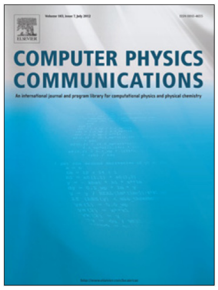
Accepted Manuscript



A high-order cross-platform incompressible Navier–Stokes solver via artificial compressibility with application to a turbulent jet

N.A. Loppi, F.D. Witherden, A. Jameson, P.E. Vincent

PII: S0010-4655(18)30224-8
DOI: https://doi.org/10.1016/j.cpc.2018.06.016
Reference: COMPHY 6548

To appear in: *Computer Physics Communications*

Received date : 22 December 2017
Revised date : 12 June 2018
Accepted date : 15 June 2018

Please cite this article as: N.A. Loppi, F.D. Witherden, A. Jameson, P.E. Vincent, A high-order cross-platform incompressible Navier–Stokes solver via artificial compressibility with application to a turbulent jet, *Computer Physics Communications* (2018), https://doi.org/10.1016/j.cpc.2018.06.016

A High-Order Cross-Platform Incompressible Navier-Stokes Solver via Artificial Compressibility with Application to a Turbulent Jet

N.A. Loppi[a,*], F.D. Witherden[b], A. Jameson[b], P.E. Vincent[a]

[a]*Department of Aeronautics, Imperial College London, SW7 2AZ, United Kingdom*
[b]*Department of Aeronautics and Astronautics, Stanford University, CA 94305, USA*

Abstract

Modern hardware architectures such as GPUs and manycore processors are characterised by an abundance of compute capability relative to memory bandwidth. This makes them well-suited to solving temporally explicit and spatially compact discretisations of hyperbolic conservation laws. However, classical pressure-projection-based incompressible Navier-Stokes formulations do not fall into this category. One attractive formulation for solving incompressible problems on modern hardware is the method of artificial compressibility. When combined with explicit dual time stepping and a high-order Flux Reconstruction discretisation, the majority of operations can be cast as compute bound matrix-matrix multiplications that are well-suited for GPU acceleration and manycore processing. In this work, we develop a high-order cross-platform incompressible Navier-Stokes solver, via artificial compressibility and dual time stepping, in the PyFR framework. The solver runs on a range of computer architectures, from laptops to the largest supercomputers, via a platform-unified templating approach that can generate/compile CUDA, OpenCL and C/OpenMP code at runtime. The extensibility of the cross-platform templating framework defined within PyFR is clearly demonstrated, as is the utility of P-multigrid for convergence acceleration. The platform independence of the solver is verified on Nvidia Tesla P100 GPUs and Intel Xeon Phi 7210 KNL manycore processors with a 3D Taylor-Green vortex test case. Additionally, the solver is applied to a 3D turbulent jet test

*Corresponding author.
E-mail address: n.loppi15@imperial.ac.uk