SIAM REVIEW
Vol. 58, No. 3, pp. 517–551

Hyperspherical Sparse Approximation Techniques for High-Dimensional Discontinuity Detection*

Guannan Zhang[†]
Clayton G. Webster[†]
Max Gunzburger[‡]
John Burkardt[‡]

Abstract. This work proposes a hyperspherical sparse approximation framework for detecting jump discontinuities in functions in high-dimensional spaces. The need for a novel approach results from the theoretical and computational inefficiencies of well-known approaches, such as adaptive sparse grids, for discontinuity detection. Our approach constructs the hyperspherical coordinate representation of the discontinuity surface of a function. Then sparse approximations of the transformed function are built in the hyperspherical coordinate system, with values at each point estimated by solving a one-dimensional discontinuity detection problem. Due to the smoothness of the hypersurface, the new technique can identify jump discontinuities with significantly reduced computational cost, compared to existing methods. Several approaches are used to approximate the transformed discontinuity surface in the hyperspherical system, including adaptive sparse grid and radial basis function interpolation, discrete least squares projection, and compressed sensing approximation. Moreover, hierarchical acceleration techniques are also incorporated to further reduce the overall complexity. Rigorous complexity analyses of the new methods are provided, as are several numerical examples that illustrate the effectiveness of our approach.

Key words. discontinuity detection, hyperspherical coordinates, adaptive approximations, sparse grid interpolation, discrete projection, least squares, compressed sensing, hierarchical methods

AMS subject classifications. 65D15, 65N35, 65N12, 65N15, 65C20, 65C30

DOI. 10.1137/16M1071699

1. Introduction.

Numerical approximation is an important tool used to define solution techniques for physical, biological, social, and economic systems. In simulations of such systems, the relationship between the inputs that drive the system

[*] Published electronically August 4, 2016. This paper originally appeared in *SIAM Journal on Numerical Analysis*, Volume 53, Number 3, 2015, pages 1508–1536. This material is based upon work supported in part by the U.S. Air Force of Scientific Research under grants 1854-V521-12 and FA9550-15-1-0001; the U.S. Defense Advanced Research Projects Agency, Defense Sciences Office under contract and award HR0011619523 and 1868-A017-15; the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics program under contracts and awards ERKJ259, ERKJE45, and DE-SC0010678; and by the Laboratory Directed Research and Development program at the Oak Ridge National Laboratory, which is operated by UT-Battelle, LLC., for the U.S. Department of Energy under contract DE-AC05-00OR22725.

http://www.siam.org/journals/sirev/58-3/M107169.html

[†] Department of Computational and Applied Mathematics, Oak Ridge National Laboratory, Oak Ridge, TN 37831 (zhangg@ornl.gov, webstercg@ornl.gov).

[‡] Department of Scientific Computing, Florida State University, Tallahassee, FL 32306 (gunzburg@fsu.edu, jburkardt@fsu.edu).

and the outputs, i.e., the system responses, are described by a multivariate function which is usually the target of the numerical approximation. Often the target function exhibits jump discontinuities, which have motivated many research efforts devoted to discontinuity detection. Traditionally, discontinuity detection has been associated with capturing jump discontinuities in a process with respect to temporal and/or spatial variables; thus, most efforts are restricted to low-dimensional problems. However, high-dimensional discontinuity detection is of significant importance in cases for which the system outputs depend on a large number of input variables. For example, this challenge arises in uncertainty quantification (UQ), where physical systems with uncertainties are described by stochastic partial differential equations (SPDEs) with random input data. It is well known that an output of interest derived from the solution of an SPDE may depend on a large number of random variables. Outputs of interest often contain jump discontinuities, sometimes because of irregular behavior with respect to random coefficients, or because the output of interest itself is defined in terms of nonsmooth functions, e.g., indicator functions. Such challenges also arise in optimization and control problems where, again, the controls are characterized using a large number of parameters, and discontinuous cost functionals typically arise. As such, the development of accurate and efficient numerical techniques for detecting high-dimensional discontinuities is important to not only UQ and control, but to other mathematics, engineering, and science research communities as well.

A straightforward approach to resolving the challenges faced when approximating discontinuous functions is to first subdivide the high-dimensional domain of definition into several subdomains, in each of which the target function is continuous or even smoother. Then, in each subdomain, construct a piecewise continuous polynomial approximation using mature methods such as sparse interpolants [6, 24, 25] or orthogonal polynomial expansions [12]. Obviously, these approaches require that the boundaries of the subdomains follow the discontinuity manifolds of the target function. Although such approaches are conceptually easy to understand, they are severely challenged numerically when one requires accurate representations of the detected discontinuities in even moderate dimensions.

Recently, several attempts have been made to alleviate the challenges in locating discontinuities. In [1, 2], a polynomial annihilation approach, originally developed for one- and two-dimensional edge detection, was extended to solve problems in higher dimensions. However, such methods rely on the evaluation of the target function based on a set of local tensor-product grids, so that the number of function evaluations grows exponentially as the dimension increases. Improvements were made in [20] by incorporating an adaptive sparse grid (SG) interpolant in order to reduce the computational cost. SG methods have been demonstrated to be effective in, for example, high-dimensional function approximation [6, 31, 34], numerical integration [5, 13, 14, 16, 26], finite difference methods [15], and Bayesian inference [22, 33, 35]. Moreover, SG methods also have successfully impacted a variety of important applications, including data mining [7], manifold learning [10], image processing [27], and financial engineering [19]. However, the effectiveness of SG approximations inextricably relies on the smoothness of the target function. When approximating a discontinuous function, mesh refinement is invariably needed in the vicinity of discontinuities, resulting in a significant deterioration in the sparsity of the grid; i.e., using an SG method, a discontinuity of an N-dimensional function, which typically occurs across an $(N-1)$-dimensional hypersurface, has to be approximated using a “dense” grid, as illustrated by Example 3.1 in section 3.2. This disadvantage dramatically limits the applicability of SG methods for high-dimensional discontinuity detection.

To combat these challenges, in this work, we propose a novel hyperspherical sparse approximation framework for capturing jump discontinuities of N-dimensional functions. Our approach achieves the desired performance and retains most of the sparsity of existing sparse approximation techniques that are known to be effective for recovering smooth functions. The basic idea is to approximate the discontinuity hypersurface directly instead of approximating the discontinuous function, motivated by observing that the hypersurface itself is often continuous or even smoother. Therefore, the number of samples needed to approximate the hypersurface can be significantly reduced compared to the classic SG interpolation in the Cartesian coordinate system. To achieve this, the first step is to define a function representation of the $(N-1)$-dimensional discontinuity hypersurface. Under a mild assumption about its geometry, the hypersurface is transformed into a function in a hyperspherical coordinate system. Note that the transformed function is defined in the subspace constituted by the $N-1$ angle coordinates; the function value at a certain point is the Euclidean distance between the origin of the hyperspherical coordinate system and the discontinuity along the direction determined by the $N-1$ angles. The next step is to develop an approach to evaluate the transformed function, i.e., calculating the desired Euclidean distance at a given point. Fortunately, this is much easier to implement because it reduces to a one-dimensional discontinuity detection problem along each of the directions determined by the $N-1$ angles. Many existing techniques can be used to fulfill this relatively straightforward task, such as the polynomial annihilation or an existing adaptive method. In particular, if the discontinuous function has a *characteristic property* (defined in section 2), e.g., an indicator function, then root-finding methods can be applied as well. Based on the above two steps, a sparse approximation of the discontinuity hypersurface can be constructed in the $(N-1)$-dimensional subspace, with the use of the hyperspherical coordinate system. In this effort, our new strategy is realized by incorporating two types of sparse approximations. The first one is based on *deterministic* sampling, specifically an adaptive SG interpolation of the hypersurface; the second is based on *random* sampling, specifically radial basis function (RBF) interpolation [4, 32], discrete least squares (DLS) projection [9, 23], and compressed sensing (CS) approximation [8, 11, 30].[1]

The efficiency of our algorithm is characterized by the total number of function evaluations required by the approximation. Thus, the computational complexity is not the number of sample points, but is the sum of the number of function evaluations consumed by all the one-dimensional discontinuity detection problems. Taking the bisection method as an example, the number of iterations required to achieve a prescribed accuracy is determined by the length of the initial search interval. Thus, to further improve the computational efficiency, we incorporate the hierarchical acceleration technique proposed in [12, 18] into the hyperspherical SG interpolation. Specifically, the approximation on a coarse sparse grid is used to predict the value of the transformed function at the newly added points on a finer sparse grid. In this way, the length of the initial search interval for each bisection simulation is significantly reduced, as is the necessary number of search iterations. It is worth noting that the proposed method differs from existing efforts on manifold learning [10, 28], which requires a set of data points that are on/near the target manifold, and the solution procedure is to find the optimal approximation in the sparse polynomial subspace by

[1] The RBF, DLS, and CS approximations can also be constructed from *deterministic* abscissas, but the goal of using random sampling in this effort is to handle the irregular geometry of the domain of the transformed function in the hyperspherical system.

minimizing objective functions. In our approach, we do not require a set of points on/near the target hypersurface to run our algorithms. The sample points on/near the hypersurface are captured using one-dimensional root-finding algorithms.

The main contributions of this paper are summarized as follows.

- A hyperspherical sparse approximation framework for high-dimensional discontinuity detection is constructed.
- The performance of several approaches for the evaluation of the transformed function is investigated.
- The new framework is realized by incorporating existing sparse approximation techniques based on either deterministic or random sampling.
- The computational efficiency of the hyperspherical SG interpolation is improved by incorporating hierarchical acceleration techniques.
- Rigorous complexity analyses are provided for the proposed algorithms in the context of using hyperspherical SG interpolation.

The rest of the paper is organized as follows. The specific problem definition and preliminary notions are discussed in section 2. In section 3, classic adaptive SG interpolation is briefly reviewed and an example is given to illustrate its disadvantages when attempting to detect discontinuities in even moderate dimensions. In section 4, we discuss how hyperspherical coordinates can be used to represent a surface across which a multivariate function is discontinuous. Our main results are given in sections 5 and 6, where hyperspherical sparse approximations of a discontinuity surface are constructed using sparse grids and random sampling, respectively. Numerical tests are provided in section 7, and concluding remarks are provided in section 8.

2. Problem Setting.

Let Γ denote an open bounded domain in $\mathbb{R}^N$, $N \geq 1$, and let $\partial\Gamma$ denote its boundary. We assume there exists an $(N-1)$-dimensional hypersurface in Γ, denoted by γ, separating the domain Γ into disjoint open subdomains Γ_1 and Γ_2, such that $\Gamma = \Gamma_1 \cup \gamma \cup \Gamma_2$, $\overline{\Gamma}_1 \cap \overline{\Gamma}_2 = \gamma$, and $\Gamma_1 \cap \gamma = \Gamma_2 \cap \gamma = \Gamma_1 \cap \Gamma_2 = \emptyset$. We observe that the volume of γ in $\mathbb{R}^N$ is zero and Γ_1 and Γ_2 are both open along γ. We consider a generic N-dimensional discontinuous function $f(\boldsymbol{y}) : \Gamma \to \mathbb{R}$ given by

$$f(\boldsymbol{y}) = \begin{cases} f_1(\boldsymbol{y}) & \text{if } \boldsymbol{y} \in \overline{\Gamma}_1, \\ f_2(\boldsymbol{y}) & \text{if } \boldsymbol{y} \in \overline{\Gamma}_2 \backslash \gamma, \end{cases} \tag{2.1}$$

where $\boldsymbol{y} = (y_1, \ldots, y_N) \in \mathbb{R}^N$ and $f_1(\boldsymbol{y})$ and $f_2(\boldsymbol{y})$ are continuous functions in $\overline{\Gamma}_1$ and $\overline{\Gamma}_2 \backslash \gamma$, respectively. Based on the fact that $f(\boldsymbol{y}) = f_1(\boldsymbol{y})$ for $\boldsymbol{y} \in \gamma \subseteq \partial\Gamma_1$, we assume $f(\boldsymbol{y})$ has a jump discontinuity on γ such that $f_1(\boldsymbol{y}^*) = \lim_{\Gamma_1 \ni \boldsymbol{y} \to \boldsymbol{y}^* \in \gamma} f_1(\boldsymbol{y}) \neq \lim_{\Gamma_2 \ni \boldsymbol{y} \to \boldsymbol{y}^* \in \gamma} f_2(\boldsymbol{y}) < +\infty$ for $\boldsymbol{y}^* \in \gamma$, which means the discontinuity only occurs when approaching γ from the subdomain Γ_2. The goal is to accurately capture the discontinuity hypersurface γ. We also assume that $\partial\Gamma_1$ is a *continuous* hypersurface such that Γ_1 and Γ_2 are disjoint. As such, there exists a *continuous* function $G(\boldsymbol{y})$ defined in $\overline{\Gamma}$ such that $\gamma = \{\boldsymbol{y} \in \overline{\Gamma} \,|\, G(\boldsymbol{y}) = 0\}$, i.e., γ is implicitly defined by the equation $G(\boldsymbol{y}) = 0$, and such that $f = f_1(\boldsymbol{y})$ for $G(\boldsymbol{y}) > 0$ (i.e., $\boldsymbol{y} \in \overline{\Gamma}_1 \backslash \gamma$) and $f = f_2(\boldsymbol{y})$ for $G(\boldsymbol{y}) < 0$ (i.e., $\boldsymbol{y} \in \overline{\Gamma}_2 \backslash \gamma$). Note that $G(\boldsymbol{y}) = 0$ is only an abstract representation of γ and that its availability is not necessary for detecting the discontinuity. Moreover, for a specific γ, the function $G(\boldsymbol{y})$ is not unique.

In one dimension ($N = 1$), γ reduces to one or two points in $\Gamma \subset \mathbb{R}$, so that it is relatively easy to capture the discontinuity of $f(\boldsymbol{y})$. However, in higher dimensions ($N > 1$), detecting discontinuities becomes difficult because γ is, in general, an $(N-1)$-dimensional hypersurface with zero measure in $\mathbb{R}^N$. What is worse, there is no direct information available about the location or geometry of γ, so that we can only rely on indirect information about $f(\boldsymbol{y})$ and $G(\boldsymbol{y})$ to infer the location of γ. In this work, $f(\boldsymbol{y})$ in (2.1) is treated as a black-box function; i.e., given any $\boldsymbol{y} \in \overline{\Gamma}$ as an input, the function value can be obtained as an output without any knowledge about the analytical expressions of $f(\boldsymbol{y})$ or $G(\boldsymbol{y})$. Before moving forward, we provide two examples of discontinuous functions of interest.

Example 2.1. Consider the generic function $f(\boldsymbol{y}) : \overline{\Gamma} \to \mathbb{R}$ defined by

$$f(\boldsymbol{y}) = \begin{cases} f_1(\boldsymbol{y}) & \text{if } \ y_1^2 + \cdots + y_N^2 \leq \mu^2, \\ f_2(\boldsymbol{y}) & \text{if } \ y_1^2 + \cdots + y_N^2 > \mu^2, \end{cases} \tag{2.2}$$

where f_1 and f_2 are continuous functions and μ is a positive real constant. In this case, the function $G(\boldsymbol{y})$ is defined by $G(\boldsymbol{y}) = \mu^2 - \sum_{n=1}^N y_n^2$. The discontinuity $\gamma = \{\boldsymbol{y} \in \mathbb{R}^N \mid G(\boldsymbol{y}) = 0\}$ is a sphere in $\mathbb{R}^N$ with radius μ and $\partial\Gamma_1 \cup \partial\Gamma = \emptyset$, $\gamma = \partial\Gamma_1$. There are three specific scenarios one must consider:

(S_1) $f(\boldsymbol{y})$ can be evaluated as a black-box function;

(S_2) $f(\boldsymbol{y})$ is the characteristic function of Γ_1, e.g., $f_1(\boldsymbol{y}) = 1$ and $f_2(\boldsymbol{y}) = 0$;

(S_3) Both $f(\boldsymbol{y})$ and $G(\boldsymbol{y})$ can be evaluated as black-box functions.

Example 2.2 (probability of an event depending on the solution of a stochastic model). Let the function $u(\boldsymbol{y}, \mathbf{x}) : \Gamma \times D \to \mathbb{R}$ denote the solution of an SPDE given by $\mathcal{L}(a(\boldsymbol{y}, \mathbf{x}))[u(\boldsymbol{y}, \mathbf{x})] = h(\boldsymbol{y}, \mathbf{x})$, where the coefficient $a(\boldsymbol{y}, \mathbf{x})$ of the differential operator $\mathcal{L}$ and the right-hand side $h(\boldsymbol{y}, \mathbf{x})$ are random fields, and $\mathbf{x} \in D \subset \mathbb{R}^d$ ($d = 1, 2, 3$) and $\boldsymbol{y} \in \Gamma \subset \mathbb{R}^N$ denote physical and random variables, respectively. By assuming that $\boldsymbol{y}$ has a joint probability density function $\varrho(\boldsymbol{y}) \colon \Gamma \to \mathbb{R}_+$ with $\varrho(\boldsymbol{y}) \in L^\infty(\Gamma)$, in practice, we may be interested in quantifying the probability of an event about $u(\boldsymbol{y}, \mathbf{x})$. For example, such an event may be the spatial average $F(u) = \frac{1}{|D|} \int_D u(\boldsymbol{y}, \mathbf{x}) d\mathbf{x}$ exceeding a threshold value $\overline{u}$, where $|D|$ denotes the volume of the physical domain D. This probability can be expressed as

$$\mathbb{P}\big(F(u) \geq \overline{u}\big) = \int_\Gamma \mathcal{X}_{\{F(u) \geq \overline{u}\}}(\boldsymbol{y}) \rho(\boldsymbol{y}) d\boldsymbol{y}, \tag{2.3}$$

where $\mathcal{X}_{\{F(u) \geq \overline{u}\}}(\boldsymbol{y})$ is the characteristic function. As such, the target function $f(\boldsymbol{y})$ is $\mathcal{X}_{\{F(u) \geq \overline{u}\}}(\boldsymbol{y}) \rho(\boldsymbol{y})$ and γ is determined by $G(\boldsymbol{y}) = F(u(\boldsymbol{y})) - \overline{u} = 0$.

From the above examples, we observe that, in practice, there may be additional indirect information available about $f(\boldsymbol{y})$ and $G(\boldsymbol{y})$ that can help one capture discontinuities. For instance, in Example 2.2, when defining $\Gamma_1 = \{\boldsymbol{y} \in \Gamma \mid \mathcal{X}_{\{F(u) \geq \overline{u}\}}(\boldsymbol{y}) = 1\}$, the function $G(\boldsymbol{y})$ can be evaluated as well and the membership of a given $\boldsymbol{y} \in \Gamma$ in the subdomain Γ_1 can be determined by the computable value of $f(\boldsymbol{y})$. Thus, we consider discontinuity detection problems under one of the following three assumptions:

A_1: given $\boldsymbol{y} \in \Gamma$, only $f(\boldsymbol{y})$ can be evaluated;

A_2: given $\boldsymbol{y} \in \Gamma$, the value $f(\boldsymbol{y})$ can determine if $f(\boldsymbol{y}) = f_1(\boldsymbol{y})$ or $f(\boldsymbol{y}) = f_2(\boldsymbol{y})$, i.e., if $\boldsymbol{y} \in \Gamma_1$ or $\boldsymbol{y} \in \Gamma_2$;

A_3: given $\boldsymbol{y} \in \Gamma$, both $f(\boldsymbol{y})$ and $G(\boldsymbol{y})$ can be evaluated.

It is easy to see that A_2 is a sufficient condition for A_1 and that A_3 is a sufficient condition for both A_1 and A_2. Under A_1, it is known that there exist jump discontinuities in Γ, but no information about the location of γ can be inferred from the function values of $f(\boldsymbol{y})$. In the context of A_2, function values of $f(\boldsymbol{y})$ can indicate the membership of a given point $\boldsymbol{y} \in \Gamma$ in the subdomain $\overline{\Gamma}_1 \in \Gamma$, which is referred to as the *characteristic property.* Under A_3, because $G(\boldsymbol{y})$ can be evaluated directly, detecting γ is equivalent to finding all the roots of the implicit equation $G(\boldsymbol{y}) = 0$. In one dimension ($N = 1$), this is straightforward to accomplish using classic root-finding algorithms, e.g., the bisection method. In higher dimensions, classic root-finding methods might make it easy to find one root, but approximately determining the whole surface γ is, in general, difficult. It is natural to look for more efficient algorithms for dealing with discontinuous functions satisfying A_2 or A_3. Such improved methods are discussed in detail in section 4.

Because it is almost impossible to solve for the analytical expression describing the hypersurface γ, the main goal of our effort is to efficiently construct, in N dimensions, an accurate approximate hypersurface, denoted by $\widetilde{\gamma}$. To assess the performance of our approaches, the *accuracy* of $\widetilde{\gamma}$ as an approximation of γ is measured by the distance between γ and $\widetilde{\gamma}$ defined as

$$e_\infty = \mathrm{dist}(\gamma, \widetilde{\gamma}) = \sup_{\boldsymbol{x}\in\gamma} \inf_{\boldsymbol{x}'\in\widetilde{\gamma}} |\boldsymbol{x} - \boldsymbol{x}'|. \tag{2.4}$$

In addition, as indicated in (2.3), we are also interested in estimating the integral of $f(\boldsymbol{y})$ over a subdomain of interest, i.e., either Γ_1 or Γ_2. Without loss of generality, the *accuracy* of $\widetilde{\gamma}$ is thus also assessed by the metric

$$e_{\mathrm{int}} = \left| \int_{\Gamma_1} f(\boldsymbol{y}) d\boldsymbol{y} - \int_{\widetilde{\Gamma}_1} f(\boldsymbol{y}) d\boldsymbol{y} \right|, \tag{2.5}$$

where $\widetilde{\Gamma}_1$ is the approximation of Γ_1 resulting from the approximation $\widetilde{\gamma}$ of γ. On the other hand, as shown in Example 2.2, the computational cost on evaluating $f(\boldsymbol{y})$ or $G(\boldsymbol{y})$ often dominates the total cost of constructing $\widetilde{\gamma}$, e.g., because of the complexity of the PDE solver required to perform those evaluations. Thus, we use the number of function evaluations of either $f(\boldsymbol{y})$ or $G(\boldsymbol{y})$ as the metric to assess the *efficiency* of constructing $\widetilde{\gamma}$.

As discussed in section 1, a straightforward way to estimate the integral $\int_{\Gamma_1} f(\boldsymbol{y}) d\boldsymbol{y}$ is to use Monte Carlo methods, but the computational cost may not be affordable due to the slow convergence. Alternatively, an adaptive SG method has been employed to approximate $f(\boldsymbol{y})$ [20], but its efficiency deteriorates dramatically as the dimension N increases. The new approach proposed in section 4 is a variant of the classic sparse approximation methods but features much improved efficiency. To set the stage, before introducing our approach, we briefly review, in section 3, the standard SG method and illustrate its unsatisfactory performance for discontinuity detection.

3. A Review of Adaptive Sparse Grid Approximations for Discontinuity Detection.

In section 3.1, we briefly review hierarchical SG interpolation (see [6] for details). In section 3.2, the adaptive SG method is introduced and its shortcomings for high-dimensional discontinuity detection are illustrated via a numerical example.

3.1. Hierarchical Sparse Grid Interpolation.

The goal is to construct a Lagrange interpolant to a function $\eta(\boldsymbol{y}) : \Gamma \to \mathbb{R}$ using hierarchical piecewise polynomials [6]. We begin with the one-dimensional hat function having support $[-1, 1]$, given by $\psi(y) = \max\{0\,, 1 - |y|\}$. An arbitrary hat function with support $(y_{l,i} - h_l, y_{l,i} + h_l)$ can be generated by dilation and translation, i.e., $\psi_{l,i}(y) = \psi\big(\frac{y+1-ih_l}{h_l}\big)$, where l denotes the resolution level, $h_l = 2^{-l+1}$ for $l = 0, 1, \ldots$ denotes the grid size of the level l grid, and $y_{l,i} = i\, h_l - 1$ for $i = 0, 1, \ldots, 2^l$ denotes the grid points. The basis function $\psi_{l,i}(y)$ has local support with respect to the level l grid and is centered at the grid point $y_{l,i}$; the number of grid points in the level l grid is $2^l + 1$. The space spanned by the nodal basis $\{\psi_{l,i}(y)\}_{i=0}^{2^l}$ is the standard piecewise-linear finite element space. For each grid level $l > 0$, the interpolant of a continuous function $\eta(y)$ in terms of the nodal basis $\{\psi_{l,i}(y)\}_{i=0}^{2^l}$ is given by $\mathcal{U}_l(\eta) = \sum_{i=0}^{2^l} \eta(y_{l,i})\psi_{l,i}(y)$. By observing that $\mathcal{U}_{l-1}(\eta) = \mathcal{U}_l\big(\mathcal{U}_{l-1}(\eta)\big)$, we define the incremental operator

$$\Delta_l = \mathcal{U}_l - \mathcal{U}_{l-1} \quad \text{for } l \geq 0 \qquad \text{with} \quad \mathcal{U}_{-1} = 0, \tag{3.1}$$

where $\Delta_l(\eta)$ only involves the basis functions $\psi_{l,i}(y)$ for $i = 1, 3, 5, \ldots, 2^l - 1$. Then the interpolant $\mathcal{U}_l(\eta)$ can be represented in the *hierarchical* form

$$\mathcal{U}_l(\eta) = \mathcal{U}_{l-1}(\eta) + \Delta_l(\eta) = \cdots = \mathcal{U}_0(\eta) + \sum_{l'=1}^{l} \Delta_{l'}(\eta), \tag{3.2}$$

where the basis functions needed by $\mathcal{U}_0(\eta)$ and $\Delta_{l'}(\eta)$ for $l' = 1, \ldots, l$ constitute the one-dimensional hierarchical basis of level l.

Next, we consider the hierarchical SG interpolation of a multivariate function $\eta(\boldsymbol{y})$ defined, without loss of generality, over the unit hypercube $\Gamma = [-1, 1]^N \subset \mathbb{R}^N$. The one-dimensional hierarchical basis can be directly extended to the N-dimensional domain Γ using tensorization. Specifically, the N-variate basis function $\psi_{\mathbf{l},\mathbf{i}}(\boldsymbol{y})$ associated with the point $\boldsymbol{y}_{\mathbf{l},\mathbf{i}} = (y_{l_1,i_1}, \ldots, y_{l_N,i_N})$ is defined using tensor products, i.e., $\psi_{\mathbf{l},\mathbf{i}}(\boldsymbol{y}) := \prod_{n=1}^{N} \psi_{l_n,i_n}(y_n)$, where $\mathbf{l} = (l_1, \ldots, l_N) \in \mathbb{N}^N$ is a multi-index indicating the resolution level of the basis function. Note that the resolution level can be different in each of the N directions. Then the level L hierarchical SG approximation $\eta_L(\boldsymbol{y})$ of the target function $\eta(\boldsymbol{y})$ is given by

$$\begin{aligned} \eta_L(\boldsymbol{y}) &= \sum_{l=0}^{L} \sum_{|\mathbf{l}|=l} \left(\Delta_{l_1} \otimes \cdots \otimes \Delta_{l_N}\right)(\eta)(\boldsymbol{y}) \\ &= \eta_{L-1}(\boldsymbol{y}) + \sum_{|\mathbf{l}|=L} \left(\Delta_{l_1} \otimes \cdots \otimes \Delta_{l_N}\right)(\eta)(\boldsymbol{y}) \\ &= \eta_{L-1}(\boldsymbol{y}) + \sum_{|\mathbf{l}|=L} \sum_{\mathbf{i} \in B_{\mathbf{l}}} [\eta(\boldsymbol{y}_{\mathbf{l},\mathbf{i}}) - \eta_{L-1}(\boldsymbol{y}_{\mathbf{l},\mathbf{i}})]\psi_{\mathbf{l},\mathbf{i}}(\boldsymbol{y}) \\ &= \eta_{L-1}(\boldsymbol{y}) + \sum_{|\mathbf{l}|=L} \sum_{\mathbf{i} \in B_{\mathbf{l}}} \omega_{\mathbf{l},\mathbf{i}}\psi_{\mathbf{l},\mathbf{i}}(\boldsymbol{y}) = \sum_{l=0}^{L} \sum_{|\mathbf{l}|=l} \sum_{\mathbf{i} \in B_{\mathbf{l}}} \omega_{\mathbf{l},\mathbf{i}}\psi_{\mathbf{l},\mathbf{i}}(\boldsymbol{y}), \end{aligned} \tag{3.3}$$

where the multi-index set $B_{\mathbf{l}}$ is given by

$$B_{\mathbf{l}} := \left\{ \mathbf{i} \in \mathbb{N}^N \;\middle|\; \begin{array}{lll} i_N \in \{0\} & \text{for } n = 1, \ldots, N & \text{if } l_n = 0 \\ i_N \in \{0, 2\} & \text{for } n = 1, \ldots, N & \text{if } l_n = 1 \\ i_N \in \{1, 3, 5, \ldots, 2^{l_N} - 1\} & \text{for } n = 1, \ldots, N & \text{if } l_n > 1 \end{array} \right\}, \tag{3.4}$$

$|\mathbf{l}| \equiv l_1 + \cdots + l_N \leq l$ defines the resolution level of the incremental operator $\Delta_{l_1} \otimes \cdots \otimes \Delta_{l_N}$, and $\omega_{\mathbf{l},\mathbf{i}} = \eta(\boldsymbol{y}_{\mathbf{l},\mathbf{i}}) - \eta_{L-1}(\boldsymbol{y}_{\mathbf{l},\mathbf{i}})$ denotes the *multidimensional hierarchical surplus*. This interpolant is a direct extension, via the Smolyak algorithm, of the one-dimensional hierarchical interpolant. The definition of the surplus $w_{\mathbf{l},\mathbf{i}}$ is based on the facts that $\eta_l(\eta_{l-1}(\boldsymbol{y})) = \eta_{l-1}(\boldsymbol{y})$ and $\eta_{l-1}(\boldsymbol{y}_{\mathbf{l},\mathbf{i}}) - \eta(\boldsymbol{y}_{\mathbf{l},\mathbf{i}}) = 0$ for $|\mathbf{l}| = l$.

We denote by $\mathcal{H}_{\mathbf{l}}(\Gamma) = \{\boldsymbol{y}_{\mathbf{l},\mathbf{i}} | \mathbf{i} \in B_{\mathbf{l}}\}$ the set of points corresponding to the operator $\Delta_{l_1} \otimes \cdots \otimes \Delta_{l_N}$ with $\mathbf{l} = (l_1, \ldots, l_N)$. Then the sparse grid corresponding to the interpolant η_l is given by

$$\mathcal{H}_l(\Gamma) = \bigcup_{|\mathbf{l}| \leq l} \mathcal{H}_{\mathbf{l}}(\Gamma),$$

where $\mathcal{H}_l(\Gamma)$ is nested, i.e., $\mathcal{H}_{l-1}(\Gamma) \subset \mathcal{H}_l(\Gamma)$. In addition, with $\Delta\mathcal{H}_0(\Gamma) = \mathcal{H}_0(\Gamma)$, we denote by $\Delta\mathcal{H}_l(\Gamma) = \mathcal{H}_l(\Gamma) \backslash \mathcal{H}_{l-1}(\Gamma)$ the set of newly added grid points on level l.

3.2. Adaptive Sparse Grids. By virtue of the surpluses $\omega_{\mathbf{l},\mathbf{i}}$, the interpolant in (3.3) can be represented in a hierarchical manner, i.e.,

$$\eta_L(\boldsymbol{y}) = \eta_{L-1}(\boldsymbol{y}) + \Delta\eta_L(\boldsymbol{y}),$$

where $\eta_{L-1}(\boldsymbol{y})$ is the SG interpolant and $\Delta\eta_L(\boldsymbol{y})$ is the hierarchical surplus interpolant. According to the analyses in [6], for smooth functions, the surpluses $\omega_{\mathbf{l},\mathbf{i}}$ of the sparse grid interpolant $\eta_L(\boldsymbol{y})$ tend to zero as L tends to infinity. For example, in the context of using piecewise-linear hierarchical bases and $\eta(\boldsymbol{y})$ having bounded mixed second-order derivatives, the surplus $\omega_{\mathbf{l},\mathbf{i}}$ can be bounded by

$$|\omega_{\mathbf{l},\mathbf{i}}| \leq C_{\text{surp}} 2^{-2\cdot|\mathbf{l}|} \quad \text{for } \mathbf{i} \in B_{\mathbf{l}}, \tag{3.5}$$

where the constant C_{surp} is independent of the level $|\mathbf{l}|$ and the dimension N (see Lemma 3.3 in [6]). This provides a good avenue for constructing adaptive sparse grids using the magnitude of the surplus as an error indicator, especially for irregular functions having, e.g., steep slopes or jump discontinuities. An alternative adaptive SG approach based on wavelet basis is described in [17].

Specifically, the one-dimensional hierarchical grid has a tree-like structure where each grid point $y_{l,i}$ on level l has two children on level $l+1$, namely, $y_{l+1,2i-1}$ and $y_{l+1,2i+1}$. At each successive interpolation level, the basic idea of adaptivity is to use the hierarchical surplus as an error indicator to refine the grid by adding two new points on the next level for each point on the current level for which the magnitude of its surplus is larger than the prescribed error tolerance. Then it is straightforward to extend the adaptivity from the one dimension to the multidimensional adaptive sparse grid. In general, a grid point in an N-dimensional space has $2N$ children which are also the neighbor points of the parent point. We start with an isotropic sparse grid of level $L_{\min} > 0$ and build an approximation $\eta_{L_{\min}}(\boldsymbol{y})$ in order to capture the main profile of $\eta(\boldsymbol{y})$. Thus, for $L \geq L_{\min}$, we only add those grid points on level L whose parent on level $L-1$ has a surplus greater than the prescribed tolerance. In this way, the N-dimensional adaptive sparse grid interpolant of level L with the error tolerance being $\alpha > 0$ can be represented by

$$\eta_{L,\alpha}(\boldsymbol{y}) = \sum_{l=0}^{L} \sum_{|\mathbf{l}|=l} \sum_{\mathbf{i} \in B_{\mathbf{l}}^{\alpha}} \omega_{\mathbf{l},\mathbf{i}} \psi_{\mathbf{l},\mathbf{i}}(\boldsymbol{y}), \tag{3.6}$$

where the multi-index set $B_{\mathbf{l}}^{\alpha} \subseteq B_{\mathbf{l}}$ is defined by $B_{\mathbf{l}}^{\alpha} = \big\{\mathbf{i} \in B_{\mathbf{l}} \,|\, |\omega_{\mathbf{l},\mathbf{i}}| \geq \alpha\big\}$. The corresponding adaptive sparse grid can be represented by $\mathcal{H}_{L,\alpha}(\Gamma) = \cup_{l=0}^{L} \Delta\mathcal{H}_{l,\alpha}(\Gamma)$, where $\Delta\mathcal{H}_{l,\alpha}(\Gamma) = \Delta\mathcal{H}_l(\Gamma)$ for $l \leq L_{\min}$, and $\Delta\mathcal{H}_{l,\alpha}(\Gamma)$ for $l > L_{\min}$ only contains the sparse grid points added by the mesh refinement.

In the literature, the adaptive SG method has been used to approximate irregular functions [6, 21] in low-dimensional spaces ($N \leq 3$). However, in these cases, the SG

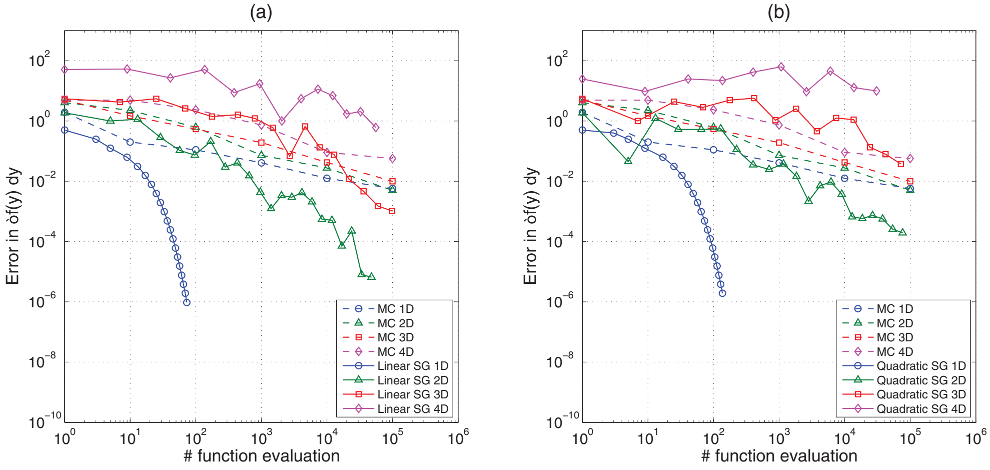

Fig. 1 *The error in the approximations of the integral of $f(\boldsymbol{y})$ given by* (3.7) *vs. the number of function evaluations, using the SG and Monte Carlo (MC in figure) methods with $N = 1, 2, 3, 4$ for* (a) *the SG approximations with a piecewise-linear hierarchical basis and* (b) *the SG approximations with a piecewise-quadratic hierarchical basis.*

method cannot achieve the desired efficiency as in approximating smooth functions. What is worse, it will eventually converge more slowly than a simple Monte Carlo method, even for a moderate four-dimensional discontinuous function, as shown in the following example.

Example 3.1. The target $f(\boldsymbol{y})$ is the characteristic function in $\mathbb{R}^N$ given by

$$f(\boldsymbol{y}) = \begin{cases} 1 & \text{if } 1 - y_1^2 - \cdots - y_N^2 \geq 0, \\ 0 & \text{otherwise}, \end{cases} \tag{3.7}$$

where the discontinuity hypersurface γ is the unit hypersphere in $\mathbb{R}^N$. For $N = 1, 2, 3, 4$, $L_{\min} = 4$, and $L_{\max} = 100$, we build the SG approximation $f_{L,\alpha}(\boldsymbol{y})$ with $\alpha = 0.01$. The error is measured by the metric e_{int} defined in (2.5). Because the surplus will not decay to zero around the hypersphere, mesh refinement will not stop until the level L reaches $L_{\max}$. Thus, we compute and plot, in Figure 1, the error e_{int} vs. the number of function evaluations by increasing the resolution level L up to $L_{\max}$. For comparison, the error of Monte Carlo simulations are also plotted in Figure 1. We observe that the SG approximation outperforms Monte Carlo in the one- and two-dimensional cases, but performs similarly in three dimensions, and in four dimensions Monte Carlo outperforms the adaptive SG method.

To investigate the reason of such failures, we plot in Figure 2 the resulting adaptive sparse grids in two and three dimensions for an error $e_{\text{int}} < 0.01$. Note that mesh refinement places a dense set of grid points in the vicinity of the discontinuities, resulting in a loss of the desired grid sparsity. In fact, the N-dimensional hypershere γ, across which the function is discontinuous, is approximated by an extremely dense grid. It is the loss of sparsity that makes the SG approximation fail when attempting discontinuity detection in high-dimensional spaces. Moreover, because the target function $f(\boldsymbol{y})$ is discontinuous, the accuracy of the SG approximation cannot be improved by using a high-order hierarchical basis [6]. In fact, the accuracy is worse for

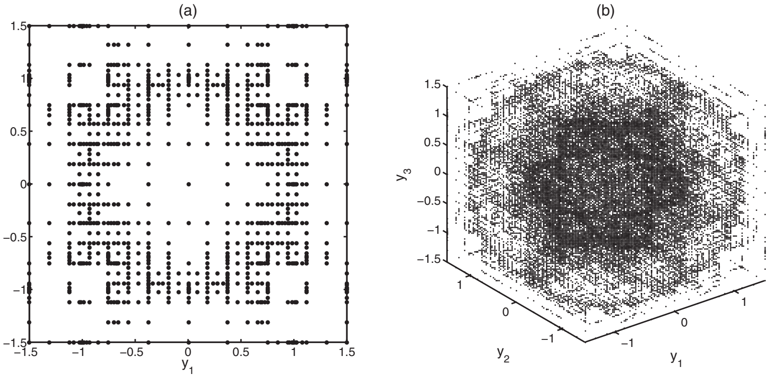

Fig. 2 *The grids produced by the SG method for approximating the integral of* $f(\boldsymbol{y})$ *given by* (3.7) *with linear hierarchical basis and* $e_{int} < 0.01$*:* (a) *the two-dimensional sparse grid has* 969 *points;* (b) *the three-dimensional sparse grid has* 936,093 *points.*

piecewise-quadratic approximations than it is for piecewise-linear approximations; see Figure 1.

4. Hyperspherical Transformation of a Discontinuity Surface.

Later in this paper, we propose a hyperspherical sparse approximation framework that overcomes the disadvantages of the adaptive SG method of section 3 for high-dimensional discontinuity detection. The basic idea is to directly approximate the discontinuity hypersurface γ itself, instead of refining the sparse grid in the vicinity of γ. Unlike the discontinuous function $f(\boldsymbol{y})$ in (2.1), the hypersurface γ is continuous or even smoother, so that the drawbacks of the SG method mentioned above can be avoided when directly approximating γ. However, in general, γ is not a function in the Cartesian coordinate system in $\mathbb{R}^N$, so that the hyperspherical transformation is introduced into our approach to convert γ into a function in the hyperspherical coordinate system. In this section, details about the conversion and the evaluation of the transformed function are discussed in sections 4.1 and 4.2, respectively. In sections 5 and 6, hyperspherical coordinate representations of the discontinuity surface are used to develop a deterministic SG and random sampling approaches, respectively, for efficiently detecting the discontinuity surface.

4.1. Function Representation in the Hyperspherical Coordinates.

A hyperspherical coordinate system is a generalization of the three-dimensional spherical coordinate systems. It has one radial coordinate r ranging over $[0, \infty)$; one angular coordinate θ_{N-1} ranging over $[0, 2\pi]$; and $N-2$ angular coordinates $\theta_1, \dots, \theta_{N-2}$ ranging over $[0, \pi]$. Denoting $\Gamma_s = [0, \pi]^{N-2} \times [0, 2\pi]$, the relation between the hyperspherical coordinates $(r, \theta_1, \dots, \theta_{N-1}) \in [0, \infty) \times \Gamma_s$ and the Cartesian coordinates $\boldsymbol{y} = (y_1, \dots, y_N) \in \mathbb{R}^N$ is given by

$$
\left\{
\begin{aligned}
y_1 &= y_{0,1} + r\cos(\theta_1),\\
y_2 &= y_{0,2} + r\sin(\theta_1)\cos(\theta_2),\\
&\ \vdots\\
y_N &= y_{0,N} + r\sin(\theta_1)\cdots\sin(\theta_{N-2})\sin(\theta_{N-1}),
\end{aligned}
\right.
\tag{4.1}
$$

where $\boldsymbol{y}_0 = (y_{0,1}, \dots, y_{0,N})$ denotes the origin of the hyperspherical coordinate system. Based on this transformation, we would like to transform the discontinuity hypersurface γ defined by the *implicit* equation $G(\boldsymbol{y}) = 0$ into the hyperspherical coordinate system, and represent it by an *explicit* function of $\boldsymbol{\theta} = (\theta_1, \dots, \theta_{N-1})$. To this end, we make the following assumption about the geometry of the subdomain Γ_1 and the origin $\boldsymbol{y}_0$.

Assumption 4.1. For the underlying domain $\Gamma = \Gamma_1 \cup \gamma \cup \Gamma_2$ in (2.1), we assume that Γ_1 is a star-convex domain in $\mathbb{R}^N$ and that a point $\boldsymbol{y}_0$ in Γ_1 is given such that, for all $\boldsymbol{y} \in \Gamma_1$, the line segment $\{\boldsymbol{y}_0 + t\boldsymbol{y} \,|\, t \in [0, 1]\}$ from $\boldsymbol{y}_0$ to $\boldsymbol{y}$ is in Γ_1.

Remark 4.2. When Γ_1 is a convex domain, it is also star convex, and any point in Γ_1 can be used as the origin $\boldsymbol{y}_0$; the function given in Example 2.1 provides an example of this case. If $\boldsymbol{y}_0$ is not known a priori, it can be obtained by Monte Carlo sampling in Γ, as long as $f(\boldsymbol{y})$ has the characteristic property. In practice, $\boldsymbol{y}_0$ is sometimes available for the problem of interest. For instance, as discussed in Example 2.2, the interest of investigating the probability of an event usually results from the occurrence of such an event in a physical experiment with a specific set of parameter values. In this case, these values can be used to define the origin $\boldsymbol{y}_0$. On the other hand, if Γ_1 is not convex, the set of points qualified to be used as $\boldsymbol{y}_0$ is only a subset of Γ_1. In this case, especially when the target function has no characteristic property, it is much more difficult to choose a qualified $\boldsymbol{y}_0$.

Based on the transformation (4.1) with origin $\boldsymbol{y}_0$ satisfying Assumption 4.1, there exists a unique $(N-1)$-dimensional continuous function $g(\boldsymbol{\theta}) : \Gamma_s \to [0, \infty)$ such that $\partial\Gamma_1 = \{(g(\boldsymbol{\theta}), \boldsymbol{\theta}) \,|\, \forall\, \boldsymbol{\theta} \in \Gamma_s\}$. The value of $g(\boldsymbol{\theta})$ is the Euclidean distance between $\boldsymbol{y}_0$ and $\partial\Gamma_1$ along the direction $\boldsymbol{\theta}$. Under the definitions in section 2, given $\boldsymbol{\theta} \in \Gamma_s$, there are two possibilities for the location of $(g(\boldsymbol{\theta}), \boldsymbol{\theta})$, i.e., $(g(\boldsymbol{\theta}), \boldsymbol{\theta}) \in \partial\Gamma \cap \partial\Gamma_1$ or $(g(\boldsymbol{\theta}), \boldsymbol{\theta}) \in \gamma$. Thus, $g(\boldsymbol{\theta})$ is the desired function representation of the discontinuity hypersurface γ. Unlike the equation $G(\boldsymbol{y}) = 0$, $g(\boldsymbol{\theta})$ is an *explicit* representation of γ, so that it becomes feasible to estimate γ directly by approximating $g(\boldsymbol{\theta})$ in Γ_s. However, Assumption 4.1 only guarantees the existence of $g(\boldsymbol{\theta})$, and the value of $g(\boldsymbol{\theta})$ at $\boldsymbol{\theta} \in \Gamma_s$ is unknown a priori. Therefore, a strategy of evaluating $g(\boldsymbol{\theta})$ is provided in section 4.2.

4.2. Evaluation of γ in the Hyperspherical Coordinates. Before moving forward, for clarity we list and explain in Table 1 the notation used in what follows. Essentially, the evaluation of $g(\boldsymbol{\theta})$ becomes a discontinuity detection problem for the one-dimensional function $f_{\boldsymbol{\theta}}(r)$ in the interval $[S_r(\boldsymbol{y}_0), S_r(\boldsymbol{\beta}_{\boldsymbol{\theta}})]$. If $(g(\boldsymbol{\theta}), \boldsymbol{\theta}) \in \partial\Gamma \cap \partial\Gamma_1$,

Table 1 *Notation.*

Notation	Explanation
$g(\boldsymbol{\theta})$	the function representation of $\partial\Gamma_1$ in $\Gamma_s = [0, \pi]^{N-2} \times [0, 2\pi]$
$\widetilde{g}(\boldsymbol{\theta})$	the approximation of $g(\boldsymbol{\theta})$ using root-finding methods
$S(\boldsymbol{y})$	the transformation from Cartesian coordinates $\boldsymbol{y}$ to hyperspherical coordinates $(r, \boldsymbol{\theta})$
$S^{-1}(\boldsymbol{y})$	the inverse transformation of $S(\boldsymbol{y})$
$S_r(\boldsymbol{y})$	the transformation from $\boldsymbol{y}$ to the radial coordinate r
$S_\theta(\boldsymbol{y})$	the transformation from $\boldsymbol{y}$ to the angular coordinates $\boldsymbol{\theta} = (\theta_1, \dots, \theta_{N-1})$
$\boldsymbol{\beta}_{\boldsymbol{\theta}}$	the Cartesian coordinates $(\beta_{\boldsymbol{\theta},1}, \dots, \beta_{\boldsymbol{\theta},N})$ of the intersection point of $\partial\Gamma \cap \partial\Gamma_1$ and the ray from $\boldsymbol{y}_0$ along the direction $\boldsymbol{\theta}$
$f_{\boldsymbol{\theta}}(r)$	the target function f restricted to the ray along the direction $\boldsymbol{\theta}$, i.e., $f_{\boldsymbol{\theta}}(r) = f(S^{-1}(r, \boldsymbol{\theta}))$

then $f_{\boldsymbol{\theta}}(r)$ is a continuous function on the line segment $\{\boldsymbol{y}_0+t\boldsymbol{\beta}_{\boldsymbol{\theta}}\,|\,t\in[0,1]\}$, such that $g(\boldsymbol{\theta})=|\boldsymbol{\beta}_{\boldsymbol{\theta}}-\boldsymbol{y}_0|$; if $(g(\boldsymbol{\theta}),\boldsymbol{\theta})\in\gamma=\partial\Gamma_1\backslash\partial\Gamma$, then $f_{\boldsymbol{\theta}}(r)$ is discontinuous at $r=g(\boldsymbol{\theta})$ so that $g(\boldsymbol{\theta})$ can be estimated by capturing the discontinuity of $f_{\boldsymbol{\theta}}(r)$. We discuss the evaluation of $f(\boldsymbol{y})$ in the absence of the characteristic property (under Assumption A_1) in section 4.2.1 and with the characteristic property (under Assumption A_2 or A_3) in section 4.2.2.

4.2.1. $f(y)$ without the Characteristic Property. Under Assumption A_1 given in section 2, we cannot distinguish whether $f_{\boldsymbol{\theta}}(r)$ is continuous or discontinuous in $[S_r(\boldsymbol{y}_0),S_r(\boldsymbol{\beta}_{\boldsymbol{\theta}})]$, so that one needs to first approximate the whole profile of $f_{\boldsymbol{\theta}}(r)$, then identify the existence and location of the discontinuity by analyzing the approximation. For each $\boldsymbol{\theta}$, we use the one-dimensional adaptive interpolation approach to construct an adaptive approximation of $f_{\boldsymbol{\theta}}(r)$ in the interval $[S_r(\boldsymbol{y}_0),S_r(\boldsymbol{\beta}_{\boldsymbol{\theta}})]$. The adaptivity will automatically refine in the region where $f_{\boldsymbol{\theta}}(r)$ has large variations, including jump discontinuities. To find a value $\widetilde{g}(\boldsymbol{\theta})$ such that $|\widetilde{g}(\boldsymbol{\theta})-g(\boldsymbol{\theta})|\le\tau$, an adaptive interpolant is constructed by setting $\eta=f_{\boldsymbol{\theta}}$ in (3.6), with the maximum level of the adaptive grid being $L_{\max}=\lceil\log_2(|\boldsymbol{y}_0-\boldsymbol{\beta}_{\boldsymbol{\theta}}|/\tau)\rceil$. Note that the hierarchical surplus decays to zero as the level L increases in the smooth region of $f_{\boldsymbol{\theta}}(r)$, but not near the jump discontinuity. Thus, if the mesh refinement stops automatically at a level $L<L_{\max}$, it means that $f_{\boldsymbol{\theta}}(r)$ is continuous in $[S_r(\boldsymbol{y}_0),S_r(\boldsymbol{\beta}_{\boldsymbol{\theta}})]$ so that $g(\boldsymbol{\theta})=|\boldsymbol{\beta}_{\boldsymbol{\theta}}-\boldsymbol{y}_0|$ and $(g(\boldsymbol{\theta}),\boldsymbol{\theta})\in\partial\Gamma\cup\partial\Gamma_1$. Otherwise, due to Assumption 4.1, $f_{\boldsymbol{\theta}}(r)$ has only one jump discontinuity at $(g(\boldsymbol{\theta}),\boldsymbol{\theta})$, and thus $\widetilde{g}(\boldsymbol{\theta})$ can be determined by

$$(r_1,r_2)=\underset{r,r'\in\mathcal{H}_{L_{\max},\alpha}([S_r(\boldsymbol{y}_0),S_r(\boldsymbol{\beta}_{\boldsymbol{\theta}})])}{\arg\max}\frac{|f_{\boldsymbol{\theta}}(r)-f_{\boldsymbol{\theta}}(r')|}{|r-r'|}\quad\text{and}\quad\widetilde{g}(\boldsymbol{\theta})=\frac{1}{2}(r_1+r_2),$$

where $[r_1,r_2]$ is the interval which contains the largest variation of $f_{\boldsymbol{\theta}}(r)$ based on the available samples in $\mathcal{H}_{L_{\max},\alpha}([S_r(\boldsymbol{y}_0),S_r(\boldsymbol{\beta}_{\boldsymbol{\theta}})])$. If τ is sufficiently small, then we have $g(\boldsymbol{\theta})\in[r_1,r_2]$ with $|r_1-r_2|<\tau$.

4.2.2. $f(y)$ with Characteristic Property. Under Assumptions A_2 or A_3, relying on the characteristic property, root-finding approaches can be employed to improve the efficiency of searching. Specifically, to evaluate $g(\boldsymbol{\theta})$ at $\boldsymbol{\theta}\in\Gamma_s$, we first evaluate $f(\boldsymbol{y})$ at $\boldsymbol{y}=\boldsymbol{\beta}_{\boldsymbol{\theta}}$. If $f(\boldsymbol{\beta}_{\boldsymbol{\theta}})=f_1(\boldsymbol{\beta}_{\boldsymbol{\theta}})$, then$(g(\boldsymbol{\theta}),\boldsymbol{\theta})\in\partial\Gamma\cap\partial\Gamma_1$ and $|\boldsymbol{\beta}_{\boldsymbol{\theta}}-\boldsymbol{y}_0|$ is the exact value of $g(\boldsymbol{\theta})$. Otherwise, we have $(g(\boldsymbol{\theta}),\boldsymbol{\theta})\in\gamma\subset\partial\Gamma_1$ and $g(\boldsymbol{\theta})$ is the location of the jump discontinuity of the function $f_{\boldsymbol{\theta}}(r)$, which can be represented by

$$f_{\boldsymbol{\theta}}(r)=\begin{cases} f_1\left(S^{-1}(r,\boldsymbol{\theta})\right) & \text{if } r\le g(\boldsymbol{\theta}),\\ f_2\left(S^{-1}(r,\boldsymbol{\theta})\right) & \text{if } r> g(\boldsymbol{\theta}),\end{cases}$$

where $S^{-1}(r,\boldsymbol{\theta})\in\Gamma$. The simplest root-finding approach is the bisection method starting with $r_{-1}=0$ and $r_0=|\boldsymbol{\beta}_{\boldsymbol{\theta}}-\boldsymbol{y}_0|$. In the kth iteration, we have $r_k=(r_{k-1}+r_{k-2})/2$, where $f_{\boldsymbol{\theta}}(r_{k-1})=f_1(S^{-1}(r_{k-1},\boldsymbol{\theta}))$ and $f_{\boldsymbol{\theta}}(r_{k-2})=f_2(S^{-1}(r_{k-2},\boldsymbol{\theta}))$. For a prescribed accuracy τ such that $|\widetilde{g}(\boldsymbol{\theta})-g(\boldsymbol{\theta})|\le\tau$, the necessary number of iterations K is given by

$$K=\left\lceil\log_2\left(|\boldsymbol{y}_0-\boldsymbol{\beta}_{\boldsymbol{\theta}}|/\tau\right)\right\rceil, \tag{4.2}$$

and the final approximation is defined by $\widetilde{g}(\boldsymbol{\theta})=(r_K+r_{K-1})/2$. Note that the bisection method is preferable when $f(\boldsymbol{y})$ has the characterization property because it only adds one neighboring point at each iteration, whereas the SG method adds two neighboring points at a time.

In the case $f(\boldsymbol{y})$ satisfies Assumption A_3, i.e., $G(\boldsymbol{y})$ can be evaluated as a black-box function, other root-finding methods with faster convergence rates can be used to improve the efficiency of searching. For instance, $G(\boldsymbol{y}) = F(u(\boldsymbol{y})) - \overline{u}$ in Example 2.2 can be evaluated such that the discontinuity of $f_{\boldsymbol{\theta}}(r)$ can also be detected by searching the root of $G(S^{-1}(r, \boldsymbol{\theta})) = F(u(S^{-1}(r, \boldsymbol{\theta}))) - \overline{u} = 0$. In this work, relying on the smoothness of $G(\boldsymbol{y})$, we use the Regula–Falsi method [29], a variant of the secant method. As is the case when using the bisection method, we start with $r_{-1} = 0$ and $r_0 = |\boldsymbol{\beta}_{\boldsymbol{\theta}} - \boldsymbol{y}_0|$; in the $(k+1)$th iteration, r_{k+1} is defined by

$$r_{k+1} = r_k - G(S^{-1}(r_k, \boldsymbol{\theta})) \cdot \frac{r_k - r_{k'}}{G(S^{-1}(r_k, \boldsymbol{\theta})) - G(S^{-1}(r_{k'}, \boldsymbol{\theta}))}, \tag{4.3}$$

where k' is the maximum index less than k such that $G(S^{-1}(r_k, \boldsymbol{\theta})) \cdot G(S^{-1}(r_{k'}, \boldsymbol{\theta})) < 0$. It is known that the Regula–Falsi method converges more slowly than the secant method, but the iterates generated by (4.3) are all contained within the initial interval $[S_r(\boldsymbol{y}_0), S_r(\boldsymbol{\beta}_{\boldsymbol{\theta}})]$. Thus, one does not need to worry about the issue of getting a negative r_k from (4.3). When $|r_k - r_{k'}|$ becomes sufficiently small, one can switch to the secant method to obtain faster convergence.

5. Hyperspherical Sparse Approximation with Deterministic Sampling.

In section 5.1, we develop our approach for detecting discontinuity surfaces by utilizing the deterministic sparse grid (SG) sampling method discussed in section 3. In section 5.2, the efficiency of the proposed algorithm is further improved by incorporating the hierarchical acceleration technique proposed in [18]. Rigorous error estimates and ε-complexity analyses are provided in section 5.3 for the algorithms discussed in sections 5.1 and 5.2.

5.1. Hyperspherical Sparse Grid Interpolation.

We now describe a complete procedure for hyperspherical SG interpolation, where sparse grids can be viewed as a deterministic sampling approach. Under Assumption 4.1, we would like to build an adaptive sparse grid interpolant of $g(\boldsymbol{\theta})$ representing γ in the $(N-1)$-dimensional domain Γ_s.

Remark 5.1. Unlike in section 3 in which sparse grids are used to build approximations in the N-dimensional Cartesian coordinates, here, sparse grids are used to build approximations in the $(N-1)$-dimensional hyperspherical coordinate system $\boldsymbol{\theta}$. The success of the latter approach is due to the smoothness of the function representation of the discontinuity surface in the hyperspherical coordinate system.

At each grid point $\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}}$, $g(\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}})$ is estimated by $\widetilde{g}(\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}})$ using the approaches discussed in section 4.2. Thus, we actually construct an interpolant of the approximation $\widetilde{g}(\boldsymbol{\theta})$. For fixed $L_{\min} > 0$ and $\alpha > 0$, the adaptive sparse grid interpolant at level L is defined by setting $\eta(\boldsymbol{\theta}) = \widetilde{g}(\boldsymbol{\theta})$ in (3.6), i.e.,

$$\widetilde{g}_{L,\alpha}(\boldsymbol{\theta}) = \sum_{l=0}^{L} \sum_{|\mathbf{l}|=l} \sum_{\mathbf{i} \in B_{\mathbf{l}}^{\alpha}} \widetilde{\omega}_{\mathbf{l},\mathbf{l}} \cdot \psi_{\mathbf{l},\mathbf{i}}(\boldsymbol{\theta}), \tag{5.1}$$

where the surpluses $\{\widetilde{\omega}_{\mathbf{l},\mathbf{i}} \mid |\mathbf{l}| \leq L, \mathbf{i} \in B_{\mathbf{l}}^{\alpha}\}$ are computed based on the set of approximate function values $\{\widetilde{g}(\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}}) \mid \boldsymbol{\theta}_{\mathbf{l},\mathbf{i}} \in \mathcal{H}_{L,\alpha}(\Gamma_s)\}$. Recall that if $(g(\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}}), \boldsymbol{\theta}_{\mathbf{l},\mathbf{i}})$ is on the boundary $\partial\Gamma \cap \partial\Gamma_1$, $\widetilde{g}(\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}}) = g(\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}})$ has no numerical error; otherwise, $\widetilde{g}(\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}})$ is computed by either the SG method discussed in section 4.2.1 or one of the root-finding methods discussed in section 4.2.2. The approximated hypersurface $\widetilde{\gamma}$ is given by

$$\widetilde{\gamma} = \left\{ (\widetilde{g}_{L,\alpha}(\boldsymbol{\theta}), \boldsymbol{\theta}) \;\middle|\; \boldsymbol{\theta} \in \Gamma_s \right\}.$$

Algorithm 1 is the main algorithm we use to construct our hyperspherical SG interpolation, where the bisection method is used under Assumption A_2.

ALGORITHM 1. The hyperspherical sparse grid interpolation.

```
Initialize $N$, $L_{\min}$, $\alpha$, $\tau$, $\boldsymbol{y}_0$
$l = -1$
**while** $l = -1$ or $\left\{\Delta\mathcal{H}_{L,\alpha}(\Gamma_s) \neq \emptyset \text{ and } l+1 \leq L_{\max}\right\}$ **do**
    Generate $\Delta\mathcal{H}_{l+1,\alpha}(\Gamma_s)$
    **for** $\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}} \in \Delta\mathcal{H}_{l+1,\alpha}(\Gamma_s)$ **do**
        Search $\boldsymbol{\beta}_{\theta_{\mathbf{l},\mathbf{i}}} = (\beta_{\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}},1}, \dots, \beta_{\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}},N}) \in \Gamma$
        **if** $f(\boldsymbol{\beta}_{\theta_{\mathbf{l},\mathbf{i}}}) = f_1(\boldsymbol{\beta}_{\theta_{\mathbf{l},\mathbf{i}}})$ **then**
            $\widetilde{g}(\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}}) = |\boldsymbol{y}_0 - \boldsymbol{\beta}_{\theta_{\mathbf{l},\mathbf{i}}}|$
        **else**
            Define $K = \left\lceil \log_2\left(|\boldsymbol{y}_0 - \boldsymbol{\beta}_{\theta_{\mathbf{l},\mathbf{i}}}|/\tau\right) \right\rceil$
            Run bisection $\widetilde{g}(\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}}) = r_K$ where $|r_K - g(\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}})| \leq |\boldsymbol{\beta}_{\theta_{\mathbf{l},\mathbf{i}}} - \boldsymbol{y}_0|/2^K$
        **end if**
        $\widetilde{\omega}_{\mathbf{l},\mathbf{i}} = \widetilde{g}(\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}}) - \widetilde{g}_{L,\alpha}(\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}})$
    **end for**
    Update to $\mathcal{H}_{l+1,\alpha}(\Gamma_s)$ by adding $\Delta\mathcal{H}_{l+1,\alpha}(\Gamma_s)$
    $l = l + 1$
**end while**
```

By building the approximation $\widetilde{g}_{L,\alpha}(\boldsymbol{\theta})$, we decompose a high-dimensional discontinuity detection problem to a set of *one-dimensional* discontinuity detection problems which are much easier to solve than the original problem. Due to the continuity of $g(\boldsymbol{\theta})$ resulting from the closeness of $\partial\Gamma_1$, the sparsity of the resulting sparse grid can be retained such that all the disadvantages of the traditional adaptive SG approach shown in Example 3.1 can be completely avoided. As mentioned in section 2, the cost of function evaluations is usually dominant, so that the total computational cost for constructing the approximation $\widetilde{g}_{L,\alpha}(\boldsymbol{\theta})$ can be measured by

$$\mathcal{C}_{\text{total}} = \sum_{l=0}^{L} \sum_{|\mathbf{l}|=l} \sum_{\mathbf{i}\in B_{\mathbf{l}}^{\alpha}} M_{\mathbf{l},\mathbf{i}}^{\tau}, \tag{5.2}$$

where $M_{\mathbf{l},\mathbf{i}}^{\tau}$ is the number of function evaluations for obtaining $\widetilde{g}(\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}})$ with accuracy τ. Note that $M_{\mathbf{l},\mathbf{i}}^{\tau} = 1$ in the sense that $f(\boldsymbol{y})$ has the characteristic property and $(g(\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}}), \boldsymbol{\theta}_{\mathbf{l},\mathbf{i}})$ is on the boundary of Γ. It is well known that the convergence of either the SG method or of root-finding methods heavily depends on the size of the search interval. So far, the search interval for each $\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}}$ is set to $[S_r(\boldsymbol{y}_0), S_r(\boldsymbol{\beta}_{\boldsymbol{\theta}})]$, which is the largest possible interval, because we assume that no knowledge about the function value $g(\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}})$ is known a priori. In the next section, the efficiency of constructing $\widetilde{g}_{L,\alpha}(\boldsymbol{\theta})$ is improved by taking advantage of its hierarchical structure.

5.2. Accelerated Approximation Using Sparse Grid Hierarchies.

In (5.2), at each grid point $\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}}$, $M_{\mathbf{l},\mathbf{i}}^{\tau}$ is determined by the prescribed accuracy τ and the initial search interval. So far, the initial search interval for each $\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}}$ is set to $[S_r(\boldsymbol{y}_0), S_r(\boldsymbol{\beta}_{\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}}})]$

because no knowledge about the value $g(\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}})$ is known a priori. Such an assumption is true on level $L = 0$. However, when $L \geq 1$, by the definition of a surplus, we have

$$\widetilde{g}(\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}}) = \widetilde{g}_{L-1,\alpha}(\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}}) + \widetilde{\omega}_{\mathbf{l},\mathbf{i}}$$

for each newly added point $\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}} \in \Delta\mathcal{H}_{L,\alpha}(\Gamma_s)$ on level L. As such, the SG approximation of level $L-1$ can provide a prediction of $\widetilde{g}(\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}})$ with the error being the unknown surplus. Based on the upper bound given in (3.5), such a prediction will become more and more accurate as L increases. Thus, for $L \geq 1$, we utilize the SG approximation of the previous level to reduce the size of the initial search interval in order to accelerate the evaluation of $\widetilde{g}(\boldsymbol{\theta})$.

Assuming the target function $f(\boldsymbol{y})$ has the characteristic property, we provide the algorithm for the accelerated bisection method in Algorithm 2, which can be extended to other approaches with relative ease. The basic idea behind Algorithm 2 is to set one of the endpoints, e.g., r_{-1}, of the initial search interval to the predicted value given by the interpolated value $\widetilde{g}_{L,\alpha}(\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}})$ at the new added point $\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}}$. Besides that, several practical issues in terms of efficiency and robustness are considered as well. First, one needs to properly define the other endpoint r_0 such that $|r_{-1} - r_0|$ will become smaller as the level L increases and the interval $[r_{-1}, r_0]$ can cover the discontinuity location $g(\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}})$. Theoretically, r_0 can be chosen according to the upper bound of the error $|g(\boldsymbol{\theta}) - \widetilde{g}_{L,\alpha}(\boldsymbol{\theta})|$. However, since the a priori error bound is only known up to a constant, in the computations, we use the hierarchical surplus, which acts as an a posteriori error estimate, to choose the other endpoint r_0. Specifically, for the new added grid points on level L, we initially set the length $|r_{-1} - r_0|$ to the maximum magnitude, denoted by ξ, of all surpluses on level $L-1$. Note that such surpluses actually characterize the error of the interpolant on level $L-2$, which means ξ is not

Algorithm 2. The accelerated bisection method to compute $\widetilde{g}_{\mathbf{l},\mathbf{i}} \approx g(\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}})$ for $\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}} \in \Delta\mathcal{H}_{L,\alpha}(\Gamma_s)$, given $\widetilde{g}_{L,\alpha}(\boldsymbol{\theta})$.

```
$\xi = \max\left\{|\omega_{\mathbf{l}'\mathbf{i}'}| \,\middle|\, \boldsymbol{\theta}_{\mathbf{l}'\mathbf{i}'} \in \mathcal{H}_{L-1,\alpha}(\Gamma_s) \text{ and } |\mathbf{l}'| = L-1\right\}$
Search $\boldsymbol{\beta}_{\theta_{\mathbf{l},\mathbf{i}}} = (\beta_{\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}},1}, \ldots, \beta_{\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}},N}) \in \Gamma$
$r_{-1} = \min\left\{\max\left\{\widetilde{g}_{L-1,\alpha}(\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}}), 0\right\}, |\boldsymbol{y}_0 - \boldsymbol{\beta}_{\theta_{\mathbf{l},\mathbf{i}}}|\right\}$
if $f(S^{-1}(r_{-1}, \boldsymbol{\theta}_{\mathbf{l},\mathbf{i}})) = f_1(S^{-1}(r_{-1}, \boldsymbol{\theta}_{\mathbf{l},\mathbf{i}}))$ then
    $r_0 = \min\{r_{-1} + \xi, |\boldsymbol{y}_0 - \boldsymbol{\beta}_{\theta_{\mathbf{l},\mathbf{i}}}|\}$
    while $f(S^{-1}(r_0, \boldsymbol{\theta}_{\mathbf{l},\mathbf{i}})) \neq f_2(S^{-1}(r_0, \boldsymbol{\theta}_{\mathbf{l},\mathbf{i}}))$ do
        $r_0 = \min\{r_0 + \xi, |\boldsymbol{y}_0 - \boldsymbol{\beta}_{\theta_{\mathbf{l},\mathbf{i}}}|\}$
    end while
else
    $r_0 = \max\{r_{-1} - \xi, 0\}$
    while $f(S^{-1}(r_0, \boldsymbol{\theta}_{\mathbf{l},\mathbf{i}})) \neq f_1(S^{-1}(r_0, \boldsymbol{\theta}_{\mathbf{l},\mathbf{i}}))$ do
        $r_0 = \max\{r_0 - \xi, 0\}$
    end while
end if
Define $K = \lceil \log_2 (|r_0 - r_{-1}|/\tau) \rceil$
Run bisection $\widetilde{g}_{\mathbf{l},\mathbf{i}} = r_K$ where $|r_K - g(\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}})| \leq |r_0 - r_{-1}|/2^K$
```

the optimal choice, but in most cases it is big enough to cover the discontinuity and also decays to zero as L increases. However, in order to avoid the scenario that both r_{-1} and r_0 are on the same side of the discontinuity, e.g., $r_{-1}, r_0 < g(\boldsymbol{\theta})$, we add two loops in Algorithm 2 to recursively enlarge the length $|r_{-1} - r_0|$ by ξ until the interval $[r_{-1}, r_0]$ covers the value $g(\boldsymbol{\theta})$.

5.3. ε-Complexity Analyses.

We provide error estimates and ε-complexity analyses of the proposed hyperspherical SG method for approximating the discontinuity hypersurface γ, i.e., the function $g(\boldsymbol{\theta})$. For simplicity, we assume the target function $f(\boldsymbol{y})$ satisfies Assumption A_2. The analyses are carried out in the context of the isotropic SG interpolation, given in (3.3), coupled with a bisection method. The underlying domain Γ is set to $(0,1)^N$. For the sake of notational convenience, we set $N_s = N-1$ in the following derivation and use $\|\cdot\|$ to denote the $L^\infty(\Gamma_s)$ norm.

First, we observe that the total error $e = g(\boldsymbol{\theta}) - \widetilde{g}_L(\boldsymbol{\theta})$ can be decomposed as

$$e = g(\boldsymbol{\theta}) - \widetilde{g}_L(\boldsymbol{\theta}) = \underbrace{g(\boldsymbol{\theta}) - g_L(\boldsymbol{\theta})}_{e_1} + \underbrace{g_L(\boldsymbol{\theta}) - \widetilde{g}_L(\boldsymbol{\theta})}_{e_2}, \tag{5.3}$$

where $g_L(\boldsymbol{\theta})$ is the isotropic SG approximation of the exact target function $g(\boldsymbol{\theta})$. Here, we study a specific type of discontinuity hypersurface satisfying the following assumption.

Assumption 5.2. Under Assumption 4.1, the transformed function $g(\boldsymbol{\theta})$ characterizing γ is in the space

$$X^q(\Gamma_s) := \big\{g : \Gamma_s \to [0,+\infty) \mid D^{\boldsymbol{\alpha}}(g) \in L^\infty(\Gamma_s), |\boldsymbol{\alpha}|_\infty \le q\big\},$$

where $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_{N_s})$, $|\boldsymbol{\alpha}|_\infty = \max_{1\le n\le N_s} \alpha_n$, and$D^{\boldsymbol{\alpha}}(g) = \partial^{|\alpha|_1} g / \partial\theta_1^{\alpha_1} \cdots \partial\theta_{N_s}^{\alpha_{N_s}}$.

An estimate for e is given in the following lemma.

Proposition 5.3. *Under Assumptions* 4.1 *and* 5.2, *if the transformed function* $g(\boldsymbol{\theta})$ *is in the space* $X^2(\Gamma_s)$, *then the error* $e = e_1 + e_2$ *in* (5.3) *can be bounded by*

$$\|e\| \le C_{\text{sg}} 2^{-2L} \sum_{k=0}^{N_s-1} \binom{L+N_s-1}{k} + 2^{N_s}\binom{L+N_s}{N_s}\tau, \tag{5.4}$$

where τ is the tolerance for the bisection method. The constant C_{sg} is independent of the level L and the dimension N_s.

Proof. The proof is given in Appendix A.1. □

Now, we analyze the cost of constructing $\widetilde{g}_L(\boldsymbol{\theta})$ with the prescribed error $\varepsilon > 0$. According to the error estimate in Proposition 5.3, a sufficient condition for $\|e\| = \|g(\boldsymbol{\theta}) - \widetilde{g}_L(\boldsymbol{\theta})\| \le \varepsilon$ is that

$$\|e_1\| \le C_{\text{sg}} 2^{-2L} \sum_{k=0}^{N_s-1} \binom{L+N_s-1}{k} \le \frac{\varepsilon}{2} \tag{5.5}$$

and

$$\|e_2\| \le 2^{N_s}\binom{L+N_s}{N_s}\tau \le \frac{\varepsilon}{2}. \tag{5.6}$$

Let $\mathcal{C}_{\min}$ denote the *minimum cost*, i.e., the minimum number of function evaluations, needed to satisfy (5.5) and (5.6). The goal is to determine an upper bound for $\mathcal{C}_{\min}$.

Note that, for fixed dimension N and level L, the total cost $\mathcal{C}_{\text{total}}$ is determined by solving the inequality (5.6). The larger L is, the smaller τ is, which means that when using the bisection method, a greater number of function evaluations are needed to achieve the accuracy τ. Therefore, the estimation of $\mathcal{C}_{\min}$ has two steps. Given N and ε, we first determine upper bounds for the minimum L needed to achieve (5.5); then we substitute the obtained value into (5.6) to get an upper bound for $\mathcal{C}_{\min}$.

To perform the first step, we need to estimate the numbers of degrees of freedom of $\mathcal{H}_l$ and $\Delta\mathcal{H}_l$ for $l \leq L$, denoted by $|\mathcal{H}_L|$ and $|\Delta\mathcal{H}_l|$, respectively. The estimation of $|\mathcal{H}_L|$ has been studied in [6, 24], but the estimate in [24] is not sufficiently sharp and the estimate in [6] has no results related to $|\Delta\mathcal{H}_l|$. In the following lemma, we provide estimates for $|\Delta\mathcal{H}_l|$ which directly leads to an estimate of $|\mathcal{H}_L|$.

Lemma 5.4. *The dimensions of the subspaces $\Delta\mathcal{H}_l$ and $\mathcal{H}_L$ for $N_s \geq 2$, i.e., the numbers of grid points in $\Delta\mathcal{H}_l(\Gamma_s)$ and $\mathcal{H}_L(\Gamma_s)$, respectively, are bounded by*

$$|\Delta\mathcal{H}_l| \leq 2^l \binom{l+N_s-1}{N_s-1} \leq 2^l \left(\frac{l+N_s-1}{N_s-1}\right)^{N_s-1} \mathrm{e}^{N_s-1}$$

for $0 \leq l \leq L$ and, correspondingly,

$$|\mathcal{H}_L| \leq 2^{L+1} \binom{L+N_s-1}{N_s-1} \leq 2^{L+1} \left(\frac{L+N_s-1}{N_s-1}\right)^{N_s-1} \mathrm{e}^{N_s-1}.$$

Proof. The proof is given in Appendix A.2. □

Similar to the analyses in [31], we solve the inequality (5.5) to obtain an upper bound for L such that the error of the isotropic sparse grid interpolant $g_L(\boldsymbol{\theta})$ is smaller than the prescribed accuracy $\frac{\varepsilon}{2}$.

Lemma 5.5. *For $\varepsilon < 2C_{\text{sg}}$ in (5.5), the accuracy $\|e_1\| \leq \frac{\varepsilon}{2}$ can be achieved with a minimum level L such that*

$$L \leq \lceil L_k \rceil = \left\lceil \frac{t_k N_s}{2\ln 2} \right\rceil \quad \textit{with} \quad h = \frac{2e}{\ln 2}\left(\frac{2C_{\text{sg}}}{\varepsilon}\right)^{\frac{1}{N_s}},$$

where $\{t_k\}_{k=0}^{\infty}$ is a monotonically decreasing sequence defined by

$$t_k = \ln(t_{k-1}h) \quad \textit{with} \quad t_0 = \frac{\mathrm{e}}{\mathrm{e}-1}\ln h.$$

Proof. The proof is given in Appendix A.3. □

Corollary 5.6. *Under Lemma 5.5, for $k \in \mathbb{N}$, we have*

$$\binom{L_k+N_s}{N_s} \leq \frac{\varepsilon}{2N_sC_{\text{sg}}} \cdot 2^{2L_k}. \tag{5.7}$$

This is an immediate result following from substituting (A.4) into (A.3).

Now we derive an upper bound for $\mathcal{C}_{\min}$ in the context of the isotropic hyperspherical SG method using a linear basis without acceleration

Theorem 5.7. *Under Lemmas 5.4 and 5.5, the minimum total cost $\mathcal{C}_{\min}$ for building the isotropic SG approximation to $g(\boldsymbol{\theta})$ with accuracy $\|g(\boldsymbol{\theta}) - \widetilde{g}_L(\boldsymbol{\theta})\| \leq \varepsilon$ based on Algorithm 1 satisfies the estimate*

$$\mathcal{C}_{\min} \leq \frac{\alpha_1}{N_s}\left\{\alpha_2 + \alpha_3\frac{\log_2\left(\frac{2C_{\text{sg}}}{\varepsilon}\right)}{N_s}\right\}^{\alpha_4 N_s}\left(\frac{2C_{\text{sg}}}{\varepsilon}\right)^{\alpha_5}\left\{\alpha_6\log_2\left(\frac{2C_{\text{sg}}}{\varepsilon}\right) + \alpha_7 N_s + \alpha_8\right\},$$

where $C_{\rm sg}$ is defined as in (5.5) *and the constants $\alpha_1, \dots, \alpha_8$ are defined by*

$$\begin{aligned}&\alpha_1 = 2, \quad \alpha_2 = \frac{2\mathrm{e}^2}{(\mathrm{e}-1)}\log_2\left(\frac{2\mathrm{e}}{\ln 2}\right), \quad \alpha_3 = \frac{2\mathrm{e}^2}{(\mathrm{e}-1)}, \quad \alpha_4 = \frac{3}{2}, \\ &\alpha_5 = \frac{1}{2}, \quad \alpha_6 = \frac{\mathrm{e}}{\mathrm{e}-1}, \quad \alpha_7 = \frac{\mathrm{e}}{\mathrm{e}-1}\log_2\left(\frac{2\mathrm{e}}{\ln 2}\right) + 1, \quad \alpha_8 = 2 - \log_2(C_{\rm sg}).\end{aligned} \tag{5.8}$$

Proof. The proof is given in Appendix A.4. □

Next, we analyze the computational cost of the accelerated Algorithm 1 by exploiting Algorithm 2. Unlike the unaccelerated Algorithm 1 for which the length τ_0 of the initial search interval is set to be of the same scale as the domain Γ, in Algorithm 2, for each new added sparse grid point $\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}}$ with $L = |\mathbf{l}| \geq 1$, first the desired function value $g(\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}})$ is predicted by the level $L-1$ interpolant $\widetilde{g}_{L-1}(\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}})$, and then this prediction is used as one endpoint of the initial search interval in the bisection simulation, i.e., $r_{-1} = \widetilde{g}_{L-1}(\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}})$. For simplicity, the other endpoint is defined by the upper bound of the error of the prediction, i.e., $|g(\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}}) - \widetilde{g}_{L-1}(\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}})|$. In this case, the interval $[r_{-1}, r_0]$ will include the exact function value $g(\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}})$. This is slightly different from the strategy used in Algorithm 2, in which the local error indicator, i.e., the surplus, is used because the upper bound of $|g(\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}}) - \widetilde{g}_{L-1}(\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}})|$ is only known up to a constant. In the following derivation, the error bound given in (5.4) is still valid, but, at sparse grid points $\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}}$ for $|\mathbf{l}| = L$, we can obtain a sharper bound for the error of $\widetilde{g}_{L-1}(\boldsymbol{\theta})$. The result is provided in the following lemma.

LEMMA 5.8. *If the transformed function $g(\boldsymbol{\theta})$ is in the space $X^2(\Gamma_s)$, then at each sparse grid point $\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}}$ with $L = |\mathbf{l}| \geq 1$ and $\mathbf{i} \in B_{\mathbf{l}}$ defined in* (3.4)*, the error $g(\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}}) - \widetilde{g}_{L-1}(\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}})$ satisfies the estimate*

$$\left|g(\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}}) - \widetilde{g}_{L-1}(\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}})\right| \leq C_{\rm surp} 2^{-2L} + 2^{N_s}\tau,$$

where $C_{\rm surp}$ is independent of L and τ is the tolerance of the bisection algorithm.

Proof. The proof is given in Appendix A.5. □

Finally, the upper bound of $\mathcal{C}_{\min}$ in the context of using the SG method with acceleration is in the following theorem.

THEOREM 5.9. *Under Lemmas* 5.4, 5.5, *and* 5.8, *the minimum total cost $\mathcal{C}_{\min}$ incurred in building the isotropic sparse grid approximation to $g(\boldsymbol{\theta})$ with accuracy $\|g(\boldsymbol{\theta}) - \widetilde{g}_L(\boldsymbol{\theta})\| \leq \varepsilon$ using the accelerated SG method satisfies the estimate*

$$\mathcal{C}_{\min} \leq \alpha_1 \left[\alpha_2 + \alpha_3 \frac{\log_2\left(\frac{2C_{\rm sg}}{\varepsilon}\right)}{N_s}\right]^{\alpha_4 N_s} \left(\frac{2C_{\rm sg}}{\varepsilon}\right)^{\alpha_5} \left[2N_s - \log_2(N_s) + \alpha_9\right],$$

where $C_{\rm sg}$ is the constant in (5.5)*, the constants $\alpha_1, \dots, \alpha_5$ are defined as in Theorem* 5.7*, and α_9 is defined by*

$$\alpha_9 = \log_2\left(\frac{C_{\rm surp}}{C_{\rm sg}}\right) + 2.$$

Proof. The proof is given in Appendix A.6. □

Remark 5.10. Theorems 5.7 and 5.9 tell us that the total cost of the hyperspherical SG method is mainly determined by the number of sparse grid points. Asymptotically, the growth rate of $|\mathcal{H}_L|$ is characterized by the constants α_4 and α_5, and the cost due to inaccurate initial searching interval is of order $\log_2(1/\varepsilon)$. According to the analyses in [6, 31], the growth rate can be reduced when using high-order hierarchical polynomial bases. In general, with a pth-order hierarchical basis, we have $\alpha_4 = (p+2)/(p+1)$ and $\alpha_5 = 1/(p+1)$. Note that the use of an acceleration technique with accurate initial guesses will reduce the total cost by a factor of $\log_2(1/\varepsilon)$ asymptotically, which will be demonstrated in the following section.

6. Hyperspherical Sparse Approximation with Random Sampling.

In this section we extend the capabilities of our method by incorporating sparse approximations using random sampling. Specifically, we are interested in the following three types of approximations:

(a) radial basis function (RBF) interpolation;

(b) discrete least squares (DLS) projection; and

(c) compressed sensing (CS) approximation.

As opposed to the SG interpolation that has to be constructed in hypercubes, the above approaches typically utilize random sampling and are more flexible when dealing with domains having irregular geometries. In fact, nonhypercubic domains appear very often in discontinuity detection, e.g., when the discontinuity surface intersects with the boundary of the search domain, as shown in Figure 4 in section 7. In this setting, we only consider the scenarios (S_2) and (S_3); i.e., the discontinuous function $f(\boldsymbol{y})$ is assumed to feature the characteristic property. In each of the three methods listed above, the aim is to construct an approximation to the discontinuity surface γ of the form

$$g_M(\boldsymbol{\theta}) := \sum_{m=1}^{M} w_m \psi_m(\boldsymbol{\theta}) \quad \text{for } \boldsymbol{\theta} \in \Gamma_\gamma := \{\boldsymbol{\theta} \in \Gamma_s \,|\, (g(\boldsymbol{\theta}), \boldsymbol{\theta}) \in \gamma\} \tag{6.1}$$

based on a set of J random samples $\{\boldsymbol{\theta}_j, j = 1, \ldots, J\} \subset \Gamma_\gamma$ and the associated function values $\{\widetilde{g}(\boldsymbol{\theta}_j), j = 1, \ldots, J\}$. The three methods (a)–(c) differ only in the choice of the basis functions $\{\psi_m\}_{m=1}^M$ and the associated approximation space $\mathcal{V}_M := \operatorname{span}\{\psi_m, m = 1, \ldots, M\}$. The construction of $g_M(\boldsymbol{\theta})$ eventually relies on the solution of a linear system, denoted by

$$\boldsymbol{\Psi} \boldsymbol{w} = \boldsymbol{g}, \tag{6.2}$$

where $\boldsymbol{w} := (w_1, \ldots, w_M)^\top$ is the coefficient vector of $g_M(\boldsymbol{\theta})$, $\boldsymbol{\Psi}$ is a $J \times M$ matrix with $\boldsymbol{\Psi}_{jm} := \psi_m(\boldsymbol{\theta}_j)$ for $j = 1, \ldots, J$, $m = 1, \ldots, M$, and $\boldsymbol{g} := (\widetilde{g}(\boldsymbol{\theta}_1), \ldots, \widetilde{g}(\boldsymbol{\theta}_J))^\top$ is the vector of data. Again, the three methods also differ in the size of the system matrix $\boldsymbol{\Psi}$, which requires different approaches for solving the corresponding linear system. In the following three paragraphs, we explain how the methods (a)–(c) are integrated into our hyperspherical framework for discontinuity detection.

Radial basis function interpolation. RBF methods possess several attractive features such as a certain insensitivity to dimension and ease of implementation. Thus, they have recently gained significant attention and have been used in many applications such as scattered data interpolation [4]. Let $\mathbb{R}^+ := \{r \in \mathbb{R}, r \geq 0\}$ denote the nonnegative half line and $\phi : \mathbb{R} \mapsto \mathbb{R}^+$ denote a continuous function with $\phi(0) \geq 0$. An RBF is a function of the form $\phi(\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_2)$, where $\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_2$ is the Euclidean distance between $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$. Then the basis function ψ_m in (6.1) can be defined as $\psi_m(\boldsymbol{\theta}) := \phi(\|\boldsymbol{\theta} - \boldsymbol{\theta}_m\|_2)$. However, when using global RBF interpolation, the system matrix $\boldsymbol{\Psi}$ in (6.2) is usually full and severely ill-conditioned, especially if certain popular RBFs such as Gaussians or inverse multiquadrics are used. To convert the ill-conditioned matrix to a well-conditioned matrix, compactly supported RBFs can be used instead of global RBFs. The most popular family of compactly supported RBFs is the Wendland function [32]. Specifically, the Wendland function starts with a truncated power function $\phi_\ell(r) := (1-r)_+^\ell$ that is strictly positive definite in $\mathbb{R}^d$ for $\ell \geq \lfloor \frac{d}{2} \rfloor + 1$. The kth order Wendland function, denote by $\phi_{d,k}(r)$, can be constructed by the recursive procedure

$$\phi_{d,k}(r) := \int_r^\infty t\phi_{d,k-1}(t)dt, \quad \text{for } k = 1, 2, 3, \ldots, \tag{6.3}$$

where $\phi_{d,0}(r) := \phi_\ell(r)$ for $\ell = \lfloor d/2 \rfloor + k + 1$ and the kth-order function $\phi_{d,k}$ is in the function space $C^{2k}(\mathbb{R})$. In the setting of interpolation, we have $M = J$ in (6.1) and the basis functions $\psi_m(\boldsymbol{\theta})$ defined by

$$\psi_m(\boldsymbol{\theta}) := \phi_{d,k}(\|\boldsymbol{\theta} - \boldsymbol{\theta}_m\|_2), \tag{6.4}$$

where $d = N_s = N - 1$ and $\{\boldsymbol{\theta}_m, m = 1, \ldots, M\}$ is the set of M samples points in Γ_γ.

Discrete least squares projection. In this case, we denote by $L^2_\varrho(\Gamma_\gamma)$ the Hilbert space of real-valued square integrable functions with respect to a finite measure ϱ and denote by $\langle \cdot, \cdot \rangle_\varrho$ and $\| \cdot \|_{L^2_\varrho}$ the associated inner product and norm, i.e.,

$$\langle u, v \rangle_\varrho := \int_{\Gamma_\gamma} u(\boldsymbol{\theta}) v(\boldsymbol{\theta}) d\varrho(\boldsymbol{\theta}) \quad \text{and} \quad \|u\|_{L^2_\varrho} := \sqrt{\langle u, u \rangle_\varrho} \;\; \forall u \in L^2_\varrho(\Gamma_\gamma).$$

We aim to approximate the function $g(\boldsymbol{\theta})$ in a finite-dimensional subspace $\mathcal{V}_M \subset L^2_\varrho(\Gamma_\gamma)$ with dimension $\#(\mathcal{V}_M) = M$. We assume that the functions belonging to $\mathcal{V}_M$ are defined everywhere over Γ_γ. In this case, $\{\psi_m\}_{m=1}^M$ represent an orthonormal basis of $\mathcal{V}_M$ with respect to the above inner product. The best approximation in the least squares sense can be obtained by defining the coefficients in (6.1) as $w_m := \langle g, \psi_m \rangle_\varrho$. When the function $g(\boldsymbol{\theta})$ is unknown and $\{\widetilde{g}(\boldsymbol{\theta}_j)\}_{j=1}^J$ are approximations of $g(\boldsymbol{\theta})$ at a set of points $\{\boldsymbol{\theta}_j\}_{j=1}^J$ that are i.i.d. random samples distributed according to ϱ, the approximation in (6.1) can be constructed in the discrete least squares (DLS) sense, i.e.,

$$g_M := \underset{v \in \mathcal{V}_M}{\arg\min} \sum_{j=1}^J |\widetilde{g}(\boldsymbol{\theta}_j) - v(\boldsymbol{\theta}_j)|^2.$$

The coefficient vector $\boldsymbol{w} \in \mathbb{R}^M$ of g_M in (6.1) is the solution to the linear system

$$\boldsymbol{\Psi}^\top \boldsymbol{\Psi} \boldsymbol{w} = \boldsymbol{\Psi}^\top \boldsymbol{g}, \tag{6.5}$$

where $\boldsymbol{\Psi}^\top \boldsymbol{\Psi}$ is the $M \times M$ Gram matrix with $(\boldsymbol{\Psi}^\top \boldsymbol{\Psi})_{ii'} := \sum_{j=1}^J \psi_i(\boldsymbol{\theta}_j)\psi_{i'}(\boldsymbol{\theta}_j)/J$. When $\mathcal{V}_M$ is an orthogonal polynomial space defined on a lower set with cardinality M, the number of samples J needs to scale like M^2 up to a logarithmic factor, so as to guarantee the stability of the least squares problem in (6.5) [9, 23]. According to the definition of $g(\boldsymbol{\theta})$, when $\Gamma_\gamma = \Gamma_s$, i.e., $\partial\Gamma_1 \cap \partial\Gamma \neq \emptyset$, we let $\varrho = 1$ in Γ_s and $\mathcal{V}_M$ be the total degree polynomial space with cardinality M expanded by Legendre polynomials. However, when $\Gamma_\gamma \neq \Gamma_s$, i.e., $\partial\Gamma_1 \cap \partial\Gamma = \emptyset$, identifying the geometry of Γ_γ in Γ_s is almost as difficult as detecting γ in Γ. Thus, instead of defining a new orthogonal polynomial basis in Γ_γ, we still use the Legendre basis; however, such a basis is no longer orthogonal under the uniform measure in Γ_γ. The loss of orthogonality may lead to instability of the system in (6.5), because the expectation of the Gram matrix $\mathbf{\Psi}^\top\mathbf{\Psi}$ does not converge to the identity matrix as $J \to \infty$. Hence, in this case, we utilize the QR factorization [29] to directly solve the overdetermined system in (6.2) and thus avoid the possible numerical instability.

Compressed sensing approximation. As opposed to the DLS method, the CS approximation starts with an enriched polynomial space $\mathcal{V}_M$, i.e., setting $M > J$, so that the linear system in (6.1) is *underdetermined*. The main idea of compressed sensing is to identify the sparse structure of the polynomial representation of $g(\boldsymbol{\theta})$ in the enriched space $\mathcal{V}_M$ using, e.g., ℓ_1 minimization. Specifically, the coefficient vector $\boldsymbol{w} \in \mathbb{R}^M$ is identified by solving the following optimization problem:

$$(6.6) \qquad \widehat{\boldsymbol{w}} := \underset{\boldsymbol{w}\in\mathbb{R}^M}{\arg\min} \|\boldsymbol{w}\|_1 \text{ such that } \|\mathbf{\Psi}\boldsymbol{w} - \boldsymbol{g}\|_2 \leq \delta,$$

where $\delta > 0$ and $\|\boldsymbol{w}\|_1 := \sum_{m=1}^M |c_m|$. We refer the reader to [11, 30] and the references therein for a comprehensive overview of CS methods. One of the main topics in polynomial approximation via compressed sensing is estimating sharp lower bounds for the number J to recover the best s-term approximation in $\mathcal{V}_M$. Such sparse recovery is strongly tied to the concept of the restricted isometry property (RIP). In the case of using standard RIP [8], M has to scale like $M \geq C\Theta^2 s\log^2(s)\log(J)$, where $\Theta = 3^{(N_s)/2}$ for the preconditioned Legendre basis and $\Theta = 2^{(N_s)/2}$ for the Chebyshev basis. In the context of recovering the best *lower* s-term approximation, which is the best among all approximations supported on lower index sets of cardinality s, a lower RIP was developed in [8] leading to an improved bound of M, i.e.,

$$M \geq CK(s)\log^2(s)(\log(s) + N_s),$$

where $K(s) = s^2$ for the Legendre basis and $K(s) = s^{\log 3/\log 2}$ in the Chebyshev basis. However, similar to the DSL case, we will lose the orthogonality of the basis when γ intersects with the boundary $\partial\Gamma$, i.e., $\partial\Gamma_1 \cap \partial\Gamma \neq \emptyset$. In this case, the theoretical results discussed above may no longer hold, but all the computational tools for ℓ_1 minimization can still be used. The last issue to be resolved is how to generate J uniformly distributed samples in Γ_γ. This can be accomplished by exploiting the characteristic property of $f(\boldsymbol{y})$.

We summarize construction of $g_M(\boldsymbol{\theta})$ in (6.1) in Algorithm 3.

Remark 6.1. As was the case for Algorithm 1, we utilize the characteristic property in Algorithm 3 to generate uniform samples $\{\boldsymbol{\theta}_j\}_{j=1}^J$ just in the subdomain Γ_γ, in the case that $\Gamma_\gamma \neq \Gamma_s$. However, because each random sample is drawn from the uniform distribution in the domain Γ_s, some function evaluations will be wasteful if the samples fall outside of the domain Γ_γ. Statistically, the number of wasteful function evaluations is of $\mathcal{O}(J\frac{\text{Volume}(\Gamma_s\backslash\Gamma_\gamma)}{\text{Volume}(\Gamma_\gamma)})$.

7. Numerical Examples.

In this section, we use four discontinuity detection problems to illustrate the performance of the proposed methods. The first example is used to test the hyperspherical SG interpolation developed in section 5 for approximating discontinuities of functions with the characteristic property. In the second example, a generic discontinuous function without the characteristic property is considered and the importance of the choice of the origin $\boldsymbol{y}_0$ is demonstrated. The third example is an application of the hyperspherical SG method in computing the probability of an event that depends on the solution of a partial differential equation with

Algorithm 3. The hyperspherical sparse approximation with random sampling.

```
Initialize $N$, $M$, $J$, $\tau$, $\boldsymbol{y}_0$
$j = 0$
while $j < J$ do
    Draw a sample $\boldsymbol{\theta}$ from the uniform distribution in $\Gamma_s$
    Search $\boldsymbol{\beta}_{\boldsymbol{\theta}} = (\beta_{\boldsymbol{\theta},1}, \ldots, \beta_{\boldsymbol{\theta},N}) \in \Gamma$
    if $f(\boldsymbol{\beta}_{\boldsymbol{\theta}}) = f_2(\boldsymbol{\beta}_{\boldsymbol{\theta}})$ then
        $j = j + 1$
        $\boldsymbol{\theta}_j = \boldsymbol{\theta}$
        $K = \lceil \log_2 \left( |\boldsymbol{y}_0 - \boldsymbol{\beta}_{\boldsymbol{\theta}_j}|/\tau \right) \rceil$
        Run bisection to get $z_j = r_K$, where $|r_K - g(\boldsymbol{\theta}_j)| \leq |\boldsymbol{\beta}_{\boldsymbol{\theta}_j} - \boldsymbol{y}_0|/2^K$
    end if
end while
Formulate the matrix $\boldsymbol{\Psi}$ and solve the system $\boldsymbol{\Psi w} = \boldsymbol{g}$
```

random inputs. The last example is used to test the hyperspherical approaches based on the RBF, DLS, and CS methods developed in section 6.

Example 7.1. Consider the two characteristic functions in $\mathbb{R}^N$,

$$F_1(\boldsymbol{y}) = \begin{cases} 1 & \text{if } \displaystyle\sum_{n=1}^{N} y_n^2 \leq 1, \\ 0 & \text{otherwise,} \end{cases} \tag{7.1}$$

$$F_2(\boldsymbol{y}) = \begin{cases} 1 & \text{if } |y_3 - y_1| \leq 0.5 \text{ for } \boldsymbol{y} \in [0,1]^N, \\ 0 & \text{otherwise,} \end{cases} \tag{7.2}$$

where the characteristic domains of $F_1(\boldsymbol{y})$ and $F_2(\boldsymbol{y})$ are a unit hypersphere and a slab in the unit hypercube, respectively. The linear hierarchical basis is used for building the SG interpolant, and the bisection method is used to estimate the value of the transformed function $g(\boldsymbol{\theta})$ at the sparse grid points.

First, to illustrate the distribution of the sparse grid points generated by our method, we set $N = 3$ and plot in Figures 3 and 4 the discontinuity surface γ, the surface of $g(\boldsymbol{\theta})$, and the sparse grid points for $F_1(\boldsymbol{y})$ and $F_2(\boldsymbol{y})$, respectively. By comparing Figures 2(b) and 3, we can see the advantage of the hyperspherical SG method. The resulting sparse grid contains only 160 points to achieve the desired accuracy, whereas the classic SG method requires 36,093 grid points. Instead of directly approximating the discontinuous function $F_1(\boldsymbol{y})$, we approximate the transformed surface shown in Figure 3(b), where its smoothness retains the sparsity of the resulting grid. In Figure 4, we see that the surface γ is only a part of the boundary $\partial\Gamma_1$, but $\overline{\Gamma}_1$ is a closed subdomain in $\overline{\Gamma}$. There are a total of 1,120 sparse grid points on $\partial\Gamma_1$ with only 349 points on γ and 771 points on $\partial\Gamma_1$. According to the discussion in section 4.2, at the sparse grid points placed on $\partial\Gamma_1 \backslash \gamma$, there is no need to run the bisection algorithm to evaluate $g(\boldsymbol{\theta})$ at the 771 points on $\partial\Gamma_1$, so that a significant amount of computational effort is saved. In Figure 4, we observe that $g(\boldsymbol{\theta})$ is not differentiable at the edges and vertices of the characteristic domain Γ_1, so that the SG approximation does mesh refinement around these regions. Although the lack of

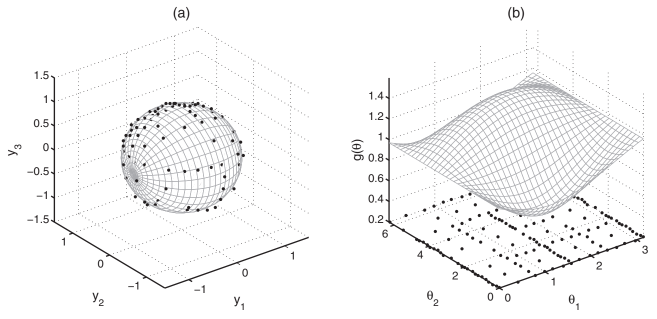

Fig. 3 (a) *The discontinuity surface* γ *with sparse grid points;* (b) *the transformed surface* $g(\boldsymbol{\theta})$ *in the hyperspherical coordinate system. The parameters for the SG approximation are* $L_{\min} = 4$*,* $L_{\max} = 12$*,* $\alpha = 0.01$*, and* $\boldsymbol{y}_0 = (0.1, 0.2, 0.3)$*; the total number of sparse grid points is* 160*.*

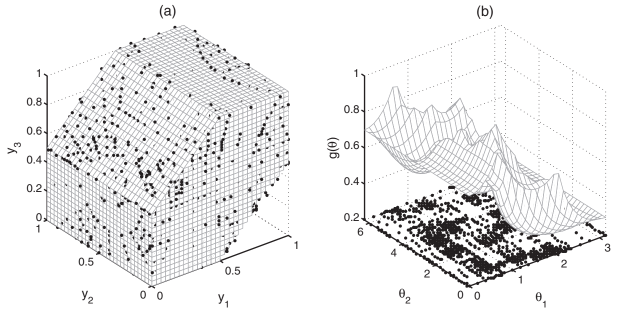

Fig. 4 (a) *The discontinuity surface* γ *with sparse grid points;* (b) *the transformed surface* $g(\boldsymbol{\theta})$ *in the hyperspherical coordinate system. The parameters for the SG approximation are* $L_{\min} = 4$*,* $L_{\max} = 12$*,* $\alpha = 0.01$*, and* $\boldsymbol{y}_0 = (0.3, 0.4, 0.5)$*; the total number of sparse grid points is* 1120*, of which only* 349 *are off the boundary.*

a derivative is not as bad as a jump discontinuity, it may result in a failure of the our method if the volume of such a singularity grows quickly as the dimension N increases. This issue will be considered in future work.

Next, we test the convergence of the hyperspherical SG method in detecting the discontinuity of $F_1(\boldsymbol{y})$ in three cases: isotropic sparse grids and no acceleration, isotropic sparse grids and acceleration, and adaptive sparse grids and acceleration. The origin of the hyperspherical coordinate system is set to $\boldsymbol{y}_0 = (0.2/\sqrt{N}, \ldots, 0.2/\sqrt{N})$ which is 0.2 away from the origin $(0, \ldots, 0)$. In Table 2, we list the computational costs of the three tested cases for $N = 2, 3, 4, 5, 7, 9$, where the interpolation error tested is measured by the metric e_∞ in (2.4). As expected, for high-dimensional discontinuity detection, our approach achieves faster convergence rates than the well-known SG method. Moreover, for the same accuracy, acceleration and adaptivity can provide a significant savings in computational cost.

Example 7.2. Consider the two-dimensional discontinuous function on $[-1.5, 1.5] \times [-1.5, 1.5]$ given by

$$f(\boldsymbol{y}) = \begin{cases} y_1^2 + y_2^2 & \text{if } \sqrt{y_1^2 + y_2^2} \le 1 + \dfrac{1}{4}\cos\left(4\arctan\left(\dfrac{y_2}{y_1}\right)\right), \\ y_1^2 + y_2^2 + \dfrac{1}{2} & \text{otherwise,} \end{cases} \tag{7.3}$$

Table 2 *Computational costs (total number of function evaluations) and savings of the hyperspherical SG method with acceleration and adaptivity for Example* 7.1.

Dim	Error	IsoSG	IsoSG+acceleration		SG+acceleration	
		Cost	Cost	Saving	Cost	Saving
2	1.0e-3	384	274	28.7%	251	34.7%
	1.0e-4	1,603	955	40.4%	832	48.1%
	1.0e-5	7,230	2,968	58.9%	2,737	62.1%
3	1.0e-3	7,046	4,461	36.7%	3,022	57.1%
	1.0e-4	42,541	18,021	57.6%	12,817	69.9%
	1.0e-5	224,978	67,721	69.9%	54,439	75.8%
4	5.0e-2	880	682	22.5%	584	33.6%
	1.0e-3	66,207	38,165	42.4%	26,329	60.2%
	1.0e-4	542,632	241,337	55.5%	161,354	70.3%
5	5.0e-2	5,135	3,645	29.0%	2,082	59.5%
	1.0e-2	23,782	16,558	30.4%	14,694	38.2%
	1.0e-3	383,884	207,862	45.9%	94,148	75.5%
7	1.0e-1	24,757	11,770	52.5%	6,327	74.4%
	5.0e-2	67,671	40,221	40.6%	25,111	62.9%
	5.0e-3	773,113	479,984	37.9%	354,040	54.2%
9	1.0e-1	26,593	14,843	44.2%	6,426	75.8%
	5.0e-2	157,851	80,507	49.0%	58,849	62.7%
	1.0e-2	1,472,441	983,101	33.2%	513,163	65.1%

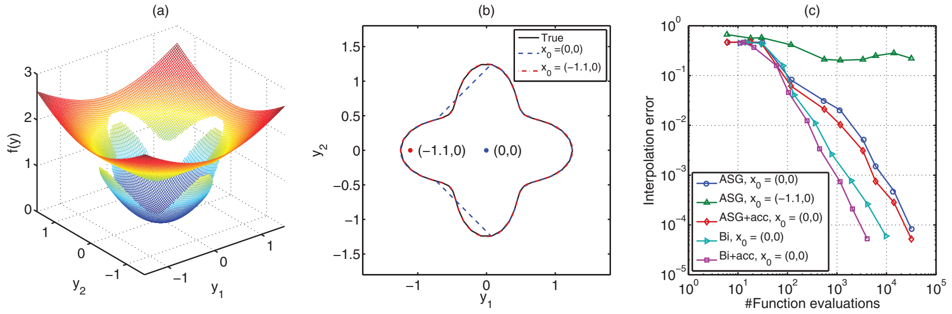

Fig. 5 (a) *The surface of the target function* $f(\boldsymbol{y})$ *in Example* 7.2; (b) *the true discontinuity curve and the approximated curves with the origin of the polar coordinate system being* $(0,0)$ *and* $(-1.1,0)$; (c) *the decay of the interpolation error measured by the metric in* (2.4).

which is plotted in Figure 5(a); the jump discontinuity is along the curve $\sqrt{y_1^2+y_2^2} = 1+\frac{1}{4}\cos\left(4\cdot\arctan\left(y_2/y_1\right)\right)$. The domain Γ_1 is defined as the interior of this curve. Note that Γ_1 is star convex but not convex, so that only a subset of points in Γ_1 can be used as the origin of the hyperspherical coordinate system. We test our approach with the origin being $(0,0)$ and $(-1.1,0)$, and the bisection method is used to approximate the transformed function $g(\boldsymbol{\theta})$. The captured discontinuity curves are plotted in Figure 5(b). It is easy to see that the point $(0,0)$ is qualified to be the origin of the polar system so that the approximate curve captures the discontinuity very well. In contrast, $(-1.1,0)$ does not satisfy Assumption 4.1, and thus the function $f_{\boldsymbol{\theta}}(r)$ has multiple roots along some directions, whereas the bisection algorithm can only find one root. Thus, the discontinuity curve is not captured correctly in a subdomain of

Γ_s, as shown in Figure 5(b). In addition, if we assume the function $f(\boldsymbol{y})$ has no characteristic property, the one-dimensional SG approach discussed in section 4.2.1 has to be used to estimate the value of the transformed function $g(\boldsymbol{\theta})$. For comparison, the decays of the interpolation error measured by the metric e_∞ in (2.4) are plotted in Figure 5(c). When setting the origin $\boldsymbol{y}_0 = (-1.1, 0)$, the error does not decay toward zero because of the violation of Assumption 4.1. When $\boldsymbol{y}_0 = (0, 0)$, we can see that the use of the bisection method based on the characteristic property and the hierarchical acceleration can significantly reduce the number of function evaluations to achieve the prescribed accuracy.

Example 7.3. Consider the two-dimensional steady heat equation with stochastic diffusivity,

$$\text{(7.4)} \qquad \begin{cases} -\nabla \cdot (\kappa(x,\omega)\nabla u(x,\omega)) = h(x) \text{ in } [0,1]^2 \times \Omega, \\ \qquad\qquad\qquad\quad u(x,\omega) = 0 \qquad \text{on } \partial D \times \Omega, \end{cases}$$

with $h(x) = 2000 + \exp(-\frac{(x_1-0.6)^2+(x_2-0.8)^2}{0.05^2})$ and

$$\text{(7.5)} \qquad \begin{aligned} \kappa(x,\omega) = y_1(\omega)^2 + \exp\big[&y_2(\omega)^4 \sin(\pi x_1) + y_3(\omega)^2 \sin(\pi x_2) \\ &+ \cos(\pi x_1) + \cos(\pi x_2)\big], \end{aligned}$$

where $\boldsymbol{y}(\omega) = (y_1(\omega), y_2(\omega), y_3(\omega))$ are independently and identically distributed random variables following the uniform distribution $U([-1,1])$. The quantity of interest is the probability of the event that the integral of the solution $u(x,\omega)$ over D is larger than the threshold value 1.2, i.e.,

$$\text{(7.6)} \qquad \text{QoI} = \int_\Omega \mathcal{X}_{\Gamma_1}(\boldsymbol{y})\rho(\boldsymbol{y})d\boldsymbol{y}, \quad \text{where} \quad \Gamma_1 = \left\{ \boldsymbol{y} \in \mathbb{R}^3 \;\middle|\; \int_D u(x,\boldsymbol{y})dx > 1.2 \right\}$$

is the domain of interest described by the characteristic function $\mathcal{X}_{\Gamma_1}(\boldsymbol{y})$. Note that this example satisfies Assumption A_3 given in section 2, and the implicit function is defined by $G(\boldsymbol{y}) = 1.2 - \int_D u(x,\boldsymbol{y})dx = 0$, which is smooth due to the regularity of the solution u. In this case, we can use more advanced root-finding methods discussed in section 4.2.2, such as the Regula–Falsi method. Here we use the SG interpolation with both acceleration and adaptivity and only compare the performances of different root-finding methods. At each point $\boldsymbol{y} \in \Gamma$, the PDE in (7.4) is solved using the finite element method on a 50×50 mesh in the physical domain. Because we are focusing on improving the accuracy of capturing discontinuities in the stochastic domain, the error from the spatial discretization is ignored, and the finite element projection is treated as the exact solution. An approximation to the exact value of QoI in (7.6) is computed using Monte Carlo sampling. The origin $\boldsymbol{y}_0$ is set to $(0.01, 0.2, 0.05)$. The surface γ and its transformed representation are plotted in Figures 6(a) and 6(b), respectively. For $L_{\min} = 3$ and $\alpha = 0.01$, we end up with a total of 344 sparse grid points in the hyperspherical domain Γ_s; those points are also plotted in Figures 6(a) and 6(b). In Figure 6(c), we show the error decay of our approach using the bisection and Regula–Falsi methods, respectively, where the error is measured by the metric e_{int} in (2.5). We observe that the Regula–Falsi method can provide additional savings in computational costs over the bisection method by taking advantage of the availability and smoothness of $G(\boldsymbol{y})$. Further evidence can been seen in Table 3.

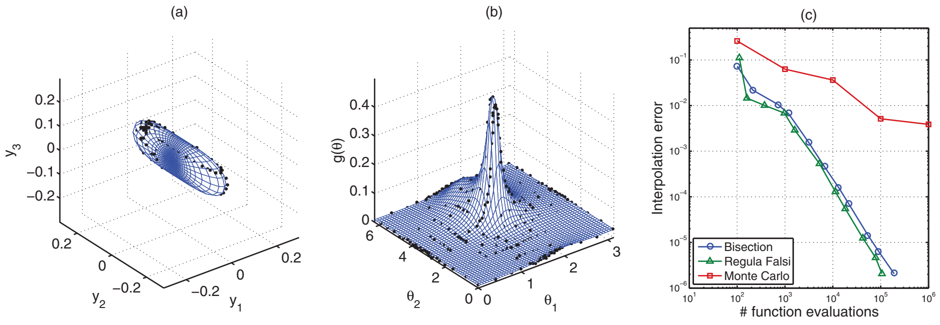

Fig. 6 (a) *The discontinuity surface γ with* 344 *sparse grid points;* (b) *the transformed surface $g(\boldsymbol{\theta})$ in the hyperspherical coordinate system;* (c) *the error decays of the Monte Carlo method and the hyperspherical SG interpolation using the bisection and the Regula–Falsi methods.*

Table 3 *The computational cost (number of function evaluations) of the hyperspherical SG method using the bisection method and the Regula–Falsi method in Example* 7.3.

Error	Bisection	Regula–Falsi	Savings
1.0e-3	4,116	3,381	17.8%
1.0e-4	17,464	13,047	25.3%
1.0e-5	68,299	48,555	28.9%

Example 7.4. Consider the discontinuous function in $\Gamma = [0,1]^N$,

$$f(\boldsymbol{y}) = \begin{cases} 1 & \text{if } (y_1 + 0.3)^2 + \sum_{n=2}^{N} y_n^2 \leq 0.64, \\ 0 & \text{otherwise}, \end{cases} \tag{7.7}$$

where the characteristic domain of $f(\boldsymbol{y})$ is the intersection of the hypercube $[0,1]^N$ and the ball centered at $(-0.3, 0, \ldots, 0)$ with radius 0.8. The discontinuity surface γ and the transformed surface $g(\boldsymbol{\theta})$ are plotted in Figures 7(a) and 7(b), respectively. It is easy to see from Figure 4 that the transformed function $g(\boldsymbol{\theta})$ may not be differentiable on the manifold where $f(\boldsymbol{y})$ and $\partial\Gamma$ intersect. In the case of using the SG interpolation in section 5, an adaptivity strategy is needed to refine the sparse grid near the intersection. However, in practice, we are interested in approximating γ without worrying about the surface $\partial\Gamma \cap \partial\Gamma_1$. Therefore, we would like to utilize the sparse approximation approaches with random sampling developed in section 6. We apply Algorithm 3 to the target function $f(\boldsymbol{y})$ for dimension $N = 3, 4, 5$, and 10; the origin of the hyperspherical system is set to $\boldsymbol{y}_0 = (0.3/\sqrt{N}, \ldots, 0.3/\sqrt{N})$. For the RBF method, we use the 3rd-order Wendland function given in (6.3) to construct interpolants. The DLS and CS approximations are built using the Legendre polynomial basis; QR factorization and the SPGL1 package [3] are used to solve the least squares and ℓ_1 minimization problems, respectively. We show, in Figures 8(a), 9(a), and 10(a) the exact surface $g(\boldsymbol{\theta})$ and the approximate surfaces in three dimensions obtained by the RBF, DLS, and CS methods, respectively. As expected, the approximate surfaces only fit the exact discontinuity surface in the subdomain Γ_γ, as all the random samples (red dots) are allocated in Γ_γ. Figures 8(b), 9(b), and 10(b) show the error

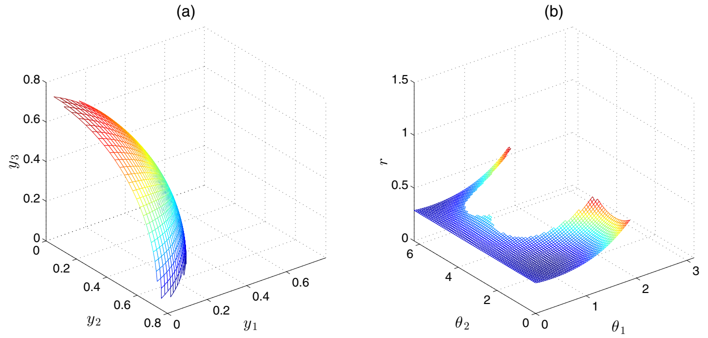

Fig. 7 (a) *The discontinuity surface γ in the Cartesian system;* (b) *the transformed surface $g(\boldsymbol{\theta})$ in the hyperspherical system.*

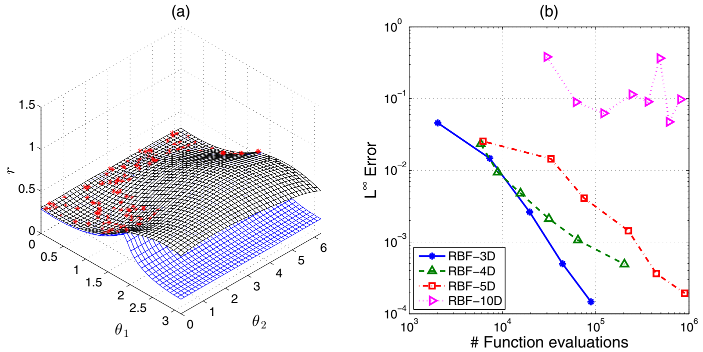

Fig. 8 (a) *The exact surface of $g(\boldsymbol{\theta})$ (blue), the RBF approximate surface (black), and the first* 100 *random data points on γ (red);* (b) *the error decay of the RBF interpolation, measured in the L^∞ norm.*

versus the number of evaluations of $f(\boldsymbol{y})$ for the three methods, where the error is measured by the norm in (2.4) based on 1,000 random samples in Γ_γ. Again, all three methods are effective in capturing γ in high-dimensional spaces. As expected, the RBF interpolation is more sensitive to the dimension than the DLS and CS methods because of the use of compactly supported local bases. By comparing the CS and the DLS methods, we can see that the CS method is more efficient than the DLS method because the CS method can effectively recover a sparse approximation from an underdetermined system. However, such a comparison does not take into account the cost of ℓ_1 minimization. This makes sense in the setting that the evaluation of $f(\boldsymbol{y})$ is so expensive that the cost of evaluating f dominates the total cost.

8. Concluding Remarks.

In this paper, we proposed a comprehensive methodology for detecting an $(N-1)$ discontinuity hypersurface of function defined in an N-dimensional bounded domain. This approach removes the continuity assumption required by adaptive sparse grid methods, thus enabling high-dimensional approximations. Both theoretical and numerical results demonstrate the significant improvements yielded by our approach when compared to existing techniques. Moreover, this technique is not restricted to extensions of sparse grid methods; it is also combined

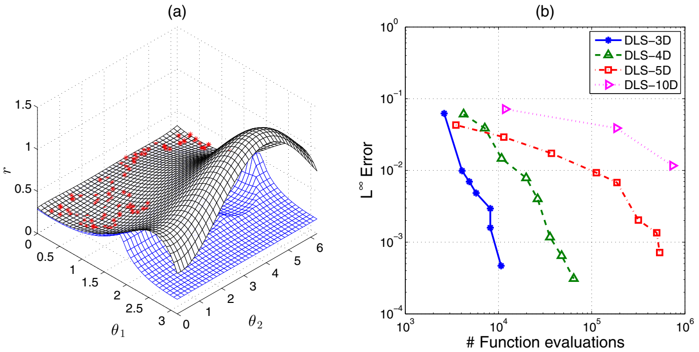

Fig. 9 (a) *The exact surface of* $g(\boldsymbol{\theta})$ *(blue), the DLS approximate surface (black), and the first* 100 *random data points on* γ *(red);* (b) *the error decay of the DLS interpolation, measured in the* L^∞ *norm.*

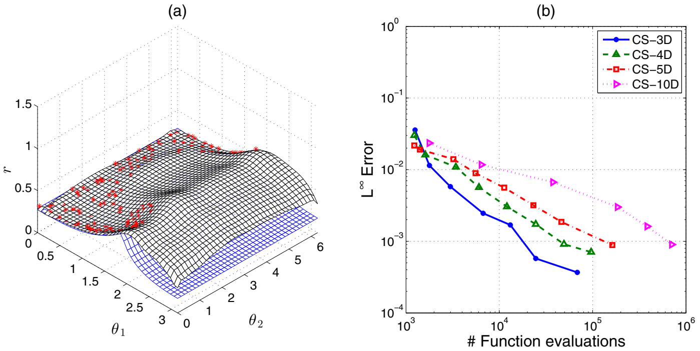

Fig. 10 (a) *The exact surface of* $g(\boldsymbol{\theta})$ *(blue), the CS approximate surface (black), and the first* 100 *random data points on* γ *(red);* (b) *the error decay of the CS interpolation, measured in the* L^∞ *norm.*

with any other numerical method for high-dimensional approximation such as radial basis function interpolation, discrete least squares projection, and compressed sensing approximation. However, we are only able to detect a single hypersurface at a time, determined by the origin of the hyperspherical system. More importantly, this choice of origin is also critical to the performance of our approach, as it dictates the shape—and therefore the steepness of the gradient—of the transformed hypersurface. Finally, our technique relies on the fact the hypersurface satisfies the star convexity assumption, and thus for general nonconvex hypersurfaces the proposed approach may fail to converge. Our future efforts will focus on relaxing this assumption to include general geometries of the domain of interest, so that problems with more complicated hypersurface geometries can be treated.

Appendix A.

A.1. The Proof of Proposition 5.3. According to the analyses in [6], for $g(\boldsymbol{\theta}) \in X^2(\Gamma_s)$, e_1 is the error arising from the linear isotropic SG interpolation which is bounded by

$$\|e_1\| \leq C_{\text{sg}} 2^{-2L} \sum_{k=0}^{N_s-1} \binom{L+N_s-1}{k}, \tag{A.1}$$

where the constant C_{sg} is independent of the level L and the dimension N_s (see Theorem 3.8 in [6]). According to the definition in (3.3), e_2 can be written as

$$e_2 = g_L(\boldsymbol{\theta}) - \widetilde{g}_L(\boldsymbol{\theta}) = \sum_{l=0}^{L} \sum_{|\mathbf{l}|=l} \left(\Delta_{l_1} \otimes \cdots \otimes \Delta_{l_{N_s}}\right)(g - \widetilde{g})(\boldsymbol{\theta}), \tag{A.2}$$

where $\|g(\boldsymbol{\theta}) - \widetilde{g}(\boldsymbol{\theta})\| \leq \tau$. Thus, it is seen that estimating e_2 is equivalent to estimating the Lebesgue constant, denoted by $\Lambda_{N_s,L}$, of the interpolation operator involved. From the representation in (3.3), $\Lambda_{N_s,L}$ can be estimated using the triangle inequality, i.e.,

$$\Lambda_{N_s,L} \leq \sum_{l=0}^{L} \sum_{|\mathbf{l}|=l} \Lambda_{\mathbf{l}} \leq \sum_{l=0}^{L} \sum_{|\mathbf{l}|=l} \prod_{n=1}^{N_s} \Lambda_{l_n},$$

where $\Lambda_{\mathbf{l}} = \Pi_{n=1}^{N_s} \Lambda_{l_n}$ is the Lebesgue constant of $\Delta_{l_1} \otimes \cdots \otimes \Delta_{l_{N_s}}$ and Λ_{l_n} is the Lebesgue constant of Δ_{l_n}. By the definition in (3.1), it is easy to show that

$$\Lambda_{l_n} = \sup \left\{ \frac{\|\Delta_{l_n}(g)\|}{\|g\|} \;\middle|\; g \text{ is continuous and } g \neq 0 \right\} \leq \lambda_{l_n} + \lambda_{l_n-1},$$

where λ_{l_n} and λ_{l_n-1} are the Lebesgue constants of $\mathcal{U}_{l_n}$ and $\mathcal{U}_{l_n-1}$, respectively. In the context of linear hierarchical polynomials, we have $\lambda_{l_n} = 1$. Thus, the Lebesgue constant $\Lambda_{N_s,L}$ can be bounded by

$$\begin{aligned} \Lambda_{N_s,L} &\leq \sum_{l=0}^{L} \sum_{|\mathbf{l}|=l} \prod_{n=1}^{N_s} (\lambda_{l_n} + \lambda_{l_n-1}) \leq \sum_{l=0}^{L} \sum_{|\mathbf{l}|=l} 2^{N_s} \\ &= 2^{N_s} \sum_{l=0}^{L} \binom{l+N_s-1}{N_s-1} = 2^{N_s} \sum_{l=0}^{L} \binom{l+N_s-1}{l} = 2^{N_s} \binom{L+N_s}{N_s}. \end{aligned}$$

Thus, the error e_2 in (A.2) can be estimated by

$$\|e_2\| \leq \Lambda_{N_s,L} \|g(\boldsymbol{\theta}) - \widetilde{g}(\boldsymbol{\theta})\| \leq 2^{N_s} \binom{L+N_s}{N_s} \tau,$$

so that, along with (A.1), (5.4) is obtained.

A.2. The Proof of Lemma 5.4. Using (3.3) and exploiting the nesting structure of the sparse grid, the dimension of $\mathcal{H}_L$ can be represented by

$$|\mathcal{H}_L| = \sum_{l=0}^{L} |\Delta \mathcal{H}_l| = \sum_{l=0}^{L} \sum_{|\mathbf{l}|=l} \prod_{n=1}^{N_s} (m_{l_n} - m_{l_n-1}),$$

where $m_{l_n} = 2^{l_n} + 1$ is the number of grid points involved in the one-dimensional interpolant $\mathcal{U}_{l_n}(\cdot)$ and $m_{-1} = 0$. For the linear hierarchical basis, $m_{l_n} - m_{l_n-1} = 2^{l_n} - 1$

for $l_n \geq 1$. We now derive an upper bound for $|\Delta\mathcal{H}_l|$ for $l \geq 1$. Note that there are $\binom{N_s-1+l}{N_s-1}$ ways to form the sum l with N_s nonnegative integers, so we have

$$|\Delta\mathcal{H}_l| = \prod_{n=1}^{N_s}(m_{l_n} - m_{l_n-1}) \binom{N_s-1+l}{N_s-1} \leq 2^l \frac{(N_s-1+l)!}{(N_s-1)! \cdot l!}.$$

By an inequality from Stirling's approximation of a factorial, i.e.,

$$d_n \leq n! \leq d_n\left(1+\frac{1}{4n}\right) \quad \text{with} \quad d_n = \sqrt{2\pi n}\left(\frac{n}{\mathrm{e}}\right)^n, \ \ n \in \mathbb{N}^+,$$

we obtain that

$$\begin{aligned}
|\Delta\mathcal{H}_l| &\leq 2^l\left(1+\frac{1}{4(N_s-1+l)}\right)\frac{d_{N_s-1+l}}{d_{N_s-1}\cdot d_l}\\
&= 2^l\frac{\left(1+\frac{1}{4(N_s-1+l)}\right)\sqrt{N_s-1+l}}{\sqrt{2\pi l(N_s-1)}}\left(\frac{N_s-1+l}{N_s-1}\right)^{N_s-1}\left(\frac{N_s-1+l}{l}\right)^l\\
&\leq 2^l\left(\frac{l+N_s-1}{N_s-1}\right)^{N_s-1}\left(1+\frac{N_s-1}{l}\right)^l \leq 2^l\left(\frac{l+N_s-1}{N_s-1}\right)^{N_s-1}\mathrm{e}^{N_s-1}.
\end{aligned}$$

It is easy to see that $|\Delta\mathcal{H}_0|$ satisfies the above inequality as well. This concludes the proof about $|\Delta\mathcal{H}^l|$. The estimate for $|\mathcal{H}_L|$ can be obtained immediately based on the estimate of $|\Delta\mathcal{H}_l|$.

A.3. The Proof of Lemma 5.5. We observe that the value of the minimal solution of the inequality (5.5) has two possibilities, i.e., $L < N_s$ and $L \geq N_s$. In the former case, all values larger than N_s are also solutions of (5.5). Hence, we assume the solution of (5.5) is larger than N_s. It is also observed that if $L \geq N_s$, we have

(A.3)
$$\sum_{k=0}^{N_s-1}\binom{L+N_s-1}{k} \leq N_s\binom{L+N_s-1}{N_s-1} \leq N_s\binom{L+N_s}{N_s} \leq N_s\left(\frac{2L}{N_s}\right)^{N_s}\mathrm{e}^{N_s}.$$

Thus, instead of solving (5.5) directly, it is sufficient to solve

$$C_{\mathrm{sg}}2^{-2L}N_s\left(\frac{2L}{N_s}\right)^{N_s}\mathrm{e}^{N_s} \leq \frac{\varepsilon}{2} \qquad \text{and} \qquad L \geq N_s. \tag{A.4}$$

Now, we temporarily treat L as a positive real number for convenience, and the desired iteration number is $\lceil L \rceil$. Let $L = tN_s/\ln 4$ in (A.4). Then we have

$$\begin{aligned}
&\left(\frac{2L}{N_s}\right)^{N_s}\mathrm{e}^{N_s}\left(\frac{2N_sC_{\mathrm{sg}}}{\varepsilon}\right) \leq 2^{2L}\\
\Longleftrightarrow \quad &\left(\frac{t}{\ln 2}\right)^{N_s}\mathrm{e}^{N_s}\left(\frac{2N_sC_{\mathrm{sg}}}{\varepsilon}\right) \leq 4^{\frac{t}{\ln 4}N_s} \quad \Longleftrightarrow \quad \left(\frac{t\mathrm{e}}{\ln 2}\right)\left(\frac{2N_sC_{\mathrm{sg}}}{\varepsilon}\right)^{\frac{1}{N_s}} \leq 4^{\frac{t}{\ln 4}}\\
\Longleftrightarrow \quad &\ln t + \ln\left[\frac{\mathrm{e}}{\ln 2}\left(\frac{2C_{\mathrm{sg}}}{\varepsilon}\right)^{\frac{1}{N_s}}N_s{}^{\frac{1}{N_s}}\right] \leq t \quad \Longleftarrow \quad \ln t + \ln\left[\frac{2\mathrm{e}}{\ln 2}\left(\frac{2C_{\mathrm{sg}}}{\varepsilon}\right)^{\frac{1}{N_s}}\right] \leq t,
\end{aligned}$$

so that (A.4) is satisfied with minimum L given by $L = tN_s/\ln 4$ if t satisfies

$$t \geq \ln t + \ln h \quad \text{with} \quad h = \frac{2\mathrm{e}}{\ln 2}\left(\frac{2C_{\mathrm{sg}}}{\varepsilon}\right)^{\frac{1}{N_s}},$$

where $h > 1$ by hypothesis. Letting $t_0 = \frac{\mathrm{e}}{\mathrm{e}-1}\ln h$, it is easy to verify that

$$t_0 - \ln h = \frac{1}{\mathrm{e}-1}\ln h \geq 1 + \ln\left(\frac{1}{\mathrm{e}-1}\ln h\right) = \ln\left(\frac{\mathrm{e}}{\mathrm{e}-1}\ln h\right) = \ln t_0$$

and that the inequality (A.4) is satisfied. Furthermore, for $k \geq 0$, $t_k = \ln(t_{k-1}h) \leq t_{k-1}$ is also the solution of (A.4) due to the fact that

$$\ln t_k + \ln h = \ln(\ln t_{k-1} + \ln h) + \ln h \leq \ln t_{k-1} + \ln h = \ln(t_{k-1}h) = t_k. \tag{A.5}$$

Thus, the sequence $\{t_k\}_{k=0}^{\infty}$ monotonically converges to a unique solution t^* such that $t^* = \ln t^* + \ln h$. Based on the sequence $\{t_k\}_{k=0}^{\infty}$, we can easily find a sequence of upper bounds $\{L_k\}_{k=0}^{\infty}$ for the minimum L satisfying the inequality (5.5).

A.4. The Proof of Theorem 5.7. According to the definition in (5.2), the cost $\mathcal{C}_{\min}$ can be bounded by

$$\mathcal{C}_{\min} \leq |\mathcal{H}_{L_k}|\, K(\tau_0, \varepsilon, L_k, N_s), \tag{A.6}$$

where L_k for $k \in \mathbb{N}$ is determined from Lemma 5.5 and $K(\tau_0, \varepsilon, L, N_s)$ is the necessary number of iterations for the bisection method to achieve the accuracy $\varepsilon/2$ in (5.6) at each grid point for fixed N_s, L, ε, and initial search interval length τ_0. We can see that the necessary tolerance τ of the bisection method is determined by (5.6), i.e.,

$$\tau(N_s, L, \varepsilon) = 2^{-N_s-1}\varepsilon \Bigg/ \binom{L+N_s}{N_s};$$

$K(\tau_0, \varepsilon, L, N_s)$ can be represented by

$$K(\tau_0, \varepsilon, L, N_s) = \log_2\left[\frac{2^{N_s+1}\tau_0}{\varepsilon}\binom{L+N_s}{N_s}\right], \tag{A.7}$$

where we temporarily treat K as a positive real number for convenience and the desired iteration number is $\lceil K \rceil$. According to the discussion in section 4.2.2, τ_0 is set to $|\boldsymbol{y}_0 - \boldsymbol{\beta}_{\boldsymbol{\theta}}|$ without any prior knowledge; thus $\tau_0 \leq (N_s+1)^{\frac{1}{2}}$, which is the length of the diagonal of $[0,1]^N$. Substituting L_0 into (A.7), we have

$$\begin{aligned}
&K(\tau_0, \varepsilon, L_0, N_s)\\
&\quad \leq \log_2\left(\frac{2^{N_s+1}\tau_0}{\varepsilon}\right) + \log_2\left(\frac{\varepsilon}{2N_sC_{\mathrm{sg}}}2^{2L_0}\right) = \log_2\left(\frac{2^{N_s+1}\tau_0}{C_{\mathrm{sg}}N_s}\right) + 2L_0\\
&\quad \leq \log_2\left(\frac{2^{N_s+1}(N_s+1)^{\frac{1}{2}}}{C_{\mathrm{sg}}N_s}\right) + \frac{\mathrm{e}N_s}{\mathrm{e}-1}\log_2\left[\frac{2\mathrm{e}}{\ln 2}\left(\frac{2C_{\mathrm{sg}}}{\varepsilon}\right)^{\frac{1}{N_s}}\right]\\
&\quad \leq N_s + \frac{\mathrm{e}N_s}{\mathrm{e}-1}\log_2\left[\frac{2\mathrm{e}}{\ln 2}\left(\frac{2C_{\mathrm{sg}}}{\varepsilon}\right)^{\frac{1}{N_s}}\right] + 2 - \log_2(C_{\mathrm{sg}})\\
&\quad = \frac{\mathrm{e}}{\mathrm{e}-1}\log_2\left(\frac{2C_{\mathrm{sg}}}{\varepsilon}\right) + N_s\left\{\frac{\mathrm{e}}{\mathrm{e}-1}\log_2\left(\frac{2\mathrm{e}}{\ln 2}\right) + 1\right\} + 2 - \log_2(C_{\mathrm{sg}})\\
&\quad = \alpha_6\log_2\left(\frac{2C_{\mathrm{sg}}}{\varepsilon}\right) + \alpha_7N_s + \alpha_8.
\end{aligned} \tag{A.8}$$

On the other hand, substituting L_1 into the upper bound of $\mathcal{H}_{L_1}$, we have

$$\begin{aligned}
|\mathcal{H}_{L_1}| &\le 2^{L_1+1}\binom{L_1+N_s-1}{N_s-1} \le 2^{L_1+1}\binom{L_1+N_s}{N_s}\\
&\le 2^{L_1+1}\left(\frac{\varepsilon}{2N_sC_{\rm sg}}\right)2^{2L_1} \le \left(\frac{\varepsilon}{N_sC_{\rm sg}}\right)2^{\frac{3t_1N_s}{2\ln 2}}\\
&= \left(\frac{\varepsilon}{N_sC_{\rm sg}}\right)2^{\frac{3\ln(t_0h)N_s}{2\ln 2}} = \left(\frac{\varepsilon}{N_sC_{\rm sg}}\right)t_0^{\frac{3}{2}N_s}\left[\frac{2\mathrm{e}}{\ln 2}\left(\frac{2C_{\rm sg}}{\varepsilon}\right)^{\frac{1}{N_s}}\right]^{\frac{3}{2}N_s}\\
&= \left(\frac{\varepsilon}{N_sC_{\rm sg}}\right)\left(\frac{\mathrm{e}}{\mathrm{e}-1}\ln h\right)^{\frac{3}{2}N_s}\left(\frac{2\mathrm{e}}{\ln 2}\right)^{\frac{3}{2}N_s}\left(\frac{2C_{\rm sg}}{\varepsilon}\right)^{\frac{3}{2}} \qquad \text{(A.9)}\\
&= \frac{2}{N_s}\left(\frac{2C_{\rm sg}}{\varepsilon}\right)^{\frac{1}{2}}\left\{\frac{2\mathrm{e}^2}{\mathrm{e}-1}\log_2\left[\frac{2\mathrm{e}}{\ln 2}\left(\frac{2C_{\rm sg}}{\varepsilon}\right)^{\frac{1}{N_s}}\right]\right\}^{\frac{3}{2}N_s}\\
&= \frac{2}{N_s}\left\{\frac{2\mathrm{e}^2}{\mathrm{e}-1}\log_2\left(\frac{2\mathrm{e}}{\ln 2}\right)+\frac{2\mathrm{e}^2}{\mathrm{e}-1}\frac{\log_2\left(\frac{2C_{\rm sg}}{\varepsilon}\right)}{N_s}\right\}^{\frac{3}{2}N_s}\left(\frac{2C_{\rm sg}}{\varepsilon}\right)^{\frac{1}{2}}\\
&= \alpha_1\left\{\alpha_2+\alpha_3\frac{\log_2\left(\frac{2C_{\rm sg}}{\varepsilon}\right)}{N_s}\right\}^{\alpha_4N_s}\left(\frac{2C_{\rm sg}}{\varepsilon}\right)^{\alpha_5}.
\end{aligned}$$

Hence, by substituting (A.8) and (A.9) into (A.6), the proof is finished.

A.5. The Proof of Lemma 5.8. As in (5.4), we split the error into two parts, i.e.,

$$g(\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}})-\widetilde{g}_{L-1}(\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}}) = \underbrace{g(\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}})-g_{L-1}(\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}})}_{e_1}+\underbrace{g_{L-1}(\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}})-\widetilde{g}_{L-1}(\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}})}_{e_2},$$

where e_1 is the definition of the hierarchical surplus $\omega_{\mathbf{l},\mathbf{i}}$ whose upper bound is given in [6], i.e., $|e_1| \le C_{\rm surp}\cdot 2^{-2L}$ with $C_{\rm surp}$ independent of L, and e_2 measures the error between the exact prediction of the surplus and the perturbed one. To estimate e_2, we need to extend the formula for calculating surpluses given in [6] by including the sparse grid points on the boundary. Based on [6, Lemma 3.2], we can see that for each grid point $\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}}$ with $|\mathbf{l}| \ge 1$, its exact surplus $\omega_{\mathbf{l},\mathbf{i}}$ can be computed from the function values of $g(\boldsymbol{\theta})$ as follows:

$$\omega_{\mathbf{l},\mathbf{i}} = \mathcal{A}_{\mathbf{l},\mathbf{i}}(g) = \left(\prod_{n=1}^{N_s}\mathcal{A}_{l_n,i_n}\right)(g),$$

where $\mathcal{A}_{\mathbf{l},\mathbf{i}}(\cdot)$ is an N_s-dimensional stencil that provides the coefficients for a linear combination of the nodal values of the function g to compute $\omega_{\mathbf{l},\mathbf{i}}$. Specifically, $\mathcal{A}_{\mathbf{l},\mathbf{i}}$ is the product of N_s one-dimensional stencils $\mathcal{A}_{l_n,i_n}$ for $l_n > 1, n = 1,\dots,N_s$, defined by

$$\begin{aligned}
\mathcal{A}_{l_n,i_n}(g) &= \begin{bmatrix}-\frac{1}{2} & 1 & -\frac{1}{2}\end{bmatrix}_{l_n,i_n}(g) \qquad \text{(A.10)}\\
&= -\frac{1}{2}g(\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}}-h_{l_n}\boldsymbol{e}_n)+g(\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}})-\frac{1}{2}g(\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}}+h_{l_n}\boldsymbol{e}_n),
\end{aligned}$$

where $\boldsymbol{e}_n$ is a vector of zeros except for the nth entry, which is one, and h_{l_n} is a scalar equal to a half of the length of the support of the basis function $\psi_{\mathbf{l},\mathbf{i}}(\boldsymbol{\theta})$ in the nth direction. Note that we have $\mathcal{A}_{0,0}(g) = [0,1,0]_{0,0}(g)$ for $l_n = 0, i_n = 0$, $\mathcal{A}_{1,0}(g) = [0,1,-1]_{1,0}(g)$ for $l_n = 1, i_n = 0$, and $\mathcal{A}_{1,2}(g) = [-1,1,0]_{1,2}(g)$ for $l_n = 1, i_n = 2$. It is easy to see that the sum of the absolute values of the coefficients of $\mathcal{A}_{\mathbf{l},\mathbf{i}}(\cdot)$ is equal to 2^{N_s}. Note that all the involved grid points in (A.10) belong to $g_{L-1}(\boldsymbol{\theta})$ except for $\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}}$. Thus, due to the fact that $|g(\boldsymbol{\theta}) - \widetilde{g}(\boldsymbol{\theta})| \le \tau$, the error e_2 can be estimated by

$$|e_2| = |\mathcal{A}_{\mathbf{l},\mathbf{i}}(g - \widetilde{g}) - (g(\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}}) - \widetilde{g}(\boldsymbol{\theta}_{\mathbf{l},\mathbf{i}}))| \le 2^{N_s}\tau,$$

which completes the proof.

A.6. The Proof of Theorem 5.9. For $L = L_1$, according to the definition in (5.2), $\mathcal{C}_{\min}$ can be bounded by

$$\begin{aligned}\mathcal{C}_{\min} &\le \sum_{l=0}^{L_1} |\Delta\mathcal{H}_l|\, K(\tau_0^l, \varepsilon, L_1, N_s)\\ &\le \sum_{l=0}^{L_1} 2^l \binom{l+N_s-1}{N_s-1} \log_2\left[\frac{2^{N_s+1}\tau_0^l}{\varepsilon}\binom{L_1+N_s}{N_s}\right],\end{aligned} \tag{A.11}$$

where we temporarily treat K as a positive real number for convenience and the desired iteration number is $\lceil K \rceil$. Based on Lemma 5.8, we define the initial search interval τ_0^l on level l by $\tau_0^l = C_{\text{surp}}2^{-2l} + 2^{N_s}\tau$, where τ is the tolerance of the bisection method. For sufficiently small ε, the logarithmic function in (A.11) is positive. Substituting such τ_0^l into (A.11), we obtain

$$\begin{aligned}\mathcal{C}_{\min} &\le \sum_{l=0}^{L_1} 2^l \binom{l+N_s-1}{N_s-1} \log_2\left[\frac{2^{N_s+1}}{\varepsilon}\binom{L_1+N_s}{N_s}\left(C_{\text{surp}}2^{-2l} + 2^{N_s}\tau\right)\right]\\ &= \sum_{l=0}^{L_1} 2^l \binom{l+N_s-1}{N_s-1} \log_2\left[\frac{2^{N_s+1}}{\varepsilon}\binom{L_1+N_s}{N_s}\left(C_{\text{surp}}2^{-2l} + \frac{\varepsilon}{2\binom{L_1+N_s}{N_s}}\right)\right]\\ &= \sum_{l=0}^{L_1} 2^l \binom{l+N_s-1}{N_s-1} \left\{\log_2\left[\frac{2^{N_s+1}C_{\text{surp}}2^{-2l}}{\varepsilon}\binom{L_1+N_s}{N_s}\right] + N_s\right\}\\ &\le \sum_{l=0}^{L_1} 2^l \binom{l+N_s-1}{N_s-1} \left\{\log_2\left[\frac{2^{N_s+1}C_{\text{surp}}2^{2(L_1-l)}}{\varepsilon}\frac{\varepsilon}{2N_sC_{\text{sg}}}\right] + N_s\right\}\\ &= \sum_{l=0}^{L_1} 2^l \binom{l+N_s-1}{N_s-1} \left[2(L_1-l) + \log_2\left(\frac{C_{\text{surp}}}{C_{\text{sg}}}\right) + 2N_s - \log_2(N_s)\right]\\ &\le \binom{L_1+N_s}{N_s}\sum_{l=0}^{L_1}(L_1-l)2^l + 2^{L_1+1}\binom{L_1+N_s}{N_s}\left[\log_2\left(\frac{C_{\text{surp}}}{C_{\text{sg}}}\right) + 2N_s - \log_2(N_s)\right]\\ &\le 2^{L_1+1}\binom{L_1+N_s}{N_s}\left[\log_2\left(\frac{C_{\text{surp}}}{C_{\text{sg}}}\right) + 2N_s + 2 - \log_2(N_s)\right]\\ &\le \alpha_1\left[\alpha_2 + \alpha_3\frac{\log_2\left(\frac{2C_{\text{sg}}}{\varepsilon}\right)}{N_s}\right]^{\alpha_4 N_s}\left(\frac{2C_{\text{sg}}}{\varepsilon}\right)^{\alpha_5}\left[2N_s - \log_2(N_s) + \alpha_9\right],\end{aligned}$$

which completes the proof.

REFERENCES

[1] R. Archibald, A. Gelb, R. Saxena, and D. Xiu, *Discontinuity detection in multivariate space for stochastic simulations*, J. Comput. Phys., 228 (2009), pp. 2676–2689. (Cited on p. 518)

[2] R. Archibald, A. Gelb, and J. Yoon, *Polynomial fitting for edge detection in irregularly sampled signals and images*, SIAM J. Numer. Anal., 43 (2005), pp. 259–279. (Cited on p. 518)

[3] E. van den Berg and M. P. Friedlander, *SPGL1: A Solver for Large-Scale Sparse Reconstruction*, 2007, http://www.cs.ubc.ca/labs/scl/spgl1. (Cited on p. 542)

[4] M. Buhmann, *Radial Basis Functions: Theory and Implementations*, Cambridge University Press, Cambridge, UK, 2003. (Cited on pp. 519, 535)

[5] H. J. Bungartz and S. Dirnstorfer, *Multivariate quadrature on adaptive sparse grids*, Computing, 71 (2003), pp. 89–114. (Cited on p. 518)

[6] H. J. Bungartz and M. Griebel, *Sparse grids*, Acta Numer., 13 (2004), pp. 147–269. (Cited on pp. 518, 522, 524, 525, 533, 535, 544, 545, 548)

[7] H. J. Bungartz, D. Pflüger, and S. Zimmer, *Adaptive sparse grid techniques for data mining*, in Modeling, Simulation and Optimization of Complex Processes, Springer, Berlin 2008, pp. 121–130. (Cited on p. 518)

[8] A. Chkifa, N. Dexter, H. Tran, and C. G. Webster, *Polynomial Approximation via Compressed Sensing of High-Dimensional Functions on Lower Sets*, preprint, arxiv:1602.05823, 2016. (Cited on pp. 519, 537)

[9] A. Cohen, M. A. Davenport, and D. Leviatan, *On the stability and accuracy of least squares approximations*, Found. Comput. Math., 5 (2013), pp. 819–834. (Cited on pp. 519, 536)

[10] C. Feuersänger and M. Griebel, *Principal manifold learning by sparse grids*, Computing, 85 (2009), pp. 267–299. (Cited on pp. 518, 519)

[11] S. Foucart and H. Rauhut, *A Mathematical Introduction to Compressive Sensing*, Birkhäuser, Basel, 2013. (Cited on pp. 519, 537)

[12] D. Galido, P. Jantsch, C. G. Webster, and G. Zhang, *Accelerating Hierarchical Stochastic Collocation Methods for Partial Differential Equations with Random Input Data*, preprint, arXiv:1505.00680v1 [math.NA], 2015. (Cited on pp. 518, 519)

[13] T. Gerstner and M. Griebel, *Numerical integration using sparse grids*, Numer. Algorithms, 18 (1998), pp. 209–232. (Cited on p. 518)

[14] T. Gerstner and M. Griebel, *Dimension-adaptive tensor-product quadrature*, Computing, 71 (2003), pp. 65–87. (Cited on p. 518)

[15] M. Griebel, *Adaptive sparse grid multilevel methods for elliptic PDEs based on finite differences*, Computing, 61 (1998), pp. 151–179. (Cited on p. 518)

[16] M. Griebel and M. Holtz, *Dimension-wise integration of high-dimensional functions with applications to finance*, J. Complexity, 26 (2010), pp. 455–489. (Cited on p. 518)

[17] M. Gunzburger, C. Webster, and G. Zhang, *An adaptive wavelet stochastic collocation method for irregular solutions of partial differential equations with random input data*, in Sparse Grids and Applications, Springer Lect. Notes Comput. Sci. Eng. 97, Springer, Cham, Switzerland, 2014, pp. 137–170. (Cited on p. 524)

[18] M. Gunzburger, C. G. Webster, and G. Zhang, *Stochastic finite element methods for partial differential equations with random input data*, Acta Numer., 23 (2014), pp. 521–650. (Cited on pp. 519, 529)

[19] M. Holtz, *Sparse Grid Quadrature in High Dimensions with Applications in Finance and Insurance*, Lect. Notes Comput. Sci. Eng. 77, Springer, Berlin, 2011. (Cited on p. 518)

[20] 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. (Cited on pp. 518, 522)

[21] X. Ma and N. Zabaras, *An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations*, J. Comput. Phys., 228 (2009), pp. 3084–3113. (Cited on p. 524)

[22] X. Ma and N. Zabaras, *An efficient Bayesian inference approach to inverse problems based on an adaptive sparse grid collocation method*, Inverse Problems, 25 (2009), article 035013. (Cited on p. 518)

[23] G. Migliorati, F. Nobile, E. von Schwerin, and R. Tempone, *Analysis of discrete L^2 projection on polynomial spaces with random evaluations*, Found. Comput. Math., 14 (2014), pp. 419–456. (Cited on pp. 519, 536)

[24] 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. (Cited on pp. 518, 533)

[25] F. Nobile, R. Tempone, and C. G. Webster, *An anisotropic sparse grid stochastic collocation method for partial differential equations with random input data*, SIAM J. Numer. Anal., 46 (2008), pp. 2411–2442. (Cited on p. 518)

[26] E. Novak and K. Ritter, *High-dimensional integration of smooth functions over cubes*, Numer. Math., 75 (1996), pp. 79–97. (Cited on p. 518)

[27] B. Peherstorfer, J. Adorf, D. Pflüger, and H.-J. Bungartz, *Image segmentation with adaptive sparse grids*, in AI 2013: Advances in Artificial Intelligence, Lecture Notes Comput. Sci. 8272, Springer, Berlin, 2013, pp. 160–165. (Cited on p. 518)

[28] D. Pflüger, I. L. Muntean, and H.-J. Bungartz, *Adaptive sparse grid classification using grid environments*, in Computational Science–ICCS 2007, Lecture Notes in Comput. Sci. 4487, Springer, Berlin, 2007, pp. 708–715. (Cited on p. 519)

[29] A. Quarteroni, R. Sacco, and F. Saleri, *Numerical Mathematics*, Springer-Verlag, New York, 2007. (Cited on pp. 529, 537)

[30] M. Rudelson and R. Vershynin, *On sparse reconstruction from Fourier and Gaussian measurements*, Comm. Pure Appl. Math. 61 (2008), 1025–1045. (Cited on pp. 519, 537)

[31] G. W. Wasilkowski and H. Woźniakowski, *Explicit cost bounds of algorithms for multivariate tensor product problems*, J. Complexity, 11 (1995), pp. 1–56. (Cited on pp. 518, 533, 535)

[32] H. Wendland, *Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree*, Adv. Comput. Math., 4 (1995), pp. 389–396. (Cited on pp. 519, 536)

[33] C. G. Webster, G. Zhang, and M. Gunzburger, *An adaptive sparse grid iterative ensemble Kalman filter approach for parameter field estimation*, Int. J. Comput. Math., 91 (2013), pp. 798–817. (Cited on p. 518)

[34] G. Zhang, M. Gunzburger, and W. Zhao, *A sparse grid method for multi-dimensional backward stochastic differential equations*, J. Comput. Math., 31 (2013), pp. 221–248. (Cited on p. 518)

[35] G. Zhang, D. Lu, M. Ye, M. Gunzburger, and C. G. Webster, *An adaptive sparse grid high-order stochastic collocation method for Bayesian inference in groundwater reactive transport modeling*, Water Res. Res., 49 (2013), pp. 6871–6892. (Cited on p. 518)