International Journal of Scientific and Innovative Mathematical Research (IJSIMR)
Volume 3, Issue 1, January 2015, PP 6-17
ISSN 2347-307X (Print) & ISSN 2347-3142 (Online)
www.arcjournals.org

Mesh-free GPU Simulation of Complex Flows in Three Dimentional Distoring Domain

Don Liu*, Yifan Wang, Wennan Ma, Haibo Zhang
Mathematics & Statistics and Mechanical Engineering, Louisiana Tech University
Ruston, Louisiana 71272, USA
*DonLiu@Latech.edu**

Abstract: *This paper used a weighted interpolation method, Smoothed Particle Hydrodynamics (SPH), to simulate three dimensional complex flow systems that are difficult to handle with a conventional mesh-based method. The advantage of this mesh-free method lies in its discrete descriptions of the governing equation in a weighted interpolation formulation. Examples of complex flows with two-phase interactions inside rapid deforming domains are demonstrated in several examples. All results were generated from self-developed codes implemented on GPU architecture in CUDA C++ and rendered with OpenGL.*

Index Terms: *Mesh-free Method, Smoothed Particle Hydrodynamics, GPU Implementation, Computational Fluid Dynamics, Two-phase flow, Von Kármán Vortex Street*

1. INTRODUCTION

Smoothed Particle Hydrodynamics (SPH) was first proposed by Gingold and Monaghan(1977)and Lucy (1977) for astrophysics problems and later wasdeveloped by Monaghan et al.(1988) and Liu et al. (2003) for problems in fluid dynamicsand solid mechanics. Originally,the method wasmainlyfor solving compressible flowsbuthadtensile instabilities and inconsistency(Liu and Liu 2003). Over the last two decades, Monaghan (1992, 1994 and 2005) and Liu et al. (2003)have further developed and improved SPH such thatit has wider applicationsto incompressible flows, fluid-solid interaction, solid mechanics, and explosion simulation etc.

Being a mesh-free weighted interpolation method, SPH is especially effective for complex problems with large domain distortion and complex physics. SPH has recently been extensively used to solve hydrodynamics and solid mechanics problems. Different from conventional mesh-based methods such as finite difference, finite volume, finite element, and spectral element methods, SPH does not rely on any fixed computational grid. This gives rise to the capability of SPH in easy handling fluid mechanics problems involving free surface, wave breaking, and rapid geometry distortion. For solid mechanics problems, it is straightforward and intuitive for SPH to mimic atoms as particles. SPH is capable of handling large distortion in solid mechanics, SPH has been adopted to mimic a variety of problems, such as near shore wave-structure interaction (Lind et al. 2012; Antocj et al. 2007; Monaghan and Kocharyan 1995; Amada et al. 2003; Dalrymple and Rogers 2006, 2009), dam break, fragmentation or crack growth in mechanical parts (Libersky and Petschek 1990; Rabczuk et al. 2003; Shao and Lo 2003), material melting, and materials impact phenomena (Randles and Libersky 1996, 1997; Deb and Pramanik 2013).

There are two types of SPH methods for fluids:the weakly compressible SPH (WCSPH)(Monaghan 1994), andthe incompressible SPH (ISPH)(Lee et al. 2008; Liu and Liu2010).Compared with weighted residual methods, SPH dramatically simplified the procedures of numerically solving Navier-Stokers equations. First, SPH simplifies the nature of the Navier-Stokes equations as SPH is a Lagrangian method and non-linear convective terms in Navier-Stokes equations disappear. Second, the momentum and energy equations could be solved explicitly in time, and there is no need to invert large linear systems any more. However, WCSPH and ISPH treat the density and

pressure differently. Since the fluid is treated as a group of particles, WCSPH computes the fluid density with a weighted summation over all neighboring particles using certain kernel functions within the influence radius. Hence, the mass conservation is automatically satisfied as the total number of particles is maintained. This saves the effort for solving the equation of mass conservation. Moreover, the pressure is directly computed from the equation of state for both gas and liquid, as the pressure depends on density only (Liu and Liu 2003). In contrast, ISPH solves the mass conservation equation to obtain the density and solves a pressure Poisson equation to acquire the pressure.

Before computing the total force on each particle and bulk physics properties, it is necessary for each particle to obtain the information about neighboring particles. For a simulation using 10,000 or more particles, the process of searching neighbor particles becomes time consuming. However, since the searching neighbor process for each particle is an independent task and highly parallelizable, GPU could be used to speed up this procedure. Once a SPH algorithm was implemented entirely on GPU, compared with using conventional CPU-based algorithms for the same computation, the GPU algorithm noticeably reduced the run time, as demonstrated in (Takahiro and Koshizuka 2007; Dalrymple et al. 2011; Amada et al. 2003; and Hérault et al. 2010)

This paper first reviews the commonly accepted formulas of SPH for simulating fluids and elastic solids, then discusses GPU-aided implementation. Simulation results from GPU-enabled self-developed SPH code are presented the pros and cons of SPH are discussed.

2. MATHEMATICAL FORMULATIONS OF SMOOTHED PARTICLE HYDRODYNAMICS

As a weighted interpolation method. SPH represents all bulk properties of the fluid and solid body at a certain location in space with a discrete interpolation over a set of surrounding particles (Monaghan 1988; Liu and Liu 2003). The interpolating function used in SPH is called kernel function and denoted as $W(\boldsymbol{r}-\boldsymbol{r}',h)$, where h is the radius of influence region around the position $\boldsymbol{r}'$.The kernel function is similar to a delta function with the following properties:

$$\int W(\boldsymbol{r}-\boldsymbol{r}',h)d\boldsymbol{r}' = 1, \tag{1}$$

$$\lim_{h\to 0} W(\boldsymbol{r}-\boldsymbol{r}',h) = \delta(\boldsymbol{r}-\boldsymbol{r}'). \tag{2}$$

Using a kernel function, a property such as I at the location $\boldsymbol{r}$ can be expressed as(Monaghan 1988):

$$I(\boldsymbol{r}) = \int I(\boldsymbol{r}')W(\boldsymbol{r}-\boldsymbol{r}',h)d\boldsymbol{r}' = \int \frac{I(\boldsymbol{r}')}{\rho(\boldsymbol{r}')}W(\boldsymbol{r}-\boldsymbol{r}',h)\rho(\boldsymbol{r}')d\boldsymbol{r}'$$
$$\approx \sum_{r'} m(\boldsymbol{r}')\frac{I(\boldsymbol{r}')}{\rho(\boldsymbol{r}')}W(\boldsymbol{r}-\boldsymbol{r}',h), \tag{3}$$

In which, the summation is over all neighbor particles within the influence region. To compute the gradient of $I(\boldsymbol{r})$ at the location $\boldsymbol{r}$, below approximation are used after an integration by parts:

$$\nabla I \approx \sum_{r'} m(\boldsymbol{r}')\frac{I(\boldsymbol{r}')}{\rho(\boldsymbol{r}')}\nabla W(\boldsymbol{r}-\boldsymbol{r}',h). \tag{4}$$

The surface integral term is dropped, since the kernel function and I both go to zero by definition. Similarly, higher order spatial derivatives could always be presented as a weighted summation of derivatives of the kernel functions.

2.1. Fluid Particles

The momentum equation of fluid in Lagrangian form is given below:

$$\frac{d\boldsymbol{v}}{dt} = -\frac{1}{\rho}\nabla P + \mu\Delta\boldsymbol{v} + \boldsymbol{f}_{ex}, \tag{5}$$

where P is pressure, ρ is the density and $\boldsymbol{f}_{ex}$ stands for external force.To conserve linear and angular momentum and remain numerical stability, the pressure term is changed (Gingold and Monaghan 1977):

$$-\frac{1}{\rho}\nabla P + \mu\Delta\boldsymbol{v} = -\nabla\left(\frac{P}{\rho}\right) - \frac{P}{\rho^2}\nabla\rho + \mu\Delta\boldsymbol{v}. \tag{6}$$

The discrete form of the Lagrangian description of the Navier-Stokes equation becomes:

$$\frac{d\boldsymbol{v}_i}{dt} = -\sum_j m_j\left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} + \Pi_{ij}\right)\nabla_i W_{ij} + \boldsymbol{f}_{ex}, \tag{7}$$

where the subscripts i and $j$areindices of current particle and neighbor particles respectively. For WCSPH, density ρ_i is given by(Monaghan 1992):

$$\rho_i = \sum_j m_j W_{ij}. \tag{8}$$

And P_i is computed from equation of state(Monaghan 1992), which depends on ρ_i only:

$$P_i = \frac{\rho_0 c_0^2}{7}\left[\left(\frac{\rho_i}{\rho_0}\right)^7 - 1\right]. \tag{9}$$

$\Pi_{ij}$is viscous pressure which has following expression(Gray and Monaghan 2001):

$$\Pi_{ij} = \begin{cases} \dfrac{-\alpha c\tilde{\mu}_{ij} + \beta\tilde{\mu}_{ij}^2}{(\rho_i + \rho_j)/2}, & if\ \boldsymbol{v}_{ij}\cdot\boldsymbol{r}_{ij} < 0, \\ 0, & otherwise. \end{cases} \tag{10}$$

In which, $\tilde{\mu}_{ij}$ is equal to $\frac{h\boldsymbol{v}_{ij}\cdot\boldsymbol{r}_{ij}}{r_{ij}^2+0.001h^2}$ andh is the radius of influence region. Specific valuesfor$\alpha\ and\ \beta$depend on the type of problems. For low Mach number flow, $\alpha = 0\ and\ 0 < \beta < 1$;for high Mach numberflow,$\alpha = 1\ and\ \beta > 1$.

2.2. Solid Particles

For an elastic body with stress tensorσ, the momentum equation in the component form (Takeda et al 1994) is given as:

$$\frac{dv^a}{dt} = \frac{1}{\rho}\frac{\partial\sigma^{ab}}{\partial x^b} + f^a, \tag{11}$$

where, f^a is the external force. To avoid the conflict in symbols, *a* and *b* are used to represent Cartesian components. Tensor stress σ^{ab} is consisted of two components, volumetric stress $P\delta^{ab}$and deviatoric stressS^{ab}. It is straightforward to calculate the volumetric stress. For the deviator stress, the change of $S^{ab}$is added to the previous deviatoric stress. The rate of change of S^{ab} is given by:

$$\frac{dS^{ab}}{dt} = 2\mu\left(\dot{\epsilon}^{ab} - \frac{1}{3}\delta^{ab}\dot{\epsilon}^{ab}\right) + S^{ac}\Omega^{bc} + \Omega^{ac}S^{cb}, \tag{12}$$

Where $\dot{\epsilon}^{ab}$and Ω^{ab} are defined as below:

$$\dot{\epsilon}^{ab} = \frac{1}{2}\left(\frac{\partial v^a}{\partial x^b} + \frac{\partial v^b}{\partial x^a}\right), \tag{13}$$

$$\Omega^{ab} = \frac{1}{2}\left(\frac{\partial v^a}{\partial x^b} - \frac{\partial v^b}{\partial x^a}\right). \tag{14}$$

Derivatives of the velocity is computed by the relative SPH scheme (Monaghan and Kocharyan 1995):

$$(\frac{\partial v^a}{\partial x^b})_i = -\sum_j \frac{m_j}{\frac{\rho_i+\rho_j}{2}}(v_i^a - v_j^a)\frac{\partial W_{ij}}{\partial x_i^b}. \tag{15}$$

For the momentum equation of elastic body, the SPH scheme is:

$$\frac{dv_i^a}{dt} = \sum_j m_j\left(\frac{\sigma_i^{ab}}{\rho_i^2} + \frac{\sigma_j^{ab}}{\rho_j^2}\right)\frac{\partial W_{ij}}{\partial x_i^b} + f^a. \tag{16}$$

2.3. Particle Interactions

The interaction force exerted between particle"i"and particle"j" is described by the Lenard-Jones potential(Liu and Liu 2003; Monaghan and Kocharyan 1995):

$$\boldsymbol{f}_{ij} = C_0\left[\left(\frac{r_0}{|\boldsymbol{r}_{ij}|}\right)^{p_1} - \left(\frac{r_0}{|\boldsymbol{r}_{ij}|}\right)^{p_2}\right]\frac{\boldsymbol{r}_{ij}}{|\boldsymbol{r}_{ij}|}, \tag{17}$$

where$\boldsymbol{r}_{ij} = \boldsymbol{r}_i - \boldsymbol{r}_j$, r_0 is the initial spacing between two particles,p_1 and $p_2$are chosen to be 12 and 6 respectively.$C_0$is an adjustable constant depends on particular problem.

2.4. Velocity Evaluation

Particle motion is computed using the XSPH scheme:

$$\frac{d\boldsymbol{r}_i}{dt} = \boldsymbol{v}_i + \varepsilon\sum_j m_j\left(\frac{\boldsymbol{v}_j - \boldsymbol{v}_i}{\bar{\rho}_{ij}}\right)W_{ij}, \tag{18}$$

with$\bar{\rho}_{ij} = \frac{\rho_i+\rho_j}{2}and0 \le \varepsilon \le 1$as a factor, which averages the velocity in the influence.

2.5. Time Evolution and Code Speedup

For simplicity, this paper uses the forward Euler method, although other methods could be implemented. As an explicit method in time, the time step should be restricted by the CFL condition:

$$\Delta t < \frac{h}{c}, \tag{19}$$

Where *h* is the characteristic length and *c* is the speed of propagation.

The procedures of the implementation for time evolution are shown in Fig. 1. The values of velocity and positions are initialized at the beginning, and the densities of fluid and solid particles and pressure are computed with the position vector. With the previous velocity of each particle is known, the deviatoric stress and viscous stress can be calculated, then the forces on particles and the acceleration of each particle. After that, the velocity and position of each particle at the next time level could be determined.

There are several ways to speed up the SPH code. For instance, one approach is to improve serial searching algorithms, and another one is by parallel implementation. The former approach mainly is to improve the efficiency of searching for neighbor particle. An example is the tree search, which reduces the operation counts down to *O(*N^2*)* to *O(NlogN)*. Another example is by introducing a link list, and only the neighbor particles within the list of neighbors of immediate neighbors are searched. SPH is ideal for parallelization and particles are task-independent.

In this paper, NVIDIA CUDA C++ was used to write GPU codes allowing individual GPU thread to search for neighbor particles and compute forces. The flow chat of GPU version SPH is illustrated in Fig. 1. The computational time is significantly reduced as shown later in the paper. However, since CUDA is still in its early stage of development, it does not support composite or nested data structure, such as class or structure. Therefore, in order to utilize GPU to perform computations that uses complex data structure, it is necessary to convert between simple and raw data structure and composite memory-access efficient data structure that was designed in the self-developed codes in this paper.

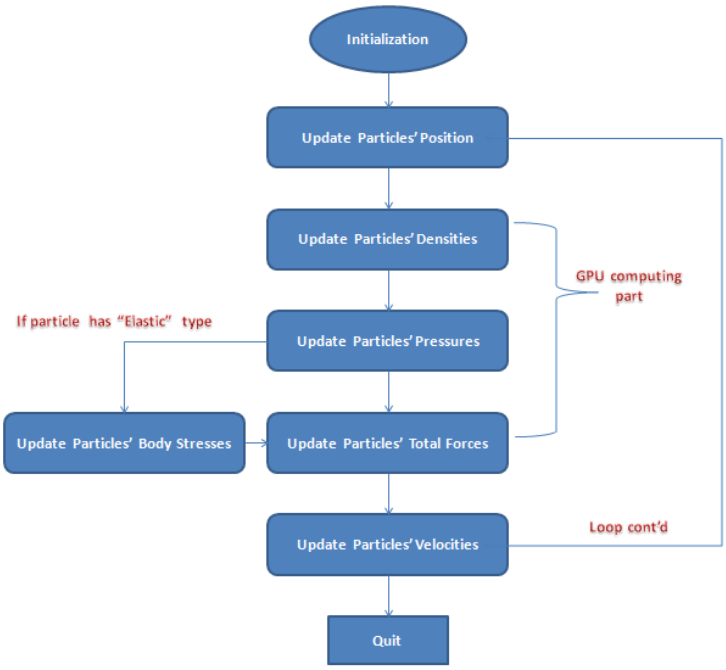


Fig.1. *Schematic of the implementation procedures*

3. Simulation Results

Three test cases in two- or three dimensions are presented in this paper to illustrate good visual effects for results from GPU computing. Three different types of particles, fluid, elastic solid, and boundary particles, are considered in this work. Each type has its own properties. The numerical simulations were performed on one Intel i5 processor with 16GB memory, NVIDIA GTX 760 graphic card with 2GB GDDR5 memory. Results were visually rendered immediately via the open graphic library (OpenGL).

3.1. Unsteady Nozzle Flow

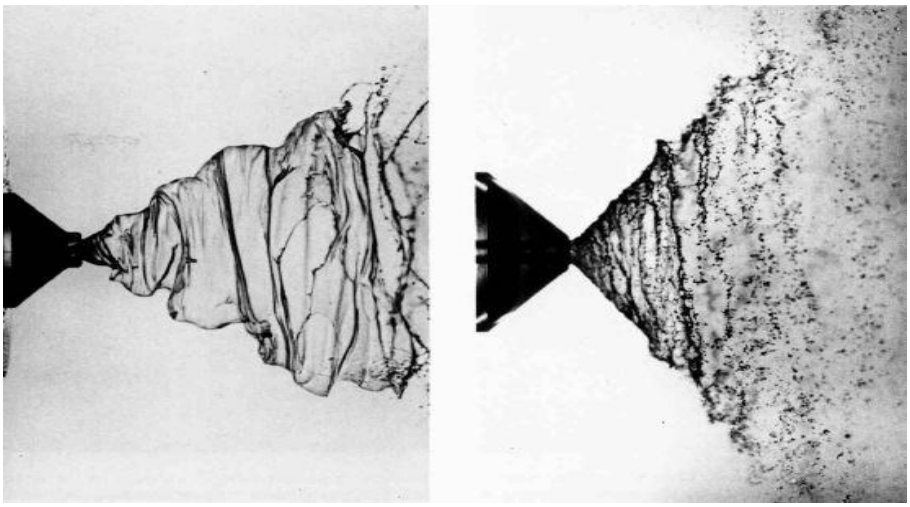
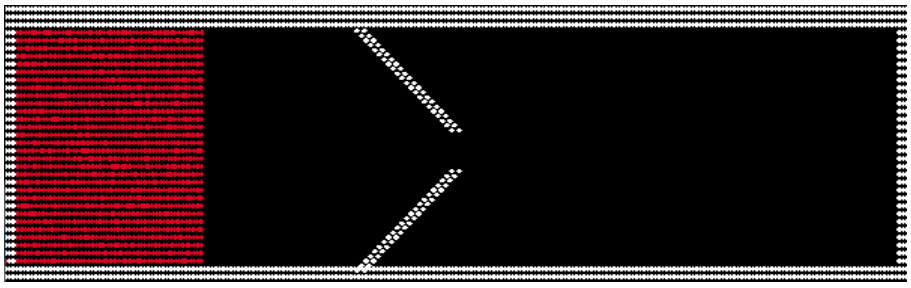


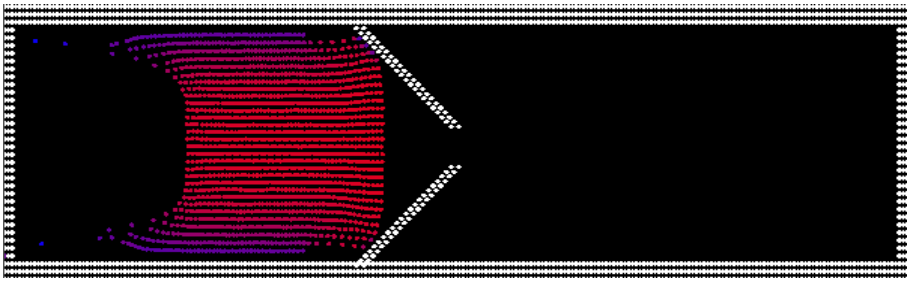
Fig. 2. *Flow ejected from a nozzle in a thin conical jet, Photographed by N. Dombrowski (Dyke 1988)*

A nozzle is commonly used in fluid dynamics and fluid mechanics to change the direction or rate of flow of the fluid. The "de Laval" nozzle is one of the most important one, since it is widely used in jet or rocket engines to obtain maximized kinetic energy in certain direction.

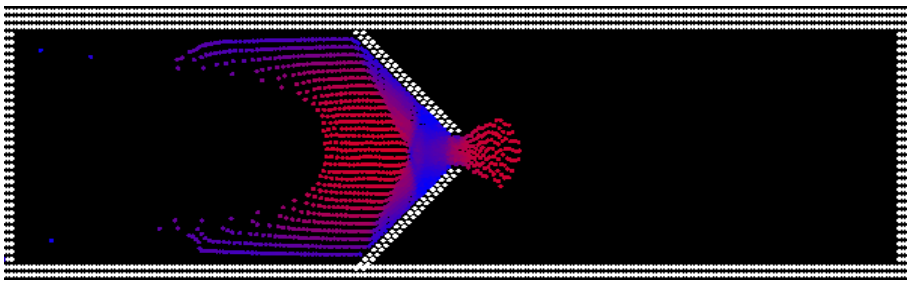
To illustrate the characteristics of a transient jet flow flow ejected from a nozzle, WCSPH formulas (previously discussed in the Section **2**) was used and implemented on GPU to simulate a two dimensional unsteady nozzle flow. Fig. 2 shows the photographs of flow ejected from a nozzle at different flow rate in a thin conical jet by Dombrowski (Dyke, 1988). Fig. 2 was used for comparison with the simulated nozzle flow as shown in Fig. 3. Afluid flows over the nozzle with an initial dimensionless velocity 20/second was set up. The gravity is ignored here. In Fig. 2, when the speed of outgoing flow increases beyond a critical point, the flow breaks into small droplets. Fig. 3 sequentially shows the time evolution flow passing through this nozzle. The red color represents the high velocity particles, while, the blue color represents particles of lower velocity. Although the total number of particles is only 3,000 as in this example, the effect of boundary layer on the top and bottom walls of the nozzle was created. Fig. 3 (b) and (c) show the Hagen–Poiseuille flow. Fig. 3 (c) to (e)show the process of droplet formation due to the effect of surface tension. The entire simulation took only 40 seconds on a single GPU card which was described earlier.



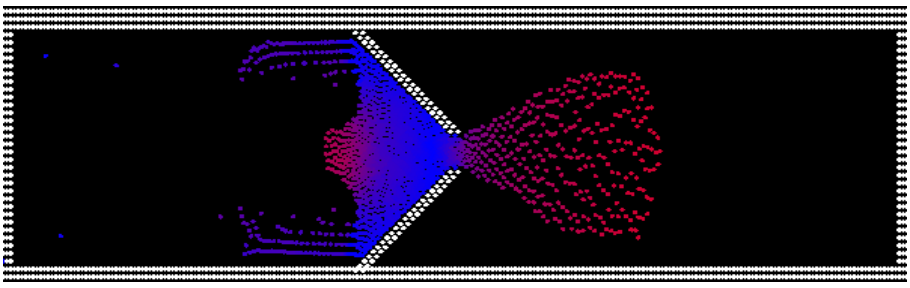
(a)



(b)



(c)



(d)

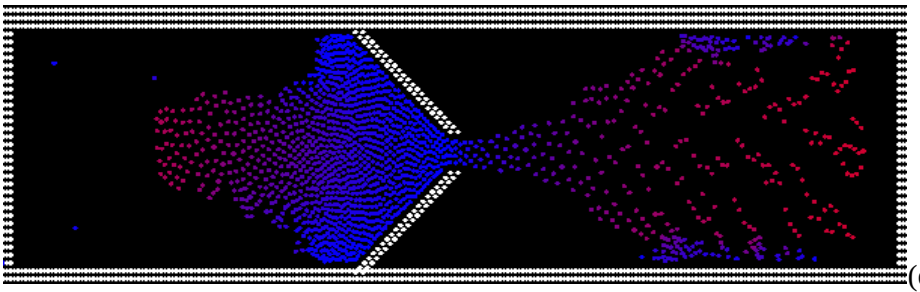

(e)

Fig.3. *Five snapshots of 2D unsteady nozzle flow*

3.2. Vortex Shedding

Vortex shedding formed by viscous fluid separated from a cylinder at Reynolds number between 40 and 150 is of engineering importance (Tritton 1959). For example, in the designing of bridges and offshore oil rigs, ignoring the effect of vortex shedding could result in serious safety issue and potential loss of equipment and lives (Holmes et al. 2011). Simulation of vortex shedding with SPH is not commonly seen in open literature (Tritton 1959; Roulund et al. 2005; Price 2012; Marrone et al. 2013).

Fig. 4 is a photo (Dyke 1988) for the experiment of the Von Kármán laminar vortex street (Kármán 1963). As seen in Fig. 4, the vortices created behind cylinder are separated boundary layer near the cylindrical surface which periodically detached from both sides of the cylinder, due to the adverse pressure zone created by skin friction on the cylinder. For comparison, a two dimensional GPU simulation of a similar flow is illustrated in Fig. 5. The flow has an initial dimensionless speed of 15/second with a peak Reynolds number of 140. A narrow domain was used here in order to use fewer particles and to speed up GPU computing. The inflow and outflow are treated as periodic boundaries. This simulation used a total of 16,000 particles and took 10 minutes 22 seconds to spin up the entire flow and to reach the stage of steady vortex street shedding. This is much faster than using a mesh-based conventional method.

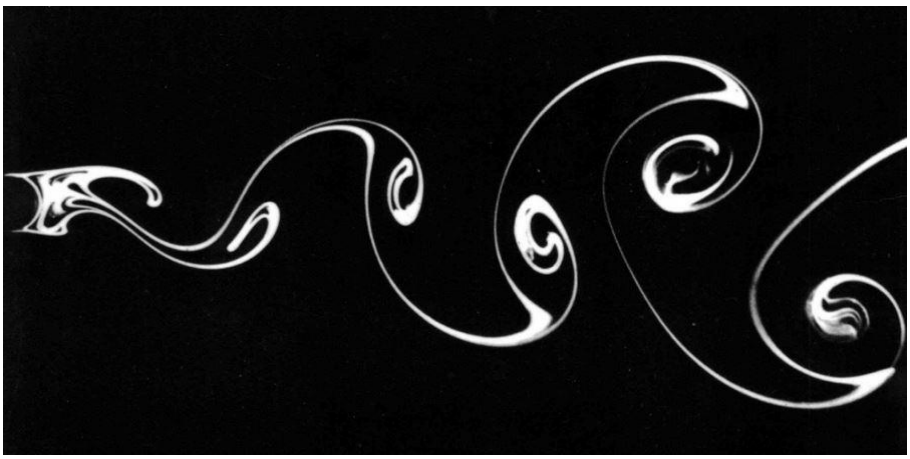


Fig.4. *Von Kármánvortex street behind a circular cylinder at Reynolds number 140 Photographed by Sadatoshi Taneda (Dyke 1988)*

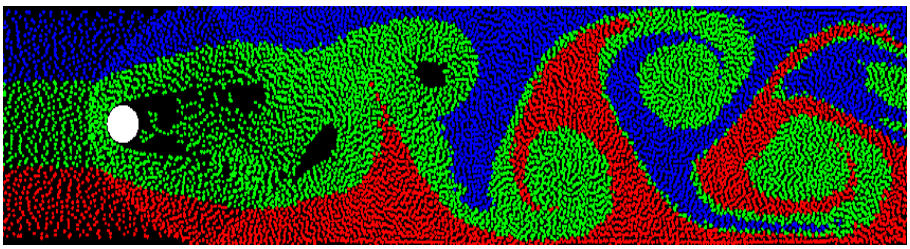


Fig.5. *Simulated laminar two dimensional vortex street at Reynolds number 140*

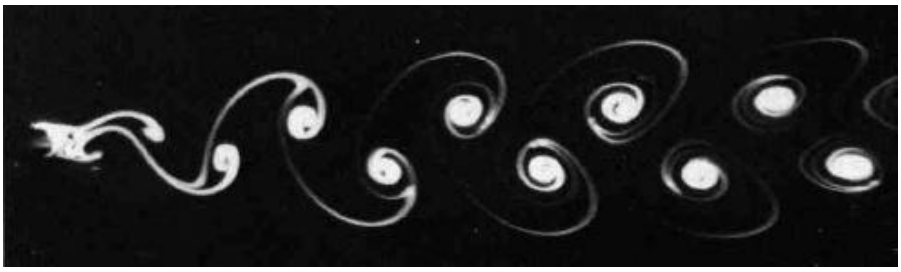


Fig.6. *Von Kármán vortex street behind a circular cylinder at Reynolds number 105Photographedby Sadatoshi Taneda (Dyke 1988)*

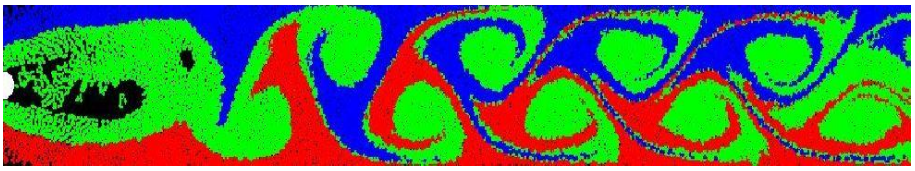


Fig. 7. *Simulated laminar two dimensional vortex street at Reynolds number 105*

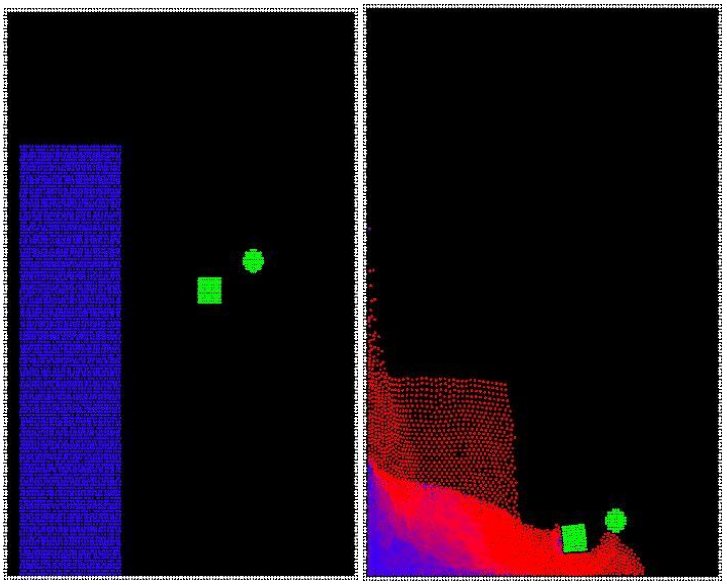
Fig. 6 is a photo (Dyke 1988) for the experiment of flow past a cylinder of unit radius at Reynolds number 105. Using results from our self-developed GPU codes, a matching run with the same setting as in Fig. 6 was performed and shown in Fig. 7. The domain is narrower than Fig. 6 so that this GPU computing could render more resolution with 34,000 particles. The total GPU computing time is 35 minutes. From the comparison between Fig. 6 and Fig. 7, it could be claimed that the GPU simulation in this paper captured the key characteristics of this vortex shedding phenomenon at the Reynolds number 105 with a relatively economical computational cost and time.

3.3. Elastic Solid and Fluid Interaction

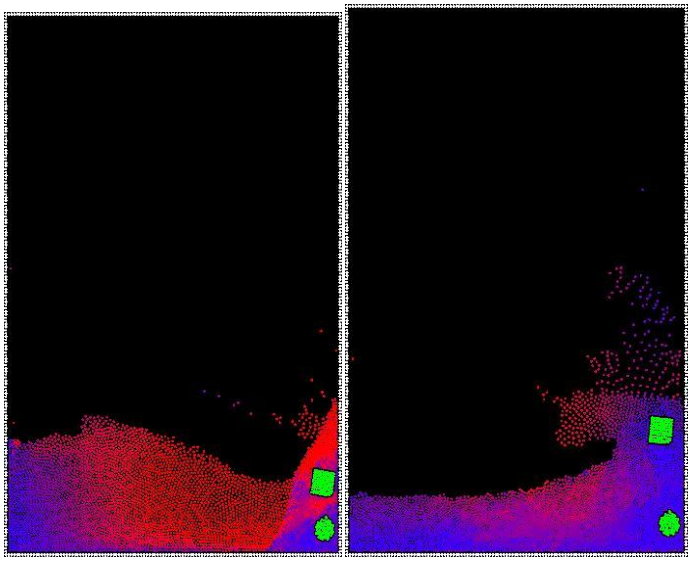
Many engineering problems involve multi-phase media and multi-phase flow, such as the dynamics of the weather development, the ocean circulations, and the near-shore sediment transportation. Under certain situations, SPH could be efficient in simulating some complex two-phase flow problems (Antocj 2007). In the following, two-phase interactions of elastic objects and the fluid are presented.

A two-dimensional simulation of the interaction of the fluid with a solid elastic cube and sphere at different time is shown in Fig. 8.The size of the simulation domain is 40 by 65in dimensionless units, same as below, surrounded by boundaries which consist of solid particles. The side of the cube is 3 and the diameter of the sphere is 3 as well. The density of the fluid is scaled to one and the density of both the cube and sphere is 0.6. Both solids are at free falling under gravity before colliding with collapsed fluid beam on the left. Initially, the fluid was fixed still on the left side of the domain by a confinement immediately on its right. Once the simulation started, the confinement was removed and the fluid was released from the left under gravity and then interacted with the elastic solids. Both objects were washed away by the inertial of the fluid with some spinning. The color of the fluid particles indicates their magnitude of velocity. Zero velocity is in blue and higher velocity is in red with purple and pink denotes the intermediate values. A total of 5,500 particles are used in this simulation and 3 minutes was used to complete this GPU computing, as shown in Table 1, compared with 14 minutes for a serial run on a singlei5 CPU. Therefore, the GPU run is about 4~5 times faster than the CPU run.

Similarly a three-dimensional simulation of one solid elastic sphere of diameter 5 interacting with the fluid is shown in Fig. 9. The simulation domain, a little large than the two-dimensional run, is 40 by 75 by 16. Exactly the same density settings as in the two-dimensional case are used here. Four snapshots indicate the free falling and interaction processes at different moments. The velocity magnitude of the fluid is indicated by the same color map as in the previous run. For a better visual effect, only the fluid and sphere are displayed and the boundary walls are skipped. A total of 28,000 particles were used to render the three-dimensional effect. Compared with over two hours of CPU time with a serial algorithm without GPU computing, as shown in Table 1, the total run time for the GPU simulation is only less than 29 minutes. Therefore, the GPU run is at least four times faster than the CPU one.

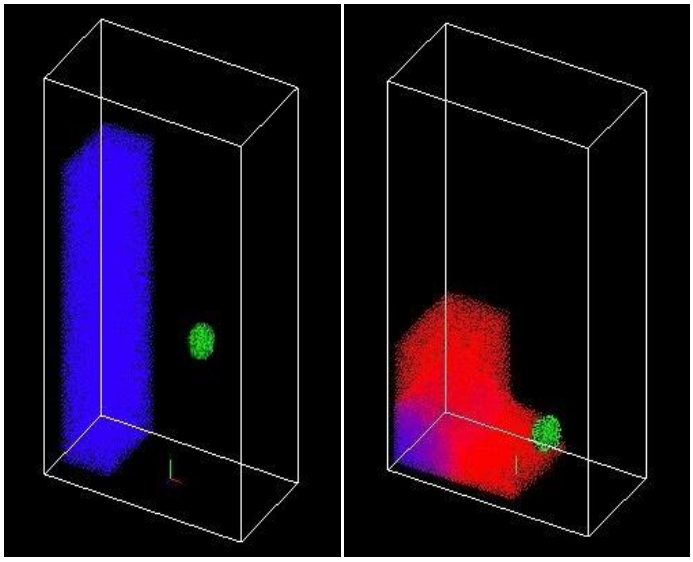


(a) (b)

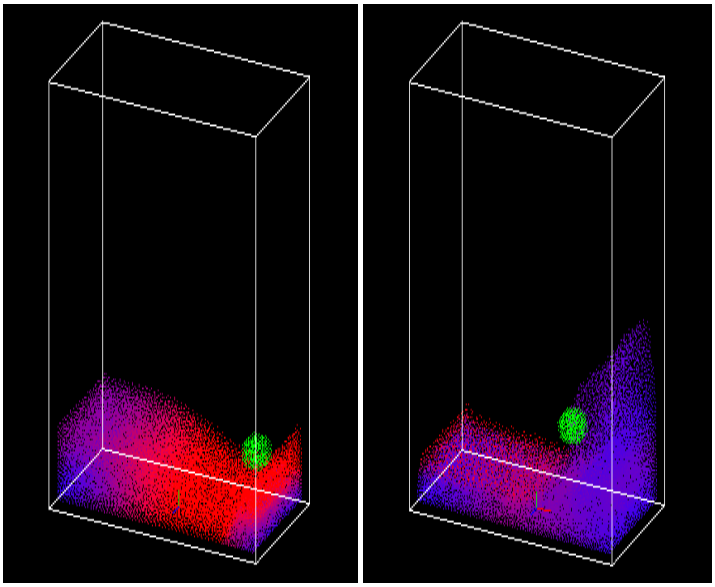


(c) (d)

Fig.8. *Snapshots of two-dimensional solid-fluid interaction*



(a) (b)



(c) (d)

Fig.9. *Snapshots of three-dimensional solid-fluid interaction*

3.4. Simulation Execution Time

In this section, the execution time for GPU simulation versus CPU simulation is listed in Table 1. For all these five runs with exactly the same number of particles and unknowns, the GPU computing is 4 to 6 times faster than the corresponding CPU computation which does have a parallel computing involved.

Table 1. *Execution time (seconds) and specifics of GPU and CPU runs*

Simulations	Number of Particles	GPU Version	Serial Version	Time Ratio
Unsteady nozzle flow	3,000	40	255	6.38
Vortex shedding Ex. (1)	16,000	622	2,605	4.18
Vortex shedding Ex. (2)	34,000	2,100	8,176	3.89
2D impinging flow	5,500	180	840	4.67
3D impinging flow	28,000	1,731	6,840	3.95

4. CONCLUSIONS

Simulation results in this paper have demonstrated that the mesh-free SPH is advantageous to handle problems involving free-surface flows, wave breaking, two-phase flows, polymorphic domain with large deformation. However, SPH is not an intrinsically accurate method due to its formulation and nature. To improve the resolution and accuracy, large numbers of particles are required for the simulation and parallel implementation is essential. A GPU-based multi-thread parallel approach could achieve a noticeable speedup comparing with a CPU serial algorithm. However, several limits restrict the GPU computing. First, the memory on a GPU card is limited; therefore, there is a limit of the max number of particles allowed for a simulation. Second, the total number of CUDA cores is finite, for example, our GPU card has 1,024 cores. Third, GPU computing requires copying data back and forth between the CPU memory and GPU memory. This operation depends on the latency of the system, e.g., the bus bandwidths of both GPU and motherboard, and cache size of the system. Therefore, for large simulations such as one involving 100,000 particles, distributing the total computational load to multiple GPUs with some CUDA inherent functions such as cuda Get Device, cuda Set Device and cuda Memcpy Peer Async (a peer-to-peer memory copy function) etc. is necessary to further accelerate the speed of computation. This multiple GPU cards parallel computing is similar in idea to Open MP and MPI in CPU-based computing.

SPH using GPU computing technique is an emerging method with many engineering application such as simulating hurricane surges in coastal regions, near-shore wave breaking (Dalrymple and Rogers 2006),micro-nano-fluids, and maritime optimization(Randles and Libersky 1996).

ACKNOWLEDGMENT

This study was funded by National Science Foundation under grants DMS-1115546 and DMS-1318988. The authors are grateful to Drs. Le Yan, Alexander Pacheco, Supada Laosooksathit, and Box Leangsuksun for their constructive comments and generous help. The authors would like to thankLouisiana Optical Network Initiative, the Center for Computation and Technology, Louisiana State University, and XSEDE for providing technical support and computational facilities.

REFERENCES

Amada, T., Imura, M., Yasumuro, Y., Manabe, Y. and Chihara, K. (2003). "Particle- based fluid simulation on the GPU." *Proc. ACM Workshop on General-purpose Computing on Graphics Processors*.

Antocj, C., Gallati, M. and Sibilla, S. (2007). "Numerical simulation of fluid-structure interaction by SPH." *Computers and Structures*, **85**, 879-890.

Dalrymple, R. A. and Rogers, B. D. (2006). "Numerical modeling of water waves with SPH method." *Coastal Engng*.,**53**, 141–147.

Dalrymple, R. A., Gómez-Gesteira, M., Rogers, B. D., Panizzo, A., Zou, S., Crespo, A. J. C., Cuomo, G. and Narayanaswamy, M. (2009). "Smoothed Particle Hydrodynamics for Water Waves, Advances in Numerical Simulation of Nonlinear Water Waves." World Scientific Publishing.

Dalrymple, R. A., Hérault, A., Bilotta, G., Farahani, R. J. (2011). "GPU-accelerated SPH model for water waves and free surface flows." *Proc. 32nd Intl. Conf. Coastal Engineering*, North America, 1.

Deb, D. and Pramanik, R. (2013). "Failure process of brittle rock using smoothed particle hydrodynamics." *Journal of Engineering Mechanics*, (ASCE) EM.1943-7889.0000592.

Dyke, M. V. (1988). "An Album of Fluid Motion." The Parabolic Press.

Gingold,R. A. and Monaghan, J. J. (1977). "Smoothed particle hydrodynamics: Theory and application to non-spherical stars." *Monthly Notices of the Royal Astronomical Society*, **181**, 375-389.

Gray, J. P. and Monaghan, J. J. (2001). "SPH elastic dynamics." *Comput. Methods Appl. Mech. Engrg.*, **190**, 6641-6662.

Harada, T. and Koshizuka, S. (2007). "Smoothed particle hydrodynamics on GPUs." *Computer Graphics International*, 63-70.

Hérault, A., Bilotta, G., Dalrymple, R. A. (2010). "SPH on GPU with CUDA." *Journal of Hydraulic Research*, **48**, 74-79.

Holmes, D. W., Williams, J. R. and Tilke, P. (2011). "Smooth particle hydrodynamics simulations of low Reynolds number flows through porous media." *Int. J. Numer. Anal. Methods Geomech.*,**35**, 419-437.

Kármán, T. V. (1963). "Aerodynamics." McGraw-Hill.

Lee, E. S., Moulinec, C., Xu, R., Violeau, D., Laurence, D. and Stansby, P. (2008). "Comparisons of weakly compressible and truly incompressible algorithms for the SPH mesh free particle method." *J. Comput. Phys*., **227**, 8417-8436.

Liu, M. B. and Liu, G. R. (2010). "Smoothed Particle Hydrodynamics (SPH): an Overview and Recent Developments." *Archives of Computational Methods in Engineering*, **17**, 25-76.

Liu, G. R. and Liu, M. B. (2003). "Smoothed Particle Hydrodynamics: A Mesh-free Particle Method." Singapore: World Scientific.

Lind, S. J., Xu, R., Stansby, P. K. and Rogers, B. D. (2012). "Incompressible smoothed particle hydrodynamics for free-surface flows: A generalised diffusion-based algorithm for stability and validations for impulsive flows and propagating waves." *Journal of Computational Physics*, **231**, 1499-1523.

Libersky, L. D. and Petschek, A. G. (1990). "Smooth particle hydrodynamics with strength of materials, advances in the free lagrange method." *Lecture Notes in Physics*, **395**, 248–257.

Lucy, L. B. (1977). "A numerical approach to the testing of the fission hypothesis." *Astron. J.*, **82**, 1013–1024.

Marrone, S., Colagrossi, A., Antuono, M., Colicchio, G. and Graziani, G. (2013). "An accurate SPH modeling of viscous flows around bodies at low and moderate Reynolds numbers." *J. Comp. Physics*, **245**, 456-475.

Monaghan, J. J. (1988). "An introduction to SPH." *Comp. Phys. Comm*., **48**, 89-96.

Monaghan, J. J. (1994). "Simulating free surface flows with SPH." *J. Comp. Phys*., **110**, 339-406.

Monaghan, J. J. (1992). "Smoothed particle hydrodynamics." *Annual Review of Astronomy and Astrophysics*, **30**, 543-574.

Monaghan, J. J. (2005). "Smoothed particle hydrodynamics." *Rep. Prog. Phys*., **68**, 1703-1759.

Monaghan,J. J. and Kocharyan, A. (1995). "SPH simulation of multi-phase flow."*Computer Physics Communications*, **87**, 225-235.

Price, D. J. (2012). "Resolving high Reynolds numbers in smoothed particle hydrodynamics simulations of subsonic turbulence." *Monthly Notices of the Royal Astronomical Society*, **420**, 375-408.

Rabczuk, T., Eibl,J. and Stempniewski, L. (2003). "Simulation of high velocity concrete fragmentation using SPH/MLSPH." *Int. J. Numer. Methods Eng.*, **56**, 1421–1444.

Randles, P. W. and Libersky, L. D. (1997). "Recent improvements in SPH modeling of hypervelocity impact." *Int. J. Impact Eng*., **20**, 525–532.

Randles, P. W. and Libersky, L. D. (1996). "Smoothed particle hydrodynamics: some recent improvements and applications." *Comput. Methods Appl. Mech. Eng.*, **139**, 375-408.

Roulund, A., Sumer, M. B. and Fredse, J. (2005). "Numerical and experimental investigation of flow and scour around a circle pile." *J. Fluid. Mech*. **534**, 351-401.

Shao, S. and Lo, E. Y. M. (2003). "Incompressible SPH method for simulating Newtonian and non-Newtonian flows with a free surface." *Adv. Water Resour*.,**26**, 787-800.

Takeda, H., Miyama, S. M., Sekiya, M. (1994). "Numerical simulation of viscous flow by smoothed particle hydrodynamics."*Progress of Theoretical Physics*, **92**, 939-960.

Tritton, D. J. (1959)."Experiments on the flow past a circular cylinder at low Reynolds numbers."*J. Fluid Mech.*, **6**, 547-567.

AUTHORS' PROFILE

Dr. Don Liu received his Ph.D degree from Brown University, Providence, Rhode Island, USA in 2004. Dr. Liu is currently the Contractor's Trust Endowed Associate Professor in Mathematics and Mechanical Engineering at Louisiana Tech University. Dr. Liu serves as the members of Editorial Board of two international journals and is an active researcher in computational fluid dynamics, computational mathematics and parallel computing.