Development of GPU-based Smoothed Particle Hydrodynamics Code for Analysis of Multi-physics Phenomena in Nuclear Reactors

Young Beom Jo, So-Hyun Park, Hae Yoon Choi, Eung Soo Kim*
Department of Nuclear Engineering, Seoul National University
E-mail: kes7741@snu.ac.kr

1. Introduction

Complicated multi-physics, multi-phase phenomena typically occur in nuclear thermal-hydraulics such as steam-water two phase flow in PWR severe accidents, molten fuel-fission gas behavior and fuel-cladding interactions in fuel pin, air–trapped sloshing in various types of water tank of power plants, and so on. Various Eulerian numerical method including PIC, VOF, LS were developed to analyze the multi-physics and multi-phase phenomena, however, these traditional grid-based methods have difficulties in detecting interface between phases and analyzing free surface flow and highly deformable flow due to the nature of Eulerian scheme.

Using particle based numerical methods such as Smoothed Particle Hydrodynamics (SPH) and Moving Particle Semi-implicit (MPS) is one possible approach to overcome above-mentioned problems of grid-based methods. This paper introduces the recent progress and on-going activities of SOPHIA code, the 3-D multi-physics SPH code under development in Seoul National University (SNU). The SOPHIA code has been developed and parallelized using CUDA C++ language in order to simulate various complicated thermal-hydraulic phenomena associated with nuclear reactor. Following sections summarize basic principle of SPH method and overall features of SOPHIA code including physical models, key implementation strategies, and also some multi-dimensional test simulations of SOPHIA code.

2. SPH Methods

2.1. Mathematical Concept of SPH

Smoothed Particle Hydrodynamics (SPH) is a fully Lagrangian, particle method for analyzing fluid flows. In the SPH method, the entire fluid system is represented by a finite number of particles which carry individual properties, and the material values including momentum, energy are computed by smoothing over the neighboring particles [1]. Fig. 1 shows particle based fluid system in SPH calculation. The SPH method exhibits large advantages over the traditional grid-based numerical methods in dealing with free surfaces or complex boundary geometries. The boundary at a free surface and moving interface are automatically imposed and simply determined by the trajectory of particle nodes. Thus, there are no constraints on the geometry of the fluid system. Recently, this method has been rapidly developed with the improvement of computational environment and widely utilized in various engineering areas in recent years.

The SPH formulation is obtained by using kernel functions that approximate a delta function and approximating the integral by a summation [1],

$$f_i(r) = \sum_j f_j W(r_i - r_j) V_j \tag{1}$$

where f_i is a function at the position i, subscript j is the nearby particles of center particle i, V is the particle volume, and $W(r_i - r_j)$ standing for the kernel function with h denotes the smoothing length that is the influencing area of kernel weighting function. Fig.2 shows the particle distribution with the kernel function. The kernel weighting function is a function of particle distance and it must be normalized over its support domain.

The kernel approximation about the derivative function can be obtained by applying the Gauss integral formula and divergence theorem to the above field approximation equation (1). Finally, the first derivative of the field function $f(r)$ is expressed as a function of kernel derivative, mass and density [1].

$$\nabla \cdot f_i(r) = \sum_j f_j \frac{m_j}{\rho_j} \nabla W\left(r_i - r_j\right) \tag{2}$$

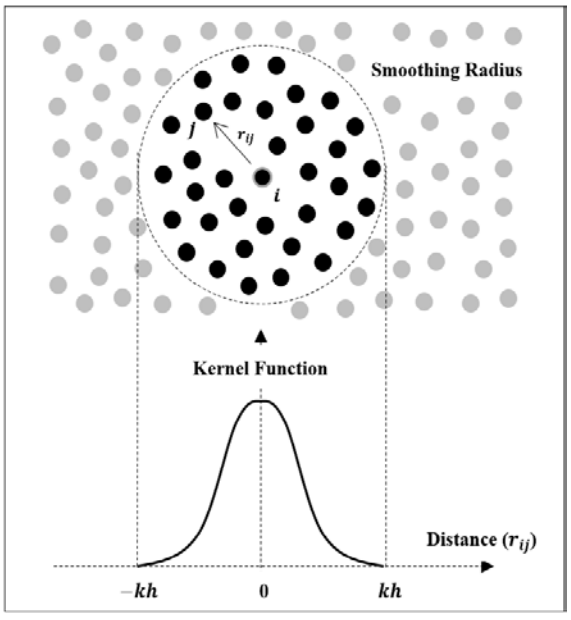

Fig. 1. Particle distribution with kernel function

Table 1
SPH notation for basic governing equations of SOPHIA

Governing Equation	SPH Formulation	
Mass Conservation	$\left(\frac{d\rho}{dt}\right)_i = \rho_i \sum_j \frac{m_j}{\rho_j}\left(\vec{u_i} - \vec{u_j}\right)\cdot\nabla_i W_{ij}$	(6)
Momentum Equation	$\left(\frac{du}{dt}\right)_i = -\sum_j m_j\left(\frac{p_i + p_j}{\rho_i\rho_j} + \prod_{ij}\right)\nabla W_{ij} + \sum_j \frac{4m_j\mu_j \vec{r_{ij}}\cdot\nabla_i W_{ij}}{\left(\rho_i+\rho_j\right)\left(\left\|\vec{r_{ij}}\right\|^2 + \eta^2\right)}\left(\vec{u_i} - \vec{u_j}\right) + \vec{g}$	(7)
Equation of State	$p = \frac{{c_0}^2\rho_0}{\gamma}\left[\left(\frac{\rho}{\rho_0}\right)^\gamma - 1\right]$	(8)

2.2. SPH Governing Equations

The basic governing equations for the motion of an isothermal, incompressible, newtonian fluid in a Lagrangian frame are the mass conservation equation (3), and the momentum conservation equation (4),

$$\frac{d\rho}{dt} = -\rho\nabla\cdot\vec{u} \tag{3}$$

$$\rho\frac{du}{dt} = -\nabla p + \mu\nabla^2 u + \rho g \tag{4}$$

where ρ, $\vec{u}$ are the density and velocity of the fluid, and p , μ , g denote pressure, dynamic viscosity, and gravitational constant, respectively.

To close the equations, equation of state (EOS) is indispensable. In general SPH methods, incompressible fluid is assumed to be weakly compressible so that the Tait equation below is used for EOS [2],

$$p = \frac{{c_0}^2\rho_0}{\gamma}\left[\left(\frac{\rho}{\rho_0}\right)^\gamma - 1\right] \tag{5}$$

where ρ_0 , c_0 are the standard reference density and speed of sound, and γ is the polytrophic constant which determines the sensibility of pressure calculation. Table 1 summarizes general SPH equation for mass, momentum conservation and equation of state [1-4].

3. SOPHIA Physical Models

3.1. Multi-phase Model

In SPH multi-phase calculation, there can be other fluid particles with a large density difference within the smoothing radius near the interface between the phases. Thus, the SPH equations should be volume-based form in order to prevent unphysical smoothing of particle properties near the interface. In addition, density renormalization for each phase can improve the stability of multi-phase simulation. In the SOPHIA multi-phase model, density renormalization is performed in every 20-50 time steps.

The surface tension force plays an important role in multi-phase simulation depending on the scale of simulation. In SOPHIA code, a macroscopic continuum surface force (CSF) model is implemented as following equation [5],

$$\left(\frac{d\vec{u}}{dt}\right)_i = -\frac{\sigma}{\rho_i}\kappa_i\left(\nabla c\right)_i \tag{9}$$

where σ is surface tension coefficient, κ is a curvature, and c is a color field for the calculation of gradient vector. In the above equation, gradient vector ∇c and the curvature κ is calculated as below [6],

$$\vec{n_i} = \left(\nabla c\right)_i = \frac{1}{V_i}\sum_j\left(V_i^2 + V_j^2\right)\frac{c_i^{\,i} + c_i^{\,j}}{2}\nabla W_{ij} \tag{10}$$

$$\kappa_i = -\nabla\cdot\left(\frac{\vec{n_i}}{\left|\vec{n_i}\right|}\right) = -n\frac{\sum_j V_j\left(\frac{\vec{n_i}}{\left|\vec{n_i}\right|} - \varphi_{ij}\frac{\vec{n_j}}{\left|\vec{n_j}\right|}\right)\cdot\nabla W_{ij}}{\sum_j V_j\left|r_i - r_j\right|\left|\nabla W_{ij}\right|} \tag{11}$$

where V is a volume of each particle, and φ is a parameter which is 1 if particle j is same phase with particle i and 0 for otherwise.

3.2. Heat Transfer Model

Since the SPH technique is fully Lagrangian numerical method, convection heat transfer is naturally reflected in the motion of particles. The SOPHIA heat transfer equation is given below [7],

$$\left(\frac{dH}{dt}\right)_i = \sum_j \frac{4m_j}{\rho_i\rho_j}\left(\frac{k_i k_j}{k_i + k_j}\right)T_{ij}\frac{\vec{r_{ij}}}{\left|\vec{r_{ij}}\right|}\cdot\nabla W_{ij} \tag{12}$$

where k is a heat conductivity of particle.

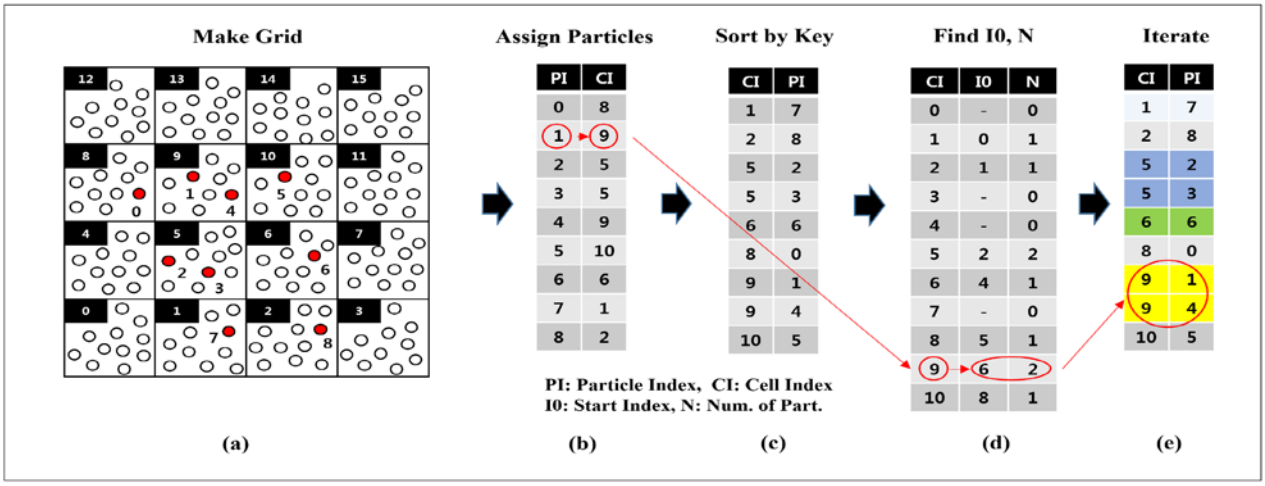

Fig. 3. SOPHIA Nearest Neighboring Particles Search (NNPS) algorithm

4. SOPHIA Code Implementation

4.1. GPU based Parallelization

Particle-based SOPHIA code has time step limitations since its time integration is based on fully explicit scheme. Thus, some strategies are required to improve the efficiency of code calculation. Fortunately, solving the governing equations for each particle is relatively easy to parallelize [8], so GPU-based code parallelization can significantly improve the code efficiency.

Fig. 2 is a simplified representation of CUDA GPU memory structure. Blocks and threads in the figure are used in various parallelization processes of SOPHIA code including mapping, reduction, cumulative sum, and so on. In general cases except for the nearest neighboring particles search (NNPS) algorithm, one block is mapped to each center particle and threads are assigned to each neighboring particle as shown in Fig. 2. After the parallelization, the calculation speed of SOPHIA code increases up to 2 orders of magnitude in some cases.

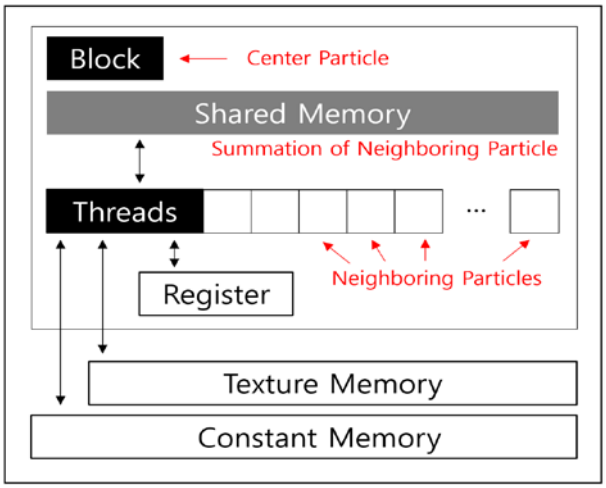

Fig. 2. CUDA GPU memory structure

4.2. Nearest Neighboring Particles Search

The SPH calculation solves the physical models as a form of discretized summation for the neighboring particles, so the neighboring particle search procedure (NNPS) for each particle must be performed before solving the governing equations. This NNPS step is the most time-consuming part of the SPH calculation, so the performance of the whole algorithm largely depends on the efficiency of NNPS procedure.

In the SOPHIA code, counting sort algorithm (Fig. 3) is implemented for NNPS process in which particles are rearranged based on the particle numbers of the cell. GPU parallelization is applied to all the processes of sorting and NNPS step such as particle counting, cumulative summation, and so on.

5. Test Simulation

5.1. Multi-dimensional Dam Break Simulation

The dam-break simulation is a good benchmark problem for the Lagrangian-based numerical analysis code since it is associated with complex phenomena including surface breaking up, high impact pressure, and so on. Fig. 4 shows the snapshots of 2d dam-break simulation and Fig. 5 compares the non-dimensional position on each non-dimensional time between the results of SOPHIA code and experiment of Martin & Moyce (1996) [9], and it shows good agreement. Fig.6 shows the geometry and results of 3d dam-break simulation with square shaped structure.

5.2. Multi-phase Dam Break Simulation

Two-dimensional multi-phase dam break simulation is also conducted to qualitatively validate the SOPHIA multi-phase model. The snapshots of Fig. 7 shows that the interface between two phases and the trapping of low-density fluid are effectively simulated.

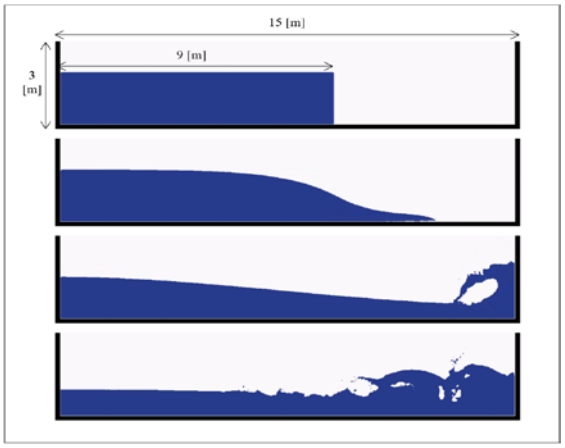

Fig. 4. 2D Dam-break simulation (SOPHIA)

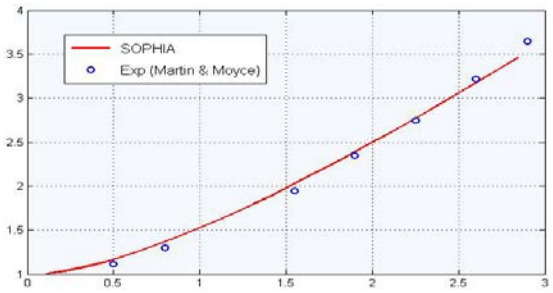

Fig. 5. Validation of 2D Dam-break Simulation

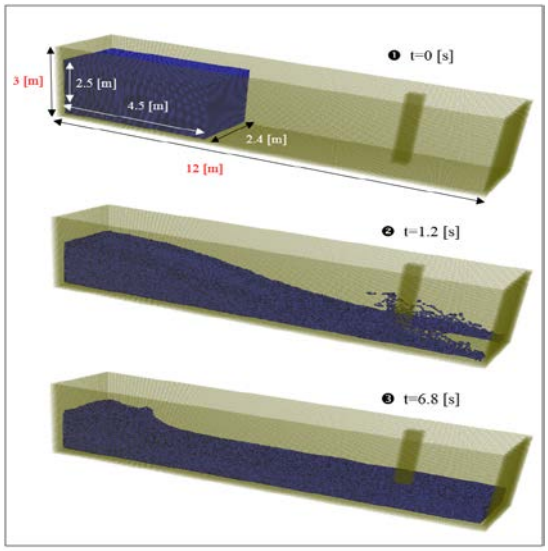

Fig. 6. 3D Dam-break simulation with structure

6. Summary

This paper summarized the recent progress and on-going activities of SOPHIA code, the 3 dimensional multi-physics parallelized SPH code under development in Seoul National University. Basic conservation and physical models for two-phase flow and heat transfer are implemented in current SOPHIA code and all of these interaction models and functions including NNPS are parallelized based on GPU using CUDA. The results of dam-break simulations quantitatively and qualitatively

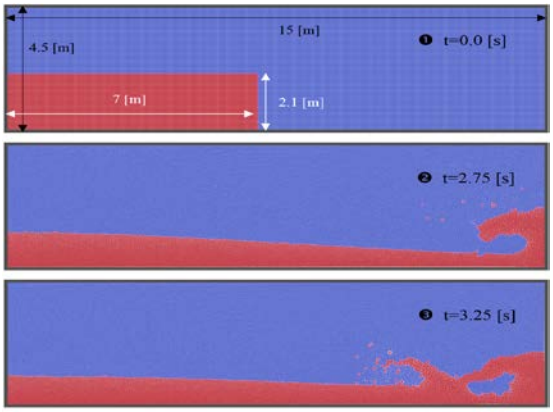

Fig. 7. Multi-phase 2D dam-break simulation (SOPHIA)

shows that SOPHIA code well simulates free surface flows with different phase. The SOPHIA code is expected to be used for various applications related to the thermal-hydraulics on nuclear engineering.

ACKNOWLEDGEMENTS

This research was supported by the National Nuclear R&D Program through the National Research Foundation of Korea (NRF) funded by MSIP; Ministry of Science ICT & Future Planning (No. NRF-2013M2B2B1075735) and the Nuclear Safety Research Program through the Korea Foundation Of Nuclear Safety (KoFONS), granted financial resource from the Nuclear Safety and Security Commission(NSSC), Republic of Korea. (NO. 1403005).

REFERENCES

[1] Pukhov, A., and J. Meyer-ter-Vehn. "Relativistic magnetic self-channeling of light in near-critical plasma: three-dimensional particle-in-cell simulation." *Physical review letters* 76.21 (1996): 3975.

[2] Monaghan, Joe J. "Simulating free surface flows with SPH." *Journal of computational physics* 110.2 (1994): 399-406.

[3] Monaghan, Joe J. "Smoothed particle hydrodynamics." *Reports on progress in physics* 68.8 (2005): 1703.

[4] Mayrhofer, Arno, et al. "Combining Blender with SPHysics, an Introduction." (2010).

[5] Grenier, Nicolas, et al. "An Hamiltonian interface SPH formulation for multi-fluid and free surface flows." *Journal of Computational Physics* 228.22 (2009): 8380-8393.

[6] Adami, S., X. Y. Hu, and N. A. Adams. "A new surface-tension formulation for multi-phase SPH using a reproducing divergence approximation." *Journal of Computational Physics* 229.13 (2010): 5011-5021.

[7] Cleary, Paul W., and Joseph J. Monaghan. "Conduction modelling using smoothed particle hydrodynamics." *Journal of Computational Physics* 148.1 (1999): 227-264.

[8] Harada, Takahiro, Seiichi Koshizuka, and Yoichiro Kawaguchi. "Smoothed particle hydrodynamics on GPUs." *Computer Graphics International*. Petropolis: SBC, 2007.

[9] Koshizuka, Seiichi, and Y. Oka. "Moving-particle semi-implicit method for fragmentation of incompressible fluid." *Nuclear science and engineering* 123.3 (1996): 421-434.