Extending adaptive sparse grids for stochastic collocation to hybrid parallel architectures

Richard K. Archibald,[1] Ralf Deiterding,[1] and John Jakeman[2]

[1]Computer Science and Mathematics Division, Oak Ridge National Laboratory, P.O. Box 2008, MS-6367, Oak Ridge, TN 37831

[2]Department of Mathematics, Purdue University, 150 N. University St., West Lafayette, IN 47907-2067

E-mail: deiterdingr@ornl.gov

Abstract. We are developing an adaptive sparse grid library tailored for emerging architectures that will allow the solution of stochastic problems of unprecedented size. This paper gives a brief overview of the problem at hand and presents initial results for a small GPU-based cluster. An outlook on large-scale distributed memory parallelization and our hybrid design approach is also included.

1 Stochastic collocation

In a stochastic simulation, one is typically interested in the relationship between the variables that drive the system (inputs) and the system response (outputs). For the "forward problem," the inputs $\mathbf{Z} = (Z_1, \ldots, Z_d)$ are random variables with distributions that we assume to be known. The outputs are some known functions g of the simulation state $u = u(\mathbf{x}, t; \mathbf{Z})$, which depends on $\mathbf{Z}$ and the deterministic variables $\mathbf{x}$ and (possibly) time t. The mapping from $\mathbf{Z}$ to $g(u)$ can be given abstractly by the function G

$$\mathbf{Z} \mapsto G(\mathbf{Z}; \mathbf{x}, t) \triangleq g(u(\mathbf{x}, t; \mathbf{Z})). \tag{1}$$

The key idea behind stochastic collocation is to select a set of nodes in the random space and then conduct a repetitive *deterministic* simulation at each node.

Consider the direction Z_i, and let $\{G(Z_{i,j})_{j=1}^{m_i}\}$ be the mapping applied to the numerical solution at these nodes. We can approximate the one-dimensional component of the solution u over the range of Z_i using the interpolation formula

$$\mathcal{G}_i[G] = \sum_{j=1}^{m_i} G(Z_{i,j}) \cdot \phi_{i,j}(Z_i), \tag{2}$$

where $\phi_{i,j}$ is the interpolating basis. The solution in the entire space can obviously be approximated by tensor products of one-dimensional approximations. That is,

$$\mathcal{G}_N = \mathcal{G}_{i_1} \otimes \cdots \otimes \mathcal{G}_{i_d}. \tag{3}$$

The difficulty is the *curse of dimensionality*, as the number of terms (or nodes) grows exponentially fast and renders the construction useless for even moderately high dimensions, for example, $d > 5$. A popular approach is the Smolyak sparse grid construction [1]

$$\mathcal{G}_N = \sum_{N-d+1 \le |\mathbf{i}| \le N} (-1)^{N-|\mathbf{i}|} \cdot \binom{d-1}{N-|\mathbf{i}|} \cdot (\mathcal{G}_{i_1} \otimes \cdots \otimes \mathcal{G}_{i_d}). \tag{4}$$

Without too much detail, it suffices to realize that this formula is nothing but a (complex) subset of the full tensor construction. The number of nodes required by the Smolyak formula grows only logarithmically with the dimensionality, while keeping most of high-order approximation properties intact.

2 Sparse grids with adaptation

The types of basis functions are dependent on the type of one-dimensional grids employed. The most frequently used [2, 3], and simplest choice for our purpose, are the multilinear piecewise basis functions, which are based on the one-dimensional formula

$$\phi_{i,j}(Z) = \max(1 - |2^i Z - j|), \quad 0 < j < m_i \tag{5}$$

where, without loss of generality, we have assumed that $Z \in [0, 1]$. These 1D basis functions can be used to form a set of d-dimensional basis functions

$$\phi_{\mathbf{i},\mathbf{j}}(\mathbf{Z}) = \prod_{n=1}^{d} \phi_{i_n,j_n}(Z_n)\,. \tag{6}$$

With the basis function Eq. (5), an equivalent formulation to the one by Smolyak, Eq. (4), is to construct an interpolant hierarchically, that is,

$$G_{N,d}(\mathbf{Z}) = \sum_{|\mathbf{i}|_1 \le N} g_{\mathbf{i}}(\mathbf{Z}), \quad g_{\mathbf{i}}(\mathbf{Z}) = \sum_{\mathbf{j} \in B_{\mathbf{i}}} v_{\mathbf{i},\mathbf{j}} \cdot \phi_{\mathbf{i},\mathbf{j}}(\mathbf{Z}) \in W_{\mathbf{i}}, \tag{7}$$

where

$$B_{\mathbf{i}} = \{j_n = 1, \ldots, m_{i_n},\ j \text{ odd } n = 1, \ldots, d\} \tag{8}$$

and $v_{\mathbf{i},\mathbf{j}}$ is known as the hierarchical surplus. The hierarchical surplus represents the contribution of the associated basis function to the interpolant. This approach splits up the interpolant into contributions from hierarchical difference spaces

$$W_{\mathbf{i}} = \text{span}\,\{\phi_{\mathbf{i},\mathbf{j}}\,|\,\mathbf{j} \in B_{\mathbf{i}}\}\,. \tag{9}$$

Each point in the sparse grid exists only in one subspace, as depicted in two dimensions in Figure 1. Sparse grids can be used to delay the curse of dimensionality by selecting the difference spaces $W_{\mathbf{i}}$ so that decreasing importance is given to the higher-dimensional subspaces. This is achieved by choosing all $W_{\mathbf{i}}$ with $|\mathbf{i}|_1 \le l$.

Since the interpolation subspaces are constructed hierarchically, adaptation is provided naturally for sparse grids. An intrinsic refinement indicator can be developed based on the magnitude of the hierarchical surplus, $v_{\mathbf{i},\mathbf{j}}$. A typical two-dimensional adaptive sparse grid is shown in the left graphic of Figure 2, where the sketch additionally denotes the subspace level l, in which each approximation point resides. It is apparent that the organization of an adaptive sparse grid represents a considerable problem to computer science. Points are clustered along lines, the spacing is quite irregular, and the hierarchical nature of the construction complicates the access to the point data. In particular, efficient parallel implementations for high-dimensional adaptive spare grids are currently not available, thereby severely limiting the applicability to realistic large-scale stochastic collocation methods.

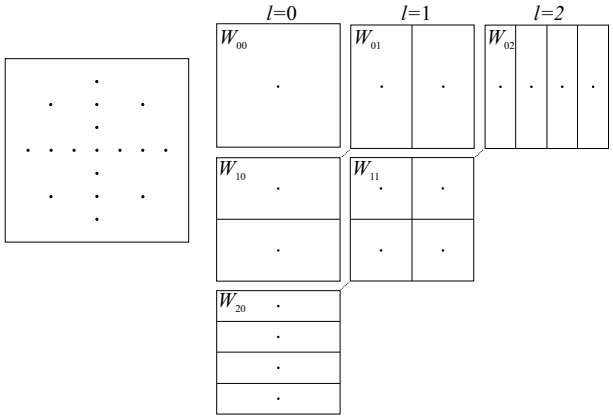

Figure 1: Left: 2D uniform sparse grid with three levels. Right: Assignment of grid points to polynomial subspaces $W_{\mathbf{i}}$. The domain of influence of each point is indicated.

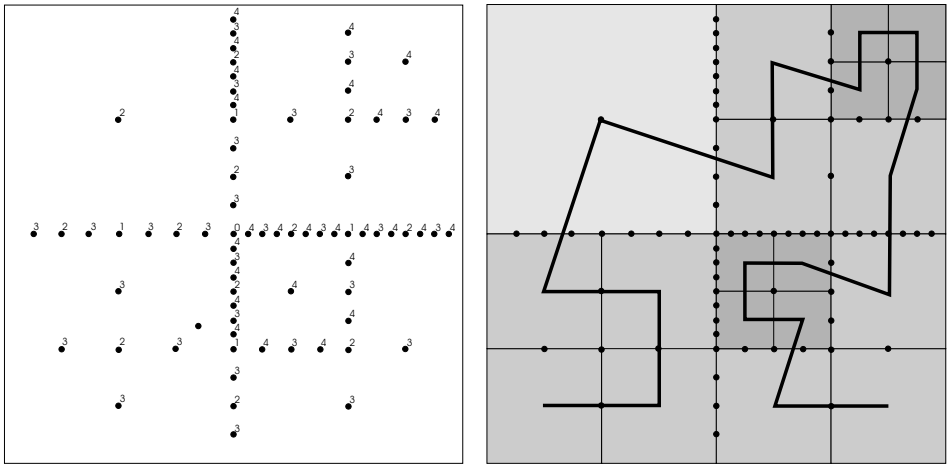

Figure 2: Two-dimensional adapted sparse grid using five subspace levels $l = 0, \ldots, 4$. The left sketch notes the subspace level for each approximation point used. The right graphic visualizes the domains of the respective highest level in different gray shades and shows a generalized space-filling curve used for decomposition.

3 Parallelization

In the following, we detail our design approach for the development of an adaptive sparse grid library for hybrid architectures. To our best knowledge, no such effort has been attempted so far for higher-dimensional sparse grids, and some of the description in here just sketches the concepts that we are currently pursuing. In Section 3.1, we discuss the first promising results utilizing GPUs effectively on a small hybrid cluster to speed the evaluation of the hierarchical surpluses and for sparse grid interpolation. Section 3.2 sketches the domain decomposition approach for a large adaptive sparse grid that we are implementing to achieve strictly local sparse=grid operations on compute nodes.

3.1 Graphics processing units

Graphics processing units (GPUs) will be a prominent component of future designs of exascale supercomputing systems. Individual processor cores in GPUs are optimized to provide high levels of performance for large amounts of data streaming through a memory hierarchy in single instruction multiple thread (SIMT) mode. Optimal performance on GPUs requires abundant low-level SIMT parallelism.

As a first prototype, we have developed a sparse-grid coefficient transformation and interpolation code for nonadapted sparse grids that pursues the parallelization idea of mapping each hierarchical subgrid $W_{\mathbf{i}}$ to a single-thread block. The approach follows the three general requirements to maximize GPU performance, which are in decreasing order of importance sufficient parallelism, coherent memory access, and coherent flow control. Additionally, the code is designed to distribute any block of the sparse grid across a small number of hybrid nodes. On any particular hybrid node, the sparse grid information is passed between multicore CPU and GPU in further refined subgrid blocks. High-order reconstruction algorithms are used within this block where the beneficial accuracy is achieved with computational efficiency through shared memory spaces.

Figure 3 displays the strong scaling of the sparse-grid construction code on one to eight hybrid nodes with all communication costs accounted for in the timings. The sparse grid used for these tests utilizes high-order hierarchical Lagrangian interpolation [2] in 14 dimensions and

6 levels of uniform refinement, with a total of 38,760 subgrids that altogether contain 1,009,905 grid points. The code was run on Yona at the Oak Ridge Leadership Computing Facility, a hybrid 16-node system with each node containing two sockets, with a 6-core Opteron 2435 in one socket and a NVIDIA Fermi GPU C2050 in the other. For reference, we display the strong scaling of using two multicore CPUs per node as opposed to a hybrid GPU/multicore CPU node. One can see that a single hybrid node is comparable to two multicore nodes. It is also noted that the current code does all calculations on the GPU using only one core of the multicore CPU for communication, thereby allowing the future possibility for additional work to be performed on the other cores. Furthermore, scaling declines for this high-order reconstruction algorithm when the number of nodes become very large.

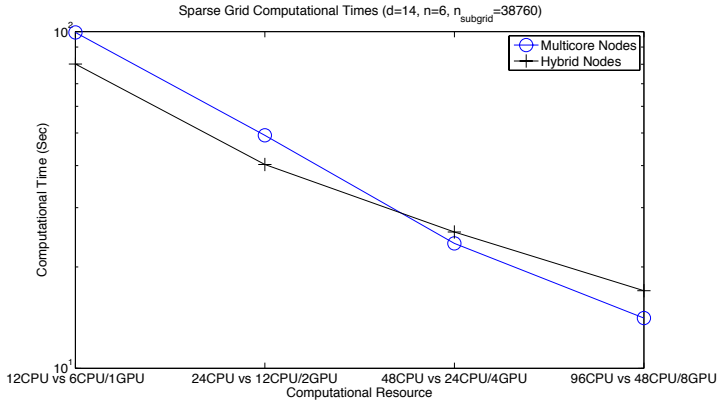

Figure 3: Scaling of high-order sparse grid reconstruction, comparing multicore with hybrid nodes.

3.2 Distributed memory

The inherent assumption for parallelization on distributed-memory petascale architectures (using MPI) is that already on such systems (adaptive) sparse grids will be encountered that are too large to allow aggregation of the data on a single node and that communication costs are non-neglectable. Under these assumptions, the hierarchical nature of the sparse grid approximation becomes the greatest obstacle in devising and implementing efficient data access and parallelization strategies. As depicted in Figure 1, the geometric domain of influence of each approximation point is shrinking successively in a non-trivial manner for higher levels in the subgrid space hierarchy. Yet, evaluation of the interpolation formula, Eq. (7), requires knowledge of all hierarchical surpluses, through all levels of the hierarchy in whose domain of influence an arbitrary point $\mathbf{Z}$ comes to lie. Our first-pursued design approach to this problem is to combine the locality-preserving properties of space-filling curves [4] with the efficiency of a local hash-based data access to the approximation point data [3, 5]. Basically, the space-filling curve will be used to define an ordering of the geometric regions at the highest level of local adaptation, which can easily be split into portions of similar size, yielding load-balanced distributions. Here, we assume a workload estimation criterion based on the number of approximation points in each geometric region. A sketch depicting a generalized space-filling curve in two space dimensions is shown on the right of Figure 2.

The approach assigns the approximation space unambiguously to compute nodes and the key idea for a strictly local sparse-grid interpolation is then to distribute all points, whose domain of influence overlaps with the locally assigned domain, to the same node. Clearly, this concept must involve the utilization of some halo points, with values synchronized by parallel communication whenever the decomposition changes. In particular, coarser-level points will have to be replicated; for example, the data associated to $l = 0$ will be available on every node. A decomposition of the adaptive sparse grid of Figure 2 to four nodes, based on the shown space-filling curve, is depicted in Figure 4. The sketches of Figure 4 show all approximation points that are required to allow a strictly local evaluation of Eq. (7). In contrast to domain-decomposition

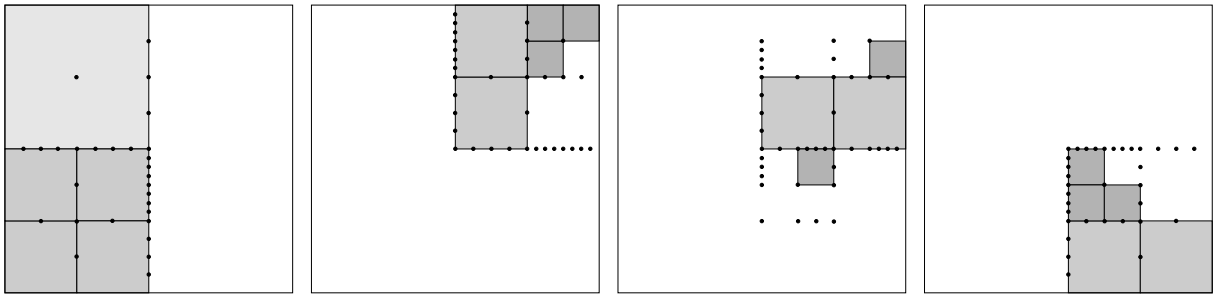

Figure 4: Domain decomposition of the adaptive sparse grid of Figure 2 to four processors for distributed interpolation based on a generalized space-filling curve.

techniques on conventional grids, knowledge of hierarchical approximation points outside of the local domain is required. Since fairly general point distributions will be encountered in practice, each node will employ a local hash table for data storage and efficient access.

4 Conclusions and outlook

The described strategies for MPI and GPU parallelization are obviously complementary and can be applied in unison thanks to the strict locality preservation of the distributed memory approach. Our current emphasis is on the development of a d-dimensional space-filling curve algorithm and communication routines for general, distributed sparse grids to evaluate our design approach on available parallel hybrid systems. Further, we are developing advanced criteria tailored for sparse grids to detect and refine discontinuities in the stochastic space effectively and reliably [6, 7].

Acknowledgments

This work is sponsored by the Office of Advanced Scientific Computing Research, Applied Mathematics Research Program; U.S. Department of Energy (DOE) and was performed at the Oak Ridge National Laboratory, which is managed by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725.

References

[1] S. Smolyak. Quadrature and interpolation formulas for tensor products of certain classes of functions. *Soviet Math. Dokl.*, 4:240–243, 1963.

[2] H.-J. Bungartz and M. Griebel. Sparse grids. *Acta Numerica*, 1–123, 2004.

[3] M. Griebel. Adaptive sparse grids multilevel methods for elliptic PDEs based on finite differences. *Computing*, 61:151–179, 1998

[4] H. Sagan. *Space-Filling Curves*. Springer-Verlag, New-York, 1994

[5] T. Schiekofer and G. Zumbusch. Software concepts of a sparse grid finite difference code. In Hackbusch, W. and Wittum, G., editors, *Proc. 14th GAMM Seminar. Concepts of Numerical Software*, Notes on Numerical Fluid Mechanics. Vieweg, 1998.

[6] R. Archibald, A. Gelb, R. Saxena, and D. Xiu. Discontinuity detection in multivariate space for stochastic simulations. *J. Comput. Phys.*, 228(7):2676–2689, 2009.

[7] J. Jakeman, R. Archibald, and D. Xiu. Characterization of discontinuities in high-dimensional stochastic problems on adaptive sparse grids. *J. Comput. Phys.*, 230(10):3977–3997, 2011.