Ab initio NMR Chemical Shift Calculations for Biomolecular Systems Using Fragment Molecular Orbital Method

A Large-scale Two-phase Flow Simulation on GPU Supercomputer

Evolutive Image/Video Coding with Massively Parallel Computing

http://www.gsic.titech.ac.jp/TSUBAME_ESJ
A Large-scale Two-phase Flow Simulation on GPU Supercomputer

Takayuki Aoki* Kenta Sugihara**

* Global Scientific Information and Computing Center, Tokyo Institute of Technology
** Graduate School of Engineering, Tokyo Institute of Technology

Two-phase flows such as gas-water mixing are often observed in nature; however, their numerical simulation is one of the most challenging themes in computational fluid dynamics, and it takes a long computational time. A state-of-the-art interface capture method, sparse matrix solver, higher-order advection scheme, and others have been introduced and entirely implemented for GPU computing. It has become possible to carry out large-scale two-phase flow simulations that have never been achieved before. In a violent air-water flow, small bubbles entrained in the water are described clearly and good performance scalability is also shown for multi-GPU computing.

Introduction and Motivation

Recently movie scenes of violent flows mixing air with water have been produced by computer graphics in Hollywood film productions. It is notable that they carry out larger-scale computations with higher-resolution than those of scientific and engineering works. For two-phase flows, particle methods such as SPH (smoothed particle hydrodynamics) have been used due to the simple algorithm and success of astrophysical N-body simulation in the beginning of GPU computing. Each particle interacts with all particles within the kernel radius to compute the particle motion. In three-dimensional simulation, the number of interacting particles increases, and particle methods have disadvantages from the viewpoints of numerical accuracy, random memory access, and amount of computation. In particular, the sparse matrix for the pressure Poisson equation has a wide bandwidth of non-zero elements in the semi-implicit time integration and is inefficiently solved in such memory distributed systems as multi-node clusters or supercomputers. In addition, there are problems involving non-physical oscillation at the gas-liquid interface, inaccurate evaluation of the surface tension, and large numerical viscosity.

In mesh methods such as FDM (finite difference method), FVM (finite volume method), and FEM (finite element method), the computation for a mesh point, a cell, or an element requires only accesses to some neighboring points. Higher-order numerical schemes are easily applicable to the FDM in structured meshes. In Hollywood film productions, they have changed the particle methods to mesh methods to make realistic water scenes. In a mesh method, the gas and liquid phases are treated as one fluid with different material properties: density, viscosity, and surface tension. It is necessary to introduce an interface capturing technique to identify the different properties. The density changes 1000 times from the gas to the liquid at the interface, and the profile is expressed with a few meshes.

In this article, we show a large-scale gas-liquid two-phase simulation by full GPU computing, which has never been achieved before.

Gas-liquid interface capturing method

The gas-liquid interface is often described as a surface of a three-dimensional identical function. The 3-D information seems to be too much for the 2-D surface information; however, the method makes it easy to express such topology changes as splitting and merging of bubbles. The Level-Set function\(^1\) and the VOF (volume of fluid) are often used as an identical function. The former is a signed distance function from the interface as shown in Fig. 1. In the gas region, it has negative distance, and in the liquid region, it has positive distance. The interface is expressed as the zero-level iso-surface of the smooth profile, which is able to accurately trace the interface.

---

Figure 1: Iso-surface plots of the Level-Set function

**Current position: Japan Atomic Energy Agency**
The Level-Set method cannot guarantee conservation of the volumes of the gas and the liquid. We lose small bubbles and droplets during the computation. On the other hand, the VOF method conserves the volume and has poor interface shapes when the curvature radius comes closer to the mesh size.

We introduce the VOF-based THINC\(^{(3)}\) WLIC\(^{(4)}\) method to our simulation, in which the anti-diffusion keeps the interface sharp. We apply the Level-Set method only to evaluate the interface curvature and the contact angle.

**Incompressible Navier-Stokes Solver**

3-1 Advection term

For the advection terms of the Navier-Stokes equation and re-initializing the Hamilton-Jacobi equation, we used the 5th-order WENO scheme. The high-wavelength filter to preserve monotonicity contributed the numerical stability. Figure 2 illustrates the stencil access of the 5th WENO scheme. When the on-chip shared memory was used as a software-managed cache, we reduced the same access to the off-chip memory (recently the L1 cache is available for this purpose in Fermi-core GPU). For this part, we have achieved more than 300 GFLOPS on the NVIDIA GTX 280.

3-2 Pressure Poisson Solver

It takes a major portion of the computational time to solve the pressure Poisson equation for two-phase flows. In the case of a structured mesh, non-zero elements are located regularly in the sparse matrix; however, their values change 1000 times. We have developed the BiCGStab method in the Krylov sub-space iteration in collaboration with MIZUHO Information and Research Institute as a GPU matrix library. The convergence was drastically improved by coupling with a V-cycle multi-grid preconditioner shown in Fig.3. In the multi-grid process, the ILU method was applied to the smoother with the Red and Black algorithm.

**Single Bubble Rising**

To confirm the gas-liquid two-phase flow simulation, we compared it with the experiment of a single bubble rising. In accordance with the Grace diagram\(^{(5)}\), the bubble shape can be spherical, ellipsoidal, skirted, or dimpled, depending on the dimensionless parameters: Eotvos number (Eo), Morton number (Mo), and Reynolds number (Re). For such parameters, the simulation results are in good agreement with the experimental shapes and rising speeds \(^{(6)}\).

**Figure 2** Stencil of the 5th WENO scheme.

**Figure 3** V-cycle of the multi-grid preconditioner.

**Figure 4** Bubble shapes depending on Eotvos number (Eo) and Morton number (Mo).
It is well known that a milk crown is formed by dropping a droplet into a glass of shallow milk, and its mechanism and the number of fingers is still being discussed. The simulation for a liquid with the same viscosity, surface tension, and impact velocity as the experiment by Krechetnikov \(^1\) was carried out. Figure 5 exhibits the typical shape of the milk crown, and it is found that the simulation reproduces the shapes changing drastically with the impact velocity.

When dropping a droplet on a dry wall, the edge of the milk membrane jumps from the wall, which is typically observed in experiments (see Fig. 6).

To check the simulation for more complex conditions, we studied the dam-break problem as a typical benchmark in collaboration with Prof. Hu’s group of the Research Institute for Applied Mechanics, Kyushu University. By opening the shutter, the dam water launches onto a dry floor, and the speed of the water edge is often examined. In our experiment, the floor ahead of the dam was wet, and a violent wave breaking happened immediately since the inundating water was decelerated on the wet floor and the wave behind overtakes it.

In the vessel with the size of 72cm × 13cm × 36cm, water was filled to a height of 1.8 cm, and a dam that was 15 cm wide and 36 cm high was set initially. We show the time-evolving snapshots of the simulation results with 576 × 96 × 288 mesh in Fig.8. The last image is a snapshot of the experiment.
In our computation, all the components of the two-phase flow simulation have been implemented on CUDA code. The dependent variables such as flow velocity, pressure, Level-Set function, and VOF function were allocated on the on-board GPU memory. The GPU execution function calls were only requested from the host CPU, and we removed frequent CPU-GPU communication, which is a major overhead of GPU computing.

For a large-scale simulation, the computational domain is decomposed into small domains in which each GPU computation runs on the on-board video memory. Similar to multi-node CPU computing, it is required to communicate data among neighbor domains. The communication between the on-board memories of different GPUs consists of three steps through a host memory as shown in Fig. 9.

Since the communication time between GPUs becomes a large overhead for large-scale GPU computing, we have introduced an overlapping technique between communication and computation[9].

We used 60 nodes of TSUBAME 1.2 with two NVIDIA Tesla S1070s per node. The performances of the computations with mesh number 1923, 3843, and 7683 are plotted in Fig. 10. The 4 TFLOPS performance was achieved in single precision with 108 GPUs[10].

The simulation reproduces the wave breaking process well. When the water impacted with the side wall, it was found that many small bubbles are entrained into the water[8]. This simulation is also available for evaluation of the damage caused by a tsunami impacting on a construction.
Gas-liquid two-phase flow simulation is one of the challenging CFD topics, and we executed a simulation on the GPU supercomputer TSUBAME. A full GPU computation makes it possible to carry out large-scale computing that has never been done before. However, there are still problems to be solved, for example, sub-grid modeling of LES (large eddy simulation) for small bubbles in high-Reynolds turbulent flows.

Acknowledgements

This research was supported in part by KAKENHI, Grant-in-Aid for Scientific Research (B) 23360046 from the Ministry of Education, Culture, Sports, Science and Technology (MEXT), two CREST projects, “ULP-HPC: Ultra Low-Power, High Performance Computing via Modeling and Optimization of Next Generation HPC Technologies” and “Highly Productive, High Performance Application Frameworks for Post-Petascale Computing” from Japan Science and Technology Agency (JST), and JSPS Global COE program “Computationism as a Foundation for the Sciences” from Japan Society for the Promotion of Science (JSPS).