Massively Parallel Simulation of Plasma Turbulence with the Sparse Grid Combination Technique

Mario Heene
Institute of Parallel and Distributed Systems
University of Stuttgart
Stuttgart, Germany
Email: mario.heene@ipvs.uni-stuttgart.de

Alfredo Parra Hinojosa
Chair of Scientific Computing
Technische Universität München
Munich, Germany
Email: hinojosa@in.tum.de

Dirk Pflüger
Institute of Parallel and Distributed Systems
University of Stuttgart
Stuttgart, Germany
Email: mario.heene@ipvs.uni-stuttgart.de

I. Introduction

The problem sizes of high dimensional problems are often severely limited due the exponential increase in degrees of freedom with the dimensionality. This so-called curse of dimensionality can, for example, be observed in the simulation of plasma turbulence in a fusion device as described by the gyrokinetic equations, a five-dimensional PDE.

A scalable approach to solve higher-dimensional problems is the sparse grid combination technique (SGCT) [1]. It is based on an extrapolation scheme that decomposes a single large problem (i.e. discretized with a fine resolution) into multiple moderately-sized problems that have coarse and anisotropic resolutions. This introduces a second level of parallelism, enabling one to compute the component problems in parallel, independently and asynchronously of each other. This minimizes the demand for global communication and synchronisation, which is expected to be one of the limiting factors with classical discretization techniques to achieve scalability on future exascale systems. By mitigating the curse of dimensionality, it offers the means to tackle problem sizes that would be out of scope for the classical discretization approaches. Furthermore, the inherent data redundancy of this technique enables fault-tolerance on the algorithmic level at low cost without the need of checkpointing or process replication. These features make the SGCT a promising candidate to tackle the algorithmic challenges that are expected from future exascale systems. It is one of the major goals within the project EXAHD of Germany's priority program *Software for Exascale Computing* (SPPEXA) to demonstrate this.

II. Sparse Grid Combination Technique

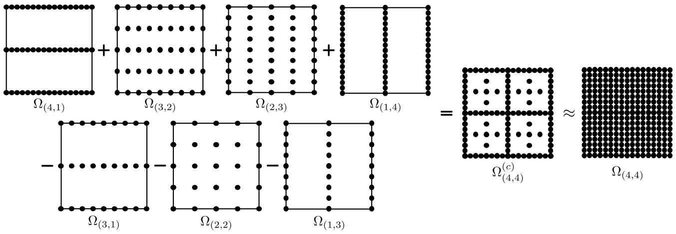

Fig. 1. A two-dimensional combination technique with $n = 4$.

In the most general form the SGCT can be stated as

$$u_n^{(c)}(\boldsymbol{x}) = \sum_{\boldsymbol{l} \in \mathcal{I}} c_{\boldsymbol{l}} u_{\boldsymbol{l}}(\boldsymbol{x}). \tag{1}$$

The sparse grid approximation $u_n^{(c)}$ (the so-called combined solution) is a weighted sum of the component solutions $u_{\boldsymbol{l}}$ and corresponding combination coefficients $c_{\boldsymbol{l}}$. Each $u_{\boldsymbol{l}}$ is the solution of a PDE on a regular grid $\Omega_{\boldsymbol{l}} := \Omega_{l_1} \times ... \times \Omega_{l_d}$, which has mesh size $h_{\boldsymbol{l}} := (2^{-l_1}, ..., 2^{-l_d})$. $\boldsymbol{l}$ denotes the discretization level in each of the d-dimensions and n is a parameter that defines the maximum discretization level in each dimension. Fig. 1 shows an example. The index set $\mathcal{I}$ defines the set of component grids used for the combination. There exist different strategies to choose $\mathcal{I}$. For instance, the classical combination technique is given by

$$u_{\boldsymbol{n}}^{(c)}(\boldsymbol{x}) = \sum_{q=0}^{d-1} (-1)^q \binom{d-1}{q} \sum_{\|\boldsymbol{l}\|_1 = n+d-1-q} u_{\boldsymbol{l}}(\vec{x}) \tag{2}$$

If u fulfills certain smoothness conditions, the approximation quality of $u_n^{(c)}$ is comparable to the approximation quality of u_n, the solution on the full grid $\Omega_{\boldsymbol{n}}$. Note that for the problem sizes we aim for, direct computation of $u_{\boldsymbol{n}}$ would not be possible any more due to the enormous number of unknowns.

III. Application: GENE

A limiting factor for the generation of clean sustainable energy from plasma fusion reactors are microinstabilities that arise during the fusion process [2]. Simulation codes like GENE [3] play an important role in understanding the mechanisms of the resulting anomalous transport phenomena. The combination technique has already been successfully applied to eigenvalue computations in GENE [4]. It has also been used to study fault tolerance on the algorithmic level [5].

The (5+1)-dimensional gyrokinetic equation describes the dynamics of hot plasmas. In very simplified form it can be written as:

$$\frac{\partial F_\sigma}{\partial t} + \frac{\partial X}{\partial t} \cdot \nabla F_\sigma + \frac{\partial v_{||}}{\partial t} \frac{\partial F_\sigma}{\partial v_{||}}. \tag{3}$$

See [6], [7] for a thorough description of the model.

One of the goals of this project is to push the computational limits of plasma turbulence simulations by applying the combination technique. This especially addresses global non-linear simulations of large fusion devices, as, e.g., ITER, which theoretically would require the performance of future exascale systems. However, it is not straightforward to assume that the code will scale on future systems, which are expected to have millions of compute cores. In this context, the SGCT is a promising approach not only to ensure the scalability of plasma turbulence simulations (and other higher-dimensional problems) on future systems, but also to enable larger problem sizes on current systems.

IV. Scalable Parallelization Concept

We are developing a general software framework for large-scale computations with the combination technique as a part of our sparse grid library SG++ [8]. The framework implements the manager-worker pattern to realize the two-level parallelism of the combination technique.

The available compute resources, in form of MPI processes, are arranged in process groups. A dedicated manager process distributes the component grids to the process groups. Each group then computes the component grids one after another by executing the application code on all processes of the group. The component grids are computed independently and asynchronously of each other. There is no need for global communication or synchronization between processes in different process groups.

With this concept we can already achieve very good load balancing with a simple load model where the computation cost of a component grid simply is its number of unknowns. We could further improve the load balancing by a model tailored to the specific application which also considers effects caused the anisotropy of the discretization [9].

V. Efficient Global Communication

In the case of time-dependent initial value problems (this is the case with GENE) advancing the combined solution in time requires to combine the component solutions (according to Eq. (1)) every few time steps and to set the combined solution as new initial value on each component grid. After that, the independent computations are continued until the next combination point. This procedure is necessary to guarantee convergence and stability of the combined solution. Being the only remaining step that involves global communication (between processes of different process groups), an efficient and scalable implementation is crucial for the overall performance of computations with the combination technique.

In [10] we have presented efficient communication schemes that exploit the hierarchical structure of the sparse grid. In [11] we have demonstrated that our combination algorithm scales up to more than 180 000 cores on *Hazel Hen* and is faster than a single time of the application. It is not yet clear how frequently one has to combine the component grids and in general it depends on the application and the scenario at hand. However, our results show that even the worst case scenario, combining after each time step, would be feasible.

VI. Algorithm-based Fault Tolerance

The inherent redundancy of the SGCT can be exploited to deal with crashing processes (or nodes) on the algorithmic level, without the need for checkpointing or process replication. Furthermore, it is possible to detect so-called silent faults by comparing the component grids with each other [12].

The fault-tolerant combination technique (FTCT) was proposed in [13]. When processes fail during the computation, we lose only a subset of our component grids. Likewise, grids might be affected by silent faults. Instead of recomputing the missing (or corrupted) grids, the idea of the FTCT is to find a different combination which excludes them. Experiments showed that the accuracy of the corrected combined solution is still astonishingly well.

In [14] we presented a parallel implementation of the FTCT for arbitrary dimensionality and obtained similar results. Furthermore, we demonstrated that the actual overhead to recover from process failures is very low, even on a large number of cores of an HPC system.

References

[1] M. Griebel, M. Schneider, and Zenger, “A combination technique for the solution of sparse grid problems,” 1992.

[2] E. Doyle, Y. Kamada, T. Osborne *et al.*, “Chapter 2: Plasma confinement and transport,” *Nuclear Fusion*, vol. 47, no. 6, p. S18, 2007.

[3] T. Görler, X. Lapillonne, S. Brunner, T. Dannert, F. Jenko, F. Merz, and D. Told, “The global version of the gyrokinetic turbulence code GENE,” *Journal of Computational Physics*, vol. 230, no. 18, pp. 7053 – 7071, 2011.

[4] C. Kowitz and M. Hegland, “The sparse grid combination technique for computing eigenvalues in linear gyrokinetics,” *Procedia Computer Science*, vol. 18, no. 0, pp. 449 – 458, 2013, 2013 International Conference on Computational Science.

[5] A. Parra Hinojosa, C. Kowitz, M. Heene, D. Pflüger, and H.-J. Bungartz, “Towards a fault-tolerant, scalable implementation of GENE,” in *Proceedings of ICCE 2014*, ser. Lecture Notes in Computational Science and Engineering. Springer-Verlag, 2015.

[6] T. Dannert, “Gyrokinetische simulation von plasmaturbulenz mit gefangenen teilchen und elektromagnetischen effekten,” Ph.D. dissertation, Technische Universität München, 2004.

[7] A. Brizard and T. Hahm, “Foundations of nonlinear gyrokinetic theory,” *Rev. Mod. Phys.*, vol. 79, pp. 421–468, Apr 2007.

[8] “SG++ library.” [Online]. Available: http://sgpp.sparsegrids.org/

[9] M. Heene, C. Kowitz, and D. Pflüger, “Load balancing for massively parallel computations with the sparse grid combination technique.” in *Parallel Computing: Accelerating Computational Science and Engineering (CSE)*, ser. Advances in Parallel Computing, vol. 25, 2014, pp. 574–583.

[10] P. Hupp, M. Heene, R. Jacob, and D. Pflüger, “Global communication schemes for the numerical solution of high-dimensional {PDEs},” *Parallel Computing*, vol. 52, pp. 78 – 105, 2016.

[11] M. Heene and D. Pflüger, “Scalable algorithms for the solution of higher-dimensional PDEs,” in *SPPEXA Workshop*, ser. LNCSE. Springer, 2016.

[12] A. Parra Hinojosa *et al.*, “Handling silent data corruption with the sparse grid combination technique,” in *Proceedings of the SPPEXA Workshop*, ser. LNCSE. Springer-Verlag, 2016, accepted.

[13] B. Harding *et al.*, “Fault tolerant computation with the sparse grid combination technique,” *SIAM Journal on Scient. Comp.*, vol. 37, no. 3, pp. C331–C353, 2015.

[14] M. Heene, A. Parra Hinojosa, H.-J. Bungartz, and D. Pflüger, “A massively-paralle, fault-tolerant solver for high-dimensional pdes,” in *Euro-Par 2016*, ser. LNCS. Springer, accepted.