Chapter 10
Computational Challenges for Simulating Strongly Elastic Flows in Biology

Robert D. Guy and Becca Thomases

Abstract Understanding the behavior of complex fluids in biology presents mathematical, modeling, and computational challenges not encountered in classical fluid mechanics, particularly in the case of fluids with large elastic forces that interact with immersed elastic structures. We discuss some of the characteristics of strongly elastic flows and introduce different models and methods designed for these types of flows. We describe contributions from analysis that motivate numerical methods and illustrate their performance on different models in a simple test problem. Biological problems often involve the coupled dynamics of active elastic structures and the surrounding fluid. The immersed boundary method has been used extensively for such problems involving Newtonian fluids, and the methodology extends naturally to complex fluids in conjunction with the algorithms described earlier in this chapter. We focus on implicit-time methods because the large elastic stresses in complex fluids necessitate high spatial resolution and long time simulations. As an example to highlight some of the challenges of strongly elastic flows, we use the immersed boundary method to simulate an undulatory swimmer in a viscoelastic fluid using a data-based model for the prescribed shape.

There are many different kinds of complex fluids in biology, and they frequently contain dynamic active or passive structures in the fluid. Numerical simulations of these complex flows can be a powerful tool in understanding these biological systems. Existing techniques in computational fluid dynamics are often sufficient for problems with weak flows and low elasticity. However, when elastic forces become large due to, for example, long relaxation times, extra forces from internal structures, or interactions with complex boundaries, more care must be taken to properly simulate these flows. This chapter is devoted to the challenges that

R.D. Guy (✉) • B. Thomases
Department of Mathematics, University of California, Davis,
1 Shields Ave., Davis, CA 95616, USA
e-mail: guy@math.ucdavis.edu; thomases@math.ucdavis.edu

S.E. Spagnolie (ed.), *Complex Fluids in Biological Systems*, Biological and Medical Physics, Biomedical Engineering,
DOI 10.1007/978-1-4939-2065-5_10

arise when internal elastic forces are modeled in complex fluids with an eye towards recognizing, understanding, and properly treating the features of strongly elastic flows.

We will focus on the Oldroyd-B model as one of the simplest closed continuum models of viscoelastic fluids. In the original derivation [1], Oldroyd set out requirements for constitutive equations so that the material properties would be frame invariant in a coordinate system which convected with the material. This procedure leads to the upper-convected time derivative (also called the Oldroyd derivative, see Eq. (10.5)) which gives the rate of change of a tensor property of a small volume of fluid written in a coordinate system rotating and stretching with the fluid. The most widely used Oldroyd model is the Oldroyd-B model, in part because this model can also be derived from a theory of dilute polymer solutions [2]. The Oldroyd-B model is given below for $\mathbf{u}$ the velocity of the fluid, p the pressure, and $\boldsymbol{\tau}$ the deviatoric stress tensor; see also Chap. 1. From balance of momentum and mass conservation for an incompressible fluid we have

$$\rho \frac{D\mathbf{u}}{Dt} = -\nabla p + \nabla \cdot \boldsymbol{\tau} \tag{10.1}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{10.2}$$

with

$$\boldsymbol{\tau} = \eta_s \dot{\boldsymbol{\gamma}} + \boldsymbol{\tau}_p. \tag{10.3}$$

Here $\eta_s \dot{\boldsymbol{\gamma}}$ is the viscous stress from the Newtonian solvent, with viscosity η_s and rate-of-strain tensor $\dot{\boldsymbol{\gamma}} = \nabla \mathbf{u} + (\nabla \mathbf{u})^T$, and $\boldsymbol{\tau}_p$ is the polymeric stress contribution. In the Oldroyd-B model the polymer stress evolves by

$$\boldsymbol{\tau}_p + \lambda \overset{\triangledown}{\boldsymbol{\tau}}_p = \eta_p \dot{\boldsymbol{\gamma}}, \tag{10.4}$$

with polymer viscosity η_p and relaxation time λ. The upper-convected derivative is defined as

$$\overset{\triangledown}{\boldsymbol{\tau}}_p \equiv (D/Dt)\boldsymbol{\tau}_p - (\nabla \mathbf{u})^T \cdot \boldsymbol{\tau}_p - \boldsymbol{\tau}_p \cdot \nabla \mathbf{u}. \tag{10.5}$$

The relaxation time characterizes the time it takes for a material to adjust to applied stresses or deformations. For the majority of this chapter we will focus on the low Reynolds number (or creeping flow) regime where inertial forces are small compared with viscous forces.

There are two important dimensionless parameters related to the relaxation time of a fluid used in rheology. The Weissenberg number (Wi) is the ratio of the relaxation time of the fluid and a specific process time. For example, in steady shear, the Weissenberg number is defined as the shear rate ($\dot{\gamma} = \sqrt{\dot{\boldsymbol{\gamma}} : \dot{\boldsymbol{\gamma}}/2}$, see Chap. 1) times the relaxation time $\mathrm{Wi} = \dot{\gamma}\lambda$. The Deborah number (De) is used to characterize

flows under specific flow conditions and is defined as the ratio of the relaxation time to the characteristic time scale for fluid deformations, such as an imposed oscillation period, $\mathrm{De} = \frac{\lambda}{T_f}$.

Weakly elastic flows (De or $\mathrm{Wi} \ll 1$) can be handled using standard techniques from computational fluid dynamics, and special treatment of the stress tensor is not necessary. By contrast, strongly elastic flows (De or $\mathrm{Wi} \gg 1$) create regions of high stress and fine features that require high resolution for accurate flow solutions. Naive implementations of standard CFD techniques may fail.

In Sect. 1 we identify and demonstrate some of the characteristics of strongly elastic flows. These include large stresses and large gradients which require fine meshes for accurate representation. We introduce a host of models and methods designed to overcome these challenges and use a simple test problem to demonstrate some of these techniques and models. In Sect. 2 we briefly present the ideas of the immersed boundary method, a popular technique for simulating problems in biology which typically involve fluid-structure interactions with large deformations and complex flow patterns. One advantage of the immersed boundary method is that it can be easily coupled with a preexisting fluid solver; however, long time simulations and fine meshes necessitate the use of an implicit time stepping method. In Sect. 3 we use the immersed boundary method to simulate an undulatory swimmer with a data-based model for the target shape. This problem highlights some of the additional challenges associated with strongly elastic flows in biology, due, in part, to the large forces which arise in the coupling of fluid to dynamic structures.

1 Strongly Elastic Flows

Simulating strongly elastic flows is difficult and requires care in the choice of model and numerical method to ensure that the chosen technique treats the elastic stresses and corresponding time scale properly. In this chapter we emphasize the importance of recognizing these difficulties and understanding their origin, and we provide some approaches for fixing them. As an example of one difficulty with the Oldroyd-B model, we present a sample simulation of an extensional point flow in Sect. 1.6 which demonstrates fine scales: near-singularities in the stress field and near-jumps in the vorticity. In these simulations a body force drives a flow in which $\mathbf{u} \sim \dot{\varepsilon}(x, -y)$ near the origin, where $\dot{\varepsilon}$ is the strain rate. To demonstrate the flow behavior we examine the vorticity and the trace of the polymer stress tensor, which is proportional to the elastic energy. Figure 10.1a shows a contour plot of the trace of the conformation tensor, $\mathrm{Tr}(\mathbf{C})$, which is related to the polymer stress tensor for the Oldroyd-B model by $\boldsymbol{\tau}_p = \mathrm{Wi}^{-1}(\mathbf{C} - \mathbf{I})$. The vorticity of the flow ($\nabla \times \mathbf{u}$) is plotted in Fig. 10.1b. These plots show the near-steady behavior for a strongly elastic flow where large stress and stress gradients have developed at the extensional stagnation point at the origin. The fine structures can be seen in slices of $\mathrm{Tr}(\mathbf{C})$ and vorticity, which are shown in Figs. 10.1c and d. The details of the simulation are given in Sect. 1.6.

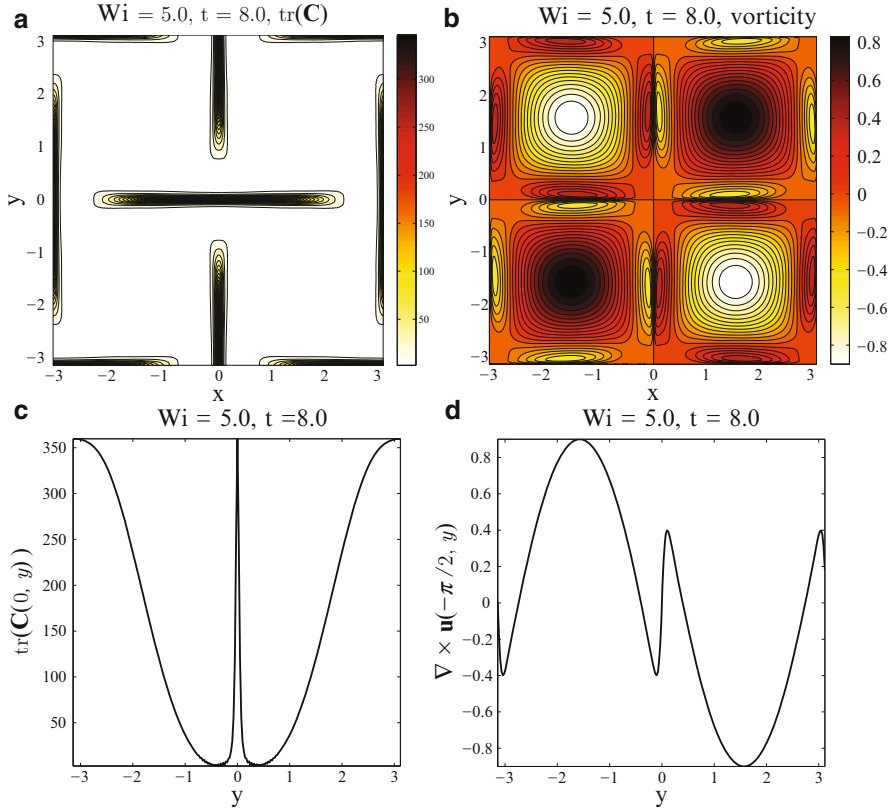

Fig. 10.1 Contour plots of (**a**) $\mathrm{Tr}(\mathbf{C})$ and (**b**) vorticity for $\mathrm{Wi} = 5.0$ at $t = 8$. (**c**) Trace of conformation tensor along line $(0,y)$: $\mathrm{Tr}(\mathbf{C}(0,y))$. (**d**) Vorticity along line $(-\pi/2,y)$: $\nabla \times \mathbf{u}(-\pi/2,y)$

1.1 Historical Perspective

Computational simulations of strongly elastic flows have historically suffered from difficulties not seen in comparable Newtonian flows. These difficulties frequently manifest in numerical methods as a breakdown beyond a critical Deborah (or Weissenberg) number. These computational challenges have been observed since the earliest numerical approximations of complex fluids were attempted in the late 1970s. Standard finite difference and Galerkin finite element approaches that were successful for Newtonian flows were converging only for $\mathscr{O}(1)$ Deborah (or Weissenberg) numbers [3–5]. There are modeling and analytical questions underlying the somewhat mysterious "high-Weissenberg number problem," and so the appropriate choice of numerical method is a delicate and extremely important question which has received much attention over the years. Significant progress has been made in the subsequent decades, but many questions related to the

high-Weissenberg number problem are still subjects of active research. We discuss some of the open analytical questions related to these problems in Sect. 1.2 and some current numerical approaches which address these difficulties in Sect. 1.4. We focus on the Oldroyd-B model in Sect. 1.3 and models derived from different molecular assumptions in Sect. 1.5.

Several benchmark problems in engineering have been developed to test proposed constitutive laws and numerical methods. These problems include flow around a cylinder or sphere [6–12] and planar contractions [13–19]. A simplified explanation for why flow around obstacles or near boundaries can cause difficulties in viscoelastic flows which are not seen in corresponding Newtonian flows is that when the velocity of the flow is near zero (due to a no-slip boundary condition or a stagnation point in the flow) the internal elastic structures in the fluid have a long time to get stretched and induce areas of high stress. Large stress gradients then induce more stress on the flow and this leads to sharp boundary layers which require high spatial resolution. Long time simulations or time-dependent simulations for finding steady-state solutions are particularly difficult.

One must be aware of other challenges which arise when modeling viscoelastic fluids such as possible change of type or loss of evolution in the flow. These problems do not arise with the Oldroyd-B and the other models discussed here. We leave further discussions of those problems and more detailed reviews of numerical issues to the following books and review articles, and the citations contained within, [5, 20–26].

1.2 Advances from Analysis

To begin to understand the complicated high-Weissenberg problem we note that fundamental analytical questions about the Oldroyd-B model (Eq. (10.1)–(10.4)) are still open. For example, it is unknown whether global solutions to the Oldroyd-B equations exist. Even for the Stokesian limit in two space dimensions, the question is challenging due to the lack of scale-dependent dissipation in the polymer stress advection equation[1]. These basic questions are important because if a model is well posed, i.e., unique solutions exist and are sufficiently smooth on some time interval, then appropriate numerical approximations of these solutions are reliable. If not, then even convergence of the method cannot guarantee that the correct solution has been chosen.

Though having a well-posedness theory is useful for a numerical study of a particular model, in many cases, notably the 3D Euler or Navier-Stokes equations,

[1]When derived from the kinetic theory of dumbbells [2, 27] there is polymer stress diffusion; however the stress diffusion coefficient is proportional to the square of the ratio of the bead diameter (or polymer radius of gyration) to the flow length scale, and even in the context of microfluidics, it is minute, $\mathcal{O}(10^{-9})$, and is typically ignored.

this is not available. However, there are often mathematical theorems which can be helpful for guiding numerics. One particular example is a well-known condition for the 3D Euler equations, referred to as the Beale-Kato-Majda criteria [28], which states that the maximum norm of the vorticity controls the breakdown of smooth solutions. More specifically this criteria states that the breakup of solutions in any norm will imply the divergence of the supremum norm of the vorticity. This makes investigating the loss of existence of solutions more tractable as you only have to keep track of one quantity. A similar result can be derived for the Oldroyd-B equations and the quantity to track is the supremum norm of the polymer stress tensor [29, 30].

Along with this nonexistence criterion, some progress in well posedness has been made. For example, if the initial data are sufficiently small, solutions to the Oldroyd-B model are globally well posed [29, 31, 32], but this prescribes an unrealistic constraint for many problems of interest. Additionally, these analytical proofs require either an unbounded or periodic domain, while the question of how to treat problems in complicated geometries remains an issue of current research that is of significant importance in many applications.

As mentioned above, one of the main difficulties in obtaining analytical proofs of global existence for the Oldroyd-B model is the lack of scale-dependent dissipation in the stress advection equation, Eq. (10.4). A simple regularization is to include a dissipative term in such as $\alpha\nabla^2\boldsymbol{\tau}_p$. With this type of polymer stress diffusion, some analytical results are available [27], and, in particular, the Oldroyd-B equations in two space dimensions are globally well posed for all initial data [33, 34]. The addition of polymer stress diffusion is not without physical justification as stress diffusion does arise in the physical model [2, 27, 35], but the diffusion is at such small scales that it is typically ignored. However, artificially large polymer stress diffusion can be introduced as a regularization parameter in numerical simulations [36, 37]. The effect of artificially large stress diffusion was studied in [36] where the authors concluded that the stress diffusion had a stabilizing effect, in particular for large Reynolds number calculations. Adding polymer stress diffusion is particularly common in simulations of turbulent drag reduction [38, 39]. Although the polymer stress diffusion may be artificially large in the context of the molecular derivation, if the length scale of the artificial polymer stress diffusion is on the scale of the grid, then the errors due to the regularization are on the order of the spatial discretization errors.

1.3 High-Weissenberg Number Problem in the Oldroyd-B Model

If the simulations presented in Fig. 10.1 are continued (without decreasing the grid size), numerical oscillations will cause the solutions to break down at some point in time. A "local" analytical solution given in [40] indicates that the polymer

stress is likely approaching a diverging solution exponentially in time. Hence the breakdown of the numerical solution appears to be due to the fact that at a fixed grid size it becomes impossible to resolve the steeper and steeper gradients of an underlying diverging solution. These oscillations can already be seen if one looks closely in Fig. 10.1c at the polymer stress near $y = 0$. There are many possible reasons for the breakdown of the numerical simulation; in particular it may be the case that the solution itself does not exist for all time, and this is simply not known. However, even if the solution remains smooth in time, other problems including loss of positive-definiteness of the stress tensor, problems with the model (such as infinite extension of polymer coils), or large stress gradients which create even larger forces on the fluid which must be resolved can lead to the breakdown of a numerical method. Problems with the Oldroyd-B model are typically blamed on the linear elastic nature of the model, but as we will see in Sect. 1.6 this is probably not the heart of the problem.

The Oldroyd-B model is attractive because it is the simplest closed continuum model which can be derived from molecular assumptions. One derivation of the Oldroyd-B model comes from representing immersed polymer coils in a Newtonian solvent as two beads connected by a linear spring with a Hookean spring force [2,41]. Additional forces on the beads include drag from the Newtonian solvent and randomly fluctuating Brownian forces. Using statistical mechanics, one can derive an expression for the stress tensor which evolves according Eq. (10.4).

This model predicts that in steady extensional flows, such as uniaxial extension, the extensional viscosity, defined as the ratio of extensional stress to extensional strain rate, will become infinite at finite strain rate. This happens when the frictional drag force that stretches the dumbbell overcomes the spring force. When the strain rate is "small" relative to the relaxation time the spring force dominates and the dumbbell remains coiled. As the strain rate is increased the molecules undergo a "coil-stretch" transition and the steady-state extensional viscosity goes to infinity [41]. This is related to the fact that the linear Hooke's law puts no limit on the length of a dumbbell and has been seen as an underlying cause of the "high Weissenberg number problem." Rallison and Hinch questioned the "physics" of the constitutive model [42], and later it was noted that even below the coil-stretch transition the "smoothness of stresses should be expected to deteriorate with increasing Weissenberg number" [43].

Numerical simulations in [40] found solutions that exhibit the Weissenberg number-dependent smoothness described above. When an extensional flow is posited, namely $\mathbf{u} = \dot{\varepsilon}(x, -y)$, the equation for the polymer stress tensor (10.4) decouples and can be solved exactly via the method of characteristics. In two space dimensions the solution for one component of the stress tensor can be written as

$$S_{11}(x,y,t) = \frac{1}{1-2\dot{\varepsilon}\mathrm{Wi}} + e^{(2\dot{\varepsilon}\mathrm{Wi}-1)t} F(xe^{-\dot{\varepsilon}\mathrm{Wi}t}, ye^{\dot{\varepsilon}\mathrm{Wi}t}), \tag{10.6}$$

where the unknown function F must be determined with proper initial and boundary conditions. This solution indicates that for $\dot{\varepsilon}\mathrm{Wi} < 1/2$ the stress should be bounded

but for $\dot{\epsilon}\mathrm{Wi} > 1/2$ the stress is diverging exponentially in time. Furthermore, the solutions have a collapsing inner length scale in y which also depends on $\dot{\epsilon}\mathrm{Wi}$.

One time-independent solution which was found to be in close agreement with numerical simulations and which demonstrates decreasing regularity in the Weissenberg number is

$$S_{11}^{\infty} = |y|^{(1-2\dot{\epsilon}\mathrm{Wi})/\dot{\epsilon}\mathrm{Wi}}. \tag{10.7}$$

It is important to be aware of exponential-in-time stress near-singularities, which are not removed by simply modifying the constitutive model, as we will show in Sect. 1.6. Next we look at a few ways to treat these near-divergent solutions.

1.4 Numerical Approaches

Defeating the "high-Weissenberg number problem" has been the aim of many numerical methods developed over the past several decades. Techniques have been developed to address issues of stability and convergence, for example, the use of upwinding [44,45] has been successfully applied to viscoelastic flow problems using finite element methods [46]. Other variations of finite element methods applied to viscoelastic fluid simulations include using discontinuous Galerkin techniques [47–49] and splitting techniques, such as EVSS (elastic viscous stress splitting) [19,23,50–52], among others [53–55]. Another way to provide local diffusion is by applying ENO schemes (essentially non-oscillatory shock capturing) [56] which use upwinding with a high-order correction [57]. To resolve fine structures in problems like the flow around a cylinder, techniques to improve accuracy include using hp-spectral elements [12] and highly accurate finite volume methods [9]. For a more complete review of the literature we refer the interested reader to a computational rheology book [5] and several review articles that deal with specific aspects of the numerical simulations of viscoelastic fluid flow [22–26]. In what follows we describe in more detail a few ways to handle the near-singularities that arise in the Oldroyd-B model discussed in Sect. 1.3.

1.4.1 Log-Conformation Method

The *log-conformation method* [58,59] is a numerical approach specifically designed to address exponential singularities in the polymer stress tensor. The method was designed for a large class of differential constitutive models (including Oldroyd-B) in which an equation is derived for the matrix logarithm of the conformation tensor or configuration tensor, $\mathbf{C}(\mathbf{x},t)$. $\mathbf{C}$ is the conformational average of the dumbbells,

$$\mathbf{C} = \int \mathbf{RR}\,\Psi dR, \tag{10.8}$$

where $\mathbf{R}$ is the end-to-end vector of the dumbbell and $\Psi(\mathbf{x},\mathbf{R},t)$ is the probability that a dumbbell at position $\mathbf{x}$ in the flow has orientation and extension $\mathbf{R}$ at time t (see Chaps. 1 and 9). For the Oldroyd-B model, the conformation tensor is related to the polymer stress tensor by

$$\boldsymbol{\tau}_p = \mathrm{Wi}^{-1}(\mathbf{C}-\mathbf{I}), \tag{10.9}$$

and is advected by

$$(\mathbf{C}-\mathbf{I}) + \mathrm{Wi}\overset{\triangledown}{\mathbf{C}} = 0. \tag{10.10}$$

From the molecular derivation, the conformation tensor should be symmetric positive definite, and it will remain so according to Eq. (10.10) if it is initially. Loss of positivity of $\mathbf{C}$ is one source of numerical errors. The log-conformation method maintains positivity by definition which can be a source of increased stability.

The log-conformation method replaces Eq. (10.10) with an equation for the matrix logarithm of $\mathbf{C}$:

$$\mathbf{A}(\mathbf{x},t) = \log \mathbf{C}(\mathbf{x},t).$$

This is possible because a symmetric positive definite matrix, $\mathbf{S}$, can always be diagonalized, $\mathbf{S} = \mathbf{R}\boldsymbol{\Lambda}\mathbf{R}^T$ and hence $\log\mathbf{S} = \mathbf{R}(\log\boldsymbol{\Lambda})\mathbf{R}^T$. The method relies on the fact that if $\mathbf{u}$ is a divergence-free velocity field and $\mathbf{C}$ is a symmetric positive definite tensor, then there is a decomposition

$$\nabla\mathbf{u} = \boldsymbol{\Omega} + \mathbf{B} + \mathbf{N}\cdot\mathbf{C}^{-1}, \tag{10.11}$$

where $\boldsymbol{\Omega}$ and $\mathbf{N}$ are anti-symmetric and $\mathbf{B}$ is symmetric, traceless, and commutes with $\mathbf{C}$. With this decomposition, the evolution of $\mathbf{A}$ is

$$\frac{\partial\mathbf{A}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{A} - (\boldsymbol{\Omega}\cdot\mathbf{A} - \mathbf{A}\cdot\boldsymbol{\Omega}) - 2\mathbf{B} = \mathrm{Wi}^{-1}e^{-\mathbf{A}}(\mathbf{I}-e^{\mathbf{A}}). \tag{10.12}$$

Under this transformation, the extensional components of the deformation act additively, rather than multiplicatively. Higher Wi values can be achieved than in similar studies without the matrix logarithm, and the log-conformation method has been particularly successful in some standard benchmark problems [10,60–62]. This method has been implemented in many different numerical frameworks including finite difference, finite volume, and finite elements. There is a nontrivial cost associated with implementation of this method, both in obtaining the decomposition in Eq. (10.11) and in computing the matrix exponential to obtain $\mathbf{C}$ from $\mathbf{A}$.

1.4.2 Square-Root Method

A method which is simpler to implement and also maintains positive definiteness of the polymer stress tensor is the *square-root method* [63]. An exact equation for the square root of the conformation tensor is advected rather than the conformation tensor itself, and therefore, the conformation tensor will remain positive. In this method Eq. (10.10) is replaced with an equation for $\mathbf{b}(\mathbf{x},t)$, the unique positive symmetric square root of $\mathbf{C}(\mathbf{x},t)$. The equation for $\mathbf{b}$ is

$$\frac{\partial \mathbf{b}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{b} = \mathbf{b}\cdot\nabla\mathbf{u} + \mathbf{a}\cdot\mathbf{b} + \frac{1}{2\mathrm{Wi}}\left((\mathbf{b}^T)^{-1} - \mathbf{b}\right), \tag{10.13}$$

where $\mathbf{a}$ is any anti-symmetric matrix. Furthermore, $\mathbf{a}$ can be prescribed uniquely so that if $\mathbf{b}$ is initially symmetric it will remain symmetric. In two dimensions the form of $\mathbf{a}$ is

$$\mathbf{a} = \begin{pmatrix} 0 & a_{12} \\ -a_{12} & 0 \end{pmatrix}, \tag{10.14}$$

where

$$a_{12} = \frac{\left(b_{12}\dfrac{\partial u}{\partial x} - b_{11}\dfrac{\partial v}{\partial x}\right) + \left(b_{22}\dfrac{\partial u}{\partial y} - b_{12}\dfrac{\partial v}{\partial y}\right)}{b_{11}+b_{22}},$$

for $\mathbf{u} = (u,v)$. An exact formula is also available in 3 space dimensions, but the details are more complicated [63]. This method was tested in a spectral framework, and it was observed that in practice the square-root method can be applied at higher Wi and for longer time than methods for evolving the conformation tensor directly [63]. This method does not address the exponential nature of the singularities of the polymer stress tensor like the log-conformation method, but the cost of implementation is no different than a direct implementation of the original model.

1.4.3 Polymer Stress Diffusion

As mentioned in Sect. 1.2, adding polymer stress diffusion will regularize the Stokes-Oldroyd-B equations, and in two space dimensions, the problem is well posed. In [37] it was shown that the exponential-in-time singularity obtained in [40] is removed with the addition of polymer stress diffusion, and smooth and bounded steady-state solutions can be found. Consider the polymer stress advection equation with diffusion, which we write in nondimensional form as

$$(\mathbf{C}-\mathbf{I}) + \mathrm{Wi}\overset{\triangledown}{\mathbf{C}} = \alpha\nabla^2\mathbf{C}. \tag{10.15}$$

If a steady flow, $\mathbf{u} = \mathrm{Wi}^{-1}(x, -y)$, is prescribed, this leads to a decoupling of Eq. (10.15), which is now linear in $\mathbf{C}$. An exact solution can be found which has the form

$$\mathbf{C} = \begin{pmatrix} -1 + Ae^{-y^2/(2\alpha)} & 0 \\ 0 & 1/3 \end{pmatrix}. \tag{10.16}$$

The Gaussian structure of $\mathbf{C}_{11}$ is a regularization of the delta-like singularities seen without diffusion in [40]. When comparing this local analytical solution to the numerical simulations the dependence on α and Wi is

$$\mathbf{C}_{11}(0, y) \approx -1 + C\,\mathrm{Wi}\,\alpha^{-1/2} e^{-y^2/(2\alpha)}. \tag{10.17}$$

These solutions are bounded and smooth for all $\alpha > 0$, and hence polymer diffusion may be used with some confidence as a regularization of the Oldroyd-B equation as long as care is taken to choose the length scale over which diffusion acts to be at or below the grid discretization.

1.5 Molecular Models

Instead of trying to address the singularities of the Oldroyd-B model directly, it is reasonable to criticize the molecular derivation of the Oldroyd-B model and use a model with a bounded extensional viscosity or a model which penalizes infinite extension of polymer coils. Many such models exist, and new molecular models are still being developed to match desired experimental data. Unfortunately, many modifications made at the molecular scale cannot be closed at the macroscopic level, resulting in multiscale or micro-macro models which are extremely computationally expensive. We discuss a few of the macroscopic models which are related to the Oldroyd-B model here and refer the interested reader to [2, 5, 41] for many other models. For various approaches to multiscale modeling of viscoelastic fluids see [26, 64, 65].

1.5.1 Giesekus Model

Like the Oldroyd-B model, the Giesekus model is also derived using a simple dumbbell model [66]. Giesekus proposed introducing an anisotropic drag force on the dumbbell which depends on the stress tensor based on the reasoning that the drag should be lower in the "direction" the fluid has been stressed. This could be true for a polymer solution in which the stress causes the molecules to line up in one direction resulting in lower drag in the direction of alignment. In this model, the drag

coefficient becomes a drag tensor and adds a nonlinear term to the Oldroyd-B model. Eq. (10.4) is replaced with

$$\boldsymbol{\tau}_p + \lambda \overset{\triangledown}{\boldsymbol{\tau}}_p + \alpha \frac{\lambda}{\eta_p} (\boldsymbol{\tau}_p)^2 = \eta_p \dot{\boldsymbol{\gamma}}. \tag{10.18}$$

This additional nonlinear term leads to physically realistic normal stress differences and bounded extensional viscosity but does not address infinite extension of polymer coils.

1.5.2 PTT Model

A very different way to derive a similar constitutive model comes from transient network model theory. Phan-Thien and Tanner [67] derived a model (called the PTT model) which assumes that polymers are entangled but they can break and reform. If the breaking rate increases with increasing average chain length, then a closed constitutive model can be derived which predicts shear thinning and bounded extensional viscosity. Eq. (10.4) is replaced with

$$\boldsymbol{\tau}_p + \lambda \overset{\triangledown}{\boldsymbol{\tau}}_p + \alpha \frac{\lambda}{\eta_p} \mathrm{Tr}(\boldsymbol{\tau}_p) \boldsymbol{\tau}_p = \eta_p \dot{\boldsymbol{\gamma}}. \tag{10.19}$$

1.5.3 Finite Extension Models

Another modification at the molecular level involves enforcing finite extensibility of polymer coils. The following modification to the linear Hooke's law was proposed by Warner [68]. The force is penalized if the polymers stretch beyond some given maximum stretch length, R_0, which results in a force law

$$\mathbf{F} = \frac{H\mathbf{R}}{(1 - \mathrm{Tr}\left(\mathbf{RR}\right)/R_0^2)}, \tag{10.20}$$

where H is a spring constant and $\mathbf{R}$ is the end-to-end vector of the dumbbell. The main drawback with this force law is that one cannot obtain a closed continuum model for the polymer stress tensor. There have been many closure approximations suggested, see for example [69–72]. The simplest and most commonly used approximation is to assume that the force depends on the average extension of the distribution of springs, and this leads to the FENE-P model [73]; see also Chap. 1. Including the full form for the force in Eq. (10.20) leads to micro-macro models which are numerically expensive and beyond the scope of this chapter.

1.6 Extensional Flow Simulations

Simulations with these models in place of the Oldroyd-B model show that neither finite extensibility nor maintaining positive definiteness of the polymer stress tensor will solve the high-Weissenberg number problem. An unavoidable cause of the problem seems to be that advection of the polymer stress near extensional points causes large stress gradients which create large forces on the fluid. In [40] it was shown that although the polymer stress was bounded for the FENE-P model, sharp gradients and corner singularities were still found in the simulations, perhaps as a consequence of the force penalization. Here we show new simulations using the Giesekus and PTT models, which are more stable than the Oldroyd-B model, but they still break down in long time simulations beyond a critical Wi. These simulations illustrate some of the ideas mentioned in Sects. 1.4 and 1.5 and highlight the need for careful consideration of the polymer stress tensor. We compare the PTT, Giesekus, and polymer stress diffusion models to the (Stokes) Oldroyd-B model in a simple 2D (periodic) extensional flow. We repeat the numerical experiment performed in [40] which involved solving the Stokes-Oldroyd-B equations in two space dimensions with a background force prescribed to enforce an extensional flow.

The various models used for the following example all have the same form, given non-dimensionally as

$$\nabla^2 \mathbf{u} - \nabla p + \xi \nabla \cdot \boldsymbol{\tau}_p + \mathbf{f} = 0, \tag{10.21}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{10.22}$$

$$(\mathbf{C} - \mathbf{I}) + \mathrm{Wi}\, \overset{\nabla}{\mathbf{C}} + \alpha \mathscr{R}(\mathbf{C}) = 0, \tag{10.23}$$

where the conformation tensor is related to the polymer stress tensor as $\boldsymbol{\tau}_p = \mathrm{Wi}^{-1}(\mathbf{C} - \mathbf{I})$. For the Stokes-Oldroyd-B equations we set $\mathscr{R} \equiv 0$; the other models are defined below.

Giesekus	$\mathscr{R}(\boldsymbol{\tau}_p) = (\boldsymbol{\tau}_p)^2$
PTT	$\mathscr{R}(\boldsymbol{\tau}_p) = \mathrm{Tr}(\boldsymbol{\tau}_p)\boldsymbol{\tau}_p$
polymer stress diffusion	$\mathscr{R}(\boldsymbol{\tau}_p) = -\nabla^2 \boldsymbol{\tau}_p$

$\mathrm{Wi} = \lambda / T_f$ is the *Weissenberg number*, with λ the polymer relaxation time and T_f the time scale of the fluid flow. The dimensional scaling F of the forcing $\mathbf{f}$ is used to set the flow time scale as $T_f = \eta_s / \rho L F$, where η_s is the solvent viscosity, ρ the fluid density, and L the system size. This sets the dimensionless force and the time scale of transport to be order one. The parameter $\xi = G T_f / \eta_s$ measures the relative contribution of the polymer stress to momentum balance, where G is the isotropic stress in the polymer field in the absence of flow. Note that the parameter ξ is the ratio of the polymer viscosity to the solvent viscosity and in what follows we set $\xi = 0.5$.

The Giesekus and PTT models both have bounded extensional viscosity, unlike Oldroyd-B, although we will see below that they still have large stress gradients near extensional points in the flow for sufficiently large Wi. In what follows, the parameter α is fixed at 0.001. We note that for this value, the resulting length scale for the polymer stress diffusion is smaller than the grid spacing.

The numerical experiment in [40] involved analyzing the stress near hyperbolic extensional points in the flow. The background force

$$\mathbf{f} = \begin{pmatrix} 2\sin x \cos y \\ -2\cos x \sin y \end{pmatrix}. \tag{10.24}$$

sets up a four vortex "mixer" in each $[-\pi,\pi]^2$ cell, which in a purely Newtonian Stokes flow (Wi $= 0$) has solution $\mathbf{u} = -\mathbf{f}/2$.

This 2D periodic "4-roll mill" geometry provides an opportunity to compare solutions to these different models in an extensional flow with no boundary effects. Given the regular domain, this problem is well suited to a pseudo-spectral method. Furthermore, as the local analytical solution in Eq. (10.6) suggests, beyond a critical Wi the polymer stress $\boldsymbol{\tau}_p$ grows exponentially near extensional points, and using a pseudo-spectral method allows one to analyze the regularity and evolution of the singularity in Fourier space [74–76].

The algorithm used in [40] for solving (10.21)–(10.23), which is similar here for $\mathscr{R} \neq 0$, was to prescribe initial data for $\boldsymbol{\tau}_p$, invert the Stokes equation to find the velocity, and with that update the polymer stress via any appropriate time stepping method (second-order Adams-Bashforth was used in [40] and will be used here). We set $\mathbf{C}(0) = \mathbf{I}$ for isotropic initial data. Inverting the Stokes equation amounts to solving for u_j in Fourier space ($j = 1, 2$)

$$\hat{u}_j = \frac{1}{|\mathbf{k}|^2}\left[ik_j\hat{p} + i\xi k_\ell(\hat{\boldsymbol{\tau}}_p)_{\ell j} + \widehat{f_j}\right], \tag{10.25}$$

where the pressure is found using the incompressibility constraint

$$\hat{p} = \frac{1}{|\mathbf{k}|^2}\left[\xi k_j k_\ell(\hat{\boldsymbol{\tau}}_p)_{\ell j} - ik_j\widehat{f_j}\right], \tag{10.26}$$

with summation convention applied for all repeated indices. The stress update is also performed in Fourier space but care must be taken as the stress advection equation is nonlinear. In Eq. (10.23) the quadratic nonlinearities are computed using de-aliasing techniques [77]. We use a filter to zero the high wave numbers before inverting and multiplying the terms in real space to avoid aliasing errors. The filter we apply is similar to a simple 2/3 cutoff, but instead we apply a smooth rapidly decaying exponential cutoff [78] which helps stabilize the simulations. The Fourier transform is applied again and the nonlinear terms are used to update the polymer stress.

Simulations of the Stokes-Oldroyd-B model ($\mathscr{R} = 0$) were performed with this algorithm for $n^2 = 1024^2$ grid points in the $[-\pi,\pi]^2$ domain. In [40] two critical

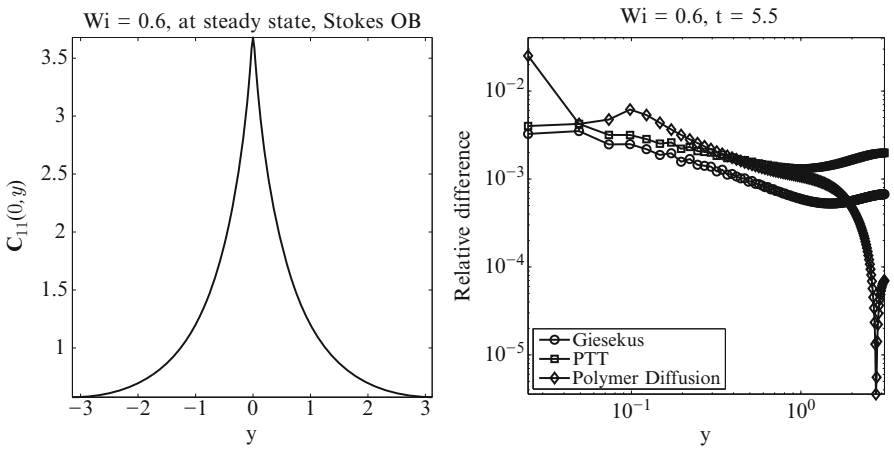

Fig. 10.2 Wi = 0.6 at $t = 5.5$: (*left*) $\mathbf{C}_{11}(0,y)$ the first component of the conformation tensor along the axis of compression where solution is at steady state. (*right*) Relative difference between models (Giesekus, PTT, and Polymer diffusion) and the Stokes-OB solution

Weissenberg numbers were identified from the numerical simulations: for Wi < 0.5 it was seen that the polymer stress was smooth, for $0.5 <$ Wi < 1 the stress approached a finite-valued cusp exponentially in time and for Wi > 1 the stress was diverging exponentially in time. These different polymer stress solutions correspond to different modifications to the velocity field. For sufficiently small Weissenberg number, Wi $\lesssim 1$, the stress perturbation only modifies the amplitude of the flow, namely $\mathbf{u} \approx C\mathbf{f}$ where C depends on Wi. The dependence of C on Wi is described in [40]. However for Wi $\gtrsim 1$ the 4-roll mill structure persists, but oppositely signed vortices arise along the axis of extension and compression. Figure 10.1a shows a contour plot of $\mathrm{Tr}(\mathbf{C})$ at $t = 8$ for Wi $= 5.0$. At $t = 8$ the solution is at "near-steady state." Although the polymer stress is still increasing in a neighborhood of the axis of compression and extension, the size of that neighborhood is decreasing, and the L^2 norm of the stress is nearly constant as is the L^2 norm of the velocity. The stress has concentrated along the stable and unstable manifolds of the extensional stagnation points in the flow. These stress regions are localized and need to be well resolved for accurate simulations. Figure 10.1b shows a contour plot of the vorticity at the same time. The near-delta-function stress creates near-cusps in the velocity field which yield near-jumps in the vorticity. These features are seen in detail when looking at slices of the polymer stress and vorticity in Figs. 10.1c and d.

In what follows, to compare the different models, we look at two singular cases separately, namely Wi $= 0.6$ which has a cusp solution and Wi $= 2.0$ which has diverging solutions. Simulations were done for the Stokes-Oldroyd-B model ($\mathscr{R} \equiv 0$) for Wi $= 0.6$ and Wi $= 2.0$ with $n^2 = 1024^2$ grid points in the $[-\pi,\pi]^2$ domain. These "exact solutions" are compared with coarser solutions of the various models: Giesekus, PTT, polymer diffusion, at $4\times$ coarser resolution, $n^2 = 256^2$, and $\alpha = 0.001$.

1.6.1 Cusp Solution: Wi = 0.6

Figure 10.2 (left) shows the solution, $\mathbf{C}_{11}(0,y)$, for Stokes OB with $\mathrm{Wi} = 0.6$ at $t = 5.5$. At this time, the solution has reached steady state, the maximum difference in the stress over 0.1 time units is $\mathcal{O}(10^{-3})$, and the maximum difference in the velocity is 2 orders of magnitude smaller. The cusp in the polymer stress is located at the extensional point in the flow $(0,0)$. The other constitutive models were run at $4\times$ coarser resolution, $n^2 = 256^2$ grid points, and are compared with the Stokes-OB solution in Fig. 10.2 (right). The different models agree well with Stokes OB. As expected, the polymer stress diffusion model fails to capture the cusp. However, outside an $\mathcal{O}(10^{-1})$ region near the extensional point all three models are accurate to 2 digits. Examining the Fourier spectrum for each of these models (not shown) reveals that the polymer diffusion model has a decaying spectrum similar to that of the higher resolution Stokes-OB solution. The spectra of the Giesekus and PTT models decay much less rapidly, which indicates the approach of a singularity. Long time simulations with polymer diffusion go to steady state, while the other models and the Stokes-OB simulations will eventually break down from oscillations as the solutions become more singular over time.

1.6.2 Diverging Solution: Wi = 2.0

When $\mathrm{Wi} = 2.0$ the results from [40] and the local analytical solution indicate that the polymer stress should be diverging near the extensional point at $(0,0)$. Although the stress is diverging, the set on which the stress is growing diminishes in time so that the resultant velocity field approaches a steady state. In these simulations between $t = 6.9$ and $t' = 7.0$ we see $\sup_{(x,y)} |\mathbf{u}(t) - \mathbf{u}(t')| = \mathcal{O}(10^{-4})$, although the polymer stress is diverging. If we revisit Fig. 10.1, we see that the effect of the concentration of polymer stress is to create recirculation cells in the vorticity. Figure 10.3a shows a slice of the $x-$component of the velocity $\mathbf{u}_1(\pi/2,y)$, at $t = 7$ (near-steady state) for $\mathrm{Wi} = 2$. This value of Wi is beyond the coil-stretch transition and we see that the diverging stress leads to a more significant modification to the flow than the near-cusp stresses for $\mathrm{Wi} = 0.6$. In Fig. 10.3a the flow perturbation occurs near $u_1(\pi/2,0)$ and as $t \to \infty$ this becomes a near-corner singularity. These near-corners lead to the near-jumps in the vorticity seen in Fig. 10.1b.

Figure 10.3b compares the other constitutive models (at $4\times$ coarser resolution) to the Stokes-OB solution for the velocity $u_1(\pi/2,y)$ for $\mathrm{Wi} = 2$, at $t = 7$. The results are plotted on a log-log scale for $0 < y < \pi/2$. Here the polymer stress diffusion model captures the near-corner singularity almost one order of magnitude better than the Giesekus and PTT models (whose data lies practically on top of one another in the figure). Figures 10.3c and d show the Fourier spectra of the spatial data in Fig. 10.3a and b, respectively. The Stokes-OB solution is well resolved at this time with $n^2 = 1024^2$ grid points, but the polymer diffusion model captures the essential features of the flow and is well resolved with $n^2 = 256^2$ grid points. The Giesekus and PTT models are both beginning to lose accuracy in the high frequencies which

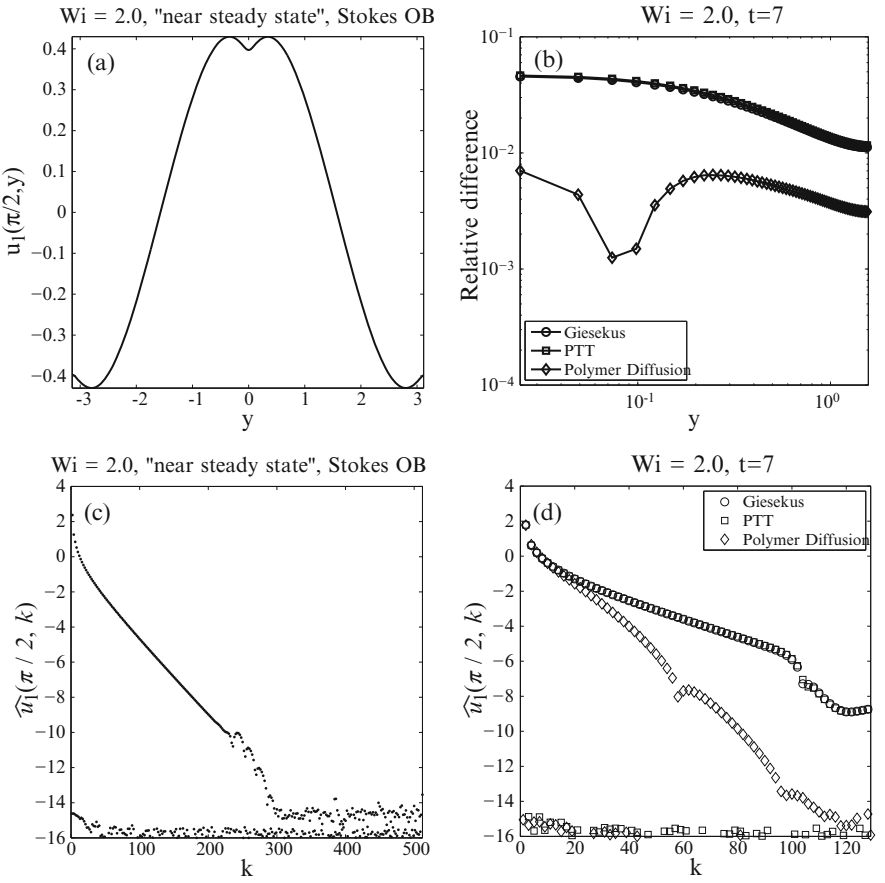

Fig. 10.3 Wi = 2.0 at $t = 7$: (**a**) $\mathbf{C}_{11}(0,y)$ the first component of the conformation tensor along the axis of compression where solution is "near-steady state." (**b**) Difference between models (Giesekus, PTT, and polymer diffusion) and the Stokes-OB solution. (**c**) and (**d**) Fourier spectra $\hat{\mathbf{u}}_1(\pi/2,k)$ for different models

may indicate that the solutions are approaching a singularity. Long time simulations of this problem break down for the Stokes-OB, PTT, and Giesekus model, whereas the polymer stress diffusion smooths the singularity and the solutions converge to steady state.

Simulations of FENE-P in this simple framework were done in [40]. It was shown that simply adding a cutoff to the polymer stress does not solve the problem of large stress gradients in extensional flows. The cusp-type singular solutions for $1/2 < \mathrm{Wi} < 1$ were still found for a sufficiently large maximum extension parameter ℓ and for $\mathrm{Wi} > 1$ the diverging solutions became bounded for finite ℓ, but numerical evidence indicated an exponential approach to a pair of corner singularities. Some numerical smoothing appears to be necessary even for this finite extension model for accurate long time simulations.

1.6.3 Simulation Conclusions

We see from this simple numerical test that long time simulations for strongly elastic flows require care in choice of model and method. These high-stress regions arise in many continuum models based on the upper-convected derivative (Eq. (10.5)) and care must be taken to resolve these regions for accurate information about the polymer stress contribution to the flow. Adding polymer stress diffusion leads to a nice balance of smoothing the stress locally while closely matching the flow outside the small smoothing region.

It is important to be aware of underlying near-singularities in these complex fluid models when designing numerical experiments. Model modifications which add finite extension (FENE-P) or which produce finite steady-state extensional viscosity (PTT and Giesekus) do not entirely get away from the underlying difficulties with the polymer stress advection equation given in Eq. (10.4). When long time simulations are required, these stress near-singularities can lead to loss of accuracy in numerical simulations unless care is taken to identify and address those features. The addition of polymer stress diffusion as a numerical smoothing, carefully chosen, does a good job capturing the fine structures while maintaining accuracy over long times.

Biological applications nearly always involve complex flows near walls and around obstacles and these flows contain extensional stagnation points which are subject to high-Weissenberg number problems. Near these regions, the flow has to be sufficiently well resolved to incorporate the fine stress structures which arise and feedback to create nontrivial flow patterns. The relaxation time of the elastic stress also introduces a new time scale which needs to be considered. Coupling fluids with structures adds further challenges which need to be addressed separately.

2 Immersed Boundary Methods

A common challenge of computational fluid mechanics is solving the equations of motion in complex geometry. Biological systems are particularly challenging because they often involve the coupled dynamics of active elastic structures and the surrounding fluid. There are many examples of such systems involving complex fluids, some of which were discussed in earlier chapters: beating cilia in mucus in the respiratory system, swimming sperm in the mucus of the female reproductive tract, and peristaltic pumping in the reproductive and digestive systems. In each of these systems, the collective dynamics of the material is an emergent phenomenon and is the product of the interactions between the fluid and the elastic material.

A popular method for biological fluid dynamics problems involving large deformations is the immersed boundary (IB) method [79]. The IB method was originally developed to study blood flow in the heart [80], and it has been applied to a large range of biological and nonbiological systems over the past thirty years. The IB method uses two coordinate systems: a moving Lagrangian coordinate

system for the immersed structures and a fixed Eulerian coordinate system for the fluid. The Eulerian domain is discretized with a regular, Cartesian mesh which permits the use of fast methods for solving the equations of fluid mechanics. A key feature of the method is that it does not require conforming discretizations of the fluid and structure; instead, the curvilinear mesh is free to cut through the background Cartesian grid in an arbitrary manner. Consequently, IB simulations do not require dynamic grid generation, even for problems involving very large structural deformations.

The popularity and longevity of the immersed boundary method are partly due to its robustness and simplicity. The method is highly adaptable, and codes require few changes to be modified for different applications. Typical implementations of the IB method generally require only solvers for the fluid equations along with routines to compute elastic forces and to transfer data between the Lagrangian and Eulerian grids. IB codes can be built on top of existing codes for solving the equations of the fluid. In fact, the computational examples presented in Sect. 3 use the same viscoelastic fluid solver for Stokes Oldroyd-B with polymer diffusion that was used to generate the results in Sect. 1. In Chap. 11, a boundary integral method is presented to simulate the dynamics of a suspension. We note that boundary integral methods can only be used for linear equations, and they do not extend to the Stokes Oldroyd-B model. In this way, immersed boundary methods are more general.

Recently, the IB method has been used to investigate several classical low Reynolds number bio-fluid problems involving complex fluids in place of Newtonian fluids. For example, [57, 81] use the IB method to investigate peristaltic pumping of viscoelastic fluids, and [82–84] use the IB method to explore the swimming of microorganisms in complex fluids. The goal of this chapter is not to review the many variants and applications of immersed boundary methods, but rather to highlight key considerations when using immersed boundary methods for applications involving strongly elastic fluids at low Reynolds number. In the remainder of this section, we briefly introduce the key ideas of the immersed boundary method. We pay particular attention to time stepping algorithms. As we show in the later sections, traditional explicit-time methods are severely limited for strongly elastic flows with stiff boundaries.

As discussed in Sect. 1, viscoelastic flows tend to form regions of high stress with sharp velocity gradients near extensional points. These regions of highly concentrated stress can be generated by active immersed objects in the fluid, and resolving these stresses requires finer grid resolution than similar problems of Newtonian fluid mechanics. Viscoelastic stresses introduce additional time scales into the problem, and at moderate to high-Weissenberg numbers, the stress evolves on very long time scales. Capturing these slow dynamics requires performing long time simulations. In Sect. 3, we illustrate these ideas using an example problem of a swimming organism in a viscoelastic fluid.

2.1 Immersed Boundary Equations

Let $\mathbf{x} \in \boldsymbol{\Omega}$ denote fixed physical coordinates, with $\boldsymbol{\Omega} \subset \mathbb{R}^{n_\mathrm{f}}$ being the physical domain where $n_\mathrm{f} = 2$ or 3 is the dimension of the fluid. Let $\mathbf{s} \in \Gamma$ denote material coordinates attached to the immersed structure, with $\Gamma \subset \mathbb{R}^{n_\mathrm{s}}$ denoting the Lagrangian coordinate domain and n_s denoting the dimension of the structure. The physical location of material point $\mathbf{s}$ at time t is given by $\mathbf{X}(\mathbf{s},t) \in \boldsymbol{\Omega}$. The name *immersed boundary method* suggests that the elastic structure is a thin interface, i.e., an object of codimension one with respect to the fluid ($n_\mathrm{s} = n_\mathrm{f} - 1$). While this is the case in many applications of the IB method, this formulation applies equally well to immersed structures that have nonzero thickness.

In many biological problems, the structure has the same density as the fluid, and thus the combined fluid and structure can be described by a single momentum balance equation, and their motion can be described by a single velocity field. In the absence of other loading, the forces generated by the deformations of the structure drive the motion of the fluid through a body force term in the balance of momentum equation. In this chapter we consider the boundary to be immersed in an Oldroyd-B fluid (with diffusion coefficient α) at zero Reynolds number. The system of equations describing the fluid and structure is

$$\eta_s \Delta \mathbf{u} - \nabla p + \nabla \cdot \boldsymbol{\tau}_p + \mathbf{f} = 0, \tag{10.27}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{10.28}$$

$$\boldsymbol{\tau}_p + \lambda \overset{\triangledown}{\boldsymbol{\tau}}_p = \mu_p \dot{\boldsymbol{\gamma}} + \lambda \alpha \Delta \boldsymbol{\tau}_p, \tag{10.29}$$

$$\mathbf{f}(\mathbf{x},t) = \int_\Gamma \mathbf{F}(\mathbf{s},t)\, \delta(\mathbf{x} - \mathbf{X}(\mathbf{s},t))\, \mathrm{d}\mathbf{s}, \tag{10.30}$$

$$\frac{\partial \mathbf{X}(\mathbf{s},t)}{\partial t} = \mathbf{U}(\mathbf{s},t) = \int_{\boldsymbol{\Omega}} \mathbf{u}(\mathbf{x},t)\, \delta(\mathbf{x} - \mathbf{X}(\mathbf{s},t))\, \mathrm{d}\mathbf{x}, \tag{10.31}$$

in which $\mathbf{u}(\mathbf{x},t)$ is the velocity field, $\mathbf{U}(\mathbf{s},t)$ is the velocity of the structure, $p(\mathbf{x},t)$ is the pressure, $\boldsymbol{\tau}_p(\mathbf{x},t)$ is the viscoelastic stress, $\mathbf{f}(\mathbf{x},t)$ is the Eulerian elastic force density generated by the immersed structure, $\mathbf{F}(\mathbf{s},t)$ is the Lagrangian elastic force density generated by the immersed structure, and δ denotes the Dirac delta function.

The integral operator in (10.30) that determines the Eulerian force density from the Lagrangian force density is called the *spreading* operator, which we denote by S. The *interpolation* operator that transfers the velocity to the structure is the adjoint of the spreading operator. Using this notation, equations (10.30) and (10.31) can be compactly expressed as

$$\mathbf{f} = S\mathbf{F}, \tag{10.32}$$

$$\frac{\partial \mathbf{X}}{\partial t} = \mathbf{U} = S^* \mathbf{u}, \tag{10.33}$$

respectively. Note that the spreading operator depends on the current position of the structure. For a discussion of discretizing these operators see [79].

A constitutive law which relates the boundary configuration to the force is needed to complete the description of the system. The form of the constitutive law depends on the application being considered. In Sect. 3 we present an example in which penalty forces are used to actively drive the immersed boundary to follow a prescribed shape to mimic the undulatory stroke of a swimming worm.

2.2 Explicit-Time Stepping

Typical implementations of the IB method use a fractional time stepping approach to solve the equations of motion. In the simplest version of such a scheme, the fluid velocity, pressure, and viscoelastic stress are updated while keeping the position of the structure fixed, and then the structural position is updated using the newly computed velocity. In this section, we consider that the viscoelastic stress is known at the beginning of a time step and thus can be described by its force density $\mathbf{f}_{\text{ve}} = \nabla \cdot \boldsymbol{\tau}_p$. For the model equations considered here, the explicit-time method advances the solution variables from time $t^n = n\Delta t$ to time $t^{n+1} = (n+1)\Delta t$ via

$$\eta_s \Delta_h \mathbf{u}^{n+1} - \nabla_h p^{n+1} + \mathbf{f}_{\text{ve}}^{n+1} + S_h^n \mathbf{F}(\mathbf{X}^n) = 0, \tag{10.34}$$

$$\nabla_h \cdot \mathbf{u}^{n+1} = 0, \tag{10.35}$$

$$\mathbf{X}^{n+1} = \mathbf{X}^n + \Delta t \, (S_h^n)^* \, \mathbf{u}^{n+1}. \tag{10.36}$$

Notice that the explicit-time method effectively decouples the computation of the boundary mechanics from the computation of the fluid velocity, pressure, and viscoelastic stress. This decoupling makes it easy to use existing codes for solving the fluid mechanics for IB simulations. In this way, the IB method for complex fluids is identical to the IB method for Newtonian fluids.

It is well known that for applications involving stiff elastic structures, the explicit-time method requires very small time steps to maintain stability. We let $\mathscr{L}^{-1}$ denote the inverse Stokes operator which maps force densities to velocity fields. The boundary update equation (10.36) can be expressed as

$$\mathbf{X}^{n+1} = \mathbf{X}^n + \Delta t \, (S_h^n)^* \, \mathscr{L}^{-1} S_h^n \mathbf{F}(\mathbf{X}^n), \tag{10.37}$$

where we have suppressed the viscoelastic forces for simplicity. This expression shows that the explicit-time scheme is essentially a forward-Euler method for the boundary positions, which explains the origin of the stability restriction. For many constitutive laws, as the grid is refined the problem becomes increasingly stiff. Because viscoelastic fluids require high grid resolution and sometimes involve long time integration, the stability restriction may be more limiting than for Newtonian fluids.

2.3 Implicit-Time Stepping

Much effort has been devoted both to understanding and to alleviating the severe time step restriction of fractional step IB methods [85–87]. The solution methods used in early implicit IB methods were not efficient and were not competitive with explicit methods [88], and some semi-implicit methods intended to allow for large time steps still suffered from significant time step restrictions [89,90]. Newren *et al.* [86] analyzed the origin of instability in semi-implicit IB methods using energy arguments, and they gave sufficient conditions to achieve unconditional stability in the sense that the total energy is bounded independent of the time step size. An important result by Newren *et al.* [86] is that it is not necessary to employ a *fully* implicit-time discretization to achieve unconditional stability, but the stable time stepping schemes proposed therein do simultaneously solve for both the Eulerian velocity field and the Lagrangian structural configuration. As indicated by the early experience with implicit IB methods, however, developing efficient solvers for the coupled equations is challenging.

More recently, a number of stable semi-implicit [91–94] and fully implicit [95, 96] IB methods have been developed. The efficiency of these methods is generally competitive with explicit methods, and in some special cases, these implicit schemes can be faster than explicit methods by several orders of magnitude. Many implicit methods use a Schur complement approach to reduce the coupled Lagrangian-Eulerian equations to purely Lagrangian equations [94, 95, 97]. These methods achieve a substantial speedup over explicit methods especially when there are relatively few Lagrangian mesh nodes [94]. An open question is whether there exist robust, general-purpose implicit methods that are more efficient than explicit methods or whether specialized methods must be developed for specific problems.

We present an example of a semi-implicit method which is very similar to methods presented in [91, 94, 95]. In our implicit-time method, the fluid velocity at time t^{n+1} depends on structure forces at time t^{n+1}, rather than on the forces at time t^n as in the explicit method. Again we consider the viscoelastic forces as given. The system of equations that must be solved for the fluid velocity, pressure, and structure position is

$$\eta_s \Delta_h \mathbf{u}^{n+1} - \nabla_h p^{n+1} + \mathbf{f}_{\text{ve}}^{n+1} + S_h^n \mathbf{F}(\mathbf{X}^{n+1}) = 0, \tag{10.38}$$

$$\nabla_h \cdot \mathbf{u}^{n+1} = 0, \tag{10.39}$$

$$\mathbf{X}^{n+1} = \mathbf{X}^n + \Delta t \, (S_h^n)^* \, \mathbf{u}^{n+1}. \tag{10.40}$$

Notice that in this time stepping scheme, the structural positions used to define the spreading and interpolation operators are lagged in time. As shown by Newren *et al.* [86], this scheme is unconditionally stable, despite the fact that the positions of the spreading and interpolation operators are treated explicitly rather than implicitly. This type of scheme is often termed semi-implicit to emphasize that not all of the terms are treated implicitly.

Rather than solve for the velocity, pressure, and boundary position simultaneously, we reduce the equations to a single equation for the unknown boundary positions. Using the inverse operator for the Stokes equations, $\mathscr{L}^{-1}$, the boundary update equation (10.40) can be written as

$$\mathbf{X}^{n+1} = \mathbf{X}^n + \Delta t\,(S_h^n)^*\,\mathscr{L}^{-1} S_h^n \mathbf{F}(\mathbf{X}^{n+1}), \tag{10.41}$$

which resembles a backward-Euler scheme for advancing the boundary positions. This is a nonlinear equation, which we solve by applying Newton's method to

$$\mathbf{G}(\mathbf{X}) = \mathbf{X} - \mathbf{X}^n - \Delta t\,(S_h^n)^*\,\mathscr{L}^{-1} S_h^n \mathbf{F}(\mathbf{X}) = 0. \tag{10.42}$$

Let $\mathbf{X}^k$ represent the approximate solution at the k^{th} step of the Newton iteration. Each Newton step involves the update

$$\mathbf{J}\delta\mathbf{X} = -\mathbf{G}(\mathbf{X}^k) \tag{10.43}$$

$$\mathbf{X}^{k+1} = \mathbf{X}^k + \delta\mathbf{X}, \tag{10.44}$$

where $\mathbf{J}$ is the Jacobian of $\mathbf{G}$. We do not explicitly form $\mathbf{J}$. Rather, we perform multiplication by $\mathbf{J}$ as described below, and we solve equation (10.43) using the generalized minimum residual method (GMRES).

Let $\mathbf{J}_F$ represent the Jacobian of the force function, $\mathbf{F}$. The Jacobian of $\mathbf{G}$ may be expressed as

$$\mathbf{J} = \mathbf{I} - \Delta t\,(S_h^n)^*\,\mathscr{L}^{-1} S_h^n \mathbf{J}_F. \tag{10.45}$$

The product $\mathbf{J}\delta\mathbf{X}$ is accomplished by first explicitly multiplying by $\mathbf{J}_F$ (which is sparse), then spreading the result to the grid, solving the Stokes equations, and interpolating back to the structure. This procedure avoids the need to explicitly form $\mathbf{J}$. The GMRES solver requires a good preconditioner for efficient performance. As a preconditioner, we ignore the Stokes solve and use

$$\mathbf{M} = \mathbf{I} - \Delta t\,(S_h^n)^*\,S_h^n \mathbf{J}_F. \tag{10.46}$$

This matrix is sparse and relatively small (number of the IB points) and can be factored quickly. The performance of the method is discussed in Sect. 3.5.

Note that each evaluation of the objective function, $\mathbf{G}$, as well as each application of the Jacobian, J, involves the application of $\mathscr{L}^{-1}$. This operator is never explicitly constructed. Its application is achieved by solving the Stokes equations for the fluid velocity. This implicit-time method uses the same code as the explicit-time method for finding the fluid velocity. Although the immersed boundary code for the implicit-time method is significantly more involved than the explicit-time method, this method retains the appealing modularity of the original IB method and allows the algorithms presented in Sect. 1 to be used without modification.

3 Locomotion of Undulatory Swimmers

Locomotion of microorganisms at low Reynolds number occurs in numerous biological processes, and swimming in a Stokesian Newtonian fluid has been extensively studied and the underlying physics is well understood. See [98] for a review of low Reynolds number locomotion. There have been many recent theoretical studies on locomotion in complex fluids [83, 84, 99–106]. Asymptotic analyses of infinitely long, small-amplitude, undulatory swimmers in a viscoelastic fluid showed that swimming is hindered by the addition of elastic stresses [100,101]. However, numerical simulation of finite-length large-amplitude swimmers in a viscoelastic fluid showed that under some conditions, the swimming speed may be enhanced by the elastic stresses [83]. A similar enhancement was shown for numerical simulations of infinite-length helical swimmers with large pitch angles [105]. The results from these papers highlight the importance of computational methods in exploring problems that are beyond the reach of asymptotic analysis.

In the remainder of this chapter, we use the problem of a finite-length free swimmer to illustrate the ideas of the immersed boundary method. We pay particular attention to the additional complications introduced from the viscoelastic fluid, namely, the need for high grid resolution and the presence of long time dynamics at high-Weissenberg numbers. The problem we explore is very similar to that presented in [83], except that the stroke pattern of the swimmer is based on data for a swimming nematode presented in Chap. 7 of this book and in [107]. In this chapter, we primarily use this problem as a computational example of the methodology. For a more in-depth analysis of how the fluid elasticity, body elasticity, and stroke kinematics affect swimming speed in this problem, see [106].

3.1 Swimmer Model

The worm is modeled as an inextensible infinitely thin sheet, which in two dimensions is a curve in the plane. The position of the worm is given by $\mathbf{X}(s,t)$, where $s \in [0,L]$ is the Lagrangian coordinate. The swimming is driven using a prescribed target curvature $\kappa_0(s,t)$, which in the absence of resistance from the surrounding medium represents the shape of the worm.

3.1.1 Immersed Boundary Forces

Both the inextensibility and the shape are enforced by forces that are designed to penalize extension and deviation from the prescribed curvature. These forces are derived from expressions for the bending and extension (stretching) energy, which are given below. For a given configuration of the worm the energy from stretching is

$$E_{\mathrm{s}} = \frac{k_s}{2} \int_{\Gamma} \left(|\mathbf{X}_s| - 1 \right)^2 ds, \tag{10.47}$$

where k_s is a stiffness coefficient. The bending energy is

$$E_{\mathrm{b}} = \frac{k_b}{2} \int_{\Gamma} \left(\kappa - \kappa_0(s,t) \right)^2 ds, \tag{10.48}$$

where k_b is the bending stiffness, κ is the curvature of the worm, and κ_0 is the prescribed target curvature. The total energy is the sum of the bending and stretching energy:

$$E = E_{\mathrm{s}} + E_{\mathrm{b}}. \tag{10.49}$$

The Lagrangian force densities come from the variational derivative of the total energy:

$$\frac{\delta E}{\delta \mathbf{X}} \tilde{\mathbf{X}} = -\int_0^L \mathbf{F} \tilde{\mathbf{X}} ds = -\int_0^L (\mathbf{F}_s + \mathbf{F}_b) \tilde{\mathbf{X}} ds, \tag{10.50}$$

where $\mathbf{F}_s$ and $\mathbf{F}_b$ are the force densities corresponding to stretching and bending, respectively. The expressions for the force densities are derived by first discretizing the structure (and hence the energy functional) and then taking the variational derivative of the discrete energy. The advantage of this approach is that it guarantees that the total forces discretely sum to zero, which is a requirement to be able to find a solution to Stokes equations in a periodic domain.

The signed curvature is computed using a discretized version of

$$\kappa = \hat{\mathbf{n}} \cdot \frac{\partial \hat{\mathbf{t}}}{\partial s} = \hat{\mathbf{n}} \cdot \frac{\partial^2 \mathbf{X}}{\partial s^2}, \tag{10.51}$$

where $\hat{\mathbf{t}}$ is the tangent vector and $\hat{\mathbf{n}}$ is the normal vector. We are assuming that the material is inextensible and the tangent vector is expressed as $\hat{\mathbf{t}} = \mathbf{X}_s$. We compute the discrete curvature at an interior point using the expression

$$\kappa_j = \left(\frac{\hat{\mathbf{n}}_{j+1/2} + \hat{\mathbf{n}}_{j-1/2}}{2} \right) \cdot \left(\frac{\hat{\mathbf{t}}_{j+1/2} - \hat{\mathbf{t}}_{j-1/2}}{\Delta s} \right), \tag{10.52}$$

where the discrete tangent vector is

$$\hat{\mathbf{t}}_{j+1/2} = \frac{\mathbf{X}_{j+1} - \mathbf{X}_j}{\Delta s} \tag{10.53}$$

and the discrete normal is the $\pi/2$ rotation of the tangent. The discretized expression for the total energy is

$$E = \frac{k_s}{2}\sum_{j=1}^{N-1}\left(\left|\frac{\mathbf{X}_{j+1}-\mathbf{X}_j}{\Delta s}\right|-1\right)^2\Delta s + \frac{k_b}{2}\sum_{j=2}^{N-1}\left(\kappa_j-\kappa_{0j}\right)^2\Delta s. \tag{10.54}$$

3.1.2 Curvature Model

We take shape data of a nematode swimming in the water to identify the target curvature.[2] The measured curvature as a function of time and position along the worm are shown in Fig. 10.4. The body coordinate (in units of mm) runs from $s = 0$ at the head to $s = 1.2$ at the tail. The worm shows a periodic motion with a dominant frequency around 2 Hz (period of 0.5 s), which is evident in the curvature data.

For the model swimmer, we use a curvature function of the form

$$\kappa_0 = A(s)\cos\left(\frac{2\pi}{T}\left(t+\phi(s)\right)\right), \tag{10.55}$$

where A is the amplitude, ϕ is the phase, and $T = 0.5$ s is the period. We identify the phase function, ϕ, by finding the peak cross-correlation between the head and points along the body. This phase function and a linear fit to it are shown in Fig. 10.5. The slope of the linear fit is -0.250 s/mm, and so we use a phase function of $\phi(s) = -s/4 + \phi_0$, where ϕ_0 is the value of the phase assigned to the head.

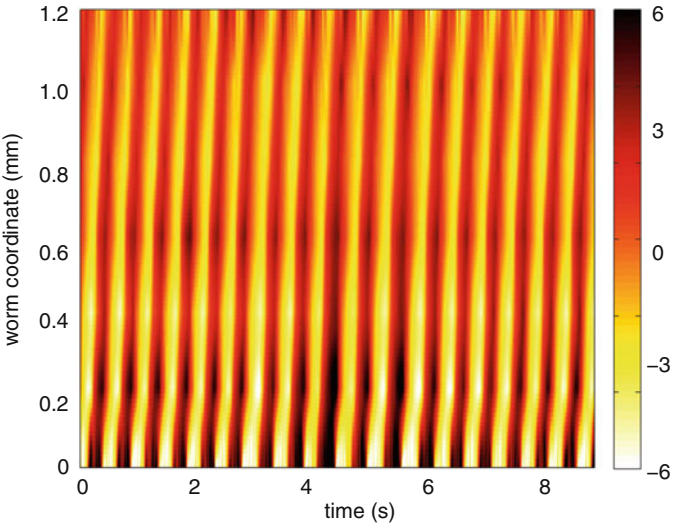

Fig. 10.4 Color field of the curvature (in mm^{-1}) data of the swimming nematode as a function of time and body coordinate along the worm. $s = 0$ is the head and $s = 1.2$ is the tail

[2]The data are kindly provided by Paulo Arratia and Xiaoning Shen.

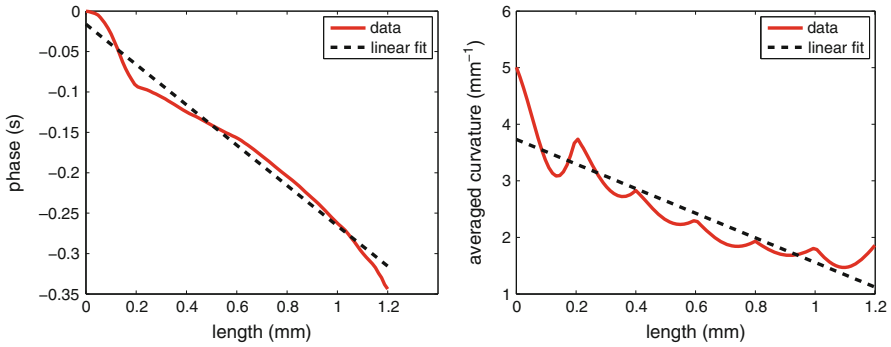

Fig. 10.5 Phase (*left*) and amplitude (*right*) functions based on the curvature data of a swimming nematode and their linear fits that are used to define the curvature in the model computations

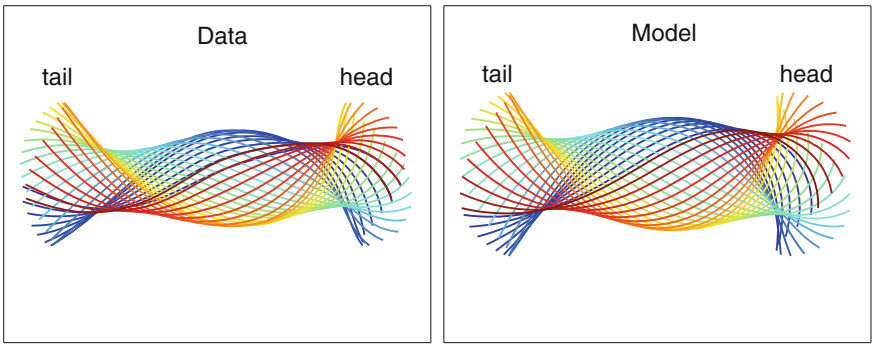

Fig. 10.6 Time sequence of worm shapes from the data (*left*) and model (*right*). The colors correspond to the time progressing from blue to red. The body is positioned with a fixed center of mass and horizontal end-to-end vector

We compute the time-averaged curvature of the data by

$$\bar{\kappa}(s) = \sqrt{\frac{1}{N}\sum_{t_j} \kappa(s,t_j)^2} \tag{10.56}$$

and fit this with a linear function. The averaged curvature and the linear fit are displayed in Fig. 10.5. We choose the amplitude of the curvature in the model by $A(s) = \sqrt{2}\bar{\kappa}_{\text{fit}}$, so that the model matches the data in the mean squared curvature.

Figure 10.6 shows a time sequence of the actual shapes and the shapes produced by the model fit. Changes in curvature propagate as phase waves from the head ($s = 0$), where the amplitude of the curvature is largest, to the tail. In what follows we call this type a swimmer a "burrower." The exact form of the curvature function used to drive the swimming worm is

$$\kappa_0^{\text{burrower}} = (5.3 - 3.1s)\cos\big(4\pi(t - s/4 + \phi_0)\big), \tag{10.57}$$

where $\phi_0 = L/4$ is the phase shift chosen to have zero phase at the tail. We note that in [83] the desired motion of the swimmer involved waves of curvature that propagated with an increasing amplitude along the direction of the wave. We call this type of swimmer a "kicker." For comparison we simulate a kicker using the change of coordinates

$$\kappa_0^{\text{kicker}}(s,t) = \kappa_0^{\text{burrower}}(L - s, t_0 - t), \tag{10.58}$$

where L is the length of the swimmer and t_0 is a phase shift that keeps the head at the same phase as the burrower.

3.1.3 Simulation Parameters/Nondimensionalization

The length of the swimmer is 1.2 mm, the period of the oscillation is 0.5 s, and the phase velocity of the bending motion is 4 mm/s. We nondimensionalize the equations using a characteristic length scale of $L = 1$ mm, a time scale of 1 s, and a velocity scale of $U = 1$ mm/s. The viscoelastic stresses are scaled by $\eta_p U/L$. As in the previous sections, we set the ratio of the polymer viscosity to the fluid viscosity to be $\xi = \eta_p/\eta_s = 0.5$. In the viscoelastic fluid we use the Deborah number, De, to characterize the ratio of the relaxation time of the polymers to the time scale of the flow. We note that we use the characteristic time scale of 1 s to define the De rather than the period of swimmer.

For the bending stiffness we choose a characteristic bending stiffness of nematodes: $k_b = 2 \times 10^{-15}$ Nm^2 [108], which when nondimensionalized becomes $k_b = 2$. For the stretching stiffness we choose a nondimensional value of $k_s = 2500$. We note that with this value of the bending stiffness, the actual curvature may be significantly different from the target curvature. We return to examining the effect of changing the bending stiffness in Sect. 3.4.

The computations are performed in a 2 mm by 1 mm doubly periodic domain, with the worm initially aligned in the x-direction. The domain is discretized using a regular $N_x = 256$ by $N_y = 128$ grid, and the points on the swimmer are spaced so that $\Delta s \approx \Delta x$. We use the implicit-time stepping scheme with a time step of $\Delta t = 10^{-3}$ and a stopping tolerance of 5×10^{-5} for the nonlinear solver.

The viscoelastic fluid solver used here is the same solver described in Sect. 1.6. The dimensionless polymer stress diffusion coefficient is fixed at $\alpha = 0.01$. The length scale associated with polymer stress diffusion over a time step is smaller than the grid spacing ($\sqrt{\Delta t \alpha} \approx 0.0032 \lesssim \Delta x = 1/2^7 \approx 0.0078$).

3.2 Swimming Speed

We simulate the swimming of both the burrower and the kicker until time $t = \max(10, 10\,\mathrm{De})$ which allows sufficient time to establish a periodic motion independent of the initial transients. We measure their steady-state swimming speed by computing the displacement of the center of mass over the last period of the simulation. In Fig. 10.7 we show the swimming speeds for both the burrower and the kicker as a function of De, scaled by the swimming speeds for $\mathrm{De} = 0$. While the burrower always swims slower in a viscoelastic fluid, the kicker swims up to 25% faster in a viscoelastic fluid. The swimming speed changes non-monotonically as a function of De for both swimmers, with a local maximum a little beyond $\mathrm{De} = 1$. The relative swimming speed for the kicker is consistent with the result reported in [83].

Figure 10.8 shows contours of the elastic energy, $\mathrm{Tr}(\boldsymbol{\tau}_p)$, for both the burrower and kicker at the ending time of the simulation for $\mathrm{De} = 0.1, 0.5, 1.0, 2.0$. These plots demonstrate the significant differences between both the size and location of the elastic stresses for the burrower and the kicker. Generally, the elastic stresses are larger for the kicker, and much more concentrated at the tail.

In Sect. 1, we showed that large, highly concentrated elastic stresses were produced around extensional points. It is not a priori clear whether the swimmer problem is prone to exhibit the flow characteristics associated with high-Weissenberg number problems. In Fig. 10.9 we show the extensional points, centers, and streamlines at four representative times in a steady frame translating at the worm's average swimming speed for the burrower at $\mathrm{De} = 0$. Periodically, a new extensional point and center are generated at the head, and shortly after this, an extensional point and center coalesce at the tail. In between these events, there are a pair of extensional points and a pair of centers traveling from the head to the tail. In the plots shown, the strengths of the extensional points range from 0.9 to 2.8, which is in the range subject to high-Weissenberg number phenomena.

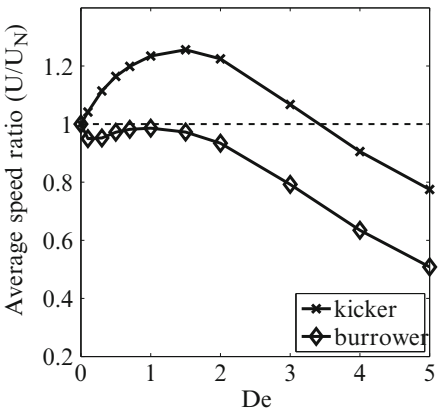

Fig. 10.7 Scaled swimming speed of both the burrower and the kicker as a function of De. The swimming speed is scaled by the speed at $\mathrm{De} = 0$

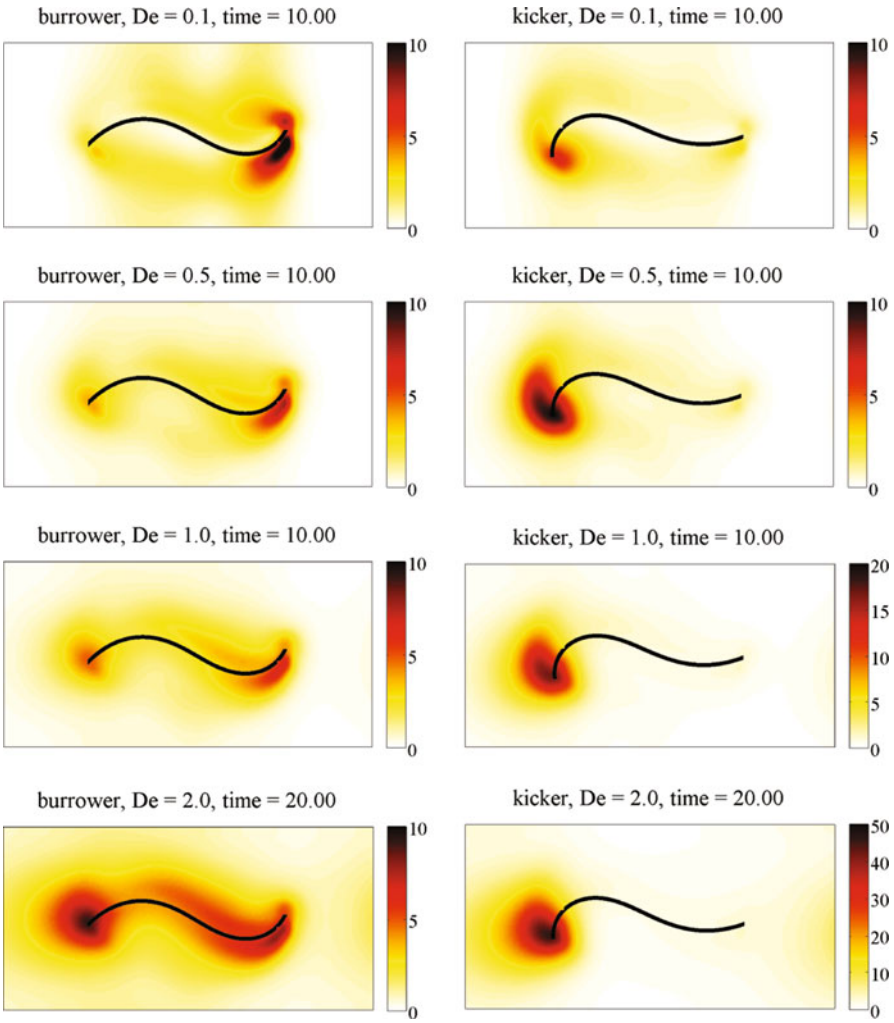

Fig. 10.8 Contours of the elastic energy at the end of the simulation for both the burrower and kicker for different De. The head is on the right

However, examining streamlines provides only limited information. It remains an open question whether these moving extensional points are related to high-Weissenberg number phenomena. See Chap. 7 for a similar discussion, and see [106] for a more detailed study of the effects of fluid elasticity, body mechanics, and stroke kinematics on swimming speed for this problem.

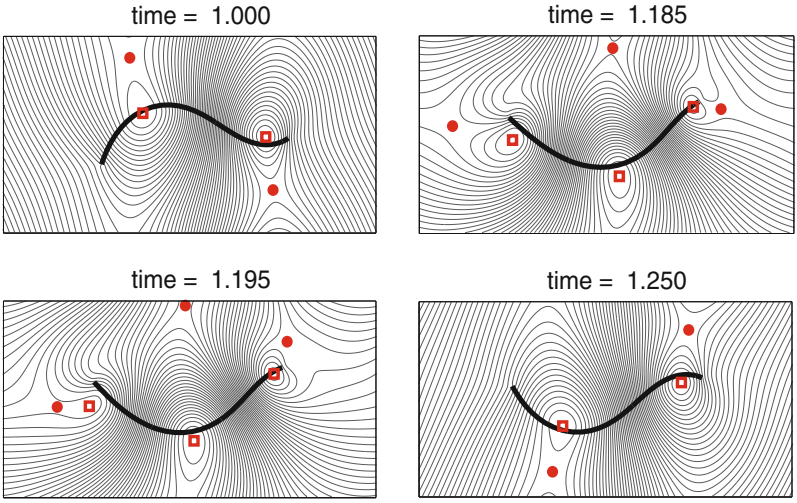

Fig. 10.9 Streamlines in the frame translating with the average swimming speed of the worm for the burrower in a Newtonian fluid (De = 0). The extensional points are marked with *dots* and centers are marked with *squares*. Four representative times are shown including the creation of a new extensional point at the head and the disappearance of an extensional point at the tail. The head is on the right

3.3 Time and Space Resolution

The results of the previous section show that the swimmer problem involves strong extensional points and highly concentrated elastic stresses that are associated with high-Weissenberg number problems. Here we demonstrate the importance of long time scales and fine spatial scales for the swimmer problem.

Figure 10.10 (left) shows the polymer elastic energy, measured as the trace of the polymer stress tensor, for three different De up to time 20. These data demonstrate that for high De the elastic stresses evolve very slowly. For De = 5 the elastic energy is still growing substantially at time $t = 20$, while the energy is near-steady state for the lower values of De. The elastic energy for De = 5 up to time 50 is shown in the inset. The effect of this slow evolution of stress on the swimming speed is demonstrated in Fig. 10.10 (right) which shows the swimming speed (averaged over the previous period) as a function of time for different De. For De = 1, the swimming speed rapidly approaches a steady state, but for high De the swimming speed increases on a time scale on the order of De, but on long time scales it slowly decays to a much smaller value. These data demonstrate that understanding the effects of elastic stresses in systems with large De (or large Wi) requires performing long-time simulations.

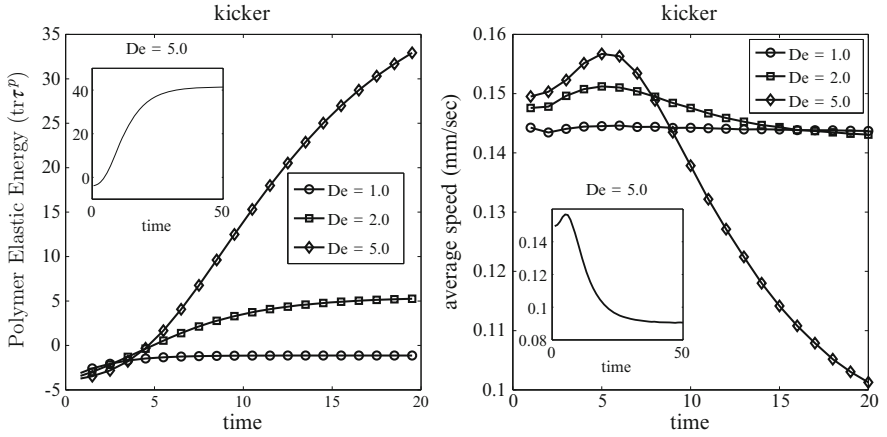

Fig. 10.10 (*left*) Time course of the elastic energy in the kicker simulations up to $t = 20$ for De $= 1, 2, 5$. The inset shows the data for De $= 5$ up to time 50. (*right*) Swimming speed averaged over previous period as a function of time for the same parameters

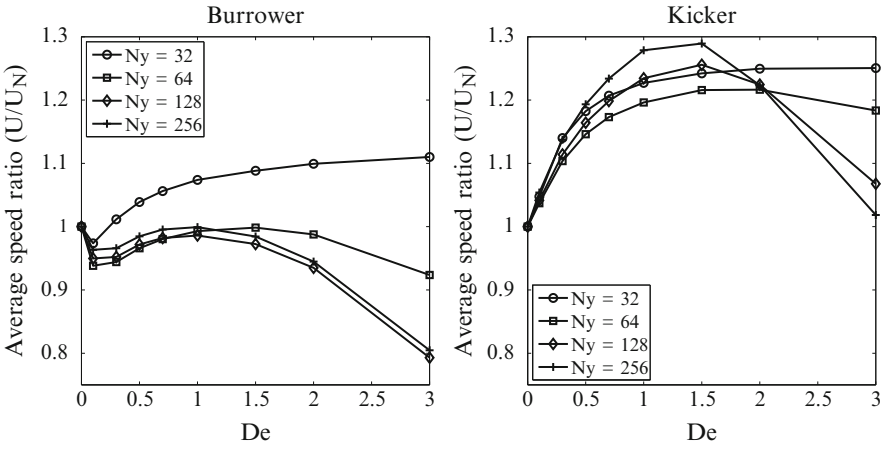

Fig. 10.11 Swimming speed for burrower and kicker for $Ny = 32, 64, 128, 256$ and varying De

As Fig. 10.8 shows, highly localized stresses develop around the swimmer for large De. Here we demonstrate the importance of adequately resolving these large stress gradients. Figure 10.11 shows the swimming speeds as a function of De for both the burrower and the kicker as the grid is refined. The results from the two coarsest meshes and the two finest meshes are notably different. In fact, on the coarsest mesh, the simulations predict the wrong dependence of swimming speed on De.

3.4 *Effect of Increasing Bending Stiffness*

In these simulations, the active forces of the worm are driven by changes in the desired curvature, κ_0. In the limit that the bending stiffness goes to infinity, the curvature of the worm approaches the desired curvature, but for any finite value of the bending stiffness, the curvature of the worm and the desired curvature are different. In the simulations presented in the previous sections, the bending stiffness was chosen based on the measured bending stiffness of nematodes. In this section we examine the effect of changing the bending stiffness on the swimming speed.

Figure 10.12 shows the swimming speed at De $= 0.5$ for eight different bending stiffness values from four times lower than that used in the previous sections to 50 times higher. As the bending stiffness increases, the curvature of the simulated worm more closely follows the prescribed curvature, and the swimming speed approaches a constant. It is notable that the swimming speed for $k_b = 100$ is about 80 % larger than for $k_b = 2$, which was the value used in the previous sections.

We repeat the calculations of the steady-state swimming speed from Sect. 3.2 for a range of De for a larger bending stiffness. Figure 10.13 shows the relative swimming speeds for both the burrower and the kicker for De $= 0 - 5$ for both $k_b = 2$ and $k_b = 40$. For both swimmers, the results are very different between the two stiffness values. In contrast to the softer worm, with the larger bending stiffness, the burrower's swimming speed decreases monotonically with De, and the kicker's swimming speed is always slower than in a Newtonian fluid.

For the softer bending stiffness, there is a significant difference between the prescribed curvature and the realized curvature. The target curvature represents the equilibrium shape that would be achieved in the absence of resistance from the surrounding medium. The product $k_b \kappa_0$ can be related to the internal torque density resulting from muscular contraction. In the soft regime, the realized shape

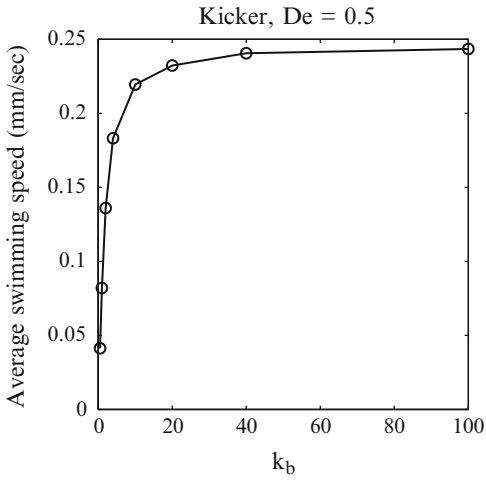

Fig. 10.12 Average swimming speed at De $= 0.5$ for different values of the bending stiffness

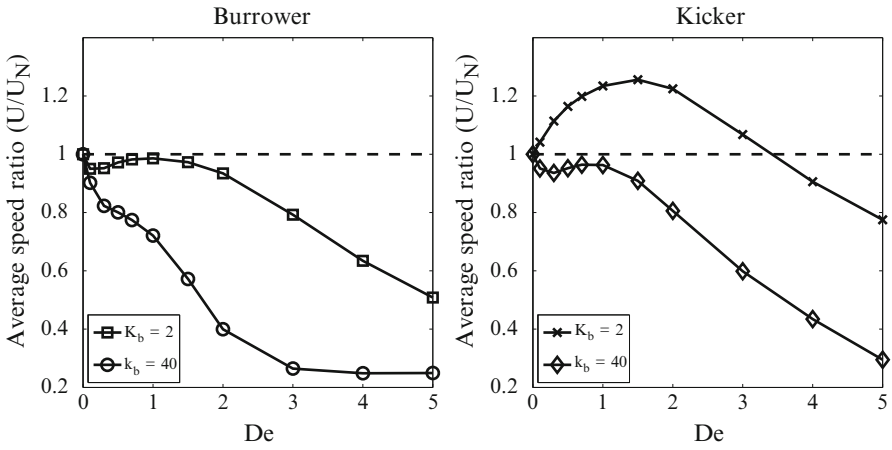

Fig. 10.13 Scaled swimming speed of both the burrower and the kicker as a function of De for two different bending stiffnesses. The swimming speed is scaled by the speed at De = 0. The data for $k_b = 2$ is the same as that from Fig. 10.7

results from the complex interaction between the fluid, passive body mechanics, and actively generated torques. For large elastic stresses, the fluid offers significant resistance to deformation, and the bending forces must be stiff enough to maintain the prescribed shape if the intention is to simulate that prescribed shape.

These results demonstrate that care must be exercised when using penalty methods to ensure that the forces are sufficiently stiff so that the choice of numerical parameters does not affect the predicted results. Even with the lower bending stiffness, the equations are extremely numerically stiff, and explicit-time methods would be very inefficient (see next section). The implicit-time method is necessary to be able to simulate on long time scales, with a fine grid, and with sufficiently large stiffness to capture the correct dynamics of the prescribed shape swimmers.

3.5 Efficiency of the Implicit-Time Method

For given stiffness coefficients and grid spacing, we let Δt_{exp} represent the largest time step that gives a stable simulation in the explicit-time method. We identify Δt_{exp} using a bisection algorithm. The worm simulation is run until time 0.1, and considered unstable if the elastic energy rises above a prescribed threshold. In Table 10.1, we report Δt_{exp} for two different bending stiffnesses and four different grid resolutions, for De = 0.5. The table shows that when the grid is refined by a factor of 2, the time step shrinks by a factor of about 8, which is as expected for bending forces at low Reynolds number [109]. On the finest grid, the reported time step restriction is an estimate from the coarse grid values.

Each time step of the implicit method requires repeated solution of the Stokes equations. Each evaluation of the objective function in the Newton's method and

Table 10.1 Maximum stable time step for the explicit method. The value reported on the finest grid is estimated from the coarser grids

Δx	$k_b = 2$	$k_b = 20$
2^{-5}	2.76e-4	2.65e-5
2^{-6}	3.28e-5	3.14e-6
2^{-7}	4.01e-6	3.86e-7
2^{-8} (est)	4.81e-7	3.49e-8

Table 10.2 Average number of Stokes solves over 1,000 time steps of the implicit code with $\Delta t = 10^{-3}$

	tol = 5×10^{-5}		tol = 1×10^{-5}	
Δx	$k_b = 2$	$k_b = 20$	$k_b = 2$	$k_b = 20$
2^{-5}	17.82	21.70	22.21	32.34
2^{-6}	20.99	33.45	33.10	43.02
2^{-7}	24.23	44.11	39.48	67.44
2^{-8}	39.58	–	64.41	102.93

each application of the Jacobian require solving the Stokes equations. In Table 10.2 we report the number of times the Stokes equations are solved per time step for 1,000 time steps of the implicit method with $\Delta t = 10^{-3}$ (two periods of the swimming motion). We report these values for two different stiffnesses, four different grid resolutions, and two different stopping tolerances for the Newton's method.

Note that on the finest grid with the largest stiffness, the solver failed with the larger tolerance during the simulation. With a smaller tolerance, no failure occurred. A more sophisticated nonlinear solver, e.g., including line searching or trust region methods, would likely be able to compute with the larger tolerance [110, 111]. This failure illustrates one of the many added difficulties of working with implicit-time methods in place explicit-time methods.

To estimate the efficiency of the implicit method we compare the number of times the Stokes equations are solved per time unit with the explicit-time method. We measure the efficiency as the ratio of Stokes solves in the explicit method to the implicit method per unit time. This efficiency estimate ignores the extra work involved in the GMRES iteration in the implicit method. However, it also ignores the extra updates of the stress equation in the explicit method, and the fact that one is not likely to run the explicit method exactly at the stability limit. With this in mind, one may loosely interpret the efficiency measurement as the expected speedup gained by using the implicit method in place of the explicit method.

The efficiency is reported in Table 10.3 for two different bending stiffnesses, four different grid resolutions, and two different stopping tolerances for the Newton's method. Only on the coarsest grid is the explicit method ever significantly more efficient than the implicit method. However, as noted previously, the solutions on the two coarsest grids have very large errors. The coarsest grid spacing which we consider usable is $\Delta x = 2^{-7}$. At this resolution and above, the implicit solver significantly outperforms the explicit method.

As the grid is refined, the problem gets significantly stiffer, but the work involved in the nonlinear solver grows only mildly. The number of Stokes solves required per

Table 10.3 Efficiency of the implicit method measured as the ratio of the number of times the Stokes equations are solved per time unit in the explicit method to the implicit method

	tol = 5×10^{-5}		tol = 1×10^{-5}	
Δx	$k_b = 2$	$k_b = 20$	$k_b = 2$	$k_b = 20$
2^{-5}	0.20	1.15	0.16	1.16
2^{-6}	1.45	6.84	0.92	7.39
2^{-7}	10.28	58.68	6.31	38.38
2^{-8}	52.42	–	32.27	278.23

Numbers greater than one indicate that the implicit method is more efficient than the explicit method

time step increases by only a factor between about 2 and 3 as the grid is refined by a factor of 8, while in the explicit method, the number of Stokes solves would increase by a factor of over 500. Similarly, when the bending stiffness is increased by a factor of 10, the work increases by less than a factor of 2 in the implicit method and by about a factor of 10 in the explicit method. These results illustrate that implicit methods far outperform explicit methods for very stiff problems on fine grids.

4 Conclusions

Strongly elastic flows at low Reynolds numbers share some characteristics of high Reynolds number flows, namely, regions of highly localized stress and sharp gradients in the velocity. One does not expect to be able to use the same algorithms for low and high Reynolds number, and similarly, special care must be used to simulate flows at high-Weissenberg numbers. We have highlighted some recent analytical work on low Reynolds number viscoelastic fluids which has led to new algorithms for successfully simulating flows at high-Weissenberg numbers. The unbounded stress growth exhibited by the Oldroyd-B model is not present in many other models with nonlinear relaxation rates. However, these models also exhibit large stress gradients at high-Weissenberg numbers, and the algorithms developed for Oldroyd-B are often still necessary to mitigate these numerical challenges. In particular, we have demonstrated that the polymer stress diffusion modification regularizes the Oldroyd-B model so that accurate, smooth, and bounded solutions are obtained in periodic extensional flow.

It is not obvious how the standard benchmark problems for the high-Weissenberg number phenomenon such as flow in a contracting channel or steady elongational flow are directly related to biological problems. We used the classical problem of an undulatory swimmer at low Reynolds number to demonstrate that when elastic stresses are under-resolved, the predicted relative swimming speed is qualitatively different from simulations with resolved stresses. We note that the immersed boundary method is first-order accurate near structures, and this low accuracy may make the resolution of stresses more difficult.

The popularity of the immersed boundary method comes from its simplicity and robustness which make it more attractive than using a significantly more complicated high-order method, in particular for problems in biology where high-order accuracy is often not paramount. Any algorithm for solving the forced equations of motion for the fluid on a structured grid can be used in an immersed boundary simulation. The methodology extends to complex fluids without modification. The method is not without its drawbacks; notably, its low accuracy and the severe stability restriction imposed by explicit-time stepping schemes in problems involving stiff elastic structures. Accurately resolving the stress in strongly elastic flows requires high grid resolution and long time integration of the equations. The implicit-time method presented in this chapter makes it possible to perform simulations on long time scales, with a fine grid, and with large elastic stiffness.

Acknowledgment The authors would like to thank Lisa Fauci and Michael Shelley for interesting discussions and suggestions on this work. We also acknowledge Paulo Arratia and Xiaoning Shen for allowing us to use their data. This work was supported in part by NSF grants DMS-1160438 and DMS-1226386 to RDG.

References

1. J. Oldroyd, Proceedings of the Royal Society of London. Series A. Math. Phys. Sci. **200**(1063), 523 (1950)
2. R.B. Bird, O. Hassager, R. Armstrong, C. Curtiss, *Dynamics of Polymeric Liquids, Vol. 2: Kinetic Theory* (John Wiley and Sons, London 1980)
3. M. Mendelson, P.W. Yeh, R. Brown, R. Armstrong, J. Non-Newtonian Fluid Mech. **10**(1), 31 (1982)
4. R. Brown, R. Armstrong, A. Beris, P. Yeh, Comput. Meth. Appl. Mech. Eng. **58**(2), 201 (1986)
5. R.G. Owens, T.N. Phillips, *Computational Rheology*, vol 2 (World Scientific, Singapore 2002)
6. G.H. McKinley, R.C. Armstrong, R.A. Brown, Philos. Trans. R. Soc. Lond. Ser. A: Phys. Eng. Sci. **344**(1671), 265 (1993)
7. M. Chilcott, J. Rallison, J. Non-Newtonian Fluid Mech. **29**, 381 (1988)
8. A.W. Liu, D.E. Bornside, R.C. Armstrong, R.A. Brown, J. Non-Newtonian Fluid Mech. **77**(3), 153 (1998)
9. M. Alves, F. Pinho, P. Oliveira, J. Non-Newtonian Fluid Mech. **97**(2), 207 (2001)
10. M.A. Hulsen, R. Fattal, R. Kupferman, J. Non-Newtonian Fluid Mech. **127**(1), 27 (2005)
11. H.S. Dou, N. Phan-Thien, Chem. Eng. Sci. **62**(15), 3909 (2007)
12. S. Claus, T. Phillips, J. Non-Newtonian Fluid Mech. **200**, 131 (2013)
13. H. Nguyen, D. Boger, J. Non-Newtonian Fluid Mech. **5**, 353 (1979)
14. D. Boger, Annual Rev. Fluid Mech. **19**(1), 157 (1987)
15. G.H. McKinley, W.P. Raiford, R.A. Brown, R.C. Armstrong, J. Fluid Mech. **223**, 411 (1991)
16. S.C. Xue, N. Phan-Thien, R. Tanner, J. Non-Newtonian Fluid Mech. **74**(1), 195 (1998)
17. M.A. Alves, P.J. Oliveira, F.T. Pinho, J. Non-Newtonian Fluid Mech. **122**(1), 117 (2004)
18. M. Webster, H. Tamaddon-Jahromi, M. Aboubacar, J. Non-Newtonian Fluid Mech. **118**(2), 83 (2004)
19. D. Trebotich, P. Colella, G. Miller, J. Comput. Phys. **205**(1), 315 (2005)
20. D.D. Joseph, *Fluid Dynamics of Viscoelastic Liquids*, vol. 84 (Springer, New York, 1990)
21. M. Renardy, *Mathematical Analysis of Viscoelastic Flows*, vol 73 (SIAM, Philadelphia, 2000)

22. M.J. Crochet, Rubber Chem. Technol. **62**(3), 426 (1989)
23. F. PT Baaijens, J. Non-Newtonian Fluid Mech. **79**(2), 361 (1998)
24. R. Keunings, in *Proceedings of the XIIIth International Congress on Rheology*, vol 1 (British Soc. Rheol, 2000), vol 1, pp. 7–14
25. R. Keunings, Rheology Rev. pp. 167–196 (2003)
26. R. Keunings, Rheology Rev. pp. 67–98 (2004)
27. A.W. El-Kareh, L.G. Leal, J. Non-Newton. Fluid Mech. **33**, 257 (1989)
28. J.T. Beale, T. Kato, A. Majda, Comm. Math. Phys. **94**(1), 61 (1984)
29. J.Y. Chemin, N. Masmoudi, SIAM J. Math. Anal. **33**(1), 84 (2001)
30. R. Kupferman, C. Mangoubi, E.S. Titi, arXiv preprint arXiv:0709.1455 (2007)
31. F.H. Lin, C. Liu, P. Zhang, Comm. Pure Appl. Math. **58**(11), 1437 (2005)
32. T.C. Sideris, B. Thomases, Comm. Pure Appl. Math. **58**(6), 750 (2005)
33. P. Constantin, Remarks on complex fluid models. Mathematical aspects of fluid mechanics. London Mathematical Society Lecture Note Series (No. 402) (Cambridge University Press, Cambridge, 2012), pp. 70–87
34. P. Constantin, M. Kliegl, ARMA **206**(1), 725 (2012)
35. R. Larson, *The Structure and Rheology of Complex Fluids* (Oxford University Press, Oxford, 1998)
36. R. Sureshkumar, A.N. Beris, J. Non-Newtonian Fluid Mech. **60**, 53 (1995)
37. B. Thomases, J. Non-Newt. Fluid Mech. **166**, 1221 (2011)
38. C.D. Dimitropoulos, R. Sureshkumar, A.N. Beris, J. Non-Newt. Fluid Mech. **79**, 433 (1998)
39. K.D. Housiadas, A.N. Beris, Phys. Fluids **15**, 2369 (2003)
40. B. Thomases, M. Shelley, Phys. Fluids **19**, 103103 (2007)
41. R.G. Larson, *The structure and rheology of complex fluids*, vol. 2 (Oxford university press, New York, 1999)
42. J.M. Rallison, E.J. Hinch, J. Non-Newt Fluid Mech. **29**, 37 (1988)
43. M. Renardy, J. Non-Newt. Fluid Mech. **138**, 204 (2006)
44. A.N. Brooks, T.J. Hughes, Comput. Meth. Appl. Mech. Eng. **32**(1), 199 (1982)
45. M.D. Gunzburger, *Finite Element Methods for Viscous Incompressible Flows: A Guide to Theory, Practice, and Algorithms* (Elsevier, Amsterdam, 2012)
46. J. Marchal, M. Crochet, J. Non-Newtonian Fluid Mech. **26**(1), 77 (1987)
47. M. Fortin, A. Fortin, J. Non-Newtonian Fluid Mech. **32**(3), 295 (1989)
48. H.P. Baaijens, G.W. Peters, F.P. Baaijens, H.E. Meijer, J. Rheol. (1978–present) **39**(6), 1243 (1995)
49. B. Cockburn, G.E. Karniadakis, C.W. Shu, *The Development of Discontinuous Galerkin Methods* (Springer, NewYork 2000)
50. D. Rajagopalan, R.C. Armstrong, R.A. Brown, J. Non-Newtonian Fluid Mech. **36**, 159 (1990)
51. F. Baaijens, J. Non-Newtonian Fluid Mech. **75**(2), 119 (1998)
52. H. Matallah, P. Townsend, M. Webster, J. Non-Newtonian Fluid Mech. **75**(2), 139 (1998)
53. R.C. King, M.R. Apelian, R.C. Armstrong, R.A. Brown, J. Non-Newtonian Fluid Mech. **29**, 147 (1988)
54. R. Guénette, M. Fortin, J. Non-Newtonian Fluid Mech. **60**(1), 27 (1995)
55. I. Keshtiban, F. Belblidia, M. Webster, J. Non-Newtonian Fluid Mech. **126**(2), 123 (2005)
56. C.W. Shu, S. Osher, J. Comput. Phys. **77**(2), 439 (1988)
57. H.D. Ceniceros, J.E. Fisher, J. Non-Newtonian Fluid Mech. **171**, 31 (2012)
58. R. Fattal, R. Kupferman, J. Non-Newtonian Fluid Mech. **123**(2), 281 (2004)
59. R. Fattal, R. Kupferman, J. Non-Newtonian Fluid Mech. **126**(1), 23 (2005).
60. A. Afonso, M. Alves, F. Pinho, P. Oliveira, in *3rd Annual European Rheology Conference* (2006), pp. 27–29
61. A. Afonso, P. Oliveira, F. Pinho, M. Alves, J. Non-Newtonian Fluid Mech. **157**(1), 55 (2009)
62. H. Damanik, J. Hron, A. Ouazzi, S. Turek, J. Non-Newtonian Fluid Mech. **165**(19), 1105 (2010)
63. N. Balci, B.Thomases, M. Renardy, C. Doering, J. Non-Newt. Fluid Mech. (2011)
64. M. Laso, H. Öttinger, J. Non-Newtonian Fluid Mech. **47**, 1 (1993)

65. A. Lozinski, C. Chauvière, J. Comput. Phys. **189**(2), 607 (2003)
66. H. Giesekus, J. Non-Newtonian Fluid Mech. **11**(1), 69 (1982)
67. N.P. Thien, R.I. Tanner, J. Non-Newtonian Fluid Mech. **2**(4), 353 (1977)
68. H.R. Warner Jr, Ind. Eng. Chem. Fundamentals **11**(3), 379 (1972)
69. M. Chilcott, J. Rallison, J. Non-Newtonian Fluid Mech. **29**, 381 (1988)
70. G. Lielens, P. Halin, I. Jaumain, R. Keunings, V. Legat, J. Non-Newtonian Fluid Mech. **76**(1), 249 (1998)
71. G. Lielens, R. Keunings, V. Legat, J. Non-Newtonian Fluid Mech. **87**(2), 179 (1999)
72. Q. Du, C. Liu, P. Yu, Multiscale Modeling and Simulation **4**(3), 709 (2005)
73. A. Peterlin, J. Polymer Sci. Part B: Polymer Lett. **4**(4), 287 (1966)
74. C. Sulem, P.L. Sulem, H. Frisch, J. Comput. Phys. **50**(1), 138 (1983)
75. R. Krasny, J. Fluid Mech. **167**, 65 (1986)
76. M. Shelley, J. Fluid Mech. **244**(1), 493 (1992)
77. R. Peyret, *Spectral Methods for Incompressible Viscous Flow*, vol 148 (Springer, Newyork, 2002)
78. T.Y. Hou, R. Li, J. Comput. Phys. **226**(1), 379 (2007)
79. C.S. Peskin, Acta Numer. **11**(-1), 479 (2002)
80. C.S. Peskin, J. Comput. Phys. **25**(3), 220 (1977)
81. J. Chrispell, L. Fauci, Math. Modelling of Nat. Phenomena **6**(05), 67 (2011)
82. J.C. Chrispell, L.J. Fauci, M. Shelley, Phys. Fluids **25**(1), 013103 (2013)
83. J. Teran, L. Fauci, M. Shelley, Phys. Rev. Lett. **104**, 038101 (2010).
84. J. Du, J.P. Keener, R.D. Guy, A.L. Fogelson, Phys. Rev. E **85**, 036304 (2012)
85. J.M. Stockie, B.R. Wetton, J. Comput. Phys. **154**(1), 41 (1999)
86. E.P. Newren, A.L. Fogelson, R.D. Guy, R.M. Kirby, J. Comput. Phys. **222**(2), 702 (2007)
87. Z. Gong, H. Huang, C. Lu, Commun. Comput. Phys. **3**, 704 (2008)
88. C. Tu, C.S. Peskin, SIAM J. Sci. Stat. Comput. **13**(6), 1361 (1992).
89. A.A. Mayo, C.S. Peskin, in *Fluid Dynamics in Biology (Seattle, WA, 1991)*, *Contemp. Math.*, vol. 141 (Amer. Math. Soc., Providence, RI, 1993), pp. 261–277
90. L. Lee, R.J. LeVeque, SIAM J. Sci. Comput. **25**(3), 832 (2003).
91. E.P. Newren, A.L. Fogelson, R.D. Guy, R.M. Kirby, Comput. Methods Appl. Mech. Engrg. **197**(25–28), 2290 (2008).
92. T.Y. Hou, Z. Shi, J. Comput. Phys. **227**(20), 8968 (2008)
93. T.Y. Hou, Z. Shi, J. Comput. Phys. **227**(21), 9138 (2008)
94. H.D. Ceniceros, J.E. Fisher, A.M. Roma, J. Comput. Phys. **228**(19), 7137 (2009).
95. Y. Mori, C.S. Peskin, Comput. Methods Appl. Mech. Engrg. **197**(25–28), 2049 (2008).
96. D. Le, J. White, J. Peraire, K. Lim, B. Khoo, J. Comput. Phys. **228**(22), 8427 (2009).
97. H.D. Ceniceros, J.E. Fisher, J. Comput. Phys. **230**(12), 5133 (2011).
98. E. Lauga, T. Powers, Rep. Prog. Phys. **72**, 096601 (2009)
99. T. Normand, E. Lauga, Phys. Rev. E **78**, 061907 (2008)
100. E. Lauga, Phys. Fluids **19**, 083104 (2007)
101. H.C. Fu, T.R. Powers, C.W. Wolgemuth, Phys. Rev. Lett. **99**, 258101 (2007)
102. E. Lauga, Europhys. Lett. **86**, 64001 (2009)
103. A.M. Leshansky, Phys. Rev. E **80**, 051911 (2009)
104. H. Fu, V. Shenoy, T. Powers, Europhys. Lett. **91**, 24002 (2010)
105. S.E. Spagnolie, B. Liu, T.R. Powers, Phys. Rev. Lett. **111**(6), 068101 (2013)
106. B. Thomases, R.D. Guy, Phys. Rev. Lett. **113**(9), 098102 (2014)
107. X.N. Shen, P.E. Arratia, Phys. Rev. Lett. **106**(20), 208101 (2011)
108. M. Backholm, W.S. Ryu, K. Dalnoki-Veress, Proc. Natl. Acad. Sci. **110**(12), 4528 (2013)
109. R.D. Guy, B. Philip, Commun. Comput. Phys. **12**, 378 (2012)
110. C. Kelley, *Iterative Methods for Linear and Nonlinear Equations* (SIAM, Philadelphia, 1995)
111. J. Nocedal, S.J. Wright, *Numerical Optimization* (Springer, New York, 1999)