Scalable Interpolation on GPUs for Thermal Fluids Applications

Neil Lindquist[c], Paul Fischer[a,b], Misun Min[a]

[a] *Mathematics and Computer Science Division, Argonne National Laboratory, Lemont, IL*
[b] *Department of Computer Science, University of Illinois Urbana-Champaign, Urbana, IL*
[c] *Department of Computer Science, University of Tennessee, Knoxville, TN*

Abstract

Most computational fluid dynamics (CFD) codes are based on Eulerian representations, where the equations are discretized in a fixed frame of reference through which the fluid flows. A common aspect of this approach is that the solution is typically represented on a fixed grid, or mesh, comprising nodes where solution values are represented. Differential operators are approximated by relationships between neighboring nodal values. These approaches can be quite efficient both for forward operator evaluation (e.g., differentiation) and for system solution (e.g, solving a Poisson problem, which is usually done iteratively for 3D problems). Post-processing quantities, such as drag, heat-transfer coefficients, and turbulence statistics, are also readily accessible with this Eulerian-based representation as one can use the mesh-based representation to evaluate gradients, boundary integrals, and volume integrals.

There are many instances, however, where there is a distinct need to evaluate *off grid* quantities. Applications include overset grids, particle tracking, and numerous solution [1, 2**?**], interrogation techniques such as profile plotting and 2D slices through the domain for visualization or spectral analysis of turbulent flows. Such applications require general, unstructured, interpolation from a predefined set of mesh points to an arbitrary array of values.

In this talk, we present a fast and robust GPU-enable interpolation utility for NekRS, a spectral-element, Navier-Stokes solver [3]. This utility is based on the `findpts` and `findpts_eval` routines from the *gslib* library [4]. We address GPU performance issues, parallel scalability, and relative per-point costs of general interpolation versus Navier-Stokes time advancement. We illustrate the utility of this routine for an overset grid application and for Lagrangian particle tracking.

References

[1] G. Chesshire, W. Henshaw, Composite overlapping meshes for the solution of partial differential equations, J. Comput. Phys. 90 (1990) 1–64.

[2] B. Merrill, Y. Peet, P. Fischer, J. Lottes, A spectrally accurate method for overlapping grid solution of incompressible Navier-Stokes equations, J. Comput. Phys. 307 (2016) 60–93.

[3] P. Fischer, S. Kerkemeier, M. Min, Y. Lan, M. Phillips, T. Rathnayake, E. Merzari, A. Tomboulides, A. Karakus, N. Chalmers, T. Warburton, Nekrs, a gpu-accelerated spectral element navier-stokes solver, CoRR abs/2104.05829 (2021). `arXiv:2104.05829`.
URL `https://arxiv.org/abs/2104.05829`

[4] gslib: Gather-scatter library (2020).
URL `github.com/Nek5000/gslib`

March 28, 2022