GPU-Based HPC and AI Developments for CFD

S. Posey[*], J. Luitjens[*] and O. Hennigh[*]
Corresponding author: sposey@nvidia.com

[*] NVIDIA Corporation, USA.

Abstract: Current trends in computational fluid dynamics (CFD) include the use of graphics processing units (GPUs) as parallel co-processors to CPUs in order to accelerate numerical operations and algorithms common to CFD solvers. The first topic will examine advances in GPU method development for various CFD software including FUN3D from NASA and OpenFOAM from OpenCFD. The second topic will introduce novel CFD methods in artificial intelligence (AI) capable of encoding the Navier-Stokes equations into physics-informed neural networks (PINNs), while being agnostic to geometry, or initial and boundary conditions. Examples will include NVIDIA use of such techniques applied in electronics cooling design.

Keywords: GPU, HPC, AI, Computational Fluid Dynamics, Neural Network Solver

1 Introduction

Efficient use of computational resources and CFD simulation turn-around times are critically important factors behind engineering decisions to expand CFD technology to support more product design. Recent developments in GPU-based high performance computing (HPC) and AI have improved computational speeds by orders of magnitude for a broad range of CFD simulations relevant to engineering practice.

2 HPC Developments

HPC systems with GPUs can provide significantly increased levels of parallel processing for CFD software that is developed using a GPU programming model such as CUDA, OpenACC, or various API approaches. The programming strategies of choice can depend on several factors, and examples will be presented that include:

- <u>FUN3D:</u> GPU developments in CUDA where 1 GPU (V100) achieves equivalent performance of 24 CPUs (SKL, 11 nodes) for a NASA CRM configuration of 14M cells. Strong scaling for ~500 x V100 GPUs for a 6.5B cell Mars lander model on ORNL Summit system, #1 on Top500.
- <u>OpenFOAM:</u> GPU developments in CUDA that demonstrate nominal ~3x speedups single GPU (V100) over single CPU, and intranode strong scaling efficiency of 65% for a 25M cell model.

NVIDIA developed libraries are also available, and provide GPU acceleration at the math kernel level, (cuBLAS, cuSPARSE, etc.) and full linear solver level with cuSOLVER and open source AmgX with a variety of Krylov and multigrid methods for unstructured grid CFD solutions.

3 AI Developments

AI research has given rise to applications of physics-informed neural networks (PINNs) that leverage the underlying laws of physics, often described in the form of partial differential equations (PDEs), to solve forward, inverse/data-assimilation and model discovery problems. Advantages over traditional methods of solving PDEs include (i) usability: not requiring arduous meshing, (ii) speed: ability to solve multiple geometries simultaneously, (iii) scalability: embarrassingly parallel across clusters of GPUs, and (iv) expertise: ability to leverage training experience.

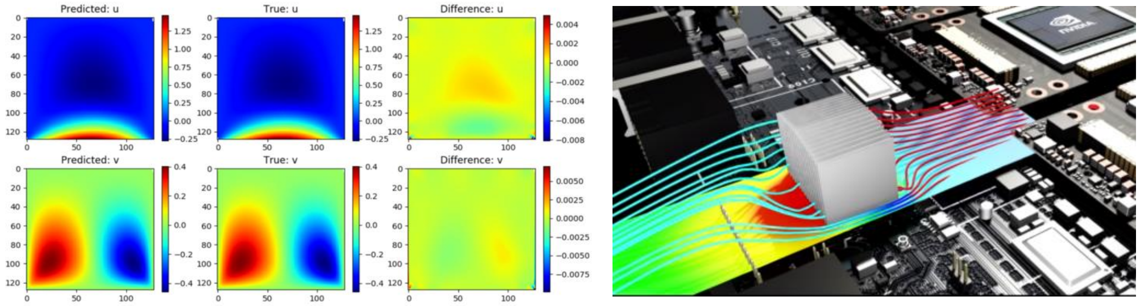

Figure 1: Validation of PINN Predicted results vs. True by an open source CFD solver, for a lid driven cavity, and PINN Predicted thermal results of an NVIDIA GPU heat sink design candidate.

NVIDIA thermal management engineers are applying PINN research in an evaluation to improve the design and effectiveness of heat sinks where thousands of design configurations can be analyzed within hours as opposed to weeks with traditional CFD simulations. The PINN method provides a forward solution of parameterized, multi-physics problems, starting with only the geometry and other physical parameters like material properties, boundary conditions, etc., typical of any RANS-based CFD solver.

Standard neural networks that are driven by data alone are inadequate for modeling such engineering multi-physics problems on various geometries. Considerations must include specific features of a neural network architecture, such as – sampling insensitivity, impact of the order of derivatives on the network structure, weighting the various PDEs for loss convergence acceleration, activation functions that do not reduce down to constants, or vanish when differentiated, and gradients and discontinuities due to geometrical effects and considerations of local versus global mass balance equations. Requirements will be examined and tradeoffs presented for this PINN approach vs. a conventional CFD approach.

4 Conclusion and Future Work

As CFD simulation demands increase and motivate the need for more transients, higher-resolutions, and multi-scale, multi-physics simulations, GPUs will become an essential HPC technology. Based on current trends, GPU-based HPC combined with novel AI techniques will enable a level of applied CFD that can grow as a common practice to support engineering design and optimization procedures.

References

[1] M. Zubair, E. Nielsen, J. Luitjens and D. Hammond, "An Optimized Multicolor Point-Implicit Solver for Unstructured Grid Applications on Graphics Processing Units," 6th Workshop on Irregular Applications: Architecture and Algorithms (IA3), Salt Lake City, UT, 2016, pp. 18-25.

[2] M. Raissi, P. Perdikaris, and G. E. Karniadakis. "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics* 378 (2019): 686-707.