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*— In recent years, increasing computational power and usage of high-speed internet connections have led to an explosion in the amount of data produced. Generally, this kind of data is used in Machine Learning problems or Data Analytics. This data has a high dimension and is hard to process with classical grid-based approaches. However, some techniques can be used to process this data. This paper presents an adaptive sparse grid approach to handle this data type, where the classical sparse grid approach is insufficient. A spatially adaptive sparse grid approach is presented.**

***Index Terms*— Sparse Grid,Interpolation,Multivariate, Multidimension, Adaptivity, Interpolation, Spatially Adaptive Sparse Grids, Spatial Adaptation**

I. Introduction

Ever-increasing internet speed increased the data produced worldwide, which led to a boom in Machine Learning and Data Analytics fields. Those areas generally have many dimensional large data sets to handle. As with all high-dimensional problems, they suffer from the curse of dimensionality, i.e., they have an exponential dependency on dimension. This is a barrier to the numerical treatment of high-dimensional problems. This exponential dependency makes it harder to use classical mesh-based approaches to solve this problem. One could also use mesh-free methods like Monte-Carlo quadratures.

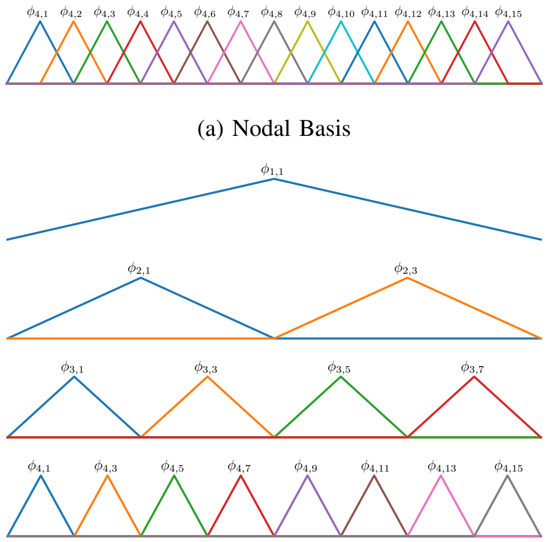
In order to overcome such a problem, the sparse grid method gains more and more popularity. The sparse grid method is a general numerical discretization technique first introduced by the Russian mathematician Smolyak in 1963 [1].

In this paper we will outlook into how to construct a classical sparse grid in section II, and look at the adaptivity of the sparse grid method in section III. We will also look at the use of the spatially adaptive sparse grid method in the context of different test functions in section IV. A final conclusion and outlook will be drawn in section V.

II. Sparse Grids

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

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



(b) Hiearchical Basis

Fig. 1: Comparision of piecewise linear basis functions.

One can use the advantage of adaptivity for problems that do not satisfy smoothness criteria or require further reduction in mesh size. The hierarchical basis is a direct indicator of areas where further refinement is 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).

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

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

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

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

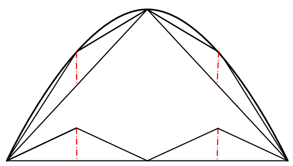
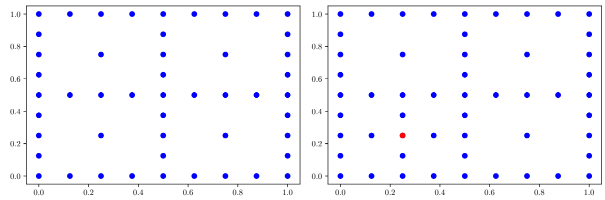


Fig. 2: Interpolation of a parabola using 2 level hierarchical basis and surpluses, surpluses are shown in red lines.



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

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

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 needs 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 a suitable piecewise d-linear basis function at each grid point $x_{l,i}$

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

More information can be found on Bungartz [3].

III. Adaptivity

A regular sparse grid is constructed in a way that takes a cut in a diagonal hyperplane, which means it treats all the dimensions equally. However, there can be an important difference between the dimensions, i.e. one dimension might be more important than others. This can be solved by so-called dimensional adaptivity [4].

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

Moreover, some cases exist where dimensional adaptivity is insufficient to solve the problem. For instance, take a primarily flat function with peaks at specific domain regions. Franke's function [5] eq. (8) is a good example for this, which will be used in section IV.

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

A dimensionally adaptive grid would also add more points to regions where the function is mostly flat. A spatially adaptive grid [6] 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 an initial coarse grid, or one knows, a grid tailored to the problem. Using an iterative process, one could add new points (neighboring grid points in the next higher level) to the region of interest. A consistency constraint exists to enable the usage of sparse grid algorithms; the grid should contain all the hierarchical ancestors of all grid points. This may lead to the number of points added is larger than $2 \cdot d$ Figure 3 shows how this process is done in the two-dimensional regular sparse grid for one refinement step.

The choice of adaptivity criteria to choose the grid points which will be refined is defined by the user. One of the most popular criteria is the surplus-based criterion, which is used in the example section in this paper. The surpluses on a parabolic equation are shown in fig. 2. One can easily observe that at finer levels, surpluses are getting smaller and reduced to zero, where function has a flat behavior. This criterion uses the absolute value of the hierarchical surpluses. It is based on the assumption that a more significant absolute surplus corresponds to a larger second derivative. The choice of adaptivity criteria to choose the grid points which will be refined is defined by the user. One of the most popular criteria is the surplus-based criterion, which is used in the example section in this paper. This criterion uses the absolute value of the hierarchical surpluses. It is based on the assumption that a more significant absolute surplus corresponds to a larger second derivative.

IV. Examples

In this section using SG++ software package [6], interpolation operation on sparse grids are tested for selection functions namely Franke's function and Genz test functions.

For sake of simplicity and visual inspectation only 2 dimensional cases are considered in this section. Thus, a visual inspectation can be done.

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. 3a which has 49 grid points initially. The refinement has been done for 10 refinement step and each step ten grid point is refined. The final grid have 570 grid points, and shown in fig. 5.

Figure 4 shows surface plots obtained from a 100 by 100 full grid fig. 4a and using an adaptive sparse grid with only 570 nodes fig. 4b. Even with just only 570 grid points the sparse grid resolve whole local properties of function very well both two peaks and the dip. Moreover, fig. 6 shows that RMS error on the grid, it is in order of 10^{-6}.

B. Genz Test Functions

A set of test functions are taken from Genz [7], where a discontinuous function has a discontinuity in both dimensions and another function that has a single peak at the product of dimensions in our case, it is located at $(1, 1)$

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

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

1) Discontinuous Function: In fig. 7 it is clear that like Franke's function, both calcualted fig. 7a and interpolated surface fig. 7b from sparse grid looks very similar.

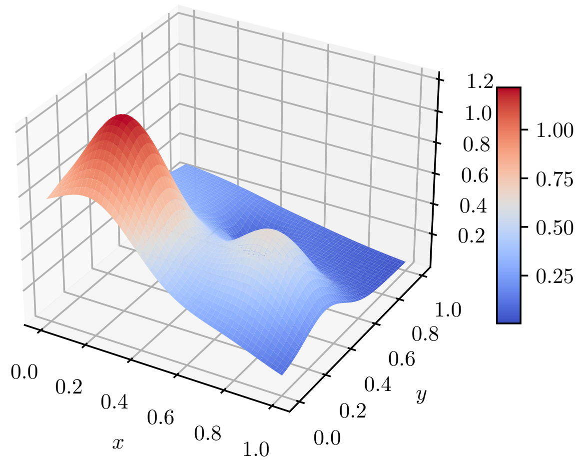
Moreover, here at each refinement step, 10 points are refined, the effect is clearly visible in RMS error fig. 8, the error is almost linearly decreases.

Figure 9 shows the final sparse grid after ten refinement steps. More points are clustered around the intersection of discontinuities in both directions since it is harder to resolve cross derivatives in sparse grids. A good observation here is that the discontinuity is not in a mixed direction sparse grid shows a pretty good performance.

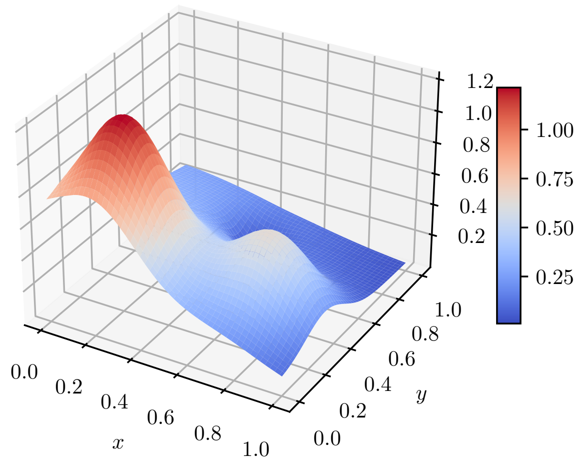
2) Product Peak Function: Another test has been done with the product peak function eq. (10), where the domain has a peak at corner $(1, 1)$.

As in the previous example, ten grid points are refined at each refinement step. Figure 10 shows that interpolation using a sparse grid remarkably agrees with the calculated result.

As shown in fig. 11, the product peak Genz function has a higher error on the regular sparse grid, as a result of high



(a) Calculated



(b) Interpolated

Fig. 4: Comparision of calculated and interpolated Franke's function.

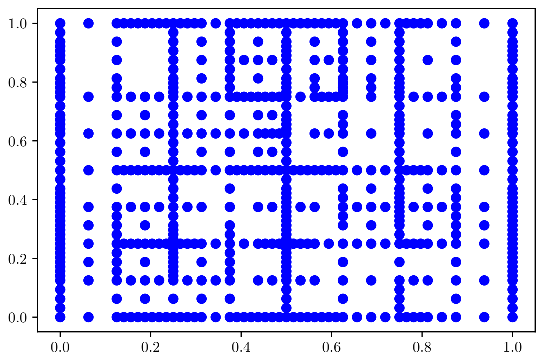


Fig. 5: The final sparse grid with 570 grid points.

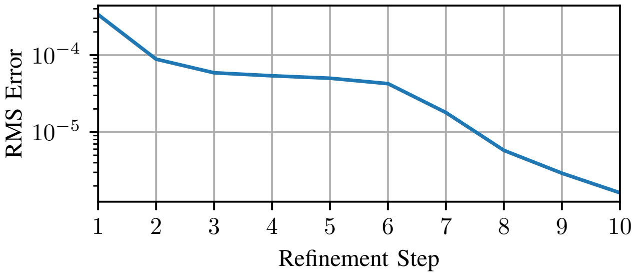
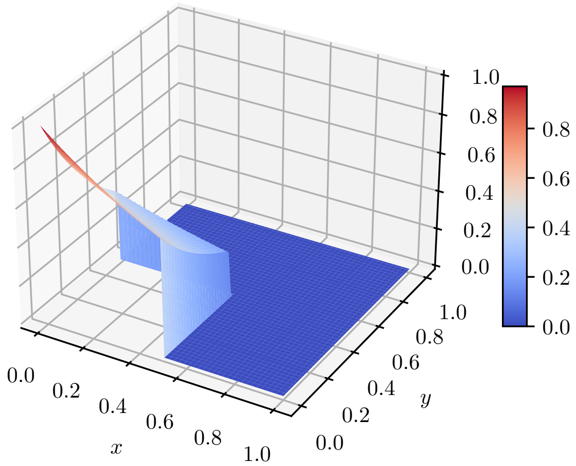
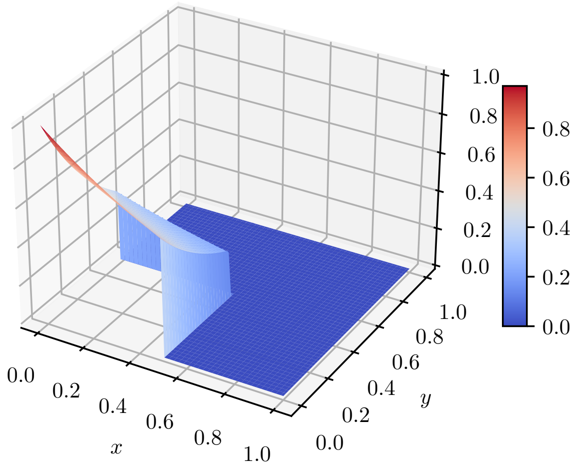


Fig. 6: Error reduction plot of Franke's function w.r.t. refinement steps.



(a) Calculated



(b) Interpolated

Fig. 7: Comparision of calculated and interpolated discontinuous Genz function.

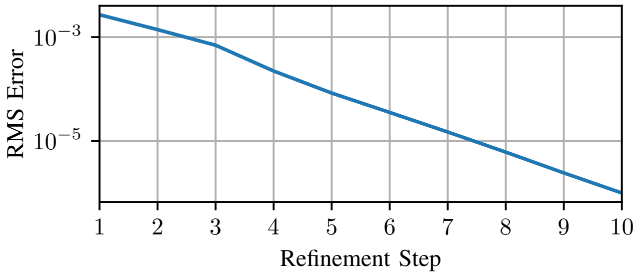


Fig. 8: Error reduction plot of discontinuous Genz function w.r.t. refinement steps.

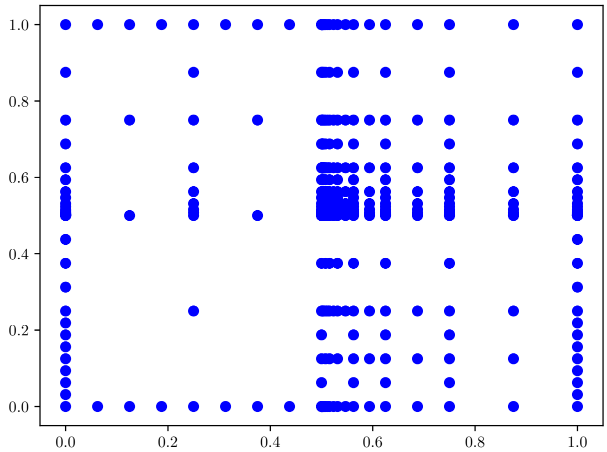


Fig. 9: The final sparse grid of discontinuous Genz function.

gradient in cross product direction. Adaptation helps to reduce the error in the sparse grid, however 10 refinement step is not enough to reduce error to an acceptable level. The error is still high on the sparse grid.

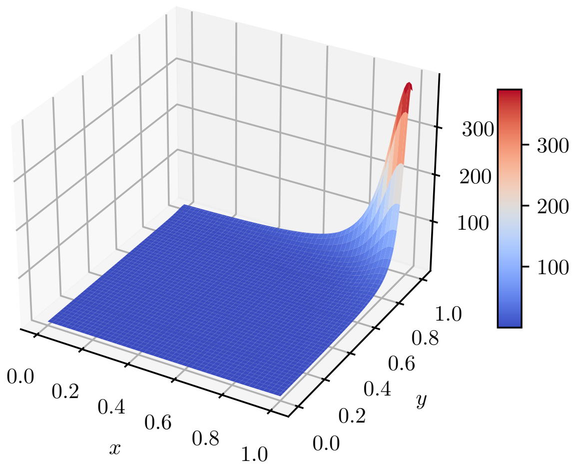
From fig. 12, one can observe that grid points are concentrated around the peak where we have the highest derivatives.

V. Conclusion

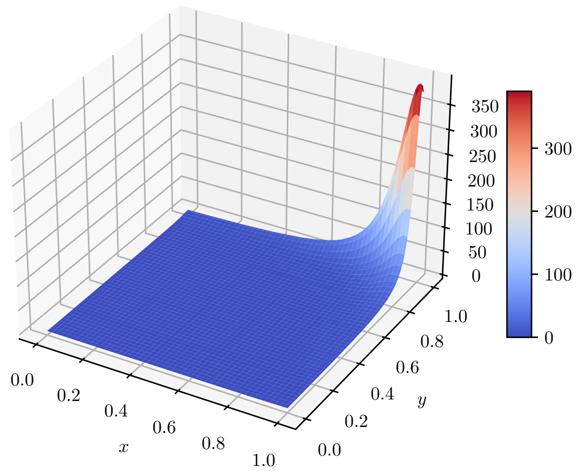
In this work, we have shown examples of spatially adaptive sparse grid method for multidimensional interpolation problems that can be used to solve a wide range of problems. The main idea is to use a sparse grid to construct a multidimensional basis of functions that can be used to interpolate a function on a given domain and use an adaptive refinement scheme to adapt the grid to the function's local behavior.

For demonstrative purposes, we have used free and open-source software SG++ to construct the sparse grids and adaptation. We have used a surplus-based adaptation strategy to adapt the grid to the function's local behavior.

It has already known that a sparse grid method is a powerful tool for solving multidimensional problems without suffering from the curse of dimensionality. It has also been shown that the sparse grid method can be extended by employing the spatial adaptation technique to solve problems efficiently with a wide range of local behavior. The strong and weak points



(a) Calculated



(b) Interpolated

Fig. 10: Comparision of calculated and interpolated product peak Genz function.

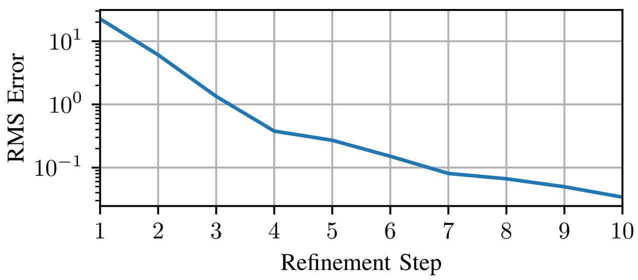


Fig. 11: Error reduction plot of product peak Genz function w.r.t. refinement steps.

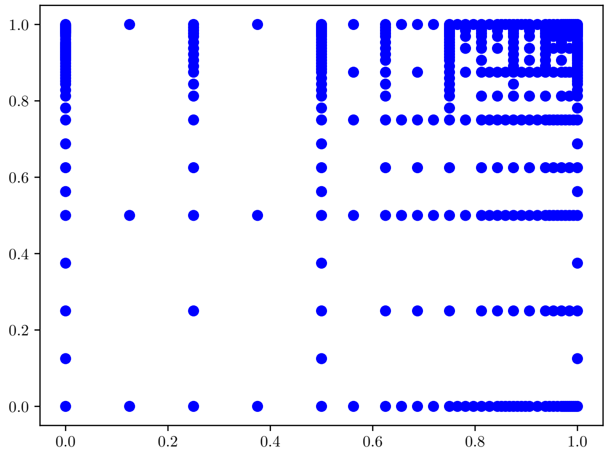


Fig. 12: The final sparse grid of product peak Genz function.

of the method are presented in section IV. The smoothness of the function affects the interpolated function on the grid drastically. It has shown that discontinuities can be handled using adaptivity. However, higher cross derivatives are not handled well. The presented technique can be employed for not only interpolation but also quadrature, regression, classification, and other problems. This enables the use of sparse grids for a wide range of applications.

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] H.-J. Bungartz and M. Griebel, "Sparse grids," *Acta Numerica*, vol. 13, pp. 147–269, May 2004. [Online]. Available: https://doi.org/10.1017/s0962492904000182

[4] 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

[5] 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

[6] 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

[7] 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