GPU accelerated flow computation by the streamfunction-velocity (ψ-v) formulation

Jiten C Kalita*, Parikshit Upadhyaya* and Murli M Gupta†

*Department of Mathematics, Indian Institute of Technology Guwahati
Guwahati 781 039 Assam INDIA
†Department of Mathematics, The George Washington University
Washington DC 20052 USA

Abstract. In this work, we present an optimization strategy for implementing the BiCGStab iterative solver on graphic processing units (GPU) for computing incompressible viscous flows governed by the unsteady Navier-Stokes (N-S) equations on a CUDA platform. A recently developed ψ-v formulation is used to discretize the biharmonic form of the N-S equation and we obtain remarkable speed up of 40 times on finer grids for the lid-driven square cavity flow. The GPU implementation enabled us to compute the flow in extremely finer grids and very small scales were resolved with remarkable accuracy.

Keywords: Navier-Stokes, GPU, Re, speed-up, ψ-v formulation, BiCGStab, CUDA

PACS: 47.11.Bc,47.32.ck,47.10.ad

INTRODUCTION

GPU, or Graphics Processing Unit, initially developed for the purpose of video gaming, has found tremendous interest amongst researchers in the field of high performance computing over the past decade. This is because of the fact that while the performance improvement of general-purpose microprocessors have slowed down significantly, the GPUs have continued to improve relentlessly. On an average, the ratio between many-core GPUs and multicore CPUs for peak floating point throughput is about 10 to 1. Consequently, GPUs can achieve performance which is very hard for a conventional CPU to accomplish. Therefore, more and more researchers are drawn into GPU computing trying to explore efficient parallel strategies on GPUs.

Several successful GPU implementations of numerical methods for partial differential equations can be found in the existing literature. It includes the simulation of incompressible viscous flows governed by the Navier-Stokes (N-S) equations [7, 8, 9] by all the three basic approaches, viz., finite difference, finite volume and finite element, as well as the Lattice Boltzman models. However, all of them either used the primitive variable or the streamfunction vorticity (ψ-ω) formulation of the N-S equations. In the present work, we apply the recently developed compact finite difference scheme by Kalita and Gupta [6] by the streamfunction-velocity (ψ-v) formulation to discretize the biharmonic form of the N-S equation in GPU. The main emphasis here is the GPU implementation of the Krylov subspace based iterative solver BiCGStab [1] which constitutes the core of the computation and we present an optimization strategy for the same.

As the numerical test case, we have chosen the famous benchmark problem of the flow in a lid-driven square cavity. Computations were performed on grids of size $2^N \times 2^N$ with $N$ ranging from 5 to 10 and we found that the speed up of using GPU vs. CPU goes up as $N$ increases. We also present comprehensive steady-state data for the flow parameters in the finest grids considered here and compare them with existing benchmark results; excellent comparison is obtained in all the cases. This is probably the first time that tertiary vortex data is presented for the lid-driven cavity flow for a grid of size as fine as $1024 \times 1024$. The GPU implementation of the code is carried out by writing it in CUDA C, which is a programming platform developed by NVIDIA corporation [2]. Computations were performed at Colonial One High Performance Computing Initiative of the George Washington University, Washington DC USA, using GPU nodes featuring dual Intel Xeon E-2620 2.0GHz 6-core processors with dual nVidia Kepler K20 GPUs, 128GB of RAM and standard CPU nodes featuring dual Intel Xeon E5-2670 2.6GHz 8-core processors with 64GB of RAM capacity.

The paper is organized in the following way. In section 2, we provide a brief discussion on the numerical scheme and the algorithm used in the computation. Section 3 deals with the GPU implementation of the BiCGStab solver and finally section 4 contains results and discussion.