Local-metrics error-based Shepard interpolation as surrogate for highly non-linear material models in high dimensions

Juan M. Lorenzi,[1,a)] Thomas Stecher,[1] Karsten Reuter,[1] and Sebastian Matera[2]
[1]*Chair for Theoretical Chemistry and Catalysis Research Center, Technische Universität München, Lichtenbergstr. 4, 85747 Garching, Germany*
[2]*Fachbereich für Mathematik und Informatik, Freie Universität Berlin, Otto-von-Simson-Str. 19, D-14195 Berlin, Germany*

(Received 23 July 2017; accepted 4 October 2017; published online 24 October 2017)

Many problems in computational materials science and chemistry require the evaluation of expensive functions with locally rapid changes, such as the turn-over frequency of first principles kinetic Monte Carlo models for heterogeneous catalysis. Because of the high computational cost, it is often desirable to replace the original with a surrogate model, e.g., for use in coupled multiscale simulations. The construction of surrogates becomes particularly challenging in high-dimensions. Here, we present a novel version of the modified Shepard interpolation method which can overcome the *curse of dimensionality* for such functions to give faithful reconstructions even from very modest numbers of function evaluations. The introduction of local metrics allows us to take advantage of the fact that, on a local scale, rapid variation often occurs only across a small number of directions. Furthermore, we use local error estimates to weigh different local approximations, which helps avoid artificial oscillations. Finally, we test our approach on a number of challenging analytic functions as well as a realistic kinetic Monte Carlo model. Our method not only outperforms existing isotropic metric Shepard methods but also state-of-the-art Gaussian process regression. *Published by AIP Publishing.*
https://doi.org/10.1063/1.4997286

I. INTRODUCTION

The interest in multiscale modeling approaches for materials science and chemistry has exploded in the last two decades. One important class of such approaches employs sequential (or hand-shaking) strategies, where a smaller scale model is employed as the closure of a larger scale model. In the simplest case, this just requires the adjustment of a finite set of parameters, e.g., the viscosity of an isothermal Newtonian fluid. In the general setting, the analytic form of the closure is not known and the small-scale model is required to determine functions of the large-scale variables. A prototypical example is *ab initio* molecular dynamics, where the functional dependence of the Potential Energy Surface (PES) is obtained from first-principles electronic structure simulations. Employing a microscale simulation every time the function is evaluated is, of course, very time-consuming and then usually the bottleneck of such multiscale approaches. One way to accelerate this is to parametrize a surrogate model using small-scale simulations and employ this in the large-scale simulations instead of the microscale simulator. A number of different general purpose surrogate models have been used in this way, including neural networks,[1] Gaussian processes,[2–4] full[5] and sparse grid splines,[6] and modified Shepard interpolation.[7–9]

The present work grew out of our efforts to couple first-principles kinetic Monte Carlo (1p-kMC) to Computational Fluid Dynamics (CFD) simulations for reactive flows over a heterogeneous catalyst using local modified Shepard interpolations[8,9] and extensions thereof.[10,11] Here, the surrogate model is used to interpolate the catalytic turnover frequency (TOF) obtained from the mesoscopic 1p-kMC simulations as a function of the temperature, T, and the partial pressures $\{p_i\}$ of the N_{spec} different gas phase species. The surrogate model then serves as a boundary condition in CFD. A corresponding use of surrogate models for coupling mean-field microkinetic models to CFD is widespread, including the use of splines[12,13] or *in situ* adaptive tabulation.[14] The latter has also been employed in the kMC+CFD context for the simulation of crystal growth and catalysis,[15,16] albeit with a phenomenological kMC model.

Efficiently and reliably interpolating 1p-kMC based TOF maps is a challenging problem. Under the appropriate coordinate transformation (i.e., logarithmic pressures and TOF, as well as inverse temperature), the maps display an approximately linear behavior for large parts of the $(\{p_i\}, T)$-space. Usually these linear regions correspond to steady-state kinetic "phases," characterized by a defined coverage regime on the catalyst.[17,18] In contrast, the behavior at the boundaries between such "phases" is highly non-linear and characterized by a rapid change of the TOF value and gradient within a narrow range of p_i and T values. This is challenging for most interpolation methods and normally it is necessary to sample such regions densely to get satisfactory results. This is aggravated further in higher dimensions (i.e., for problems with a larger number of gas-phase species N_{spec}) because the number of points required to densely fill space grows

[a)]juan.lorenzi@tum.de

exponentially with the number of dimensions (the so-called *curse of dimensionality*). For this reason, 1p-kMC+CFD studies have, up to now, been limited to problems involving only a small number of gas-phase species, such as CO oxidation,[10,11,18] where, in addition to the temperature, only the CO and oxygen partial pressures play a role. Modeling more complex pathways would, of course, be of great interest. For example, in competitive CO + NO oxidation,[19] the dimensionality is already five (at least) because the TOFs also depend on the partial pressures of NO and NO_2.

In this article, we present an extension to the popular local modified Shepard interpolation[20] addressing the problem of approximating functions with sharp transitions in higher dimensions. Our approach constructs a local metric for each data point (node), which is then used to determine local polynomial approximations (the nodal functions), which are combined to estimate function values at arbitrary points (query points). In this way, we can exploit (local) low-dimensionality of the target functions: sharp variations typically occur only along a few directions, while the function is smoother along the others. Having metrics that are *local* is then crucial because the direction of rapid change might vary across the domain. In the 1p-kMC context, sharp variations in rates are often associated with phase transitions in the surface coverage. These transitions have interfaces which are quasi-$(D-1)$-dimensional for D dimensional problems. Close to such regions, only the direction perpendicular to the transition region presents rapidly changing behavior, and thus the function is approximately one-dimensional there. Where two interfaces meet (i.e., around points where three phases coexist), the behavior will be approximately two-dimensional. The idea of a local metric is shared with locally weighted projection regression,[21] which differs, however, in the way the metric is determined and the nodal functions are blended. Most significantly, we do not base the blending on the distances between the query point and the nodes, but on estimates of the approximation quality of the nodal functions at the query points.[10,22]

This combination of a local metric with error estimate based weighting largely suppresses artificial wiggles and especially overshoots close to sharp changes, while the resulting interpolant is once differentiable by construction. Our method produces accurate and qualitatively correct interpolations of a number of test functions with rapid, localized transitions, even in higher dimensions (up to at least 7) and from small data sets. All these properties are desirable in a multiscale context: overshoots and wiggles might introduce qualitatively wrong behavior, e.g., artificial hysteresis in 1p-kMC/CFD couplings; large-scale solvers often require continuous derivatives, e.g., many CFD codes incorporate the stiff chemistry using implicit ordinary differential equation (ODE) solvers;[23] finally, the small-scale models are often very costly and a large number of function evaluations are usually not affordable. While our present focus is on activity data, especially from 1p-kMC, our approach is very general and should also be of help in other fields, possibly with suitable adaptations.

This paper is structured as follows. In Sec. II, we present the methodology of our interpolation as well as the details of other versions of the Shepard interpolant that are relevant to this work. In Sec. III, we use examples to demonstrate the performance of our interpolant. The examples include a collection of analytic test functions (cf. Subsection III A) and a realistic 1p-kMC reactivity map (cf. Subsection III B). In Sec. IV, we offer conclusions on our findings and discuss future directions which might lead to an improvement of the devised methodology.

II. METHODS

Our approach belongs to the class of modified Shepard (MS) interpolation methods. They are *meshless*, scattered data interpolation methods because they require neither the input data to lay on a predefined grid nor any kind of triangulation (meshing). The defining characteristic of the MS approach is the use of a collection of local approximations of the target function, centered on the data points. The interpolant itself is evaluated as a weighted sum of these approximations.

In Sec. II A we introduce the common features of MS interpolation methods as well as one of the standard versions, which we will call distance-based MS (cf. Sec. II A). In Sec. II B, we discuss some of the limitations of distance-based MS and a way to overcome these by using an estimate of the error of the local approximations as the basis for the weighting. This constitutes what we call error-based MS, first introduced in Ref. 10. In Sec. II C we consider problems arising from the use of isotropic weighting schemes when dealing with high-dimensional functions with localized regions of rapid change. We explain how local metrics can be constructed and combined with the error estimates to solve such issues, resulting in the error-based local metric MS (EBLMMS) method. Finally, in Sec. II D, we discuss our choice of input data, i.e., the set of independent variables for which we evaluate the original function.

A. Modified Shepard interpolation

Formally, our aim is to interpolate a *target function*

$$f : \mathbb{R}^D \to \mathbb{R} \tag{1}$$

within a certain D-dimensional domain $\Omega = [x_1^{\min}, x_1^{\max}] \times \cdots \times [x_D^{\min}, x_D^{\max}] \subset \mathbb{R}$. D is the number of parameters that define the value of the function (e.g., $D = N_{\text{spec}} + 1$ in the above 1p-kMC/CFD coupling example). The interpolant is constructed using a set of points $X = \{ \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N \} \subset \Omega$, known as *nodes*, and the corresponding function values $F = \{f_1, f_2, \ldots, f_N\} \subset \mathbb{R}$, with $f_i = f(\mathbf{x}_i)$.

The formula for the modified Shepard interpolant is[20]

$$g(\mathbf{x}) = \frac{\sum_{k=1}^{N} w_k(\mathbf{x}) Q_k(\mathbf{x})}{\sum_{k=1}^{N} w_k(\mathbf{x})} = \sum_{k=1}^{N} W_k(\mathbf{x}) Q_k(\mathbf{x}), \tag{2}$$

where the *nodal functions* Q_k are local approximations of f around the nodes $\mathbf{x}_k$, w_k are the *relative interpolation weights*, and

$$W_k(\mathbf{x}) = \frac{w_k(\mathbf{x})}{\sum_{i=1}^{N} w_k(\mathbf{x})} \tag{3}$$

are the *normalized interpolation weights* or simply the *weights*.

Typically, the nodal functions are low-order polynomials, mostly first or second order. In this work, we will only consider the linear case and take

$$\begin{aligned} Q_k(\mathbf{x}) &= f_k + \mathbf{a}_k \cdot (\mathbf{x} - \mathbf{x}_k) \\ &= f_k + \sum_{i=1}^{D} a_{k,i}(x_{k,i} - x_i). \end{aligned} \tag{4}$$

The coefficients $a_{k,i}$ are obtained by minimizing the weighted sum of squared errors

$$\sum_{\substack{i=1 \\ i\neq k}}^{N} \tilde{w}_k(\mathbf{x}_i)\,(Q_k(\mathbf{x}_i) - f_i)^2\,, \tag{5}$$

where we have introduced the *relative construction weights* $\tilde{w}_k(\mathbf{x}_i)$.

The flexibility in the selection of the weights w_k and $\tilde{w}_k$ allows for the definition of different classes of Shepard interpolants. In this work, however, we only consider relative interpolation weights that satisfy

$$w_k(\mathbf{x}) \geq 0, \tag{6a}$$

$$\lim_{\mathbf{x}\to\mathbf{x}_k} w_k(\mathbf{x}) = +\infty, \tag{6b}$$

$$\lim_{|\mathbf{x}-\mathbf{x}_k|\to+\infty} w_k(\mathbf{x}) = 0. \tag{6c}$$

This guarantees that the normalized weights constitute a Shepard partition of unity, i.e.,

$$W_i(\mathbf{x}) \geq 0, \tag{7a}$$

$$W_i(\mathbf{x}_k) = \delta_{ik}, \tag{7b}$$

$$\sum_{i=1}^{N} W_i(\mathbf{x}) = 1 \qquad \forall \mathbf{x}. \tag{7c}$$

The property (7b) and the fact that $Q_k(\mathbf{x}_k) = f_k$ ensure that the interpolant goes through each of the datapoints exactly [i.e., $g(\mathbf{x_i}) = f_i\ \forall i$]. By releasing one (or both) of these conditions, the method could easily be extended to also deal with noisy input data. However, this is outside the scope of this work.

A simple ansatz for the weights would be $w_k(\mathbf{x}) = \tilde{w}_k(\mathbf{x}) = |\mathbf{x} - \mathbf{x}_k|^{-2}$, i.e., inverse-square decay, which was used for the interpolation weights by Shepard in his original work.[24] In most cases, however, such long-range weights are undesirable and we want to construct the local approximations Q_k using only points close to the corresponding node $\mathbf{x}_k$. Accordingly, we can only expect such functions to be predictive near $\mathbf{x}_k$.

For this reason, alternative versions of Shepard interpolation use weights which either (a) decay (much) faster than inverse-square at longer distances[7,21] or (b) have finite support, i.e., the weights are only non-zero in the vicinity of the nodes.[20,25,26] In the latter case, which is the one we use in this work, each of the nodal functions Q_k is built using only a subset of the nodes $\sigma_k \subset X$. Such subsets are called *stars*. Correspondingly, the range of influence of each node $\mathbf{x}_k$ is limited to a region

$$\omega_k = \{\mathbf{x} \mid w_k(\mathbf{x}) > 0\} \subset \Omega \tag{8}$$

around it. Such regions are called *clouds*. The simplest choice is to make clouds and stars isotropic. This is most easily achieved by making $w_k(\mathbf{x})$ and $\tilde{w}_k(\mathbf{x})$ non-zero only inside D-balls centered around $\mathbf{x}_k$.[25] In the seminal work of Renka,[20] relative interpolation weights are correspondingly defined according to

$$w_k(\mathbf{x}) = \frac{\left(1 - \dfrac{d_k(\mathbf{x})}{R_{w,k}}\right)_+^2}{\left(\dfrac{d_k(\mathbf{x})}{R_{w,k}}\right)^2} \tag{9}$$

and construction weights are defined according to

$$\tilde{w}_k(\mathbf{x}) = \frac{\left(1 - \dfrac{d_k(\mathbf{x})}{R_{q,k}}\right)_+^2}{\left(\dfrac{d_k(\mathbf{x})}{R_{q,k}}\right)^2}, \tag{10}$$

where $d_k(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}_k\|_2$ is the Euclidean distance between query point $\mathbf{x}$ and node $\mathbf{x}_k$ and

$$(x)_+ = \begin{cases} x, & \text{if } x \geq 0 \\ 0, & \text{if } x < 0 \end{cases}. \tag{11}$$

The radii $R_{w,k}$ are chosen such that a given number of nodes N_w fall inside each cloud σ_k. Similarly, $R_{q,k}$ are chosen such that all stars σ_k contain a given number N_q of nodes. A representation of these elements is given in Fig. 2(a).

Alternatively, clouds can be defined such that each query point is inside the clouds of exactly N_w nodes. In this formulation, distance based weights can be defined according to

$$w_k(\mathbf{x}) = \frac{\left(1 - \dfrac{d_k(\mathbf{x})}{R_w(\mathbf{x})}\right)_+^2}{\left(\dfrac{d_k(\mathbf{x})}{R_w(\mathbf{x})}\right)^2}, \tag{12}$$

where $R_w(\mathbf{x})$ depends on the query point $\mathbf{x}$ and is set to the distance to its N_wth neighbor. Such a method is implemented in the numerical subroutine library ALGLIB.[26]

N_q and N_w are the two adjustable parameters of this method. The smallest reasonable value for N_q is the number of free parameters in the nodal functions (i.e., D for linear nodal functions), in order to be able to fit them to the N_q nodes. In practice, N_q is chosen considerably larger than D to avoid overfitting of the nodal functions. N_w represents the range of validity of nodal functions and controls how much clouds overlap. *A priori*, we would expect that N_q and N_w should not differ very much, as they ultimately represent the range in which we expect the target function to be reasonably approximated by linear functions.

In what follows, we will refer to methods using construction weights from Eq. (10) and evaluation weights from either Eq. (9) or Eq. (12) as distance-based MS (DBMS) to differentiate them from the error-based methods which we define in Sec. II B.

B. Error-based modified Shepard (EBMS) interpolation

Purely distance-based weights are a natural choice when we expect target function values at a given query point to be predicted better by nodal functions of closer nodes than by nodal functions of more distant nodes. This assumption might be violated by functions with concentrated regions of large gradient changes. An illustration of this is presented in Fig. 1,

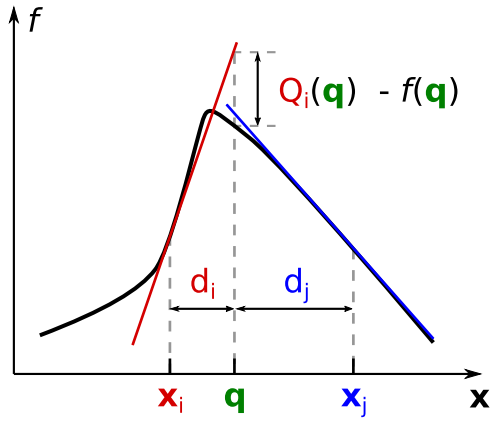

FIG. 1. The cause for overshoots with distance-based weights. The black curve represents the target function; colored straight lines represent the nodal functions Q_i and Q_j of nodes $\mathbf{x}_i$ and $\mathbf{x}_j$, respectively. The distance-based relative interpolation weight associated with Q_i at query point $\mathbf{q}$, i.e., $w_i(\mathbf{q})$, will be larger than that for $\mathbf{x}_j$, i.e., $w_j(\mathbf{q})$, even though the latter's nodal function predicts the target function value $f(\mathbf{q})$ considerably better.

where nodes $\mathbf{x}_i$ and $\mathbf{x}_j$ are located on different sides of such a region. The query point $\mathbf{q}$ is on the same side as $\mathbf{x}_j$ but closer to $\mathbf{x}_i$ ($d_i < d_j$). The prediction of nodal function Q_i (red straight line in the figure) at point $\mathbf{q}$ is much worse than that of Q_j (blue straight line), but the distance-based weight of the former will be higher [$w_i(\mathbf{q}) > w_j(\mathbf{q})$]. We therefore obtain a largely overpredicted function value even though we have a better approximation available.

Alternatively, we propose to weight nodal functions according to how well they predict the target function, e.g., inversely proportional to their error,

$$w_k(\mathbf{x}) \propto \frac{1}{|Q_k(\mathbf{x}) - f(\mathbf{x})|}. \tag{13}$$

Using an expression like Eq. (13) results in a larger weight for Q_j at query point $\mathbf{q}$ than for Q_i because $|Q_i(\mathbf{q}) - f(\mathbf{q})| \gg |Q_j(\mathbf{q}) - f(\mathbf{q})|$. Of course, the target function value and, consequently, the nodal function error $|Q_k(\mathbf{x}) - f(\mathbf{x})|$ are unknown at arbitrary query points.

The key idea behind error-based modified Shepard (EBMS) interpolation[10] is to use computationally cheap *error estimates* instead

$$\epsilon_k(\mathbf{x}) \sim |Q_k(\mathbf{x}) - f(\mathbf{x})|. \tag{14}$$

An analytic expression for such estimates can be obtained from a formula giving upper bounds of the nodal function's error,[22] which can be parametrized using the *known errors* of the nodal functions on nearby nodes.

We can formally derive the EBMS error estimates as follows: Let $\partial_i f$, with $i = 1, \ldots, D$, be the (unknown) partial derivatives of the target function f. From the theory of Taylor expansions, we have

$$\begin{aligned} f(\mathbf{x}) &= T_k(\mathbf{x}) + Z_k(\mathbf{x}) \\ &= f_k + \sum_{i=1}^{D} \partial_i f(\mathbf{x}_k)(x_i - x_{k,i}) + Z_k(\mathbf{x}), \end{aligned} \tag{15}$$

where T_k is the first-order Taylor expansion of f around $\mathbf{x}_k$ and Z_k is the residual. It can be shown that, for continuously differentiable target functions, there exists a scalar $b_{k,2} \geq 0$ such that[22]

$$|Z_k(\mathbf{x})| \leq b_{k,2}\,(d_k(\mathbf{x}))^2 \qquad \forall \mathbf{x} \in \omega_k, \tag{16}$$

where $d_k(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}_k\|^2$ as before. With this in mind, we obtain a bound for the error $|Q_k(\mathbf{x}) - f(\mathbf{x})|$ within the cloud ω_k,

$$\begin{aligned} |Q_k(\mathbf{x}) - f(\mathbf{x})| &= |Q_k(\mathbf{x}) - T_k(\mathbf{x}) - Z_k(\mathbf{x})| \\ &\leq \sum_{i=1}^{D} |\partial_i f(\mathbf{x}_k) - a_{k,i}||x_i - x_{k,i}| - |Z_k(\mathbf{x})|, \end{aligned} \tag{17}$$

where we have applied the definition of the nodal functions, Eq. (4), and the triangle inequality. Combining the fact that $\sum_{i=1}^{D} |x_i - x_{k,i}| \leq D d_k(\mathbf{x})$ with Eq. (16) and taking

$$b_{k,1} = D \max_{1 \leq i \leq N} (\{|\partial_i f(\mathbf{x}_k) - a_{k,i}|\}),$$

we obtain a formula for a bound on the errors of nodal functions

$$|Q_k(\mathbf{x}) - f(\mathbf{x})| \leq b_{k,1} d_k(\mathbf{x}) + b_{k,2}\,(d_k(\mathbf{x}))^2, \tag{18}$$

which we can use as an analytic expression for our error estimates

$$\epsilon_k(d_k(\mathbf{x})) = b_{k,1} d_k(\mathbf{x}) + b_{k,2}(d_k(\mathbf{x}))^2. \tag{19}$$

We need the error estimates ϵ_k to approximate the prediction error of Q_k. To achieve this, the coefficients $b_{k,1}$ and $b_{k,2}$ are fitted by minimizing the sum of squared differences between the error estimates and the known errors in the cloud,

$$\sum_{\mathbf{x_i} \in \omega_k} (\epsilon_k(d_k(\mathbf{x}_i)) - |Q_k(\mathbf{x}_i) - f_i|)^2. \tag{20}$$

In order to be consistent with the derivation of ϵ_k, this minimization is performed under the constraints

$$0 \leq b_{k,1}, b_{k,2} \tag{21a}$$

and

$$|Q_k(\mathbf{x}_i) - f_i| \leq \epsilon_k(d_k(\mathbf{x}_i)) \qquad \text{for all } \mathbf{x}_i \in \omega_k. \tag{21b}$$

Having obtained an expression for the error estimates ϵ_k, we can now formally define the EBMS interpolant: the nodal functions Q_k are built exactly as in DBMS [cf. Eqs. (4), (5), and (10)], but the interpolation weights are given by

$$w_k(\mathbf{x}) = \frac{\lambda(R_w, r_w; d_k(\mathbf{x}))}{\epsilon_k(d_k(\mathbf{x}))}, \tag{22}$$

where λ is a localization function

$$\lambda(R, r; d) = \begin{cases} 1, & \text{if} \quad d < R - r \\ -2\left(\dfrac{R-d}{r}\right)^2 + 3\left(\dfrac{R-d}{r}\right)^3, & \text{if} \quad R - r \leq d < R \\ 0, & \text{if} \quad R \leq d. \end{cases} \tag{23}$$

λ guarantees that w_k have finite support and that the resulting interpolant is once differentiable. The width of the transition, i.e., the region where $0 < \lambda < 1$, can be made small by choosing $r_w << R_w$, which ensures that the weights are purely error based (except for the localization). It has already been shown that error-based weights very effectively alleviate overshoots in DBMS for low-dimensional cases.[10] In Sec. III A, we show that this also holds for higher dimensional functions.

Much like in Sec. II A, the radius R_w of Eq. (22) can be chosen to depend either on the node [like $R_{k,w}$ in Eq. (9)] or on the query point [like $R_k(\mathbf{x})$ in Eq. (12)], which changes the shape of the clouds. The EBMS implementation we use in Sec. III is based on ALGLIB's DBMS implementation[26] and thus uses the query point based interpolation weights.[10]

C. Local metric based modified Shepard

In both DBMS and EBMS, the relative construction weights $\tilde{w}_k(x)$ and the interpolation weights $w_k(\mathbf{x})$ depend only on the distance $\|\mathbf{x} - \mathbf{x}_k\|_2$. This isotropy corresponds to the implicit assumption that the nodal functions approximate the function equally well in all directions. However, this may not reflect the true behavior of the target function. An example of such an anisotropic function is depicted in Fig. 2. The (linear) nodal function corresponding to node $\mathbf{x}_k$ approximates the function very well at query point $\mathbf{q}_1$, but we expect a large error at query point $\mathbf{q}_2$ (at the same distance from the node as $\mathbf{q}_1$) because the function behaves highly non-linearly in the direction $\mathbf{x}_k - \mathbf{q}_2$ (as indicated by the isolines).

To get an accurate interpolation of such a function using isotropic weights, we would need to densely sample the domains of rapid change. This becomes intractably expensive in higher dimensions even if we were able to detect these domains. An alternative is to introduce anisotropic stars and clouds. Intuitively, stars and clouds that are narrow in the directions of rapid variation and wide in the other directions are needed, as illustrated in Fig. 2. Instead of the (hyper-)spherical cloud for isotropic weights in Fig. 2(a), we thus introduce a cloud which is contracted in the $\mathbf{x}_k - \mathbf{q}_2$ direction, as shown in Fig. 2(b). This reduces the deviation of the target function from the linear nodal function within the cloud even if the cloud still has the same volume.

A straightforward way to achieve anisotropic clouds is to introduce a set of $D \times D$ matrices $M_k = (m_{k,ij})$, each associated with a node. We can then introduce a set of local distance measures

$$d_k(\mathbf{x}) := ||M_k(\mathbf{x} - \mathbf{x}_k)||_2 \tag{24}$$

and use this local metric to naturally extend the formulae from the isotropic interpolant case. Since M_k is only used to define distances, it suffices to consider *symmetric*, *positive definite* matrices. Consequently, each of them is determined by $D(D + 1)/2$ coefficients.

In this formalism, the interpolation weights are given by

$$w_k(\mathbf{x}) = \frac{\lambda(1, r_0, d_k(\mathbf{x}))}{\epsilon_k(d_k(\mathbf{x}))}, \tag{25}$$

where the local metric, Eq. (24), is used instead of the Euclidean distance. The clouds resulting from these error weights are D-dimensional *ellipsoids*. Notice that M_k already contain the information giving the size of the ellipsoids. Consequently, there is no explicit radius appearing in Eq. (25). The parameter r_0 defines the thickness of the shell in which $0 < \lambda < 1$. To make this parameter scale adequately with the dimension D of the problem, we define it as

$$r_0 = 1 - \sqrt[D]{1 - \rho_0}. \tag{26}$$

This ensures that the fraction of the cloud's volume taken by the shell is ρ_0 (the proof is given in the supplementary material). In this work, we use $\rho_0 = 0.1$.

In this local metric based MS (LMMS), the nodal functions Q_k are still constructed by minimizing Eq. (5), but now using anisotropic construction weights. We can generalize the isotropic construction weights from Eq. (10) to

$$\tilde{w}_k(\mathbf{x}) = \frac{(1 - \tilde{d}_k(\mathbf{x}))_+^2}{\left(\tilde{d}_k(\mathbf{x})\right)^2}, \tag{27}$$

where we have introduced the local distance measure $\tilde{d}_k(\mathbf{x}) = ||\tilde{M}_k(\mathbf{x} - \mathbf{x}_k)||_2$ and the $\tilde{M}_k$ construction matrices. The supports of $\tilde{w}_k$, also D-dimensional ellipsoids, define the anisotropic stars.

Intuitively, we expect the ellipsoids defining stars and clouds to be aligned and geometrically similar, i.e., they can be mapped to each other through isotropic scaling, since their orientation and shape should only depend on the local gradients of the target function. Therefore, we expect $M_k = \gamma \tilde{M}_k$ for

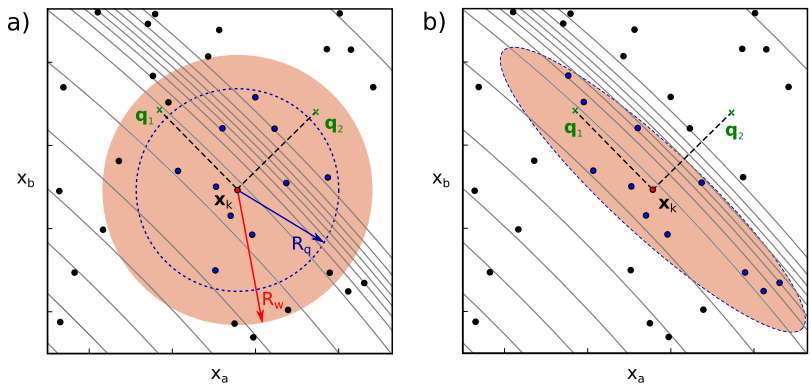

FIG. 2. Schematic representation of the different geometric elements associated with the interpolation methods described in this work. Small circles represent the nodes and gray lines represent contour levels of the target function, which accumulate in regions of rapid change. The shaded region corresponds to the cloud ω_k, i.e., the support of the interpolation weight w_k. The region delimited by the blue dashed line represents the support of construction weight $\tilde{w}_k$ and defines the star σ_k. Nodes belonging to the star are colored blue. Query points $\mathbf{q}_1$ and $\mathbf{q}_2$ are separated from $\mathbf{x}_k$ by the same distance. In the isotropic case (left panel), both regions are hyper-spheres with radii R_w and R_q, respectively. The weight associated with nodal function Q_k is the same at query points $\mathbf{q}_1$ and $\mathbf{q}_2$, i.e., $w_k(\mathbf{q}_1) = w_k(\mathbf{q}_2)$. In the anisotropic case (right panel), the supports of w_k and $\tilde{w}_k$ are coinciding hyper-ellipsoids, such that the weight associated with Q_k is positive at query point $\mathbf{q}_1$ but zero at point $\mathbf{q}_2$, i.e., $w_k(\mathbf{q}_1) > w_k(\mathbf{q}_2) = 0$.

some scalar γ. For simplicity, we will only consider $\gamma = 1$ in this work, thus restricting our study to stars and clouds of equal size and shape. Although a preliminary analysis has shown the effects of γ to be small, we postpone a detailed analysis to a future study. For the sake of simplicity of exposition, in what follows, we will often use the term *clouds* instead of *clouds and stars*, even where strict conceptual analogy with the isotropic methods would favor the term *stars*.

Constructing an error estimate for use with the anisotropic version of the interpolant is straightforward. To rationalize this, it is sufficient to notice that distances from Eq. (24) are simply the Euclidean distance in the transformed coordinates $\mathbf{x}'$ given by

$$\mathbf{x} \to \mathbf{x}' = M_k \mathbf{x}. \tag{28}$$

Thanks to this, the formal derivation of the error estimates from Sec. II B [cf. Eqs. (15)–(19)] is valid also in the transformed coordinates. This means the error estimates ϵ_k in Eq. (25) are still given by Eq. (19), but using the local distance $d_k(\mathbf{x})$ from Eq. (24). Correspondingly, the coefficients $b_{k,1}$ and $b_{k,2}$ are obtained by minimizing expression (20) under the constraints of Eq. (21) using the local distance $d_k(\mathbf{x})$.

The only element missing is then a procedure to obtain the matrices M_k. As discussed at the beginning of this section, we want local matrices that minimize the prediction error of the nodal functions. We can quantify this with a (distance) weighted sum of errors given by

$$E(M_k) = \sum_{\substack{i=1 \\ i \neq k}}^{N} \tilde{w}_k(M_k, \mathbf{x}_i) \left(Q_k(M_k, \mathbf{x}_i) - f_i\right)^2, \tag{29}$$

where we have highlighted the dependence of the nodal functions Q_k and the distance-based construction weights $\tilde{w}_k$ [cf. Eq. (25)] on M_k.

Directly minimizing Eq. (29) would often not provide useful results since clouds will tend to shrink until the number of nodes they contain is $\leq D$, trivially resulting in $E = 0$. This would lead to overfitting of the nodal functions and/or to gaps in the cloud coverage of the domain. Instead, we would like to control the number of points that fall within a cloud, as we could in the isotropic versions of the method. We would thus want to perform the minimization, constraining the search space to M_k-matrices that produce clouds with a given number of nodes in them. However, introducing a discontinuous constraint like node counts into a numerical minimization scheme is technically difficult. To alleviate this, we have decided to impose softer constraints.

We can define two estimates of the number of points in a cloud: One given by

$$\eta_-(M_k) = \sum_{i=1}^{N} \lambda(1, r_-; d_k(\mathbf{x}_i)), \tag{30}$$

which is always equal to or smaller than the actual number of nodes in the cloud $\eta_0(M_k)$, and the other given by

$$\eta_+(M_k) = \sum_{i=1}^{N} \lambda(1 + r_+, r_+; d_k(\mathbf{x}_i)), \tag{31}$$

which is always $\eta_+(M_k) \geq \eta_0(M_k)$. Proof of the inequality relations for η_+ and η_- are given in the supplementary material.

The width parameters are given by

$$r_- = 1 - \sqrt[D]{1 - \rho_{\text{soft}}}, \tag{32a}$$

$$r_+ = \sqrt[D]{1 + \rho_{\text{soft}}} - 1. \tag{32b}$$

In the limit $\rho_{\text{soft}} \to 0$, both width parameters tend to zero and the bounds η_- and η_+ both converge to the actual number of nodes in the cloud. However, for very low values of ρ_{soft}, the λ function shows very steep gradients, which would disqualify the estimates from Eqs. (30) and (31) for use in numerical optimization routines. It is necessary, therefore, to work with a finite ρ_{soft} and we have employed $\rho_{\text{soft}} = 0.2$ throughout this study.

We use Eqs. (30) and (31) to define the following optimization constraints:

$$\eta_-(M_k) \geq N_t, \tag{33a}$$

$$\eta_+(M_k) \leq 2N_t, \tag{33b}$$

which ensures that each cloud contains more than N_t but less than $2N_t$ nodes. Here, N_t is a free parameter of the interpolation method, analogous to N_q and N_w in DBMS and EBMS. In this formulation, the points in the cloud determine not only the D coefficients of the nodal functions but also the $D(D+1)/2$ coefficients of the local matrices. For this reason, it is necessary to have $N_t \geq D(D+1)/2$.

The final necessary element, avoiding another potential pitfall, is a limit on the skewness of the matrices. For node distributions, in which they are approximately aligned, clouds will tend to extend in one direction and shrink indefinitely in the others, leading to overfitted, spurious nodal functions and gaps in the cloud coverage. We can quantify the skewness of an ellipsoid as the ratio between the length of its longest and its shortest principal semi-axes, which is equal to the ratio between the absolute values of the largest and the smallest eigenvalues of the corresponding matrix. This is the *condition number*[27] of the matrix and can be approximated by

$$\kappa(M_k) = ||M_k||_1 ||M_k^{-1}||_1, \tag{34}$$

where $||\cdot||_1$ represents the 1-norm for $D \times D$ matrices. We penalize M_k with large κ by introducing a multiplicative factor to our cost function of the form

$$K(M_k) = 1 + \left(\frac{\kappa(M_k) - \kappa_0}{\kappa_0}\right)_+^p, \tag{35}$$

where κ_0 is the value at which this penalization term starts to take effect, and $p \geq 2$. In this work, we take $\kappa_0 = 100$ and $p = 4$.

In summary, local matrix coefficients are obtained by minimizing the cost function

$$C(M_k) = E(M_k)K(M_k) \tag{36}$$

subject to the constraints of Eq. (33).

For the rest of this work, we will refer to the interpolation method described in this section as error-based LMMS interpolation, EBLMMS for short. We have implemented the EBLMMS method as a python[28] package, with computationally critical parts implemented as C-extensions with the help of the SWIG[29] interface generator. Linear algebra operations on bigger matrices (mainly the matrix of node coordinates),

as well as the least squares optimization of the nodal function coefficients, cf. Eq. (5), are done using the BLAS[30] and LAPACK[31] linear algebra libraries. Both local matrix and error estimate parameter optimizations were implemented with the help of the NLopt non-linear optimization library.[32]

The initial guess for the local matrices is taken as $M_k^0 = I_D/R_0$, where I_D is the $(D \times D)$ identity matrix and

$$R_0 = \sqrt[D]{\frac{(R_k^{NN}(N_t))^D + (R_k^{NN}(2N_t))^D}{2}}, \tag{37}$$

with $R_k^{NN}(n)$ representing the distance from $\mathbf{x}_k$ to its n^{th} nearest neighbor. The algorithm used to fit the matrices is Constrained Optimization by Linear Approximations (COBYLA),[33] which is a derivative-free optimization algorithm able to handle non-linear constraints. COBYLA works by solving consecutive linear approximations of the target optimization problem. The approximations are constructed using the points in a simplex (similar to the one used in the well-known Nelder-Mead method), which is reduced in size during the optimization.

We have explored three different stopping criteria for the optimization: (a) the value of the cost function, Eq. (36), falls below some threshold value $C_{\min}$, (b) the size of the simplex (i.e., the maximum distance between vertices) is smaller than some value m_0 (which represents convergence of the change of the matrix coefficients $m_{k,ij}$), and (c) the number of cost function evaluations exceeds N_{eval}. As expected, the choice of the stopping criterion affects both the quality of the interpolant and the time to reach convergence. For criterion (a), larger values of $C_{\min}$ (obviously) reduce the CPU time for determining M_k but can affect the quality of the interpolant when the number of nodes is large (and thus the value of the cost function is small). In this work, we use $C_{\min} = 10^{-8}$ for all calculations presented in Sec. III. We find that reducing the value of this parameter further provides only negligible gains in interpolant quality. For criterion (b), we find that it is most robust to employ a scale dependent on the local environment of the node and thus select $m_0 = 0.05(R_k^{NN}(2N_t) - R_k^{NN}(N_t))$. Finally, criterion (c) is simply the backup for when the optimization takes too long and we employ $N_{\text{eval}} = 1000$ in this work.

In summary, we observe that the stopping criteria as well as the initial value of the M_k matrix can have a strong impact on both the quality and construction time of the interpolant. We believe this is a direction in which considerable improvements could be made to the method. However, as we will see in Sec. III A, the method as presented can already provide better results than other state-of-the-art methods for the class of functions which are of interest in this work.

The fitting of the error estimate parameters $b_{k,1}$ and $b_{k,2}$ is achieved using the derivative-based Method of Moving Asymptotes (MMA)[34] optimization. The initial guess for the parameters is a conservative guess

$$b_{0,1} = 10\frac{\Delta F}{\Delta X},$$
$$b_{0,2} = 10\frac{\Delta F}{(\Delta X)^2},$$

where ΔF and ΔX are the range of variation of the function values and the (maximum) range of variation of the coordinate values, respectively. We stop the optimization, when both $b_{k,1}$ and $b_{k,2}$ change by less than 10^{-5} between two consecutive iteration steps.

For clarity, we finally present a summary of the algorithm for EBLMMS construction as pseudocode in Algorithm 1.

D. Interpolant construction and quality evaluation

In Sec. III we construct interpolants for different target functions. In order to do this, we first need to select the location of the nodes. As this work targets applications in higher dimensions, using nodes on regular grids is not a viable option because the number of total nodes for a given grid resolution grows exponentially with the number of dimensions, i.e., $N = n^D$, where n is the number of nodes in each coordinate. This is the so-called *curse of dimensionality*. The use of sequences of pseudo-random vectors is also not ideal for our problem. Such sequences typically show regions with a locally high or

Algorithm 1. Construction of the EBLMMS interpolant.

procedure BUILD
  Set κ_0, N_t
  Load nodes $X = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$
  Load function values $F = (f_i, \ldots, f_N)$
  for k = 1 to N **do**
    $DX \leftarrow (\mathbf{x}_1 - \mathbf{x}_k, \ldots, \mathbf{x}_N - \mathbf{x}_k)$
    $DF \leftarrow (f_1 - f_k, \ldots, f_N - f_k)$
    Get $R_k^{NN}(N_t)$, $R_k^{NN}(2N_t)$
    $M_k^0 \leftarrow I_D/R_0$ [cf. Eq. (37)]
    $\mathbf{a}_k, M_k \leftarrow$ Minimize C(...) [cf. Eq. (36)] under constraints (33) using COBYLA
    $\{b_{k,1}, b_{k,2}\} \leftarrow$ Minimize Eq. (20) under constraints (21a) using MMA
function C(M_k, DX, DF, κ_0) ▷ Eq. (36)
  $DX_{\text{scaled}} \leftarrow M_k DX$
  Get Q_k by minimizing Eq. (5)
  Evaluate E [cf. Eq. (29)]
  Evaluate K [cf. Eq. (35)]
  return $E \cdot K$

low density of points (compared to the overall density). For our purposes, regions where nodes accumulate are undesirable as these nodes could become redundant (especially in smooth regions of the test function). Correspondingly, regions locally devoid of nodes could leave parts of the domain outside of the cloud coverage.

For these reasons, we use *low-discrepancy* sequences (also known as *quasi-random* sequences or quasi-Monte Carlo points), which are deterministic vector sequences covering a given domain more evenly than pseudo-random vector sequences. This property is often exploited to perform high-dimensional numerical integration, where it allows accurate estimates to be obtained from relatively few function evaluations. We expect this to be beneficial for the determination of the nodal functions as well since fitting the expansion coefficients of a polynomial approximation is closely related to integration. Specifically, we employ Sobol sequences,[35,36] which are very frequently used and widely implemented. For each of the cases presented in Sec. II, we construct the database for the interpolant by evaluating the target function on the first N vectors of the Sobol sequence of the corresponding dimension $\{\mathbf{x}_i^{\text{sobol}}\}_{i=1}^{N}$, for a number of different values of N.

III. RESULTS

In this section, we analyze the performance of the newly developed interpolant qualitatively and quantitatively. This is done in comparison to isotropic versions of the MS method as well as against the state-of-the-art Gaussian process regression (GPR) method, which has recently gained popularity in computational physics and materials science.[2–4] To obtain a quantitative measure of the interpolant's quality, we estimate the L1 integral of the error of the interpolation as

$$\Phi = \frac{1}{\Delta F} \frac{\sum_{i=1}^{N_{\text{test}}} |g(\mathbf{y}_i) - f(\mathbf{y}_i)|}{N_{\text{test}}}, \tag{38}$$

where g is the interpolant, f is the target function, and ΔF is a measure of the variation of the function values given by

$$\Delta F = \max_{\Omega}(f) - \min_{\Omega}(f). \tag{39}$$

In all cases, we will be using the N_{test} vectors of the Sobol sequence *immediately following* the points used as nodes, i.e., $\{\mathbf{y}_i\}_{i=1}^{N_{\text{test}}} = \{\mathbf{x}_i^{\text{sobol}}\}_{i=N+1}^{N+N_{\text{test}}}$. Taking $N_{\text{test}} = 2 \times 10^5$ was sufficient to converge Φ values for all tests.

In Sec. III A, we first use a collection of analytic functions designed specifically to emulate the challenging features this method intends to tackle and to show how the method performs for problems of different dimensionality. In Sec. III B, we then test the method by interpolating results from a realistic 1p-kMC model of heterogeneous catalysis.

A. Analytic test functions

We define two function classes, which can be used to construct related functions of arbitrary dimension. In this work, we test dimensions from $D = 2$ to $D = 7$. A representation of the 2D test functions is given in Fig. 3. All functions from both classes have small gradients across most of the domain

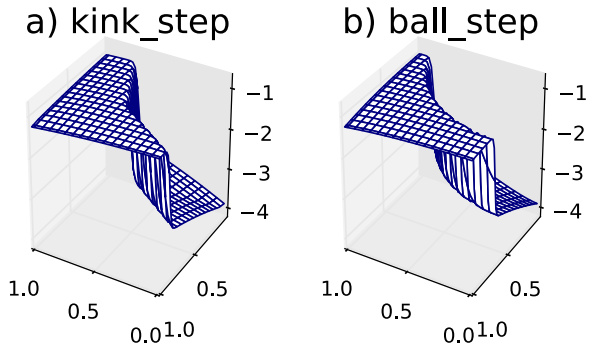

FIG. 3. Representation of the 2D version of the analytic functions used to test the interpolation method.

but show a sharp (but differentiable), step-like transition concentrated around a $(D-1)$-dimensional hypersurface. What differentiates the two function classes is the shape of this surface, which also determines the *intrinsic dimensionality* of the functions. The transition of kink_step is on the union of two $(D-1)$-dimensional half-hyperplanes which meet on a common $(D-2)$-dimensional hyperplane. Therefore, this function is intrinsically two-dimensional for all D values. ball_step functions have the transition on the surface of a D-ball and are thus fully D-dimensional. However, they are approximately one-dimensional at length scales smaller than the radius of the ball. As we will see, our locally adaptive method is capable of exploiting this fact to improve the quality of the interpolant. The detailed definition of the test functions is given in the supplementary material.

1. Analysis of anisotropic clouds

To demonstrate the working principle of the method in a concrete example, we analyze the shape of the clouds resulting from the EBLMMS interpolation of the 2D version of kink_step [cf. Fig. 3(a)] using 256 nodes. Figure 4 shows a comparison between the isotropic clouds ($N_q = N_w = 20$, left) and the anisotropic clouds from EBLMMS ($N_t = 20$, right) for selected nodes, marked by colored symbols. The clouds are represented by ellipses and are colored by the estimated

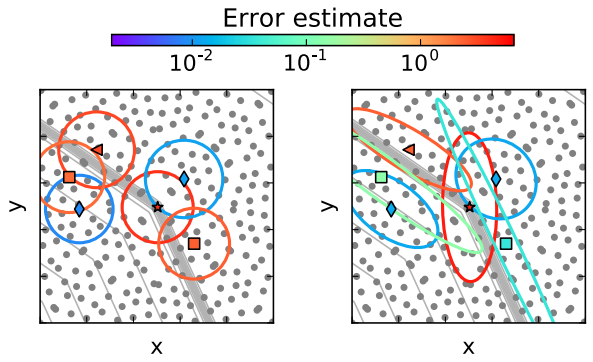

FIG. 4. Representation of clouds from selected nodes. Gray lines are contour lines of the 2D kink_step target function (cf. Fig. 3), and gray dots mark the position of the nodes. The total number of nodes is 256. The left panel corresponds to EBMS with $N_q = N_w = 20$, and the right panel corresponds to EBLMMS with $N_t = 20$. The clouds are represented by ellipses, colored according to the value of the error estimate $\epsilon_k(\mathbf{x})$. The symbols marking selected nodes (star, squares, diamonds, and triangle) are used to assist the discussion in the text.

error of the corresponding nodal functions at their boundaries. The target function is represented by gray equidistant contour lines. In the isotropic case, all clouds close to the sharp transition show a high error. In the local metric case, the clouds generally align to the expected directions and also become narrower as they get closer to the region of strong gradient changes. Moreover, for any given node, the error at the cloud boundaries is typically smaller in the latter case. An exception is the node marked with a star in the plot, located very close to the point in which the transition region bends. Since there is no satisfactory orientation for the cloud, the shape of the corresponding ellipsoid is spurious. This is where the error weighting scheme comes into play. The error associated with the mentioned point is very high compared to those of neighboring nodes, thus ensuring that the effect of this ill-defined cloud is minimized. Another exception is the point marked by a triangle. The resulting cloud aligns in the expected direction but still contains several nodes that lie across the transition region. As a consequence, the quality of the corresponding nodal function is low and, correspondingly, the associated weight is low in the flat regions, where there are multiple nodal functions which better predict the function values. The nodes marked with squares are examples of these. For these nodes, the local matrix optimization has shrunk the ellipses to let them lie fully within a single smooth sub-domain. In the isotropic case (left panel), the corresponding clouds extend across the transition region and, consequently, their nodal functions are not accurate within the respective clouds and the error estimates are large. Nodes marked with diamonds are located well within a region of smooth behavior, so the isotropic method is expected to work well. Here, the error estimates for both cases are small and the EBLMMS clouds also have a roughly circular shape.

2. Quantitative analysis

As a quantitative test of the quality of our interpolant, we interpolate each of the test functions from Fig. 3 in the domain $\Omega = [0,1]^D \subset \mathbb{R}^D$ for dimensions $D = 2, 3, \ldots, 7$ and evaluate the L1 error norm from Eq. (38) in each case. The EBLMMS method, as described above, includes several free parameters. While we leave the systematic assessment of the effect of each parameter to a future study, we concentrate on the N_t parameter here, which has a clear geometric interpretation and is analogous to the N_q parameter in traditional MS. We construct interpolants using $N_t = n_t D$, with n_t = 4, 10, 20, 50, 150, to cover a wide range of reasonable values of this parameter. For each function class, dimension D, and n_t value, we build interpolants for different numbers of nodes using the first N elements of the Sobol sequence (cf. Sec. II D). We take N as the powers of 2 between 64 and 32 768 and evaluate Φ for each interpolant using Eq. (38). In Fig. 5 we plot the best (smallest) Φ value obtained from all calculations (for a given test function class, dimension D, and number of nodes N).

To quantify the specific effect of introducing anisotropic clouds, we also calculate the L1 error resulting from the EBMS interpolation (cf. Sec. II B). EBMS interpolants are constructed using N_q values equal to the N_t values used for EBLMMS. Moreover, $N_w = N_q/2$, $N_w = N_q$, and $N_w = 2N_q$ are tested.

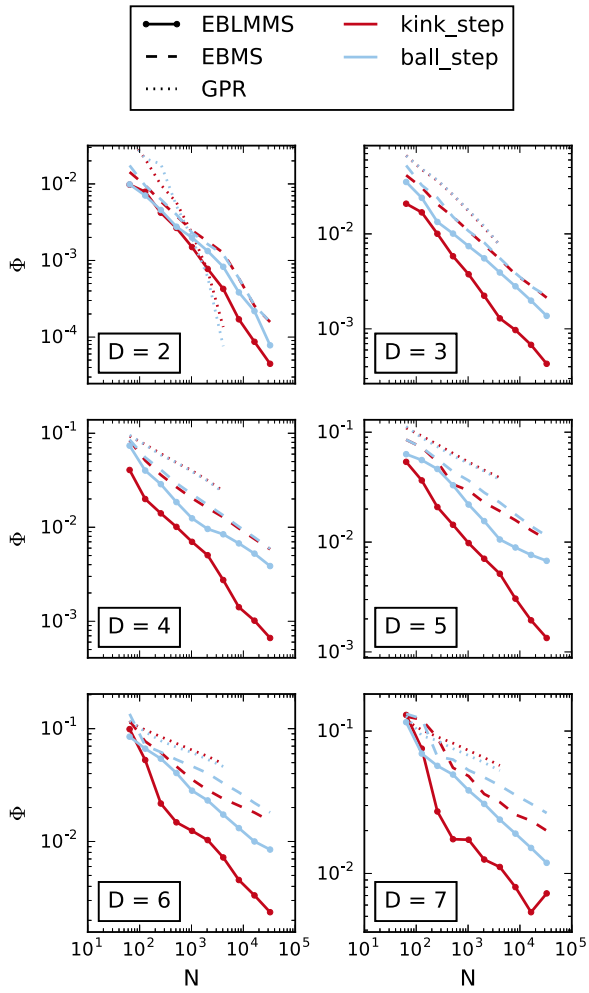

FIG. 5. Scaling of the Φ error with increasing number of nodes for the kink_step (dark, red lines) and the ball_step (light, blue lines) functions in different dimensions. The EBLMMS method (solid lines) is compared to EBMS (dashed lines) and GPR (dotted lines) methods.

The best Φ values for each (N_q,N_w) and each dimension are included in Fig. 5 as dashed lines. Naturally, the error for the anisotropic case is always smaller. As expected, it can also be seen that the relative improvement due to using EBLMMS is larger for higher dimensions. This is due to the fact that the rate of improvement of Φ with N (scaling) decreases more slowly for EBLMMS than for EBMS. It is also important to note that the benefit of using local metrics is much more pronounced for the intrinsically low-dimensional kink_step test functions than for the fully D-dimensional ball_step test functions.

To compare also to a non-MS method, we assess the performance of Gaussian process regression.[37,38] Since the popular squared exponential kernel yields very low quality results for the target functions in this work, we employ the more flexible neural network kernel, which has been shown to be able to cope with discontinuities.[37] In particular, we use the diagonally anisotropic version of this kernel, which is the most general version available in the GPy library.[39] The working equations of the methods and further details are provided in the supplementary material. Due to computer memory constraints, we only present GPR results up to node counts of N = 4096.

The GPR results are included in Fig. 5 as dotted lines. We observe that the errors for GPR are larger than those for EBLMMS in all cases except for $D = 2$ at larger node counts. In this low-dimensional case, node counts $N \gtrsim 1000$ correspond to very high node densities, which usually is not practical. Moreover, the error is already very low when GP becomes more accurate. Even for $D = 3$, the Shepard interpolations outperform GPR and the improvement continues to increase with dimensionality.

To complement the results presented in this section, we perform analogous calculations for two additional function classes, including one with rapid change of the gradient but not the function value. Since the conclusions drawn from the analysis of these functions are very similar to the ones just presented, we only present these results in the supplementary material.

3. Graphical analysis

It is important to point out that the small differences in the values of Φ in Fig. 5 do not always fully capture the qualitative improvement provided by the EBLMMS approach. To show this, we compare different interpolations of the 5D ball_step-function graphically. The wireframe plots in Fig. 6 show a number of interpolants evaluated in a 2D cut of the full 5D domain, which passes through the center and is parallel to coordinate directions 2 and 4 (cf. the supplementary material; we observe qualitatively very similar behavior for the nine other possible pairs of axes). All interpolants shown in Fig. 6 were constructed using the same 1024 nodes. As well as for EBLMMS, EBMS, and GPR, which were used in Sec. III A 2, we also present results for the traditional, distance-based isotropic DBMS (cf. Sec. II A) and for what we call DBLMMS (distance-based local metric MS), in which we use the local matrices M_k from EBLMMS to define the clouds but evaluate the interpolant using distance-based interpolation weights [i.e., analogous to Eq. (9), but using the local anisotropic distances instead of Euclidean distances]. In the figure, we highlight overshoots by changing the wireframe-color when the value of the interpolant is above the (true) maximum of the target function. The overshoot value is reported as a percentage of the step-height (which here is 3). Regions of the wireframe in which the interpolant is equal to or lower than the maximum remain colored in dark blue (i.e., according to the lower end of the colorbar).

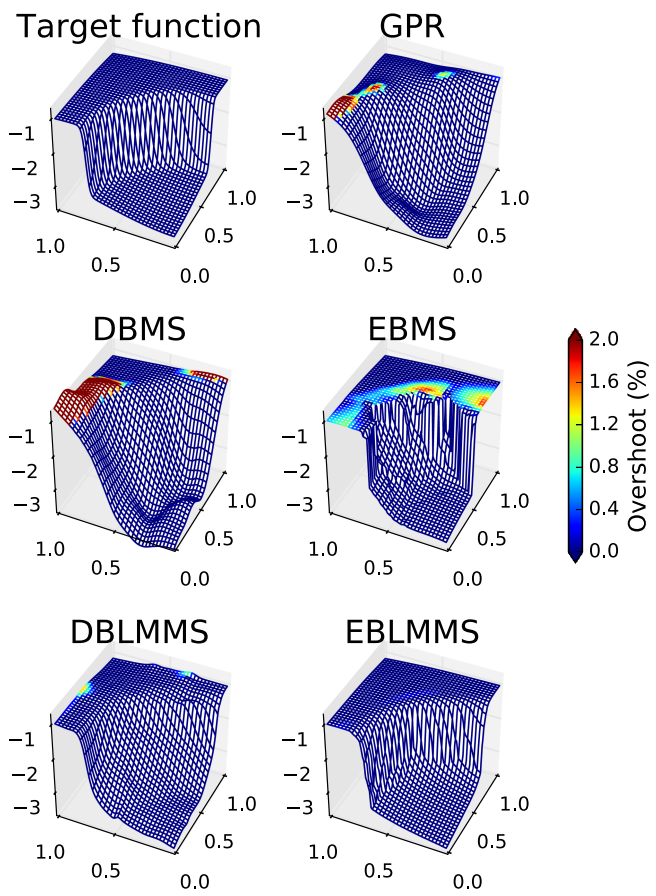

FIG. 6. Comparison of different approximations of the 5D ball_step test function, using a database of size $N = 1024$. Highlighted regions are colored according to how much the interpolant has exceeded the maximum value of the target function (as a percentage of the step height, i.e., 3).

We start by discussing the differences between methods from the MS family, as they illustrate many of the effects discussed in Sec. II. The isotropic, distance-based DBMS interpolant suffers from overshoots, which are rather large in this example. The shape of the step is barely reproduced and the function is heavily smoothed out. In the isotropic EBMS interpolant, such overshoots are considerably reduced and the interpolant matches the target function very well far from the highly non-linear region. However, this method is still unable to reproduce the shape of the transition region (the 5-ball), and some spurious features appear.

Looking at the bottom two panels, we can see that anisotropic clouds improve the interpolation quality significantly. However, the shape of the DBLMMS interpolant still shows several flaws. In particular, small oscillations appear, even in regions relatively far from the highly non-linear transition. In addition, the transition region is smoothed out considerably and its shape is not particularly well reproduced. The EBLMMS interpolant, finally, gives a much better qualitative match than any of the other cases. Even with such a small number of nodes, the shape of the highly non-linear transition region is traced very precisely. Moreover, there are no overshoots or oscillations detectable. The main source of the observed integral error Φ is a smoothing of the sharp transition, which, to some extent, is probably unavoidable using such a small dataset.

For completeness, we also show a comparison with GPR. All the spurious features observed for DBMS are present, albeit somewhat less pronounced, namely, overshoots, artificial oscillations, smoothing of the step, and an inability to properly capture the step's shape. As a final note, we point out that the good quality observed for EBLMMS is not strongly dependent on the exact choice of N_t. In the supplementary material, we provide wireframe plots comparing different values of this parameter to illustrate this.

B. Realistic 1p-kMC based data

Having demonstrated the capabilities of the EBLMMS method for a variety of analytic function classes and in multiple dimensions, we next tackle a realistic example, interpolating

the reactivity map arising from a 1p-kMC model of heterogeneous catalysis. Specifically, we use a reduced version of the well-established and frequently studied model of CO oxidation at $RuO_2(110)$ by Reuter and Scheffler.[40] The original model is based on an extensive set of Density Functional Theory (DFT) calculations and has been shown to accurately capture experimental results.[17] It considers two adsorption site types, bridge (br) and coordinately unsaturated (cus), and two surface species, CO and O. The elementary steps modeled include molecular CO adsorption and desorption, dissociative adsorption/associative desorption of O_2, irreversible CO + O reaction, and diffusional hops.

The reduced version of the model employed here was introduced by Gelß *et al.*[41] and is obtained by excluding all processes involving br sites. It has been shown that chemical kinetics is mainly controlled by the cus sites[42–44] and that the reduced model reproduces the results of the full model quantitatively for many reaction conditions.[41] Being computationally cheap, the reduced model can be evaluated for a large number of different input parameter values, which makes it a valuable test problem for our interpolation method.

The reduced model contains 7 elementary reaction steps. Single-site processes include unimolecular adsorption and desorption of CO; two-site processes, defined on pairs of nearest neighbors, include dissociative adsorption and associative desorption of O_2, CO_2 desorption as an immediate result of reaction of a pair of adsorbed CO and O, and diffusional hops of both species. The whole reaction mechanism is summarized in Table I.

In the context of 1p-kMC/CFD coupling, the TOF for this model is a function of 3 parameters, namely, the partial pressures of CO and oxygen, p_{CO} and p_{O_2}, and the temperature T. As we intend to demonstrate the capabilities of the EBLMMS method in higher dimensional problems, in this work, we will study the TOF as a function of the individual rate constants instead, i.e., we consider the 7D function

$$f : \mathbb{R}^D \to \mathbb{R},$$
$$(k^{\text{ads}}_{\text{CO}}, k^{\text{ads}}_{\text{O}_2}, k^{\text{des}}_{\text{CO}}, k^{\text{des}}_{\text{O}_2}, k^{\text{reac}}, k^{\text{diff}}_{\text{CO}}, k^{\text{diff}}_{\text{CO}}) \to \text{TOF}. \quad (40)$$

On the one hand, understanding the parametric dependence of the TOF on the rate constants is useful to perform local or global sensitivity analyses,[42,43,45] which are crucial to quantify the effects of uncertainty in the determination of rate constants (due to, e.g., DFT errors). On the other hand, and more importantly for our purposes, the characteristics of this 7D function in rate constant-space are very similar to 1p-kMC TOF maps in $(\{p_\alpha\},T)$-space. This is highlighted in models such as the one used here, in which the rate constants for non-activated adsorption are directly proportional to the corresponding partial pressures,

$$k^{\text{ads}}_\alpha = \frac{A_{\text{uc}}}{n_\alpha} \frac{p_\alpha}{\sqrt{2\pi m_\alpha k_{\text{B}} T}}, \qquad \alpha = \text{CO}, \text{O}_2, \quad (41)$$

where m_α are the molecular masses, A_{uc} is the surface area of the $RuO_2(110)$ unit cell, k_{B} is the Boltzmann constant, and n_α is a factor arising from the multiplicity of the adsorption processes included in the model. The specific values are $A_{\text{uc}} = 20.06\ \text{Å}^2$, $n_{\text{CO}} = 2$, and $n_{\text{O}_2} = 4$. Variation in temperature would correspond to concerted changes of the rate constants for the activated processes, which are also included in the domain of the 7D TOF function. For example, the rate constant for CO oxidation is given by

$$k^{\text{reac}} = \frac{k_{\text{B}} T}{h} \exp\left(-\frac{\Delta E^{\text{reac}}}{k_{\text{B}} T}\right), \quad (42)$$

where h is the Planck constant and ΔE^{reac} is the activation barrier for the CO oxidation elementary process. Moreover, the 7D TOF function of Eq. (40) also includes variations in the parameters of the model beyond those accessible by simple changes in p_{CO}, p_{O_2}, and T. For these reasons, this function is a useful proxy for a reactivity map arising from 1p-kMC containing more species.

The advantages of using such a proxy are twofold. On the one hand, the computational cost to run this 1p-kMC model is reasonably low. This allows us to perform the 200 000 + kMC calculations used as systematic test data. Just for the contour plots presented in Sec. III B 2, we needed 10 000 results for reference, which with more complex multi-species models could cost vast amounts of computational time. On the other hand, the RuO_2 CO oxidation model has been characterized in detail both in $(p_{\text{CO}}, p_{\text{O}_2}, T)$-space[40,46–49] and in rate constant space[42–44] and its behavior is well understood. This makes it ideal for testing new theoretical developments such as the one presented here. In particular, the conditions under which the model presents rapid changes in reactivity are well known.

TABLE I. List of elementary reaction events included in the reduced model for CO oxidation at RuO_2. The default value for each rate constant and its range of variation are indicated. The default values correspond to reaction conditions $T = 600$ K, $p_{\text{CO}} = p_{\text{O}_2} = 1$ bar.

Name	Expression	Default rate constant (1/s)	Range (1/s)
CO adsorption	$* \to \text{CO}*$	$k^{\text{ads}}_{\text{CO}} = 2.0 \times 10^8$	2.0×10^6–2.0×10^{10}
O_2 adsorption	$2* \to 2\text{O}*$	$k^{\text{ads}}_{\text{O}_2} = 9.7 \times 10^7$	9.7×10^5–9.7×10^9
CO desorption	$\text{CO}* \to *$	$k^{\text{des}}_{\text{CO}} = 9.2 \times 10^6$	9.2×10^4–9.2×10^8
O_2 desorption	$2\text{O} \to 2*$	$k^{\text{des}}_{\text{O}_2} = 2.8 \times 10^1$	2.8×10^{-1}–2.8×10^3
CO oxidation	$\text{O} + \text{CO} \to 2*$	$k^{\text{reac}} = 1.7 \times 10^5$	1.7×10^3–1.7×10^7
CO diffusion	$\text{CO} + * \to * + \text{CO}$	$k^{\text{diff}}_{\text{CO}} = 5.0 \times 10^{-1}$	5.0×10^{-3}–5.0×10^1
O_2 diffusion	$\text{O} + * \to * + \text{O}$	$k^{\text{diff}}_{\text{O}_2} = 6.6 \times 10^{-2}$	6.6×10^{-4}–6.6×10^0

Therefore, we can focus our study there, where interpolation becomes most challenging.

Taking this into account, we consider the rate constants corresponding to $T = 600$ K, $p_{CO} = p_{O_2} = 1$ bar, as the default (central) values. Such values lay close to the anticipated (second order) phase transition.[44] We define the limits of the interpolation domain such as to encompass a change in each of the rate constants of four orders of magnitude in total (i.e., two orders of magnitude higher and lower than the default, cf. Table I). Considering the Arrhenius-dependence of the rate constants for activated processes on energy barriers, e.g., as in Eq. (42), this accounts for changes in activation barriers ΔE of up to ~0.25 eV. For the non-activated adsorption processes, cf. Eq. (41), this corresponds to a span of variation of 4 orders of magnitude in the partial pressures. To perform the interpolation, the domain is mapped onto a logarithmic scale and into the $[0, 1]^7$ unit hypercube.

The 1p-kMC predicted CO oxidation TOF is calculated for the first 3×10^5 vectors of the 7D Sobol sequence to build the database that will later be split into nodes and test points (cf. Sec. II D). The 1p-kMC model is implemented with the help of the kmos kinetic Monte Carlo simulation package[50] using a simulation cell containing 400 individual cus sites. A total of 3×10^8 kMC steps are used for relaxation and another 5×10^8 steps for steady-state sampling. To build the interpolant, the TOF values are also log-scaled. As the upper limit of the TOF scale, we take $\mathrm{TOF}_{max} = 3 \times 10^6\ \mathrm{s}^{-1}$, which is (slightly) larger than the maximum TOF in the database. For low TOF values, kMC sampling is challenging and simulations can sometimes result in rates equal to zero, which cannot be log-scaled. However, very low TOF conditions are of little interest in catalysis, and we therefore sidestep this problem by capping the rates from below at a value of $\mathrm{TOF}_{min} = 10^{-4}\ \mathrm{s}^{-1}$. The interpolation is then performed using the transformed TOF values, in which the $[\mathrm{TOF}_{min}, \mathrm{TOF}_{max}]$ interval is logarithmically mapped to the [0, 1] interval. While we observe that the relaxation times used are sufficient to reach the kinetic steady state, we also find that a small amount of statistical noise remains in the data, as can be seen in the line plots of Fig. 8. To quantify this error, we recalculate 10^4 1p-kMC data points with different random number seeds and find an average absolute error value due to noise of $\sim 3 \times 10^{-3}$ (in the transformed TOF coordinates). However, for some points, the kMC sampling error can be as large as $\sim 3 \times 10^{-1}$.

1. Quantitative analysis and method comparison

Similar to Sec. III A, the EBLMMS interpolant is built for $N_t = 28, 70, 140, 350, 700, 1050$ and the number of nodes N equal to the powers of 2 from 64 to 32 768. The value of the error measure Φ, Eq. (38), is evaluated for the transformed TOF values. A summary of the results is presented in Fig. 7, where we again plot the smallest value of Φ (varying N_t) obtained for each N (exactly as in Fig. 5). Even for sample sizes as small as $N \approx 2 \times 10^3$, we can achieve global errors of $\Phi \approx 10^{-2}$. Considering the highly non-linear behavior of the TOF (cf. Figs. 8 and 9) and the fact that such low values of N would correspond to a regular grid with only ~3 points in each coordinate direction, we think this is a remarkably good approximation.

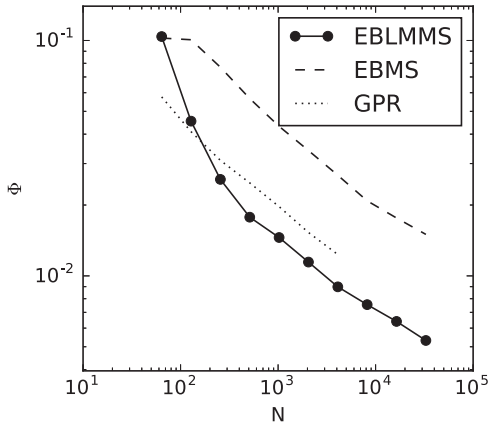

FIG. 7. Scaling of the Φ error with increasing number of nodes for the 1p-kMC model. The EBLMMS method (solid lines) is compared to EBMS (dashed lines) and GPR (dotted lines) methods.

In Fig. 7 we also present results from the EBMS (cf. Sec. II B) and GPR (cf. the supplementary material) methods for comparison. For EBMS, N_q values equal to the N_t values above were used, as well as $N_w = N_q/2$, $N_w = N_q$, and $N_w = 2N_q$. As for the case of the analytic test functions, comparing EBLMMS and EBMS shows that incorporating the local metrics produces a noticeable improvement in the quality measure. Interestingly, the error values for GPR and EBLMMS are very similar, with EBLMMS's being slightly lower except for very low node counts. However, a careful investigation reveals that important qualitative differences are not sufficiently reflected by this error measure. In Sec. III B 2, we demonstrate that GPR is not able to capture features with rapid function value and gradient changes as well as EBLMMS. Since such regions are localized in a small volume fraction of the domain, this difference is not properly captured by an integral error measure such as Φ. In Sec. III B 3, we show that this difference can have large impacts in the results of coupled 1p-kMC/CFD simulations.

2. Graphical analysis

To provide a clearer understanding of the quality and scaling of the interpolant, we present 2D line plots of the CO oxidation TOF as a function of selected rate constants in Fig. 8. Both EBLMMS and GPR are compared to a set of additional 1p-kMC data points (not included in the interpolants' input database). The curves show the values the 7D interpolants take along 1D cuts of the domain in which all but one of the parameters are kept constant. We have decided to focus on the directions of the adsorption rate constants, which can be directly associated with changes in the partial pressures p_{CO} and p_{O_2}, cf. Eq. (41), and of the CO oxidation rate constant k^{reac}, which can be associated with potential errors in the activation barrier for oxidation, cf. Eq. (42). These associated dependencies have been indicated by extra axes in the plots. In all three directions, the TOF presents a rapid, step-like change in value and gradient, which presents a challenge for the interpolation methods.

From the plots, it can already be seen that EBLMMS presents the correct qualitative behavior even at a very low

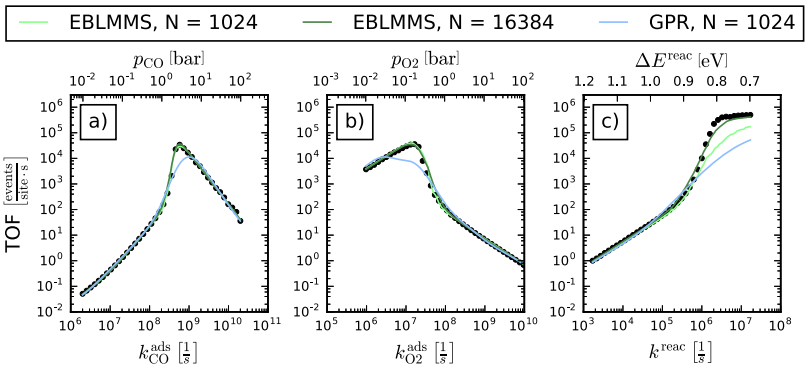

FIG. 8. 1D cuts of different 7D interpolants compared to 1p-kMC data not included in their input (dots). For each case, all but one of the rate constants were fixed at their default values (cf. Table I). Lines correspond to EBLMMS built with 1024 (light green) and 16 384 (dark green) nodes and GPR built using 1024 nodes (light blue lines). Axes indicating partial pressures associated with the adsorption rate constants [cf. Eq. (41)] and the activation barrier associated with the CO oxidation rate constants [cf. (42)] are also included.

node count of $N = 1024$. Although there are quantitative errors for high k^{reac} values, we see that EBLMMS is able to reproduce the step-like nature of the transition. For the k^{ads}_{CO} (p_{CO}) and $k^{ads}_{O_2}$ (p_{CO}) cases, there is even very good quantitative agreement. In contrast, GPR is qualitatively poorer in all cases at such (desirably) a low node count. It is never able to reproduce the shape of the curves and misses matching the high TOF peak in all cases. For the k^{reac} case, it does not even hint the step-like shape. In 1p-kMC/CFD, such interpolant deficiencies are crucial: The highest TOF values at the ridge need to be reproduced quantitatively, as this corresponds precisely to the region of highest activity targeted in catalysis research. Steep TOF increases over small pressure regions are also critical topological features that govern potential reactor instabilities or gas-phase coupled activity oscillations. If such features are washed out as by the GPR interpolant in Fig. 8, the very targets of coupled microkinetic-fluid dynamical multiscale simulations cannot be met by construction.

Further increasing the number of nodes systematically improves the quality of the EBLMMS interpolant. In Fig. 8, this is illustrated for $N = 16\,384$ nodes, in which very good quantitative agreement is found for all the cases shown. In the supplementary material, we have included a plot similar to Fig. 8 presenting examples for other choices of number of nodes N. There it can be seen that the quality of GPR remains low even up to 4096 nodes (the highest value analyzed).

Another impression of the behavior of the interpolation method can be gained by looking at the contour plots presented in Fig. 9. The plots show TOF contour plots across cuts of the 7D domain in which all rate constants except for k^{ads}_{CO} and k^{reac} are now kept fixed at their default values. The panels on the left (the same data top and bottom) show 1p-kMC data explicitly calculated on a (100×100) grid. The central panels show the TOF values predicted by the 7D EBLMMS interpolant built using $N = 1024$ (top) and $N = 8192$ (bottom) nodes. The panels on the right show the corresponding absolute value of the

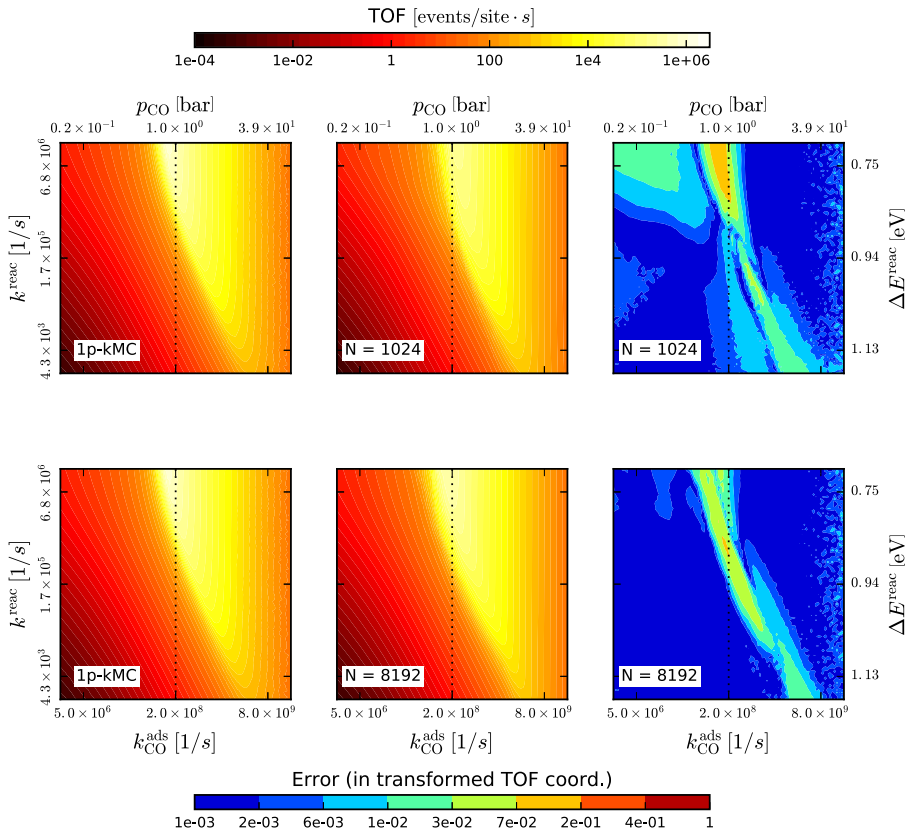

FIG. 9. Contours of CO oxidation TOF explicitly calculated with the 1p-kMC model (left, plot repeats top and bottom), compared with interpolations generated by EBLMMS using 1024 nodes (center, top) and 8192 nodes (center, bottom). The panels on the right present the error, measured in the transformed TOF coordinates. The domain is cut through its center (where all rate constants have their default value) and parallel to the axes corresponding to the CO adsorption and CO oxidation rate constants. The dotted lines mark the conditions plotted in Fig. 8(c). Additional axes have been added to show the correspondence between changes in k^{ads}_{CO} and p_{CO} (k^{reac} and ΔE^{reac}).

error, calculated in the transformed (logarithmic) TOF scale. The plots demonstrate that even for $N = 1024$ the qualitative features of the target function are remarkably well reproduced. For $N = 8192$, the 1p-kMC and EBLMMS results are almost indistinguishable by eye. In addition, it can be observed that the errors are concentrated in the regions of rapid gradient change and are very small in the rest of the domain. The conditions of Fig. 8(c) are marked in the contour plots by a dotted line. We can rationalize the difficulty of reproducing TOF values for high k^{reac} by noting that these conditions fall into the TOF peak observed in the contours (i.e., located in the top-center region). This peak is relatively localized and thus very difficult to predict using only a small number of nodes. In the supplementary material, an analogous contour plot with GPR predictions using 1024 nodes is presented. It shows that the qualitative behavior is poorer than the corresponding EBLMMS interpolant.

3. A stagnation flow example

To further underscore our general remarks on the required accuracy of TOF interpolants in coupled 1p-kMC/CFD simulations, we consider an isothermal and stationary stagnation flow,[51] where a mixture of CO, O_2, and argon streams from a sieve-like inlet against a disk-shaped catalyst. This is a suitable reactor model for flat-faced single-crystal model catalysts as in the reduced $RuO_2(110)$ 1p-kMC CO oxidation model, which we will continue to use for this demonstration. As illustrated in Fig. 10, the geometry of the axisymmetric reactor problem is fully determined by the vertical height L of the inlet. For the calculations, we employ $L = 3$ cm and an inlet velocity of 10 cm/s. The oxygen partial pressures at the inlet is chosen as $p_{\text{O}_2}^{\text{inlet}} = 1\text{bar}$, the CO partial pressure $p_{\text{CO}}^{\text{inlet}}$ varies between one and 4 bars, and 50% of the mixture is always argon, i.e., $p_{\text{Ar}}^{\text{inlet}} = p_{\text{O}_2}^{\text{inlet}} + p_{\text{CO}}^{\text{inlet}}$. We obtain numerical solutions to the resulting one-dimensional boundary value problem using our previously employed perturbative approach (see the supplementary material in Ref. 11) and a stagnation flow solver.[8,9]

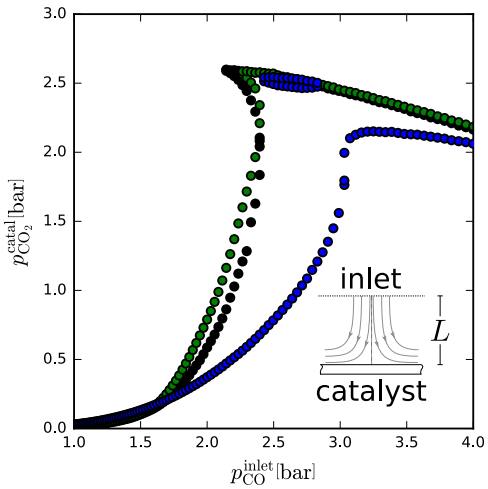

FIG. 10. Steady-state CO_2 partial pressure immediately over the catalyst as a function of the CO partial pressure of the inlet of the stagnation flow reactor schematically depicted in the inset. Detailed conditions of the simulation are summarized in the text. We compare the results obtained using as catalyst boundary condition a reference solution (black) with those using the EBLMMS-based (green) and GPR-based (blue) interpolants discussed in Fig. 8. Both surrogates are built using a total of 1024 nodes.

The interpolated TOF enters the stagnation flow equations as a non-linear boundary condition. As a reference, we use a dense regular 2D grid of 100×100 1p-kMC data points in the pressure range $(p_{\text{CO}}, p_{\text{O}_2}) \in [10^{-2}, 10^{2}]^2$, which we interpolate piecewise linearly. Against this reference, we assess the performance of the high-dimensional GPR and EBLMMS surrogate models with a low number of interpolation nodes $N = 1024$ (cf. Fig. 8), i.e., we employ the full 7D interpolations and Eq. (41) to obtain the partial pressures. This way, we can assess the impact of interpolation errors in multidimensional TOF maps onto CFD simulation results. Such multidimensional TOF maps naturally arise from 1p-kMC models with many reactive species but also the here considered TOF map of the individual rates might be beneficial in practice, e.g., when fitting reaction parameters to experimental reactor data.

Figure 10 shows the CO_2 partial pressure directly above the catalyst $p_{\text{CO}_2}^{\text{catal}}$, which would be the central experimental observable, e.g., when employing planar laser induced fluorescence measurements,[11,52] and which is related to the catalytic activity. For our reference calculations (black dots), we find a low, monotonically increasing activity ($p_{\text{CO}_2}^{\text{catal}}$) for $p_{\text{CO}}^{\text{inlet}}$ below the stoichiometric ratio. For high $p_{\text{CO}}^{\text{inlet}}$, the activity is higher and monotonically decreasing. These two regimes are connected by a relatively narrow regime for $p_{\text{CO}}^{\text{inlet}}$ slightly above the stoichiometric ratio, in which we find multiple stationary solutions that could give rise to gas-phase coupled oscillatory behavior of the catalytic activity. The EBLMMS-based model reproduces the behavior of the reference calculation with only minor quantitative differences. In contrast, the GPR-based model deviates largely from the reference calculation for most of the CO pressure range. Particularly at the phase transition, it provides a qualitatively wrong picture. Multiple solutions appear where they should not be, while the true regime with multiple solutions is missed. This is a direct consequence of its inability to properly trace the steep TOF variations predicted by the 1p-kMC model.

IV. SUMMARY AND OUTLOOK

We have presented an interpolation technique able to faithfully approximate high-dimensional functions with locally rapid changes, such as those arising from first-principles kinetic Monte Carlo models for heterogeneous catalysis. Exploiting the fact that such functions often show locally low-dimensional behavior, small global errors can be achieved with this error-based local metric modified Shepard (EBLMMS) method even with modest numbers of function evaluations. Furthermore, the method successfully suppresses undesired behavior, such as oversmoothing and artificial wiggles.

Compared with existing methods from the Shepard family as well as with state-of-the-art Gaussian process regression, our approach proved to be superior for tested target functions ranging from analytic test cases up to numerical 1p-kMC data. In higher dimensions in particular, our combination of a locally changing metric and error estimate based blending

proved to be advantageous. The superior accuracy was also shown to be very important when building surrogates for use in 1p-kMC/CFD coupling.

Another strength of the approach is its basis in geometrical considerations and a conceptually simple mathematical description. Most input parameters either have an intuitive geometrical meaning or can be interpreted as an error, either in the error function or in the coefficients of the local metric. Nevertheless, methodologically there is still room for improvement. Most importantly, this concerns the determination of the ellipsoids which define the clouds. A better initial guess of the local metric and the exploitation of more efficient and robust optimization algorithms are desirable. Furthermore, data structures can be developed which exploit the finite support of the interpolation weights and make evaluation times scale sub-linearly with the number of nodes. Finally, the cheaply available error estimates could be used for parameter set optimization. Instead of having to split the calculated data into construction and test sets, the parameters could be obtained by minimizing the error estimates at points in which the true function value is unknown.

SUPPLEMENTARY MATERIAL

See supplementary material for the following: (a) the justification of Eqs. (26), (30), and (31); (b) the exact definition of the analytic test functions; (c) a description of the GPR method used; (d) error scaling plots for two additional test function classes; (e) a graphical comparison of EBLMMS interpolants with different N_t values; (f) a plot similar to Fig. 8 for interpolants built with different numbers nodes; and (g) a graphical comparison between EBLMMS and GPR for the 1p-kMC model.

ACKNOWLEDGMENTS

We gratefully acknowledge support from the German Research Council (DFG) and the TUM Faculty Graduate Center Chemistry, as well as generous computing time at the Leibniz Rechenzentrum (LRZ) of the Bavarian Academy of Sciences. S.M.'s research is carried out in the framework of Matheon supported by the Einstein Foundation Berlin. J.M.L. warmly thanks A. Martinez and F. Busnengo for their hospitality during his stay at Instituto de Física Rosario, Argentina, where parts of this work were carried out.

[1] J. Behler, J. Phys.: Condens. Matter **26**, 183001 (2014).
[2] A. P. Bartók, M. C. Payne, R. Kondor, and G. Csányi, Phys. Rev. Lett. **104**, 136403 (2010).
[3] T. Stecher, N. Bernstein, and G. Csányi, J. Chem. Theory Comput. **10**, 4079 (2014).
[4] L. Mones, N. Bernstein, and G. Csányi, J. Chem. Theory Comput. **12**, 5100 (2016).
[5] H. F. Busnengo, A. Salin, and W. Dong, J. Chem. Phys. **112**, 7641 (2000).
[6] D. Strobusch and C. Scheurer, J. Chem. Phys. **140**, 074111 (2014).
[7] M. A. Collins, Theor. Chem. Acc. **108**, 313 (2002).
[8] S. Matera and K. Reuter, Catal. Lett. **133**, 156 (2009).
[9] S. Matera and K. Reuter, Phys. Rev. B **82**, 085446 (2010).
[10] S. Matera, M. Maestri, A. Cuoci, and K. Reuter, ACS Catal. **4**, 4081 (2014).
[11] S. Matera, S. Blomberg, M. J. Hoffmann, J. Zetterberg, J. Gustafson, E. Lundgren, and K. Reuter, ACS Catal. **5**, 4514 (2015).
[12] M. Votsmeier, Chem. Eng. Sci. **64**, 1384 (2009).
[13] M. Votsmeier, A. Scheuer, A. Drochner, H. Vogel, and J. Gieshoff, Catal. Today **151**, 271 (2010).
[14] S. Pope, Combust. Theory Modell. **1**, 41 (1997).
[15] A. Varshney and A. Armaou, Chem. Eng. Sci. **60**, 6780 (2005).
[16] A. Varshney and A. Armaou, Comput. Chem. Eng. **32**, 2136 (2008).
[17] K. Reuter, D. Frenkel, and M. Scheffler, Phys. Rev. Lett. **93**, 116105 (2004).
[18] K. Reuter, Catal. Lett. **146**, 541 (2016).
[19] J. M. Lorenzi, S. Matera, and K. Reuter, ACS Catal. **6**, 5191 (2016).
[20] R. J. Renka, ACM Trans. Math. Software **14**, 139 (1988).
[21] S. Vijayakumar, A. D'souza, and S. Schaal, Neural Comput. **17**, 2602 (2005).
[22] C. Zuppa, Appl. Numer. Math. **49**, 245 (2004).
[23] M. Maestri and A. Cuoci, Chem. Eng. Sci. **96**, 106 (2013).
[24] D. Shepard, in *Proceedings of the 1968 23rd ACM National Conference, ACM '68* (ACM, New York, NY, USA, 1968), pp. 517–524.
[25] R. Franke and G. Nielson, Int. J. Numer. Methods Eng. **15**, 1691 (1980).
[26] S. Bochkanov, "ALGLIB," www.alglib.net.
[27] J. W. Demmel, *Applied Numerical Linear Algebra* (SIAM, Philadelphia, PA, USA, 1997).
[28] Python Software Foundation, "Python 2.7," http://www.python.org/.
[29] See http://www.swig.org for SWIG: Simplified wrapper and interface generator.
[30] L. S. Blackford, J. Demmel, J. Dongarra, I. Duff, S. Hammarling, G. Henry, M. Heroux, L. Kaufman, A. Lumsdaine, A. Petitet, R. Pozo, K. Remington, and R. C. Whaley, ACM Trans. Math. Software **28**, 135 (2002).
[31] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK Users' Guide*, 3rd ed. (SIAM, Philadelphia, PA, 1999).
[32] S. G. Johnson, "The NLopt nonlinear-optimization package," http://ab-initio.mit.edu/nlopt.
[33] M. J. D. Powell, in *Advances in Optimization and Numerical Analysis*, edited by S. Gomez and J.-P. Hennart (Kluwer Academic, Dordrecht, 1994), pp. 51–67; M. Powell, Acta Numer. **7**, 287 (1998).
[34] K. Svanberg, SIAM J. Optim. **12**, 555 (2002).
[35] I. Sobol, USSR Comput. Math. Math. Phys. **7**, 86 (1967).
[36] P. L'Ecuyer and C. Lemieux, in *Modeling Uncertainty: An Examination of Stochastic Theory, Methods, and Applications*, edited by M. Dror, P. L'Ecuyer, and F. Szidarovszky (Springer, Boston, MA, 2005), pp. 419–474.
[37] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*, Adaptive Computation and Machine Learning Series (MIT Press, 2005).
[38] D. J. C. Mackay, *Information Theory, Inference and Learning Algorithms*, 1st ed. (Cambridge University Press, 2003).
[39] GPy, "GPy: A gaussian process framework in python," http://github.com/SheffieldML/GPy, since 2012.
[40] K. Reuter and M. Scheffler, Phys. Rev. B **73**, 045433 (2006).
[41] P. Gelß, S. Matera, and C. Schütte, J. Comput. Phys. **314**, 489 (2016).
[42] H. Meskine, S. Matera, M. Scheffler, K. Reuter, and H. Metiu, Surf. Sci. **603**, 1724 (2009).
[43] S. Döpking and S. Matera, Chem. Phys. Lett. **674**, 28 (2017).
[44] M. J. Hoffmann, F. Engelmann, and S. Matera, J. Chem. Phys. **146**, 044118 (2017).
[45] J. E. Sutton, W. Guo, M. A. Katsoulakis, and D. G. Vlachos, Nat. Chem. **8**, 331 (2016).
[46] B. Temel, H. Meskine, K. Reuter, M. Scheffler, and H. Metiu, J. Chem. Phys. **126**, 204711 (2007).
[47] S. Matera, H. Meskine, and K. Reuter, J. Chem. Phys. **134**, 064713 (2011).
[48] D.-J. Liu and J. W. Evans, J. Chem. Phys. **142**, 134703 (2015).
[49] G. J. Herschlag, S. Mitran, and G. Lin, J. Chem. Phys. **142**, 234703 (2015).
[50] M. J. Hoffmann, S. Matera, and K. Reuter, Comput. Phys. Commun. **185**, 2138 (2014).
[51] R. Kee, M. Coltrin, and P. Glarborg, *Chemically Reacting Flow, Theory and Practice* (Wiley, Hoboken, NJ, 2003).
[52] J. Zetterberg, S. Blomberg, J. Gustafson, Z. W. Sun, Z. S. Li, E. Lundgren, and M. Aldén, Rev. Sci. Instrum. **83**, 053104 (2012).