



Semi-implicit operator splitting for the simulation of Herschel–Bulkley flows with smoothed particle hydrodynamics

Chang Yoon Park¹ · Tarek I. Zohdi¹

Received: 26 April 2019 / Revised: 27 June 2019 / Accepted: 19 November 2019 / Published online: 10 December 2019
© OWZ 2019

Abstract

Smoothed particle hydrodynamics (SPH) has become a popular numerical framework of choice for simulating free-surface flows, mainly for Newtonian fluids. The topic regarding the simulation of non-Newtonian free-surface flows, however, remains relatively untouched due to difficulties regarding the computation of viscous forces. In previous approaches, the viscous forces acting on each SPH particle were computed explicitly. Non-Newtonian fluids such as Herschel–Bulkley fluids, the effective viscosity between yielded and unyielded regions can differ by several orders of magnitudes; imposing severe time step restrictions for the simulation for explicit methods. Numerically, this can be seen as a stiff problem. We propose a semi-implicit time-stepping approach where the viscous forces are computed implicitly, within the context of SPH. We demonstrate the convergence of the method via a simple 2D test case.

Keywords Smoothed particle hydrodynamics · Non-Newtonian fluids · Meshfree methods

1 Introduction

Originally developed for astrophysical simulations, SPH has become a popular numerical method of choice for simulating free-surface Newtonian flow problems [4,5,9]. Naturally, the topic regarding the simulation of non-Newtonian flows with this method has been in question for some time, although successful attempts were limited due to the explicit nature of SPH. Such efforts can be found in previous work such as in Hosseini et al. [7]. Although satisfactory results were obtained for some cases, the effective viscosity values used within the simulation had little variance. For example, for a Herschel–Bulkley type fluid, unyielded regions exhibit viscosity values that are multiple magnitudes larger compared to yielded regions. In our experience, it becomes very difficult to resolve a SPH system with such viscosity variance exclusively with explicit methods, especially with SPH. For this article, we will employ Chorin’s projection method to split the viscous forces and the pressure forces. We then introduce

an iterative process to solve for the viscous forces implicitly for the next time step. A brief convergence study for a simple 2D flow field of a Herschel–Bulkley fluid was given at the end.

2 Problem formulation: smoothed particle hydrodynamics

We first discretize the Navier–Stokes equation via SPH formalism. For each SPH particle, we have

$$\frac{D\mathbf{u}_i}{Dt} = - \left\langle \frac{1}{\rho} \nabla p \right\rangle_i + \left\langle \frac{\mu}{\rho} \nabla^2 \mathbf{u} \right\rangle_i + \mathbf{b}_i \quad (1)$$

$$\frac{D\rho_i}{Dt} = \langle \rho \nabla \cdot \mathbf{u} \rangle_i \quad (2)$$

$$p_i = f_{\text{EOS}}(\rho_i) \quad (3)$$

The physical quantity included within the angles brackets $\langle \cdot \rangle$ represents the physical value discretized via SPH, and f_{EOS} represents the equation of state. Tait’s equation of state [13] was used for this paper. Note that taking such approach results in a weakly-compressible SPH model (WCSPH). The pressure gradient discretization (1) is usually chosen to be [13]:

✉ Chang Yoon Park
changyoonpark@berkeley.edu

Tarek I. Zohdi
zohdi@berkeley.edu

¹ University of California, Berkeley, Etcheverry Hall, Berkeley, CA 94720-1740, USA

$$-\left\langle \frac{1}{\rho} \nabla p \right\rangle_i = - \sum_{j \in \mathcal{N}_i} \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) \nabla_i W_{ij} m_j \quad (4)$$

The index i represents the i th SPH particle, while $j \in \mathcal{N}_i$ represents the set of neighbors of the i th particle. W_{ij} is the kernel function centered around particle i , $\nabla_i W_{ij}$ represents the gradient of the kernel, p_i is the pressure of particle i , p_j is the pressure of particle j , ρ_i is the density of particle i , ρ_j is the density of particle j , m_i and m_j represent the mass of particle i and j , respectively.

We may split the pressure and the viscous contributions via a projection method [3]:

$$\mathbf{u}_i^* = \mathbf{u}_i^n + \left(\frac{\Delta t}{\rho_i} \right) (\mathbf{a}_p)_i \quad (5)$$

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^* + \left(\frac{\Delta t}{\rho_i^*} \right) (\mathbf{a}_v)_i \quad (6)$$

where \mathbf{u}_i^n and \mathbf{u}_i^{n+1} , respectively, represent the velocity of particle i at the n th and $n+1$ th time step, and \mathbf{u}_i^* represents the “intermediate velocity”. The acceleration of each particle due to pressure forces can be computed using (4). We now look into the acceleration contribution for the viscous step ($\mathbf{a}_{a,v}$). Influenced by the viscous operator utilized in Shao and Lo [15], we explicitly expand out the viscous operator. For an explicit version, we will have:

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^n + (\Delta t) \left(\sum_{j \in \mathcal{N}_i} \frac{4m_j (\bar{\mu}_{\text{eff}})_{ij} (\mathbf{x}_i^* - \mathbf{x}_j^*) \cdot \nabla W_{ij}}{(\rho_i^* + \rho_j^*) (\mathbf{x}_i^* - \mathbf{x}_j^*) \cdot (\mathbf{x}_i^* - \mathbf{x}_j^*)} (\mathbf{u}_i^* - \mathbf{u}_j^*) \right) \quad (7)$$

For the average effective viscosity $\bar{\mu}_{\text{eff},ab}$, we pick

$$(\bar{\mu}_{\text{eff}})_{ij} = \frac{(\bar{\mu}_{\text{eff}})_i + (\bar{\mu}_{\text{eff}})_j}{2} \quad (8)$$

Note that there exists different methods for “mixing” the viscosities between particles [4,8].

Later on, we will re-formulate the above via an implicit time-stepping scheme.

The effective viscosity of each particle is computed according to the non-Newtonian fluid model before the viscous step. For a regularized Herschel–Bulkley fluid model (Tanner/milthrope 1983):

$$\mu_{\text{eff}} = \begin{cases} k \dot{\gamma}_o^{n+1} + \tau_o \dot{\gamma}_o^{-1} & \dot{\gamma} < \dot{\gamma}_o \\ k \dot{\gamma}^{n-1} + \tau_o \dot{\gamma}^{-1} & \dot{\gamma} \geq \dot{\gamma}_o \end{cases} \quad (9)$$

With the viscous shear ($\boldsymbol{\tau}_i$) computed using the above effective viscosity being

$$\boldsymbol{\tau} = 2(\mu_{\text{eff}})_i (\dot{\gamma}) \mathbf{D}_i, \quad \mathbf{D}_i = \frac{1}{2} \left(\nabla \mathbf{u}_i^* + \nabla \mathbf{u}_i^{*T} \right) \quad (10)$$

For continuity, the method found in Antuono et al. [1] was employed (widely known within the SPH community as δ -SPH):

$$\langle \rho \nabla \cdot \mathbf{u} \rangle_i = -\rho_i \sum_{j \in \mathcal{N}_i} (\mathbf{u}_j - \mathbf{u}_i) \cdot \nabla_i W_{ij} V_j + \delta h c_0 \mathcal{D}_i \quad (11)$$

where \mathcal{D}_i is the “diffusion” term, defined as

$$\mathcal{D}_i = 2 \sum_{j \in \mathcal{N}_i} \psi_{ij} \frac{\mathbf{r}_{ji} \cdot \nabla W_{ij}}{r_{ij}^2} V_j, \quad (12)$$

where δ is a tunable constant (usually chosen to be 0.1), h is the SPH smoothing length, c_0 is the speed of sound, and ψ_{ij} is defined as:

$$\psi_{ij} = (\rho_j - \rho_i) - \frac{1}{2} \left(\langle \nabla_1 \rho_i \rangle + \langle \nabla_1 \rho_j \rangle \right) \cdot \mathbf{r}_{ji}, \quad (13)$$

where $\langle \nabla_1 (\cdot) \rangle$ represents the renormalized gradient operator. $\langle \nabla_1 \rho_i \rangle$ can then be written as:

$$\langle \nabla_1 \rho_i \rangle = \mathbf{B}_i \sum_{j \in \mathcal{N}(i)} (\rho_j - \rho_i) \nabla_i W_{ij} V_j \quad (14)$$

Here, \mathbf{B}_i is the *first derivative renormalization matrix* suggested by Oger et al. [14], which is defined as:

$$\mathbf{B}_i = \left[- \sum_{j \in \mathcal{N}(i)} \mathbf{r}_{ij} \otimes \nabla W_{ij} V_j \right]^{-1} \quad (15)$$

Physically, this term adds “artificial diffusivity” to the continuity equation. Although exact conservation of mass no longer holds, the added diffusivity onto the density greatly increases stability throughout the system. As a result, δ -SPH was proven to be an accurate, versatile approach for many SPH applications as an alternative to a pressure Poisson Equation (PPE) solver for estimating the pressure field [1,11].

3 Viscous forces

3.1 Semi-implicit time stepping

Non-Newtonian flows usually consist of yielded and non-yielded regions within the fluid. Many practical fluid models usually generate effective viscosities that are magnitudes

lower for yielded regions, allowing them to “flow”. Obtaining stability and convergence for such systems has been a challenge for SPH due to this property of non-Newtonian fluid models. In order to address this issue (1), we propose an implicit time-stepping scheme for the viscous step. A fixed-point iteration scheme is used to solve for the whole particle system, similar to [16,17].

Consider a system where the acceleration of each particle only depends on \mathbf{x} and \mathbf{v} . Using an implicit trapezoidal rule, for the displacement field and the acceleration field we have:

$$\begin{bmatrix} \mathbf{r}^{t+1} \\ \mathbf{u}^{t+1} \end{bmatrix} = \begin{bmatrix} \mathbf{r}^* \\ \mathbf{u}^* \end{bmatrix} + \frac{\Delta t}{2} \cdot \left(\begin{bmatrix} \mathbf{u}^{t+1} \\ \mathbf{a}_v^{t+1} \end{bmatrix} + \begin{bmatrix} \mathbf{u}^* \\ \mathbf{a}_v^* \end{bmatrix} \right) \tag{16}$$

where $\begin{bmatrix} \mathbf{r}^* \\ \mathbf{u}^* \end{bmatrix}$ represents the vector of displacements and velocities of the whole particle system at time t . Plugging in $\begin{bmatrix} \mathbf{u}^{t+1} \\ \mathbf{a}_v^{t+1} \end{bmatrix} = \begin{bmatrix} \mathbf{u}^* \\ \mathbf{a}_v^* \end{bmatrix} + \frac{\Delta t}{2} (\mathbf{a}_v^{t+1} + \mathbf{a}_v^*)$, we have for $\begin{bmatrix} \mathbf{x}^{t+1} \\ \mathbf{v}^{t+1} \end{bmatrix}$:

$$\begin{bmatrix} \mathbf{r}^{t+1} \\ \mathbf{u}^{t+1} \end{bmatrix} = \begin{bmatrix} \mathbf{r}^* \\ \mathbf{u}^* \end{bmatrix} + \frac{\Delta t}{2} \cdot \left(\begin{bmatrix} \mathbf{u}^* \\ \mathbf{a}_v^* \end{bmatrix} + \frac{\Delta t}{2} \cdot \left(\begin{bmatrix} \mathbf{a}_v^{t+1} \\ \mathbf{a}_v^* \end{bmatrix} + \begin{bmatrix} \mathbf{u}^* \\ \mathbf{a}_v^* \end{bmatrix} \right) \right) \tag{17}$$

$$= \begin{bmatrix} \mathbf{r}^* \\ \mathbf{u}^* \end{bmatrix} + \Delta t \cdot \begin{bmatrix} \mathbf{u}^* \\ \mathbf{a}_v^* \end{bmatrix} + \frac{(\Delta t)^2}{4} \left(\begin{bmatrix} \mathbf{a}_v^{t+1} \\ \mathbf{a}_v^* \end{bmatrix} + \begin{bmatrix} \mathbf{u}^* \\ \mathbf{a}_v^* \end{bmatrix} \right) \tag{18}$$

Thus, we have for $\begin{bmatrix} \mathbf{r}^{t+1} \\ \mathbf{u}^{t+1} \end{bmatrix}$:

$$\begin{bmatrix} \mathbf{r}^{t+1} \\ \mathbf{u}^{t+1} \end{bmatrix} = \begin{bmatrix} \mathbf{r}^* \\ \mathbf{u}^* \end{bmatrix} + \frac{\Delta t}{2} \cdot \left(\begin{bmatrix} \mathbf{u}^* + \frac{\Delta t}{2} \cdot \left(\begin{bmatrix} \mathbf{a}_v^{t+1} \\ \mathbf{a}_v^* \end{bmatrix} + \begin{bmatrix} \mathbf{u}^* \\ \mathbf{a}_v^* \end{bmatrix} \right) \\ \begin{bmatrix} \mathbf{a}_v^{t+1} \\ \mathbf{a}_v^* \end{bmatrix} \end{bmatrix} \right) \tag{19}$$

Notice that the acceleration vector \mathbf{a}_v^{t+1} is a nonlinear function of the velocity and the position vectors at time $t + 1$; thus $\begin{bmatrix} \mathbf{a}_v^{t+1} \\ \mathbf{a}_v^* \end{bmatrix} = \xi \left(\begin{bmatrix} \mathbf{r}^{t+1} \\ \mathbf{u}^{t+1} \end{bmatrix} \right)$, where

$$\begin{aligned} & (\mathbf{a}_v^{t+1})_i \\ &= \left(\sum_{j \in \mathcal{N}_i} \frac{4m_j (\bar{\mu}_{\text{eff}})_{ij} (\mathbf{r}_i^{t+1} - \mathbf{r}_j^{t+1}) \cdot \nabla W_{ij}}{(\rho_i^{t+1} + \rho_j^{t+1}) (\mathbf{r}_i^{t+1} - \mathbf{r}_j^{t+1}) \cdot (\mathbf{r}_i^{t+1} - \mathbf{r}_j^*)} (\mathbf{u}_i^{t+1} - \mathbf{u}_j^{t+1}) \right) \end{aligned} \tag{20}$$

Writing everything out again,

$$\begin{bmatrix} \mathbf{r}^{t+1} \\ \mathbf{u}^{t+1} \end{bmatrix} = \Psi \left(\begin{bmatrix} \mathbf{r}^{t+1} \\ \mathbf{u}^{t+1} \end{bmatrix} \right) \stackrel{\text{def}}{=} \begin{bmatrix} \mathbf{r}^* \\ \mathbf{u}^* \end{bmatrix} + \frac{\Delta t}{2} \cdot \left(\begin{bmatrix} \mathbf{u}^* + \frac{\Delta t}{2} \cdot \left(\xi \left(\begin{bmatrix} \mathbf{r}^{t+1} \\ \mathbf{u}^{t+1} \end{bmatrix} \right) + \mathbf{a}_v^* \right) \\ \xi \left(\begin{bmatrix} \mathbf{r}^{t+1} \\ \mathbf{u}^{t+1} \end{bmatrix} \right) \end{bmatrix} + \begin{bmatrix} \mathbf{u}^* \\ \mathbf{a}_v^* \end{bmatrix} \right) \tag{21}$$

Thus, we have a system of nonlinear equations with the unknown vector $\begin{bmatrix} \mathbf{r}^{t+1} \\ \mathbf{u}^{t+1} \end{bmatrix}$. We now have the form suitable for a fixed-point iteration solution process :

$$\begin{bmatrix} \mathbf{r}^{t+1} \\ \mathbf{u}^{t+1} \end{bmatrix}^{k+1} = \Psi \left(\begin{bmatrix} \mathbf{r}^{t+1} \\ \mathbf{u}^{t+1} \end{bmatrix}^k \right) \tag{22}$$

where k is the iteration counter. We simply use $\begin{bmatrix} \mathbf{r}^{t+1} \\ \mathbf{u}^{t+1} \end{bmatrix}^0 = \begin{bmatrix} \mathbf{r}^* \\ \mathbf{u}^* \end{bmatrix}$ as the initial “guess”.

3.2 Boundary conditions

A ghost-particle approach was used for our no-slip boundary implementation. Such approach is fairly straightforward, although for Newtonian fluids one must be cautious when assigning velocities to the ghost particles. In previous work regarding Newtonian fluids such as [2,12], the ghost particles were assigned mirrored velocities in order to improve

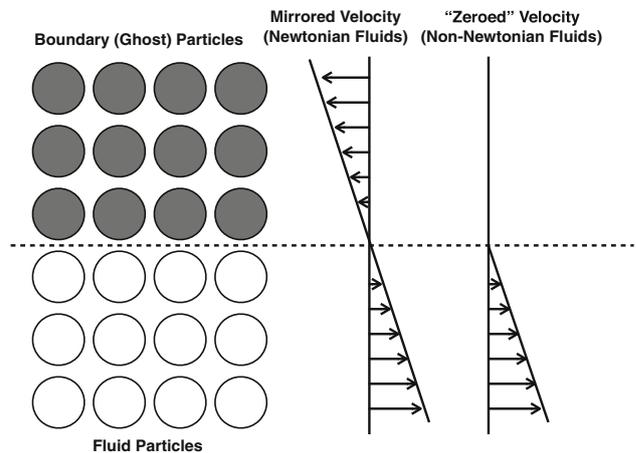


Fig. 1 No-slip boundary conditions for Newtonian fluids and non-Newtonian fluids

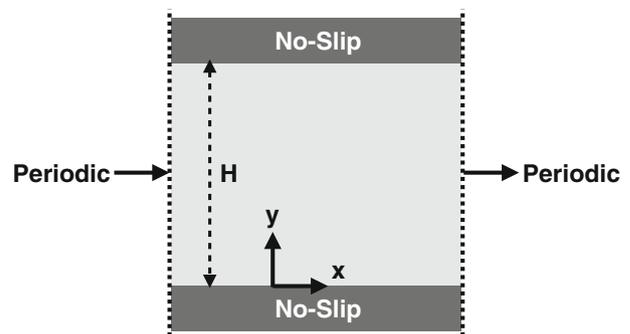


Fig. 2 Physical parameters: $\frac{\partial p}{\partial x} = 10$ [Pa/m], $\rho = 1$ [kg/m³], $H = 0.5$ [m], $\mathbf{u}(y = 0) = 0$, $\mathbf{u}(y = H) = 0$, $k = 0.5$ [Pa s], $\tau_o = 0.5$ [Pa], $n = 0.5$, $\gamma_o = 0.001$

Algorithm 1 Fixed-Point Iteration

```

function FIXEDPOINTITERATION
   $\begin{bmatrix} \mathbf{r}^{t+1} \\ \mathbf{u}^{t+1} \end{bmatrix}^0 = \begin{bmatrix} \mathbf{r}^t \\ \mathbf{u}^t \end{bmatrix}$  ▷ Initialize Guess
  while  $\text{NORM}\left(\begin{bmatrix} \mathbf{r}^{t+1} \\ \mathbf{u}^{t+1} \end{bmatrix}^{k+1} - \begin{bmatrix} \mathbf{r}^{t+1} \\ \mathbf{u}^{t+1} \end{bmatrix}^k\right) > \text{TOL}$  do
    for  $Particle_a$  in  $SolidParticles$  do ▷ Compute  $\Psi^{k+1}$ 
      COMPUTEACCELERATIONS( $Particle_a$ )
    end for
    for  $Particle_a$  in  $SolidParticles$  do ▷ Update Guess
       $\begin{bmatrix} \mathbf{r}^{t+1} \\ \mathbf{u}^{t+1} \end{bmatrix}^{k+1} = \Psi\left(\begin{bmatrix} \mathbf{r}^{t+1} \\ \mathbf{u}^{t+1} \end{bmatrix}^k\right)$ 
    end for
  end while
end function
  
```

convergence near the boundaries when implementing no-slip conditions. Unfortunately, for non-Newtonian fluids, this approach would be inappropriate since the rate of deformation tensor that is required to compute the shear tensor

($\tau = 2(\mu_{\text{eff}})_i \dot{\gamma} D_i$, $D_i = \frac{1}{2}(\nabla u_i^* + \nabla u_i^{*T})$) would incorrectly indicate that the fluid is “yielding” near the boundary. Thus, the velocities are required to be zero for the ghost particles (Fig. 1).

4 Numerical example: Herschel–Bulkley 2D poiseuille flow

A simple steady-state numerical example is presented to show convergence properties of our method, as shown in Fig. 2. The left/right boundaries were treated with a periodic boundary condition, where the leaving particles were fed back into the domain. The quintic Wendland kernel was exclusively used as our kernel function, due to its many known benefits [10]. The kernel support radius was chosen to be 3 times the particle diameter ($3 \times \Delta x$).

To compare our results, the analytical solution of the above problem [6] was used:

Fig. 3 Steady-state velocity profile compared to analytical solution

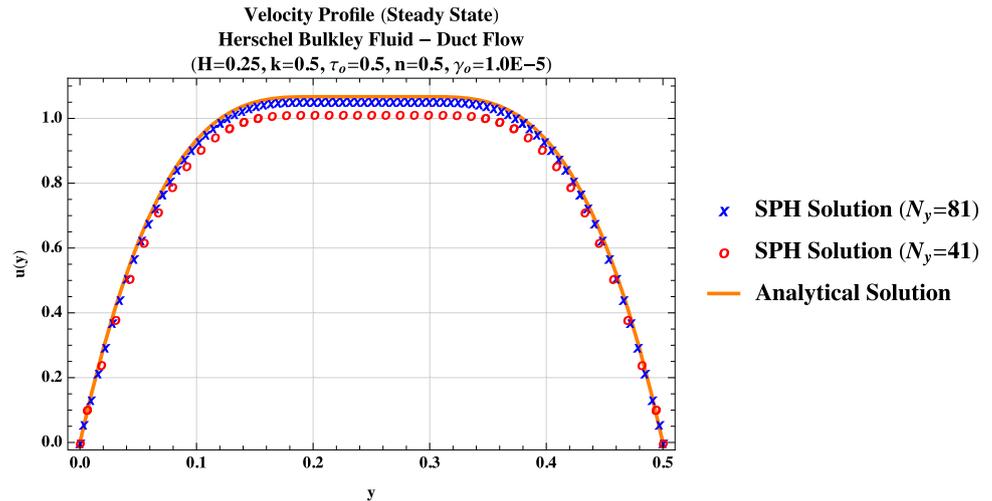
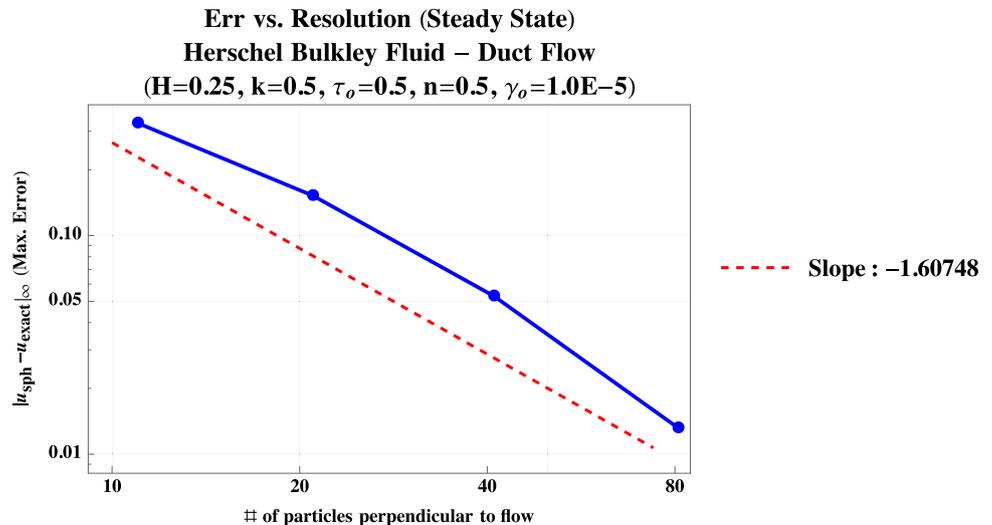


Fig. 4 The method showed a convergence order of 1.6 in space



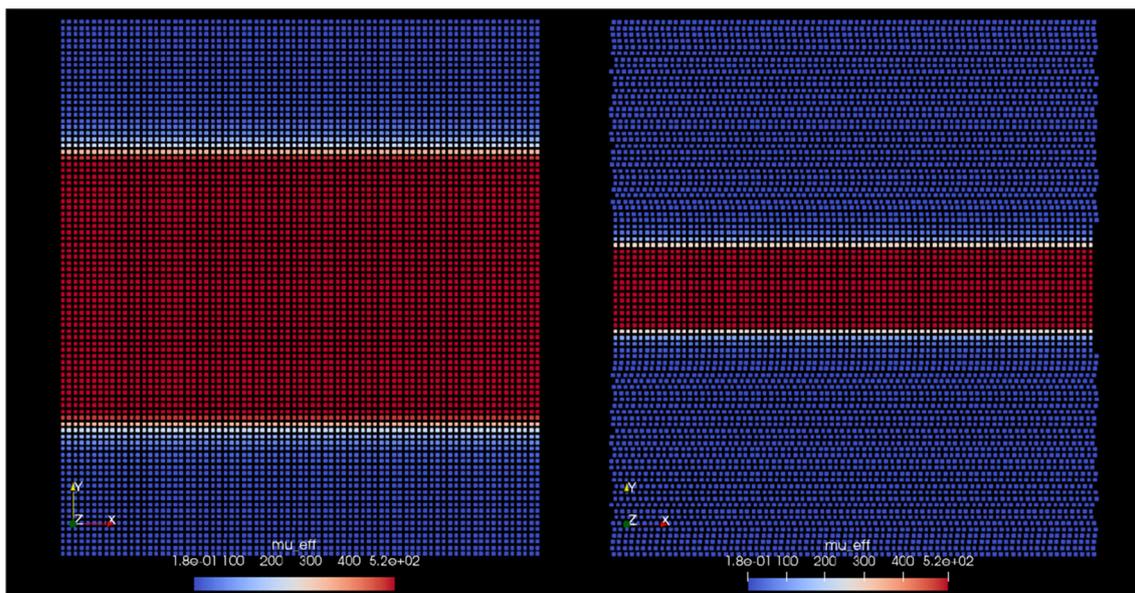


Fig. 5 Red region is the “unyielded” fluid where the effective viscosity is significantly larger. At startup (left), the region is significantly larger compared to steady state (right).

$$u_x(y) = \begin{cases} \left(\frac{n}{n+1}\right) \left[\left(\frac{\partial p}{\partial x}\right)^{\frac{1}{n}} \left(\left(\frac{H}{2} - \frac{\tau_0}{\left(\frac{\partial p}{\partial x}\right)}\right)^{\frac{n+1}{n}} - \left|y - \frac{H}{2}\right| - \frac{\tau_0}{\left(\frac{\partial p}{\partial x}\right)} \right)^{\frac{n+1}{n}} \right] & |y - \frac{H}{2}| \geq \frac{\tau_0}{\left(\frac{\partial p}{\partial x}\right)} \\ \left(\frac{n}{n+1}\right) \left[\left(\frac{\partial p}{\partial x}\right)^{\frac{1}{n}} \left(\frac{H}{2} - \frac{\tau_0}{\left(\frac{\partial p}{\partial x}\right)}\right)^{\frac{n+1}{n}} \right] & |y - \frac{H}{2}| < \frac{\tau_0}{\left(\frac{\partial p}{\partial x}\right)} \end{cases} \tag{23}$$

4.1 Numerical results

The results of the numerical solution are shown in Figs. 3, and 5. 4 simulations with different resolutions (11, 21, 41, 81 particles in the *y*-direction, respectively).

Figure 3 compares the velocity profile of the SPH solution to the analytical solution. Figure 4 shows the convergence rate of our proposed method. The results show that our semi-implicit approach is capable of resolving large viscosity contrasts between the two different regions. This shows that SPH can be a possible method of choice for producing high-quality results for non-Newtonian fluid flow simulations.

Nevertheless, compared to the simulation with the highest resolution ($N_x = 81$), the lower resolution cases seemed to show rather large errors. Since the shear rate $\dot{\gamma} = \sqrt{2\mathbf{D} : \mathbf{D}}$ is dependent on the rate of deformation tensor, which is computed at each time step with the SPH operators, it is difficult to maintain “sharp” transitions for the rate of deformation tensor. This transition region can also be observed in Fig. 5).

5 Conclusion

A new semi-implicit time-stepping scheme was proposed to overcome the large difference in effective viscosities in non-Newtonian fluids. For such systems, explicit methods are virtually fruitless. Convergence was obtained by applying our proposed scheme, with high effective viscosity ratios. However, relatively fine discretization was required in order to obtain satisfactory results. This indicates that the boundaries between the yielded region and the non-yielded regions are being excessively smoothed (which SPH operators are known to do), especially for lower resolution simulations. For full 3D simulations, this may result in very high computational expense. Additional numerical remedies that may obviate these issues are currently under investigation by the authors.

Acknowledgements The research was not funded by any third party organization.

Compliance with ethical standards

Conflict of interest We declare that there was no conflicting interests of any type during the production of this research.

Ethical statement No animals or humans participants were involved with any of the research, therefore informed consent was not required.

References

1. Antuono M, Colagrossi A, Marrone S, Molteni D (2010) Free-surface flows solved by means of SPH schemes with numerical diffusive terms. *Comput Phys Commun* 181(3):532–549. <https://doi.org/10.1016/j.cpc.2009.11.002>
2. Bierbrauer F, Bollada P, Phillips T (2009) A consistent reflected image particle approach to the treatment of boundary conditions in smoothed particle hydrodynamics. *Comput Methods Appl Mech Eng* 198(41–44):3400–3410. <https://doi.org/10.1016/j.cma.2009.06.014>
3. Chorin AJ (1968) Numerical solution of the Navier–Stokes equations. *Math Comput* 22(104):745–762. <https://doi.org/10.2307/2004575>
4. Cleary PW (1998) Modelling confined multi-material heat and mass flows using SPH. *Appl Math Model* 22(12):981–993
5. Colagrossi A, Landrini M (2003) Numerical simulation of interfacial flows by smoothed particle hydrodynamics. *J Comput Phys* 191:448–475
6. Ferrás L, Nóbrega J, Pinho F (2012) Analytical solutions for newtonian and inelastic non-newtonian flows with wall slip. *J Nonnewton Fluid Mech* 175:76–88
7. Hosseini S, Manzari M, Hannani S (2007) A fully explicit three-step SPH algorithm for simulation of non-newtonian fluid flow. *Int J Numer Methods Heat Fluid Flow* 17(7):715–735
8. Hu X, Adams N (2006) A multi-phase SPH method for macroscopic and mesoscopic flows. *J Comput Phys* 213(2):844–861
9. Liu M, Liu G (2010) Smoothed particle hydrodynamics (SPH): an overview and recent developments. *Arch Comput Methods Eng* 17(1):25–76
10. Macià F, Colagrossi A, Antuono M, Souto-Iglesias A (2011) Benefits of using a Wendland kernel for free-surface flows. In: 6th ERCOFTAC SPHERIC workshop on SPH applications, Hamburg University of Technology, pp 30–37
11. Marrone S, Antuono M, Colagrossi A, Colicchio G, Le Touzé D, Graziani G (2011) Delta-SPH model for simulating violent impact flows. *Comput Methods Appl Mech Eng* 200(13–16):1526–1542. <https://doi.org/10.1016/j.cma.2010.12.016>
12. Monaghan J, Kajtar J (2009) SPH particle boundary forces for arbitrary boundaries. *Comput Phys Commun* 180(10):1811–1820. <https://doi.org/10.1016/j.cpc.2009.05.008>
13. Monaghan JJ (1994) Simulating free surface flows with SPH. *J Comput Phys* 110(2):399–406
14. Oger G, Doring M, Alessandrini B, Ferrant P (2007) An improved SPH method: towards higher order convergence. *J Comput Phys* 225(2):1472–1492
15. Shao S, Lo EY (2003) Incompressible SPH method for simulating newtonian and non-newtonian flows with a free surface. *Adv Water Resour* 26(7):787–800
16. Zohdi T (2007) Particle collision and adhesion under the influence of near-fields. *J Mech Mater Struct* 2(6):1011–1018
17. Zohdi T (2010) On the dynamics of charged electromagnetic particulate jets. *Arch Comput Methods Eng* 17(2):109–135

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.