Mini-course: Uncertainty Quantification and Approximation Theory for Parameterized PDEs

Part VI: local adaptive sparse grid methods and applications

Guannan Zhang[⋆]

Clayton G. Webster[†⋆], Hoang Tran[⋆]

[⋆]Computer Science and Mathematics Division, Oak Ridge National Laboratory
[†]Department of Mathematics, University of Tennessee

Supported by DOE (ASCR), DOD (DARPA), NSF (Comp math)

Vietnam Institute for Advanced Study in Mathematics

November 16, 2016

1. Piecewise hierarchical polynomial basis
2. Adaptive sparse grid interpolation
3. Sparse grids with other types of basis functions
4. Application 1: high-dimensional discontinuity detection
5. Application 2: hierarchical acceleration of stochastic collocation methods

- Sparse grids can be constructed based on many types of piecewise (local) bases. We first use the hat function to explain the construction, and then introduce several other types of bases.

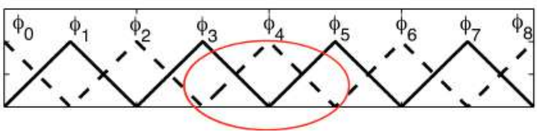
- We consider the 1-D hat function having support $[-1, 1]$ defined by

$$\psi(y) = \max\{1 - |y|, 0\},$$

from which an arbitrary hat function with support $(y_{l,i} - h_l, y_{l,i} + h_l)$ can be generated by dilation and translation, that is,

$$\psi_{l,i}(y) := \psi\left(\frac{y + 1 - ih_l}{h_l}\right),$$

where h_l denotes the grid size on the resolution level l.



- Delta property: each nodal basis function is zero at other grid poins, i.e.,

$$\psi_{l,i}(y_{l,i'}) = \delta_{ii'},$$

  with δ being the Kronecker delta

- A sequence of nodal bases can be generated by defining a sequence of mesh sizes $\{h_l, l = 0, 1, \ldots\}$.

- The most common choice is the dyadic rule, i.e.,

$$h_{l+1} = \frac{h_l}{2}, \quad N_l = 2^l + 1, \quad \text{for} \quad l = 0, 1, \ldots.$$

- We define V_l to represent the space expanded by the nodal basis on level l, i.e.,

$$V_l := \text{span}\left\{\psi_{l,i}(y) \mid 0 \leq i \leq 2^l\right\}.$$

- Due to the dyadic rule, the sequence $\{V_l\}$ is nested, i.e.,

$$V_0 \subset V_1 \subset \cdots \subset V_l \subset V_{l+1} \subset \cdots \subset V.$$

One-dimensional hierarchical basis

Based on the hat function

- Due to the nesting structure of $\{V_l\}_{l=0}^{\infty}$, we can define the incremental subspace

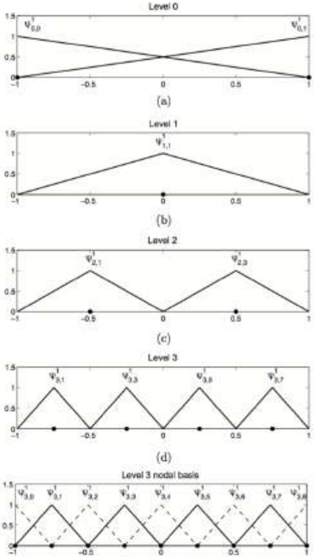
$$W_l := V_l / V_{l-1} \quad \Longrightarrow \quad V_l = V_{l-1} \oplus W_l.$$

- Then, we have a hierarchical subspace splitting of V_l given by

$$V_l = V_0 \oplus W_1 \oplus \cdots \oplus W_l \ \text{ for } \ l = 1, 2, \ldots.$$

- Each W_l contains about half of the basis functions of the associated V_l.

- Non-overlapping property:
  For $l \geq 1$, the supports of the basis functions in W_l do not overlap.



- The hierarchical and the nodal bases expand the same subspace V_l.
- The hierarchical basis only possesses a partial delta property:

Partial Delta Property

The basis functions corresponding to a specific level possess the delta property with respect to its own level and coarser levels, but not with respect to finer levels.

$$\begin{aligned}
&\text{for } 0 \le l' < l, && \psi_{l,i}(y_{l',i'}) = 0 && \text{for all } i' \in B_{l'},\\
&\text{for } l' = l, && \psi_{l,i}(y_{l,i'}) = \delta_{i,i'} && \text{for all } i' \in B_{l'},\\
&\text{for } l < l' \le l, && \psi_{l,i}(y_{l',i'}) \neq 0 && \text{for all } i' \in B_{l'}.
\end{aligned}$$

- Our goal is to use such basis to build hierarchical interpolation in V_l.

- The interpolant of a function $g(y)$ in the subspace V_l in terms of the its nodal basis $\{\psi_{l,i}(y)\}_{i=0}^{2^l}$ is given by

$$\mathcal{I}_l[g](y) := \sum_{i=0}^{2^l} g(y_{l,i})\psi_{l,i}(y).$$

- Due to the nesting property $V_l = V_{l-1} \oplus W_l$, we have $\mathcal{I}_{l-1}[g] = \mathcal{I}_l\left[\mathcal{I}_{l-1}[g]\right]$, based on which we define the incremental interpolation operator

$$\begin{aligned}\Delta_l[g] &= \mathcal{I}_l[g] - \mathcal{I}_{l-1}[g] = \mathcal{I}_l\left[g - \mathcal{I}_{l-1}[g]\right] \\ &= \sum_{i=0}^{2^l}\left\{g(y_{l,i}) - \mathcal{I}_{l-1}[g](y_{l,i})\right\}\psi_{l,i}(y) \\ &= \sum_{i\in B_l}\left\{g(y_{l,i}) - \mathcal{I}_{l-1}[g](y_{l,i})\right\}\psi_{l,i}(y) = \sum_{i\in B_l} c_{l,i}\psi_{l,i}(y),\end{aligned}$$

where $c_{l,i} = g(y_{l,i}) - \mathcal{I}_{l-1}\left[g(y_{l,i})\right]$. Then we have

$$\mathcal{I}_l[g] = \mathcal{I}_{l-1}[g] + \Delta_l[g] = \cdots = \mathcal{I}_0[g] + \sum_{l'=1}^{l}\Delta_{l'}[g].$$

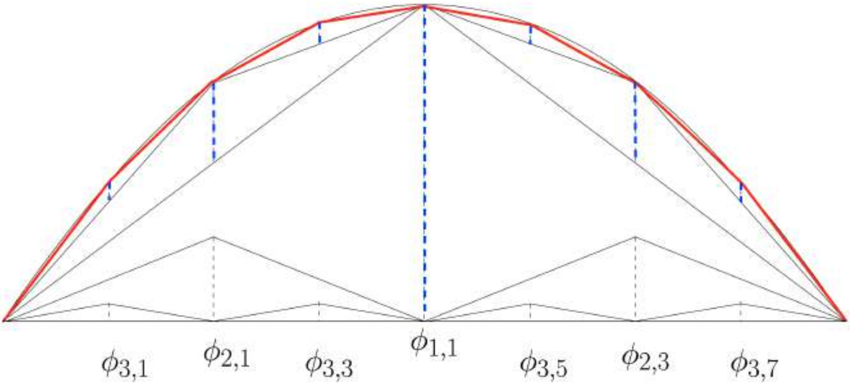


Figure: (Solid red line) the piecewise linear interplolant; (Dashed blue line) the absolute value of the coefficients $c_{l,i}$

- The one-dimensional hierarchical polynomial basis can be extended to the d-dimensional parameter domain Γ using tensorization.

- The d-variate basis function $\psi_{\mathbf{l},\mathbf{i}}(\boldsymbol{y})$ associated with $\boldsymbol{y}_{\mathbf{l},\mathbf{i}} = (y_{l_1,i_1}, \ldots, y_{l_d,i_d})$ is defined using tensor products, i.e.,

$$\psi_{\mathbf{l},\mathbf{i}}(\boldsymbol{y}) := \prod_{n=1}^{d} \psi_{l_n,i_n}(y_n),$$

  where $\mathbf{l} = (l_1, \ldots, l_d)$ and $\mathbf{i} = (i_1, \ldots, i_d)$ are multiindices indicating the resolution level and the grid point within the level.

- The multidimensional expanded by nodal basis of level $\mathbf{l}$:

$$\mathcal{V}_{\mathbf{l}} := \text{span}\{\psi_{\mathbf{l},\mathbf{i}}(\boldsymbol{y}) \mid i_n = 0, \ldots, 2^{l_n},\ n = 1, \ldots, d\},$$

  which might be anisotropic, i.e., $l_n \neq l_{n'}$ for some $n \neq n'$.

- The d-dimensional hierarchical incremental subspace $W_{\mathbf{l}}$ can be defined by

$$W_{\mathbf{l}} = \bigotimes_{n=1}^{N} W_{l_n} = \text{span}\left\{\psi_{\mathbf{l},\mathbf{i}}(\boldsymbol{y}) | \mathbf{i} \in B_{\mathbf{l}}\right\},$$

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

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

- A subspace $\mathcal{V}_{\mathcal{J}}$ can be defined by the direct sum of a set of $W_{\mathbf{l}}$, i.e.,

$$\mathcal{V}_{\mathcal{J}} := \bigoplus_{\mathbf{l}\in\mathcal{J}} W_{\mathbf{l}}.$$

- $W_{\mathbf{l}}$ are like "building blocks", and the multi-index set $\mathcal{J}$ is like a "blueprint" determining which set of building blocks are chosen to construct $\mathcal{V}_{\mathcal{J}}$.

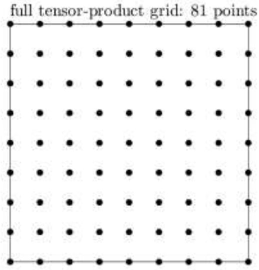
- In any subspace $\mathcal{V}_{\mathbf{l}}$, we can define a tensor-product interpolation operator

$$\mathcal{I}_{\mathbf{l}}[g] := \sum_{i_1=0}^{2^{l_1}} \cdots \sum_{i_d=0}^{2^{l_d}} g(y_{l_1,i_1}, \ldots, y_{l_d,i_d}) \left(\prod_{n=1}^{d} \psi_{l_n,i_n}(y_n) \right),$$

- In any subspace $\mathcal{W}_{\mathbf{l}}$, we can define a tensor-product incremental operator

$$\begin{aligned}\Delta_{\mathbf{l}}[g] &:= \Delta_{l_1} \otimes \cdots \otimes \Delta_{l_d}[g] \\ &= \bigotimes_{n=1}^{d} (\mathcal{I}_{l_n} - \mathcal{I}_{l_n - 1})[g] \\ &= \sum_{\boldsymbol{\alpha} \in \{0,1\}^d} \left((-1)^{|\boldsymbol{\alpha}|} \bigotimes_{n=1}^{d} \mathcal{I}_{l_n - \alpha_n}[g] \right),\end{aligned}$$

where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_d)$ and $|\boldsymbol{\alpha}| := \alpha_1 + \cdots + \alpha_d$.



- The isotropic sparse grid interpolant is defined by choosing the index set $\mathcal{J}$ in the following way

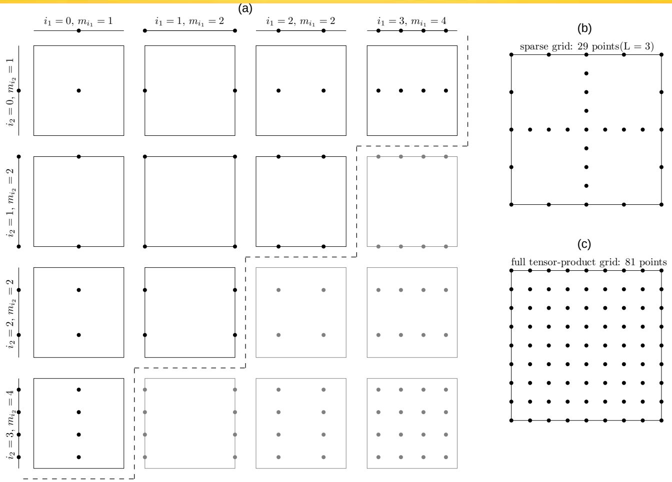
$$\mathcal{J} = \mathcal{J}_L^{\rm sg} := \left\{ \mathbf{l} \in \mathbb{N}^d \;\middle|\; |\mathbf{l}| = l_1 + \cdots + l_d \leq L \right\}.$$

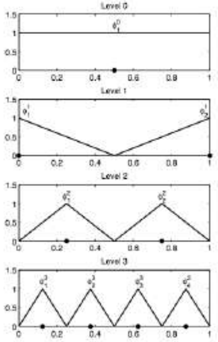
- The corresponding polynomial subspace $\mathcal{V}_{\mathcal{J}}$ is given by

$$\mathcal{V}_{\mathcal{J}} := \mathcal{V}_{\mathcal{J}_L^{\rm sg}} := \bigoplus_{\mathbf{l} \in \mathcal{J}_L^{\rm sg}} W_{\mathbf{l}} = \bigoplus_{l=0}^{L} \bigoplus_{|\mathbf{l}|=l} W_{\mathbf{l}}.$$

- The sparse grid interpolant can be naturally obtained by summing all the $\Delta_{\mathbf{l}}$ associated with $\mathcal{J}_L^{\rm sg}$,

$$\mathcal{I}_L^{\rm sg}[g](\boldsymbol{y}) := \sum_{l=0}^{L} \sum_{|\mathbf{l}|=l} \underbrace{\Delta_{l_1} \otimes \cdots \otimes \Delta_{l_d}}_{\Delta_{\mathbf{l}}}[g](\boldsymbol{y}).$$





Sparse grid v.s. full tensor product

$$\mathcal{J}_L^{\mathrm{sg}} := \left\{ \mathbf{l} \in \mathbb{N}^d \mid |\mathbf{l}| \leq L \right\} \quad \text{v.s.} \quad \mathcal{J}_L^{\mathrm{tp}} := \left\{ \mathbf{l} \in \mathbb{N}^d \mid \max_n l_n \leq L \right\}$$

- Computing the coefficients of any Lagrange interpolant is equivalent to solving a linear system

$$\boldsymbol{\Psi c} = \boldsymbol{g},$$

where Ψ_{ij} is the value of the j-th basis function evaluated at the i-th interpolation point.

- The interpolation matrix of a tensor product interpolant is an identity matrix, due to the delta property, i.e., $\Psi_{ij} = \delta_{ij}$.

- The sparse grid interpolant can also be written as the linear combination of the basis functions in $\mathcal{V}_{\mathcal{J}_L^{\rm sg}}$

$$\begin{aligned}\mathcal{I}_L^{\rm sg}[g](\boldsymbol{y}) &= \sum_{l=0}^{L} \sum_{|\mathbf{l}|=l} \Delta_{l_1} \otimes \cdots \otimes \Delta_{l_d}[g](\boldsymbol{y}) \\ &= \sum_{l=0}^{L} \sum_{|\mathbf{l}|=l} \sum_{\mathbf{i} \in B_{\mathbf{l}}} c_{\mathbf{l},\mathbf{i}} \, \psi_{\mathbf{l},\mathbf{i}}(\boldsymbol{y})\end{aligned}$$

- The resulting linear system can be solved by some linear solvers, but we would like to see if we could exploit the partial delta property to solve it explicitly.

- The sparse grid interpolation can be written as a recursive formulation,

$$\begin{aligned}\mathcal{I}_L^{\rm sg}[g](\boldsymbol{y}) &:= \sum_{l=0}^{L} \sum_{|\mathbf{l}|=l} \Delta_{l_1} \otimes \cdots \otimes \Delta_{l_d}[g](\boldsymbol{y}) \\ &= \mathcal{I}_{L-1}^{\rm sg}[g](\boldsymbol{y}) + \sum_{|\mathbf{l}|=L} \Delta_{l_1} \otimes \cdots \otimes \Delta_{l_d}[g](\boldsymbol{y}) \\ &= \mathcal{I}_{L-1}^{\rm sg}[g](\boldsymbol{y}) + \sum_{|\mathbf{l}|=L} \sum_{\mathbf{i}\in B_{\mathbf{l}}} c_{\mathbf{l},\mathbf{i}}\, \psi_{\mathbf{l},\mathbf{i}}(\boldsymbol{y}).\end{aligned}$$

- For any $\mathbf{l}$ satisfying $|\mathbf{l}| = L$ and any $\mathbf{l}'$ satisfying $|\mathbf{l}'| \leq L$, there exists one component $l_n > l'_n$, such that $\psi_{l_n,i_n}(y_{l'_n,j_n}) = 0$ due to the partial delta property. Thus, we have

$$\psi_{\mathbf{l},\mathbf{i}}(\boldsymbol{y}_{\mathbf{l}',\mathbf{j}}) := 0 \quad \text{for} \quad \mathbf{l} \geq \mathbf{l}'.$$

Computation of the coefficients

- Suppose we are given $\mathcal{I}_{L-1}^{\text{sg}}[g](\boldsymbol{y})$, and now we add new points on level L.
- Substituting any interpolation points $\boldsymbol{y}_{\mathbf{l},\mathbf{i}}$ satisfying $|\mathbf{l}| < L$, we have

$$\mathcal{I}_{L}^{\text{sg}}[g](\boldsymbol{y}_{\mathbf{l},\mathbf{i}}) = \mathcal{I}_{L-1}^{\text{sg}}[g](\boldsymbol{y}_{\mathbf{l},\mathbf{i}}),$$

  which means the new added basis functions on level L will not change the coefficients of $\mathcal{I}_{L-1}^{\text{sg}}[g]$.
- Substituting any new added point $\boldsymbol{y}_{\mathbf{l},\mathbf{i}}$ satisfying $|\mathbf{l}| = L$, we have

$$\mathcal{I}_{L-1}^{\text{sg}}[g](\boldsymbol{y}_{\mathbf{l},\mathbf{i}}) + c_{\mathbf{l},\mathbf{i}}\psi_{\mathbf{l},\mathbf{i}}(\boldsymbol{y}_{\mathbf{l},\mathbf{i}}) = g(\boldsymbol{y}_{\mathbf{l},\mathbf{i}}),$$

  such that $c_{\mathbf{l},\mathbf{i}}$ can be computed explicitly by

$$c_{\mathbf{l},\mathbf{i}} = g(\boldsymbol{y}_{\mathbf{l},\mathbf{i}}) - \mathcal{I}_{L-1}^{\text{sg}}[g](\boldsymbol{y}_{\mathbf{l},\mathbf{i}}),$$

  where $c_{\mathbf{l},\mathbf{i}}$ is called the "surplus".
- In other words, the interpolation matrix is a lower triangular matrix.

- By defining the mixed derivative and a norm

$$D^{\boldsymbol{\alpha}} g := \frac{\partial^{|\boldsymbol{\alpha}|} g}{\partial y_1^{\alpha_1} \cdots \partial y_d^{\alpha_d}}, \text{ and } \|g\|^2_{H^s_{\text{mix}}} := \sum_{0 \le \boldsymbol{\alpha} \le s} |D^{\boldsymbol{\alpha}} g|_2^2,$$

the space H^s_{mix} can be defined in natural way:

$$H^s_{\text{mix}} := \left\{ g : \Gamma \to \mathbb{R} \Big| \|g\|_{H^s_{\text{mix}}} < \infty \right\}.$$

- For a function $g \in H^2_{\text{mix}}$, the error of the sparse grid interpolant $\mathcal{I}^{\text{sg}}_L[g]$ is

$$\|g - \mathcal{I}^{\text{sg}}_L[g]\|_{L^2} = \mathcal{O}\left(h_L^2 \log(h_L^{-1})^{d-1} \right),$$

while the error of the full tensor product interpolant is

$$\|g - \mathcal{I}^{\text{tp}}_L[g]\|_{L^2} = \mathcal{O}\left(h_L^2 \right)$$

- However the complexity comparison, i.e., the number of grid points, is

$$\#(\mathcal{V}^{\text{sg}}_L) = \mathcal{O}\left(h_L^{-1} \log(h_L^{-1})^{d-1} \right) \text{ v.s. } \#(\mathcal{V}^{\text{tp}}_L) = \mathcal{O}\left(h_L^{-d} \right).$$

- Adaptive mesh refinement (AMR) has been widely used to approximate functions with irregular behavior, e.g., steep gradient, sharp transition, and jump discontinuities.
- The key of AMR is to exploits an a-posteriori error indicator to measure the error of the current approximation, and guide us where to add new grid points.
- Can we do mesh refinement on sparse grids?
- If so, what is the error indicator?

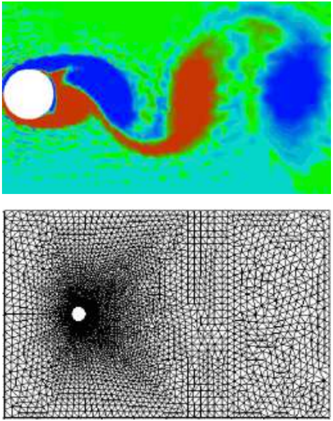
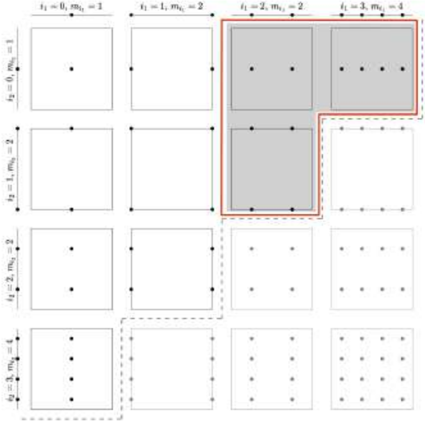
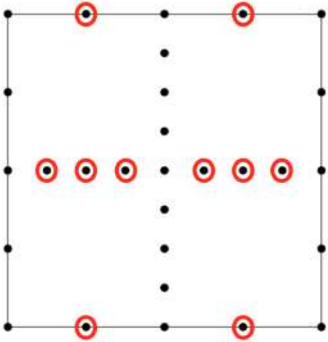


Figure: (Top) Turbulent flow past a cylinder; (Bottom) An adaptive triangulation [Tran, Webster, Z, 16]

- This strategy shares the same idea as the global sparse grids. Instead of having the same resolution along each direction, we can add the anisotropy by defining a weighted norm for the multi-index $\mathbf{l}$, i.e.,

$$\mathcal{J}_L^{\text{aniso}} := \left\{ \mathbf{l} \in \mathbb{N}^d \;\middle|\; |\mathbf{l}|_w = w_1 l_1 + \cdots + w_d l_d \le L \right\}.$$





- Recall the expression of the surplus $c_{\mathbf{l},\mathbf{i}}$

$$c_{\mathbf{l},\mathbf{i}} = g(\boldsymbol{y}_{\mathbf{l},\mathbf{i}}) - \mathcal{I}^{\rm sg}_{L-1}[g](\boldsymbol{y}_{\mathbf{l},\mathbf{i}}),$$

which can be bounded by [Bungartz, Griebel, 04]

$$|c_{\mathbf{l},\mathbf{i}}| \leq C 2^{-2|\mathbf{l}|},$$

such that the surplus can be used as an error indicator to guide the refinement.

- For a given threshold $\tau > 0$, the level L interpolant $\mathcal{I}^{\rm sg}_{\tau,L}[g]$ retains only the terms of the isotropic SG interpolant $\mathcal{I}^{\rm sg}_{L}[g]$ for which the magnitudes of the corresponding surpluses are larger than τ, i.e.,

$$\mathcal{I}^{\rm sg}_{\tau,L}[g](\boldsymbol{y}) = \sum_{l=0}^{L} \sum_{|\mathbf{l}|=l} \sum_{\mathbf{i}\in B^{\tau}_{\mathbf{l}}} c_{\mathbf{l},\mathbf{i}}\, \psi_{\mathbf{l},\mathbf{i}}(\boldsymbol{y}) \quad \text{with} \quad B^{\tau}_{\mathbf{l}} = \{\mathbf{i} \in B_{\mathbf{l}} \,|\, |c_{\mathbf{l},\mathbf{i}}| > \tau\}.$$

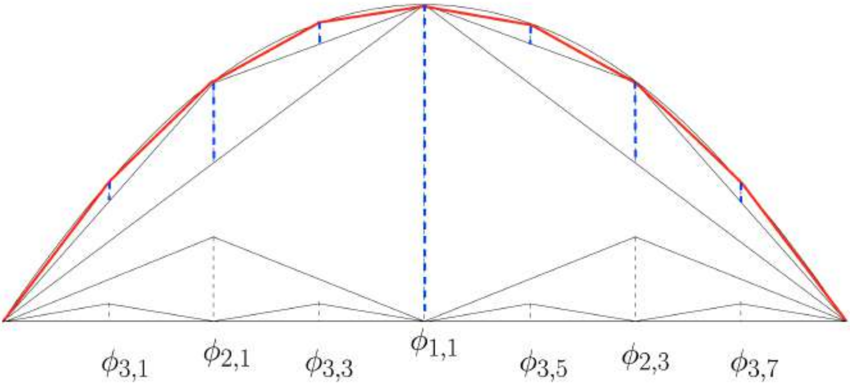


Figure: (Solid red line) the piecewise linear interplolant; (Dashed blue line) the absolute value of the coefficients $c_{l,i}$

One-dimensional example

Level 6 adaptive hierarchical interpolation with $\tau = 0.01$

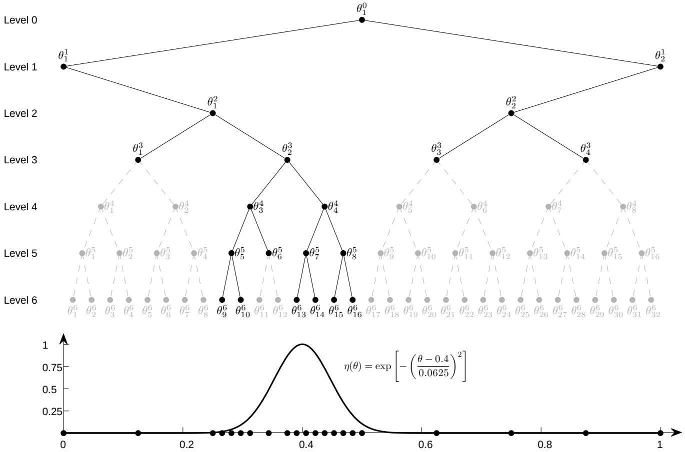


Figure: The resulting adaptive grid has 21 points (black points) whereas the full grid has 65 points (black and gray points)

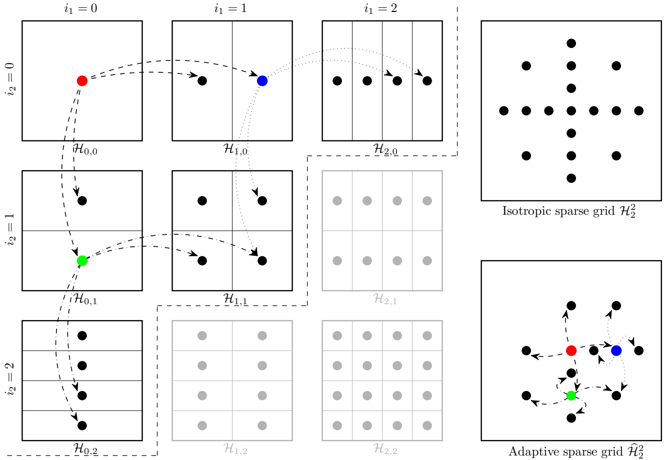
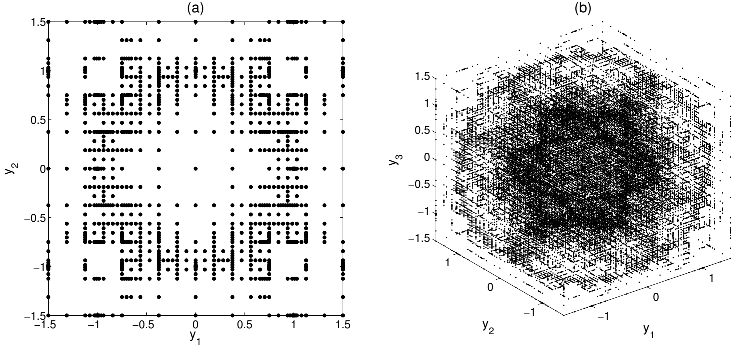


Figure: With adaptivity, each point that corresponds to a large surplus, e.g., the points in red, blue, or green, lead to 2 children points added in each direction resulting in the adaptive sparse grid

We approximate a characteristic function $g(\boldsymbol{y})$ with $\boldsymbol{y} = (y_1, \ldots, y_d)$ as

$$g(\boldsymbol{y}) = \begin{cases} 1, & \sqrt{y_1^2 + \cdots + y_d^2} \leq 1 \\ 0, & \text{otherwise} \end{cases}$$



- When extending the linear basis to a high-order basis, we would like to retain the partial delta property, i.e.,

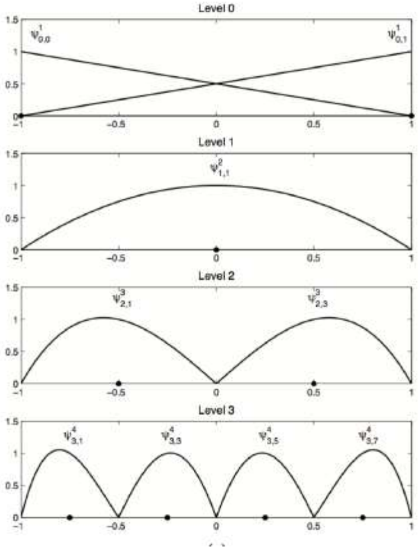
(P_1) New basis function on level L is equal to zero at all the grid points for $L' < L$

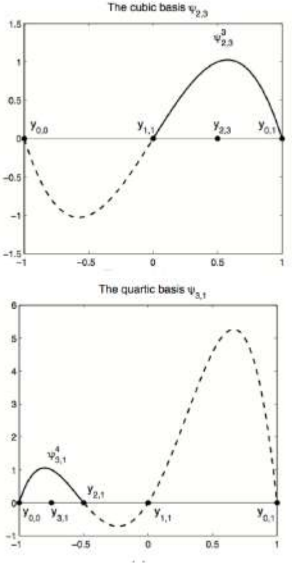
(P_2) New basis function on level L is equal to zero at all the grid points for $L' = L$

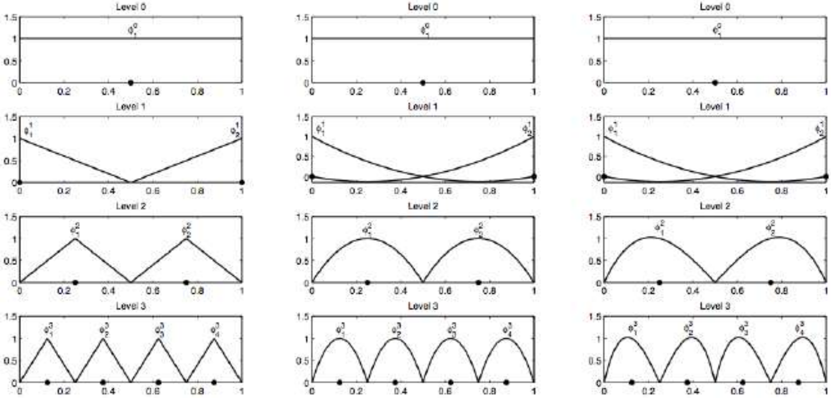
- High-order polynomials requires more grid points to define, 2nd-order needs 3 points, 3rd-order needs 4 points, etc.

- The idea is to use some points at lower levels plus one point on the current level to define the polynomial to satisfy (P_1), and then cut off part of the polynomial to satisfy (P_2).

High-order hierarchical polynomial bases for sparse grids
[Bungartz, Griebel, 04] [Gunzburger, Webster, Z, 14]






Level 0
Level 1
Level 2
Level 3

For a function $g \in H_{\text{mix}}^{p+1}$, the error of the sparse grid interpolant $\mathcal{I}_L^{\text{sg}}[g]$ is

$$\|g - \mathcal{I}_L^{\text{sg}}[g]\|_{L^2} = \mathcal{O}\left(h_L^{p+1} \log(h_L^{-1})^{d-1}\right).$$

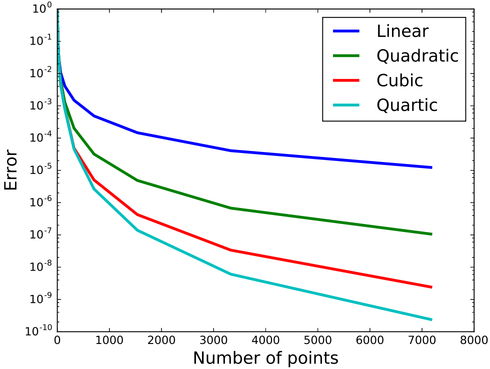
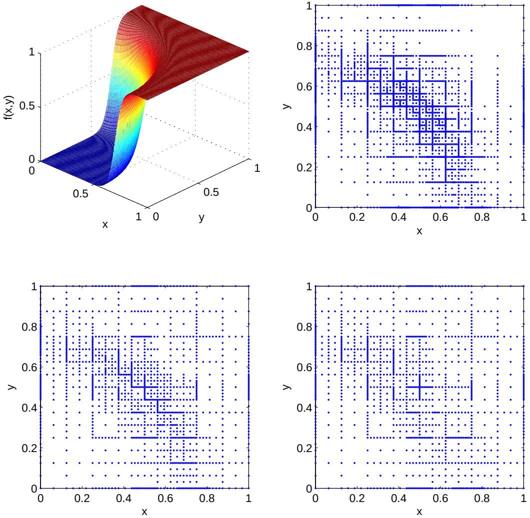


Figure: The error decay for a 2D function $g(\boldsymbol{y}) = \exp(-y_1^2 - y_2^2)$.

Adaptivity with high-order sparse-grid interpolation

linear, quadratic and cubic approximations with $tol = 10^{-3}$



- Motivation: The aforementioned hierarchical bases may have some stability issues when doing adaptivity.

- For example, the projection of the target function $g(\boldsymbol{y})$ in the subspace $\mathcal{V}_{\mathcal{J}_L^{\rm sg}}$ can be bounded from above, i.e.,

$$\|\mathcal{I}_L^{\rm sg}[g]\|_{L^2}^2 \leq C \sum_{l=0}^{L} \sum_{|\mathbf{l}|=l} \sum_{\mathbf{i}\in B_{\mathbf{l}}} |c_{\mathbf{l},\mathbf{i}}|^2 ,$$

  which may be an over estimate, meaning that a big coefficient may only contribute very little to the approximation.

- **Riesz basis**: there exists constants c and C independent of level L such that

$$c \sum_{l=0}^{L} \sum_{|\mathbf{l}|=l} \sum_{\mathbf{i}\in B_{\mathbf{l}}} |c_{\mathbf{l},\mathbf{i}}|^2 \leq \|\mathcal{I}_L^{\rm sg}[g]\|_{L^2}^2 \leq C \sum_{l=0}^{L} \sum_{|\mathbf{l}|=l} \sum_{\mathbf{i}\in B_{\mathbf{l}}} |c_{\mathbf{l},\mathbf{i}}|^2 .$$

- **Idea:** The lifting scheme is a process of taking an existing hierarchical basis and modifying it by adding linear combinations of hierarchical basis at the coarser level.

- The approximation space $V_l = \text{span}\{\psi_{l,i} | 0 \leq i \leq 2^l\}$ has a decomposition $V_l = V_{l-1} \oplus W_l$ where V_{l-1} and W_l are defined by

$$V_{l-1} = \text{span}\left\{\psi_{l-1,i} | 0 \leq i \leq 2^{l-1}\right\}, \quad W_l = \text{span}\left\{\psi_{l,i} | 0 \leq i \leq 2^l,\ i \text{ odd}\right\}.$$

- For any $\psi_{l,i} \in W_l$, the corresponding second-generation wavelet $\psi_{l,i}$ is constructed by "lifting" $\psi_{l,i}$ as

$$\psi_{l,i} \equiv \psi_{l,i} + \sum_{\hat{i}=0}^{2^{l-1}} \alpha_{\hat{i},i}^{l-1} \psi_{l-1,\hat{i}},$$

where the weights $\alpha_{\hat{i},i}^{l-1}$ in the linear combination are chosen in such a way that the new wavelet $\psi_{l,i}$ satisfies the Riesz property.

In the linear case, the second-generation wavelets are defined by

$$\phi_{l,i} = \psi_{l,i} - \frac{1}{4}\psi_{l-1,\frac{i-1}{2}} - \frac{1}{4}\psi_{l-1,\frac{i+1}{2}} \quad \text{for } 1 < i < 2^l - 1,\ i \text{ odd}$$
$$\phi_{l,i} = \psi_{l,i} - \frac{3}{4}\psi_{l-1,\frac{i-1}{2}} - \frac{1}{8}\psi_{l-1,\frac{i+1}{2}} \quad \text{for } i = 1,$$
$$\psi_{l,i} = \psi_{l,i} - \frac{1}{8}\psi_{l-1,\frac{i-1}{2}} - \frac{3}{4}\psi_{l-1,\frac{i+1}{2}} \quad \text{for } i = 2^l - 1,$$

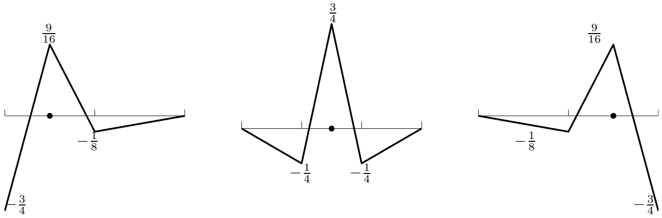
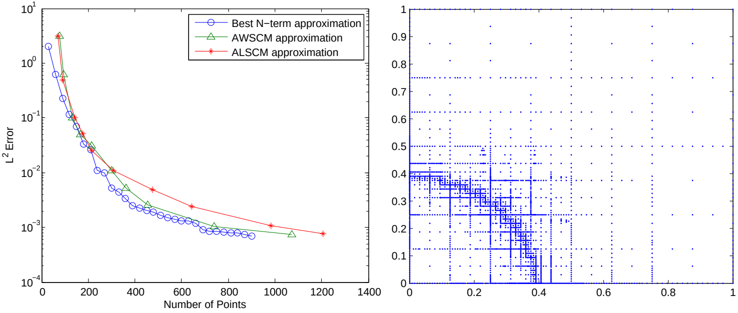


Figure: Left wavelet (left), central wavelet (middle), right wavelet (right)

- The target function is a bivariate function

$$f(x, y) = \frac{1}{|0.15 - x^2 - y^2| + 0.1}$$



- The L^2 error of the wavelet approximation is closer to that of the best N-term approximation parameters $\boldsymbol{y} \in \Gamma \subset \mathbb{R}^N$ $\longrightarrow$ **PDE model:** $\mathcal{L}(u, \boldsymbol{y}) = f$ in $D \subset \mathbb{R}^d$, $d = 1, 2, 3$ $\longrightarrow$ quantity of interest $Q[u(\cdot, \boldsymbol{y})]$

- The linear operator $\mathcal{L}$ depends on an q parameters $\boldsymbol{y} = (y_1, y_2, \ldots, y_q) \in \Gamma$, which can be deterministic or stochastic.
- The parameters $\boldsymbol{y}$ may be affected by uncertainty (measurement error, incomplete description of parameters), and are modeled as a random vector $\boldsymbol{y} : \Omega \to \Gamma$ with joint PDF $\varrho(\boldsymbol{y}) = \prod_{i=1}^{q} \varrho_i(y_i)$.
- Quantity of interest $F(\boldsymbol{y}) = F(u(\boldsymbol{y}))$ is a functional of u which may
  - be a smooth function of $\boldsymbol{y}$
  - have steep gradients with respect to $\boldsymbol{y}$
  - have dicontinuities with respect to $\boldsymbol{y}$

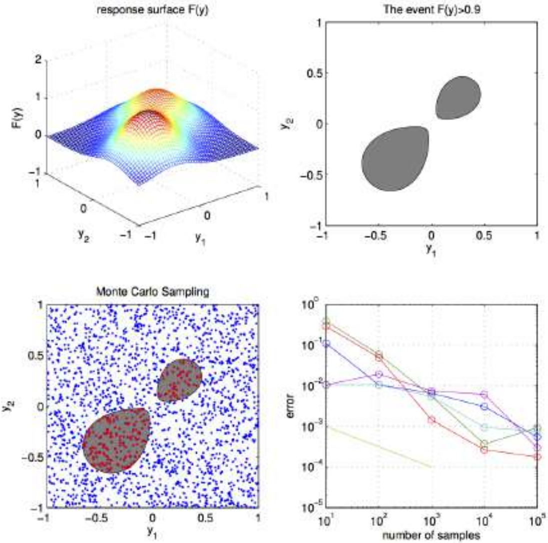
- For $F(\boldsymbol{y})$ with **discontinuities** in the parameter space Γ, we want to
  - identify the points of discontinuity
  - subdivide the geometry into subregions of smooth behavior
  - construct a piecewise approximation which is smooth over each subregion
- For any $F(\boldsymbol{y})$, **continuous or discontinuous**, the problem we want to solve is:
  - given the PDF $\rho(\boldsymbol{y})$ for the input parameter $\boldsymbol{y} \in \Gamma$
  - given the threshold F_0
  - given an output of interest $F(\boldsymbol{y}) = F(u(\boldsymbol{y}))$

Probability of the event $F(\boldsymbol{y}) \geq F_0$

$$\mathbb{P}\left[F(u(\boldsymbol{y})) \geq F_0\right] = \int_\Gamma \chi_{\{F(\boldsymbol{y}) \geq F_0\}}(\boldsymbol{y})\rho(\boldsymbol{y})d\boldsymbol{y}$$

Motivation: discontinuities from thresholds

a simple example



M	MC estimate	MC error
1	0.000000	0.110691
10	0.200000	0.089309
100	0.090000	0.020691
1,000	0.106000	0.004691
10,000	0.108300	0.002391
100,000	0.110430	0.000261
1,000,000	0.110564	0.000127
exact	0.110691	

- Monte Carlo is slow to converge
- lots of solutions of PDE are needed
- quadrature rules with global polynomial approximation do not work

We often begin with a bounded domain $\Gamma \subset \mathbb{R}^N$ but we are interested in a subdomain D which can only be described implicitly, e.g. by **a characteristic function** $f(\boldsymbol{y}) : \Gamma \to \mathbb{R}$ defined by

$$f(\boldsymbol{y}) = \begin{cases} 1, & \text{if } \boldsymbol{y} \in D \subset \Gamma, \\ 0, & \text{otherwise.} \end{cases}$$

- Can we detect the boundary ∂D of the discontinuous function $f(\boldsymbol{y})$?
- Can we accurately and efficiently estimate the integral:

$$\int_\Gamma f(\boldsymbol{y})\rho(\boldsymbol{y})d\boldsymbol{y} = \int_D \rho(\boldsymbol{y})d\boldsymbol{y}$$

- Our goals is to combat (we are not so ambitious as to believe we can beat it) the **curse of dimensionality** in
  - Building approximations to ∂D
  - Estimating the above integral faster than conventional Monte Carlo sampling

- adaptive triangle mesh refinement
- discontinous Galerkin methods
- Monte Carlo sampling
- polynomial annihilation
- adaptive hierarchical sparse grids

- In multi-D, an adaptive process (based on surpluses) is used to select a subset of the tensor product grid that is concentrated near the discontinuity surface.

- For discontinuous functions, the adaptive hierarchical sparse-grid method can incur very high cost, even in low dimensions, because
  - the sparse-grid interpolant does not converge in L^∞ norm, which means the surplus does not decay to zero
  - the adaptivity generates a dense grid around the discontinuity surface
  - many grid points do not contribute much to the approximation
  - high-order hierarchical basis functions are useless

We approximate a characteristic function $g(\boldsymbol{y})$ with $\boldsymbol{y} = (y_1, \ldots, y_d)$ as

$$g(\boldsymbol{y}) = \begin{cases} 1, & \sqrt{y_1^2 + \cdots + y_d^2} \leq 1 \\ 0, & \text{otherwise} \end{cases}$$

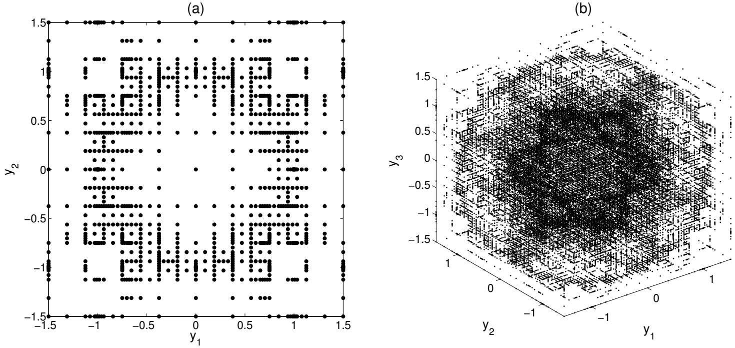


Figure: 2D adaptive sparse grid requires 5,925 points; 3D adaptive sparse-grid requires 21,501 points

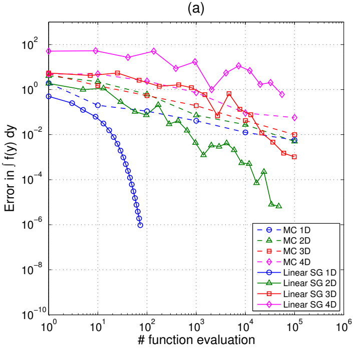


Figure: The error is measured by $\left| \int f(\boldsymbol{y}) d\boldsymbol{y} - \int \mathcal{I}_L^{\text{sg}}[f](\boldsymbol{y}) d\boldsymbol{y} \right|$

- Consider a bounded domain $\Gamma \subset \mathbb{R}^N$ and a characteristic function $f(\boldsymbol{y}) : \Gamma \to \mathbb{R}$ defined by

$$f(\boldsymbol{y}) = \begin{cases} 1, & \text{if } \boldsymbol{y} \in D \subset \Gamma, \\ 0, & \text{otherwise} \;, \end{cases}$$

  - D is the characteristic domain
  - ∂D is the discontinuity surface described by an implicit equation $G(\boldsymbol{y}) = 0$ in Γ. For example, a hyper-sphere can be represented by $G(\boldsymbol{y}) \equiv \sum_{n=1}^{N} y_n^2 - \lambda^2 = 0$

- The goal is to find two bounded domains D_1 and D_2 such that
  - $D_1 \subset D \subset D_2 \subset \Gamma$
  - $\text{dist}\,(\partial D_1, \partial D_2) \le \varepsilon$

  where ε is a prescribed accuracy.

- It is easy to see that $f(\boldsymbol{y}) = 0$ for $\boldsymbol{y} \in \partial D_2$ and $f(\boldsymbol{y}) = 1$ for $\boldsymbol{y} \in \partial D_1$

We put the following assumption about the domain $D \in \Gamma$ of interest:

Assumption

Assume that D is a star-convex domain in Γ and a point $\boldsymbol{y}_0$ in D is given such that for all $\boldsymbol{y}$ in D, the line segment $\{\boldsymbol{y}_0 + t\boldsymbol{y} | t \in [0, 1]\}$ from $\boldsymbol{y}_0$ to $\boldsymbol{y}$ is in D.

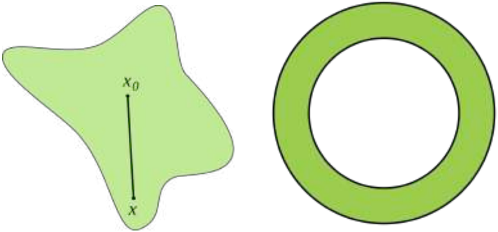


Figure: (left) A star-convex domain is not necessarily convex; (right) An annulus is not a star-convex domain (The two figures are from Wikipedia)

A hyper-spherical coordinate system is a generalization of the 2D polar and 3D spherical coordinate systems

- one radial coordinate r ranging over $[0, +\infty)$
- one angular coordinate θ_{N-1} ranging over $[0, 2\pi)$
- $N-2$ angular coordinates $\theta_1, \ldots, \theta_{N-2}$ ranging over $[0, \pi)$

Hyper-spherical coordinates are converted Cartesian coordinates by

$$
\begin{aligned}
y_1 &= y_{0,1} + r\cos(\theta_1) \\
y_2 &= y_{0,2} + r\sin(\theta_1)\cos(\theta_2) \\
y_3 &= y_{0,3} + r\sin(\theta_1)\sin(\theta_2)\cos(\theta_3) \\
&\vdots \\
y_{N-2} &= y_{0,N-2} + r\sin(\theta_1)\cdots\sin(\theta_{N-2})\cos(\theta_{N-1}) \\
y_{N-1} &= y_{0,N-1} + r\sin(\theta_1)\cdots\sin(\theta_{N-2})\sin(\theta_{N-1})
\end{aligned}
$$

- Transform the Cartesian coordinates $y_1, \ldots, y_N$ to the hyper-spherical coordinates $r, \theta_1, \ldots, \theta_{N-1}$ with the given origin point $\boldsymbol{y}_0$.

- Each point $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_{N-1})$ corresponds to a ray in $\mathbb{R}^N$ out from $\boldsymbol{y}_0$ in a specific direction

- Due to the star-convexity of the domain D, there is only one jump discontinuity in each direction $\boldsymbol{\theta}$

- ∂D can be represented by a function $r = g(\boldsymbol{\theta})$ on the bounded N-1 dimensional domain

$$\Gamma_\theta = \prod_{n=1}^{N-1} [0, \pi] \times [0, 2\pi]$$

  where for any $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_{N-1}) \in \Gamma_\theta$, $(g(\boldsymbol{\theta}), \boldsymbol{\theta})$ is on ∂D.

- Build an L-level sparse grid $\mathcal{H}_L^{N-1}$ on Γ_θ with a total of M grid points.

$$\mathcal{H}_L^{N-1} = \{\boldsymbol{\theta}_i \in \Gamma_\theta, \text{ for } i = 1, \ldots, M\}$$

- For an accuracy tolerance ε and for $m = 1, \ldots, M$, from $\boldsymbol{y}_0$, along the direction corresponding $\boldsymbol{\theta}_m \in \mathcal{H}_L^{N-1}$, use 1-D bisection method to find two values g_m^1 and g_m^2 such that

$$g_m^1 \leq g(\boldsymbol{\theta}_m) \leq g_m^2 \qquad \text{and} \qquad |g_m^1 - g_m^2| \leq \varepsilon$$

- Build sparse-grid interpolants $g^1(\boldsymbol{\theta})$ and $g^2(\boldsymbol{\theta})$ based on $\{g_m^1, m = 1, \ldots, M\}$ and $\{g_m^2, m = 1, \ldots, M\}$, respectively. Then we have

$$\begin{aligned} \left(g^1(\boldsymbol{\theta}), \boldsymbol{\theta}\right) &\quad \Longrightarrow \quad \partial D_1 \\ \left(g^2(\boldsymbol{\theta}), \boldsymbol{\theta}\right) &\quad \Longrightarrow \quad \partial D_2 \end{aligned}$$

- $\#$ function evaluations $= \sum_{m=1}^{M} \#$ bisection trials for $\boldsymbol{\theta}_m$

- According to smoothness of the hyper-surface ∂D, different types of basis functions can be used, e.g. high-order hierarchical basis or wavelet basis

Consider the two characteristic functions in $\mathbb{R}^N$

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

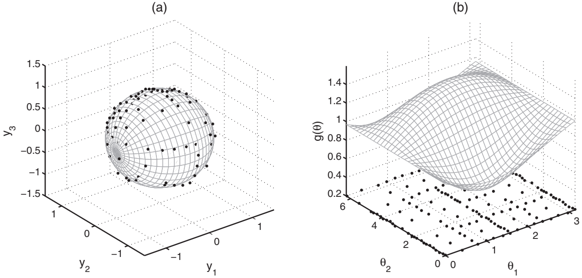


Figure: (a) The discontinuity surface γ with sparse grid points; (b) the transformed surface $g(\boldsymbol{\theta})$ in the hyperspherical coordinate system. The parameters for the SG approximation are $L_{\min} = 4$, $L_{\max} = 12$, $\alpha = 0.01$, and $\boldsymbol{y}_0 = (0.1, 0.2, 0.3)$; the total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

Consider the two characteristic functions in $\mathbb{R}^N$

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

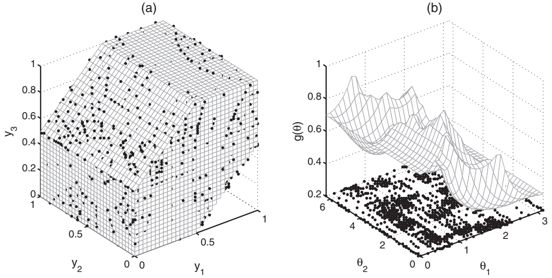


Figure: (a) The discontinuity surface γ with sparse grid points; (b) the transformed surface $g(\boldsymbol{\theta})$ in the hyperspherical coordinate system. The parameters for the SG approximation are $L_{\min} = 4$, $L_{\max} = 12$, $\alpha = 0.01$, and $\boldsymbol{y}_0 = (0.3, 0.4, 0.5)$; the total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

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

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

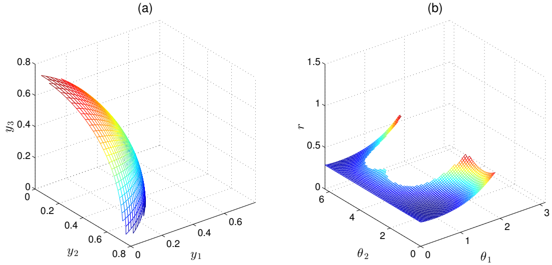
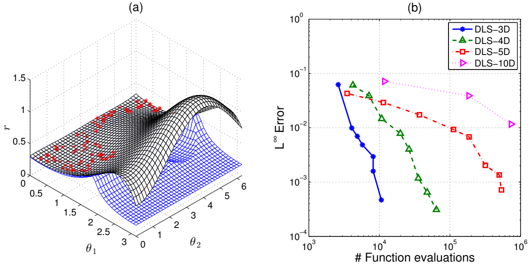
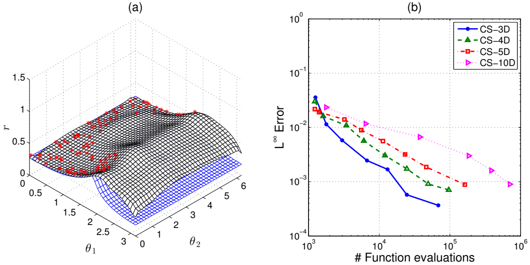


Figure: (a) The discontinuity surface γ in the Cartesian system; (b) the transformed surface $g(\boldsymbol{\theta})$ in the hyperspherical system.


(a)
1.5
1
0.5
0
r
0
0.5
1
1.5
2
2.5
3
θ_1
0
1
2
3
4
5
6
θ_2
(b)
DLS–3D
DLS–4D
DLS–5D
DLS–10D
L^∞ Error
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^3
10^4
10^5
10^6
Function evaluations


(a)
1.5
1
0.5
0
r
0
0.5
1
1.5
2
2.5
3
θ_1
0
1
2
3
4
5
6
θ_2
(b)
CS–3D
CS–4D
CS–5D
CS–10D
L^∞ Error
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^3
10^4
10^5
10^6
Function evaluations

- At each collocation point $\boldsymbol{y}_{\mathbf{l},\mathbf{i}}$, $u_{N_h}(x, \boldsymbol{y}_{\mathbf{l},\mathbf{i}})$ is approximated based on the solution from the selected linear system solver, i.e.

$$u_{N_h}(x, \boldsymbol{y}_{\mathbf{l},\mathbf{i}}) = \sum_{j=1}^{N_h} u_{j,\mathbf{l},\mathbf{i}} \phi_j(x) \approx \widetilde{u}_{N_h}(x, \boldsymbol{y}_{\mathbf{l},\mathbf{i}}) = \sum_{j=1}^{N_h} \widetilde{u}_{j,\mathbf{l},\mathbf{i}} \phi_j(x)$$

  where $\widetilde{\boldsymbol{u}}_{\mathbf{l},\mathbf{i}} = (\widetilde{u}_{1,\mathbf{l},\mathbf{i}}, \ldots, \widetilde{u}_{N_h,\mathbf{l},\mathbf{i}})^\top$ is the output of the solver.
- In the case of using conjugate gradient methods, the error $\boldsymbol{e}^k_{\mathbf{l},\mathbf{i}} = \boldsymbol{u}_{\mathbf{l},\mathbf{i}} - \boldsymbol{u}^k_{\mathbf{l},\mathbf{i}}$ is bounded by

$$\|\boldsymbol{e}^k_{\mathbf{l},\mathbf{i}}\|_{\boldsymbol{A}_{\mathbf{l},\mathbf{i}}} \leq 2 \left(\frac{\sqrt{\kappa_{\mathbf{l},\mathbf{i}}} - 1}{\sqrt{\kappa_{\mathbf{l},\mathbf{i}}} + 1} \right)^k \|\boldsymbol{e}^0_{\mathbf{l},\mathbf{i}}\|_{\boldsymbol{A}_{\mathbf{l},\mathbf{i}}}$$

- We describe the total computational cost for constructing $\widetilde{u}_{N_h,M_L} \approx u_{N_h,M_L}$ is represented by

$$\mathcal{C}_{\text{total}} = \sum_{l=0}^{L} \sum_{|\mathbf{l}|=l} \sum_{\mathbf{i} \in B_{\mathbf{l}}} \mathcal{M}_{\mathbf{l},\mathbf{i}}$$

  where $\mathcal{M}_{\mathbf{l},\mathbf{i}}$ is the number of iterations needed at the collocation point $\boldsymbol{y}_{\mathbf{l},\mathbf{i}}$.

- The approximation $\widetilde{u}_{N_h,M_L}(x,\boldsymbol{y})$ can be represented in a hierarchical manner,

$$\widetilde{u}_{N_h,M_L}(x,\boldsymbol{y}) = \widetilde{u}_{N_h,M_{L-1}}(x,\boldsymbol{y}) + \sum_{g(\mathbf{l})=L} \sum_{\mathbf{i}\in B_\mathbf{l}} \widetilde{c}_{\mathbf{l},\mathbf{i}}(x) \cdot \psi_{\mathbf{l},\mathbf{i}}(\boldsymbol{y})$$

- At each collocation point $\boldsymbol{y}_{\mathbf{l},\mathbf{i}}$ on level L, $\boldsymbol{u}_{\mathbf{l},\mathbf{i}} = (u_{1,\mathbf{l},\mathbf{i}}, \ldots, u_{N_j,\mathbf{l},\mathbf{i}})^\top$ can be represented by

$$u_{j,\mathbf{l},\mathbf{i}} = u_{N_h,M_{L-1}}(x_j, \boldsymbol{y}_{\mathbf{l},\mathbf{i}}) + c_{j,\mathbf{l},\mathbf{i}}, \quad \text{for} \quad j = 1, \ldots, N_h$$

Key idea

Due to the decay of $|c_{j,\mathbf{l},\mathbf{i}}|$ as $|\mathbf{l}| \to \infty$, the initial guess for the CG solver is given by

$$\widetilde{\boldsymbol{u}}^0_{\mathbf{l},\mathbf{i}} = \left(\widetilde{u}_{N_h,M_{L-1}}(x_1, \boldsymbol{y}_{\mathbf{l},\mathbf{i}}), \ldots, \widetilde{u}_{N_h,M_{L-1}}(x_{N_h}, \boldsymbol{y}_{\mathbf{l},\mathbf{i}})\right)^\top$$

where the error of such prediction is, for $j = 1, \ldots, N_h$,

$$\left|\widetilde{u}^0_{j,\mathbf{l},\mathbf{i}} - u(x_j, \boldsymbol{y}_{\mathbf{l},\mathbf{i}})\right| \leq \left|\widetilde{u}_{N_h,M_{L-1}}(x_j, \boldsymbol{y}_{\mathbf{l},\mathbf{i}}) - u_{N_h,M_{L-1}}(x_j, \boldsymbol{y}_{\mathbf{l},\mathbf{i}})\right| + c_{j,\mathbf{l}\,\mathbf{i}}$$

- Without hierarchical acceleration, $\tau_0 = \mathcal{O}(\|u\|_\infty)$, so that the minimum cost $\mathcal{C}_{\min}$ to achieve $\|e\|| \leq \varepsilon$ can be bounded by

$$\mathcal{C}_{\min} \leq |\mathcal{H}_L(\Gamma)| \cdot J(\tau_0, \varepsilon, \overline{\kappa}, L_k, N)$$

whose estimate is given as follows:

Theorem [Gunzburger, Webster, Z, 14], complexity without hierarchical acceleration

The minimum cost $\mathcal{C}_{\min}$ for building the standard piecewise linear SG approximation $\widetilde{u}_{N_h, M_L}(x, \boldsymbol{y})$ with the prescribed accuracy $\varepsilon > 0$ can be bounded by

$$\mathcal{C}_{\min} \leq \frac{\alpha_1}{N}\left[\alpha_2 + \alpha_3 \frac{\log_2\left(\frac{3C_{\text{sg}}}{\varepsilon}\right)}{N}\right]^{\alpha_4 N}\left(\frac{3C_{\text{sg}}}{\varepsilon}\right)^{\alpha_5}$$
$$\times \frac{1}{\log_2\left(\frac{\sqrt{\overline{\kappa}}+1}{\sqrt{\overline{\kappa}}-1}\right)}\left[\alpha_6 \log_2\left(\frac{3C_{\text{sg}}}{\varepsilon}\right) + \log_2(\sqrt{\overline{\kappa}}\tau_0) + \alpha_7 N + \alpha_8\right],$$

where the constants $\alpha_1, \ldots, \alpha_8$ are independent of L, N and ε.

- With hierarchical acceleration, $\tau_0^l \leq C_{\mathsf{sg}} 2^{-2l} + 2^N e_{\mathsf{cg}}$ for $l = 1, \ldots, L$, so that the minimum cost $\mathcal{C}_{\min}$ to achieve $\|e\|| \leq \varepsilon$ can be bounded by

$$\mathcal{C}_{\min} \leq \sum_{l=0}^{L_k} |\Delta \mathcal{H}_l(\Gamma)| \cdot J(\tau_0^l, \varepsilon, \overline{\kappa}, L_k, N)$$

whose estimate is given as follows:

Theorem [Gunzburger, Webster, Z, 14], complexity with hierarchical acceleration

The minimum cost $\mathcal{C}_{\min}$ for building the standard piecewise linear SG approximation $\widetilde{u}_{N_h, M_L}(x, \boldsymbol{y})$ with the prescribed accuracy $\varepsilon > 0$ can be bounded by

$$\mathcal{C}_{\min} \leq \alpha_1 \left[\alpha_2 + \alpha_3 \frac{\log_2\left(\frac{2C_{\mathsf{sg}}}{\varepsilon}\right)}{N} \right]^{\alpha_4 N} \left(\frac{2C_{\mathsf{sg}}}{\varepsilon}\right)^{\alpha_5}$$
$$\times \frac{1}{\log_2\left(\frac{\sqrt{\overline{\kappa}}+1}{\sqrt{\overline{\kappa}}-1}\right)} \left[2N - \log_2(N) + \alpha_9 + \log_2(\sqrt{\overline{\kappa}}) \right],$$

where the constants $\alpha_1, \ldots, \alpha_5$ and α_9 are independent of L, N and ε.

We consider the 2D Poisson equation with random diffusivity and forcing term, i.e.,

$$\begin{cases} \nabla \cdot (a(x, \boldsymbol{y}) \nabla u(x, \boldsymbol{y})) = f(x, \boldsymbol{y}) & \text{in} \quad [0,1]^2 \times \Gamma, \\ u(x, \boldsymbol{y}) = 0 \quad \text{on} \quad \partial D \times \Gamma, \end{cases}$$

where a and f arethe nonlinear functions of the random vector $\boldsymbol{y}$ given by

$$a(x, \boldsymbol{y}) = 0.1 + \exp\left[y_1 \cos(\pi x_1) + y_2 \sin(\pi x_2)\right],$$

and

$$f(x, \boldsymbol{y}) = 10 + \exp\left[y_3 \cos(\pi x_1) + y_4 \sin(\pi x_2)\right],$$

where y_n for $n = 1, 2, 3, 4$ are i. i. d. random variables following the uniform distribution $U([-1, 1])$. The quantity of interest is the mean value of the solution over $D \times \Gamma$, i.e.

$$\text{QoI} = \mathbb{E}\left[\int_D u(x, \boldsymbol{y}) dx\right],$$

Table: The computational savings of the piecewise SG approach with hierarchical acceleration

Basis type	Error	# SG points	hSGSC	hSGSC+acceleration	
			cost	cost	saving
Linear	1.0e-2	377	13,841	7,497	45.8%
	1.0e-3	1,893	81,068	38,670	52.2%
	1.0e-4	7,777	376,287	167,832	55.3%
Quadratic	1.0e-3	701	29,874	11,877	60.2%
	1.0e-4	2,285	110,744	36,760	66.8%
	1.0e-5	6,149	329,294	100,420	69.5%
Cubic	1.0e-4	1,233	59,344	23,228	60.8%
	1.0e-5	3,233	172,845	57,777	66.5%
	1.0e-6	7,079	415,760	129,433	68.8%

TASMANIAN

Toolkit for Adaptive Stochastic Modeling and Non-Intrusive ApproximatioN
ORNL Laboratory Directed Research and Development
DoE: Office for Advanced Scientific Computing Research

- Download at http://tasmanian.ornl.gov
- Global and Local hierarchical basis functions;
- Arbitrary order local polynomial basis;
- C++ library and CLI and MATLAB interfaces;
- Different types of local refinement techniques

References

- A. Cohen and R. DeVore, *Approximation of high-dimensional parametric PDEs*, **Acta Numerica**, (24) 2015, pp. 1–159.
- H. Bungartz and M. Griebel, *Sparse grids*, **Acta Numerica**, Vol. 13, pp. 147–269, 2004.
- G. Zhang, C. Webster, M. Gunzburger and J. Burkardt, *Hyperspherical sparse approximation techniques for high-dimensional discontinuity detection*, **SIAM Review**, 58 (2016), pp. 517-551.
- G. W Wasilkowski, H. Wożniakowski, *Explicit cost bounds of algorithms for multivariate tensor product problems*, **J. Complexity**, Vol. 11, pp. 1–56, 1995
- D. Pflüger, B. Peherstorfer, and H-J. Bungartz, *Spatially adaptive sparse grids for high-dimensional data-driven problems.*, **J Complexity**, 2010 vol. 26 (5) pp. 508-522.
- M. Hegland, *Adaptive sparse grids*, **ANZIAM J**, 2002 vol. 44 (C) pp. C335-C353.
- T. Gerstner and M. Griebel, *Dimension-Adaptive Tensor-Product Quadrature*, **Computing**, 2003 vol. 71 (1) pp. 65-87.
- G. Zhang, C. Webster, M. Gunzburger, J. Burkardt, *A Hyper-Spherical Adaptive Sparse-Grid Approach for High-Dimensional Discontinuity Detection*, **SIAM Journal of Numerical Analysis**, 53 (2015), pp. 1508-1536.
- M. Gunzburger, C. Webster G. Zhang, *An Adaptive Wavelet Stochastic Collocation Method for Irregular Solutions of PDEs with Random Input Data*, **Springer Lecture Notes on CS&E**, Vol. 97, pp. 137–170, 2014
- M. Gunzburger, C. Webster, G. Zhang, *Stochastic finite element methods for partial differential equations with random input data*, **Acta Numerica**, Vol. 23, pp. 521–650, 2014