Adaptive Sparse Grids

Seminar High Dimensional Methods in Scientific Computing

Muhammed Kürşat Yurt
TUM School of Computation, Information and Technology
Technische Universität München Email: kursat.yurt@tum.de

***Abstract*—**

***Index Terms*— Sparse Grid,Interpolation,Multivariate, Multidimension,Adaptivity,Interpolation**

I. Introduction

Ever increasing internet speed increased the data produced all around the world, that leads to boom in Machine Learning and Data Analytics fields. Which are generally have many dimensional large data sets. As like all high dimensional problems, they are suffer from curse of dimensionality, i.e. they have an exponential dependency on dimension. This is a barrier in numerical treatment of the high-dimensonal problems. This exponential dependency makes harder to use classical mesh based approaches to solve this kind of problems. One could also use mesh-free methods like Monte-Carlo quadratures.

In order to overcome such a problem, the sparse grid method gains more and more popularity. The sparse grid method is a general numerical discretization technique which is first introduced by the Russian mathematician Smolyak in 1963 [1].

Sparse grids offers a new way to reduce the required number of grid points by the order of magnitude $O(2^{nd})$ to just only $O(2^n n^{d-1})$ while preseving a similar error as using the full grid [2]. A comparision of storage requirement and error is listed in table I. In order to achieve these bounds, the mixed second derivatives have to be bounded.

The sparse grid uses a hierarchical formulation as shown in fig. 1 for one dimensional case. It has an incremental and adaptive behaviour inherently. In order to extend to a general d-dimensional setting it expolits tensor prodoct approach.

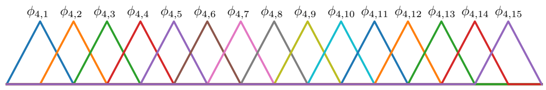
For the problems which are do not satisfy smoothness criteria or required further reduce in mesh size, one can use advantage of adaptivity. The hierarchical basis is a direct indicator of areas where further refinement required.

TABLE I: Comparision of Sparse and Full Grid Approaches.

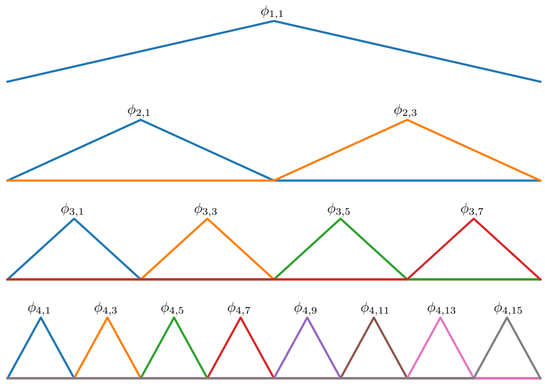
	Storage Requirement	L2 Norm of Interpolation Error
Full Grid	$O(2^{nd})$	$O(2^{-2n})$
Sparse Grid	$O(2^n n^{d-1})$	$O(2^{-2n} n^{d-1})$

In this work we will use standard hat function given by eq. (1).

$$\phi(x) = \begin{cases} 1-|x| & \text{if } x \in [-1,1], \\ 0 & \text{otherwise} \end{cases} \tag{1}$$



(a) Nodal Basis



(b) Hiearchical Basis

Fig. 1: Comparision of piecewise linear basis functions.

On a equidistant grid Ω_l of level l on a unit interval $\bar{\Omega} = [0,1]$. The mesh width is given by 2^{-l}. The grid points on a certain level is given bibliography

$$x_{l,i} = i \cdot h_l, 0 \leq i \leq 2^l \tag{2}$$

Using eq. (1) a family of basis functions $\phi_{l,i}(x)$ with a support of $[x_{l,i} - h_l, x_{l,i} + h_l]$, by dilation and translation one could get eq. (4). This process gives all possible basis function in level l as shown in fig. 1a.

$$\phi_{l,i}(x) = \phi\left(\frac{x - i \cdot h_l}{h_l}\right) \tag{3}$$

$$V_l = \text{span}\left\{\phi_{l,i} : 1 \leq i \leq 2^l - 1\right\} \tag{4}$$

One need hierarchical ones in order to construct the sparse grid. The hierarchical increment spaces are given by eq. (5).

$$W_l = \text{span}\left\{\phi_{l,i} : i \in I_l\right\} \tag{5}$$

where the index set is,

$$I_l = \left\{i \in \mathbb{N} : 1 \leq i \leq 2^l - 1, i \text{ odd}\right\} \tag{6}$$

Using the resulting basis functions as input to the tensor product construction, one can obtain suitable piecewise d-linear basis function at each grid point $x_{l,i}$

$$\phi_{l,i}(x) = \prod_{j=1}^{d} \phi_{l_j,i_j}(x_j) \tag{7}$$

II. Adaptivity

A regular sparse grid is constructed in a way that takes a cut in diagonal hyperplane, which means it treats all the dimensions equally. However, there can be an importance difference between the dimensions, i.e. one dimension might be more important than others. This can be solved by so called dimensional adaptivity [3].

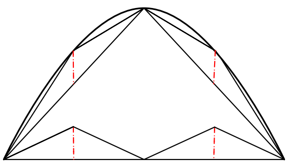


Fig. 2: Interpolation of a par abola using 2 level hierarchical basis and surpluses, surpluses are shown in red lines.

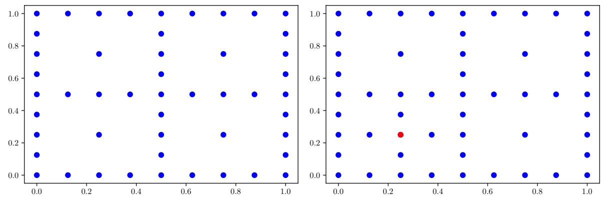
The most straightforward approach for this type of refinement is adding some new subspaces in the dimension which function changes rapidly. In order to add a new subspace $W_{\vec{l}}$ one should include all the backward neighbours in the current set of subspaces. This refiment treads all the grid points in one dimension as a uniform way,and called as dimensionally adaptive refinement. Leads more point in one dimension than other one.

Morever, there are some cases exist, where dimensional adaptivity is not to be enough to solve problem. For instance take a function that is mostly flat, have peaks at certain regions in the domain. Franke's function eq. (8) is a good example for this [4], which will be used in section III.

$$\begin{aligned} f(x_1,x_2) &= \frac{3}{4}\exp\left(-\frac{(9x_1-2)^2}{4}-\frac{(9x_2-2)^2}{4}\right) \\ &+\frac{3}{4}\exp\left(-\frac{(9x_1+1)}{49}-\frac{9x_2+1}{10}\right) \\ &+\frac{1}{2}\exp\left(-\frac{(9x_1-7)^2}{4}-\frac{(9x_2-3)^2}{4}\right) \\ &-\frac{1}{5}\exp\left(-(9x_1-4)^2-(9x_2-7)\right) \end{aligned} \tag{8}$$

A dimensionally adaptive grid would also add more point to regions where the function is mostly flat. A spatially adaptive grid [5] would overcome this problem. Instead of adding whole incremental grids, spatial adaption would only include a subset of those points near the region of interest and saves points.

The general approach for spatially adaptive grids is doing the refinement process iteratively. One could start and initial coarse grid or if knows a grid which is tailored to problem. Using an iterative process one could simply adds new points (neighboring grid point in the next higher level) to region of interest. To enable usage of sparse grids algorithms there is also a consistency constraint exist, the grid should contain all the hierarchical ancestors of all grid points. This may lead, the number of point which are added might be larger than $2 \cdot d$ Figure 3 shows how this process is done in two-dimensional regular sparse grid for one refinement step.



(a) Regular grid. (b) The refined grid.

Fig. 3: Spatial refinement of a regular level 3 grid. Refinement point is marked with red.

The choice of adaptivity criteria to choose the grid points which will be refined is defined by user. One of the most popular criterion is the surplus-based criterion, which used in example section in this paper. This criterion uses absolute value of the hierarchical surpluses. It based on the assumption that a larger absolute surplus corresponds a larger second derivatives.

III. Examples

In this section using SG++ software package [5], interpolation operation on sparse grids are tested for selection functions namely Franke's function and Genz test functions.

A. Franke's Function

Franke's function is a well known test function for interpolation. It has two Gaussian peaks of different heights, and a smaller dip. The function is illustrated in fig. 4a.

The operation is started with a level 3 regular sparse grid which can be seen in fig. 10a which has 49 grid points initially. The refinement has been done before having approximately 400 grid points, which roughly takes 10 refinement step when every step only one grid point is selected for refinement. The final refined grid is shown in fig. 5.

B. Genz Test Functions

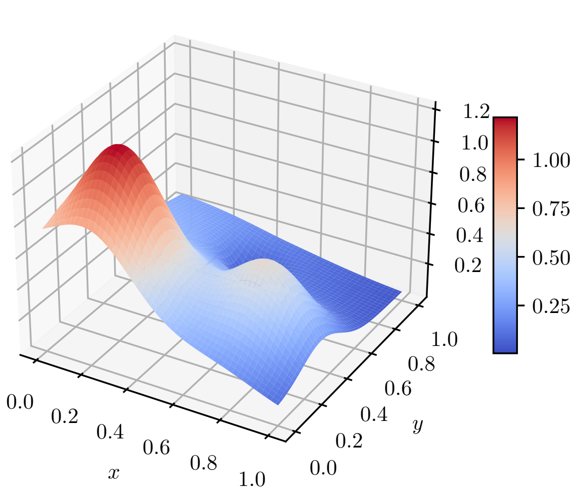
A set of test functions are taken from [6].

$$f_{disc}(x) = \begin{cases} 0 & \text{for} x \geq 0.2, \\ \exp\left(-\sum_{i=1}^{d} i \cdot x_i\right) & \text{otherwise} \end{cases} \tag{9}$$

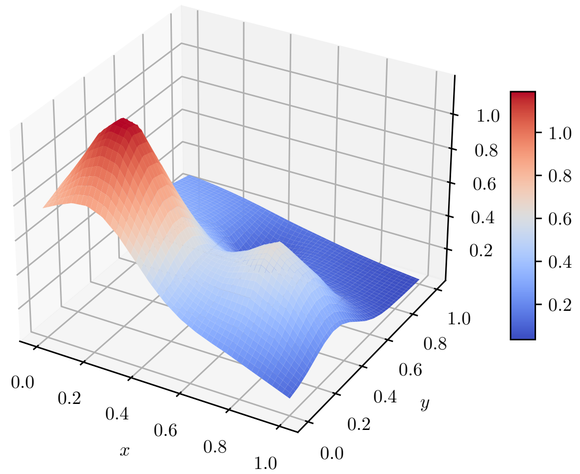
$$f_{prod}(x) = \frac{10^{-d}}{\prod_{i=1}^{d} (10i)^{-2} + (x_i - 0.99)^2} \tag{10}$$

1) Discontinuous Function:

2) Product Peak Function:



(a) Regular grid.



(b) The refined grid.

Fig. 4: Spatial refinement of a regular level 3 grid. Refinement point is marked with red.

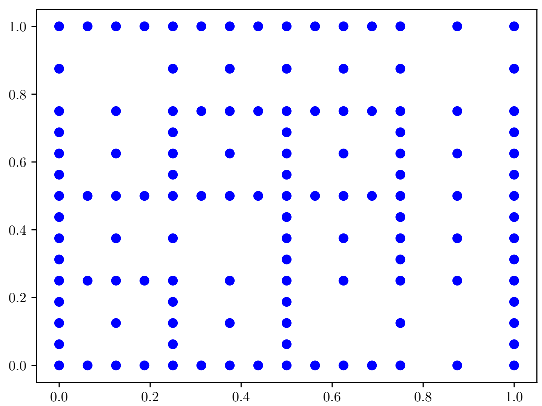


Fig. 5: The final sparse grid with 400 grid points.

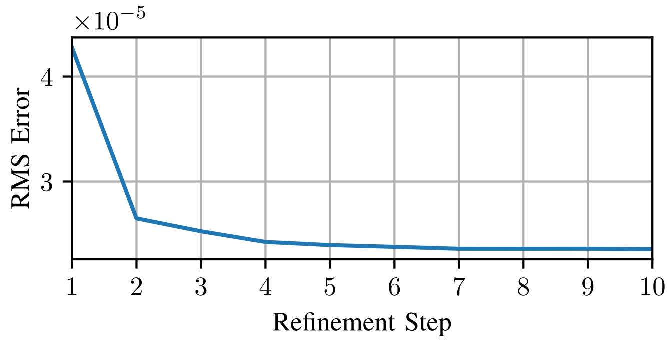
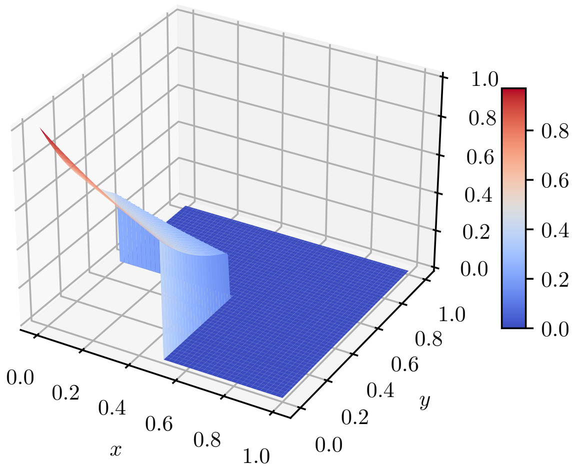
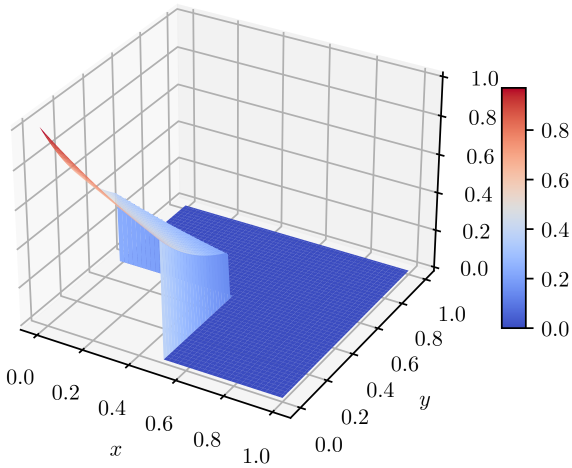


Fig. 6: Error reduction plot of Franke's function w.r.t. refinement steps.



(a) Regular grid.



(b) The refined grid.

Fig. 7: Spatial refinement of a regular level 3 grid. Refinement point is marked with red.

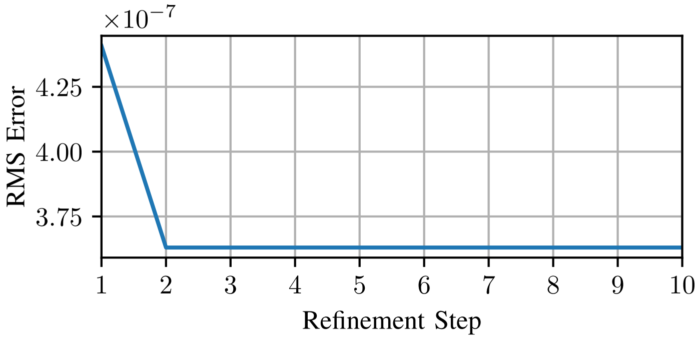


Fig. 8

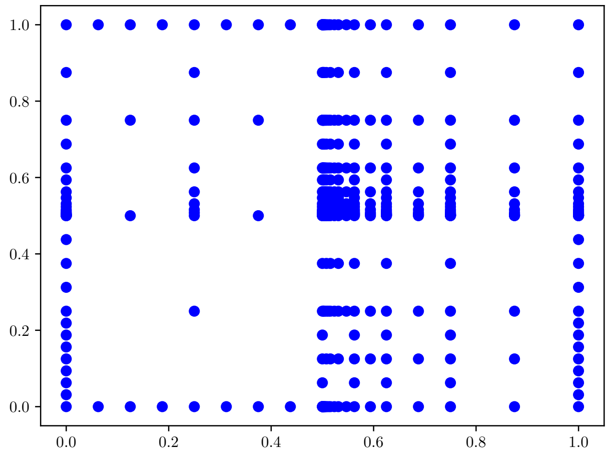
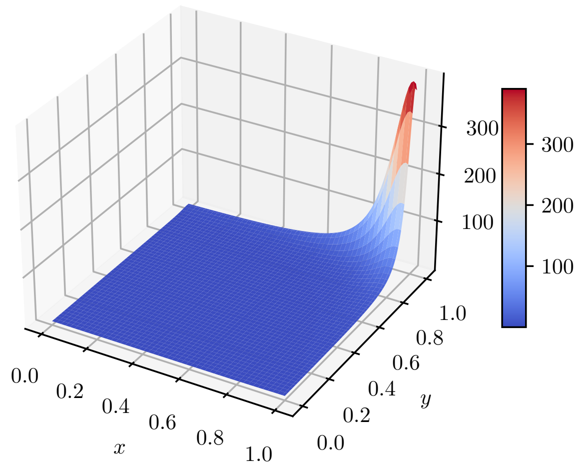
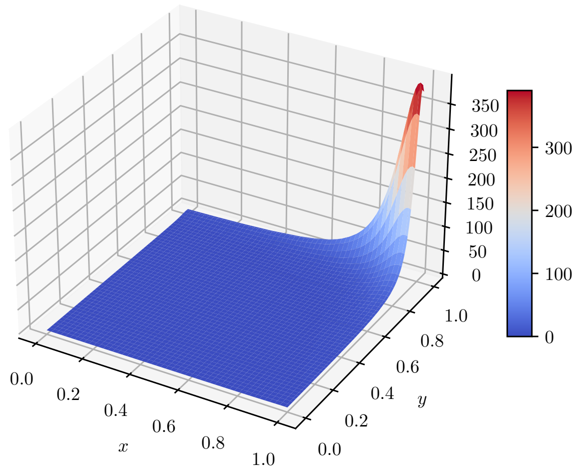


Fig. 9



(a) Regular grid.



(b) The refined grid.

Fig. 10: Spatial refinement of a regular level 3 grid. Refinement point is marked with red.

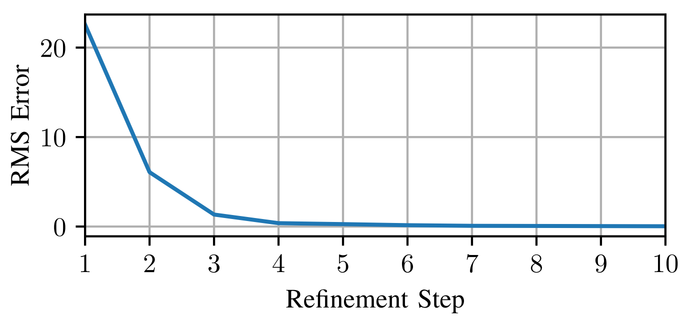


Fig. 11

References

[1] S. A. Smolyak, “Quadrature and interpolation formulas for tensor products of certain classes of functions,” in *Doklady Akademii Nauk*, vol. 148, no. 5. Russian Academy of Sciences, 1963, pp. 1042–1045.

[2] J. Garcke, “Sparse grids in a nutshell,” in *Lecture Notes in Computational Science and Engineering*. Springer Berlin Heidelberg, 2012, pp. 57–80. [Online]. Available: http://doi.org/10.1007/978-3-642-31703-3_3

[3] T. Gerstner and M. Griebel, “Dimension adaptive tensor product quadrature,” *Computing*, vol. 71, no. 1, pp. 65–87, Aug. 2003. [Online]. Available: https://doi.org/10.1007/s00607-003-0015-5

[4] R. Franke, “A critical comparison of some methods for interpolation of scattered data,” Tech. Rep., Dec. 1979. [Online]. Available: https://doi.org/10.21236/ada081688

[5] D. Pflüger, *Spatially Adaptive Sparse Grids for High-Dimensional Problems*. München: Verlag Dr. Hut, Aug. 2010. [Online]. Available: http://www5.in.tum.de/pub/pflueger10spatially.pdf

[6] A. Genz, “A package for testing multiple integration subroutines,” in *Numerical Integration*. Springer Netherlands, 1987, pp. 337–340. [Online]. Available: https://doi.org/10.1007/978-94-009-3889-2_33

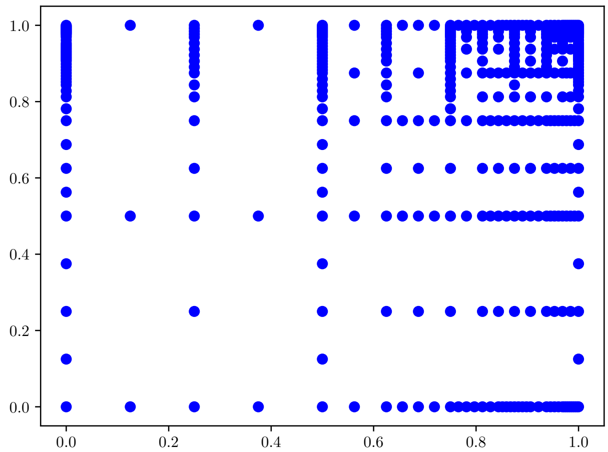


Fig. 12