Developing a Modern CFD Framework with Parallel Algorithms and Mesh Adaption

J. McKee, Y. Mileyko, A. Fisher, and A. Koniges
Corresponding author: koniges@hawaii.edu

University of Hawaii USA.

Abstract: We discuss recent advances in a Computational Fluid Dynamics (CFD) framework that uses a combination of of Arbitrary Langrangian Eulerian (ALE) Dynamics and Adaptive Mesh Refinement (AMR). Significant updates to the frame work to allow it to build efficiently and scale appropriately on the latest high performance computing (HPC) platforms are necessary to adapt the framework to modern processors with GPUs. We also describe some of the more unusual algorithms that are available in the framework such as those which model surface tension effects in three dimensions. We consider how these algorithms will scale on these latest platforms such as the new Perlmutter system at NERSC which is a HPE Cray EX supercomputer. This system is heterogeneous with both GPU-accelerated and CPU-only nodes, and we consider which nodes are most appropriate for applications in a variety of areas.

Keywords: Numerical Algorithms, Computational Fluid Dynamics, Surface Tension, Mesh Adaptation, ALE Methods.

1 Methodology

Our CFD framework, known as PISALE (Pacific Island Structured-AMR with ALE), uses a staggered-grid, Lagrangian formulation with position and velocity being nodal variables and density, internal energy, temperature, pressure, strain, and stress being zonal (cell centered) variables. In this Lagrangian formulation (in vector and indicial notation $i, j, k = 1, 2, 3$) we have:

$$\frac{\mathrm{D}\rho}{\mathrm{D}t} = -\rho\nabla\cdot\vec{U} = \rho U_{i,i} \tag{1}$$

$$\frac{\mathrm{D}\vec{U}}{\mathrm{D}t} = \frac{1}{\rho}\nabla\cdot\boldsymbol{\sigma} = \frac{1}{\rho}\sigma_{ij,j} \tag{2}$$

$$\frac{\mathrm{D}e}{\mathrm{D}t} = \frac{1}{\rho}V\boldsymbol{s}:\dot{\boldsymbol{\varepsilon}} - P\dot{V} = \frac{1}{\rho}V\left(s_{ij}\dot{\varepsilon}_{ij}\right) - P\dot{V} \tag{3}$$

where

$$\frac{\mathrm{D}}{\mathrm{D}t} = \frac{\partial}{\partial t} + \vec{U}\cdot\nabla$$

is the substantial derivative, ρ is the density, $\vec{U} = (u, v, w)$ is the material velocity, t is time, $\boldsymbol{\sigma}$ is the total stress tensor, P is the pressure, e is the internal energy, V is the relative volume ($\rho V = \rho_0$ where ρ_0 is the reference density), $\boldsymbol{s}$ is the deviatoric stress defined as,

$$s_{ij} = \sigma_{ij} + P\delta_{ij} \tag{4}$$

where δ is the Kronecker delta, and $\dot{\boldsymbol{\varepsilon}}$ is the strain rate tensor defined as

$$\dot{\varepsilon}_{ij} = \frac{1}{2}\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right) \tag{5}$$

An interface reconstruction algorithm is used for multiple materials, and a VOF (volume of fluid) method for the remapping of the Lagrangian grid. During a simulation, the volume fraction of each material in a mixed zone is stored and the actual interface is only reconstructed as needed, such as during mesh refinement. Volume fractions in mixed zones are used for weighting the pressures, densities, stresses, etc., to obtain composite quantities. Volume fractions are used during refinement of neighboring zones to determine the orientation of each interface and the actual location is determined by the volume fraction in the zone. A 2nd-order predictor-corrector model is utilized for time integration. We mix the ALE scheme with adaptive mesh refinement (AMR) and in the code both the ALE and the AMR schemes can work independently to give zonal modification and potentially improve accuracy and/or runtime to solution.

Surface tension models within the PISALE framework have been implemented in a few different ways [1]. One is implemented by adding a third-order space derivative to the stress that is derived thermodynamically based on the Van der Waals-Cahn-Hilliard free energy. A second approach is based on two-fluid VOF models, focusing on an implementation which calculates the curvature based on the height function approach [2,3]. Although this is a two fluid model, PISALE uses a version that converts a single material density into the volume of fluid methodology and utilizes the same algorithm for a single-fluid implementation and the method is extended to three dimensions. We describe the behavior of these methods in the context of efficiency and parallelization.

2 Discussion

This paper focuses primarily on how to apply this framework to a variety of modern applications such as hypersonic flows, high energy density experiments, and droplets for viral spread. We discuss which parts of the code need updating to run efficiently on modern processes and how are parallelization schemes adapt to very large problems on current HPC architectures. A major focus of the paper is the surface tension models, and we review the methodologies for including surface tension in our calculations.

References

[1] W. Liu, A. Koniges, K. Gott, D. Eder, et al.. Surface tension models for a multi-material ale code with amr. *Int Computers and Fluids*, 2017;151:91–101.

[2] J U. Brackbill, D. B. Kothe, C. A. Zemach. Continuum method for modeling surface tension. *J Comput Phys* 1992;100:335–54.

[3] S. J. Cummins, M. M. Francois, D. B. Kothe. Estimating curvature from volume fractions. *Comput Struct* 2005;83:425–34.