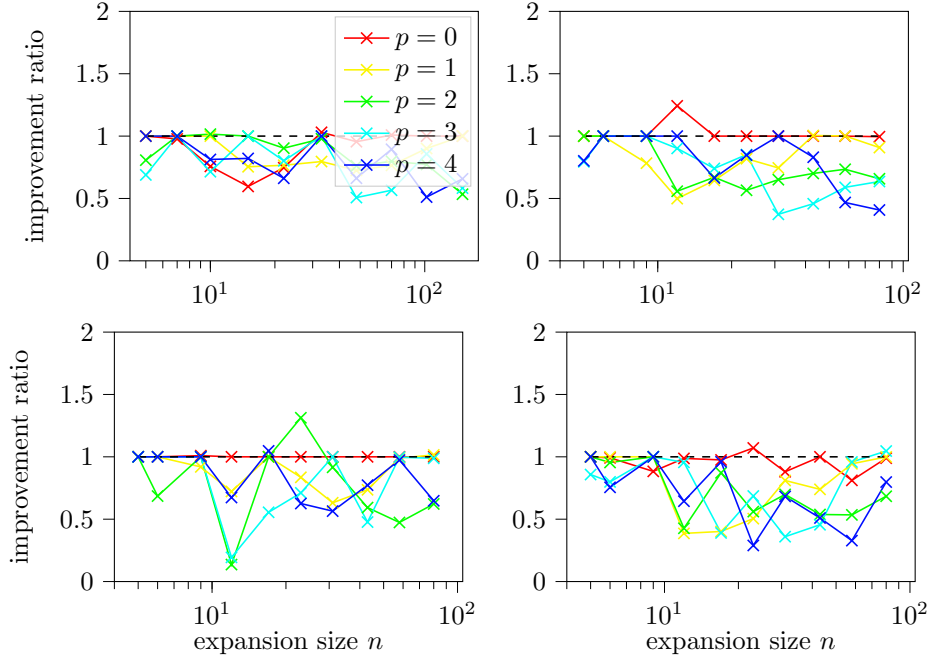


FIG. 4.2. Visualization of the improvement ratio $\|f - s_{n,KEA}\|_{L^\infty(\Omega)} / \|f - s_n\|_{L^\infty(\Omega)}$ (y -axis) over the kernel model expansion size n (x -axis) for the four two-dimensional test functions from Section 4.2: For values in $(0, 1)$, KEA yields improvements; for values in $(1, \infty)$, KEA yields deterioration. Five Matérn kernels with different smoothness parameters $p \in \{0, 1, 2, 3, 4\}$ were used.

test functions, which were also used in [26] as test functions for greedy approximation:

$$f_5(x) = e^{-4(\sum_{j=1}^5 x_j - 0.5)^2}$$

$$f_6(x) = e^{-4\sum_{j=1}^5 (x_j - 0.5)^2} + 2|x_1 - 0.5|$$

Again we test the five Matérn kernels from Section 4.2, however additionally making use of a two-layered kernel structure as $k(Ax, Az) = \Phi(A(x - z))$, with a matrix $A \in \mathbb{R}^{d \times d}$ that is optimized. This matrix A allows to adapt the shape of the initially radial kernel k to the data to be approximated, which usually improves the accuracy for medium- to high-dimensional problems. For details on two-layered kernels and the corresponding optimization procedure to obtain a suitable matrix A , we refer to [26].

The layout of the numerical experiment is the same as previously in Section 4.2, with a couple of minor changes: In order to take into account the higher dimensionality of the domain Ω , we employ 10^4 uniformly randomly sampled points for X_{train} as well as X_{test} . The greedy insertion algorithm uses again the f -greedy criterion, and selects up to 100 centers for f_5 respective 200 centers for f_6 . KEA is applied again to intermediate models of size n for 10 logarithmically equally spaced values of n between 5 and the maximal expansion size 100 respective 200. The improvement ratio of Eq. (4.2) is considered, and the results are displayed in Figure 4.3:

As in the low dimensional examples, one can observe that the use of KEA further reduces the approximation error for most expansion sizes as well as most kernels. The improvement is more pronounced for smoother kernels (i.e. large values of p),

especially for $p = 0$ there is frequently no improvement. The same explanation as given around Eq. (4.3) also applies here. Only in rare cases, there is a deterioration instead of an improvement of the ratio Eq. (4.2).

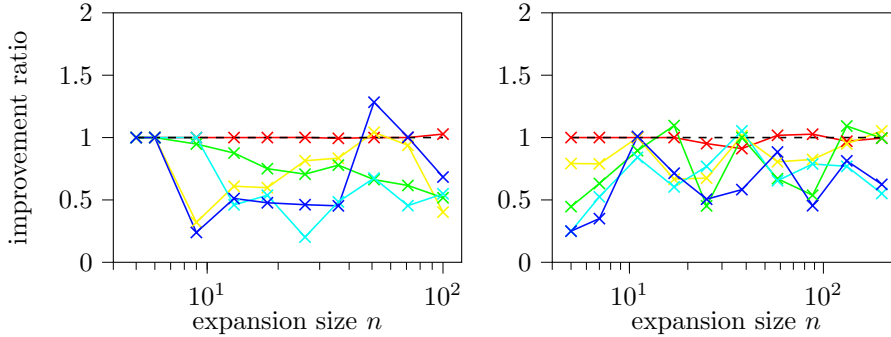


FIG. 4.3. Visualization of the improvement ratio $\|f - s_{n,KEA}\|_{L^\infty(\Omega)} / \|f - s_n\|_{L^\infty(\Omega)}$ (y -axis) over the kernel model expansion size n (x -axis) for the two high-dimensional test functions from Section 4.3: For values in $(0, 1)$, KEA yields improvements; for values in $(1, \infty)$, KEA yields deterioration. Five Matérn kernels with different smoothness parameters $p \in \{0, 1, 2, 3, 4\}$ were used.

5. Conclusion & Outlook. In this work, two approaches of greedy kernel algorithms for interpolation were considered and compared, namely greedy insertion and greedy removal of points. The driving motivation for these algorithms is to derive sparse and efficient kernel models in a computational feasible way.

In order to finetune these greedy kernel models, we introduced and investigated a *kernel exchange algorithm* (KEA): Based on an initial set of centers, provided e.g. by a greedy algorithm, exchange steps are performed to further optimize the selected subset of centers, without increasing (or decreasing) the amount of centers. Doing so, we showed that it is indeed possible to further improve the accuracy of the final kernel model.

Future work may address the quantification of the possible improvement, in particular in comparison to a theoretically optimal center distribution, which is however computational infeasible in most cases.

Acknowledgements: The authors acknowledge financial support through the projects LD-SODA of the *Landesforschungsförderung Hamburg* (LFF) and support from the RTG 2583 “Modeling, Simulation and Optimization of Fluid Dynamic Applications” funded by the *Deutsche Forschungsgemeinschaft* (DFG).

REFERENCES

- [1] S. De Marchi, R. Schaback, and H. Wendland. Near-optimal data-independent point locations for radial basis function interpolation. *Advances in Computational Mathematics*, 23(3):317–330, 2005.
- [2] L. Demaret, N. Dyn, and A. Iske. Image compression by linear splines over adaptive triangulations. *Signal Processing*, 86(7):1604–1616, 2006.
- [3] L. Demaret and A. Iske. Adaptive image approximation by linear splines over locally optimal delaunay triangulations. *IEEE Signal Processing Letters*, 13(5):281–284, 2006.
- [4] L. Demaret and A. Iske. Optimal N -term approximation by linear splines over anisotropic delaunay triangulations. *Mathematics of Computation*, 84(293):1241–1264, 2015.

- [5] F. Döppel, T. Wenzel, R. Herkert, B. Haasdonk, and M. Votsmeier. Goal-Oriented Two-Layered Kernel Models as Automated Surrogates for Surface Kinetics in Reactor Simulations. *Chemie Ingenieur Technik*, 2024.
- [6] S. Dutta, M. W. Farthing, E. Perracchione, G. Savant, and M. Putti. A greedy non-intrusive reduced order model for shallow water equations. *Journal of Computational Physics*, 439:110378, 2021.
- [7] G. E. Fasshauer and M. J. McCourt. *Kernel-based Approximation Methods using MATLAB*, volume 19. World Scientific Publishing Company, 2015.
- [8] M. S. Floater and A. Iske. Multistep scattered data interpolation using compactly supported radial basis functions. *Journal of Computational and Applied Mathematics*, 73(1-2):65–78, 1996.
- [9] M. S. Floater and A. Iske. Thinning algorithms for scattered data interpolation. *BIT Numerical Mathematics*, 38:705–720, 1998.
- [10] R. Franke. A critical comparison of some methods for interpolation of scattered data. Technical report, Monterey, California: Naval Postgraduate School., 1979.
- [11] B. Haasdonk, H. Kleikamp, M. Ohlberger, F. Schindler, and T. Wenzel. A New Certified Hierarchical and Adaptive RB-ML-ROM Surrogate Model for Parametrized PDEs. *SIAM Journal on Scientific Computing*, 45(3):A1039–A1065, 2023.
- [12] D. S. Hochbaum. Approximation algorithms for NP-hard problems. *ACM Sigact News*, 28(2):40–52, 1997.
- [13] T. Hofmann, B. Schölkopf, and A. J. Smola. Kernel methods in machine learning. *The Annals of Statistics*, 36(3):1171 – 1220, 2008.
- [14] E. J. Kansa. Multiquadrics — A scattered data approximation scheme with applications to computational fluid-dynamics. II. Solutions to parabolic, hyperbolic and elliptic partial differential equations. *Computers & Mathematics with Applications*, 19(8-9):147–161, 1990.
- [15] F. Marchetti. The extension of Rippa’s algorithm beyond LOOCV. *Applied Mathematics Letters*, 120:107262, 2021.
- [16] F. Marchetti and E. Perracchione. Efficient Reduced Basis Algorithm (ERBA) for kernel-based approximation. *Journal of Scientific Computing*, 91(2):41, 2022.
- [17] S. Müller. *Komplexität und Stabilität von kernbasierten Rekonstruktionsmethoden (Complexity and Stability of Kernel-based Reconstructions)*. PhD thesis, Fakultät für Mathematik und Informatik, Georg-August-Universität Göttingen, 2009.
- [18] M. Pazouki and R. Schaback. Bases for kernel-based spaces. *Journal of Computational and Applied Mathematics*, 236(4):575–588, 2011.
- [19] S. Rippa. An algorithm for selecting a good value for the parameter c in radial basis function interpolation. *Advances in Computational Mathematics*, 11:193–210, 1999.
- [20] G. Santin and B. Haasdonk. Convergence rate of the data-independent P -greedy algorithm in kernel-based approximation. *Dolomites Research Notes on Approximation*, 10:68–78, 2017.
- [21] G. Santin and B. Haasdonk. Kernel methods for surrogate modeling. In P. Benner, S. Grivet-Talocia, A. Quarteroni, G. Rozza, W. Schilders, and L. M. Silveira, editors, *Model Order Reduction*, volume 2. De Gruyter, 2021.
- [22] G. Santin, T. Wenzel, and B. Haasdonk. On the optimality of target-data-dependent kernel greedy interpolation in Sobolev Reproducing Kernel Hilbert Spaces. *arXiv preprint arXiv:2307.09811*, 2023.
- [23] R. Schaback and H. Wendland. Adaptive greedy techniques for approximate solution of large RBF systems. *Numerical Algorithms*, 24(3):239–254, 2000.
- [24] I. Steinwart and A. Christmann. *Support vector machines*. Springer Science & Business Media, 2008.
- [25] H. Wendland. *Scattered Data Approximation*, volume 17 of *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge University Press, Cambridge, 2005.
- [26] T. Wenzel, F. Marchetti, and E. Perracchione. Data-driven kernel designs for optimized greedy schemes: A machine learning perspective. *SIAM Journal on Scientific Computing*, 46(1):C101–C126, 2024.
- [27] T. Wenzel, G. Santin, and B. Haasdonk. A novel class of stabilized greedy kernel approximation algorithms: Convergence, stability and uniform point distribution. *Journal of Approximation Theory*, 262:105508, 2021.
- [28] T. Wenzel, G. Santin, and B. Haasdonk. Analysis of target data-dependent greedy kernel algorithms: Convergence rates for f -, f - P - and f/P -greedy. *Constructive Approximation*, 57(1):45–74, 2023.