Efficient High-Performance and Numerical Computing for Turbulence Simulations – Adriana Ladera

Turbulent flows are some of the most complex and interesting phenomena in the world, making appearances in vortex dynamics, climate, and weather, from scales as small as blood flow in the valves of a heart muscle to scales as large as Jupiter's vortices and the Orion nebula. In computational fluid dynamics, numerical simulations are a key tool to understanding turbulence in computational science and high-performance computing (HPC). However, direct numerical simulations of turbulent flows are computationally expensive for most real-world problems. For this reason, researchers resort to two other models that dominate turbulence simulations: (i) Reynolds averaged-Navier Stokes (RANS), which are lower in fidelity, but are computationally affordable, and (ii) large-eddy simulations (LES), which are higher in fidelity than RANS but increasingly costly for more complex systems. Still, HPC for turbulence simulations proves to be a challenge. With Post-Moore's Law computing in sight, we must devise more efficient computing algorithms to compensate for approaching hardware bottlenecks. I plan to answer the question, "how can we efficiently simulate turbulent flows?" by finding transformative high performance and numerical computing tools for turbulence simulations.

Background: Eddies, a staple feature of turbulent flow, are vortices within the flow system that can vary in size. Near the wall boundary, eddies can be in the order of micrometers, whereas their size increases farther away from the boundary. Computational grids for turbulent flows ideally have a high enough resolution (i.e. finer grid points) to resolve the small near-wall eddy scales, but simulating a system this fine is intractable even with today's most advanced high performance computers. More affordable computational grids are therefore a tradeoff that leave the smaller-scale eddies unresolved. A compromise to this problem that resolves small near-wall scales is a hybrid of RANS and LES, called wall-modeled LES (WMLES), which uses LES (below grid resolution) near the wall boundary and RANS (at grid resolution) farther away, maintaining good accuracy while saving computational time. Finite element methods can provide approximate solutions of these models, of which the discontinuous Galerkin (DG) finite element method has become highly favorable [1]. In the context of WMLES, DG spatial discretization has been shown to improve solution accuracy on significantly under-resolved grids. Additionally, the good scalability and data locality (e.g. DG elements communicate with their nearest neighbors) of the high-order DG method makes it a promising tool for enabling HPC methods on today's computer architectures.

Proposal: I plan to employ novel parallel algorithms and fast numerical computational techniques to construct an efficient, heterogeneously parallelizable, and power efficient framework for DG methods of WMLES.

One prospective avenue for parallelism is the work developed by Ashraf *et al* [2], a massively parallel computational model for moving into HPC exa-scale systems. This approach merges MPI, OpenMP, and CUDA (MOC) to deliver heterogenous levels of parallelism, achieving massive performance while consuming less power.

Numerical techniques are also a natural suggestion for the DG method. A major challenge in finding appropriate linear solvers for high-order DG methods is employing computationally inexpensive but reliable preconditioners [1]. Persson *et al* [3] establish a preconditioner based on block-incomplete LU factorization of a matrix A with zero fill-in (ILU0) that outperforms all other alternatives tested (*fill-in* refers to zero matrix entries in A that are nonzero in the L or U factors), and demonstrated ILU0 preconditioning a DG spatial discretization for RANS flows. ILU0 may potentially be extended to WMLES.

Additionally, high precision computational costs are an outstanding problem. While higher-precision floating point operations have superior accuracy, they can get extremely expensive and inefficient for larger computations [4]. A possibility to address this is the use of mixed precision iterative refinement (MPIR), which has been shown by Haidar *et al* [5] to increase both performance and power efficiency of LU factorization while maintaining similar accuracy to an LU factorization carried out in higher precision.

I aim to apply MOC parallelism and the combined ILU0-MPIR linear solver to the DG method for WMLES. Employing the MOC framework for the DG method could open doors for other exa-scale finite element method systems to become highly parallelizable, whereas the ILU0-MPIR linear solver could increase power and performance efficiency of DG computations while conserving accuracy. The underlying

objective would be to introduce a parallel computational model and a linear solver that is a fresh and efficient approach for turbulence simulations.

Research Plan: The MOC method in [2] accomplishes massive parallelism by merging course-grained and fine-grained parallelism via inter-node computation, intra-node computation, and accelerated GPU devices. Inter-node computation provides coarse grain parallelism by using MPI processes that communicate with host CPUs to partition data across all connected nodes. OpenMP then conducts intra-node computations by parallelizing CPU threads to achieve fine grain parallelism. The accelerated GPU environment is then invoked, where CUDA kernels further compute the data to obtain even finer grain parallelism. I will introduce novel algorithmic improvements to adapt DG for WMLES to the MOC method and investigate strategies to reduce data communication, such as adding CUDA statements that restrict communication with host CPUs to only when necessary. I will then execute current state-of-the-art HPC implementations for DG methods and compare the performance of this state-of-the-art with my modifications on the MOC method. The desired outcome is that this application will establish the viability of MOC as a suitable and more efficient parallel paradigm for DG methods of WMLES.

While [3] applied ILU0 to DG spatial discretization for RANS flows, I will extend their approach to WMLES. I propose to devise a DG preconditioner using an MPIR linear solver based on ILU0 factorization. The algorithm will first solve the linear system with ILU0 factorization in precisions lower than single precision, and then iteratively increase precision for each residual calculation until the desired precision is achieved. I will then compare the MPIR methods with standard single and double precision ILU0 factorization schemes to ensure that solutions with lower precision in earlier iterations could still maintain similar accuracy to their higher precision counterparts. Since ILU0 performance is shown to be highly dependent on the element ordering, I will implement the greedy heuristic algorithm inspired by [3] to induce an element ordering that can improve convergence. The overall aim of this combined ILU0+MPIR approach is to increase power and performance efficiency for DG methods for WMLES.

Intellectual Merit: If the methods described above are proven to be successful, then MOC parallelism could pave the way for a wide array of computationally efficient computational fluid dynamics methods beyond DG, making the field of finite element methods highly parallelizable for a broad variety of applications. Successful implementation of ILU0-MPIR linear solvers for DG can also open doors for increased energy efficiency in aspects of computational fluid dynamics that rely on the use of linear solvers.

Broader Impact: This optimized parallel and numerical framework addresses a key challenge in turbulence simulation efficiency. In DG, it can provide researchers with more efficient ways to understand problems in other areas where DG methods have received particular interest, such as electrodynamics and plasma. Additionally, understanding the effects of turbulence can help to improve infrastructure, such as bridge supports over rivers, and transportation. In the case of aerospace engineering, the framework could accelerate the development of advanced aircraft design via fluid dynamics simulations and cost-effective virtual testing. Advanced aircraft designs would also reduce the fuel consumption normally induced by competing against aerodynamic drag, and the environmental impact of reduction in pollutant emissions that is enabled by the proposed computational savings will be equally impressive.

References

[1] Kronbichler, M., Persson, P.O. (2021). High-Performance Implementation of Discontinuous Galerkin Methods with Application in Fluid Flow. *In Efficient High-Order Discretizations for Computational Fluid Dynamics*, (pp. 57-115). Springer International Publishing. **[2]** Ashraf, M.U., Eassa, F.A., Albeshri, A.A., Algarni, A. (2018). Performance and Power Efficient Massive Parallel Computational Model for HPC Heterogenous Exascale Systems. *IEEE Access*, 6, 10.1109. **[3]** Persson, P.O., Peraire, J. (2008). Newton-GMRES Preconditioning for Discontinuous Galerkin Discretizations of the Navier Stokes Equations. *SIAM J. Sci. Comput.*, 30, 6, pp. 2709 – 2733. **[4]** Dongarra, J., Grigori, L., Higham, N.J. (2019). Numerical algorithms for high-performance computational science. *Phil. Trans. R. Soc. A*, 378, 20190066. **[5]** Haidar, A., Abdelfattah, A., Zounon, M., Wu, P., Pranesh, S., Tomov, S., & Dongarra, J. (2018). The Design of Fast and Energy-Efficient Linear Solvers: On The potential Of Half Precision Arithmetic And Iterative Refinement Techniques. In *Computational Science – ICCS 2018* (pp. 586-600).