Export Citatio

A numerical solver for active hydrodynamics in three dimensions and its application to active turbulence **(**)

Cite as: Phys. Fluids **35**, 105155 (2023); doi: 10.1063/5.0169546 Submitted: 26 July 2023 · Accepted: 3 October 2023 · Published Online: 30 October 2023

Abhinav Singh,^{1,2,3,4} 🕞 Philipp H. Suhrcke,^{1,2,3} 🍺 Pietro Incardona,^{1,2,3,a)} 🝺 and Ivo F. Sbalzarini^{1,2,3,4,5,b)} 🝺

AFFILIATIONS

¹Technische Universität Dresden, Faculty of Computer Science, Dresden, Germany

²Max Planck Institute of Molecular Cell Biology and Genetics, Dresden, Germany

³Center for Systems Biology Dresden, Dresden, Germany

⁴Center for Scalable Data Analytics and Artificial Intelligence, Dresden, Germany

⁵Cluster of Excellence Physics of Life, TU Dresden, Dresden, Germany

^{a)}Present address: IGSB, University Hospital Bonn, Bonn, Germany. ^{b)}Author to whom correspondence should be addressed: sbalzarini@mpi-cbg.de

ABSTRACT

We present a higher-order convergent numerical solver for active polar hydrodynamics in three-dimensional domains of arbitrary shape, along with a scalable open-source software implementation for shared- and distributed-memory parallel computers. This enables the computational study of the nonlinear dynamics of out-of-equilibrium materials from first principles. We numerically solve the nonlinear active Ericksen–Leslie hydrodynamic equations of three-dimensional (3D) active nematics using both a meshfree and a hybrid particle-mesh method in either the Eulerian or Lagrangian frame of reference. The solver is validated against a newly derived analytical solution in 3D and implemented using the OpenFPM software library for scalable scientific computing. We then apply the presented method to studying the transition of 3D active polar fluids to spatiotemporal chaos, the emergence of coherent angular motion in a 3D annulus, and chiral vortices in symmetric and asymmetric 3D shapes resembling dividing cells. Overall, this provides a robust and efficient open-source simulation framework for 3D active matter with verified numerical convergence and scalability on parallel computers.

© 2023 Author(s). All article content, except where otherwise noted, is licensed under a Creative Commons Attribution (CC BY) license (http:// creativecommons.org/licenses/by/4.0/). https://doi.org/10.1063/5.0169546

I. INTRODUCTION

C Author(s) 2023

Active matter refers to material systems whose constituents convert chemical energy into mechanical work, resulting in self-organized emergent motion. The constant input of energy at the microscopic scale maintains the system out of thermodynamic equilibrium. The type of emerging motion depends on many factors, including the rate of energy injection,¹ material properties,^{2,3} and constituent concentrations.⁴ This gives rise to rich nonlinear dynamics, including topological active hydrodynamics.^{5–7} In addition, active matter often comprises anisotropic or orientable constituents, endowing it with nematic (liquid crystal) or polar order, respectively. Examples of nematic or polar active matter include cytoskeletal structures in living cells,^{8–10} bacterial suspensions,^{11,12} multicellular tissues,¹² and flocks of birds.¹³ The governing partial differential equations (PDEs) for the mean-field description of polar or nematic active matter have been derived from first

principles (symmetries and conservation laws),^{8–10,14} as comprehensively reviewed by Marchetti *et al.*¹² Studying the fascinating emergent properties of this class of materials, however, relies on numerical simulations, as the equations are analytically intractable in all but the simplest one-dimensional cases.

Owing to this importance of computer simulations in the study of active matter, various numerical methods have been proposed. On the microscopic level, active matter has been simulated "bottom-up," resolving the individual molecular components and their stochastic dynamics, as oriented active particle systems,¹⁵ by Brownian dynamics,¹⁶ or by stochastic simulation algorithms.¹⁷ Furthermore, molecular dynamics simulations using HOOMD-Blue^{18,19} have been used for active matter studies at the molecular scale.^{20,21} On the mesoscopic level, hybrid Lattice-Boltzmann methods (hLBM) have established themselves as a standard for simulating active matter in both two dimensions (2D) and three dimensions (3D), due to their versatility in modeling complex fluids.²² hLBM for active matter derives from a classic algorithm for passive liquid crystals,^{23–25} which was extended to active nematics. This has been successfully applied to a range of problems in active hydrodynamics, including chaotic flows (active turbulence)^{2,26–32} and topological defect dynamics in activity gradients.^{28,29,31,33,34} Despite their popularity, though, hLBM methods are limited to periodic or no-slip boundary conditions, and convergence to the correct hydrodynamic mean-field limit is only guaranteed for certain lattice symmetries³⁵ under specific moment constraints on the equilibrium distribution.^{22,33} If they converge, the order of convergence is at most 1 in domains with curved boundaries,³⁶ but a systematic convergence validation for active hydrodynamics is missing in the literature. Most importantly, an open-source implementation of hLBM for active matter does not seem to be available, limiting reproducibility and utility for the scientific community.

On the macroscopic, continuum level, spectral or pseudospectral methods have been used to numerically solve the governing PDEs.³⁷ This has, e.g., enabled studying active suspensions,³⁸⁻⁴¹ bacterial suspensions,⁴² and chaotic behavior in 2D active nematics.^{43,44} Spectral methods have superior accuracy (exponential convergence to machine precision) in periodic domains, but are limited to simple domain geometries and lose their spectral convergence for non-periodic boundary conditions.⁴⁵⁻⁴⁷ Ramaswamy et al.⁴⁸ therefore presented a hybrid Lagrangian particle-mesh method for 2D active polar fluids, which allows for arbitrary boundary conditions in rectangular domains and was the first method solving the most general Ericksen-Leslie PDEs of active polar fluids with Lagrange multipliers for unit polarity magnitude.8-10,14 This method has been used to characterize spontaneous transitions of 2D contractile active fluid from a resting initial state to spontaneous laminar flow, and further to traveling vortices and waves, and eventually spatiotemporal chaos.⁴⁹ Due to the use of staggered Cartesian grids, this method is also limited to simple (rectangular or isomorphic to a rectangle) domain geometries. A simplified unsteady Stokes-flow model of active hydrodynamics has been solved using finite differences in order to study topological defect dynamics and spatiotemporal chaos in 2D active nematics.^{50,51} In 3D, a box-shooting algorithm on a staggered Cartesian grid was proposed for solving a minimal phenomenological model of active hydrodynamics that did not include flows induced by the rotation of molecules along their axis in the nematic field.⁵² Jointly, these works emphasize the progress, but also the challenges and limitations of the different numerical methods applied to active polar hydrodynamics. Importantly, a convergence-validated method that solves the macroscopic symmetry-preserving PDE model in 3D with arbitrary boundary conditions and in arbitrary 3D geometries is, to our knowledge, so far missing.

This work focuses on the computational challenges to numerically solve the full nonlinear theory of active polar hydrodynamics arising from symmetry relations. The model considered here contains a 3D vectorial orientation field, which is nonlinearly coupled to the flow velocity field. More specifically, we present a numerical method to solve the 3D active Ericksen–Leslie equations that model polar or nematic active matter in the mean-field. The proposed hybrid particle-mesh approach can incorporate arbitrary boundary conditions on complexshaped 3D domain geometries, and it is implemented as a scalable open-source software for shared-memory and distributed-memory parallel computers. We use Discretization-Corrected Particle Strength Exchange (DC-PSE)⁵³ to consistently discretize differential operators on geometry-adapted distributions of collocation points. This is coupled with a pressure-correction scheme to enforce incompressibility of the fluid, avoiding the use of staggered grids, which are challenging in complex domains. We validate convergence of the presented solver for the incompressible force balance on regular and irregular collocation-point distributions. The computational cost associated with 3D problems is addressed by using the open-source framework OpenFPM for scalable and efficient scientific computing.54 The complexity of the governing equations is tackled using a C++ template expression system for PDEs, greatly reducing code complexity and improving readability and extensibility.⁵⁵ We present the derivation of the governing equations in 3D, a detailed description of the algorithm, as well as its performance and convergence. The algorithm is then applied to study the spontaneous flow transition in an extensile active polar fluid with friction boundary conditions, and the transition of the same system to spatiotemporal chaos (a.k.a. active turbulence) as well as to traveling waves in a 3D annulus. Finally, the application of the presented solver to asymmetric 3D geometries is demonstrated, simulating the onset of vortical flows in geometries resembling dividing biological cells. Together, these results fully validate the proposed numerical method and illustrate its application to cases where the symmetry-preserving PDE model of active matter was hitherto impossible to solve.

II. HYDRODYNAMIC DESCRIPTION OF 3D ACTIVE LIQUID CRYSTALS

The hydrodynamic theory of active matter describes the dynamics of out-of-equilibrium systems that are driven by energy conversion at the microscopic scale.^{8–10,14} The average orientation of the microscopic constituents in a control volume defines the macroscopic orientation field and is referred to as polarity field **p** with components p_{α} , where α stands for any coordinate direction [{*x*, *y*, *z*} in Cartesian coordinates]. A typical example of such a material is the cytoskeleton in biological cells. There, the polarity field describes the average orientation of the cytoskeletal filaments.⁵⁶ Motor proteins bind to the filaments and convert chemical energy (in the form of the "fuel" molecule ATP) to mechanical work (in the form of forces/stresses). This deforms the filaments and sets the surrounding fluid in motion. At the same time, passive and elastic stresses relax the polarity field toward uniform nematic order, and the viscosity of the surrounding fluid dissipates energy. The resulting force balance leads to self-organized emergent flows with velocity field **v** with components v_{α} . Based on first principles (conservation laws, symmetry, entropy production, and Onsager reciprocal relations), the constitutive equations can be derived, linking all stresses to the fluid strain rate.

We consider an incompressible active polar fluid in the hydrodynamic limit of active viscous fluids with constant activity levels at a low Reynolds number. In a 3D domain, this results in three PDEs for the velocity components, three PDEs for the polarity components, and the continuity equation for the incompressibility condition. Using Einstein's index summation convention, wherein repeated indices imply summation, the governing equations can be written as

$$\frac{Dp_{\alpha}}{Dt} = \frac{h_{\alpha}}{v} - \nu u_{\alpha\beta} p_{\beta} + \lambda \Delta \mu p_{\alpha}, \qquad (1a)$$

$$\partial_{\beta}\sigma^{(\text{tot})}_{\alpha\beta} - \partial_{\alpha}\Pi = 0, \tag{1b}$$

$$\partial_{\gamma} v_{\gamma} = 0, \qquad (1c)$$

$$2\eta u_{\alpha\beta} = \sigma_{\alpha\beta}^{(s)} - \zeta \Delta \mu \left(p_{\alpha} p_{\beta} - \frac{1}{2} p_{\gamma} p_{\gamma} \delta_{\alpha\beta} \right)$$

$$-\frac{\nu}{2}\left(p_{\alpha}h_{\beta}+p_{\beta}h_{\alpha}-\frac{2}{3}p_{\gamma}h_{\gamma}\delta_{\alpha\beta}\right),$$
 (1d)

where $\alpha, \beta, \gamma \in \{x, y, z\}$ represent spatial vector components, η is the fluid viscosity, γ is the rotational viscosity of the polarity field (not to be confused with the repeated subscript γ), h is the so-called molecular field as introduced below, and ν is the coupling coefficient between mechanical stress and polarization that controls the flow-aligning $(|\nu| > 1)$ or flow-tumbling $(|\nu| < 1)$ nature of the nematic order. Additionally, λ is the coefficient coupling the polarity dynamics with the active chemical potential $\Delta \mu$. The coefficient ζ couples the activity potential $\Delta \mu$ to the active stress. If $\Delta \mu > 0$, the sign of ζ controls the contractile or extensile nature of the active fluid, with $\zeta < 0$ leading to contractile stress and $\zeta > 0$ to extensile stress. The time evolution of the polarity field $\mathbf{p} = (p_x, p_y, p_z)^{\top}$ is governed by Eq. (1a) in terms of the co-rotational Lagrangian derivative

$$\frac{\mathrm{D}p_{\alpha}}{\mathrm{D}t} = \frac{\partial p_{\alpha}}{\partial t} + v_{\gamma}\partial_{\gamma}p_{\alpha} + \omega_{\alpha\beta}p_{\beta}, \qquad (2)$$

where $\omega_{\alpha\beta} = \frac{1}{2}(\partial_{\alpha}v_{\beta} - \partial_{\beta}v_{\alpha})$ is the vorticity tensor and $u_{\alpha\beta} = \frac{1}{2}(\partial_{\alpha}v_{\beta} + \partial_{\beta}v_{\alpha})$ the strain-rate tensor. In the Eulerian frame of reference, Eq. (1a) thus becomes

$$\frac{\partial p_{\alpha}}{\partial t} = \frac{h_{\alpha}}{\gamma} - \nu u_{\alpha\beta} p_{\beta} + \lambda \Delta \mu p_{\alpha} - v_{\gamma} \partial_{\gamma} p_{\alpha} - \omega_{\alpha\beta} p_{\beta}$$

Equation (1b) is the Stokes equation, with $\sigma_{\alpha\beta}^{(tot)}$ denoting the total deviatoric stress tensor and Π the pressure. Equation (1c) is the incompressible continuity equation, imposing mass conservation. The constitutive equation in Eq. (1d) relates the strain-rate tensor $u_{\alpha\beta}$ to the symmetric, the active, and the passive stresses, which are the first, second, and third terms of the right-hand side, respectively.

In the constitutive equation, the so-called molecular field **h** appears. It is defined as the functional derivative $h_{\alpha} = -\delta F/\delta p_{\alpha}$ = $-\partial f/\partial p_{\alpha} + \partial_{\beta}(\partial f/\partial(\partial_{\beta}p_{\alpha}))$ of the Frank free energy *F*, which in 3D reads

$$F_{3D} = \int \underbrace{\left[\frac{K_s}{2} \left(\partial_{\alpha} p_{\alpha}\right)^2 + \frac{K_t}{2} \left(p_{\alpha} \varepsilon_{\alpha\beta\gamma} \partial_{\beta} p_{\gamma}\right)^2 + \frac{K_b}{2} \left(\varepsilon_{\delta\varepsilon\alpha} p_{\varepsilon} \left(\varepsilon_{\alpha\beta\gamma} \partial_{\beta} p_{\gamma}\right)\right)^2 - \frac{1}{2} h_{||}^0 p_{\alpha} p_{\alpha}\right]}_{f} \, \mathrm{d}V \,, \tag{3}$$

where the constants K_s , K_b and K_b control the splay, twist, and bend elasticity of the fluid, respectively, and $\varepsilon_{\alpha\beta\gamma}$ is the Levi-Civita symbol. The whole integrand is referred to as the Frank free-energy density f. Squares over vectors imply a scalar product. The free energy is constrained to unit polarity magnitude by $h_{||}^0$ acting as a Lagrange multiplier. The free energy in Eq. (3) does not explicitly enforce a polarity in the material; hence, the model considered here is equivalent to the Qtensor evolution of a nematic field for systems of infinite size, except for defect nucleation and defect dynamics at sufficient activity levels.^{12,14} However, confinement with anchoring boundary conditions does enforce a polarity, as the equations describe the evolution of a polarity vector field. The molecular field h can be decomposed into polarity-parallel $h_{||} = p_{\alpha}h_{\alpha}$ and polarity-perpendicular $h_{\perp,\alpha} =$ $\epsilon_{\alpha\beta\gamma}p_{\beta}h_{\gamma}$ components in a local co-moving frame (see the Appendix A). Since molecular fields that differ by a factor of $h_{\parallel}^0 p_{\alpha}$ are equivalent,⁵⁷ $h_{||} = h_{||}^0$ becomes the Lagrange multiplier to enforce unit polarity magnitude, i.e., the vector field **p** indicates the direction of nematic order with $||\mathbf{p}|| = 1$. Using Eq. (1a) and enforcing $p_{\gamma} \frac{Dp_{\gamma}}{Dt} = 0$, which is equivalent to constraining the Frank free energy to only allow unit polarity magnitude, leads to the Lagrange multiplier

$$h_{||}^{0} = h_{||} = -\gamma \left[\lambda \Delta \mu - \frac{2\nu}{p_{\gamma} p_{\gamma}} (u_{\alpha\beta} p_{\alpha} p_{\beta}) \right].$$
(4)

The resulting expressions for the perpendicular components, of which there are three in 3D $(h_{\perp,x}, h_{\perp,y}, h_{\perp,z})$, are given in Appendix A. The total deviatoric stress tensor $\sigma_{\alpha\beta}^{(tot)}$ can be

decomposed into its symmetric part $\sigma_{\alpha\beta}^{(s)}$, its antisymmetric part $\sigma_{\alpha\beta}^{(a)}$, and the Ericksen stress tensor $\sigma_{\alpha\beta}^{(e)}$. The symmetric stresses $\sigma_{\alpha\beta}^{(s)}$ are given by Eq. (1d), whereas $\sigma_{\alpha\beta}^{(a)}$ and $\sigma_{\alpha\beta}^{(e)}$ are defined as

$$\sigma_{\alpha\beta}^{(a)} = \frac{1}{2} (p_{\alpha} h_{\beta} - p_{\beta} h_{\alpha}), \quad \sigma_{\alpha\beta}^{(e)} = -\frac{\partial f}{\partial (\partial_{\beta} p_{\gamma})} \partial_{\alpha} p_{\gamma}.$$
(5)

The pressure Π acts as another Lagrange multiplier to enforce the incompressibility condition on the velocity field, $\partial_{\gamma} v_{\gamma} = 0$. Because of the Lagrange multiplier $h_{||}$ entering the stress tensor, additional flow-field derivatives appear in the force-balance equation when computing the divergence of the total stresses $\partial_{\beta} \sigma_{\alpha\beta}^{(tot)}$. This leads to the final governing equations as given in Appendix B. Since those equations would be challenging to hand-code in a computer programming language, we leverage a C++ template expression system⁵⁵ to directly translate the mathematical expressions into executable code.

III. NUMERICAL ALGORITHM

The numerical algorithm for the equations in Appendix B starts by solving the force balance, because knowing the velocity field is prerequisite for evolving the polarity in time. It then computes the polarity time evolution in either an Eulerian or Lagrangian frame of reference. This results in a hybrid particle-mesh method with normalization correction for the polarity field as outlined in the flow chart in Fig. 1. All computer codes are implemented in C++ using the open-source scalable scientific computing framework OpenFPM.⁵⁴ 30 October 2023 13:32:10



FIG. 1. Overall flow chart of the numerical solver.

The present approach is fundamentally different from the popular hybrid Lattice-Boltzmann method, as it directly solves the mean-field PDE model of the system in the long time limit. Unlike Lattice-Boltzmann, the present method does not evolve the velocity field in time, but instead solves the force balance for the velocity at each time step. This allows flow velocities to relax on a different timescale than the polarity. Previous simulation methods also confined predictions to 2D or to solving a transient Stokes equation in 3D accompanied by a finite speed of sound. The present approach relaxes these limitations and also affords more freedom than Lattice-Boltzmann methods in choosing the space and time resolution of the simulations. One should always carefully check the relationship between viscosity, relaxation time, and mesh resolution in order to avoid unphysical oscillations in the computed flow fields. This is easier to do in the present framework than in Lattice-Boltzmann methods, where the viscosity inherently depends on the mesh resolution. This becomes particularly important when simulating active turbulence, where numerical compressibility can influence the results. Unlike Lattice-Boltzmann methods, the solver presented here converges toward the incompressible limit as resolution increases, offering precise tolerance-based control over compressibility artifacts without needing to monitor the speed of sound.

A. Force-balance solver

Given the polarity field **p** for any point in time *t*, the induced velocity field **v** is determined by discretizing Eqs. (B1)–(B3) using DC-PSE differential operators.⁵³ Incompressibility is imposed using a pressure-correction scheme as previously described in $2D^{58.59}$ and

successfully used with DC-PSE operators.⁶⁰ Here, we extend this algorithm to 3D. The solver can be used on irregular distributions of collocation points, which are therefore referred to as particles. This renders the algorithm suitable for simulations in both Eulerian and Lagrangian frames of reference, where particles are stationary or move with the velocity field of the flow, respectively. The resulting linear system of equations is assembled using the OpenFPM template expression system⁵⁵ and is solved with the KSPGMRES solver from the PETSc library.⁶¹ The pressure-correction algorithm iteratively improves the estimate of the flow velocity to a desired numerical tolerance ϵ_{ν} . The tolerance ϵ_{ν} is bounded from below by the approximation error of the spatial derivatives, which converges with increasing resolution. This tolerance is set it to 10^{-2} for h = 0.2 and to 10^{-3} for h = 0.1. The full pressure-correction solver is described in Algorithm 1.

ALGORITHM 1: Incompressible force-balance solver for a given polarity field **p**.

Input:

- 1. ϵ_{ν} : Numerical relative tolerance for the incompressible flow solver.
- 2. n_{max} : Maximum number of pressure-correction iterations. 3. \mathbf{v}^0 , Π^0 : Initial guess for velocity and pressure
- **Output: v**: Incompressible flow velocity satisfying Eq. (1b) and the boundary conditions.

 $n = 0, \epsilon = \infty$ while $||\mathbf{v}^n - \mathbf{v}^{n-1}||_2 < \epsilon_v$ and $n \le n_{\max}$ do Solve Stokes equations for \mathbf{v}^n using Π^{n-1} Correct pressure: $\Pi^n = \Pi^{n-1} - \partial_k v_k^n$ if $||\mathbf{v}^n - \mathbf{v}^{n-1}||_2 > \epsilon$ then Stop with error: "Requested tolerance is unachievable for the chosen resolution" end $\epsilon = ||\mathbf{v}^n - \mathbf{v}^{n-1}||_2$ $\mathbf{v}^{n-1} = \mathbf{v}^n$ n = n + 1end

B. Time evolution of the polarity field

Once the velocity field has been computed, a time-integration step for the polarity field is performed. Equation (1a) can be integrated in time in either an Eulerian or Lagrangian frame of reference until a desired final time t_f . In an Eulerian frame of reference, the advection operator in Eq. (2) is be computed using DC-PSE spatial operators. To achieve numerical stability, however, predictor–corrector time integration is necessary, as the operators are centered in space and not upwind. Hence, for the Eulerian frame of reference, we use the second-order Adams–Bashforth–Moulton predictor–corrector scheme for time integration, as implemented in OpenFPM.⁶² For Eulerian simulations, the time step δt is limited by the CFL condition $\frac{u_s \delta t}{h} + \frac{u_r \delta t}{h} \leq 1$ for numerical stability, where *h* is the average inter-particle spacing.⁶³ This severely limits the time step size. The issue is exacerbated at higher activity $|\zeta|$, leading to stronger active stresses causing faster flows. However, when computing flow steady states, and for

imposing boundary conditions in complex domain shapes, the Eulerian frame of reference still provides better stability than the Lagrangian approach. Steady states are detected with a user-defined tolerance of ϵ_{steady} , checked against the maximal rate of change of polarity in the entire domain.

The Lagrangian frame of reference allows for the less stringent Lagrangian CFL condition $\delta t ||\nabla \mathbf{v}|| \leq 1$, imposing that particle trajectories never cross flow streamlines.⁶⁴ Moving the particles changes their positions, which requires DC-PSE kernels to be recomputed at each simulation time step. Time integration in the Lagrangian case is also the second-order Adams-Bashforth-Moulton done using predictor-corrector scheme. Alternatively, Runge-Kutta 4 can be used in this case. This adds additional computational cost, but is amortized in most cases by the increased stability permitting larger time steps.⁵¹ Detecting non-equilibrium steady states with nonzero flow, however, is not easily done in the Lagrangian frame of reference. It is only possible if the rate of change of the polarity can be equated to an approximation of the advection operator. Computing the advection operator, however, introduces additional numerical error and lowers the tolerance with which the steady state can be detected. Nevertheless, the superior timestepping stability of the Lagrangian method makes it the preferred choice for simulating unsteady flows in simple Cartesian geometries.

The present method supports all types of polarity boundary conditions including homogeneous and inhomogeneous Dirichlet, Neumann, and Robin (mixed) boundary conditions. Dirichlet boundary conditions are straightforward, as the polarity simply remains constant at boundary particles. Neumann boundary conditions are imposed using the method of images by creating a "ghost layer" of particles in a thin tubular neighborhood outside the domain boundary. The width of this ghost layer is given by the radius of the DC-PSE operators used to discretize derivatives. The ghost layer is created by mirroring particles at the boundary along the normal **n**. If the particles do not move, this is done only once at the beginning of a simulation, otherwise at each time step. In a distributed-memory simulation, it is ensured that ghost and source particles are stored on the same process to minimize communication. The polarity values of the "ghost particles" are reset at each time step, ensuring the correct value of $\partial \mathbf{p}/\partial \mathbf{n}$ at the boundary.

C. Remeshing

Another issue in Lagrangian simulations is that particles may eventually evolve to a distorted distribution at long simulation times of steady flows. Then, remeshing becomes necessary,⁶⁴ whereby the particle positions are reset to a regular Cartesian lattice and the field quantities are "interpolated" from the old to the new set of particles. Multiple methods are available for moment-conserving particle-to-mesh remeshing.⁶⁵ Here, we use the lambda kernels $\Lambda_{m,n}$, which exactly conserve the first m moments of the field⁶⁴ and are of smoothness class \mathcal{C}^n . However, using such conservation-enforcing kernels only leads to higher-order (>1) convergent results if the zeroth moment of the remeshed field is actually conserved over time.⁶⁶ The present solver only remeshes the polarity field, since the velocity field is recomputed from the force balance at every time step. Since the polarity field is constrained to unit magnitude, its zeroth moment is constant. This justifies the use of moment-conserving remeshing kernels, which converge with the desired order in this case. Specifically, we use the $\Lambda_{2,1}$ kernel,⁶⁴ which is equivalent to the classic M'_4 kernel,⁶⁷ for remeshing Lagrangian particles

$$\Lambda_{2,1}(d) = M'_4(d) = \begin{cases} 1 - \frac{1}{2}(5d^2 - 3d^3) & 0 \le d < 1, \\ \frac{1}{2}(2-d)^2(1-d) & 1 \le d \le 2, \\ 0 & d > 2, \end{cases}$$
(6)

where d is the distance of a particle to a grid node, measured in units of the grid spacing h. The algorithm loops through all particles and, for each particle, over all mesh nodes in the support of the $\Lambda_{2,1}$ kernel around it. For each particle-node pair, the above formula is used to compute the fraction of the particle's field value that is attributed to that mesh node. The mesh nodes accumulate (sum) the contributions they receive from all the particles. We remesh the polarity field after every advection time step. To ensure full particle support for the interpolation kernel close to boundaries, we use the method of images.⁴⁸ This method mirrors all particles within a distance of 3h from a boundary along the boundary normal. Specifically, if a particle *p* inside the domain carries a polarity of $p_{\scriptscriptstyle p}$ and is distance ℓ away from the boundary, a mirror particle will be created at distance $-\ell$ with a polarity of $\mathbf{p}_{\text{mirror}} = 2\mathbf{p}_b - \mathbf{p}_p$, where \mathbf{p}_b is the polarity boundary condition. The algorithm also takes care of renormalizing the polarity magnitude to compensate for errors accumulated during polarity time evolution. This is necessary, as these errors would otherwise amplify to larger numerical errors for long-time simulations. Hence, the polarity is projected onto the unit sphere if its magnitude deviates from 1 by more than a user-defined tolerance ϵ_p . The pseudo-code for the complete Lagrangian advection with remeshing and polarity renormalization is given in Algorithm 2. It is a hybrid particle-mesh method.

ALGORITHM 2: Lagrangian advection with remeshing and polarity renormalization.

Input:

1. *S*₀: Particle set with polarity **p** and velocity **v** stored on the particles located on a mesh.

2. ϵ_p : Numerical tolerance for deviation in polarity magnitude. **Output:** Polarity field **p** on S_0 after advection with velocity *v*. Create a particle distribution S_1 by advecting particles from S_0 . Create mirror particles outside the domain boundaries as required

for the method of images to impose boundary conditions.⁴⁸ Interpolate polarization \mathbf{p} from S_1 to S_0 .

for each particle do

if
$$|1 - ||\mathbf{p}||_2| > \epsilon_p$$
 then
 $|\mathbf{p} = \frac{\mathbf{p}}{||\mathbf{p}||_2}$
end

end

IV. RESULTS

The theoretically expected order of convergence of the method presented above depends on the spatial discretization operators and the time-integration scheme used. In the presented solver, all spatial differential operators are discretized using DC-PSE with consistency of $O(N^{-2})$, where N is the total number of particles used in the simulation. The $\Lambda_{2,1}$ remeshing scheme converges with third order $O(N^{-3})$ and does therefore not limit the convergence rate. Time is integrated

using the two-step Adams-Bashforth-Moulton predictor-corrector scheme, which has a global convergence order of $O(N_t^{-2})$, where N_t is the total number of time steps in a simulation. We therefore expect the method to overall be second-order accurate in both space and time.

A. Validation: Three-dimensional active incompressible channel flow

We validate the numerical convergence of the present solver (and its software implementation) in a case with $\nu = 0$ for which an analytical steady-state solution can be derived in 3D. Consider a 3D rectangular channel that is infinitely long along the x and the z directions and has thickness L in the y direction. The channel wall at y = L is stress-free $(\sigma_{xv}^{(\text{tot})}(x,L,z,t)=0)$, has no slip in z, $[v_z(x,L,z,t)=0]$, and is impenetrable ($v_y(x, L, z, t) = 0$). The surface at y = 0 is also impenetrable $(v_y(x,0,z,t)=0)$ and has no slip in x and z $[v_x(x,0,z,t)]$ $= v_z(x, 0, z, t) = 0$]. The polarity is anchored at both surfaces as $(p_x, p_y, p_z)(x, L, z, t) = (0, 1, 0)$ and $(p_x, p_y, p_z)(x, 0, z, t) = (1, 0, 0)$. Under these conditions, $v_v = 0$ everywhere due to incompressibility and translational invariance in x and z. For the initial condition

$$\mathbf{p}(x, y, z, 0) = \left(\cos\left(\frac{\pi y}{2L}\right), \sin\left(\frac{\pi y}{2L}\right), 0\right),\tag{7}$$

we can derive the asymptotic (for $t \to \infty$) steady-state solution

$$\mathbf{p}(x, y, z, t \to \infty) = (\cos \theta(y), \sin \theta(y), 0), \tag{8a}$$

$$u_{xy}(x, y, z, t \to \infty) = u_{yx}(x, y, z, t \to \infty)$$
$$= \frac{K}{\gamma [1 + \nu \cos \theta(y)]} \frac{d^2 \theta(y)}{dy^2}.$$
 (8b)

For $\nu = 0$, the hydrodynamic equations reduce to

with

$$\frac{d^2\theta}{dy^2} = \frac{\Delta\mu\gamma\zeta}{K(4\eta+\gamma)}\sin 2\theta,$$
(9)
$$\theta(y) = \operatorname{am}\left(\sqrt{c_1 - a}(c_2 + y), -\frac{2a}{c_1 - a}\right),$$
(10)

(9)

where am(u, k) is the Jacobi amplitude function. This solution in 3D is invariant along z and is therefore an extension of a known 2D problem.4

We numerically solve this problem for $\eta = 1$, $\gamma = 1$, $\zeta = -1$, $\lambda = 1$, $K_s = K_b = K_t = 1$, $\Delta \mu = -1$, and L = 10 in the Lagrangian frame of reference with remeshing. For this set of parameters, the analytical solution has $\theta(y) = \operatorname{am}(\sqrt{0.2(1+y)}, -0.5)$. We use the proposed hybrid particle-mesh method with $\epsilon_v = 10^{-5}, \epsilon_p = 10^{-16},$ $n_{\rm max} = 30$, and $\epsilon_{\rm steady} = 10^{-9}$ to perform the simulation in a boxshaped domain of size (10, 10, 5h) discretized by a uniform Cartesian mesh of resolution h. The domain has periodic boundary conditions in x and z to model the infinite extent of the fluid in these directions. In the y direction, the boundary conditions are as specified by the problem.

The polarity field is invariant along the x and z axes [Figs. 2(a) and 2(b), left]. Its one-dimensional gradient in y creates coherent active stresses that generate a unidirectional flow of material along the x axis. This flow has a velocity gradient that respects the no-slip boundary condition at the bottom wall and the stress-free boundary condition at the top wall [Figs. 2(a) and 2(b), right]. Reversing the sign of the active stress in the simulation reverses the flow direction, respecting the symmetry of the problem.

Refining *h*, we observe second-order convergence to the analytical solution at t = 40, as theoretically expected. In order to satisfy the CFL condition, the time step size is refined along with the grid resolution according to the relation $N_x = \sqrt{0.64N_t + 1}$, where N_x and N_t are the number of particles in x direction and number of time steps to reach t = 40, respectively. The initial polarity and velocity fields at t = 0 are visualized in Fig. 2(a). The steady-state numerical solution at t = 40 is visualized in Fig. 2(b). We plot the $L_2(\circ, \bullet)$ and $L_{\infty}(\diamondsuit, \blacklozenge)$ errors in both the polarity field **p** and the nonzero components of the strain rate u_{xy} for increasing N_x ($h = 1/N_x$) in Fig. 2(c). Figure 2(d) further verifies that the error in the polarity magnitude $||\mathbf{p}||$ plateaus over time and converges for decreasing h, confirming numerical convergence of the Lagrange multiplier h_{\parallel} . The jaggedness in Fig. 2(d) reflects numerical "ringing" of the Lagrange multiplier. This ringing is caused by the remeshing error causing the predictor-corrector time integration scheme with constant time step size to overshoot the flow steady state. The amplitude of this ringing, however, does converge with the correct order for increasing resolution. Together with earlier convergence results of the DC-PSE pressure-correction scheme for steady-state Stokes flow in a 3D ball with irregular particle distributions,⁵⁵ this validates the correctness of the present method also in the active polar case.

B. Benchmarks: Computational cost and scalability

Next, we benchmark the computational performance of the solver. The KSPGMRES algorithm⁶⁸ implemented in the PETSc library uses the Arnoldi iterations to find the orthonormal basis of the Krylov subspace. In each Arnoldi iteration, a new basis vector is found and orthonormalized using the Gram-Schmidt process. This requires storing the vectors from all k previous iterations. The number of multiplications over k Arnoldi iterations thus is $O(\frac{1}{2}k^2N)$, where N is the total number of particles and k = N in the worst case. The storage requirement is O(Nk). We prevent the explosion of the cost with k^2 by restarting the solver every m = 5000 iterations. This restarted version requires O((m+3+1/m)N+Z) multiplications, where Z is the number of nonzero elements in the sparse force-balance system matrix of size $3N \times 3N$ for the three spatial components.⁶⁹ The memory requirement reduces to O((m+2)N). For each iteration of the pressure-correction algorithm, the KSPGMRES solver is run to convergence. This is repeated at most n_{max} times. The cost for the two-step Adams-Bashforth-Moulton time-integration scheme is O(N). Likewise, remeshing has a computational cost of O(N), as each particle assigns to a constant number of neighboring grid nodes. In the present Eulerian simulation, the particle positions do not change over time. Therefore, the DC-PSE kernels only need to be calculated once at the beginning of the simulation. Evaluating the kernels is an O(N) operation. In summary, the overall computational cost of the simulation in an Eulerian frame of reference is dominated by KSPGMRES, as it needs to solve a linear system of equations for each pressurecorrection iteration, the cost of which depends on the condition number of the system. In a Lagrangian frame of reference, the DC-PSE kernels need to be recomputed at every time step, incurring an



FIG. 2. Convergence of the presented 3D active polar fluid solver for the channel-flow problem with the analytical solution given in Eq. (8). (a) Streamlines of initial (t = 0) polarity (left) and velocity (right) fields with arrow heads indicating direction and color indicating the Frank energy density and velocity magnitude, respectively. (b) Streamlines at steady state (t = 40) for the polarity (left) and velocity (right) fields. (c) Convergence of the absolute numerical errors at t = 40 for the polarity **p** and strain rate u_{xy} fields. The L_2 norm of the errors across all particles is plotted as circles (°), the L_{∞} norms as diamonds (\diamondsuit). Open symbols represent errors in the polarity field **p**, filled symbols in the strain rate field u_{xy} . The theoretically expected convergence slope is indicated by the solid line. (d) Evolution of the L_{∞} error in the numerically computed polarity magnitude $||\mathbf{p}||_2$ over time for different resolutions *h* demonstrating convergence of the Lagrange multiplier h_{\parallel} .

additional computational cost of O(N), albeit with a large pre-factor. This cost then dominates the one from the linear system solver. Overall, we thus expect the computational cost of the whole simulation to scale linearly with the number of particles *N*. This is empirically confirmed in our benchmarks with *N* ranging from 10³ to 10⁵.

The software implementation of the presented method relies on OpenFPM to leverage shared- and distributed-memory parallel computers to address the high computational cost of 3D simulations. We therefore check how the runtime of the simulation from Sec. IV A scales when distributing it across an increasing number of CPU cores. For each configuration, ten time steps are performed on Intel Xeon E5–2680v3 CPUs at 2.50 GHz with 24 cores on each compute node. Individual compute nodes are connected by a 4-lane FDR InfiniBand network (at 14 Gb/s per lane) with a latency of 0.7 μ s for message passing using the OpenMPI library. Each experiment is repeated 10 times, and the results are averaged over the 10 independent repetitions. We use the ParMETIS graph domain decomposition⁷⁰ of OpenFPM to parallelize the computation over spatial particle sets.⁵⁴

For a fixed number of particles ($N_x = 64$, $N_y = 65$, and $N_z = 5$), the number of CPU cores increases from 1 to 192, hence measuring the strong scaling of the algorithm [Fig. 3(a)]. The time measurements show an almost linear scaling within one compute node. Using more cores per node, the memory bandwidth becomes the bottleneck, such that the speedup on 24 cores (1 full node) is about 14-fold. Scaling beyond one node, communicating data via the interconnect network of the machine, the scaling slows down only slightly up until 128 cores. Using more than 128 cores, runtimes start to increase, as this problem is too small to be distributed over more than 128 cores. We next measure the weak scaling of the simulation where the number of particles increases proportionally to the number of CPU cores. For simplicity, this test is performed in a cubic computational domain and increase the number of particles from $N = 16 \times 17 \times 16$ on 1 core to $N = 164 \times 165 \times 164$ (one additional particle is required in y to impose the boundary conditions) on 1012 cores [Fig. 3(b)]. We observe perfect scaling in one node up to 4 cores, while the parallel efficiency when using all 24



FIG. 3. Scaling of the simulation wall-clock time to complete 10 time steps when parallelizing the simulation from Sec. IV A over increasingly many CPU cores. Dots show the average time over ten repetitions of each experiment with standard deviation error bars smaller than the symbol size. Dashed vertical lines show compute node boundaries beyond which communication happens over the interconnect network. (a) Strong scaling when distributing a problem of fixed size ($N_x = 64$, $N_y = 65$, and $N_z = 5$) across an increasing number of processor cores. The dotted line represents the ideal linear speedup. (b) Weak scaling when increasing the problem size proportional to the number of processor cores.

cores of a node is 58%. Scaling to more nodes, the communication overhead of the linear system solver eventually becomes limiting, as it requires global all-to-all communication with a volume that grows with the total problem size. The communication time depends on the node interconnect topology, the condition number of the system matrix, the number of GMRES iterations until convergence, the shape of the domain decomposition, and the memory bandwidth of the compute nodes. As the number of processor cores increases, communication overhead on average increases. The nonmonotic behavior of the measured times in Fig. 3(b) is not caused by measurement noise, as the error bars over the 10 independent repetitions of each run are smaller than the symbol size. Instead, these variations are deterministic. They are caused by the condition number of the system matrix changing with increasing resolution (weak scaling), the node interconnect network topology changing as more nodes are used, and the shape of the domain decomposition changing as data distributes across more cores within a node.

Taken together, these results demonstrate that the parallel OpenFPM implementation of the present code can reduce the wall-clock time of the simulations by more than one order of magnitude (strong scaling) and allows performing large simulations (weak scaling). This enables using the code for real-world applications.

V. APPLICATIONS

We illustrate the use of the present simulation method to active polar flow problems in increasingly complex 3D spatial domains and with different boundary conditions. While analytical solutions are not available in these cases, we compare with published results where possible in order to corroborate the correctness of our simulations. Together, these example simulations showcase the broad applicability of the present method.

A. Active Fréedericksz transition in extensile polar fluids

We simulate the spontaneous flow transition of an incompressible extensile active polar fluid with friction boundary conditions and $\nu \neq 0$.⁷¹ For this, we simulate the channel-flow problem from Sec. IV A in a cuboidal domain with edge lengths $L_x = L_y = L$, $L_z = 5h$, discretized with a uniform Cartesian mesh of N_{xo} $N_y = N_x + 1$, and $N_z = 5$ particles in the *x*, *y*, and *z* directions, respectively. $N_y = N_x + 1$ ensures equal spacing *h* along all dimensions, since periodic boundary conditions are imposed along *x* (and *z*), but not along *y*. At the boundaries $y = L_y$ and y = 0, the polarity is fixed to be aligned with the *x* axis. For the velocity, the friction boundary conditions are as follows:

$$\sigma_{xy}(x,0,z,t) = \mu_b v_x(x,0,z,t),$$
(11a)

$$\sigma_{xy}(x,L_y,z,t) = -\mu_t v_x(x,L,z,t), \qquad (11b)$$

$$\sigma_{zy}(x, 0, z, t) = \mu_b v_z(x, 0, z, t),$$
(11c)

$$\sigma_{zy}(x,L_y,z,t) = -\mu_t v_z(x,L,z,t), \qquad (11d)$$

where μ_t and μ_b are the friction coefficients at the top and bottom boundary, respectively. Initially, all polarity vectors point along the positive x axis. We then induce a small perturbation to the polarity vector of the central particle in the domain by setting $\theta = 0.01$ and $\varphi = \pi/2 + 0.01$, where θ and φ are the azimuthal and polar angle of the polarity vector at that particle, respectively. The remaining model and simulation parameters are: $L = L_x = L_y = 10$, $\eta = 1$, $\gamma = 1$, $\zeta = 1$, $\lambda = 1$, and $\nu = -2/5$ (i.e., flow-tumbling regime), $K_s = K_b = K_t = 1$, $\mu_t/\mu_b = 4$, $\Delta\mu = 3.0$, $\epsilon_v = 5 \times 10^{-4}$, $\epsilon_p = 10^{-16}$, $n_{max} = 30$, $\epsilon_{steady} = 10^{-9}$. We perform the simulation using the hybrid particle-mesh method due to its less restrictive CFL condition of the Lagrangian frame of reference. The spatial resolution is $N_x = 32$, the time step size is $\Delta t = 0.01$, and the final simulation time is $t_f = 200$. The initial condition with the perturbation at the center is visualized in Fig. 4(a).

In the absence of active stress, the system behaves according to equilibrium thermodynamics, and the perturbation relaxes to the



FIG. 4. Active Fréedericksz transition and transition to active turbulence in a 3D extensile polar fluid. (a) Streamlines of initial (t = 0) polarity (left) and velocity (right) fields with color indicating the Frank free-energy density and velocity magnitude, respectively. A small single-point perturbation is introduced at the center of the domain (see main text). (b) Streamlines at steady state (t = 200) for the polarity (left) and velocity (right) fields for activity $\zeta \Delta \mu = 3$, showing an active Fréedericksz transition to spontaneous flow with polarity-bending instability. (c) Numerically computed maximum finite-time Lyapunov exponent (dots: mean; error bars: standard deviation) as a function of the activity level. For increasing activity, the system transits from no flow, to flow, to traveling vortices, and eventually chaos. The increasing standard deviation in the waves regime is caused by polarity oscillations. (d) Streamlines of the polarity (left) and velocity fields (right) in the turbulent regime for $\zeta \Delta \mu = 50$ at t = 200.

homogeneous state with no flow. This also generates the expected back-flows as the polarity relaxes to the homogeneous state. Sufficiently strong active stresses, however, can generate contractile or extensile forces that permanently maintain the system out of equilibrium, according to the description from Sec. II. Indeed, for activity levels above an analytically known⁷¹ critical threshold, we observe bending of the polarity field lines. This is the well-known bending instability of extensile active polar fluids.^{2,27,72} The bending of the polarity field in the *xz* plane induces directed fluid flow in the *z* direction, causing a transition away from the initial homogeneous steady state [Fig. 4(b)]. The dynamics of this transition are characteristic of a pitchfork bifurcation. Computing the maximum velocity in the domain as a function of $\Delta \mu$ empirically identifies the critical activity for this case to be around $\Delta \mu = 2.15$, which is higher than the analytically predicted value⁷¹

$$\Delta \mu_{c} = -\frac{5\pi^{2}K(\gamma(\nu-1)^{2}+4\eta)}{2\gamma L^{2}(\nu-1)(\gamma\lambda\nu+\zeta)} = 1.75$$
(12)

for stress-free walls. In the present case, the walls have friction, for which it is known that $\Delta \mu_c \leq \Delta \mu \leq 4\Delta \mu_c$,^{49,56} which our simulation fulfills. The locations of maximum velocity magnitude coincide with the maxima in the Frank free-energy density, and both are located at the hilltops and valley bottoms of the bending deformation. The numerically solved fields also perfectly preserve translational

invariance along the z direction and the mirror symmetry of the inward and outward bends. Simulations in a cubic domain with $N_z = N_x$ show the exact same behavior. At the final time of the simulation, the maximum rate of change in the polarity field measured by the L_2 -norm stays on the order of 10^{-6} , indicating a non-equilibrium steady state. All of these observations confirm that the numerically computed solution behaves physically correctly.

B. Transition to active turbulence

Section V A has shown that the present numerical solver accurately recapitulates the space–time dynamics of the active Fréedericksz transition in extensile polar fluids when the activity exceeds the critical value $\Delta \mu_c$. Increasing the activity even more first leads to the emergence of traveling waves and traveling vortices, followed by a transition to spatiotemporal chaos. To identify these flow transitions, we numerically compute the maximum finite-time Lyapunov exponent λ_1 , which measures the exponential growth rate of a small perturbation in the system. Negative λ_1 indicate that the flow relaxes back to the unperturbed state. Positive λ_1 are indicative of chaos transitions. If the perturbation remains in the system forever, λ_1 is zero. We numerically compute λ_1 using the classic method of Benettin *et al.*⁷³ and Skokos,⁷⁴ as previously implemented.⁴⁹ Therefore, two Lagrangian simulations are run with the same model and parameters, except that the second simulation has an additional initial perturbation of $d_0 = 0.001$ to both

 θ and φ of the polarity vector of the central particle. Every $T_L = 50$ time steps, it is checked whether the difference |d| between the two simulations (measured by the L^2 -norm over all particles) increased or decreased. In order to detect the small differences in the polarity field, we decrease $\epsilon_v = 1 \times 10^{-4}$. This provides one estimation for $\lambda_1 = \frac{1}{T_L} \ln \frac{|d|}{d_0}$. Then, the perturbation is reset to d_0 . With a time step size of 0.01, a total of 800 λ_1 estimates are thus collected by the final time $t_f = 400$. We report the mean and standard deviation over the last 300 λ_1 estimates when the flow has transitioned into its final state.

A plot of the measured λ_1 (mean \pm standard deviation) as a function of $|\zeta \Delta \mu|$ is shown in Fig. 4(c) with error bars indicating the standard deviation. The "No Flow" and "Flow" regimes are characterized by a homogeneous and an inhomogeneous steady state, respectively. This transition, where λ_1 changes from being negative to being zero, therefore is the active Fréedericksz transition in extensile polar fluids as studied in Subsection V A. At an activity level of $\Delta \mu \approx 4.35$, the system starts to exhibit self-organized traveling waves of polarity and traveling vortices of velocity. Both fields travel in x direction at constant group velocity. The transition can be empirically localized by computing the maximum rate of change or the slope of the power spectrum of the polarity field. Such empirically estimated values, however, are always approximate and may vary slightly for different spatial and temporal resolutions of the simulation. Within the "Waves" regime, there is sub-transition happening around $\Delta \mu \approx 6.75$. There, the polarity field starts to exhibit oscillations leading to oscillating values of λ_1 , as evidenced by the growing standard deviation of λ_1 beyond this point. At $\Delta \mu \approx 11$, the previously oscillation frequency starts to wobble, which is a sign of an imminent transition to spatiotemporal chaos. The mean maximum Lyapunov exponent become positive beyond $\Delta \mu \approx 12$, but the oscillatory behavior still dominates at this activity. In the region between $\Delta \mu \approx 11$ and $\Delta \mu \approx 16$, the system exhibits a mixture of signatures for oscillatory vortices and chaos. This soft transition is indicated by a color gradient in Fig. 4(c). In the "Chaos" regime beyond $\Delta \mu \approx 16$, the flow fields are turbulent as quantified by a growing positive Lyapunov exponent. In this regime, we observe disclination lines in the polarity field as one-dimensional line defects preferentially forming at the walls and growing into the bulk. The defect loops move chaotically and fuse with other disclination lines upon encounter. They always form closed loops either with themselves of by starting and ending at a wall. A visualization of the polarity and velocity fields in the chaotic regime is shown in Fig. 4(d) for the same resolution $N_x = 32$. To our knowledge, this constitutes the first numerical validation of the transition to turbulence for the full three-dimensional Ericksen-Leslie hydrodynamic model with extensile stress.

C. Coherent motion in a 3D annulus

Moving to a non-Cartesian geometry, we test our algorithm and implementation in a 3D annulus. The behavior of active matter in 2D annular geometries has been extensively studied both experimentally^{75–78} and numerically.^{79–81} The main focus of these studies was on confinement-induced transitions to active turbulence. In 3D, an experimental study reported coherent motion of microtubule assays confined to an annulus,⁸² but we are not aware of any numerical simulations of this system in 3D.

We here solve Eq. (1) of active polar hydrodynamics in a 3D annulus of inner radius 1 (dimensionless units), outer radius of 5, and thickness of 2.75 in the periodic z direction. The annulus is centered at (0, 0, 1.375). At the inner and outer ring surfaces, the polarity field is anchored tangentially and parallel to the xy plane. For the velocity field, physically realistic no-slip boundary conditions are imposed. We use the same model and simulation parameters as in Subsection VB, but set $\Delta \mu = 60$. The simulation is performed in the Eulerian frame of reference, which would be harder to do in the Lagrangian reference frame in this case. At t = 0, polarity is homogeneously aligned tangentially to the annulus and parallel to the xy plane except for a small perturbation setting $\theta = 0.01$ and $\varphi = \pi/2 + 0.01$, where θ and φ are the azimuthal and polar angles, respectively, for particles with 1 < |x| < 2.5 and -1 < y < 1 [Fig. 5(a)]. We observe that the polarity develops a circular variant of the bend instability at $t \approx 5$ [Fig. 5(b)]. This bent state instability develops a sixfold angular symmetry exhibits coherent flows rotating it clockwise [Figs. 5(c) and 5(d)]. This confirms the experimentally observed⁸² emergence of coherent angular motion in 3D annular domains and suggests that extensile active materials can use the bend instability to sustain a bent mode for generating coherent motion.

D. Chiral vortices in asymmetric 3D geometries

Finally, we demonstrate the robustness and versatility of the present simulation algorithm in more challenging 3D geometries that have never been considered in the literature so far. Two different geometries are created using Blender⁸³ that resemble shapes of dividing biological cells. The first geometry is a rotationally and left-right symmetric "peanut" shape, discretized with 1952 boundary particles given by the vertices of Blender's triangular surface mesh and 10 627 bulk particles placed in a regular Cartesian grid with spacing h = 0.1. The second geometry is an asymmetric geometry created by manually indenting the peanut mesh at arbitrary locations, different for the left and right sides. It has the same number of boundary particles and 10 898 bulk particles with the same h = 0.1. At the boundary, polarity is anchored to be perpendicular to the boundary, and the velocity is zero. The initial condition consists of a random polarity field with polarity at each bulk particle sampled uniformly randomly on the unit sphere. The initial velocity field is zero everywhere. We use the same model and simulations parameters as in Sec. V A, except for $\nu = 0$ and $\epsilon_v = 5 \times 10^{-4}$. All simulations run until $t_f = 10.0$. Due to the difficulty of applying the method of images in complex geometries, we use the Eulerian frame of reference for this simulation.

Without activity, the polarity field aligns with the boundary condition, and the velocity decays to zero everywhere in the domain (not shown). This is as expected for a purely dissipative system and confirms the physical plausibility of the simulation. At $\Delta \mu = 20.0$, a steady state with nonzero flow develops in both the symmetric and the asymmetric geometries (Fig. 6). The polarity and velocity fields reflect the symmetry properties of the domain. In the symmetric shape [Figs. 6(a)-6(c)], the fields are approximately left-right symmetric, but not exactly, since the simulation starts from a random initial polarity field. Over time (panels from left to right), the polarity fields develops two stable asters in either body of the peanut, and the velocity field becomes chiral with multiple stable vortices stemming from the antisymmetric part of the stress tensor. In the asymmetric shape [Figs. 6 (d)–6(f)], a similar pattern initially forms at $t \approx 1.0$, but later one aster "eats" the other, and a strong steady-state streaming velocity is induced through the neck between the two sides of the peanut. Taken together, these simulations not only showcase the application of the present simulation framework to arbitrary 3D geometries, but predict the



FIG. 5. Coherent angular motion of an active fluid in a 3D annular domain with visualized polarity (top, color bar: Frank free-energy density) and velocity (bottom, color bar: velocity magnitude) streamlines. (a) Initial state at time t = 0. (b) Initial appearance of the circular bend instability at time t = 5. (c) Fully developed bent state rotating clockwise at time t = 7. (d) Rotation at time t = 9.

emergence of chiral flows in axially symmetric shapes and geometryinduced left-right symmetry breaking in asymmetric domain shapes.

VI. CONCLUSION

We have presented a numerical method for solving the full, symmetry-preserving 3D active Ericksen-Leslie equations with unit polarity constraint. These equations model the physics of active polar fluids from first principles. The presented method numerically solves these complex PDEs using a hybrid particle-mesh method with Discretization-Corrected Particle Strength Exchange (DC-PSE) to consistently discretize differential operators with second-order accuracy in space over potentially irregular distributions of discretization points (a.k.a. particles). This simplifies the implementation of arbitrary boundary conditions and of simulations in irregularly shaped domains. We presented an iterative pressure-correction solver for imposing incompressibility of the flow using DC-PSE in 3D and validated its second-order accurate convergence. Thanks to the flexibility afforded by DC-PSE, remeshing, and iterative pressure-correction, the method can be used in both the Eulerian and Lagrangian frames of reference. In the Eulerian frame of reference, particles do not move. Remeshing is then not needed, but the CFL condition imposes a smaller time step limit on the simulations. Detecting flow steady states and imposing boundary conditions in complex-shaped domains, however, is easier in the Eulerian frame. In the Lagrangian frame of reference, the particles move with the velocity field of the flow. This allows for larger time steps, but also incurs a higher computational cost per time step, as DC-PSE operator kernels need to be recomputed at each time step and remeshing is required. The Lagrangian frame is therefore preferable for unsteady simulations in simple domain shapes, where the method of images can be applied to impose the boundary conditions.

The software implementation of the present algorithm leverages the open-source framework OpenFPM for scalable scientific computing⁵⁴ and its template-expression system for PDEs.⁵⁵ This allows the simulations to scale on shared- and distributed-memory multi-processor parallel computers, addressing the computational challenges associated with 3D active matter problems. The use of the PDE template expression system further facilitates modifying the model equations and domain geometry without needing to change the numerical solver code. We have benchmarked the parallel scalability in both the strong and the weak sense. The scalability of the simulation was predominantly contingent on the scalability of the pressure-correction solver, which currently uses the Portable Extensible Toolkit for Scientific Computation (PETSc)⁶¹ to solve the global linear system of equations.

The main results of this work were as follows: a comprehensive derivation of the governing hydrodynamic equations of 3D active polar fluids, a detailed description of a fully consistent numerical solver algorithm, a validation of its second-order accurate numerical convergence, and an analysis of its computational performance. We further validated the proposed simulation algorithm by recapitulating known phenomena in active polar fluids, including the active Fréedericksz transition in extensile polar fluids, the transition to active turbulence, and the emergence of coherent rotational motion in a 3D annulus, which was numerically simulated for the first time here. All of these simulations compared the numerical results with experimental observations or theoretical predictions from the literature. Finally, the presented solver was applied to more complex symmetric and asymmetric 3D geometries that have never been considered in the literature so far. This predicted the emergence of chiral vortices in axially symmetric domains, as well as geometry-induced left-right symmetry breaking. Together, these simulations highlighted the versatility of our approach over a range of 3D active polar hydrodynamics problems in different geometries and with different boundary conditions.

ARTICLE

pubs.aip.org/aip/pof



FIG. 6. Developing polarity (top, color bar: Frank free-energy density) and velocity (bottom, color bar: velocity magnitude) streamlines of an extensile active polar fluid in a symmetric (a)–(c) and an asymmetric (d)–(f) peanut-like 3D geometry. From left to right, we show the flow fields for increasing time (t = 0.1, 1.0, and10.0) as indicated in the panels. For better visualization, the shapes are cut open along their axial mid-planes. Gray spheres show boundary particles. A surface rendering of one half of either shape is overlaid to the first frame as a gray, semitransparent surface.

Despite these advantages, the present approach also has a number of limitations. The most important one is its high computational cost with a single simulation time step requiring tens to hundreds of seconds of wall-clock time even on a parallel computer and for low spatial resolution. This high computational cost is partly intrinsic to the use of remeshed DC-PSE and partly limited by the parallel scalability of the third-party PETSc solver used. However, the modular architecture of OpenFPM could in the future be exploited to use alternative solvers after assembling the system matrix in a distributed fashion. This could potentially reduce the computational cost of the simulations in the future and enhance their parallel scalability. Another measure that could reduce the runtime of the simulations would be to perform them on Graphics Processing Units (GPUs). This will depend on the GPU support in OpenFPM being extended to the PDE template expression system and to particle simulations. Furthermore, the presented algorithm was not tested in moving or deforming domains, albeit we did show it to work in different complex geometries. Finally, the present algorithm is limited to solving bulk active flow problems in 3D. While it can relatively easily be adapted to solving 2D bulk problems by changing the governing equations in the code, extending it to flows in curved surfaces would involve a meshfree Surface DC-PSE method,⁸⁴ but first requires a number of mathematical questions to be resolved.

Taken together, the present work enables the numerical solution of 3D active polar fluid models by providing a convergence-validated, scalable, and open-source method capable of handling complex geometries and boundary conditions. This opens up new avenues for exploring and investigating of the rich and at times puzzling physics of active fluids and their use to model living systems, including tissue morphogenesis and animal flocking behavior.

ACKNOWLEDGMENTS

This work was funded in parts by the Federal Ministry of Education and Research (Bundesministerium für Bildung und Forschung, BMBF) under Funding Code 031L0160 (project "SPlaT-DM—computer simulation platform for topology-driven morphogenesis") and in the framework of the Federal Center for Scalable Data Analytics and Artificial Intelligence, ScaDS.AI, Dresden/Leipzig.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Abhinav Singh and Philipp H. Suhrcke contributed equally to this work.

APPENDIX A: MOLECULAR FIELD

Abhinav Singh: Conceptualization (equal); Data curation (lead); Formal analysis (equal); Investigation (equal); Methodology (lead); Resources (equal); Software (lead); Supervision (supporting); Validation (lead); Visualization (lead); Writing - original draft (lead); Writing - review & editing (supporting). Philipp Suhrcke: Data curation (equal); Formal analysis (equal); Investigation (equal); Methodology (equal); Resources (equal); Software (equal); Validation (equal); Visualization (equal); Writing - original draft (lead); Writing - review & editing (supporting). Pietro Incardona: Methodology (supporting); Resources (supporting); Software (equal); Validation (supporting); Writing - review & editing (supporting). Ivo Fabian Sbalzarini: Conceptualization (lead); Formal analysis (supporting); Funding acquisition (lead); Investigation (equal); Methodology (equal); Project administration (lead); Resources (equal); Software (supporting); Supervision (lead); Validation (equal); Visualization (supporting); Writing - original draft (supporting); Writing - review & editing (lead).

DATA AVAILABILITY

The computer codes that support the findings of this study are openly available in the 3Dactive-hydrodynamics github repository located at https://github.com/mosaic-group/3Dactive-hydrodynamics, Ref. 85.

The molecular field h_{α} is decomposed into its polarity-perpendicular $(h_{\perp,x}, h_{\perp,y}, h_{\perp,z})$ and polarity-parallel (h_{\parallel}) components

$$h_x = h_{\parallel} p_x - h_{\perp,z} p_y + h_{\perp,y} p_z \quad h_{\perp,x} = p_y h_z - p_z h_y, \tag{A1a}$$

$$h_{y} = h_{\perp,z}p_{x} + h_{\parallel}p_{y} - h_{\perp,x}p_{z}$$
 $h_{\perp,y} = p_{z}h_{x} - p_{x}h_{z},$ (A1b)

$$h_z = -h_{\perp,y}p_x + h_{\perp,x}p_y + h_{\parallel}p_z \quad h_{\perp,z} = p_x h_y - p_y h_x \,. \tag{A1c}$$

The parallel component is defined as the Lagrange multiplier in Eq. (4) to enforce $||\mathbf{p}|| = 1$. The perpendicular components are then as above.

APPENDIX B: COMPLETE FORCE-BALANCE EQUATION

x–component :

$$\begin{split} \eta &(2\partial_{xx}v_{x} + \partial_{xy}v_{y} + \partial_{yy}v_{x} + \partial_{xz}v_{z} + \partial_{zz}v_{x}) + \nu^{2}\gamma \left[\frac{1}{(p_{x}^{2} + p_{y}^{2} + p_{z}^{2})^{2}} \left[-2p_{x}p_{y}(p_{x}\partial_{y}p_{x} + p_{y}\partial_{y}p_{y} + p_{z}\partial_{y}p_{z})(p_{x}^{2}\partial_{x}v_{x} + p_{x}p_{y}(\partial_{x}v_{y} + \partial_{y}v_{x}) + p_{y}^{2}\partial_{y}v_{y} + p_{x}p_{z}(\partial_{x}v_{z} + \partial_{z}v_{x}) + p_{y}p_{z}(\partial_{y}v_{z} + \partial_{z}v_{y}) + p_{z}^{2}\partial_{z}v_{z}) - 2p_{x}p_{z}(p_{x}\partial_{z}p_{x} + p_{y}\partial_{z}p_{y} + p_{z}\partial_{z}p_{z})(p_{x}^{2}\partial_{x}v_{x} + p_{x}p_{y}(\partial_{x}v_{y} + \partial_{y}v_{x}) + p_{y}^{2}\partial_{y}v_{y} + p_{x}p_{z}(\partial_{x}v_{z} + \partial_{z}v_{x}) + p_{y}p_{z}(\partial_{y}v_{z} + \partial_{z}v_{y}) + p_{z}^{2}\partial_{z}v_{z}) + \frac{1}{3}(2(p_{x}\partial_{x}p_{x} + p_{y}\partial_{x}p_{y} + p_{z}\partial_{x}p_{z})(-2p_{x}^{2} + p_{y}^{2} + p_{z}^{2})(p_{x}^{2}\partial_{x}v_{x} + p_{x}\partial_{y}p_{y} + p_{x}p_{y}(\partial_{x}v_{y} + \partial_{y}v_{x}) + p_{y}^{2}\partial_{y}v_{y} + p_{x}p_{z}(\partial_{x}v_{z} + \partial_{z}v_{x}) + p_{y}p_{z}(\partial_{y}v_{z} + \partial_{z}v_{y}) + p_{z}^{2}\partial_{z}v_{z}) \right] + \frac{1}{p_{x}^{2} + p_{y}^{2} + p_{z}^{2}} \left[\left(\frac{4p_{x}\partial_{x}p_{x}}{3} + p_{x}\partial_{y}p_{y} + p_{x}\partial_{z}p_{z} - \frac{2p_{y}\partial_{x}p_{y}}{3} + p_{y}\partial_{y}p_{x} - \frac{2p_{z}\partial_{x}p_{z}}{3} + p_{z}\partial_{z}p_{x}} \right) (p_{x}^{2}\partial_{x}v_{x} + p_{x}p_{y}(\partial_{x}v_{y} + \partial_{y}v_{x}) + p_{y}^{2}\partial_{y}v_{y} + p_{x}p_{z}(\partial_{x}v_{z} + \partial_{z}v_{y}) + p_{y}\partial_{y}v_{z} + p_{x}\partial_{x}p_{y}(\partial_{x}v_{y} + \partial_{y}v_{x}) + p_{y}\partial_{x}p_{z}(\partial_{x}v_{z} + \partial_{z}v_{y}) + p_{y}\partial_{x}p_{z}(\partial_{x}v_{z} + \partial_{z}v_{y}) + p_{y}\partial_{y}p_{y}(\partial_{x}v_{y} + \partial_{y}v_{x}) + p_{y}\partial_{x}p_{x}\partial_{x}v_{x} + p_{x}^{2}\partial_{x}v_{x} + p_{x}^{2}\partial_{x}v_{x} + p_{x}\partial_{x}p_{y}(\partial_{x}v_{y} + \partial_{y}v_{x}) + p_{y}\partial_{x}p_{z}(\partial_{x}v_{x} + \partial_{x}v_{y}) + p_{y}\partial_{x}p_{z}(\partial_{x}v_{y} + \partial_{y}v_{x}) + p_{y}\partial_{x}p_{z}(\partial_{x}v_{y} + \partial_{y}v_{y}) + p_{x}\partial_{x}p_{z}(\partial_{x}v_{y} + \partial_{y}v_{y}) + p_{x}\partial_{x}p_{z}(\partial_{x}v_{y} + \partial_{y}v_{y}$$

ARTICLE

pubs.aip.org/aip/pof

$$= -\left[-0.5p_{x}h_{\perp,y}\partial_{z}p_{x} + 0.5p_{x}h_{\perp,z}\partial_{y}p_{x} + 0.5p_{y}h_{\perp,x}\partial_{z}p_{x} + 0.5p_{y}h_{\perp,z}\partial_{y}p_{y} + 0.5p_{y}h_{\perp,z}\partial_{z}p_{z} - 0.5p_{x}(-h_{\perp,x}\partial_{z}p_{y} + h_{\perp,y}\partial_{z}p_{x} + p_{x}\partial_{z}h_{\perp,y}) - p_{y}\partial_{z}h_{\perp,x}) - 0.5p_{z}h_{\perp,x}\partial_{y}p_{x} - 0.5p_{z}h_{\perp,y}\partial_{y}p_{y} - 0.5p_{z}h_{\perp,y}\partial_{z}p_{z} + 0.5p_{x}(-h_{\perp,x}\partial_{y}p_{z} + h_{\perp,z}\partial_{y}p_{x} + p_{x}\partial_{y}h_{\perp,z} - p_{z}\partial_{y}h_{\perp,x}) + 0.5p_{y}(-h_{\perp,y}\partial_{y}p_{z} + h_{\perp,z}\partial_{y}p_{y} + p_{y}\partial_{y}h_{\perp,z} - p_{z}\partial_{y}h_{\perp,y}) - 0.5p_{z}(h_{\perp,y}\partial_{z}p_{z} - h_{\perp,z}\partial_{z}p_{y} - p_{y}\partial_{z}h_{\perp,z} + p_{z}\partial_{z}h_{\perp,y}) + 0.5\nu\left[-p_{x}h_{\perp,x}\partial_{y}p_{z} + p_{x}h_{\perp,x}\partial_{z}p_{y} + 2p_{x}h_{\perp,y}\partial_{x}p_{z} - 2p_{x}h_{\perp,y}\partial_{z}p_{x} - 2p_{x}h_{\perp,z}\partial_{x}p_{y} + 2p_{x}h_{\perp,z}\partial_{y}p_{x} + p_{x}^{2}\partial_{y}h_{\perp,z} - p_{x}^{2}\partial_{z}h_{\perp,y} + p_{y}h_{\perp,x}\partial_{z}p_{x} + p_{y}h_{\perp,y}\partial_{y}p_{z} - 2p_{y}h_{\perp,z}\partial_{x}p_{x} - 2p_{x}h_{\perp,z}\partial_{x}p_{y} - 2p_{x}h_{\perp,z}\partial_{y}p_{x} + p_{x}^{2}\partial_{y}h_{\perp,z} - p_{z}^{2}\partial_{z}h_{\perp,y} + p_{y}h_{\perp,x}\partial_{z}p_{x} + p_{z}h_{\perp,y}\partial_{y}p_{y} + 2p_{x}h_{\perp,z}\partial_{x}p_{y} - 2p_{x}h_{\perp,z}\partial_{x}p_{y} + 2p_{x}h_{\perp,z}\partial_{y}p_{x} + p_{x}^{2}\partial_{y}h_{\perp,z} - p_{z}^{2}\partial_{z}h_{\perp,y} + p_{y}h_{\perp,x}\partial_{z}p_{x} + p_{z}h_{\perp,y}\partial_{y}p_{z} - 2p_{y}h_{\perp,z}\partial_{x}p_{x} - 2p_{x}h_{\perp,y}\partial_{x}p_{x} - 2p_{x}h_{\perp,z}\partial_{x}p_{y} - 2p_{x}h_{\perp,z}\partial_{y}p_{x} + p_{x}p_{y}\partial_{z}h_{\perp,z} - p_{z}^{2}\partial_{x}h_{\perp,z} - p_{z}^{2}\partial_{x}h_{\perp,y} + p_{y}h_{\perp,x}\partial_{z}p_{x} + p_{z}h_{\perp,y}\partial_{y}p_{y} + 2p_{z}h_{\perp,y}\partial_{z}p_{x} - 2p_{x}h_{\perp,y}\partial_{y}p_{y} - p_{x}p_{z}\partial_{z}h_{\perp,z} + p_{y}p_{z}\partial_{y}h_{\perp,z} - p_{z}h_{\perp,x}\partial_{y}p_{x} + 2p_{z}h_{\perp,y}\partial_{x}p_{x} + p_{z}h_{\perp,y}\partial_{y}p_{y} + 2p_{x}h_{\perp,y}\partial_{z}p_{x} - 2p_{x}h_{\perp,z}\partial_{y}p_{y} - p_{x}p_{z}\partial_{y}h_{\perp,z} - p_{y}\partial_{y}h_{\perp,z} - p_{z}h_{\perp,z}\partial_{y}p_{x} + 2p_{z}h_{\perp,y}\partial_{y}p_{y} + p_{z}h_{\perp,y}\partial_{y}p_{y} + 2p_{z}h_{\perp,y}\partial_{z}p_{x} + 2p_{z}h_{\perp,y}\partial_{y}p_{x} + 2p_{z}h_{\perp,y}\partial_{y}p_{x} + 2p_{z}h_{\perp,y}\partial_{y}p_{x} + 2p_{z}h_{\perp,y}\partial_{y}p_{x} + 2p_{z}h_{\perp,y}\partial_{y}p_{x} + p_{z}h_{\perp,y}\partial_{y}p_{x} + 2p_{z}h_{\perp,y}\partial_{y}p_{x} + 2p_{z}h_{\perp,y}\partial_{z}p_{x} + 2p_{z}h_{\perp,y}\partial_{y}p_{x} + 2p$$

y–component :

$$\begin{split} \eta(2\partial_{yy}v_{y} + \partial_{xx}v_{y} + \partial_{yy}v_{z} + \partial_{xz}v_{z} + \partial_{xz}v_{y}) + v^{2}\gamma \left[\frac{1}{(p_{x}^{2} + p_{y}^{2} + p_{z}^{2})^{2}} \left[-2p_{x}p_{y}(p_{x}\partial_{x}p_{x} + p_{y}\partial_{x}p_{y} + p_{z}\partial_{x}p_{z})(p_{z}^{2}\partial_{x}v_{z} + p_{x}p_{y}(\partial_{x}v_{y} + \partial_{y}v_{z}) \\ &+ p_{y}^{2}\partial_{y}v_{y} + p_{x}p_{z}(\partial_{x}v_{z} + \partial_{z}v_{z}) + p_{y}p_{z}(\partial_{y}v_{z} + \partial_{z}v_{z}) + p_{z}^{2}\partial_{z}v_{z}) - 2p_{y}p_{z}(p_{x}\partial_{z}p_{x} + p_{y}\partial_{z}p_{y} + p_{z}\partial_{z}p_{z})(p_{z}^{2}\partial_{x}v_{z} + p_{x}p_{y})(\partial_{x}v_{y} + p_{y}v_{z}) \\ &+ p_{y}^{2}\partial_{y}v_{y} + p_{x}p_{z}(\partial_{x}v_{z} + \partial_{z}v_{z}) + p_{y}p_{z}(\partial_{y}v_{z} + \partial_{z}v_{y}) + p_{z}^{2}\partial_{z}v_{z}) + \frac{1}{3}(2(p_{x}\partial_{y}p_{x} + p_{y}\partial_{z}p_{y} + p_{z}\partial_{z}p_{z})(p_{x}^{2}\partial_{x}v_{x} + p_{y}p_{y}(\partial_{x}v_{y} + \partial_{y}v_{z})) \\ &+ \frac{4p_{y}\partial_{y}p_{y}}{9} + p_{y}\partial_{z}p_{z} - \frac{2p_{z}\partial_{y}p_{z}}{3} + p_{z}\partial_{z}p_{y})(p_{z}^{2}\partial_{x}v_{z} + p_{z}p_{y}(\partial_{x}v_{z} + \partial_{y}v_{z}) + p_{z}^{2}\partial_{x}v_{z} + p_{z}^{2}\partial_{y}v_{z} + p_{z}p_{z}\partial_{x}v_{z}) \\ &+ \frac{4p_{y}\partial_{y}p_{y}}{3} + p_{y}\partial_{z}p_{z} - \frac{2p_{z}\partial_{y}p_{z}}{3} + p_{z}\partial_{z}p_{y})(p_{z}^{2}\partial_{x}v_{z} + p_{z}p_{y}(\partial_{x}v_{z} + \partial_{y}v_{z}) + p_{y}p_{z}(\partial_{y}v_{z} + \partial_{z}v_{z}) + p_{y}p_{z}(\partial_{y}v_{z} + \partial_{z}v_{z}) \\ &+ p_{z}^{2}\partial_{z}v_{z}) + p_{y}p_{y}(2p_{x}\partial_{x}p_{z} + p_{z}\partial_{x}v_{z}) + p_{z}^{2}\partial_{x}v_{z} + p_{z}\partial_{y}p_{z}v_{z} + p_{z}\partial_{y}p_{z}(\partial_{y}v_{z} + \partial_{y}v_{z}) + p_{z}\partial_{z}p_{z}\partial_{z}v_{z} \\ &+ p_{x}p_{y}(\partial_{x}v_{z} + \partial_{x}v_{z}) + p_{x}\partial_{x}p_{z}(\partial_{x}v_{z} + \partial_{z}v_{z}) + p_{z}\partial_{x}p_{z}(\partial_{x}v_{z} + \partial_{y}v_{z}) + p_{z}\partial_{y}p_{z}(\partial_{y}v_{z} + \partial_{z}v_{z}) + p_{z}\partial_{z}p_{z}\partial_{z}v_{z} \\ &+ p_{x}p_{y}(\partial_{x}v_{z} + \partial_{z}v_{z}) + p_{y}\partial_{y}p_{z}(\partial_{y}v_{z} + \partial_{z}v_{z}) + p_{z}\partial_{y}p_{z}(\partial_{y}v_{z} + \partial_{y}v_{z}) + p_{z}\partial_{y}p_{z}(\partial_{x}v_{z} + \partial_{z}v_{z}) + p_{z}\partial_{y}p_{z}\partial_{z}v_{z} \\ &+ p_{x}p_{z}(\partial_{x}v_{z} + \partial_{z}v_{z}) + p_{y}\partial_{y}p_{z}(\partial_{y}v_{z} + \partial_{z}v_{z}) + p_{z}\partial_{y}p_{z}(\partial_{y}v_{z} + \partial_{y}v_{z}) + p_{z}\partial_{y}p_{z}\partial_{z}v_{z} \\ &+ p_{y}\partial_{y}p_{z}(\partial_{x}v_{z} + \partial_{x}v_{z}) + p_{y}\partial_{y}p_{z}(\partial_{y}v_{z} + \partial_{y}v_{z}) + p_{z}\partial_{y}p_{z$$

z-component :

$$\begin{split} \eta (2\partial_{zz}v_{z} + \partial_{xx}v_{z} + \partial_{yx}v_{z} + \partial_{yy}v_{z} + \partial_{yz}v_{y}) + v^{2}\gamma \bigg[\frac{1}{(p_{x}^{2} + p_{y}^{2} + p_{z}^{2})^{2}} \bigg[-2p_{x}p_{z}(p_{x}\partial_{x}p_{x} + p_{y}\partial_{x}p_{y} + p_{z}\partial_{y}p_{z})(p_{x}^{2}\partial_{x}v_{x} + p_{x}p_{y}(\partial_{x}v_{y} + \partial_{y}v_{x}) \\ &+ p_{y}^{2}\partial_{y}v_{y} + p_{x}p_{z}(\partial_{x}v_{z} + \partial_{v}v_{x}) + p_{y}p_{z}(\partial_{y}v_{z} + \partial_{z}v_{y}) + p_{z}^{2}\partial_{z}v_{y} - 2p_{y}p_{z}(p_{x}\partial_{y}p_{x} + p_{y}\partial_{y}p_{y} + p_{z}\partial_{y}p_{z})(p_{x}^{2}\partial_{x}v_{x} + p_{x}p_{y}(\partial_{x}v_{y} + \partial_{y}v_{x}) \\ &+ p_{y}^{2}\partial_{y}v_{y} + p_{x}p_{z}(\partial_{x}v_{z} + \partial_{v}v_{x}) + p_{y}p_{z}(\partial_{x}v_{z} + \partial_{z}v_{y}) + p_{z}^{2}\partial_{z}v_{z} + \frac{1}{3}(2(p_{x}\partial_{x}p_{x} + p_{y}\partial_{y}p_{y} + p_{z}\partial_{z}p_{z})(p_{x}^{2}\partial_{x}v_{x} + p_{x}p_{y}(\partial_{x}v_{y} + \partial_{y}v_{y})) \bigg] + \frac{1}{p_{x}^{2} + p_{y}^{2}} + p_{z}^{2}(p_{x}^{2})(p_{x}^{2}\partial_{x}v_{x} + p_{y}p_{y}(\partial_{x}v_{y} + \partial_{y}v_{y}) + p_{y}p_{z}(\partial_{x}v_{z} + \partial_{z}v_{y}) + p_{z}p_{z}(\partial_{x}v_{z} + \partial_{z}v_{y})) \bigg] + \frac{1}{p_{x}^{2} + p_{y}^{2}} + p_{z}^{2}(p_{x}^{2})(p_{x}^{2}\partial_{x}v_{x} + p_{y}\partial_{y}p_{y} + p_{z}\partial_{z}p_{z})(p_{y}^{2}\partial_{x}v_{x} + p_{x}p_{y}(\partial_{x}v_{y} + \partial_{y}v_{x}) + p_{y}p_{z}(\partial_{x}v_{z} + \partial_{x}v_{y}) + p_{z}p_{z}(\partial_{x}v_{z} + \partial_{x}v_{y}) + p_{z}p_{z}(\partial_{x}v_{z} + \partial_{z}v_{z}) + p_{z}p_{z}(\partial_{x}v_{z} + \partial_{x}v_{y}) + p_{z}p_{z}(\partial_{x}v_{z} + \partial_{x}v_{y}) + p_{z}p_{z}(\partial_{x}v_{z} + \partial_{z}v_{y}) + p_{z}p_{z}(\partial_{x}v_{z} + \partial_{x}v_{y}) + p_{z}p_{z}(\partial_{x}v_{z} + \partial_{x}v_{y}) + p_{z}\partial_{x}p_{z}(\partial_{x}v_{z} + \partial_{x}v_{y}) + p_{z}\partial_{x}p_{z}(\partial_{$$

REFERENCES

- ¹F. J. Nédélec, T. Surrey, A. C. Maggs, and S. Leibler, "Self-organization of microtubules and motors," Nature **389**, 305–308 (1997).
- ²F. Bonelli, G. Gonnella, A. Tiribocchi, and D. Marenduzzo, "Spontaneous flow in polar active fluids: The effect of a phenomenological self propulsion-like term," Eur. Phys. J. E **39**, 1–10 (2016).
- ³K. T. Stanhope, V. Yadav, C. D. Santangelo, and J. L. Ross, "Contractility in an extensile system," Soft Matter **13**, 4268–4277 (2017).
- ⁴M. Dogterom and G. H. Koenderink, "Actin–microtubule crosstalk in cell biology," Nat. Rev. Mol. Cell Biol. 20, 38–54 (2019).
- ⁵G. Duclos, R. Adkins, D. Banerjee, M. S. E. Peterson, M. Varghese, I. Kolvin, A. Baskaran, R. A. Pelcovits, T. R. Powers, A. Baskaran, F. Toschi, M. F. Hagan, S. J. Streichan, V. Vitelli, D. A. Beller, and Z. Dogic, "Topological structure and dynamics of three-dimensional active nematics," *Science* 367, 1120–1124 (2020).
- ⁶N. Kralj, M. Ravnik, and Ž. Kos, "Defect line coarsening and refinement in active nematics," Phys. Rev. Lett. **130**, 128101 (2023).

- ⁷J. Binysh, Ž. Kos, S. Čopar, M. Ravnik, and G. P. Alexander, "Three-dimensional active defect loops," Phys. Rev. Lett. **124**, 088001 (2020).
- ⁸K. Kruse, J. F. Joanny, F. Jülicher, J. Prost, and K. Sekimoto, "Generic theory of active polar gels: A paradigm for cytoskeletal dynamics," Eur. Phys. J. E 16, 5–16 (2005).
- ⁹F. Jülicher, K. Kruse, J. Prost, and J. F. Joanny, "Active behavior of the cytoskeleton," Phys. Rep. **449**, 3–28 (2007).
- ¹⁰J. F. Joanny and J. Prost, "Active gels as a description of the actin-myosin cytoskeleton," HFSP J. 3, 94–104 (2009).
- ¹¹C. Dombrowski, L. Cisneros, S. Chatkaew, R. E. Goldstein, and J. O. Kessler, "Self-concentration and large-scale coherence in bacterial dynamics," Phys. Rev. Lett. **93**, 098103 (2004).
- ¹²M. C. Marchetti, J. F. Joanny, S. Ramaswamy, T. B. Liverpool, J. Prost, M. Rao, and R. A. Simha, "Hydrodynamics of soft active matter," Rev. Mod. Phys. 85, 1143–1189 (2013).
- ¹³J. Toner and Y. Tu, "Long-range order in a two-dimensional dynamical XY model: How birds fly together," Phys. Rev. Lett. 75, 4326–4329 (1995).

30 October 2023 13:32:10

(B3)

- ¹⁴F. Jülicher, S. W. Grill, and G. Salbreux, "Hydrodynamic theory of active matter," Rep. Prog. Phys. 81, 076601 (2018).
- ¹⁵J. Roostalu, J. Rickman, C. Thomas, F. Nédélec, and T. Surrey, "Determinants of polar versus nematic organization in networks of dynamic microtubules and mitotic motors," Cell **175**, 796–808 (2018).
- ¹⁶C. A. Lugo, E. Saikia, and F. J. Nédélec, "A typical workflow to simulate cytoskeletal systems," J. Visualized Exp. **194**, e64125 (2023).
- ¹⁷W. Yan, S. Ansari, A. Lamson, M. A. Glaser, R. Blackwell, M. D. Betterton, and M. Shelley, "Toward the cellular-scale simulation of motor-driven cytoskeletal assemblies," eLife 11, e74160 (2022).
- ¹⁸J. A. Anderson, C. D. Lorenz, and A. Travesset, "General purpose molecular dynamics simulations fully implemented on graphics processing units," J. Comput. Phys. 227, 5342–5359 (2008).
- ¹⁹T. D. Nguyen, C. L. Phillips, J. A. Anderson, and S. C. Glotzer, "Rigid body constraints realized in massively-parallel molecular dynamics on graphics processing units," Comput. Phys. Commun. **182**, 2307–2313 (2011).
- ²⁰ M. Spellings, M. Engel, D. Klotsa, S. Sabrina, A. M. Drews, N. H. P. Nguyen, K. J. M. Bishop, and S. C. Glotzer, "Shape control and compartmentalization in active colloidal cells," Proc. Natl. Acad. Sci. U. S. A. **112**, E4642–E4650 (2015).
- ²¹M. Agrawal, I. R. Bruss, and S. C. Glotzer, "Tunable emergent structures and traveling waves in mixtures of passive and contact-triggered-active particles," Soft Matter 13(37), 6332–6339 (2017).
- ²²L. N. Carenza, G. Gonnella, A. Lamura, G. Negro, and A. Tiribocchi, "Lattice Boltzmann methods and active fluids," Eur. Phys. J. E 42, 81 (2019).
- ²³C. Denniston, E. Orlandini, and J. M. Yeomans, "Simulations of liquid crystal hydrodynamics in the isotropic and nematic phases," Europhys. Lett. 52, 481–487 (2000).
- ²⁴C. Denniston, E. Orlandini, and J. M. Yeomans, "Lattice Boltzmann simulations of liquid crystal hydrodynamics," Phys. Rev. E 63, 056702 (2001).
- ²⁵C. Denniston, D. Marenduzzo, E. Orlandini, and J. M. Yeomans, "Lattice Boltzmann algorithm for three-dimensional liquid-crystal hydrodynamics," Philos. Trans. R. Soc., A 362, 1745–1754 (2004).
- ²⁶L. Giomi, L. Mahadevan, B. Chakraborty, and M. F. Hagan, "Excitable patterns in active nematics," Phys. Rev. Lett. **106**, 218101 (2011).
- E. Tjhung, M. E. Cates, and D. Marenduzzo, "Nonequilibrium steady states in polar active fluids," Soft Matter 7, 7453–7464 (2011).
 S. P. Thampi, R. Golestanian, and J. M. Yeomans, "Velocity correlations in an
- ²⁸S. P. Thampi, R. Golestanian, and J. M. Yeomans, "Velocity correlations in an active nematic," Phys. Rev. Lett. **111**, 118101 (2013).
- ²⁹S. P. Thampi, R. Golestanian, and J. M. Yeomans, "Vorticity, defects and correlations in active turbulence," Philos. Trans. R. Soc., A **372**, 20130366 (2014).
- ³⁰S. Chandragiri, A. Doostmohammadi, J. M. Yeomans, and S. P. Thampi, "Flow states and transitions of an active nematic in a three-dimensional channel," *Phys. Rev. Lett.* **125**, 148002 (2020).
- ³¹M. R. Nejad and J. M. Yeomans, "Active extensile stress promotes 3D director orientations and flows," Phys. Rev. Lett. **128**, 048001 (2022).
- ³²L. J. Ruske and J. M. Yeomans, "Activity gradients in two- and threedimensional active nematics," Soft Matter 18, 5654–5661 (2022).
- ³³D. Marenduzzo, E. Orlandini, M. E. Cates, and J. M. Yeomans, "Steady-state hydrodynamic instabilities of active liquid crystals: Hybrid lattice Boltzmann simulations," Phys. Rev. E 76, 031921 (2007).
- ³⁴D. Marenduzzo, E. Orlandini, and J. M. Yeomans, "Hydrodynamics and rheology of active liquid crystals: A numerical investigation," Phys. Rev. Lett. 98, 118102 (2007).
- ³⁵U. Frisch, B. Hasslacher, and Y. Pomeau, "Lattice-gas automata for the Navier-Stokes equation," Phys. Rev. Lett. 56, 1505 (1986).
- ³⁶Y. G. Yong, C. L. Lin, and J. Huang, "Accuracy and efficiency study of lattice Boltzmann method for steady-state flow simulations," Numer. Heat Transfer, Part B 39, 21–43 (2001).
- ³⁷E. J. Hemingway, P. Mishra, M. C. Marchetti, and S. M. Fielding, "Correlation lengths in hydrodynamic models of active nematics," Soft Matter 12, 7943–7952 (2016).
- ³⁸T. Gao, M. D. Betterton, A.-S. Jhang, and M. J. Shelley, "Analytical structure, dynamics, and coarse graining of a kinetic model of an active fluid," Phys. Rev. Fluids 2(9), 093302 (2017).
- ³⁹L. Ohm and M. J. Shelley, "Weakly nonlinear analysis of pattern formation in active suspensions," J. Fluid Mech. **942**, A53 (2022).

- ⁴⁰S. Weady, D. B. Stein, and M. J. Shelley, "Thermodynamically consistent coarse-graining of polar active fluids," Phys. Rev. Fluids 7, 063301 (2022).
- ⁴¹B. Chakrabarti, M. J. Shelley, and S. Fürthauer, "Collective motion and pattern formation in phase-synchronizing active fluids," Phys. Rev. Lett. **130**, 128202 (2023).
- ⁴²V. Bratanov, F. Jenko, and E. Frey, "New class of turbulence in active fluids," Proc. Natl. Acad. Sci. U. S. A. **112**, 15048–15053 (2015).
- ⁴³R. Alert, J.-F. Joanny, and J. Casademunt, "Universal scaling of active nematic turbulence," Nat. Phys. 16, 682–688 (2020).
- ⁴⁴R. Assante, D. Corbett, D. Marenduzzo, and A. Morozov, "Active turbulence and spontaneous phase separation in inhomogeneous extensile active gels," Soft Matter 19, 189 (2023).
- ⁴⁵M. M. Hejlesen, G. Winckelmans, and J. H. Walther, "Non-singular Green's functions for the unbounded Poisson equation in one, two and three dimensions," Appl. Math. Lett. 89, 28–34 (2019).
- ⁴⁶M. M. Hejlesen, J. T. Rasmussen, P. Chatelain, and J. H. Walther, "A high order solver for the unbounded Poisson equation," J. Comput. Phys. 252, 458–467 (2013).
- ⁴⁷M. M. Hejlesen, J. T. Rasmussen, P. Chatelain, and J. H. Walther, "High order Poisson solver for unbounded flows," Proceedia IUTAM 18, 56–65 (2015).
- ⁴⁸R. Ramaswamy, G. Bourantas, F. Jülicher, and I. F. Sbalzarini, "A hybrid particle-mesh method for incompressible active polar viscous gels," J. Comput. Phys. 291, 334–361 (2015).
- ⁴⁹R. Ramaswamy and F. Jülicher, "Activity induces traveling waves, vortices and spatiotemporal chaos in a model actomyosin layer," Sci. Rep. 6, 20838 (2016).
- spatiotemporal chaos in a model actomyosin layer," Sci. Rep. 6, 20838 (2016).
 ⁵⁰L. Giomi, M. J. Bowick, P. Mishra, R. Sknepnek, and M. C. Marchetti, "Defect dynamics in active nematics," Philos. Trans. R. Soc., A 372, 20130365 (2014).
- ⁵¹L. Giomi, "Geometry and topology of turbulence in active nematics," Phys. Rev. X 5, 031003 (2015).
- ⁵²M. Varghese, A. Baskaran, M. F. Hagan, and A. Baskaran, "Confinementinduced self-pumping in 3D active fluids," Phys. Rev. Lett. **125**, 268003 (2020).
- ⁵³B. Schrader, S. Reboux, and I. F. Sbalzarini, "Discretization correction of general integral PSE operators for particle methods," J. Comput. Phys. 229, 4159–4182 (2010).
- ⁵⁴P. Incardona, A. Leo, and Y. Zaluzhnyi, "OpenFPM: A scalable open framework for particle and particle-mesh codes on parallel computers," Comput. Phys. Commun. 241, 155–177 (2019).
- ⁵⁵A. Singh, P. Incardona, and I. F. Sbalzarini, "A C++ expression system for partial differential equations enables generic simulations of biological hydrodynamics," Eur. Phys. J. E 44, 117 (2021).
- ⁵⁶R. Voituriez, J. F. Joanny, and J. Prost, "Spontaneous flow transition in active polar gels," Europhys. Lett. **70**, 404–410 (2005).
- ⁵⁷P. G. de Gennes and J. Prost, *The Physics of Liquid Crystals*, International Series of Monographs on Physics, 2nd ed. (Clarendon Press, Oxford, England, 1995).
- ⁵⁸P. K. Papadopoulos, "An auxiliary potential velocity method for incompressible viscous flow," Comput. Fluids 51, 60–67 (2011).
- ⁵⁹G. C. Bourantas and V. C. Loukopoulos, "A meshless scheme for incompressible fluid flow using a velocity-pressure correction method," Comput. Fluids 88, 189-199 (2013).
- ⁶⁰G. Bourantas, B. L. Cheeseman, R. Ramaswamy, and I. F. Sbalzarini, "Using DC PSE operator discretization in Eulerian meshless collocation methods improves their robustness in complex geometries," Comput. Fluids 136, 285–300 (2016).
- ⁶¹S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith, "Efficient management of parallelism in object-oriented numerical software libraries," in *Modern Software Tools for Scientific Computing*, edited by E. Arge, A. M. Bruaset, and H. P. Langtangen (Birkhäuser, Boston, MA, 1997), pp. 163–202.
- ⁶²A. Singh, L. Kraatz, P. Incardona, and I. F. Sbalzarini, "A distributed algebra system for time integration on parallel computers," arXiv:2309.05331 (2023).
- ⁶³J. H. Ferziger and M. Perić, Computational Methods for Fluid Dynamics (Springer, Berlin Heidelberg, 2002).
- ⁶⁴G.-H. Cottet, J.-M. Etancelin, F. Perignon, and C. Picard, "High order semi-Lagrangian particle methods for transport equations: Numerical analysis and implementation issues," ESAIM: M2AN 48, 1029–1060 (2014).
- ⁶⁵Although these methods are commonly referred to as "interpolation" in the literature, they are not actually interpolating, as they do in general not maintain the exact field values at the original data points.

- ⁶⁶A. Magni and G.-H. Cottet, "Accurate, non-oscillatory, remeshing schemes for particle methods," J. Comput. Phys. 231, 152–172 (2012). ⁶⁷J. J. Monaghan, "Extrapolating B splines for interpolation," J. Comput. Phys.
- 60(2), 253-262 (1985).
- ⁶⁸Y. Saad and M. H. Schultz, "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems," SIAM J. Sci. Stat. Comput. 7(3), 856-869 (1986).
- ^{69}Z is related to N in a nontrivial way, as it depends on the DC-PSE kernel coefficients and the particle neighborhood used to evaluate the spatial derivatives when constructing the system matrix. The DC-PSE kernel coefficients differ from particle to particle, and so does the distribution of particles in a Lagrangian simulation.
- 70G. Karypis and V. Kumar, "A parallel algorithm for multilevel graph partitioning and sparse matrix ordering," J. Parallel Distrib. Comput. 48(1), 71-95 (1998).
- 71 A. Singh, Q. Vagne, F. Jülicher, and I. F. Sbalzarini, "Spontaneous flow instabilities of active polar fluids in three dimensions," Phys. Rev. Res. 5, L022061 (2023).
- 72S. Ramaswamy and M. Rao, "Active-filament hydrodynamics: Instabilities, boundary conditions and rheology," New J. Phys. 9, 423 (2007).
- 73G. Benettin, L. Galgani, A. Giorgilli, and J. M. Strelcyn, "Lyapunov characteristic exponents for smooth dynamical systems and for Hamiltonian systems; A method for computing all of them. Part 2: Numerical application," Meccanica 15, 21-30 (1980).
- 74C. Skokos, "The Lyapunov characteristic exponents and their computation," in Dynamics of Small Solar System Bodies and Exoplanets (Springer, 2009), pp. 63-135.
- 75A. Opathalage, M. M. Norton, M. P. N. Juniper, B. Langeslay, S. A. Aghvami, S. Fraden, and Z. Dogic, "Self-organized dynamics and the transition to turbulence of confined active nematics," Proc. Natl. Acad. Sci. U. S. A. 116(11), 4788-4797 (2019).

- 76 J. Hardoüin, J. Laurent, T. Lopez-Leon, J. Ignés-Mullol, and F. Sagués, "Active microfluidic transport in two-dimensional handlebodies," Soft Matter 16(40), 9230-9241 (2020).
- 77 J. Hardoüin, C. Doré, J. Laurent, T. Lopez-Leon, J. Ignés-Mullol, and F. Sagués, "Active boundary layers in confined active nematics," Nat. Commun. 13(1), 6675 (2022).
- 78C. Joshi, Z. Zarei, M. M. Norton, S. Fraden, A. Baskaran, and M. F. Hagan, "From disks to channels: Dynamics of active nematics confined to an annulus," arXiv:2304.04895 (2023).
- ⁷⁹M. Neef and K. Kruse, "Generation of stationary and moving vortices in active polar fluids in the planar Taylor-Couette geometry," Phys. Rev. E 90(5), 052703 (2014).
- 80S. Chen, P. Gao, and T. Gao, "Dynamics and structure of an apolar active suspension in an annulus," J. Fluid Mech. 835, 393-405 (2018).
- ⁸¹T. N. Shendruk, A. Doostmohammadi, K. Thijssen, and J. M. Yeomans, "Dancing disclinations in confined active nematics," Soft Matter 13(21), 3853-3862 (2017).
- 82K.-T. Wu, J. B. Hishamunda, D. T. N. Chen, S. J. DeCamp, Y.-W. Chang, A. Fernández-Nieves, S. Fraden, and Z. Dogic, "Transition from turbulent to coherent flows in confined three-dimensional active fluids," Science 355(6331), eaal1979 (2017).
- 83B. O. Community, Blender—A 3D Modelling and Rendering Package (Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018).
- ⁸⁴A. Singh, A. Foggia, P. Incardona, and I. F. Sbalzarini, "A meshfree collocation scheme for surface differential operators on point clouds," J. Sci. Comput. 96(3), 89 (2023).
- 85See https://github.com/mosaic-group/3Dactive-hydrodynamics for the C++ source code of the presented examples, which use the OpenFPM simulation framework located at https://github.com/mosaic-group/openfpm_pdata.