Trough OpenMP Platform for Reducing Computational Time Cost in Underwater Landslide Simulation on Inclined Bottom

P. H Gunawan[1,*], Fadhil Lobma[2]

School of Computing, Telkom University
Jl. Telekomunikasi No 1 Terusan Buah Batu, Bandung 40257, Indonesia
[1]phgunawan@telkomuniversity.ac.id, [2]fadhillobma@student.telkomuniversity.ac.id

ABSTRACT

Simulation of underwater landslide becomes important, since the real underwater landslide phenomena is very dangerous in real life. One of the enormous disasters caused by landslide phenomena can be a Tsunami. Computer simulation of underwater landslide can reduce cost of time and money from conventional simulation (using laboratory). However, to obtain high resolution of computer simulation, large discrete points should be computed. In this paper, the numerical simulation of underwater landslide on inclined bottom using two-layers shallow water equations (SWE) and OpenMP platform is elaborated. Here, the finite volume method framework using upwinding dispersive correction hydrostatic reconstruction (UDCHR) scheme is used. The results of numerical simulation are in a good agreement with the numerical simulation using Nasa-Vof2d numerical scheme. Moreover, parallel computing using OpenMP is observed can reduce the computational time in numerical simulation. in parallel performance, speedup and efficiency of this numerical simulation are observed 2.8 times and 76% respectively at t=0.8 s final time simulation.

Keywords: OpenMP, Parallel computing, Underwater landslide, Simulation, Speedup, Efficiency.

1. INTRODUCTION

Underwater landslide is one of the interesting phenomena in the nature disaster problems. This landslide has a high impact to the living organisms on surrounding water area. One of the dangerous problems is Tsunami, which can be generated by avalanche or landslide of underwater sediment. Indeed, study of this underwater landslide becomes important trough simulation. The conventional simulation (in laboratory) of underwater landslide needs high cost of time and money. Therefore, the computer simulation can be a best choice to reduce this cost. Several computer simulations of underwater landslide can be seen in several references [1, 2, 3, 4].

In this paper, the water level and landslide will be governed using two-layer model of shallow water equations (SWE). The first layer is used to simulate the water flow and second layer is used to perform landslide movement. The two-layer SWE model [5, 6, 7, 8, 9, 10] is given as follow,

$$\partial_t h_1 + \partial_x (h_1 u_1) = 0, \tag{1}$$

$$\partial_t \ (h_1 u_1) + \partial_x \left(h_1 u_1^2 + \frac{gh_1^2}{2}\right) = -gh_1 \partial_x \ (h_2 - z), \tag{2}$$

$$\partial_t \ h_2 + \partial_x \ (h_2 u_2) = 0, \tag{3}$$

$$\partial_t \ (h_2 u_2) + \partial_x \left(h_2 u_2^2 + \frac{gh_2^2}{2}\right) = -\frac{\rho_1}{\rho_2} \ gh_2 \partial_x \ (h_1 - z) \tag{4}$$

where the total depth of water is denoted by h, water velocity is described by u, gravitational force is given as g, fix bottom/bathymetry is denoted by z, density of layer is denoted by ρ, time and spatial is given by t and x respectively. Moreover, subscripts 1 and 2 denotes first and second layer of water. In this model, equations (1-2) are called the mass and momentum conservation equation for the water movement. Meanwhile equations (3-4) are called mass and momentum conservation equation for the sediment flow.

To approximate (1-4), several numerical scheme can be used, see [11, 12, 13]. In the references [11, 12], Eqs (1-4) are approximated using source-centered hydrostatic reconstruction (SCHR) scheme. This scheme is shown has a good result for simulating under water avalanche and erodible dambreak. However according to [14], this scheme is failed to tackle upwind data when the eigenvalues have the same sign. Therefore, in this paper, the upwinding dispersive correction hydrostatic reconstruction (UDCHR) numerical scheme will be used. This scheme is used since this scheme is mathematically proved satisfying the following properties such as preserving wet-dry simulation, well-balanced scheme, satisfying a semi-discrete entropy condition, working for arbitrary number of layers and densities, etc (see [14] for more detail).

For minimizing computational time cost due to the increasing of discrete points, parallel computing will be used. There are several platforms can be used in parallel computing which depend on their architecture. In multi-core parallel programming, there are two types of parallel architecture, distributed and shared parallel architecture. Here, shared parallel architecture with OpenMP platform is chosen since its simplicity and straightforward to implement in serial code. The advantages of using OpenMP in numerical simulation also can be found in several references, for instance see [11, 13, 12].

The structure of the rest of this paper is given as follows, in Section 2, the brief explanation about UDCHR numerical scheme is given. In Section 3, the algorithm of UDCHR in parallel and serial architecture is elaborated. The results and discussion of numerical simulation of underwater landslide and parallel performance are shown in Section 4. Moreover, the conclusion of this paper is presented in Section 5.

2. NUMERICAL SCHEME: UDCHR

The upwinding dispersive correction reconstruction (UDCHR) is a modified scheme of source centered hydrostatic reconstruction (SCHR) [14]. The improvement of this scheme is in the ability of UDCHR for handling upwind data, where the eigenvalues produce same sign. Next, the brief explanation of UDCHR will be given.

This scheme approximates (1-4) in finite volume method framework which given as follow, for each layers,

$$U_i^{\{n+1\}} - U_i^n + \frac{\Delta t}{\Delta x}\left(F_{\{i+\frac{1}{2}\}} - F_{\{i-\frac{1}{2}\}}\right) = 0, \tag{5}$$

where

$$\begin{cases} F_{\{i+\frac{1}{2}\}} = \mathcal{F}_l^{\{HR\}}(U_i, U_{i+1}, z_i, z_{i+1}) + \mathcal{J}_l, \\ F_{\{i-\frac{1}{2}\}} = \mathcal{F}_r^{\{HR\}}(U_{i-1}, U_i, z_{i-1}, z_i) + \mathcal{J}_r. \end{cases} \tag{6}$$

This finite volume framework is known as hydrostatic reconstruction scheme where the correction of bottom energy is involved. Moreover the detail of numerical fluxes (6) is written as,

$$\mathcal{F}_l^{\{HR\}}(U_l, U_r, z_l, z_r) = \mathcal{F}(U_l^*, U_r^*) + \begin{pmatrix} 0 \\ p(h_l) - p(h_l^*) \end{pmatrix},$$

$$\mathcal{F}_r^{\{HR\}}(U_l, U_r, z_l, z_r) = \mathcal{F}(U_l^*, U_r^*) + \begin{pmatrix} 0 \\ p(h_r) - p(h_r^*) \end{pmatrix}. \tag{7}$$

Here, in (7), numerical flux $\mathcal{F}(U_l^*, U_r^*)$ is a numerical flux without bottom friction/ bathymetry effect in one layer shallow water equation and $p(h) = gh^2/2$. Additionally, the reconstructed states are given as

$$U_l^* = (h_l^*, h_l^* u_l), \quad h_l^* = \max(0, h_l + z_l - z^*), \quad U_r^* = (h_r^*, h_r^* u_r)$$
$$h_r^* = \max(0, h_r + z_r - z^*), \quad z^* = \max(z_l, z_r). \tag{8}$$

Moreover in (6), the numerical fluxes $\mathcal{J}_{l/r} = \left(\mathcal{J}^0, \mathcal{J}_{l/r}^1\right)$ should satisfy,

$$\mathcal{J}_l^1 = \frac{g\kappa}{2}(1 + \theta)(h_r - h_l + \Delta z) + u_l \max(0, \mathcal{J}^0) + u_r \min(0, \mathcal{J}^0), \tag{9}$$

$$\mathcal{J}_l^1 = -\frac{g\kappa}{2}(1 - \theta)(h_r - h_l + \Delta z) + u_l \max(0, \mathcal{J}^0) + u_r \min(0, \mathcal{J}^0), \tag{10}$$

$$\mathcal{J}^0 = \frac{\kappa}{2}\left((u_l + U_r) + \theta(u_l - u_r)\right), \tag{11}$$

$$\theta = \min\left(\frac{1, max(0, u_l)}{\sqrt{gh_l}}\right) - \min\left(1, \frac{max(0, u_r)}{\sqrt{gh_r}}\right), \tag{12}$$

$$\kappa = \tilde{\kappa} \; if \; |\tilde{\kappa}| \le \frac{5}{2}\min(h_l, h_r), \text{otherwise } \kappa = \frac{5\tilde{\kappa}}{2|\tilde{\kappa}|}\min(h_l, h_r) \tag{13}$$

here, $\Delta z = z_r - z_l$ and for simplicity the definition of $\tilde{\kappa}$ can be seen in the reference [14]. Note that, the numerical fluxes (6-13) are implemented for each layers in two-layer model of SWE. In the next section, the algorithm of this numerical scheme will be given in serial and parallel architecture.

3. PARALLEL ARCHITECTURE AND ITS ALGORITHM

The algorithm for simulating underwater landslide using numerical scheme (5) and its numerical fluxes (6-13) can be seen in Figure 1.
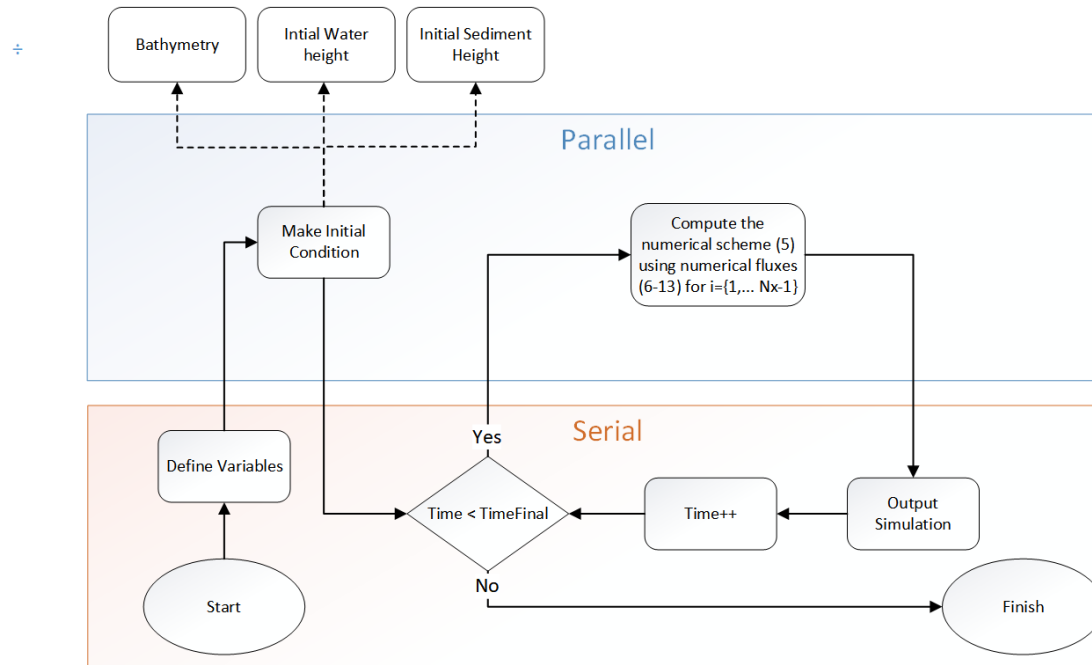

FIGURE 1. The parallel mechanism for simulating underwater landslide using UDCHR numerical scheme in OpenMP platform.

In Figure 1, defining variables, time evolution and output simulation are done in serial part. Since these operations are not suitable for parallelization using multi-core architecture. In parallel part, setting the initial condition and computing the numerical scheme using numerical fluxes UDCHR (6-13) are elaborated. Here, the parallel is done since the independence of data is high. The discrete spatial points in current time are calculated using the discrete spatial points in previous time. Here, the OpenMP platform is used since its simplicity. The OpenMP is implemented directly in the process of parallel. Generally, OpenMP syntax is carried out in looping process in the serial codes. See [15] for more detail about OpenMP framework or see [16, 17, 18, 19, 16] for the application of OpenMP in several numerical simulations.

In the next simulation, the parallel performance will be analyzed using the following computers specifications:

TABLE 1.
The specifications of computer for simulation.

Computer	Operating System	Processor	Memory
I	Ubuntu 16.04 LTS	Intel Core i7-7500U	8 Gb
II	Windows 10	Intel Core i3-6006U	4 Gb

4. RESULT AND DISCUSSION

Here, numerical simulation of water landslide using two-layer SWE and UDCHR scheme is given. The results are compared with the numerical result in the paper of [20], where the Nasa-vof2d is used to simulate the problem. Moreover, the parallel computing performance in reducing computational time cost using OpenMP is elaborated.

4.1 NUMERICAL SIMULATION: UNDERWATER LANDSLIDE

Here, the initial configuration of each layer of two-layer SWE model and its bathymetry for simulating underwater landslide simulation should be defined. The configuration of each layers and inclined bottom (bathymetry) is given as follows,

$$h_1(x, 0) = \max\big(0, 1.5 - h_2(x, 0) - z(x)\big), \quad u_1(x, 0) = u_2(x, 0) = 0, \tag{14}$$

$$h_2(x) = \begin{cases} \max\big(0, 1.35 - z(x)\big), & \text{if } 0.25 < x \le 0.8 \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

$$z(x) = \begin{cases} 1.4 & \text{if } x \le 0.2 \\ -\dfrac{(1.4x + 1.95)}{1.2} & \text{if } 0.2 < x \le 1.4 \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

From these initial conditions (14-16), the illustration of these initial conditions can be described in Figure 2. The first layer describes the water level h_1 and the second layer describes the landslide sediment layer h_2. The inclined bottom is given by fix bottom equation (16) and describes by grey color in Figure 2. This inclined bottom is adjusted as in the reference [20].
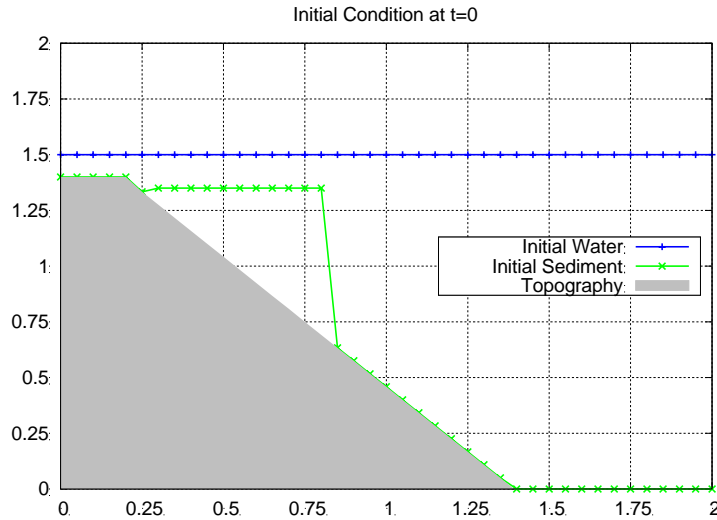
FIGURE 2. Initial condition for underwater landslide simulation

Using the ratio of density $\rho1/\rho2 = 0.83$, the results of this simulation in final time $t = 0.4$ and $t = 0.8$ s can be found in Figure 3. Here, the comparison of numerical results using UDCHR with the numerical results from [20] using Nasa-Vof2d numerical scheme is presented. It can be seen that the results of UDCHR is in a good agreement with the numerical results of Nasa-Vof2d.
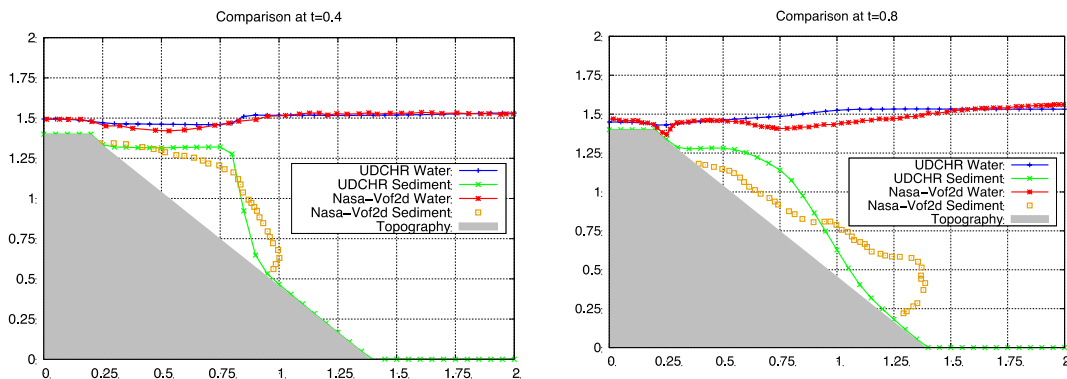

FIGURE 3. The comparison of numerical results UDCHR and Nasa-

From Figure 3, the water and sediment profile of UDCHR and Nasa-Vof2d at time $t = 0.4$ and $t = 0.8$ s are very close. As explained in [20], Nasa-Vof2d numerical scheme includes the characteristics of sediment. Meanwhile, here the UDCHR scheme is used for two-layer water model where the characteristics of sediment are omitted. However, using two-layer water model, underwater landslide simulation is well elaborated.

4.2 OPENMP PERFORMANCE

In order to show the performance of parallel computing OpenMP in this simulation, the tables of CPU time in serial and parallel in final time $t = 0.4$ and $t = 0.8$ seconds are given in Table 2 and Table 3 respectively. Here, the measurement

uses six different numbers of points. The lowest and largest number of discrete points in this simulation is 50 and 1600 points respectively.

TABLE 2.

The comparison of CPU time in serial and parallel at final time simulation t = 0.4 s.

Number of Points	Serial		Parallel		Speedup	
	Comp. I	Comp. II	Comp. I	Comp. II	Comp. I	Comp. II
50	0.071041	0.682768	0.028087	0.3999375	2.52932	1.707187
100	0.135972	1.36624	0.046746	0.69852	2.908741	1.955907
200	0.241486	1.72921	0.085994	0.824401	2.808173	2.097535
400	0.473676	3.19469	0.157073	1.41573	3.015642	2.256567
800	0.956489	6.52481	0.31758	2.77189	3.011805	2.353921
1600	1.92857	13.1842	0.635463	5.53812	3.034905	2.380627

TABLE 3.

The comparison of CPU time in serial and parallel at final time simulation t = 0.8 s.

Number of Points	Serial		Parallel		Speedup	
	Comp. I	Comp. II	Comp. I	Comp. II	Comp. I	Comp. II
50	0.13562	1.64418	0.049642	0.853534	2.731961	1.92632
100	0.268165	2.282755	0.09148	1.42747	2.931406	1.599161
200	0.463308	3.49409	0.156112	1.44475	2.967792	2.418474

400	0.925117	7.04056	0.312607	2.47288	2.959361	2.847109
800	1.90866	12.47	0.623427	5.7169	3.061561	2.181252
1600	3.80306	25.7822	1.31378	10.445	2.894746	2.468377

In Table 2, the performance of Computer I is observed better than Computer II. It can be seen clearly that, using large number of points (1600 discrete points), the speedup performance of parallel computing in Computer I is 3 times, meanwhile in Computer II is observed 2.3 times. Moreover, even if the time of simulation is increased until t = 0.8 (Table 3), I is observed has higher speedup which is approximately 2.9 times. Meanwhile using Computer II, computer speedup is obtained approximately 2.4 times at time t = 0.8 of final time simulation.
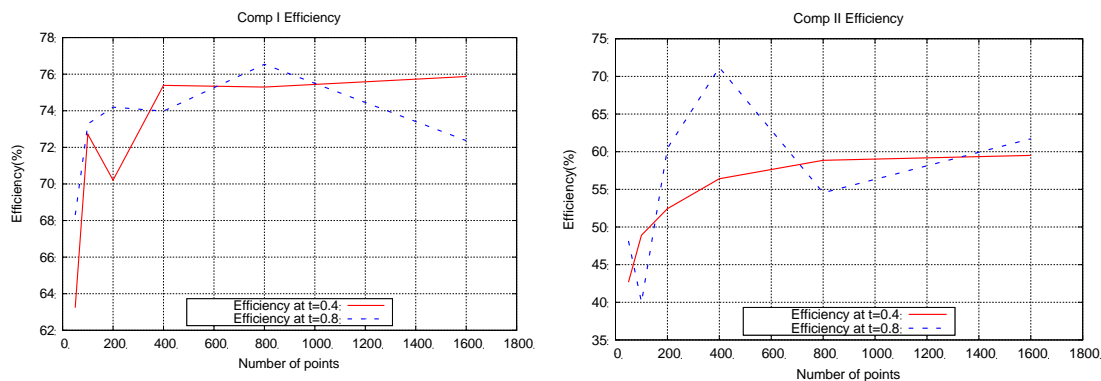

FIGURE 4. The efficiency profile of OpenMP using Computer I (left) and Computer II (right).

Further, the efficiency of using parallel computing in this simulation should be investigated. In Figure 4, efficiency using parallel computing for each computers are given. In Figure 4 (left), Computer I has high efficiency performance, running large time of simulation, the efficiency is in stable along the increasing of number of points. Here, using 1600 discrete points and final time t = 0.8 s, around 76% of efficiency is observed in Computer I. Meanwhile in Computer II, the efficiency is obtained around 61%.

From the parallel performance (speedup and efficiency), Computer I shows has a good performance rather than Computer II. This results is supported by the specification of each computer as shown in Table 1. Obviously, Computer I has high processor level which is Intel Core i7 compared with Computer II which has processor Intel Core i3. Note that Intel Core i7 is highest generation of Intel Core recently. Therefore, the performance results are in a good agreement with the specification of computers.

5. CONCLUSION

The numerical simulation of underwater landslide on inclined bottom using two-layer SWE model and UDCHR numerical scheme is presented. Here, results of numerical simulation are observed in a good agreement with the numerical simulation using Nasa-vof2d as shown in paper of [20]. In this paper, the parallel performance using OpenMP platform is shown satisfying. The speedup using Computer I is obtained 2.89 times, higher than Computer II which has speedup 2.4 times at final time simulation t = 0.8 s. Moreover, Computer I has higher efficiency performance around 76%, meanwhile Computer II has approximately 61% of efficiency performance.

REFERENCES

[1] E. D. Fernández-Nieto, "Modelling and numerical simulation of submarine sediment shallow flows: transport and avalanches," Boletín de la Sociedad Española de Matemática Aplicada, no. 49, pp. 83–103, 2009.

[2] S. T. Grilli and P. Watts, "Modeling of waves generated by a moving submerged body. Applications to underwater landslides," Engineering Analysis with boundary elements, vol. 23, no. 8, pp. 645–656, 1999.

[3] P. Lynett and P. L.-F. Liu, "A numerical study of submarine-landslide-generated waves and run–up," in Proceedings of the Royal Society of London A: Mathe-matical, Physical and Engineering Sciences, vol. 458, no. 2028. The Royal Society, 2002, pp. 2885–2910.

[4] E. B. Pitman, C. C. Nichita, A. Patra, A. Bauer, M. Sheridan, and M. Bursik, "Computing granular avalanches and landslides," Physics of fluids, vol. 15, no. 12, pp. 3638–3646, 2003.

[5] M. Castro-Dı́az, E. D. Fernández-Nieto, J. M. González-Vida, and C. Parés-Madroñal, "Numerical treatment of the loss of hyperbolicity of the two-layer shallow-water system," Journal of Scientific Computing, vol. 48, no. 1-3, pp. 16–40, 2011.

[6] M. J. Castro, J. A. Garcıa-Rodrıguez, J. M. González-Vida, J. Macıas, C. Parés and M. E. Vázquez-Cendón, "Numerical simulation of two-layer shallow water flows through channels with irregular geometry," Journal of Computational Physics, vol. 195, no. 1, pp. 202–235, 2004.

[7] R. Abgrall and S. Karni, "Two-layer shallow water system: a relaxation approach," SIAM Journal on Scientific Computing, vol. 31, no. 3, pp. 1603–1627, 2009.

[8] M. Castro, J. Macías, and C. Parés, "A Q-scheme for a class of systems of coupled conservation laws with source term. Application to a two-layer 1-D shallow water system.," ESAIM: Mathematical Modelling and Numerical Analysis, vol. 35, no. 1, pp. 107–127, 2001.

[9] F. Bouchut and T. M. de Luna, "An entropy satisfying scheme for two-layer shallow water equations with uncoupled treatment," ESAIM: Mathematical Modelling and Numerical Analysis, vol. 42, no. 4, pp. 683–698, 2008.

[10] A. Kurganov and G. Petrova, "Central-upwind schemes for two-layer shallow water equations," SIAM Journal on Scientific Computing, vol. 31, no. 3, pp. 1742–1773, 2009.

[11] C. A. Simanjuntak, B. A. R. H. Bagustara, and P. H. Gunawan, "Computational multicore on two-layer 1d shallow water equations for erodible dambreak," in Journal of Physics: Conference Series, vol. 971, no. 1. IOP Publishing, 2018, p. 012034.

[12] B. A. R. H. Bagustara, C. A. Simanjuntak, and P. H. Gunawan, "Multicore runup simulation by under water avalanche using two-layer 1d shallow water equations," in Journal of Physics: Conference Series, vol. 971, no. 1. IOP Publishing, 2018, p. 012026.

[13] C. A. Simanjuntak and P. H. Gunawan, "Computing two-layer swe for simulating submarine avalanches on openmp," in Control, Electronics, Renewable Energy and Communications (ICCREC), 2017 International Conference on. IEEE, 2017, pp. 190–195.

[14] F. Bouchut and V. Zeitlin, "A robust well-balanced scheme for multi-layer shallow water equations," Discrete and Continuous Dynamical Systems-Series B, vol. 13, no. 4, pp. 739–758, 2010.

[15] P. H. Gunawan, "Scientific parallel computing for 1d heat diffusion problem based on openmp," in Information and Communication Technology (ICoICT), 2016 4th International Conference on. IEEE, 2016, pp. 1–5.

[16] M. L. P. Siagian and P. H. Gunawan, "Parallel processing for simulating surface gravity waves by non-hydrostatic model using arakawa grid," in Control, Electronics, Renewable Energy and Communications (ICCREC), 2017 International Conference on. IEEE, 2017, pp. 164–168.

[17] S. Juliati and P. H. Gunawan, "Openmp architecture to simulate 2d water oscillation on paraboloid," in Information and Communication Technology (ICoIC7), 2017 5th International Conference on. IEEE, 2017, pp. 1–5.

[18] M. N. A. Alamsyah, A. Utomo, and P. H. Gunawan, "Analysis openmp performance of amd and intel architecture for breaking waves simulation using mps," in Journal of Physics: Conference Series, vol. 971, no. 1. IOP Publishing, 2018, p. 012022.

[19] I. B. P. A. Pranidhana and P. H. Gunawan, "Computational parallel for shallow water-sediment concentration coupled model," in Journal of Physics: Conference Series, vol. 971, no. 1. IOP Publishing, 2018, p. 012032.

[20] S. A. Rzadkiewicz, C. Mariotti, and P. Heinrich, "Numerical simulation of submarine landslides and their hydraulic effects," Journal of Waterway, Port, Coastal, and Ocean Engineering, vol. 123, no. 4, pp. 149–157, 1997.