Shear-induced migration of rigid particles near an interface between a Newtonian and a viscoelastic fluid

Shear-Induced Migration of Rigid Particles near an Interface between a Newtonian and a Viscoelastic Fluid - Supporting Information

Nick. O. Jaensson,*,†,‡ Christos Mitrias,† Martien A. Hulsen,† and Patrick D. Anderson†

†*Department of Mechanical Engineering, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands*

‡*Dutch Polymer Institute (DPI), P.O. Box 902, 5600 AX Eindhoven, The Netherlands*

E-mail: n.o.jaensson@tue.nl

Numerical method

The finite element method with adaptive meshing is employed to solve the Cahn-Hilliard equation, the mass- and momentum balance and the evolution equation for the conformation tensor. Here a short overview of the numerical method is given. For a more detailed explanation, the reader is referred to our previous work on particles in Cahn-Hilliard fluids in Newtonian flows[1,2] and viscoelastic flows[3].

At the start of the simulation, a mesh is generated using Gmsh[4] which is aligned with the particle boundary and is refined near the fluid-fluid interface, as shown in Fig. 1. Due to the symmetry and periodic assumptions, only half of the particle is simulated. The equations are discretized in space to yield the unknowns $\boldsymbol{u}$, p, ϕ, μ and $\boldsymbol{c}$. Tetrahedral $P_2P_1P_1$ elements are used for $\boldsymbol{u}$, p and $\boldsymbol{c}$, whereas P_2P_2 elements are used for ϕ and μ. The physical boundaries

of the domain are ∂P, Γ_5 and Γ_6, where no-flux and contact angle boundary conditions are imposed for the Cahn-Hilliard problem, and no-slip is imposed for the fluid velocity. Periodicity is enforces between the planes Γ_1 and Γ_2 using constraint equations, yielding Lagrange multipliers as additional unknowns. Symmetry conditions are imposed on the planes Γ_3 and Γ_4.

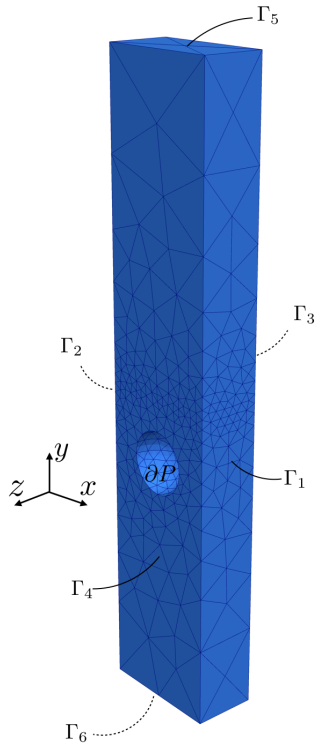


Fig. 1: An example of the computational mesh indicating the boundaries (note: in the actual computations the mesh is much more refined).

At the beginning of each time step, the mesh is moved using an arbitrary Lagrange Euler approach[5], which moves the mesh close to the particle with the particle, but keeps the mesh stationary further away from the particle. When the mesh becomes too distorted, or if the interface moves out of the refined region, a remeshing is performed. Second-order, adaptive time stepping schemes are used, and the mass and momentum balance and Cahn-Hilliard equation are solved in one system, with a prediction of the polymer stress according to[6]. Solving the mass and momentum balance and Cahn-Hilliard equation in one system was shown to drastically improve stability with respect to a decoupled approach[7], allowing for the

use of larger time steps. The (angular) velocities of the particle are included as a additional unknowns[8] using a constraint equation, and the particle velocities are such that the force- and torque-balance on the particle boundary are satisfied. This system is solved using a GMRES iterative solver from the Sparskit library[9], with a customized preconditioner as described in the next section. To solve the evolution equation for the conformation tensor, the log-conformation approach is applied[10]. Furthermore, SUPG stabilization is used[11]. After all variables at the new time step have been computed, the simulation can continue to the next time step.

The element size near the interface is chosen equal to ξ, which yields ten second-order points on the interface, which has been shown to be sufficient for mesh-convergence[12]. Furthermore, the element size near the particle is chosen as $2\pi a/64$, which yields 64 elements on the particle circumference. Using an adaptive time-stepping scheme, the time step is varied between $\Delta t\dot{\gamma} = 0.08$ and $\Delta t\dot{\gamma} = 0.32$, where the larger time step is used when the dynamics are slow (e.g when the particle migrates toward the interface) and the smaller time step is used when the dynamics are fast (e.g. the adsorption of a particle at an interface).

GMRES preconditioner

The discretized version of the mass- and momentum balance and Cahn-Hilliard equation is written as a linear system of equations:

$$\begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix}, \tag{1}$$

where x contains the unknowns in the mass- and momentum balance (velocity, pressure and Lagrange multipliers for the particle motion/periodic boundary conditions) and y contains the unknowns in the Cahn-Hilliard equation (composition, chemical potential and Langrange multipliers for the periodic boundary conditions). The vectors f and g contain information about previous time steps, needed for the integration in time. Since the full LU-

decomposition of the system given in Eq. (1) can become too large to solve efficiently, iterative solvers are essential. Our choice is the GMRES iterative solver from the Sparskit library[9], for which proper preconditioning is essential. However, the system as shown in Eq. (1) can become badly conditioned and designing a suitable preconditioner is not straightforward. Our approach is to make use of the natural block structure that arises from the separate discretized equations to define the preconditioner P:

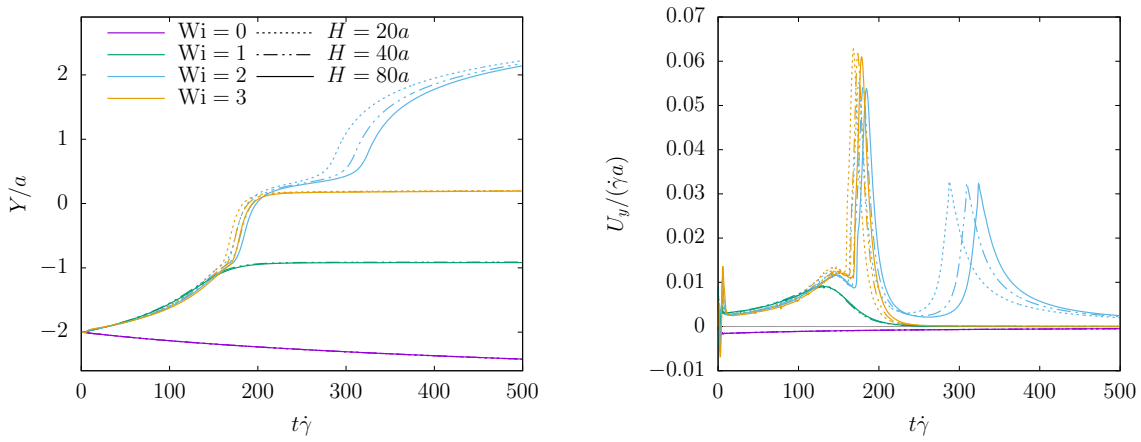
$$P = \begin{bmatrix} M_{11} & M_{12} \\ 0 & M_{22} \end{bmatrix}, \tag{2}$$

where the M_{21}-block was disregarded, i.e. the convection term in the Cahn-Hilliard equation. Using the triangular shape of P, linear systems of equations with the preconditioner P can easily be solved by first solving $M_{22}y = g$, followed by $M_{11}x = f - M_{12}y$. These subsystems are solved using a direct solver from the HSL library[13], which allows to reuse the LU decompositions during the entire GMRES iteration. Moreover, the true increase in performance arises due to the reuse of the LU decompositions across several time steps, which was found to maintain fast convergence of the GMRES iteration. In practice, the LU decompositions could be reused until a remeshing was performed, after which they had to be recomputed.

Size of the domain

In this research, a domain size in the y direction (i.e. the wall-to-wall distance) of $H = 40a$ is used. To ensure that the influence of the walls on the migration results is small, we have also performed the simulations as presented in Fig. 5 in the main article with domain sizes of $H = 20a$ and $H = 80a$. The results are presented in Fig. 2, where it can be seen that the final location of the particle, which determines the migration regime used in the morphology plots, is similar for all values of H. We therefore use $H = 40a$ in all simulations presented in this

paper.



(a) (b)

Fig. 2: The particle vertical position (a) and particle velocity (b). Ca = 1, $\theta_c = 90°$ and S = 0.1 and varying wall-to-wall distance H.

References

(1) Jaensson, N.; Hulsen, M.; Anderson, P. Stokes-Cahn-Hilliard formulations and simulations of two-phase flows with suspended rigid particles. *Computers and Fluids* **2015**, *111*, 1–17.

(2) Jaensson, N.; Hulsen, M.; Anderson, P. A comparison between the XFEM and a boundary-fitted mesh method for the simulation of rigid particles in Cahn-Hilliard fluids. *Computers & Fluids* **2017**, *148*, 121–136.

(3) Jaensson, N.; Hulsen, M.; Anderson, P. On the use of a diffuse-interface model for the simulation of rigid particles in two-phase Newtonian and viscoelastic fluids. *Computers & Fluids* **2017**, *156*, 81–96.

(4) Geuzaine, C.; Remacle, J.-F. Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering* **2009**, *79*, 1309–1331.

(5) Hu, H.; Patankar, N.; Zhu, M. Direct numerical simulations of fluid-solid systems using the arbitrary Lagrangian-Eulerian technique. *Journal of Computational Physics* **2001**, *169*, 427–462.

(6) D'Avino, G.; Hulsen, M. Decoupled second-order transient schemes for the flow of viscoelastic fluids without a viscous solvent contribution. *Journal of Non-Newtonian Fluid Mechanics* **2010**, *165*, 1602–1612.

(7) Aland, S. Time integration for diffuse interface models for two-phase flow. *Journal of Computational Physics* **2014**, *262*, 58–71.

(8) Glowinski, R.; Pan, T.; Périaux, J. Distributed Lagrange multiplier methods for incompressible viscous flow around moving rigid bodies. *Computer Methods in Applied Mechanics and Engineering* **1998**, *151*, 181–194.

(9) Saad, Y. SPARSKIT: A Basic Tool Kit for Sparse Matrix Computations. NASA Ames Research Center, Moffett Field, CA, Technical Report 2090, 2001.

(10) Hulsen, M.; Fattal, R.; Kupferman, R. Flow of viscoelastic fluids past a cylinder at high Weissenberg number: stabilized simulations using matrix logarithms. *Journal of Non-Newtonian Fluid Mechanics* **2005**, *127*, 27–39.

(11) Brooks, A.; Hughes, T. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Computational Methods in Applied Mechanical Engineering* **1982**, *32*, 199–259.

(12) Yue, P.; Feng, J.; Liu, C.; Shen, J. A diffuse-interface method for simulating two-phase flows of complex fluids. *Journal of Fluid Mechanics* **2004**, *515*, 293–317.

(13) HSL, A collection of Fortran codes for large scale scientific computation. 2013; `http://www.hsl.rl.ac.uk`.