A HIGH PERFORMANCE PARALLEL NAVIER-STOKES SOLVER FOR INCOMPRESSIBLE, NONISOTHERMAL, NEWTONIAN FLUIDS

PHILIP MURZYNOWSKI

1. Abstract. The aim of this project is to develop a high performance and parallel implementation of a Navier-Stokes Solver for single-phase, nearly incompressible, nonisothermal, and Newtonian fluids in three dimensions. The motivations for this particular topic center around the opportunity for exposure to computational fluid dynamics, experience in the implementation of an accurate and efficient partial differential equation solver, and a bias towards simulations with subjectively interesting graphical output. The problem is constrained to single phase, approximately incompressible, and Newtonian fluids to narrow the scope to a project appropriate for a single semester. The nearly incompressible assumption is defined through the Boussinesq approximation, which states that the flow field is not affected by variations in density except for when they result in buoyancy forces, and thus allows the incorporation for buoyancy driven flows and a more accurate simulation for a nonisothermal fluid [3]. Compressible Navier-Stokes equations are more prone to instability due to their highly nonlinear nature, therefore the Boussinesq approximation has the additional advantage of providing more stable solutions under conditions with moderate temperature differentials while lowering memory usage as the number of variables needed to solve the equations with respect to a varying density is decreased [3][4]. Consequently, excepting the buoyancy term and necessary conservation equations, the implementation centrally relies on the incompressible form of the Navier-Stokes Equations.

The core of the implementation makes use of Finite Volume Methods (FVM), subdividing the region of interest into an array of cuboids that are uniformly spaced and of equivalent size, shape, and orientation, and employing a staggered grid approach which specifies certain quantities at the faces of the cells instead of the centers. The staggered grid is utilized to circumvent odd-even decoupling between velocity and pressure and therefore decrease potential coarsening of the simulation [6], while the structured meshes are implemented as they are more readily amenable to the implementations of cache oblivious and parallel cache-efficient algorithms based on stencil decompositions due to their more regular definition in regards to stencils and lower overhead related to geometric computations. Specifically, as the evolution of a cuboid is dependent on the cuboids immediately neighboring it, a local region of cuboids can be brought into cache, and instead of updating all the cuboids by one timestep and loading another set into the cache, the inner cuboids can be updated by a number of timesteps proportional to their depth inside the local cuboid, known as a trapezoidal decomposition, making for fewer cache misses. [9][5]. Even though the number of cuboids within the local region that can be updated another timestep for each timestep decreases with increasing dimension, the approach is amenable to parallelization and can provide noticeable differences in performance for low dimensions.[5]

The underlying algorithm utilized to update the cuboids is the PISO algorithm, from the SIMPLE family of algorithms. The original SIMPLE algorithm uses the momentum and continuity equations to derive an equation for pressure and then satisfy the continuity equation by deriving a corrector for the velocity field using explicit methods, and iteratively updates pressure and velocity until convergence to a

specified tolerance as part of an implicit method. [8][1]. However, the closely related PISO algorithm variation is used as has the advantage that it has a tighter loop is more suitable for unsteady or transient flows. [2]. The SIMPLE family is chosen as it separates the update matrix into a diagonal and nondiagonal component, but only requires the inversion of the diagonal component, and the algorithms contained within it are largely modifiable for the incorporation of heat transfer [8][7].

Performance is compared to benchmarked implementations using similar methods and potentially spectral and pseudo spectral implementations with similar timescales, environments, and fluid parameters.

REFERENCES

[1] Ansys Inc., *Ansys Pressure-Velocity Coupling*, 2009, https://www.afs.enea.it/project/neptunius/docs/fluent/html/th/node373.htm (accessed 2020/10/25).

[2] Barton, I.E, *Comparison of SIMPLE- and PISO-type algorithms for transient flows*, Int. J. Numer. Meth., 26 (1998), pp. 459–483, https://doi.org/10.1002/(SICI)1097-0363(19980228)26:4⟨459::AID-FLD645⟩3.0.CO;2-U.

[3] Comsol Multiphysics Cyclopedia, *COMSOL Software*, 2017, https://uk.comsol.com/multiphysics/boussinesq-approximation (accessed 2020/10/20).

[4] Donaly D. Gray and Aldo Giorgini, *The validity of the boussinesq approximation for liquids and gases*, International Journal of Mass and Heat Transfer, 19 (1976), pp. 545–551, https://doi.org/10.1016/0017-9310(76)90168-X.

[5] MIT 6.172 Lecturers 2008-2018, *Cache-Oblivious Algorithms*, 2018, https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-172-performance-engineering-of-software-systems-fall-2018/lecture-slides/MIT6_172F18_lec15.pdf (accessed 2020/10/21).

[6] Oceananigans Documentation, *Oceananigans.jl*, 2020, https://clima.github.io/OceananigansDocumentation (accessed 2020/10/20).

[7] Open Foam, *Buoyant Boussinesq Piso Foam*, 2014, https://openfoamwiki.net/index.php/BuoyantBoussinesqPisoFoam (accessed 2020/10/30).

[8] QuickerSim Inc., *Tutorial 2 - Numerics SIMPLE Scheme*, 2017, https://quickersim.com/tutorial/tutorial-2-numerics-simple-scheme/ (accessed 2020/10/28).

[9] Yuan Tang and Rezal Alam Chowdhury and Bradley C. Kuszmaul, Chi-Keung Luk and Charles E. Leiseron, *The Pochoir Stencil Compiler*, SPAA '11, (2011), pp. 117–128, https://doi.org/10.1145/1989493.1989508.