Multiphase Fluid Simulations on a Multiple GPGPU PC
Using Unsplit Time Integration VSIAM3

Akio IKEBATA1,* Yoshibiko MURAOKA1 and Feng XIAO2

1 TOTO LTD., 2-1-1, Nakashima, Kokurakita-ku, Kitakyushu 802-8601, Japan
2 Tokyo Institute of Technology, 4259 Nagatsuta, Midori-ku, Yokohama, 226-8502, Japan

This talk presents the implementation of simulations on multiphase fluid dynamics on hardware of multiple GPGPU architecture by using robust and efficient numerical methods. An unsplit formulation for the advection computation is proposed to take the place of the original split formulation in the so-called VSIAM3 method. The new formulation improves dimensional symmetry of numerical results without losing the advantages of the original formulation. We also devised a numerical scheme named STAA to capture the moving interface between different fluids. The volume of fluid function is transported by the above advection scheme, and then modified by an anti-diffusion step based on a well designed numerical flux that rigorously conserves the mass and effectively eliminate the numerical smearing around the interface. Our experiments show that it is able to maintain the VOF function in a well-regulated shape with compact transition layer between different fluids. We have developed a 3D numerical model for multi-fluid simulations with above methods incorporated, and implemented it on a multiple GPGPU computer. It is shown that the present model is able to adequately make use of the advantage of the GPGPU architecture. Through this research, we have built a numerical tool with adequate reliability for sanitary ware product design. Computational efficiency which is of particular importance for real-case design process is achieved by implementing the code on a multiple GPGPU hardware.

KEYWORDS: multiphase flow, numerical method, finite volume method, CIP, VSIAM3, GPGPU, OpenCL

I. Introduction

Tremendous efforts for developing numerical simulation technology for gas and liquid multiphase fluid have been dedicated for a long time. In the so-called one-fluid model, the interfaces between different fluids are explicitly solved by the interface capturing (or tracking) schemes. Among the representative ones of this category, VOF (volume of fluid) and level set methods have been suggested to identify different phases.1–4 These methods transport the identification functions through the advection equations based on the velocity field computed from the Navier-Stokes equations. In practical applications, interface capturing schemes of both numerical accuracy and computational efficiency are of much interests. Following the recent trend in the expanding use of the new type hardware architectures, such as the GPGPU, numerical schemes that are well suited for multi-thread processing are particularly demanded. Level set method is suitable for GPGPU because the calculation of the scheme can be constructed as the assembly of simple vector processing. However, the mass conservation is not inherently guaranteed in the level set method. As a consequence, small liquid drops and bubbles tend to be lost. At this point, the VOF method that exactly conserves the fluid volume is more attractive. However, the geometrical reconstruction procedure in the conventional VOF method has algorithmic complication which prevents it from being well oriented for the GPU processing.

In this work, we propose a new advection scheme using multi-moment concept which is very suitable for GPU computing. Different from the conventional VOF method, an interface capturing scheme that doesn’t require geometrical reconstruction has been also suggested. An improved VSIAM3 model for multiphase simulation has been developed and implemented on a multiple GPU computer system. An integrated simulation tool which is able to use the advantage of GPU acceleration has been established for practical applications in product design process.

II. The Advection Transport Scheme

We consider the following two-dimensional advection equation

\[
\frac{\partial \phi}{\partial t} + u \frac{\partial \phi}{\partial x} + v \frac{\partial \phi}{\partial y} = 0.
\] (1)

It can be alternatively written as

\[
\frac{\partial \phi}{\partial t} + \frac{\partial u \phi}{\partial x} + \frac{\partial v \phi}{\partial y} = \phi \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right). \] (2)

For incompressible flow, it is obvious that the conservation of scalar value \(\phi\) is guaranteed if Eq. (2) is solved by the finite volume method (FVM).

The CIP-CSL (Constrained Interpolation Profile-Conservative Semi-Lagrangian) schemes are a class of
conservative advection schemes of high numerical accuracy and computational efficiency.\textsuperscript{5-9} The most remarkable feature of a CIP-CSL scheme which distinguishes it from the existing methods is using multi-moments and treating them simultaneously as the computational variables. When implementing the CIP-CSL schemes in multi-dimensions, dimension splitting is usually used to get around the complexities of integrating the piece-wisely constructed interpolation functions over distorted volume. An economical splitting is introduced in the study to simplify the algorithm that uses all moments of three dimensions.\textsuperscript{2-8} The simplified scheme makes use of only two types of moments, i.e. the volume integrated average (VIA) and surface integrated average (SIA) for discretizations in three dimensions. The resultant numerical formulations are called VSIAM3 (Volume/Surface Integrated Average based Multi-Moment Method). In two dimensional context, we define two kinds of moments, VIA and SIA as follows,

\[ \nabla \phi_{ij} \equiv \frac{1}{\Delta x \Delta y} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \phi(x, y) dxdy \]  

\( (3) \) and

\[ \nabla \phi_{i+\frac{1}{2}, j} \equiv \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \phi(x_{i+\frac{1}{2}}, y) dy, \]  

\( (4) \)

\[ \nabla \phi_{ij+\frac{1}{2}} \equiv \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \phi(x, y_{j+\frac{1}{2}}) dy. \]  

\( (5) \)

Instead of the splitting in the existing VSIAM3, we propose here an alternative to update the SIA moment and numerical fluxes in an unsplit manner.

In the unsplit method, SIA is calculated by semi-Lagrangian scheme based on Eq. (1), while VIA is updated with Eq. (2) by a finite volume formulation over the control volume (mesh element).

As shown in Fig. 1 we approximate the numerical fluxes by the Gaussian quadrature. The flux on right face of the control volume, for example, is calculated by

\[ F_{x_{i+\frac{1}{2}, j}} = \frac{u \Delta t}{2} \left\{ \nabla \phi^0 \left( x_{i+\frac{1}{2}} - g_1 u \Delta t, y_j - g_1 v \Delta t \right) \right. \]  

\[ + \left. \nabla \phi^0 \left( x_{i+\frac{1}{2}} - g_1 u \Delta t, y_j - g_2 v \Delta t \right) \right\}, \]  

where \( g_1 \equiv (3 - \sqrt{3})/6 \) and \( g_2 \equiv (3 + \sqrt{3})/6 \). In the same way, the flux on bottom face of the control volume can be calculated by

\[ F_{y_{i,j+\frac{1}{2}}} = \frac{v \Delta t}{2} \left\{ \nabla \phi^0 \left( x_i - g_1 u \Delta t, y_{j+\frac{1}{2}} - g_1 v \Delta t \right) \right. \]  

\[ + \left. \nabla \phi^0 \left( x_i - g_2 u \Delta t, y_{j+\frac{1}{2}} - g_2 v \Delta t \right) \right\}. \]  

\( (7) \)

The VIA is then updated by

\[ \nabla \phi^{n+1}_{ij} = \nabla \phi^0_{ij} \left( \frac{\partial F_{x_{i+\frac{1}{2}, j}}}{\Delta x} + \frac{\partial F_{y_{i,j+\frac{1}{2}}}}{\Delta y} \right) \]  

\[ + \Delta t \nabla \phi^0_{ij} \left( \frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} \right)_{ij}. \]  

\( (8) \)

On the other hand, SIA is updated by

\[ \nabla \phi^{n+1}_{i+\frac{1}{2}, j} = \nabla \phi^0_{i+\frac{1}{2}, j} \left( x_{i+\frac{1}{2}} - u \Delta t, y_j - v \Delta t \right) \]  

\[ + \nabla \phi^0_{i+\frac{1}{2}, j} \left( x_{i+\frac{1}{2}} - u \Delta t, y_{j+\frac{1}{2}} - v \Delta t \right). \]  

\( (9) \)

\( (10) \)

The key issue to compute Eqs. (6), (7), (9) and (10) is to properly update the numerical solutions for SIA variables. Without losing the generality, we assume a locally frozen velocity and denote the effective velocity components by \( u' \) and \( v' \). The solutions of \( \nabla \phi^0 \left( x_{i+\frac{1}{2}} - u' \Delta t, y_j - v' \Delta t \right) \) and \( \nabla \phi^0 \left( x_i - u' \Delta t, y_{j+\frac{1}{2}} - v' \Delta t \right) \) can be computed by the following two-step procedure as shown in Fig. 2.

\[ \text{Step 1} \quad \text{In} \ x \ \text{direction, the intermediate values for SIA} \ \nabla \phi^0 \left( x_{i+\frac{1}{2}} - u' \Delta t, y_j \right) \ \text{and flux} \ F_x^* \left( x_{i+\frac{1}{2}} - u' \Delta t, y_j \right) \ \text{are calculated by one dimensional CIP-CSL scheme. In the same manner, those in} \ y \ \text{direction,} \ \nabla \phi^0 \left( x_i, y_{j+\frac{1}{2}} - v' \Delta t \right) \ \text{and} \ F_y^* \left( x_i, y_{j+\frac{1}{2}} - v' \Delta t \right), \ \text{are calculated.} \]

\[ \text{Step 2} \quad \text{From the intermediate results obtained at step 1, we update the SIA value by,} \]

\[ \nabla \phi^0 \left( x_{i+\frac{1}{2}} - u' \Delta t, y_j - v' \Delta t \right) \]  

\[ = \nabla \phi^0 \left( x_{i+\frac{1}{2}} - u' \Delta t, y_j \right) - \Delta t \left( v' \frac{\partial \phi}{\partial y} \right) \]  

\[ = \nabla \phi^0 \left( x_{i+\frac{1}{2}} - u' \Delta t, y_j \right) - \Delta t \left( \frac{\partial v' \phi}{\partial y} - \phi \left( \frac{\partial v'}{\partial y} \right) \right), \]  

\( (11) \)

The flux gradient in \( y \) direction on the right hand side of (11) can be approximated by a linear interpolation in terms of the intermediate flux function at step 1 as,

\[ \frac{\partial v' \phi}{\partial y} = \frac{F^+ - F^-}{\Delta y} \]  

\( (12) \)

with

\[ F^+ \equiv \left( \frac{1}{2} + \xi \right) F_y^* \left( x_{i+1}, y_{j+\frac{1}{2}} - v' \Delta t \right) \]  

\[ + \left( \frac{1}{2} - \xi \right) F_y^* \left( x_{i+1}, y_{j+\frac{1}{2}} - v' \Delta t \right) \]  

\( (13) \)
By repeating above procedure, fluxes can be calculated by Eqs. (6) and (7) with the effective velocity \( u' = g_1.v_t \) and \( v' = (g_1,2 v_t) \), and thus the VIA is updated by Eq. (8). The SIA is updated by letting \( u' = u \) and \( v' = v \).

This scheme can be straightforwardly extended to three dimensions. To distinguish the present method from the existing VSIAM3 algorithm, we call this scheme "UTI-VSIAM3" (Unsplit Time Integration VSIAM3). Because of unsplit time integration, the dimensional symmetry is much improved by the present scheme. We use UTI-VSIAM3 as the transport scheme for both moving interface between different fluids and the fluid dynamic solver.

### III. Numerical Tests

We tested the present scheme by solving 2D advection problem. The advection velocity \( u \) and \( v \) are specified to give a solid rotation around the center of computational domain with a mesh of 75 × 75. We specified the initial advected profile being centered over the computational domain and having a shape of letter "T" (for TOTO as the name of our company) with discontinuous jumps between 0 and 1.

The result of UTI-VSIAM3 under CFL=0.3 (775 cycle per rotation) is shown in Fig. 3. It is obvious that the present scheme is more accurate than the conventional MUSCL scheme, and is comparable to the original VSIAM3. Moreover, without the directional sweep, the present scheme guarantees the dimensional symmetry, which provides a great convenience in optimizing the partitioning for parallel and GPU processing.

It should be noted that the UTI-VSIAM3 also suffers from numerical diffusion as other Eulerian type schemes. Thus, extra numerical manipulation will be needed to reduce the smearing of the transition layer of the moving interface. In the present model, we make use of an efficient anti-diffusion method which is to be presented in the next section.11)

### IV. Correction of Numerical Diffusion

After the calculation of \( \partial F/\partial t + (u \cdot \nabla)F = 0 \), where \( F \) denotes VOF function, we make modifications to reduce the numerical diffusion at interface between liquid \( (F = 1) \) and gas \( (F = 0) \).

Our approach consists of two steps. At first step, we construct a level set function \( \phi \), that is a signed distance function from the interface based on the VOF function \( F \). As the second step, the cells where \( |\phi| > \epsilon \) are corrected to \( F = 0 \) or \( F = 1 \) by an effective anti-diffusion formulation of flux form so that the conservation of \( F \) is exactly guaranteed.

In level set function calculation, initial value of \( \phi \) must be set from \( F \). We use the following expression,

\[
\phi_0 = 2\alpha (F - 0.5) .
\]  

(15)

The level set function \( \phi \) is then computed iteratively by

\[
\phi_{n+1}^\ast(x,y) = \phi_n(x - C\delta_x, y - C\delta_y)
\]

(16)

\[
\phi_{n+1}(x,y) = \phi_{n+1}^\ast(x,y) + C\text{sgn}(\phi) \delta
\]

(17)

\[
\delta \equiv \frac{\text{sgn}(\phi)}{|\nabla \phi|} \nabla \phi
\]

(18)

where \( C \equiv \text{CFL} \times \text{mesh spacing} \). If \( \phi_{n+1}^\ast(x,y) \cdot \phi_n(x,y) < 0 \), CFL must be reduced locally until \( \phi_{n+1}^\ast(x,y) \cdot \phi_n(x,y) \geq 0 \) is met. \( \phi_{n+1}^\ast(x,y) \) can be interpolated by bi-linear function.
In the anti-diffusion flux calculation, we evaluate the amount of VOF to be redistributed by $\Delta F \equiv F - \overline{F}$ where $\overline{F}=0$ for $\phi<0$ and $\overline{F}=1$ for $\phi>0$. $\Delta F$ is then transported toward the interface along the opposite direction of the signed normal vector given by Eq. (18). In practice, we distribute $\Delta F_{i,j}$ to its neighboring cells $(iup, j)$, $(i, jup)$, $(iup, jup)$ according to the orientation of the interface, i.e. $\Delta F_{i,j}$ is transported to $(iup, j)$, $(i, jup)$ if $|\delta_x|>|\delta_y|$ and to $(i, jup)$, $(iup, jup)$ if $|\delta_x|<|\delta_y|$. The corresponding ratio for distributing $\Delta F_{i,j}$ between $(iup, j)$ and $(iup, jup)$ is calculated by $1 - |\delta_y|/|\delta_x|$ in the former case. In the latter case, the ratio between $(i, jup)$ and $(iup, jup)$ is calculated by $1 - |\delta_x|/|\delta_y|$.

The result of 2D advection test is shown in Fig. 4. The sharpness of interface between $F = 1$ and $F = 0$ are well maintained while the transported profile of shape “T” is faithfully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully reproduced. As shown in the latter numerical tests, the present scheme works well as an interface capturing scheme fully repro
simple vector calculations. The solution of Poisson equation of pressure, on the other hand, requires extra manipulations. Among the popular solvers for Poisson equation, preconditioned ICCG or Bi-CGSTAB are widely used due to their robustness and efficiency. However, recursive dependencies are usually involved in the preconditioner calculation. In the present study, we used PCG solver with the diagonal scaling preconditioner, and found that owing to the reduction of total floating point operations per cycle its computational time is comparable to that of the ICCG. The number of cycles in our test with ICCG was roughly 1/4 of PCG while ICCG calculation time per cycle using single CPU core was two times of PCG because of the heavy precondition calculation. Since the PCG is more parallelism oriented, we have not explored further the implementation of the ICCG on the GPU.

We implemented present code with OpenCL in order to retain compatibility among various hardwares. It is known that a code using OpenCL appears to be more complicated compared to those using CUDA Runtime API. We simplified our code by using macro definition or our own basic API package. For example, although each arguments of calling kernel function must be set with "clSetKernelArg" one by one and called with "clEnqueueNDRangeKernel", we wrapped these instructions with our API "CL_one by one and called with "clEnqueueNDRangeKernel", calling kernel function must be set with "clSetKernelArg" API package. For example, although each arguments of fied our code by using macro definition or our own basic comparison to those using CUDA Runtime API. We simpli-
shortens the elapse time for each real-case application which gain of practical significance in terms of cost-performance im-
provement since a 10 time leap of the GPGPU code largely shortens the elapse time for each real-case application which usually involves number of runs in product design and opti-

We computed the dam break test problem and show the re-

<table>
<thead>
<tr>
<th>Devices</th>
<th>Elapse</th>
<th>Mem. BW</th>
<th>Ideal BW</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xeon</td>
<td>1 Core</td>
<td>20.3 s</td>
<td>9.4 GB/s</td>
</tr>
<tr>
<td>W5580</td>
<td>2 Core</td>
<td>12.4 s</td>
<td>15.5 GB/s</td>
</tr>
<tr>
<td>3.2 GHz 3 Core</td>
<td>9.7 s</td>
<td>19.7 GB/s</td>
<td>32.0 GB/s</td>
</tr>
<tr>
<td>4 Core</td>
<td>9.0 s</td>
<td>21.3 GB/s</td>
<td>32.0 GB/s</td>
</tr>
<tr>
<td>Geforce</td>
<td>2 GPU</td>
<td>2.59 s</td>
<td>74.1 GB/s</td>
</tr>
<tr>
<td>GTX 295</td>
<td>3 GPU</td>
<td>1.12 s</td>
<td>170.8 GB/s</td>
</tr>
<tr>
<td>4 GPU</td>
<td>0.95 s</td>
<td>201.7 GB/s</td>
<td>447.6 GB/s</td>
</tr>
</tbody>
</table>

Fig. 6 Result of dam break test (X-Z plane at Y=75 mm)
respondingly.

It is obvious that the transition layer between water and air has been kept with a compact thickness in the simulation, and small droplets and bubbles are well resolved. Moreover, we would like to emphasize that the calculation was quite stable from the beginning to the end.

VI. Application as a Design Tool for Products

In our company, various products like sanitary wares, kitchens and bath rooms among many others, are produced. Because many products involve water flows, simulations of water/gas/body (deformable or un-deformable) multiphase flows are important for design and improvement of our products. A few examples of applications in our product designs are shown in Figs. 7 and 8 where the free surface and moving solid bodies are reasonably reproduced.

Our codes can simulate real-case problems which involve multiphases including arbitrarily deformable or un-deformable bodies. The calculation of moving bodies is also accelerated with GPGPU computing. The moving bodies shown in Fig. 7 are calculated by method of a finite difference calculation procedure in which bodies represented by color functions are accelerated by integral of forces in each cells. In the simulation shown in Fig. 8, the bucket motion is specified by users. The motion of the solid bodies are calculated by coordinate transformation.

As a matter of fact, the elapse times of these simulations are almost less than one or two seconds per time step with only one GPGPU system described above. We expect the code will be used to the various applications in real-case design processes.

VII. Conclusion

Some recent efforts are made toward the establishment of an integrated simulation tool for designing products that involve multiphase flows. We have proposed a more accurate and efficient formulation for the advection computation by using the unsplit solution procedure, so-called UTI-VSIAM3, to improve the original VSIAM3. In order to compute the moving interface in multiphase flows, a numerical correction method named STAA has been incorporated with UTI-VSIAM3, which results in a conservative and accurate interface capturing scheme well suited for GPU architecture. Owing to the simplicity of the proposed schemes, we were able to implement the multiphase flow solver on a multiple GPGPU hardware with practical computing performance and scalability. We expect the code will be used to the various applications in real-case design processes.

References


