

Jurnal Ilmu Komputer dan Informasi (Journal of a Science and Information). 12/1 (2019), 31-39
DOI: <http://dx.doi.org/10.21609/jiki:v12i1:635>

SHARED MEMORY ARCHITECTURE FOR SIMULATING SEDIMENT-FLUID FLOW BY OPENMP

P. H. Gunawan^{1,*} and A. Utomo²

School of Computing, Telkom University, Jl. Telekomunikasi No 1, Terusan Buahbatu, Bandung, 40257, Indonesia

E-mail: ardhitoutm@student.telkomuniversity.ac.id¹, phgunawan@telkomuniversity.ac.id²

*Corresponding Author

Abstract

This paper elaborates the simulation of fluid flow using shallow water equations (SWE) and sediment movement below it using Exner equation. Both of those equations are combined using splitting technique, in which SWE would be computed using collocated Finite Volume Method (FVM) with Harten-Lax-van Leer and Einfeldt (HLLC) numerical flux, then Exner would be computed semi-implicitly. This paper gives several steps of constructing numerical scheme for approximating SWE-Exner model. In order to validate the numerical scheme, two problems will be elaborated: (1) comparison between analytical solution and numerical solution, and (2) parallelism using OpenMP for transcritical over a granular bump. For the first problem, the measurement of numerical scheme is observed by discrete L^p - norm error. Here, using 1600 grid points, discrete L^1 -, L^2 -, and L^∞ - norm errors for sediment are obtained 1.257E-3, 1.299E-3 and 1.492E-2 respectively. Moreover, for the second problem, speedup and efficiency of parallel computing are given to show the successful of parallel technique. In this second simulation, parallel computing can increase the speedup from serial computing almost 3 times for number of grids more than 800 points. Furthermore, efficiency is observed reach maximum up to 70%, which in average around 56.44%.

Keywords: SWE, Exner, HLLC, analytical, parallel computing, OpenMP

Abstrak

Jurnal ini membahas mengenai simulasi persamaan air dangkal menggunakan *shallow water equations* (SWE) dan pergerakan sedimen di bawahnya menggunakan *Exner equation*. Penggunaan persamaan-persamaan tersebut dipadukan secara *splitting*, dimana SWE dihitung menggunakan *collocated Finite Volume Method* (FVM) dengan fluks numerik *Harten-Lax-van Leer and Einfeldt*, lalu *Exner* akan dicari solusinya secara semi-implicit. Jurnal ini menjabarkan mengenai beberapa langkah pembuatan skema numerik untuk menghampiri model SWE-Exner. Untuk memvalidasi hasil skema numerik, dua buah masalah dikerjakan: (1) perbandingan dengan skema analitik, dan (2) penggunaan paralel pada skema *transcritical over a granular bump* menggunakan platform openMP. Untuk masalah pertama, pengukuran akan hasil skema numerik diberikan menggunakan *discrete L^p - norm error*. Dalam jurnal ini, menggunakan 600 titik grid, *discrete L^p - norm errors* untuk variabel sedimen didapatkan 1.257E-3, 1.299E-3 dan 1.492E-2 secara berurutan. Selanjutnya, pada masalah kedua, speedup dan efisiensi dari komputasi paralel diberikan untuk memperlihatkan kesuksesan dari teknik paralel. Pada simulasi kedua ini, komputasi paralel dapat menaikkan nilai speedup dari komputasi serial hampir 3 kalinya menggunakan jumlah grid melebihi 800 titik. Selain itu, efisiensi diamati mencapai maksimum sampai 70%, dengan rata-ratanya sekitar 56.44%.

Kata Kunci: SWE, Exner, HLLC, analitik, komputasi paralel, OpenMP

1. Introduction

Particle movements below the surface of fluid flow are divided into three categories: saltation, suspension, and bedload. Saltation is defined as an alteration of particles position from one place to another. Meanwhile, suspension is defined as the articles moving together with the fluid when they

are affected by huge amount of flux, and bed-load is defined as a movement of particles at the bottom of the current [1], [2]. Generally, those particle movements can produce the sedimentation under water flow. However, sedimentation can produce negative impact or natural hazard for living organisms near water area, such as flooding, erosion, etc.

Through this paper, the simulation of sediment movement by mathematical model and its discretization will be elaborated [2]. Here, the focus of sediment is on bed-load particles movement. Moreover, the mathematical models shallow water equations (SWE) and Exner equation are used to model the fluid current and the sediment movement respectively. The illustration of one-dimensional model of SWE-Exner can be seen in Figure 1.

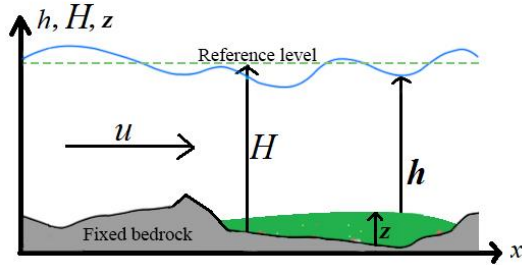


Figure 1. The one-dimensional illustration of shallow water with sediment and fixed bedrock in horizontal slice.

Since SWE-Exner equations in this paper are discretized using Finite Volume Method (FVM), the degree of fineness in making the simulation is dependent on how many grid points used. In numerical simulation, explicit numerical scheme of FVM is used. Thus the time step of simulation for covering stability analysis depends on number of grids. Therefore, the more number the grid point has, the longer numerical computation would be taken [3].

With parallel technique, the execution time of the code in numerical scheme can be reduced. Some of the aspects that determine the efficiency between the parallel and serial technique are the simplicity of the code, the model of shared memory technique, the parallel architecture of the computer, and the robustness of the parallel code [4]. A time comparison between parallel and serial technique would help to determine how big the impact given by the parallel. Then, the measurement of parallel computing is given by calculating its speedup and efficiency.

OpenMP is one of the method in parallel technique [5]. Its simplicity (see [4], [6]) is one of the main reasons people choose it instead of other technique (you can also see the detail of process of OpenMP in Section 3). For instance, computing MPS method in [4] is shown to give 49.57% of efficiency in approximation. Moreover, in the simulation of water oscillation on paraboloid using SWE in [7], its efficiency of OpenMP is obtained up to 84.4%. Both of them show that OpenMP is a good choice to accelerate the computation and shorten the computation time.

This paper will point out two things: (1) the errors of the SWE-Exner model, by comparing between the solution of analytical scheme and numerical scheme and (2) simulating a transcritical over a granular bump model and discover its speedup and efficiency in serial and parallel technique. The errors in the first problem will be seen by showing the discrete L^1 -, L^2 -, and L^∞ -norm error. The parallel architecture's model in this paper for the second problem is OpenMP with Intel Core i5-4200 @2.3GHz using 4 threads.

2. Shallow water-Exner equation and its discretization

2.1 Governing equations

Shallow Water Equations (SWE) is one of the hyperbolic system in conservation law which is used to solve geophysical flow problems [8], [9]. Generally, SWE is applied for water flow and gas flow. For water flow, SWE may be applied if a current has a wave length more than twenty times bigger than its depth, or $\lambda \geq 20h$ [10].

SWE is formed by two equations: (1) continuity equation which is obtained from mass conservation law, and (2) momentum equation which is obtained from momentum conservation law. And because in this paper the depth of the bedrock to the reference level (H) is fixed, thus it becomes $\partial_x H = 0$, so SWE can be described as following system of equations,

$$\partial_t h + \partial_x(hu) = 0, \quad (1)$$

$$\partial_t hu + \partial_x(hu^2 + \frac{1}{2}gh^2) = -gh\partial_x(z) - ghS_f, \quad (2)$$

where x describes the spatial coordinate, t as time, $h(x, t)$ as water depth, g as gravity, $u(x, t)$ as velocity of the fluid, S_f as friction slope, and $z(x, t)$ as the height of sediment (see Figure 1 for more detail about these variables). The friction slope used in this paper is obtained using Darcy-Weisbach friction force, because of the fluid stream tend to be turbulent [11]. Its function is given as,

$$S_f = C_f \frac{|u|u}{8gh}, \quad (3)$$

with C_f is Darcy-Weisbach friction factor.

In order to describe the interaction between fluid (SWE) and sediment, then the mathematical model of sediment should be given. Here, Exner model is used to simulate the movement of sedimentation. Exner equation is an equation derived from conservation mass that is used to model sediment that moves because of the interaction with the flow of fluid [9]. Exner equation may be combined with SWE to model bed-load movement below the fluid current as sediment elevation z , which can be affected by the

velocity of water u [12]. Its equation can be explained as,

$$(1 - \phi)\partial_t z + \partial_x (A_g(u)|u|^{m-1}u) = 0, \quad (4)$$

where ϕ is sediment porosity, u is the velocity as in Eqs. (1-2), m is a parameter for the sediment with range $1 \leq m \leq 4$ (see [13] for more detail about this parameter), and A_g is a number around $0 \leq A_g \leq 1$ to show the interaction strength between the fluid and the sediment; the higher value A_g , the stronger the interaction between those two. Note that, this Exner equation uses the Grass formula [13] as the bed load function. See references [9], [14] for more detail about other bed load functions.

2.2 Numerical Scheme

Using Finite Volume Method (FVM), SWE is applied from system of conservation law [15]. The equation of fluid flow (1-2) can be given in a compact form [16] such as,

$$\partial_t U + \partial_x (F(U)) = S(U), \quad (5)$$

where,

$$U = \begin{pmatrix} h \\ hu \end{pmatrix}, \quad F(U) = \begin{pmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \end{pmatrix}, \quad (6)$$

$$S(U) = \begin{pmatrix} 0 \\ -gh(\partial_x(z) + S_f) \end{pmatrix}.$$

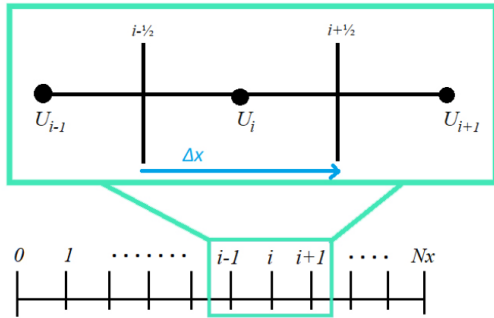


Figure 2. The one-dimensional explicit stencil in FVM shows that Δx is found by subtracting right and left side of every grid point, and every point has the same gap length to other point before or after it (same value of Δx).

Value of spatial length can be obtained by decreasing one spatial point and the number before it (see Figure 2). It also can be written as,

$$\Delta x_i = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}} \geq 0. \quad (7)$$

Then, discrete value of time can be obtained using $t^n = n\Delta t$ with n as a natural number and $\Delta t \geq 0$. Using conservation scheme in Eq. (5), SWE is transformed in numerical scheme to,

$$U_i^{n+1} - U_i^n + \frac{\Delta t}{\Delta x} (F_{i+\frac{1}{2}} - F_{i-\frac{1}{2}}) = 0. \quad (8)$$

The fluxes F are defined using hydrostatic reconstruction scheme, which is,

$$F_{i+\frac{1}{2}} = \mathcal{F}(U_{i+\frac{1}{2}L}^n, U_{i+\frac{1}{2}R}^n) + \begin{pmatrix} 0 \\ \frac{g(h_{i+\frac{1}{2}L}^n)^2}{2} - \frac{g(h_{i+\frac{1}{2}R}^n)^2}{2} \end{pmatrix}, \quad (9)$$

$$F_{i-\frac{1}{2}} = \mathcal{F}(U_{i-\frac{1}{2}L}^n, U_{i-\frac{1}{2}R}^n) + \begin{pmatrix} 0 \\ \frac{g(h_{i-\frac{1}{2}L}^n)^2}{2} - \frac{g(h_{i-\frac{1}{2}R}^n)^2}{2} \end{pmatrix}, \quad (10)$$

with \mathcal{F} is a numerical flux without source terms. This paper will use Harten-Lax-Van Leer and Einfeldt (HLLE) numerical flux. Then, $U_{i+\frac{1}{2}R}^n$ and $U_{i+\frac{1}{2}L}^n$ can be written as,

$$U_{i+\frac{1}{2}R}^n = (h_{i+\frac{1}{2}R}^n, h_{i+\frac{1}{2}R}^n u_{i+1}^n)^T, \quad (11)$$

$$U_{i+\frac{1}{2}L}^n = (h_{i+\frac{1}{2}L}^n, h_{i+\frac{1}{2}L}^n u_i^n)^T, \quad (12)$$

where,

$$h_{i+\frac{1}{2}R} = (h_{i+1}^n - (-\Delta Z)_+)_+, \quad (13)$$

$$h_{i+\frac{1}{2}L} = (h_i^n - (\Delta Z)_+)_+, \quad (14)$$

$$\Delta Z = z_{i+1}^n - z_i^n, \quad (15)$$

with the notation $a_+ = \max(a, 0)$.

For the friction force in topography, Darcy-Weisbach is used by substituting ΔZ with ΔZ_{app} , which is,

$$\Delta Z_{app} = \Delta Z + \Delta F_{i+\frac{1}{2}}, \quad (16)$$

with $\Delta F_{i+\frac{1}{2}}$ can be further known in [2].

Finally, the numerical flux, Harten-Lax-van Leer and Einfeldt (HLLE) numerical flux is given as,

$$\mathcal{F}(U_{i+\frac{1}{2}L}^n, U_{i+\frac{1}{2}R}^n) = t_1 F(U_{i+\frac{1}{2}R}^n) + t_2 F(U_{i+\frac{1}{2}L}^n) - t_3 (U_{i+\frac{1}{2}R}^n - U_{i+\frac{1}{2}L}^n), \quad (17)$$

with the value of t_1 , t_2 , and t_3 could be further known in [2].

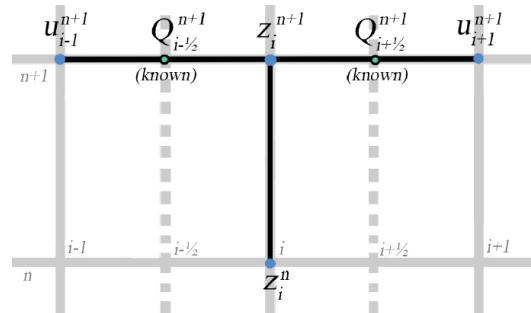


Figure 3. Semi-implicit stencil in Exner Eqs. 18. Note that $Q_{i\pm\frac{1}{2}}^{n+1}$ are found using $u_{i\pm\frac{1}{2}}^{n+1}$ which could be obtained by processing the value of u_{i+1}^{n+1} , u_i^{n+1} , and u_{i-1}^{n+1} .

The solution of Exner equation will be searched after the solutions of SWE (8-10) have found.

Splitting technique is applied when HLLC method is no longer used in Exner, but semi-implicit method instead. It means that the value of z_i^{n+1} not only needs z_i^n , it also needs u^{n+1} that is found earlier from SWE (see Figure 3). Exner equation can be discretized as follows [14].

$$z_i^{n+1} = z_i^n - \frac{\Delta t}{(1-\phi)\Delta x}(Q_{i+\frac{1}{2}}^{n+1} - Q_{i-\frac{1}{2}}^{n+1}), \quad (18)$$

with the value of $Q_{i+\frac{1}{2}}^{n+1}$ and $Q_{i-\frac{1}{2}}^{n+1}$ are found using *grass formula*, so they become,

$$Q_{i\pm\frac{1}{2}}^{n+1} = A_g u_{i\pm\frac{1}{2}}^{n+1} |u_{i\pm\frac{1}{2}}^{n+1}|^{m-1}, \quad (19)$$

with $m = 3$ is usually used [9]. After that, the value of $u_{i+\frac{1}{2}}^{n+1}$ and $u_{i-\frac{1}{2}}^{n+1}$ are determined using approximation, which are given as,

$$u_{i\pm\frac{1}{2}}^{n+1} = \frac{u_{i\pm 1}^{n+1} + u_i^{n+1}}{2}. \quad (20)$$

3. Parallel Architecture

Parallel architecture is a method used to reduce memory access time. They work by splitting tasks to threads, then each thread will work on the task until it is done. One of the most popular shared-memory programming model is Open Multi Processing (OpenMP). In OpenMP, each worker is given a chunk of tasks (iterations of a parallel-for-loop, or sections in a parallel-sections block), and a barrier construct is used to synchronize the workers (not the tasks) [17].

OpenMP program is essentially a sequential program augmented with compiler directives to specify parallelism. Each thread, if it works below the OpenMP program, has its own program counter and executes one task at a time, similar to sequential program execution [18] (see Figure 4). For more information, see [19] about OpenMP description. This paper uses OpenMP as its shared-memory programming model because of its simplicity [3], [4], [6], [7], [20]–[22].

OpenMP has its own popularity because,

- it is implemented incrementally [23],
- it can still running serial program without modification needed [24], and
- it has directive based interface [25].

OpenMP directives are categorized to three parts [23],

- Work-sharing specified (*omp parallel for*, *omp parallel section*, *omp single*), that focus on partition work between process. One of the example is in *omp parallel for*, if you run

```
#pragma omp parallel for
for(int k=1;k<12;k++){
    printf("thread %d, i %d \n"
        ,omp_get_thread_num(), i);
}
```

then the output would likely be like Figure 5. In Figure 5, the number of threads is 4 and the number of arrays is 12. Thus by `#pragma omp parallel for` command, the array indexes 0-2 will be executed by thread 1, then followed by other threads.

- Data properties specified (*openmp shared*, *openmp private*, *reduction*, *firstprivate* etc.) that focus on constructing the variables in the parallel part. The default of variables in OpenMP standard is shared.
- Synchronization (*omp barrier*, *omp atomic*, *omp critical*, *omp flush* etc.) that is incorporated into the control flow graph.

To see the details of every directives, reader is encouraged to see [25] for more detail. OpenMP also has basis syntax and runtime library routines,

- **#pragma omp construct [clause] ...** as basic syntax, to explicitly instructs the compiler to parallelize the chosen block of code,
- **omp get num thread()** to return the number of threads currently working in a parallel region,
- **omp get threads num()** to get the thread rank within the team,
- **omp set num threads()** to set the number of threads used in a parallel region,
- etc.

To see other runtime library routines, reader is encourage to check the reference [24].

The flowchart of SWE-Exner implementation in serial and parallel part is shown at Figure 6. In the serial part, variables are declared and the current time is inspected. The current time inspection has to be executed in the serial part because the looping has to be increased step by step. Still in the serial part, the definition of z boundary condition is executed because it only needs two execution lines to define it, so parallel technique isn't needed. Last but not least, the time step is executed in serial because it only need computed once.

In OpenMP part, the new value of h_{new} , hu_{new} , and z_{new} are found by finding the solution of SWE and Exner. Because each iteration doesn't need the value of other iterations, so parallel technique is possible to be used. The boundary conditions of h_{new} and hu_{new} requires four lines, so parallel technique is used by attaching each line to available threads. Note that each processor

executes the parallel part by dividing the maximum spatial number and the number of processor in the architecture (*maximum_grid/processor*). After that, u is counted using parallel technique because each of its iteration doesn't need other iterations. It also applies in updating the value of sediment (z) and water (η) profile.

4. Results and Discussions

In this section two numerical simulations will be elaborated: (1) Comparison between numerical solution and analytical solution, and (2) Transcritical flow over a granular bump. The parallel part will be performed in second simulation, and then its speedup and efficiency of parallel method will be given.

4.1 Comparison between numerical and analytical solution

Here the goal is to model a numerical solution where the analytical solution is available, from the paper of [26]. The numerical solution is initialized with space domain $\Omega = [0,7]$, grass formula, and friction slope $S_f = 0$ as

$$(hu)_{ini}(x) = 1, \quad u_{ini}(x) = \left[\frac{\alpha x + \beta}{Ag} \right]^{1/3},$$

$$h_{ini}(x) = \frac{(hu)_{ini}(x)}{u_{ini}(x)},$$

$$Z_{ini}(x) = 1 - \frac{u_{ini}^3(x) + 2g(hu)_{ini}^2(x)}{2gu_{ini}(x)}.$$

Coefficients $\alpha = \beta = A_g = 0.005$ are used in this simulation. In detail, the analytical solution of this simulation can be found in [26].

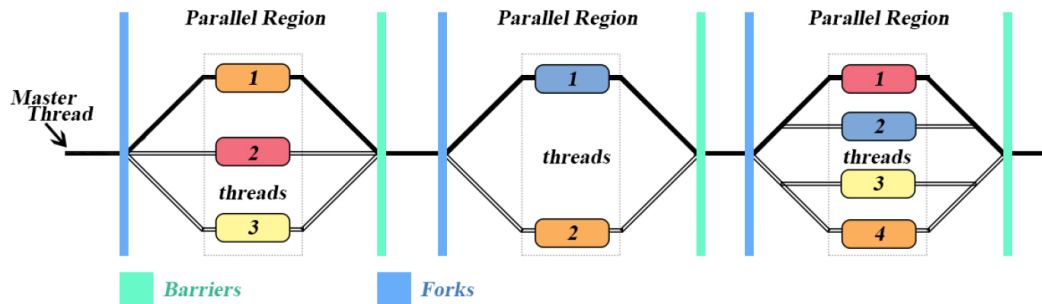


Figure 4. OpenMP work scheme in 3, 2, and 4 threads. Barrier and Forks will show which part is performed using parallel technique, and even if in parallel, there will be a master thread where it would run in thread 0 by default.

0	1	2	3	4	5	6	7	8	9	10	11
P1	P1	P1	P2	P2	P2	P3	P3	P3	P4	P4	P4

Figure 5. The illustration of array distribution using `#pragma omp parallel` for in OpenMP Parallel

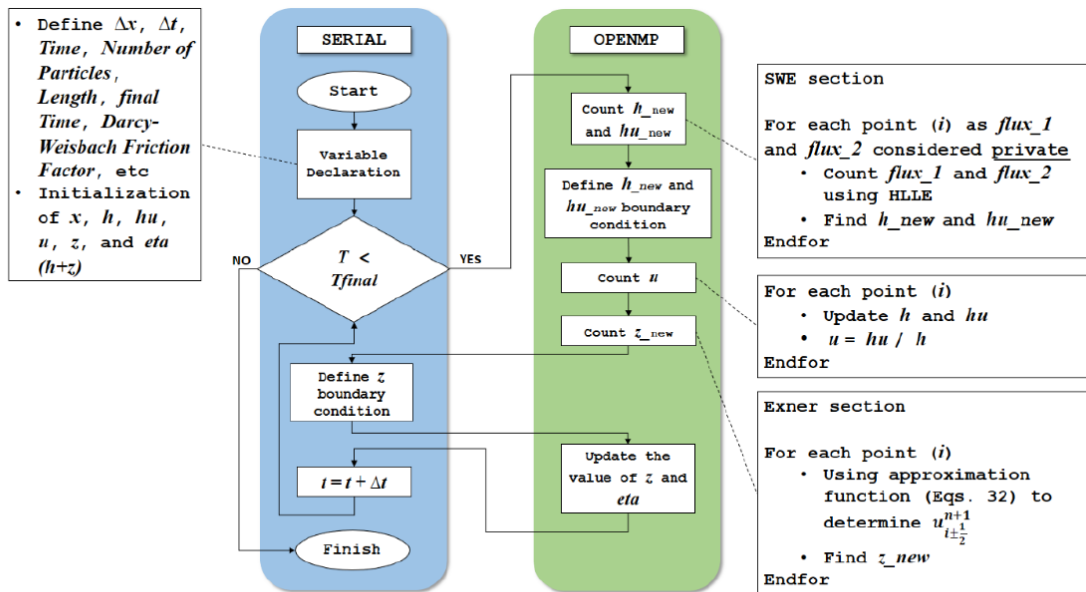


Figure 6. The Serial - OpenMP flowchart of SWE-Exner Implementation.

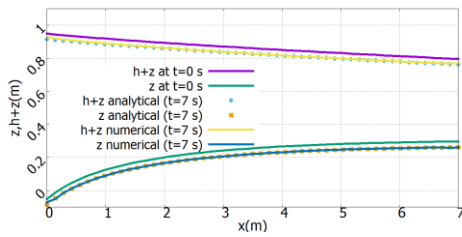


Figure 7. Analytical and numerical comparison in $t = 7s$ and their initial condition at $t = 0s$.

The result in 50 grid points shows that the numerical scheme has good agreements with the analytical solution (see Figure 7). In order to validate the distance between numerical and analytical solution, then the discrete L^p -norm error will be given (here $p = 1, 2, \infty$). Here, discrete L^1 -, L^2 -, and L^∞ -norm are a non-negative quantities that explain the magnitude of the object. Indeed, discrete L^p -norm error is used as the measurement for validating the numerical solution in numerical analysis [27]. Discrete L^p -norm error is defined as follows depend on the value of p ,

$$\|v\|_{L^p} = \left(\sum_{i=0}^{N_x} |v_i|^p \right)^{1/p} \quad (21)$$

$$\|v\|_{L^\infty} = \max_{i=0, \dots, N_x} |v_i|. \quad (22)$$

Tables 1, 2, and 3 show the results of discrete L^1 -, L^2 -, and L^∞ -norm error of simulation in 100, 200, 400, 800, and 1600 grid points. Be noted that h_a , u_a , and Z_a stand for analytical solution of h , u , and Z respectively. Moreover, the notations $\tilde{h} = h - h_a$, $\tilde{u} = u - u_a$ and $\tilde{Z} = Z - Z_a$ are defined the distance between approximation and analytical solution.

TABLE 1
 DISCRETE L^1 -norm ERROR OF THE SCHEME AT
 $t = 7s$

N_{grid}	$\ \tilde{h}\ _{L^1}$	$\ \tilde{u}\ _{L^1}$	$\ \tilde{Z}\ _{L^1}$
100	3.351E-2	7.058E-2	3.812E-3
200	2.938E-2	4.723E-2	2.985E-3
400	1.877E-2	2.255E-2	2.091E-3
800	1.0177E-2	1.0033E-2	1.475E-3
1600	5.872E-3	6.196E-3	1.257E-3

TABLE 2
 DISCRETE L^2 -norm ERROR OF THE SCHEME AT
 $t = 7s$

N_{grid}	$\ \tilde{h}\ _{L^2}$	$\ \tilde{u}\ _{L^2}$	$\ \tilde{Z}\ _{L^2}$
100	1.366E-2	2.934E-2	4.024E-3
200	1.152E-2	2.016E-2	2.886E-3
400	7.386E-3	1.019E-2	2.0141E-3
800	4.195E-3	4.584E-3	1.582E-3
1600	2.616E-3	2.983E-3	1.299E-3

TABLE 3
 DISCRETE L^∞ -norm ERROR OF THE SCHEME AT
 $t = 7s$

N_{grid}	$\ \tilde{h}\ _{L^\infty}$	$\ \tilde{u}\ _{L^\infty}$	$\ \tilde{Z}\ _{L^\infty}$
100	1.681E-2	1.529	1.236E-2
200	1.603E-2	1.037E-2	1.035E-2
400	1.575E-2	5.744E-3	1.259E-2
800	1.684E-2	3.930E-3	1.525E-2
1600	1.596E-2	4.868E-3	1.492E-2

Tables 1, 2 and 3 show the numerical error of h , u , and z by several grid points. By increasing the number of points, discrete L^1 -norm error for h , u , and z are shown decreasing significantly. Since the error is tend to zero, it means the numerical results is in a good agreement with the analytical solution and the numerical solution is acceptable. Here for instance, using 1600 points, the numerical error of sediment (z) is obtained 1.257×10^{-3} , 1.299×10^{-3} and 1.492×10^{-2} in discrete L^1 -, L^2 -, and L^∞ -norm error respectively.

4.2 Visualization Images

In this test case, the transcritical flow over a granular bump problem from [14] is elaborated. In [14], the staggered numerical scheme is used. Meanwhile here, numerical scheme (Eqs. (8-20)) is used. Considering an initial state of subcritical and supercritical flow over a granular bump with boundary conditions in space domain $\Omega = [0,10]$ are $(hu)(0, t) = 0.6$, $Z(0, t) = 0.1$, $Z_{ini}(x) = 0.1 + 0.1e^{-(x-5)^2}$, $(hu)_{ini}(x) = 0.6$, $h_{ini}(x) + Z_{ini}(x) = 0.4$, and $A_g = 0$. These conditions remain until the scheme reaches steady state (see Figure 8). It means that when a bump appear in the bottom of the flow, it would take some time to make the flow become appropriately formed because of the bump. Once it reached that state, then it is called as steady state. Changing A_g once it reaches steady state will make it more realistic because the effect of the bump should only happens after the condition becomes stable.

After that, the value of $A_g = 0.0005$ is applied to the scheme as the time starts. The movement of sediment can be seen in Figure 9. The image shows

that from the steady state to the flat state, both fluid flow and sediment has to reach certain time to make it to the flat state. Furthermore, the movement of both fluid and sediment (see Figure 9) tells the harmony of both of them.

Using parallel technique with shared memory architecture can be seen in the last paragraph of Section 1, it can be seen that the time used to compile the numerical computation is reduced for every grid points. Table 4 shows the time execution in serial and parallel technique. It shows that using parallel technique, the execution time of simulation can be reduced. For instance, using 3200 mesh points/grids, the execution time of serial and parallel are obtained 6118.536118 and 2015.589241 seconds respectively. In other words, parallel technique gives almost speedup 3 times from serial one.

In order to make it easier, graphics of speedup and efficiency could be seen in Figures 10 and 11 respectively. Figure 10 show that the more mesh points used in the model, the more it would increase the speedup. Moreover, similar in Figure 11, high efficiency is obtained along the increasing of mesh points. However, the progress of speedup is observed only growing fast until 800 mesh points. The formula of speedup and efficiency can be seen for examples in [7], [20], [21].

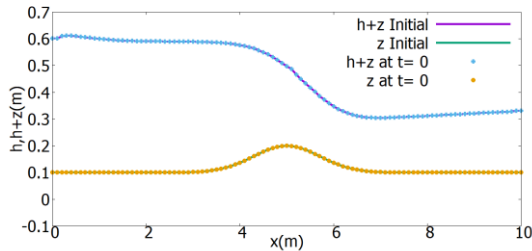


Figure 8. The initial condition of transcritical flow over a granular bump

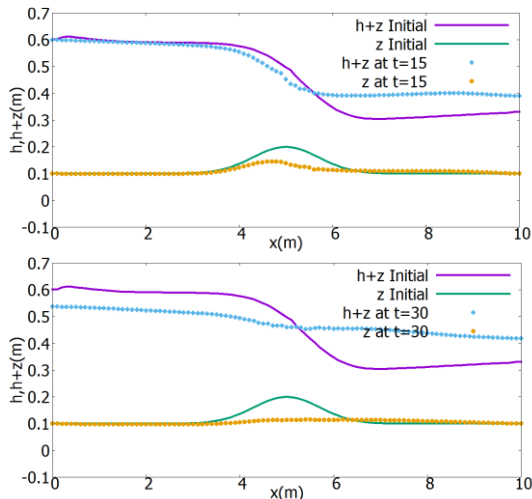


Figure 9. The solution of transcritical flow over a granular bump at: (a) $t = 15s$ and (b) $t = 30s$

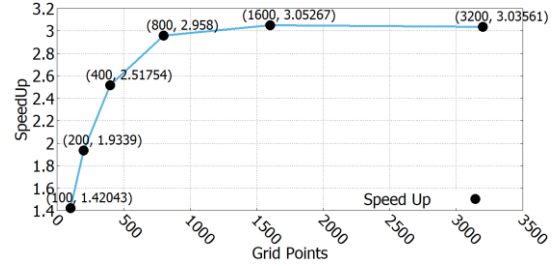


Figure 10. Time speedup of Transcritical flow over a granular bump numerical simulation

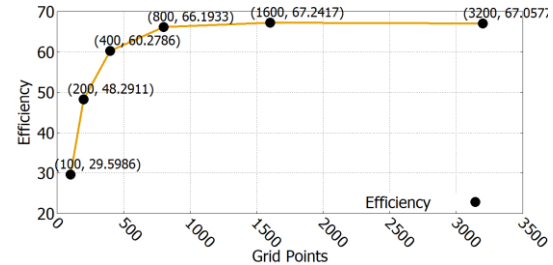


Figure 11. Time efficiency of Transcritical flow over a granular bump numerical simulation

TABLE 4
DETAILS OF EXECUTION TIME SIMULATION FOR EACH MESH POINTS IN SERIAL AND PARALLEL

Mesh Points	Execution Time (s)	
	Serial	Parallel
100	5.22702	3.679893
200	20.826099	10.768957
400	92.197323	36.622046
800	393.978355	133.190932
1600	1559.242544	510.780677
3200	6118.536447	2015.589241

Here, good speedup (3 times serial time) is observed if the grid number is greater than 800. This phenomena is natural behaviour in parallel computing, since the communication time of threads dominates execution time due to small amount of data. Therefore, here stable speedup profile is shown when the number of grids is increased. Similarly, in efficiency measurement of using parallel technique is shown stable when the number of grids is more than 800. Here in Figure 11, this simulation is observed reach up to 70% efficient by using parallel algorithm.

5. Conclusion

In this paper, numerical scheme of semi-implicit collocated FVM using average velocity for approximating SWEExnner model is given. Two numerical problems are successfully elaborated: (1) comparison scheme of analytical solution and numerical solution, and (2) parallel implementation for transcritical flow over the granular bump. In the first problem, it shows that the results of the proposed scheme in this paper are in a good agreement with the analytical solution.

The agreement is shown by the discrete L^p – norm errors (with p may be replaced with 1, 2, or ∞) which can be seen in Section 4.1. The second problem shows that in the term of algorithm, the efficiency of parallel implementation is gradually increasing until it reaches up to 70% by using grid number more than 800 points. Moreover, the average of parallel implementation's efficiency is obtained around 56.44%. Another parallel measurement, speedup of parallel computation is shown 3 times faster than using serial computation when the number of grids is more than 800. Therefore, through this speedup and efficiency, parallel technique is shown to be useful when less computation cost is needed. Moreover, parallel computing is shown able to decrease execution time in numerical simulation for approximating SWE-Exner model by semi-implicit collocated FVM.

References

- [1] M. C. D'iaz, E. D. Fern'andez-Nieto, and A. Ferreiro, "Sediment transport models in shallow water equations and numerical approach by high order finite volume methods," *Computers & Fluids*, vol. 37(3), pp. 299–316, 2008..
- [2] A. Utomo and P. H. Gunawan, "Implementasi skema collocated semi-implicit untuk swe-exner menggunakan fluks hlle," *Indonesian Journal on Computing (Indo-JC)*, vol. 3(1), pp. 85–96, 2018.
- [3] P. H. Gunawan, "Scientific parallel computing for 1d heat diffusion problem based on openmp," in *Information and Communication Technology (ICoICT)*, 2016 pp. 1–5.
- [4] M. 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(1). 2018.
- [5] O. OpenMP, "A proposed industry standard api for shared memory programming," *OpenMP Architecture Review Board*, vol. 27, 1997.
- [6] M. N. A. Alamsyah, C. A. Simanjuntak, B. A. R. H. Bagustara, W. A. Pradana, and P. H. Gunawan, "Openmp analysis for lid driven cavity simulation using lattice boltzmann method," in *Information and Communication Technology (ICoICT), 5th International Conference on. IEEE*, 2017 pp. 1–6.
- [7] S. Juliati and P. H. Gunawan, "Openmp architecture to simulate 2d water oscillation on paraboloid," in *Information and Communication Technology (ICoICT), 2017 5th International Conference on. IEEE*, 2017, pp. 1–5.
- [8] J. A. Liggett and D. A. Caughey, "Fluid mechanics: An interactive text." ASCE, 1998.
- [9] P. H. Gunawan and X. Lh'ebard, "Hydrostatic relaxation scheme for the 1d shallow water-exner equations in bedload transport," *Computers & Fluids*, vol. 121, pp. 44–50, 2015.
- [10] J. N. Newman and J. Grue, *Marine hydrodynamics*. MIT press, 2018.
- [11] Z. Cao, G. Pender, and J. Meng, "Explicit formulation of the shields diagram for incipient motion of sediment," *Journal of Hydraulic Engineering*, vol. 132(10), pp. 1097–1099, 2006.
- [12] S. Coleman and V. Nikora, "Exner equation: A continuum approximation of a discrete granular system," *Water Resources Research*, vol. 45(9), 2009.
- [13] A. Grass, *Sediment transport by waves and currents*. University College, London, Dept. of Civil Engineering, 1981.
- [14] P. H. Gunawan, R. Eymard, and S. Pudjaprasetya, "Staggered scheme for the exner–shallow water equations," *Computational Geosciences*, vol. 19(6), pp. 1197–1206, 2015.
- [15] F. Bouchut, *Nonlinear Stability of Finite Volume Methods for Hyperbolic Conservation Laws and Well-Balanced Schemes for Sources*. Birkhauser, 2000.
- [16] C. Berthon and F. Marche, "A positive preserving high order vfroe scheme for shallow water equations: a class of relaxation schemes," *SIAM Journal on Scientific Computing*, vol. 30, no. 5, pp. 2587–2612, 2008.
- [17] R. Aloor and V. K. Nandivada, "Unique worker model for openmp," in *Proceedings of the 29th ACM on International Conference on Supercomputing. ACM*, 2015, pp. 47–56.
- [18] T. Mattson, "A "hands-on" introduction to openmp," available at [https://www.openmp.org/wpcontent/uploads/Intro To OpenMP Mattson.pdf](https://www.openmp.org/wpcontent/uploads/Intro_To_OpenMP_Mattson.pdf) [Online; accessed 31-05-2018].
- [19] B. Barney, "Introduction to parallel computing. lawrence livermore national laboratory," Available at https://computing.llnl.gov/tutorials/parallel_comp, 2012.
- [20] Mulyani, N. D. Putri, and P. H. Gunawan, "The performance of openmp architecture for simulating fire spreading in forest area by cellular automata," in *Information and Communication Technology (ICoICT), 2017 5th International Conference on. IEEE*, 2017, pp. 1–5.

- [21] M. R. Pahlevi and P. H. Gunawan, "Parallel processing for simulating 2d radial dambreak using fvm hllc flux on openmp," in *Information and Communication Technology (ICoIC7), 2017 5th International Conference on. IEEE, 2017*, pp. 1–4.
- [22] Iryanto and P. H. Gunawan, "An openmp parallel godunov scheme for 1d two phase oil displacement problem," in *Information and Communication Technology (ICoIC7), 2017 5th International Conference on. IEEE, 2017*, pp. 1–5.
- [23] A. Basumallik, S.-J. Min, and R. Eigenmann, "Programming distributed memory systems using openmp," in *Parallel and Distributed Processing Symposium, 2007. IPDPS 2007. IEEE International. IEEE, 2007*, pp. 1–8.
- [24] R. Chandra, L. Dagum, D. Kohr, D. Maydan, R. Menon, and J. McDonald, *Parallel programming in OpenMP*. Morgan kaufmann, 2001.
- [25] A. Prabhakar, V. Getov, and B. Chapman, "Performance comparisons of basic openmp constructs," in *International Symposium on High Performance Computing. Springer, 2002*, pp. 413–424.
- [26] C. Berthon, S. Cordier, O. Delestre, and M. H. Le, "An analytical solution of the shallow water system coupled to the exner equation," *Comptes Rendus Mathematique*, vol. 350(3), pp. 183–186, 2012.
- [27] F. M. Zain, M. G. Khadafi, and P. Gunawan, "Analisis konvergensi metode beda hingga dalam menghampiri persamaan difusi," *E-Jurnal Matematika*, vol. 7, no. 1, pp. 1–4.