A Numerical Exploration of Viscoelastic Fluids

By

JEFFREY NICHOLS
DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Applied Mathematics

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

Robert Guy, Chair

Becca Thomases, Co-Chair

Harishankar Manikantan

Committee in Charge

2024

Contents

Abstract

Dissolving stretchable polymers in a Newtonian fluid can give the fluid a new property called viscoelasticity. Viscoelastic fluids exhibit fascinating behaviors due to the feedback force created by stretched polymers trying to return to their original length. These fluids come with many mathematical challenges, including open questions about the dynamics of flows as well as numerical difficulties. The two parts of this dissertation each relate to one of those overarching categories.

The first part of this dissertation is a broad exploration of a doubly periodic parallel shear flow known as Kolmogorov flow. This flow was chosen in part because it is one of the simplest flows known to exhibit coherent structures known as both ‘narwhals’ and ‘arrowheads,’ as well as a chaotic phenomenon known as elastic turbulence, both of which are active areas of study. Elastic turbulence shares many features of Newtonian turbulence, but is a distinct and poorly understood phenomenon. In particular, finding a dynamical origin or clear “route to chaos” has not been previously achieved for viscoelastic fluids.

We catalog a large variety of flow states, including elastic turbulence, that can be reached with varying levels of elasticity in the fluid and various domains. These chapters contain many intriguing results, the most significant of which is the first evidence of a route to chaos in viscoelastic fluids. An initial instability in Kolmogorov flow generates traveling wave solutions in the form of coherent structures, which in turn lose stability as elasticity is increased. Oscillations in the traveling waves emerge, which proceed through a period doubling cascade and eventually become chaotic.

The second part of this dissertation relates to numerical methods and will introduce what we call the Double Immersed Boundary (DIB) method. This is a modification of the existing Immersed Boundary (IB) method, which is very useful for simulating fluid-structure interactions, but cannot achieve convergence of the velocity gradients near boundaries, which are of particular importance to viscoelastic fluid simulations. The DIB method remedies this problem for the special case where solutions are only required on one side of the boundary. Naturally, there are tradeoffs, and the DIB method leads to challenging conditioning and stability issues that can be addressed in a variety of ways.

Acknowledgments

Thank you to my first advisor, Becca Thomases, for sticking with me through a long and unusual PhD process. From our meetings in your garage during the pandemic to the countless Zoom calls across the country, your belief in me has been unwavering for a very long time. Beyond the invaluable mathematical mentorship you provided, you have a unique ability to transform a student who enters a meeting too discouraged to even discuss their results into a student who is not only ready, but excited to press on.

Thank you to my co-advisor, Bob Guy, for stepping in and providing essential support when Becca moved away. Your patience in explaining concepts, no matter how many times, and your willingness to answer any question, no matter how small, have been fundamental to my growth as a mathematician.

Thank you to my final committee member, Harishankar Manikantan, and the rest of our fluids research group. Our meetings were not only academically enriching, but also enjoyable, supportive, and even full of laughter, an ingredient which I, for one, rely upon.

Thank you to all of the other math teachers I've had, particularly the faculty at the Colorado School of Mines. You took a young person with doubts about his future in engineering and showed him the wonders and joys that could be found in both mathematics and in teaching. Your work as math instructors continues to inspire me in my own teaching career.

Thank you to my family, Cheryl, Kelly, and Steven, for supporting my decision to pursue this degree from the beginning and for the boundless love and encouragement along the way.

A special thank you to my partner, Dasha, for so many things. You've been my source of laughter on the darkest days, and you've seen this process break me in ways nothing else has, only to help put me back together, time and again.

Thank you to my friends in grad school and beyond: Eli Moore, Avishai Halev, Jeonghoon Kim, Raaghav Ramani, Kayden Mimmack, James Hughes, Michael Ragone, Brian Knight, Nathan Yoshino, Tait Weicht, Vincent Lovero, Kyle Chickering, Alex McDonough, Julia Schedler, Ryan Berg, Kara Davis, the UC Davis Triathlon Team, the Davis cycling and acroyoga communities, and many more. I could not have asked for, nor would I have reached this point without, a more kind, generous, and encouraging group of friends and colleagues.

To properly recognize the support and contributions of everyone who has helped me on this journey would require an acknowledgment section longer than the thesis itself. Grad school has been an extraordinarily challenging process that I could never have completed alone. From the bottom of my heart, thank you all. This thesis is as much yours as it is mine.

CHAPTER 1

Introduction

1.1. Viscoelasticity

What most people think of as fluids – water, oil, etc. – are called Newtonian fluids. These fluids exhibit viscous behavior, meaning that their resistance to being deformed is akin to internal friction and that the resistance becomes stronger if the deformation happens more quickly. Put more formally, the stress in a Newtonian fluid changes linearly with the *rate* of deformation, known as the strain rate. Another characteristic of Newtonian fluids is a lack of "memory." That is, the material has no ability to return to a previous shape after it has been deformed. As a fun example, note that if you find yourself battling to deform a Newtonian fluid that is putting up a significant fight (e.g. squeezing honey out of its container, which forces it to deform), you can make things easier by slowing down. The honey can be more easily extracted if you increase your patience and decrease the strain rate.

On the other hand, solid materials exhibit elastic behavior, meaning that their resistance to deformation is a spring-like force which becomes stronger as the amount of deformation increases, regardless of the rate of deformation. More formally, the stress in a (linear) elastic material changes linearly with the strain. In contrast to viscous materials, elastic materials have an indefinite memory, meaning that when the force that caused a deformation is removed, the material returns to its original shape. So, unfortunately, if you are battling with elastic materials (e.g. compressing too many clothes into a suitcase that won't quite close), slowing down the deformation is no longer a helpful strategy. You must either push harder or pack fewer clothes, because the only way to reduce an elastic force is to apply a smaller deformation.

Viscoelasticity, as the name suggests, is a combination of viscosity and elasticity. This property can arise when long stretchable polymers are dissolved in a Newtonian solvent. The polymers become stretched as the fluid deforms and, like microscopic springs, endeavor to return to their relaxed, unstretched state. In doing so they push back on the solvent, creating a feedback force, and giving the material some "memory" of its former shape. Because polymers relax over time, however, this memory is only temporary. Due to the feedback force, the dynamics of the solvent and the dynamics of the polymers are coupled together and complex, nonlinear behavior of the system results. To give a few examples:

- If a rod is placed vertically in a container of viscoelastic fluid and rotated, the fluid can climb up the rod (Newtonian fluid will do the opposite). The reader is encouraged to try this out with a variety of batters in their kitchen! [2, 26]
- When a thin filament of viscoelastic fluid is stretched out, it can form so-called "beads on a string." This can be done with saliva between the fingers. [9]
- Even when inertia is negligible in a viscoelastic flow, chaotic, turbulent-like dynamics can occur. A more thorough discussion of this phenomenon is given in Ch. 2.
- When polymers are added to turbulent flow through a pipe, such as oil pipelines or fire-fighting hoses, the amount of drag can decrease dramatically, meaning much less power is required to achieve a given flow rate. [67]

An important parameter in viscoelastic flows is the relaxation time, λ, which is the typical time needed for stretched polymers to relax back to their unstretched state. When relaxation is fast relative to the flow timescale, polymer forces are short-lived and the flow is dominated by viscous behavior. On the other hand, when relaxation is slow relative to the flow, elastic forces become more important. The relative importance of viscous and elastic behavior is captured by the Deborah number, a dimensionless number used to describe viscoelastic flows. It is defined as $\mathrm{De} = \lambda/T$, where T is the timescale of the flow.

While comprehending the Deborah number is certainly useful for a general understanding of viscoelastic fluids, we will not be using it in our studies. Instead we use another dimensionless number, the Weissenberg number, denoted Wi. The Weissenberg number can also be thought of as a ratio of timescales, specifically the ratio of polymer relaxation time to the characteristic *deformation* time. Wi captures the degree of nonlinearity in a viscoelastic flow, and can be defined as $\mathrm{Wi} = \lambda\dot{\gamma}$, where $\dot{\gamma}$ is the characteristic strain rate of the flow. Note that the units of strain rate are inverse time, so $\lambda\dot{\gamma}$ does indeed represent a ratio of timescales. Distinguishing between De and Wi can be quite confusing, and while we will not attempt to articulate the difference here, the interested reader may find [14, 46] to be illustrative.

Viscoelastic fluids are common in nature and in synthetic materials, so the study of viscoelastic fluids has many applications. These include industrial processing of polymeric materials [15,50], 3D printing [52,60], biological sciences [37,54], and enhancement of mixing in microfluidic devices [53, 68,69].

1.2. The Stokes-Oldroyd B model

Throughout this work, we will use the Stokes-Oldroyd B model for viscoelastic fluids. This model couples Stokes equations, which govern inertialess incompressible Newtonian flow, with the Oldroyd B model. The Oldroyd B model is the simplest closed continuum viscoelastic model which can be derived from kinetic molecular theory. In particular, the molecular theory treats polymers as "dumbbells" which consist of a pair of beads which experience drag from the solvent and are connected by a Hookean spring [37, 48]. Flow of the solvent can move, stretch, and rotate the dumbbells, which creates a feedback force on the flow as the springs return to their unstretched lengths. The direction of the feedback force thus depends on the orientation of the dumbbells. In order to track both the magnitude and orientation of stretching for a continuum of dissolved dumbbells, we will need a tensor field. Omitting the details, which can be found in [37], the ensemble average dumbbell configuration at each point in space is described by the conformation tensor, which we denote $\mathbf{C}$. In n dimensions, this is a symmetric positive definite $n \times n$ matrix whose eigenvectors give the principle directions of polymer stretch and the associated eigenvalues give the average squared length of stretching in that direction. Thus, the trace of the conformation tensor, tr$\mathbf{C}$, gives the average squared length of polymers and is proportional to the strain energy density (see Section 2.3.4). The fully relaxed state of the conformation tensor corresponds to the identity.

The polymer stress tensor is related to the conformation tensor by

$$\tau_p = \frac{\eta_p}{\lambda}(\mathbf{C} - \mathbf{I}), \tag{1.1}$$

where η_p is the polymer contribution to the fluid viscosity and λ is the characteristic relaxation time of polymers. Note that $(\mathbf{C} - \mathbf{I})$ is a strain tensor and η_p/λ is an elastic modulus. The force due to stretched polymers is given by the divergence of the polymer stress tensor. When Stokes equations are coupled with the Oldryod B model, this polymeric forces appears as an extra force term in the momentum balance.

The governing equations for the velocity, $\mathbf{u}$, pressure, p, and conformation tensor, $\mathbf{C}$, are

$$\eta_s \Delta \mathbf{u} - \nabla p + \mathbf{f}_{ext} + \nabla \cdot \tau_p = 0, \tag{1.2}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{1.3}$$

$$\partial_t \mathbf{C} + \mathbf{u} \cdot \nabla \mathbf{C} - \left(\nabla \mathbf{u} \mathbf{C} + \mathbf{C} \nabla \mathbf{u}^T \right) = -\frac{1}{\lambda}(\mathbf{C} - \mathbf{I}), \tag{1.4}$$

$$\tau_p = \frac{\eta_p}{\lambda}(\mathbf{C} - \mathbf{I}) \tag{1.5}$$

where $\mathbf{f}_{ext}$ is an external force and we use the convention $(\nabla \mathbf{u})_{i,j} = \partial u_i / \partial x_j$.

Eqs. 1.2 and 1.3 are Stokes equations with the added polymer force mentioned previously. These equations correspond to the balance of momentum and incompressibility, respectively. Eq. (1.4) is the evolution equation for the conformation tensor. The first two terms in this equation may be familiar to students of fluid dynamics as the material derivative of $\mathbf{C}$. These two terms advect the polymer stress along streamlines of the flow. The next two terms,

$$-\left(\nabla \mathbf{u} \mathbf{C} + \mathbf{C} \nabla \mathbf{u}^T \right),$$

are responsible for the stretching of polymers. The right hand side of Eq. (1.4) models polymer relaxation and makes $\mathbf{C}$ relax back to the identity in the absence of stretching. Together, the four terms on the left hand side of Eq. (1.4) are known as the upper convected derivative of $\mathbf{C}$, which is denoted $\overset{\triangledown}{\mathbf{C}}$. This derivative is similar to the material derivative, but with added terms to track deformation and orientation of the frame of reference. The result is a derivative which maintains frame invariance of material properties in a coordinate system that moves and deforms with the flow. If we write Eq. (1.4) as

$$\overset{\triangledown}{\mathbf{C}} = -\frac{1}{\lambda}(\mathbf{C} - \mathbf{I}), \tag{1.6}$$

the physical interpretation becomes more clear: in a reference frame that moves, rotates, and stretches with the solvent flow, polymer molecules do nothing more than relax back to their unstretched state.

Numerical simulations of viscoelastic fluids are computationally challenging, and a common strategy is to add a diffusive term to Eq. 1.4. We implement this strategy in two distinct ways for the two projects presented. Thus, we leave discussion of added stress diffusion for Chapters 2 and 6.

1.3. Project summaries

The two projects presented in this dissertation both relate to the study of viscoelastic fluids, originally inspired by works relating to waves and turbulence in viscoelastic flow around obstacles [**31**, **43**, **47**, **66**]. Any attempt to make progress in this direction will face several challenges. Two that are commonly faced are numerical difficulties [**23**] as well as the general lack of understanding of physical mechanisms driving transitions in the flow [**13**, **17**, **34**]. Over time the two projects grew apart, but each managed to address one of these common challenges.

The first project is presented in Chapters 2-4, and relates to two dimensional viscoelastic turbulence. The work was inspired by the presence of similar coherent structures, which we call "narwhals," in both viscoelastic channel flow [**36**] and viscoelastic Kolmogorov flow (periodic parallel shear flow) [**7**,**8**]. In 2D Kolmogorov flow with low inertia, upon increasing polymer relaxation time, the narwhals lose stability. This leads to oscillations and eventually to elastic turbulence. On the other hand, narwhals in 2D channel flow with low inertia have not been found to lose stability. To explore possible connections between Kolmogorov flow and channel flow, we have simulated Kolmogorov flow with low frequency driving force. In an informal sense, this low frequency driving force makes Kolmogorov flow more like channel flow than it would be with higher frequency forcing. Background and experimental methods for this project are presented in Chapter 2. Our studies of Kolmogorov flow demonstrated many different flow types and transitions, which are the subject of Chapters 3 and 4. The highlight of this work is the uncovering of a period doubling cascade in Kolmogorov flow with a single period in the driving force. This cascade is, to our knowledge, the first discovery of a mechanistic route to chaos for viscoelastic fluids.

The second project is presented in Chapters 5 and 6, and relates to numerical challenges in simulating viscoelastic flows. Specifically, we introduce a modification to the Immersed Boundary method [**45**], which is used to simulate flows with physical boundaries. The Immersed Boundary method has many advantages including flexibility and easy coupling of a structured Eulerian grid to a nonconforming Lagrangian boundary grid. However, velocity gradients near the boundaries do not converge with this method. For many applications, this lack of convergence is unimportant, but for others such as viscoelastic fluids, the velocity gradient plays a crucial role (in our case, stretching polymers). We have implemented a novel modification to the Immersed Boundary method which achieves convergence of velocity gradients at boundaries of the domain.

CHAPTER 2

Introduction to Viscoelastic Kolmogorov Flow

2.1. Introduction

This chapter covers background material and methods that are relevant to our studies of viscoelastic Kolmogorov flow, the results of which are presented in Chapters 3 and 4. We begin with an overview of a phenomenon known as elastic turbulence and discuss existing results that inspired our studies. We then cover the methods used for our studies. Most of the information in this chapter and the next is available in [41].

Polymer molecules in Newtonian solvents can align and stretch with the flow causing complex flow dynamics even at low or vanishing Reynolds number. In some cases, the dynamics become chaotic, with excitation across a wide range of scales. This phenomenon, referred to as elastic turbulence, has been documented in experiments [21, 22, 43] and seen in numerical simulations [7, 34], but there is no understanding of the mechanisms involved in the transition to turbulence. Nevertheless, applications of elastic turbulence and viscoelastic instabilities abound including industrial polymer processing [15, 50], 3D printing [52, 60], and enhancement of mixing in microfluidic devices [53, 68, 69].

Significant effort has gone into understanding the theoretical underpinnings of the flow instabilities and dynamics in elastic turbulence both with and without inertia [4,5,13,17,18,39]. Purely elastic turbulence (inertialess viscoleastic flow) was first identified by Steinberg [21] and relevant theory [4, 18] and numerical studies [7, 8, 63, 64] have been performed to study the transitions in and properties of these flows. While early efforts to understand viscoelastic flow instabilities focused on curved geometries [35, 51], more recent effort has determined that chaotic flows can be reached through subcritical bifurcations and sustained in parallel shear flows [27, 34, 36, 38, 39, 43].

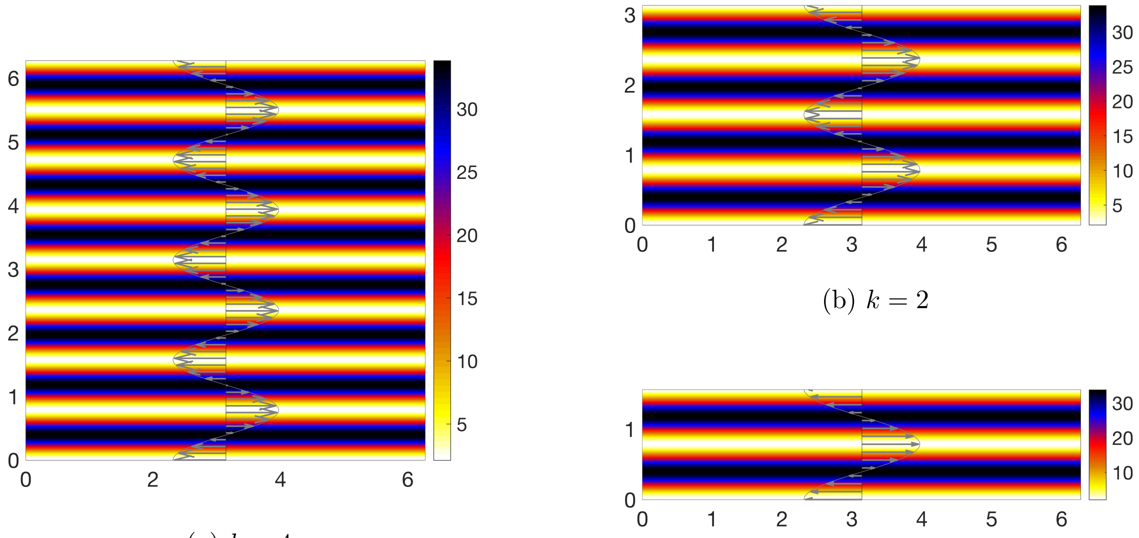
A number of numerical studies of viscoelastic fluids have featured related coherent structures. In studies with Re $\gtrsim$ 1 [16, 42] they have been referred to as arrowheads and in studies with

Re ≈ 0 [33, 34, 41], they have been referred to as narwhals. The structures were observed at least as early as 2008 in [7], with related work done in [6, 8], though they were unnamed at that time.

In 2D Kolmogorov flow (periodic shear flow with a sinusoidal driving force) at low inertia these coherent structures arise as traveling wave solutions (TWS). Upon increasing the Weissenberg number, Wi, these TWS lose stability and exhibit oscillations. At higher Wi the coherent structures repeatedly appear and disappear in chaotic flow [8]. In 2D channel flow, also at low inertia, these TWS co-exist with the stable uniform state and can be obtained using finite size (not infinitesimal) perturbations at sufficiently high Wi, but do not lead to a turbulent regime [36]. With non-negligible inertia in channel flow, TWS and chaotic regimes are found and can coexist [16]. While coherent TWS have been found in both Kolmogorov and channel flow, their relation to chaotic flow regimes is not well understood [5, 10, 13, 28].

We have studied 2D viscoelastic Kolmogorov flow at Re $= 0$. Our studies began in the same geometry as [8]: a doubly periodic, $2\pi \times 2\pi$ box, and a horizontal driving force comprised of a sinusoid with 4 periods (we use $\boldsymbol{f} = (-A\cos(4y), 0)$. We also studied Kolmogorov flow with fewer periods in the driving force (we will use k for the number of periods). Reducing k was motivated by an idea to bridge the gap between studies of channel flow [36] and those of Kolmogorov flow with $k = 4$ [8]. The laminar state in channel flow has a region of high stress along each wall, while the laminar Kolmogorov state has 2 horizontal bands of high stress per period in the background force. Thus, it is natural to wonder if reducing to $k = 1$ (one period) and matching up the number of "stress bands" to that of channel flow might yield interesting results, and in fact it does.

In Chapter 3, we discuss the results obtained with $k = 1$, where the dynamics are the simplest. These results are also presented in [41]. We then move on to $k = 2$ and finally $k = 4$ in Chapter 4. Fig. 2.1 shows the laminar flow state in all three domains. Movies with examples of most of the flow states that we discuss in Chapters 3 and 4 are available as supplemental material.



(a) $k = 4$

(b) $k = 2$

(c) $k = 1$

FIGURE 2.1. The three domains used for Kolmogorov flow. tr**C** is shown with color, and the velocity profile with grey arrows. Note the scaling of the arrows in relation to the x-axis is not meaningful. The maximum x-velocity in all cases is 4. In all cases shown here, Wi = 4.

2.2. Model

The Stokes-Oldroyd B model was introduced in Section 1.2 and will be used for all of our Kolmogorov flow studies. Here we will put the model into a slightly different form, as well as add a diffusive term to the stress evolution equation.

Recall the momentum balance, Eq. (1.2),

$$\eta_s \Delta \mathbf{u} - \nabla p + \mathbf{f}_{ext} + \nabla \cdot \tau_p = 0. \tag{2.1}$$

We divide this equation by η_s, and rescale the pressure and external force. The notation for pressure is left unchanged, and we use $\mathbf{f} = \mathbf{f}_{ext}/\eta_s$. This yields

$$\Delta \mathbf{u} - \nabla p + \mathbf{f} + \frac{1}{\eta_s} \nabla \cdot \tau_p = 0. \tag{2.2}$$

Next, we write the polymer force in terms of the conformation tensor, $\mathbf{C}$ (recall $\tau_p = (\eta_p/\lambda)(\mathbf{C} - \mathbf{I})$). The parameter $\xi = \eta_p/\eta_s$ is used to represent the ratio of viscosities. This gives us the momentum equation

$$\Delta \mathbf{u} - \nabla p + \mathbf{f} + \frac{\xi}{\lambda} \nabla \cdot \mathbf{C} = 0. \tag{2.3}$$

Note that ξ is related to the more commonly used ratio of polymer to total viscosity, β. The relation between the two is $\xi = (1 - \beta)/\beta$.

Finally, we add the term $\nu \Delta \mathbf{C}$ to the evolution equation for $\mathbf{C}$, Eq.(1.4), to give diffusion of the polymer stress, which is discussed more below. The governing equations can now be restated in the form that they will be used in this chapter:

$$\Delta \mathbf{u} - \nabla p + \mathbf{f} + \frac{\xi}{\lambda} \nabla \cdot \mathbf{C} = 0, \tag{2.4}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{2.5}$$

$$\partial_t \mathbf{C} + \mathbf{u} \cdot \nabla \mathbf{C} - \left(\nabla \mathbf{u} \mathbf{C} + \mathbf{C} \nabla \mathbf{u}^T \right) = -\frac{1}{\lambda}(\mathbf{C} - \mathbf{I}) + \nu \Delta \mathbf{C}. \tag{2.6}$$

Recall that the parameter λ is the polymer relaxation time and ξ is the ratio of polymer viscosity to solvent viscosity. Throughout our work (both Kolmogorov flow studies and Double Immersed Boundary), we fix $\xi = 1/2$ (or equivalently, $\beta = 2/3$) and vary λ. In all Kolmogorov flow simulations, we use a driving force of $\boldsymbol{f} = (-A\cos(ny), 0)$, with $n = 4$. The amplitude A is chosen such that the analytic solution (see Section 2.3.1 below) has a maximum velocity of 4, in alignment with [8]. We define the Weissenberg number as $\text{Wi} = \max|\partial U/\partial y|\lambda$, where U is the analytic velocity. With the chosen velocity, we have the relation $\text{Wi} = 16\lambda$. Rather than change the frequency, n, of this force directly, we modify the wave number, k, of the driving force by changing the size of the domain in the y-direction, see Fig. 2.1.

We have added the stress diffusion term $\nu \Delta \mathbf{C}$ to aid in stabilizing numerical simulations. However, its presence is not entirely unrealistic. Brownian diffusion of polymers can be accounted for in the underlying kinetic theory [59], but the associated diffusion coefficient, which depends on the nature of the flow being simulated, is usually far too small to stabilize numerical simulations. So, an artificially large coefficient is often used in viscoelastic flow simulations. It would be ideal to run simulations with a very small value for ν, but this requires using a high resolution, adding computational cost. A balance must be struck which allows for stable and reasonably efficient simulations, while not introducing nonphysical effects which significantly alter the results. The bulk of the results presented here use $\nu = 5 \cdot 10^{-4}$. A select number of simulations were run with $\nu = 5 \cdot 10^{-5}$ which is comparable to the value determined in [36] to realistically arise from kinetic theory.

2.3. Methods

2.3.1. Analytic solution. The system given in Eqs. (2.4)-(2.6) can be solved analytically in the case of Kolmogorov flow. For the given force the analytic solution for the velocity is

$$(u, v) = (-B\cos ny, 0), \tag{2.7}$$

and the conformation tensor is

$$C_{11} = 1 + E\sin^2(ny) + 2\lambda\nu E n^2, \tag{2.8}$$

$$C_{12} = \frac{Bn\lambda}{1+\lambda\nu n^2}\sin(ny), \tag{2.9}$$

$$C_{22} = 1, \tag{2.10}$$

where $E = 2(Bn)^2\left(\frac{1}{\lambda^2} + \frac{5\nu n^2}{\lambda} + 4\nu^2 n^4\right)^{-1}$ and $A = Bn^2\left(1 + \frac{\xi}{1+\nu n^2\lambda}\right)$. In the simulations we set $B = 4$, $n = 4$, $\xi = 1/2$, $\nu = 5 \cdot 10^{-4}$, and vary the relaxation time λ. We report results in terms of $\mathrm{Wi} = 16\lambda$.

2.3.2. Numerical solutions. We perform direct numerical simulation of the system given in Eqs. (2.4)-(2.6) with doubly periodic boundary conditions. To advance in time from t to $t + \Delta t$, we first determine $\mathbf{C}(t + \Delta t)$ using $\mathbf{C}(t)$ and $\mathbf{u}(t)$. This step is done with a fractional stepping method, using classical Runge-Kutta 4 (RK4) to update all but the diffusive term, which is done with backward Euler. We use a psuedospectral method, with spatial derivatives computed in Fourier space, and nonlinear multiplication done in real space. Prior to this multiplication a spectral filter,

$$\Lambda(K_x, K_y) = \exp\left[-36\left(\frac{K_x}{\max(K_x)}\right)^{36}\right] \cdot \exp\left[-36\left(\frac{K_y}{\max(K_y)}\right)^{36}\right], \tag{2.11}$$

is applied to each factor [25]. Note that the wave number is represented with a capital K here and in Eq. (2.12). This is done to avoid confusion with k, which represents the number of periods in the driving force.

Once $\mathbf{C}(t+\Delta t)$ is known, we use it to determine $\mathbf{u}(t+\Delta t)$. To do this, we invert the Stokes equations in Fourier space. We take the mean velocity to be 0, meaning we can invert the Laplacian in Fourier space as

$$\widehat{\Delta^{-1} f} = \begin{cases} 0 & |K| = 0 \\ \frac{-1}{|K|^2}\hat{f} & |K| > 0. \end{cases} \tag{2.12}$$

A grid resolution of $\Delta x = \Delta y = (2\pi)/512$ and a time-step of $\Delta t = 1.25 \cdot 10^{-3}$ were used for all of our Kolmogorov flow simulations, except those select few that used lower diffusion. For the simulations with lower diffusion in Section 3.6.3, the grid was refined to $\Delta x = \Delta y = (2\pi)/1024$ and the time step was halved.

2.3.3. Perturbations. The initial condition used for each simulation consists of either the analytic solution or, in the case of numerical continuations, the final state of a previous simulation. In either case, a small random perturbation is applied to the first component of the conformation tensor, C_{11}. The first component of the initial data for the conformation is

$$C_{11,init} = C_{11} + \delta(x,y),$$

where $\delta(x,y)$ is the perturbation, which is of the form

$$\delta(x,y) = 10^{-6} \cdot M \cdot |R(x,y)|.$$

At each (x,y) location, the value of R is random, drawn from a standard normal distribution (mean zero, standard deviation one). M is the max of C_{11} before the perturbation is applied.

2.3.4. Energies. In many cases, results of the simulations are presented in terms of the following two quantities:

$$E_s = \int\int \mathrm{tr}\mathbf{C}\, dxdy, \qquad E_k = \frac{1}{2}\int\int |\mathbf{u}|^2 \quad dxdy. \tag{2.13}$$

E_k is the kinetic energy of the flow. E_s is proportional to the strain energy, which is given by

$$\text{strain energy} = \frac{\eta_p}{2\lambda}\int\int \mathrm{tr}\mathbf{C}\, dxdy. \tag{2.14}$$

E_s scales differently in λ (and thus also Wi) than the true strain energy given in Eq. (2.14). Despite this difference, we will use the phrase "strain energy" to refer to E_s as well as "strain energy density" to refer to tr**C** moving forward.

2.3.5. Windowing data for Fourier transforms. Many of our results involve identifying frequencies of oscillation in our data by computing its spectrum, for which we use Matlab's fast Fourier transform (fft) function. Rather than using the raw data as input to the fft, we use a step of preprocessing to obtain higher quality results. The idea behind this step is to avoid taking Fourier transforms of data that is not smooth, because Fourier coefficients decay at a rate which depends on the smoothness of the function being transformed. In particular, a C^n function will have Fourier coefficients that decay like $1/\omega^{n+1}$, though the exact rate is not important to this discussion.

The fft computes a discrete Fourier transform, so if the input data has domain $[0, T]$, the algorithm will compute approximate Fourier coefficients for a T-periodic extension of the data. This can quickly create a problem because, even if our data consists of smooth, steady oscillations, we must provide an integer number of periods in order for the extension to be smooth. If the input contains a noninteger number of periods, the extension will almost certainly have a discontinuity, though in principle it could also be continuous but lack smoothness.

The quality of the spectrum returned by the fft is also impacted by the length of the signal used as input. For data with any amount of noise, or (as often happened in our studies) a small decaying transient, providing the fft a short input signal can result in a spectrum where the desired peaks are obscured by noise. A longer signal can make these peaks more distinct and aid in identifying frequencies of oscillation. Mathematically, we cannot solve the problems associated with non-smooth periodic extensions simply by providing signals of greater length to the fft. But, one might wonder if in practice longer signals could compensate for a lack of smoothness. In our work with Kolmogorov flow, we did not find using longer input data to be a useful strategy, in part because obtaining a signal with many oscillations requires a very long runtime.

Fig. 2.2 shows the results of a simple experiment that explores the impacts of both signal length and smoothness of the periodic extension. We compute six different spectra from the function $\cos(3x \cdot 2\pi)$ on different domains, whose lengths we denote L. Because this function oscillates with a frequency of 3, we expect the spectrum to contain a peak at $\omega = 3$.

The spectra in the top row (b and c) were computed using inputs whose extensions are smooth, and we can see that the peak values are several orders of magnitude above the off-peak values (in fact the difference is approximately $1/\epsilon_{mach}$), and that the spectrum computed with a longer domain of $L = 10$ (c) has a narrower peak. The spectra in the middle row (e and f) were computed using inputs whose extensions are discontinuous. The peaks at $\omega = 3$ are still visible, but separated from the off-peak values by dramatically smaller amounts than in the case of smooth extensions. Again, the spectrum computed with a longer domain (f) has a narrower peak. The bottom row shows results in the case where the extensions are continuous but not differentiable. The story is essentially the same here as in the case with discontinuous extensions.

Numerous options exist to improve the quality of results when computing spectra, including several built-in Matlab routines. Many of these routines provide a window function, which we will denote $q(x)$, that decays to 0 (or nearly so) at the beginning and end of its domain, which is chosen to match that of the data. This function is then multiplied with the data, which we will denote $f(x)$, prior to applying the fft, so that a smoother periodic extension is achieved. The idea is that $\text{fft}(q(x) \cdot f(x))$ should provide a higher quality spectrum than $\text{fft}(f(x))$. We have found, however, that we can obtain better results (particularly with data still containing early transients) with a custom window.

Our window is referred to as a rectangular window, meaning that we use an indicator function for our $q(x)$. The effect of this choice is to cut off some of our input data before the fft is applied, and we optimize the position of the cutoff. The optimization is done under the assumption that the input data is smooth and periodic, and so the spectrum should decay at high frequencies. Our approach is to seek out small values in the spectrum at the highest frequencies.

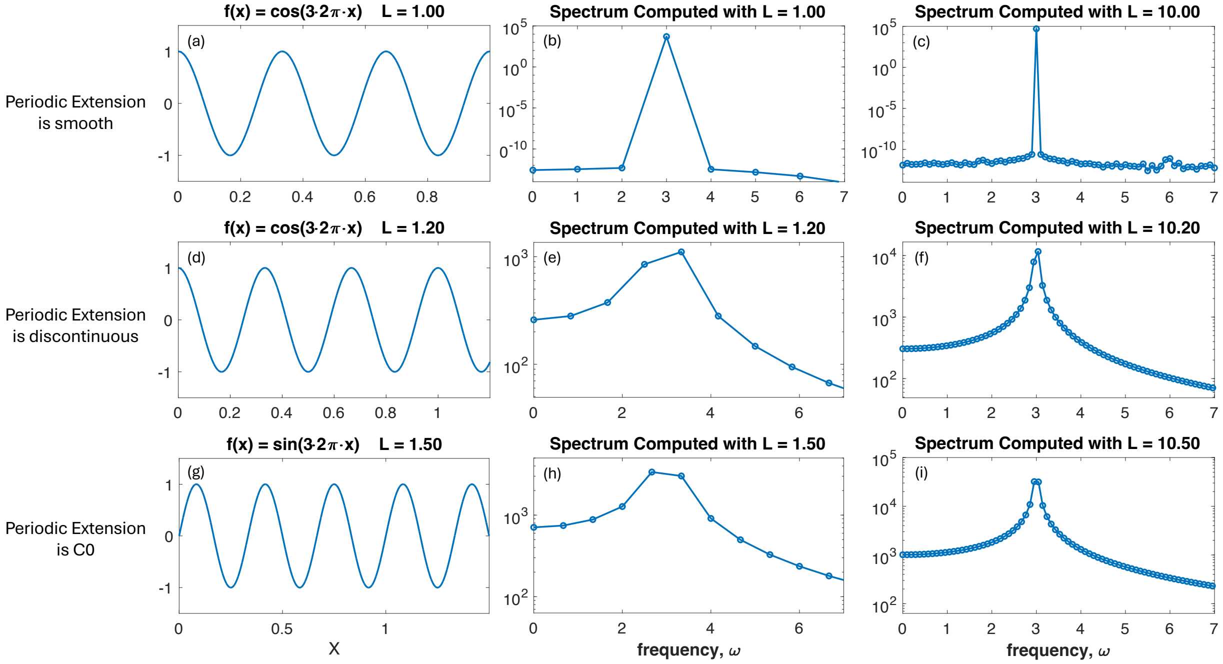


Figure 2.2. Demonstration of the need for careful data windowing. Spectra are computed from different length data without windowing. Note the dramatically different scales of the spectra plots. The top row (a-c) shows the desired result in which data that oscillates with a frequency of 3 produces a spectrum with a spike at $\omega = 3$. The domain of the data was chosen so that its periodic extension is smooth. The middle row (d-f) shows the result when the domain of the data produces a discontinuous extension. The bottom row (g-i) shows the result of an extension which is continuous but not differentiable. In all cases, a spatial resolution of $\Delta x = 1 \cdot 10^{-4}$ is used in the signal.

We proceed in a brute force manner, computing the fft with many different cutoff points and checking the size of the highest frequency Fourier coefficient each time. Because our data may contain the remnants of transient dynamics in the early part of the signal, we always cut off data from the beginning of the signal during this process. Also, we make sure to test cutoff points through at least one period of oscillation to ensure that we do not inadvertently fail to test a cutoff point that creates an integer number of frequencies. Once we have checked all the desired cutoff points, we keep the cutoff that gave us the smallest spectral value at the frequency being checked.

This method works well enough, but we found that it can be improved by also considering low frequencies. We assume that the input data is smooth and periodic with a lowest frequency of oscillation that is significantly higher than the minimum frequency that is output by the fft. In other words, the desired peak in the spectrum is situated away from the first Fourier coefficient, and thus we expect the first coefficient to be small. We use a similar brute force method to check the first Fourier coefficients corresponding to various cutoff points.

Checking the first Fourier coefficient is used as a second priority in choosing the cutoff point for our window. That is, after we have checked high frequencies, we take only a selection of the best performing cutoff points and check their first coefficient values to determine a winner. With this careful windowing of data prior to using the fft, we were better able to identify frequencies of oscillation in our data. In the early stages of our work, particularly while studying the period doubling cascade, this was quite helpful.

We conclude this section with Fig. 2.3, showing sample spectra from the early part (the first 300 time units) of a Kolmogorov flow simulation. We show spectra that have been computed with no windowing, with one of the built-in routines, and with our custom window. This simulation is a continuation from a previous simulation at a different Wi, and thus it contains some transient dynamics. The peak showing a frequency of $\omega = 0.032$ is only visible with the custom window, while all three spectra show the harmonic frequency at $\omega = 0.064$.

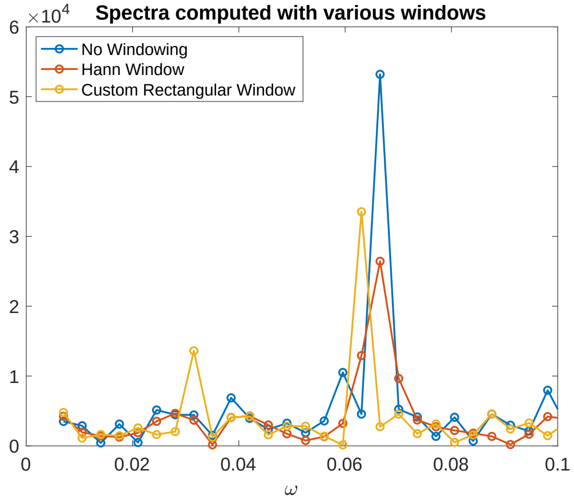


FIGURE 2.3. Spectra computed from the first 300 time units of a simulation at Wi = 27.28. At this point in the simulation, transient dynamics have not yet died out, but the spectrum with the custom window already suggests (correctly) that the fundamental frequency is $\omega = 0.032$.

2.3.6. Criteria for steady oscillations. For simulations involving oscillatory narwhals, care must be taken to ensure that a steady state has been reached because transient dynamics, sometimes manifesting as quasi-periodic behavior, can exist for a very long time. To be considered steady oscillations we examine a scalar quantity, such as the strain energy, and impose requirements on the number of oscillations and the consistency of maximum and minimum values seen through all the oscillations. To test a set of oscillations, we remove early data containing transient dynamics and apply the following procedure to the remaining data. The procedure is applied separately to the set of minima and to the set of maxima in the data.

Let x_i represent either an absolute minimum or maximum at a sample point in the data collected during one period of oscillation. We fit a second degree polynomial to the sample points around this value, $\{x_{i-1}, x_i, x_{i+1}\}$ and find the extremum of this polynomial, which we denote p_i. This helps account for deviations due to discrete sample points not being aligned with the true location of an extreme value.

Now, define $P = \{p_1, p_2, ..., p_N\}$ to be the set of all such maximum or minimum (not both) values under consideration. Let m be the mean of P and let A be the average amplitude of the oscillations. We compute the following three averages, which reflect deviation from the mean at the start (S), end (E), and through all (L) of the signal, normalized by the amplitude:

$$S = \frac{1}{A} \cdot \frac{1}{5} \cdot \sum_{i=1}^{5} |p_i - m|, \tag{2.15}$$

$$E = \frac{1}{A} \cdot \frac{1}{5} \cdot \sum_{i=N-4}^{N} |p_i - m|, \tag{2.16}$$

$$L = \frac{1}{A} \cdot \frac{1}{N} \cdot \sum_{i=1}^{N} |p_i - m|. \tag{2.17}$$

The requirements for steady oscillations are

$$N \geq 50, \qquad L < 10^{-3}, \qquad |S - L| < 10^{-4}, \quad \text{and} \quad |E - L| < 10^{-4}.$$

The first requirement is simply that the data includes at least 50 oscillations. The requirement on L ensures that across all the data under consideration, extrema occur at approximately the same value. The requirements on S and E ensure that any transient dynamics early in the signal have sufficiently died out and that the end of the signal is not showing evidence of instability.

2.3.7. Numerical methods for linear stability analysis. We begin by linearizing the Stokes-Oldroyd B equations about the analytic solution, which we denote $(\mathbf{C}_0, \mathbf{u}_0, p_0)$. Adding a perturbation, we have $(\mathbf{C}, \mathbf{u}, p) = (\mathbf{C}_0 + \tilde{\mathbf{C}}, \mathbf{u}_0 + \tilde{\mathbf{u}}, p_0 + \tilde{p})$. Plugging this into the governing equations and dropping quadratic terms yields an equation for $\tilde{\mathbf{C}}$, subject to a constraint equation involving $\tilde{\mathbf{u}}$ and $\tilde{p}$. We can write the linearized equations as

$$\partial_t \tilde{\mathbf{C}} = N(\mathbf{u}_0, \tilde{\mathbf{C}}) + N(\tilde{\mathbf{u}}, \mathbf{C}_0) - \frac{1}{\lambda}\tilde{\mathbf{C}} + \nu \Delta \tilde{\mathbf{C}}, \tag{2.18}$$

$$L(\tilde{\mathbf{u}}, \tilde{p}) = \frac{-\xi}{\lambda} \nabla \cdot \tilde{\mathbf{C}}, \tag{2.19}$$

where $N(\mathbf{u}, \mathbf{C}) = -\mathbf{u} \cdot \nabla \mathbf{C} + \left(\nabla \mathbf{u} \mathbf{C} + \mathbf{C} \nabla \mathbf{u}^T\right)$ is the collection of nonlinear terms in the Stokes-Oldroyd-B equations and L is the Stokes operator. The velocity perturbation is determined by the perturbation to the conformation tensor: $\tilde{\mathbf{u}} = L^{-1}(\frac{-\xi}{\lambda} \nabla \cdot \tilde{\mathbf{C}})$. Thus Eqs. (2.18)-(2.19) can be written in terms of $\tilde{\mathbf{C}}$ as $\partial_t \tilde{\mathbf{C}} = A(\mathbf{C}_0, \mathbf{u}_0, p_0)\tilde{\mathbf{C}}$ where A is a linear operator depending on the steady solution.

We then build the discrete operator $A(\mathbf{C}_0, \mathbf{u}_0, p_0)$ directly. That is, we assemble the operator column-by-column by applying the RHS of Eq. (2.18) to unit vectors. Once the operator is built, the Matlab function 'eig' is used to compute the eigenvalues of the operator. The analytic solution is determined to be stable when all eigenvalues have negative real part, and unstable otherwise.

As Wi is increased, Kolmogorov flow first loses and later regains linear stability, and thus there are two critical values. To determine each of these values we use a bisection search method. Before this process begins, we must manually find a lower bound Wi_l, and an upper bound Wi_u, so that the critical value lies between these bounds, i.e. $\mathrm{Wi}_l < \mathrm{Wi}_c < \mathrm{Wi}_u$. Testing the midpoint $(\mathrm{Wi}_l + \mathrm{Wi}_u)/2$ for stability allows the search window to be cut in half. This process is repeated until the upper and lower boundaries are within a tolerance of 10^{-3} of one another. At this point, the average of the upper and lower boundaries is taken to be the critical value.

When building the discretized operator $A(\mathbf{C}_0, \mathbf{u}_0, p_0)$, the grid resolution is limited by the large memory requirement of building the operator. However, coarse resolutions yield accurate results due to the low mode nature of the analytic solution. Using a resolution of $\Delta x = \Delta y = 2\pi/96$ was found to be satisfactory in terms of both accuracy and memory. Higher resolutions were occasionally used to verify results.

2.3.8. Modal analysis method. We have an analytic equilibrium solution for the conformation tensor, given in Section 2.3.1, which we will call $\mathbf{C}_{eq}$. We will call deviations from this equilibrium $\mathbf{C}_\delta = \mathbf{C} - \mathbf{C}_{eq}$. For flows with Wi just above the critical value where the equilibrium becomes linearly unstable, we expect this deviation to be approximately equal to a linear combination of the unstable modes of the linearized system. These modes, as well as the coefficients in the linear combination, are complex valued. However, since $\mathbf{C}_\delta$ is real, we equivalently expect the deviations to be approximated by a linear combination of the real and imaginary parts of the unstable modes, with real coefficients. That is,

$$\mathbf{C}_\delta \approx \sum_{i=1}^{2n} c_i \boldsymbol{M}_i, \tag{2.20}$$

where $\boldsymbol{M}_i$ represents either the real or imaginary part of an unstable mode, and we have n unstable modes.

In each case examined here, we have a double Hopf bifurcation, so there are $n = 2$ complex valued unstable modes. We will examine how well these modes capture dynamics near bifurcations by considering the traces of $\mathbf{C}_\delta$ and the unstable modes, $\mathrm{tr}\mathbf{C}_\delta$ and $\mathrm{tr}\mathbf{M}_i$ respectively.

At a single moment in time, we would like to find the best approximation to $\mathrm{tr}\mathbf{C}_\delta$ using the unstable modes. So, we need to determine coefficients c_i so that

$$\mathrm{tr}\mathbf{C}_\delta \approx \sum_{i=1}^{4} c_i(\mathrm{tr}\mathbf{M}_i). \tag{2.21}$$

To do this, we can flatten all the arrays and find a least squares solution for the coefficients. Define A_m to be the linear operator from coefficients to the corresponding (flattened) sum of the modes:

$$A_m\,\boldsymbol{c} = \sum_{i=1}^{4} c_i(\mathrm{tr}\mathbf{M}_i)_f, \tag{2.22}$$

where the subscript f indicates a flattened array. Given that $\mathbf{M}_i$ are known, we can construct A_m directly, where the i^{th} column is simply $(\mathrm{tr}\mathbf{M}_i)_f$. So, finding optimal coefficients can be done using the backslash operator to solve the least squares problem,

$$\boldsymbol{c} = A_m \backslash \mathrm{tr}\mathbf{C}_{\delta,f} \tag{2.23}$$

CHAPTER 3

Viscoelastic Kolmogorov Flow with Single Period Forcing

3.1. Introduction

In this chapter we begin the presentation of our results for 2D viscoelastic Kolmogorov flow. Relevant background and methods were presented in Chapter 2. The current chapter is concerned with flows driven by a single period forcing ($k = 1$), and Chapter 4 will discuss flows driven with multiple period forcing. Note that with a few exceptions, the material in this chapter is available in [41].

We evolve the Stokes-Oldroyd B model given in Section 2.2 on a doubly periodic domain of $[0, 2\pi] \times [0, \pi/2]$, using the driving force $\boldsymbol{f} = (-A\cos(4y), 0)$. The analytic solution on this domain is visualized in Fig. 2.1c and Fig. 3.1a. We study a range of Wi by varying the polymer relaxation time, λ. Recall that we have the relation $\mathrm{Wi} = 16\lambda$.

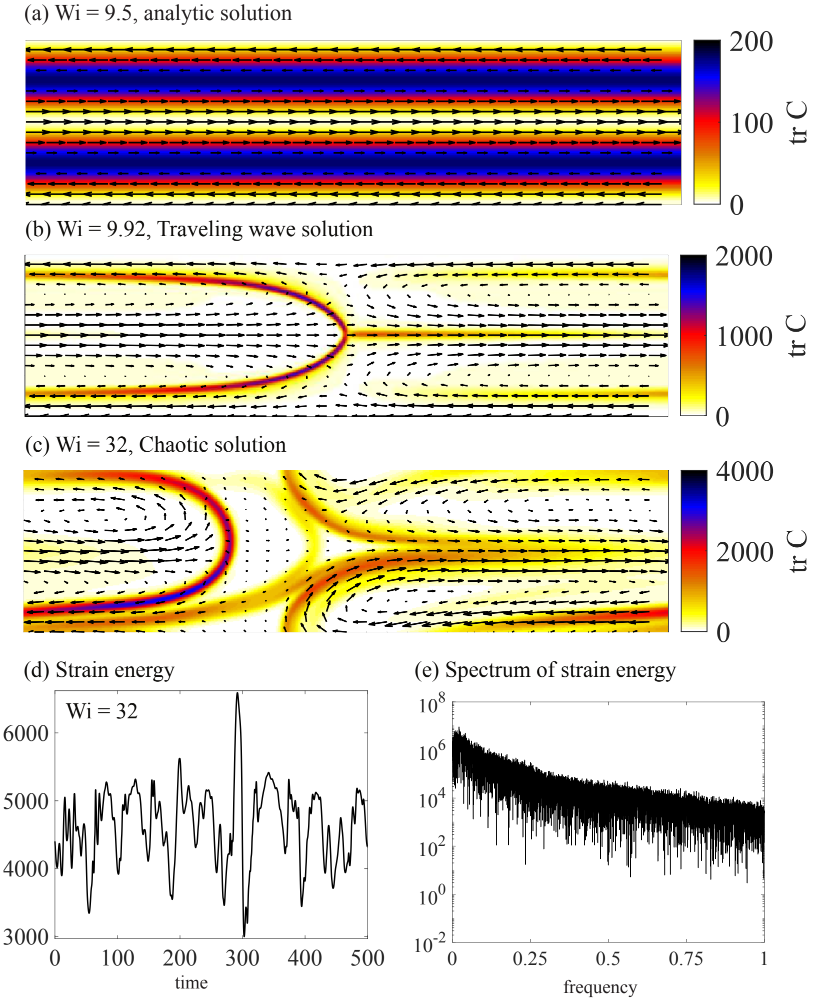
Much of this chapter is focused on coherent structures which we call narwhals. The distinctive pattern in the strain energy density that led to this naming choice can be seen in 3.1b. A similar traveling wave solution (TWS) has been seen in channel flow [36], and in Kolmogorov flow with higher spatial frequency in the driving force similar traveling structures arise as part of the solution [8]. In low Reynolds number 2D channel flow these TWS have not been shown to lose stability [36], while in Kolmogorov flow there is a transition to a chaotic state at higher Wi [8]. We similarly see chaotic behavior for $\mathrm{Wi} \gtrsim 27.3$.

This chapter is organized into two parts. In Sections 3.2 - 3.5 we present the most fundamental results, beginning with the loss of linear stability of the analytic solution and working our way upwards in Wi. Following this, additional details will be presented in Section 3.6.

3.1.1. Overview of dynamics. At low Wi, the analytic solution given in Section 2.3.1 is linearly stable. This solution is shown in Fig. 3.1a at $\mathrm{Wi} = 9.50$. As Wi is increased, the analytic solution becomes linearly unstable. As we progress upwards in Wi, we obtain standing waves (to be discussed in Section 3.6.1), followed by traveling wave solutions (TWS) in the form of coherent structures we call narwhals. Fig. 3.1b shows the strain energy density and velocity field in a flow containing a narwhal at Wi = 9.92. Further increasing Wi causes these structures to lose stability and begin oscillating, after which they proceed through a cascade of period doubling bifurcations. Above the doublings, a single period tripling was found, followed by intermittency and chaos.

A time snapshot of the strain energy density and flow are given in Fig. 3.1c for Wi = 32, where a structure resembling the narwhal is present.

In Fig. 3.1d we see aperiodic behavior of the strain energy over time, and the energy spectra, Fig. 3.1e, shows a wide range of excited frequencies. This behavior is typically classified as elastic turbulence [7].


(a) Wi = 9.5, analytic solution

(b) Wi = 9.92, Traveling wave solution

(c) Wi = 32, Chaotic solution

(d) Strain energy

(e) Spectrum of strain energy

FIGURE 3.1. Strain energy density with velocity vectors overlayed for (a) analytic solution at Wi = 9.5, (b) traveling wave solution at Wi = 9.92, (c) chaotic solution at Wi = 32. (d) Time series of the strain energy for Wi = 32 over a representative time interval. (e) Spectrum of strain energy, Wi = 32.

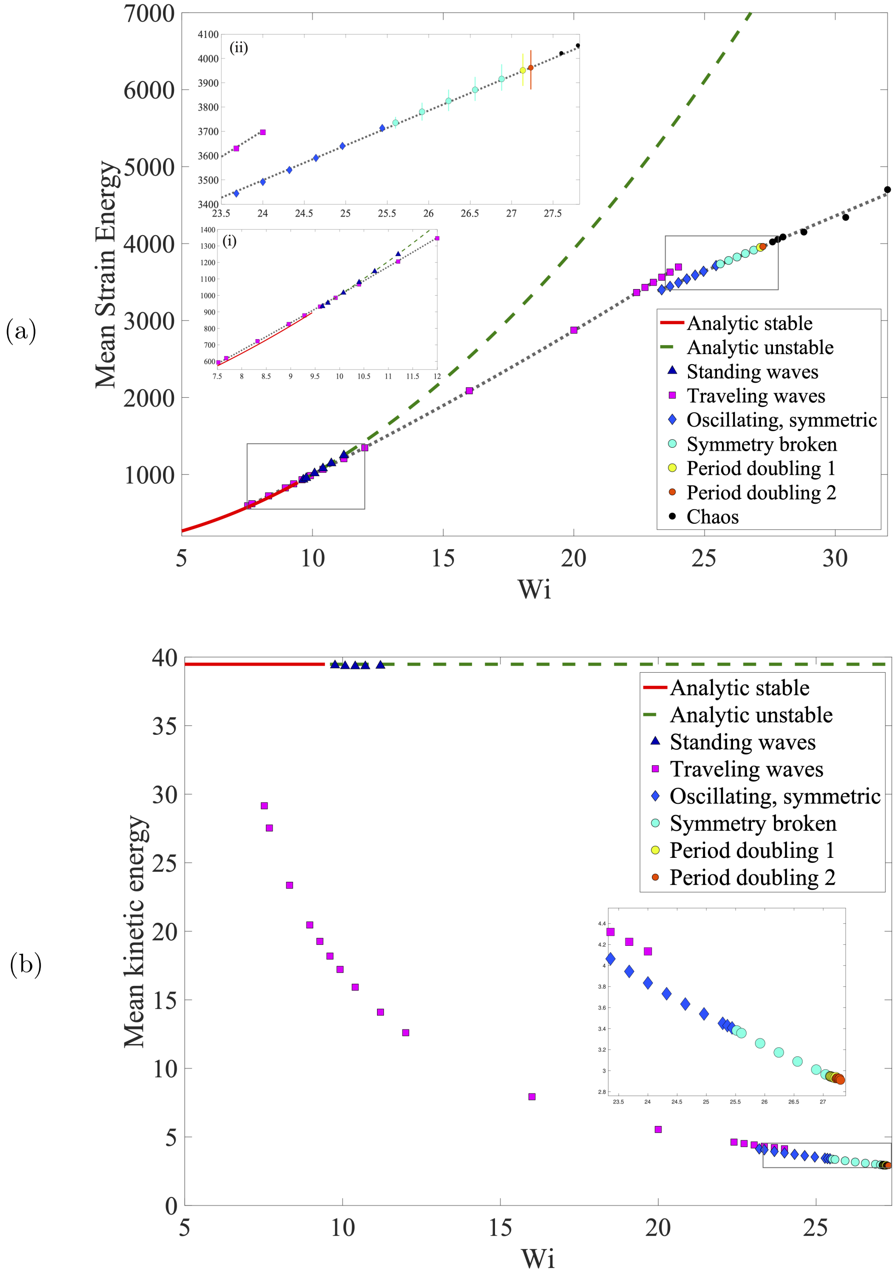


Figure 3.2. As a function of Wi, we show the mean strain energy (a) and mean kinetic energy (b), along with classification of solution types. In (a), the analytic solution is quadratic in Wi. A dotted line shows a $3/2-$power law fit to the data for the TWS branch. A dotted line shows a linear fit to the data for the period doubling and chaotic branch. The lines on the markers in inset (ii) show the maximum and minimum values of the strain energy for the oscillating solutions.

Fig. 3.2 contains bifurcation diagrams which organize the various flow types we have classified as functions of Wi. Diagrams are shown in terms of both temporal mean of the strain energy, Fig. 3.2a, and temporal mean of the kinetic energy, Fig. 3.2b. These diagrams will be referred to several times throughout the chapter.

3.2. Results with low Wi: Standing and traveling waves

We perform linear stability analysis and determine that the analytic solution is stable for Wi < 9.61. The values of Wi where stability is lost at increasing resolutions are reported in Table 3.1.

Nx	Wi_c $(\pm 5 \cdot 10^{-4})$
32	8.169
48	11.194
64	9.619
96	9.615
128	9.613
192	9.614

Table 3.1. Critical Wi values found at various grid resolutions. As Wi is increased past Wi_c, Kolmogorov flow switches from linearly stable to linearly unstable. This computation is done with the stress diffusion coefficient $\nu = 5 \cdot 10^{-4}$.

In the stable region the analytic solution for the strain energy density has regions of high stress that correspond to where the shear rate is strongest; see Fig. 3.1a for results at Wi $= 9.5$. Just above this threshold ($9.65 \leq$ Wi ≤ 11.2) small perturbations to the analytical solution result in standing wave solutions, which are discussed in Section 3.6.1.

Above the threshold at Wi $= 11.2$ where the standing wave solutions exist, TWS are obtained starting from small perturbations from the analytic solution. By downward continuation in Wi the TWS are found to co-exist with the standing wave solutions and the analytic solutions. As an example and to illustrate the TWS structure, the strain energy and velocity for the TWS at Wi $= 9.92$ are shown in Fig. 3.1b. The bifurcation diagram with kinetic energy shows a significant drop in E_k from the analytic to the TWS branch. As we decrease Wi below 9.92, the TWS persists down to Wi $= 7.52$ (see Fig. 3.2a inset (i)). Below this value, the analytic solution is the only solution we found by continuation. In the range $7.52 <$ Wi < 9.61 both the traveling wave and analytic solutions are stable. This is similar to channel flow [5, 36] where the stable traveling wave solution and analytic solutions co-exist above some threshold in Wi, but in channel flow the base state remains stable.

3.3. Results with moderate Wi: Narwhals and oscillations

Increasing Wi, the numerical continuation produces steady traveling narwhal solutions until about Wi ≈ 24. A new lower energy stable branch corresponding to oscillatory narwhals appears as a subcritical bifurcation, see Fig. 3.2a inset (ii). Both steady TWS and oscillating solutions exist for $23 \lesssim \mathrm{Wi} \lesssim 24$.

The strain energy for the analytic solution demonstrates a quadratic scaling in Wi which is seen in Fig. 3.2a, and a 3/2 power-law (dotted line) is the best fit to the strain energy data in the TWS regime. The sub-critical bifurcation to oscillating symmetric solutions appears again as a lower energy branch with approximately linear (dotted line) scaling in Wi. As Wi increases the amplitude of the oscillations are initially small but grow with Wi as indicated by vertical bars in Fig. 3.2a inset (ii).

3.3.1. Symmetric oscillating solutions. The first oscillatory solutions that emerge after traveling waves have a vertical symmetry which is lost as Wi increases. In Fig. 3.3 we show the strain energy density in the oscillatory regime where the narwhal exhibits a periodic vertical deformation.

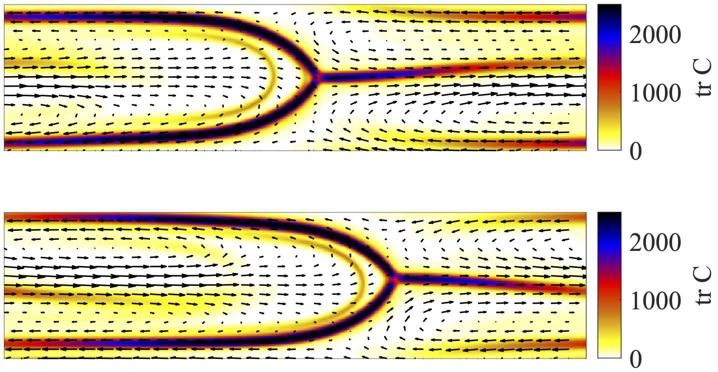
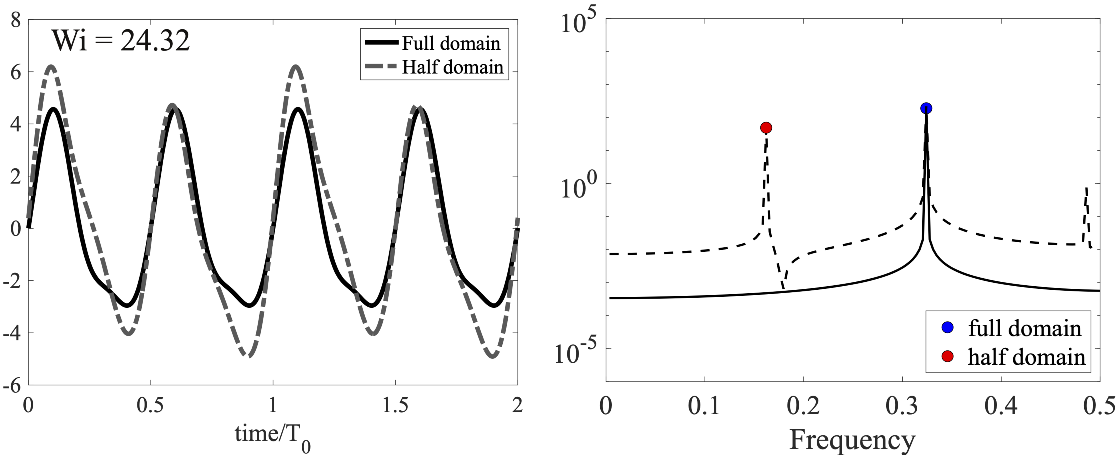


FIGURE 3.3. The strain energy density and velocity vectors for Wi = 24.32 at $t = 0$ and $t = T_0/2$, period $T_0 = 6.17$.

The symmetry can be seen by examining the strain energy and spectrum over both the full domain as well as only the (bottom) half of the domain. Fig. 3.4a shows deviations from the mean strain energy where the mean is taken over the full domain and also over only the bottom half of the domain. When the period is computed using the dominant frequency over the full domain a period of 3.085 is calculated, whereas over only the bottom half the period is 6.17, as shown in Fig. 3.4b.



(a) Deviation from mean strain energy over time.

(b) Spectrum of strain energy.

FIGURE 3.4. (a) The deviation from the mean strain energy averaged over both the full domain and bottom half of the domain. (b) The spectra of the energy over full and half domain show different dominant frequencies. The full domain has dominant frequency 0.3241 (period 3.085), and over half the domain the dominant frequency is 0.162 (period 6.17). This difference occurs due to the vertical symmetry for this solution.

3.3.2. Period doubling cascade. In Fig. 3.5a the period of the strain energy, T_0, is plotted for a portion of the branch of periodic solutions below the chaotic regime, $24.5 < \mathrm{Wi} < 27.3$. The period slowly increases in Wi, and near $\mathrm{Wi} = 27.12$ the period jumps from 7.72 to 15.47. The period again doubles around $\mathrm{Wi} = 27.216$. The harmonics $T_0/2$ and $T_0/4$ are highlighted for the first and second doublings to help illustrate the period doubling phenomena.

We further illustrate period doubling by visualizing the strain energy (Figs. 3.5b - d) and its spectrum (Figs. 3.5h - j) for three solutions on the period doubling branch at $\mathrm{Wi} = 26.24, 27.168,$ and 27.264. For each of these three Wi we additionally visualize the strain energy versus the kinetic energy in Figs. 3.5e - g. For solutions with the shortest period this is a loop, and at each successive doubling the loop splits in two. The colors of the loops correspond to the strain energy over time.

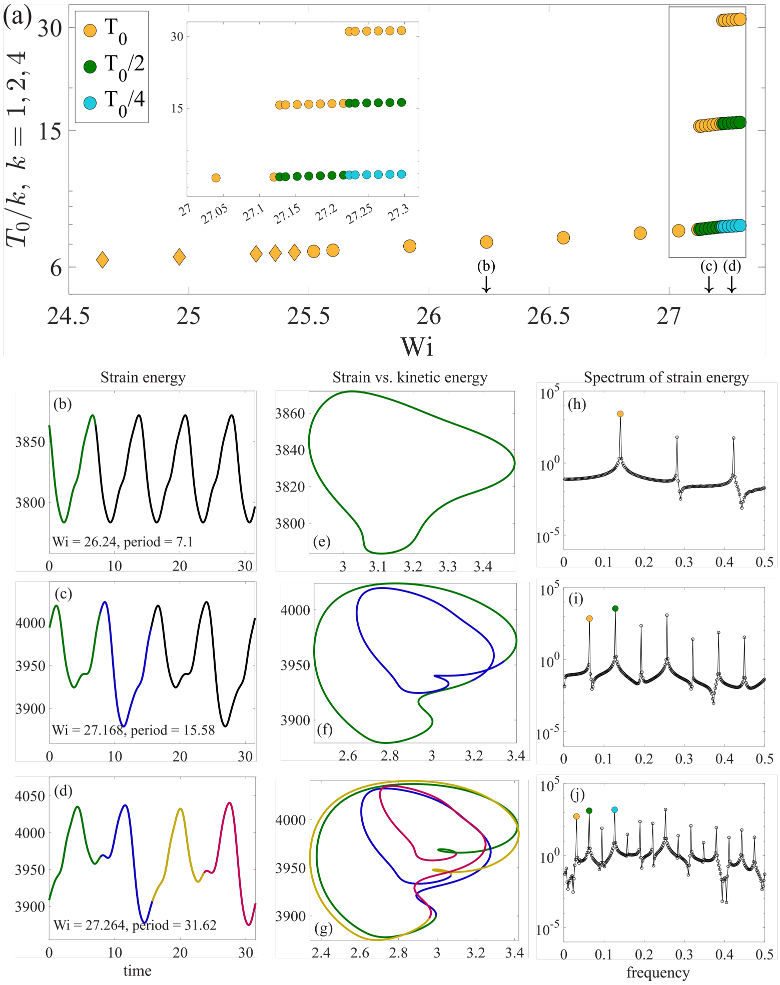


FIGURE 3.5. (a) Period and harmonics as a function of Wi. Diamond markers represent symmetric solutions. (b)-(d) Strain energy over time for Wi $= 26.24, 27.168, 27.264$ with a period highlighted in color(s). (e)-(g) Strain energy versus kinetic energy over a period with color labeling matching (b)-(d). (h-j) Spectrum of strain energy with dominant frequencies highlighted, corresponding to the period and harmonics labeled in (a).

In Fig. 3.6a, the bifurcations in the region $25 < \mathrm{Wi} < 28.1$ are visualized using local extrema of deviations from the mean of the kinetic energy. For solutions below $\mathrm{Wi} \leq 27.296$ extrema of the periodic solutions are colored according to the legends in the bifurcation diagrams of Fig. 3.2. The grayscale markers for $\mathrm{Wi} > 27.296$ also represent local extrema, but in this regime the flow has become increasingly chaotic and the extrema no longer repeat periodically. In this regime the extrema are binned according to the number of extrema with similar values in that bin with darker markers indicating more occurrences in a bin. This branching diagram has similarities with orbit

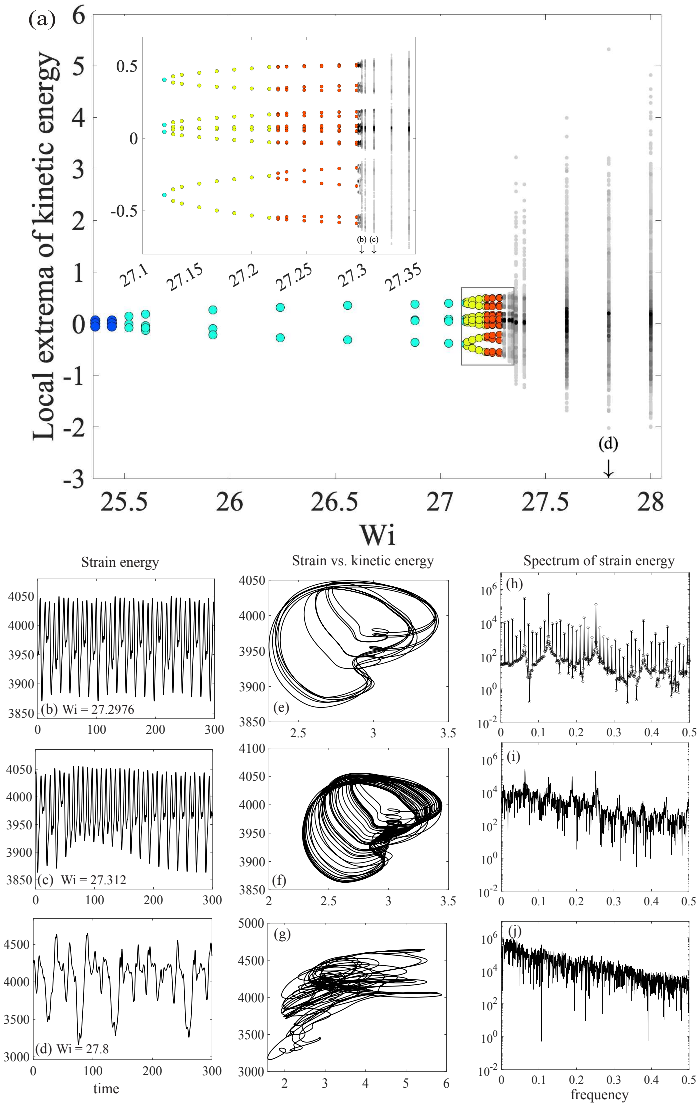


FIGURE 3.6. (a) The local extrema of the deviation from the mean kinetic energy. Periodic solutions are colored according to the legend in the bifurcation diagram (Fig. 3.2a. (b)-(d) Strain energy over time for Wi $= 27.2976, 27.312, 27.8$ over a representative time interval. (e)-(g) Strain energy versus kinetic energy. (h)-(j) Spectrum of strain energy.

diagrams of iterated maps [58]. Below the first period doubling (Wi ≤ 27.12) there are between 2-4 local extrema for each solution. At the first doubling (cyan to yellow markers) 4 local extrema become 8, and at the next doubling (between yellow and orange markers) the 8 extrema split into 16.

3.4. Results with high Wi: Transition to chaos

Beyond Wi $\approx$ 27.264 the flow in these regions may have very long periods, may be intermittently chaotic, or may be chaotic. For example, at Wi = 27.2976 we find a period of 95.24, indicating a potential period tripling. To demonstrate this we show Fig. 3.7, which visualizes the strain energy, strain energy vs kinetic energy, and spectrum of the strain energy for one solution below (in Wi) the tripling and one solution above. Note that data from the Wi = 27.2976 simulation is also included in Fig. 3.6b,e,h.

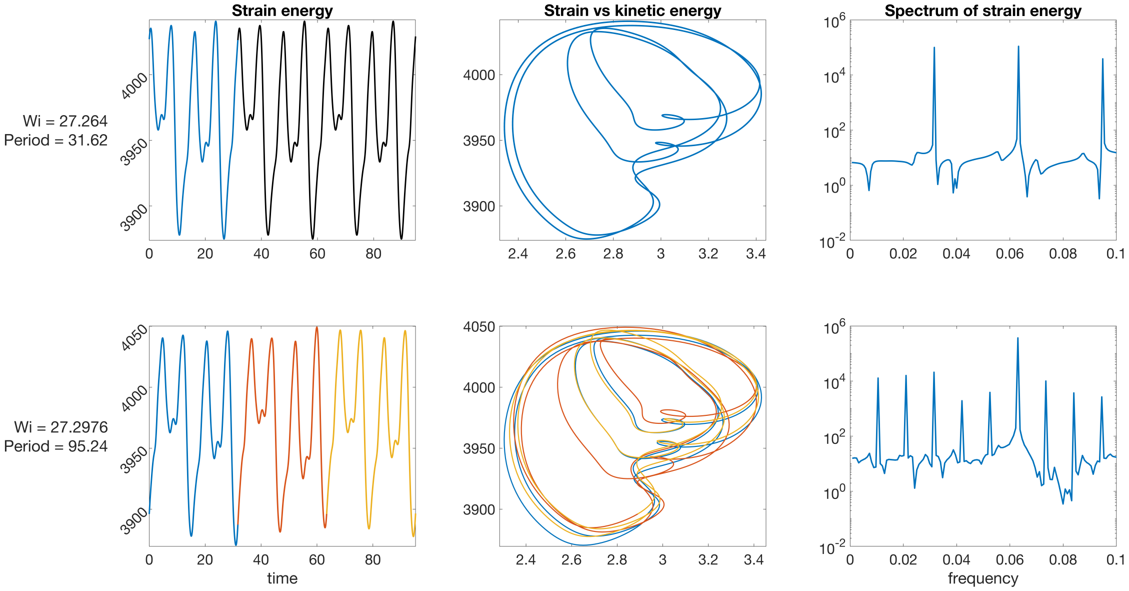


FIGURE 3.7. Period tripling illustrated by the strain energy over time (left column), strain energy vs kinetic energy (middle column), and spectra (right column)

Further increasing to Wi = 27.312 (Figs.3.6c,f,i), the strain energy over time shows structure to the oscillations but also regions of intermittency. The frequency still shows distinct peaks in the signal, but there is also a lot of additional noise with many frequencies being active. Increasing further to Wi = 27.8 (Figs. 3.6d,g,j) the dynamics more closely resemble those previously described as chaotic for Wi = 32 (Figs. 3.1d,e).

3.5. Summary of main results

Using forward simulations and numerical continuation we identified a subcritical bifurcation from traveling wave solutions to oscillating solutions in a 2D viscoelastic fluid driven by a periodic shear flow. The oscillating solutions then undergo a series of period-doubling bifurcations below the chaotic regime. As these bifurcations occur, waves in the narwhal “tusk" appear to oscillate and interact with other regions of high stress in the coherent structure. This self-interaction of a single narwhal structure appears to lead to longer periods of oscillations as Wi increases until a chaotic regime is reached. This bifurcation structure is the first systematic connection between coherent structures to chaotic dynamics in a viscoelastic fluid.

Chaotic flows have been seen in Kolmogorov flow with higher frequency forcing [6, 8] but the transition to chaos involves more complex dynamics due to the interaction of multiple coherent structures. However, the existence of the narwhal traveling wave solution does not imply that a flow will become chaotic; in fact a single structure appears to be stable in channel flow with low inertia [36]. Unlike 2D, in 3D the narwhal in a channel appears to become unstable leading to chaotic flows [33, 34]. The relatively simple framework of a single narwhal in 2D allows the identification of bifurcations that may also be underpinning those found in other more complex flows on the route to chaos.

The single narwhal structure is reminiscent of channel flow yet it still exhibits chaos; this comparatively simple flow and subsequent flow transitions give the first description of a mechanistic route to chaos. Further study of this system may help unlock other mechanisms driving the transition to chaos in more complex systems through numerical simulations and analysis. We now transition to a detailed look at additional results related to Kolmogorov flow with $k = 1$

3.6. Additional results with $k = 1$

3.6.1. Standing waves. Just above the critical value $\mathrm{Wi}_c = 9.61$ where the analytic solution becomes linearly unstable, there is a small range, observed in simulations from $9.65 \leq \mathrm{Wi} \leq 11.2$, that supports solutions which are distinct from the others discussed in this Chapter. These solutions are standing waves in the strain energy density, as shown in Fig. 3.8. The waves alternate between two configurations ($t = 0$ and $t = 2\mathsf{T}/4$ in the figure), each with four concentrated, slightly tilted “stress islands.” The difference between these two states is merely a horizontal shift by $\pi/2$.

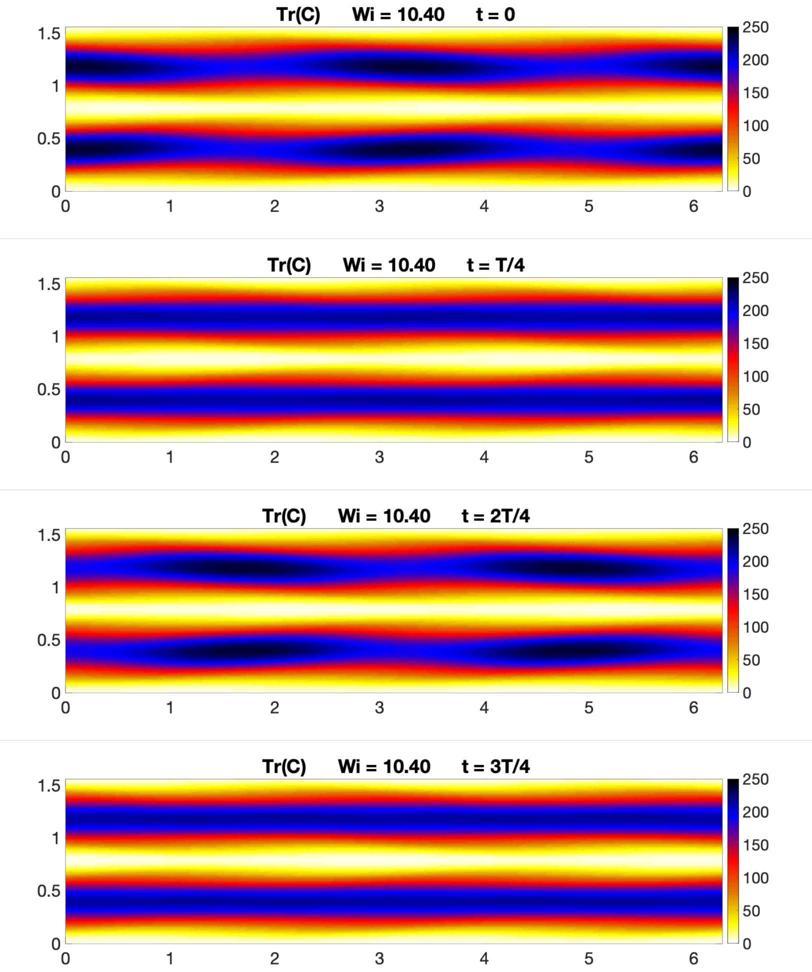
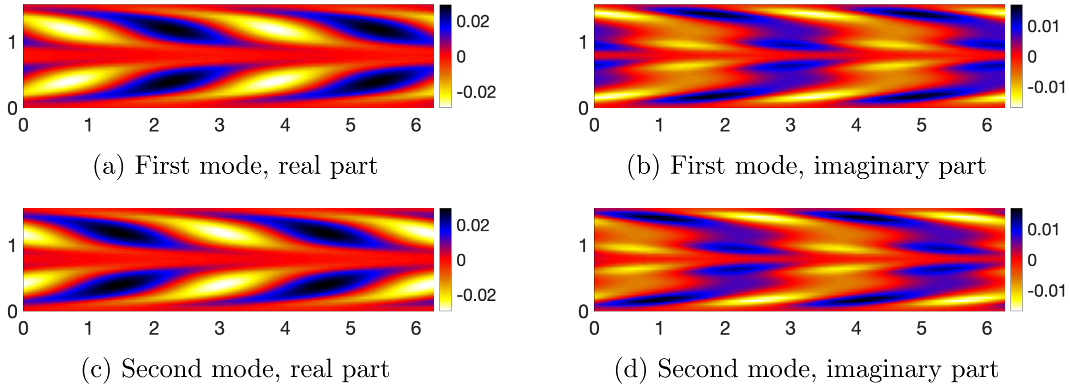


FIGURE 3.8. Standing waves over one period of oscillation for Wi $=$ 10.4. The period, $\mathsf{T} = 0.80$.

Note, however, that the stress islands are *not* traveling horizontally. Rather, they disappear ($t = \mathsf{T}/4$ and $t = 3\mathsf{T}/4$) and reform in the new location.

The standing waves emerge from the analytic solution in a continuous manner as Wi is increased, suggesting a supercritical bifurcation. Continuing upwards to Wi > 11.2, traveling wave narwhals emerge, which are discussed in the main text. When the narwhals form, there is a large decrease in the kinetic energy, which is visible in the bifurcation diagram in Fig. 3.2b.

3.6.2. Modal analysis results. Here we will examine standing wave solutions at $\mathrm{Wi} = 9.65$, which is just above the critical value of $\mathrm{Wi} = 9.613$ where the analytic solution becomes linearly unstable. We carry out a modal analysis using the methods described in Section 2.3.8. $\mathrm{tr}\mathbf{C}$ of the unstable modes of the linearized system are shown in Fig. 3.9.



(a) First mode, real part

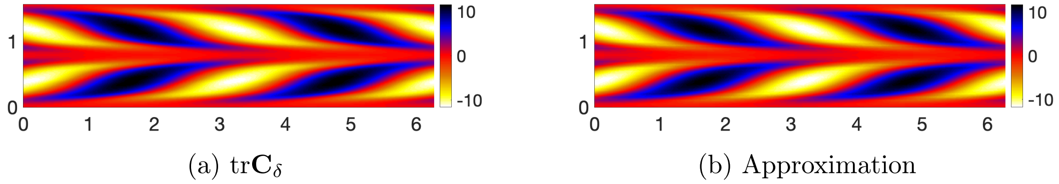
(b) First mode, imaginary part

(c) Second mode, real part

(d) Second mode, imaginary part

FIGURE 3.9. Unstable modes from linear stability analysis at $\mathrm{Wi} = 9.65$. Computed on a 128x32 grid and interpolated to 512x128.

In Fig. 3.10a, we show a time snapshot of the deviation of the standing wave solution from the analytic solution, which we denote $\mathrm{tr}\mathbf{C}_\delta$. In the standing wave regime, the unstable modes account for most of the deviation from the analytic solution that is seen in simulations of the full, nonlinear system. We use the unstable modes to build a least squares best approximation to $\mathrm{tr}\mathbf{C}_\delta$, and display this approximation in Fig. 3.10b.



(a) $\mathrm{tr}\mathbf{C}_\delta$

(b) Approximation

FIGURE 3.10. Deviation from the analytic solution (a) and the approximation formed with a linear combination of the unstable modes. For this approximation, the relative error is 0.053 measured with the Frobenius norm.

We measure the difference between $\mathrm{tr}\mathbf{C}_\delta$ and the approximation with the Frobenius norm. Because we are approximating a solution to the nonlinear system with modes from the linearization, we do not expect the norm to be small. It merely gives some indication that, because the standing waves exist near enough (in Wi) to the linear instability, much of the deviation from equilibrium is due to linear terms. Computing the Frobenius norm for many snapshots in time reveals temporal information about the standing waves. In Fig. 3.11, we can see oscillations in time of the size of the coefficients and of the error in the approximation.

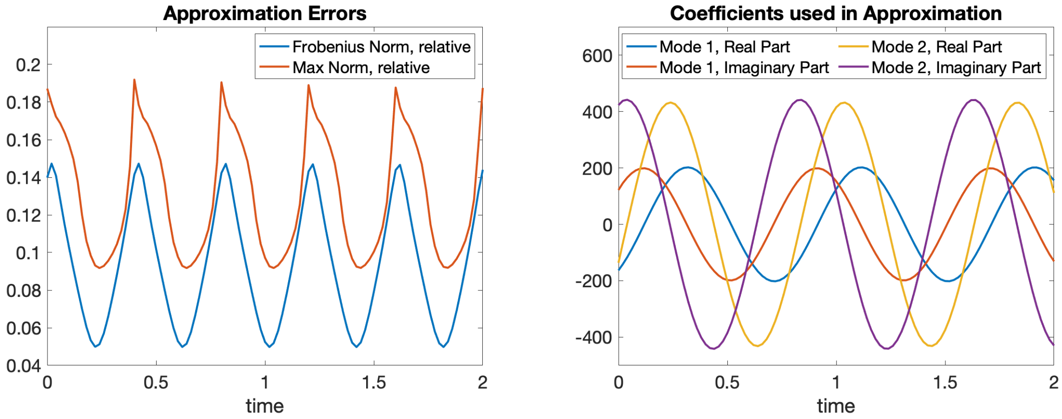
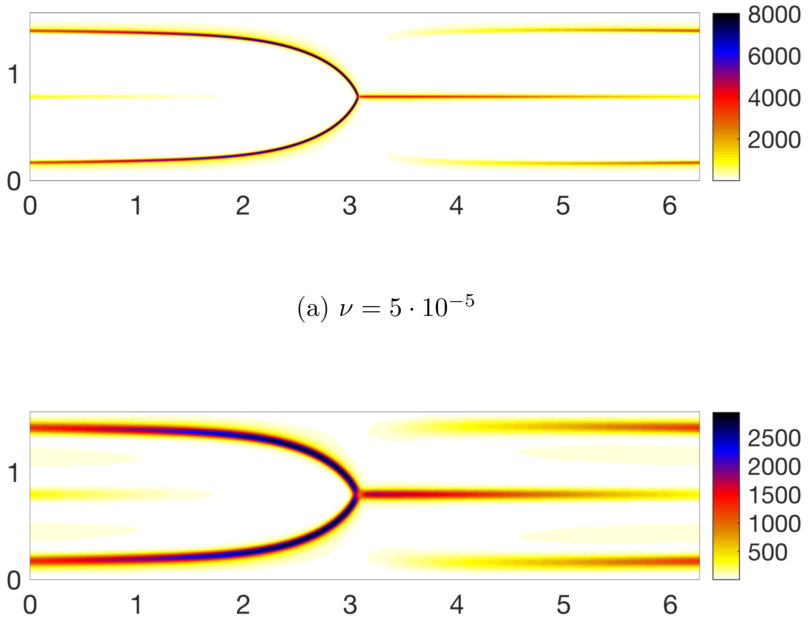


FIGURE 3.11. Left: Relative Frobenius norm of error over time for unstable mode approximation to standing wave. Right: Coefficients for each mode over time.

In Fig. 3.8, we can see that the standing wave oscillates between states which are reminiscent of the analytic solution (stress bands nearly constant in the x-direction) and states in which the stress shows spatial oscillations in the x-direction. The analytic-like states roughly correspond to the largest errors in the approximation and the times at which the coefficients are all relatively small.

3.6.3. Checking results at lower diffusion. Upon finding the route to chaos presented in Section 3.3.2, a natural question to ask is whether the size of the stress diffusion ($\nu = 5 \cdot 10^{-4}$) played an important role in the dynamics. To help answer this question, we ran a select number of simulations with $\nu = 5 \cdot 10^{-5}$, which is comparable to the value determined in [**36**] to realistically arise from kinetic theory. With this lower value of diffusion, we found that the route to chaos via period doubling bifurcations remains intact, and discovered no indications of substantially different dynamics as a result of reducing diffusion. The low diffusion simulations revealed traveling waves and oscillations which first break symmetry and then undergo a period doubling. Simulations with this amount of diffusion required a grid resolution of 1024×256 and a timestep of $\Delta t = 6.25 \cdot 10^{-4}$.

From $8.8 \leq \mathrm{Wi} \leq 22.4$, we obtained traveling wave solutions taking the form of narwhals. An example is shown in Fig. 3.12. These solutions contain exceptionally high peak stresses with sharp gradients, which necessitated a smaller timestep ($\Delta t = 3.125 \cdot 10^{-4}$) than other simulations to maintain stability.



(a) $\nu = 5 \cdot 10^{-5}$

(b) $\nu = 5 \cdot 10^{-4}$

FIGURE 3.12. Strain energy density of a traveling wave solution at $\mathrm{Wi} = 16$ with $\nu = 5 \cdot 10^{-5}$ (top) compared to $\nu = 5 \cdot 10^{-4}$ (bottom). Note the significantly higher peak stress in the lower diffusion case.

Oscillatory narwhals emerged at higher Wi through a subcritical bifurcation. The oscillating behavior was continued downwards to $\mathrm{Wi} = 21.6$, demonstrating the coexistence of oscillatory narwhals with steady TWS narwhals. The first oscillations to emerge are characterized by a vertical symmetry which results in the kinetic and strain energies oscillating at half the period of the narwhal's oscillation. The energies' oscillatory periods in this regime range from 2.68 to 3.12.

Continuing upwards in Wi, the symmetry in the oscillations breaks, and at $\mathrm{Wi} = 24$ the energies oscillate with a period of 6.44. Above this, a period doubling occurs and at $\mathrm{Wi} = 24.16$ the period is 13.20. The strain energy over time and its spectrum before and after this period doubling is shown in Fig. 3.13.

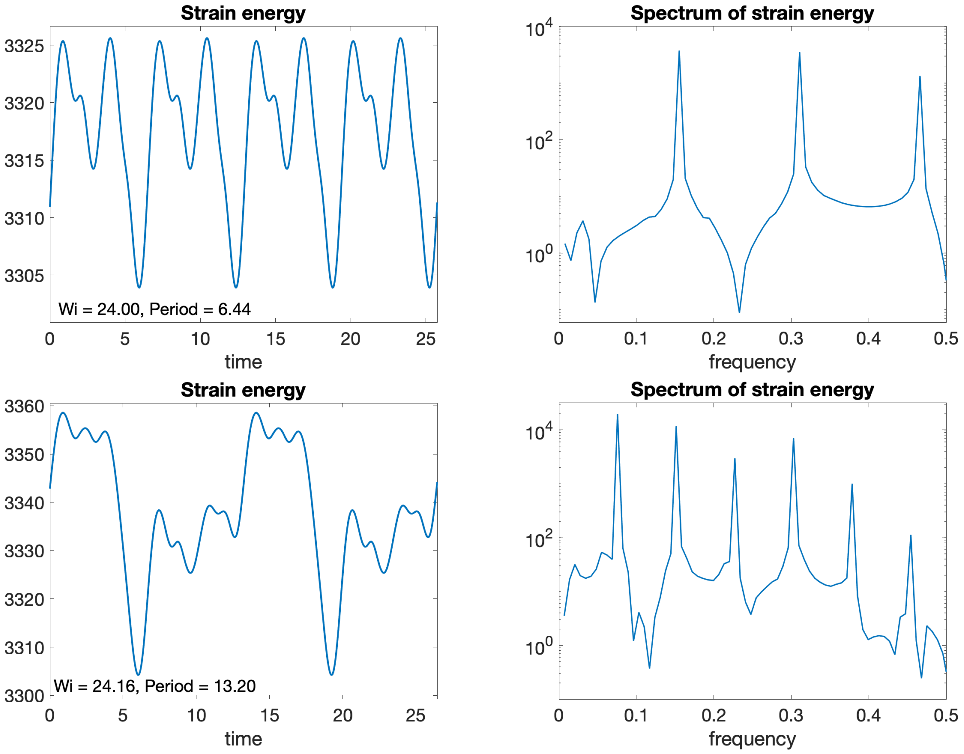


FIGURE 3.13. A period doubling with $\nu = 5 \cdot 10^{-5}$ is apparent in the strain energy over time (left column) and the spectra of the strain energies (right column).

3.6.4. Checking results at higher resolution. As an additional verification of our results, we ran a select number of simulations at the original size of diffusion, $\nu = 5 \cdot 10^{-4}$, and twice the resolution. That is, on a 1024×256 grid. Our objective was to verify the values of Wi around which period doubling bifurcations occur.

Testing began with the end states of simulations at our standard resolution (512×128), with Wi near bifurcation points. The resolution was then doubled and the simulations continued. In all cases, we found the expected periods of oscillations which support our conclusions. These tests are summarized in Table 3.2

Wi	Simulation Length	Period at standard resolution	Period at high resolution
27.04	370	7.66	7.66
27.12	380	7.72	7.72
27.128	500	15.47	15.46
27.216	500	15.73	15.72
27.224	500	31.48	31.46
27.296	500	31.74	31.73

TABLE 3.2. Verification of Wi values around period doublings with high resolution simulations.

3.6.5. Stability region in the ν-Wi plane. Linear stability analysis reveals that as Wi is increased, viscoelastic Kolmogorov flow first loses stability and later regains it. The critical values of Wi where stability changes depend on the amount of stress diffusion. In Fig. 3.14 we report the region in the $\nu - \text{Wi}$ plane where the flow is linearly unstable. Diffusion is generally stabilizing (the exception being small amounts of diffusion resulting in an increase in the upper critical Wi), and above $\nu \approx 1.17 \cdot 10^{-2}$, the instability disappears entirely.

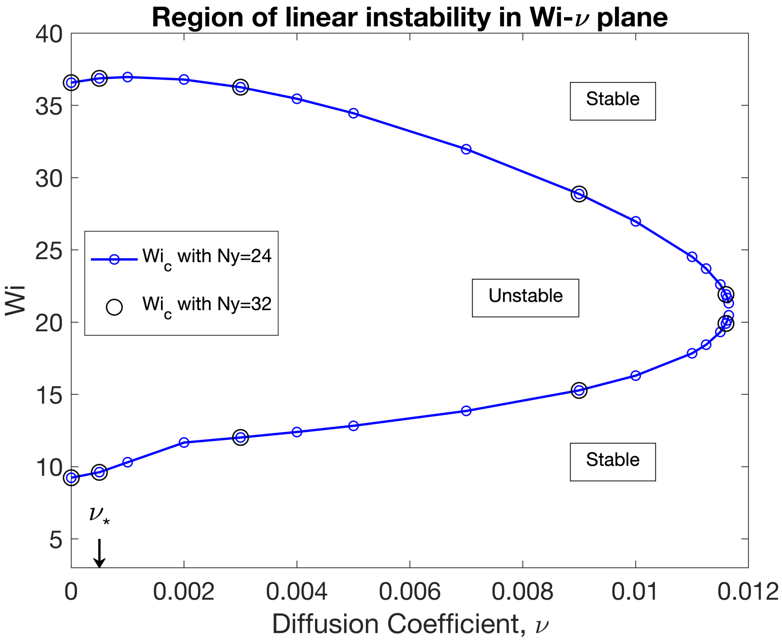


FIGURE 3.14. Region of instability for viscoelastic Kolmogorov flow with $k = 1$. The amount of diffusion used for the results in the main text is marked on the horizontal axis, $\nu_* = 5 \cdot 10^{-4}$

3.6.6. Run-times for non-periodic data. Wi = 27.2976 is the highest value at which we obtained data with oscillations that appear to be steady. Above this value, we ran simulations to extended times to ascertain whether stable periodic solutions would emerge.

Just above Wi = 27.2976, simulations were run for at least 2500 time units. As a reference point, the 3 largest Wi simulations with steady oscillations had reached steady states before a time of 1600. Two simulations (Wi = 28 and Wi = 32) were run for over 10,000 time units. With these firmly established as aperiodic, the simulations with $28 < \mathrm{Wi} < 32$ were only run for 1000 time units, which was sufficient to generate the desired data. All run times are listed in Table 3.3.

Wi	Simulation Length
27.2992	3200
27.3000	4000
27.3008	3520
27.3040	3000
27.3120	2500
27.3280	4000
27.3440	4000
27.3600	3510
27.4000	3000
27.6000	3000
27.8000	3000
28.0000	10030
28.8000	1000
30.4000	1000
32.0000	10040

Table 3.3. Simulation lengths (in time) for non-periodic data

3.6.7. Results on a longer domain. We carried out a limited number of tests on a domain lengthened to $[4\pi \times \pi/2]$. We found steady TWS consisting of a pair of narwhals at Wi $= 22.40$, shown in Fig. 3.15. To obtain this solution, we used two horizontally arranged copies of a single narwhal, taken from a simulation with domain length of 2π, as an initial condition. This TWS translated to the right stably for over 100 time units, after which the simulation was stopped.

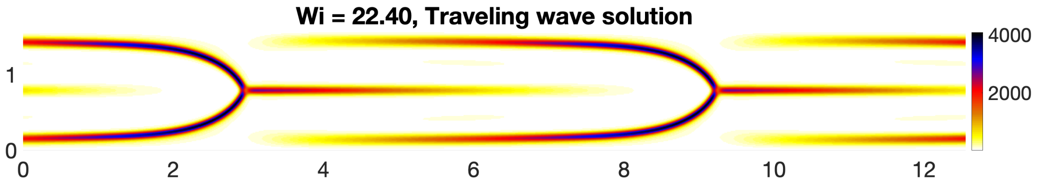


FIGURE 3.15. Strain energy density of a traveling wave narwhal pair on a $4\pi \times \pi/2$ domain.

Upward and downward continuation in Wi was attempted, with interesting results. The TWS readily lost stability and at times appeared to begin oscillating vertically, reminiscent of the oscillations seen in the period doubling cascade. However, these oscillations were not found to be stable.

Stable oscillations did appear in a different manner, however. After the TWS lost stability in some tests, the flow evolved to a state with one narwhal traveling to the left and one to the right, offset vertically from one another. As these narwhals repeatedly pass and interact with one another, they create temporal oscillations in the energies. Unsurprisingly, chaotic flow was also found with this domain length.

We emphasize that our testing with this domain was limited. Uncovering more cohesive results, particularly in relation to the transition to turbulence, would be an interesting avenue of future research.

CHAPTER 4

Viscoelastic Kolmogorov Flow with Multiple Period Forcing

4.1. Overview

In this chapter we study Kolmogorov flow with more than one period in the driving force, specifically two and four periods. We again use the driving force $\boldsymbol{f} = (-A\cos(4y), 0)$, and use domains of $[0, 2\pi] \times [0, \pi]$ and $[0, 2\pi] \times [0, 2\pi]$ to obtain two and four periods, respectively. The analytic solution on these domains is shown in Fig. 2.1a,b. As before, we study a range of Wi. Other parameters will remain the same as those used with $k = 1$.

The collections of flows generated with $k = 1$ and $k = 2$ have certain similarities. In both cases, as Wi increases, we obtain standing waves, traveling waves and oscillating flows which include narwhal structures, and finally intermittency and chaos. Like $k = 1$, the oscillatory states with $k = 2$ also exhibit symmetry breaking and period doubling.

The $k = 2$ case differs from $k = 1$ in several fundamental ways. In particular, with $k = 2$ the analytic solution loses stability at Wi $= 4.34$, while this occurs at Wi $= 9.61$ with $k = 1$. With $k = 2$, we find traveling wave solutions (TWS) which do not take the form of narwhals. Some flows include a pair of narwhals while others include a single narwhal accompanied by a tilted "stress island." Lastly, oscillations arise from the interaction of two narwhals rather than from the self-interaction seen with $k = 1$. Also, we have obtained only one period doubling with $k = 2$, in contrast to the cascade obtained with $k = 1$. However, additional doublings may well be found with further study.

Moving on to $k = 4$, we see additional differences but the results are strongly reminiscent of $k = 2$. In particular, for both $k = 2$ and $k = 4$, the analytic solution loses stability through a double Hopf bifurcations at Wi $= 4.34$, and the unstable modes with $k = 4$ appear to be 2 vertically repeated copies of those with $k = 2$. While the transitions between flow states are not identical, we have not observed fundamentally new states with $k = 4$. Also, the transition from oscillatory to chaotic flow occurs at much closer values of Wi with $k = 2$ and 4 (Wi ≈ 15 and Wi ≈ 13)

as compared to $k = 1$ (Wi $\approx$ 28). In what follows, we first discuss the case of $k = 2$ in detail, proceeding from low to high Wi. We then discuss the $k = 4$ case.

4.1.1. Summary of flow states with $k = 2$. Table 4.1.1 contains a compilation of the various flows with $k = 2$ which will be discussed, as well as the range of Wi where they were obtained in our simulations.

Flow Description	**Range of Wi**
Analytic solution is stable	Wi < 4.34
Standing waves	$4.37 \le$ Wi ≤ 4.55
Symmetric traveling waves	$4.75 \le$ Wi ≤ 5.40
Asymmetric traveling waves	$5.44 \le$ Wi ≤ 7.0
Narwhal with oscillating stress islands	$7.25 \le$ Wi ≤ 7.35
Two narwhals, one repeatedly passing	$7.40 \le$ Wi ≤ 8.0
Two narwhals jockeying, symmetric	$8.50 \le$ Wi ≤ 11.50
Two narwhals jockeying, asymmetric	$11.75 \le$ Wi ≤ 11.85
Two narwhals jockeying, period doubled	$11.85 \le$ Wi ≤ 12.125
Aperiodic flows	Wi > 12.125

4.2. Results with $k = 2$ and low Wi: Standing and traveling waves

4.2.1. Linear stability analysis. We perform linear stability analysis and determine that the analytic solution is stable for Wi < 4.34. The values of Wi where stability is lost at increasing resolutions are reported in Table 4.1, along with the values found with $k = 1$ (these were first presented in Table 3.1). In the stable region, just as with $k = 1$, the strain energy density is high in the areas with the strongest shear rate and constant in the x-direction. Fig. 4.1 shows the solution at Wi $= 4.0$. At Wi $= 4.35$, there are 2 conjugate pairs of eigenvalues with positive real part, indicating a double Hopf bifurcation.

Nx	Wi_c, $k = 2$	Wi_c, $k = 1$
32	2.972	8.169
48	4.208	11.194
64	4.346	9.619
96	4.344	9.615
128	4.345	9.613
192	-	9.614

TABLE 4.1. Results of linear stability analysis, searching for critical Wi with $\nu = 5 \cdot 10^{-4}$. The analytic solution switches from linearly stable to unstable as the value of Wi increases past Wi_c. Values were found using a binary search with a tolerance of $\pm 5 \cdot 10^{-4}$.

.

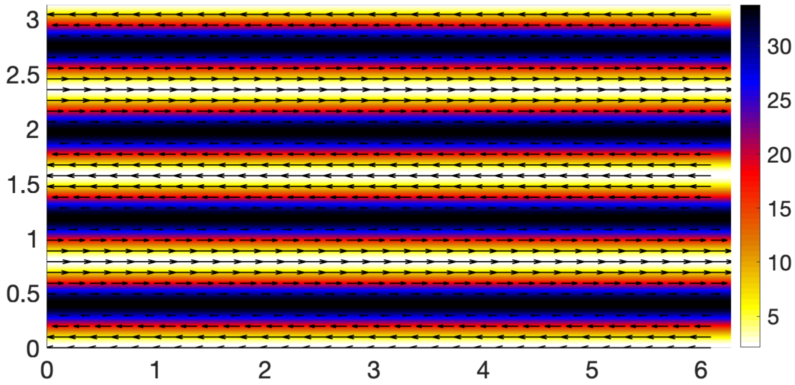


FIGURE 4.1. Strain energy density of the analytic solution with $k = 2$ at $\mathrm{Wi} = 4.0$. The velocity field is displayed with overlayed arrows.

4.2.2. Standing waves. In a small range ($4.37 \leq \mathrm{Wi} \leq 4.55$), perturbations to the analytic solution lead to standing waves in the stress and associated oscillations in the strain and kinetic energies. Similar dynamics with $k = 1$ were found with $9.65 \leq \mathrm{Wi} \leq 11.2$. The $k = 2$ energy oscillations are displayed in Fig. 4.2 and the strain energy density of these standing waves is shown in Fig. 4.3 as it evolves over time. In the standing wave regime, the stress bands seen in the analytic solution (Fig. 4.1) are still present, but no longer uniform in the x-direction. The stress alternately concentrates and dissipates over time at different locations in each band. The top and bottom half of the domain are horizontally shifted copies of one other. That is, $\mathbf{C}(x, y) = \mathbf{C}(x + \pi/2, y + \pi/2)$. Within each half of the domain, the two stress bands oscillate out of phase by $\pi/2$ from one another,

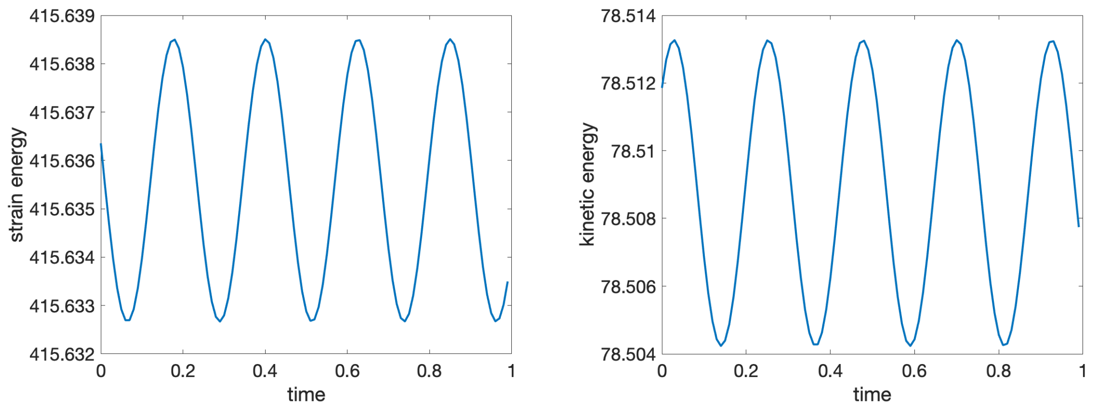


FIGURE 4.2. Oscillations in the strain and kinetic energy of standing waves at Wi = 4.37.

which is apparent in the right two columns of Fig. 4.3. At Wi = 4.37, the system oscillates with a period of 0.90. Due to the symmetries present in the flow, the oscillations in the energies have $1/4^{th}$ this period, or 0.225.

4.2.3. Modal analysis. As with the $k = 1$ case, we perform a modal analysis of the standing wave regime. See section 3.6.2 for the $k = 1$ analysis and section 2.3.8 for the methods used. In the standing wave regime, the unstable modes of the linearized system account for most of the deviation from the analytic solution that is seen in simulations of the full, nonlinear system. The trace of the unstable modes is shown in Fig. 4.4.

Fig. 4.5 shows the trace of the deviation from the analytic solution, denoted $\mathrm{tr}\mathbf{C}_\delta$, at one moment in time from a simulation at Wi = 4.37, as well as the best (in a least squares sense) approximation to $\mathrm{tr}\mathbf{C}_\delta$ consisting of a linear combination of the unstable modes. For this snapshot, the approximation has a relative error of 21.0% measured with the Frobenius norm and 28.2% measured with the max norm. Because we are approximating a solution to the nonlinear system with modes from the linearization, we do not expect these norms to be small. They merely give some indication that, because the standing waves exist near enough (in Wi) to the linear instability, much of the deviation from equilibrium is due to linear terms.

Because the standing waves are time dependent solutions, the deviations and approximations will change in time. Fig. 4.6 displays the coefficients used and the errors in the approximation over one period of oscillation.

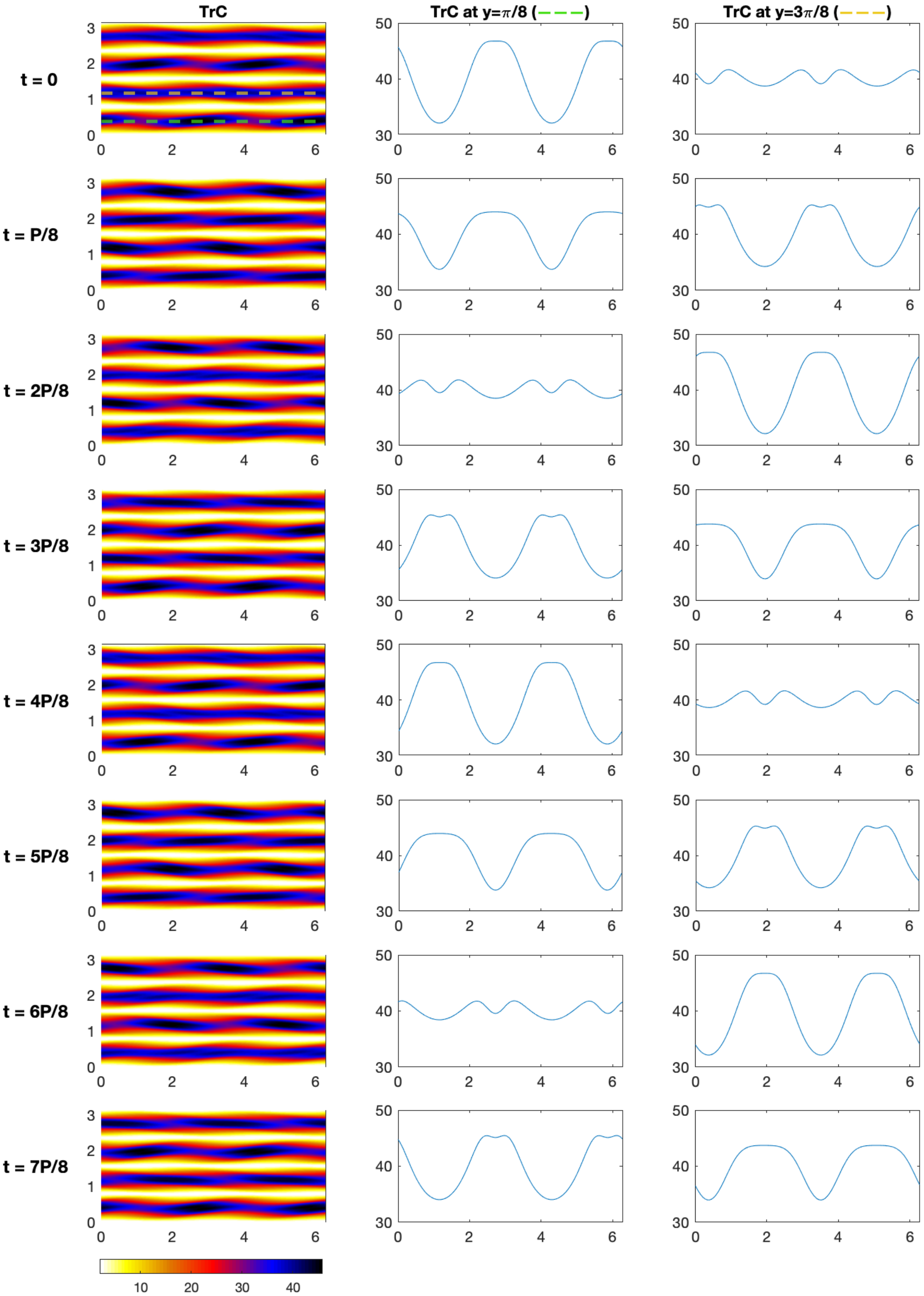
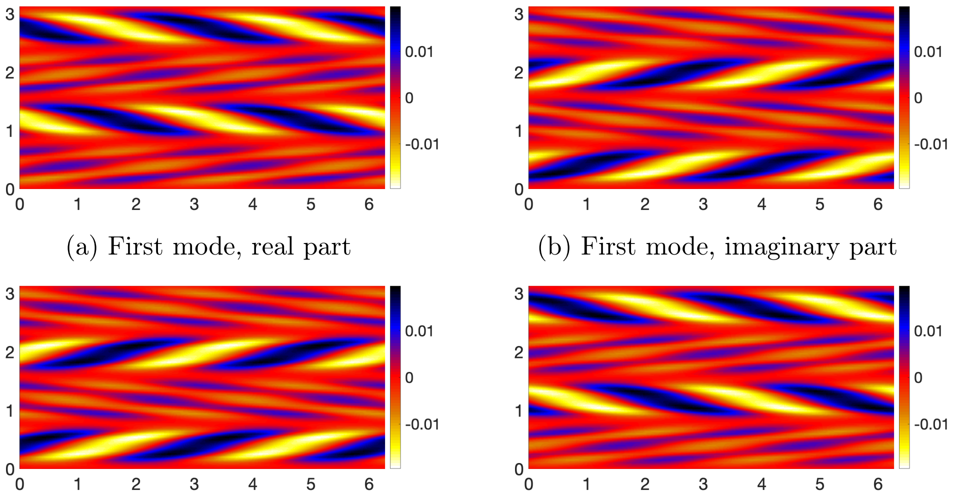


FIGURE 4.3. Standing wave at Wi = 4.37, which has period **P** = 0.90. Time progresses from top to bottom through one period. tr**C** is shown in the left column. The middle and right columns display slices of tr**C** along horizontal lines marked in the top left panel.



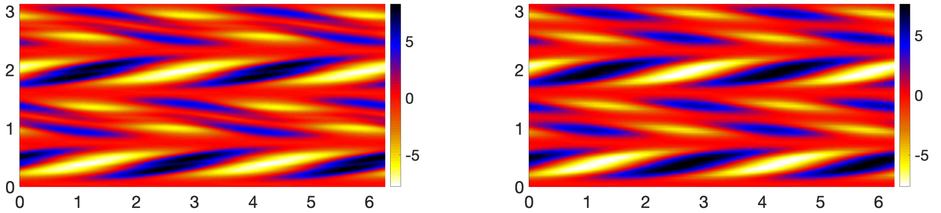
(a) First mode, real part

(b) First mode, imaginary part

(c) Second mode, real part

(d) Second mode, imaginary part

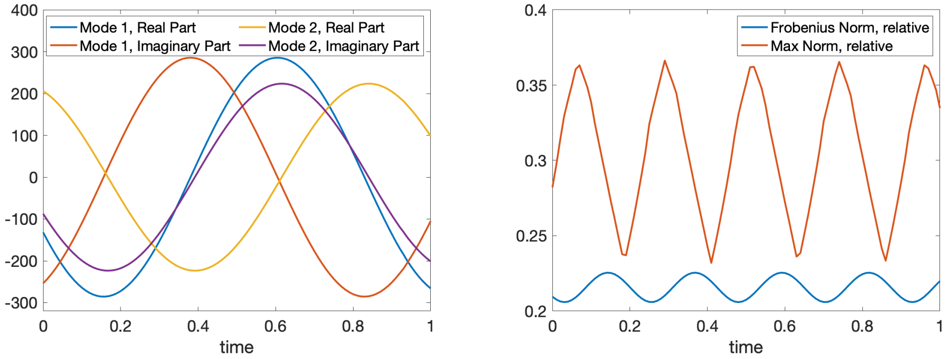
Figure 4.4. tr**C** of the unstable modes of the linearized system at Wi = 4.37.



(a) tr$\mathbf{C}_\delta$

(b) Approximation to tr$\mathbf{C}_\delta$

Figure 4.5. Deviation and approximation at $t = 0$ with Wi = 4.37. This point in time corresponds to the top row in Fig. 4.3 and the first data points in Fig. 4.6.



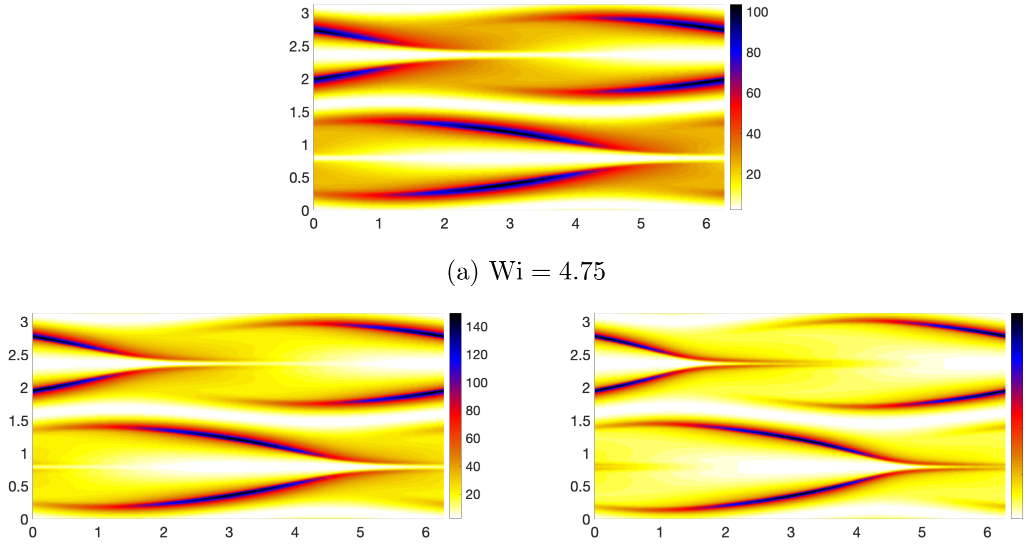
(a) Coefficients for approximations

(b) Errors in approximations

Figure 4.6. Coefficients and errors in modal approximation over time. The errors oscillate with $1/4^{th}$ the frequency of the coefficients due to symmetry of the system.

4.2.4. First traveling wave solutions: Tilted stress islands. Above the standing wave threshold, small perturbations to the analytic solution lead to traveling wave solutions (TWS), though in contrast to $k = 1$, these do not take the form of narwhals. These traveling waves appear as pairs of tilted "stress islands," and bear some resemblance both to narwhals and the standing wave solutions. These TWS were found in the range $4.75 \leq \mathrm{Wi} \leq 5.4$, and examples are shown in Fig. 4.7. Whether the structures move to the left or right appears to be random. This is expected given the symmetric forcing in Kolmogorov flow and the random perturbations used in this study. For clarity of presentation, we only show flows with structures moving to the right in this work.

As Wi is increased, the stress islands attain higher peak stresses, while the stress in surrounding areas only slightly increases. Each pair becomes nearly connected, as we can see along $y = \pi/4$ and $y = 3\pi/4$ in Fig. 4.7. In this TWS regime, as we progress upwards in Wi the pairs of stress islands begin to resemble the narwhal structures seen in Chapter 3 and seen later in this chapter, for example in Fig. 4.10. In contrast to those structures however, in the TWS seen in Fig. 4.7, each pair of stress islands is separated by a thin horizontal region of low stress, along $y = \pi/4$ for the lower pair and along $y = 3\pi/4$ for the upper pair. It is unclear if there is any definite value of Wi in this progression where the stress islands become narwhals. However, there is a clear flow transition above $\mathrm{Wi} = 5.4$ where the flow loses its vertical symmetry.



(a) Wi = 4.75

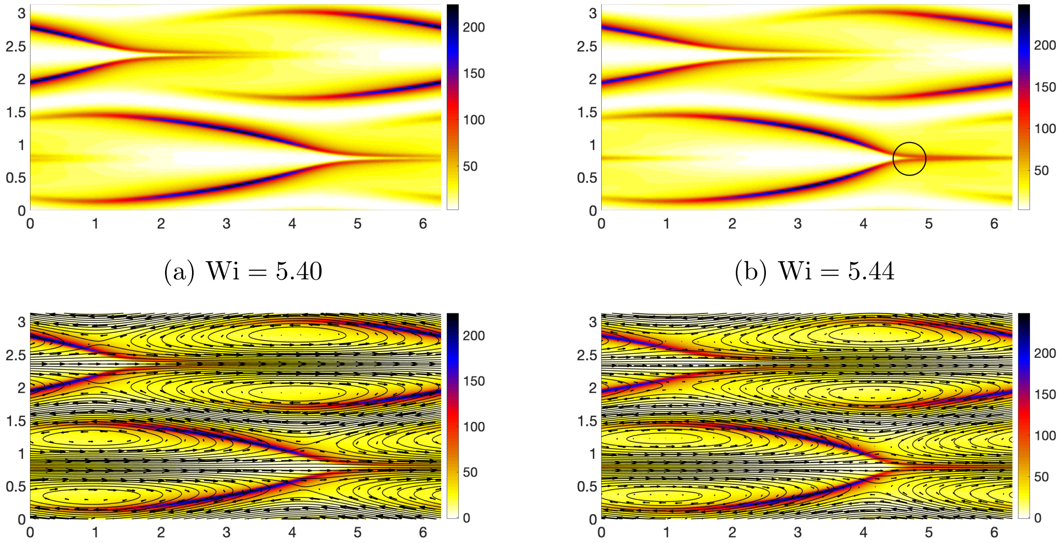
(b) Wi = 5.00

(c) Wi = 5.40

FIGURE 4.7. The first traveling waves found with $k = 2$ in our progression upwards in Wi. These structures translate to the right at a uniform speed.

4.3. Results with $k = 2$ and moderate Wi: Narwhals and oscillations

4.3.1. Transition to asymmetric traveling waves. Numerical continuation from $\mathrm{Wi} = 5.4$ to $\mathrm{Wi} = 5.44$ causes the traveling waves to transition to a new state, as seen in Fig. 4.8. The flow loses its vertical symmetry, with one pair of stress islands (in the bottom half of Fig. 4.8b, d) reaching a higher peak stress. This pair of stress islands is no longer separated by a horizontal region of low stress, having connected around, in this snapshot, the coordinates $(4.7, 0.78)$. This location is circled in Fig. 4.8b. The other pair of stress islands has a lower peak stress, and separates slightly, becoming less narwhal-like.



(a) $\mathrm{Wi} = 5.40$

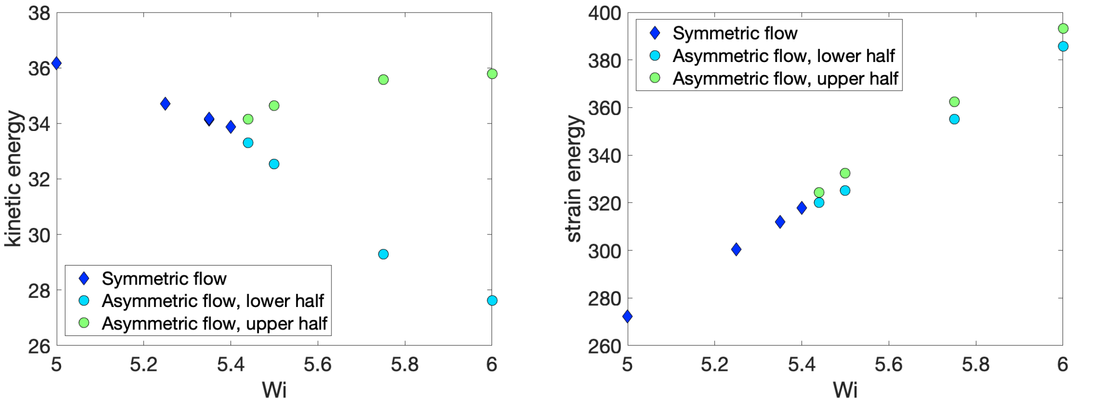
(b) $\mathrm{Wi} = 5.44$

(c) $\mathrm{Wi} = 5.40$

(d) $\mathrm{Wi} = 5.44$

FIGURE 4.8. Comparison of traveling wave solutions (a,c) before and (b,d) after transitioning to an asymmetric state with only one dominant structure, which in this case occurs in the bottom half of the domain. The circle in (b) highlights the location where the stress islands connect, a feature which distinguishes the TWS before and after this transition. Streamlines computed in traveling wave coordinates have been overlayed on (c) and (d).

This transition is hard to identify visually, and inspecting streamlines of the flow does not reveal clear topological differences between any of the structures in Fig. 4.8. Thus, we do not have enough information to assert which, if any, of these structures are or are not narwhals. Despite the difficulty in distinguishing these states, the transition is quite distinct if we inspect the energies of the flow, specifically focused on the loss of vertical symmetry. In Fig. 4.9 we show the energies taken over the top and bottom half of the domain for several values of Wi around this transition. Prior



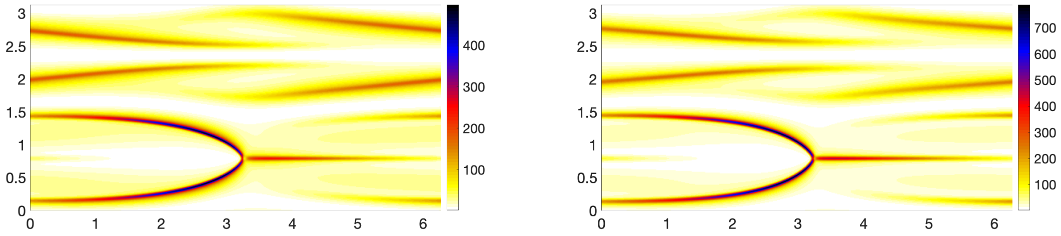
(a) Kinetic Energy

(b) Strain Energy

FIGURE 4.9. The energies of traveling wave solutions, taken separately on the upper and lower halves of the domain. In all of the asymmetric flows used here, the lower half contains the more narwhal-like structure.

to the transition, the energy is identical on the top and bottom halves. After the transition, the more narwhal-like structure has both lower strain energy and lower kinetic energy than the other structure.

As Wi is increased beyond this transition at Wi = 5.44, the difference between the structures in the top and bottom parts of the domain becomes more pronounced. At Wi = 6 and 7, shown in Fig. 4.10, the bottom of the domain contains a structure which clearly resembles the narwhal structures seen with $k = 1$, while the top contains a pair of stress islands. These flows are still traveling wave solutions, translating the the right with a uniform velocity.



(a) Wi = 6.0

(b) Wi = 7.0

FIGURE 4.10. Traveling Wave Solutions consisting of both narwhals and tilted stress islands.

4.3.2. Emergence of oscillations. Upward continuation from Wi = 7 to Wi = 7.25 brings us to the end of the traveling wave regime, and we progress through a variety of oscillatory states.

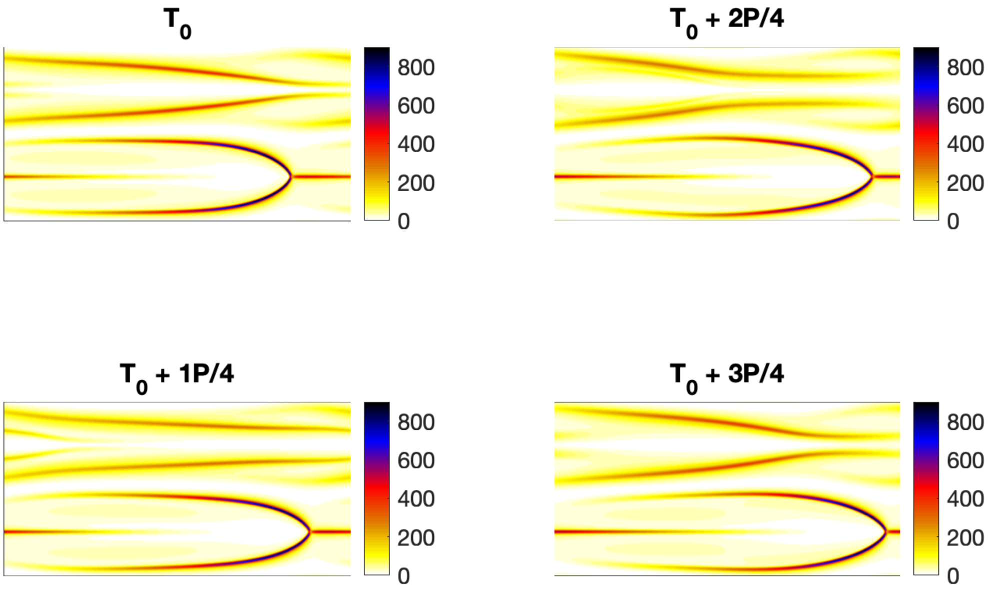
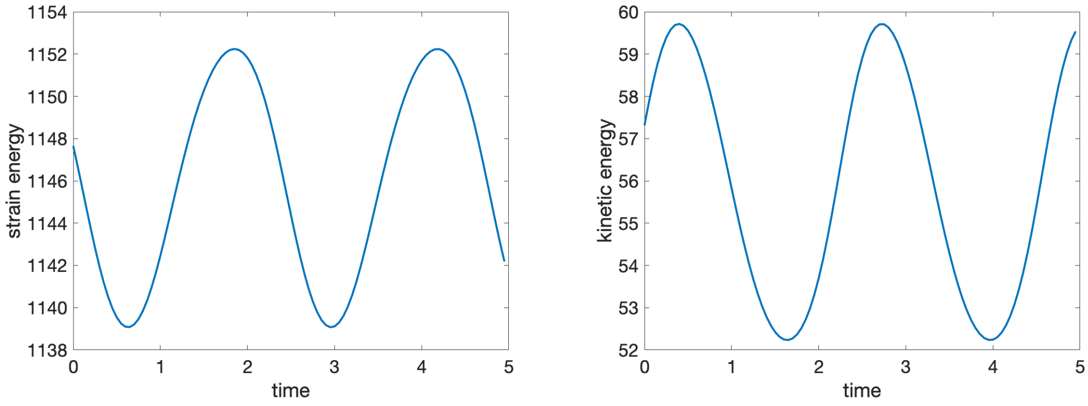


FIGURE 4.11. Snapshots of tr**C** of the oscillatory solution found at Wi = 7.35. The period of oscillation is $P = 2.33$. The average velocity of the narwhal (0.541) has been subtracted. The energies of this flow are plotted in Fig. 4.12.

At Wi = 7.25 and 7.35, we retain the structures of a narwhal and tilted stress islands, all moving to the right. However, the angle and the magnitude of stress in the stress islands begins to oscillate in time. This gives steady oscillations in the energies, with a period of 2.27 at Wi = 7.25 and 2.33 at Wi = 7.35. tr**C** in the flow with Wi = 7.35 is visualized in Fig. 4.11 and the oscillations in the energies are shown in Fig. 4.12.



(a) Strain Energy

(b) Kinetic Energy

FIGURE 4.12. Energies of the flow with Wi = 7.35.

4.3.3. Dynamics with two narwhals. Another upward continuation from Wi = 7.35 to Wi = 7.40 leads to the stress islands forming a second narwhal which repeatedly passes the original narwhal. Both narwhals are traveling to the right, but at unsteady speeds. Similar behavior, with one narwhal repeatedly passing the other as both travel, was also observed at Wi = 7.50 and 8.0. tr**C** and the energies at Wi = 7.5 are shown in Figs. 4.13 and 4.14.

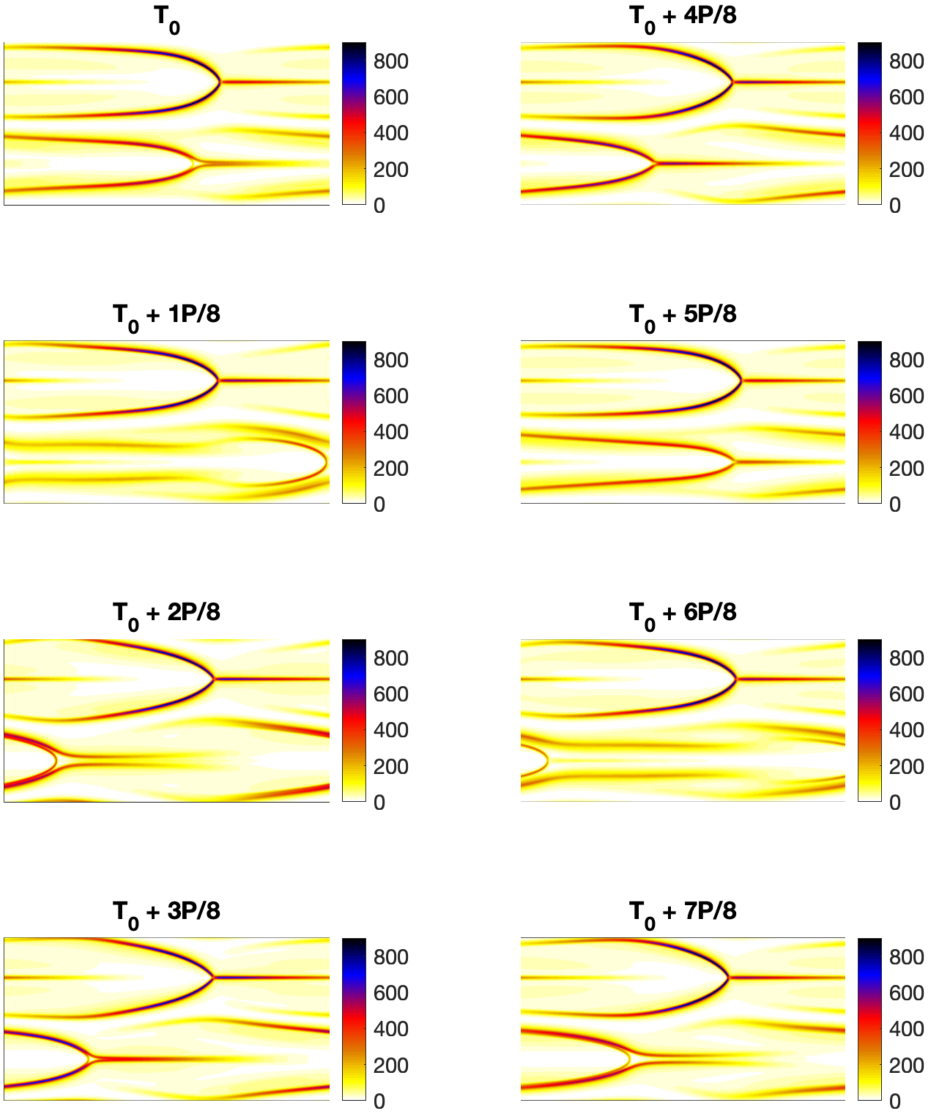


FIGURE 4.13. tr**C** with Wi = 7.50. The lower narwhal repeatedly passes the upper narwhal, alternating between a slow pass (first 6 panels) and a fast pass (last 2 panels). The average horizontal velocity (0.592) of the narwhals has been subtracted in making these images.

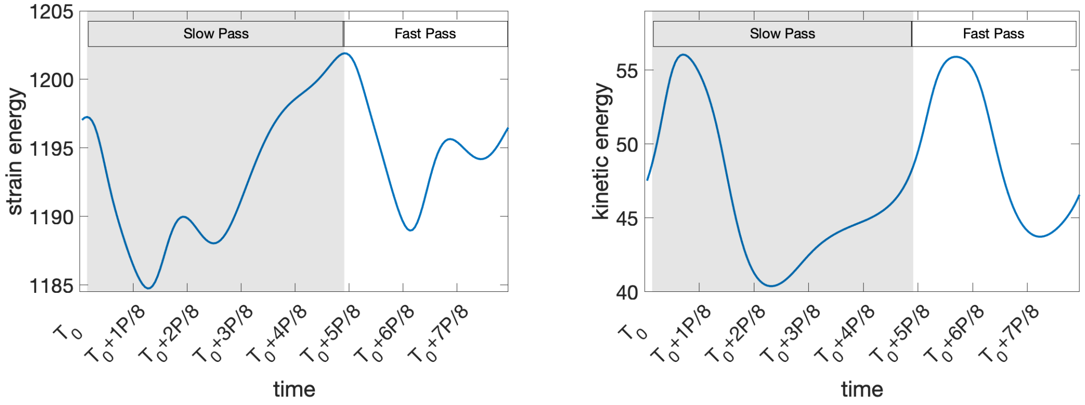


FIGURE 4.14. Energies of the flow with Wi = 7.5. The time labels correspond to the panels in Fig. 4.13

.

At Wi = 7.4 and 7.5, though one narwhal passes the other in a periodic manner, the time interval between passes is not constant. We have observed that the times at which the narwhals are vertically aligned coincide with peaks in the strain energy. We use the time intervals between these peaks to measure the time taken for one pass. At both Wi = 7.4 and 7.5, the flow alternates between a ‘fast pass’ taking approximately 3 time units and a ‘slow pass’ taking about 5 time units. In contrast, in the flow at Wi = 8 each pass takes the same amount of time (5.52 time units).

Upward continuation from Wi = 8 to Wi = 8.5 brings about another change. The flow still contains two traveling narwhals, but now both narwhals alternatively pass one another, a behavior we refer to as ‘jockeying.’ An example of jockeying at Wi = 9.0 is shown in Fig. 4.15, and Fig. 4.16 inspects the oscillations of the energies and the spectrum of the strain energy. The energies have a period of 5.46. For the pair of narwhals to progress through a full cycle (i.e. two passes take place so they return to their original relative positions), the period is 10.92. The shorter period of the energies is simply a result of a vertical symmetry in the narwhal dynamics. That is, the energies cannot distinguish between times with the top or bottom narwhal is ahead of the other. At Wi = 11.75 and 11.85, the symmetry is broken, and the period of the energies now matches that of a full cycle of narwhal passing.

4.3.4. Period doubling. In the range just discussed, $11.75 \leq \mathrm{Wi} \leq 11.85$, jockeying behavior was observed in which the energies of the flow oscillated with periods of 10.6. At Wi $\approx$ 11.85, a

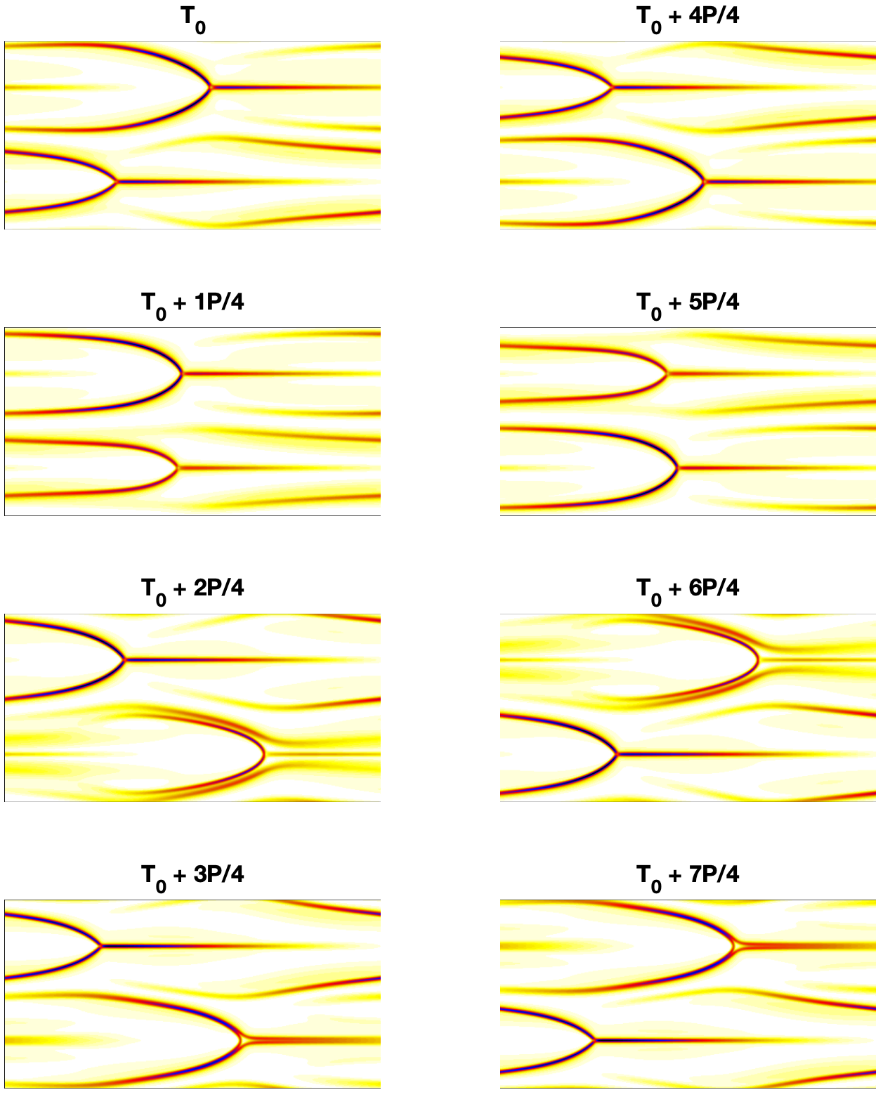


FIGURE 4.15. Jockeying narwhals with Wi = 9.0. The average horizontal velocity (0.956) of the narwhals has been subtracted. The period of the energies is P = 5.46. Notice the vertical symmetry is apparent - each panel is a vertical reflection of its neighbor, which is one period away in time.

period doubling occurs. Above this value, narwhals continue to exhibit jockeying behavior, with steady oscillations in the energies. However, in the range $11.85 \leq \text{Wi} \leq 12.125$, these oscillations have periods of roughly 21 time units. Notice that the value Wi = 11.85 appears in both of the above inequalities. Indeed, in separate simulations at this value of Wi, we observed both periods

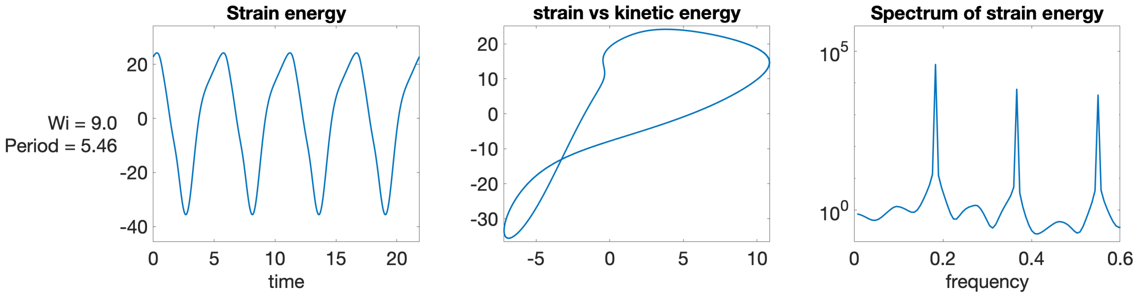


FIGURE 4.16. Data from jockeying narwhals at Wi = 9.0, where a vertical symmetry exists in the flow. A full cycle of 2 passes takes 10.92 time units.

of 10.6 and 21.2. The shorter period was found via upwards continuation from Wi = 11.75, and the longer period was found with downwards continuation from Wi = 12.0. This single point of bistability is indicative of a possible subcritical bifurcation, and in a future study it would be of interest to probe this further. Of even greater interest would be a search for further period doublings or triplings as Wi is increased, as seen with $k = 1$. It is entirely possible that the doubling shown here is merely the first in a cascade leading to chaos. Energies, Lissajous curves, and spectra before and after the doubling are shown in Fig. 4.17. To summarize, jockeying narwhals were obtained in the range $8.5 \leq \text{Wi} \leq 12.125$, and two transitions were observed within this range. Reminiscent of the $k = 1$ case, these two transitions are a symmetry breaking transition followed by a period doubling bifurcation.

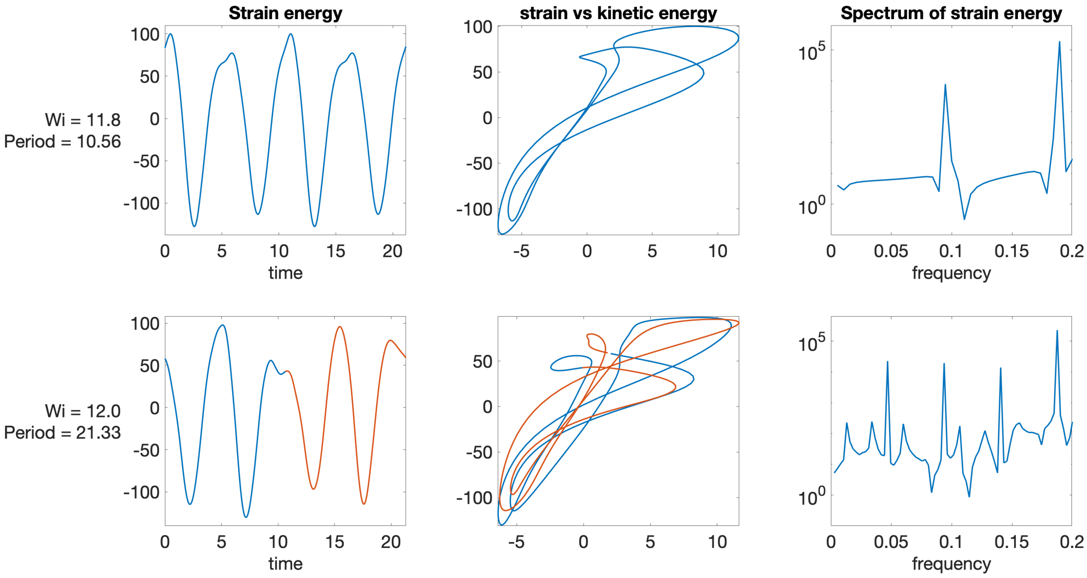


FIGURE 4.17. Data before (top row) and after (bottom row) the period doubling.

4.4. Results with $k = 2$ and high Wi: Transition to chaos

Moving to Wi > 12.125 brings us into a regime where dynamics begin to appear aperiodic, similar to that seen in the $k = 1$ case above Wi ≈ 27.3. In the lower part of this range, it is possible that very long tansients or dynamics with very long periods are obscuring some periodic behavior. For example the energies at Wi $= 12.25$, shown in the top row of Fig. 4.18, may still have a dominant period of approximately 20, though the oscillations never became steady in a simulation of 1500 time units. Though it is no longer steady, this flow appears to retain the jockeying behavior seen at lower values of Wi.

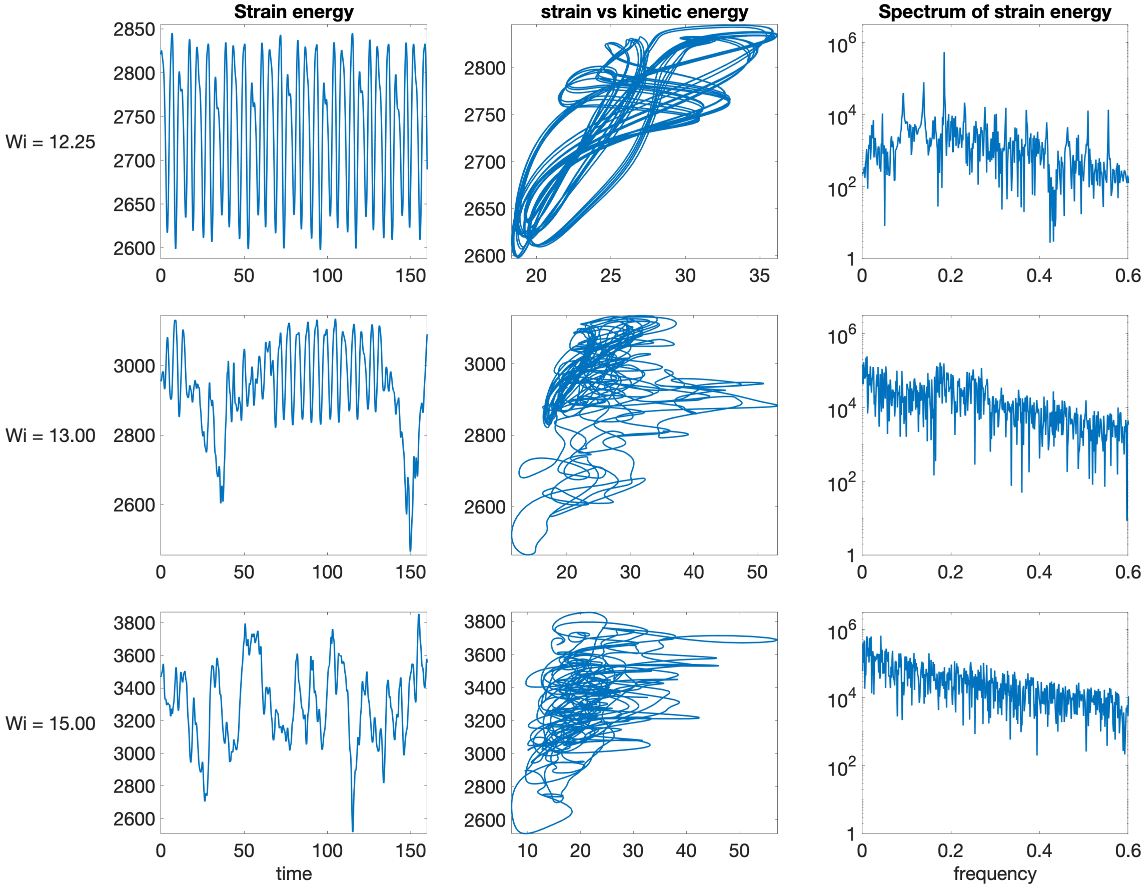


FIGURE 4.18. Transitioning to chaos through values of Wi above the loss of steady oscillations. Early in this regime at Wi $= 12.25$ (top row), oscillations in the energies are still nearly periodic and the flow contains jockeying narwhals. At Wi $= 13$ (middle row), the narwhals intermittently break apart, which is associated with a drop in strain energy. Two such drops are visible in the data shown here. Finally, at Wi $= 15$, the flow appears fully chaotic.

At Wi = 12.75 we first begin to see the narwhal structures intermittently break apart, each time associated with a drop in strain energy and rise in kinetic energy. When the narwhals break apart, the flow becomes more disorganized, at times featuring narwhals pointing in opposite directions or no narwhals at all. Wi = 13 gives similar dynamics, and representative snapshots as the flow moves into and out of a disorganized state are shown in Fig. 4.19, along with the energies of the flow.

At Wi = 15, disorganized flow is the dominant state. Snapshots of the flow are shown in Fig. 4.20. While narwhal-like structures are still visible at times, they rapidly break apart and move in unpredictable ways. The spectrum, shown in the bottom right panel of Fig. 4.18, shows excitement across a wide spectrum of frequencies, does not contain distinct peaks, and decays exponentially with frequency. Such a flow is typically classified as elastic turbulence. Similar chaotic states were simulated up to Wi = 18.4.

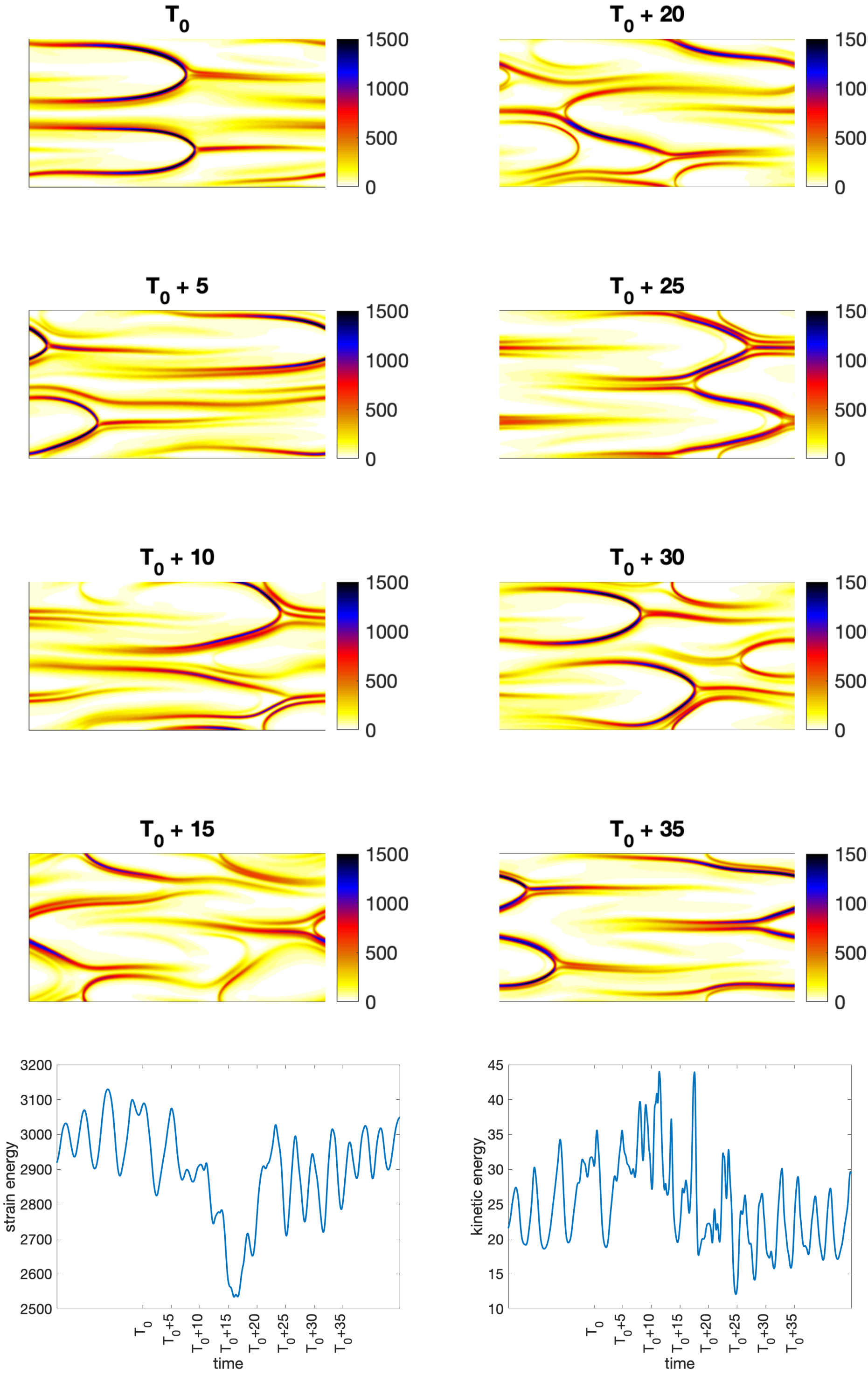


FIGURE 4.19. At Wi = 13, the flow contains two jockeying narwhals at most times, but this behavior is regularly interrupted by brief disorganized states. Such an interruption is seen here, between approximately T_0+10 and T_0+20. Two narwhals are visible before $T_0 + 10$ and after $T_0 + 20$, but break apart in between. In the bottom row we see a drop in strain energy and rise in kinetic energy associated with the disorganized state.

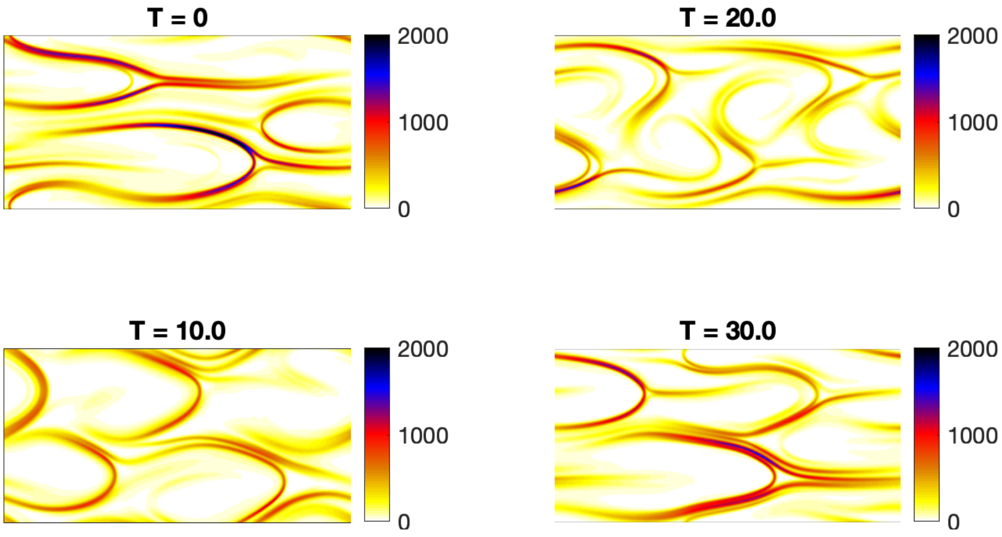


FIGURE 4.20. Snapshots in time of fully chaotic flow at Wi = 15.

4.5. Introduction to the $k = 4$ case

A broad introduction to the cases of $k = 2$ and $k = 4$ is given in Section 4.1. We reiterate that the $k = 4$ flows are simulated with a driving force of $\boldsymbol{f} = (-A\cos(4y), 0)$ on a $[0, 2\pi] \times [0, 2\pi]$ domain, shown in Fig. 2.1a. It should also be noted that our exploration with $k = 4$ is less thorough than the lower k values, for a few reasons. For one, an exploration of this flow was done by Berti and Boffetta [8], though their study was done with Re = 1. Additionally, the $k = 1$ and $k = 2$ cases contain many intriguing differences, and the discovery of a route to chaos involving a single narwhal was an exciting and novel result. $k = 4$ on the other hand appears in many ways similar to $k = 2$ and is of course more computationally expensive. Thus, we chose to prioritize simulations with lower k values.

The results presented here with $k = 4$ are useful as a comparison to those lower values, and perhaps as a bridge to other, more complicated flows. However, they should not be viewed as a complete catalogue of flows. To some extent, the same can be said of our lower k results, but in those cases a significant effort was made to perform upward and downward continuations to probe for additional flow states existing at Wi values in between the states that we have documented, as well as to establish a reasonably accurate range of Wi where each flow type exists. In contrast, our continuations with $k = 4$ often took relatively large steps in Wi and at times these steps were only in the upward direction.

4.6. Results with $k = 4$ and low Wi: Standing and traveling waves

4.6.1. Analytic solution and linear stability analysis. Just as in the cases of $k = 1$ and $k = 2$, we have an analytic solution that is stable for small Wi. In this solution, the strain energy density is high in the areas with the strongest shear rate and constant in the x-direction. Fig 4.21 shows the solution at Wi = 4.0.

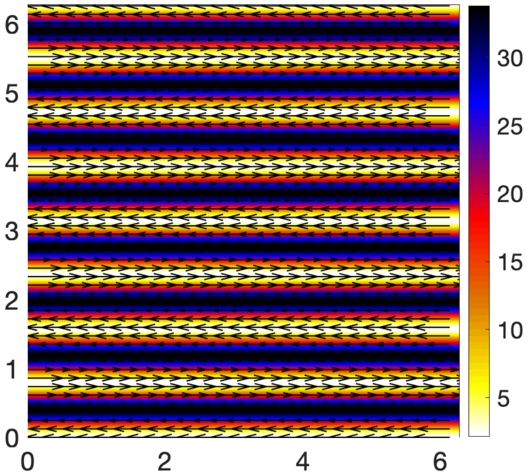


FIGURE 4.21. Strain energy density of the analytic solution with $k = 4$ at Wi = 4.0. The velocity field is displayed with overlayed arrows.

We perform linear stability analysis to determine at what value of Wi the analytic solution loses stability, with results shown in Table 4.2, along with the results for smaller k, first shown in Tables 3.1 and 4.1. The values of Wi_c with $k = 4$ and $k = 2$ agree up to the tolerance used in our tests. Again we find 2 conjugate pairs of eigenvalues crossing the imaginary axis indicating a double Hopf birfucation. The higher resolutions for $k = 2$ and $k = 4$ were not run due to computational cost.

Nx	$\mathrm{Wi}_c, k = 4$	$\mathrm{Wi}_c, k = 2$	$\mathrm{Wi}_c, k = 1$
32	2.973	2.972	8.169
48	4.206	4.208	11.194
64	4.345	4.346	9.619
96	4.345	4.344	9.615
128	-	4.345	9.613
192	-	-	9.614

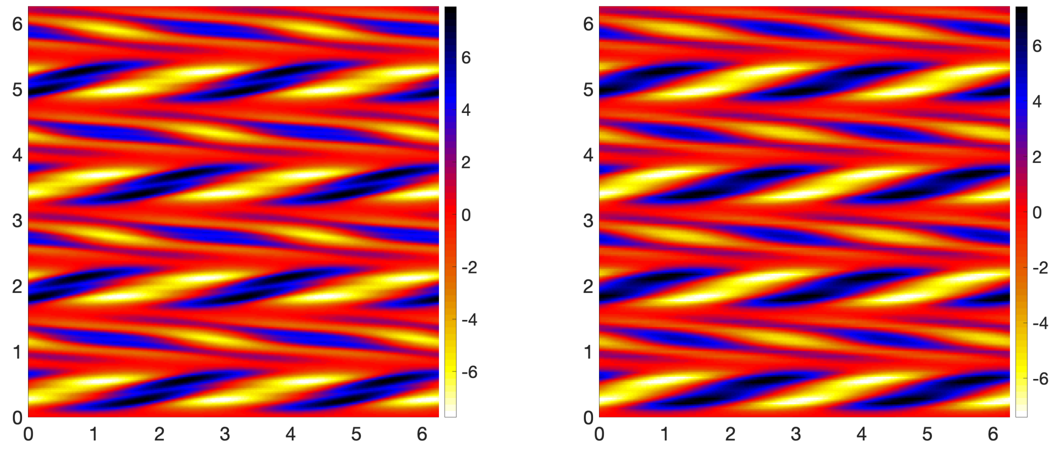
TABLE 4.2. Results of linear stability analysis, searching for critical Wi with $\nu = 5 \cdot 10^{-4}$. The analytic solution switches from linearly stable to unstable as the value of Wi increases past Wi_c. Values were found using a binary search giving a tolerance of $\pm 5 \cdot 10^{-4}$.

.

4.6.2. Standing waves. Standing waves were obtained from a perturbation to the analytic solution at Wi = 4.4. This simulation was continued downwards to Wi = 4.37, and the two simulations had very similar dynamics. The standing wave at Wi = 4.37 has the same pattern in tr**C** as in the $k = 2$ case, modulo a horizontal shift. Details are given in sections 4.2.2 and 4.2.3 for $k = 2$. The $k = 4$ standing wave appears to simply be two copies of the $k = 2$ wave, stacked in the y-direction. As we did with $k = 2$, we will inspect the solutions and build modal approximations at Wi = 4.37.

4.6.3. Modal analysis. As one might expect based upon the previous section, the modal analysis results for $k = 4$ are nearly identical to those for $k = 2$, which are presented in section 4.2.3. We omit visualizations of the unstable modes, which are merely vertically stacked copies of the modes shown in Fig. 4.4. Though they are also essentially stacked versions of the $k = 2$ case, we will include the deviations in tr**C** and the modal approximation to them, shown in Fig. 4.22. We also include plots of the coefficients used in the approximations to $\mathrm{tr}\mathbf{C}_\delta$ and the errors in these approximations over time, both in Fig. 4.23.

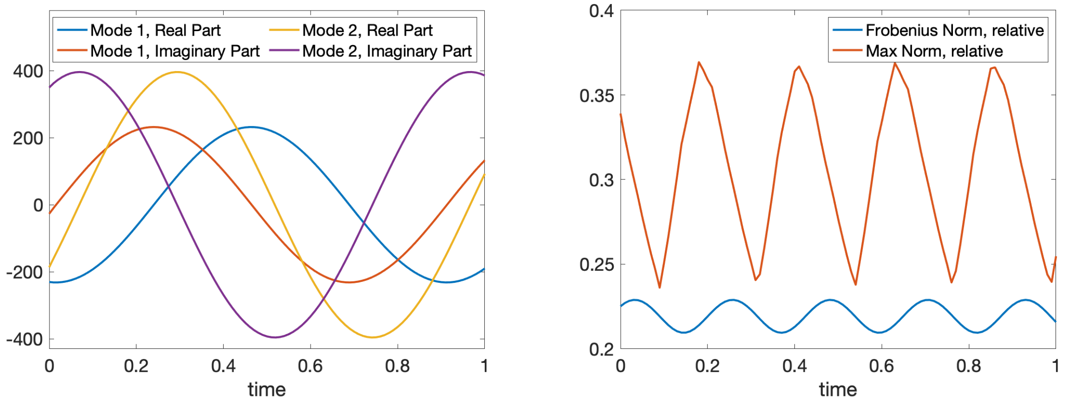
4.6.4. Traveling wave solutions: Tilted stress islands and emergence of narwhals. As with $k = 2$, above the standing wave regime we find pairs of traveling wave stress islands. With $k = 4$, these were found with $4.60 \leq \mathrm{Wi} \leq 5.5$. While the analytic solution is unstable in this range, growth caused by perturbations is very slow near Wi = 4.6. To avoid this slow growth, the



(a) $\mathrm{tr}\mathbf{C}_\delta$

(b) Approximation to $\mathrm{tr}\mathbf{C}_\delta$

FIGURE 4.22. Deviation and approximation at $t = 0$ with $\mathrm{Wi} = 4.37$. This corresponds to time $t = 0$ in Fig. 4.23.



(a) Coefficients for approximations

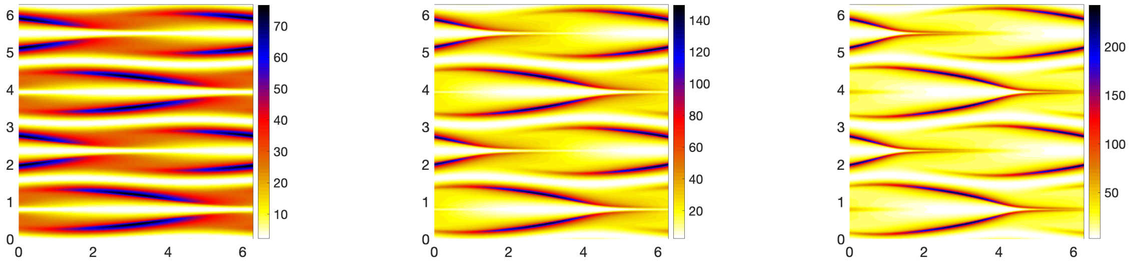
(b) Errors in approximations

FIGURE 4.23. Coefficients and errors in modal approximation over time. The errors oscillate with $1/4^{th}$ the frequency of the coefficients due to symmetry of the system.

initial condition used for TWS shown here is not a perturbation to the analytic solution. Instead, our initial conditions are a simulation from earlier work, which was carried out with higher stress diffusion ($\nu = 5 \cdot 10^{-3}$) but featured similar TWS. To produce the data shown here, the diffusion was first lowered to $\nu = 5 \cdot 10^{-4}$. From here, we performed a downwards continuation to $\mathrm{Wi} = 4.6$, as well as incremental upwards continuations. Note that the older simulation with $\nu = 5 \cdot 10^{-3}$

began with a perturbation to the analytic solution. We expect that perturbing the analytic solution with $\nu = 5 \cdot 10^{-4}$ would produce the same TWS as shown here.

As Wi is increased, the stress island TWS evolve in a manner similar to $k = 2$, reaching higher peak stresses and more closely resembling narwhals, shown in Fig. 4.24. This continues up to Wi $= 5.5$, above which a transition to an asymmetric flow occurs.



(a) Wi = 4.60

(b) Wi = 5.00

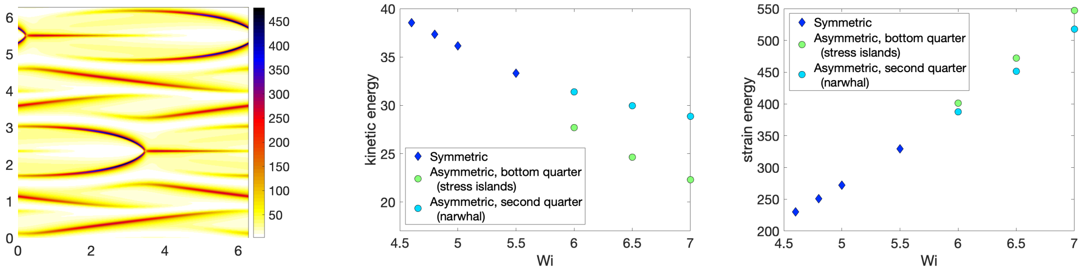
(c) Wi = 5.50

FIGURE 4.24. Strain energy density of traveling wave stress islands at various values of Wi. The structures all translate to the right at constant speeds.

Performing a continuation from Wi $= 5.5$ to Wi $= 6.0$ brings about a transition from a flow with 4 identical traveling wave structures (Fig. 4.24) to a flow with 2 pairs of stress islands and 2 narwhals (Fig. 4.25a). The transition realigns the flow, and at Wi $= 6$ the narwhals are no longer vertically aligned as they were before the transition.

In the $k = 2$ case, a similar transition occurs at a slightly lower Wi, between Wi $= 5.40$ and 5.44 (see Section 4.3.1). It is possible that running longer $k = 4$ simulations and taking smaller steps in Wi would reveal that the transition points are the same for $k = 2$ and 4. Aside from this possible discrepancy, flows with $k = 2$ and $k = 4$ appear to have essentially the same dynamics below the transition. Above the transition, however, they have a definitive difference, owing to the realignment of the $k = 4$ flow, which is possible only with the extra degrees of freedom present with more periods in the driving force. With the realignment, the stress islands as well as the trailing ends of the narwhals have space to extend further in the vertical direction than they would without realigning.

As with $k = 2$, we inspect the energies on subsets of the domain, focusing on the difference between the structures that emerge through the transition. The results are shown in Fig. 4.25b and c. The analogous results for $k = 2$ are shown in 4.9. Note that there is a distinct difference between $k = 2$ and $k = 4$ in these kinetic energy plots. With $k = 2$, the stress islands have higher kinetic energy than narwhals, while the opposite is true for $k = 4$. These TWS consisting of two narwhals and two pairs of stress islands continue to exist as we perform upwards continuations, and we obtained such solutions in the range $6.0 \leq \mathrm{Wi} \leq 8$.



(a) TWS at Wi = 6.0.

(b) Kinetic energy.

(c) Strain energy.

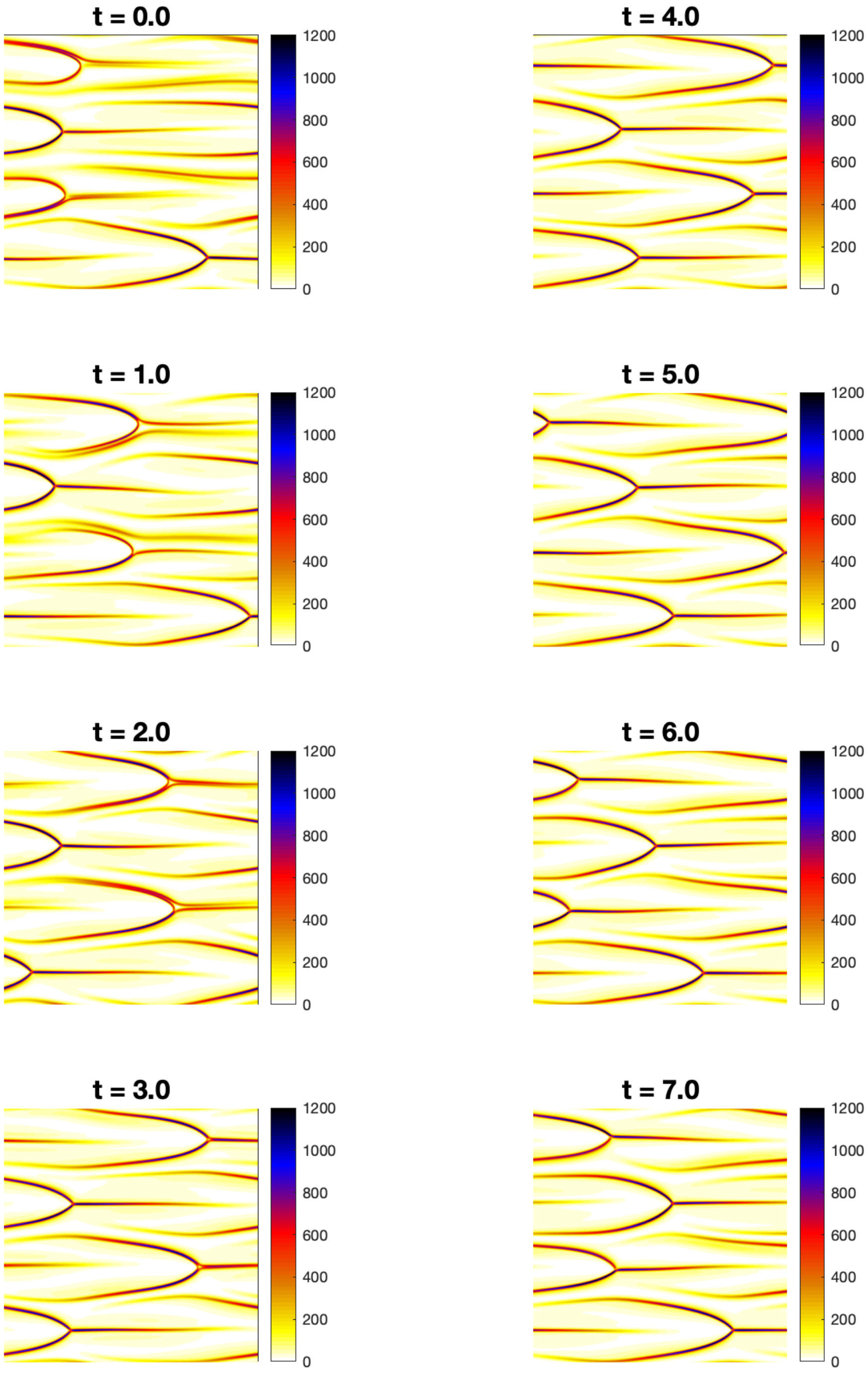
FIGURE 4.25. (a) Shows the TWS after the transition and realignment from a flow with 4 identical structures. The bottom quarter of this flow contains stress islands, and the second quarter contains a narwhal. (b) and (c) show the energies of these subsets of the domain, along with the same measurement at several other values of Wi.

4.7. Results with $k = 4$ and moderate to high Wi

At Wi = 9, we leave the traveling wave regime, and see familiar behavior with some narwhals passing others. We have not observed steady periodic behavior, such as the jockeying seen with $k = 2$, though such a pattern could certainly emerge with $8 < \mathrm{Wi} < 9$, or perhaps eventually emerge at Wi = 9 if the simulation is run for long enough. Fig. 4.26 shows a snapshots during a representative period of time.

Continuation from Wi = 9 to 10 brings about a transition, via a long transient resembling chaotic flow, to a very distinctive state which has the only steady oscillations we have found with $k = 4$. This flow consists of three narwhals and two tilted stress islands, shown in Fig. 4.27 along with the strain energy over time. Unlike all other non-chaotic flows we have encountered, the narwhals point in opposite directions, and the stress islands are separated by a narwhal rather than occurring as a pair in close proximity. These structures all travel to the right (meaning one narwhal travels 'backwards'), though at unsteady speeds.

This state may appear unusual compared to the others we have discussed. One might speculate that it is one of many somewhat random stable arrangements of coherent structures that the flow could arrive at after a chaotic transient. While we cannot rule out other arrangements, we did



t = 0.0

t = 4.0

t = 1.0

t = 5.0

t = 2.0

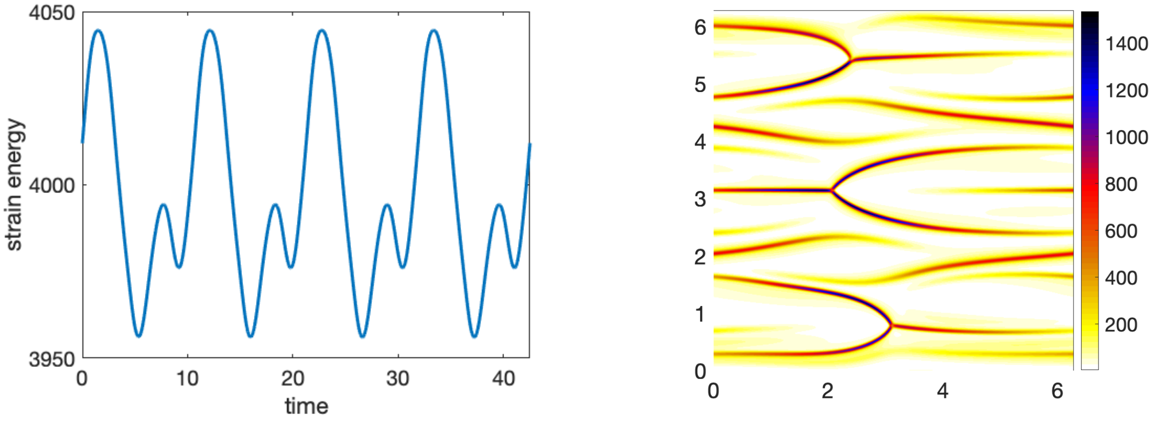
t = 6.0

t = 3.0

t = 7.0

FIGURE 4.26. Snapshots of the flow with Wi = 9. Four narwhals pass one another in an irregular pattern.

obtain this state in three separate simulations with unique initial conditions: once by continuation from Wi = 12.5 (a chaotic state); once by continuation from the flow at Wi = 9 consisting of four oscillatory narwhals (see Fig. 4.26); and once by continuation from an earlier point in time in the Wi $=$ 9 simulation, prior to its transition from a TWS to oscillatory narwhals (it resembled Fig. 4.25a at that point).



(a) Strain energy over time

(b) tr$\mathbf{C}$ at $t = 0$

FIGURE 4.27. The dynamics at Wi $=$ 10. The structures in (b) all travel to the right at unsteady speeds.

We have not thoroughly explored the transitions above Wi $=$ 10, but what data we have suggests the existence of chaotic flows. This is expected given the similarities with the $k = 2$ case and the elastic turbulence at $k = 4$ documented in [8]. A relatively short simulation of 250 time units at Wi $=$ 11 and a longer simulation of 840 time units at Wi $=$ 13 both appear chaotic. The strain energy, Lissajous curve, and spectrum of the strain energy of the Wi $=$ 13 flow are shown in Fig. 4.28, and snapshots of the flow are shown in Fig. 4.29.

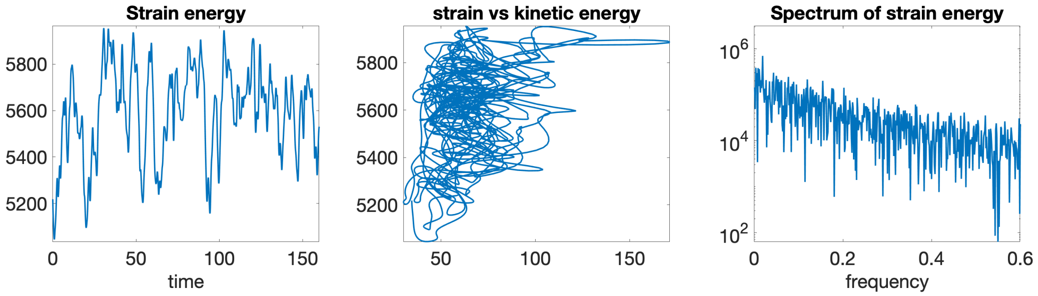


FIGURE 4.28. Energies and spectrum of chaotic flow at Wi $=$ 13

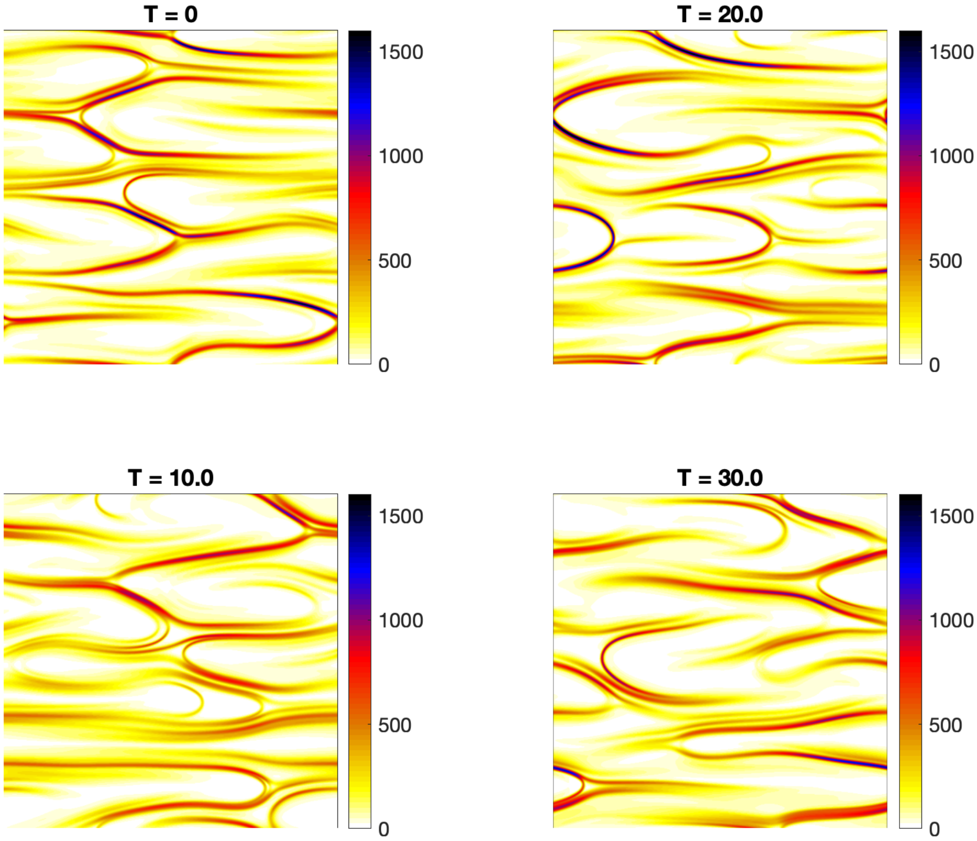


FIGURE 4.29. Fully chaotic flow at Wi = 13

4.8. Conclusions

This brings us to the end of our exploration of Kolmogorov flow and the transitions to chaos seen in three different domains. The case of $k = 1$ stands out, being quite distinct from the other cases, as well as revealing the first systematic route to chaos seen in viscoelastic fluids. In this case, the route to chaos is a cascade of period doublings and a period tripling which occur as a single narwhal interacts with itself via periodic boundary conditions.

The cases of $k = 2$ and $k = 4$ are in many ways similar to one another, and quite different than $k = 1$ due to the presence of multiple narwhals. Naturally, these flows have more complex dynamics, which is likely connected to the onset of chaos at lower Wi than with $k = 1$. Movies with examples of most of the flow states that we discuss in Chapters 3 and 4 are available as supplemental material.

Narwhals are a key coherent structure in all of these flows, and it was this structure that originally inspired this portion of our work. In particular, the results of Berti [8] involve multiple narwhals transitioning to chaos in Kolmogorov flow, and the results of Morozov [36] show a single narwhal in channel flow which appears to be stable. Studying Kolmogorov flow with fewer periods in the driving force was appealing as a way to gain insight into the differences between channel flow and Kolmogorov flow, and potentially about the role of narwhals in the transition to turbulence. This effort led to all of the results discussed in the preceding three chapters, and has raised many new questions that may be answered with further analysis and simulation. To name a few: Further simulations could uncover additional period doublings and triplings, or other systematic routes to chaos with $k = 2$ or $k = 4$. It would also be of interest to use specialized numerical methods to find additional periodic solutions and/or unstable equilibrium solutions, so that the parameter space can be more fully understood. Probing the applicability of our results to 3D Kolmogorov flow would be useful. Finally, it would be exciting to explore 2D Kolmogorov flow with longer domains more thoroughly than we have done in Section 3.6.7, especially with comparison to channel flow in mind.

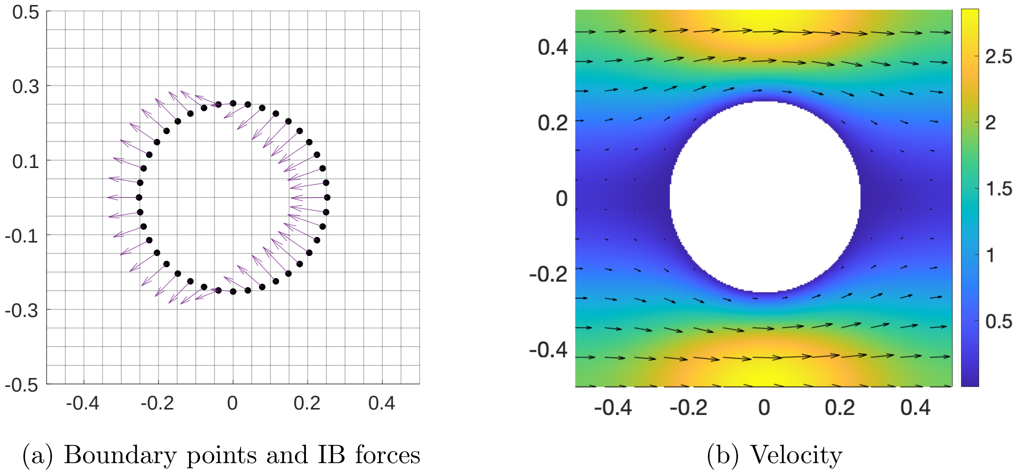
CHAPTER 5

Introduction to the Double Immersed Boundary Method and Application to Stokes Flow

5.1. Introduction to the Immersed Boundary Method

The Immersed Boundary Method, introduced by Peskin [44, 45], was designed for simulations of fluid-structure interactions and first used in simulations of blood flow around heart valves. The original idea of the Immersed Boundary (IB) Method was to simulate the coupled dynamics of elastic structures and surrounding fluid in such a way that computations can be done efficiently on an Eulerian grid, while the elastic structure is defined on a Lagrangian boundary that is not aligned with the grid and may be nonstationary. To link the Eulerian grid to the Lagrangian boundary, the IB method places a set of point forces along the boundary, which are mapped to the grid by convolution with a regularized delta function. This generates a collection of forces at nearby gridpoints which do the same work on the fluid as the Lagrangian boundary forces [32]. Thus, the IB method allows for the use of techniques that are fast and accurate on a regular grid (e.g. spectral methods), together with great flexibility in boundary geometry.

The IB method was adapted for problems with prescribed motion of the boundary in [32]. The idea was to use tether forces at boundary points, determined by the displacement of boundary points from their target positions. This method was further adapted for Stokes flow in [62]. After the tether force approach was developed a new idea emerged [61], also for use with prescribed motion of the boundary, in which the boundary conditions are viewed as a constraint which implicitly determines the boundary forces. This is similar to the way in which the constraint of incompressibility determines fluid pressure. Our work will use this constraint method with additional modifications. The basic idea is illustrated in figure 5.1, where periodic Stokes flow around a cylinder with no-slip boundary conditions is simulated.



(a) Boundary points and IB forces

(b) Velocity

FIGURE 5.1. Demonstration of the Immersed Boundary (IB) method for Stokes flow around a cylinder. The forces shown in (a) are spread to the grid and combined with a constant driving force in the x-direction. When Stokes equations are solved with this combined forcing, the resulting velocity is flow around a cylinder with no-slip boundary conditions, shown in (b). The color axis shows the magnitude of the velocity, and direction of flow is displayed with arrows.

One drawback to the IB method, which is described in detail in Section 5.1.4, is its failure to provide convergent velocity gradients near boundaries. In this chapter, we introduce the Double Immersed Boundary (DIB) method, with the goal of achieving convergence of velocity gradients everywhere on the domain, including near boundaries. An important application of this new method is simulating polymeric (viscoelastic) fluid flows, in which the polymer stress depends on the velocity gradient.

Our primary point of comparison for viscoelastic fluid simulations using the DIB method will be the Immersed Boundary Smooth Extension (IBSE) method [55, 56, 57], which provides convergent velocity gradients near the boundary. The IBSE method was tested with a standard benchmark problem for viscoelastic flows: 2D flow around a cylinder confined in a channel [1, 11, 12]. For the sake of simplicity and easing computational costs, we will focus on the related problem of flow around a cylinder with doubly period boundary conditions. We will focus first on Newtonian fluid to demonstrate the main idea of the DIB method. Viscoelastic fluids will follow in Chapter 6, along with discussion of methods to handle the additional challenges such simulations pose.

5.1.1. Linking an Eulerian grid with a Lagrangian boundary. Before we begin, it will be useful to establish the following notation for different parts of the domain.

- Computational domain, $\mathcal{C}$
- Fluid domain (outside of the cylinder), Ω
- Non-physical domain (inside cylinder), $\mathcal{E}$
- Cylinder boundary, Γ

Additionally, to help distinguish between quantities on the boundary and quantities on the grid, we will use capital letters for boundary values and lower case for the grid (e.g. we represent boundary forces with $\mathbf{F}$ and forces on the grid with $\mathbf{f}$).

The communication between our two coordinate systems will consist of two operations. The first, as described briefly in the introduction, is to map boundary forces to an approximately equivalent collection of forces at nearby grid points, which we refer to as *spreading* the forces. The second is interpolating quantities from the grid onto the boundary. Both of these operations rely upon a regularized delta function. We use the standard four-point delta constructed by Peskin [45]:

$$\delta_h = \frac{1}{h^2}\phi\left(\frac{x}{h}\right)\phi\left(\frac{y}{h}\right) \tag{5.1}$$

with $h = \Delta x = \Delta y$ and

$$\phi(r) = \begin{cases} 1/8\left(3 - 2|r| + \sqrt{1 + 4|r| - 4r^2}\right), & 0 \leq |r| < 1 \\ 1/8\left(5 - 2|r| - \sqrt{-7 + 12|r| - 4r^2}\right), & 1 \leq |r| \leq 2 \\ 0, & |r| > 2. \end{cases} \tag{5.2}$$

To spread forces from the boundary to the grid, we define the spread operator S as

$$(S\mathbf{F})(\mathbf{x}) = \int_\Gamma \mathbf{F}(s)\delta_h(\mathbf{x} - \mathbf{X}(s))\,ds. \tag{5.3}$$

To interpolate values from the grid onto the boundary, we again integrate against the delta, but here we integrate over the computational domain. We define the interpolation operator, S^*, as

$$(S^*\mathbf{u})(s) = \int_\mathcal{C} \mathbf{u}(\mathbf{x})\delta_h(\mathbf{x} - \mathbf{X}(s))\,d\mathbf{x}. \tag{5.4}$$

As the notation suggests, the spread and interpolate operators are adjoint with respect to the standard L^2 inner products. That is, $\langle S\mathbf{F}, \mathbf{u}\rangle_{\mathcal{C}} = \langle \mathbf{F}, S^*\mathbf{u}\rangle_{\Gamma}$. We refer the interested reader to Section 2.2.2 of [55] for illustrative examples of these operators in one dimension.

5.1.2. Stokes equations with the IB method. Stokes equations for fluid velocity $\mathbf{u}$, pressure p, and external force $\mathbf{f}$ with no-slip boundary conditions are

$$\Delta \mathbf{u} - \nabla p + \mathbf{f} = 0 \qquad \text{in } \Omega, \tag{5.5}$$

$$\nabla \cdot \mathbf{u} = 0 \qquad \text{in } \Omega, \tag{5.6}$$

$$\mathbf{u} = 0 \qquad \text{on } \Gamma. \tag{5.7}$$

The spread forces and interpolated velocity will be substituted into (5.5) and (5.7), respectively. First however, we will introduce a driving force and a flow rate condition. The flow will be driven by a horizontal force, $\boldsymbol{\alpha} = (\alpha, 0)$, applied uniformly in all of $\mathcal{C}$. We make the usual choice to drive the flow left-to-right, so $\alpha > 0$. The value of α will be determined by specifying an average flowrate, V. This flowrate is computed by integrating the x-velocity from the bottom of the domain at $y = y_l$ to the top at $y = y_h$:

$$\frac{1}{y_h - y_l} \int_{y_l}^{y_h} u(0, y)\, dy = V. \tag{5.8}$$

Throughout this chapter, we will use the flowrate $V = 1$. Note that in Chapter 6, we use $V = 1/2$.

Incorporating this driving force and flow rate condition in Stokes equations, along with substituting the spread forces (5.3) and interpolated velocity (5.4), we have the following system to solve:

$$\Delta \mathbf{u} - \nabla p + S\mathbf{F} + \boldsymbol{\alpha} = 0 \qquad \text{in } \mathcal{C}, \tag{5.9}$$

$$\nabla \cdot \mathbf{u} = 0 \qquad \text{in } \mathcal{C}, \tag{5.10}$$

$$S^*\mathbf{u} = 0 \qquad \text{on } \Gamma, \tag{5.11}$$

$$\frac{1}{y_h - y_l} \int_{y_l}^{y_h} u(0, y)\, dy = V. \tag{5.12}$$

We will need to invert this system, but we must address two issues before doing so. First, note that to satisfy (5.9), it must be the case that the forces are in the range of the Laplace operator, meaning they must have zero mean (pressure is not a concern because ∇p has zero mean by periodicity of p). Second, solutions to Stokes equations on a periodic domain are not unique unless the mean velocity is constrained, because adding an arbitrary mean velocity to any solution produces another solution. The same is true of the pressure but this is less concerning, as the mean pressure is rarely of interest and often set to zero, which we will do in our work. In order to properly handle the mean velocity, we will break the velocity into two components, $\mathbf{u} = \mathbf{u}_0 + \overline{\mathbf{u}}$, where the $\mathbf{u}_0$ has zero mean and $\overline{\mathbf{u}}$ is a constant. Rewriting the system with this decomposition and the solvability condition gives

$$\Delta \mathbf{u}_0 - \nabla p + S\mathbf{F} + \boldsymbol{\alpha} = 0 \qquad \text{in } \mathcal{C}, \tag{5.13}$$

$$\nabla \cdot \mathbf{u}_0 = 0 \qquad \text{in } \mathcal{C}, \tag{5.14}$$

$$S^*(\mathbf{u}_0 + \overline{\mathbf{u}}) = 0 \qquad \text{on } \Gamma, \tag{5.15}$$

$$\frac{1}{y_h - y_l} \int_{y_l}^{y_h} u_0(0, y) + \overline{u}(0, y)\, dy = V, \tag{5.16}$$

$$\int_{\mathcal{C}} S\mathbf{F} + \boldsymbol{\alpha}\, d\mathbf{x} = 0. \tag{5.17}$$

5.1.3. The Schur complement. We are now prepared for the first step in solving our system, which is to determine the boundary forces, driving force, and mean velocity, ($\mathbf{F}$, $\boldsymbol{\alpha}$, $\overline{\mathbf{u}}$) that are required in order to satisfy our chosen boundary conditions and flow rate. To do this, we will construct a Schur complement for the system, $\mathbf{M}$, which maps ($\mathbf{F}$, $\boldsymbol{\alpha}$, $\overline{\mathbf{u}}$) to the boundary velocity, flow rate, and integral of the forces. We will then use the inverse of this operator to determine the unknown quantities.

To construct $\mathbf{M}$, we will need to express $S^*\mathbf{u}$, V, and $\int_{\mathcal{C}} S\mathbf{F} + \boldsymbol{\alpha}\, d\mathbf{x}$ in terms of ($\mathbf{F}$, $\boldsymbol{\alpha}$, $\overline{\mathbf{u}}$). We begin by manipulating Eqs. (5.13–5.14) to isolate the velocity and pressure. This process starts with taking the divergence of Eq. (5.13). By incompressibility (5.14), the first term in Eq. (5.13) drops out and we can get the pressure in terms of the forces,

$$p = -\Delta^{-1} \nabla \cdot (S\mathbf{F} + \boldsymbol{\alpha}), \tag{5.18}$$

where we take the mean to be 0 when inverting the Laplacian. Inserting (5.18) into (5.13) and inverting gives us $\mathbf{u}_0$ in terms of only the forces. We add the mean (recall $\mathbf{u} = \mathbf{u}_0 + \overline{\mathbf{u}}$) to find the full velocity,

$$\mathbf{u} = \Delta^{-1}\left(\nabla(-\Delta^{-1}\nabla \cdot (S\mathbf{F} + \boldsymbol{\alpha})) + S\mathbf{F} + \boldsymbol{\alpha}\right) + \overline{\mathbf{u}}. \tag{5.19}$$

Interpolating this to the boundary gives

$$S^*\mathbf{u} = S^*\left[\Delta^{-1}\left(\nabla(-\Delta^{-1}\nabla \cdot (S\mathbf{F} + \boldsymbol{\alpha})) + S\mathbf{F} + \boldsymbol{\alpha}\right) + \overline{\mathbf{u}}\right]. \tag{5.20}$$

We now proceed to the easier tasks of expressing the flow rate and integral of the forces in terms of ($\mathbf{F}$, $\boldsymbol{\alpha}$, $\overline{\mathbf{u}}$). For the flow rate, we simply evaluate

$$V = \frac{1}{y_h - y_l}\int_{y_l}^{y_h} u(0, y)\, dy \tag{5.21}$$

using the velocity from Eq. (5.19). The integral from Eq. (5.17) only involves $\mathbf{F}$ and $\boldsymbol{\alpha}$, so no additional work is required. Of course, this integral must evaluate to 0 to solve Stokes equations. However in the process of building $\mathbf{M}$ it will generally be nonzero and we denote the quantity Φ,

$$\Phi = \int_{\mathcal{C}} S\mathbf{F} + \boldsymbol{\alpha}\, d\mathbf{x}. \tag{5.22}$$

As mentioned in the beginning of this section, the Schur complement $\mathbf{M}$ maps ($\mathbf{F}$, $\boldsymbol{\alpha}$, $\overline{\mathbf{u}}$) to $(S^*\mathbf{u}, V, \Phi)$. Thus, we can write Eqs. (5.20–5.22) compactly as

$$\mathbf{M}\begin{pmatrix}\mathbf{F}\\ \boldsymbol{\alpha}\\ \overline{\mathbf{u}}\end{pmatrix} = \begin{pmatrix}S^*\mathbf{u}\\ V\\ \Phi\end{pmatrix}. \tag{5.23}$$

We construct $\mathbf{M}$ column-by-column, by repeated application to unit vectors. Once the Schur complement is formed, we apply its inverse to determine the forces and mean velocity that are required to meet the solvability condition and our chosen boundary conditions. To satisfy no-slip boundary conditions and the solvability condition, we solve the system

$$\mathbf{M}\begin{pmatrix}\mathbf{F}\\ \boldsymbol{\alpha}\\ \overline{\mathbf{u}}\end{pmatrix} = \begin{pmatrix}0\\ V\\ 0\end{pmatrix} \tag{5.24}$$

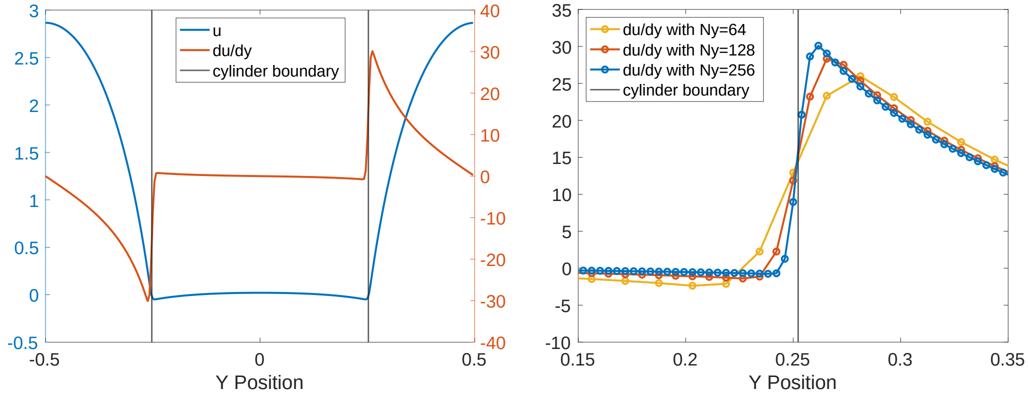
to obtain the boundary forces, driving force, and mean velocity. With these in hand, we use Eqs. (5.18–5.19) to compute the velocity and pressure on the computational domain. To summarize, there are three steps to solve Stokes equations with our implementation of the IB method:

(1) Build the Schur complement $\mathbf{M}$

(2) Apply $\mathbf{M}^{-1}$ to obtain $\mathbf{F}$, $\boldsymbol{\alpha}$, and $\overline{\mathbf{u}}$

(3) Insert these quantities into Stokes equations and solve for velocity and pressure.

5.1.4. Gradient discontinuities with the IB method. The IB method has a drawback in that using delta-like forces will create a velocity field whose gradient is discontinuous across the bounary Γ. For one dimensional analogy consider solving $u_{xx} = \delta(x)$, where two integrations of the right hand side give a C^0 function with a jump in u_x at $x = 0$. This behavior is apparent in Fig. 5.2, where we inspect slices of the x-velocity and $\partial u/\partial y$ in a solution for flow around a cylinder computed using the IB method described in the previous three sections. This discontinuity results in an incorrect value of the gradient near the boundary.



(a) u and $\partial u/\partial y$

(b) $\partial u/\partial y$ at three grid resolutions

FIGURE 5.2. The IB method creates gradient discontinuities on boundaries, as shown here for Stokes flow around a cylinder. Both plots slow vertical slices of the domain (shown in figure 5.1b) taken along $x = 0$.

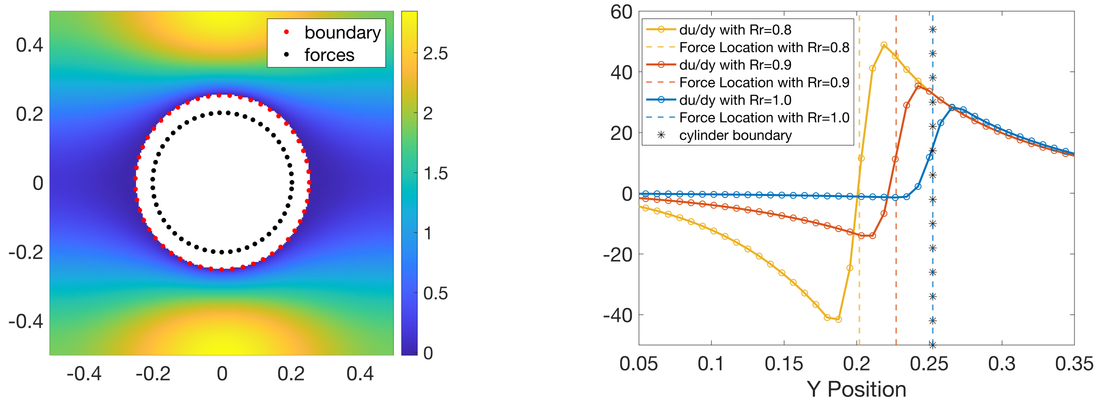
Depending on the application, accuracy of the velocity gradient may or may not be important. This work is focused in particular on viscoelastic fluid simulations, where accurate velocity gradients are important because viscoelastic stress depends upon the velocity gradient. Before exploring viscoelastic flows however, we will focus on obtaining accurate velocity gradients in Stokes flow.

5.2. Introduction to the Double Immersed Boundary Method

The main idea of the Double Immersed Boundary (DIB) is quite simple and comes from two observations. First, when using the IB method, the jump in the velocity gradient is localized near the point forces (see Fig. 5.2). Second, the locations of these point forces need not coincide with the location where boundary conditions are enforced. With this in mind, the DIB method will use point forces placed along a new curve, γ, that is separated from the boundary, Γ. The forces are placed in the non-physical domain, $\mathcal{E}$, which in our case corresponds to placing them inside the cylinder. The forces still enforce boundary conditions on Γ, just as in the IB method. This means we apply the spread and interpolate operators at different locations: the forces are spread along γ, while velocities are interpolated along Γ. This moves the gradient discontinuity out of the fluid domain. For the work presented here, γ is always a circle, and we define $\mathrm{Rr} = \mathrm{rad}(\gamma)/\mathrm{rad}(\Gamma)$ as the ratio of the radii used for the forces and for the cylinder boundary. Thus, $\mathrm{Rr} = 1$ corresponds to the IB method and $\mathrm{Rr} < 1$ to the DIB method. Fig. 5.3 demonstrates the idea of the DIB method, with 5.3a illustrating the placement of forces inside the cylinder, and 5.3b showing the jump in the velocity gradient at different locations generated by three separate simulations using a variety of force placements. Naturally, there are trade-offs, which we will begin addressing in the next section.

5.3. Conditioning issues with DIB

Recall that the condition number of an operator is defined as $\kappa = \frac{\sigma_{max}}{\sigma_{min}}$, the ratio of largest to smallest singular values. A poorly conditioned operator is one with a large condition number, which indicates that applying the inverse of the operator to an input with small errors will produce a solution with large errors. In order for DIB to work well then, we must have a reasonably well conditioned Schur complement.



(a) Horizontal velocity, with force locations and boundary points shown.

(b) $\partial u/\partial y$ from three tests.

FIGURE 5.3. Demonstration of the principle of Double Immersed Boundary. (a) shows flow around a cylinder, with forces moved away from the boundary into the non-physical domain. (b) shows slices of the velocity gradient from three tests. IB is shown in blue, and DIB with 2 force placements are in red and yellow. All three have a jump in the gradient, but we can observe the jump moving further inside the cylinder (to the left of the star markers) as well as growing in magnitude as the forces are placed further from the cylinder wall. All tests used a resolution of Ny=128.

5.3.1. A theoretical prediction. We can learn something about the conditioning of the DIB Schur complement by examining a related problem: Laplace's equation on $\mathbb{R}^2$. Recall that Laplace's equation,

$$\Delta \mathbf{u} = \delta(\mathbf{x}_0) \tag{5.25}$$

has the fundamental solution

$$\mathbf{u}(\mathbf{x}) = \frac{1}{2\pi} \ln(|\mathbf{x} - \mathbf{x}_0|). \tag{5.26}$$

Suppose we wish to solve $\Delta \mathbf{u} = \mathbf{f}(\mathbf{x})$, using $\mathbf{f}(\mathbf{x})$ to enforce boundary conditions on the unit circle. We will place the forces, $\mathbf{f}(\mathbf{x})$, on a concentric circle of radius R. Just as in our approach to the immersed boundary method, we would find and invert the operator that maps forces to $\mathbf{u}(\mathbf{x})$ on the boundary. In this case, we can determine the eigenfunctions and eigenvalues analytically. It also turns out that the operator is symmetric, meaning that the singular values are simply the absolute values of the eigenvalues.

To find the eigenvalues, begin by parameterizing the unit circle as $(\cos\theta, \sin\theta)$ and the circle of forces as $(R\cos\alpha, R\sin\alpha)$. We will represent the set of forces as $f(\alpha)$. Using Eq. (5.26) and integrating the forces, we have

$$u(\theta) = \int_0^{2\pi} \frac{1}{2\pi} \ln\left(\sqrt{(R\cos\alpha - \cos\theta)^2 + (R\sin\alpha - \sin\theta)^2}\right) f(\alpha) R\, d\alpha, \tag{5.27}$$

$$= \int_0^{2\pi} \frac{R}{4\pi} \ln\left(R^2 + 1 - 2R\cos(\alpha - \theta)\right) f(\alpha)\, d\alpha, \tag{5.28}$$

$$= I(f). \tag{5.29}$$

We will now show that the eigenfunctions of I are $f_k = \exp(ik\alpha)$ for $k \in \mathbb{Z}$.

$$I(\exp(ik\alpha)) = \int_0^{2\pi} \frac{R}{4\pi} \ln\left(R^2 + 1 - 2R\cos(\alpha - \theta)\right) \exp(ik\alpha)\, d\alpha, \tag{5.30}$$

$$= \exp(ik\theta) \int_0^{2\pi} \frac{R}{4\pi} \ln\left(R^2 + 1 - 2R\cos(\alpha - \theta)\right) \exp(ik(\alpha - \theta))\, d\alpha, \tag{5.31}$$

$$= \exp(ik\theta) \int_0^{2\pi} \frac{R}{4\pi} \ln\left(R^2 + 1 - 2R\cos(\omega)\right) \exp(ik(\omega))\, d\alpha, \tag{5.32}$$

where we used the substitution $\omega = \alpha - \theta$ and 2π periodicity to arrive at Eq. (5.32). Therefore we have eigenvalues of

$$\lambda_k = \int_0^{2\pi} \frac{R}{4\pi} \ln\left(R^2 + 1 - 2R\cos(\omega)\right) \exp(ik(\omega))\, d\alpha, \tag{5.33}$$

$$= \frac{-R}{4\pi} \int_0^{2\pi} \frac{2R\sin(\omega)}{R^2 + 1 - 2R\cos(\omega)} \frac{\exp(ik\omega)}{ik}\, d\omega. \tag{5.34}$$

We use a computer algebra system to evaluate this integral, though it can also be evaluated by hand via contour integration. For $R \leq 1$, we have

$$\lambda_k = \frac{-R^{|k|+1}}{2|k|}. \tag{5.35}$$

To relate Eq. (5.35) to a discretized problem, we consider a truncated series. We use N to denote the largest value of k, which will be proportional to the number of grid cells used. For $0 < R \leq 1$, the largest and smallest singular values are σ_1 and σ_N, respectively (recall that $\sigma_k = |\lambda_k|$). Thus, we have $\kappa = \sigma_1/\sigma_N$, or

$$\kappa = \begin{cases} N, & R = 1, \\ N\left(\frac{1}{R}\right)^{N-1}, & 0 < R < 1. \end{cases} \tag{5.36}$$

So, with either $R = 1$ (analogous to IB) or $0 < R < 1$ (analogous to DIB) the condition number grows as resolution is increased. The growth is linear for IB and exponential for DIB. Also note that in the DIB case we have polynomial growth in κ as R is reduced, or in other words as the forces are moved further away from the boundary.

In the DIB case, we can take advantage of the relation between condition number and R to slow the growth of condition number as the grid is refined. Suppose we set the distance between the force locations and the boundary at a constant number of grid cells, meaning that $R = 1 - c/N$ for some constant c with $0 < c < N$. We will still have $\kappa = \sigma_1/\sigma_N$ and thus the condition number is

$$\kappa = \frac{N}{(1-c/N)^{N-1}}, \tag{5.37}$$

which we will show grows linearly as N increases. Consider the denominator of 5.37,

$$(1-c/N)^{N-1} = \frac{(1-c/N)}{(1-c/N)^N} < \frac{1}{(1-c/N)^N}. \tag{5.38}$$

Next, because $\lim\limits_{N\to\infty}(1-c/N)^N = e^{-c}$, we know that for large enough N,

$$\frac{1}{(1-c/N)^N} < e^c + \epsilon \tag{5.39}$$

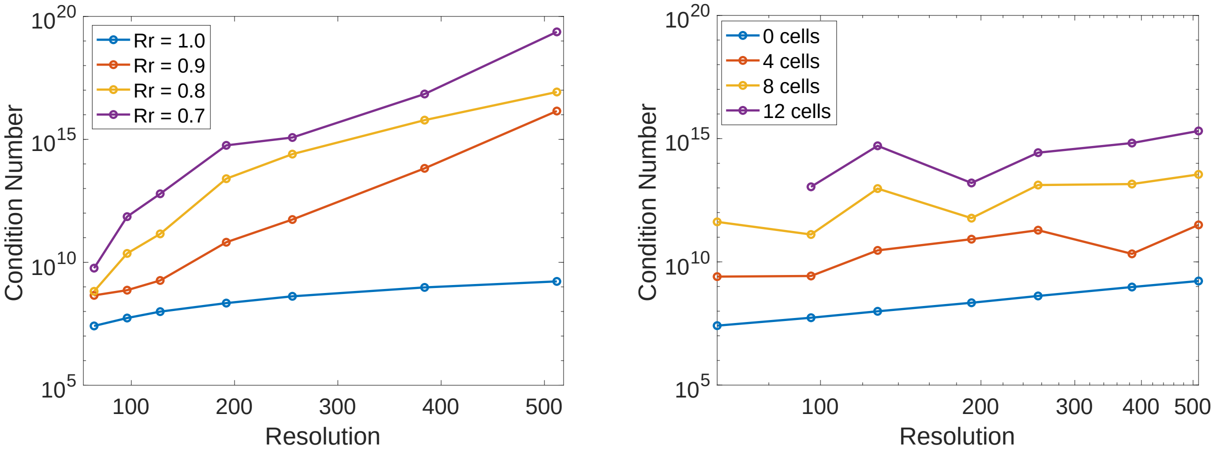
for any $\epsilon > 0$. Thus for large N, we have

$$\kappa = \frac{N}{(1-c/N)^{N-1}} < (e^c + \epsilon)N, \tag{5.40}$$

which demonstrates linear growth in κ as desired.

Recall that these results are only directly applicable to the Poisson problem on an infinite domain. We must use numerical methods to see how well it translates to periodic Stokes flow around a cylinder, which is the subject of the next section.

5.3.2. Numerical inspection of conditioning. We begin with an examination of the condition number of the Schur complement as resolution is increased in Fig. 5.4. When the distance between the force locations and boundary is kept constant (Fig. 5.4a), growth in the condition number is approximately exponential for DIB at resolutions of 256 × 256 and up. This aligns well with the prediction from the preivous section, based on the Poisson problem. When we keep a constant number of grid cells between the force locations and the boundary (Fig. 5.4b), growth is approximately polynomial. While the fixed grid cell approach does lead to smaller condition numbers than the fixed distance approach, both give large condition numbers at high resolution.



(a) Fixed distance between boundary and forces.

(b) Fixed number of grid cells between boundary and forces.

FIGURE 5.4. Changes in the condition number of the DIB Schur complement as resolution is increased. From the analysis in Section 5.3.1, we expect exponential growth when Rr < 1 and linear growth when Rr $= 1$.

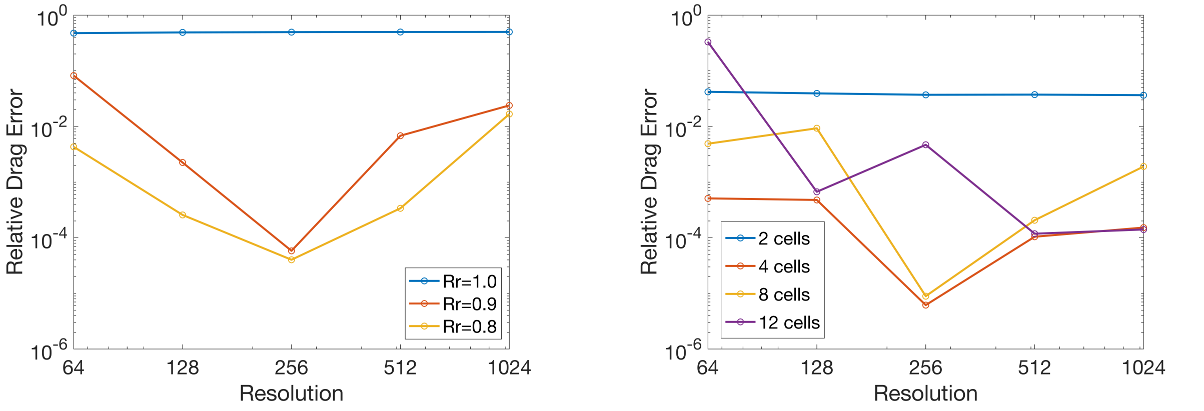
Now that we have explored the growth in the condition number, we will demonstrate the detrimental effect of poor conditioning on the DIB method when it is implemented as it has been described so far. Keeping in mind that the goal of DIB is to achieve accurate velocity gradients up to the boundary, we will inspect the drag force in periodic flow around a cylinder because this force depends upon the velocity gradient along the boundary. Additionally, we will inspect the pointwise velocity gradient on the cylinder boundary. After demonstrating poor performance, we will discuss solutions to this conditioning issue and later return to the drag force and pointwise velocity gradient in Section 5.6 to verify that our solutions provide satisfactory results.

The drag force is given by

$$\boldsymbol{F}_D = \int_\Gamma \left(-p \cdot \mathbf{I} + \mu(\nabla \mathbf{u} + \nabla \mathbf{u}^T)\right) \cdot d\boldsymbol{S}, \tag{5.41}$$

and dimensionless drag is $\boldsymbol{F}_D/(\mu V)$ (we are using $\mu = V = 1$). When integrating along the boundary, we use values of p and $\nabla \mathbf{u}$ that have been interpolated from the grid to the boundary using the interpolation operator defined in Eq. (5.4). Values for dimensionless drag computed using integral methods are available in [**20**], and we will use these as our point of comparison for DIB. We can see the impact of poor conditioning in Fig. 5.5, which shows drag results as the grid is refined. In Fig. 5.5a, the gap between the force locations and the cylinder boundary, stated in terms of Rr, is held constant as the grid is refined. For resolutions using 256 or more grid cells, using DIB (Rr $<$ 1) gives errors that grow with resolution for both values of Rr shown. Note that from Fig. 5.4a we know that the condition numbers in the simulations at resolutions of 512 and higher are over 10^{15}, so these large drag errors are expected. The IB method (Rr $=$ 1) is also shown in Fig. 5.5a, which as expected gives large errors at all resolutions. The errors are not due to conditioning, however. Rather, they are due to the lack of convergent velocity gradients on the boundary inherent in the IB method. In Fig. 5.5b, we keep a constant number of grid cells between the forces and the cylinder boundary as the grid is refined. While the drag errors do not definitively grow at high resolutions, they only drop below $\approx 10^{-4}$ at a moderate resolution of 256 grid cells. We will see later (Section 5.6) that with improved conditioning of the Schur complement, errors drop well below 10^{-4}, even at high resolution.

An interesting aspect of Fig. 5.5b is the error seen with 2 grid cells between the cylinder boundary and the forces. While the error is smaller than that seen with the IB method (3.7% vs 49%), it remains nearly constant as the grid is refined. This suggests that with gap of 2 grid cells, errors depend less on conditioning and, like the IB method, result from the jump in velocity gradient. This is expected - the regularized delta function we use for the interpolation operator is supported over four points in both the x- and y-directions. Centering the delta at boundary points means that erroneous values from the velocity gradient discontinuity are incorporated into the interpolated value.

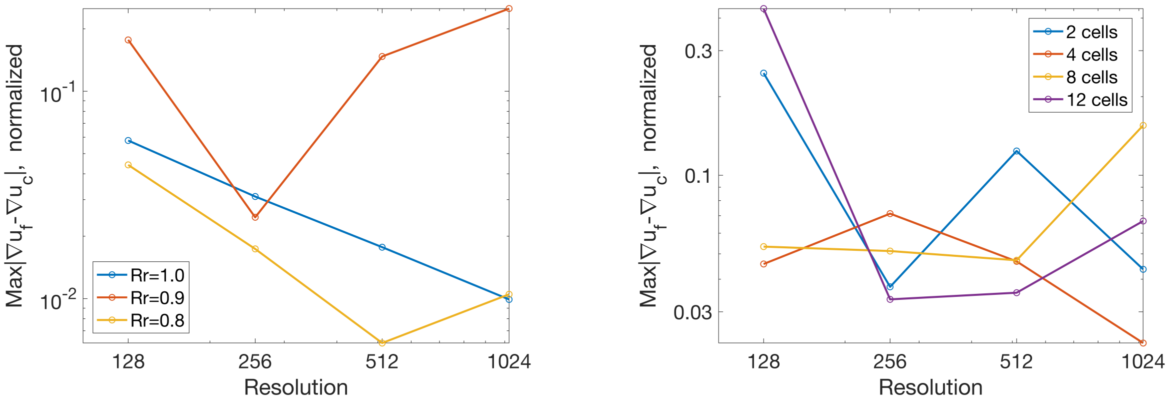


(a) Fixed distance between boundary and forces.

(b) Fixed number of grid cells between boundary and forces.

FIGURE 5.5. Relative error in dimensionless drag as a function of grid resolution. Several different boundary spacings are shown. With the gap size held constant (a), errors clearly become large at high resolution. When we keep a constant number of grid cells (b), the growth of errors is less pronounced. However, both strategies leave something to be desired. As we will see later, improving the conditioning of the Schur complement gives errors $\approx 10^{-6}$.

We can also see the effect of poor conditioning in the pointwise velocity gradient along the cylinder boundary. Fig. 5.6 shows the differences between successive refinements in $\nabla \mathbf{u}$ along the boundary, using the max norm. Like the previous tests for drag force, we have performed one refinement study with a fixed distance between the boundary and the forces (Fig. 5.6a) and one with a fixed number of grid cells between the boundary and forces (Fig. 5.6b). The only test which appears convergent uses $\mathrm{Rr} = 1$, where forces are placed on the boundary (in other words this is the IB method, not DIB). However, while the velocity gradient converges pointwise in this test, we know from our drag computations (Fig. 5.5a) that it does not converge to the correct solution. For all of the tests shown in Fig. 5.6, the differences between refinements are quite large and do not appear convergent. As a reference point, at high resolution with improved conditioning we obtain differences smaller than 10^{-3} in most tests (See Figs. 5.14 and 5.15).



(a) Fixed distance between boundary and forces.

(b) Fixed number of grid cells between boundary and forces.

FIGURE 5.6. Max norm of the difference between velocity gradients on the boundary at successive refinements. The resolution value used to plot each data point represents the fine resolution for that comparison. For example, a point plotted at resolution 512 comes from comparing results at 256 and 512. Several different boundary spacings are shown. All suffer from poor conditioning at high resolution.

5.4. Methods to improve conditioning

Now that we have introduced the Schur complement, $\mathbf{M}$, for our system, along with its poor conditioning and the resulting inaccuracies in drag and the velocity gradient, we will explore ways to improve the conditioning. We will briefly explore two methods, though it should be noted that there is a wealth of literature on solving poorly conditioned problems [30, 40].

For both methods, we will discuss the singular value decomposition

$$\mathbf{M} = \mathcal{U}\Sigma\mathcal{V}^\top, \tag{5.42}$$

where $\mathcal{U}$ and $\mathcal{V}$ are unitary and Σ is a diagonal matrix containing the singular values, σ_i, of $\mathbf{M}$. If $\mathbf{M}$ is invertible, its inverse is

$$\mathbf{M}^{-1} = \mathcal{V}\Sigma^{-1}\mathcal{U}^\top. \tag{5.43}$$

An important idea for both methods is that small singular values of $\mathbf{M}$ create large entries in Σ^{-1}, which lead to numerical errors upon inverting $\mathbf{M}$. The methods we present to improve the conditioning of the problem involve creating a regularized version of Σ^{-1}. The first method does so directly, requiring an expensive SVD computation. The second method does not require computing the SVD, but effectively regularizes Σ^{-1} nonetheless.

5.4.1. Improving conditioning by truncating singular values. Our first method is related to a common way of handling non-invertible matrices. When $\mathbf{M}$ is not invertible, it will have one or more zero singular values. Supposing there are k nonzero singular values, the most common way to define a pseudoinverse, known as the Moore-Penrose inverse, is $\mathbf{M}^+ = \mathcal{V}\Sigma^+\mathcal{U}^\top$, where

$$\Sigma^+ = \begin{bmatrix} 1/\sigma_1 & & & & & \\ & \ddots & & & & \\ & & 1/\sigma_k & & & \\ & & & 0 & & \\ & & & & \ddots & \\ & & & & & 0 \end{bmatrix}. \tag{5.44}$$

Thus, applying the Moore-Penrose inverse has the effect of inverting $\mathbf{M}$ only on the subspace associated with nonzero singular values.

Our first approach to improving the conditioning of $\mathbf{M}$ takes this idea further. We will define a pseudoinvese $\mathbf{Z} = \mathcal{V}\Sigma^+_{sv}\mathcal{U}^\top$, where Σ^+_{sv} is the regularized version of Σ^{-1}. It has the same form as Σ^+, but includes the reciprocals of only those singular values which are larger than a chosen cutoff. The remaining entries of Σ^+_{sv} are set to zero. Thus, applying $\mathbf{Z}$ has the effect of inverting $\mathbf{M}$ only on the subspace associated with the largest singular values, eliminating a source of numerical error. The trade-off is a loss of information from the Schur complement, as we are ignoring its actions on the subspace associated with small singular values. With an appropriate cutoff, this lost information is unimportant and the application of $\mathbf{Z}$ still determines forces and a mean velocity which approximately satisfy our boundary conditions. But, we must determine the cutoff carefully. If the cutoff is too large, we eliminate too many singular values, lose important information, and fail to satisfy the boundary conditions. If the cutoff is too small, we retain too many singular values and will not eliminate the problem of poor conditioning.

The cutoff is set relative to the largest singular value. This is because poor conditioning does not result from small singular values alone, but rather the ratio of the largest and smallest singular values (recall $\kappa = \sigma_{max}/\sigma_{min}$). So, the parameter we will be choosing is actually a cutoff *ratio* which we denote $\mathrm{SV_c}$. Thus Σ_{sv}^{+} will contain the reciprocals of all singular values σ_i satisfying $\sigma_i/\sigma_{max} > \mathrm{SV_c}$. With this method we are setting an upper bound of $1/\mathrm{SV_c}$ on the condition number of the problem. We do not have a precise way to choose $\mathrm{SV_c}$, so we will do so based on experimental results. We will return to this task after introducing the second method to improve the conditioning of $\mathbf{M}$.

It should be noted that this method of truncating singular values requires computing the SVD of the Schur complement, which is computationally expensive. Because the size of $\mathbf{M}$ depends on the number of boundary points, this method is not practical for large systems. For this reason we offer a second method of controlling the conditioning of $\mathbf{M}$ which does not require the SVD.

5.4.2. Improving conditioning with Tikhonov regularization. Our second method to control the conditioning of the Schur complement is Tikhonov regularization (also known as ridge regression), a well known method for solving nearly singular systems [19, 65]. As discussed in Section 5.1.2, to solve Stokes equations we invert Eq. (5.24). However, the actual system being solved is not relevant to our discussion of Tikhonov regularization in this section. So, for the sake of simplicity, in this section we will represent the system to be solved as $\mathbf{MF} = \mathbf{u}$.

Rather than inverting this system directly, Tikhonov regularization finds a least squares solution to the augmented system

$$\begin{pmatrix} \mathbf{M} \\ \mathbf{A} \end{pmatrix} \mathbf{F} = \begin{pmatrix} \mathbf{u} \\ \mathbf{b} \end{pmatrix}. \tag{5.45}$$

Finding a least squares solution to Eq. (5.45) gives an $\mathbf{F}$ that compromises between solving $\mathbf{MF} = \mathbf{u}$ and solving $\mathbf{AF} = \mathbf{b}$. By weighting $\mathbf{A}$ with $0 < \epsilon \ll 1$, we can intentionally bias this compromise towards solving $\mathbf{MF} = \mathbf{u}$, which in the context of DIB enforces the boundary conditions. There are many choices of what to use for $\mathbf{A}$ and $\mathbf{b}$. The most common choice is to take $\mathbf{A} = \epsilon\mathbf{I}$ and $\mathbf{b} = \mathbf{0}$, which gives preference to solutions with smaller L^2 norms. Another common choice is $\mathbf{A} = \epsilon\mathbf{D}$ and $\mathbf{b} = \mathbf{0}$, where $\mathbf{D}$ is a difference operator, which gives preference to solutions with small derivatives. In practice both choices may effectively regularize the problem. In our own testing we obtained reasonably good results using both choices, though testing with the difference operator was limited. Going forward we will focus only on the case $\mathbf{A} = \epsilon\mathbf{I}$, $\mathbf{b} = \mathbf{0}$.

We can gain insight into how Tikhonov regularization works by again appealing to the singular value decomposition $\mathbf{M} = \mathcal{U}\Sigma\mathcal{V}^\top$. Plugging this, along with our choice of $\mathbf{A}$ and $\mathbf{b}$, into Eq. 5.45 yields

$$\begin{pmatrix} \mathcal{U}\Sigma\mathcal{V}^\top \\ \epsilon\mathbf{I} \end{pmatrix} \mathbf{F} = \begin{pmatrix} \mathbf{u} \\ \mathbf{0} \end{pmatrix}. \tag{5.46}$$

To find a least squares solution to Eq. (5.46), we left-multiply by the transpose of the first term and solve, arriving at

$$\mathbf{F} = \left(\mathcal{V}(\Sigma^2 + \epsilon^2\mathbf{I})^{-1}\Sigma\mathcal{U}^\top\right)\mathbf{u}. \tag{5.47}$$

We will use the notation Σ_{tk}^+ for the key term in Eq. (5.47). That is,

$$\Sigma_{tk}^+ = (\Sigma^2 + \epsilon^2\mathbf{I})^{-1}\Sigma. \tag{5.48}$$

This term plays the aforementioned role of a regularized version of Σ^{-1}.

Σ_{tk}^+ is a diagonal matrix with the values $\sigma_i/(\sigma_i^2 + \epsilon^2)$ on the diagonal, where σ_i is a singular value of $\mathbf{M}$. In contrast to the arbitrarily large entries of Σ^{-1}, the entries of Σ_{tk}^+ are bounded above by $1/(2\epsilon)$ and smoothly go to zero as $\sigma_i \to 0$. For large singular values, entries of Σ_{tk}^+ only differ from entries of Σ^{-1} by an ϵ^2/σ perturbation. Putting all this together, the effect is similar to the truncated singular value approach. The action of $\mathbf{M}$ on subspaces associated with small singular values is effectively ignored, while the action of $\mathbf{M}$ on subspaces associated with large singular values is changed negligibly. Like the previous method, Tikhonov regularization puts an upper bound on the condition number of the problem. In this case, the bound is relative to the largest singular value: $\sigma_{max}/(2\epsilon)$.

Unlike the truncated singular value approach in the previous section, Tikhonov regularization does not require an expensive SVD computation. For large systems, one could use a Krylov method to solve Eq. (5.45) without computing $\mathbf{M}$ or its SVD. However for a Krylov method to work efficiently, the challenge becomes providing a good preconditioner for the system.

The question of what value to use for ϵ still needs to be answered. Based on the preceding discussion, we can see that if ϵ is too small, the conditioning will not be significantly improved. On the other hand, if ϵ is too large, the augmented system no longer closely approximates the original system. We do not have a precise way to choose the value of ϵ, so we will do so based on experimental results.

For both the truncated singular value approach and Tikhonov regularization, a parameter value (cutoff ratio and ϵ, respectively) must be chosen. We will use the same experimental approach to choose both parameters, which is the subject of the next section.

5.5. Choosing conditioning parameters

5.5.1. Testing with a manufactured solution. To choose an appropriate value for each conditioning parameter, we test the DIB method with several parameter values and search for an optimal choice. In Section 5.1, we introduced the problem of periodic flow around a cylinder. Unfortunately, this problem would not be a good choice for evaluating the conditioning parameters because there is not an analytic solution. Since the goal of DIB is to obtain accurate velocity gradients up to the boundary, it would be better to use a test problem where the velocity gradient is known. One way of doing this is to use the method of manufactured solutions, which is a very general method for evaluating computational PDE solvers [29, 49]. At a high level, the method of manufactured solutions applied to the PDE $Lu = f$ follows the steps:

(1) Manufacture a solution, u_m, to the PDE.
(2) Analytically determine the forcing term, f_m, by solving $Lu_m = f_m$.
(3) As input to the computational method being tested, use f_m along with the boundary conditions determined by u_m.
(4) Compare the output of the computational method with the original manufactured solution.

To use this test method on DIB, we will work on our usual domain: The computational domain is $\mathcal{C} = [-0.5, 0.5] \times [-0.5, 0.5]$. The non-physical domain, $\mathcal{E}$, is a cylinder of radius $\sqrt{0.2/\pi}$ centered at the origin. The fluid domain is $\Omega = \mathcal{C} \backslash \mathcal{E}$.

We begin by manufacturing a velocity and pressure, $(\mathbf{u}_m, p_m)$. We will use

$$u_m = \exp(\sin(2\pi kx))\cos(2\pi ky), \tag{5.49}$$

$$v_m = -\exp(\sin(2\pi kx))\cos(2\pi kx)\sin(2\pi ky), \tag{5.50}$$

$$p_m = \exp(\cos(4\pi kx)). \tag{5.51}$$

This solution is somewhat arbitrary, but chosen to satisfy periodic boundary conditions and incompressibility, and so that no terms in Stokes equations vanish everywhere. We take $k = 3$ so that $\nabla \mathbf{u}_m$ is comparable to the maximum velocity gradient obtained in flow around a cylinder, as shown in Figs. 5.1 and 5.3. The x-component of the manufactured velocity is shown in Fig. 5.7

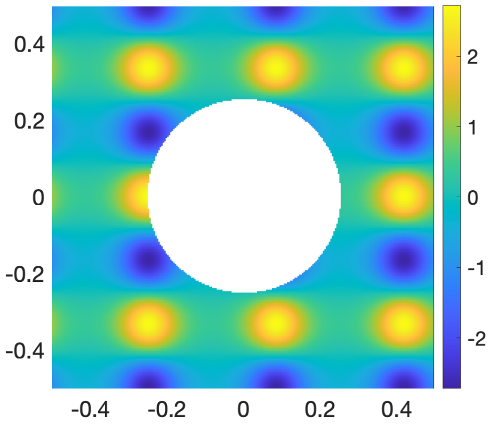


FIGURE 5.7. x-component of the manufactured velocity, u_m, on the fluid domain.

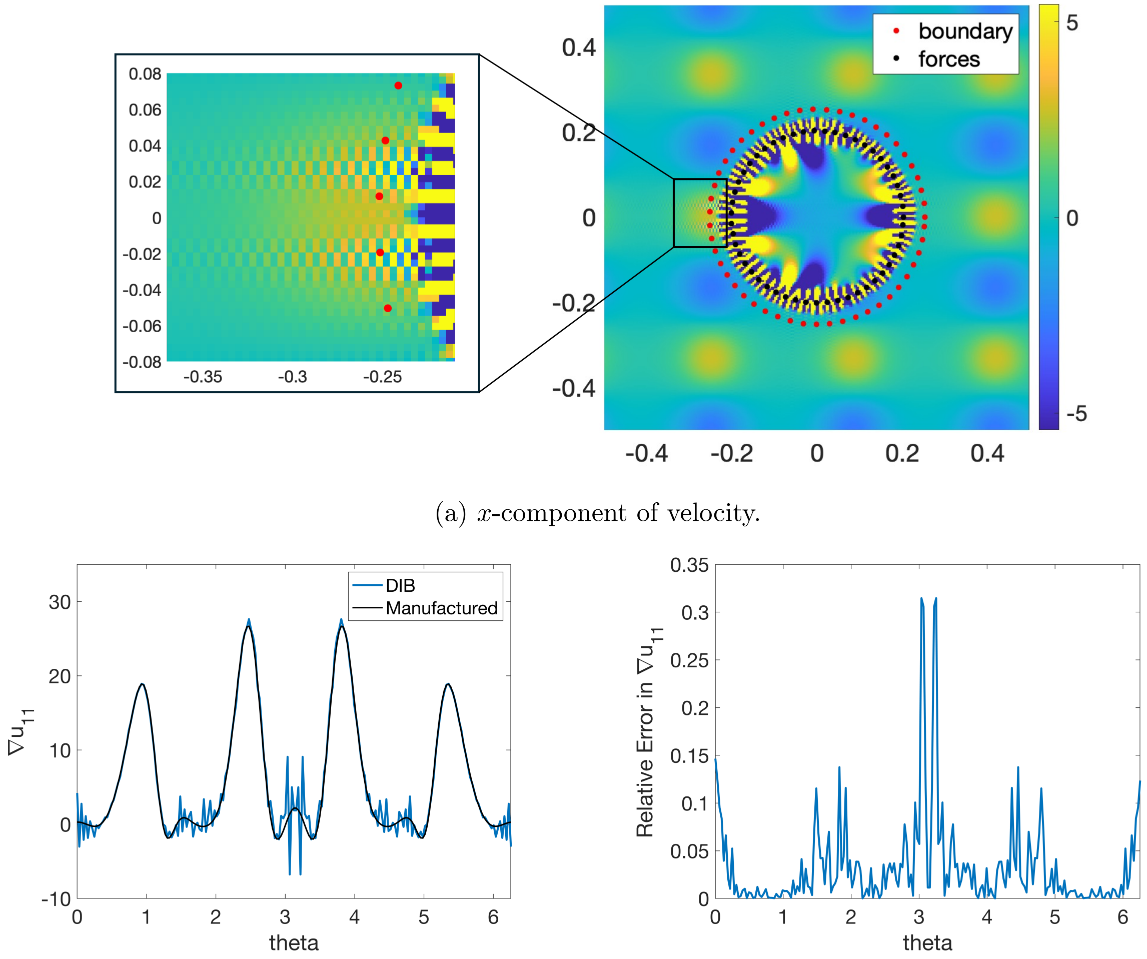
The next step is to find a forcing function, which is done analytically by evaluating $\mathcal{L}\mathbf{u}_m = \mathbf{f}_m$ on the entire computational domain, where $\mathcal{L}$ is the Stokes operator. We now apply $\mathbf{f}_m$ only on the fluid domain and attempt to recover $\mathbf{u}_m$ using the DIB method. Inside the cylinder we place point forces which are computed with the Schur complement so that they maintain the desired boundary velocity, $S^*\mathbf{u}_m$. Thus, the DIB method attempts to produce a velocity $\mathbf{u}$ satisfying

$$\mathcal{L}\mathbf{u} = \mathbf{f}_m|_\Omega + S\mathbf{F}, \tag{5.52}$$

$$S^*\mathbf{u} = S^*\mathbf{u}_m. \tag{5.53}$$

If DIB works as intended, we also expect $\nabla \mathbf{u}$ to converge to $\nabla \mathbf{u}_m$ in the fluid domain. However, based upon Section 5.3, we do not expect convergence if the conditioning of the Schur complement is not controlled. Fig. 5.8a illustrates part of the problem when conditioning is not controlled. It shows the x-velocity from a manufactured solution tests with no control of the conditioning. While the velocity far from the cylinder looks similar to the manufactured velocity, zooming in we can see grid scale oscillations which originate at the force points and extend beyond the cylinder boundary.

Given these oscillations, we cannot expect an accurate velocity gradient on the boundary. We inspect this gradient in Figs. 5.8b and 5.8c and as expected, find noisy and inaccurate results.



(a) x-component of velocity.

(b) ∇u_{11} on boundary

(c) Error in ∇u_{11} on boundary.

FIGURE 5.8. Results of a manufactured solution test without controlling conditioning of the Schur complement. Note the grid scale oscillations in (a), which extend from the forces to areas beyond the cylinder boundary. The color axis has been restricted so that the velocity outside the cylinder is still apparent. The x-velocity inside the cylinder ranges from -2699 to 3074. (b) and (c) inspect the first component of the velocity gradient interpolated to the boundary. This test was done on a 256×256 grid with $\text{Rr} = 0.8$.

In Section 5.4, we discussed how to control conditioning of the Schur complement with two methods - truncating singular values and Tikhonov regularization. Each method requires choosing the value of a parameter, which is our next task. We will carry out manufactured solution tests on each method with a wide range of parameter values in search of optimal choices. To evaluate the performance of each test, we will inspect three errors, all measured with a normalized max norm: the errors in velocity and its gradient in the fluid domain, and the error in the velocity gradient on the cylinder boundary. To obtain values on the boundary, we use the interpolation operator S^*.

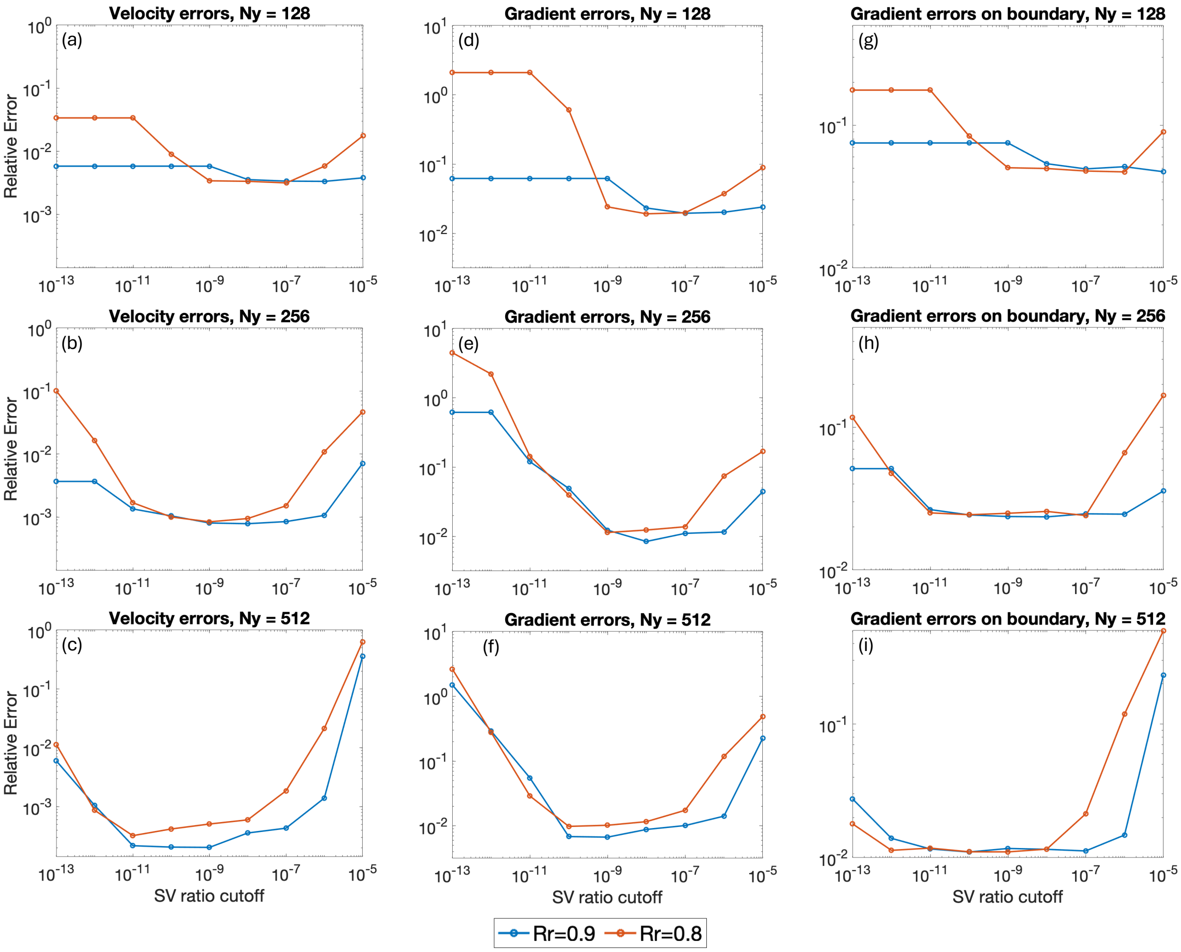


FIGURE 5.9. Results from manufactured solution testing using a range of values for the singular value ratio cutoff. The max norm, normalized by the norm of the manufactured solution, of the errors is computed for the velocity in the fluid domain (a-c), the velocity gradient in the fluid domain (d-f), and the velocity gradient interpolated to the boundary (g-i).

5.5.2. Results. Fig. 5.9 shows results from testing a variety of singular value ratio cutoffs. Recall from Section 5.4.1 that when the cutoff ratio is too small, the conditioning of the Schur complement is not adequately controlled. When the cutoff ratio is too large, we lose too much information from the Schur complement to effectively enforce boundary conditions. So, a cutoff that is either too small or too large will lead to large errors in the manufactured solutions tests, and this is indeed evident in the ‘U’ shaped error plots in Fig. 5.9. We seek a cutoff ratio that avoids both of these detrimental effects, or in other words, corresponds to a local minimum in the error plots. The various plots presented here allow us to see how the location of this minimum changes depending on resolution, boundary spacing, and whether we measure errors in the velocity, the velocity gradient, or the gradient interpolated to the boundary. Together, these plots lead us to choose a cutoff ratio of $\mathrm{SV_c} = 10^{-8}$, which works reasonably well in all of the tests. As an aside, we also note that the results from tests on a 128×128 grid with $\mathrm{Rr} = 0.9$ show relatively little variation as the cutoff ratio changes. This is because, at low resolution with the forces located near the boundary, the problem is still fairly well conditioned. From Fig. 5.4a, we can see that in this case, the condition number is below 10^{10}.

Fig. 5.10 shows results with Tikhonov regularization, using a variety of values for ϵ. The situation here is quite similar to the previous tests which explored the singular value ratio cutoff. With Tikhonov regularization, if the parameter ϵ is too small, the conditioning of the Schur complement is not adequately controlled. If ϵ is too large, then the augmented system (Eq. 5.45) is no longer a useful approximation to the original system and will not effectively enforce boundary conditions. So, again we have ‘U’ shaped graphs resulting from the large errors associated with values of ϵ that are too small or too large. It is interesting to note that using a larger than optimal value of ϵ appears less harmful than using a large value for the singular value ratio cutoff, particularly with the boundary spacing $\mathrm{Rr} = 0.9$. As before, we show results for a variety of resolutions and boundary spacings, checking the errors in velocity, velocity gradient, and the gradient interpolated to the boundary. We select $\epsilon = 10^{-7}$ because this roughly corresponds to the minimum error in all of the tests.

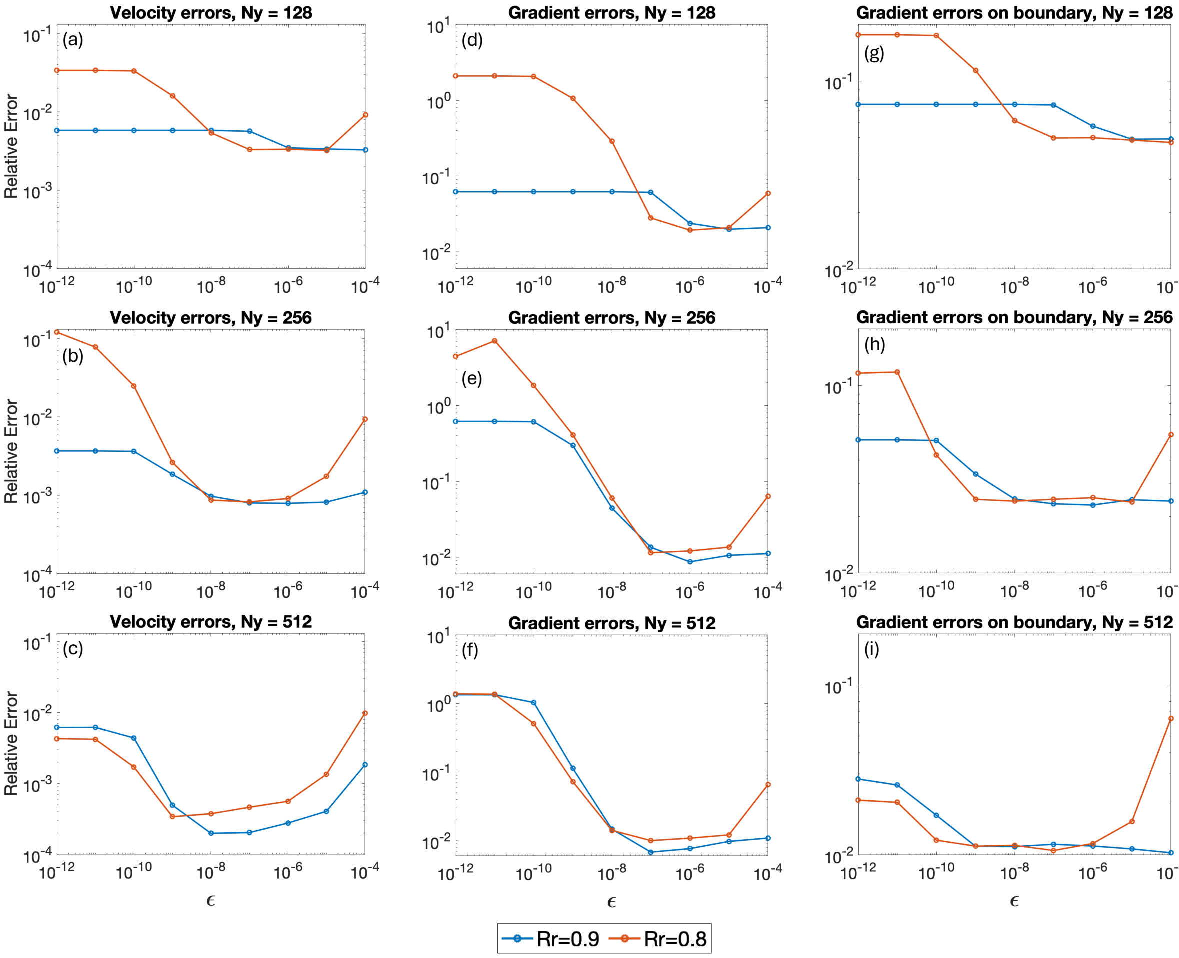
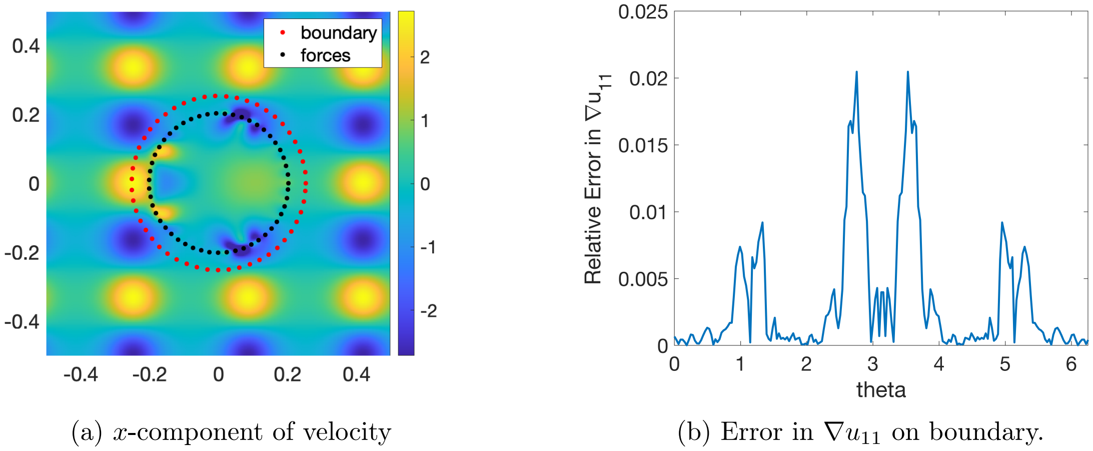


FIGURE 5.10. Results from manufactured solution testing using Tikhonov regularization with a range of values for ϵ. The max norm, normalized by the norm of the manufactured solution, of the errors is computed for the velocity in the fluid domain (a-c), the velocity gradient in the fluid domain (d-f), and the velocity gradient interpolated to the boundary (g-i).

We conclude this section with Fig. 5.11, in which we inspect the results of a manufactured solution test with well controlled conditioning. Using the truncated singular value approach with $\mathrm{SV_c} = 10^{-8}$, we reduce gradient errors on the boundary by more than an order of magnitude compared to those shown in Fig. 5.8, which were obtained without controlling the conditioning. Also, with better conditioning, there no longer appear to be numerical artifacts originating at the force locations which interfere with the solution near the cylinder boundary.



(a) x-component of velocity

(b) Error in ∇u_{11} on boundary.

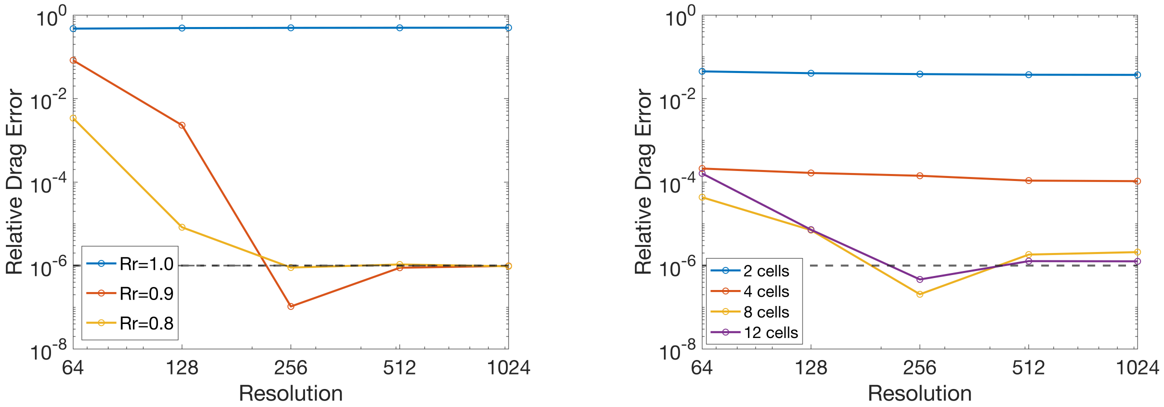
.

Figure 5.11. Results of a manufactured solution test with conditioning of the Schur complement controlled by a singular value ratio cutoff of $\mathrm{SV_c} = 10^{-8}$. This test was done on a 256×256 grid with $\mathrm{Rr} = 0.8$. Compare with the results in Fig. 5.8, where conditioning was not controlled. As done in Fig. 5.8, we again restrict the color axis in (a). However, the x-velocity inside the cylinder takes *much* smaller values with controlled conditioning, ranging from -6.9 to 2.7. (b) shows the relative error in $\nabla \mathbf{u}_{11}$ along the boundary. Notice that the errors shrink by more than an order of magnitude with controlled conditioning.

5.6. Stokes flow results with improved conditioning

Now that we have discussed methods to control the conditioning of the Schur complement and chosen parameter values, we return to simulations of Stokes flow around a cylinder. In Section 5.3.2, we saw poor results in the drag force and pointwise velocity gradient on the boundary. We will now implement both methods of controlling conditioning and re-examine these quantities.

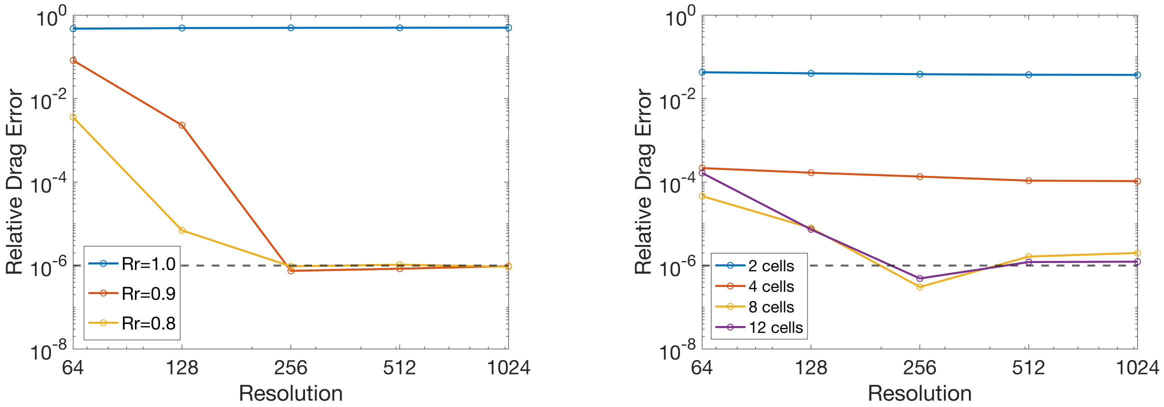
Drag force results with conditioning controlled by truncating singular values are in Fig. 5.12. Results using Tikhonov regularization are in Fig. 5.13. With each of these methods, we perform two refinement studies: One with the space between the forces and the boundary held constant and one with a fixed number of grid cells between the forces and the boundary. The results are very good, reaching a relative error of around 10^{-6} (shown with dashed lines) on a 256×256 grid in each test. On finer grids the errors are similar, but this does not indicate that the solutions are no longer converging. It is merely a product of the limited accuracy of the reference value we use from [20], which has 6 significant figures.



(a) As the grid is refined, space between boundaries is held constant.

(b) As the grid is refined, number of grid cells between boundaries is held constant.

FIGURE 5.12. Error in drag as a function of grid resolution, with conditioning of the Schur complement improved by truncating singular values. The cutoff ratio used here is 10^{-8}. Several different boundary spacings are shown. Compare to the poor results in Fig. 5.5. Note that asymptotic values we compare against are only computed to 6 digits in [20], so this test cannot give meaningful relative errors below $\sim 10^{-6}$. This value is indicated with dashed lines.



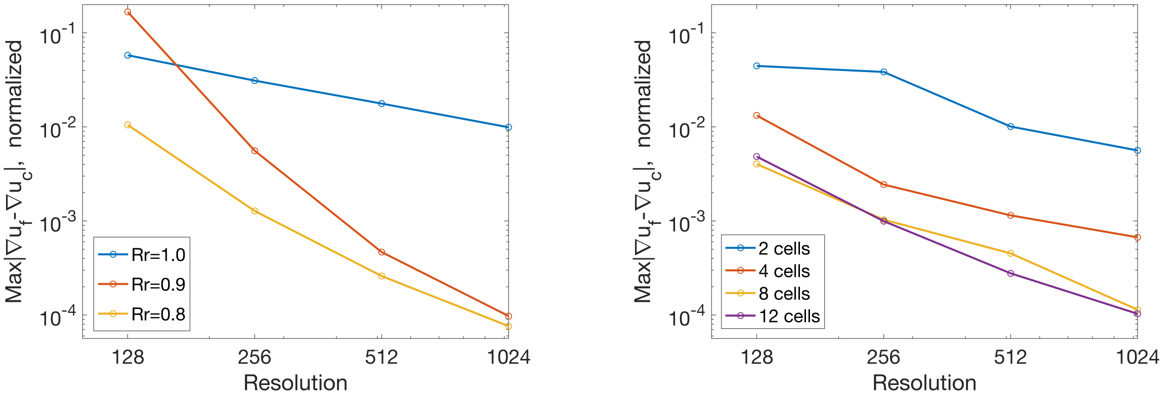
(a) As the grid is refined, space between boundaries is held constant.

(b) As the grid is refined, number of grid cells between boundaries is held constant.

FIGURE 5.13. Error in drag as a function of grid resolution, with conditioning of the Schur complement improved using Tikhonov regularization with $\epsilon = 10^{-7}$. Several different boundary spacings are shown. Compare to the poor results in Fig. 5.5. Note that asymptotic values we compare against are only computed to 6 digits in [20], so this test cannot give meaningful relative errors below $\sim 10^{-6}$. This value is indicated with dashed lines.

We now verify pointwise convergence of the velocity gradient. Results with conditioning controlled by truncating singular values are in Fig. 5.14. Results using Tikhonov regularization are in

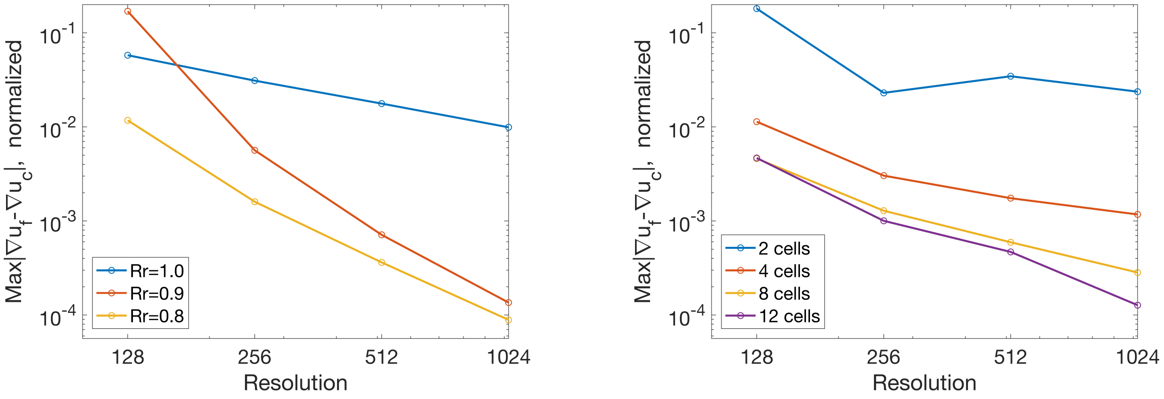
Fig. 5.15. Again for each method we perform two refinement studies: One with the space between the forces and the boundary held constant and one with a fixed number of grid cells between the forces and the boundary.



(a) As the grid is refined, space between boundaries is held constant.

(b) As the grid is refined, number of grid cells between boundaries is held constant.

FIGURE 5.14. Velocity gradient differences between successive refinements, with conditioning of the Schur complement improved by truncating singular values with a cutoff ration of 10^{-8}. Several different boundary spacings are shown. Compare to the poor results in Fig. 5.6.



(a) As the grid is refined, space between boundaries is held constant.

(b) As the grid is refined, number of grid cells between boundaries is held constant.

FIGURE 5.15. Velocity gradient differences between successive refinements, with conditioning of the Schur complement improved using Tikhonov regularization with $\epsilon = 10^{-7}$. Several different boundary spacings are shown. Compare to the poor results in Fig. 5.6.

Figs. 5.14 and 5.15 indicate that DIB provides a velocity gradient that converges pointwise on the boundary. In isolation this result does not imply that the velocity gradient is converging to the *correct* values. However, we have some assurance of this from our drag force results in Figs. 5.12a and 5.13. With this encouraging result, we conclude our work with DIB in Stokes flow, and proceed to viscoelastic flow.

CHAPTER 6

The Double Immersed Boundary Method Applied to Viscoelastic Flow

6.1. Introduction

In the previous chapter, we introduced the Double Immersed Boundary (DIB) method and developed necessary techniques for satisfactory simulations of periodic Stokes flow around a cylinder. Recall however, that Stokes flow is not the motivation for DIB. Our original goal was to develop a method which overcomes the traditional Immersed Boundary method's inability to produce convergent velocity gradients near boundaries. In particular, we had the application of polymeric (viscoelastic) flows in mind, because in these flows the polymer stress depends on the velocity gradient. The current chapter applies DIB to viscoelastic fluids.

The Immersed Boundary Smooth Extension (IBSE) method was developed with similar goals, and does in fact give pointwise convergence of the velocity gradient and polymer stress [**55**,**56**,**57**]. IBSE was a breakthrough in its own right and is very valuable as a point of comparison for the DIB method. IBSE involves inverting a large and complicated system and is thus computationally expensive. DIB, on the other hand, inverts a much smaller system. Even so, we certainly do not claim that DIB has improved upon IBSE. We merely present DIB as an alternative method to obtain convergent gradients.

We have chosen the test problem of 2D periodic flow around a cylinder, shown in Fig. 6.1. This choice was made in part to facilitate the Stokes flow drag force tests in the previous chapter, because asymptotic values for the drag force with this geometry are available in [**20**]. It is also a suitable choice for testing of viscoelastic flows, where we use IBSE as our benchmark. IBSE was tested thoroughly in 2D flow past a cylinder confined to a channel, so we are confident that it will give accurate results in the periodic case.

Our notation for parts of the domain will be the same as the previous chapter. That is,

- Computational domain, $\mathcal{C}$,
- Fluid domain (outside of the cylinder), Ω,
- Non-physical domain (inside cylinder), $\mathcal{E}$,
- Cylinder boundary Γ,
- Force locations γ.

6.1.1. The Stokes-Oldroyd B model. We will work with the Stokes-Oldroyd B model for fluid velocity $\mathbf{u}$, pressure p, and polymer conformation tensor $\mathbf{C}$. The model was first introduced in Section 1.2 and again in Section 2.2 with a stress diffusion term, $\nu\Delta\mathbf{C}$. We drop that term here to facilitate our use of the square root method, which is described in Section 6.2.1. Without stress diffusion, the Stokes-Oldroyd B model is

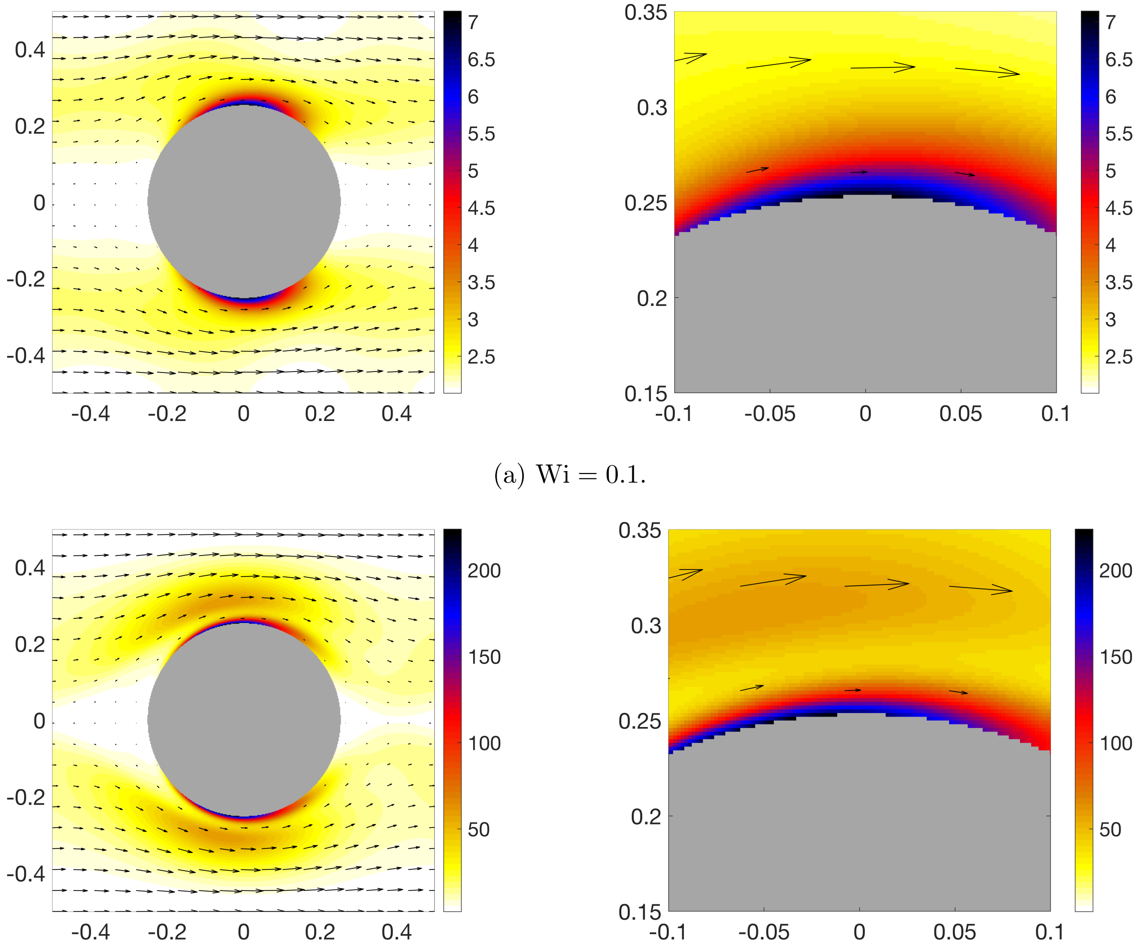
$$\Delta\mathbf{u} - \nabla p + \mathbf{f} + \frac{\xi}{\lambda}\nabla\cdot\mathbf{C} = 0, \tag{6.1}$$

$$\nabla\cdot\mathbf{u} = 0, \tag{6.2}$$

$$\partial_t\mathbf{C} + \mathbf{u}\cdot\nabla\mathbf{C} - \left(\nabla\mathbf{u}\mathbf{C} + \mathbf{C}\nabla\mathbf{u}^T\right) = -\frac{1}{\lambda}(\mathbf{C} - \mathbf{I}). \tag{6.3}$$

ξ is the ratio of polymer viscosity to solvent viscosity, and we will use $\xi = 1/2$ throughout our work. λ is the polymer relaxation time, which we will vary. The key dimensionless parameter for viscoelastic flow is the Weissenberg number, Wi, which represents the ratio of the timescale of polymer relaxation to the timescale of the flow. For flow around a cylinder, Wi is typically defined as $\text{Wi} = \lambda V/(2R)$, where V is the average horizontal velocity and R is the cylinder radius. In this chapter, we will use $R = \sqrt{0.2/\pi}$ and $V = 1/2$ (note that we used $V = 1$ for Stokes flow in Chapter 5). This gives us $\text{Wi} \approx \lambda$. The trace of the conformation tensor, $\text{tr}\mathbf{C}$, gives the strain energy density of the flow.

The forcing term $\mathbf{f}$ consists of two types of forces. The first is point forces placed inside the cylinder which enforce no-slip boundary conditions. The second is a background force that maintains the flow rate. These forces are determined using the methods developed in Section 5.1.3.



(a) $\mathrm{Wi} = 0.1$.

(b) $\mathrm{Wi} = 0.7$.

FIGURE 6.1. Viscoelastic flow around a cylinder with periodic boundary conditions, simulated with the DIB method. Colors show the strain energy density, $\mathrm{tr}\mathbf{C}$, and arrows show the velocity. The amount of elasticity is small in (a). It is near the upper limit of our (and most others) computational abilities in (b). Note that the strain energy is maximized near the cylinder boundary, illustrating the importance of obtaining convergent velocity gradients near boundaries.

6.2. Coupling DIB with Viscoelasticity

6.2.1. The square root method. In our studies of viscoelastic Kolmogorov flow (Chapters 2-4), we performed direct numerical simulations which evolved the conformation tensor, $\mathbf{C}$. Now, we take a different approach and use the square root method, which was developed in [**3**]. This method is used to improve numerical stability as well as to aid in our comparisons with IBSE, which also implemented the square root method.

This method relies on the existence of a unique symmetric positive definite matrix, $\mathbf{b}$, satisfying $\mathbf{b}^2 = \mathbf{C}$. The existence and uniqueness of $\mathbf{b}$ relies on the fact that $\mathbf{C}$ itself is symmetric positive definite (see Theorem 7.2.6 in [24]). As one might expect, the square root method reformulates (6.3) in terms of $\mathbf{b}$, allowing us to evolve the square root. The resulting equation is

$$\partial_t \mathbf{b} + \mathbf{u} \cdot \nabla \mathbf{b} = \mathbf{b} \nabla \mathbf{u} + \mathbf{a}\mathbf{b} + \frac{1}{2\lambda}(\mathbf{b}^{-\top} - \mathbf{b}), \tag{6.4}$$

where

$$\mathbf{a} = \begin{pmatrix} 0 & a_{12} \\ -a_{12} & 0 \end{pmatrix} \tag{6.5}$$

with

$$a_{12} = \frac{b_{12}\partial_x u - b_{11}\partial_x v + b_{22}\partial_y u - b_{21}\partial_y v}{b_{11} + b_{22}}. \tag{6.6}$$

The determination of the matrix $\mathbf{a}$ is nontrivial and depends on the choice to constrain $\mathbf{b}$ to be symmetric. More details can be found in [3].

6.2.2. Timestepping. Our numerical simulations of this system use the same methods as the Kolmogorov flow simulations (see Section 2.3.2), with the necessary modifications for use of the square root method and the inclusion of an immersed boundary. Our initial condition for $\mathbf{b}$ will be the identity matrix, which corresponds to no polymer stress. This means that the initial velocity and pressure are (Newtonian) Stokes flow around a cylinder.

To advance in time from t to $t + \Delta t$, we first determine $\mathbf{b}(t + \Delta t)$ using $\mathbf{b}(t)$ and $\mathbf{u}(t)$. This step uses classical Runge-Kutta 4 (RK4). We use a psuedospectral method, with spatial derivatives computed in Fourier space, and nonlinear multiplication done in real space. Prior to this multiplication a spectral filter,

$$\Lambda(K_x, K_y) = \exp\left[-36\left(\frac{K_x}{\max(K_x)}\right)^{36}\right] \cdot \exp\left[-36\left(\frac{K_y}{\max(K_y)}\right)^{36}\right], \tag{6.7}$$

is applied to each factor [25].

Once $\mathbf{b}(t+\Delta t)$ is known, we use it to determine $\mathbf{u}(t+\Delta t)$. This velocity update boils down to solving Stokes equations with appropriate DIB forces, a process that requires three distinct steps:

(i) Determine the velocity on Γ (solve Stokes and interpolate) due to the polymer force alone.

(ii) Apply the regularized inverse of the Schur complement, $\mathbf{Z}$ (see Section 5.4.1), in order to determine the necessary DIB forces, driving force, and mean velocity that will negate the velocity from (i), thus enforcing the no-slip boundary condition.

(iii) Solve Stokes equations using the sum of the DIB forces from (ii), the polymer force, and the driving force.

When the velocity has been updated, the timestep is complete and we begin the next by again updating the square root of the stress. The timestep used is $\Delta t = 5 \cdot 10^{-3}$ on a 64×64 grid. The timestep is halved for each doubling of the grid resolution. We have run our simulations for 50 relaxation times to ensure that a steady state has been reached.

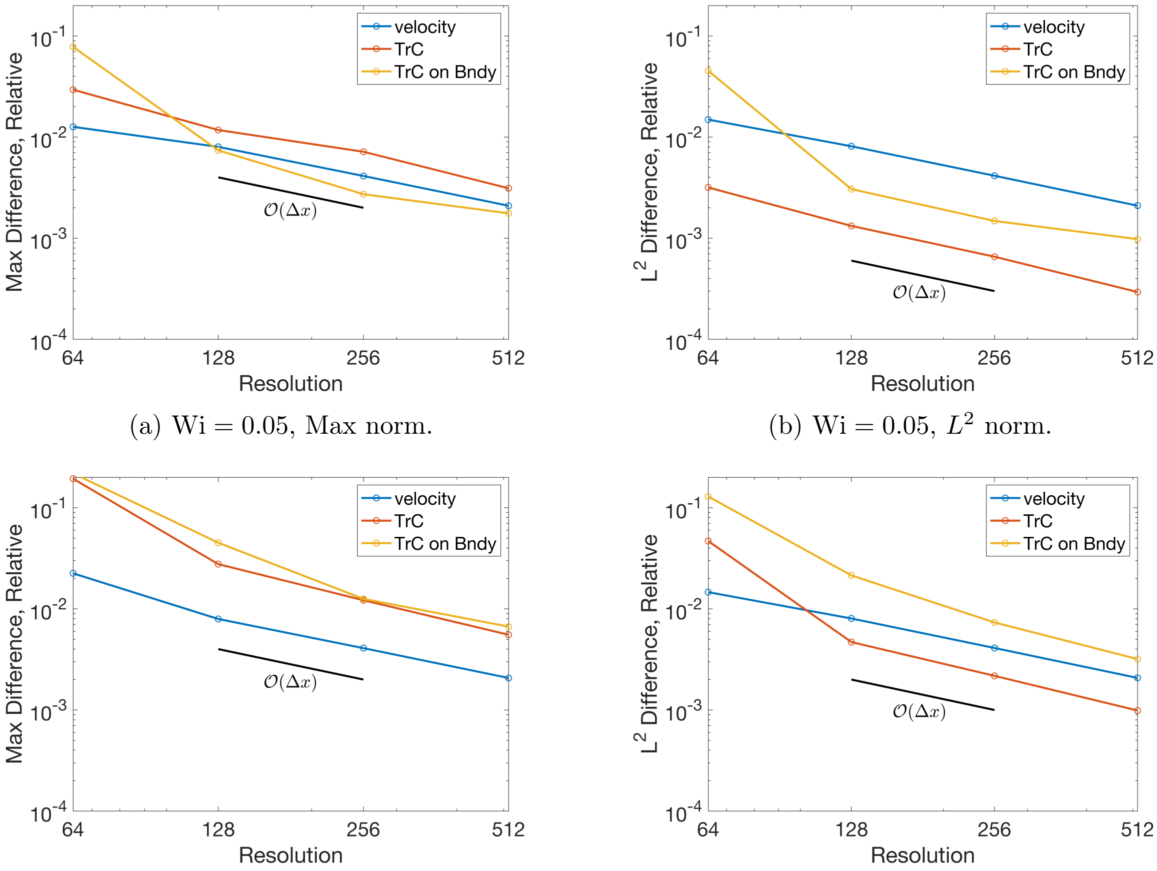
Note that the Schur complement does not change from one time step to the next, so it only needs to be computed once in a preprocessing step. It can also be stored and used for other simulations that use the same geometry.

We control the conditioning of the Schur complement using a singular value ratio cutoff of $\mathrm{SV_c} = 10^{-8}$. See Section 5.4 for more details. Different spaces between the boundary and force locations are used as needed for numerical stability. $\mathrm{Rr} = 0.9$ is used whenever possible, and works well in all simulations on 256×256 or finer grids. In the lower resolution tests with moderate to high Wi, using $\mathrm{Rr} = 0.9$ often leads to numerical instability. This may be a result of the fact that at low resolution, $\mathrm{Rr} = 0.9$ leaves very few grid cells in the gap between the force locations and the boundary. In the cases which were unstable with $\mathrm{Rr} = 0.9$, we reduced Rr in increments of 0.1 (down to a minimum of 0.6) until the simulation ran stably. In a few cases low resolution simulations never became stable, which is apparent in Figures 6.10, 6.12, and 6.13, which lack data points at low resolution.

6.3. Comparison between DIB and IBSE at low Wi

To evaluate the performance of the DIB method in simulations of viscoelastic flow around a cylinder, we will compare our results against the same flow simulated with the IBSE method. Specifically, we will compare the x-velocity and $\mathrm{tr}\mathbf{C}$ in the fluid domain as well as $\mathrm{tr}\mathbf{C}$ interpolated to the boundary. The differences in these quantities will be measured with both the L^2 and max norms. We will check that the differences converge as the grid is refined. We are comparing against 'IBSE-2' which, as shown in [57], achieves second order convergence of the polymer stress and velocity gradient.

Our comparisons begin with refinement studies at small Wi, Wi $= 0.05$ and Wi $= 0.10$, shown in Fig. 6.2. Convergence appears to be approximately first-order for all quantities. Comparing the results between Wi $= 0.05$ and Wi $= 0.10$, the stress differences (both on the boundary and in the fluid domain, and in both norms) become larger. The velocity differences on the other hand are essentially the same at Wi $= 0.10$ and Wi $= 0.05$.



(a) Wi $= 0.05$, Max norm.

(b) Wi $= 0.05$, L^2 norm.

(c) Wi $= 0.10$, Max norm.

(d) Wi $= 0.10$, L^2 norm.

FIGURE 6.2. Comparison between DIB and IBSE with Wi $= 0.05$ (top row) and Wi $= 0.10$ (bottom row). These simulations were run to 100 relaxation times. The differences are converging at approximately first-order in all cases.

6.4. Stabilizing DIB with artificial diffusion

At $\mathrm{Wi} = 0.2$, both IBSE and DIB become unstable, which is a common problem in simulations of viscoelastic fluids. Such computational challenges are discussed in Chapter 10 of [23]. A typical approach to stabilize simulations is to add artificial stress diffusion. We used this approach in our Kolmogorov flow simulations in Chapters 2 - 4, adding the term $\nu\Delta\mathbf{C}$ to the Stokes-Oldroyd B model. In our DIB simulations, we will use a similar strategy. However, because we are using the square root method we add the term $\zeta\Delta\mathbf{b}$ to Eq. (6.4). This term does not directly represent diffusion of polymer stress and should be thought of only as a numerical viscosity (the exact relation between $\Delta\mathbf{C}$ and and $\Delta\mathbf{b}$ is given in [3], but is not particularly important for our work). We will take $\zeta = (d \cdot \Delta x)^2$. This dependence of ζ on grid resolution means that the numerical viscosity is reduced as the grid is refined, and thus the model converges to Stokes-Oldroyd B with refinement. d is referred to as the diffusion constant and we will use either $d = 2$ or $d = 4$ depending on the parameters of a given simulation. In general, we strive to keep d as small as possible so that the non-physical effects of this added viscosity are minimized. As previously mentioned, timestepping of Eq. (6.4) is done with RK4. We do not add the diffusive term $\zeta\Delta\mathbf{C}$ into the RK4 step. Rather, a step of backward Euler adds the diffusion after the RK4 step, which makes our method a fractional stepping method.

Adding diffusion to IBSE allows us to run simulations at least up to $\mathrm{Wi} = 0.7$, which is the highest we will explore in this work. Diffusion is helpful to stabilize DIB, and using $d = 4$ allows us to simulate $\mathrm{Wi} = 0.2$. However, if we increase to $\mathrm{Wi} = 0.3$ we again lose stability. It is possible that a larger diffusion constant would allow simulations to run stably at $\mathrm{Wi} = 0.3$ or higher, but as previously mentioned we would like to keep diffusion small. Given that we have not advanced very far upwards in Wi, exploring additional ways to stabilize DIB seems prudent. First, we will return to $\mathrm{Wi} = 0.2$ and examine an unstable simulation to gain insight about how we might improve stability.

6.5. Stabilizing DIB with extensions on the nonphysical domain

Fig. 6.3 gives a detailed look at the growth of a numerical instability in the final moments of simulation at $\mathrm{Wi} = 0.2$, showing $\partial u/\partial x$ and $\mathrm{tr}\mathbf{C}$ around the force locations and cylinder. As time advances, we see grid scale oscillations emerge near the cylinder boundary. While this is reminiscent of the numerical artifacts resulting from poor conditioning in Fig. 5.8, the current situation is quite distinct. The conditioning of the Schur complement is being controlled, but we are now simulating a time dependent system in which the polymer stress has a 'memory' of past deformations. This may be causing numerical issues as large stresses build up inside the cylinder.

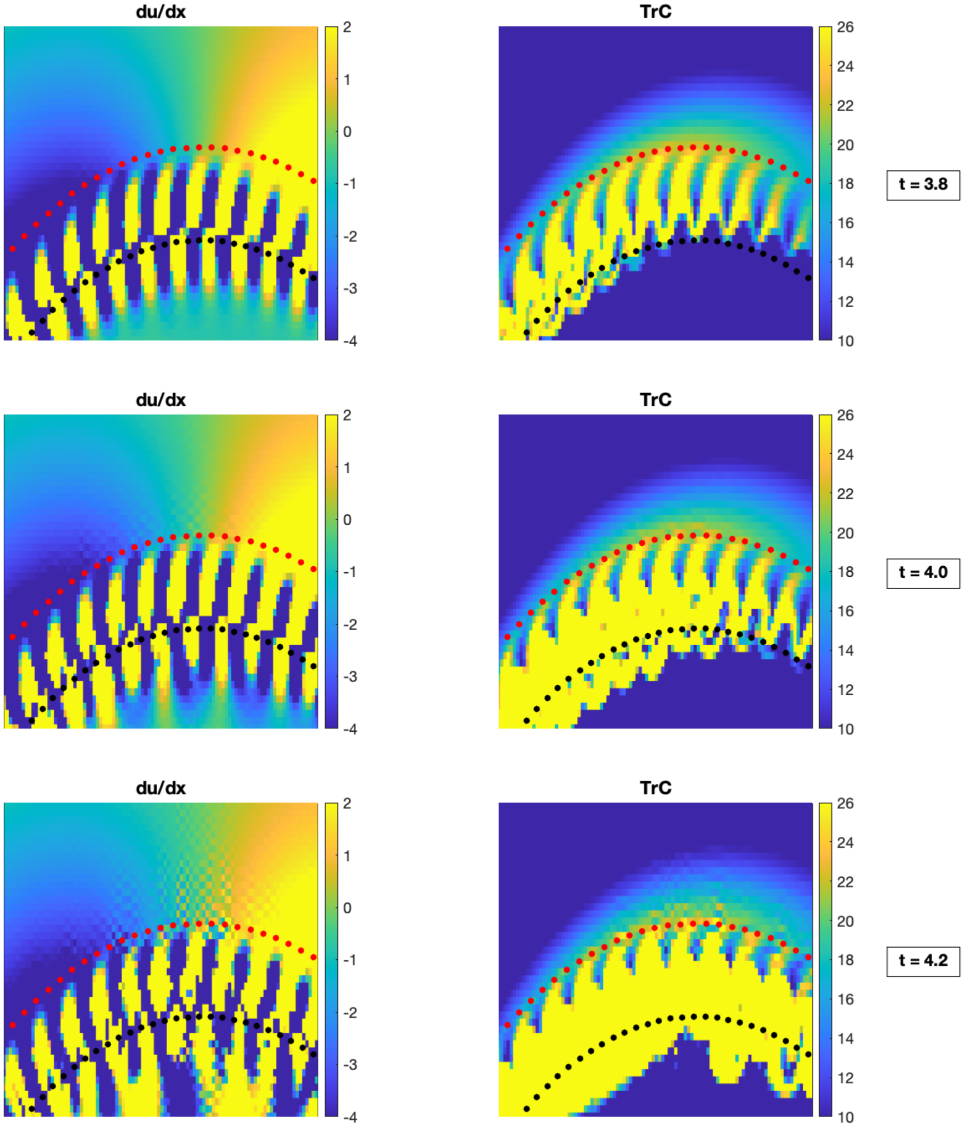


FIGURE 6.3. Visualizations of the $\partial u/\partial x$ and tr$\mathbf{C}$ as blow-up approaches. The cylinder boundary is shown with red dots and the force locations with black dots. This simulation uses a diffusion constant of 2 at Wi $= 0.2$ and $Ny = 512$. Time is increasing from top to bottom. Blow-up occurred at $t = 4.234$. The color axis has been restricted in order to make the grid scale oscillations visible. The largest values of $\partial u/\partial x$ in the panels shown here are (top to bottom) approximately ±80, ±120, and ±150. The maximum values of tr$\mathbf{C}$ are approximately 250, 600, and 1900.

After inspecting Fig. 6.3, it may seem that the core idea of DIB – to move the point forces away from the boundary – has created a new problem around the force locations. Before giving up hope, it would behoove us to recall that the forces are located within the *non-phyiscal* domain. In this region we are not attempting to find a physically relevant solution, and so we may modify the solution here in any way we please, so long as the solution on the fluid domain is left intact. There are many possible ways to modify the solution on the non-physical domain. We could, for example, set the velocity to zero or the stress to the identity each time they are computed in the timestepping loop. However, we must take care to maintain the accuracy of the velocity gradient and stress near the boundary, and avoid creating new instabilities.

We will view the task as an extension problem. That is, given some part (velocity, stress, etc.) of the solution, which we denote $\tilde{\mathbf{h}}$, we will compute a new function, $\mathbf{z}$, on the entire computational domain. $\mathbf{z}$ will be designed to match the values and possibly some number of derivatives of $\tilde{\mathbf{h}}$ on the cylinder boundary, Γ. We then create the extended function $\mathbf{h}$ by stitching together $\mathbf{z}$ and $\tilde{\mathbf{h}}$ along Γ, retaining $\mathbf{z}$ only on the nonphysical domain and keeping the original solution, $\tilde{\mathbf{h}}$, on the fluid domain. Recalling that we represent the fluid domain as Ω and the nonphysical domain as $\mathcal{E}$, this gives us the extended function

$$\mathbf{h} = \tilde{\mathbf{h}} \cdot \chi_\Omega + \mathbf{z} \cdot \chi_\mathcal{E}. \tag{6.8}$$

With this approach comes the choice of how smooth to make $\mathbf{h}$. If the extension is smooth away from the boundary, which will be the case with our methods, then the smoothness of $\mathbf{h}$ only depends on what happens at the boundary. The simplest choice is to force $\mathbf{z}$ to match only the values of the original solution along the boundary. Thus, we expect the extended function to be C^0, with a gradient discontinuity across the boundary. Another option is to match values and the first n derivatives, thus giving a C^n extension. Note that constructing such an extension requires the nontrivial task of estimating derivatives of $\tilde{\mathbf{h}}$ at the boundary.

As for what part of the solution to extend, the stress is perhaps the most natural choice, because it is the uncontrolled growth of the stress that causes DIB to crash. Stress extensions will be developed and tested in the next several sections. Note that, owing to our use of the square root method, we will actually extend the square root of the stress, $\mathbf{b}$. We will still refer to this as a "stress extension." The ideas we discuss also apply to extending $\mathbf{C}$ for simulations which evolve

Eq. (6.3) directly. In addition to stress extensions, it is reasonable to consider extensions to the velocity or the velocity gradient, because these both influence the evolution of the stress. Choosing which of these to pursue (or both) is worth a short discussion.

Consider an extension to the velocity. Inspecting Eq. (6.3), we see that the velocity itself is only present in the material derivative of $\mathbf{C}$, and thus only advects the stress. The velocity gradient, on the other hand, is directly responsible for growth of the stress. Thus, the most important effect of a velocity extension will be the fact that it also acts as a velocity gradient extension. The gradient extension will of course have one degree lower regularity than the velocity extension. This is important because, from Eq. (6.3), we expect the regularity of the stress to match that of the velocity gradient. So if a certain regularity, say C^k, of the stress is desired, then we will need a C^{k+1} extension of the velocity.

Next, consider an extension to the velocity gradient. This approach does *not* give us a velocity extension as a by-product, but this may not matter because the velocity only advects stress. Also, supposing again that we need a C^k stress for stability or accuracy, we can obtain this with a C^k velocity gradient extension. For the same stress regularity then, we can construct a velocity gradient extension that has lower regularity (C^k) than the corresponding velocity extension (C^{k+1}). Constructing extensions with lower regularity is generally easier than with higher regularity, so the velocity gradient extension has an advantage over the velocity extension. Thus, we will not pursue velocity extensions, but will move forward with velocity gradient extensions, as well as the previously mentioned stress extensions.

As an aside, we note a few other variations on the theme of extensions. It is reasonable to use multiple types of extensions simultaneously, stress and velocity for example. To increase efficiency, we could compute extensions less frequently than every time step (perhaps even in an adaptive manner), though care would need to be taken to maintain stability. It is also possible to stitch together the extension and the original function along a curve (inside the nonphysical domain) other than the boundary, Γ. In principle, relocating this curve could take any issues related to the stitching and relocate them away from the fluid domain. We have done some limited testing of this last idea, however, and found it to be destabilizing. Perhaps if implemented in a different way, this would not be the case!

Clearly, there are many options before us. Exploring them all is beyond the scope of our work, so we will move forward only with extensions to the square root of the stress and extensions to the velocity gradient. For each, we will test both C^0 and C^1 extensions. We choose these extensions because they are relatively simple, they have good potential to stabilize DIB, and they only require estimating first order derivatives on the boundary.

Incorporating the extensions into the time stepping loop (see Section 6.2.2) is simple. For simulations using stress extensions, the stress is extended immediately after each stress update, i.e. $\mathbf{b}(t+\Delta t)$, is computed. Thus, the velocity will be updated using the extended stress. For simulations using velocity gradient extensions, the gradient is extended immediately after each velocity update, i.e. $\mathbf{u}(t+\Delta t)$, is computed.

6.6. Methods for building extensions

6.6.1. C^0 extension methods. Our method for building a C^0 extension will be the same for extending both the stress and the velocity gradient. As such, we present the method with $\tilde{\mathbf{h}}(x, y)$ representing the function to be extended, be it $\nabla\mathbf{u}$ or $\mathbf{b}$. Our intention is to find a smooth extension $\mathbf{z}_0(x, y)$ which will replace $\tilde{\mathbf{h}}$ on the non-physical domain, $\mathcal{E}$. Of course, we do not modify $\tilde{\mathbf{h}}$ on the fluid domain, Ω. Thus, we will combine $\mathbf{z}_0$ and $\tilde{\mathbf{h}}$ into the extended function

$$\mathbf{h} = \tilde{\mathbf{h}} \cdot \chi_\Omega + \mathbf{z}_0 \cdot \chi_{\mathcal{E}}. \tag{6.9}$$

Our goal is that $\mathbf{h}$ does not create the numerical instabilities that $\tilde{\mathbf{h}}$ otherwise would. One way to create a smooth extension whose values we can specify on the boundary is to solve a PDE subject to boundary conditions, an idea inspired by the extensions used in the IBSE method [**56**]. Conveniently, we can use the same Schur complement approach which we developed for solving Stokes equations (see Section 5.1.3) to solve our extension PDEs. For a C^0 extension, we will solve

$$\Delta\mathbf{z}_0 - c\mathbf{z}_0 = S\mathbf{G}, \tag{6.10}$$

$$S^*\mathbf{z}_0 = S^*\tilde{\mathbf{h}}, \tag{6.11}$$

where $S\mathbf{G}$ represents point forces located on Γ that have been spread to the grid. These forces are determined by the constraint equation (6.11).

One could solve a variety of different PDEs to obtain an extension, and our choice in (6.10) follows the ideas used to developed IBSE. In order to match values on the boundary and give a smooth solution away from the boundary, the Poisson equation may at first seem like a good choice. However, because we solve the PDE in Fourier space with periodic boundary conditions, this would give us a singular problem. The term $-c\mathbf{z}_0$ is added in order to shift the eigenvalues of the Laplacian (which are nonpositive) away from 0. The value used for c will affect both the conditioning of the problem and the accuracy of the method. The trade-off between these two effects was explored in [**56**], and based upon that work, we take $c = -0.1$. Tuning this parameter specifically for DIB could be useful for future applications, but is beyond the current scope.

Before we move on, note that the forces represented by $S\mathbf{G}$ create a gradient discontinuity in $\mathbf{z}_0$ in the same way that traditional immersed boundary forces create a velocity gradient discontinuity. However, because our extension is only meant to be C^0, the gradient discontinuity introduced by $S\mathbf{G}$ is irrelevant. This fact is useful because, unlike when we solve Stokes, we can place our forces on Γ and avoid conditioning issues like those seen in Section 5.3.

We solve the system given in Eqs. (6.10)-(6.11) using the same approach as used for solving Stokes equations, which was discussed in Section 5.1.3. The first step is to determine the forces which are required to satisfy the boundary condition. To do this, we construct a Schur complement for the system, $\mathbf{M}_{\mathbf{z}_0}$, which is an operator that maps a set of forces, $\mathbf{G}$, to the boundary values that it will produce. Inverting Eq. (6.10) and interpolating to the boundary allows us to define this operator,

$$\mathbf{M}_{z_0}(\mathbf{G}) = S^* \left(\Delta - c\mathbf{I}\right)^{-1} S\mathbf{G}. \tag{6.12}$$

We can then form the matrix representation of this operator directly, with repeated application to unit vectors. That is,

$$(\boldsymbol{m}_{z_0})_j = \mathbf{M}_{\mathbf{z}_0}(\boldsymbol{e}_j), \tag{6.13}$$

where $(\boldsymbol{m}_{z_0})_j$ represents the j^{th} column of $\mathbf{M}_{z_0}$.

Once the Schur complement is formed (which only happens once, in preprocessing), its inverse is used to determine the forces which are required in order to satisfy (6.11). That is,

$$\mathbf{G} = \mathbf{M}_{z_0}^{-1}(\tilde{\mathbf{h}}). \tag{6.14}$$

These forces are then used as the right hand side of Eq. (6.10), which we can solve efficiently with a Fourier method to obtain the extension. The final step is to stitch together the extension with the original function in accordance with Eq. (6.9).

6.6.2. C^1 extension methods. After we examine some results, it will become clear that smoother extensions will be useful, so we will now discuss our methods to build C^1 extensions of the stress and the velocity gradient. The procedure for making these extensions is very similar to the procedure used for C^0 extensions, but we will need to modify it so that we can control the derivatives of the extension on the boundary. To do this, we use a higher order PDE so that both values and derivatives can be specified as boundary conditions. This method will require new spread and interpolation operators to handle normal derivatives at the boundary. These operators will rely on the normal derivative of the delta function defined in Section 5.1.1. The original delta function is

$$\delta_h = \frac{1}{h^2}\phi\left(\frac{x}{h}\right)\phi\left(\frac{y}{h}\right), \tag{6.15}$$

with $h = \Delta x = \Delta y$ and

$$\phi(r) = \begin{cases} 1/8\left(3 - 2|r| + \sqrt{1 + 4|r| - 4r^2}\right), & 0 \le |r| < 1 \\ 1/8\left(5 - 2|r| - \sqrt{-7 + 12|r| - 4r^2}\right), & 1 \le |r| \le 2 \\ 0, & |r| > 2. \end{cases} \tag{6.16}$$

If we represent the boundary's normal vector as $\boldsymbol{n} = (n_x, n_y)$, then the normal derivative of the delta function is

$$\boldsymbol{n}\cdot\nabla\delta_h = \frac{1}{h^2}\left(n_x\phi'\left(\frac{x}{h}\right)\phi\left(\frac{y}{h}\right) + n_y\phi\left(\frac{x}{h}\right)\phi'\left(\frac{y}{h}\right)\right) \tag{6.17}$$

with

$$\phi'(r) = \begin{cases} 1/4\left(1 + \frac{3+2r}{\sqrt{-7-12r-4r^2}}\right), & -2 \le r < -1 \\ 1/4\left(1 - \frac{1+2r}{\sqrt{1-4r-4r^2}}\right), & -1 \le r < 0 \\ 1/4\left(-1 + \frac{1-2r}{\sqrt{1+4r-4r^2}}\right), & 0 \le r < 1 \\ 1/4\left(-1 - \frac{3-2r}{\sqrt{-7+12r-4r^2}}\right), & 1 \le r \le 2 \\ 0, & |r| > 2. \end{cases} \tag{6.18}$$

Our new spread operator, S_n, is defined as

$$(S_n\mathbf{G})(\mathbf{x}) = \int_\Gamma \mathbf{G}(s)\left[\boldsymbol{n}\cdot\nabla\delta_h(\mathbf{x}-\mathbf{X}(s))\right] ds. \tag{6.19}$$

We can think of $\mathbf{G}$ in Eq. (6.19) as a collection of dipole strengths located at boundary points. The effect of S_n is to create an approximation to these dipoles which is supported on grid points near the boundary. We will refer to S_n as the gradient spread operator.

The adjoint, S_n^*, is an interpolation operator which interpolates a function's derivative in the direction normal to the boundary. It is defined as

$$(S_n^*\mathbf{u})(s) = \int_{\mathcal{C}} \mathbf{u}(\mathbf{x})\left[\boldsymbol{n}\cdot\nabla\delta_h(\mathbf{x}-\mathbf{X}(s))\right] d\mathbf{x}. \tag{6.20}$$

With these operators in hand, we can write down the PDE that we will solve for the extension $\mathbf{z}_1$. As with the C^0 extensions in the previous section, $\tilde{\mathbf{h}}$ will represent the function to be extended. $\mathbf{z}_1$ is subject to the constraints that it matches both the value and the normal derivative of $\tilde{\mathbf{h}}$ on the boundary. The system we will solve is

$$\Delta^2\mathbf{z}_1 + c\mathbf{z}_1 = S\mathbf{G}_1 + S_n\mathbf{G}_2, \tag{6.21}$$

$$S^*\mathbf{z}_1 = S^*\tilde{\mathbf{h}}, \tag{6.22}$$

$$S_n^*\mathbf{z}_1 = S_n^*\tilde{\mathbf{h}}. \tag{6.23}$$

Our choice of PDE is again inspired by IBSE [56] and the reasoning behind the choice follows the same logic as was used for the C^0 extensions. That is, we choose a PDE that can satisfy the required boundary conditions and produce smooth solutions away from the boundary. The term

$+c\mathbf{z}_1$ is included to make the problem nonsingular by shifting the eigenvalues of the differential operator (which are nonnegative) away from 0. As before, we take $c = 0.1$.

Note that we use two forcing terms: a set of point forces that are spread to the grid, $S\mathbf{G}_1$, and a set of dipoles that are spread to the grid, $S_n\mathbf{G}_2$. In an informal sense, we use these forcing terms to enforce the boundary value condition (6.22) and the normal derivative condition (6.23), respectively. Of course in reality the forcing terms are coupled and do not act independently.

We solve this PDE with a Schur complement approach, as we have done with the C^0 extensions in the previous section and with Stokes equations in section 5.1.3. That is, in preprocessing we construct the Schur complement, $\mathbf{M}_{z_1}$, which is the mapping from forces, $\mathbf{G}_1$, and $\mathbf{G}_2$, to the boundary values and normal derivatives that they produce. To define this mapping, we first invert Eq. (6.21). We then interpolate to the boundary, and each interpolation operator gives us an equation to work with. So, we have

$$\mathbf{M}_{z_1}\begin{pmatrix}\mathbf{G}_1\\ \mathbf{G}_2\end{pmatrix} = \begin{pmatrix} S^*\left(\Delta^2 + c\mathbf{I}\right)^{-1}\left(S\mathbf{G}_1 + S_n\mathbf{G}_2\right) \\ S_n^*\left(\Delta^2 + c\mathbf{I}\right)^{-1}\left(S\mathbf{G}_1 + S_n\mathbf{G}_2\right)\end{pmatrix}. \tag{6.24}$$

We use this definition to form the Schur complement with repeated application to unit vectors.

Once the Schur complement is formed, its inverse is used to determine the forces which are required in order to satisfy (6.22) and (6.23). That is,

$$\begin{pmatrix}\mathbf{G}_1\\ \mathbf{G}_2\end{pmatrix} = \mathbf{M}_{z_1}^{-1}\begin{pmatrix} S^*\tilde{\mathbf{h}} \\ S_n^*\tilde{\mathbf{h}}\end{pmatrix}. \tag{6.25}$$

These forces and dipoles are then used as the right hand side of Eq. (6.21), which we can solve efficiently with a Fourier method to obtain the extension $\mathbf{z}_1$. Finally, we stitch together the extension with the original function,

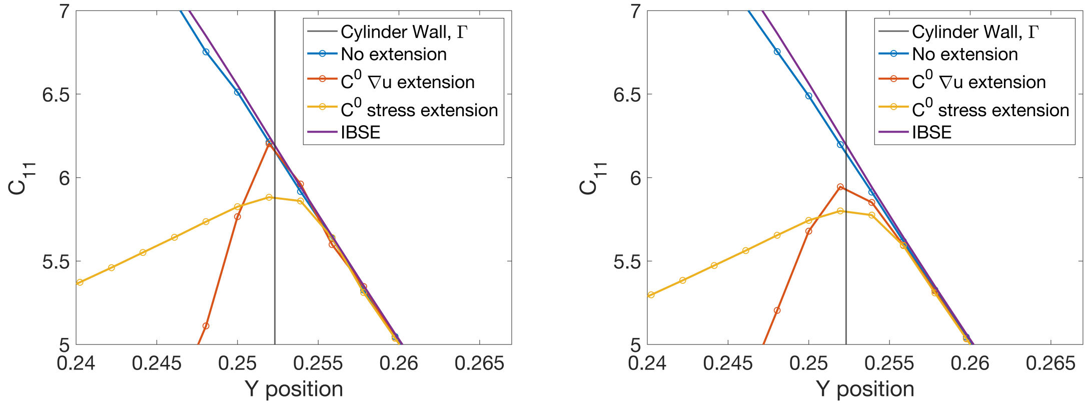
$$\mathbf{h} = \tilde{\mathbf{h}} \cdot \chi_\Omega + \mathbf{z}_1 \cdot \chi_{\mathcal{E}}. \tag{6.26}$$

With these tools developed to stabilize DIB, we will resume our testing of flow around a cylinder, and evaluate our four chosen extensions at increasing values of Wi.

6.7. Results with extensions at small Wi

We now return to comparisons against IBSE in our test problem, viscoelastic flow around a cylinder. Our previous tests, in Section 6.3, were successful up to Wi = 0.1 and we left off by mentioning that artificial diffusion allows for a small increase to Wi = 0.2. Testing our extensions will begin back at Wi = 0.1 so that we may compare the performance of DIB with and without extensions. This approach will also allow us to examine results using extensions with and without diffusion, which will reveal an important insight.

6.7.1. C^0 extensions. Fig. 6.4 displays vertical slices, taken at $x = 0$, of the first component of the conformation tensor, $\mathbf{C}_{11}$, across the cylinder boundary. Taking $x = 0$ allows us to view a computationally challenging region in the vicinity of the maximum stress in the fluid domain. With no diffusion (6.4a), and with a diffusion constant $d = 2$ (6.4b), we compare results from four simulations: IBSE, DIB without extensions, and DIB with each type of C^0 extension.



(a) No diffusion (b) Diffusion constant = 2

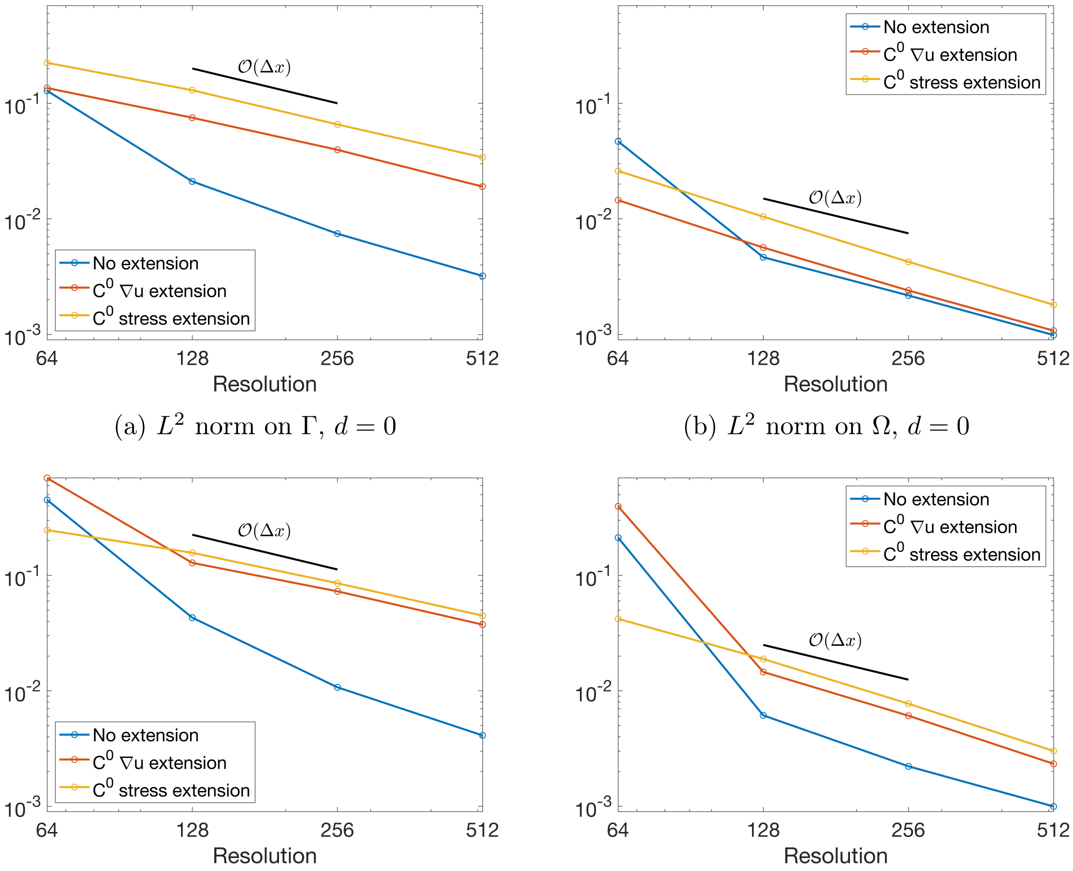
FIGURE 6.4. Vertical slices (at $x = 0$) of $\mathbf{C}_{11}$. These simulations are run with $\mathrm{Wi} = 0.1$ on a 512×512 grid. The fluid domain is on the right side of each plot and the nonphysical domain is on the left. Both C^0 extensions lead to abrupt changes in the gradient of the stress across the boundary. This is cause for concern if the diffusive term $\zeta\Delta\mathbf{b}$ is used, because diffusion will smooth out these changes. We can see the effect here, most clearly in the $\nabla\mathbf{u}$ extension, where adding diffusion causes the stress to drop away from the IBSE solution.

At this low value of Wi = 0.1, extensions are not yet necessary for stability, and IBSE is in close agreement with the extension-free DIB method. When extensions are used, the stress is continuous but not smooth around the boundary, which is the expected result with these extension methods.

In the case of a C^0 extension to $\mathbf{b}$, we are extending the stress at every timestep with a function which matches the values on the boundary but has no derivative constraints. Thus, the gradient discontinuity is an immediate consequence. To understand the lack of smoothness observed in $\mathbf{C}_{11}$ when using a C^0 extension to $\nabla\mathbf{u}$, recall that $\nabla\mathbf{u}$ is responsible for the growth in the stress. At each timestep, we are introducing a gradient discontinuity in the growth terms of 6.3, and thus we expect the steady state solution for $\mathbf{C}$ to contain a gradient discontinuity across the boundary as well.

In isolation these changes in the gradient across the boundary are not a great concern, because the solution on the fluid domain is not necessarily affected. However, introducing diffusion changes the situation because it smooths out the stress in areas with large second order derivatives, such as the location visualized in Fig. 6.4. We can see this effect most clearly in the C^0 $\nabla\mathbf{u}$ extension in Fig. 6.4a, where the peak stress notably drops away from the IBSE solution when diffusion is added. The first grid point on the interior of Ω (to the right of the boundary) has also notably dropped away from the IBSE solution. Though it is harder to see, the C^0 stress extension has a similar change when diffusion is added. At higher Wi, shown in Figs. 6.8 and 6.11, this effect is much more pronounced.

Fig. 6.5 shows results of refinement studies in which we compare tr$\mathbf{C}$ between IBSE to DIB. Differences are measured with the L^2 norm and shown separately when the norm is taken along the boundary, Γ, and when it is taken on the entire fluid domain, Ω. These simulations were run with $\mathrm{Wi} = 0.1$ with both $d = 0$ and $d = 2$.



(a) L^2 norm on Γ, $d = 0$

(b) L^2 norm on Ω, $d = 0$

(c) L^2 norm on Γ, $d = 2$

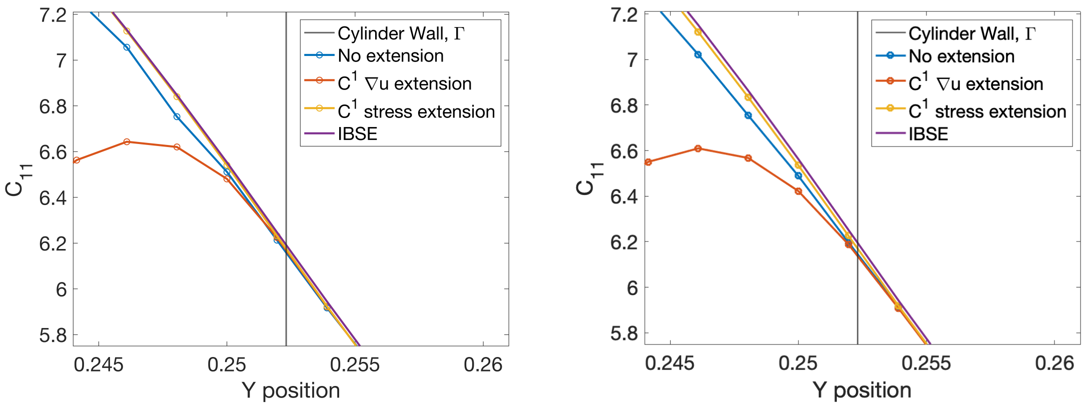
(d) L^2 norm on Ω, $d = 2$

FIGURE 6.5. Comparison of DIB and IBSE using C^0 extensions. Plots show differences in tr$\mathbf{C}$, at Wi $= 0.1$ without diffusion (top row) and with a diffusion constant $d = 2$ (bottom row). On the boundary, Γ, both extensions create errors. On the fluid domain, Ω, the $\nabla\mathbf{u}$ extension performs approximately as well as using no extension when no diffusion is used. Once diffusion is added, the two extensions perform similarly.

It is apparent in Fig. 6.5 that in all cases we have approximately first-order convergence. Examining differences on the boundary (Figs. 6.5a, 6.5c), it seems that both C^0 extensions contribute to notably larger differences from IBSE, compared to using no extension. This result is expected, because the boundary values are found by interpolating from grid points near the boundary. Thus, the changes in the stress on the non-physical domain that we observed in Fig. 6.4 are picked up in the interpolation. On the other hand, when the norm is taken on the fluid domain (Figs. 6.5b, 6.5d), everything inside the cylinder is ignored. In this case, the C^0 $\nabla\mathbf{u}$ extension seems to perform quite well without diffusion. As we have just discussed however, adding diffusion significantly modifies the stress near the boundary. Thus, when diffusion is used (which will be required at higher Wi) in conjunction with C^0 extensions, we seem to be faced with an unavoidable loss of accuracy near the boundary. As we will see later, the C^0 extensions do contribute greatly to stability. This, together with their simplicity, could make them a useful tool in the right context.

6.7.2. Results with C^1 extensions. We now examine results with C^1 extensions to the square root of the stress and to the velocity gradient. As in the previous section, we begin by inspecting slices of $\mathbf{C}_{11}$ before moving on to refinement studies.

Fig. 6.6 shows vertical slices of the first component of $\mathbf{C}_{11}$ across the cylinder boundary. This is the same location and uses the same parameters as the results shown in Fig. 6.4. The tests differ only in the type of extension used. We can see that the gradient of the stress is much smoother across the boundary when C^1 extensions are used, compared to the C^0 extensions. Adding diffusion also seems to have much less effect on the values of the stress.

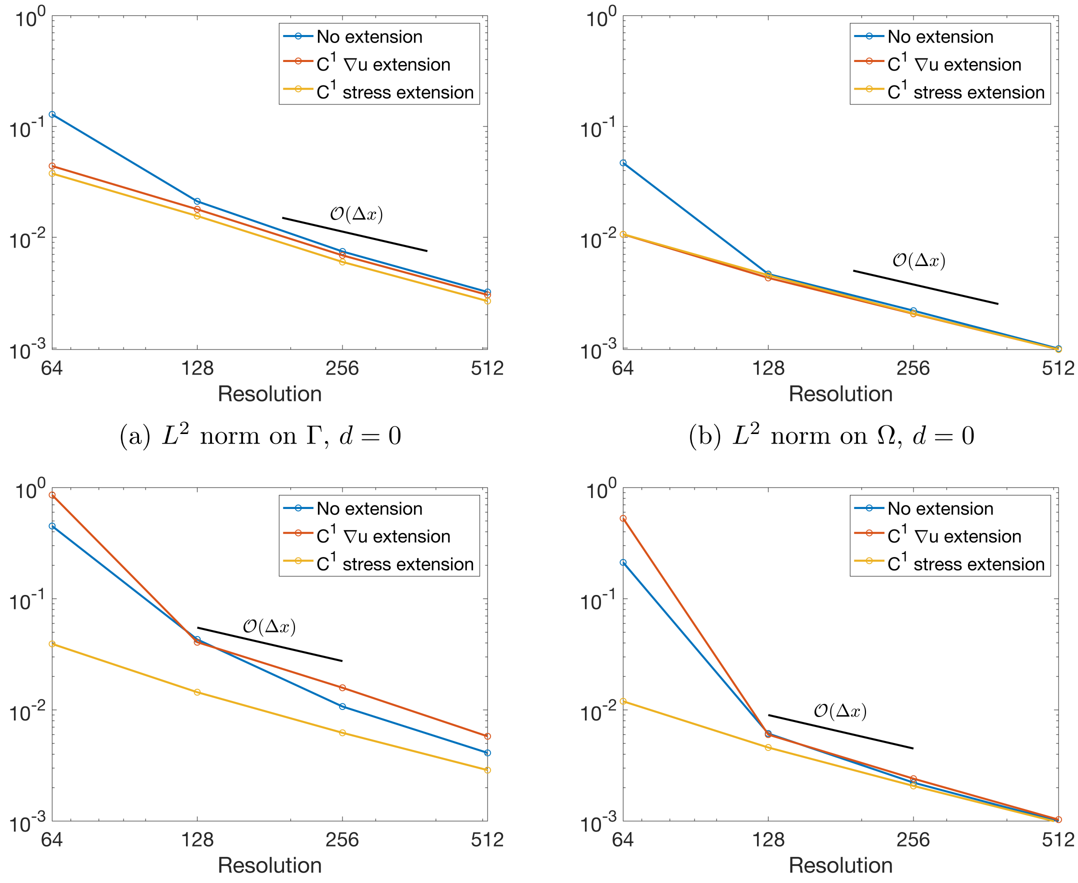


(a) No diffusion

(b) Diffusion constant = 2

FIGURE 6.6. Vertical slices (at $x = 0$) of $\mathbf{C}_{11}$ using C^1 extensions. These simulations are run with Wi = 0.1. The fluid domain is on the right side of the plot. Both extensions offer a noticeable improvement over the C^0 extensions when diffusion is used (compare to Figure 6.4b).

Fig. 6.7 shows the results of our refinement studies at Wi = 0.1 using C^1 extensions. In contrast to the C^0 extensions, the current results are generally as good or better than what we obtain with no extensions. This is a significant accomplishment given that, as discussed in Section 6.5, at higher Wi we will not have the luxury of using DIB without extensions. In both the refinement studies and the examination of the stress across the boundary, extending the stress gives slightly more accurate results (in comparison to IBSE) than extending the velocity gradient. This difference will become more pronounced at higher Wi.



(a) L^2 norm on Γ, $d = 0$

(b) L^2 norm on Ω, $d = 0$

(c) L^2 norm on Γ, $d = 2$

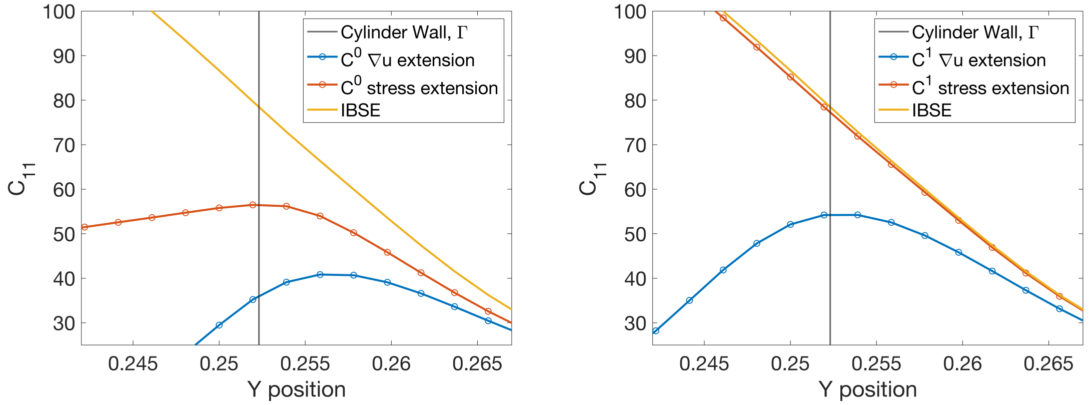
(d) L^2 norm on Ω, $d = 2$

FIGURE 6.7. Comparison of DIB and IBSE using C^1 extensions. Plots show differences in tr$\mathbf{C}$, at Wi $= 0.1$ without diffusion (top row) and with a diffusion constant $d = 2$ (bottom row). This is a notable improvement over the C^0 extensions shown in Fig. 6.5.

6.8. Results at moderate and high Wi

We now increase Wi to 0.4 and compare our extension methods. With a diffusion constant of $d = 2$, we found that DIB was not always stable, particularly on coarse grids. Additionally, on finer grids, convergence to IBSE was not consistent. It is possible that tuning the parameters used in DIB could improve performance in these tests. However, we plan to simulate as high as Wi $= 0.7$, which will require more diffusion. So, for our testing at Wi $= 0.4$, we will content ourselves with $d = 4$.

Fig. 6.8 shows vertical slices of $\mathbf{C}_{11}$, again taken across the cylinder boundary at $x = 0$. We now display the C^0 and C^1 extensions side-by-side. It is interesting to note that while the C^1 extensions did not show a large difference in accuracy at $\mathrm{Wi} = 0.1$, when we move up to $\mathrm{Wi} = 0.4$, the C^1 stress extension becomes the clear winner.

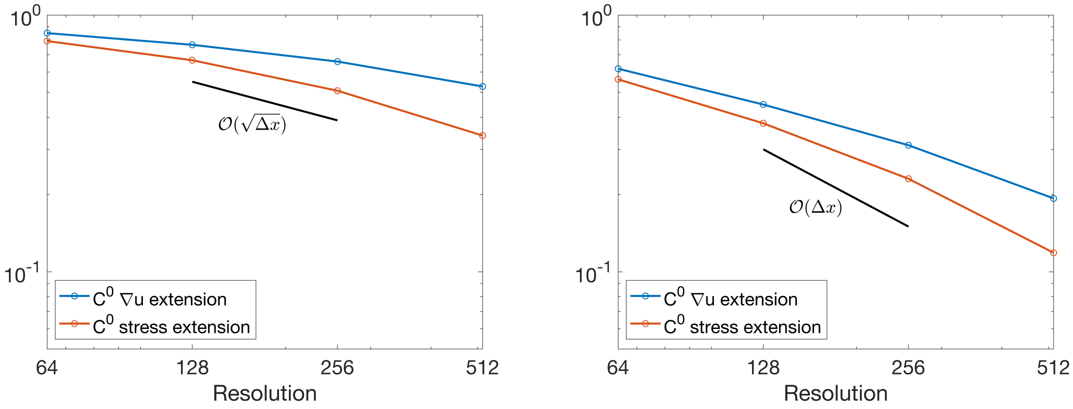


(a) C^0 extensions

(b) C^1 extensions

FIGURE 6.8. Vertical slices (at $x = 0$) of $\mathbf{C}_{11}$ using C^1 extensions. These simulations are run with $\mathrm{Wi} = 0.4$ and $d = 4$. The fluid domain is on the right side of the plot.

In Fig. 6.9, we show refinement study results for the C^0 extensions on the boundary and in the fluid domain. As expected based on the previous discussion about diffusion and Fig. 6.8a, these extensions provide very poor accuracy. Interestingly, we now see sublinear convergence on the boundary, and approximately first-order convergence in the fluid domain.

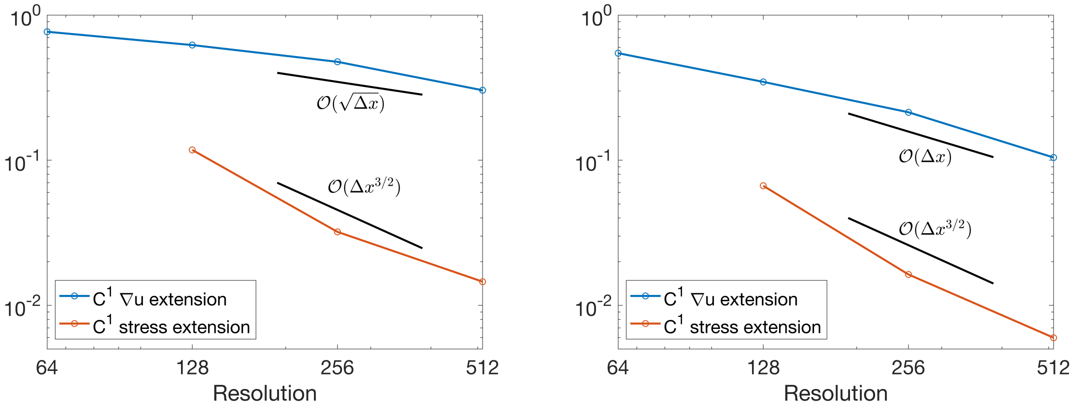


(a) L^2 norm on Γ

(b) L^2 norm on Ω

FIGURE 6.9. Comparison of DIB and IBSE using C^0 extensions. Plots show differences in $\mathrm{tr}\mathbf{C}$, at $\mathrm{Wi} = 0.4$ and $d = 4$ diffusion.

In Fig. 6.10, we show refinement study results for the C^1 extensions on the boundary and in the fluid domain. The velocity gradient extension performs poorly, which is not surprising, given the stress shown in Fig. 6.8b. It gives sublinear convergence on the boundary, and approximately first-order on the fluid domain, similar to the C^0 extensions. The C^1 stress extension performs much better, though it failed to run stably on the coarsest grid. Surprisingly, the convergence appears to be better than first-order both on and off the boundary. More testing would be needed to verify this result.



(a) L^2 norm on Γ

(b) L^2 norm on Ω

FIGURE 6.10. Comparison of DIB and IBSE using C^1 extensions. Plots show differences in tr$\mathbf{C}$, at Wi $= 0.4$ and $d = 4$ diffusion.

We now continue to our most challenging simulations, at Wi $= 0.7$, using diffusion with $d = 4$. This is the highest Wi at which IBSE has been tested, and is a common value around which numerical methods for viscoelastic fluids begin to fail [48]. Our simulations with the C^1 $\nabla \mathbf{u}$ extensions were not stable, so we only show results for the other three extensions.

Fig. 6.8 shows vertical slices of $\mathbf{C}_{11}$, again taken across the cylinder boundary at $x = 0$. Similar to our simulations at Wi $= 0.4$, the C^0 extensions produce a stress which is significantly smaller than IBSE near the boundary. The C^1 extension to the square root of the stress still appears to work well.

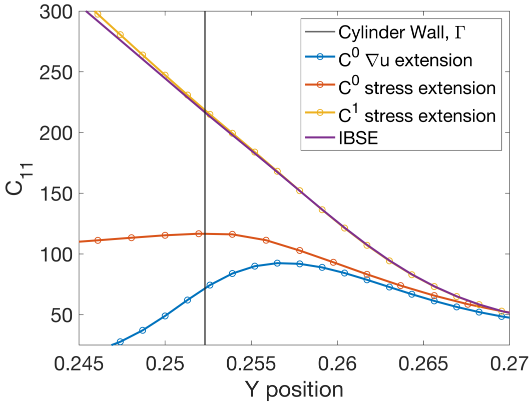
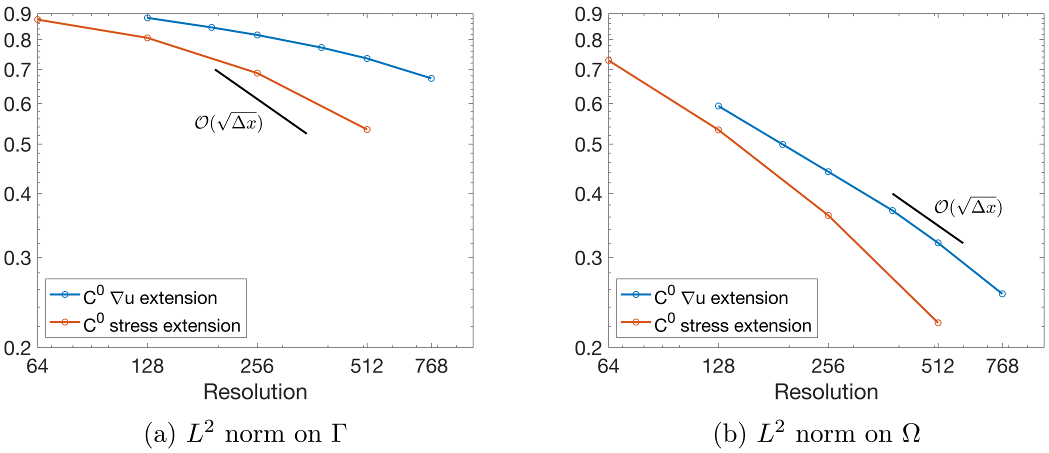


FIGURE 6.11. Vertical slices (at $x = 0$) of $\mathbf{C}_{11}$ using all stable extensions. These simulations are run with Wi $= 0.7$ and $d = 4$. The fluid domain is on the right side of the plot.

Moving on to our refinement studies with Wi $= 0.7$, we show the results from the C^0 extensions in Fig. 6.12. Predictably, the errors are abysmal. As with Wi $= 0.4$, we have sublinear convergence on boundary. Additionally, it seems that on the fluid domain, where we have had first-order convergence with these extensions at lower Wi, we now have sublinear convergence.



(a) L^2 norm on Γ

(b) L^2 norm on Ω

FIGURE 6.12. Comparison of DIB and IBSE using C^0 extensions. Plots show differences in tr$\mathbf{C}$, at Wi $= 0.7$ and $d = 4$ diffusion.

In Fig. 6.13, we show the results of our refinement study with Wi $= 0.7$ using the C^1 stress extension. In additional the usual measurement of differences between DIB and IBSE with L^2 norms on Γ and Ω, we have included measurements with the max norm. All are converging at approximately first-order and reach reasonably small values below about 3% on a 768 × 768 grid.

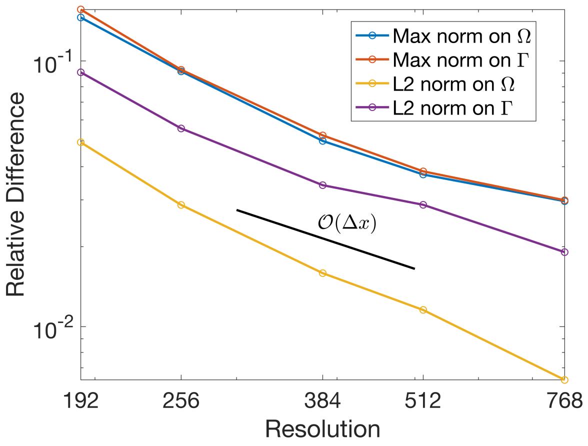


FIGURE 6.13. Comparison of DIB and IBSE using C^1 stress extension. Plot shows differences in tr$\mathbf{C}$, at $\mathrm{Wi} = 0.7$ and $d = 4$ diffusion.

6.9. Comparison of DIB and IBSE runtimes

In Table 6.1, we compare the runtimes of IBSE and DIB. We test DIB with no extensions, as well as each of the four extensions presented in this chapter. The test problem is viscoelastic flow around a cylinder, with $\mathrm{Wi} = 0.1$, no diffusion, and a simulation time of 10 time units. Only the time stepping portion of each simulation is timed, meaning that we are not accounting for the time taken to build the Schur complement. This is because the Schur complement can be built in a preprocessing step and saved for other simulations that use the same geometry. The runtimes reported are averages from 3 tests each.

	Nx = 64	**Nx = 128**	**Nx = 256**	**Nx = 512**
No Ext	29.8	130	837	7227
C^0 $\nabla\mathbf{u}$ Ext	31.0	148	986	8672
C^0 Stress Ext	29.7	143	949	7883
C^1 $\nabla\mathbf{u}$ Ext	34.6	187	1318	12376
C^1 Stress Ext	32.0	163	1108	10070
IBSE	45.4	395	2745	29291

TABLE 6.1. Average runtimes (in seconds) for flow around a cylinder, comparing DIB with each type of extension and IBSE. The simulations are 10 time units long, using $\mathrm{Wi} = 0.1$ and no diffusion ($d = 0$).

The runtime results in Table 6.1 cannot be considered a truly fair comparison of the two algorithms, in part because our implementations are in different languages (IBSE in Python and DIB in Matlab). However, it is clear that DIB runs significantly faster than IBSE.

As mentioned in the introduction, IBSE inverts a very large system, which makes it computationally expensive in comparison to DIB. If we examine the system that IBSE inverts, we will find that it is similar in many ways to DIB with extensions. Each method solves the Stokes-Oldroyd B equations on the fluid domain and solves another PDE on the nonphysical domain to create an extension. With IBSE, these two problems are coupled and therefore necessarily done in a single step, hence the larger system. In contrast, moving the DIB forces off of the boundary allows us to obtain solutions that, even without extensions, are smooth across the boundary. It turns out that extensions are necessary for the stability of DIB, but they can be computed in a separate step of the algorithm, thereby splitting the method into two steps and allowing us to invert two smaller systems.

6.10. Conclusion

Over the course of the last two chapters, we have introduced, developed, and evaluated the performance of the Double Immersed Boundary method applied to Stokes and Stokes-Oldroyd B equations. The DIB method began with the simple idea to modify the traditional Immersed Boundary method by moving forces away from the boundary, thereby obtaining a velocity gradient which is smooth across the boundary. While an elegant change, this placement of the forces creates conditioning issues. Handling these issues in Stokes flow was the subject of Chapter 5. When viscoelasticity is introduced in Chapter 6, stability issues arise which we tackle by computing extensions on the nonphysical domain. With these extensions, DIB is stable and reasonably accurate up to $\mathrm{Wi} = 0.7$ in our test problem of periodic flow around a cylinder. We obtained first order convergence of the stress, which is admittedly inferior to IBSE's second order convergence. However, it may be possible to reach second order convergence with DIB by using a smoother delta function for the spread and interpolation operators.

As discussed in the previous section, using extensions makes DIB conceptually similar to IBSE. However, with DIB the extension step is fully separated from solving the fluid equations. Besides computational efficiency, another notable advantage to this separation of the extension step is that we can change the method for creating extensions with great ease. In this way, DIB has opened an important new door for future work, as there are a myriad of extension methods that may offer new advantages, particularly if DIB is adapted for applications beyond viscoelastic flows.

Bibliography

[1] M. Alves, F. Pinho, and P. Oliveira, *The flow of viscoelastic fluids past a cylinder: finite-volume high-resolution methods*, Journal of Non-Newtonian Fluid Mechanics, 97 (2001), pp. 207–232.

[2] P. E. Arratia, *Life in complex fluids: Swimming in polymers*, Physical Review Fluids, (2022).

[3] N. Balci, B. Thomases, M. Renardy, and C. R. Doering, *Symmetric factorization of the conformation tensor in viscoelastic fluid models*, Journal of Non-Newtonian Fluid Mechanics, 166 (2011), pp. 546–553. XVIth International Workshop on Numerical Methods for Non-Newtonian Flows.

[4] E. Balkovsky, A. Fouxon, and V. Lebedev, *Turbulence of polymer solutions*, Phys. Rev. E, 64 (2001), p. 056301.

[5] M. Beneitez, J. Page, Y. Dubief, and R. R. Kerswell, *Multistability of elasto-inertial two-dimensional channel flow*, Journal of Fluid Mechanics, 981 (2024), p. A30.

[6] S. Berti, A. Bistagnino, G. Boffetta, A. Celani, and S. Musacchio, *Elastic Turbulence in 2D Viscoelastic Flows*, vol. 117, 01 2008, pp. 280–282.

[7] S. Berti, A. Bistagnino, G. Boffetta, A. Celani, and S. Musacchio, *Two-dimensional elastic turbulence*, Phys. Rev. E, 77 (2008), p. 055306.

[8] S. Berti and G. Boffetta, *Elastic waves and transition to elastic turbulence in a two-dimensional viscoelastic kolmogorov flow*, Phys. Rev. E, 82 (2010), p. 036314.

[9] P. Bhat, S. Appathurai, M. Harris, M. Pasquali, G. McKinley, and O. Basaran, *Formation of beads-on-a-string structures during break-up of viscoelastic filaments*, Nature Physics, 6 (2010), pp. 625–631.

[10] G. Buza, M. Beneitez, J. Page, and R. R. Kerswell, *Finite-amplitude elastic waves in viscoelastic channel flow from large to zero reynolds number*, Journal of Fluid Mechanics, 951 (2022), p. A3.

[11] M. Chilcott and J. Rallison, *Creeping flow of dilute polymer solutions past cylinders and spheres*, Journal of Non-Newtonian Fluid Mechanics, 29 (1988), pp. 381–432.

[12] S. Claus and T. Phillips, *Viscoelastic flow around a confined cylinder using spectral/hp element methods*, Journal of Non-Newtonian Fluid Mechanics, 200 (2013), pp. 131–146. Special Issue: Advances in Numerical Methods for Non-Newtonian Flows.

[13] S. S. Datta, A. M. Ardekani, P. E. Arratia, A. N. Beris, I. Bischofberger, G. H. McKinley, J. G. Eggers, J. E. López-Aguilar, S. M. Fielding, A. Frishman, M. D. Graham, J. S. Guasto, S. J. Haward, A. Q. Shen, S. Hormozi, A. Morozov, R. J. Poole, V. Shankar, E. S. G. Shaqfeh,

H. Stark, V. Steinberg, G. Subramanian, and H. A. Stone, *Perspectives on viscoelastic flow instabilities and elastic turbulence*, Phys. Rev. Fluids, 7 (2022), p. 080701.

[14] J. Dealy, *Weissenberg and deborah numbers – their definition and use*, Rheol. Bull., 79 (2010), pp. 14–18.

[15] M. Denn, *Extrusion instabilities and wall slip*, Annu. Rev. Fluid Mech, 33 (2001), pp. 265–87.

[16] Y. Dubief, J. Page, R. R. Kerswell, V. E. Terrapon, and V. Steinberg, *First coherent structure in elasto-inertial turbulence*, Phys. Rev. Fluids, 7 (2022), p. 073301.

[17] Y. Dubief, V. Terrapon, and B. Hof, *Elasto-inertial turbulence*, Annual Review of Fluid Mechanics, 55 (2023).

[18] A. Fouxon and V. Lebedev, *Spectra of turbulence in dilute polymer solutions*, Physics of Fluids, 15 (2003), pp. 2060–2072.

[19] M. S. Gockenbach, *Linear inverse problems and Tikhonov regularization*, vol. 32, American Mathematical Soc., 2016.

[20] L. Greengard and M. Kropinski, *Integral equation methods for stokes flow in doubly-periodic domains*, Journal of Engineering Mathematics, 48 (2004), pp. 157–170.

[21] A. Groisman and V. Steinberg, *Elastic turbulence in a polymer solution flow*, Nature, 405 (2000), pp. 53–55.

[22] A. Groisman and V. Steinberg, *Efficient mixing at low reynolds numbers using polymer additives*, Nature, 410 (2001), pp. 905–908.

[23] R. D. Guy and B. Thomases, *Computational Challenges for Simulating Strongly Elastic Flows in Biology*, Springer New York, New York, NY, 2015, pp. 359–397.

[24] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, 2 ed., 2012.

[25] T. Y. Hou and R. Li, *Computing nearly singular solutions using pseudo-spectral methods*, Journal of Computational Physics, 226 (2007), pp. 379–397.

[26] D. D. Joseph and R. Fosdick, *The free surface on a liquid between cylinders rotating at different speeds part i*, Archive for Rational Mechanics and Analysis, 49 (1973), pp. 321–380.

[27] M. R. Jovanović and S. Kumar, *Transient growth without inertia*, Physics of Fluids, 22 (2010), p. 023101.

[28] R. Kerswell and J. Page, *Asymptotics of the centre mode instability in viscoelastic channel flow: with and without inertia*, 2023.

[29] P. M. Knupp and K. Salari, *Verification of Computer Codes in Computational Science and Engineering*, Chapman & Hall/CRC, 2002.

[30] R. Kress, *Ill-Conditioned Linear Systems*, Springer New York, New York, NY, 1998, pp. 77–92.

[31] M. Kumar and A. M. Ardekani, *Elastic instabilities between two cylinders confined in a channel*, Physics of Fluids, 33 (2021), p. 074107.

[32] M.-C. Lai and C. S. Peskin, *An immersed boundary method with formal second-order accuracy and reduced numerical viscosity*, Journal of Computational Physics, 160 (2000), pp. 705–719.

[33] M. Lellep, M. Linkman, and A. Morozov, *Linear stability analysis of purely elastic travelling-wave solutions in pressure-driven channel flows*, Journal of Fluid Mechanics, 959 (2023), p. R1.

[34] M. Lellep, M. Linkmann, and A. Morozov, *Purely elastic turbulence in pressure-driven channel flows*, Proceedings of the National Academy of Sciences, 121 (2024), p. e2318851121.

[35] G. H. McKinley, P. Pakdel, and A. Öztekin, *Rheological and geometric scaling of purely elastic flow instabilities*, Journal of Non-Newtonian Fluid Mechanics, 67 (1996), pp. 19–47.

[36] A. Morozov, *Coherent structures in plane channel flow of dilute polymer solutions with vanishing inertia*, Phys. Rev. Lett., 129 (2022), p. 017801.

[37] A. Morozov and S. E. Spagnolie, *Introduction to Complex Fluids*, Springer New York, New York, NY, 2015, pp. 3–52.

[38] A. N. Morozov and W. van Saarloos, *Subcritical finite-amplitude solutions for plane couette flow of viscoelastic fluids*, Phys. Rev. Lett., 95 (2005), p. 024501.

[39] A. N. Morozov and W. van Saarloos, *An introductory essay on subcritical instabilities and the transition to turbulence in visco-elastic parallel shear flows*, Physics Reports, 447 (2007), pp. 112–143. Nonequilibrium physics: From complex fluids to biological systems I. Instabilities and pattern formation.

[40] A. Neumaier, *Solving ill-conditioned and singular linear systems: A tutorial on regularization*, SIAM Review, 40 (1998), pp. 636–666.

[41] J. Nichols, R. D. Guy, and B. Thomases, *A period-doubling route to chaos in viscoelastic kolmogorov flow*, 2024.

[42] J. Page, Y. Dubief, and R. R. Kerswell, *Exact traveling wave solutions in viscoelastic channel flow*, Phys. Rev. Lett., 125 (2020), p. 154501.

[43] L. Pan, A. Morozov, C. Wagner, and P. E. Arratia, *Nonlinear elastic instability in channel flows at low reynolds numbers*, Phys. Rev. Lett., 110 (2013), p. 174502.

[44] C. S. Peskin, *Flow patterns around heart valves: A numerical method*, Journal of Computational Physics, 10 (1972), pp. 252–271.

[45] C. S. Peskin, *The immersed boundary method*, Acta Numerica, 11 (2002), p. 479–517.

[46] R. Poole, *The deborah and weissenberg numbers*, The British Society of Rheology - Rheology Bulletin, 53 (2012), pp. 32–39.

[47] B. Qin, P. F. Salipante, S. D. Hudson, and P. E. Arratia, *Upstream vortex and elastic wave in the viscoelastic flow around a confined cylinder*, Journal of Fluid Mechanics, 864 (2019), p. R2.

[48] M. Renardy and B. Thomases, *A mathematician's perspective on the oldroyd b model: Progress and future challenges*, Journal of Non-Newtonian Fluid Mechanics, 293 (2021), p. 104573.

[49] P. J. Roache, *Code Verification by the Method of Manufactured Solutions* , Journal of Fluids Engineering, 124 (2001), pp. 4–10.

[50] L. Sangroniz, M. Fernández, and A. Santamaria, *Polymers and rheology: A tale of give and take*, Polymer, 271 (2023), p. 125811.

[51] E. G. Shaqfeh, *Purely elastic instabilities in viscometric flows*, Annual Review of Fluid Mechanics, (1996), pp. 129–185.

[52] F. D. C. Siacor, Q. Chen, J. Y. Zhao, L. Han, A. D. Valino, E. B. Taboada, E. B. Caldona, and R. C. Advincula, *On the additive manufacturing (3d printing) of viscoelastic materials and flow behavior: From composites to food manufacturing*, Additive Manufacturing, 45 (2021), p. 102043.

[53] P. C. Sousa, F. T. Pinho, and M. A. Alves, *Purely-elastic flow instabilities and elastic turbulence in microfluidic cross-slot devices*, Soft Matter, 14 (2018), pp. 1344–1354.

[54] S. E. Spagnolie and P. T. Underhill, *Swimming in complex fluids*, Annual Review of Condensed Matter Physics, 14 (2023), pp. 381–415.

[55] D. B. Stein, *The Immersed Boundary Smooth Extension (IBSE) Method: A Flexible and Accurate Fictitious Domain Method, and Applications to the Study of Polymeric Flow in Complex Geometries*, PhD thesis, University of California Davis, 2016. Copyright - Database copyright ProQuest LLC; ProQuest does not claim copyright in the individual underlying works; Last updated - 2023-03-04.

[56] D. B. Stein, R. D. Guy, and B. Thomases, *Immersed boundary smooth extension: A high-order method for solving pde on arbitrary smooth domains using fourier spectral methods*, Journal of Computational Physics, 304 (2016), pp. 252–274.

[57] D. B. Stein, R. D. Guy, and B. Thomases, *Convergent solutions of stokes oldroyd-b boundary value problems using the immersed boundary smooth extension (ibse) method*, Journal of Non-Newtonian Fluid Mechanics, 268 (2019), pp. 56–65.

[58] S. H. Strogatz, *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering*, CRC press, 2018.

[59] R. Sureshkumar and A. N. Beris, *Effect of artificial stress diffusivity on the stability of numerical calculations and the flow dynamics of time-dependent viscoelastic flows*, Journal of Non-Newtonian Fluid Mechanics, 60 (1995), pp. 53–80.

[60] S. Tabakova and R. Kotsilkova, *Modelling of polymer flow instabilities with application to 3d printing*, Journal of Theoretical and Applied Mechanics, 53 (2023), pp. 389–403.

[61] K. Taira and T. Colonius, *The immersed boundary method: A projection approach*, Journal of Computational Physics, 225 (2007), pp. 2118–2137.

[62] J. Teran and C. Peskin, *Tether force constraints in stokes flow by the immersed boundary method on a periodic domain*, SIAM J. Scientific Computing, 31 (2009), pp. 3404–3416.

[63] B. Thomases and M. Shelley, *Transition to mixing and oscillations in a stokesian viscoelastic flow*, Phys. Rev. Lett., 103 (2009), p. 094501.

[64] B. Thomases, M. Shelley, and J.-L. Thiffeault, *A stokesian viscoelastic flow: Transition to oscillations and mixing*, Physica D: Nonlinear Phenomena, 240 (2011), pp. 1602–1614. Special Issue: Fluid Dynamics: From Theory to Experiment.

[65] A. N. Tikhonov and V. Y. Arsenin, *Solutions of ill-posed problems*, V. H. Winston & Sons, Washington, D.C.: John Wiley & Sons, New York, 1977. Translated from Russian, Preface by translation editor Fritz John, Scripta Series in Mathematics.

[66] A. Varshney and V. Steinberg, *Elastic alfven waves in elastic turbulence*, Nature Communications, 10 (2019), p. 652.

[67] W.-H. Zhang, H.-N. Zhang, Z.-M. Wang, Y.-K. Li, B. Yu, and F.-C. Li, *Repicturing viscoelastic drag-reducing turbulence by introducing dynamics of elasto-inertial turbulence*, Journal of Fluid Mechanics, 940 (2022), p. A31.

[68] J. Zhou and I. Papautsky, *Viscoelastic microfluidics: progress and challanges*, Microsystems & Nanoengineering, 6 (2020), p. 113.

[69] A. Zizzari, M. Cesaria, M. Bianco, L. del Mercato, M. Carraro, M. Bonchio, R. Rella, and V. Arima, *Mixing enhancement induced by viscoelastic micromotors in microfluidic platforms*, Chemical Engineering Journal, 391 (2020), p. 123572.