Bubble Merging and Dam Break Simulations using OpenACC/OpenMP Hybrid Structure

Chih-Wen Chang[1,*], Pao-Hsiung Chiu[2], Chia-Yung Jui[1], Chia-Chen Kuo[1]
1,* National Center for High-performance Computing, 0903040@nchc.narl.org.tw
2 Institute of High Performance Computing, chiuph@ihpc.a-star.edu.sg
1 National Center for High-performance Computing, 1403036@nchc.narl.org.tw
1 National Center for High-performance Computing, cckuo@nchc.narl.org.tw

ABSTRACT

We utilize the OpenACC/OpenMP accelerated structure to deal with the three-dimensional two-phase incompressible Navier-Stokes equations. For the time-saving aspect, the Poisson equation solver for pressure correction term is employed to the GPU computation of OpenACC directives and the symmetric multiprocessing (SMP) parallelization for accelerating the rest of calculations. To verify this proposed algorithm, the bubble merging and dam break problems are tested and demonstrated to have good agreements with the referenced results. It also displays about **7.32x** faster for the bubble merging problem and **3.75x** faster for the dam break problem by using the current framework.

***Keywords*:** Bubble Merging Problem, Dam Break Problem, Incompressible Navier-Stokes Equations, OpenACC, OpenMP, Graphics Processing Unit (GPU).

1. Introduction

Recently, [Chang et al. 2014] have utilized the GPU structure to tackle the milkcrown problem. For the detailed derivations of proposed method and the three-dimensional two-phase incompressible Navier-Stokes equations, the reader can refer to [Chiu and Sheu 2009; Kuo et al. 2010; Chiu and Lin 2011]. Furthermore, there were many applications by utilizing GPU construction in computational fluid dynamics as follows: [Linear algebra operators; Krüger and Westermann (2003)], [Programmable graphics hardware; Goodnight et al. (2003)], [Navier-Stokes flow solvers; Harris (2004)], [Unstructured grid solver; Corrigan et al. (2009)], [High-order finite difference schemes and Chorin's projection approach; Zaspel and Griebel (2013)]. All the above-mentioned references have presented time-saving techniques by utilizing the GPU implementation.

In this study, we combine the GPU(OpenACC) and multi-CPU(OpenMP) cores to develop the proposed algorithm. Good performance can be acquired by applying the GPU to accelerate the Poisson solver with employing the multi-CPU cores for the rest of calculations. OpenACC allows that programmers can provide the simple directives to the compiler and identify which areas of code to accelerate. By employing the parallelism to the compiler, the directives consent the compiler to operate the detailed work of mapping the computation onto the accelerator. Besides, without rewriting the original program is the main advantage of OpenACC and empowers programmers to make the performance easier, more efficient and portable than the Array Building Blocks, CUDA, OpenCL and FLIP with GPU.

2. Results and Discussion

The bubble merging problems with coaxial coalescence and oblique coalescence are contemplated. A single computing node with two Intel Xeon E5-2687W CPUs and one NVIDIA Tesla C2075 GPU card is used for this present case. Furthermore, we display these good numerical results from t = 0 to t = 6 in Figure 1 and the results of dam break problem with t = 0~10 in Figure 2. We only show the differences with OpenACC and **Maya** software of t = 0~6(bubble merging problem) and **RealFlow** software of t = 0~60(dam break problem) in Figures 3-5 because of no "upload video" function in the submission website. Besides, we enlarge the meshes to 160x320x160 of bubble merging problem, and the compared results are demonstrated in Table 1.

3. Conclusions

We apply the OpenACC/OpenMP framework to cope with the bubble merging problem and dam break problem. Moreover, the benchmark problems have been numerically investigated with the SMP parallelization. We can acquire **7.32x** and **3.75x** faster than the sequential program on a single computing node by employing 8 CPU cores and one GPU card, respectively. In addition, to the authors' best knowledge, there is no published literature presenting that the numerical schemes with this framework for these two problems. In the future, we will carry out the plug-in and combine it with the **Maya** software.

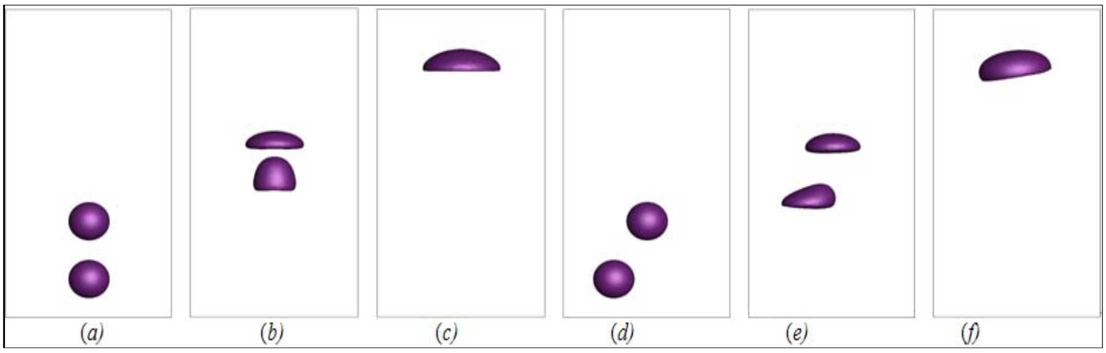

Figure 1. For the coaxial and oblique bubble merging problems at different time with 80x160x80 meshes in the predicted time-evolving surfaces, respectively. (a) t = 0.0; (b) t = 3.0; (c) t = 6.0; (d) t = 0.0; (e) t = 3.0; (f) t = 6.0.

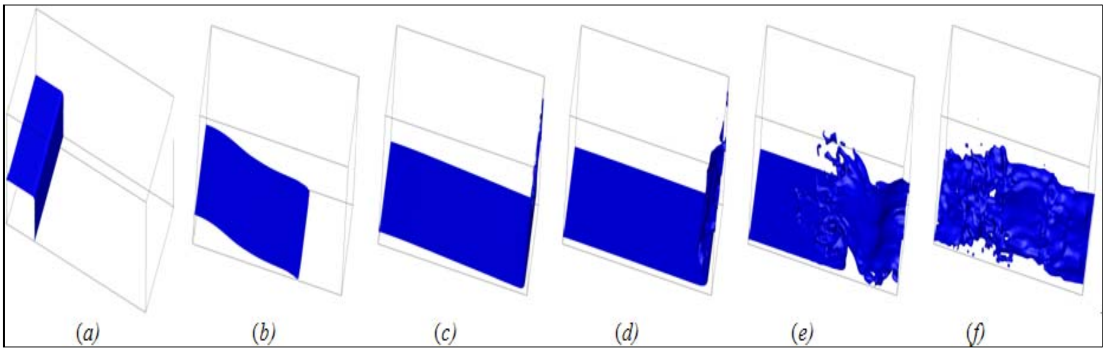

Figure 2. For the dam break problem at different time with 200x100x100 meshes in the predicted time-evolving surfaces. (a) t = 0.0; (b) t = 2.0; (c) t = 4.0; (d) t = 6.0; (e) t = 8.0; (f) t = 10.0.

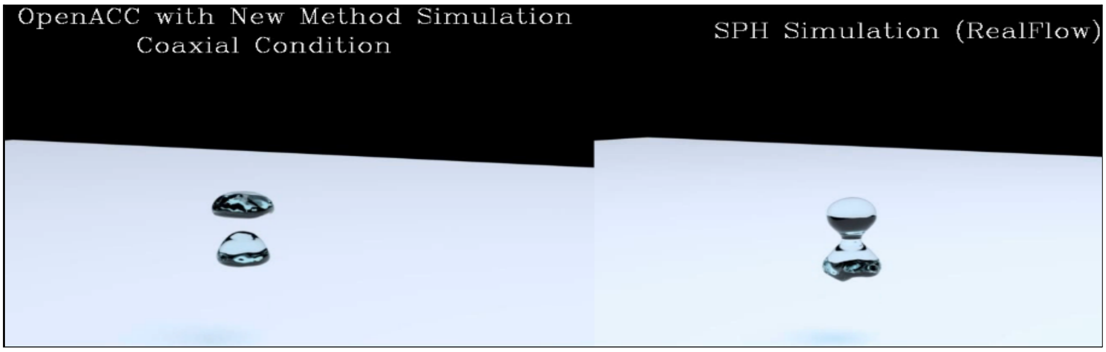

Figure 3. The picture was captured from our video.

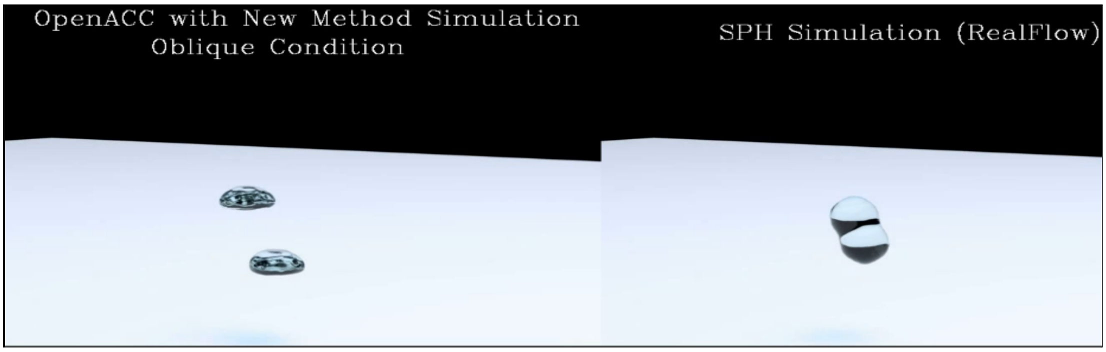

Figure 4. The picture was captured from our video.

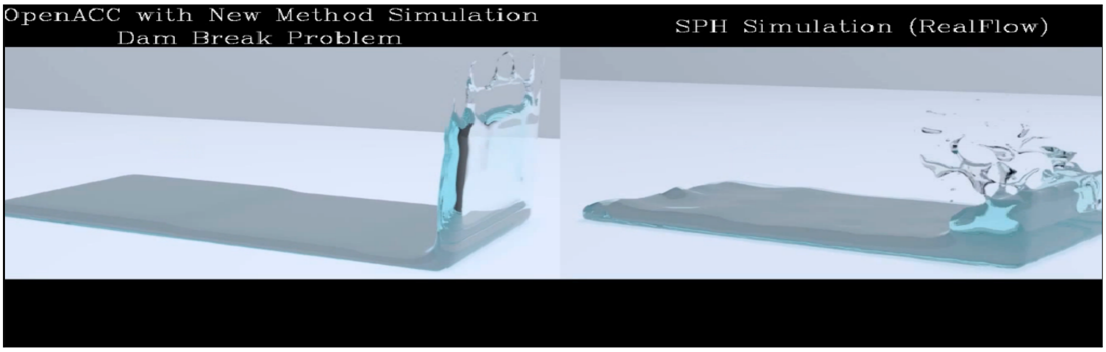

Figure 5. The picture was captured from our video.

Table 1. Average elapsed time per time step and speed up in the bubble merging problem.

Mesh size	Sequential	8 cores+GPU	**Speedup**
40x80x40	8 sec	4 sec	**1.96**
80x160x80	230 sec	37 sec	**6.22**
160x320x160	6788 sec	927 sec	**7.32**

References

Chang, C.-W., Chiu, P.-H., & Kuo, C.-C. (2014). Milkcrown simulation using OpenACC accelerated framework. *SIGGRAPH Asia* 45.

Chiu, P.-H., & Sheu, Tony W.-H. (2009). On the development of a dispersion-relation-preserving dual-compact upwind scheme for convection-diffusion equation. *Journal of Computational Physics, 228*(10), 3640-3655.

Kuo, S.-H., Chiu, P.-H., Lin, R.-K., & Lin, Y.-T. (2010). GPU implementation for solving incompressible two-phase flows. *International Journal of Information and Mathematical Sciences 6*(4), 241-249.

Chiu, P.-H., & Lin, Y.-T. (2011). A conservative phase field method for solving incompressible two-phase flows. *Journal of Computational Physics, 230*(1), 185-204.

Krüger, J., & Westermann R., (2003). Linear algebra operators for GPU implementation of numerical algorithms. *ACM Trans. Graphics, 22* (3), 908-916.

Goodnight, N., Woolley C., Lewin G., Luebke D., & Humphreys G., (2003). A multigrid solver for boundary value problems using programmable graphics hardware. *Graphics Hardware,* 1-11.

Harris, M. J., (2004). Fast fluid dynamics simulation on the GPU. *GPU Gems,* 637-665 (Chapter 38).

Corrigan, A., Camelli, F., Löhner, R., & Wallin, J., (2009). Running unstructured grid based CFD solvers on modern graphics hardware. *19th AIAA Computational Fluid Dynamics.* AIAA Paper

2009-4001.

Zaspel, P., & Griebel, M., (2013). Solving incompressible two-phase flows on multi-GPU clusters. *Computers & Fluids, 80*, 356-364.