SIAM J. SCI. COMPUT.
Vol. 41, No. 6, pp. A3624–A3648

AN ADAPTIVE MINIMUM SPANNING TREE MULTIELEMENT METHOD FOR UNCERTAINTY QUANTIFICATION OF SMOOTH AND DISCONTINUOUS RESPONSES[*]

YOUS V. HALDER[†], BENJAMIN SANDERSE[‡], AND BARRY KOREN[§]

Abstract. A novel approach for nonintrusive uncertainty propagation is proposed. Our approach overcomes the limitation of many traditional methods, such as generalized polynomial chaos methods, which may lack sufficient accuracy when the quantity of interest depends discontinuously on the input parameters. As a remedy we propose an adaptive sampling algorithm based on minimum spanning trees combined with a domain decomposition method based on support vector machines. The minimum spanning tree determines new sample locations based on both the probability density of the input parameters and the gradient in the quantity of interest. The support vector machine efficiently decomposes the random space in multiple elements, avoiding the appearance of Gibbs phenomena near discontinuities. On each element, local approximations are constructed by means of least orthogonal interpolation, in order to produce stable interpolation on the unstructured sample set. The resulting minimum spanning tree multielement method does not require initial knowledge of the behavior of the quantity of interest and automatically detects whether discontinuities are present. We present several numerical examples that demonstrate accuracy, efficiency, and generality of the method.

Key words. multielement method, stochastic collocation, support vector machines, minimum spanning trees, uncertainty quantification, discontinuity detection

AMS subject classifications. 35, 76

DOI. 10.1137/18M1219643

1. Introduction.

Uncertainty quantification (UQ) has become increasingly important for complex engineering applications. Determining and quantifying the influence of parametric and model-form uncertainties is essential for a wide range of applications: from turbulent flow phenomena [1, 2], aerodynamics [3], biology [4, 5], to design optimization [6, 7, 8]. We are interested among others in liquid-impact problems [9, 10, 11].

For problems which have a complex underlying model, one often uses so-called nonintrusive methods, also known as *sampling* methods. The model is solved deterministically a number of times, and a stochastic solution is constructed using these deterministic samples. A well-known sampling method for propagating uncertainties through a model is the *Monte Carlo* method [12]. Despite its easy implementation and wide applicability, the Monte Carlo method suffers from slow convergence with increasing number of model evaluations, when approximating the quantity of interest (QoI). As a consequence of this slow convergence rate, many samples are required for obtaining high quality stochastic solutions. Even though Monte Carlo methods are

[*] Submitted to the journal's Methods and Algorithms for Scientific Computing section October 8, 2018; accepted for publication (in revised form) September 20, 2019; published electronically November 14, 2019.
https://doi.org/10.1137/18M1219643

Funding: This work is part of the research programme "SLING" (Sloshing of Liquefied Natural Gas), which is (partly) financed by the Netherlands Organisation for Scientific Research (NWO).

[†] Scientific Computing, Centrum Wiskunde and Informatica, Amsterdam, Noord-Holland, 1098 XG, The Netherlands (y.van.halder@cwi.nl).

[‡] Centrum Wiskunde and Informatica, Amsterdam, 1090 GB, The Netherlands (b.sanderse@cwi.nl).

[§] Mathematics and Computer Science, Eindhoven University of Technology, Eindhoven, Noord-Brabant, 5600 MB, The Netherlands (b.koren@tue.nl).

often superior for high-dimensional problems, proper alternatives have been proposed to speed up convergence in many cases. As an alternative to Monte Carlo methods, *stochastic collocation* (SC) methods [13, 14, 15] were introduced, replacing the slow convergence of Monte Carlo by an exponential convergence rate. The introduction of SC methods resulted in a decrease of required samples to achieve a certain accuracy in comparison to Monte Carlo methods. For a smooth QoI as a function of the uncertainties, fast convergence is achieved. However, if the QoI is highly nonlinear or discontinuous, *Gibbs phenomena* [16] may occur, which deteriorate the accuracy globally. To avoid the occurrence of Gibbs phenomena, several alternatives to the SC methods were introduced [17, 18], but they focus solely on discontinuous QoIs, leading to a significant increase in the number of samples needed for approximating smooth QoIs. One interesting method is the *multielement SC* method (ME-SC) [19, 20]. The idea of ME-SC is to decompose the domain, spanned by the uncertainties, into smaller nonoverlapping elements, in each of which the QoI is amenable for using an SC method. Gibbs phenomena still appear in the elements where there is a discontinuity in the QoI, but they are confined to these specific elements. Improving the multi-element approach is an active field of research and focuses on more efficient and robust domain decomposition. Jakeman, Narayan, and Xiu [21] proposed the minimal multielement method, which uses discontinuity detection based on polynomial annihilation to detect discontinuities, and divides the domain along the discontinuities. As a result, the Gibbs phenomena are removed completely. Although this discontinuity detection algorithm is accurate, the total number of samples needed to determine the discontinuity location may easily be still too high if sampling the model is expensive. Therefore, a discontinuity detection algorithm which performs well for a lower number of samples was proposed by Gorodetsky and Marzouk [22]. This discontinuity detection uses polynomial annihilation in combination with *support vector machines* (SVMs) to divide the domain into elements along the so-called SVM classification boundary. Even though both discontinuity detection algorithms [21, 22] perform well, using them for approximating smooth QoIs can become prohibitively expensive, as both methods focus solely on finding the discontinuities. It is often unknown in advance if a QoI is smooth or discontinuous. Hence, choosing a method which is suited for either smooth or discontinuous responses is often not recommended. Our goal is to create a surrogate model that works for both smooth and discontinuous QoIs and which requires no initial knowledge about the QoI.

A novel domain decomposition method in combination with adaptive sampling of the QoI is therefore proposed for constructing this surrogate. The adaptive sampling procedure in our method is based on *minimum spanning trees* (MSTs) [23, 24], which add new sample points at places which are associated with a high probability density and/or where the QoI changes rapidly. The adaptively placed samples are classified and an SVM [22, 25, 26, 27, 28] is used to obtain a classification boundary, which serves as an approximation for the discontinuity location. The decomposition of the random space in this way leads to elements on which each local QoI is amenable for interpolation without Gibbs phenomena. For constructing a surrogate model in each element a *least orthogonal interpolant* [29] is employed, which is suited for interpolation on the scattered data points that we obtain with our adaptive sampling. Our proposed method is abbreviated as the MST-ME (MST multielement) method and is designed mainly for the purpose of uncertainty propagation. However, when assuming uniformly distributed probability density functions for the input parameters, the MST-ME method is also well suited to obtain a parametric solution of the partial differential equation (PDE) under consideration. This will also be illustrated in this paper. Although the MST-ME method is designed to work in any number of dimensions, very high-dimensional problems are not the main focus of our work, but rather the accurate treatment of discontinuities.

This paper is outlined as follows: section 2 briefly introduces the problem. Section 3 introduces the MST-ME method in detail. Finally, section 4 demonstrates efficiency and accuracy of our method when applied to analytical test cases. Complex test cases related to sloshing impact problems, i.e., shallow water dam break and a three-dimensional (3D) dam break, are also studied.

2. Problem description. Quantifying the effects of uncertainties in computational engineering typically consists of three steps: (i) each of the input uncertainties is characterized in terms of a probability density function (PDF), which follows from observations or physical evidence; (ii) the uncertainties are propagated through the model; (iii) the outputs are postprocessed, where the QoI is expressed in terms of its statistical properties. In the present work we focus on the propagation step, and the input distributions are assumed to be given. The goal is to solve the following stochastic problem:

$$\mathcal{L}(v;\mathbf{z}) = 0 \ , \tag{2.1}$$

where $I_{\mathbf{z}}$ is the support set of the uncertain inputs $\mathbf{z}$, referred to as a *random space*, $\mathbf{z} = (z_1, \dots, z_d) \in I_{\mathbf{z}}$ is the d-dimensional vector containing uncertain inputs, $v = v(\mathbf{z})$ is the solution and $\mathcal{L}$ an operator, representing the model. The operator $\mathcal{L}$ can be a nonlinear partial differential operator, or any mathematical model that relates input $\mathbf{z}$ to the solution $v(\mathbf{z})$. The QoI u is calculated by applying an operator Q to the solution v, i.e., $u(\mathbf{z}) = Q(v(\mathbf{z}))$. Both continuous QoIs $u \in C^0(I_{\mathbf{z}})$, and discontinuous QoIs $u \notin C^0(I_{\mathbf{z}})$, are considered in this paper. The uncertain inputs are assumed to be characterized by a joint PDF $\rho(\mathbf{z})$. The stochastic problem is solved nonintrusively by sampling the model (2.1) at different locations $\mathbf{z}_i$ in the random space, i.e.,

$$\mathcal{L}(v_i;\mathbf{z_i}) = 0 \ \Rightarrow u_i = u(\mathbf{z}_i) = Q(v_i) \ , \tag{2.2}$$

where $v_i = v(\mathbf{z}_i)$ and $u_i = u(\mathbf{z}_i)$ are the sampled solution and QoI at the collocation node $\mathbf{z}_i$, respectively. Since the sampling is nonintrusive, black-box solvers for the operator $\mathcal{L}$ can be used. In this paper we are interested in finding the entire functional relation u as a function of the uncertainties $\mathbf{z}$, in terms of a surrogate model. A surrogate model $\tilde{u}$ of u is constructed by interpolation, such that

$$\tilde{u}(\mathbf{z}) \approx u(\mathbf{z}) \ \text{ for all } \ \mathbf{z} \in I_{\mathbf{z}} \ . \tag{2.3}$$

When u is smooth, it is possible to construct an approximation $\tilde{u}$ which converges exponentially fast to the exact solution u. However, if the QoI exhibits highly nonlinear or discontinuous behavior, then the accuracy of the approximation deteriorates globally, due to Gibbs phenomena. Multielement methods divide the random space $I_{\mathbf{z}}$ into a set of N_E smaller elements E_i, such that the negative impact of the Gibbs phenomena is confined to a limited number of elements surrounding the discontinuity. The elements E_i are nonoverlapping and span the entire random space, i.e.,

$$\cup_{i=1}^{N_E} E_i = I_{\mathbf{z}} \ \text{ and } \ E_i \cap E_j = \emptyset \quad \text{if} \quad i \neq j \ . \tag{2.4}$$

A local surrogate $\tilde{u}^{(i)}$ is constructed in each E_i:

$$\tilde{u}^{(i)}(\mathbf{z}) \approx u(\mathbf{z}) \ \text{ for all } \ \mathbf{z} \in E_i \ . \tag{2.5}$$

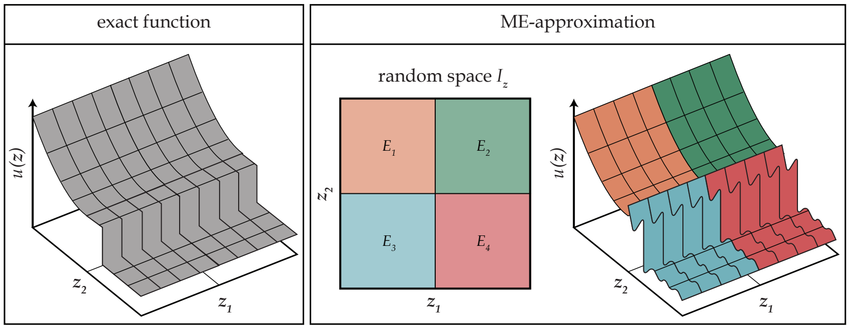

FIG. 1. *Example of the standard multielement approach for approximating a two-dimensional (2D) function exhibiting a discontinuity.*

The global surrogate is given by patching the local surrogate models:

$$\tilde{u}(\mathbf{z}) = \sum_{i=1}^{N_E} \tilde{u}^{(i)}(\mathbf{z})\mathcal{I}_{E_i}(\mathbf{z}) \ , \tag{2.6}$$

where $\mathcal{I}_{E_i}(\mathbf{z})$ is the indicator function satisfying $\mathcal{I}_{E_i}(\mathbf{z}) = 1$ if $\mathbf{z} \in E_i$ and 0 otherwise. Standard multielement methods utilize a tensor construction of hypercubes for defining the elements E_i [19, 20]. While such a tensor construction removes the global effect of Gibbs phenomena, they can still appear locally in several elements.

An example of the standard tensorized multielement approach for the approximation of a 2D function is shown in Figure 1. Gibbs phenomena appear in the elements $E_{3,4}$, where a discontinuity is present in the exact function.

3. MST-ME. The art of constructing an efficient ME-SC method lies in the choice of sampling points $\mathbf{z}_i$, the choice of the elements E_i, and the reconstruction of the local approximations $\tilde{u}^{(i)}$. These are the focuses of this paper. Accordingly, the MST-ME method introduced here is divided into three stages: (I) sampling, (II) domain decomposition, and (III) local approximation construction:

I *choice of* z_i: adaptive sampling of the QoI while taking into account both smooth and discontinuous regions, and the underlying PDF;

II *choice of* E_i: division of the random space into a minimal number of elements, such that the QoI is smooth within each element;

III *construction of* $\tilde{u}^{(i)}$: interpolation of the samples, while producing a stable interpolant.

The QoI is adaptively sampled (I) by taking into account both the PDF and the QoI gradient information. The samples are distributed among different classes, such that within each class the local QoI is smooth. The classified samples are given as input to the domain classification algorithm (II). Instead of using a tensor based domain decomposition, the random space is divided into a minimal number of elements, in which the QoI is amenable to interpolation without Gibbs phenomena. This domain decomposition methodology was already introduced in [21], but in that work the number of samples required for determining proper elements was too high. In contrast, our domain decomposition method (II) uses the samples from the sampling algorithm (I) for determining proper elements, without the need to perform additional sampling. Local approximations (III) are constructed in each element, by using the

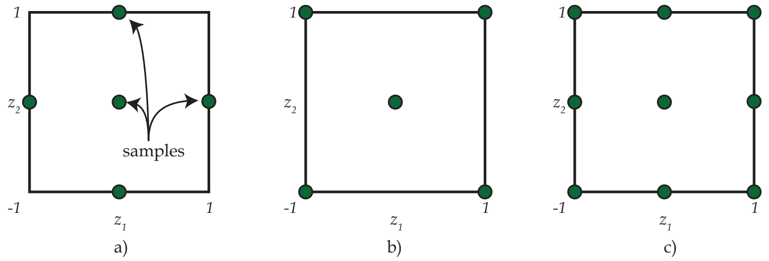

Fig. 2. *Initial sample location configurations.* (a) *One sample in the middle and one sample at each face center.* (b) *One sample in the middle and one sample at each corner of the domain.* (c) *The combination of* (a) *and* (b).

least orthogonal interpolation method [29]. The global approximation, the surrogate model, is given by the patched local approximations (2.6).

Our method distinguishes itself from other methods, such as [18, 21], by automatically detecting if a QoI is smooth or discontinuous. By utilizing weight function (3.4), which is proposed later, the placed samples can be used for proper discontinuity detection, while focusing on regions with a high PDF. As a result, the placed samples are not only suited for finding the discontinuity, but also for constructing an accurate surrogate in the regions with high PDF. The combination of these sample locations and an SVM leads to accurate discontinuity detection, without placing samples in the random space, having relatively small contributions to the statistical moments of the QoI.

I. Sampling algorithm. The main idea behind our adaptive sampling algorithm is that we want to refine our surrogate model based on the QoI behavior and the associated joint PDF of the random input variables. We achieve this by creating a graph that links the samples, and by assigning weights to edges of this graph, constructing an MST, and then adding samples on the most important edges of this MST.

The methodology is explained for a 2D QoI $u(z_1, z_2)$, but can be generalized easily to a higher-dimensional QoI.

Initial sample placement. The sampling procedure is started by placing initial sample points. Straightforward choices of the initial sample locations are shown in Figure 2. These initial sample configurations are extensible to high-dimensional random spaces. Initial sample placement in both Figures 2(a) and 2(b) introduces an anisotropy in the placement of subsequent samples. The initial sample placement in Figure 2(c) is a good trade-off between the number of initial samples and the isotropy of subsequent samples. In this paper, the initial sample grid shown in Figure 2(c) is employed, unless stated otherwise.

Graph construction with neighboring samples. The existing samples are connected based on Voronoi diagram construction [30]. A Voronoi diagram is a partitioning of a d-dimensional space into regions based on the distance to a specific set of samples [31]. Each Voronoi cell contains one sample, and the cell corresponds to all the points that are closer to this sample than to any other sample. The Voronoi diagram for the initial sample configuration is shown in Figure 3 (left).

The boundary of each Voronoi cell contains several vertices. At each of these vertices the neighboring Voronoi cells are determined. The midpoints of these neighboring cells are then connected to obtain a local graph; see Figure 3 (middle). This

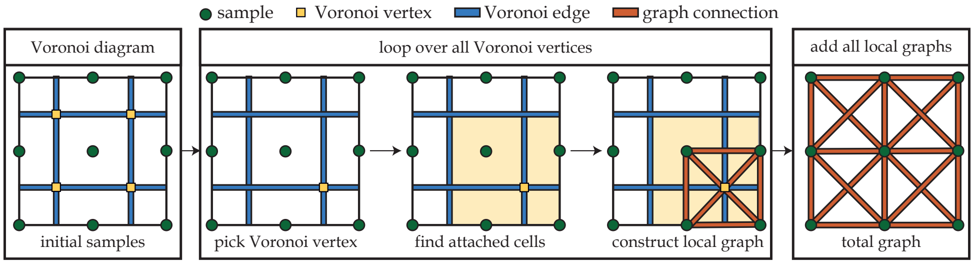

Fig. 3. *Algorithm for constructing the total graph of a set of sample points. (left) The Voronoi diagram with the corresponding Voronoi vertices. (middle) The local graphs connect the sample points of which the Voronoi cells have a common Voronoi vertex. (right) The total graph combines all the local graphs.*

local graph is similar to a Delaunay triangulation, but gives more isotropic behavior on regular grids. The local graphs are subsequently connected together to obtain a global graph; see Figure 3 (right).

Assignment of weights to edges. Not all the edges in the total graph are equally attractive for placing new samples. Edges that are long, and have large variation between QoI values and/or are in a region associated with a high PDF value, are good candidates for refinement. A weight function $w(\mathbf{z}_1, \mathbf{z}_2)$ assigns a weight to the edge between the two graph vertices $\mathbf{z}_1$ and $\mathbf{z}_2$. A low weight means that the edge is a good candidate for refinement and vice versa. Weighting based solely on either gradient [21] or PDF [32, 33] is the most straightforward:

$$w_{\mathrm{PDF}}(\mathbf{z}_1, \mathbf{z}_2) = \left(\rho \left(\frac{\mathbf{z}_1 + \mathbf{z}_2}{2} \right) \|\mathbf{z}_1 - \mathbf{z}_2\|_2 \right)^{-1}, \tag{3.1}$$

$$w_{\mathrm{grad}}(\mathbf{z}_1, \mathbf{z}_2) = \left(\frac{|u(\mathbf{z}_1) - u(\mathbf{z}_2)|}{\|\mathbf{z}_1 - \mathbf{z}_2\|_2} \|\mathbf{z}_1 - \mathbf{z}_2\|_2 \right)^{-1}, \tag{3.2}$$

where ρ is the PDF and $\|\cdot\|_2$ the Euclidean distance. These weight functions are not always efficient, as they place most samples in either the smooth or the discontinuous regions. A PDF-weighed gradient across an edge is more meaningful, as it incorporates both the rate of change of response values and the probability of occurrence. Therefore, when compared to (3.1)–(3.2), a more meaningful measure would be:

$$\left(\int_e |\mathbf{e} \cdot \nabla u(\mathbf{z})| \rho(\mathbf{z}) d\mathbf{z} \right)^{-1}, \tag{3.3}$$

where $\mathbf{e}$ is a vector representing an edge. Equation (3.3) can be seen as the reciprocal of the expectation of the gradient across an edge. The expectation of the gradient is an important quantity in stochastic optimization routines such as stochastic gradient descent [34], where it justifies the use of a subset of samples for determining the gradient, which is used for minimizing the objective function. In our case the gradient is not used for finding a descent direction, but it indicates the amount of local variation in the QoI u. By weighting the gradient with the PDF and integrating it over an edge, we get the expected variation across an edge for the given PDF, which indicates if refinement is necessary. The integral in (3.3) cannot be computed directly, but can be approximated as

$$w_{\mathrm{PDF+grad}}(\mathbf{z}_1, \mathbf{z}_2) = \left(\left(\rho \left(\frac{\mathbf{z}_1 + \mathbf{z}_2}{2} \right) \frac{|u(\mathbf{z}_1) - u(\mathbf{z}_2)|}{\|\mathbf{z}_1 - \mathbf{z}_2\|_2} \right) \|\mathbf{z}_1 - \mathbf{z}_2\|_2 \right)^{-1} . \tag{3.4}$$

To clarify, we multiply each weight function with the distance of the edge $\|\mathbf{z}_1 - \mathbf{z}_2\|_2$ to account for regions in the random space that have a low number of samples. After weighting all the edges, we normalize by dividing by the maximum weight. The different weight functions are compared in section 4.

MST for refinement of sample grid. New samples are placed at the middles of the edges that have a sufficiently low weight. However, if all edges with sufficiently low weight are refined, undesirable clustering of samples may occur at early stages of the sampling procedure.

To prevent this, an MST is used to obtain a subset of edges, such that this subset reaches all the samples with a minimal total edge weight. The most important edges are contained in this MST, while still exploring a significant portion of the random space. The MST prevents the undesirable sample clustering at early stages. An edge in the MST is refined, if its weight is sufficiently low compared to the minimum weight among all edges, $w_{\min}$:

$$w_i \leq c\, w_{\min} \ , \tag{3.5}$$

where $c > 1$. The value of c determines how many samples are added each iteration, i.e., low values of c result in a low number of samples added and vice versa. Low values of c produce the most accurate results, but many iterations are needed to reach a specified total number of samples. Iterations can become prohibitively expensive in high-dimensional random spaces. Therefore, in this paper we set $c = 2$, which is a trade-off between the number of samples added and the number of iterations to be performed. The edge with minimum weight $w_{\min}$ is not necessarily included in the MST, and will be added if it was not already included, to prevent the sampling algorithm from not adding any samples.

Complete sampling algorithm. The complete sampling strategy (I) is an iterative procedure, which is illustrated in Figure 4. The procedure starts with choosing the initial sample points. Next we loop over the three steps: *graph construction*, *edge weighting*, and *MST edge refinement*. The loop is terminated when the specified number of iterations $i_{\max}$ has been performed or when the total number of sample points exceeds a specified threshold $N > N_{\max}$.

II. Domain decomposition. The idea of the domain decomposition step in our method is to divide the random space into nonintersecting elements E_i, such that the sampled QoI values from section I exhibit smooth behavior locally in each element. The elements are constructed by first classifying the QoI values according to the QoI gradients. Second, the sample classes are separated by means of a classification boundary, based on SVMs. This classification boundary cuts the random space into several elements.

Sample classification based on QoI gradients. An SVM determines a classification boundary based on a set of classified samples. Since a classification boundary is an approximation to a discontinuity in the QoI, the classification is based on the difference in values between two neighboring samples. To clarify, samples are put into different groups if the jump in values across an edge exceeds a threshold. The choice for the threshold in the procedure is crucial. We have observed that the polynomial

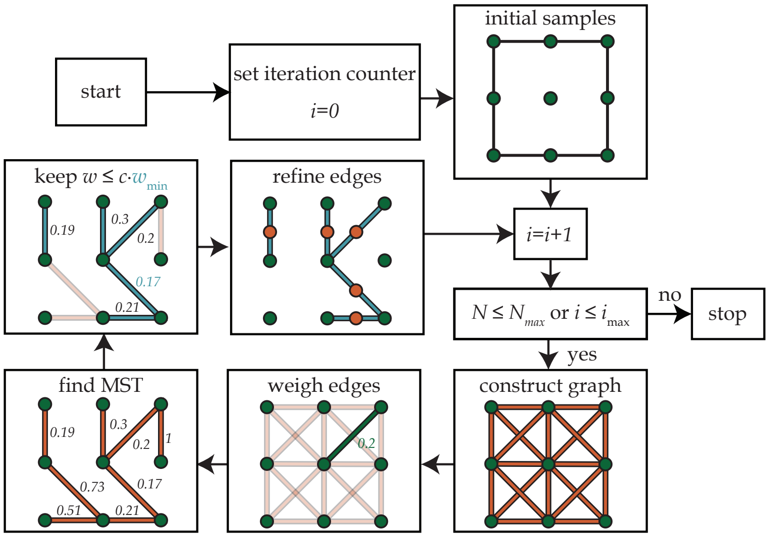

FIG. 4. *Schematic representation of the adaptive sampling strategy.*

annihilation procedure from [22] works very well and a similar methodology is therefore employed in this paper. Polynomial annihilation, using all samples close to a straight line crossing an edge, is used to estimate a jump value across an edge and labels two points as the same class if the difference in function values is less than the jump value. An in-depth discussion of polynomial annihilation is discussed in [35]. A schematic representation of the classification procedure is shown in Figure 5.

The classification procedure shown in Figure 5 is able to detect multiple discontinuities and able to divide the samples into multiple classes by using only the sample locations and corresponding function values. There is no need for specifying the number of classes beforehand, as the procedure detects the total number of classes automatically. Even though the classification procedure works in many cases, it is not able to consistently detect discontinuities which do not divide the random space into multiple subdomains. For this reason, we do not consider such discontinuities in the remainder of this manuscript.

Classification boundary from SVMs. The classes assigned to the different samples in the random space will be used as training data for a supervised machine learning algorithm. SVM is a supervised machine learning technique, used for building a classification boundary between samples that belong to different classes [22, 28]. SVM is used in this paper because it is defined by a convex optimization problem, for which efficient methods are available [36], which makes it viable for high-dimensional problems.

Assume N adaptive samples $Z = \{\mathbf{z}_i\}_{i=1}^{N}$ are classified into N_c different classes c_i, where $c_i \in \{1, \dots, N_c\}$ is the class belonging to sample $\mathbf{z}_i$. The idea behind an SVM is to construct a classifier S_λ of the form

$$S_\lambda(\mathbf{z}) = \sum_{i \in SV} \alpha_i K(\mathbf{z}, \mathbf{z}_i) \ , \tag{3.6}$$

where α_i is the coefficient associated with the sample point $\mathbf{z}_i$, λ a regularization parameter, $SV \subset Z$ the set of support vectors, and K a kernel [27]. If $\alpha_i > 0$, then

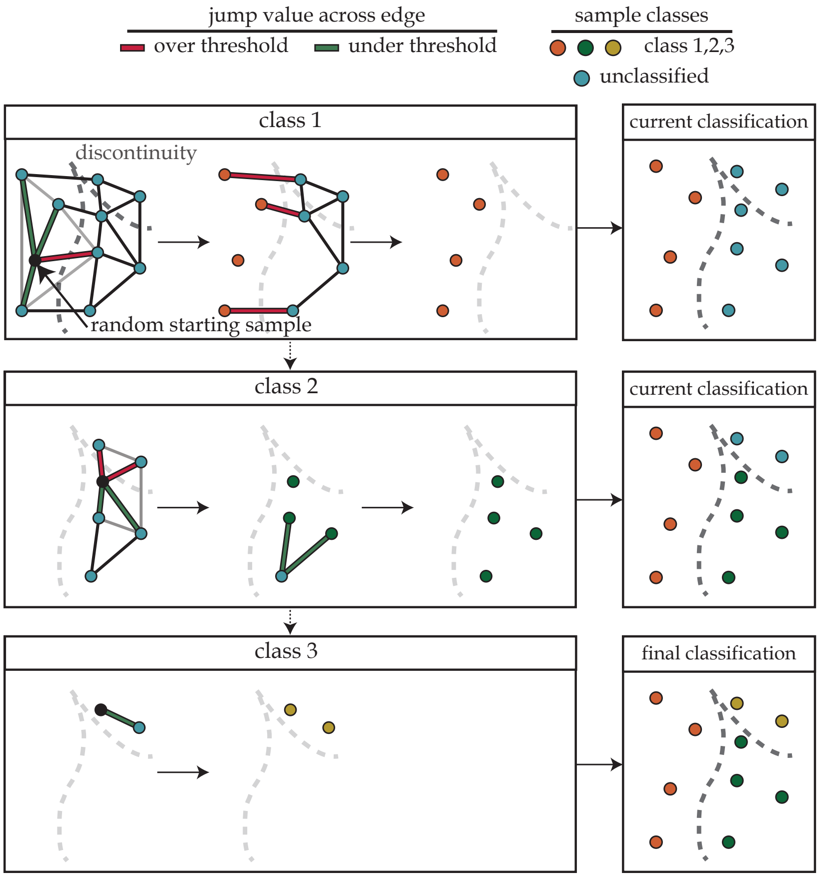

FIG. 5. *Schematic representation of the classification procedure. The sample connectivity is given by the Voronoi construction performed on the final sampling grid from the adaptive sampling algorithm.*

$\mathbf{z}_i$ is a support vector. Depending on the application, different kernels are available [28]:

$$K(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x}, \mathbf{y} \rangle \quad \text{(linear)}\ , \tag{3.7}$$

$$K(\mathbf{x}, \mathbf{y}) = (\gamma\langle \mathbf{x}, \mathbf{y} \rangle + c_t)^r \quad \text{(polynomial)}\ , \tag{3.8}$$

$$K(\mathbf{x}, \mathbf{y}) = \exp(-\gamma\|\mathbf{x} - \mathbf{y}\|_2^2) \quad \text{(radial basis function)}\ , \tag{3.9}$$

$$K(\mathbf{x}, \mathbf{y}) = \tanh(\gamma\langle \mathbf{x}, \mathbf{y} \rangle + c_t) \quad \text{(sigmoid)}\ , \tag{3.10}$$

where $\langle \cdot, \cdot \rangle$ is the standard inner product in $\mathbb{R}^d$, γ a regularization constant, c_t a translation constant, and r the polynomial degree. Choosing the proper kernel depends on the regularity of the classification boundary which is often not known beforehand. Radial basis function kernels (3.9) are a common choice and are also employed in this paper. The constant γ is normally advised to be chosen as $1/N_c$ [36], but we will also investigate other choices in the result section of this paper. High values for γ result in a classification boundary with a fine resolution, but this may result in overfitting. Low values for γ result in a coarser estimation of the location of the classification boundary. The optimal value for γ differs for different functions. In section 4 an

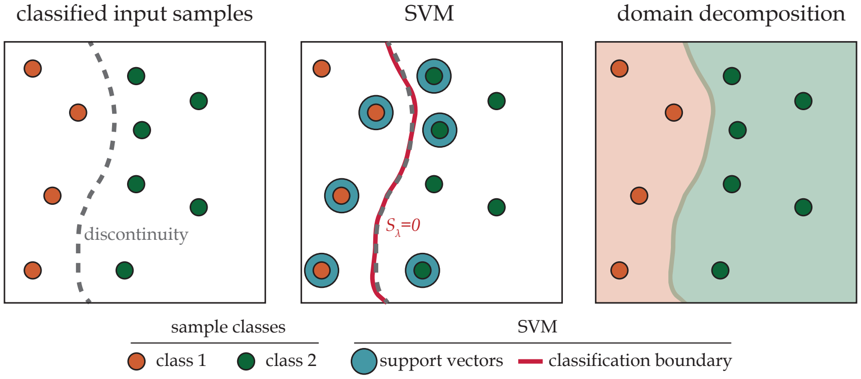

FIG. 6. *An example of an SVM domain decomposition for two different classes.*

optimal value for γ is found for a specific family of functions, which is important in our test cases. Once K has been chosen, the classifier S_λ is obtained by solving the following least-squares problem:

$$S_\lambda(\mathbf{z}) = \arg \min_{S \in L_2(I_\mathbf{z})} \left\{ \frac{1}{N} \sum_{i=1}^{N} \max(0, 1 - c_i S(\mathbf{z}_i)) + \lambda \|S\|_{L_2(I_\mathbf{z})} \right\} . \tag{3.11}$$

The classification boundary is given by the 0-contour of S_λ. It separates the different classes from each other with a hypersurface and is obtained with the LIBSVM library [36]. The classification boundary decomposes the domain into several elements. Figure 6 shows an example of a classification boundary for two different classes. SVM can deal with multiple classes and, hence, multiple discontinuities, quite easily, which makes it a suitable discontinuity finder for a wide range of QoIs. However, SVM in combination with the classification procedure is not yet able to properly detect discontinuities that have at least one endpoint which is not intersecting with itself or with the boundaries of the random space.

Gorodetsky and Marzouk [22] show how SVMs can be used to efficiently localize discontinuities in a high-dimensional space. Their approach differs from our approach in the strategy for placing samples which are used by the SVM. They propose sample placement which is solely suited for detecting discontinuities and samples are placed without taking the PDF into account. As opposed to this, our approach incorporates the PDF and makes our sample placement more suitable for surrogate construction and statistical moment calculations, while being slightly less efficient in localizing discontinuities.

III. Local approximations. The elements E_i from the domain decomposition (II) are arbitrarily shaped and the samples $\mathbf{z}_i$ are distributed in such a way that interpolation is not a trivial task. Least orthogonal interpolation is able to perform interpolation on sample distributions on such arbitrarily shaped domains. The sampling strategy (I) does not necessarily choose points that are optimal for interpolation. Therefore, attempting to construct an interpolant on this set of interpolation nodes is not always a good idea and may produce unstable interpolants. We therefore use an extended version of the original least orthogonal interpolation, which selects a subset of samples that is better suited for stable interpolation [21]. This enables the MST-ME method to place sample points in the random space, where we want to further resolve the QoI, without focusing on the stability of the interpolation. In practice, the least orthogonal interpolation can become unstable when an element has a highly irregular shape and in such cases we propose that a different interpolation procedure should be used [37, 38]. In this work we use the least orthogonal interpolation in the remainder of this manuscript because of its fast convergence and flexibility for scattered data sets, and we do not consider such highly irregular shapes.

Least orthogonal interpolation. We briefly introduce the least orthogonal interpolation procedure, which is discussed in more detail in [29]. Assume we have sampled our model response at the unstructured locations $\{\mathbf{z}_i\}_{i=1}^N$ in one element $E \in \mathbb{R}^d$ of the SVM domain decomposition. Assume a pdf $\rho(\mathbf{z})$ on E, with corresponding orthonormal polynomials $\{\phi_i(\mathbf{z})\}_{i=1}^\infty$. The orthonormal polynomials are ordered with natural numbers according to the graded reverse lexicographic ordering. We order the orthonormal polynomials by using a multi-index $\mathbf{i} \in \mathbb{N}^d$. Now let Π be the space of all d-variate polynomials and let Π_k be the space of all d-variate polynomials of degree less than or equal to k. For any $u \in \Pi$, define the projection P_k onto Π_k as

$$u = \sum_{\mathbf{i}} \hat{u}_{\mathbf{i}} \phi_{\mathbf{i}} \Leftrightarrow P_k u = \sum_{\|\mathbf{i}\| \leq k} \hat{u}_{\mathbf{i}} \phi_{\mathbf{i}} \ , \tag{3.12}$$

where the expansion coefficients $\hat{u}_{\mathbf{i}}$ are given by

$$\hat{u}_{\mathbf{i}} = \langle u, \phi_{\mathbf{i}} \rangle_\rho = \int_E u(\mathbf{z}) \phi_{\mathbf{i}}(\mathbf{z}) \rho(\mathbf{z}) d\mathbf{z} \ . \tag{3.13}$$

The goal of least orthogonal interpolation is now, *given the nodes* $Z = \{\mathbf{z}_i\}_{i=1}^N$, *define a polynomial interpolation space* $\Pi_{Z,\rho}$ *in which we can uniquely perform interpolation on the sample responses* $\{u(\mathbf{z}_i)\}_{i=1}^N$.

To accomplish this, we use the sample locations to define a new set of functions

$$h_n(\cdot) = \sum_{\mathbf{i}} \phi_{\mathbf{i}}(\mathbf{z}_j) \phi_{\mathbf{i}}(\cdot) \ , \quad j = 1, \ldots, N \ . \tag{3.14}$$

An operator is introduced that maps the functions h_n to polynomials:

$$h_{\downarrow,\rho} = P_m h \ , \quad m = \min \{ \in \mathbb{N}_0 \ : \ P_k h \neq 0 \} \ , \tag{3.15}$$

where we introduce a subscript ρ to emphasize the fact that the operator P depends on the choice of ρ. We extend the operator $(\cdot)_{\downarrow,\rho}$ to vector spaces H by $H_{\downarrow,\rho} = \text{span}\{h_{\downarrow,\rho} \ : \ h \in H\}$. If $H = \text{span}\{h_i\}_{i=1}^N$ one can show that the space of polynomials

$$\Pi_{Z,\rho} = H_{\downarrow,\rho} \tag{3.16}$$

is an N-dimensional space of d-variate polynomials that is isomorphic to interpolation data at the sample locations Z. Therefore, a polynomial $g \in \Pi_{Z,\rho}$, satisfying $g(\mathbf{z}_i) = u(\mathbf{z}_i)$, exists and is unique. The polynomial g is called the least orthogonal interpolant. The construction of both the interpolation space and the interpolation coefficients is accomplished by standard linear algebra operations on the Vandermonde matrix. Notice that when using Gauss quadrature sample locations associated with the specified PDF as input for least orthogonal interpolation, the classic orthogonal basis polynomials result, and the procedure boils down to common nonintrusive SC. For an in-depth discussion of least orthogonal interpolation, see [29].

Surrogate construction. We denote the least orthogonal interpolation operator by $I[\cdot]$, which operates on a subset of $(\mathbf{z}_i, u(\mathbf{z}_i))_{i=1}^n$, and we assume that the random

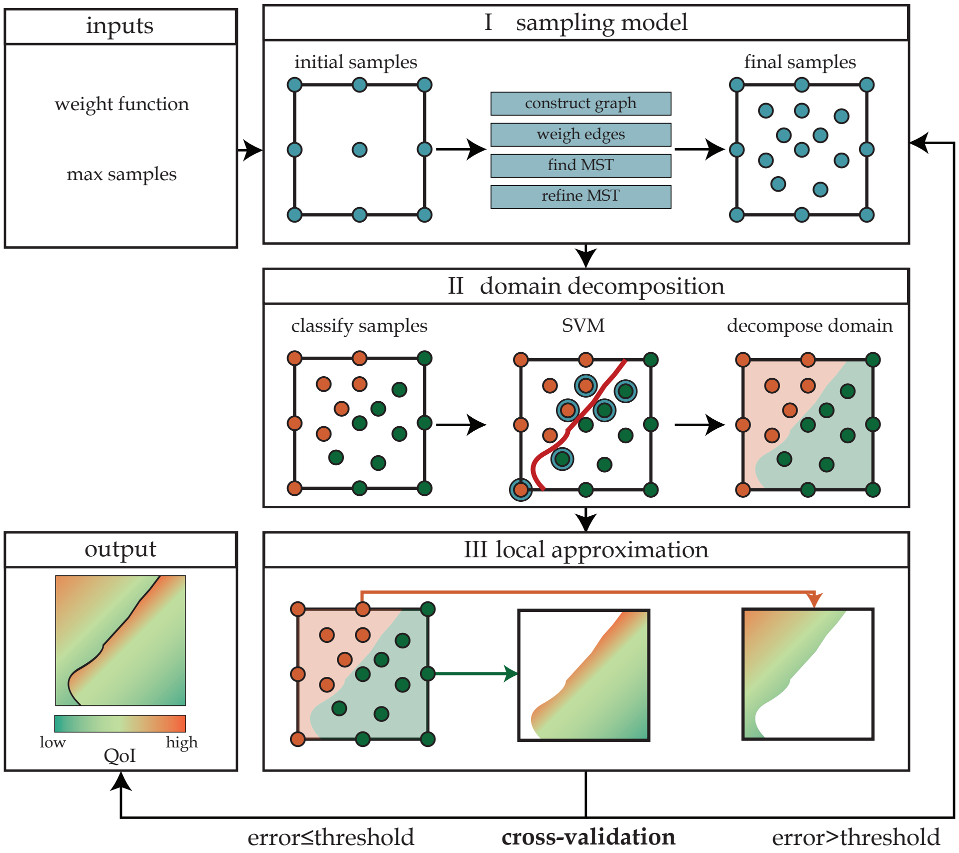

FIG. 7. *A schematic overview of the complete MST-ME method in a 2D random space.*

space is decomposed into N_c elements E_i. Each element E_i comprises a single class c_i, which consists of the samples $(\mathbf{Z}_i, \mathbf{U}_i)$. The global approximation is given by (see (2.6))

$$\tilde{u}(\mathbf{z}) = \sum_{i=1}^{N_c} I[(\mathbf{Z}_i, \mathbf{U}_i)]\mathcal{I}_{E_i}(\mathbf{z}) \ , \tag{3.17}$$

where $\mathcal{I}_{E_i}(\mathbf{z})$ is the indicator function satisfying $\mathcal{I}_{E_i}(\mathbf{z}) = 1$ if $\mathbf{z} \in E_i$ and 0 otherwise.

IV. Complete algorithm. The complete MST-ME method is shown in Figure 7. Apart from the weight function and sampling threshold, no additional input from the user is required, which makes the method suitable for generic problems.

A suitable stopping criterion is given by checking if cross validation errors in the surrogate drop below a specified threshold. Computing the cross-validation errors is straightforward and is based on the leave-one-out principle. The errors in the surrogate given by cross validation are unbiased [39, 40, 41] and give a good indication on how well the algorithm performs. In fact, as we use the adaptive least orthogonal interpolation, some samples are not used in the interpolation procedure and are left out. These samples can be used to compute the validation error without the need to leave out relevant samples.

Global approximation error. The adaptive nature and incorporation of SVMs in our approach makes it difficult to derive error bounds. However, in this subsection we discuss how the error is composed and how it can be monitored. The global approximation error can be decomposed into two parts.

The first part comprises the error made by the SVM, i.e., the error in the approximation for the discontinuity location. Strict error bounds are hard to derive for SVMs in general. However, Vapnik and Chapelle showed that the cross-validation error of the SVM gives an unbiased estimate of the error in the domain decomposition step [39], and is therefore a proper way to monitor the error of the SVM approximation.

The second part comprises the error made by interpolating the samples. Least orthogonal interpolation is used, which is able to interpolate scattered data sets adaptively. A quality measure of our sample set can be given by the Lebesgue constant, which is a measure of how well an interpolant does in comparison with the best polynomial approximation. In the least orthogonal interpolation method, the Lebesgue constant in one dimension is defined as [29]

$$\Lambda_N = \sup_{\mathbf{z}\in I_{\mathbf{z}}} \lambda(\mathbf{z}) \ , \tag{3.18}$$

where

$$\lambda(\mathbf{z}) = \sum_{i=1}^{N} |l_i(\mathbf{z})| \ , \tag{3.19}$$

where N is the number of samples and $l_i(\mathbf{z})$ is the Lagrange interpolating polynomial corresponding to node i. The Lebesgue constant is related to the error in the interpolant as follows:

$$\|u - \tilde{u}\| \leq (1 + \Lambda_N)\|u - u^*\| \ , \tag{3.20}$$

where u, $\tilde{u}$, and u^* are the exact function, the surrogate, and the best approximating polynomial amongst the set of polynomials of degree $N-1$, respectively. The adaptively chosen sample sets that are used for interpolation on the elements can be used to compute the Lebesgue constant, which gives an indication of the quality of the nodal locations. However, for the one-dimensional (1D) test cases in section 4, the Lebesgue constant gave no additional insight and is therefore not discussed in the remainder of this manuscript.

Computational cost for N samples in d-dimensional random space. The computational cost of the sampling strategy (I) is determined by the cost of computing the Voronoi diagrams and finding the MST. Computing a Voronoi diagram on N samples in $\mathbb{R}^d$ can be done in $\mathcal{O}(N\log(N) + N^{\lceil d/2\rceil})$ time [42]. In the worst-case scenario, only a single sample is added in each iteration and, hence, we have to compute N Voronoi diagrams on N samples. The maximum computational cost is therefore $\mathcal{O}(N^2\log(N) + N^{\lceil d/2\rceil+1})$. Notice that the Voronoi diagrams are not suited for high-dimensional spaces because of the factor $d/2$ in the exponent. Instead, when Voronoi diagrams are too expensive to compute, a simple radius search algorithm can be used. Computing the MST can be done with Prim's algorithm [43], which has an algorithmic complexity of $\mathcal{O}(|E|\log(|V|))$, where $|E|$ is the total number of edges and $|V|$ the total number of samples. An upper bound for $|E|$ and $|V|$ is given by $N(N+1)/2$ and N, respectively. Again, in the worst-case scenario we have to perform N iterations, which can be done in $\mathcal{O}(N^3\log(N))$ time.

The computational cost of the domain decomposition (II) is based on the complexity of the classification procedure and the SVM. Classification of N samples can be performed in $\mathcal{O}(N^2)$ time. The SVM has a complexity ranging between $\mathcal{O}(N^2)$ and $\mathcal{O}(N^3)$, depending on the number of classes and the kernel [36]. Hence, domain decomposition can be performed in $\mathcal{O}(N^3)$ time and is independent of d.

The local approximation (III) uses LU- and QR-decomposition for determining the interpolant. Computing the QR-decomposition is the dominant factor in (III), it can be done in $\mathcal{O}(N^3)$ time, using the standard implementation in MATLAB. If least orthogonal interpolation is employed in an adaptive fashion, the complexity increases to $\mathcal{O}(N^4)$, as N QR-decompositions are computed in the worst case.

The complexity of the complete algorithm is determined by the complexities of (I), (II), and (III). This results in an overall complexity of $\mathcal{O}(N^4)$ if $d \leq 5$ and $\mathcal{O}(N^3 \log(N) + N^{\lceil d/2 \rceil + 1})$ otherwise.

4. Results.

In this section we present multiple examples that illustrate the robustness and flexibility of the MST-ME method. For computing the error between the exact surrogate and approximation we use the following weighted $L_{2,\rho}$-norm:

$$\|\tilde{u} - u\|_{2,\rho}^2 = \frac{1}{N_{MC}} \sum_{i=1}^{N_{MC}} \rho(\mathbf{z}_i^{MC}) \cdot |\tilde{u}(\mathbf{z}_i^{MC}) - u(\mathbf{z}_i^{MC})|^2 \ , \tag{4.1}$$

where the surrogate model $\tilde{u}$ is constructed using MST-ME and evaluated at Monte Carlo sample locations $\mathbf{z}_i^{MC}$ drawn from the PDF ρ. The exact solution u is the evaluation of the model sampled at the same Monte Carlo samples. To clarify, the error compares the constructed surrogate with the exact solution at the Monte Carlo sample locations, and this error should converge to 0 if the constructed surrogate converges to the exact response. We multiply the difference between the surrogate and exact solution with the PDF ρ to emphasize on regions in the random space that are likely to occur. Samples that are within a distance from the discontinuity, which is lower than the minimum distance of the adaptive MST-ME samples, are discarded. Discarding samples is motivated by the fact that these lie below the resolution of the SVM discontinuity detection, where the fidelity of the classification is questionable [21].

The first example shows the approximation of three 2D, piecewise constant functions. This example focuses solely on the domain classification and shows convergence of the SVM domain decomposition (II). The second example shows the approximation of 1D and multidimensional Genz functions [44]. The third test case shows the MST-ME method applied to a more complicated model, which is defined by an underlying PDE, namely, the shallow water equations. Last, we apply the MST-ME method to a 3D dam break problem simulated through an incompressible smoothed particle hydrodynamics (SPH) model, to indicate that our methodology can be applied to complex engineering problems.

4.1. Domain classification.

The accuracy of the SVM domain decomposition (II) is investigated as a function of the parameter γ.

SVM domain decomposition works best for discontinuities without corners. The SVM domain decomposition (II) is tested by approximating three piecewise constant 2D functions. The functions have a discontinuity in the shape of a circle, a rectangle, and a triangle, respectively. Notice that all discontinuities divide the domain into two subdomains and, therefore, are detectable by the classification procedure in II. The sampling algorithm (I), with a weight function that focuses solely on the gradient (3.2), is used to determine the sample locations. The SVM domain decomposition uses the radial basis function kernel (3.9) in which γ is set to the advised value $1/N_c$

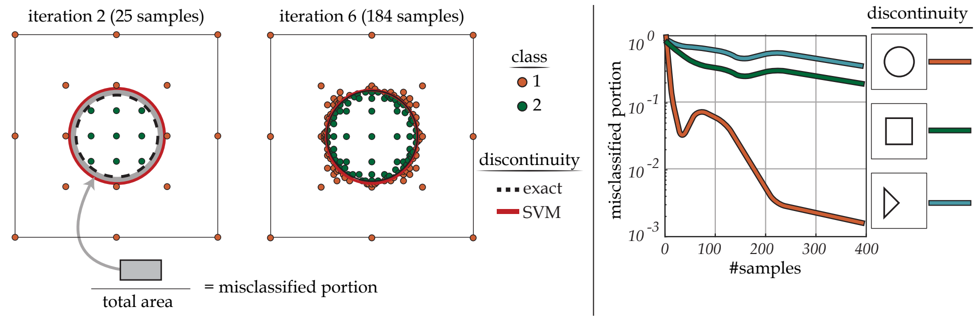

FIG. 8. *SVM domain decomposition results. (left) Sample locations at different iterations for the circular test case with $\gamma = \frac{1}{2}$. (right) Convergence of the misclassified portion for increasing number of samples.*

[36], where N_c is the number of classes. The sample locations for the circle test case are shown in Figure 8 (left), along with the definition of the error measure (the misclassified portion). Clustering of samples appears around the circularly shaped discontinuity location, because the gradient based weight function (3.2) does not lead to refinement if there is no intersection with the discontinuity. The sample locations show symmetry, but after 5 iterations, the symmetry is slightly lost, although this is not noticeable in Figure 8 (left). The domain decomposition error as a function of the number of samples is shown in Figure 8 (right).

The misclassified portion of the domain decreases rapidly with the increasing number of samples, but is in general not monotonically decreasing. As stated in section II, discontinuity lines with sharp corners, in particular, the square and triangle, are hard to approximate for the SVM with a radial basis function kernel. However, the regularity of the classification boundary is often not known beforehand and therefore we opt for the commonly used radial basis function kernel. The value for γ however may significantly influence the accuracy of the discontinuity approximation, and the advised value $1/N_c$ is in general not optimal. Therefore we search for a value of γ that is optimal for the remaining test cases.

The optimal value for γ for our test cases is $3/N_c$. We now investigate the optimal value for γ by testing several candidate values for a large set of piecewise constant functions. To clarify, the optimal shape parameter γ for the radial basis function kernel (3.9) is sought for a specific family of discontinuous functions, such that the classification procedure works best. The fact that this value may be optimal for the radial basis function kernel, does not necessarily imply that it is optimal for other kernels as well. Kernel (3.9) is used in this paper for classification; we focus on the optimal shape parameter for this specific kernel. The parameter γ may influence the accuracy of the SVM domain classification. Many discontinuous QoIs in engineering applications possess a discontinuity without corners, i.e., the exact discontinuity surface is a smooth hypersurface. For such discontinuities, the value of $1/N_c$ might not be optimal and therefore we search for an optimal value among a set of candidates that are obtained by adding perturbations to the advised value: $\{\frac{1}{N_c+4}, \frac{1}{N_c+3}, \frac{1}{N_c+2}, \frac{1}{N_c+1}, \frac{1}{N_c}, \frac{2}{N_c}, \frac{3}{N_c}, \frac{4}{N_c}, \frac{5}{N_c}\}$. A set of 10^6 piecewise constant functions, possessing up to 3 discontinuities, on the domain $[-1, 1]^2$, is randomly generated. The discontinuity location is given by up to 3 nonintersecting polynomial parametric curves up to degree 5, which have random coefficients. For each of these

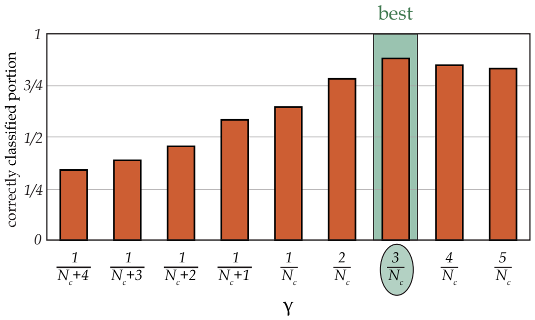

FIG. 9. *Average correctly classified portion for different γ-values.*

functions, an SVM domain decomposition is performed for each of the possible values of γ. This domain decomposition is performed on 50 adaptively sampled points from the MST-ME method, based on the weight function (3.2) and the initial configuration shown in Figure 2(c). We add all correctly classified portions for each γ-value, and divide the resulting sum value by 10^6 to obtain an average correctly classified portion for all the randomly generated functions. The number of adaptive samples influences the average correctly classified portion. A similar trend between the γ-values is obtained. The results are shown in Figure 9. Figure 9 shows that $\gamma = 3/N_c$ is the most accurate choice for the generated family of piecewise constant functions. The inconsistency with the value for γ suggested in literature ($1/N_c$) and our value is possibly due to the fact that we limit ourselves to a family of discontinuous functions, which possess no sharp corners in the discontinuity surface. Hence, the value $3/N_c$ might not be the optimal value for other families of discontinuous functions. The discontinuities considered in the remainder of this paper have no sharp corners in the discontinuity surface and $\gamma = 3/N_c$ is therefore used there as well.

4.2. Genz functions approximation. To illustrate the accuracy/efficiency, the proposed MST-ME method is applied to a standard benchmark problem, namely, approximation of 1D Genz functions [44].

Edge weighting based on PDF and gradient is most robust. We consider the following Genz functions:

$$g_1(x,\alpha) = \cos(\alpha x)\ , \tag{4.2}$$

$$g_3(x,\alpha,\beta) = \left(\frac{1}{1+\alpha x}\right)^{1+\beta}\ , \tag{4.3}$$

$$g_5(x,\alpha,\beta) = \exp(-(\alpha|x|-\beta))\ , \tag{4.4}$$

$$g_6(x,\alpha,\beta) = \begin{cases} 0, & x > \beta\ , \\ \exp(\alpha x), & \text{otherwise}\ . \end{cases} \tag{4.5}$$

A uniform PDF is assumed on the interval $[-1, 1]$ and the initial grid consists of the two endpoints plus the middle point of this interval. As a reference, the MST-ME solution is compared with the SC solution on a Gauss–Legendre grid, which is essentially a least orthogonal interpolation using Gauss–Legendre sample locations (see section III). Figure 10 shows that weighting based on the PDF only results in a uniformly spaced sample grid. Interpolation on such a grid is in general not a good idea, as it may produce unstable interpolants. The least orthogonal interpolation

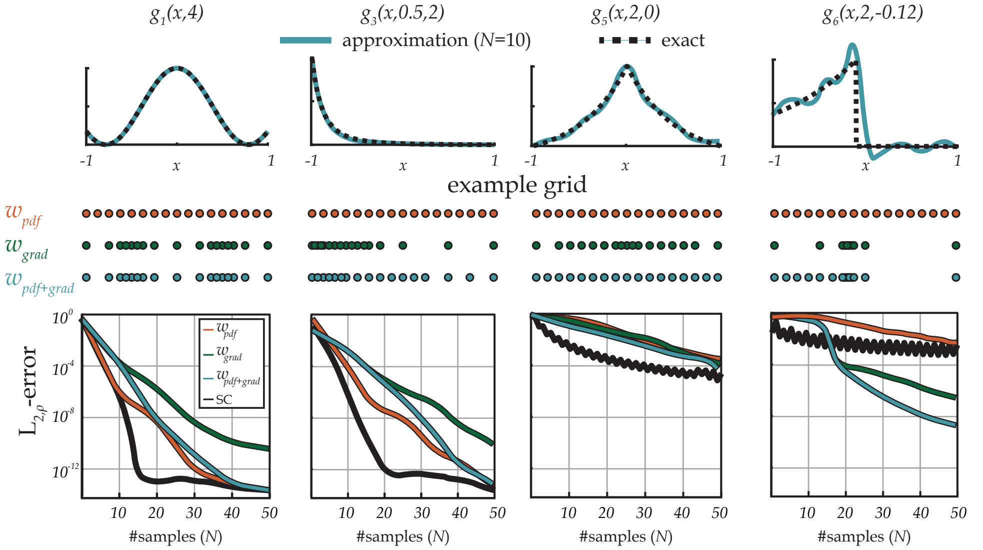

FIG. 10. *Error of the approximation with respect to the exact function.*

method partially circumvents this issue by choosing a subset of these samples in constructing an interpolant. Consequently, the smooth Genz functions g_1 and g_3 are well approximated, but g_5 and g_6 are not. Weighting based on the gradient alone leads to improved results for the discontinuous function g_6, but leads to less accurate results for the smooth functions g_1 and g_3. The standard SC method performs well in smooth cases, but converges slowly and also shows an oscillating error in some cases, due to the Gauss–Legendre grid that includes the middle point of the domain only for an odd number of samples. In contrast, the weight function based on both the PDF and the gradient performs the best overall, by keeping track of the discontinuity location, while still maintaining a sample distribution which resolves parts of the random space away from the discontinuity. In the remainder of this paper we therefore use weighting based on both PDF and gradient, (3.4).

Notice that all weight functions show slow convergence in the approximation of g_5. This is due to the absence of the second derivative in the weight function (3.4). By basing the classification on the second derivative in the QoI, we can circumvent the slow error convergence in the presence of a discontinuity in the first derivative. However, we will not have any discontinuities in the first derivatives for the remaining test cases in this paper. We therefore use classification based on the first derivative only in the remainder of this paper.

The underlying PDF changes the sample grid. The effect of the underlying PDF is now investigated. The two PDFs that we consider are a symmetric and an asymmetric β-distribution, with parameters $(10,10)$ and $(2,7)$, respectively. The support of both PDFs is scaled to $[-1,1]$ and we use the uniform distribution as a reference. The error convergence for g_1 and g_6 is shown in Figure 11. The error convergence is similar to the error convergence for the uniform distribution. Again the sample grid is not ideal for interpolation, but the adaptive least orthogonal interpolation circumvents this by using a subset of nodes.

MST-ME outperforms conventional Latin hypercube sampling (LHS) in multiple dimensions. To investigate how the MST-ME method performs in multiple dimen-

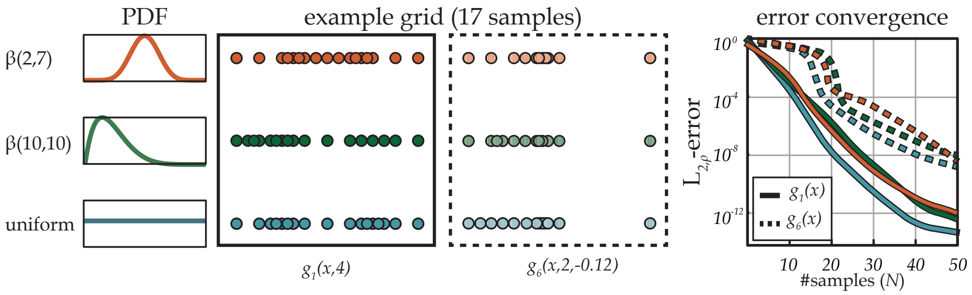

Fig. 11. *Error convergence for different choices of PDFs.*

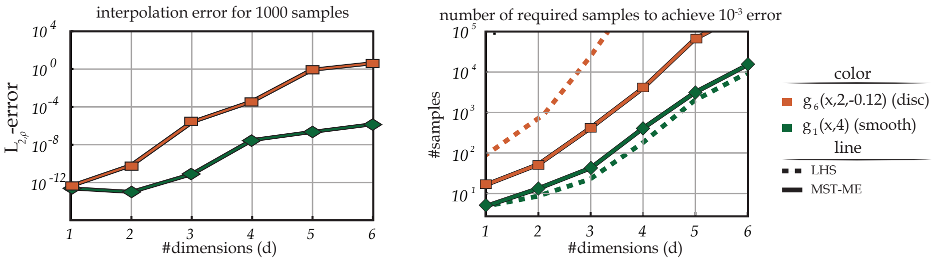

Fig. 12. *Comparison of the MST-ME method and LHS+least orthogonal interpolation with increasing dimensionality. (left) Error of the approximation with respect to the exact function as a function of the dimension. (right) The number of samples needed to attain a certain accuracy as a function of the dimension.*

sions, the error is plotted for the smooth Genz function g_1 and the discontinuous Genz function g_6, with increasing dimension d, in Figure 12 (left). These multidimensional Genz functions are tensor products of the 1D Genz functions. The error for each dimension is based on 1000 adaptively sampled points with an initial sampling configuration equal to the one shown in Figure 2(c). The number of required samples needed to attain a specific accuracy is also plotted as a function of the dimension of the random space (right). As SC scales poorly to multiple dimensions when using a tensor grid construction, we compare the MST-ME method with a method which scales better to higher dimensions [45, 46]. As a comparison, we use isotropic LHS in combination with the least orthogonal interpolation method, and the results are shown in Figure 12. We clearly see the exponential increase of the required number of samples for the MST-ME method, which is the well-known curse of dimensionality. The LHS approach scales similarly when compared to the MST-ME for the smooth case, but requires significantly more samples when compared to the MST-ME method.

4.3. Shallow water dam break. We study the performance of the MST-ME method applied to a system of 1D conservation laws. This system consists of the 1D shallow water equations (SWEs), which describe the inviscid flow of a layer of fluid with a free surface, under the action of gravity, with the thickness of the fluid layer small compared to the other length scales [47]:

$$\frac{\partial}{\partial t}\begin{pmatrix} h \\ hv \end{pmatrix} + \frac{\partial}{\partial x}\begin{pmatrix} hv \\ hv^2 + gh^2/2 \end{pmatrix} = \mathbf{0}\ , \tag{4.6}$$

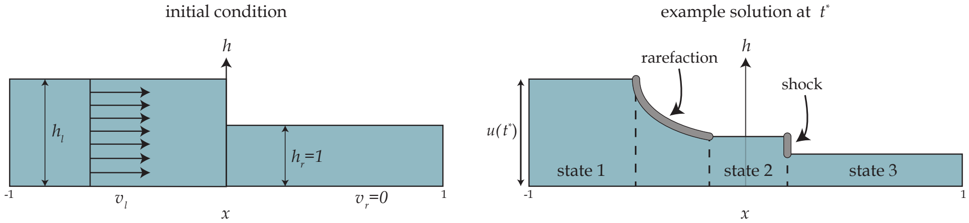

Fig. 13. *Schematic representation of the shallow water test case.*

where h is the free surface height (thickness of the fluid layer), v the velocity, and g the acceleration of gravity. The initial condition for the system of PDEs is given by

$$\begin{pmatrix} h \\ v \end{pmatrix}(x, t = 0) = \begin{cases} \begin{pmatrix} h_l \\ v_l \end{pmatrix}, & x \leq 0 \ , \\ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, & x > 0 \ , \end{cases} \tag{4.7}$$

leading to a Riemann problem shown in Figure 13. The solution of the Riemann problem for these initial conditions can be computed exactly when working on an infinite spatial domain [48]. The solution consists of two characteristic waves traveling through the spatial domain; see Figure 13. Each wave is a shock or rarefaction wave. When solid boundary conditions are imposed at $x = \pm 1$, an exact solution cannot be obtained for all initial solutions. We therefore employ a finite volume method with an exact Riemann solver [49] to compute the cell face fluxes, and solve the SWEs using 256 cells. A Crank–Nicolson scheme is used to integrate the SWEs in time and a ghost-cell method with reflective properties is used for the boundaries. The initial left state (h_l, v_l) is assumed to be uncertain and uniformly distributed $\mathcal{U}$ between $[1.5, 2.5]$ and $[-0.5, 0.5]$, respectively, i.e.,

$$\mathbf{z} = \begin{pmatrix} h_l \\ v_l \end{pmatrix} \sim \begin{pmatrix} \mathcal{U}(1.5, 2.5) \\ \mathcal{U}(-0.5, 0.5) \end{pmatrix} . \tag{4.8}$$

The uncertainty in the initial conditions is large in order to ensure that we get different characteristic behaviors of the QoI. The average thickness of the fluid layer is not shallow compared to the domain length, but this is not important for showing the performance of the MST-ME method. The QoI u is defined as the fluid height at $x = -1$ at a certain time t^*. A schematic representation of this setup is shown in Figure 13. Notice that the QoI is time dependent and that the characteristics of this QoI will change significantly as time progresses. Either a transition between a shock and rarefaction wave, or a difference in wave speeds, can result in a discontinuity in u. This allows us to study the robustness of the MST-ME method, as this test case comprises both smooth and discontinuous QoI responses. We emphasize that the constructed surrogate for the QoI at $t = t^*$ cannot be reused for other time instances, because the MST-ME method uses the QoI at the current time t^* as a measure to place new samples.

MST-ME automatically detects if a function is smooth or discontinuous. The MST-ME method is used for three different QoIs, $u = h(x = -1, t^* \in \{1.67, 4.16, 2.21\})$, which correspond to a mildly nonlinear, highly nonlinear, and close to discontinuous QoI, respectively. The surrogate model and sample grids after 10 iterations are shown

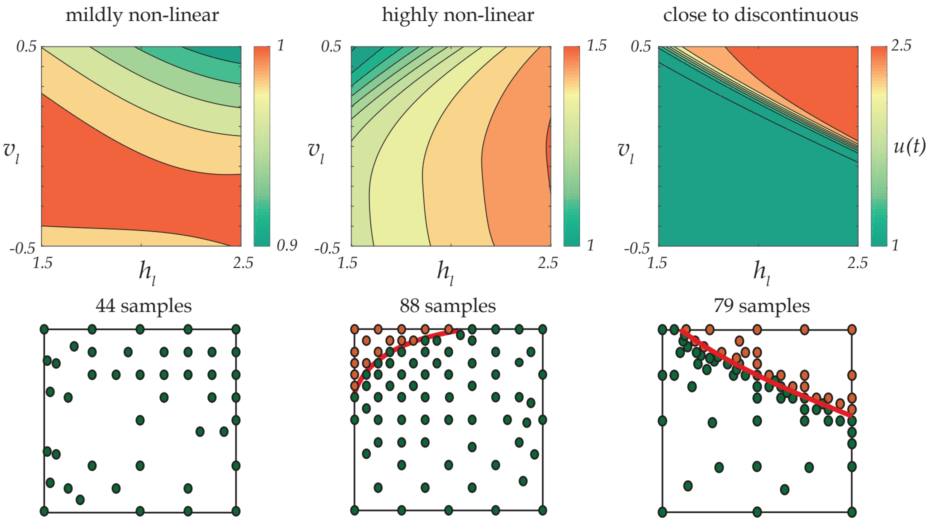

FIG. 14. *The three QoI surrogate models and corresponding sample grids after* 10 *iterations.*

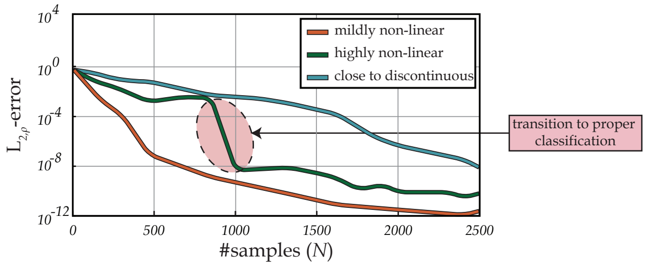

FIG. 15. *Convergence of the MST-ME solution in the* $L_{2,\rho}$*-error* (4.1)*.*

in Figure 14. The discontinuity in the QoI at time $t^* = 2.21$ is caused by a shock wave, which hits the left boundary for certain values in the random space, but does not yet hit the left boundary for other values in the random space.

To investigate the accuracy of the MST-ME method, we determine the convergence. The error is based on 10^6 Monte Carlo samples. The convergence of the $L_{2,\rho}$-error (4.1) is shown in Figure 15. The results show that the error as a function of the samples decays fast for the mildly and highly nonlinear cases, as expected. The highly nonlinear case shows a sudden drop in the error, which is caused by transition in the domain decomposition. First the classification procedure detects a large enough jump in the sampled QoI to conclude that there is a discontinuity present in the QoI. As the MST-ME progresses, samples are added in the area of the possible discontinuity, until the jumps in the QoI values become small enough to classify the samples properly. This transition to a correct classification explains the sudden drop in the error for the highly nonlinear case. MST-ME automatically detects the smoothness of the QoI, as the number of samples increases.

4.4. 3D dam break. As a last test case, the MST-ME method is applied to a complex engineering problem, namely, a 3D fluid dam break problem with parametric

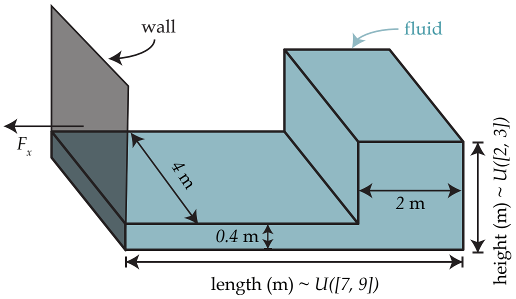

Fig. 16. *Schematic of initial condition for* 3*D dam break.*

initial conditions. This test case is similar to the shallow water dam break, with the difference that it describes fluid motion in 3D and contains more physics, i.e., viscous effects and no shallow water assumption. Dam break problems are commonly used as a benchmark in, for example, sloshing applications [9] and have been studied extensively [9, 10, 11]. MST-ME can be used to gain physical insight into this parametric problem, by constructing a surrogate model in the full parameter space, and this is our goal in this test case. We do not focus on convergence, as it is computationally infeasible to construct a reliable reference solution. A schematic representation of the test case is shown in Figure 16. We consider two uncertain parameters, namely, the length of the tank and the height of the right column of liquid. When the fluid is released, it starts to flow to the left side of the domain and impacts the wall, shown in Figure 16. The QoI is the maximum perpendicular wall force component F_x during first impact. SPH is used to simulate the free surface flow, induced by the initial condition in Figure 16. We use an open-source SPH solver, DualSPHysics [50]. The acceleration of gravity is set to 9.81m/s^2 and the kinematic viscosity ν is set to 10^{-6}m^2/s, which is the value for water at room temperature. Surface tension is neglected. Approximately 10^6 particles are used for the simulations, which is considered as medium to high resolution for free surface flows of these length scales. The simulations are performed on a single GPU unit with 2048 cores and the average simulation time is approximately 14 hours. A typical time-dependent result for a height of 3 m and a length of 7 m, along with the perpendicular wall force component F_x, is shown in Figure 17. The four consecutive instances of the example simulation show the following: initial liquid configuration; wave development due to pressure gradient; breaking wave impact on wall; liquid after impact. Depending on the height difference between both liquid columns, wave breaking may occur before the wave impacts the wall. In reality, a gas pocket may be entrapped during this process. The gas pocket may be compressed and next lead to an inside pressure buildup. However, the simulations performed here are single phase free surface flows and the gas phase is not taken into account, so the physics in an entrapped gas pocket is ignored. Obtaining a parametric solution is the target, where both parameters are assumed to be uniformly distributed. The MST-ME method is used to construct the unknown QoI response. A total of 6 iterations of the sampling algorithm is performed with weight function (3.4), which results in 30 samples. We expect the MST-ME method to automatically distinguish smooth and discontinuous behavior of the QoI, which is important for this problem, since we have limited initial knowledge of the QoI as a function of the parameters. The results are shown in Figure 18. The results indicate a smooth QoI response, which is approximately constant along the lines height/length = constant. This implies that

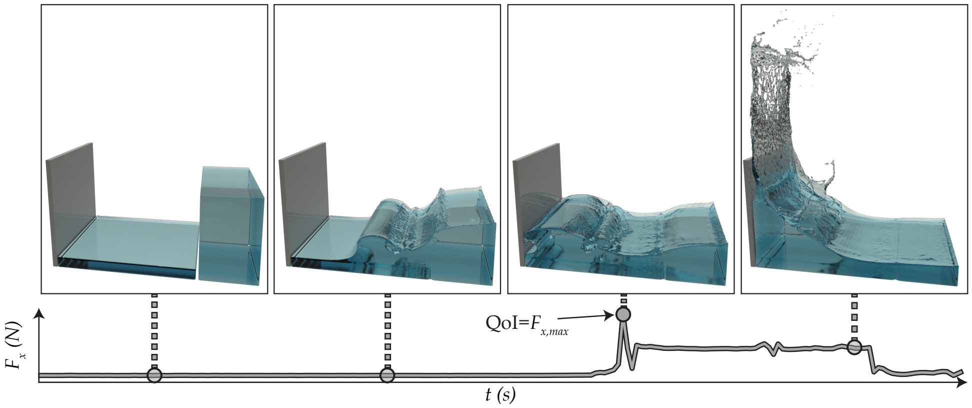

FIG. 17. *Single simulation with height of* 3 *m and a length of* 7 *m. The QoI is the maximum of the wall force F_x.*

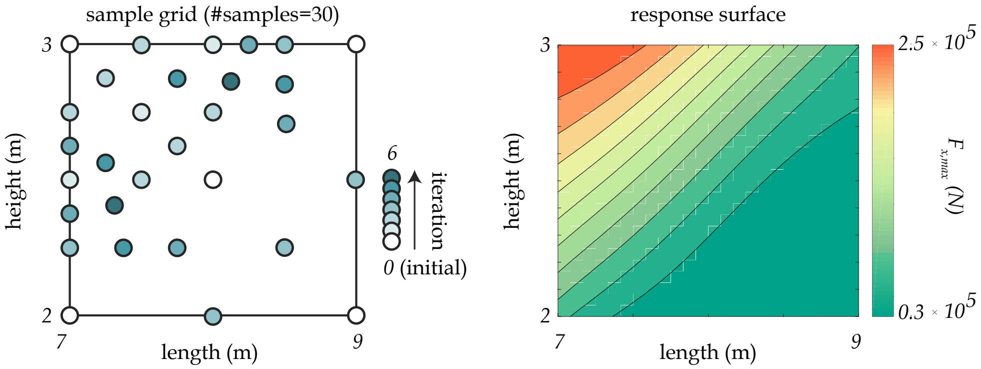

FIG. 18. *QoI of the* 3*D dam break problem, obtained by the MST-ME method with* 6 *iterations.*

for single phase free surface flow, the force on the wall depends roughly on the ratio of height and length, and not on their separate values. The wall force increases when this ratio increases, which is intuitive from a physical point of view. Figure 18 (right) shows that this increase is nonlinear, which corresponds to results previously reported for dry-bed dam break problems [51].

Interestingly, in contrast to the SWE test case, the QoI shows no discontinuity in the parameter space. This is possibly due to neglecting the gas phase in the single phase free surface simulations. When simulated with a gas phase, the pressure buildup in the entrapped gas pocket (see Figure 17) may lead to a discontinuity in the QoI. The strength of our proposed MST-ME method is that we do not require knowledge about the characteristics of the QoI beforehand, as it distinguishes smooth and discontinuous behavior automatically.

5. Conclusion.

In this paper we have presented a novel domain decomposition based interpolation method, the MST-ME method. The unique property of the MST-ME method is that it adaptively constructs a surrogate model as a function of a set of uncertain parameters for both smooth and discontinuous quantities of interest. The three key ingredients in the MST-ME method are (I) adaptive sampling based on an

MST with a smart weight function, (II) discontinuity detection and sample classification with an SVM algorithm, and (III) least orthogonal interpolation to construct local approximations. This combination of robust methods makes the MST-ME method a very practical method that is applicable to a wide range of UQ problems.

The MST-ME method has been applied to several numerical examples: domain decomposition, Genz function approximation, 1D SWEs, and a 3D dam break problem. Discontinuities present in these test cases are effectively captured by the method. In all cases, fast convergence is obtained, leading to an accurate surrogate model already at a relatively low number of model runs. This surrogate model can be directly used as a fast tool for UQ (for example, with Monte Carlo type methods), but it is also a great tool for the parametric solution of black-box models, including PDEs. We also foresee application of this surrogate model in the solution of inverse problems. The freedom in the weighting function of the MST offers many applications, such as adaptive sampling for reliability analysis, where the weighting function may be adapted such that samples are placed in regions of low probability.

Currently, the MST-ME method does not include the option to detect discontinuities that do not divide the random space into several simply connected subdomains. Furthermore, the MST-ME method cannot yet preserve the symmetry in the node distributions, which might be advantageous in certain special cases (e.g., when both the model and the underlying PDF of the random variables are symmetric). Last, the least orthogonal interpolation does not necessarily use all sample points in the construction of the local approximation. By adding a term to the weight function of the MST, which accounts for the stability of the interpolant (as is done for example in Leja nodes [52]), the sample locations may be further improved, such that all samples are used in the construction of the interpolant.

REFERENCES

[1] H. Xiao, J. L. Wu, J. X. Wang, R. Sun, and C. J. Roy, *Quantifying and reducing model-form uncertainties in Reynolds-averaged Navier-Stokes simulations: A data-driven, physics-informed Bayesian approach*, J. Comput. Phys., 324 (2016), pp. 115–136.

[2] W. N. Edeling, R. P. Dwight, and P. Cinnella, *Simplex-stochastic collocation method with improved scalability*, J. Comput. Phys., 310 (2016), pp. 301–328.

[3] F. Simon, P. Guillen, P. Sagaut, and D. Lucor, *A gPC-based approach to uncertain transonic aerodynamics*, Comput. Methods Appl. Mech. Engrg., 199 (2010), pp. 1091–1099.

[4] K.-H. Cho, S.-Y. Shin, W. Kolch, and O. Wolkenhauer, *Experimental design in systems biology, based on parameter sensitivity analysis using a Monte Carlo method: A case study for the TNF-mediated NF-B signal transduction pathway*, Simulation, 79 (2003), pp. 726–739.

[5] R. Abagyan and M. Totrov, *Biased probability Monte Carlo conformational searches and electrostatic calculations for peptides and proteins*, J. Molec. Biol., 235 (1994), pp. 983–1002.

[6] E. M. Constantinescu, V. M. Zavala, M. Rocklin, S. Lee, and M. Anitescu, *A computational framework for uncertainty quantification and stochastic optimization in unit commitment with wind power generation*, IEEE Trans. Power Systems, 26 (2011), pp. 431–441.

[7] J. Mateos, T. Gonzalez, D. Pardo, V. Hoel, and A. Cappy, *Monte Carlo simulator for the design optimization of low-noise HEMTs*, IEEE Trans. Electron Devices, 47 (2000), pp. 1950–1956.

[8] M. Papadrakakis and N. D. Lagaros, *Reliability-based structural optimization using neural networks and Monte Carlo simulation*, Comput. Methods Appl. Mech. Engrg., 191 (2002), pp. 3491–3507.

[9] E. J. Hopfinger and V. Baumbach, *Liquid sloshing in cylindrical fuel tanks*, in Progress in Propulsion Physics, Vol. 1, EDP Sciences, Les Ulis, France, 2009, pp. 279–292.

[10] R. Marsooli and W. Wu, *3-D finite-volume model of dam-break flow over uneven beds based on VOF method*, Adv. Water Res., 70 (2014), pp. 104–117.

[11] L. A. Larocque, J. Imran, and M. H. Chaudhry, *3-D numerical simulation of partial breach dam-break flow using the LES and k-ε turbulence models*, J. Hydraulic Res., 51 (2013), pp. 145–157.

[12] J. Hammersley, *Monte Carlo Methods*, Springer, Dordrecht, The Netherlands, 1964.

[13] D. Xiu, *Fast numerical methods for stochastic computations*, Commun. Comput. Phys., 5 (2009), pp. 242–272.

[14] I. Babuška, F. Nobile, and R. Tempone, *A stochastic collocation method for elliptic partial differential equations with random input data*, SIAM J. Numer. Anal., 45 (2007), pp. 1005–1034.

[15] F. Nobile, R. Tempone, and C. G. Webster, *A sparse grid stochastic collocation method for partial differential equations with random input data*, SIAM J. Numer. Anal., 46 (2008), pp. 2309–2345.

[16] C. Canuto, M. Y. Hussaini, A. M. Quarteroni, and T. A. Zang, Jr., *Spectral Methods in Fluid Dynamics*, Springer, New York, 1988.

[17] J. A. S. Witteveen, A. Loeven, and H. Bijl, *An adaptive stochastic finite elements approach based on Newton-Cotes quadrature in simplex elements*, Comput. & Fluids, 38 (2009), pp. 1270–1288.

[18] J. D. Jakeman and S. G. Roberts, *Local and Dimension Adaptive Sparse Grid Interpolation and Quadrature*, preprint, arXiv:1110.0010, 2011.

[19] X. Wan and G. E. Karniadakis, *Multi-element generalized polynomial chaos for arbitrary probability measures*, SIAM J. Sci. Comput., 28 (2006), pp. 901–928.

[20] J. Foo, X. Wan, and G. E. Karniadakis, *The multi-element probabilistic collocation method (ME-PCM): Error analysis and applications*, J. Comput. Phys., 227 (2008), pp. 9572–9595.

[21] J. D. Jakeman, A. Narayan, and D. Xiu, *Minimal multi-element stochastic collocation for uncertainty quantification of discontinuous functions*, J. Comput. Phys., 242 (2013), pp. 790–808.

[22] A. Gorodetsky and Y. Marzouk, *Efficient localization of discontinuities in complex computational simulations*, SIAM J. Sci. Comput., 36 (2014), pp. A2584–A2610.

[23] J. B. Kruskal, *On the shortest spanning subtree of a graph and the traveling salesman problem*, Proc. Amer. Math. Soc., 7 (1956), pp. 48–50.

[24] R. L. Graham and P. Hell, *On the history of the minimum spanning tree problem*, Ann. Hist. Comput., 7 (1985), pp. 43–57.

[25] G. S. El-tawel and A. K. Helmy, *An edge detection scheme based on least squares support vector machine in a contourlet HMT domain*, Appl. Soft Comput., 26 (2015), pp. 418–427.

[26] M. K. S. Varma, N. K. K. Rao, K. K. Raju, and G. P. S. Varma, *Pixel-based classification using support vector machine classifier*, in 2016 IEEE 6th International Conference on Advanced Computing (IACC), Piscataway, NJ, 2016, pp. 51–55.

[27] C. J. C. Burges, *A tutorial on support vector machines for pattern recognition*, Data Min. Knowl. Disc., 2 (1998), pp. 121–167.

[28] B. Scholkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, MIT Press, Cambridge, MA, 2001.

[29] A. Narayan and D. Xiu, *Stochastic collocation methods on unstructured grids in high dimensions via interpolation*, SIAM J. Sci. Comput., 34 (2012), pp. A1729–A1752.

[30] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, *The quickhull algorithm for convex hulls*, ACM Trans. Math. Software, 22 (1996), pp. 469–483.

[31] J. E. Goodman and J. O'Rourke, eds., *Handbook of Discrete and Computational Geometry*, CRC Press, Boca Raton, FL, 1997.

[32] J. A. S. Witteveen and G. Iaccarino, *Simplex elements stochastic collocation for uncertainty propagation in robust design optimization*, in 48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, American Institute of Aeronautics and Astronautics, Reston, VA, 2010.

[33] J. Witteveen and G. Iaccarino, *Simplex stochastic collocation with random sampling and extrapolation for nonhypercube probability spaces*, SIAM J. Sci. Comput., 34 (2012), pp. A814–A838.

[34] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, Cambridge, MA, 2016, http://www.deeplearningbook.org.

[35] J. D. Jakeman, R. Archibald, and D. Xiu, *Characterization of discontinuities in high-dimensional stochastic problems on adaptive sparse grids*, J. Comput. Phys., 230 (2011), pp. 3977–3997.

[36] C.-C. Chang and C.-J. Lin, *LIBSVM: A library for support vector machines*, ACM Trans. Intell. Systems Technol., 2 (2011), 27.

[37] Y. Ohtake, A. Belyaev, and H.-P. Seidel, *3D scattered data interpolation and approximation with multilevel compactly supported RBFs*, Graph. Models, 67 (2005), pp. 150–165.

[38] A. Gorodetsky and Y. Marzouk, *Mercer kernels and integrated variance experimental design: Connections between Gaussian process regression and polynomial approximation*, SIAM/ASA J. Uncertain. Quantif., 4 (2016), pp. 796–828.

[39] V. Vapnik and O. Chapelle, *Bounds on error expectation for support vector machines*, Neural Comput., 12 (2000), pp. 2013–2036.

[40] D. W. Scott and G. R. Terrell, *Biased and unbiased cross-validation in density estimation*, J. Amer. Statist. Assoc., 82 (1987), pp. 1131–1146.

[41] G. Blatman, *Adaptive Sparse Polynomial Chaos Expansions for Uncertainty Propagation and Sensitivity Analysis*, Ph.D. Thesis, Université Blaise Pascal, Clermont-Ferrand, France, 2009.

[42] S. Fortune, *A sweepline algorithm for Voronoi diagrams*, in Proceedings of SCG '86, ACM, New York, 1986, pp. 313–322.

[43] R. C. Prim, *Shortest connection networks and some generalizations*, Bell System Tech. J., 36 (1957) 1389–1401.

[44] A. Genz, *Testing multidimensional integration routines*, in Proceedings of International Conference on Tools, Methods and Languages for Scientific and Engineering Computation, Amsterdam, North-Holland, 1984, pp. 81–94.

[45] A. Narayan, J. D. Jakeman, and T. Zhou, *A Christoffel function weighted least squares algorithm for collocation approximations*, Math. Comp., 86 (2017), pp. 1913–1947.

[46] J. Helton and F. Davis, *Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems*, Reliab. Eng. Syst. Saf., 81 (2003), pp. 23–69.

[47] C. B. Vreugdenhil, *Numerical Methods for Shallow-Water Flow*, Springer, Dordrecht, The Netherlands, 2011.

[48] R. J. LeVeque, *Finite Volume Methods for Hyperbolic Problems*, Cambridge University Press, Cambridge, 2002.

[49] E. F. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction*, Springer, Berlin, 1999.

[50] A. J. C. Crespo, J. M. Dominguez, B. D. Rogers, M. Gomez-Gesteira, S. Longshaw, R. Canelas, R. Vacondio, A. Barreiro, and O. Garcia-Feal, *DualSPHysics: Open-source parallel CFD solver based on smoothed particle hydrodynamics (SPH)*, Comput. Phys. Commun., 187 (2015), pp. 204–216.

[51] L. Lobovský, E. Botia-Vera, F. Castellana, J. Mas-Soler, and A. Souto-Iglesias, *Experimental investigation of dynamic pressure loads during dam break*, J. Fluids Struct., 48 (2014), pp. 407–434.

[52] A. Narayan and J. D. Jakeman, *Adaptive Leja sparse grid constructions for stochastic collocation and high-dimensional approximation*, SIAM J. Sci. Comput., 36 (2014), pp. A2952–A2983.