# Chapter 3
# Smoothed particle hydrodynamics-based viscous deformable object modelling

Jung-Tae Kim, Fabio Ruggiero, Vincenzo Lippiello, Bruno Siciliano

**Abstract** Materials like fluids are long since important research objects of continuum mechanics as well as of computer graphics. Smoothed particle hydrodynamics(SPH) is one of the representation methods employed for continuous materials. Its simplicity in implementation and its realistic representation are drastically improved during the last decades. More recently, highly viscous fluids like honey, jam, and bread dough based on the SPH formulation have gained attention with impressive results. In this chapter, a novel implicit viscosity method is proposed. The internal viscosity forces are recursively calculated from the difference of the nearby velocities of the particles until they are small enough to be neglected. The proposed approach has longer time-steps compared with existing explicit viscosity methods, resulting in shorter computation time. Besides, the proposed method uses a physical viscosity coefficient, not an artificial one like in existing implicit viscosity methods, which helps predict the viscous behavior of continuous materials more accurately. The obtained results show that the computational time for

Jung-Tae Kim
MIKNMEK Inc., CheongAm-ro 87, Nam-gu, Pohang-si, Gyeongsangbuk-do, 37673, Republic of Korea, e-mail: tickleti1221@gmail.com

Fabio Ruggiero
CREATE Consortium & University of Naples Federico II, Department of Electrical Engineering and Information Technology, PRISMA Lab, Via Claudio 21, 80125, Naples, Italy, e-mail: fabio.ruggiero@unina.it

Vincenzo Lippiello
CREATE Consortium & University of Naples Federico II, Department of Electrical Engineering and Information Technology, PRISMA Lab, Via Claudio 21, 80125, Naples, Italy, e-mail: vincenzo.lippiello@unina.it

Bruno Siciliano
CREATE Consortium & University of Naples Federico II, Department of Electrical Engineering and Information Technology, PRISMA Lab, Via Claudio 21, 80125, Naples, Italy, e-mail: bruno.siciliano@unina.it

Table 3.1: Main symbols used in this chapter.

| **Definition** | **Symbol** |
|---|---|
| Material or substantial derivative | $\frac{\mathrm{D}}{\mathrm{D}t}$ |
| Fluid density | $\rho > 0$ |
| Space dimension | $r = \{2, 3\}$ |
| Flow vector velocity field | $\mathbf{v} \in \mathbb{R}^r$ |
| Hydrostatic pressure | $p > 0$ |
| Viscous stress | $\mathbf{T} \in \mathbb{R}^{r \times r}$ |
| Body force due to gravity, surface tension, or friction | $\mathbf{f}^{body} \in \mathbb{R}^r$ |
| $i$-th fluid sample point | $\mathbf{x}_i \in \mathbb{R}^r$ |
| Dynamic viscosity coefficient | $\mu > 0$ |
| Neighbourhood of the $i$-the particle | $\mathcal{N}_i$ |
| Mass of the $i$-th particle | $m_i > 0$ |
| Reynold number | $Re > 0$ |

the proposed approach is predictable, while the accuracy in modelling the viscosity behaviour is similar or higher than existing methods.

## 3.1 Brief introduction

Continuous materials (*e.g.*, solid objects, fluids) were widely and deeply investigated a long time after Augustin-Louis Cauchy firstly formulated them in the 19-th century. Various methods for modelling continuous materials are categorized as Lagrangian versus Eulerian approach, mesh-based versus mesh-free approach, or a hybrid combination. Below, a list of well-established methods is revised.

The finite difference method(FDM) is an Eulerian grid method constructing regular grids [57], while the finite volume method(FVM) generates subdomains referred to as cells [90]. The finite element method(FEM) is instead a grid method requiring mesh generation for their particle elements [140, 351]. Within the GSMs, gradient smoothing operations based on relevant gradient smoothing domains are employed to approximate derivatives [174]. The SPH consists of Lagrangian particles carrying the convection properties like mass, pressure, and velocity [176, 205]. Among the hybrid methods, it is worth mentioning the FLIP approach [39, 63], which is represented with Lagrangian particles while the projection step is performed on an Eulerian grid, and the MPM [299, 300], which handles collision and fracture of Lagrangian particles with the use of an Eulerian grid. The PBFM [185] applies geometric constraints of the PBDM [211] to enforcing constant density in the SPH framework. The integration of the PIC approach [124] and the FLIP one was proposed in [350] for animating granular materials such as the sand. The

combination of the SPH's incompressibility solver with the position-based velocity correction for volume preservation of viscoelastic fluids was presented in [306].

Each of the methods listed above has its benefits and issues compared to the others. Recently, thanks to the drastically increasing computation power of computers and the development of novel algorithms, the mesh-free Lagrangian modelling methods, like the SPH, are becoming more practical [176]. The intuitive and straightforward formulation of the SPH approach is suitable to describe highly deformable objects with complex surfaces. For this reason, such a method fits well the continuous material handled by this chapter.

The SPH method was firstly designed in [108, 181, 206] for analyzing astrophysical problems. Naturally, it emphasizes convection properties like mass rather than geometric volume conservation, precisely the benefits and drawbacks of a Lagrangian approach compared to an Eulerian approach. While SPH methods were applied to highly dense continuous materials like water, addressing the incompressibility of the materials would be a significant improvement. The first SPH method to simulate less compressible continuous materials, the so-called XSPH variant for matter diffusion, was proposed in [203, 204, 205] by including an artificial viscosity approach, the equation of state, and boundary conditions. Another contribution in applying the SPH method to incompressible materials can be found in [79]. An SPH technique based on a projection method to model incompressible flows was proposed in [72] by projecting a velocity onto a divergence-free subspace using pressure correction. An ISPH approach, in which the incompressibility of the material is satisfied through a pressure Poisson's equation, was proposed in [279]. A WCSPH method was proposed in [21] by using Tait's equation, resulting in fast computation and less density fluctuation. The success of the PCISPH method proposed in [292, 293] raised the popularity of implicit methods over explicit ones. The IISPH method in [136] computes density deviation not based on the position but based on the velocity, resulting in a robust time-integration scheme. The DFSPH approach computes impulse force to maintain the initial constant density and the divergence-free velocity field [22].

This chapter deals with highly viscous continuous materials like honey, jam, and bread dough. Based on the SPH formulation, several viscosity methods exist modelling incompressible behaviour of continuous materials. A conventional SPH method to model the incompressible flows with a low Reynolds number was proposed in [208]. A double density relaxation procedure to enforce incompressibility and particle anti-clustering was used in [59]. The work in [50] integrated the pressure force and the viscosity force in [209] with an additional elastic force derived from a modification of Hooke's law.

The Cross' model [69] for variable viscosity under shear stress in non-Newtonian fluids was employed in [279]. Later, a similar approach was used in [5, 6], resulting in accurate modelling of viscous jet buckling. An implicit viscosity integration method was presented in [307]: it was good at generating rotational viscous fluid behaviours like coiling or buckling with long

time-steps. At the same time, a different implicit viscosity formulation was proposed in [241]. An impulse force-based implicit viscosity method, which is a similar approach to a previous incompressibility solver [22], was introduced in [23]. The DC-PBD solver proposed in [15] handles both position-based and velocity-based constraints efficiently so that large-scaled and highly viscous fluids are animated with high speed. A further implicit viscosity solver, which is controllable by physical viscosity coefficients, was proposed in [336]. Besides these SPH-based methods, other approaches model continuous viscous materials as well. Namely, a variant of the MAC algorithm was proposed in [48] to simulate wax, while an implicit Eulerian method for simulating free-surface viscous fluids was proposed in [19].

Recently, the vorticity diffusion of continuous material related to the turbulence or the eddy near solid boundaries in high Reynolds number flows has gained attention within the research community. For instance, the implicit viscosity formulation in [241] was improved in [240] to include vorticity diffusion. The DVH was proposed in [261] to generate a highly accurate vorticity field in the 2D space. However, this chapter deals with high viscous fluids with a related flow characterized by a low Reynolds number and a slow velocity. Therefore, the vorticity diffusion is not addressed here.

The method introduced in this chapter takes inspiration from the implicit viscosity method addressed in [23, 241, 307]. Hence, the proposed method has long time-steps as the other implicit viscosity methods do, resulting in a shorter computation time to simulate the continuous material. In addition, the following novelties are introduced by this chapter: (*i*) the proposed method uses a physical viscosity coefficient, instead of an artificial one as commonly done in the literature, and thus it is possible to predict the viscous behaviour of the continuous materials accurately; (*ii*) while most of the existing implicit viscosity methods use optimization methods (*e.g.*, conjugate gradient or precomputed Jacobian to find the proper velocities of the particles), the proposed approach can control the viscosity accuracy and predict the computation time, as it will be verified in the proposed experiments.

The outline of the chapter is as follows. The mathematical background for Navier-Stokes equations used to represent the continuous materials and the SPH formulation, which is a mesh-free Lagrangian method implementing the Navier-Stokes equations, are briefly revised in the next section. The proposed approach is presented in Section 3.3 along with a brief description of existing viscosity approaches. At the same time, Section 3.4 gives additional ways to improve the introduced SPH implementation to be more accurate and fast in simulating the continuous materials. Experiments are described in Section 3.5 where the performance of the proposed algorithm is competitive with the conventional methods. Conclusions are provided within Section 3.6.

## 3.2 Theoretical background about Navier-Stokes's theorem and SPH

The basic concept of Navier-Stokes equations is briefly introduced in this section. The SPH method is revised in a nutshell as well .

### 3.2.1 Navier-Stokes' theorem for continuous materials

The two equations regarding the conservation of mass and momentum for continuous materials are considered [56, 251, 309]. The former is also known as the mass continuity equation

$$\frac{\mathrm{D}\rho}{\mathrm{D}t} = -\rho\nabla\cdot\mathbf{v}. \tag{3.1}$$

The latter equation is about the momentum conservation

$$\rho\frac{\mathrm{D}\mathbf{v}}{\mathrm{D}t} = -\nabla\cdot p\mathbf{I}_r + \nabla\cdot\mathbf{T} + \rho\,\mathbf{f}^{body}. \tag{3.2}$$

For any material property $\mathbf{A}_i$ of the fluid sample point $\mathbf{x}_i$ with proper dimension, in the Eulerian approach, the material derivative is defined as $\frac{\mathrm{D}\mathbf{A}}{\mathrm{D}t} = \frac{\partial\mathbf{A}}{\partial t} + (\mathbf{v}\cdot\nabla)\mathbf{A}$, in which $\frac{\partial}{\partial t}$ is the time derivative at a fixed Eulerian sample point $\mathbf{x}_i$ and $(\mathbf{v}\cdot\nabla)\mathbf{A}$ is the so-called advection term. In the Lagrangian approach , he material derivative is defined as $\frac{\mathrm{D}\mathbf{A}}{\mathrm{D}t} = \frac{d\mathbf{A}}{dt}$, in which $\frac{d}{dt}$ is the time derivative at an advected Lagrangian sample point $\mathbf{x}_i$ [137, 151].

### 3.2.2 SPH formulation

As briefly mentioned in Section 3.1, the SPH formulation is a mesh-free Lagrangian approach initially developed for astrophysical problems in [108, 181]. They treated the stars distributed sparsely in the universe as particles with mass and other properties, and they researched about interval physical properties among neighbour stars. Lately, the SPH formulation was applied to compressible fluid problems in continuum mechanics, in which each particle represents a collection of nearby atoms or molecules.

According to the general SPH formulation [151, 205, 203], a physical quantity $\mathbf{A}$ of any point $\mathbf{x}\in\mathbb{R}^r$ of the continuous material can be calculated by

$$\mathbf{A}(\mathbf{x}) = \int_{-\infty}^{\infty}\mathbf{A}(\mathbf{x}')\,W(\mathbf{x}-\mathbf{x}',h)\,d\mathbf{x}', \tag{3.3}$$

where $W : \mathbb{R}^r \to \mathbb{R}^{\geq 0}$ is an interpolating kernel, and $h$ is the radius of the kernel domain. The value for the outer of the kernel domain is zero. Such a kernel satisfies the following two properties

$$\int_{-\infty}^{\infty} W(\mathbf{x} - \mathbf{x}', h)\, d\mathbf{x}' = 1, \qquad \lim_{h \to 0} W(\mathbf{x} - \mathbf{x}', h) = \delta(\mathbf{x} - \mathbf{x}'), \tag{3.4}$$

where $\delta : \mathbb{R}^n \to \mathbb{R}^{\geq 0}$ is the Dirac delta function, whose output is equal to zero everywhere except for $\mathbf{x} = \mathbf{x}'$, and whose integral over the domain is equal to one. Moreover, decreasing monotonically with distance may be an additional useful property for a kernel function $W$.

The discrete approximation of (3.3) is given by

$$\mathbf{A}_i = \sum_j \frac{m_j}{\rho_j} \mathbf{A}_j\, W(\mathbf{x}_i - \mathbf{x}_j, h), \tag{3.5}$$

where $m_j > 0$ and $\rho_j > 0$ are the mass and the density of the $j$-neighbour particle, respectively. The gradient $\nabla \mathbf{A}$ for equation (3.5) can be defined using the second golden rule of SPH, which is to rewrite formulae with the density placed inside operators [111, 203, 241], as

$$\nabla \mathbf{A}_i = \frac{1}{\rho_i} \sum_j m_j (\mathbf{A}_j - \mathbf{A}_i) \nabla W(\mathbf{x}_i - \mathbf{x}_j, h). \tag{3.6}$$

Similarly, the Laplacian $\nabla^2 \mathbf{A}$ for equation (3.5) can be defined as

$$\nabla^2 \mathbf{A}_i = \sum_j \frac{m_j}{\rho_j} (\mathbf{A}_j - \mathbf{A}_i) \nabla^2 W(\mathbf{x}_i - \mathbf{x}_j, h), \tag{3.7}$$

as proposed in [61, 209] by applying the second golden rule of SPH twice for incompressible continuous materials.

From now in this chapter, $W_{ij}$ will be used as short notation for $W(\mathbf{x}_i - \mathbf{x}_j, h)$.

## 3.3 Viscosity property and various viscosity methods for SPH

The Navier-Stokes equation (3.2) for incompressible fluids with a Lagrangian approach can be written as follows for the $i$-th element

$$\frac{\mathrm{D}\, \mathbf{v}_i}{\mathrm{D}\, t} = -\frac{1}{\rho_i} \nabla \cdot p_i\, \mathbf{I}_r + \frac{1}{\rho_i} \nabla \cdot \mathbf{T}_i + \mathbf{f}_i^{body}, \tag{3.8}$$

where it is possible to recognize the net motion force, $\mathbf{f}_i^{motion} = \frac{\mathrm{D}\mathbf{v}}{\mathrm{D}t}$, equal to the sum of the pressure force, $\mathbf{f}_i^{pressure} = -\frac{1}{\rho_i}\nabla \cdot p\,\mathbf{I}_r$, occurred by pressure difference, the viscosity force, $\mathbf{f}^{vis} = \frac{1}{\rho_i}\nabla \cdot \mathbf{T}_i$, occurred by shear stress, and the body force, $\mathbf{f}_i^{body}$, acting on a continuous fluid and given by gravity, inertial accelerations, elastostatic accelerations, and so on. Incompressibility of continuous materials is guaranteed by solvers like IISPH [136] or DFSPH [22], which generates the proper pressure force $\mathbf{f}_i^{pressure}$ to keep constant the density, that is, $\rho_i = \rho_0$, or at least $\frac{\mathrm{D}\rho_i}{\mathrm{D}t} = 0$. The SPH is purely based on Lagrangian approach, hence the material derivative for a velocity $\mathbf{v}_i$ is calculated as $\frac{\mathrm{D}\mathbf{v}_i}{\mathrm{D}t} = \frac{d\mathbf{v}_i}{dt}$ [137].

Concerning incompressible fluids, the viscosity force $\mathbf{f}^{vis}$ is defined as

$$\begin{aligned}\mathbf{f}_i^{vis} &= \frac{1}{\rho_i}\nabla \cdot \mathbf{T}_i = \frac{1}{\rho_i}\nabla \cdot \left(2\,\mu\frac{\nabla\mathbf{v}_i + \nabla\mathbf{v}_i^T}{2}\right) \\ &= \frac{1}{\rho_i}\left(\mu\underbrace{\nabla \cdot \nabla\mathbf{v}_i}_{\nabla^2\mathbf{v}_i} + \mu\underbrace{\nabla \cdot (\nabla\mathbf{v}_i)^T}_{\nabla(\nabla\cdot\mathbf{v}_i)=0}\right) = \frac{1}{\rho_i}\mu\nabla^2\mathbf{v}_i,\end{aligned} \tag{3.9}$$

where $\nabla^2\mathbf{v}_i$ is the Laplacian of the velocity.

Regarding the SPH formulation, a double discretization of the velocity generates error accumulation, and it is prone to be too much sensitive with respect to the velocity of the particles and the kernel function. Hence, the conventional explicit viscosity methods [204, 208, 279] use alternative formulations by employing the first derivative kernel function only, while the existing implicit viscosity methods [23, 241, 307] try not to calculate the viscosity force explicitly. The formulation proposed in [209] is employed in this chapter to define the viscosity force $\mathbf{f}_i^{vis}$ as

$$\mathbf{f}_i^{vis} = \frac{1}{\rho_i}\mu\nabla^2\mathbf{v}_i = \frac{1}{\rho_i}\mu\sum_j \frac{m_j}{\rho_j}(\mathbf{v}_j - \mathbf{v}_i)\,\nabla^2 W_{ij}. \tag{3.10}$$

The goal of most of the viscosity methods is to find out the velocity $\mathbf{v}_i$ satisfying the Navier-Stokes equation (3.8). However, it is not easy to find out such a velocity because the viscosity force $\mathbf{f}_i^{vis}$ is relative to both the velocity of a particle and the velocities of its neighbours, as evident from (3.10).

To easy explain the presented methodology, all the particles in the continuous material are assumed to be initially stable, *i.e.*, $\mathbf{v}_i = \mathbf{v}_0\ \forall i$ where $\mathbf{v}_0 \in \mathbb{R}^r$ is a constant velocity, for instance zero. Let $\mathbf{f}_i^{ext} = \mathbf{a}_i \in \mathbb{R}^r$ be any other force applied to the $i$-th particle, assimilated to an acceleration $\mathbf{a}$, except its viscosity force $\mathbf{f}_i^{vis}$. Then, consider that only a particle $i$ of the continuous material is subject to an external force $\mathbf{f}_i^{ext}$, while the other particles are not $\mathbf{f}_{j\neq i}^{ext} = \mathbf{0}_r$. When the $i$-th particle is accelerated by the external force $\mathbf{f}_i^{ext}$, its closest particles drag it to be less accelerated. Such a dragging force is equal to the viscosity force $\mathbf{f}_i^{vis}$ for the $i$-th particle as

$$\mathbf{f}_i^{vis} = \sum_j \mathbf{f}_{i \leftarrow j}^{vis}, \tag{3.11}$$

where $i \leftarrow j$ means the effect of a particle $j$ toward a particle $i$. Therefore, folding equation (3.10) into equation (3.8), with the assumption $\mathbf{v}_i = \mathbf{v}_0 \ \forall k$, yields

$$\frac{\Delta \mathbf{v}_i}{\Delta t} \simeq \frac{1}{\rho_i} \mu \sum_j \frac{m_j}{\rho_j} (\mathbf{v}_j - (\mathbf{v}_i + \Delta \mathbf{v}_i)) \, \nabla^2 W_{ij} + \mathbf{a}_i = -\frac{1}{\rho_i} \mu \, \Delta \mathbf{v}_i \sum_j \frac{m_j}{\rho_j} \, \nabla^2 W_{ij} + \mathbf{a}_i, \tag{3.12}$$

where the equality holds if the change of velocity, $\Delta \mathbf{v}$, is constant during a given time-step $\Delta t > 0$. Then, the increase of velocity for the particle $i$, $\Delta \mathbf{v}_i$, caused by the external force $\mathbf{f}_i^{ext}$ can be obtained from

$$\Delta \mathbf{v}_i \simeq (\Delta t \cdot \mathbf{a}_i) / (1 + \Delta t \cdot \frac{\mu}{\rho_i} \sum_j \frac{m_j}{\rho_j} \, \nabla^2 W_{ij}). \tag{3.13}$$

Moreover, the viscosity force occurred by a neighbourhood particle $j$ can be retrieved from $\mathbf{f}_{i \leftarrow j}^{vis} = -\frac{1}{\rho_i} \mu \, \Delta \mathbf{v}_i \frac{m_j}{\rho_j} \nabla^2 W_{ij}$. According to the third Newton's law, $m_i \mathbf{f}_{i \leftarrow j}^{vis} + m_j \mathbf{f}_{j \leftarrow i}^{vis} = \mathbf{0}_r$, the dragging force from a particle $j$ to the particle $i$ also affects the particle $j$ itself such that

$$\mathbf{f}_{j \leftarrow i}^{vis} = \frac{1}{\rho_j} \mu \, \frac{m_i}{\rho_i} ((\mathbf{v}_i + \Delta \mathbf{v}_i) - \mathbf{v}_j) \nabla^2 W_{ij}. \tag{3.14}$$

Then, the force $\mathbf{f}_{j \leftarrow i}^{vis}$ can be regarded as a new external force $\mathbf{f}_j^{ext'} \in \mathbb{R}^r$ for the particle $j$, that is, $\mathbf{f}_j^{ext'} = \mathbf{f}_{j \leftarrow i}^{vis}$.

The described procedure is resumed within the schematic diagram in Fig. 3.1. Assume that five particles exist, each of which is denoted with subscript $j1$, $j2$, $j3$, $j4$, and $i$, serially and stably placed in a row. The particle $i$ is affected by four neighbourhood particles $j1$, $j2$, $j3$, and $j4$ in the SPH formulation. The particle $i$ is subject to a given external force $\mathbf{f}_i^{ext}$ vertically applied in the upward direction. Since the surrounding particles are still stable, the initial viscosity force is given by $\mathbf{f}_i^{vis} = \sum_j \mathbf{f}_{i \leftarrow j}^{vis}$ , where $j = \{j1, j2, j3, j4\}$. Then, the summation of the provided external force $\mathbf{f}_i^{ext}$ and the viscosity force $\mathbf{f}_i^{vis}$ is the net motion force $\mathbf{f}_i^{motion}$ of the $i$-th particle. Subsequently, the change of velocity $\Delta \mathbf{v}_i$ caused by the motion force $\mathbf{f}_i^{motion}$ creates the viscosity forces, $\mathbf{f}_{j1 \leftarrow i}^{vis}$, $\mathbf{f}_{j2 \leftarrow i}^{vis}$, $\mathbf{f}_{j3 \leftarrow i}^{vis}$, and $\mathbf{f}_{j4 \leftarrow i}^{vis}$. Each of these viscosity forces can be regarded as new external forces, $\mathbf{f}_{j1}^{ext'}$, $\mathbf{f}_{j2}^{ext'}$, $\mathbf{f}_{j3}^{ext'}$, and $\mathbf{f}_{j4}^{ext'}$, for the neighbourhood particles.

Now, consider the case where multiple particles in a continuous material are subject to external forces $\mathbf{f}_{j \neq i}^{ext}$, simultaneously. Following the previous equations (3.13)-(3.14), a particle $i$ might be subject to multiple viscosity forces $\sum_j \mathbf{f}_{i \leftarrow j}^{vis}$ from its neighbourhoods, so that the new external force $\mathbf{f}_i^{ext'}$

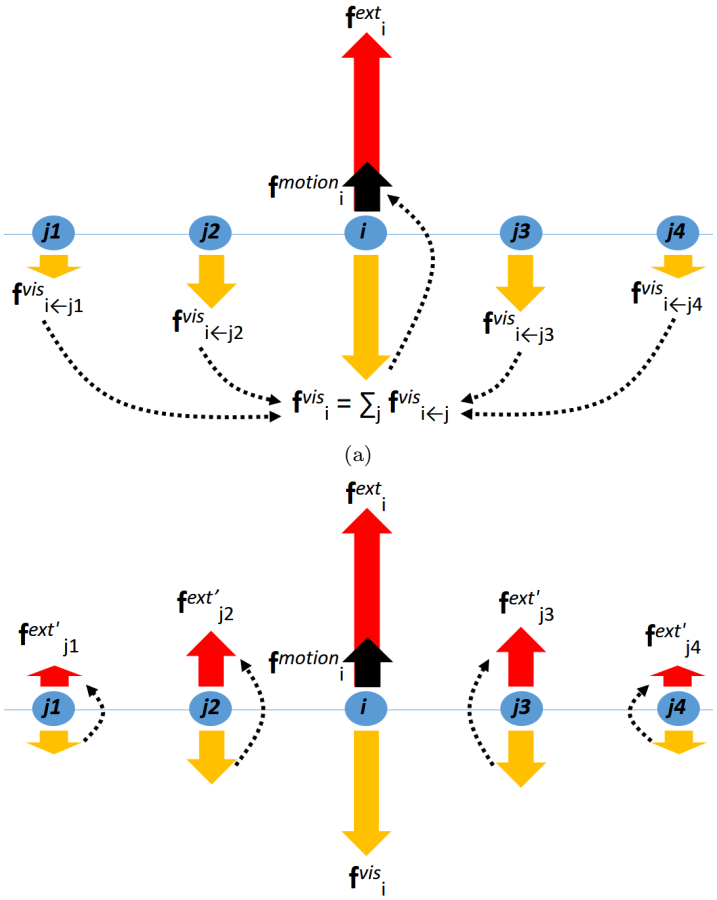



Fig. 3.1: When an external force $\mathbf{f}_i^{ext}$ is applied on the $i$-th particle of a series of particles arranged in a row, not only the $i$-th one gets the viscosity force $\mathbf{f}_i^{vis}$ but also the particles $j1 \sim j4$ are subject to new external forces $\mathbf{f}_{j1}^{ext'} \sim \mathbf{f}_{j4}^{ext'}$ according to Newton's third law. Red, yellow, and black arrows indicate external, viscosity, and net motion forces, respectively.

can be extended to

$$\begin{aligned}\mathbf{f}_i^{ext'} = \sum_j \mathbf{f}_{i \leftarrow j}^{vis} &= \frac{1}{\rho_i}\mu \sum_j \frac{m_j}{\rho_j}\Big((\mathbf{v}_j + \Delta\mathbf{v}_j) - (\mathbf{v}_i)\Big)\nabla^2 W_{ij} \\ &= \frac{1}{\rho_i}\mu \sum_j \frac{m_j}{\rho_j}(\Delta\mathbf{v}_j)\nabla^2 W_{ij}.\end{aligned} \tag{3.15}$$

Furthermore, removing the assumption that $\mathbf{v}_i = \mathbf{v}_0 \ \forall i$ yields

$$\frac{\Delta \mathbf{v}_i}{\Delta t} \simeq \frac{1}{\rho_i} \mu \sum_j \frac{m_j}{\rho_j} (\mathbf{v}_j - (\mathbf{v}_i + \Delta \mathbf{v}_i)) \nabla^2 W_{ij} + \mathbf{a}_i, \tag{3.16}$$

which is equivalent to

$$\frac{\Delta \mathbf{v}_i}{\Delta t} \simeq -\frac{1}{\rho_i} \mu \, \Delta \mathbf{v}_i \sum_j \frac{m_j}{\rho_j} \nabla^2 W_{ij} + \underbrace{\left( \frac{1}{\rho_i} \mu \sum_j \frac{m_j}{\rho_j} (\mathbf{v}_j - \mathbf{v}_i) \nabla^2 W_{ij} + \mathbf{a}_i \right)}_{\mathbf{f}_i^{ext'} = \mathbf{a}_i'}. \tag{3.17}$$

Note that, considering the combination of the last two terms in the right side of the above equation as a new external force $\mathbf{f}_i^{ext'} = \mathbf{a}_i'$, then equation (3.17) becomes equivalent to (3.12).

In view of the law of conservation of energy and the condition that the kernel Laplacian is $\nabla^2 W \geq 0$, the following expressions hold

$$\begin{aligned} & \mathbf{f}_i^{motion} = \mathbf{f}_i^{vis} + \mathbf{f}_i^{ext} = \sum_j \mathbf{f}_{i \leftarrow j}^{vis} + \mathbf{f}_i^{ext} \\ \Rightarrow \quad & \mathbf{f}_i^{ext} = \mathbf{f}_i^{motion} + \sum_j \mathbf{f}_{j \leftarrow i}^{vis} \\ \Rightarrow \quad & \left\| \mathbf{f}_i^{ext} \right\| \geq \sum_j \left\| \mathbf{f}_{j \leftarrow i}^{vis} \right\| = \sum_j \left\| \mathbf{f}_j^{ext'} \right\|. \end{aligned} \tag{3.18}$$

Therefore, the magnitude of the sum of the external forces, $\| \sum \mathbf{f}^{ext} \|$, always decreases and converges to zero as the procedure is iterated.

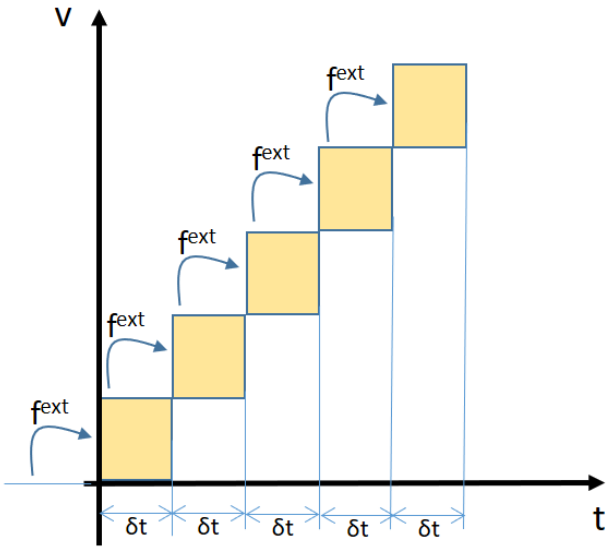
So far, a viscosity equation has been retrieved from the knowledge of an applied external force, and such a viscosity term has been used as a further external force as well. There is not so much difference with conventional explicit viscosity methods. However, imagine the sequential procedure with a very short time-step $\delta t \ll \Delta t$ in an explicit approach. The external force $\mathbf{f}_i^{ext}$ affects the particle $i$: after $\delta t$, the particle $i$ changes its velocity accordingly to $\delta \mathbf{v}_i$ and, subsequently, any particle $j \in \mathcal{N}_i$ in the neighbours of $i$ gets an external force $\mathbf{f}_j^{ext}$ from $i$. During the next $\delta t$, the particle $j$ subject to the external force $\mathbf{f}_j^{ext}$ changes its velocity accordingly to $\delta \mathbf{v}_j$ and, subsequently, any particle $k \in \mathcal{N}_j$ in its neighbour, including the particle $i \in \mathcal{N}_j$, gets an additional external force $\mathbf{f}_k^{ext}$ from the particle $j$ again. This procedure is iterated for $\Delta t / \delta t$ times, and thus the number of interactions tends to infinite as $\delta t \to 0$.

For a Newtonian fluid, since the dynamic viscosity coefficient $\mu$ is constant, it is possible to separate the computation of the new external forces, $\mathbf{f}^{ext'}$, from the sequential processing without increasing the time-step. In this way, it is possible to take a longer time-step $\Delta t$ for viscosity calculation like other implicit viscosity methods.

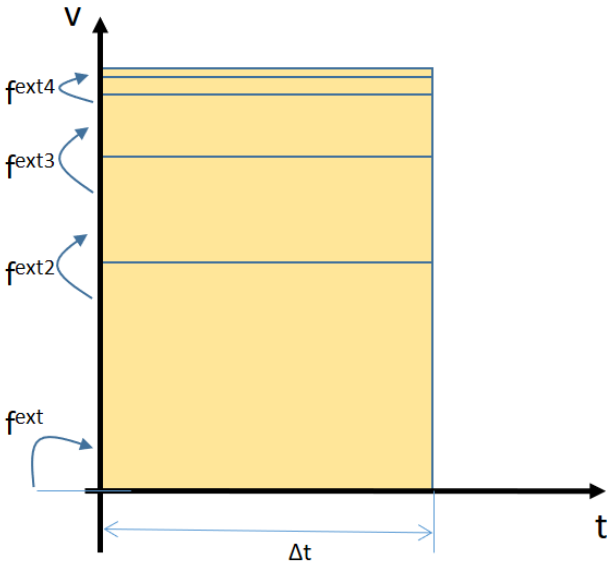
A schematic diagram is shown in Fig. 3.2. On the one hand, Fig. 3.2-(a) shows that the velocity changes $\delta\mathbf{v}$ are accumulated linearly during a short time $\delta t$ as in the existing explicit viscosity methods. To use these approaches, the short time-step $\delta t$ should be selected efficiently. Furthermore, suppose there is an SPH routine including neighbour searching and incompressible solving for each iteration as the conventional explicit viscosity methods do. In that case, the computation time of the SPH algorithm also drastically increases. On the other hand, Fig. 3.2-(b) illustrates the approach proposed in this paper, which calculates the velocity changes $\Delta\mathbf{v}$ during a given time-step $\Delta t \gg \delta t$, and then the induced external force $\mathbf{f}_i^{ext'} = \sum_j \mathbf{f}_{i \leftarrow j}^{vis}$ is applied to the surrounding particles subsequently without adding a time-step. Moreover, the magnitude of the change of velocity $\|\Delta\mathbf{v}\|$ decreases as the routine is repeated: hence, the viscosity error can be controlled, as well as the number of iterations, with a trade-off between them.

The schematic pseudo-codes of both the SPH algorithm and the one related to the proposed viscosity method are shown in Algorithm 2 and Algorithm 3, respectively. In Algorithm 2, the properties related to the particles are initialized, like the space dimension, the fluid density, the mass of a particle, the position $\mathbf{x}$ and velocity $\mathbf{v}$ of each particle, as well as the kernel function with the radius of its domain $h$, the time-step, the total simulation time, and so on (line 2). Then, the main routine (lines 4-9) is repeated until the end of the given simulation time. The neighbour search for each particle is carried out (line 4), then the kernel value $W$, its gradient, its Laplacian, and its density are calculated using the distances to the neighbours (line 5). The neighbour search and the computation of the kernel function are the most time-consuming parts. The related calculations can be improved with various methods like implementing a look-up table for the kernel function. The body force $\mathbf{f}^{body}$, like gravity, is applied to each particle (line 6). There are two main sub-routines: one for the incompressibility of the continuous material and the other one to calculate the viscosity force. In this chapter, the conventional incompressibility methods, like the IISPH approach in [241] and the DFSPH method in [22] , are employed. They guarantee incompressibility of the continuous materials through $\rho = \rho_0$, or at least through $\frac{\mathrm{D}\rho}{\mathrm{D}t} = 0$ (line 7). Under the assumption that a given continuous material is incompressible, the proposed viscosity method is run (Algorithm 3). It calculates the velocity of a particle $\mathbf{v}$ satisfying viscosity (line 8). Finally, the position of a particle $\mathbf{x}$ is updated based on the velocity $\mathbf{v}$ (line 9).

In detail, Algorithm 3 consists of an initial part (line 2-6) and a loop (line 8-16). Firstly, the external force $\rho\,\mathbf{a}'$ is re-calculated, including not only the original external body force, like gravity, but also the viscosity force $\mathbf{f}^{vis}$ occurred by the relative velocity between a particle $i$ and its neighbours $\mathcal{N}_i$ using the equation (3.17). Within the loop part, two termination conditions are checked: the former to verify when the number of iterations is less than the maximum number *max_iter*; the latter to prove that the maximum magnitude of the accelerations of the particles $\|\mathbf{a}\|$ is greater than the minimum threshold




(a) conventional explicit approach




(b) proposed method

Fig. 3.2: While the conventional explicit approaches require a short time-step $\delta t$ for sequential calculations, on the top, the proposed method uses a given time-step $\Delta t \gg \delta t$ for calculating the velocity changes and the additional external forces, on the bottom.

**ALGORITHM 2:** OUTLINE OF THE SPH ALGORITHM

```
1  begin
2      initialize particles of a continuous material in SPH formulation
3      repeat
4          search neighbourhood for each particle
5          compute W, ∇W, ∇²W, and ρ for each particle
6          apply the body force f^body e.g. gravity
7          correct the density error s.t. ρ = ρ0 or at least Dρ/Dt = 0
8          apply the viscosity effect on the continuous material (Algorithm 3)
9          update the position x of each particle
10     until
11 end
```

keeping a reasonable computation time and preventing the computation of negligible small external forces (line 7). For these two termination conditions, *max_acc* and *cur_iter* are calculated (line 5-6 and line 15-16). Within the loop, two main parts can be recognized: the former is the update of the velocity of each particle (line 8-11) using (3.13); the latter is the calculation of the new external force $\mathbf{f}^{ext'} = \mathbf{a}'$ using (3.15). When each external force is negligible or the max iterations number is overcome, the viscosity method terminates.

**ALGORITHM 3:** PROPOSED VISCOSITY METHOD FOR INCOMPRESSIBLE NEWTONIAN FLUIDS

```
1  begin
2      for each particle i of a continuous material in SPH formulation do
3          a'_i = a_i + (μ/ρ_i) Σ_j (m_j/ρ_j)(v_j − v_i)∇²W_ij          --- Eq. (3.17)
4      end
5      max_acc = max{ ‖a'_1‖ , ··· , ‖a'_N‖ }
6      cur_iter = 0
7      while (max_acc > threshold) ∧ (cur_iter < max_iter) do
8          for each particle i do
9              Δv_i = (Δt · a'_i)/(1 + Δt · (μ/ρ_i) Σ_j (m_j/ρ_j) ∇²W_ij)   --- Eq. (3.13)
10             v_i = v_i + Δv_i
11         end
12         for each particle i do
13             a'_i = (μ/ρ_i) Σ_j (m_j/ρ_j)(Δv_j) ∇²W_ij                  --- Eq. (3.15)
14         end
15         max_acc = max{ ‖a'_1‖ , ··· , ‖a'_N‖ }
16         cur_iter = cur_iter + 1
17     end
18 end
```

## 3.4 Other components for the SPH-based modelling

### 3.4.1 Kernel functions

The smoothing kernel function $W$ is crucial for the accuracy of the SPH-based continuum mechanic simulation. As mentioned, there is no necessary condition for the kernel except that

$$\int_{-\infty}^{\infty} W(\mathbf{x}-\mathbf{x}', h)\,\mathrm{d}\mathbf{x}' = 1, \tag{3.19}$$

or the discretized approximated version

$$\sum_j \frac{m_j}{\rho_j} W(\mathbf{x}_i - \mathbf{x}_j, h) \simeq 1, \tag{3.20}$$

for the center position $\mathbf{x}_i$ of the kernel. Its gradient function is given by

$$\nabla_i W(\mathbf{x}_i - \mathbf{x}_j, h) = \frac{\partial W}{\partial q}\nabla_i q, \tag{3.21}$$

while its Laplace function is

$$\nabla_i^2 W(\mathbf{x}_i - \mathbf{x}_j, h) = \frac{\partial^2 W}{\partial q^2}\left\|\nabla_i q\right\|^2 + \frac{\partial W}{\partial q}\nabla_i^2 q, \tag{3.22}$$

where $q = \dfrac{\|\mathbf{x}_i - \mathbf{x}_j\|}{h}$, $\nabla_i q = \dfrac{\mathbf{x}_i - \mathbf{x}_j}{\|\mathbf{x}_i - \mathbf{x}_j\|\,h}$, and $\nabla_i^2 q = \dfrac{1}{\|\mathbf{x}_i - \mathbf{x}_j\|\,h} - \dfrac{\sqrt{\|\mathbf{x}_i - \mathbf{x}_j\|}}{h}$.

Usually, the radius of the kernel domain $h$ is set to twice the default particle spacing, that is, four times longer than the particle radius for maintaining sufficient but not too much neighbour particles. The following Gaussian kernel is commonly used

$$W(\mathbf{x}_i - \mathbf{x}_j, h) = \frac{1}{(2\pi h^2)^{\frac{r}{2}}} \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2h^2}\right). \tag{3.23}$$

Even though a Gaussian kernel is highly recommended statistically, the computational cost for evaluating the exponential function is expensive, and it does not have compact support, that is, the range spans $-\infty$ to $\infty$. Therefore, an approximated cubic spline kernel function is employed in this work

$$W(\mathbf{x}_i - \mathbf{x}_j, h) = \frac{1}{\sigma} \begin{cases} \frac{4}{3} - 2q^2 + q^3 & \text{if } 0 \leq q < 1 \\ \frac{1}{3}(2-q)^3 & \text{if } 1 \leq q < 2 \\ 0 & \text{otherwise,} \end{cases} \tag{3.24}$$

where $q = \dfrac{\|\mathbf{x}_i - \mathbf{x}_j\|}{h/2}$, $\sigma$ is a normalization constant equal to $\{2h, \pi h^2, \frac{4}{3}\pi h^3\}$ for $r = \{1, 2, 3\}$, respectively [203, 205]. A particular kernel to handle viscosity effect, whose Laplacian is always non-negative, $\nabla^2 W \geq 0$, so that the viscosity force does not increase the relative velocity and avoid to create unstable status, was introduced in [151].

An additional method to limit the computational burden concerning the kernel function is to use a look-up table [23]. The kernel values are precomputed for sample distances, called keys, between two particles with a specific span. The keys and the computed values are saved into the look-up table. Using the closest key to the requested one or calculating the linear interpolation using two narrowest points, finding out the desired approximated value of the kernel function is possible.

A cubic spline kernel function has been employed within the performed simulations, while its Laplacian value is set to zero if it is negative. Furthermore, when convenient, the look-up tables storing the kernel value, its gradient value, and its Laplacian value of the kernel domain, $h = 0.005$ m and $h = 0.010$ m with a span $10^{-7}$ m, have been exploited in some simulations using the particle spacing $1.25 \cdot 10^{-3}$ m or $02.5 \cdot 10^{-3}$ m, respectively.

### 3.4.2 Incompressible fluid

Particle-based continuum mechanic methods like SPH have incompressibility problem, that is, they need some special treatments for modelling incompressible continuous materials. To address the incompressibility problem, various methods, like the PSPH [72], the ISPH [279], the WCSPH [21], the PCISPH [292], and the PBFM [185] were proposed. Recently, methods like the IISPH [136] and the DFSPH [22] have shown outstanding performance in addressing incompressible fluid. The IISPH approach tries to find out a proper pressure $p$ so that it satisfies the derivative of the density $\frac{\mathrm{D}\rho}{\mathrm{D}t} = 0$ using the continuity equation $\frac{\mathrm{D}\rho}{\mathrm{D}t} = -\rho\nabla \cdot \mathbf{v}$ and the velocity update equation $\mathbf{v}(t + \Delta t) = \mathbf{v}(t) + \Delta t \frac{\mathbf{f}^{adv} + \mathbf{f}^p}{m}$, where $\mathbf{f}^{adv} \in \mathbb{R}^r$ is an advection force and $\mathbf{f}^p \in \mathbb{R}^r$ is a pressure force computed as

$$\mathbf{f}_i^p = -m_i \sum_j m_j \left( \frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) \nabla W_{ij}. \tag{3.25}$$

The DFSPH method, instead, does not find out the pressure directly but it retrieves an impulse pressure force defined as

$$\mathbf{f}_i^p = \kappa_i^v \sum_j m_j \nabla W_{ij}, \tag{3.26}$$

where $\kappa^v$ is the stiffness parameter, so that it satisfies $\frac{\mathrm{D}\rho_i}{\mathrm{D}t} + \Delta\frac{\mathrm{D}\rho_i}{\mathrm{D}t} = 0$, where $\Delta\frac{\mathrm{D}\rho_i}{\mathrm{D}t} = \Delta t \sum_j m_j \left(\frac{\mathbf{f}_i^p}{\rho_i} - \frac{\mathbf{f}_{j\leftarrow i}^p}{\rho_i}\right)\nabla W_{ij}$.

In the proposed simulations dealing with viscosity, the IISPH method has been employed: nevertheless, other incompressibility approaches are doable. As a contribution, instead of the pressure force equation (3.25) in the original IISPH method, the one from [209] has been employed

$$\mathbf{f}_i^p = -\frac{m_i}{\rho_i} \sum_j \frac{m_j}{\rho_j} \left(\frac{p_i + p_j}{2}\right) \nabla W_{ij}, \tag{3.27}$$

since more stable performance are obtained within the carried out simuatlions.

## 3.5 Simulations

The set-up employed to test the proposed method is composed of an IntelCore i7-6500U CPU@2.50 GHz, 8.0 Gb of memory with Windows 10 x64 OS, equipped with MFC of Microsoft, Eigen[1], and OpenCV[2] for 2D graphic or Open Scene Graph[3] for 3D graphics libraries based on C++ programming language. Houdini software from SideFX[4] has been employed to reconstruct the mesh from particles, and Blender[5] for the graphical rendering. A CUDA version has been implemented to compare both the proposed algorithm and the existing conventional viscosity methods. They have been tested on an NVIDIA GeForce 940MX graphic card. It has been possible to verify that CUDA implementation has sped up the execution from the 20% to the 50% compared to OpenMP-based[6] implementation.

[1] https://eigen.tuxfamily.org/

[2] https://opencv.org

[3] http://www.openscenegraph.org

[4] https://www.sidefx.com

[5] https://www.blender.org

[6] https://www.openmp.org

### 3.5.1 Accuracy and time analysis

The accuracy of the proposed algorithm is firstly addressed. A free-fall experiment is considered, where the velocity of the particles of the continuous material is measured after applying a uniform external force like gravity. Because there is no degradation for the gravity during the free-fall, the movement of the particles is the same as the product of the acceleration with the delay time, $\Delta t\mathbf{a}$. The error introduced by the proposed viscosity method is due to the threshold $\eta$, the viscosity $\mu$, and $\mathbf{a}$. Two values are considered for the external force, namely $\mathbf{a} = 10$ m/s$^2$ and $\mathbf{a} = 100$ m/s$^2$. The velocity of the continuous materials is measured after 0.001 s. The particle spacing and the density $\rho$ of the SPH particles are set as 0.0025 m and $\rho = 100$ kg/m$^3$, respectively. The results are summarized in Table 3.2 and Table 3.3.

| | $\eta = 0.01$ | 0.1 | 0.2 | 0.4 | 0.6 | 0.8 | 1.0 |
|---|---|---|---|---|---|---|---|
| $\mu = 0.1$ | 0.01 | 0.05 | 0.05 | 0.50 | 0.50 | 0.50 | 0.50 |
| 1 | 0.02 | 0.20 | 0.34 | 0.98 | 1.69 | 1.69 | 2.95 |
| 10 | 0.05 | 0.56 | 1.16 | 2.40 | 3.69 | 5.02 | 6.05 |
| 100 | 0.09 | 0.92 | 1.83 | 3.66 | 5.49 | 7.30 | 9.14 |
| 200 | 0.10 | 0.96 | 1.91 | 3.82 | 5.73 | 7.65 | 9.54 |
| 300 | 0.10 | 0.97 | 1.94 | 3.88 | 5.82 | 7.76 | 9.71 |
| 400 | 0.10 | 0.98 | 1.96 | 3.91 | 5.86 | 7.82 | 9.78 |
| 500 | 0.10 | 0.98 | 1.96 | 3.93 | 5.89 | 7.85 | 9.82 |

Table 3.2: Error rates (%) of free-fall experiment with $\mathbf{a} = 10$ m/s$^2$ and various viscosity $\mu$ and threshold $\eta$ at time 0.001 s in 3D space.

| | $\eta = 0.1$ | 1.0 | 2.0 | 4.0 | 6.0 | 8.0 | 10.0 |
|---|---|---|---|---|---|---|---|
| $\mu = 0.1$ | 0.01 | 0.05 | 0.05 | 0.50 | 0.50 | 0.50 | 0.50 |
| 1 | 0.02 | 0.20 | 0.34 | 0.98 | 1.69 | 1.69 | 2.95 |
| 10 | 0.05 | 0.56 | 1.16 | 2.40 | 3.69 | 5.02 | 6.05 |
| 100 | 0.09 | 0.92 | 1.83 | 3.66 | 5.49 | 7.30 | 9.14 |
| 200 | 0.10 | 0.96 | 1.91 | 3.82 | 5.73 | 7.65 | 9.54 |
| 300 | 0.10 | 0.97 | 1.94 | 3.88 | 5.82 | 7.76 | 9.71 |
| 400 | 0.10 | 0.98 | 1.96 | 3.91 | 5.86 | 7.82 | 9.78 |
| 500 | 0.10 | 0.98 | 1.96 | 3.93 | 5.89 | 7.85 | 9.82 |

Table 3.3: Error rates (%) of free-fall experiment with $\mathbf{a} = 100$ m/s$^2$ and various viscosity $\mu$ and threshold $\eta$ at time 0.001 s in 3D space.

Based on these data, the graph in Fig. 3.3 is depicted with a threshold rate, $TR = \dfrac{\eta}{\|\mathbf{a}\|}$. From the data, it is possible to appreciate that, as long as the

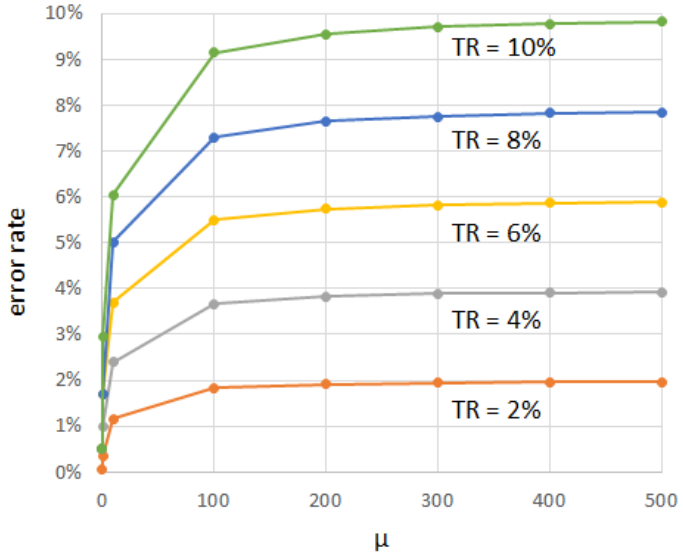



Fig. 3.3: The relation between $\mu$ and error rate with various threshold rates, $TR = \dfrac{\eta}{\|\mathbf{a}\|}$.

viscosity coefficient increases, the error rate, $ERR = \dfrac{\|\mathbf{a}\| - \|\mathbf{a}'\|}{\|\mathbf{a}\|}$, where $\mathbf{a}$ is the input acceleration and $\mathbf{a}'$ is the measured one, converges to the threshold rate given by the external force, that is,

$$ERR \leq \frac{\eta}{\|\mathbf{a}\|}. \tag{3.28}$$

Besides, it is possible to estimate the execution time of the proposed viscosity method, which is relative to the number of iterations, $ITER$, because the time of each iteration can be considered as a constant. The number of iterations are measured from the previous simulations: the results are summarized in Table 3.4 and Table 3.5.

We used the threshold rate, $TR = \dfrac{\eta}{\|\mathbf{a}\|}$, to display the graphical relation in Fig. 3.4. Furthermore, we display the gradient of each relation equation at each threshold rate, $TR = \dfrac{\eta}{\|\mathbf{a}\|}$, in Fig. 3.5.

Here, we derived the following equation between them

$$ITER \simeq -A \ln\left(\frac{\eta}{\|\mathbf{a}\|}\right) \cdot \mu, \tag{3.29}$$

with the coefficients $A = 1.646$ in this experiment.

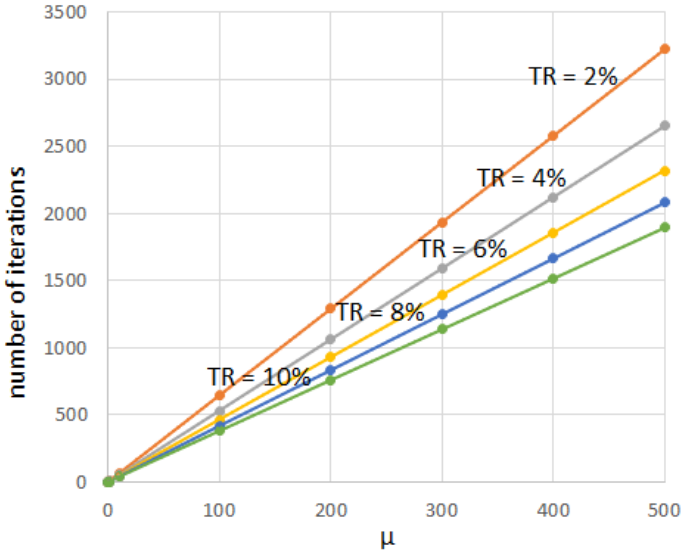



Fig. 3.4: The relation between $\mu$ and the number of iterations with various threshold rates, $TR = \frac{\eta}{\|\mathbf{a}\|}$.

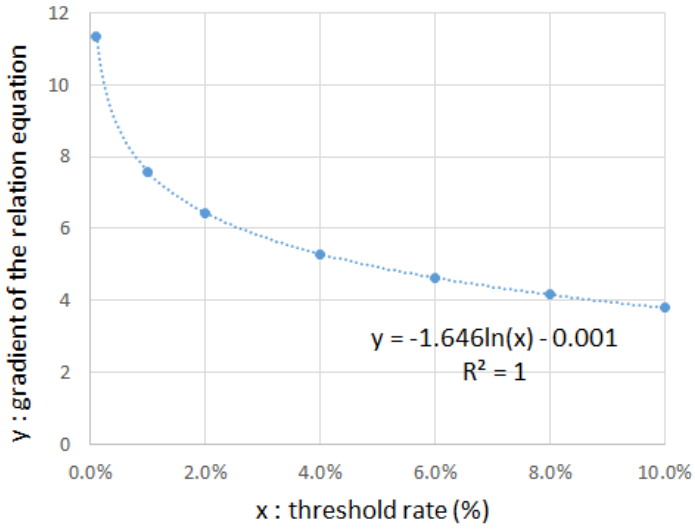



Fig. 3.5: The relation between threshold rates, $TR = \frac{\eta}{\|\mathbf{a}\|}$, and the gradients of the relation equations in Fig. 3.4.

| | $\eta = 0.01$ | 0.1 | 0.2 | 0.4 | 0.6 | 0.8 | 1.0 |
|---|---|---|---|---|---|---|---|
| $\mu = 0.1$ | 4 | 3 | 3 | 2 | 2 | 2 | 2 |
| 1 | 15 | 10 | 9 | 7 | 6 | 6 | 5 |
| 10 | 118 | 79 | 67 | 55 | 48 | 43 | 40 |
| 100 | 1142 | 761 | 647 | 532 | 465 | 418 | 381 |
| 200 | 2278 | 1519 | 1290 | 1062 | 928 | 833 | 760 |
| 300 | 3415 | 2277 | 1934 | 1591 | 1391 | 1249 | 1138 |
| 400 | 4551 | 3034 | 2577 | 2121 | 1854 | 1664 | 1517 |
| 500 | 5688 | 3792 | 3221 | 2650 | 2317 | 2080 | 1896 |

Table 3.4: The number of iterations of the free-fall simulation with $\mathbf{a} = 10$ m/s$^2$ and various viscosity $\mu$ and threshold $\eta$ at time 0.001 s in 3D space.

| | $\eta = 0.1$ | 1.0 | 2.0 | 4.0 | 6.0 | 8.0 | 10.0 |
|---|---|---|---|---|---|---|---|
| $\mu = 0.1$ | 4 | 3 | 3 | 2 | 2 | 2 | 2 |
| 1 | 15 | 10 | 9 | 7 | 6 | 6 | 5 |
| 10 | 118 | 79 | 67 | 55 | 48 | 43 | 40 |
| 100 | 1142 | 761 | 647 | 532 | 465 | 418 | 381 |
| 200 | 2278 | 1519 | 1290 | 1062 | 928 | 833 | 760 |
| 300 | 3415 | 2277 | 1934 | 1591 | 1391 | 1249 | 1138 |
| 400 | 4551 | 3034 | 2577 | 2121 | 1854 | 1664 | 1517 |
| 500 | 5688 | 3792 | 3221 | 2650 | 2317 | 2080 | 1896 |

Table 3.5: The number of iterations of the free-fall simulation with $\mathbf{a} = 100$ m/s$^2$ and various viscosity $\mu$ and threshold $\eta$ at time 0.001 s in 3D space.

Based on these results, it is possible to conclude that the proposed viscosity method has a predictable accuracy and computation time.

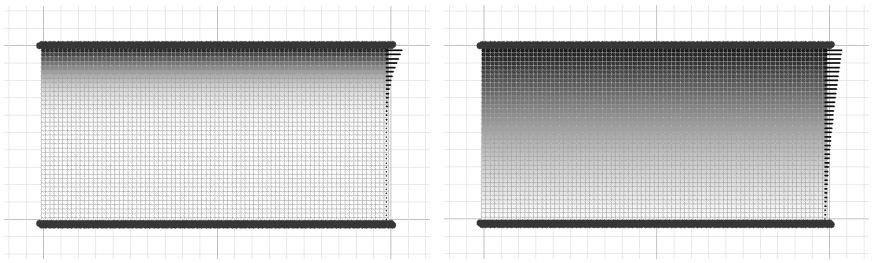
### 3.5.2 Couette flow experiment

The Couette flow appears within a viscous fluid between two parallel plates. The upper plate is moving with a constant velocity while the lower is stationary: because of the difference of the velocities between the two plates, the speed of each layer of the flow is different. As time goes to infinity, the gradient of the velocity to the vertical coordinate, $\dfrac{\partial \mathbf{v}_x}{\partial y}$, is constant, where $y$ is the axis orthogonal to the fluid movement that is along the $x$ direction. The viscosity of the fluid affects how fast the gradient becomes constant. The following equation is an analytic solution for the Couette flow [208]:

$$\mathbf{v}_x(y,t) = \mathbf{v}_0 \frac{y}{y_0} + \sum_{n=1}^{\infty} \frac{2\,\mathbf{v}_0}{n\pi}(-1)^n \sin\left(\frac{n\pi}{y_0} y\right) \exp\left(-\nu \left(\frac{n\pi}{y_0}\right)^2 t\right), \quad (3.30)$$

where $\nu = \dfrac{\mu}{\rho}$ is the kinematic viscosity, $\mathbf{v}_0 \in \mathbb{R}^r$ is the velocity of the top flow, and $y_0 \in \mathbb{R}$ is the height of the top flow. By comparing the analytic solution with the experimental results, it is possible to verify the accuracy of the viscosity algorithm.

The carried out simulation about the Couette flow has height $y_0 = 0.1$ m, velocity of the upper plate $\mathbf{v}_0 = 1$ m/s, and time $t0.01$ s, The parameters relative to the SPH formulation are: $\rho = 100$ for the fluid density, the particle spacing is equal to $1.25 \cdot 10^{-3}$ m, and the time-step is $\Delta t = 0.1 \cdot 10^{-3}$ s.



(a) (b)

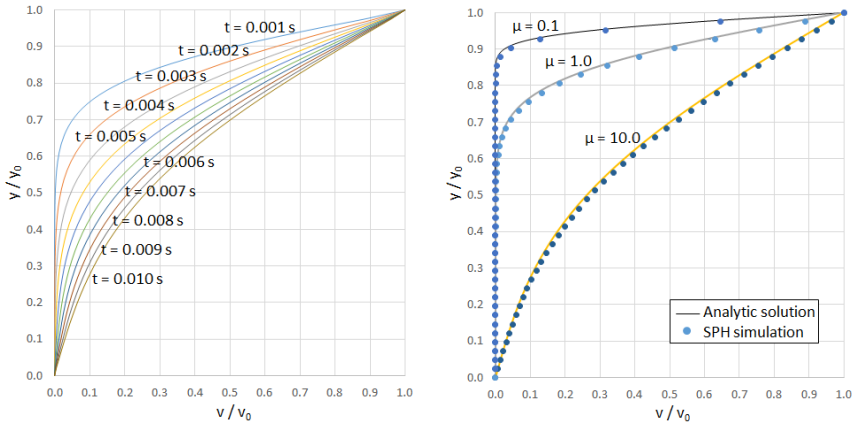
Fig. 3.6: Couette flow simulation with the gap between two plates $y_0 = 0.1$ m, the velocity of the upper plate $\mathbf{v}_0 = 1.0$ m/s, the total simulation time of 0.01 s, the density of the fluid $\rho = 100$, the particle spacing $1.25 \cdot 10^{-3}$ m, time-step $\Delta t = 0.1 \cdot 10^{-3}$ s, and dynamic viscosity coefficients : (a) $\mu = 0.1$, Re = 25, (b) $\mu = 1.0$, Re = 2.5. The right side bar indicates the magnitude of the velocity of each layer.

The results of two flows with $\mu = 0.1$ and $\mu = 1.0$ after $t = 0.01$ s are depicted in Fig. 3.6, in which the gray colour shows the velocity magnitudes of each layer in the fluid and lines in the right side. Reynolds number for Couette flow is defined [84] as

$$\mathrm{Re} := \frac{\rho\,\mathbf{v}_0\,\frac{1}{2}y_0}{\mu}. \quad (3.31)$$

Hence, Reynolds numbers for $\mu = 0.1$ and $\mu = 1.0$ are Re = 25 and Re = 2.5, respectively.

The results are more easily collected in Fig. 3.7. It is possible to verify that the top and bottom fluid layers' speeds are almost identical to the velocity of the upper and lower plates, respectively. The curves in Fig. 3.7-(a) indicate the velocities of each fluid layer with time $t$: as time goes, the gradient of




(a) (b)

Fig. 3.7: Couette flow experiment. (a) The results from time 0.001 s to 0.01 s with viscosity coefficient $\mu = 10$. (b) The results from time 0.01 s with various viscosity coefficients, namely: $\mu = 0.1$, $\mu = 1.0$, and $\mu = 10.0$. Lines and dots indicate the analytic solution and SPH simulation results, respectively.

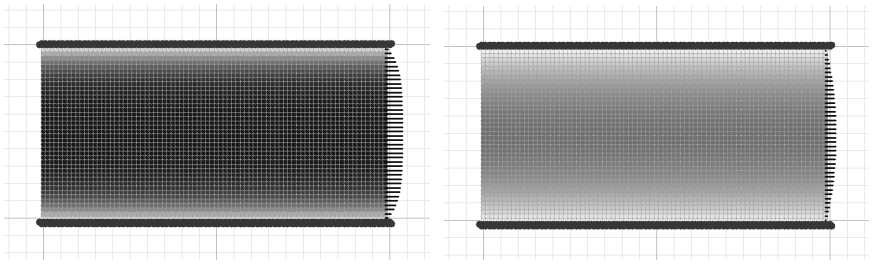
the velocity equation becomes constant. The velocity results of the proposed algorithm has been compared with the analytic solution of equation (3.30) in Fig. 3.7-(b) for three different viscosity fluids, namely $\mu = 0.1$, $\mu = 1.0$, and $\mu = 10.0$. The velocity decreases smoothly as the height decreases in the higher viscosity fluid than the lower one. Hence, the proposed algorithm has almost the same results as the analytic solutions.

### 3.5.3 Poiseuille plane flow experiment

Next simulation test deals with the Poiseuille plate flow [208, 342]. Like the Couette flow, there are two parallel plates with distance $y_0$, but both of them are stationary. A constant force is applied into the fluid between the two plates. Depending on how strong the viscosity of the fluid is, the velocities of each layer of the fluid are determined at a certain time. The following equation is the analytic solution for the Poiseuille plate flow [208]

$$\mathbf{v}_x(y,t) = -\frac{\mathbf{a}}{2\nu} y(y-y_0) - \sum_{n=0}^{\infty} \frac{4\,\mathbf{a}\,y_0^2}{\nu\,\pi^3(2n+1)^3} \sin\left(\frac{(2n+1)\pi}{y_0} y\right) \exp\left(-\nu\left(\frac{(2n+1)\pi}{y_0}\right)^2 t\right).$$

The simulation test is carried out with height $y_0 = 0.1$ m, acceleration $\mathbf{a} = 10.0$ m/s$^2$, time $t = 0.1$ s, and the SPH relative parameters equal to $\rho = 100$, the particle spacing equal to $1.25 \cdot 10^{-3}$ m, and a time-step $\Delta t = 0.0001$ s.



(a) (b)

Fig. 3.8: Poiseuille plane flow experiment with the distance between two plates of $y_0 = 0.1$ m, a fluid acceleration of $\mathbf{a} = 10.0$ m/s$^2$, the total simulation time of $t = 0.1$ s, $\rho = 100$, the particle spacing equal to $1.25 \cdot 10^{-3}$ m, the time-step $\Delta t = 0.1 \cdot 10^{-3}$ s, and dynamic viscosity coefficients equal to, namely, (a) $\mu = 0.1$, Re $= 33.3$ (b) $\mu = 1.0$, Re $= 3.3$. The right side bar indicates the magnitude of the velocity of each layer.

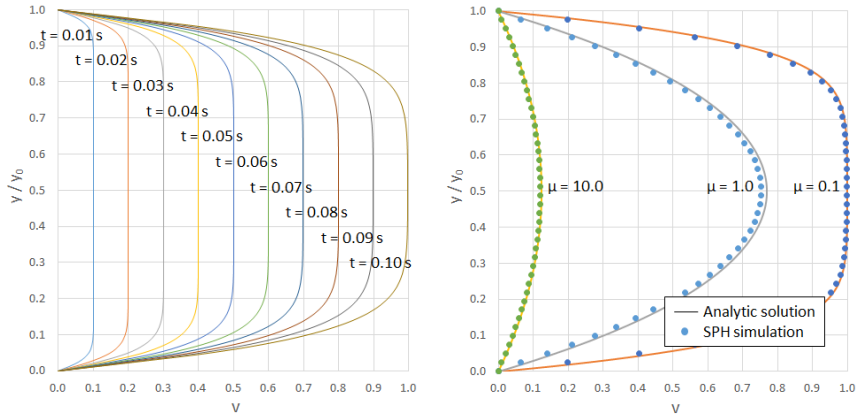
The experimental results of the two flows with $\mu = 0.1$ and $\mu = 1.0$ are shown in Fig. 3.8. Reynolds number for Poiseuille plane flow is defined as [84]

$$\text{Re} := \frac{\rho\, \overline{\mathbf{v}}\, y_0}{\mu}, \tag{3.32}$$

where $\overline{\mathbf{v}}$ is the average velocity. Another definition of Reynolds number is also used, $\text{Re} := \dfrac{\rho\, \mathbf{v}_0\, \frac{1}{2} y_0}{\mu}$, where $\mathbf{v}_0$ is the velocity at the mid-plane of the channel [317]. The average velocity for Poiseuille plane flow is $\overline{\mathbf{v}} = \frac{2}{3}\mathbf{v}_0$. Hence, the Reynolds numbers for $\mu = 0.1$ and $\mu = 1.0$ are Re $= 33.3$ and Re $= 3.3$, respectively.

Because the upper and lower plates are stationary, the speeds of the top and bottom fluid layer are zero, while the speed of the center fluid is maximum. The average velocity of the fluid is faster if $\mu = 0.1$, rather than in case of $\mu = 1.0$.

The results with three different viscosity fluids $\mu = 0.1$, $\mu = 1.0$, and $\mu = 10.0$ are represented in Fig. 3.9. Since the acceleration is $\mathbf{a} = 10.0$ m/s$^2$, and the time $t = 0.1$ s, the maximum speed is 1 m/s. It is possible to appreciate that the higher the viscosity of the fluid, the lower the maximum reached speed. Simulation results are verified to be similar to the analytic solutions.




(a) (b)

Fig. 3.9: Poiseuille flow experiment. (a) The results from time 0.01 to 0.10 s with viscosity coefficient $\mu = 0.1$. (b) The results from time 0.1 s with various viscosity coefficients, namely, $\mu = 0.1$, $\mu = 1.0$, and $\mu = 10.0$. Lines and dots indicates the analytic solution and SPH simulation results, respectively.

### 3.5.4 Comparison with conventional viscosity methods

A comparison with the existing viscosity approaches mentioned in Section 3.1 is carried out to prove the efficiency of the proposed method. The Couette flow experimentation presented in Section 3.5.2 is chosen as a test-bed for such a comparison. It is worth remarking that the comparison results of the viscosity methods might differ depending on what kind of test-bed is used.

The simulations are configured with the height and the velocity of the top plate as $y_0 = 0.1$ m and $v_0 = 1$ m/s, respectively. The SPH particles for the incompressible Newtonian fluid have particle spacing equal to $1.25 \cdot 10^{-3}$ m and rest density $\rho_0 = 100$. Then, the velocity of each particle along the vertical direction is measured at time 0.01 s and compared with the analytic solution for a fluid of viscosity 1 using an RMSE method. Because each viscosity method has its parameters, it is impossible to use the same numbers for them. Hence, the best parameters matching the analytic solution have been selected: such a best matching induces longer time-step and less RMSE.

It is worth saying that, within the conventional methods, adjusting the parameters has frequently caused explosion phenomena of the fluid. This can be explained by the fact that too close particles have a strong repulsive force. Therefore, they move very fast in opposite directions. Usually, a fast movement triggers subsequent explosions of nearby particles. Within the compari-

son experiments, the parameters for which an explosion occurs are neglected. The best settings corresponding to stable behaviour are thus selected.

The obtained results are summarized in Table 3.6 and their velocity curves are shown in Fig. 3.10 and Fig. 3.11. In these figures, the gray solid lines with dynamic viscosity coefficients $\mu$ indicate the reference viscosity curve calculated by the analytic solution in (3.30).

| Method | max $\Delta t$ (s) | iterations | real time / simul time | RMSE | parameters |
|---|---|---|---|---|---|
| [203] | 0.00001 | 1000 | 6209.3 | 0.026075 | $\alpha = 0.01$, $\beta = 0$, $c = 1484$ |
| [208] | 0.0001 | 100 | 635.9 | 0.006468 | $\mu = 0.4452$ |
| [279] | 0.0001 | 100 | 706.2 | 0.006747 | $\mu = 0.4452$ |
| [6] | 0.0001 | 100 | 710.8 | 0.082348 | $\nu_0 = 0.000005$, $k = 0$ |
| [307] | 0.001 | 44 | 223.4 | 0.009944 | $\mu = 1.2$, PCG tol. = 1% |
| [241] | 0.001 | 434 | 81.2 | 0.072388 | $\xi = 0.9$, PCG tol. = 1% |
| [23] | 0.0001 | 265686 | 37863.9 | 0.155345 | $\gamma = 0.89$, $\eta^{visco} = 0.1$ |
| PM | 0.001 | 259 | 90.6 | 0.007486 | $\mu = 0.88$, $\eta = 0.1$ |
| PM | 0.001 | 278 | 92.4 | 0.012060 | $\mu = 1.00$, $\eta = 0.1$ |

Table 3.6: Comparison of the proposed viscosity method with explicit / implicit methods. Couette flow experiments [208] for simulating a physical fluid of dynamic viscosity 1 at simulation time 0.01 s, particle spacing 0.00125 m, and flow rest density $\rho_0 = 100$. No optimization like parallelization or precompiled kernel. PM stands for proposed method.

In the following, conventional explicit and implicit viscosity methods are compared separately.

The first four methods shown in Table 3.6 are explicit viscosity methods. The viscosity force is calculated according to the related viscosity equations at each time step $\Delta t$. Therefore the total iteration number is equal to $t/\Delta t$. The classic viscosity equation proposed in [203] requires the shortest time step, $\Delta t = 0.1 \cdot 10^{-4}$ s. Both of the viscosity equations proposed in [208] and in [279] produce very accurate results. The viscosity equation proposed in [6] generates a zigzag velocity curve. Such a zigzag velocity curve occurs because the calculation of the shear stress and the calculation of its derivative are separated so that the particle updated by the shear stress is different from the particle updated by the viscosity force. However, the overall pattern of the velocity curve is similar to the one of the analytic solution.

The following three viscosity methods shown in Table 3.6 are categorized into implicit viscosity methods. They update their velocities iteratively so that the relation equation about viscosity for all the fluid particles is solved. Therefore, viscosity diffusion can occur within a time step, and these implicit viscosity methods usually work well with relatively long time steps. In the comparison experiment, the viscosity algorithm proposed in [307] has recorded the fewest iterations among the conventional methods. The viscosity algorithm proposed in [241] has finished the simulation with the shortest running time, while the velocity curve is somewhat different from the one of

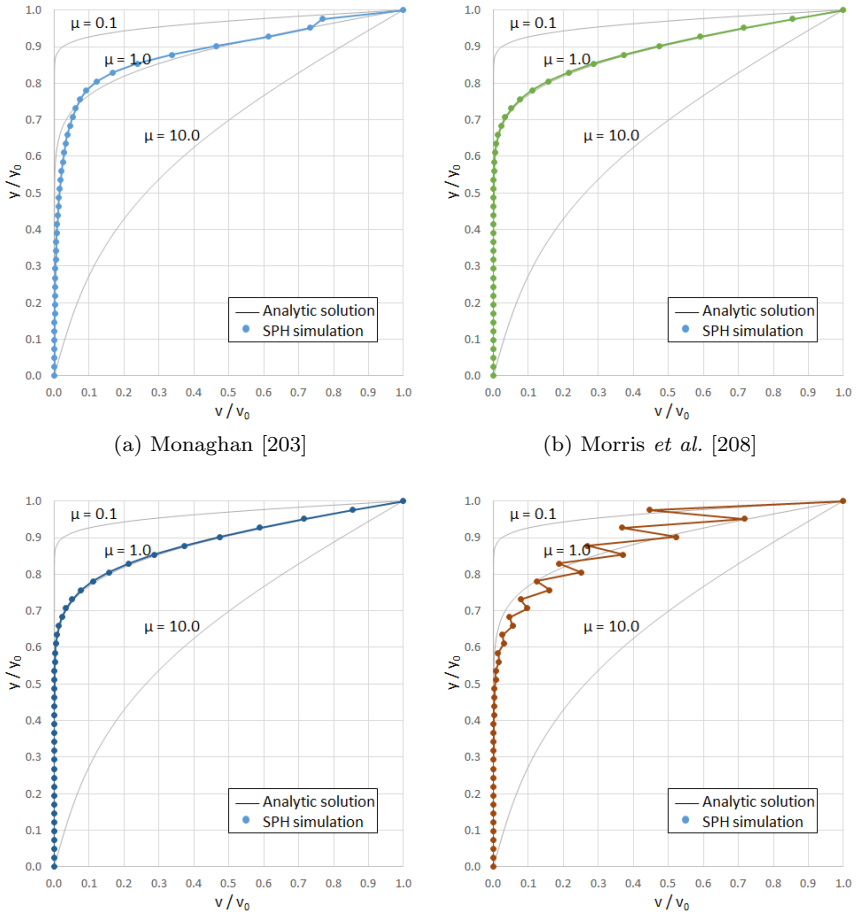
the analytic solution. The viscosity algorithm in [23] has also shown a zigzag curve, with reasons similar to the ones mentioned above.

The method proposed in this paper can work with a long time-step $\Delta t = 10^{-3}$ s. The time ratio between the real-time to simulation time of the proposed approach ranked second among the compared viscosity methods. Even though the best time performance is recorded by the viscosity method proposed in [241], the proposed method has higher accuracy. Moreover, it has been appreciated that the proposed method is more stable when there is a change in the viscosity parameter. Based on these results, it is possible to affirm that the proposed viscosity is comparable to the existing viscosity methods regarding accuracy, speed, and stability.

### 3.5.5 Additional simulations

Two further simulations are presented. The former is about free-falls of several viscous fluids (water, honey, ketchup, and shortening) over a bunny sculpture. The dynamic viscosity coefficients of these fluids are $\mu = 8.94 \times 10^{-4}$, $2 - 10$, $50 - 100$, and $\approx 250$, respectively. The following parameters have been considered to simulate these fluids, having in mind the above range $\mu = 0.0,\ 5.0,\ 70.0$, and $250.0$. The simulation scenes for each case study are shown in Fig. 3.12. Because the time gap between the viscous fluids is big, *i.e.*, the movement of shortening is languid compared to other fluids, the best scene for each liquid is illustrated. With the help of additional graphical treatment, it is possible to verify how the simulated case is close to the real one.

The latter simulation is about the stretching of a bread dough. Doughs are a non-Newtonian fluid, and they have complex properties including viscosity, elasticity, and plasticity [201, 202, 291]. Hence, the proposed viscosity method for Newtonian fluids does not perfectly match the simulation of a dough, but it can approximately simulate its behaviour. Taking data from physical experiments, the shear stress of general a dough has a relation equation $\tau = 298.76 + 5177\,\dot{\gamma}^{0.417}$ [202], where $\dot{\gamma}$ is the strain rate. The dynamic viscosity coefficient $\mu = 2500.0$ has been selected for a simulation with 6000 particles with spacing $1.25 \cdot 10^{-3}$ m. Some results are shown in Fig. 3.13 at time 0 s, 1.5 s, and 3 s, respectively. The dough is pushed, stretched, and torn: the simulated behaviours are similar to a real dough.




(a) Monaghan [203]

(b) Morris *et al.* [208]

(c) Shao & Lo [279]

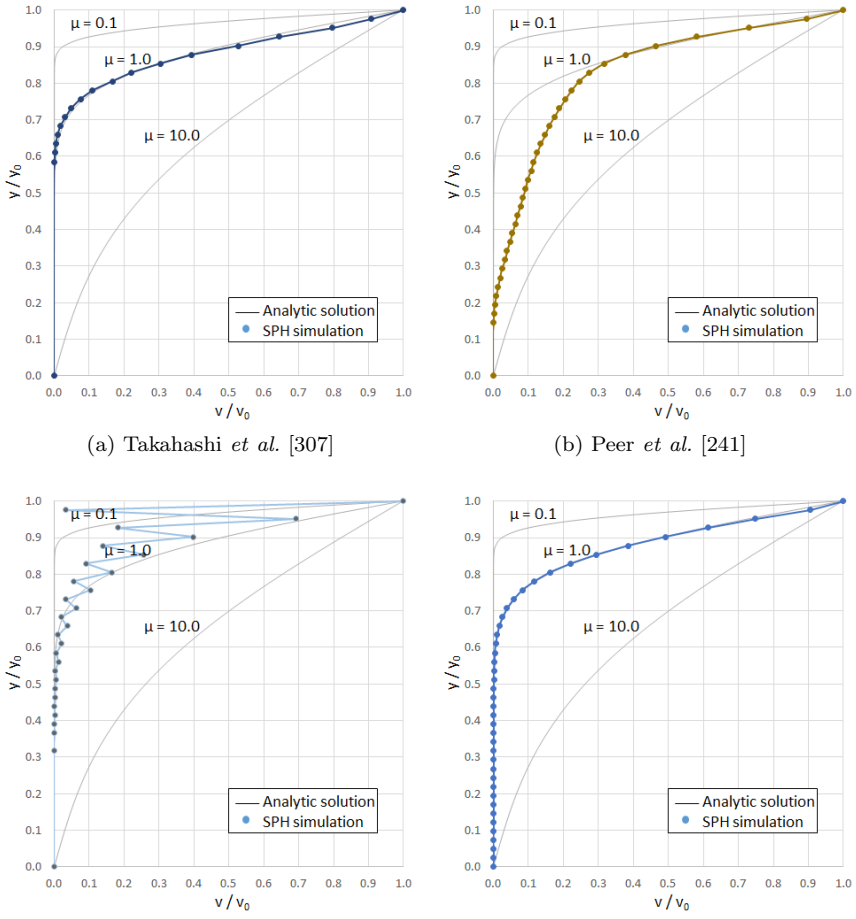
(d) Andrade *et al.* [6]

Fig. 3.10: Couette flow experiments with various conventional viscosity methods and the proposed one for simulating a physical fluid of viscosity 1 at simulation time 0.01 s, particle spacing of $1.25 \cdot 10^{-3}$ m, and flow rest density $\rho_0 = 100$. The relative data are shown in Table 3.6.

## 3.6 Discussion and conclusion

A novel implicit viscosity method based on the SPH formulation for incompressible Newtonian fluids has been proposed. Such formulation can simulate wide-ranged viscosity behaviours in continuous materials.

Concerning the contributions, the proposed method exhibits a shorter computation time; it employs a physical viscosity coefficient, which helps




(a) Takahashi *et al.* [307]

(b) Peer *et al.* [241]

(c) Bender & Koschier [23]

(d) Proposed method

Fig. 3.11: Couette flow experiments with various conventional viscosity methods and the proposed one for simulating a physical fluid of viscosity 1.0 at simulation time $t = 0.01\,s$, particle spacing $r = 0.00125\,m$, and flow rest density $\rho_0 = 100$. Relative data are shown in table 3.6.

predict the viscous behaviour of the continuous material more accurately. The carried out simulations show that the computational time is predictable. Finally, the accuracy is similar or higher than conventional viscosity methods in modelling continuous and viscous materials. Additionally, some graphical simulations show that the approach might be suitable for graphics animations.

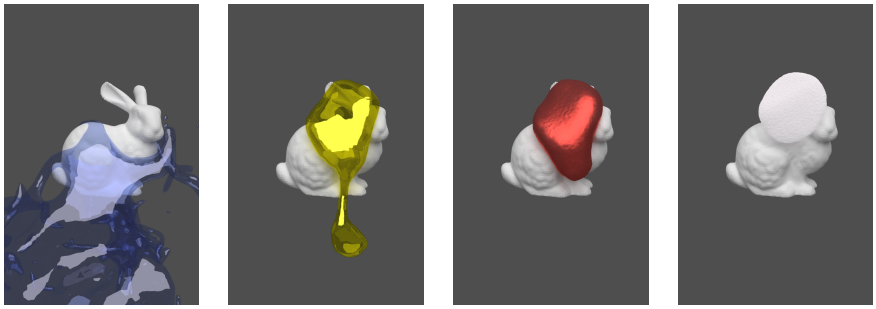


Fig. 3.12: Simulations of viscous fluids: water $\mu = 8.94 \times 10^{-4}$ (0.0), honey $\mu = 2-10$ (5.0), ketchup $\mu = 50-100$ (70.0), and shortening $\mu \approx 250$ (250.0). The numbers indicate physical dynamic viscosity coefficients and the numbers in parenthesis are the coefficients for these simulations.

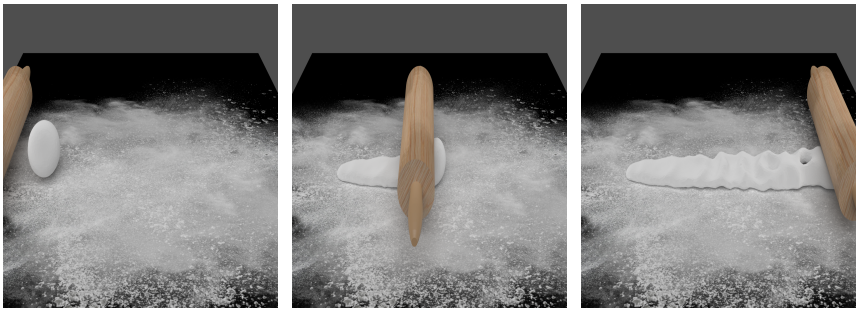


Fig. 3.13: Simulation of stretching of a bread dough with the dynamic viscosity coefficient $\mu = 2500.0$ at simulation time (left) 0 s, (middle) 1.5 s, and (right) 3 s.