Computers & Fluids 104 (2014) 20–29

Contents lists available at ScienceDirect

Computers & Fluids

journal homepage: www.elsevier.com/locate/compfluid

Proposal for fast calculation of particle interactions in SPH simulations

Buntara Sthenly Gan [a,*], Dinh Kien Nguyen [b], AyLie Han [c], Sofia W. Alisjahbana [d]

[a] Department of Architecture, College of Engineering, Nihon University, Koriyama, Japan
[b] Department of Solid Mechanics, Institute of Mechanics, VAST, Hanoi, Viet Nam
[c] Engineering Faculty, Diponegoro University, Semarang, Indonesia
[d] Department of Civil Engineering, Universitas Bakrie, Jakarta, Indonesia

ARTICLE INFO

Article history:
Received 27 January 2014
Received in revised form 9 July 2014
Accepted 10 August 2014
Available online 22 August 2014

Keywords:
Smoothed Particle Hydrodynamic
SPH
Algorithm
Smoothing length
Particle interaction
Navier–Stokes equations
Fluid Simulation

ABSTRACT

A simple and easy algorithm is presented for a fast calculation of kernel functions which are required in fluid simulations using the Smoothed Particle Hydrodynamic (SPH) method. The present proposed algorithm improves the Linked-list algorithm and adopts the Pairwise interaction technique, both of which are widely used for evaluating kernel functions in fluid simulations using the SPH method. The proposed algorithm is easy to implement without any complexities in programming. Some benchmark examples are used to show the simulation time saved by using the proposed algorithm. Parametric studies on the number of divisions for sub-domains, ratios between the sub-domain size and smoothing length and total amount of particles are conducted to determine the range of applicability and effectiveness of the proposed technique. A handy formulation which relates the ratio between the sub-domain size and smoothing length and the total amount of particles used in the simulation using the SPH method is proposed for practical usage.

1. Introduction

The Smoothed Particle Hydrodynamic (SPH) method which was first introduced [1,2] for modeling astrophysical phenomena is one amongst many particle methods that has been used for simulating the physical behavior of fluid and continuum solid bodies. Recent progress in using the SPH method has been applied in the fields of fluid and solid interaction [3,4], multi-phase fluids and free surface flows [5]. In the SPH method, the so-called smoothing function or kernel function which is based on particle approximation plays a very important role in carrying out the integration of governing partial differential equations within the supporting domain.

One of the important issues for implementing the SPH method using the particle approximation is how to perform effectively the evaluation of kernel functions based on a set of particles scattered in an arbitrary manner. A lot of effort has been made to improve the governing equations used in the simulations and variation of kernel functions, as summarized in [6]; however, little research work has been done in enhancing the technique to carry out the interaction among particles in the supporting domain.

Since the SPH method was introduced, three well-known particles search algorithms have been widely used in the evaluation of kernel functions. They are: the all particles searching method; the Linked-list technique; and the Hierarchical Tree technique.

The simplest of these algorithms is the all particles searching method. The search is performed at a particle to find another particle inside its supporting domain within the entire simulation domain. The searching process is necessary at every time step, and thus the computation effort required for this all particles search method is very time-consuming, and not feasible for problems with very large amounts of particles.

The Linked-list technique was introduced in [7] before the SPH method was invented, and the technique is still being widely used to perform the SPH method. The Linked-list algorithm uses uniform meshes for particle bookkeeping with the size of κh where κh is the radius of the compact support domain of the kernel function. Thus, all particles in the neighboring sub-domains can then contribute to the properties of particles in the sub-domain. An improvement was made in [8], where a cylindrical sub-domain is used as a particle bookkeeping device to simulate shocks in accretion disks. However, the cylindrical domain will lose its capability to cover an arbitrary simulation domain which is not circular, in general problems. Unlike a rectangular sub-domain, the cylindrical sub-domain leaves the four corners of its bounding rectangle untouched, and therefore overlapping between cylindrical sub-domains for particle bookkeeping is required, which makes this technique less effective.

* Corresponding author.
E-mail address: buntara@arch.ce.nihon-u.ac.jp (B.S. Gan).

http://dx.doi.org/10.1016/j.compfluid.2014.08.004
0045-7930/

Hierarchical Tree coding has also been widely used [9–11]. However, due to the complexity in implementing the algorithm, this technique has not gained any popularity in practice. In [12], the application of Hierarchical Tree coding was used in parallel programming to boost its performance; nevertheless, the efforts add further complexities when adopting the algorithm proposed.

In this study, a simple and easy algorithm based on the creation of fixed sub-domains and their outlines is presented for a faster calculation of kernel evaluations, as is required in simulations using the SPH method. The proposed technique is based on the same concept as the Linked-list, but the sub-domain width does not need to be κh. In most simulation problems, the value of κh is very small compared to the whole simulation domain. Parametric studies conducted have shown that too small a sub-domain size could result in a considerable increase of the computation time. For easy use, the size of the sub-domain is determined by equally dividing the entire domain of simulation by an integer. The proposed technique is then further facilitated by the Pair-Wise Interaction method to register all the particles within the outlined sub-domain which contribute to the particles inside the sub-domain where the kernel functions are being evaluated.

Previous study [13] in internuclear distance calculations between atoms in cells inside molecular simulations also pointed out a growing computation time when the sub-divisions were increased. The so-called off-mapping algorithm was introduced to reduce unnecessary internuclear distance calculations for larger and complex systems. However, the study did not recommend the appropriate condition for applying the technique neither proposed any concrete solution for practical usage.

The outline of the paper is as follows. First, the SPH formulation for Navier–Stokes equation is highlighted. Second, the outlined sub-domain technique is explained. Third, verifications of simulation results were conducted for some benchmark problems. Fourth, parametric studies by changing the number of particles were done to observe the varying computation times. A fifth, a handy formulation was established for practical usage. Finally, the effectiveness of the formulation was used for are presented.

2. SPH formulation for Navier–Stokes equations

SPH can be considered as a kind of interpolation method for interactions of arbitrary particles in a support domain inside the fluid simulation system [14]. In the present study, the SPH method is used for solving Navier–Stokes equations problems. Fig. 1 shows a typical kernel function W. The kernel function used in this study was taken from the cubic spline family [15], which is known as a B-spline function, as given in Eq. (1).

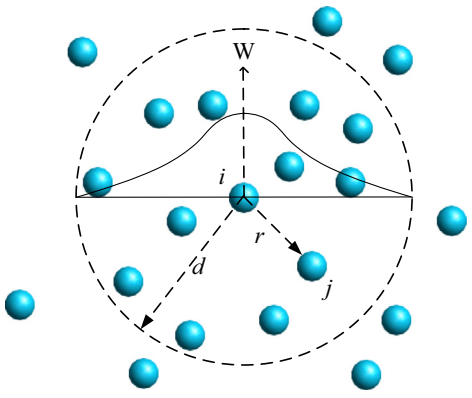

Fig. 1. Support domain of the kernel function W of particle i.

$$W(\xi,d)=\alpha_i\times\begin{cases}\frac{2}{3}-\xi^2+\frac{1}{2}\xi^3 & 0\leqslant\xi<1\\ \frac{1}{6}(2-\xi)^3 & 1\leqslant\xi<2\\ 0 & \xi\geqslant 2\end{cases}\tag{1}$$

where $\xi = 2r/d$, r is the distance between two nodes, d is the radius of the supporting domain which is given by $d = \kappa h$, κ is a multiplier factor, h is the smoothing length and $\alpha_1 = 2/d$, $\alpha_2 = 60/7\pi d^2$, $\alpha_3 = 12/\pi d^3$ for one-, two- and three-dimensional space respectively.

For the conservation of mass governing equation, the particle approximation of density can be expressed as

$$\frac{D\rho_i}{Dt}=\sum_{j=1}^{N}m_j v_{ij}^{\beta}\frac{\partial W_{ij}}{\partial x_i^{\beta}}\tag{2}$$

where ρ_i is the density of particle i, m_j is the mass of particle j, $v_{ij}^{\beta}=v_i^{\beta}-v_j^{\beta}$ is the relative velocity between particles i and j.

For the conservation of momentum and energy, the particle approximation of momentum and energy governing equations, taking into account artificial viscosity, are given as

$$\frac{Dv_i^{\alpha}}{Dt}=-\sum_{j=1}^{N}m_j\left(\frac{\sigma_i^{\alpha\beta}}{\rho_i^2}+\frac{\sigma_j^{\alpha\beta}}{\rho_j^2}+\prod_{ij}\right)\frac{\partial W_{ij}}{\partial x_i^{\beta}}\tag{3}$$

$$\frac{De_i}{Dt}=\frac{1}{2}\sum_{j=1}^{N}m_j\left(\frac{P_i}{\rho_i^2}+\frac{P_j}{\rho_j^2}+\prod_{ij}\right)v_{ij}^{\beta}\frac{\partial W_{ij}}{\partial x_i^{\beta}}+\frac{\mu_i}{2\rho_i}\varepsilon_i^{\alpha\beta}\varepsilon_i^{\alpha\beta}\tag{4}$$

where σ_i, σ_j are the stresses for particles i, j, P_i, P_j are the pressures at particles i, j, ε_i is the viscous strain rate for particle i, μ_i is the dynamic viscosity for particle i, and the artificial viscosity $\prod_{ij}$ [16,17].

There have been many variations of the governing equations, and these are summarized in [6]. Regardless of the governing equations being used in the SPH simulations, the present technique is generally applicable and effective for evaluating the kernel functions.

3. Outlined sub-domain technique

The proposed algorithm improves the Linked-list algorithm by allowing an arbitrary spacing for the fixed sub-domains and outlining the sub-domain with the κh width to guarantee that interacting particles are taken into account in the calculations.

The proposed technique uses the Pairwise interaction technique [11,18,19] as a particle bookkeeping device that is adopted by limiting the search for interacting particles within each sub-domain. The Pairwise interaction technique is carried out with the process of searching for the nearest neighboring particle and stores the necessary data for the SPH summation process. By dividing the whole of the simulation domain into equal size sub-domains, the time needed for storing particle data becomes longer, but the time required for searching for the nearest neighboring particle is reduced, which results in a considerable total time saving.

To illustrate the present outlined sub-domain technique, a schematic two-dimensional arbitrary domain as shown in Fig. 2 is adopted. The whole simulation domain is first divided into smaller rectangles and outlines the sub-domain of the rectangles. Each rectangle is then used for registering all the particles inside the area after each time step calculation. At the same time, an outer rectangle outlining the inner rectangle is also used for particle registration. The width of the outer rectangle is determined from the size of the inner rectangle increased by the radius d of the support domain of kernel function W in both its width and height. There are overlapping areas in the adjacent sub-domains due to this outline sub-domain technique, hence coverage of the compact support of the kernel function from a particle at the edge of the inner sub-domain is guaranteed.

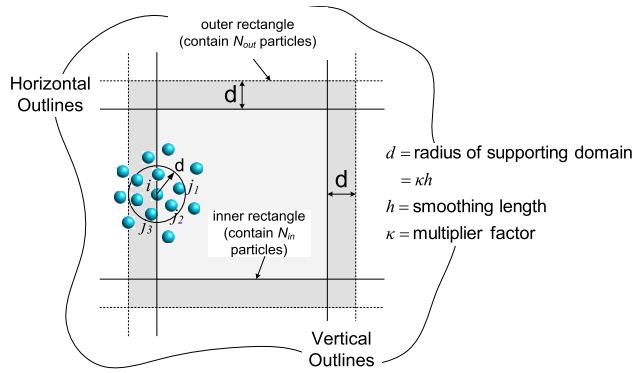

Fig. 2. A 2D schematic representation of the outlined sub-domain technique.

In evaluating Eqs. (2)–(4), the summations of governing equations are accounted for each of N_{in} particles which were registered in the inner rectangle using the index *i*. The nearest particle searching for a particle *i* is sought among N_{out} particles with index *j* which are registered within the outer rectangle domain.

Here, each particle inside the inner rectangle that is *i* indexed acts as a center for evaluating the governing equations to interact with the surrounding particles within the outer rectangle that are *j* indexed. Hence, searching for interacting particles will only be required inside the outer rectangle, which requires less time than searching for a particle in the whole domain of simulation. The reduction of computation time in the present technique depends on the number of sub-domains, particles, and smoothing length.

The additional time required in the present technique is for registering all the particles inside the predefined sub-domains and their outlined area to account for all particles at the utmost boundary inside the sub-domains for calculation of kernel functions. Registration of particles into their sub-domains and the outlined area is conducted after each time step calculation is finished. The process of registration is repeated at all sub-domains within the simulation domain.

The following algorithm applies for the proposed outlined sub-domain technique.

- *Start CPU timing.*
- *Divide the whole simulation domain into equal divisions of sub-domains in all dimensional direction(s).*
- *Start the simulation LOOP.*
- *Create link lists for each sub-domain:*
  - *Registration 1: N_{in} particles inside a sub-domain.*
  - *Registration 2: N_{out} particles inside the outlined sub-domain.*
  - *Create a link list for the N_{in} particles to search from the N_{out} particles inside the support domain of kernel function W.*
- *Calculate the SPH formulation for the whole simulation domain using the link lists created.*
- *Calculate the new step time value for next simulation.*
- *If not at the end of the loop, repeat the LOOP.*
- *Stop CPU timing.*

As a final point, the proposed technique will considerably improve the efficiency of the Pairwise interaction method by limiting the sub-domain searching which is defined only by the overlapping of outline widths. The advantage of this method is that the proposed number of divisions for sub-domains can significantly reduce the required time evaluation for the same simulation to the complexity of work in order $O(\alpha N)$. The value of α varies from one tenth to a thousandth of fraction of the required simulation time without division.

4. Verification of simulation results

Before conducting a parametric study, several benchmark problems are chosen to determine the applicability of the proposed technique. As references for verifying the results of simulations from the proposed technique, several benchmark problems were selected from [6].

4.1. Shock tube 1D problem

The shock tube problem is a good one-dimensional benchmark which has been frequently used by many researchers using the SPH method [11,20]. The shock tube is a long straight tube filled with gas, which is separated by a membrane into two equal parts each of which is initially in an equilibrium state of constant pressure, density and temperature. When the membrane is taken away suddenly, a shock wave, a rarefaction wave and a contact discontinuity will be produced.

The dimensionless initial conditions of the simulation are similar to [11], then introduced by [20] which were taken from [21], for $x \leqslant 0$ ($\rho = 1$; $v = 0$; $e = 2.5$; $p = 1$; $\Delta x = 0.001875$) and for $x > 0$ ($\rho = 0.25$; $v = 0$; $e = 1.795$; $p = 0.1795$; $\Delta x = 0.0075$). Here, ρ, p, e and v are the density, pressure, internal energy and velocity of the gas, respectively. Δx is the space between two particles. A constant time step of 0.00015 is used for a 1000-step calculation. A constant smoothing length $h = 0.015$ and multiplier factor $\kappa = 2$ are used in this simulation. Fig. 3 shows the scheme of the outlined sub-domain technique applied to the shock tube problem.

In Fig. 3, M is the number of divisions, L is the sub-domains length, d is the outlined width, and ℓ is the entire length of the simulation domain.

There were 400 particles used in the simulation. 320 particles are evenly distributed in the high-density region and 80 particles are evenly distributed in the low-density region. Figs. 4–7 show the density, pressure, velocity and internal energy distribution along the *x*-axis, respectively. The square box markers are the reference solutions obtained from the execution of codes provided in [6] using the Pairwise interaction method for particle interactions. The circle markers are the solution obtained after the augmentation of the present outlined technique using 4 equal divisions along the *x*-axis.

From the results, it can be concluded that the results by using the proposed technique fit exactly with the reference solutions without any discrepancies found. The purpose of this verification is to show that the augmentation of the present outlined technique does not influence the results obtained.

4.2. Shear driven cavity 2D problem

The classical shear driven cavity problem is the fluid flow within a closed square rectangle generated by moving the top side of the rectangle at a constant velocity while the other sides remain fixed. The flow will reach a steady state and form a recirculation pattern. In the simulation, the dimensions of the kinetic viscosity

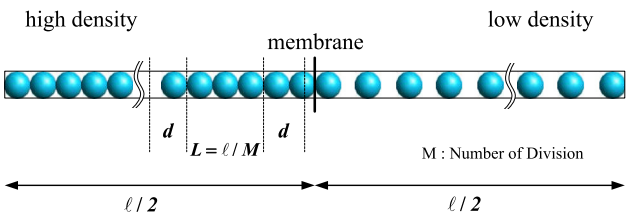

Fig. 3. Outlined sub-domain scheme in the shock tube problem.

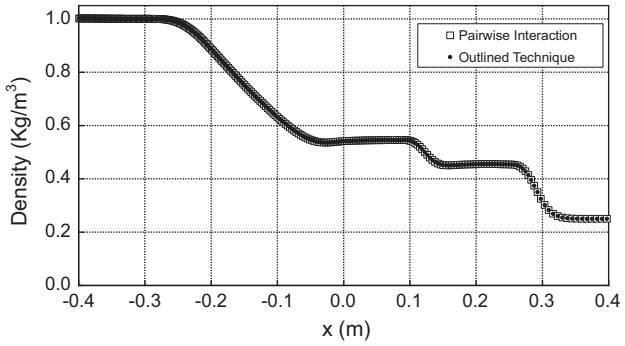

Fig. 4. Density distribution in the shock tube at $t = 0.20$ s.

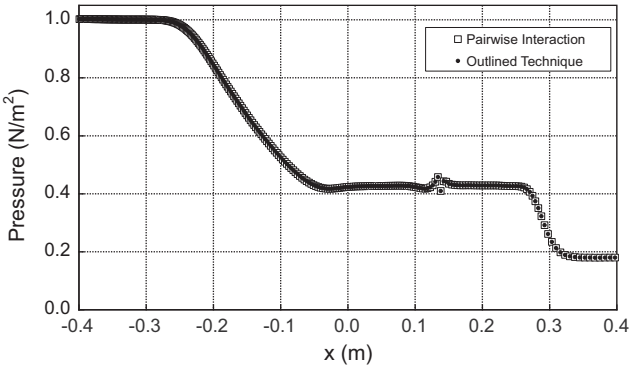

Fig. 5. Pressure distribution in the shock tube at $t = 0.20$ s.

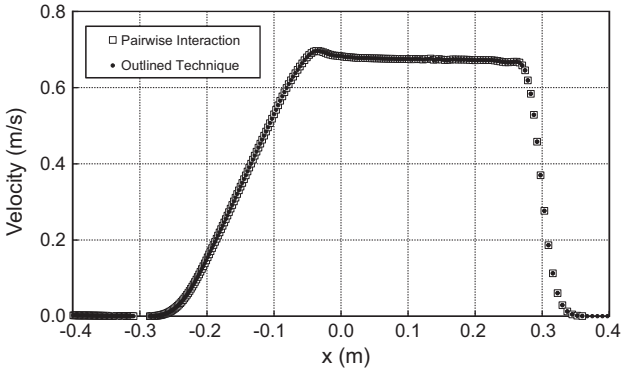

Fig. 6. Velocity distribution in the shock tube at $t = 0.20$ s.

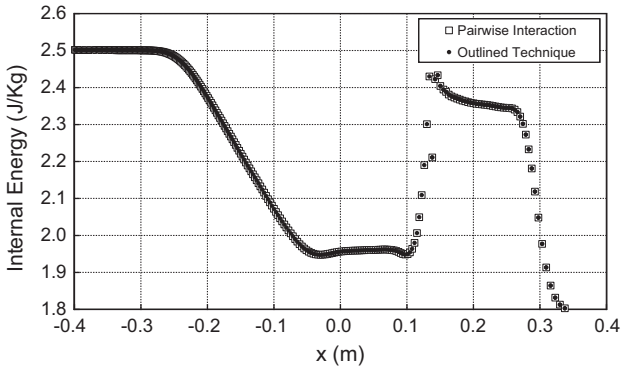

Fig. 7. Internal energy distribution in the shock tube at $t = 0.20$ s.

and density are $v = 10^{-6}$ m^2/s and $\rho = 10^3$ kg/m^3 respectively. The top side of the rectangle moves at a velocity of $V = 10^{-3}$ m/s, thus the Reynolds number for this problem is one. A constant time step of 5×10^{-5} s is used. A constant smoothing length of 2.5×10^{-5} is used. Fig. 8 shows the scheme of the outlined sub-domain technique applied to the shear driven cavity problem.

There were 1600 particles filling a square rectangle used in the simulation. Figs. 9 and 10 show the vertical distribution velocities along the horizontal centerline of the rectangle and the horizontal

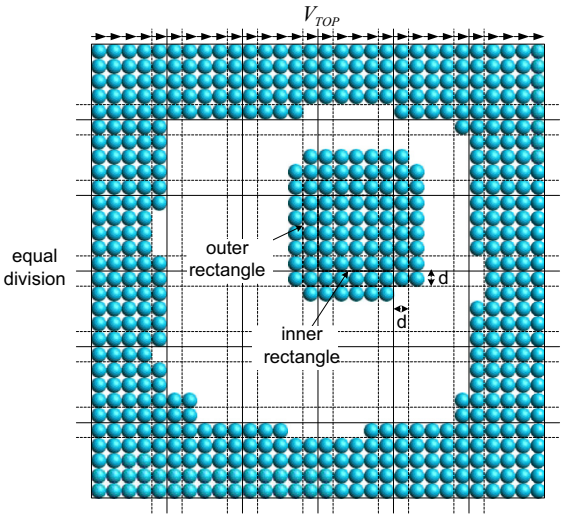

Fig. 8. Outlined sub-domain scheme in the 2D shear driven cavity problem.

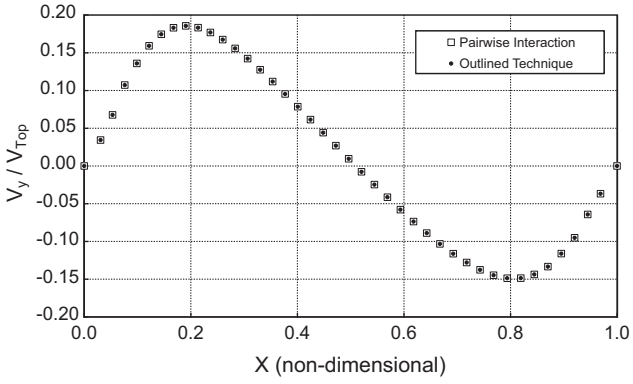

Fig. 9. Vertical velocities distribution along the horizontal centerline.

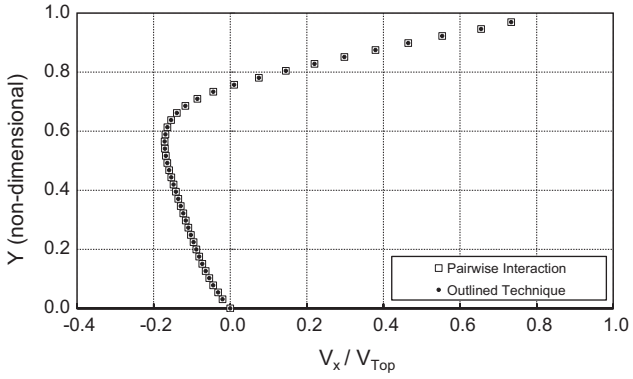

Fig. 10. Horizontal velocities distribution along the vertical centerline.

velocities distribution along the vertical centerline of the rectangle, respectively. The square box markers are the reference solutions obtained from the works of [6] using the Pairwise interaction method for particle interactions. The circle markers are the solution obtained after the augmentation of the present outlined technique by using 4 equal divisions along both the *x*-axis and *y*-axis directions. Both results are taken after 3000 steps at $t = 0.15$ s.

From the results, it can be concluded that the results by using the proposed technique fit exactly with the reference solutions without any discrepancies found. The purpose of this verification is to show that the augmentation of the present outlined technique does not influence the results obtained.

4.3. Shear driven cavity 3D problem

Since there is no benchmark available for a 3D problem, the shear driven cavity 2D problem is extended to a 3D closed cube problem by moving the top side of the cube at a constant diagonally 45 degree velocity generated while the other sides remain fixed. By using the same parameters as given in the 2D problem, the flow will reach a steady state and form a recirculation pattern. In the simulation, the dimensions of the kinetic viscosity and density are $v = 10^{-6}$ m^2/s and $\rho = 10^3$ kg/m^3 respectively. The top side of the cube moves at velocities of $V_{0x} = 10^{-3}$ m/s and $V_{0z} = 10^{-3}$ m/s in the *x* and *z* directions, respectively. A constant time step of 5×10^{-5} s is used. A constant smoothing length of 2.5×10^{-5} is used.

There were 64,000 particles filling a cube box used in the simulation. Figs. 12–14 show the orthogonal distribution velocities

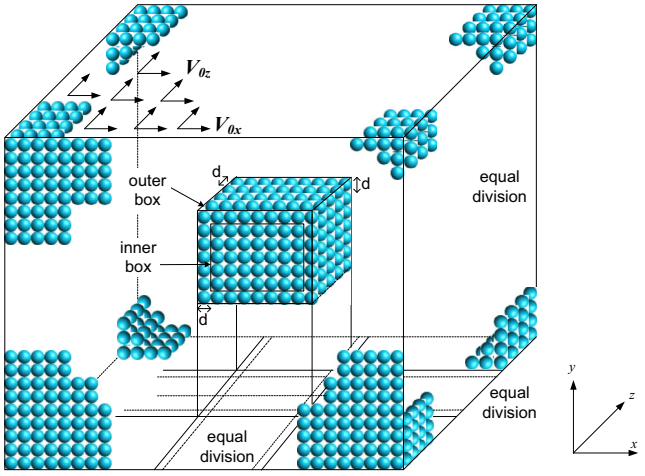

Fig. 11. Outlined sub-domain scheme in the 3D shear driven cavity problem.

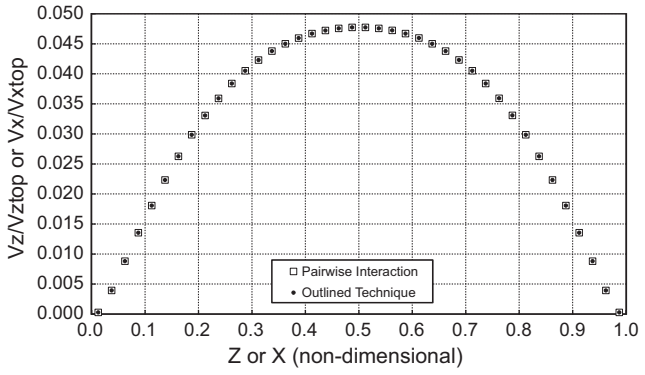

Fig. 12. Vz/Vztop or Vx/Vxtop distribution along the *Z* or *X* centerline direction.

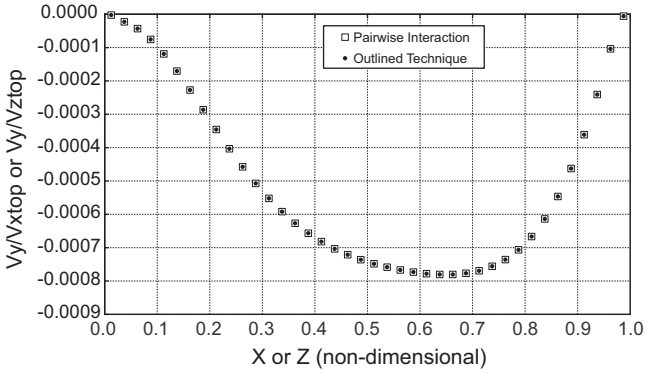

Fig. 13. Vy/Vxtop or Vy/Vztop distribution along the *X* or *Z* centerline direction.

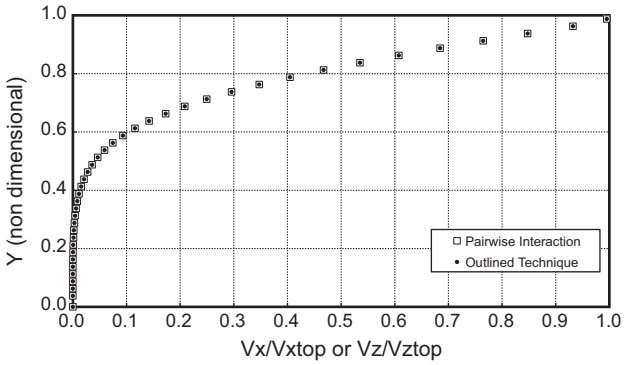

Fig. 14. Vx/Vxtop or Vz/Vztop distribution along the *Y* centerline direction.

along each axis centerline of the cube. By using the source code provided in [6], the results of the simulations are compared with the proposed technique. Both results are taken after 20 steps at $t = 0.001$ s.

From the results, it can be concluded that the results by using the proposed technique fit exactly with the reference solutions without any discrepancies found. The purpose of this verification is to show that the augmentation of the present outlined technique does not influence the results obtained.

5. Parametric study using benchmarks

In the following examples, the timing evaluations were performed on a personal computer with Core2Duo E7500 CPU, clock rate of 2.93 Hz, FSB speeds of 1066 MHz and 4 GB memory of RAM. A Fortran compiler was used to edit, modify, compile, debug and run the source codes available from [6].

The execution time ratio is computed as

$$\text{Execution time ratio} = \frac{t_{ND}}{t_1} \tag{5}$$

where t_{ND} is the required CPU timing evaluation obtained from dividing the whole simulation domain into *ND* divisions; t_1 is the required CPU timing evaluation without dividing the whole simulation domain, and *ND* is the number of equal divisions.

5.1. Shock tube 1D problem

The shock tube problem previously used for verifying the results of the proposed technique is then adopted for the following parametric study. The gas properties and parameters used in the simulations for the parametric study are the same as the previous

data. In this parametric study, the length of the shock tube is fixed, but the number of particles used in the simulation is varied by adjusting the particle mass and smoothing length.

Table 1 shows three different numbers of particles used in the study. The parametric study is conducted with the same initial percentage distribution of particles in two equal parts of the tube, that is, 80% and 20% for the high-density and low-density regions. To maintain a constant density in the tube, the length of the tube is increased along with the increasing number of particles.

For each case of the three shock tubes shown in Table 1, the total length of the tube is divided by vertical sub-domains which result in 2, 4, 8, 16, 32, 64, 128 and 256 divisions.

Fig. 15 depicts the results of timing evaluation ratios by using the proposed technique for the different total numbers of particles in the simulation and an equal number of divisions of the vertical sub-domains.

Fig. 15 shows the effectiveness of the proposed technique in giving better results for larger numbers of particles used in the simulation. Also shown in Fig. 15, for the total number of 100 particles in the simulation case, the time for registering the particles became higher than for evaluating the kernel functions, resulting in reduced effectiveness if the number of particles used in the simulation is few. However, the present technique shows good results for the larger numbers of particles used; for 10,000 particles the computation time can be reduced to less than 10% when the number of divisions of sub-domains is from 64 to 128.

5.2. Shear driven cavity 2D problem

The shear driven cavity problem previously used for verifying the results of the proposed technique is then adopted for the following parametric study. The fluid properties and parameters used in the simulations for the parametric study are the same as the previous data. In this parametric study, the size of the rectangle is fixed, but the number of particles used in the simulation is varied by adjusting the particle mass and smoothing length.

Table 2 shows four different numbers of particles used in the simulation to show the effectiveness of the proposed technique.

Table 1
Parametric study for shock tube 1D problem.

Case	Number of particles		Total length
	High density	Low density	
I	80	20	0.3
II	800	200	3.0
II	8000	2000	30.0

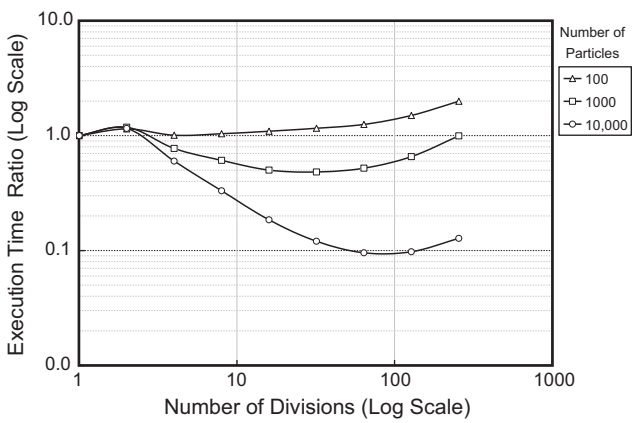

Fig. 15. Results of execution time for 1D shock tube problem.

Table 2
Parametric study for shear driven cavity 2D problem.

Case	Number of particles inside the square	Side length	Computation steps
I	50×50	0.00125	1000
II	100×100	0.00250	1000
II	200×200	0.00500	1000
IV	1000×1000	0.02500	10

For each case of the four shear driven cavity problems shown in Table 2, each side of the simulation domain is divided by vertical and horizontal sub-domains of 2, 4, 8, 16, 32, 64 and 128 equal divisions.

A similar tendency to the results from the shock tube problem is seen, and the effectiveness of the present proposed outlined sub-domain technique gives better results for larger numbers of particles used in the simulation. As shown in Fig. 11, for a lesser number of particles, the time for registering the particles became higher than for evaluating the kernel functions, resulting in reduced effectiveness. The present technique shows even better results for larger numbers of particles used; as shown for 1000×1000 particles, a computation time of less than 1% can be achieved when around 32 divisions of the sub-domains are used. However, further increase of the number of divisions will result in a longer computation time.

5.3. Shear driven cavity 3D problem

The shear driven cavity problem previously used for verifying the results of proposed technique is then adopted for the following parametric study. The fluid properties and parameters used in the simulations for the parametric study are the same as the previous data. In this parametric study, the size of the cube is fixed, but the number of particles used in the simulation is varied by adjusting the particle mass and smoothing length.

Table 3 shows eight different numbers of particles used in the simulation to show the effectiveness of the proposed technique.

For each case of the eight shear driven cavity problems shown in Table 3, each side of the simulation domain is divided by vertical and horizontal sub-domains of 2, 4, 8 and 16 equal divisions.

A similar tendency can be observed from the results, where the effectiveness of the proposed outlined sub-domain technique gives better results for larger total numbers of particles used in the simulation. As shown in Fig. 17, for smaller total numbers of particles, the time for registering particles became higher than for evaluating the kernel functions, resulting in reduced effectiveness. The present technique shows even better results for larger numbers of particles used. As shown for $60 \times 60 \times 60$ particles, reduction of the computation time to less than 3% can be achieved when around 8 divisions of the sub-domain are used. However, further increase of the number of divisions will result in longer computation time.

Table 3
Parametric study for shear driven cavity 3D problem.

Case	Number of particles inside the cube	Side length	Computation steps
I	$10 \times 10 \times 10$	0.00100	1000
II	$15 \times 15 \times 15$	0.00100	1000
II	$20 \times 20 \times 20$	0.00100	1000
IV	$25 \times 25 \times 25$	0.00100	1000
V	$30 \times 30 \times 30$	0.00100	1000
VI	$40 \times 40 \times 40$	0.00100	1000
VII	$50 \times 50 \times 50$	0.00100	1000
VIII	$60 \times 60 \times 60$	0.00100	1000

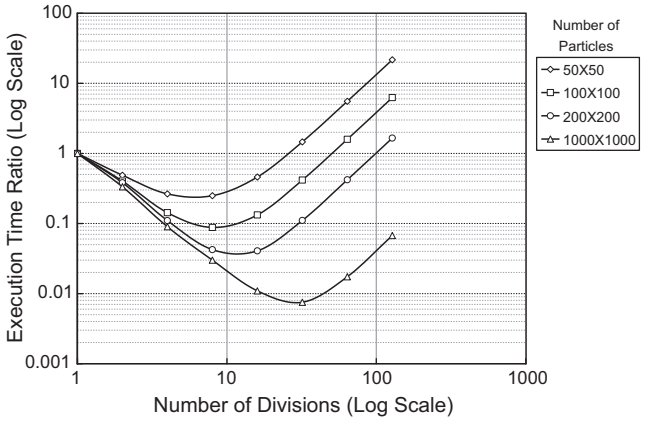

Fig. 16. Results of execution time for 2D shear driven cavity problem.

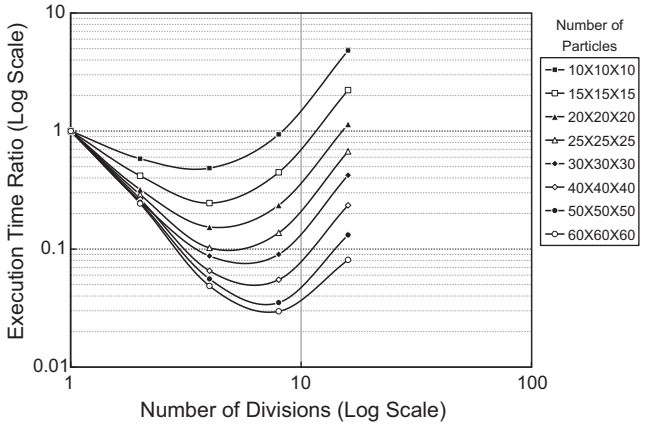

Fig. 17. Results of execution time for 3D shear driven cavity problem.

Table 4
The most reduced execution time from parametric studies.

Case		Number of particles	L/h ratio	ND
Shock tube 1D	II	1000	9.563	1000
	III	10,000	31.313	10,000
Shear driven cavity 2D	I	50×50	9.375	50
	II	100×100	12.500	100
	III	200×200	18.750	200
	IV	1000×1000	31.250	1000
Shear driven cavity 3D	I	$10 \times 10 \times 10$	3.125	10
	II	$15 \times 15 \times 15$	3.750	15
	III	$20 \times 20 \times 20$	5.000	20
	IV	$25 \times 25 \times 25$	5.469	25
	V	$30 \times 30 \times 30$	5.682	30
	VI	$40 \times 40 \times 40$	6.250	40
	VII	$50 \times 50 \times 50$	7.031	50
	VIII	$60 \times 60 \times 60$	7.485	60

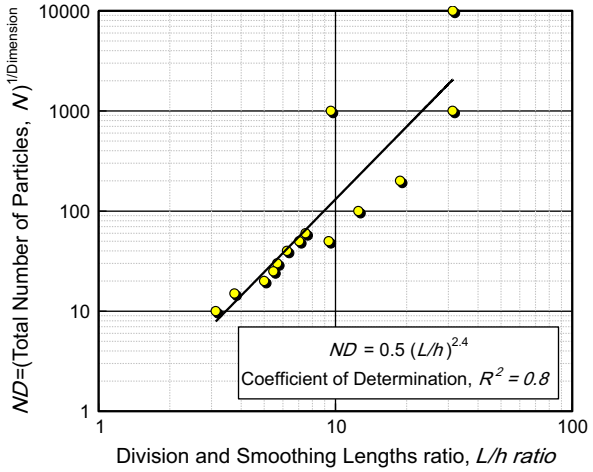

Fig. 18. Results of parametric studies with the greatest reduction time ratio.

6. Proposal for practice

The effectiveness of the proposed technique in reducing the execution time was found to depend on the dimensional type of the problem, the number of particles in the simulations and the number of divisions used in the simulations, as shown in Figs. 15–17 from the parametric studies. For practical purposes, a simple formulation that relates all the parameters used in the simulation is to be established. A unique relationship between the number of particles, the radius of the smoothing length and the number of divisions is to be sought. From the results of the parametric studies depicted in Figs. 15–17, the cases which are ineffective when using the proposed technique are omitted. From each case, the number of divisions where the execution time ratios give the lowest value are collected and summarized in Table 4.

In Table 4, L is the size of the sub-domain; h is the smoothing length; N is the total number of particles used in the simulation. Here, ND is defined as $ND = N^{1\backslash D}$, where D is the dimension of the simulations type.

Fig. 18 shows a logarithmic relationship between the L/h *ratio* and ND. By applying linear regression analysis to the data plotted, a unique relationship can be obtained as follows:

$$ND = 0.5(L/h)^{2.4} \tag{6}$$

The goodness of fit is given by the coefficient of determination $R^2 = 0.8$. However, for real application, the total number of particles which are used in the simulation, N, is considered more practical, thus Eq. (6) can be expressed as follows:

$$N = ND^D = 0.5^D(L/h)^{2.4D} \tag{7}$$

Depending on the problem type and the number of particles used in the simulations, generally a reduction of execution time varying from one-hundredth to one-tenth ratio can be achieved by using the proposed technique. Unlike the other techniques proposed in the past, Eq. (7) is very simple and easy to use, and is thus recommended for practical simulation purposes.

7. Efficiency of the proposed method

The proposed method's advantage is demonstrated by comparing a system of N particles. For N particles system, the traditional All-search method requires a complexity of work in the order of $O(N^2)$ [6] to calculate the particle interaction of each particle with every other particle. The Tree-search method requires less complexity of work i.e. in the order of $O(N \log N)$ [6]. The Linked-list method requires only a complexity of work in the order of $O(N)$ [6], while on the other hand the proposed method can reduce the complexity of work in the order of $O(0.008N \sim 0.1N)$ depending on the classification of the problem. The coefficient values of 0.008∼0.1 are the most reduced execution time ratio obtained from the results of parametric study using benchmarks in Section 5.

Fig. 19 illustrates the comparison of the required amount of work for all the aforementioned particle searching methods with the proposed method. In common to all particle searching methods, the amount of work are efficient especially for larger number

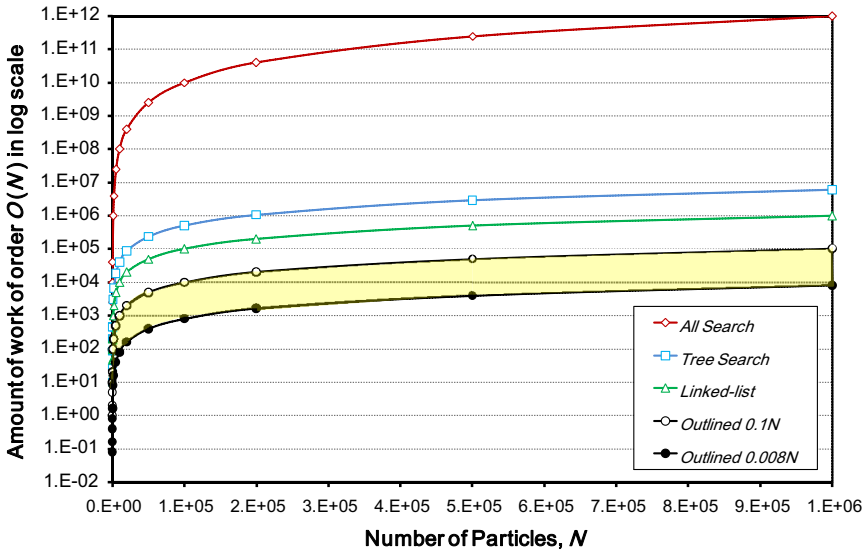

Fig. 19. Comparison of work of order between methods.

of particles used in the simulation. It can be seen that the proposed outline sub-domain technique shows the highest efficiency degree as compared to all other particle searching methods. The range of applicability of the proposed method is shown in the yellow filled shaded between both curves.

8. Applications

In order to verify the effectiveness of the proposed technique, 1D, 2D and 3D type fluid simulation problems are presented. The proposed formula in Eq. (7) was used for all the problems below.

8.1. Modified shock tube 1D problem

The original shock tube problem which was used as the benchmark problem is modified by dividing the long straight tube filled with gas into four equal parts along its length. The tube is separated by membranes at three locations which divide the tube into four equal parts. All parts of the tube are initially in an equilibrium state of assumed constant pressure, density and temperature. When all the membranes are taken away instantaneously, shock waves, rarefaction waves and contact discontinuities will be produced. Fig. 20 shows the scheme of the outlined sub-domain technique applied to the modified shock tube problem.

Considering that the flow of gas inside the tube is adiabatic in smooth regions, the functional entropy can be set as a constant, and thus the relationships between density and pressure follow the isentropic law. The initial conditions of the simulation are then given: for $x \leqslant 0.15$ ($\rho = 1.0$; $v = 0$; $e = 2.5$; $p = 1.0$; $\Delta x = 0.0000375$); for $0.15 < x \leqslant 0.30$ ($\rho = 0.5$; $v = 0$; $e = 1.895$; $p = 0.379$; $\Delta x = 0.000075$); for $0.30 < x \leqslant 0.45$ ($\rho = 0.8$; $v = 0$; $e = 2.287$; $p = 0.732$; $\Delta x = 0.000046875$) and for $x > 0.45$ ($\rho = 0.3$; $v = 0$; $e = 1.544$; $p = 0.185$; $\Delta x = 0.000125$). Here, ρ, p, e and v are the density, pressure, internal energy and initial velocity of the gas, respectively. Δx is the space between two particles. A constant time step of 5×10^{-5} s is used for a 1000-step calculation.

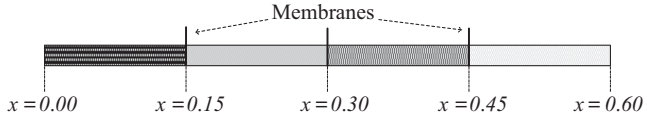

Fig. 20. Modified 1D shock tube problem.

In total 10,400 particles were used in the simulation. The radius of the supporting domain used in the simulation was determined from two times the smoothing length, $d = 0.00025$, which is two times the largest distance between particles in the lowest density region of $\Delta x = 0.000125$, to ensure that all the particles are within the support domain of the kernel function. From $d = 0.00025$, the smoothing length $h = 0.000125$ can be computed using the multiplier factor $\kappa = 2$. By using Eq. (7), where $D = 1$ for the 1D type problem, the division length of $L = 0.0075$ can be calculated, which gives the rounded number of 80 equal divisions along the tube length.

The CPU time required for conducting the 1000-step simulation was about 86.2 s, which is only 7.1% of the time required when the proposed technique was not used, namely 1210.8 s. From the comparison of both execution CPU times, the proposed technique shows its effectiveness in reducing the time required for conducting the modified 1D Shock Tube simulation.

Fig. 21 shows the density distribution along the x-axis at a 2400-step calculation after all the membranes that separated density varied regions were removed instantaneously at the same time.

8.2. Modified shear driven cavity 2D problem

The shear driven cavity 2D problem previously used as the benchmark problem is modified by cutting off the four corners of the rectangle domain and by removing the center area. The fluid flow within a closed area is generated by moving the inner top and bottom sides inside the center area at a constant velocity in reverse horizontal directions. In the simulation, the kinetic viscosity and density are $v = 10^{-6}$ m^2/s and $\rho = 10^3$ kg/m^3, respectively. The inner top and bottom sides of the center area move at a constant velocity of $V = 10^{-3}$ m/s. A constant time step of 5×10^{-5} s is used. A constant smoothing length of 2.5×10^{-5} m is used. Fig. 22 shows the holed corner-cut shear driven cavity problem.

In total 6000 particles were used in the simulation. From the defined smoothing length $h = 2.5 \times 10^{-5}$ m, the division length $L = 0.0002$ m can be obtained from Eq. (7), where $D = 2$ for the 2D type problem. The largest dimension of the simulation domain is then divided by L and gives the rounded number of 12 equal divisions in the horizontal and vertical directions.

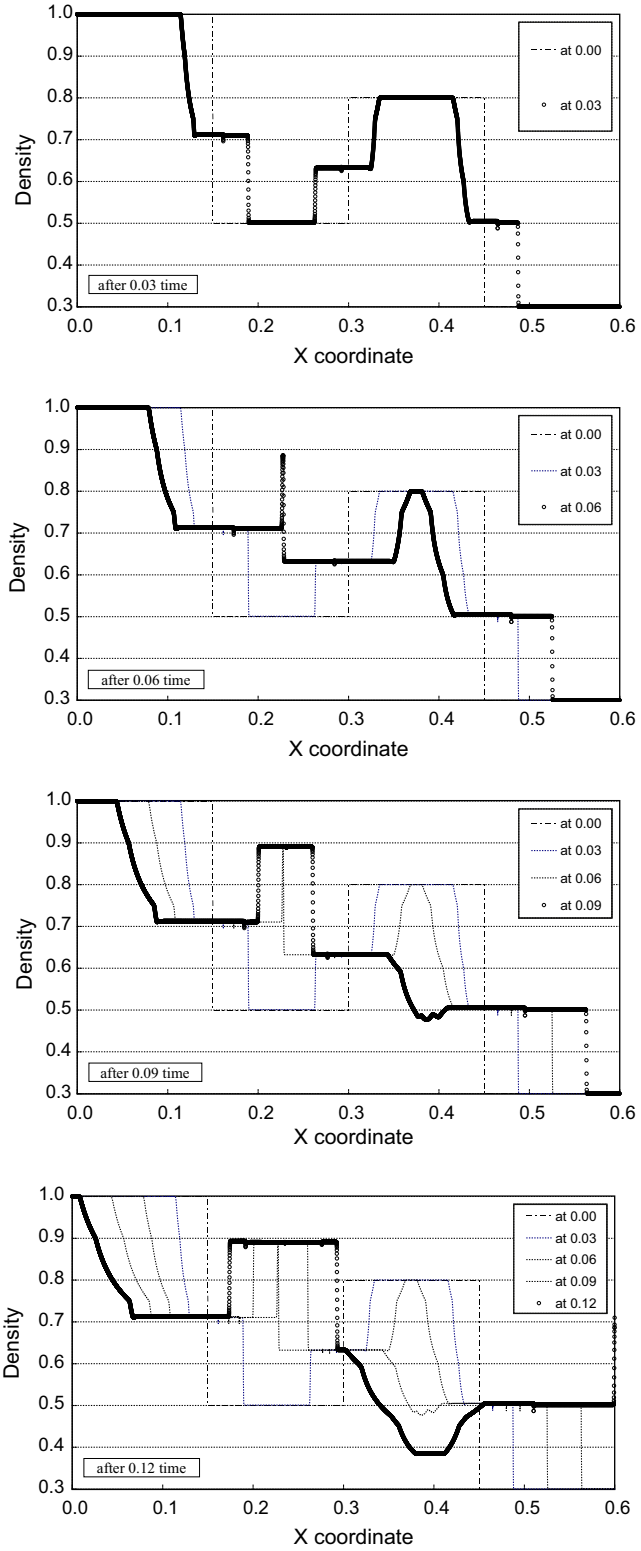

Fig. 21. Density distributions along the tube from the modified 1D shock tube simulation results at 0.03; 0.06; 0.09 and 0.12 s, respectively.

The CPU time required for conducting the 6000-step simulation was about 932 s, which is about 17% of the time required when the proposed technique was not used, namely 5492 s. From the comparison of both execution CPU times, the proposed technique shows its effectiveness in reducing the time required for conducting the modified 2D Shear Driven Cavity simulation.

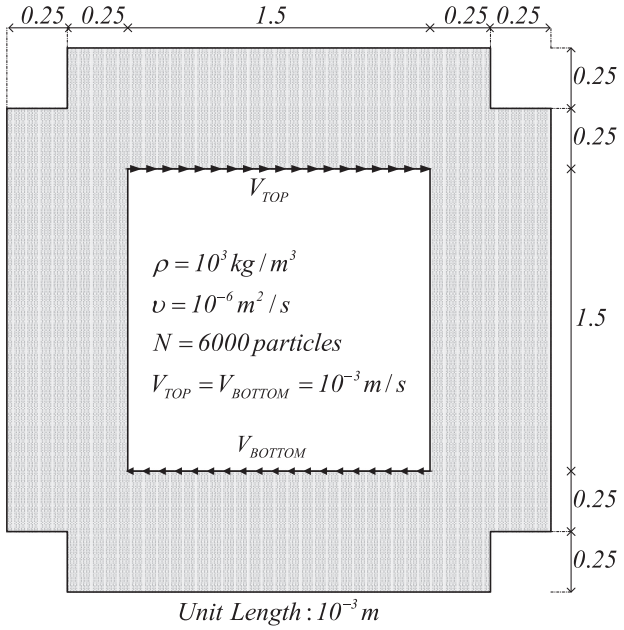

Fig. 22. Holed corner-cut 2D shear driven cavity problem.

Fig. 23 shows the velocity distribution using constant-length vectors of the results at time 0.3 s after 6000 steps of simulation.

8.3. Example of shear driven cavity 3D problem

The shear driven cavity 3D problem previously used as the benchmark problem is modified by making middle square holes in three directions along the centerline axes of the cube. The fluid flow within the modified volume of the cube is generated by moving the top side of the cube at a constant diagonal 45 degree velocity generated while the other sides remain fixed. In the simulation, the kinetic viscosity and density are $\nu = 10^{-6}$ m^2/s and $\rho = 10^3$ kg/m^3, respectively. At the top side of the cube, particles are moved at a constant velocity of $V = 10^{-3}$ m/s in the x and z directions. A constant time step of 5×10^{-5} s is used. A constant smoothing

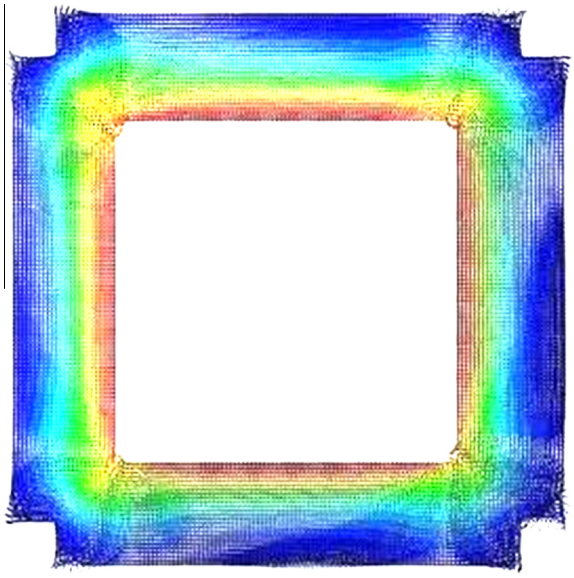

Fig. 23. Holed corner-cut 2D shear driven cavity velocity distribution result.

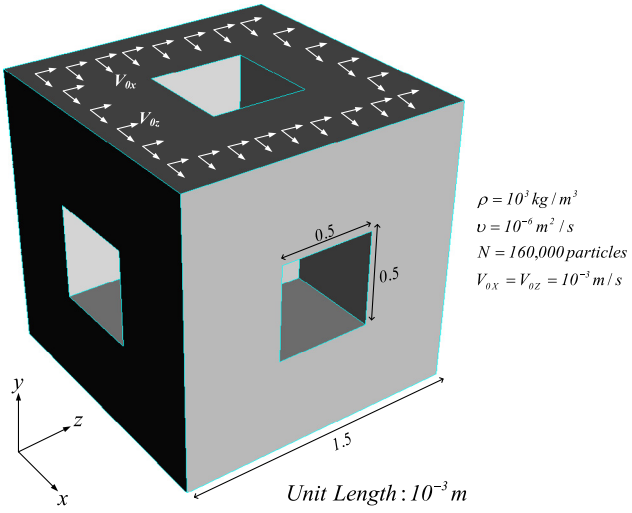

Fig. 24. Center hollowed cube 3D shear driven cavity problem.

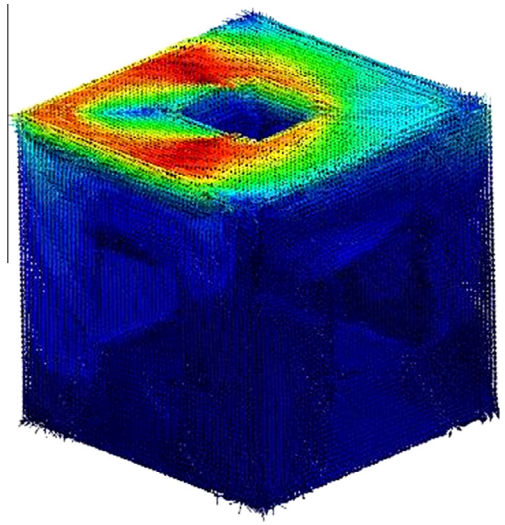

Fig. 25. Center hollowed cube 3D shear driven cavity velocity distribution result.

length of 2.5×10^{-5} m is used. Fig. 24 shows the center hollowed cube 3D shear driven cavity problem.

In total 160,000 particles were used in this simulation. By using Eq. (7), the division length of $L = 0.00015$ m was determined, and thus the number of horizontal and vertical equal divisions of the problem calculated is 10 in the x, y, and z directions.

From the defined smoothing length $h = 2.5 \times 10^{-5}$ m, the division length $L = 0.00018$ m can be obtained from Eq. (7), where $D = 3$ for the 3D type problem. The largest dimension of the simulation domain is then divided by L, giving the rounded number of nearly 10 divisions in the x, y, and z directions.

The CPU time required for conducting the 100 steps simulation was about 91.5 min which is about 2.5% of the time required when the proposed technique was not used, namely 3662.4 min. From the comparison of both execution CPU times, the proposed technique shows its effectiveness in reducing the time required for conducting the modified 3D example of shear driven cavity simulation.

Fig. 25 shows the velocity distribution using constant-length vectors of the results at time 0.3 s after 6000 steps of simulation.

9. Conclusion

From parametric studies, it can be concluded that the proposed technique is effective in a range of applications if certain numbers of divisions are used, especially when large numbers of particles are used in the SPH simulation. This tendency is shown by the concave lowest curve in the figures as the results of parametric studies. Further increase in the number of particles used in the simulation will help the proposed technique to achieve a significant level of improvement in terms of saving execution time. The most significant contribution of this paper in conducting fluid simulation by using the SPH method is that the proposed technique will reduce the execution time considerably using the handy formulation established. Therefore, the proposed technique is recommended for practice.

There are subjects for further work: to apply a similar algorithm to the other particle methods; to combine the technique with parallel computing technologies for the particle bookkeeping process.

References

[1] Lucy LB. Numerical approach to testing the fission hypothesis. Astron J 1977;82:1013–24.
[2] Gingold RA, Monaghan JJ. Smoothed particle hydrodynamics: theory and application to non-spherical stars. Mon Not R Astron Soc 1977;181:375–89.
[3] Bui HH, Sako K, Fukagawa R. Numerical simulation of soil-water interaction using smoothed particle hydrodynamics (SPH) method. J Terrramech 2007;44(5):339–46.
[4] Potapov S, Maurel B, Combescure A, Fabis J. Modeling accidental-type fluid-structure interaction problems with the SPH method. Comput Struct 2009;87(11–12):721–34.
[5] Grenier N, Antuono M, Colagrossi A, Le Touze D, Alessandrini B. An Hamiltonian interface SPH formulation for multi-fluid and free surface flows. J Comput Phys 2009;228(22):8380–93.
[6] Liu GR, Liu MB. Smoothed particle hydrodynamics: a meshfree particle method. World Scientific Publishing; 2003.
[7] Hockney RW, Goel SP, Eastwood JW. A 10000 particle molecular dynamics model with long range forces. Chem Phys Lett 1973;21(3):589–91.
[8] Gerardi G, Molteni D. Parallelization of a smoothed particle hydrodynamic code for simulation of shocks in accretion disks. Memorie Soc Astron Ital 2003;4:46–54.
[9] Appel AW. An efficient program for many-body simulations. SIAM J Sci Stat Comput 1985;6(1):85–103.
[10] Hernquist L. Hierarchical N-body methods. Comput Phys Commun 1988;48:107–15.
[11] Hernquist L, Katz N. TREESPH – a unification of SPH with the hierarchical tree method. Astrophys J Supp Ser 1989;70:419–46.
[12] Warren MS, Salmon JK. A portable parallel particle program. Comput Phys Commun 1995;87:266–90.
[13] Mattson W, Rice BM. Near-neighbor calculations using a modified cell-linked list method. Comput Phys Commun 1999;119:135–48.
[14] Monaghan JJ. Why particle methods work (hydrodynamics). SIAM J Sci Stat Comput 1982;3:422–33.
[15] Monaghan JJ, Lattanzio JC. A refined particle method for astrophysical problems. Astron Astrophys 1985;149:135–43.
[16] Lattanzio JC, Monaghan JJ, Pongracic H, Schwartz MP. Controlling penetration. SIAM J Sci Stat Comput 1986;7(2):591–8.
[17] Monaghan JJ. On the problem of penetration in particle methods. J Comput Phys 1989;82:1–15.
[18] Hockney RW, Eastwood JW. Computer simulations using particles. New York: Adam Hilger; 1988.
[19] Riffert H, Herold H, Flebbe O, Ruder H. Numerical aspects of the smoothed particle hydrodynamics method for simulating accretion disks. Comput Phys Commun 1995;89:1–16.
[20] Monaghan JJ, Gingold RA. Shock simulation by the particle method of SPH. J Comput Phys 1983;52:374–89.
[21] Sod GA. A survey of several finite difference methods for systems of hyperbolic conservation laws. J Comput Phys 1978;27:1–31.