



## Attempt to Suppress Numerical Viscosity in Incompressible SPH Method

Y. Fukunishi<sup>†</sup>, Y. Takahashi, Y. Nishio and S. Izawa

*Tohoku University, Aramaki-Aoba 6-6-01, Sendai, 980-8579, Japan*

<sup>†</sup>*Corresponding Author Email: [fushi@fluid.mech.tohoku.ac.jp](mailto:fushi@fluid.mech.tohoku.ac.jp)*

(Received September 11, 2018; accepted November 18, 2018)

### ABSTRACT

In this study, attempts to suppress numerical viscosity in incompressible smoothed particle hydrodynamics (SPH) computations are reported. Two-dimensional computations are performed for inviscid and viscous flows to evaluate the effects of numerical viscosity suppression. The first approach is to reduce numerical viscosity at the wall by considering only the wall-normal components of the forces between fluid particles and wall particles. The second approach is to reduce numerical viscosity within the flow field by employing elliptic kernel functions whose major axes are aligned with the local mean flow direction. It is found that special treatment of the wall radically reduces the numerical wall friction. Using an elliptic kernel function is found to work reasonably well in reducing numerical viscosity.

**Keywords:** Incompressible SPH computation, Numerical viscosity, Wall friction, Elliptic kernel function, Special wall treatment. .

### NOMENCLATURE

$f$	fluttering factor	$\rho$	density
$h$	smoothing length	$\alpha$	major axis radius
$m$	mass	$\beta$	minor axis radius
$r$	radial coordinate	$u_0$	initial velocity
$\mathbf{r}$	position vector	$u$	streamwise velocity
$t$	time	$u_\theta$	tangential velocity
$W$	kernel	$\mathbf{v}$	velocity vector
$x$	Streamwise coordinate		
$y$	wall-normal coordinate		

### 1. INTRODUCTION

Smoothed particle hydrodynamics (SPH) is a Lagrangian particle method in which the fluid continuum is discretized as a group of particles and the flow is directly represented by the movement of the particles (Monaghan 2005). Lagrangian methods are free from the numerical instability deriving from the convection term in the Eulerian formulations. Besides, they are also advantageous for calculating moving boundaries, since these boundaries are automatically represented by the motion of the particles. The SPH method was originally developed to simulate galaxy formation, which can be treated as a compressible flow in astrophysics (Gingold and Monaghan 1977). In the SPH method, a kernel function is defined for smoothly distributing the physical quantities for each particle. A physical quantity at an arbitrary position is obtained by

summing the values of its surrounding particles, which are weighted according to the kernel function. When solving partial differential equations, the derivatives at the center of a particle can be obtained as the vector sum of the gradients of the kernels of the surrounding particles. The SPH method has attracted attention because it is easy to simulate flows that contain a large deformation at the boundary as well as to simulate multi-phase flows. The technique is currently utilized for elastic body problems and heat conduction analyses, to name several examples.

There have been attempts to modify SPH to use it for computing incompressible flows. For instance, Monaghan (1994) used an artificial equation of state to simulate free-surface flows at low Reynolds numbers on the assumption that the flow is weakly compressible in which the Mach number is nearly equal to 0.1. This approach is named the “weakly

compressible” SPH (WCSPH). However, WCSPH requires very small time step. Instead, Cummins and Rudman (1999) proposed an alternative approach based on the projection method in which a pressure Poisson equation derived from an approximate pressure projection was solved to obtain the divergence-free velocity field. This projection SPH (PSPH) technique is capable to simulate incompressible flows at higher Reynolds numbers in the order of several hundred; however, it has a downside that the numerical error is accumulated in the density field. Pozorski (2002) and Szewc *et al.* (2012) proposed a correction algorithm to resolve the problem of accumulation of the density errors in PSPH. Also, Lee *et al.* (2008) formulated the governing equations based on the fractional step method and Imasato and Sakai (2006) applied the simplified marker and cell (SMAC) algorithm-generally to the SPH method.

The moving particle semi-implicit (MPS) method (Koshizuka and Oka 1996) is an extension of the SPH method for the incompressible flow computations. In the MPS method, the incompressible condition is satisfied by keeping the particle number density constant. The constant particle number density is obtained by solving Poisson’s equation for pressure. Although the MPS method allows increasing the time step, the computational cost increases significantly because of the large linear system of (Poisson’s) equations that must be solved. In addition, it is difficult to set boundary conditions, and nonphysical pressure vibration sometimes appears in the computation.

Our research group has analyzed low Mach number flows in various fields, double-diffusive convection (Shigeta *et al.* 2009), welding operation (Ito *et al.* 2015), drifting object swept by sudden water flow (Wakui *et al.* 2018) and flapping flags (Nishiura *et al.* 2018), by incorporating an incompressible approximation technique into the SPH method. In this technique, incompressible flows are calculated by correcting the particle velocities at each time step while keeping the density variation relatively small. During each iteration, the velocities of the particles are corrected by the density gradients caused by spatial non-uniformity of the particle distribution. However, with this approach, the particles become likely to form a regular close-packed structure as the number of iterations for density correction increases. Once such structures are formed in the flow field, the particles tend to move as a group and their moments spread instantaneously within the structure. As a result, the numerical viscosity increases. In this study, the principle underlying this change in numerical viscosity in the aforementioned incompressible SPH method is investigated in detail. Furthermore, techniques to reduce the numerical viscosity are proposed.

## 2. COMPUTATIONAL METHOD

### 2.1 Incompressible SPH Method

In the SPH method, a continuous function, termed the kernel function, provides each particle a smooth

distribution of physical quantities. The physical quantities at any arbitrary position can be obtained by superimposition of these kernel functions. The density  $\rho(\mathbf{r}_a)$  at the position  $\mathbf{r} = \mathbf{r}_a$ , for example, is represented by

$$\rho(\mathbf{r}_a) = \sum_b m_b W_{ab} \quad (1)$$

where  $W$  is the kernel function. We use the M4 spline kernel Monaghan and Lattanzio (1985), based on the cubic B-spline function,

$$W(q, h) = \frac{\sigma}{h^2} \begin{cases} 1 - \frac{3}{2}q^2 + \frac{3}{4}q^3 & (0 \leq q < 1) \\ \frac{1}{4}(2 - q)^3 & (1 \leq q < 2) \\ 0 & (2 \leq q) \end{cases} \quad (2)$$

where  $\sigma$  is a normalization constant which is  $10/7\pi$  in two-dimensional case, and  $q = |\mathbf{r}|/h$  is the distance normalized using the effective size of the kernel  $h$ , which corresponds to the effective radius of the particles. This effective radius  $h$  also corresponds to the shortest distance between the neighboring particles. This function gives non-zero values only within the range of  $r < 2h$  and, accordingly, the particle interactions occur only between particles that exist within that range. The accuracy of this kernel was discussed in detail by Monaghan and Lattanzio 1985.

Incompressibility is enforced by a density homogenization algorithm based on the predictor-modifier method in which the particle velocity of the predictor is corrected such that the local density becomes smooth instead of solving the pressure Poisson equation. An outline of the algorithm is shown in Fig. 1. First, the particle velocity  $\mathbf{v}(t)$  at time  $t$  is obtained from the difference in location between the previous time step  $t - \Delta t$  and the current time  $t$ . This velocity is used as the initial value for the predictor-corrector method, as follows:

$$\mathbf{v}^{(0)}(t) = \mathbf{v}(t). \quad (3)$$

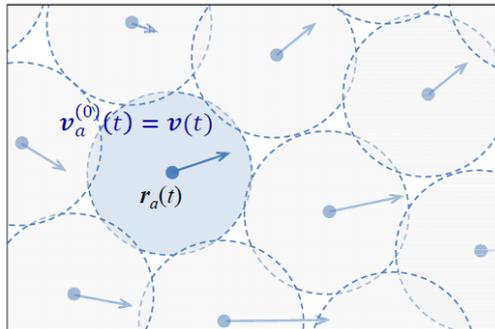
Tentatively, the particles move distances  $\mathbf{v}^{(0)}(t)\Delta t$ . Because the velocity predictor shown in Eq. (3) provides an estimate irrespective of the motion of other particles, the density distribution becomes uneven after all of the particles move. Thus, the velocity  $\mathbf{v}^{(0)}(t)$  is corrected as follows to even out the density distribution:

$$\Delta \mathbf{v}_a^{(n)} = -C \sum_b m_b \left( \frac{\rho_a^* - \rho_0}{\rho_a^{*2}} + \frac{\rho_b^* - \rho_0}{\rho_b^{*2}} \right) \nabla_a W_{ab}^* \quad (4)$$

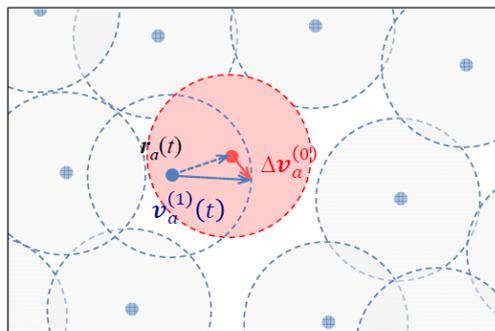
where  $C$  is a constant that determines the degree of correction,  $\rho_0$  is the reference density of the fluid, and  $\rho^*$  is the density at the tentative particle location. The correction  $\Delta \mathbf{v}_{corr}$  is added to the

velocity  $\mathbf{v}_{pred}$ , and each particle moves again in accordance with the new velocity. The  $n$ -th velocity is expressed by the following equation:

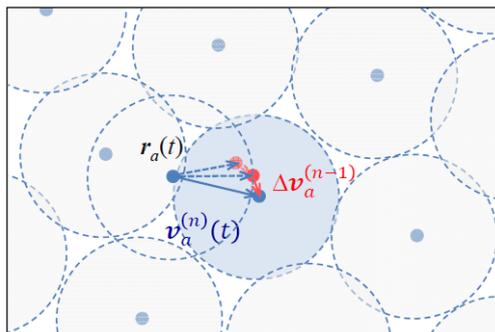
$$\begin{aligned} \mathbf{v}^{(n)} &= \mathbf{v}^{(n-1)} + \Delta\mathbf{v}^{(n-1)} \\ &= \mathbf{v}(t) + \sum_{i=0}^{n-1} \mathbf{v}^{(i)}. \end{aligned} \quad (5)$$



(a) predictor



(b) 1st corrector



(c)  $n$ th corrector

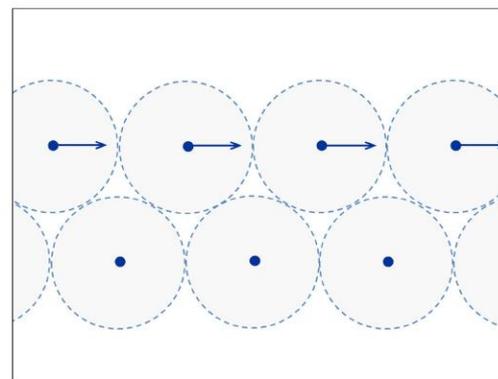
**Fig. 1. Adjustment of velocity by predictor-corrector method.**

By choosing an appropriate value for  $C$ , the density inhomogeneity improves iteratively. The number of iterations  $N$  is constant during the calculation.

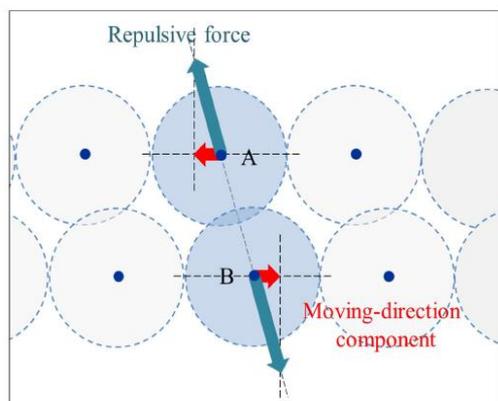
### 2.2 Increase in Numerical Viscosity by Crystallization

Figure 2 illustrates one possible cause for the

numerical viscosity. Following the density homogenization process, the particles tend to form a closest packing structure, as shown in Fig. 2(a), because it is the optimal solution. When particles are arranged as shown in Fig. 2(a) and the upper layer is moving rightward relative to the lower layer, after a short time  $\Delta t$ , the particles in the upper layer will move to positions shown in Fig. 2(b). Then, a repulsive force (denoted by blue arrows) will act between particle “A” in the upper layer and particle “B” in the lower layer. Consequently, the components of the repulsive forces in their respective directions of motion (signified by red arrows) will push particle “A” to the left and particle “B” to the right, which is equivalent to an exchange of momentum between the two particles. Similarly, particle “A” is also pushed leftward by the attractive force between particle “A” and particle next on the left side of particle “B”. Such nonphysical momentum exchange between the particles acts to increase the numerical viscosity. Once the particles form a packed formation during the density homogenizing process, the momenta of the particles will be distributed throughout the group via momentum exchanges. This phenomenon is equivalent to a strong viscous diffusion of momentum.

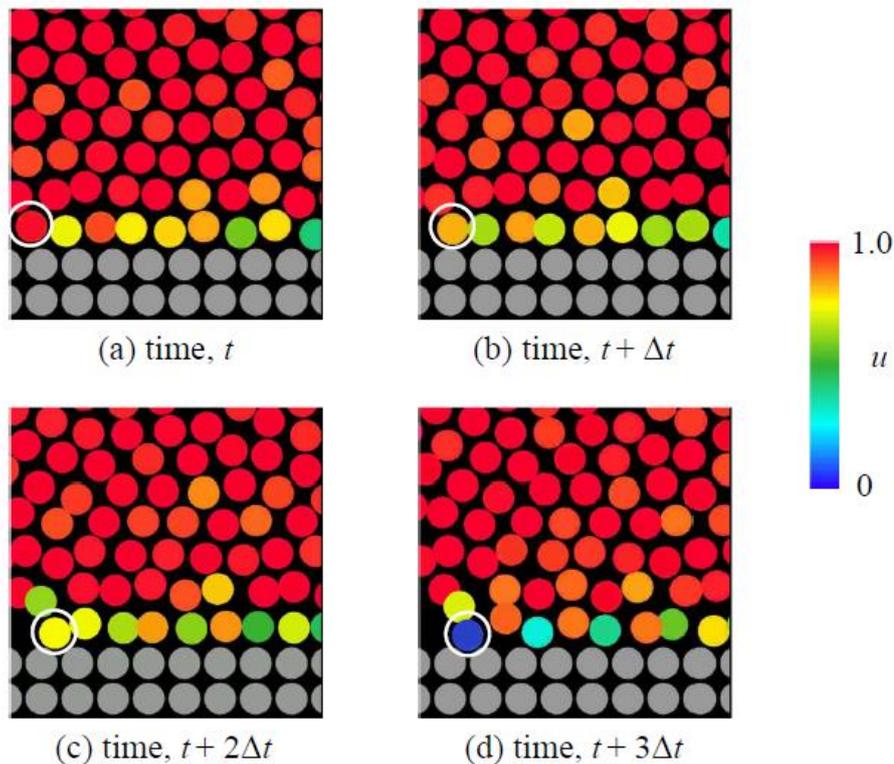


(a) Time,  $t$



(b) Time,  $t + \Delta t$

**Fig. 2. Density adjustment process and numerical viscosity.**



**Fig. 3. Behavior of particles near wall.**

### 2.3 New Wall Surface Model to Reduce Wall Resistance

The effect of numerical viscosity is particularly apparent near the wall. Figure 3 shows an enlarged view of the particles near the lower wall, a region of inviscid uniform flow between the parallel plates calculated using the incompressible SPH method. The vectors indicate the x-direction velocities. The velocities of particles near the wall decrease even though the slip condition is applied at the wall surface. A fluid particle, marked by a circle, can be observed getting stuck between the particles of the wall. This is because the fluid particle and the wall particles form a triangular structure in an attempt to satisfy the equal-density condition during the density homogenization process. Once the particle stops moving, it starts to behave as if part of the wall, thereby blocking the flow. Afterward, the surrounding fluid particles lose their momenta due to the blockage and, consequently, the velocity of the fluid near the wall rapidly attenuates.

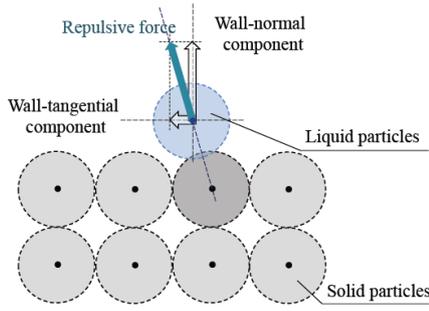
To prevent the aforementioned phenomenon from occurring, the technique whose schematic is shown in Fig. 4 is proposed. In the conventional method, the velocity corrector vector has both a wall-normal component and a wall-tangential component, as shown in Fig. 4(a). The wall-tangential component causes numerical viscosity, as previously described. The proposed idea is to use only the wall-normal component when correcting the predictor velocity during the iterative process. This technique is intended to allow fluid particles near the wall to flow smoothly past the wall particles. It is noted here that the reduction of numerical viscosity at the wall can

also be achieved by making the size of wall particles smaller than that of fluid particles. However, the use of smaller particles will result in the increase of the computational cost.

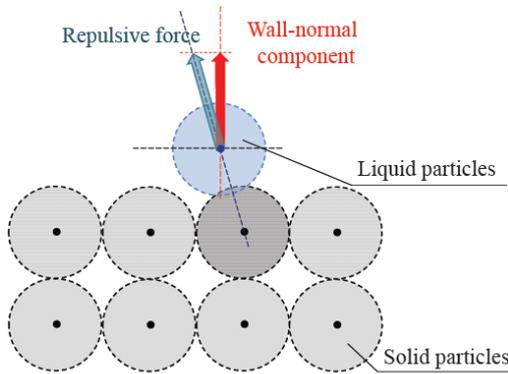
### 2.4 Anisotropic Elliptic Kernel to Reduce Numerical Viscosity within Flow Field

As demonstrated by the sample calculation of flow between parallel plates in section 2.2, the numerical viscosity promotes the transfer of low-velocity momentum to the center of the channel with time. Thus, we attempt to suppress the numerical viscosity within the flow as follows.

In the density homogenization process, Eq. (4) is used for the velocity corrector calculation and, accordingly, the velocity correction is carried out in the direction indicated by the gradient of the kernel. For example, when particle “A” in Fig. 5 is moving rightward relative to particle “B,” the velocity corrector vector will appear as shown in Fig. 5(a), which is caused by the interaction between the two particles. Then, because of the tangential component—that is, the component tangential to the velocity of the particle relative to the other particle—of the vector, an exchange of tangential momentum occurs between the two particles. This exchange of momentum serves to increase the numerical viscosity. The proposed idea is to alter the shape of the kernel function. Since the velocity corrector vector is in the direction of the gradient of the kernel function, when the shape of the kernel function is (b) Velocity corrector in new wall model elliptic, the velocity corrector vector will appear as shown in Fig. 5(b). Therefore, the tangential component of the vector becomes considerably shorter.



(a) Velocity corrector in conventional wall model



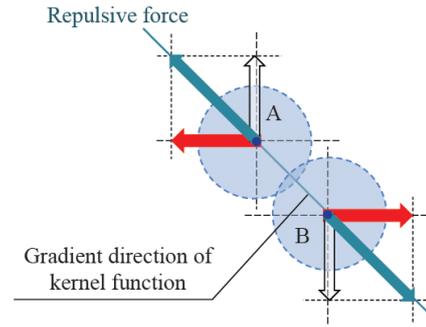
(b) Velocity corrector in new wall model

**Fig. 4. Schematic view of proposed technique to handle interaction between solid and liquid particles.**

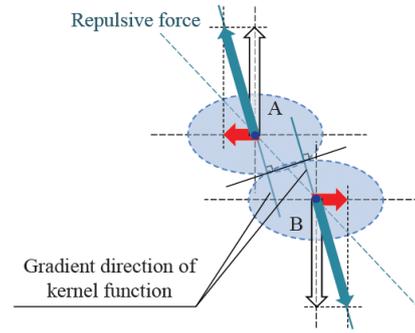
By suppressing the aforementioned exchange of tangential momentum, excessive momentum exchanges between neighboring particles along the directions of the minor axes should be suppressed. As a result, if the major axes of the elliptic kernel functions are aligned with the velocity vectors of the particles in the shear layer, the numerical viscosity should be suppressed. It should be noted that, when elliptic kernels are used, since the vectors of the forces acting between the particles do not pass through the centers of the particles, moments that induce rotation of the particles appear. These moments are ignored in this model. Moreover, the directions of the elliptic kernels are not preserved, and they must be set along the directions of local flow at every time step.

M4 spline curves are used to create an elliptic kernel function, which is equivalent to the conventional circular kernel function case. For computation in two dimensions, the conventional M4 spline kernel function is expressed as Eq. (2) using  $q = |\mathbf{r}|/h$ , where  $q$  is redefined to construct an anisotropic elliptic kernel, as follows:

$$q = \frac{1}{h} \sqrt{\left(\frac{x}{\alpha}\right)^2 + \left(\frac{y}{\beta}\right)^2} \quad (6)$$



(a) Density correcting vectors using conventional kernel



(b) Density correcting vectors using elliptic kernel

**Fig. 5. Schematic view of interaction between particles when elliptic kernel is used.**

where  $x, y$  denote the major axis and minor axis directions, respectively,  $\alpha$  is the major axis radius of the kernel at  $W = 0$ , and  $\beta$  is the minor axis radius.  $\alpha$  and  $\beta$  are given by the following equations using a flattening factor  $f$  ( $0 \leq f < 1$ ):

$$\alpha = \frac{h}{\sqrt{1-f}}, \quad \beta = h\sqrt{1-f}. \quad (7)$$

In the expression above, when  $f = 0$ ,  $\alpha$  is equal to  $\beta$ , which is an expression of the ordinary circular kernel given by Eq. (2). The flattening factor  $f$  determines the aspect ratio of the ellipse, and a flat elliptic kernel can be obtained by increasing  $f$ . For example, when  $f = 0.5$ , the kernel assumes an anisotropic elliptical shape as shown in Fig. 6. It should be noted that  $\alpha\beta = h^2$  regardless of  $f$ . In other words, even if  $f$  is varied between  $0 \leq f < 1$ , the area does not change on the same  $W$  section of the kernel. Therefore, the integrated value of the kernel is constant regardless of the value of  $f$ , which corresponds to the mass of a particle being constant.

## 2.5 Calculating the Effect of Viscosity When Elliptic Kernel is Used

When an elliptic kernel is used, the gradient of the kernel function depends not only on the distance between two particles but also on the direction of the line that passes through the centers of both particles. Furthermore, whereas the gradient of the kernel function itself can be used in the case of a circular

particle, the gradient corresponding to the steepest direction of the kernel function at each local point, which is given as the direction normal to the kernel contours, the must be obtained for the elliptic particle case. Another issue that must be taken into account is that, when the elliptic kernel is vertically short and horizontally wide, diffusion increases in the vertical direction but decreases in the horizontal direction. Thus, the viscous term for two-dimensional flows can be calculated by

$$\sum_b \frac{2m\tilde{\rho}v}{\rho_a\rho_b} (\mathbf{v}_a - \mathbf{v}_b) \frac{\mathbf{r}_{ab} \cdot \nabla_a W_{ab}}{|\mathbf{r}_{ab}|^2} C_{ab} \quad (8)$$

where  $\tilde{\rho}=(\rho_a + \rho_b)/2$  and  $C_{ab}$  as a transform coefficient, which is written by

$$C_{ab} = \frac{1}{h} \sqrt{\frac{(\beta r_x)^2 + (\alpha r_y)^2}{r_x^2 + r_y^2}} \quad (9)$$

Here,  $\mathbf{r}_{ab} = (r_x, r_y)$ . The value of  $C_{ab}$  represents a window size that determines the local diffusion flux at each direction: the maximum value is  $\sqrt{a/b}$  on the minor axis and the minimum is  $\sqrt{b/a}$  on the major axis, while  $C_{ab}=1$  for the circular kernel function. It is noted that, in two-dimensional case, the swept area of a relative position vector  $\mathbf{r}_{ab}$  is the same for both kernel functions. In other words, total diffusion flux is independent of kernel functions.

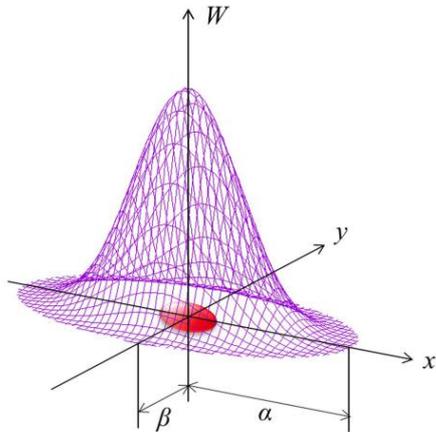


Fig. 6. Elliptic kernel function ( $f = 0.5$ ).

### 3. RESULTS AND DISCUSSION

#### 3.1 New Wall Model Tested in Uniform Flow between Parallel Plates

To verify the effect of the new wall model in reducing numerical viscosity at the wall, a uniform flow between the parallel plates is computed. Figure 7 shows the computational domain. The dimensions in the figure are normalized lengths. The dimensional quantities (length, time) are normalized by the effective radius of the particles  $h$  and the initial uniform velocity. The normalized computational time step is  $\Delta t = 0.05$ . A periodic boundary condition

is applied in the x direction. In both the conventional method and the new technique, the fluid is treated as inviscid and the slip condition is applied at the wall surface. A uniform velocity is used for the initial velocity, and changes in the velocity field caused by the numerical viscosity are examined. The initial particle distribution satisfies a uniform density. This arrangement is obtained by distributing the particles equally in space, slightly disturbing them randomly, and then carrying out the density homogenization process.

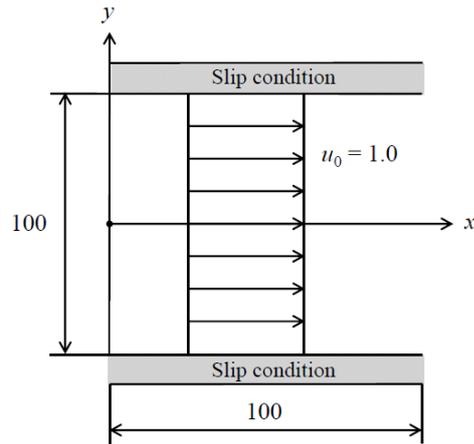


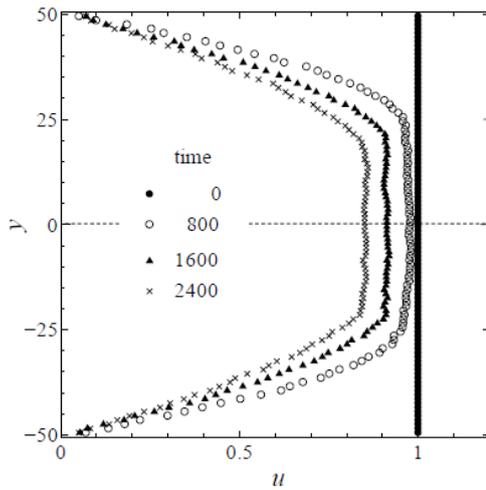
Fig. 7. Computational domain and initial velocity profile for inviscid uniform flow between parallel plates using new wall model.

Figure 8 compares the time variations of the averaged profiles of flow direction velocity. It is noted here that the velocity ranges are different in both figures. With the conventional method, although the slip condition is applied at the top and bottom walls, the particle velocities near the walls decrease and the profile becomes like that of a Poiseuille flow. Subsequently, the velocity of the entire flow gradually decreases. Since the fluid is treated as inviscid, the numerical viscosity must be the cause of the change in the velocity profile. The result clearly shows that the influence of the numerical viscosity is very large when the conventional method is used and that the slip condition at the wall cannot be realized.

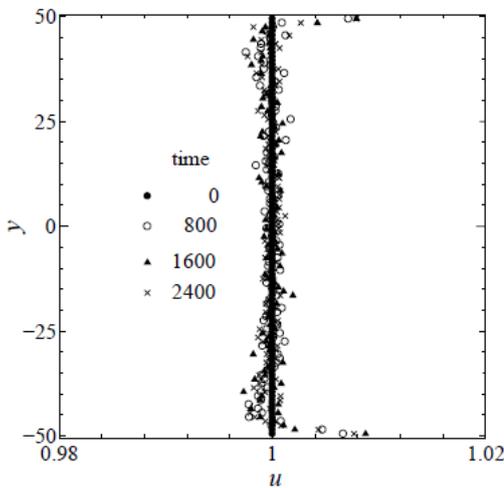
On the contrary, when the new technique is applied, the velocity profile changes little with time, and the fluid slips at the wall surface. It can clearly be observed that the proposed technique can more effectively suppress the numerical viscosity at the wall relative to the conventional method. In the following calculations, this technique is applied at the wall for all cases.

#### 3.2 Elliptic Kernel Model Applied to Inviscid Flow between Parallel Plates

Two-dimensional computation of inviscid flow between two parallel plates is carried out, focusing on the numerical viscosity. Computations using the normal circular kernel function and the newly proposed elliptic kernel function are performed and the results are compared.



(a) Conventional wall model



(b) New wall model

Fig. 8. Time variation of streamwise velocity profiles in testing the new wall model.

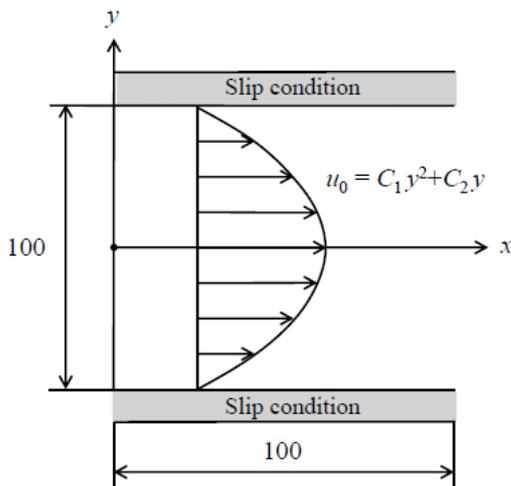


Fig. 9. Computational domain and initial velocity profile for inviscid flow between parallel plates using elliptic kernel functions.

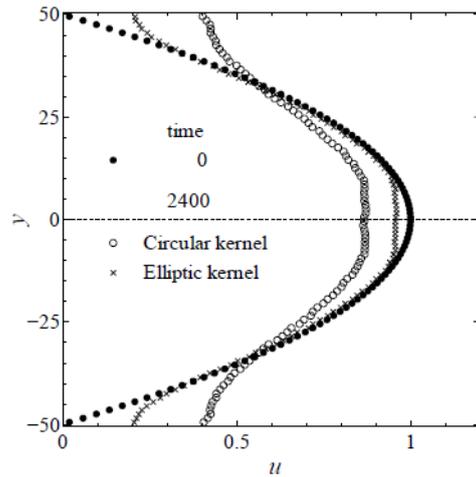


Fig. 10. Velocity profile of  $u$  at  $t = 2400$  in testing the elliptic kernel function of  $f = 0.5$ .

The computational domain is shown in Fig. 9. A slip condition is applied at both wall surfaces and a periodic boundary condition is applied in the  $x$  direction. The dimensional quantities are normalized by the effective radius of the particles  $h$  and the maximum value of the initial velocity distribution. A parabolic velocity distribution,

$$u_0 = C_1 y^2 + C_2 y \quad (10)$$

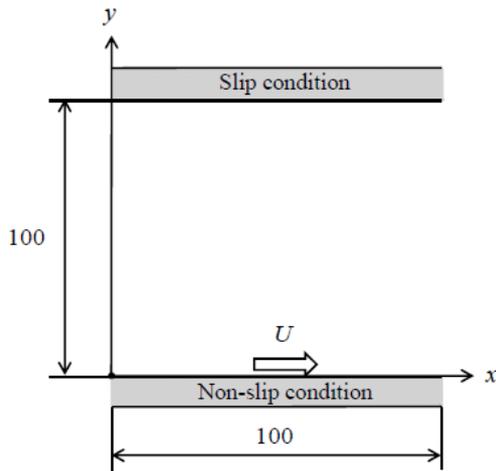
is used for the initial velocity profile. Here the values of the coefficients are  $C_1 = -2.84 \times 10^3$  and  $C_2 = 1.07 \times 10^2$ , and the normalized computational time step is  $\Delta t = 0.05$ .

Figure 10 shows the averaged profile of the velocity in the mean flow direction at  $t = 2400$ . In the figure, the closed-circle symbol represents the initial velocity profile, the open-circle symbol shows the case in which the normal circular kernel is used, and the cross symbol shows the case in which the elliptic kernel is used. When the ordinary circular kernel is used, the velocity profile quickly flattens at the center and the velocity decreases. This is due to the high numerical viscosity, where the momentum of particles in the central region is rapidly transferred to particles in the near-wall regions. On the other hand, when the elliptic kernel is used, the change from the initial profile is smaller. The result shows that the elliptic kernel results in a lower numerical viscosity.

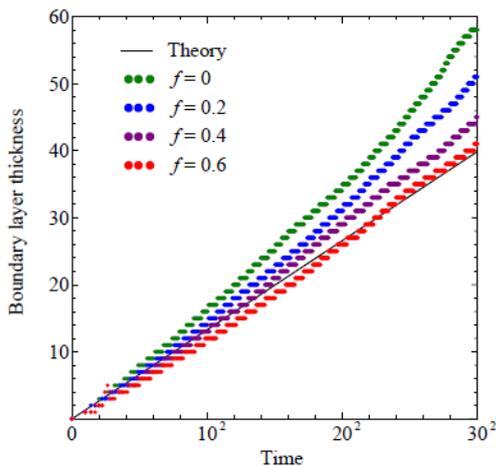
### 3.3 Elliptic Kernel Model Applied to Rayleigh Flow Computation

Two-dimensional computation of a Rayleigh flow in which the velocity of the bottom surface diffuses due to viscosity is carried out, and computations using the normal circular kernel function and the proposed elliptic kernel function are compared. Again, the periodic boundary condition is applied in the  $x$  direction. The calculation domain is shown in Fig. 11. The slip condition is applied to the upper wall while the non-slip condition is applied to the bottom wall. The bottom wall begins to move at a constant velocity  $U = 1.0$  in the positive  $x$  direction starting at  $t = 0$ . All of the particles have zero velocity prior to  $t$

= 0. Here, the dimensional quantities are normalized as well by  $h$  and  $U$ , and  $\Delta t = 0.25$ . In this calculation, the fluid is treated as a viscous fluid of kinematic viscosity coefficient, given by  $\nu = 1.50 \times 10^{-3} \text{ m}^2/\text{s}$ . The remaining computational conditions are identical to those from the previous section.



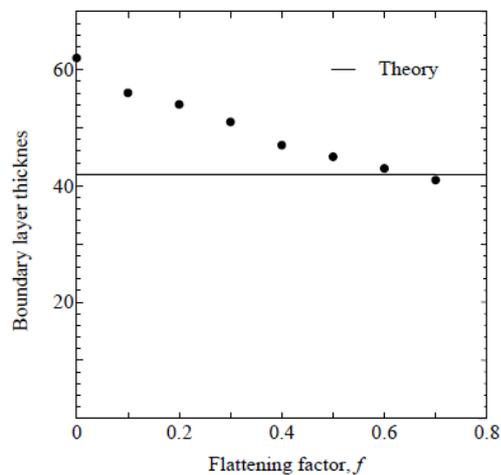
**Fig. 11. Computational domain for Rayleigh flow computations using elliptic kernel functions.**



**Fig. 12. Time variations in boundary layer thicknesses.**

Figure 12 shows the variation of the boundary layer thickness with time, where the boundary layer thickness is defined as the distance between the wall and the  $y$  coordinate at which the average velocity reaches  $0.01U$ . The thickness at the final time step, 1000, is also plotted in Fig. 13. The solid line in each figure represents the theoretical value. Theoretically, the boundary layer thickness, which represents the height that has been reached by the influence of viscous action, should increase proportionally with the square root of the product of the kinematic viscosity coefficient and time. The computational result in Fig. 12 shows that the boundary layer is much thicker when a normal circular kernel is used, due to the additional viscosity resulting from the numerical viscosity. However, the additional

viscosity gradually decreases when an elliptic kernel with a large flattening factor  $f$  is used. Figure 13 shows that, when a circular kernel is used, the net kinematic viscosity in the computation is approximately 1.5 times the true value. On the other hand, this value is 0.98 for an elliptic kernel with  $f = 0.6$ . This result demonstrates that the elliptic kernel performs reasonably well at suppressing the numerical viscosity. However, when the flattening factor exceeds 0.7, a small amount of noise begins to appear in the flow field, causing the calculation to become unstable. This is probably due to the deterioration of calculation accuracy of diffusion flux. In the preset computations, the diffusion flux is simply obtained under the assumption that the radial flux is just an orthogonal projection component of exact diffusion flux vector whose direction is normal to the kernel contour at each location. This assumption is not satisfied when the flattening factor becomes very large.



**Fig. 13. Relationship between flattening factor of elliptic kernel and boundary layer thickness at time step = 1000 (Rayleigh flow).**

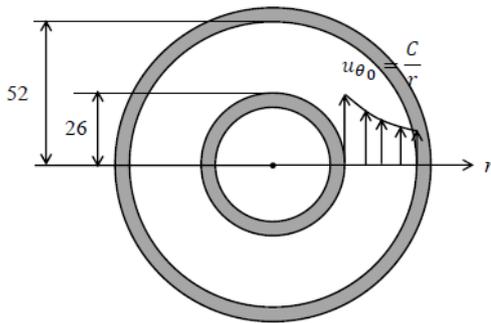
### 3.4 Elliptic Kernel Model Applied to Flow Computation in Concentric Circular Annulus

Finally, newly-proposed kernel model was applied to two-dimensional flow in a concentric circular annulus, which is a simple case of a non-parallel flow. Fig. 14 shows the computational domain. The fluid is inviscid and the slip condition is applied at both inner and outer wall surfaces. The dimensional quantities are normalized by the effective radius of the particles  $h$  and the maximum value of the initial velocity distribution. A free-vortex, or no-vorticity, velocity distribution,

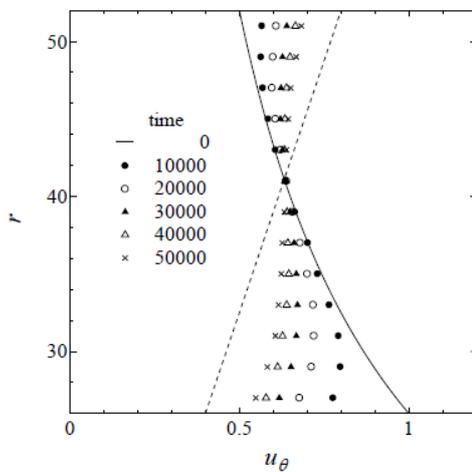
$$u_{\theta 0} = \frac{C}{r}, \quad (11)$$

is used as the initial velocity profile. Here, the value of coefficient  $C$  is set to 26 to give the velocity at the inner wall unity. The normalized computational time step is  $\Delta t = 0.25$ . In this computation, the velocity caused by the centripetal acceleration is added to the

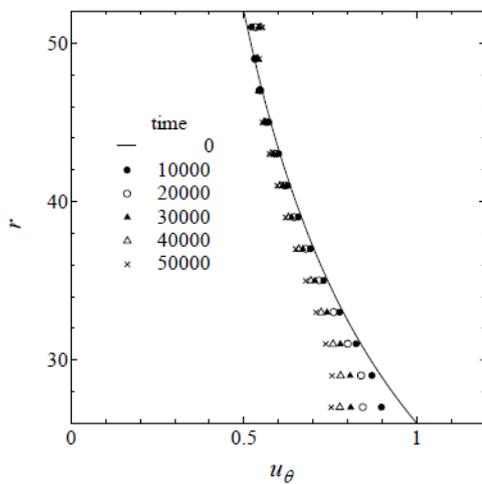
velocity predictor in the density homogenizing process for faster convergence.



**Fig. 14. Computational domain for flow computations in concentric circular annulus using elliptic kernel functions.**



(a) Conventional circular kernel function



(b) Elliptic kernel function of  $f = 0.6$

**Fig. 15. Time variation of circumferential velocity profiles in testing the elliptic kernel function  $f = 0.6$ . Dotted line in the figure represents the velocity profile of a solid body rotation.**

Figure 15 shows the time variations of the profiles of circumferential direction velocity averaged

depending on the radius from the center of the flow rotation. The case when an elliptic kernel of  $f=0.6$  is used is compared with the conventional circular kernel case. Since the fluid is inviscid, ideally, the velocity profile should remain the same. In the circular kernel case, the circumferential velocities of the particles in the inner regions quickly decrease while those in the outer regions increase. Eventually, the velocity profile approaches that of a solid body rotation, which is shown by a dotted line in Fig. 15, exhibiting the strong effect of numerical viscosity. On the other hand, change in the velocity profile for the elliptic kernel of  $f = 0.6$  case is much slower, showing much lower numerical viscosity. The result clearly shows the advantage of using the elliptic kernel to avoid the numerical viscosity.

#### 4. CONCLUSIONS

The cause of the numerical viscosity in an incompressible SPH method was discussed and new techniques to suppress the numerical viscosity were proposed.

First, to suppress the numerical viscosity at the wall, a new model that only takes into account the wall-normal components of the forces between fluid particles and wall particles was proposed. This new model was tested for the case of inviscid flow between parallel plates. The new wall model demonstrated that it could significantly suppress the numerical viscosity at the wall.

Next, to suppress the numerical viscosity within the flow field, an elliptic kernel function was proposed and tested for the case of inviscid flow between two parallel plates, between two concentric cylinders, and the Rayleigh flow. As a result, the anisotropic elliptic kernel was found to work reasonably well in reducing the numerical viscosity. In the Rayleigh flow case, the net kinematic viscosity was almost identical to the theoretical value when an elliptic kernel with  $f = 0.6$  was used, whereas the viscosity was approximately 1.5 times larger using the conventional circular kernel.

In conclusion, the newly proposed techniques appeared to be promising methods for reducing the numerical viscosity in incompressible SPH computations. However, with regard to the elliptic kernel function, computations became unstable when the flattening factor of the elliptic kernel was too large. Furthermore, it is unclear whether moments that induce rotation of the particles during the density homogenization process can be ignored. And, we do not have a method to determine the appropriate directions of the major axes of elliptic kernels. These issues must be solved before the new techniques can be applied to more general unsteady flow simulations.

#### REFERENCES

Cummins, S. J. and M. Rudman (1999). An sph projection method. *Journal of Computational Physics* 152, 584–607.

- Gingold, R. A. and J. J. Monaghan (1977). Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society* 181, 375–389.
- Imasato, J. and Y. Sakai (2006). Free surface flow analysis by smoothed particle hydrodynamics. *ASME Pressure Vessels and Piping Conference* 2, 533–537.
- Ito, M., Y. Nishio, S. Izawa, Y. Fukunishi, and M. Shigeta (2015). Numerical simulation of joining process in a tig welding system using incompressible sph method. *Japan Welding Society Journal* 33(2), 34–38.
- Koshizuka, S. and Y. Oka (1996). Moving- particle semi-implicit method for fragmentation of incompressible fluid. *Nuclear Science and Engineering* 123(3), 421–434.
- Lee, E. S., C. Moulinec, R. Xu, D. Violeau, D. Laurence, and P. Stansby (2008). Comparisons of weakly compressible and truly incompressible algorithms for the sph mesh free particle method. *Journal of Computational Physics* 227, 8417–8436.
- Monaghan, J. J. (1994). Simulating free surface flow with sph. *Journal of Computational Physics* 110, 399–406.
- Monaghan, J. J. (2005). Smoothed particle refined particle method for astrophysical problems. *Astronomy and Astrophysics* 149(1), 135–143.
- Monaghan, J. J. and J. C. Lattanzio (1985). A refined particle method for astrophysical problems. *Astronomy and Astrophysics* 149(1), 135–143.
- Nishiura, K., Y. Nishio, S. Izawa, and Y. Fukunishi (2019). Flapping motion of a permeable flag in uniform flow. *Fluid Dynamics Research* 51(2).
- Pozorski, J. (2002). Sph computation of incompressible viscous flows. *Journal of Theoretical and Applied Physics* 40(4), 917–937.
- Shigeta, M., T. Watanabe, S. Izawa, and Y. Fukunishi (2009). Incompressible sph simulation of double-diffusive convection phenomena. *International Journal of Emerging Multidisciplinary Fluid Sciences* 1(1), 1–18.
- Szewc, K., J. Pozorski, and J. P. Minier (2012). Analysis of the incompressibility constraint in the smoothed particle hydrodynamics method. *International Journal for Numerical Methods in Engineering* 92(4), 343–369.
- Wakui, N., M. Yamauchi, Y. Nishio, S. Izawa, and Y. Fukunishi (2018, September). Three-dimensional sph simulation of solid body washed away by water. In *Proceedings of the 8th International Conference on Fluid Mechanics*, Sendai, Japan.