28th International Conference on Parallel Computational Fluid Dynamics
Parallel CFD2016

INVESTIGATION OF GPU-BASED IMMERSED BOUNDARY METHOD SOLVERS WITH DIRECT FORCING

CHRISTOPHER S. MINAR, KYLE E. NIEMEYER*

School of Mechanical, Industrial, and Manufacturing Engineering
Oregon State University, Corvallis OR, United States
*e-mail: `kyle.niemeyer@oregonstate.edu`, web page: `http://kyleniemeyer.com`

Key words: CFD, GPU, Immersed boundary method, Direct forcing

Abstract. Many engineering applications require fast, accurate solutions of fluid flow around freely moving bodies. The high performance and massive parallelism of GPUs offers a promising alternative to traditional solver acceleration via multicore CPUs. However, fully harnessing GPU parallelism requires specialized algorithms and computing strategies. This work modifies a direct-forcing IBM to model coupled fluid-structure interaction and investigates its behavior on GPUs.

1 INTRODUCTION

Peskin [1] first introduced the immersed boundary method (IBM) to solve the Navier–Stokes equations around complex bodies on a simple Cartesian grid. The original IBM adds a forcing term to the Navier–Stokes equations to account for the immersed boundary and treats Lagrangian body nodes as springs:

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla(\mathbf{u}\mathbf{u}) = -\nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f}\,, \tag{1}$$

where the forcing term $\mathbf{f}$ is only non-zero at nodes adjacent to the body.

Peskin's method works well for modeling elastic bodies but requires a large spring constant for rigid bodies, which can induce stiffness. Direct forcing IBM techniques offer an alternative for better performance with rigid bodies. Instead of treating the body as springs, direct forcing methods solve for the forcing term to enforce no slip directly on the body. Our work builds on the direct forcing method of Fadlun et al. [2], which modifies the velocity solution at the nodes closest to the body u_i using a linear interpolation between the body velocity u_B and the second closest node u_{i+1}.

Our GPU-based solver builds on `cuIBM` of Krishnan and Barba [3], with all computational routines implemented in CUDA. However, rather than their projection method for forcing, we implemented a modified direct forcing method. Linear system solutions for velocity and pressure (Poisson equation) use the `CUSP` conjugate gradient solver, with diagonal and smoothed aggregation preconditioners, respectively. Following the approach commonly used in the literature, the diffusion and convection terms are advanced in time with the implicit Crank–Nicolson and second-order Adams–Bashforth schemes, respectively, and spatial derivatives use central differences.

2 VALIDATION

Figure 1 shows the present work compared to the results of Ghia et al. [4] for lid-driven cavity flow. Figure 2 shows the present work compared to the results of Koumoutsakos and Leonard [5] for flow over an impulsively started cylinder. We will show further validation of our solver for flow around freely moving bodies using a vortex-induced vibration test case, and investigate the computational performance on GPUs.

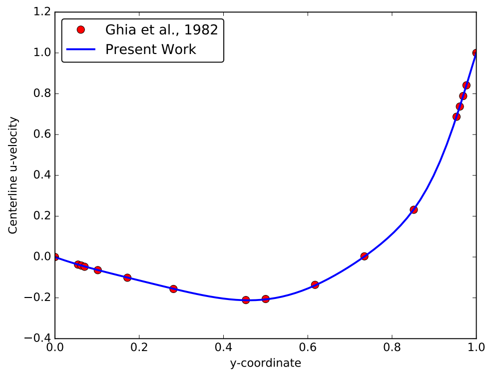

Figure 1: u velocity at vertical centerline velocity for lid-driven cavity flow at Re = 100.

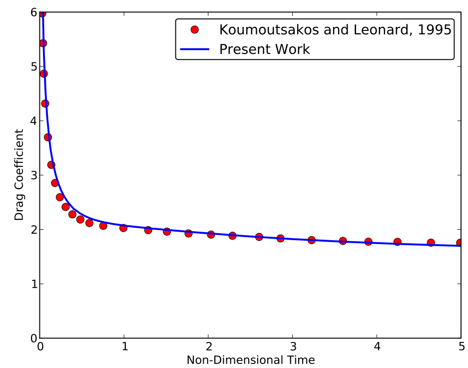

Figure 2: Drag coefficient for impulsively started flow over a cylinder at Re = 40.

REFERENCES

[1] Peskin, C. S. (1972). Flow patterns around heart valves: a numerical method. *J. Comput. Phys.*, 10(2), 252–271.

[2] Fadlun, E. A., Verzicco, R., Orlandi, P., & Mohd-Yusof, J. (2000). Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations. *J. Comput. Phys.*, 161(1), 35–60.

[3] Krishnan, A., & Barba, L. A. (2012). Validation of the cuIBM code for Navier–Stokes equations with immersed boundary methods. figshare. doi:10.6084/m9.figshare.92789.v3

[4] Ghia, U., Ghia, K. N., & Shin, C. T. (1982). High-Re solutions for incompressible flow using the Navier–Stokes equations and a multigrid method. *J. Comput. Phys.*, 48(3), 387–411.

[5] Koumoutsakos, P., & Leonard, A. (1995). High-resolution simulations of the flow around an impulsively started cylinder using vortex methods. *J. Fluid Mech.*, 296, 1–38.