11th. World Congress on Computational Mechanics (WCCM XI)
5th. European Conference on Computational Mechanics (ECCM V)
6th. European Conference on Computational Fluid Dynamics (ECFD VI)
July 20 - 25, 2014, Barcelona, Spain

DEVELOPMENT OF EXPLICIT UNSTRUCTURED MESH-BASED CFD SOLVER FOR LOW-MACH NUMBER FLOWS USING GRAPHICS PROCESSOR UNITS

Anton Karpenko[1], Vladislav Emelyanov[2] and Konstantin Volkov[3]

[1] Faculty of Mathematics and Mechanics, St Petersburg State University, Universitetsky prospect, Old Petergof, St Petersburg, 198504, Russia, E-mail: aspera.2003.ru@mail.ru
[2] Faculty of Power Engineering, Baltic State Technical University, 1, 1-ya Krasnoarmeyskaya ulitsa, St Petersburg, 190005, Russia, E-mail: vlademelyanov@gmail.com
[3] Centre for Fire and Explosion Studies, Kingston University, Friars Avenue, Roehampton Vale, London, SW15 3DW, United Kingdom, E-mail: k.volkov@kingston.ac.uk

Key words: *Computational Fluid Dynamics, Parallel Algorithm, Finite Volume Method, Unstructured Mesh, Preconditioning, Graphics Processor Unit, CUDA.*

The methods of computational fluid dynamics (CFD) are extensively applied in design and optimization of industrial devices to get more insight into 3D unsteady flows through fluid or gas passages. Accurate prediction of internal flows still remains a challenging task despite a lot of work in this area.

The stagnation in the clock-speed of central processing units (CPU) has led to significant interest in parallel architectures that offer increasing computational power by using many separate processing units. Modern graphics hardware contains such an architecture in the form of the graphics processing units (GPU). These platforms make it possible to achieve speedups of an order of magnitude over a standard CPU in many CFD applications and are growing in popularity [1].

The GPU employs a parallel architecture so each generation improves on the speed of previous ones by adding more cores, subject to the limits of space, heat and cost. CPUs, on the other hand, have traditionally used a serial design with a single core, relying instead on greater clock speeds and shrinking transistors to drive more powerful processors. While this approach has been reliable in the past, it is now showing signs of stagnation as the limit of current manufacturing technology is being reached. Recent CPUs, therefore, tend to feature two or more cores, but GPUs still enjoy a significant advantage in this area for the time being.

Explicit time-marching algorithms are the most convenient ones to be ported on to the GPU. This is because there is no iteration, and the new value of a variable depends only

on the old time values. Hence, the update of a given variable is done independent of variables being updated on other threads. There is no recursive relation between the variables on the threads, since they are all known at the old time step. However, even for explicit algorithms, a few changes are needed for efficiently implementation of numerical algorithms on the GPU. These relate to the use of shared memory and the layout of data structures. Memory coalescing and block size influence the speed achieved. The data should be organized such that adjacent threads access adjacent nodal data. In addition, data should be, where possible, copied to shared memory and re-used as much as possible. Therefore, even explicit algorithm based CFD codes need to be reorganized to take advantage of the GPU architecture.

Possibilities of the use of graphics processing units (GPU) for the simulation of flows of viscous compressible and incompressible fluid are discussed. The finite volume method is applied to solve 3D Euler and Navier–Stokes equations on hybrid and unstructured meshes [2]. The unstructured hybrid code developed uses an face-based data structure to give the flexibility to run on meshes composed of a variety of cell types. The non-linear CFD solver works in an explicit time-marching fashion, based on a Runge–Kutta stepping procedure. Convergence to a steady state is accelerated by the use of low-Mach number preconditioning method for low-speed flows. Parallel computation is implemented using the MPI (clusters) and CUDA technology (GPU). Solution of some fluid dynamics problems on GPUs is presented. Speed-up of solution on GPUs in according to the solution on CPU is compared with the use of different meshes and different methods of distribution of input data into blocks. The results obtained are generally in reasonable agreement with the available experimental and computational data, although some important sensitivities are identified and discussed. Capabilities and accuracy of various finite difference schemes, acceleration efficiency calculations due to parallelization of computational algorithms are compared.

The computational procedure is used as a part of CFD package LOGOS developed in the Institute of Theoretical and Mathematical Physics of the Russian Federal Nuclear Center (Sarov, Russia). LOGOS package is widely used in mechanical engineering and aerospace applications.

REFERENCES

[1] J. Sanders and E. Kandrot. CUDA by example: an introduction to general-purpose GPU programming. Boston, Pearson Education, 2011.

[2] V. Emelyanov, A. Karpenko and K. Volkov. Development of advanced CFD tools and their application to simulation of internal turbulent flows, *Proceedings of the 5th European Conference for Aeronautics and Space Sciences*, 1–15, 2013.