HiFiLeD 2021
2^{nd} High-Fidelity Industrial LES/DNS Symposium
22-24 September 2021, Toulouse, France

Enabling larger and faster simulations from mesh symmetries

Xavier Álvarez-Farré[1], Àdel Alsalti-Baldellou[1,2], Andrey Gorobets[3], Assensi Oliva[1] and F.Xavier Trias[1]

[1] Heat and Mass Transfer Technological Center, Technical University of Catalonia, Carrer Colom 11, 08222 Terrassa (Barcelona), Spain.
[2] TermoFluids S.L. (https://www.termofluids.com), Sabadell (Barcelona), Spain.
[3] Keldysh Institute of Applied Mathematics, Russian Academy of Sciences, Miusskaya Sq. 4, 125047 Moscow, Russia.

Key words: Parallel CFD, Mesh symmetries, SpMV, SpMM, Memory footprint

Abstract. The evolution in hardware technologies enables scientific computing to advance incessantly and reach further aims. Many algorithms employed in numerical simulations, particularly in computational fluid dynamics (CFD), can significantly benefit from using massively parallel accelerators with high computing capabilities and fast integrated memory. Especially memory-bound algorithms with low arithmetic intensity, for which the memory bandwidth is the principal limiting factor [1]. Therefore, the use of graphics processing units (GPUs) in scientific computing has become rather mature, and there are many successful examples in the literature [2, 3]. However, the integrated memory capacity in GPUs is much smaller than the traditional off-chip DRAM controlled by CPUs, which becomes an additional constraint.

To take advantage of the hybridization of high-performance computing systems, the computing subroutines that form the algorithms, the so-called kernels, must be adapted to complex paradigms such as distributed-memory and shared-memory multiple-instruction, multiple-data parallelism, and stream processing. This programming complexity encourages the demand for portable and sustainable implementations of scientific simulation codes [4]. In this line, we proposed in [5] an algebra-based framework for heterogeneous computing as a portable solution for the scale-resolution of incompressible turbulent flows on unstructured meshes. Briefly, the CFD algorithm relies on a set of only three algebraic kernels. Thus, the kernel code shrinks to hundreds of lines; portability becomes natural, and maintaining the OpenMP, OpenCL, and CUDA implementations requires little effort. Besides, we can easily use standard libraries (*e.g.*, cuSPARSE of NVIDIA, clSPARSE of AMD) optimized for a particular architecture, in addition to our specialized in-house implementations.

By profiting from the (spatial) mesh symmetries, we aim at tackling both the bottlenecks mentioned above at once. Roughly, considering an n-dimensional domain exhibiting $p \leq n$ symmetric (by reflection) directions, we build the mesh and the respective discrete differential operators for only one of the composing symmetric blocks. Then, the scalar and vector fields involved in the simulation become a set of 2^p vectors belonging to a reduced vector space. On the one hand, this allows for using the sparse matrix-matrix product (SpMM), also known as sparse matrix-multiple-vector product, which increases the arithmetic intensity with respect to the classical SpMV: it allows for reusing the coefficients of the sparse matrix. Moreover, the application of this kernel to symmetric meshes can be combined with the resolution of multiple transport equations at once (*e.g.*, three components of the velocity in collocated formulations plus the temperature field). Figure 1 (left) shows the theoretical gain of using SpMM instead of SpMV. On the other hand, the size of the discrete differential operators (*i.e.*, sparse matrices) in a "symmetry-aware" framework is reduced by a factor of 2^p. Figure 1 (right) illustrates the theoretical diminution of memory footprint in the direct numerical simulation of incompressible

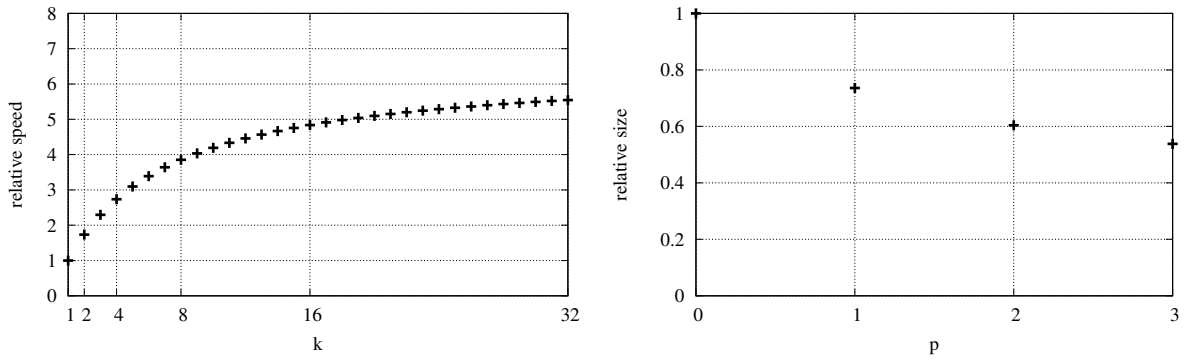

Figure 1: (left) Theoretical gain of SpMM with respect to SpMV as the number of vectors, k, increases. (right) Theoretical diminution of memory footprint as the number of symmetries, p, increases.

turbulent flows. Note that this diminution is not directly proportional to 2^p because the scalar and vector fields maintain their original size in this framework, although divided into smaller blocks. As a result, a symmetry-aware numerical simulation could solve faster the time-step of a larger simulation.

At the conference, we aim to analyze the theoretical magnitude of these advantages and contrast them with the obtained results. Besides, we will discuss the application of symmetry-aware simulations to academic and industrial large-scale simulations.

Acknowledgments. The work of A. G. has been funded by the Russian Science Foundation, project 19-11-00299 À. A. B. is supported by the predoctoral grants DIN2018-010061 and 2019-DI-90 given by the Spanish Ministry of Science, Innovation and Universities (MICINN), and the Catalan Agency for Management of University and Research Grants (AGAUR). The work of X. Á. F., À. A. B., A. O. and F. X. T. was supported by the competitive R+D project ENE2017-88697-R by the Spanish Research Agency. The work has been carried out using the MareNostrum 4 supercomputer of the Barcelona Supercomputing Center. The authors thankfully acknowledge the institution.

REFERENCES

[1] Samuel Williams, Andrew Waterman, and David Patterson. Roofline: An insightful model for Performance Visual multicore Architectures. *Communications of the ACM*, 52(4):65–76, apr 2009.

[2] Peter Vincent, Freddie Witherden, Brian Vermeire, Jin Seok Park, and Arvind Iyer. Towards Green Aviation with Python at Petascale. In *SC16: International Conference for High Performance Computing, Networking, Storage and Analysis*, Salt Lake City, nov 2016. IEEE.

[3] A.N. Bocharov, N.M. Evstigneev, V.P. Petrovskiy, O.I. Ryabkov, and I.O. Teplyakov. Implicit method for the solution of supersonic and hypersonic 3D flow problems with Lower-Upper Symmetric-Gauss-Seidel preconditioner on multiple graphics processing units. *Journal of Computational Physics*, 406:109189, apr 2020.

[4] Mohammed Al Farhan, Ahmad Abdelfattah, Stanimire Tomov, Mark Gates, Dalal Sukkari, Azzam Haidar, Robert Rosenberg, and Jack Dongarra. MAGMA templates for scalable linear algebra on emerging architectures. *The International Journal of High Performance Computing Applications*, 34(6):645–658, nov 2020.

[5] Xavier Álvarez-Farré, Andrey Gorobets, and F. Xavier Trias. A hierarchical parallel implementation for heterogeneous computing. Application to algebra-based CFD simulations on hybrid supercomputers. *Computers & Fluids*, 214:104768, jan 2021.