6-1-2019

Convergent Solutions of Stokes Oldroyd-B Boundary Value Problems using the Immersed Boundary Smooth Extension (IBSE) Method

David B. Stein
Simons Foundation

Robert D. Guy
University of California, Davis

Becca Thomases
University of California, Davis, bthomases@smith.edu

Convergent solutions of Stokes Oldroyd-B boundary value problems using the Immersed Boundary Smooth Extension (IBSE) Method

David B. Stein[a,*], Robert D. Guy[b], Becca Thomases[b]

[a] *Center for Computational Biology, Flatiron Institute, New York, NY 10010, USA*
[b] *Department of Mathematics, University of California, Davis, Davis, CA 95616, USA*

Abstract

The Immersed Boundary (IB) method has been widely used to solve fluid-structure interaction problems, including those where the structure interacts with polymeric fluids. In this paper, we examine the convergence of one such scheme for a well known two-dimensional benchmark flow for the Oldroyd-B constitutive model, and we show that the traditional IB-based scheme fails to adequately capture the polymeric stress near to embedded boundaries. We analyze the reason for such failure, and we argue that this feature is not specific to the case study chosen, but a general feature of such methods due to lack of convergence in velocity gradients near interfaces. In order to remedy this problem, we build a different scheme for the Oldroyd-B system using the Immersed Boundary Smooth Extension (IBSE) scheme, which provides convergent viscous stresses near boundaries. We show that this modified scheme produces convergent polymeric stresses through the whole domain, including on embedded boundaries, and produces solutions in good agreement with known benchmarks.

Keywords: Complex Fluids, Oldroyd-B, Immersed Boundary, Complex geometry, Partial Differential Equations, High-order

1. Introduction

The Immersed Boundary (IB) method was originally developed for the study of moving, deformable structures immersed in a fluid, and it has been widely applied to such problems since its introduction [1–3]. Recently, the method has been adapted to more general fluid-structure problems, including the motion of rigid bodies immersed in a fluid [4], fluid flow through a domain with either stationary boundaries or boundaries with prescribed motion [5, 6], and fluid-structure problems in which the boundaries interact with a polymeric fluid [7–12]. In this broadened context, we use the term *Immersed Boundary method* to refer only to methods in which (*i*) the boundary is treated as a Lagrangian structure embedded in a geometrically simple domain, (*ii*) the background PDE (e.g. the Navier-Stokes equations) are solved on a Cartesian grid everywhere in that domain, and (*iii*) all communication between the Lagrangian structure and the underlying PDE is mediated by convolutions with regularized δ-functions. These methods have many desirable properties: they make use of robust and efficient Cartesian-grid methods for solving the underlying PDE, are flexible to a wide range of problems, and are simple to implement, requiring minimal geometric information and processing to describe the boundary. The accuracy and convergence properties of the IB method have been well-studied for Netwonian fluids, but it has not been carefully validated against known benchmarks for polymeric fluids. In this paper, we examine the IB method applied to a classical test problem: the flow of an Oldroyd-B fluid at zero Reynolds number past a stationary cylinder in confinement. This problem has been extensively studied [13–17], and high quality benchmarks are available at low values

*Corresponding author
Email address: dstein@flatironinstitute.org (David B. Stein)

of the Weissenberg number. We show that the IB method, uncorrected, produces incorrect stresses in near-boundary regions, and we identify the cause of this simply as a failure of the viscous stress tensor to converge in these regions. The errors in the velocity gradients enter into the evolution equation for the polymeric stress, and cause persistent errors that do not converge with grid refinement. We then summarize a recently introduced modification of the IB method, the Immersed Boundary Smooth Extension method [18, 19], which captures the viscous stress tensor accurately near boundaries, and show that this method coupled to a standard pseudo-spectral Oldroyd-B solver produces accurate solutions that converge to known benchmarks.

In this paper, we first consider coupling an unaltered Immersed Boundary scheme to a pseudo-spectral solver for the Stokes Oldroyd-B (SOB) equations for a viscoelastic fluid to simulate polymeric flows in two-dimensional stationary complex geometries. The evolution of the polymeric stress in the SOB equations is an ODE along streamlines that contains growth terms dependent on the gradient of the fluid velocity $\nabla \mathbf{u}$. When fixed obstacles exist in a flow, large velocity gradients are typically present in near-boundary regions. For many embedded boundary methods, including the IB method, gradients of the velocity field are not captured accurately in the near-boundary region: for general flows they show persistent $\mathcal{O}(1)$ errors that do not disappear with grid refinement [19]. It would thus be surprising if the IB method, unmodified, provided accurate solutions for the polymeric stress near to boundaries, and to our knowledge the convergence properties of these schemes when applied to complex, nonlinear equations such as the SOB system has not been rigorously analyzed. In this paper, we show that it is indeed the case that this quantity fails to converge in the L^∞ norm; with large errors precisely in the near-boundary regions where velocity gradients are not accurately.

Significant effort has been expended on improving the accuracy of embedded boundary schemes [23–39], an overview is provided in [18]. In addition, methods which build on the IB or Immersed Finite Element (IFEM) method but with modifications such as the use of one-sided interpolation and spread operators near to boundaries have been used to model flow past fixed objects and deformable particles in a viscoelastic fluid [41–43]; these works have shown that averaged flow features are resolved but convergence of the stress at and near boundaries is not considered. A subset of these improved embedded boundary methods [18, 19, 39] generate solutions to the fluid equations that are globally smooth in a simple domain, and as such allow discretizations of more complicated, nonlinear equations to be constructed in a way that is nearly unaltered from solvers used on simple geometries. In [18], we demonstrate that the Immersed Boundary Smooth Extension (IBSE) method allows for simple and accurate discretizations of the 2D viscous Burgers' and Fitzhugh Nagumo equations, and in [19], we show that the solver applied to the Stokes and Navier-Stokes equations is able to obtain high-accuracy solutions for all elements of the Newtonian stress tensor, including velocity gradients, up to the boundary. In light of these observations, it is natural to consider using the IBSE method, rather than the IB method, as the underlying fluid solver when constructing an embedded boundary scheme for the SOB equation in complex geometries. In the remainder of the paper, we show that a simple scheme based on the IBSE method and a Fourier pseudo-spectral evolution scheme for the SOB equations indeed converges, up to and on embedded boundaries, allowing the consistent computation of local tractions imparted by the polymeric fluid onto immersed obstacles. To the best of our knowledge, these results provide the first rigorous demonstration of an embedded boundary scheme providing L^∞ convergence of the polymeric stress up to and on embedded boundaries for the flow of a polymeric fluid — including convergence to numerically challenging benchmarks [14–16].

This paper is organized as follows: in Section 2, we describe the Stokes Oldroyd-B (SOB) model, a pseudo-spectral evolution scheme for the simulation of the SOB system in a periodic rectangle, and a standard benchmark flow experiment. In Section 3, we show how this method can be simply extended to complex geometries using the IB method, show that this method fails to provided convergent solutions for the polymeric stress near to domain boundaries, and provide a brief analysis explaining this inaccuracy. In Section 4, we show how to alter this scheme to instead use the IBSE method, demonstrate convergence for the relatively simple flow for which the IB based scheme failed, and finally show convergence results for more challenging numerical benchmarks at higher Weissenberg number.

2. Evolution of the Stokes Oldroyd-B Equation

The total stress ($\tau_{\text{tot}} = \tau_{\text{N}} + \tau_{\text{P}}$) in an incompressible polymeric fluid is often decomposed into a Newtonian portion ($\tau_{\text{N}} = \eta_s \dot{\gamma} - P\mathbb{I}$) and a polymeric portion (τ_{P}), where $\mathbf{u}$ is the fluid velocity, η_s is the Newtonian (solvent) viscosity, P is the pressure, $\mathbb{I}$ is the identity tensor, and $\dot{\gamma}$ is the strain rate tensor $\dot{\gamma} = \nabla\mathbf{u} + (\nabla\mathbf{u})^\intercal$. Letting λ denote the relaxation time of the polymers, we assume that the polymeric stress (τ_{P}) evolves as

$$\tau_{\text{P}} + \lambda \overset{\nabla}{\tau_{\text{P}}} = H(\tau_{\text{P}}, \dot{\gamma}), \tag{1}$$

where the notation $\overset{\nabla}{\tau_{\text{P}}}$ denotes the upper-convected time derivative of τ_{P}:

$$\overset{\nabla}{\tau_{\text{P}}} = \partial_t \tau_{\text{P}} + \mathbf{u} \cdot \nabla \tau_{\text{P}} - (\nabla\mathbf{u})^\intercal \tau_{\text{P}} - \tau_{\text{P}} \nabla\mathbf{u}. \tag{2}$$

We have taken the convention that $(\nabla\mathbf{u})_{ij} = \partial_{x_i} \mathbf{u}_j$. With η_p the polymer viscosity, the Oldroyd-B model is given by $H(\tau_{\text{P}}, \dot{\gamma}) = \eta_p \dot{\gamma}$; other common models may be represented with different choices of the function H, e.g. for the Giesekus model, $H(\tau_{\text{P}}, \dot{\gamma}) = \eta_p \dot{\gamma} - \alpha \frac{\lambda}{\eta_p} {\tau_{\text{P}}}^2$ [44]. The symmetric positive-definite conformation tensor σ is related to the polymeric stress by $\tau_{\text{P}} = \frac{\eta_p}{\lambda}(\sigma - \mathbb{I})$. Substituting this into eq. (1), along with $H(\tau_{\text{P}}, \dot{\gamma}) = \eta_p \dot{\gamma}$, and noting that $\overset{\nabla}{\mathbb{I}} = -\dot{\gamma}$ leads to an evolution equation for σ:

$$\overset{\nabla}{\sigma} = \frac{-1}{\lambda}(\sigma - \mathbb{I}). \tag{3}$$

Rescaling the pressure P as $p = P/\eta_s$, an external force $\mathbf{F}$ as $\mathbf{f} = \mathbf{F}/\eta_s$, and defining a coupling constant $\xi = \eta_p/(\eta_s\lambda)$ gives the system

$$-\Delta\mathbf{u} + \nabla p = \xi \nabla\cdot\sigma + \mathbf{f}, \tag{4a}$$
$$\nabla\cdot\mathbf{u} = 0, \tag{4b}$$
$$\overset{\nabla}{\sigma} = -\lambda^{-1}(\sigma - \mathbb{I}). \tag{4c}$$

Further rescaling space by a system size L and time by an inverse strain rate L/U (where U is a typical velocity scale) leads to a nondimensional system of the same form with the relaxation time λ replaced by the Weissenberg number $\text{Wi} = U\lambda/L$. In the literature, it is common to work with a constant β, the ratio of Newtonian viscosity to total viscosity, given by $\beta = \eta_s/\eta$, where the total viscosity η is defined to be the sum of the Newtonian viscosity (η_s) and the polymeric viscosity (η_p). λ and β are related by $\xi = \frac{1-\beta}{\beta\lambda}$.

2.1. A pseudo-spectral evolution scheme for the SOB system

We initially consider solving the Stokes Oldroyd-B equations on the periodic torus $\mathbb{T}^2 = [0, 2\pi] \times [0, 2\pi]$ in two spatial dimensions, with the flow driven by a force $\mathbf{f}(x, y, t)$, with zero polymeric stress at $t = 0$:

$$-\Delta\mathbf{u} + \nabla p = \xi\nabla\cdot\sigma + \mathbf{f}(x, y, t) \qquad \text{in } \mathbb{T}^2, \tag{5a}$$
$$\nabla\cdot\mathbf{u} = 0 \qquad \text{in } \mathbb{T}^2, \tag{5b}$$
$$\overset{\nabla}{\sigma} = -\lambda^{-1}(\sigma - \mathbb{I}) \qquad \text{in } \mathbb{T}^2, \tag{5c}$$
$$\sigma(x, y, 0) = \mathbb{I}. \tag{5d}$$

Note that zero polymeric stress corresponds to $\sigma(x, y, 0) = \mathbb{I}$, and that no initial condition is needed for $\mathbf{u}$ — as we are solving the *Stokes* Oldroyd-B model, $\mathbf{u}(x, y, 0)$ is determined by the solution to

$$-\Delta\mathbf{u}(x, y, 0) + \nabla p(x, y, 0) = \xi\nabla\cdot\sigma(x, y, 0) + \mathbf{f}(x, y, 0) \qquad \text{in } \mathbb{T}^2, \tag{6a}$$
$$\nabla\cdot\mathbf{u}(x, y, 0) = 0 \qquad \text{in } \mathbb{T}^2. \tag{6b}$$

Throughout this paper, we will be interested primarily in steady-state behavior. To discretize this system in time, we thus consider $\mathbf{u}$ to be fixed, advance the equation for σ forward in time by Δt, and then recompute $\mathbf{u}$ using the updated value of σ. The Runge-Kutta 4 (RK4) timestepping scheme is well suited to evolving the convection dominated evolution equation for σ [46], although in Section 4.4 we will utilize an IMEX-BDF based scheme. Derivatives are computed spectrally by utilizing the Fast Fourier Transform (FFT), and nonlinear terms are computed pseudo-spectrally with the filter defined in [45].

Analytically, it is known that if σ is initially positive-definite, then the solution $\sigma(x, y, t)$ to Equation (5) remains positive-definite [47]. Unfortunately, the numerical scheme described above does not retain this property, and a loss of positive-definiteness of σ is typically accompanied by numerical instability and failure [17]. Several numerical solutions have been proposed for this problem, including one in which the matrix logarithm of σ is advanced in time [48], and one in which the matrix square-root of σ is advanced in time [49]. We will use the matrix square-root method: letting $b = \sqrt{\sigma}$ be the unique positive-definite matrix square-root of σ, we solve the evolution equation [49]:

$$\partial_t b + \mathbf{u} \cdot \nabla b = b \nabla \mathbf{u} + ab + \frac{1}{2\lambda}\left(b^{-\intercal} - b\right), \tag{7}$$

where a is a skew symmetric matrix with off-diagonal elements depending on b and $\nabla \mathbf{u}$. As in the previous discussion for the evolution of σ, this evolution is accomplished using explicit pseudo-spectral methods. Once b has been updated, to update $\mathbf{u}$ we first compute $\sigma = b^2$, and then solve for $\mathbf{u}$ as before. Since $\sigma = b^2$, σ is guaranteed to stay at least positive semi-definite [49].

The preceding discussion is sufficient to evolve the SOB system in a periodic box. Many flows of interest are set in more complex domains, for which we will need more complex algorithms. In this paper, we introduce two: one based on the IB method (Section 3), and one based on the IBSE method (Section 4.2). Before introducing these methods, we discuss a standard benchmark problem that we will use to assess the quality of the solutions produced by each method.

2.2. A standard benchmark Stokes Oldroyd-B flow and flow characteristics

A standard benchmark flow for evaluating numerical solvers for polymeric flow problems is the confined flow of a fluid around a cylinder in a channel, studied in [14–16, 50]. The behavior of the flow is well understood at low Weissenberg numbers but the behavior of the flow is unknown for $\mathrm{Wi} \gtrsim 0.7$ (for this problem, $\mathrm{Wi} = \lambda$). The typical computational setup is in a rectangular domain, $[-20, 20] \times [-2, 2]$ with no-slip ($\mathbf{u} = 0$) boundaries imposed on the bottom and top of the domain, the inflow condition $\mathbf{u}(-20, y) = (u_{\mathrm{in}}(y), 0)$ with

$$u_{\mathrm{in}}(y) = \frac{3}{2}\left(1 - \frac{y^2}{4}\right), \tag{8}$$

and an outflow boundary condition (which may differ depending on the computational setup). The parameter $\beta = \frac{\eta_s}{\eta}$ is fixed at 0.59.

In order to assess the quality of the solutions, we examine two quantities. The first is the scalar drag coefficient, which may be computed as

$$C_d = \frac{1}{\eta}\int_0^{2\pi} \left[\left(\tau_{\mathrm{P}xx} + \eta_s \dot{\gamma}_{xx} - P\right)\cos\theta + \left(\tau_{\mathrm{P}xy} + \eta_s \dot{\gamma}_{xy}\right)\sin\theta\right] \, \mathrm{d}\theta. \tag{9}$$

This benchmark has been computed to high accuracy using a variety of methods [13, 51–63], with close agreement at low values of Wi. In comparing the IB and IBSE based methods, it will be sufficient to focus only on the very low Weissenberg number case of $\mathrm{Wi} = 0.1$, for which there is good agreement in the literature for a value of $C_d = 130.364$. The second quantity that we examine is the value of the xx-component of the extra polymeric stress (τ_{P}) near to and on the cylinder walls. Although exact values are not reported in the literature, the value may be inferred from plots to be between 17 and 19 [16].

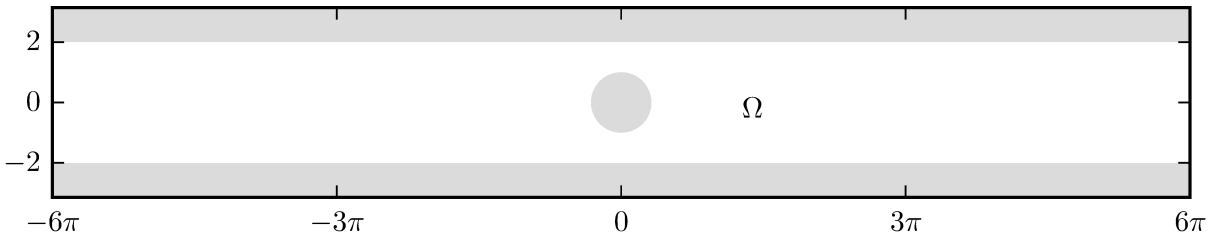


Figure 1: The computational setup we will use to solve the confined flow around a cylinder problem for a Stokes Oldroyd-B fluid. Note that in contrast to [14–16, 50], the domain is $12\pi \approx 37.7$ long, instead of 40 long. The domain Ω where fluid is present is shown in white, the entire computational domain C is the full box. What we will refer to as the extension domain E in Section 4, is shown in gray.

3. Evolution of the SOB system in complex geometry using the IB method

In the classical IB method, known forces on an interface defined by the parametrized curve $\mathbf{X}(s)$ are transferred to a grid by convolving those forces with a regularized delta function δ_h:

$$\mathbf{f}(s) = (S\mathbf{F})(\mathbf{x}) = \int_{\mathbf{X}(s)} \mathbf{F}(s)\delta_h(\mathbf{x} - \mathbf{X}(s))\, ds, \tag{10}$$

the fluid equations are solved for $\mathbf{u}$, and the structure is moved with the fluid velocity according to

$$\mathbf{U}(s) = (S^*\mathbf{u})(s) = \int_C \mathbf{u}(\mathbf{x})\delta_h(\mathbf{X}(s) - \mathbf{x})\, dx. \tag{11}$$

The operators S and S^* are typically referred to as the *spread* and *interpolation* operators, respectively.

Boundary value problems are slightly different. For these problems, the velocities on the boundary are known, while the forces that impose these boundary conditions are unknown, and the fluid may be defined only in some subset Ω of a larger, simple domain C (see Figure 1). In the tether point method [6], these forces are approximated by connecting the boundary to artificial fixed points using springs. In the direct forcing IB method [4, 5], these forces are instead computed directly as Lagrange multipliers that serve to enforce the velocity boundary conditions. For a general Stokes problem, set in a domain Ω, we may solve the simple computational domain C by adding singular forces $\mathbf{F}$ supported on the boundary that act as Lagrange multipliers which enforce the boundary condition. For Dirichlet problems this can be represented as

$$-\Delta\mathbf{u} + \nabla p + S\mathbf{F} = \mathbf{f} \qquad \text{in } C, \tag{12a}$$

$$\nabla \cdot \mathbf{u} = 0 \qquad \text{in } C, \tag{12b}$$

$$S^*\mathbf{u} = \mathbf{U}_b. \tag{12c}$$

In order to simulate the confined channel flow past a cylinder geometry described in Section 2.2, we embed the channel into the larger periodic box $[-6\pi, 6\pi] \times [-\pi, \pi]$, denoted by C. The entire computational setup is shown in Figure 1. Internal boundaries representing the channel walls are placed at $y = -2$ and $y = 2$, along the entire length of the channel, and around the circle of radius 1 centered at the point $(0, 0)$. The IB method will be used to enforce a no-slip $\mathbf{u} = 0$ boundary condition along the channel walls, and at the surface of the cylinder, corresponding to $\mathbf{U}_b = 0$ in Equation (12). Discretization of the S and S^* operators has been covered in detail elsewhere [1]. For these computations we use the fourth-order C^3 regularized δ function introduced in [18], so that the results are directly comparable to those given in Section 4. We have run the identical simulations with other choices of the regularized δ functions, and find little difference in the results. We note that the length of the channel is $12\pi \approx 37.7$, which is slightly less than the channel length of 40 used in other studies [14–16, 50]. In this setup, direct imposition of the inflow and outflow boundary

conditions is nontrivial. Instead, we approximate these by adding a constant forcing to the domain whose magnitude is treated as a Lagrange multiplier to enforce the average inflow condition

$$\int_{-2}^{2} u(-6\pi, y)\, dy = 1, \tag{13}$$

discretized using Simpsons rule. Because of the fact that the IB method relies on simple, underlying Cartesian grid solvers, integration of the pseudo-spectral solver for the polymeric stress and the IB solver for the Stokes equations is simple. The full system to be evolved is:

$$\partial_t b + \mathbf{u}\cdot\nabla b = b\nabla\mathbf{u} + ab + \frac{1}{2\lambda}\left(b^{-1} - b\right), \qquad \text{in } C, \tag{14a}$$

$$-\Delta\mathbf{u} + \nabla p + S\mathbf{F} - \alpha\hat{x} = \xi\nabla\cdot\sigma \qquad \text{in } C, \tag{14b}$$

$$\nabla\cdot\mathbf{u} = 0 \qquad \text{in } C, \tag{14c}$$

$$S^*\mathbf{u} = 0, \tag{14d}$$

$$\int_{-2}^{2} u(-6\pi, y)\, dy = 1. \tag{14e}$$

This system is evolved in exactly the same way as is described in Section 2.1, with the exception that $\mathbf{u}(t+\Delta t)$ is found by solving the Stokes equations as given in Equations (14b) to (14e).

3.1. Results: Flow past a cylinder at low Wi *using the IB-SOB solver*

To analyze the quality of the solutions produced by the IB-SOB scheme introduced in Section 3, we solve the flow around a cylinder problem described in Section 2.2 for both a coarse discretization ($n_y = 64$), and a relatively fine discretization ($n_y = 256$), where n_y denotes the number of points discretizing the domain in the span-wise direction. For all computations the time-step is set to $\Delta t = 0.64/n_y$, and the simulations are run to $t_{\text{final}} = 20\text{Wi} = 2$.

The stress in the upper half of the channel, in a region near the cylinder, is shown in Figure 2. Benchmark solutions are qualitatively different, with stresses that are larger and maximized at the boundary [14–16, 50], while the solutions produced by the IB method are smaller and maximized away from the boundary. Interpolating the stress to the boundary (by computing the stress in C and applying S^*) produces stresses that are far smaller than the benchmark solution (with maximum value no greater than 6 at all discretizations tested), and does not converge to the known solution (with a maximum value between 17 and 19) with grid refinement.

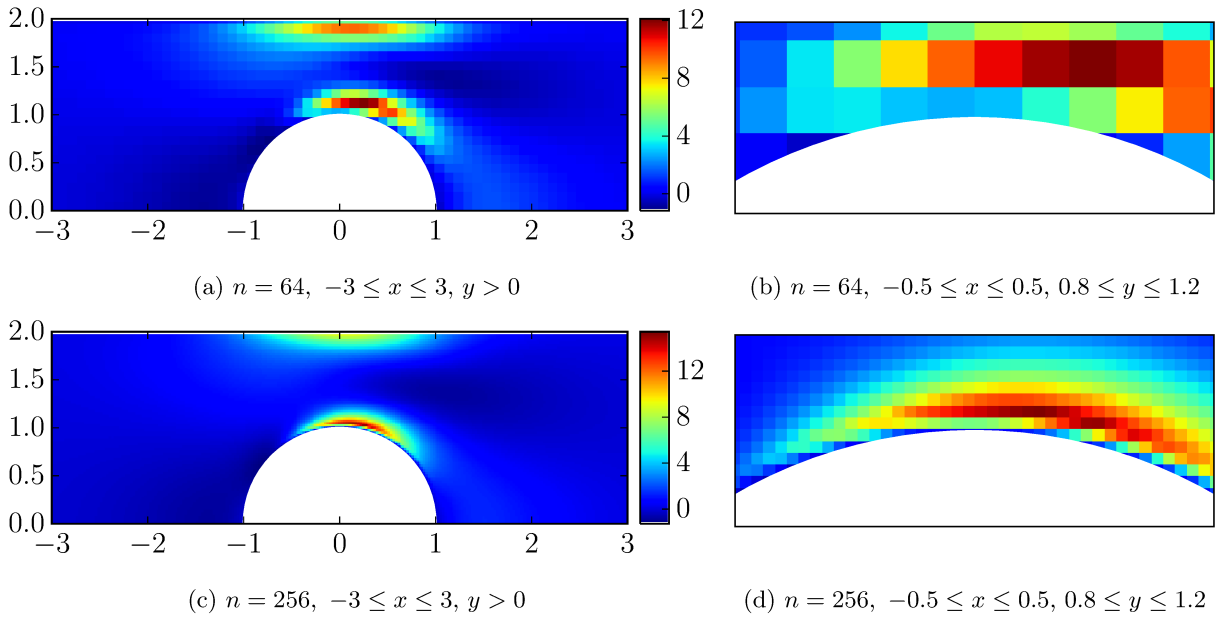
The drag-coefficient, as defined by Equation (9), involves an integral of elements of the viscous and polymeric stress tensor on the boundary. Because these fail to converge near to the boundary, computation of this quantity by computing the relevant quantities in the domain, interpolating to the boundary using S^*, and computing the integral given in Equation (9) gives an incorrect result. Instead, we may compute the drag coefficient as a sum over the singular forces:

$$C_d = \frac{1}{\eta}\int_{\Gamma} \eta_s \mathbf{F}\cdot\hat{x}\, \mathrm{d}S. \tag{15}$$

Note that $\mathbf{F}$ gives the jump in stress, for this problem the interior solution converges to 0; for the flow of an incompressible, viscous Newtonian fluid around a cylinder, this is known to provide a consistent result [23]. For $n_y = 64$ and $n_y = 256$, we find values of $C_d = 173.951$ and $C_d = 138.647$, giving errors (as computed to the benchmark value of 130.364) of approximately 33% and 6.4%, respectively. Perhaps surprisingly, this appears convergent, despite the inaccuracy in the stress in the near-boundary region.

3.2. Analysis of Errors in the IB-SOB solver

To understand the failure for the convergence of the stress in the preceding section, we turn to the simple one-dimensional Poisson problem. We will solve $\Delta u = f$, where C is the periodic interval $[0, 2\pi]$,



(a) $n = 64,\ -3 \leq x \leq 3,\ y > 0$

(b) $n = 64,\ -0.5 \leq x \leq 0.5,\ 0.8 \leq y \leq 1.2$

(c) $n = 256,\ -3 \leq x \leq 3,\ y > 0$

(d) $n = 256,\ -0.5 \leq x \leq 0.5,\ 0.8 \leq y \leq 1.2$

Figure 2: The xx-component of the polymeric stress τ_{P} for the Stokes Oldroyd-B flow around the cylinder problem, computed by the Immersed Boundary method, with $n_y = 64$ (a, b) and $n_y = 256$ (c, d), and Weissenberg number $\mathrm{Wi} = 0.1$. In (a, c), the stress $\tau_{\mathrm{P}xx}$ is shown in the upper half of the channel ($y > 0$), for $-3 \leq x \leq 3$. In (b, d), a zoom near the cylinder wall is shown, with $-0.5 \leq x \leq 0.5$ and $0.8 \leq y \leq 1.2$.

the physical domain Ω is chosen to be $[0, 2\pi] \setminus [3, 4]$, and the extension domain E is taken to be $(3, 4)$. We take $f = \sin x$ to be given in Ω, and seek to impose homogeneous Dirichlet boundary conditions enforced at $x = 3$ and $x = 4$. Some choice of f must be made in E. We will explore two options:

$$\tilde{f}_1 = \begin{cases} \sin x, & x \in \Omega, \\ \sin x, & x \in E, \end{cases} \tag{16}$$

and

$$\tilde{f}_2 = \begin{cases} \sin x, & x \in \Omega, \\ 0, & x \in E, \end{cases} \tag{17}$$

In the first case, $\tilde{f}_1 \in C^\infty(C)$, while in the second case $\tilde{f}_2$ is piecewise smooth but discontinuous at the boundary points $x = 3$ and $x = 4$. We now solve the problem analytically in the two subdomains Ω and E. Results are shown in Figure 3; in Panel (a) for $\tilde{f}_1$ and in Panel (b) for $\tilde{f}_2$. For both choices of u, the solution is only C^0 across the interface, despite the different regularity of the input forcing $\tilde{f}$.

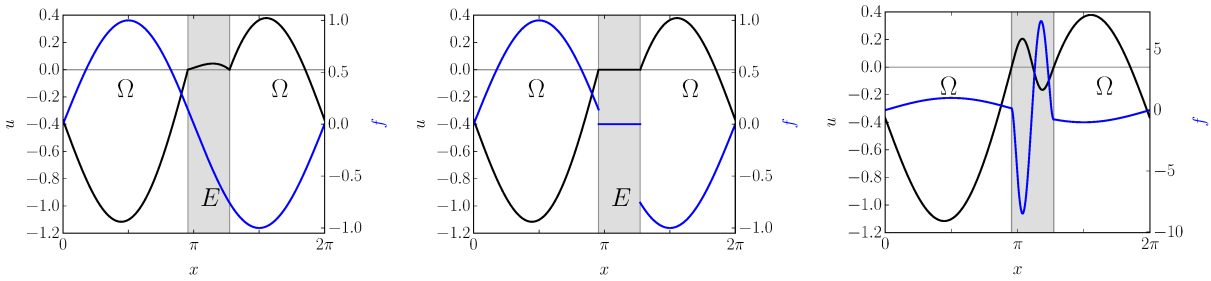
We define $\tilde{u}$ as the function that equals u_Ω in Ω, u_E in E, and 0 on Γ (with subscripts corresponding to the subscripts on the forces, when used). Notice that both $\tilde{u}_1$ and $\tilde{u}_2$ are piecewise smooth, the derivatives have jumps across the Γ. Differentiating twice, we thus find that:

$$\Delta\tilde{u} = \tilde{f} + [\tilde{u}']_3\delta_3 + [\tilde{u}']_4\delta_4, \tag{18}$$

where δ_a is the Dirac-delta distribution centered at the point a and the bracket notation $[u']_a$ is used to denote the jump in the given quantity across the point $x = a$, i.e $[u']_a = \lim_{x\to a^+} u'(x) - \lim_{x\to a^-} u'(x)$. Solving on C, rather than on Ω, thus requires the addition of singular forces supported at Γ:

$$\Delta\tilde{u} + A_3\delta_3 + A_4\delta_4 = \tilde{f} \qquad \text{in } C, \tag{19a}$$

$$\tilde{u} = 0 \qquad \text{on } \Gamma. \tag{19b}$$



(a) Solution and forcing for $\tilde{f}_1$ (b) Solution and forcing for $\tilde{f}_2$ (c) Solution and forcing for $\tilde{f}_3$

Figure 3: The solution to the Laplace problem $\Delta u = \tilde{f}$; $u(3) = u(4) = 0$, for different choices of $\tilde{f}$. In (a), the forcing function $\tilde{f}_1$ is taken to be $C^\infty(C)$, yet the solution u is only continuous. This is the generic case — in (b), a different choice is shown where $\tilde{f}_2$ is not even continuous, with the same regularity of the solution. Only for very specific choices of $\tilde{f}$, such as $\tilde{f}_3$, is the solution u smooth, as shown in panel (c). Note that $\tilde{f}$ is the same in Ω in all graphs, but has been rescaled in Panel (c). The computation of extended forcing functions that give smooth u will be discussed in Section 4.1.

Note that $A_a = -[\tilde{u}']_a$, and depends on the choice of $\tilde{f}$. The constants A_3 and A_4 may be thought of as Lagrange multipliers that act to enforce the boundary condition that $\tilde{u} = 0$ on Γ, and that these singular forces correspond to the regularized singular forces added to enforce the boundary condition in Equation (12). The consequence of the addition of these singular forces is that the analytic solution is in general not smooth, and thus boundary-naive discretizations fail to converge rapidly: using a Fourier spectral discretization, the solution u converges at only first-order in the grid spacing h, while the derivative $\tilde{u}'$ *fails to converge* near to the boundary.

For the Stokes equations as discretized via Equation (12), the situation is similar: fluid velocities will be continuous and converge, elements of the stress tensor have discontinuities and fail to converge near to embedded boundaries. We restate here the evolution equation for the conformation tensor σ in the Oldroyd-B model:

$$\sigma_t + \mathbf{u} \cdot \nabla\sigma - (\nabla\mathbf{u})^\intercal\sigma - \sigma\nabla\mathbf{u} = -\frac{1}{\mathrm{Wi}}(\sigma - \mathbb{I}). \tag{20}$$

The nonlinear growth terms in this evolution equation depend on $\nabla\mathbf{u}$, and failure of convergence of $\nabla\mathbf{u}$ in near-boundary regions leads to failure of convergence for the polymeric stress in the same locations. We comment that this is different from the case of Navier-Stokes, where the evolution of u depends on gradients of the velocity only in the convective term, and there is additional smoothing that is not present in the Oldroyd-B model.

Finally, we note that it is possible to choose extended functions $\tilde{f}$ for which the solution $\tilde{u}$ is smooth. An example is shown in Figure 3c. For this choice, the solution $\tilde{u}$ will converge rapidly, and both u and u' will agree with the analytic solution in Ω. If such extended forcing functions could be chosen in general, we might expect to capture the evolution of the polymeric stress correctly, since $\nabla\mathbf{u}$ should be captured accurately, even in near boundary regions. We discuss the computation of such functions in the following section.

4. Evolution of the SOB system in complex geometry using the IBSE method

Having identified the reason for the failure of the polymeric stress to converge as inaccuracies in the fluid velocity gradients, we turn to remedying this problem. In Section 3.2, we make two observations: (1) failure of $\nabla\mathbf{u}$ to converge is due to the low global regularity of the solution $\tilde{\mathbf{u}}$ to the problem set in C, and (2), there is a freedom to choose f in E which we have not exploited. In [19], we show that choosing f in an appropriate manner leads to a solution $\mathbf{u}$ that converges rapidly, with consistent estimates of all elements of the fluid stress tensor including the velocity gradients $\nabla\mathbf{u}$. A full explanation of the method is presented in [19], here we present a brief review.

4.1. Review of the IBSE method

It is considerably simpler to explain the Immersed Boundary Smooth Extension (IBSE) method in the context of the Poisson, rather than the Stokes, equation. Although some details differ, the basic ideas are the same, and the differences are carefully analyzed in [19]. Suppose then that we wish to solve the Poisson problem

$$\Delta u = f \qquad \text{in } \Omega, \tag{21a}$$

$$u = g \qquad \text{on } \Gamma, \tag{21b}$$

in an arbitrary smooth domain Ω. The IBSE method works by smoothly extending the unknown solution u from the physical domain Ω to the simpler computational domain C. We assume that the boundary $\Gamma = \partial\Omega$ is smooth, not self-intersecting, and must separate C into the two disjoint regions Ω and the extension domain $E = C \setminus \overline{\Omega}$. We denote an extension of the unknown solution u by ξ. This extension of the solution is then used to define a volumetric forcing $\mathcal{F}_e = \Delta\xi$ in the region E. With this forcing, an extended problem in all of the simple domain C may be solved:

$$\Delta u_e - \chi_E \mathcal{F}_e = \chi_\Omega f \qquad \text{in } C, \tag{22a}$$

$$u_e = g \qquad \text{on } \Gamma. \tag{22b}$$

The solution u_e gives the desired solution u in Ω and is equal to ξ in E. Because ξ was chosen to be a smooth extension to u, the function u_e is globally smooth in C.

The extension ξ to the unknown function u is defined as a solution to a high-order PDE which takes for its boundary conditions matching criteria of the form $\partial^j\xi/\partial n^j = \partial^j u\partial n^j$. This allows the extension to be defined by a small number of unknowns (proportional to the number of points used to discretize the boundary). The extension PDE for ξ is solved efficiently in the simple domain C using an Immersed Boundary type method.

In order to succinctly describe the methodology we require some additional notation. We define the *spread operator*:

$$(S_{(j)}F)(\mathbf{x}) = (-1)^j \int_\Gamma F_j(s) \frac{\partial^j \delta(\mathbf{x} - \mathbf{X}(s))}{\partial n^j}\, d\mathbf{X}(s) \tag{23}$$

and the *interpolation operator*:

$$(S^*_{(j)}\xi)(s) = (-1)^j \int_C \xi(\mathbf{x}) \frac{\partial^j \delta(\mathbf{x} - \mathbf{X}(s))}{\partial n^j}\, d\mathbf{x} \tag{24}$$

for the j^{th} normal derivative, where $\mathbf{X}(s)$ is a parametrization for Γ with s in the parameter interval $\mathcal{I}_\Gamma$. We further introduce the composite operators T_k, T^*_k, and R^*_k by:

$$T_k = \sum_{j=0}^{k} S_{(j)}, \tag{25a}$$

$$T^*_k = \begin{pmatrix} S^*_{(0)} & S^*_{(1)} & \cdots & S^*_{(k)} \end{pmatrix}^{\mathsf{T}}, \tag{25b}$$

$$R^*_k = \begin{pmatrix} S^*_{(1)} & \cdots & S^*_{(k)} \end{pmatrix}^{\mathsf{T}}. \tag{25c}$$

The operator T^*_k provides an interpolation of a function and its first k normal derivatives to the boundary; R^*_k provides an interpolation of the first k normal derivatives to the boundary, but excludes the values; the spread operator T_k represents a set of singular forces (δ-like) and hyper-singular forces (like the first k normal derivatives of the δ-function) on the boundary.

The central challenge of the IBSE method is to compute the smooth extension to an *unknown* solution. We first discuss how to compute an extension to a given function. Let $v \in C^k(\Omega)$ be given. To compute a $C^k(C)$ extension to v, we solve the following high-order PDE in the region E:

$$\mathcal{H}^k \xi = 0 \qquad \text{in } E, \tag{26a}$$

$$\frac{\partial^j \xi}{\partial n^j} = \frac{\partial^j v}{\partial n^j} \qquad \text{on } \Gamma,\ 0 \le j \le k. \tag{26b}$$

Here $\mathcal{H}^k$ is an appropriate differential operator such as the polyharmonic operator Δ^{k+1}. Details regarding the specific choice of this operator are available in [18]. This problem may be solved on the simpler domain C using methodology directly analogous to the direct forcing Immersed Boundary method. The boundary conditions given in Equation (26b) that force ξ to share its first k normal derivatives with u along Γ are enforced by the addition of unknown singular and hyper-singular forces supported on the boundary:

$$\mathcal{H}^k \xi(x) + (T_k F)(x) = 0 \qquad \text{for } x \in C, \tag{27a}$$

$$(S^*_{(j)} \xi)(s) = \frac{\partial^j v}{\partial n^j}(s) \qquad \text{for } s \in \mathcal{I}_\Gamma,\ 0 \le j \le k. \tag{27b}$$

Notice that ξ is not actually an extension to v: that is, $\xi(x) \neq v(x)$ in Ω. We will only be interested in the function ξ in E, and so need not form its literal extension (which is $\chi_\Omega v + \chi_E \xi$).

To solve the Poisson problem given by Equation (21) using the IBSE method, we instead solve the extended problem given in Equation (22). The forcing function $\mathcal{F}_e$ that is specified in E must be chosen so that it forces the extended solution u_e to be $C^k(C)$. Let ξ smoothly extend u, that is, we ask that ξ is globally smooth in C and that it satisfies the constraints

$$R^*_k \xi = R^*_k u \tag{28}$$

at the interface Γ. These constraints require that the first k normal derivatives of ξ agree with the first k normal derivatives of u on the boundary. The forcing function $\mathcal{F}_e$ is then defined as $\mathcal{F}_e = \Delta\xi$. Coupling these equations together, we obtain the IBSE formulation for the Poisson problem given by Equation (21):

$$\Delta u - \chi_E \Delta \xi = \chi_\Omega f \qquad \text{in } C, \tag{29a}$$

$$\mathcal{H}^k + T_k F = 0 \qquad \text{in } C, \tag{29b}$$

$$R^*_k \xi = R^*_k u, \tag{29c}$$

$$S^* u = 0. \tag{29d}$$

In [18], we verify that the IBSE formulation of the Poisson problem given in Equation (29) produces $C^k(C)$ solutions that converge at a rate of $\mathcal{O}(\Delta x^{k+1})$, in the $L^\infty(\Omega)$ norm, for the Poisson problem, as well as the heat equation, Burgers equation, and the Fitzhugh-Nagumo equations. In [19], we derive a generalization of the IBSE formulation described here sufficient for solving the Stokes equations, and verify that it produces velocities $\mathbf{u}$ with global regularity $C^k(C)$ and that all elements of the viscous stress tensor are $C^{k-1}(C)$. The velocity and stress converge to the correct solutions at a rate of $\mathcal{O}(\Delta x^{k+1})$ and $\mathcal{O}(\Delta x^k)$, respectively.

4.2. Coupling of IBSE to the SOB solver

In Section 3, we computed solutions to a standard benchmark problem (the flow around a cylinder in a confined channel) for the Stokes Oldroyd-B model using the Immersed Boundary method, and find several deficiencies with the quality of the solutions. In particular, even for low values of the Weissenberg number ($\mathrm{Wi} = 0.1$), the solutions produce polymeric stresses that are too small, are maximized away from the boundary, and fail to converge when interpolated to the boundary. Although an integral quantity relating to the stress (the drag coefficient, C_d) does converge, it converges slowly, and produces large errors: approximately 33% on a coarse grid ($n_y = 64$) and approximately 6% on a fine grid ($n_y = 256$). The

Immersed Boundary Smooth Extension method introduced in [19] and summarized in Section 4 was designed primarily to deal with these issues. By automatically generating a globally smooth extension of the velocity field $\mathbf{u}$ and pressure field p at every timestep, the boundary can be effectively ignored when solving the Oldroyd-B update without introducing large errors at the boundary.

The IBSE based Stokes Oldroyd-B solver works in almost exactly the same way as the IB based Stokes Oldroyd-B solver described in Section 3. There is only one additional difficulty: we must decide what to do with the extra polymeric stress τ_P in the extension region E. Recall that b is defined to be the positive definite matrix square-root of the conformation tensor σ. Assuming that $\mathbf{u}_t$ and b_t are known at the discrete time t, the Stokes Oldroyd-B system is advanced in time as follows:

1. The evolution equation for b given in Equation (7) is advanced for one time-step to compute b_*, as described in Section 2.1.
2. The conformation tensor σ_* is computed as b_*^2.
3. The Stokes equation:

$$-\Delta \mathbf{u}_{t+\Delta t} + \nabla p_{t+\Delta t} = \xi \nabla \cdot \sigma_* \qquad \text{in } \Omega, \tag{30a}$$

$$\nabla \cdot \mathbf{u}_{t+\Delta t} = 0 \qquad \text{in } \Omega, \tag{30b}$$

   Supplemented with boundary conditions for $\mathbf{u}_{t+\Delta t}$, is solved using the IBSE method to find the solutions $\mathbf{u}_{t+\Delta t}$ and $p_{t+\Delta t}$ that are globally smooth in C and satisfy Equation (30) in Ω (see Section 4 for details).
4. The square root of the conformation tensor, b_*, is *re-extended*; that is, redefined in the extension region E to maintain smoothness in the entire domain, denoted by $b_{t+\Delta t} = \mathcal{R} b_*$. We describe the re-extension process in detail below.

Other than the final re-extension step, this algorithm is identical to the IB-SOB algorithm described in Section 3, with the IBSE method replacing the IB method for solving the Stokes equation given by Equation (30). The re-extension step is necessary due to the fact that the solution of u in E is non-physical, and often displays large derivatives. Naively evolving b in E using this aphysical u leads to instabilities, and thus we choose simply to reset b in E at each timestep. The re-extension operator $\mathcal{R}_{k-1}$, for use with the IBSE-k method, is defined by the following process:

1. Let $\tilde{b}$ be the solution to the equation:

$$\mathcal{H}^{k-1}\tilde{b} + T_{k-1} = 0 \qquad \text{in } C, \tag{31a}$$

$$T^*_{k-1}\tilde{b} = T^*_{k-1} b_*, \tag{31b}$$

   Note that this equation is solved elementwise for each element of the tensor. That is, we define $\tilde{b}^{ij}$ to be a $C^{k-1}(C)$ function that shares its first $k-1$ normal derivatives with b_*^{ij} on the boundary Γ.
2. Define $\mathcal{R} b_*$ by

$$\mathcal{R} b_* = \begin{cases} b_* & \text{in } \Omega, \\ \tilde{b} & \text{in } E. \end{cases} \tag{32}$$

Thus the re-extension operator $\mathcal{R}_{k-1}$ smoothly re-defines $b_{t+\Delta t}$ in the extension region E. This equation is solved using the methodology described in Section 4. Notice the choice to re-extend b to be only C^{k-1}, and not C^k. The IBSE-k method produces C^k solutions of the velocity field $\mathbf{u}$, and thus b will be only C^{k-1} in the vicinity of the boundary.

4.3. Results: Flow past a cylinder at low Wi *using the IB-SOB solver*

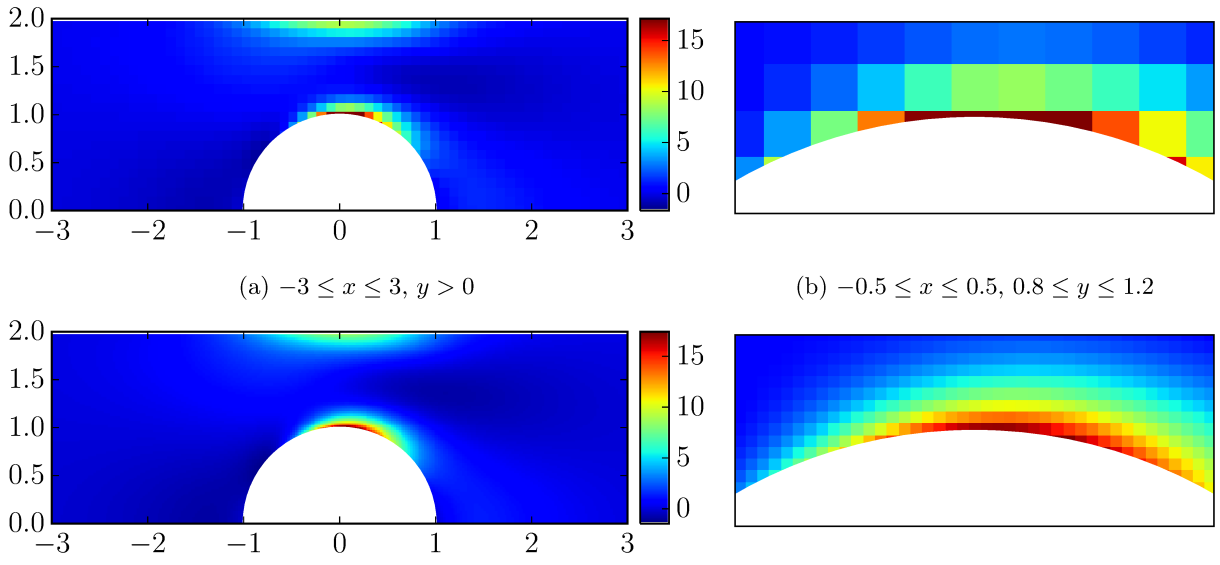
We now return to the confined channel flow around a cylinder problem defined in Section 2.2 and studied using the IB-SOB algorithm in Section 3.1. The setup for the simulations is identical, and we present results for the same coarse ($n_y = 64$) and fine ($n_y = 256$) discretizations used in Section 3.1. Evolution of the

n_y	IB (value)	IBSE-2 (value)	IB (error)	IBSE-2 (error)
64	173.951	130.457	33%	0.07%
256	138.647	130.374	6.4%	0.008%

Table 1: Drag coefficient and relative error in the drag coefficient, for Wi = 0.1, computed relative to the value 130.364 from [16]. Relative errors are computed assuming that the reference value is exact; errors from IBSE are on the scale of the last digit reported.

SOB system is done using the IBSE-SOB solver as described in Section 4.2. In all simulations presented henceforth, the value of k for the IBSE solver is chosen to be 2, providing third-order accuracy for velocities and second-order accuracy for stresses.

The xx-component of the extra polymeric stress, τ_{P}, computed using both the coarse discretization with $n_y = 64$ and the fine discretization with $n_y = 256$, is shown in Figure 4. In the zoom of the stress near the cylinder wall, it is apparent that the stress is maximized on the boundary. The maximum value of $\tau_{\mathrm{P}xx}$ is approximately 18.5 and 18.2 for the coarse and fine discretizations, respectively, substantially larger than those computed using the IB-SOB method, and consistent with the solutions shown in [16]. The drag coefficient may be computed directly using Equation (9), and is found to be $C_d = 130.457$ for $n_y = 64$ and 130.374 for $n_y = 256$, giving an error of 0.07% and 0.008% when compared with the value $C_d = 130.364$ reported by [16]. This is an improvement of three to four orders of magnitude over the errors produced by the IB-SOB method; a comparison is shown in Table 1. For both discretizations, The xx-component of τ_{P}, interpolated to the boundary Γ by applying the interpolation operator S^*, is shown in Figure 7c. This curve converges rapidly, and appears consistent with the with the solution from [16]; a more detailed analysis is given in Section 4.4.



(a) $-3 \le x \le 3$, $y > 0$ (b) $-0.5 \le x \le 0.5$, $0.8 \le y \le 1.2$

(c) $-3 \le x \le 3$, $y > 0$ (d) $-0.5 \le x \le 0.5$, $0.8 \le y \le 1.2$

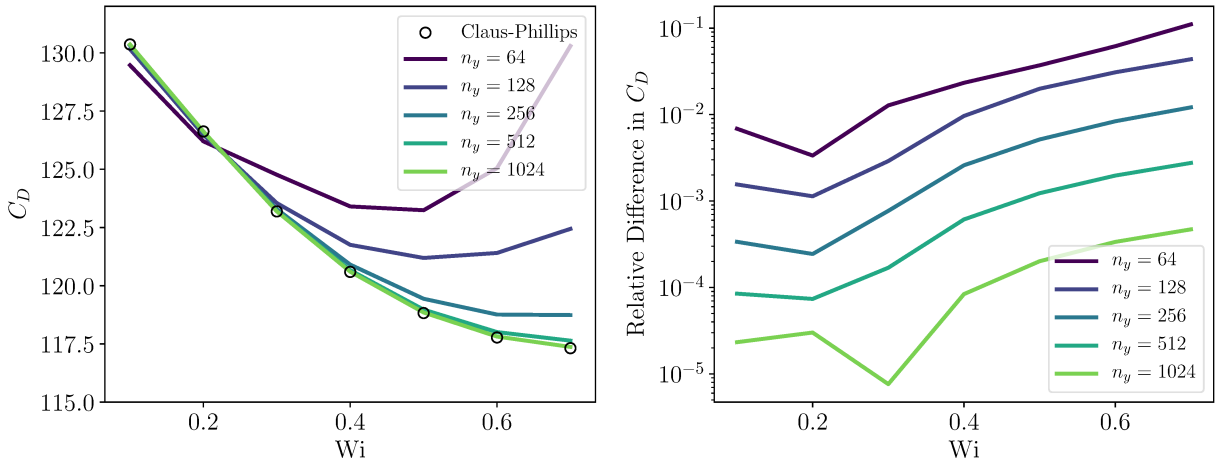
Figure 4: The xx-component of the polymeric stress τ_{P} for the Stokes Oldroyd-B flow around the cylinder problem, computed by the IBSE-SOB method, with $n_y = 64$ (a, b) and $n_y = 256$ (c, d), and Weissenberg number Wi = 0.1. In (a, c), the stress $\tau_{\mathrm{P}xx}$ is shown in the upper half of the channel ($y > 0$), for $-3 \le x \le 3$. In (b, d), a zoom near the cylinder wall is shown, with $-0.5 \le x \le 0.5$ and $0.8 \le y \le 1.2$.

*4.4. Results for moderate Weissenberg number (*Wi ≤ 0.7*)*

As can be seen from examining Figure 4d, the solution to the confined channel flow around a cylinder problem produces steep stress gradients in the vicinity of the cylinder walls, even for low Weissenberg number

(Wi = 0.1). At higher values of the Weissenberg number this problem becomes more severe, with steeper gradients at cylinder walls and the appearance of large stresses in the cylinder wake. To help alleviate the numerical problems that arise due to these steep gradients in the stress we add an artificial viscosity $\zeta\Delta b$ to the time evolution for the polymeric stress, with $\zeta = (4\Delta x)^2$. Because this is a diffusion on b, rather than σ or τ_{P}, we can not identify it as stress diffusion; it is simply an artificial viscosity added to prevent numerical breakdown. This evolution equation is marched in time using the same IMEX-BDF scheme used in [19] for solving the Navier-Stokes equations. Note that this is different from the examples in the preceding section, which where integrated in time using RK4.

In Figure 5, we show the drag coefficient computed via Equation (9), for $0.1 \leq \mathrm{Wi} \leq 0.7$ over a range of discretization sizes. For low Wi, our simulations agree quite well, even for coarse discretizations. The resolution required to obtain accurate results increases with Wi. To better highlight the convergence, we also show the relative error in the drag coefficient C_d (computed against the value provided by [16]). For the finest discretization and Wi = 0.1, the error is 0.009%, while the error at Wi = 0.7 is 0.19%. In Figure 6a, we show the xx-component of the polymeric stress τ_{P} in the region near to the cylinder for Wi = 0.7 and $n_y = 1024$ at $t = 14$. The solution shows the qualitative features that we expect: a steep boundary layer along the cylinder wall and a prominent stress island in the wake behind the stagnation point at the rear of the cylinder.



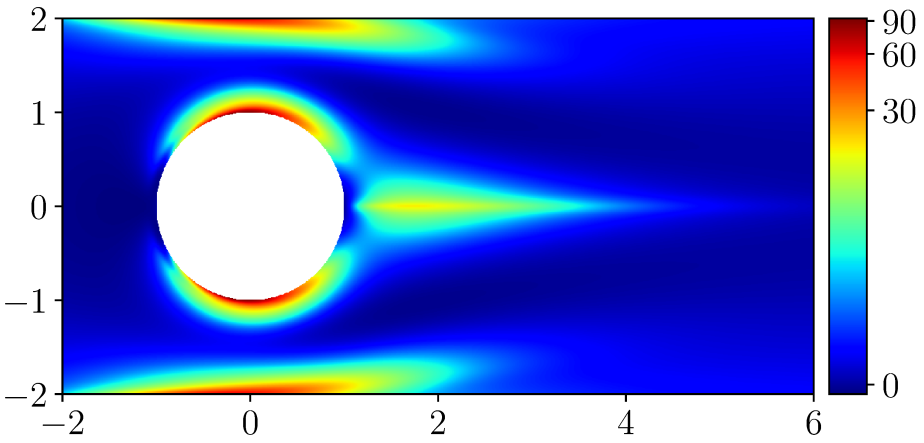
(a) Drag coefficient

(b) Relative error

Figure 5: The drag coefficient C_d and the relative error when compared against the results from [16], computed by the IBSE-2 method with $n_y = 64$, 128, 256, 512, and 1024.

4.5. Effect of the artificial viscosity ζ

Finally, we analyze the effect of the artificial viscosity $\zeta\Delta$ added to the stress evolution equation. For the grid resolution $n_y = 256$, with Wi = 0.1, we run to $t = 20\,\mathrm{Wi} = 2.0$ and analyze the drag coefficient C_d and the xx-component of the polymeric stress integrated to the boundary ($S^*\tau_{\mathrm{P}xx}$) over a wide range of values of ζ: for $\zeta = 1/2^k$ for $0 \leq k \leq 20$. The results are shown in Figure 7. The convergence study for C_d confirms that the stress converges at first-order in ζ. Because we scale ζ as Δx^2 as the grid is refined, this gives $\mathcal{O}(\Delta x^2)$ convergence of the stress. This is the maximum asymptotic accuracy deliverable by the IBSE-2 solver used. The values of ζ used for $n_y = 128,\ 256,\ 512$, and 1024, along with the associated error induced in C_d by the artificial diffusivity, are shown in Figure 7a. The polymeric stress $\tau_{\mathrm{P}xx}$, interpolated to the boundary, computed over a range of ζ, is shown in Figure 7b. At large values of ζ, the stress on the boundary differs markedly from the zero diffusion case, but is nearly indistinguishable for the values used in simulation, especially for finer discretizations. Finally, in Figure 7b, we show $\tau_{\mathrm{P}xx}$, as a function of n, interpolated to the boundary, for both the non-diffusive computations from Section 4.3 (solid lines),



(a) $\tau_{\mathrm{P}xx}$

Figure 6: Figure 6a shows the xx-component of the extra polymeric stress τ_{P} for the Stokes Oldroyd-B flow around the cylinder problem, computed by the IBSE method, with $n_y = 1024$ and Weissenberg number $\mathrm{Wi} = 0.7$. Note that the color scale has been logged, and that the flow domain is longer than is shown.

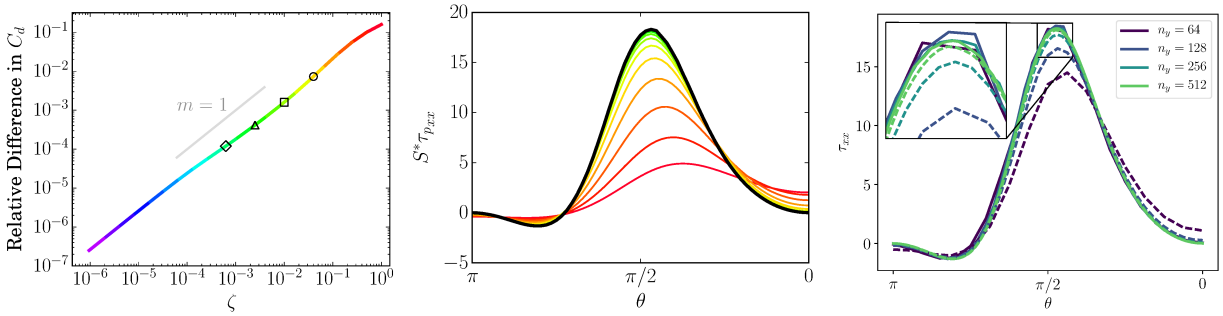
and with artificial diffusion added (dotted lines). The non-diffusive solutions converge rapidly with n; the diffusive solutions converge more slowly, with good agreement for finer discretizations.

5. Discussion

In this paper, we have analyzed the convergence of an uncorrected IB method, as well as a method based on the IBSE method, for solving the Stokes Oldroyd-B equations. The IB based method fails to capture stresses accurately on-boundary and in near-boundary regions, and is slow to converge even for averaged quantities such as the drag coefficient. The IBSE method, by contrast, produces rapid convergence of averaged quantities and convergent solutions in near- and on-boundary regions that are in agreement with benchmark solutions. Critical to this convergence is the fact that the IBSE solver accurately computes gradients of the fluid velocity, which feed directly into the evolution equation for the polymeric stresses.

There are several significant limitations to the approach that we have taken. The first of these is the use of a uniform discretization. It is apparent that the large gradients in fluid velocity near fixed objects generate large stresses, both along those objects and in the wake of hyperbolic points in the flow. The IB method has been implemented using adaptive mesh refinement in the distributed memory-parallel IBAMR (Immersed Boundary Adaptive Mesh Refinement) software [67]. In [19], we have shown how to couple the IBSE methodology to an underlying finite-difference discretization. Discretization of the physical and extension equations need not be done using the same grids or methods, as in this paper: the physical equations may be computed using finite-differences and AMR, while extensions can be computed using Fourier methods on a dense grid that is localized around obstacles in the flow.

The second major limitation is that the method, as currently implemented, requires the dense inversion of a system with size $2(k+1)N$, where k is the regularity requested from the solution and N is the number of nodes in the discretized boundary. In two-dimensions, this allows for the solution of small moving-boundary problems and relatively large stationary-boundary problems. In three-dimensions, the number of points required for discretizing the boundaries is typically large enough that dense inversion is impractical. The development of a an efficient preconditioning strategy for these matrices is crucial to scaling the methodology to large-scale moving boundary problems, in both two and three dimensions.



(a) Convergence of C_d (b) Varying over ζ (c) Varying over n

Figure 7: In Figure 7a, we show the relative error for the drag coefficient C_d, produced by the IBSE-2 method with varying values of the artificial viscosity parameter ζ, as compared to the solution produced by the IBSE-2 method with $\zeta = 0$. The values of ζ used to study higher Wi numbers, along with the corresponding relative difference in the C_d for this problem, are shown as ($\bigcirc$), for $n_y = 128$, ($\square$), for $n_y = 256$, ($\triangle$), for $n_y = 512$, and ($\Diamond$), for $n_y = 1024$. In Figure 7b, we show the xx-component of the extra polymeric stress τ_{P} for the Stokes Oldroyd-B flow around the cylinder problem, interpolated to the boundary using the operator S^*, computed by the IBSE-2 method for a range of values of ζ; the colors of each line correspond to the values of ζ shown in Figure 7a. The solution corresponding to $\zeta = 0$ is shown in black. $S^*\tau_{\mathrm{P}xx}$ is shown along the upper half of the cylinder as a function of the parameter θ, with $\theta = 0$ corresponding to the point at the centerline of the channel downstream of the cylinder, and $\theta = \pi$ corresponding to the point at the centerline of the channel upstream of the cylinder. In Figure 7c, we again show $S^*\tau_{\mathrm{P}xx}$, but now as a function of n. Solid lines correspond to the non-diffusive case, the dotted lines correspond to the solutions computed with diffusion.

Acknowledgements

R.D.G. B.T. were supported in part by NSF Grant No. DMS-1664679.

References

[1] Charles S. Peskin. The immersed boundary method. *Acta Numerica*, 11:479–517, 2002. ISSN 0962-4929. doi: 10.1017/S0962492902000077.

[2] Rajat Mittal and Gianluca Iaccarino. Immersed Boundary Methods. *Annual Review of Fluid Mechanics*, 37(1):239–261, 2005. ISSN 0066-4189. doi: 10.1146/annurev.fluid.37.061903.175743.

[3] Sarah D Olson and Anita T Layton. Simulating biofluid-structure interactions with an immersed boundary framework–a review. *Biological Fluid Dynamics: Modeling, Computations, and Applications*, 628:1, 2014.

[4] Bakytzhan Kallemov, Amneet Bhalla, Boyce Griffith, and Aleksandar Donev. An immersed boundary method for rigid bodies. *Communications in Applied Mathematics and Computational Science*, 11(1):79–141, 2016.

[5] Kunihiko Taira and Tim Colonius. The immersed boundary method: A projection approach. *Journal of Computational Physics*, 225(2):2118–2137, 2007. ISSN 00219991. doi: 10.1016/j.jcp.2007.03.005.

[6] Joseph M Teran and Charles S Peskin. Tether force constraints in Stokes flow by the immersed boundary method on a periodic domain. *SIAM Journal on Scientific Computing*, 31(5):3404–3416, 2009.

[7] Joseph Teran, Lisa Fauci, and Michael Shelley. Viscoelastic fluid response can increase the speed and efficiency of a free swimmer. *Physical review letters*, 104(3):038101, 2010.

[8] Becca Thomases and Robert D Guy. The role of body flexibility in stroke enhancements for finite-length undulatory swimmers in viscoelastic fluids. *Journal of Fluid Mechanics*, 825:109–132, 2017.

[9] Becca Thomases and Robert D Guy. Mechanisms of elastic enhancement and hindrance for finite-length undulatory swimmers in viscoelastic fluids. *Physical review letters*, 113(9):098102, 2014.

[10] Gaojin Li and Arezoo M Ardekani. Collective motion of microorganisms in a viscoelastic fluid. *Physical review letters*, 117(11):118001, 2016.

[11] Chuanbin Li, Boyang Qin, Arvind Gopinath, Paulo E Arratia, Becca Thomases, and Robert D Guy. Flagellar swimming in viscoelastic fluids: role of fluid elastic stress revealed by simulations based on experimental data. *Journal of The Royal Society Interface*, 14(135):20170289, 2017.

[12] Daniel Salazar, Alexandre M Roma, and Hector D Ceniceros. Numerical study of an inextensible, finite swimmer in stokesian viscoelastic flow. *Physics of Fluids (1994-present)*, 28(6):063101, 2016.

[13] Nhan Phan-Thien and Hua-Shu Dou. Viscoelastic flow past a cylinder: drag coefficient. *Computer methods in applied mechanics and engineering*, 180(3-4):243–266, 1999.

[14] Hua-Shu Dou and Nhan Phan-Thien. The flow of an Oldroyd-B fluid past a cylinder in a channel : adaptive viscosity vorticity (DAVSS- 3) formulation. 87(1999), 2006.

[15] M.A. Alves, F.T. Pinho, and P.J. Oliveira. The flow of viscoelastic fluids past a cylinder: finite-volume high-resolution methods. *Journal of Non-Newtonian Fluid Mechanics*, 97(2-3):207–232, feb 2001. ISSN 03770257. doi: 10.1016/S0377-0257(00)00198-1. URL http://linkinghub.elsevier.com/retrieve/pii/S0377025700001981.

[16] S. Claus and T.N. Phillips. Viscoelastic flow around a confined cylinder using spectral/hp element methods. *Journal of Non-Newtonian Fluid Mechanics*, 200:131–146, oct 2013. ISSN 03770257. doi: 10.1016/j.jnnfm.2013.03.004. URL http://linkinghub.elsevier.com/retrieve/pii/S0377025713000785.

[17] Robert G. Owens and Timothy N. Phillips. *Computational Rheology, Vol 2*. World Scientific, 2002.

[18] David B. Stein, Robert D. Guy, and Becca Thomases. Immersed Boundary Smooth Extension: A high-order method for solving PDE on arbitrary smooth domains using Fourier spectral methods. *Journal of Computational Physics*, 304: 252–274, 2015. ISSN 10902716. doi: 10.1016/j.jcp.2015.10.023. URL http://arxiv.org/abs/1506.07561.

[19] David B Stein, Robert D Guy, and Becca Thomases. Immersed boundary smooth extension (ibse): A high-order method for solving incompressible flows in arbitrary smooth domains. *Journal of Computational Physics*, 335:155–178, 2017.

[20] Zhilin Li and Kazufumi Ito. *The immersed interface method: numerical solutions of PDEs involving interfaces and irregular domains*, volume 33. Siam, 2006.

[21] Ronald P Fedkiw, Tariq Aslam, Barry Merriman, and Stanley Osher. A Non-oscillatory Eulerian Approach to Interfaces in Multimaterial Flows (the Ghost Fluid Method). *Journal of Computational Physics*, 152(2):457–492, 1999. ISSN 00219991. doi: 10.1006/jcph.1999.6236. URL http://www.sciencedirect.com/science/article/pii/S0021999199962368.

[22] Philippe Angot, Charles-Henri Bruneau, and Pierre Fabrie. A penalization method to take into account obstacles in incompressible viscous flows. *Numerische Mathematik*, 81(4):497–520, 1999. ISSN 0029-599X. doi: 10.1007/s002110050401.

[23] Ming-Chih Lai and Charles S. Peskin. An Immersed Boundary Method with Formal Second-Order Accuracy and Reduced Numerical Viscosity. *Journal of Computational Physics*, 160(2):705–719, 2000. ISSN 00219991. doi: 10.1006/jcph.2000.6483. URL http://linkinghub.elsevier.com/retrieve/pii/S0021999100964830.

[24] Andreas Mark and Berend G M van Wachem. Derivation and validation of a novel implicit second-order accurate immersed boundary method. *Journal of Computational Physics*, 227(13):6660–6680, 2008. ISSN 00219991. doi: 10.1016/j.jcp.2008.03.031.

[25] Mark N. Linnick and Hermann F. Fasel. A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains. *Journal of Computational Physics*, 204(1):157–192, 2005. ISSN 00219991. doi: 10.1016/j.jcp.2004.09.017.

[26] Jian Kang Liu and Zhou Shun Zheng. Efficient high-order immersed interface methods for heat equations with interfaces. *Applied Mathematics and Mechanics*, 35(51174236):1189–1202, 2014. ISSN 02534827. doi: 10.1007/s10483-014-1851-6.

[27] Sheng Xu and Z. Jane Wang. An immersed interface method for simulating the interaction of a fluid with moving boundaries. *Journal of Computational Physics*, 216(2):454–493, 2006. ISSN 00219991. doi: 10.1016/j.jcp.2005.12.016.

[28] Xiaolin Zhong. A new high-order immersed interface method for solving elliptic equations with imbedded interface of discontinuity. *Journal of Computational Physics*, 225(1):1066–1099, 2007. ISSN 00219991. doi: 10.1016/j.jcp.2007.01.017.

[29] S Yu, Y Zhou, and G Wei. Matched interface and boundary (MIB) method for elliptic problems with sharp-edged interfaces. *Journal of Computational Physics*, 224(2):729–756, 2007. ISSN 00219991. doi: 10.1016/j.jcp.2006.10.030. URL http://dx.doi.org/10.1016/j.jcp.2006.10.030.

[30] Y. C. Zhou, Shan Zhao, Michael Feig, and G. W. Wei. High order matched interface and boundary method for elliptic equations with discontinuous coefficients and singular sources. *Journal of Computational Physics*, 213(1):1–30, 2006. ISSN 00219991. doi: 10.1016/j.jcp.2005.07.022.

[31] Frédéric Gibou and Ronald Fedkiw. A fourth order accurate discretization for the Laplace and heat equations on arbitrary domains, with applications to the Stefan problem. *Journal of Computational Physics*, 202(2):577–601, 2005. ISSN 00219991. doi: 10.1016/j.jcp.2004.07.018.

[32] John P. Boyd. Fourier embedded domain methods: extending a function defined on an irregular region to a rectangle so that the extension is spatially periodic and C^{∞}. *Applied Mathematics and Computation*, 161(2):591–597, 2005. ISSN 00963003.

[33] Alfonso Bueno-Orovio. Fourier embedded domain methods: Periodic and c^{∞} extension of a function defined on an irregular region to a rectangle via convolution with gaussian kernels. *Applied Mathematics and Computation*, 183(2):813–818, 2006. ISSN 00963003. doi: 10.1016/j.amc.2006.06.029.

[34] S. H. Lui. Spectral domain embedding for elliptic PDEs in complex domains. *Journal of Computational and Applied Mathematics*, 225(2):541–557, 2009. ISSN 03770427. doi: 10.1016/j.cam.2008.08.034. URL http://dx.doi.org/10.1016/j.cam.2008.08.034.

[35] Feriedoun Sabetghadam, Shervin Sharafatmandjoor, and Farhang Norouzi. Fourier spectral embedded boundary solution of the Poisson's and Laplace equations with Dirichlet boundary conditions. *Journal of Computational Physics*, 228(1): 55–74, 2009. ISSN 00219991. doi: 10.1016/j.jcp.2008.08.018. URL http://dx.doi.org/10.1016/j.jcp.2008.08.018.

[36] Nathan Albin and Oscar P. Bruno. A spectral FC Solver for the compressible Navier-Stokes equations in general domains I: Explicit time-stepping. *Journal of Computational Physics*, 230(16):6248–6270, jul 2011. ISSN 00219991. doi: 10.1016/j.jcp.2011.04.023. URL http://linkinghub.elsevier.com/retrieve/pii/S0021999111002695.

[37] Mark Lyon and Oscar P. Bruno. High-order unconditionally stable FC-AD solvers for general smooth domains I. Basic Elements. *Journal of Computational Physics*, 229(9):3358–3381, 2010. ISSN 00219991. doi: 10.1016/j.jcp.2010.01.006. URL http://dx.doi.org/10.1016/j.jcp.2009.11.020.

[38] Mark Lyon and Oscar P. Bruno. High-order unconditionally stable FC-AD solvers for general smooth domains II. Elliptic, parabolic and hyperbolic PDEs; theoretical considerations. *Journal of Computational Physics*, 229(9):3358–3381, 2010. ISSN 00219991. doi: 10.1016/j.jcp.2010.01.006. URL http://dx.doi.org/10.1016/j.jcp.2010.01.006.

[39] David Shirokoff and Jean-Christophe Nave. A Sharp-Interface Active Penalty Method for the Incompressible Navier-Stokes

Equations. *Journal of Scientific Computing*, 62(1):53–77, 2015. URL http://arxiv.org/abs/1303.5681.
[40] Randall J. LeVeque and Zhilin Li. The Immersed Interface Method for Elliptic Equations with Discontinuous Coefficients and Singular Sources. *SIAM Journal on Numerical Analysis*, 32(5):1704–1704, 1995. ISSN 0036-1429. doi: 10.1137/0732076.
[41] S De, S Das, JAM Kuipers, EAJF Peters, and JT Padding. A coupled finite volume immersed boundary method for simulating 3d viscoelastic flows in complex geometries. *Journal of Non-Newtonian Fluid Mechanics*, 232:67–76, 2016.
[42] Amir Saadat, Christopher J Guido, Gianluca Iaccarino, and Eric SG Shaqfeh. Immersed-finite-element method for deformable particle suspensions in viscous and viscoelastic media. *Physical Review E*, 98(6):063316, 2018.
[43] C Fernandes, SA Faroughi, OS Carneiro, J Miguel Nóbrega, and GH McKinley. Fully resolved simulations of particle-laden viscoelastic fluids using an immersed boundary method. *Journal of Non-Newtonian Fluid Mechanics*, 2019.
[44] H Giesekus. A simple constitutive equation for polymer fluids based on the concept of deformation-dependent tensorial mobility. *Journal of Non-Newtonian Fluid Mechanics*, 11(1-2):69–109, 1982.
[45] Thomas Y Hou and Ruo Li. Computing nearly singular solutions using pseudo-spectral methods. *Journal of Computational Physics*, 226(1):379–397, 2007.
[46] John Charles Butcher. *The numerical analysis of ordinary differential equations: Runge-Kutta and general linear methods*. Wiley-Interscience, 1987.
[47] Fang-Hua Lin, Chun Liu, and Ping Zhang. On hydrodynamics of viscoelastic fluids. *Communications on Pure and Applied Mathematics*, 58(11):1437–1471, 2005.
[48] Raanan Fattal and Raz Kupferman. Time-dependent simulation of viscoelastic flows at high Weissenberg number using the log-conformation representation. *Journal of Non-Newtonian Fluid Mechanics*, 126(1):23–37, 2005. ISSN 03770257. doi: 10.1016/j.jnnfm.2004.12.003.
[49] Nusret Balci, Becca Thomases, Michael Renardy, and Charles R. Doering. Symmetric factorization of the conformation tensor in viscoelastic fluid models. (3), jun 2010. URL http://arxiv.org/abs/1006.3488v1.
[50] M.D. Chilcott and J.M. Rallison. Creeping flow of dilute polymer solutions past cylinders and spheres. *Journal of Non-Newtonian Fluid Mechanics*, 29:381–432, 1988. ISSN 03770257. doi: 10.1016/0377-0257(88)85062-6.
[51] Alice W Liu, David E Bornside, Robert C Armstrong, and Robert A Brown. Viscoelastic flow of polymer solutions around a periodic, linear array of cylinders: comparisons of predictions for microstructure and flow fields. *Journal of Non-Newtonian Fluid Mechanics*, 77(3):153–190, 1998.
[52] J Sun, MD Smith, RC Armstrong, and RA Brown. Finite element method for viscoelastic flows based on the discrete adaptive viscoelastic stress splitting and the discontinuous galerkin method: Davss-g/dg. *Journal of Non-Newtonian Fluid Mechanics*, 86(3):281–307, 1999.
[53] Yurun Fan, RI Tanner, and N Phan-Thien. Galerkin/least-square finite-element methods for steady viscoelastic flows. *Journal of Non-Newtonian Fluid Mechanics*, 84(2):233–256, 1999.
[54] AE Caola, YL Joo, RC Armstrong, and RA Brown. Highly parallel time integration of viscoelastic flows. *Journal of non-newtonian fluid mechanics*, 100(1-3):191–216, 2001.
[55] MA Alves, FT Pinho, and PJ Oliveira. The flow of viscoelastic fluids past a cylinder: finite-volume high-resolution methods. *Journal of Non-Newtonian Fluid Mechanics*, 97(2-3):207–232, 2001.
[56] Robert G Owens, Cédric Chauvière, and Timothy N Philips. A locally-upwinded spectral technique (lust) for viscoelastic flows. *Journal of non-newtonian fluid mechanics*, 108(1-3):49–71, 2002.
[57] X Ma, V Symeonidis, and GE Karniadakis. A spectral vanishing viscosity method for stabilizing viscoelastic flows. *Journal of non-newtonian fluid mechanics*, 115(2-3):125–155, 2003.
[58] Ju Min Kim, Chongyoup Kim, Kyung Hyun Ahn, and Seung Jong Lee. An efficient iterative solver and high-resolution computations of the oldroyd-b fluid flow past a confined cylinder. *Journal of non-newtonian fluid mechanics*, 123(2-3):161–173, 2004.
[59] Martien A Hulsen, Raanan Fattal, and Raz Kupferman. Flow of viscoelastic fluids past a cylinder at high weissenberg number: stabilized simulations using matrix logarithms. *Journal of Non-Newtonian Fluid Mechanics*, 127(1):27–39, 2005.
[60] Yurun Fan, Huayong Yang, and Roger I Tanner. Stress boundary layers in the viscoelastic flow past a cylinder in a channel: limiting solutions. *Acta Mechanica Sinica*, 21(4):311–321, 2005.
[61] Robert G Owens and Timothy N Phillips. *Computational rheology*. World Scientific, 2002.
[62] S Claus and TN Phillips. Viscoelastic flow around a confined cylinder using spectral/hp element methods. *Journal of Non-Newtonian Fluid Mechanics*, 200:131–146, 2013.
[63] Raz Kupferman. On the linear stability of plane Couette flow for an Oldroyd-B fluid and its numerical approximation. *Journal of Non-Newtonian Fluid Mechanics*, 127(2-3):169–190, 2005. ISSN 03770257. doi: 10.1016/j.jnnfm.2005.03.002. URL http://linkinghub.elsevier.com/retrieve/pii/S0377025705000674.
[64] S. Berti and G. Boffetta. Elastic waves and transition to elastic turbulence in a two-dimensional viscoelastic Kolmogorov flow. *Physical Review E*, 82(3):036314, sep 2010. ISSN 1539-3755. doi: 10.1103/PhysRevE.82.036314. URL http://link.aps.org/doi/10.1103/PhysRevE.82.036314.
[65] SAMRAI: Structured adaptive mesh refinement application infrastructure. URL http://computation.llnl.gov/project/SAMRAI/.
[66] Richard D Hornung and Scott R Kohn. Managing application complexity in the samrai object-oriented framework. *Concurrency and Computation: Practice and Experience*, 14(5):347–368, 2002.
[67] Boyce E Griffith, Richard D Hornung, David M McQueen, and Charles S Peskin. An adaptive, formally second order accurate version of the immersed boundary method. *Journal of Computational Physics*, 223(1):10–49, 2007.