ANALYSIS AND COMPUTATION OF IMMERSED BOUNDARIES, WITH APPLICATION TO PULP FIBRES

by

JOHN MICHAEL STOCKIE

B. Math. (Applied Mathematics and Computer Science)
University of Waterloo, 1991

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

THE FACULTY OF GRADUATE STUDIES

Department of Mathematics
Institute of Applied Mathematics

We accept this thesis as conforming
to the required standard

. . . .

. . . .

. . . .

. . . .

. . . .

THE UNIVERSITY OF BRITISH COLUMBIA

September 1997

(Signature) ________________________________

Department of Mathematics
The University of British Columbia
Vancouver, Canada

Date ________________________________

Abstract

Immersed fibres are a very useful tool for modeling moving, elastic interfaces that interact with a surrounding fluid. The Immersed Boundary Method is a computational tool based on the immersed fibre model which has been used successfully to study a wide range of applications including blood flow in the heart and arteries and motion of suspended particles.

This work centres around a linear analysis of an isolated fibre in two dimensions, which pinpoints a discrete set of solution modes associated solely with the fibre. We investigate the stability and stiffness characteristics of the fibre modes and then relate the results to the severe time step restrictions experienced in explicit and semi–implicit immersed boundary computations. A subset of the modes corresponding to tangential oscillations of the fibre are the main source of stiffness, which intensifies when the fibre force is increased or fluid viscosity is decreased — this explains why computations are limited to unrealistically small Reynolds numbers.

We also investigate the effects of smoothing the fibre over a given thickness, which corresponds to the delta function approximation that is central to the discrete scheme. The results can be applied to explore the accuracy of various interpolation methods in an idealised setting.

The analysis is next extended to predict time step restrictions and convergence rates for various explicit and semi–implicit discretisations. The results are verified in numerical experiments.

Finally, we introduce a novel application of the Immersed Boundary Method to the motion of pulp fibres in a two–dimensional shear flow. The method is shown to reproduce both theoretical results and experimentally observed behaviour over a wide range of parameter values.

Table of Contents

List of Tables

List of Figures

Acknowledgement

> *I had a feeling about Mathematics — that I saw it all. Depth beyond Depth was revealed to me — the Byss and the Abyss. I saw — as one might see the transit of Venus or even the Lord Mayor's show — a quantity passing through infinity and changing its sign from plus to minus. I saw exactly how it happened and why the tergivisation was inevitable — but it was after dinner and I let it go.*
>
> — Winston Churchill.

I would like to thank Brian Wetton for his support and encouragement throughout my graduate studies at UBC. He was always generous with his time, and it was during our many long and insightful discussions that he helped to cultivate my interest in CFD and imparted to me some of his own enthusiasm for the subject. In my opinion, he has gone above and beyond the limits of what define a *supervisor*.

I am also indebted to Uri Ascher for his helpful advice. I am particularly thankful for his determination and foresight in setting up two first–rate computing laboratories, which facilitated my research and provided me with a healthy and productive computing environment.

Sheldon Green must be credited for making the suggestion that the Immersed Boundary Method could be applied to simulating pulp fibres. Thanks go also to Alexandre Kurth for giving the proposition in Chapter 3 mathematical legs to stand on.

I acknowledge the financial support of the Natural Sciences and Engineering Research Council of Canada during my graduate work.

Finally, I want to express my deepest gratitude to a person who has played an integral role in my studies these past four years: she has been a friend and confidant, advisor and proof–reader, classmate and soul mate; she was an endless source of laughter and encouragement throughout this time; but most importantly, she reminded me of the importance of my dreams, and by her own example and strength of character helped me in achieving those dreams. Vielen Dank, Petra Menz!

Chapter 1
Introduction

In modern times the belief that the ultimate explanation of all things was to be found in Newtonian mechanics was an adumbration of the truth that all science, as it grows towards perfection, becomes mathematical in its ideas.

— Alfred North Whitehead.

The physical world is replete with examples of free surfaces, material interfaces and moving boundaries that interact with a surrounding fluid. There are interfaces that separate air and water (in the case of bubbles or free surface flows) and boundaries between two materials of differing physical properties (in porous media flow or mixing layers). Alternatively, the interface may be a rigid wall that moves with some specified time–dependent motion, or an elastic membrane that deforms and stretches in response to the fluid motion.

One of the most challenging problems facing both modelers and computational scientists alike is how to handle the two–way hydrodynamic coupling between a fluid and an elastic interface or membrane that transmits forces to the fluid. Such problems are particularly common in living systems, with examples including the interaction of muscle tissue with blood in the heart and arteries; swimming motion of marine worms, fish and microorganisms; and locomotion of amoebae through inter–cellular fluid, to name a few. Peskin & McQueen remark [PM95]:

> *It is appropriate to ask whether there is any common theme that unites the diverse problems that arise in the study of living systems interacting with fluids. The answer that immediately comes to mind is this: biological fluid dynamics invariably involves*

the interaction of elastic flexible tissue with viscous incompressible fluid.

While the mathematical modeling of the interaction is a difficult problem in itself, another formidable task is developing a numerical method that solves these problems effectively and efficiently. In an excellent review article [Hou95], Hou identifies the salient issues that face investigators trying to compute solutions to free and moving boundary problems:

- sensitivity to small perturbations, which leads to instabilities when naïve discretisation schemes are applied;
- singularity formation and topological changes; and
- severe time–stepping constraints due to stiffness introduced by high order regularisation effects.

Furthermore, he points out that a common feature in many fluid interface problems is an underlying physical instability that generates rapid growth in high frequency solution components in the absence of regularising forces such as surface tension or viscosity.

Fortunately, topological changes can be ruled out for the elastic boundaries we are considering here: the fluid is incompressible and the boundaries are solid, and so there is no separation of boundary components or leakage that could cause a singularity to form. However, the magnitude and highly localised nature of forces generated by the elastic fibres also introduce a high level of stiffness in the problem. Consequently, the main difficulty in developing a numerical method for this class of problem is coupling the fibre motion with that of the fluid in a way that avoids spurious numerical instabilities and allows reasonable time steps to be taken.

A diverse range of numerical methods have been developed for dealing with these and other difficulties. Among the schemes that have been applied to moving interface problems, the most common approach used has been a moving or adaptive Eulerian mesh technique in which the fluid grid is evolved during each time step to conform with the moving boundary. One common technique is *local mesh refinement*, where a finer rectangular mesh is used near the interface, while a more efficient coarse grid is used in regions of the flow where the high level of refinement is not necessary for an equivalent level of accuracy. An example of this technique is the Adaptive Mesh Refinement algorithm, originally developed by Berger [Ber82] for hyperbolic problems and recently extended to handle incompressible flows [HB97, Rom96].

In contrast with this local strategy, there are also "global" techniques which adapt the entire grid to conform with the interface. Ohring and Lugt [OL89], for example, use a finite difference discretisation based on an orthogonal coordinate transformation to solve free surface flows — this technique has the advantage that a complex irregular domain is solved on an equally–spaced rectangular grid in "computational space." Skalak & Tözeren [ST82], on the other hand, apply a moving finite element technique to solve biofluid mechanics problems, wherein the finite element mesh is advanced within each time step to conform with a moving, deformable boundary. The main drawback of these moving mesh approaches is the extra expense and complexity of regridding all or part of the fluid domain in every time step.

A method which avoids the issue of regridding altogether is the *Level Set Method*, developed by Osher & Sethian [OS88]. This scheme propagates the interface along with the solution by introducing an additional dependent function that is convected with the fluid: the interface is simply the zero level set of this function. This method handles the difficult problems of singularity formation and changes in topology in a natural way. However, the level set formulation lacks any knowledge of the location of individual material points on the interface and hence is unable to capture the stretched state of the interface, which is needed to compute an elastic force.

The *Immersed Boundary Method* is a mixed Eulerian–Lagrangian scheme that combines the simplicity and efficiency inherent in using a fixed Cartesian grid to compute the fluid motion, along with the geometric flexibility of tracking the interface at a set of moving Lagrangian points. The key idea in this method is to replace the fluid–material interface with appropriate contributions to a force density term in the fluid equations. The internal boundaries are thereby eliminated and a simple, finite difference scheme can be used to solve the fluid equations, with the influence of the interface relegated to an inhomogeneous forcing term that is distributed onto fluid points near the interface.

The Immersed Boundary Method was originally developed by Peskin [Pes72] to simulate the motion of a heart valve in a two–dimensional fluid. The method has since been extended to three–dimensional heart valve simulations [PM89, PM92] and has been applied to a wide variety of other biological problems including swimming motions of marine worms [FP88a], sperm [FF93, FM95] and biflagellated algal cells [Fau93]; aggregation of blood platelets [Fog84, FF93]; wave propagation in the fluid–filled cavity of the inner ear [Bey92]; blood flow in arteries [VY92, Ros94, Art96]; bacterial motion and chemotaxis in biofilms [DFG95, DFFG96]; and amoeboid locomotion [Bot96]. It has also been applied to non–biological situations, such as the flow of particulate suspensions [SB91, FP88b]; interaction of particles in turbulent flows [Yus96]; and plasma simulations [LIB95] where a solid body is treated as an immersed boundary.

The method's main advantages are its simplicity and geometric flexibility, which account for its widespread use. The Immersed Boundary Method is very efficient due to its use of the Fast Fourier Transform in combination with an "alternating direction implicit" (or ADI) approach to solve the fluid equations, and it is easily vectorisable [PM89, FP88b], which makes 3D calculations feasible. The interface is modeled very simply using a data structure composed of "spring–like" links between adjacent points, which facilitates the handling of immersed boundaries of nearly arbitrary shape, size and configuration.

On the other hand, the method suffers from some major deficiencies. Computational evidence [LL97, Rom96] suggests that the method is only first order accurate in space due to the choice of interpolation scheme used to transfer quantities between fluid and fibre grid points [BL92]. Roma [Rom96] recently developed a two–dimensional implementation of the Immersed Boundary Method that uses a multi–level adaptive mesh algorithm to refine the fluid grid near the immersed interface. He achieved the same level of accuracy on a non–uniform mesh that was previously obtainable only with a uniformly fine mesh. LeVeque & Li [LL94, LL97] developed an alternative (but related) scheme, called the *Immersed Interface Method*, which attains second–order accuracy in space by replacing the interpolation between fluid and fibre points with modified difference stencils across the boundary. It has been tested on a variety of stationary flows in two dimensions and recently extended by Li to 3D elliptic problems [Li97b] and time–dependent (Stefan–like) problems in one dimension [Li97a]. However, the Immersed Interface Method has yet to be extended to time–dependent problems in two or three dimensions.

The Immersed Boundary Method does not avoid the numerical stiffness issue, and requires that extremely small time steps be taken for explicit (and also many semi–implicit) calculations. Even in two dimensions, the method is restricted to relatively viscous flows (with $\mathcal{R}e$ on the order of several hundred) and three–dimensional simulations are only practical on supercomputers [PM93a, PM93b]. We will see later on that there is an analytical justification for the stiffness observed in computations, based on how the solution behaves when viscosity is small and fibre stiffness is large. These limitations are not a consequence of any particular choice of spatial discretisation in the Immersed Boundary Method, and would manifest themselves regardless of whether a finite element or spectral approach was used. Considerable effort has gone into developing versions of the method that better couple the fibre force to the motion of the fluid [TP92, MP93, FF93]. Nevertheless, an effective semi–implicit method with a corresponding efficient solver has yet to be developed, and many computations are still being done using a simple time discretisation that is explicit in the fibre force [DFFG96, Ros94, PM92].

Despite its shortcomings, the Immersed Boundary Method is still an extremely useful modeling tool. Its utility has been demonstrated by its unparalleled ability to simulate a wide range of complex physical phenomena. The power of the method is perhaps best demonstrated by its central role in the design of an artificial heart valve, which has led to the granting of several patents [MP90, MP91].

Because of the widespread use of the Immersed Boundary Method, it therefore comes as a surprise that there has been little analysis performed to date on either the model equations or the numerical method. The limitations on the accuracy and efficiency of the scheme are well–documented, and yet no satisfactory explanation has appeared in the literature. Our main goal in this work is to gain a better understanding of the behaviour of solutions to the underlying equations of motion for immersed boundaries. We will identify the stiffness inherent in the problem and use our insight into the mathematical structure of the solution to suggest improvements to the method that will counteract the stiffness and increase efficiency.

We begin in Chapter 2 with a summary of the mathematical background on immersed fibres that has appeared in the literature. We provide an overview of the integro–differential equations governing their motion. Two mathematically equivalent formulations of the problem are given: one on which the Immersed Boundary Method is based; and the other in terms of jumps in fluid quantities across the fibre, which yields more easily to analysis. We present the Immersed Boundary Method algorithm and outline the vital role of the interpolating functions that transfer fluid quantities to fibre grid points and vice versa.

Chapter 3 contains a linear analysis of a two–dimensional model problem which has appeared in [SW95]. We identify the effect of a single fibre on the motion of a fluid by singling out a set of discrete modes that are excited by the presence of the fibre. The results are used to make conclusions about the stiffness of the problem. We also investigate the effect of smoothing the fibre force, which is inherent in any discretisation of the problem, and relate this to the stiffness and spatial accuracy of the Immersed Boundary Method. The results and conclusions are verified in numerical experiments.

The analytical results for the continuous problem are extended to time discretisations in Chapter 4. We investigate the stability of explicit time–stepping, as well as the convergence of iterative methods based on two semi–implicit time discretisations. Computations are employed to illustrate our conclusions and to demonstrate the power of the analytical technique in developing and investigating modifications and improvements to the Immersed Boundary Method.

In Chapter 5, we digress from the more theoretical treatment of immersed fibres in the previous two chapters and introduce a new application of the Immersed Boundary Method to the motion of paper pulp fibres. The relevant physics governing pulp fibres is reviewed, and we identify the importance of understanding the motion of individual pulp fibres in the paper–making process. We then present a series of numerical simulations that demonstrate the ability of the Immersed Boundary Method to reproduce fibre behaviour observed in experiments.

We conclude in Chapter 6 with a summary of the major results in this work, and the potential areas for future research.

Chapter 2
Immersed Fibres

It's good to express a matter in two ways simultaneously so as to give it both a right foot and a left. Truth can stand on one leg, but with two it can walk and get about.

— Friedrich Nietzsche.

2.1 What is an "Immersed Fibre"?

The original motivation for the development of the Immersed Boundary Method was to simulate the motion of heart muscle immersed in blood. With this in mind, we will summarise the basic assumptions of the model in terms of the application to cardiac fluid dynamics. However, it is important to remember that many of the assumptions we make can also be justified for other biological and non–biological flows where an elastic or contractile fibre or surface interacts with a surrounding fluid.

In Peskin and McQueen's three–dimensional heart model [PM89], the muscle walls and valves are composed of an interwoven mesh of fibres, as is the arterial wall model of Rosar [Ros94]. Figure 2.1 shows a schematic representation of an immersed surface in three dimensions and its representation as a fibre mesh. The fibres are suspended within a Newtonian, incompressible fluid (the blood) and assumed to be neutrally buoyant, massless, and to occupy zero volume. Consequently, the fibres are also incompressible, and the fluid–fibre system can be regarded as a

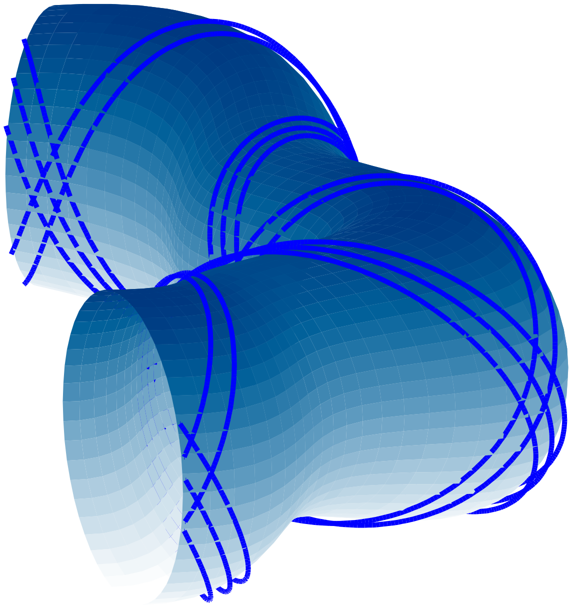


Figure 2.1: A three–dimensional immersed surface modeled as an interwoven mesh of immersed fibres, with only five fibres shown.

composite viscoelastic material.[a] The main advantage to this model, as we will soon see, is that the fluid and fibre can be described by a single system of equations. Immersed fibres by themselves are not accurate physical representations of heart muscle fibres (or arterial wall fibres), since they have no mass or volume. It is only in combination with its massive fluid component that the force–bearing fibres can be thought of *as actual physical fibres.*

It still remains to describe how the immersed boundary interacts with the surrounding fluid. The fibres exert on neighbouring fluid particles a force whose direction and magnitude depend on the configuration of the boundary. One can think of successive points on a fibre as being connected by "springs" which transmit a force to the fluid that depends on the stretched state of the springs.

To summarise, an immersed fibre is a massless, neutrally–buoyant fibre that exerts an elastic

[a] The treatment of the fibres as an incompressible material has a further physical justification in the work of Ebin and Saxton [ES86, ES87], who showed that the equations describing an incompressible elastic body resemble those of an incompressible fluid.

force on the surrounding fluid. The force can be either a simple elastic function depending only on the configuration of the fibres; or more generally an active, time–dependent “contractile” force (which is certainly the case for heart muscle fibres). Immersed elastic structures of almost any shape and function can be constructed by interweaving and joining together fibres with varying elastic force properties, thereby allowing such diverse biological structures as heart muscle, flagellated cells, and amoebae to be encompassed by the same model.

2.2 Mathematical Formulation for Immersed Fibres

The main focus of this chapter is a study of the behaviour of solutions to the equations underlying the motion of immersed boundaries. Since any immersed elastic boundary is modeled as a collection of one–dimensional fibres, we will restrict ourselves to the study of a single fibre immersed in a two–dimensional fluid.

For our two–dimensional model problem, let us consider a square domain $\Omega = [0,1] \times [0,1]$, periodic in both the $x-$ and y–directions, that is filled with an incompressible, viscous fluid. Suspended within the fluid is a fibre, which can be described by a continuous curve Γ, as pictured in Figure 2.2. The motion of the fluid–fibre composite is governed by the incompressible Navier–Stokes equations

$$\rho \frac{\partial \boldsymbol{u}}{\partial t} = -\rho \boldsymbol{u} \cdot \nabla \boldsymbol{u} + \mu \Delta \boldsymbol{u} - \nabla p + \boldsymbol{F}, \tag{2.1}$$

$$\nabla \cdot \boldsymbol{u} = 0, \tag{2.2}$$

where $\boldsymbol{u}(\boldsymbol{x},t) = (u(\boldsymbol{x},t), v(\boldsymbol{x},t))$ is the fluid velocity, $p(\boldsymbol{x},t)$ the pressure, $\boldsymbol{F}(\boldsymbol{x},t)$ is the external force, and ρ and μ are the (constant) fluid density and viscosity. Let $\boldsymbol{x} = \boldsymbol{\mathcal{X}}(s,t)$ represent the position of the fibre, where s is a parameterisation of the fibre in some reference configuration.[b] The fibre moves at the same velocity as neighbouring fluid particles, and so we can write

$$\frac{\partial \boldsymbol{\mathcal{X}}}{\partial t} = \boldsymbol{u}(\boldsymbol{\mathcal{X}}(s,t),t). \tag{2.3}$$

[b] Typically, s is taken to be the arclength of the fibre *in an unstressed state*, though as the fibre evolves in time s will not necessarily be a measure of arclength.

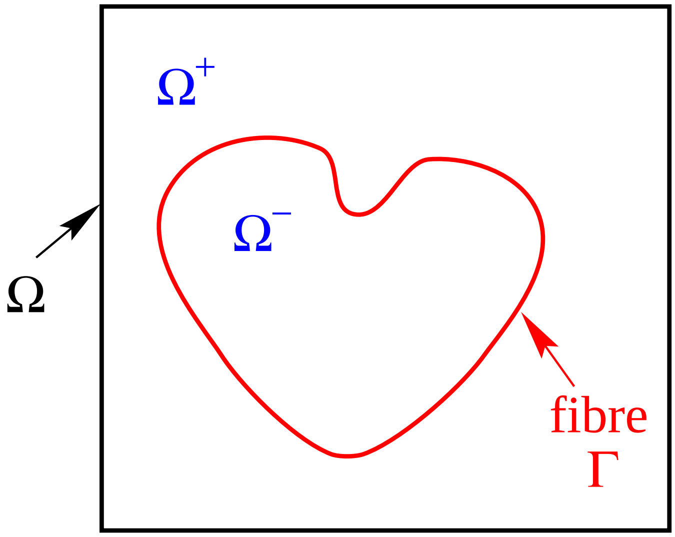


Figure 2.2: The two–dimensional model: a fluid domain, Ω, which is divided into two parts, Ω^+ and Ω^-, by an isolated fibre Γ immersed in the fluid.

One more element is needed to close the system: namely, an expression for the force $\boldsymbol{F}$ to couple the motion of the fluid and fibre in equations (2.1)–(2.3). Gravitational effects are assumed to be negligible since the fibre is neutrally buoyant, and so the external force $\boldsymbol{F}$ arises solely from the action of the elastic fibre. Let $T(s,t)$ be the tension force in the fibre and assume that T is a function of the fibre strain[c]:

$$T = T\left(\left|\frac{\partial \boldsymbol{\mathcal{X}}}{\partial s}\right|\right). \tag{2.4}$$

It can be shown under further assumptions [PM89] that the local force density per unit length ds is given by the expression

$$\boldsymbol{f}(s,t) = \frac{\partial}{\partial s}(T\boldsymbol{\tau}) \tag{2.5}$$

[c] Note that $|d\boldsymbol{\mathcal{X}}| = |\partial\boldsymbol{\mathcal{X}}/\partial s| \cdot |ds|$, where $|d\boldsymbol{\mathcal{X}}|$ is the distance between two points on the fibre and $|ds|$ is the distance between the same two points in the reference configuration. If the reference configuration is unstressed, then $(|\partial\boldsymbol{\mathcal{X}}/\partial s| - 1)$ is a measure of the fibre strain.

where

$$\boldsymbol{\tau} = \frac{\frac{\partial \boldsymbol{\mathcal{X}}}{\partial s}}{\left|\frac{\partial \boldsymbol{\mathcal{X}}}{\partial s}\right|} \tag{2.6}$$

is the unit tangent vector to the fibre. For example, if we choose the tension to be $T = \sigma\,|\partial\boldsymbol{\mathcal{X}}/\partial s|$, then the force density is simply

$$\boldsymbol{f} = \sigma\,\frac{\partial^2 \boldsymbol{\mathcal{X}}}{\partial s^2}. \tag{2.7}$$

Taking this form of the force is analogous to linking successive fibre points by linear springs with spring constant σ and zero resting length — we will see a similar forcing function appearing in the linear stability analysis in Chapter 3. Since the force is zero everywhere except on the fibre, the fluid body force $\boldsymbol{F}$ can be regarded as a distribution and written compactly as the convolution of the fibre force density with a delta function:

$$\boldsymbol{F}(\boldsymbol{x},t) = \int_\Gamma \boldsymbol{f}(s,t)\,\delta(\boldsymbol{x} - \boldsymbol{\mathcal{X}}(s,t))\,ds. \tag{2.8}$$

The two–dimensional delta function $\delta(\boldsymbol{x}) = \delta(x)\cdot\delta(y)$ is the product of two Dirac delta functions.

We can now write a coupled system of integro–differential equations for the motion of the fluid and fibre:

Delta Function Formulation

$$\rho\frac{\partial \boldsymbol{u}}{\partial t} = -\rho\boldsymbol{u}\cdot\nabla\boldsymbol{u} + \mu\Delta\boldsymbol{u} - \nabla p + \int_\Gamma \boldsymbol{f}(s,t)\,\delta(\boldsymbol{x} - \boldsymbol{\mathcal{X}}(s,t))\,ds \tag{2.9a}$$

$$\nabla\cdot\boldsymbol{u} = 0 \tag{2.9b}$$

$$\frac{\partial \boldsymbol{\mathcal{X}}}{\partial t} = \int_\Omega \boldsymbol{u}(\boldsymbol{x},t)\,\delta(\boldsymbol{x} - \boldsymbol{\mathcal{X}}(s,t))\,d\boldsymbol{x} \tag{2.9c}$$

Because of the central role of the delta function in coupling together equations (2.9a)–(2.9c), we will refer to these equations as *the delta function formulation* of the immersed fibre problem. The right hand side of the fibre evolution equation (2.3) has been rewritten in the equivalent form of a convolution of the velocity with a delta function, thereby introducing a symmetry between (2.9a) and (2.9c) that will prove to be very useful in Section 2.3 from the standpoint of constructing a numerical scheme. It is also worth mentioning that even though the integrals in (2.9a) and (2.9c) look quite similar, they are fundamentally different — the first is a line integral that evaluates to a singular function, while the latter is integrated over a two–dimensional region and so is bounded.

2.2.1 An alternate formulation

The presence of the delta-function singularity in the formulation above leads us to recast the equations in an alternate form which is more amenable to analysis. Following the derivation in [PP93], we integrate equation (2.9a) across the fibre and assume the velocity is continuous, to obtain a series of "jump conditions" relating the solution on either side of the fibre:

$$[\![\boldsymbol{u}]\!] = 0, \tag{2.10}$$

$$\mu\boldsymbol{\tau}\cdot[\![\boldsymbol{n}\cdot\nabla\boldsymbol{u}]\!] = -\frac{\boldsymbol{f}\cdot\boldsymbol{\tau}}{\left|\frac{\partial\boldsymbol{\mathcal{X}}}{\partial s}\right|}, \tag{2.11}$$

$$-[\![p]\!] + \mu\boldsymbol{n}\cdot[\![\boldsymbol{n}\cdot\nabla\boldsymbol{u}]\!] = -\frac{\boldsymbol{f}\cdot\boldsymbol{n}}{\left|\frac{\partial\boldsymbol{\mathcal{X}}}{\partial s}\right|}, \tag{2.12}$$

Here, $[\![\cdot]\!] = (\cdot)|_{\Gamma^+} - (\cdot)|_{\Gamma^-}$ denotes the difference in a quantity on either side of the fibre, and $\boldsymbol{n}$ is the unit normal vector to the fibre defined by $\boldsymbol{n}\cdot\boldsymbol{\tau} = 0$ (see Figure 2.3). The last jump condition reduces to

$$-[\![p]\!] = -\frac{\boldsymbol{f}\cdot\boldsymbol{n}}{\left|\frac{\partial\boldsymbol{\mathcal{X}}}{\partial s}\right|} \tag{2.13}$$

after using the fact that the velocity is incompressible and continuous across Γ. From (2.11) and (2.13), it is clear that both the pressure and the normal derivative of the velocity may be discontinuous across the fibre, even though the velocity itself is continuous. In fact, it is only possible for the tangential component of the normal derivative of velocity to be discontinuous, since incompressibility requires that the normal component be continuous.

Since the fibre Γ divides Ω into two subdomains, Ω^+ and Ω^-, on both of which the velocity is *smooth*, we may reformulate (2.9a)–(2.9c) as two separate Navier–Stokes problems *with zero external force*. Rather than handling the interaction between fluid and fibre using delta functions,

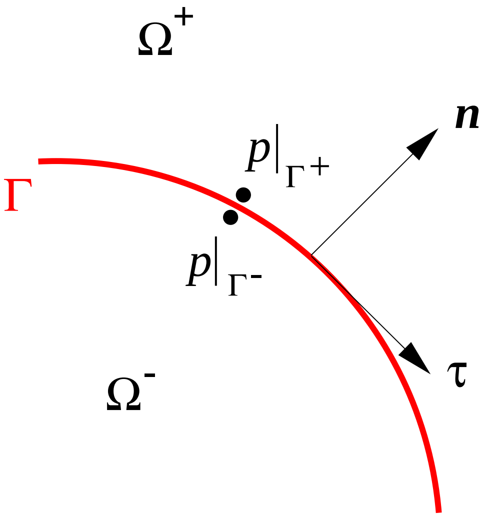


Figure 2.3: An illustration of the pressure jump across the interface, $[\![p]\!] = p|_{\Gamma^+} - p|_{\Gamma^-}$, including the unit normal and tangent vectors.

we instead use the jump conditions and the original fibre evolution equation (2.3). The resulting system of equations will be referred to as the *jump formulation* of the immersed fibre problem:

Jump Formulation

$$\rho \frac{\partial \boldsymbol{u}}{\partial t} = -\rho \boldsymbol{u} \cdot \nabla \boldsymbol{u} + \mu \Delta \boldsymbol{u} - \nabla p \qquad \text{in } \Omega^+ \cup \Omega^- \qquad (2.14a)$$

$$\nabla \cdot \boldsymbol{u} = 0 \qquad \text{in } \Omega^+ \cup \Omega^- \qquad (2.14b)$$

$$\frac{\partial \boldsymbol{\mathcal{X}}}{\partial t} = \boldsymbol{u}(\boldsymbol{\mathcal{X}}(s,t),t) \qquad \text{on } \Gamma \qquad (2.14c)$$

$$[\![\boldsymbol{u}]\!] = 0 \qquad (2.14d)$$

$$\mu \boldsymbol{\tau} \cdot [\![\boldsymbol{n} \cdot \nabla \boldsymbol{u}]\!] = -\frac{\boldsymbol{f} \cdot \boldsymbol{\tau}}{\left|\frac{\partial \boldsymbol{\mathcal{X}}}{\partial s}\right|} \qquad (2.14e)$$

$$-[\![p]\!] = -\frac{\boldsymbol{f} \cdot \boldsymbol{n}}{\left|\frac{\partial \boldsymbol{\mathcal{X}}}{\partial s}\right|} \qquad (2.14f)$$

2.3 The Immersed Boundary Method

In this section, we will describe the basic form of the Immersed Boundary Method first proposed by Peskin [Pes72, Pes77] and which is still being used with minor modifications [PM92, DFFG96]. The details of more recent improvements, particularly those related to semi-implicit discretisations, will be postponed to Chapter 4 when they are needed.

The Immersed Boundary Method (discussed briefly in the Introduction) is a mixed Eulerian–Lagrangian finite difference scheme for computing the motion of immersed fibres. The fluid variables are defined on a fixed, Eulerian, $N \times N$ grid of points labeled $\vec{x}_{i,j} = (x_i, y_j) = (ih, jh)$, with spacing $h = \frac{1}{N}$ in both directions. The fluid domain is doubly–periodic so that the points x_0 and x_N are identified with each other, and similarly with y_0 and y_N. The fibre position is a Lagrangian quantity which is discretised as a set of N_b moving points, so that the parameter $s \in [0,1]$ is taken at discrete locations $s_\ell = \ell \cdot h_b$, where $h_b = \frac{1}{N_b}$. Both fluid and fibre quantities are sampled at equally–spaced times $t_n = n \cdot k$, where k is the time step. Figure 2.4 shows a typical fluid–fibre grid.

We can now write discrete approximations of the fluid velocity, pressure and force

$$\begin{aligned} \vec{U}^n_{i,j} &\approx \boldsymbol{u}(x_i, y_j, t_n) \\ P^n_{i,j} &\approx p(x_i, y_j, t_n) \qquad\qquad i,j = 0,1,\ldots,N-1, \\ \vec{F}^n_{i,j} &\approx \boldsymbol{F}(x_i, y_j, t_n) \end{aligned}$$

and the fibre position and force density

$$\begin{aligned} \vec{X}^n_\ell &\approx \boldsymbol{\mathcal{X}}(s_\ell, t_n) \\ \vec{f}^n_\ell &\approx \boldsymbol{f}(s_\ell, t_n) \qquad\qquad \ell = 0,1,\ldots,N_b-1, \end{aligned}$$

for $n = 0, 1, \ldots$. In addition, the delta function appearing in (2.9a) and (2.9c) is replaced by a discrete approximation $\delta_h(\boldsymbol{x})$, which is the product of two one–dimensional discrete delta functions

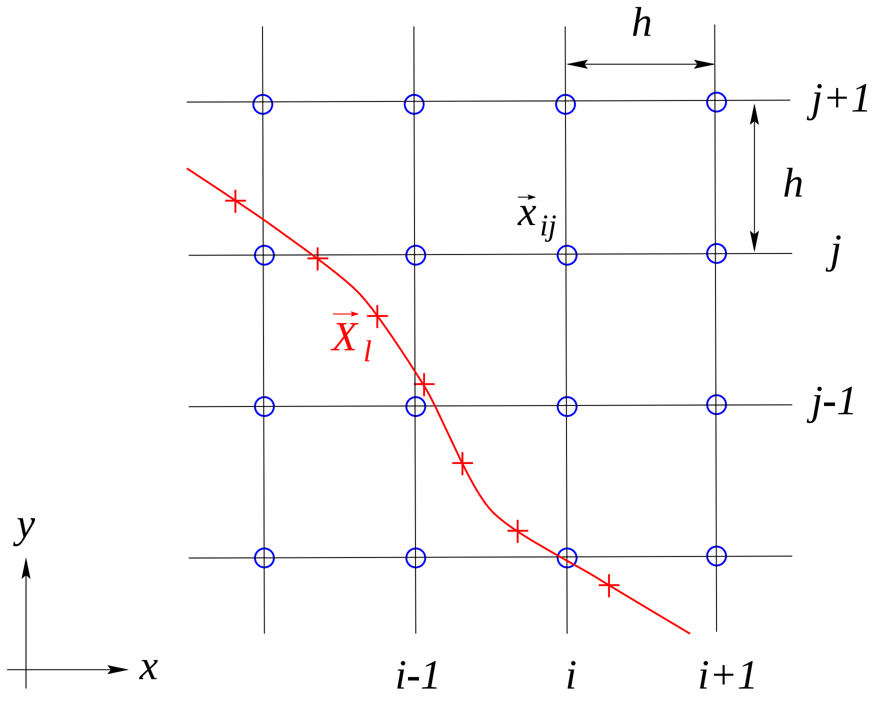


Figure 2.4: The relationship between fluid (◯) and fibre (+) grid points.

$$\delta_h(x_i, y_j) = d_h(x_i) \cdot d_h(y_j).$$

The choice of d_h typically used in immersed boundary computations is

$$d_h(r) = \begin{cases} \frac{1}{4h}\left(1 + \cos\frac{\pi r}{2h}\right) & \text{if } |r| < 2h, \\ 0 & \text{if } |r| \geq 2h. \end{cases} \tag{2.15}$$

which is pictured in its one– and two–dimensional incarnations in Figure 2.5. It will become clear in the algorithm to follow that $\delta_h(\boldsymbol{x})$ acts to interpolate quantities between the fluid and fibre grid points. This particular choice of approximate delta function is motivated by a set of discrete compatibility properties which are discussed in Section 2.3.1.

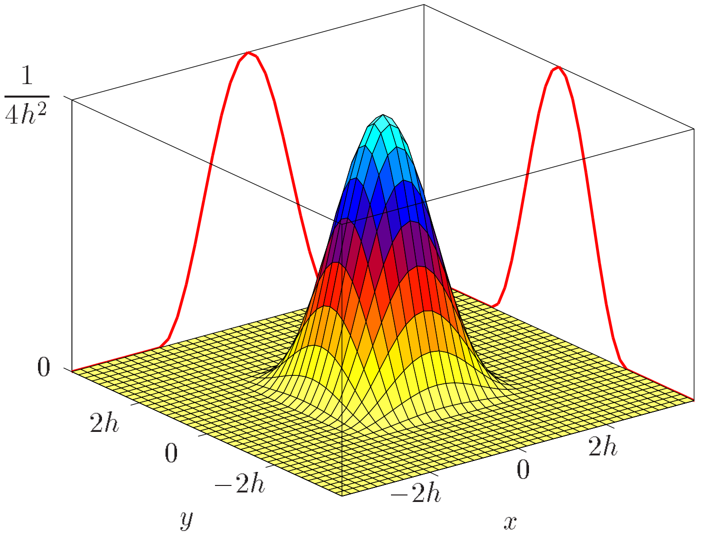


Figure 2.5: The cosine approximation $d_h(x, y)$ to the delta function.

To simplify the notation when writing the scheme, we will make use of several finite difference operators on fluid grid quantities:

- first order one–sided difference approximations to the first derivative

$$D_x^+\phi_{i,j} = \frac{\phi_{i+1,j} - \phi_{i,j}}{h},$$
$$D_x^-\phi_{i,j} = \frac{\phi_{i,j} - \phi_{i-1,j}}{h},$$

  and the second–order centered formula

$$D_x^0\phi_{i,j} = \frac{\phi_{i+1,j} - \phi_{i-1,j}}{2h};$$

- similar definitions for the y–derivative $D_y^+\phi_{i,j}$, $D_y^-\phi_{i,j}$ and $D_y^0\phi_{i,j}$;

- a centered difference formula for the gradient

$$\nabla_h\phi_{i,j} = (D_x^0,\, D_y^0)\,\phi_{i,j},$$

  and the Laplacian

$$\Delta_h\phi_{i,j} = (D_x^+D_x^- + D_y^+D_y^-)\,\phi_{i,j}.$$

Similarly, we can define one–sided difference approximations for the first derivative of fibre quantities

$$D_s^+\psi_\ell = \frac{\psi_{\ell+1} - \psi_\ell}{h_b},$$
$$D_s^-\psi_\ell = \frac{\psi_\ell - \psi_{\ell-1}}{h_b}.$$

We are now in a position to state the algorithm, which is a discrete version of equations (2.9a)–(2.9c). Assuming that the velocity $\vec{U}_{i,j}^n$ and fibre position $\vec{X}_{i,j}^n$ are known at time t_{n-1}, the procedure for updating these values to time t_n is as follows:

Immersed Boundary Method Algorithm *(FE/ADI)*

Step 1: Compute the fibre force density (assuming for simplicity that we have the linear force density function (2.7)):

$$\vec{f}^{n}_{\ell} = \sigma D^{+}_{s} D^{-}_{s} \vec{X}^{n-1}_{\ell} \tag{2.16a}$$

Step 2: Distribute the fibre force to fluid grid points:

$$\vec{F}^{n}_{i,j} = \sum_{\ell} \vec{f}^{n}_{\ell} \cdot \delta_h(\vec{x}_{i,j} - \vec{X}^{n-1}_{i,j}) \cdot h_b \tag{2.16b}$$

Step 3: Solve the Navier–Stokes equations using a split–step projection scheme (based on that proposed by Chorin [Cho68])

Step 3a: Apply the force, convection and diffusion terms using an alternating direction implicit (ADI) scheme [PT83, p. 66ff]

$$\rho \left(\frac{\vec{U}^{n,0}_{i,j} - \vec{U}^{n-1}_{i,j}}{k} \right) = \vec{F}^{n}_{i,j}, \tag{2.16c}$$

$$\rho \left(\frac{\vec{U}^{n,1}_{i,j} - \vec{U}^{n,0}_{i,j}}{k} + U^{n-1}_{i,j} D^{0}_{x} \vec{U}^{n,1}_{i,j} \right) = \mu D^{+}_{x} D^{-}_{x} \vec{U}^{n,1}_{i,j}, \tag{2.16d}$$

$$\rho \left(\frac{\vec{U}^{n,2}_{i,j} - \vec{U}^{n,1}_{i,j}}{k} + V^{n-1}_{i,j} D^{0}_{y} \vec{U}^{n,2}_{i,j} \right) = \mu D^{+}_{y} D^{-}_{y} \vec{U}^{n,2}_{i,j}, \tag{2.16e}$$

where velocity components in the convection terms are $\vec{U} = (U, V)$. Equation (2.16c) is an explicit formula for for $\vec{U}^{n,0}_{i,j}$, while the next two equations are periodic tridiagonal systems for the *intermediate velocities* $\vec{U}^{n,1}$ and $\vec{U}^{n,2}$.

Step 3b: Solve for the pressure and the velocity at the next time step using

$$\rho \left(\frac{\vec{U}^{n}_{i,j} - \vec{U}^{n,2}_{i,j}}{k} \right) + \nabla_h P^{n}_{i,j} = 0,$$

$$\nabla_h \cdot \vec{U}^{n}_{i,j} = 0.$$

This simultaneous system of equations for $\vec{U}^n_{i,j}$ and $P^n_{i,j}$ can be written in the more convenient form of a split–step projection procedure:

$$\Delta_{2h} P^n_{i,j} = \frac{\rho}{k} \nabla_h \cdot \vec{U}^{n,2}_{i,j}, \tag{2.16f}$$

$$\vec{U}^n_{i,j} = \vec{U}^{n,2}_{i,j} - \frac{k}{\rho} \nabla_h P^n_{i,j}, \tag{2.16g}$$

in which

- the Poisson equation (2.16f) is solved for the pressure $P^n_{i,j}$, where $\Delta_{2h} := \nabla_h \cdot \nabla_h$ is a wide five–point difference stencil for the Laplacian operator; and
- the computed pressure is used in (2.16g) to update the velocity field. The intermediate velocity $\vec{U}^{n,2}_{i,j}$ from the ADI step need not be incompressible, and so the pressure acts as a *Lagrange multiplier* to generate a velocity field $\vec{U}^n_{i,j}$ that is divergence–free.

This two–step process may be written more compactly as

$$\vec{U}^n_{i,j} = \mathcal{P}_h(\vec{U}^{n,2}_{i,j}),$$

where $\mathcal{P}_h$ represents the orthogonal projection operator onto the space of discretely divergence–free vector fields. Since the domain is doubly–periodic and the mesh is fixed and equally–spaced, the most efficient way to solve the pressure Poisson equation is by a Fast Fourier Transform (FFT) technique.

Step 4: Evolve the fibre at the new local fluid velocity

$$\frac{\vec{X}^n_\ell - \vec{X}^{n-1}_\ell}{k} = \sum_{i,j} \vec{U}^n_{i,j} \cdot \delta_h(\vec{x}_{i,j} - \vec{X}^{n-1}_\ell) \cdot h^2 \tag{2.16h}$$

Since this algorithm applies an ADI step to diffusion terms and Forward Euler to the fibre force and position, we will refer to it from now on as the *Forward Euler/ADI* or *FE/ADI* method. This designation will also serve to distinguish it from other schemes that we will consider later in Chapter 4.

We complete this section with a summary of the main characteristics of the Immersed Boundary Method:

1. The fluid equations are solved on a regular grid, which makes coding the method relatively simple and permits fast solvers to be applied (namely, an ADI step for the convection and diffusion terms, in combination with an FFT solve for the pressure). Furthermore, the method is easily vectorizable [FP88b, PM89].

2. Delta functions are used to interpolate quantities between the fixed fluid grid and moving fibre points. In Step 2, the force generated at a single fibre point s_ℓ is "spread out" to neighbouring fluid points that lie within a square of dimensions $4h \times 4h$ centered on the fibre point (see Figure 2.6). Similarly, $\delta_h(\boldsymbol{x})$ is used in Step 4 to compute the velocity of a fibre point as a weighted average of the fluid velocities in a $4h \times 4h$ neighbourhood.

3. A careful choice of approximation δ_h is required to ensure volume conservation, but it is also tied to the order of the scheme. Even though the Immersed Boundary Method is formally second–order accurate in space, Beyer & LeVeque prove [BL92] that for a one–dimensional model, the cosine delta function (2.15) actually reduces the scheme to first order. There is computational evidence to suggest that the use of approximate delta functions limits the discrete scheme to first order accuracy in two dimensional fluid flow as well [PM89, DFFG96, Rom96, LL97].

4. Computational results show that the time step restriction for this semi-implicit scheme is quite severe [PM92, TP92]. This limitation is due to the fact that the fibre force is handled explicitly, and many variations of the method have been proposed that handle the force implicitly, several of which will be described in Chapter 4.

To summarise, the main advantages of the scheme are its simplicity and geometric flexibility. While the fluid solver is definitely not state–of–the–art, the limitations on spatial accuracy and time step do not arise from any choice of solver but rather from the presence of the fibre and the use of approximate delta functions to smooth the fibre force. All things considered, the key role of

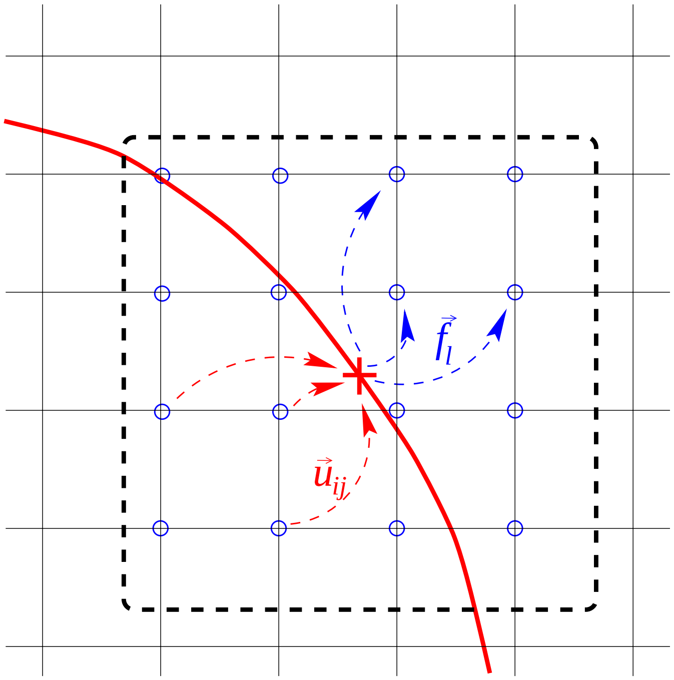


Figure 2.6: The approximate delta function is used to interpolate forces and velocities between fluid and fibre grid points in a neighbourhood of size $4h \times 4h$: it “distributes” fibre forces to fluid points, and computes the velocity of fibre points as a “weighted average” of velocities at neighbouring fluid points.

the Immersed Boundary Method as a qualitative tool for exploring complex biological phenomena is still unmatched.

2.3.1 Choice of discrete delta function

The choice of an approximation to the delta function used to transfer quantities between the Eulerian and Lagrangian mesh points is an integral part of the Immersed Boundary Method. The particular choice of a cosine shape (2.15) is *not unique*, and we will see in Section 3.2, that the choice of smoothing function for $\delta(\boldsymbol{x})$ has considerable influence on the behaviour of the solution modes for the linearised equations of motion. It also has important consequences related to the accuracy to which the solution modes for the "smoothed" problem match those of the exact or "jump" problem in an asymptotic sense (also in Section 3.2).

Peskin [Pes77, p. 230-232] presents a list of discrete compatibility conditions that should be satisfied by the approximation $d_h(x)$ in the Immersed Boundary Method. The function $d_h(x)$ is required to take the form

$$d_h(x) = \frac{1}{h}\,\phi\left(\frac{x}{h}\right),$$

where $\phi(r)$ is a function that satisfies the following properties:

I. $\boxed{\phi(r) \text{ is continuous}}$: so that the coefficients of the interpolation between fluid and fibre points appearing in (2.16b) and (2.16h) vary continuously as the fibre moves across fluid mesh lines.

II. $\boxed{\phi(r) = 0 \text{ for } |r| \geq 2}$: the support of the delta function must be finite in order that the cost of the Immersed Boundary Method be kept reasonable. Without this assumption, each fluid point would interact with every point on the fibre (and *vice versa*), and the computational cost would be prohibitive. The choice of two mesh points (that is, $|r| \leq 2$) for the width of support is the smallest integer that allows the remaining two properties to be satisfied.

III. $\boxed{\sum_{i\text{ (even)}} \phi(r-i) = \sum_{i\text{ (odd)}} \phi(r-i) = \frac{1}{2} \text{ for all } r}$: one consequence of this property is that $\sum_i \phi(r-i) = 1$ for all r, which guarantees that constant functions are interpolated exactly by d_h (this has the physical interpretation of conserving momentum when applied to force interpolation in (2.16b)). The motivation for the even/odd distinction is specific to the use of the wide stencil Δ_{2h} for Chorin's projection scheme in (2.16f), and ensures that contributions from the disjoint even and odd sub-grids are equal. This avoids oscillations that would otherwise be generated when interpolating the force.

IV. $\boxed{\sum_i \left[\phi(r-i)\right]^2 = C \text{ for all } r}$: where C is a constant. This property ensures that $\sum_i \phi(r_1 - i)\,\phi(r_2 - i) \leq C$ for all r_1 and r_2 (after applying the Schwarz inequality), which is analogous to the physically reasonable requirement that when two fibre points interact, the effect of one boundary point on the other is maximised when the points coincide. It can be shown, by setting $r = 0$ in II–IV, that $C = \frac{3}{8}$.

The sums in the above are taken over all integers i.

The cosine function (2.15) can be shown to satisfy all of these properties, with

$$\phi_c(r) = \begin{cases} \dfrac{1}{4}\left(1 + \cos\left(\dfrac{\pi r}{2}\right)\right), & \text{if } |r| \leq 2, \\ 0, & \text{if } |r| > 2. \end{cases}$$

However this choice is not unique, and Peskin and McQueen have derived an alternate approximation [PM95] which satisfies the additional property

V. $\boxed{\sum_i (r-i)\,\phi(r-i) = 0 \text{ for all } r}$: which ensures that linear functions are interpolated exactly by δ_h (with the physical implication that angular momentum is conserved when applying forces to fluid points).

This last property, in combination with I–IV above, *uniquely determines* the following function

$$\phi_2(r) = \begin{cases} \dfrac{1}{8}\left(3 - 2|r| + \sqrt{1 + 4|r| - 4r^2}\right), & \text{if } |r| \leq 1, \\ \dfrac{1}{2}\phi\left(2 - |r|\right), & \text{if } 1 < |r| \leq 2, \\ 0, & \text{if } 2 < |r|. \end{cases} \tag{2.17}$$

The function $\phi_2(r)$ is plotted in Figure 2.7 alongside the cosine approximation, from which it is clear that the two are nearly indistinguishable. It is because of this that Peskin & McQueen remark [PM95, p. 273]:

> *[the change in delta functions from $\phi_c(r)$ to $\phi_2(r)$] is not likely to have any practical effect on the computed results. We have gone ahead and made the change nonetheless, since it turns out that $\phi_2(r)$ is slightly cheaper to compute than $\phi_c(r)$, and since it is satisfying to have δ_h uniquely determined by a reasonable list of axioms.*

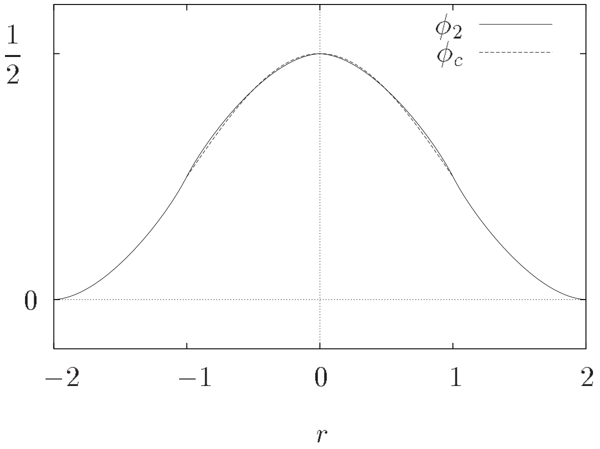


Figure 2.7: The cosine delta function ϕ_c plotted alongside the function ϕ_2 that satisfies the additional property V.

Another form of ϕ that satisfies a similar set of properties (with a reduced radius of support equal to 1.5, and no even/odd distinction in III) was derived by Roma [Rom96]. The corresponding approximate delta function is applicable to computations on a staggered *marker–and–cell* (or MAC) grid, where the problem of decoupled pressure modes inherent in Chorin's projection scheme is not an issue.

Something which has not been considered to date is the discrete second moment condition

VI. $\boxed{\sum_i (r-i)^2\,\phi(r-i) = 0 \text{ for all } r}$: which guarantees that quadratic functions are interpolated exactly by the corresponding δ_h. This is the discrete analogue of the continuous second moment condition that we will encounter in Section 3.2.2, in relation to accuracy of the interpolation in the continuous setting.

To find a discrete delta function that satisfies the additional property VI, we must increase the radius of support from $2h$ to $3h$. We can show with the help of MAPLE [C+91] that properties I–VI uniquely determine the following function:

$$\phi_{new}(r) = \begin{cases} \frac{1}{12}|r|^3 - \frac{11}{56}r^2 - \frac{11}{42}|r| + \frac{61}{112} + \frac{\sqrt{3}}{336}\Big(243 + 1584|r| \\ \qquad - 748r^2 - 1560|r|^3 + 500r^4 + 336|r|^5 - 112r^6\Big)^{\frac{1}{2}}, & \text{if } |r| \leq 1, \\ \frac{1}{6}|r|^3 - \frac{7}{8}r^2 + \frac{7}{12}|r| + \frac{21}{16} - \frac{3}{2}\,\phi_{new}\left(|r|-1\right), & \text{if } 1 < |r| \leq 2, \\ -\frac{1}{12}|r|^3 + \frac{3}{4}r^2 - \frac{23}{12}|r| + \frac{9}{8} + \frac{1}{2}\,\phi_{new}\left(|r|-2\right), & \text{if } 2 < |r| \leq 3, \\ 0, & \text{if } 3 < |r|. \end{cases} \tag{2.18}$$

The graph of ϕ_{new} is displayed in Figure 2.8. It is clear from an order of operations standpoint alone that $\phi_{new}(r)$ is more expensive to compute than any of the above-mentioned delta functions. Moreover, a more significant cost increase is engendered by an increase in the number of points over which the interpolation is performed from $5^2 = 25$ to $7^2 = 49$. We will see later in Section 3.2.2 that there may be an improvement in the accuracy of the scheme by using this more expensive

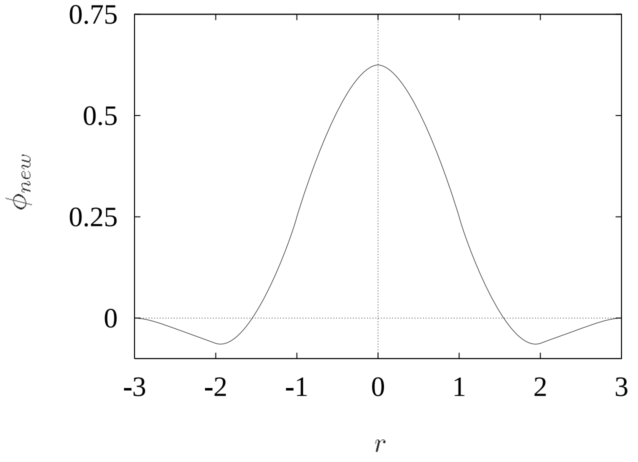


Figure 2.8: The delta function ϕ_{new} with smoothing radius 3, that satisfies the properties I–VI.

alternative. Whether or not the increase in accuracy is worth the extra computational effort is an issue that will be investigated in computations.

There has been some work done by Beyer & LeVeque [BL92] to increase the spatial accuracy of a one–dimensional analog of the Immersed Boundary Method to second order. They show that the approximate delta function must satisfy *one–sided discrete moment conditions*. An extension of this approach to two dimensions leads to the *Immersed Interface Method* [LL94], which replaces the delta function interpolation by modified difference stencils, with coefficients that are carefully chosen to interpolate the jump conditions based on truncation error expansions.

2.4 Choice of Parameters

Since the Immersed Boundary Method is so closely tied to biological applications, a description of the scheme and the underlying mathematical model would not be complete without a summary of the typical values of physical parameters appearing in the model. Furthermore, there are very standard computational test problems which appear over and over again in the literature, and so we present characteristic values of the numerical parameters as well.

The fluid within which the fibres are immersed is typically very similar to water. In studies of marine worms [FP88a] and flagellated cells [Fau93, FM95] in water, and blood flow in the heart [Pes77] and arteries [Ros94], the fluid is assumed to have density $\rho = 1 \; g/cm^3$ and viscosity $\mu = 1 \; g/cm \cdot s$. The fibre is taken to have a force with stress parameter σ in the range 10,000–250,000 $g/cm \cdot s^2$. The fluid domain is a square with sides of length $1-5 \; cm$ and periodic boundary conditions are applied in both directions.[d]

The choice for the fluid viscosity requires some explanation. The viscosity of blood is approximately 0.04 $g/cm \cdot s$, which corresponds to a flow with Reynolds number $\mathcal{R}e$ approximately equal to 300, when combined together with a characteristic length of 3.2 cm and velocity 3.7 cm/s. This

[d] This particular choice of domain and boundary conditions is reasonable for many applications and allows the use of fast solvers for the fluid equations.

choice of parameters is justified by Peskin [Pes77] for the human mitral valve. He then states that at this Reynolds number the problem is computationally intractable, since the time step required for stability in the numerical scheme decreases as $\mathcal{R}e$ increases.[e] To avoid this problem, the parameters are modified so that the Reynolds number is scaled by a factor of 0.01 to $\mathcal{R}e = 3$. This change is justified physically by Peskin as follows: there is a wide variation in Reynolds numbers for mammalian hearts (in the range 1–1000) and yet many are approximately scale models of each other. Consequently, the flow fields should not be very sensitive to the Reynolds number, and one can expect qualitatively reasonable results even though the viscosity is outside of the physical range.

We will now give a brief summary of the numerical parameters we will use in computations. The fluid grid spacing is typically taken to be of dimensions 64×64 with the fibre discretised at 128–256 points (which varies depending on the fibre configuration). This choice of $N = 64$ and $N_b = 128$–256 ensures that there is adequate resolution of the fibre relative to the fluid grid for the interpolation stages; namely, that there are at least two fibre grid points for every fluid cell (i.e., $h_b < \frac{1}{2}h$), while also not choosing h_b so small that the fibre is over–resolved. The typical time step used for a calculation is $k \approx 10^{-4}$, but depends on the choice of N and the physical parameters. The physical and numerical parameters that will be used in computations in the remainder of this thesis are summarised in Table 2.1 below.

[e] The actual dependence of the time step restriction on the parameters will be discussed in detail in Chapters 3 and 4.

Parameter	Description	Value or range
—	domain size	$1 - 5\ cm$
σ	fibre stress	$1 - 250 \times 10^3\ g/cm \cdot s^2$
μ	fluid viscosity	$1.0\ g/cm \cdot s$
ρ	fluid density	$1.0\ g/cm^3$
N	# of fluid grid pts.	64
N_b	# of fibre pts.	128–256
k	time step	$10^{-5} - 10^{-3}$

Table 2.1: Typical values of physical parameters used in immersed boundary computations.

Chapter 3
Linear Stability Analysis

> *An approximate answer to the right problem is worth a good deal more than an exact answer to an approximate problem.*
>
> — John Tukey.

As mentioned in the Introduction, a great deal of effort has gone into improving the Immersed Boundary Method and applying it to various physical problems. However, comparatively little work has been done on analysing the behaviour of solutions to the underlying equations of motion. LeVeque and others [LPL85, LPL88] applied a Fourier transform technique to find an explicit solution to a two–dimensional immersed boundary model of wave propagation in the basilar membrane, which is suspended in the fluid–filled cavity of the inner ear. They used a variation of the immersed fibre equations that was simplified in two ways: the fibre position is described by a vertical displacement function $y = h(x,t)$, so that tangential stretching of the fibre is neglected; and the pressure jump is taken to have a special functional form, justified by the physics of the problem. Fogelson [Fog92] has also applied a similar technique to determining the stability of the elastic links between platelets in a model of blood clotting.

In this chapter we will use an approach akin to that used in [LPL85, LPL88] and [Fog92] to perform a linear modal analysis of the immersed fibre problem in a more general form. It is not possible to solve the full problem explicitly, but we are able to obtain useful information about the stability and conditioning of fluid flows containing immersed fibres, which relates to the stiffness observed in immersed boundary computations. We will proceed in two stages, first by considering the exact formulation of the problem, and then introducing smoothing effects with approximate delta functions. The analysis for the exact problem, which is based on the jump formulation, has already appeared in [SW95], though we have made several minor modifications and generalisations in the discussion to follow.

3.1 Linear Stability of the Jump Problem

Consider a portion of the fluid domain on which the immersed fibre is approximately flat, labeled as Ω_0 in Figure 3.1. Suppose that the fibre lies at equilibrium along the horizontal line $y = 0$, and that the current fibre position is a small perturbation from this rest state. For the purpose of isolating the influence of the fibre on the flow, we extend the boundaries of Ω_0 to infinity in the y–direction. We can justify this modification of the fluid domain in three ways:

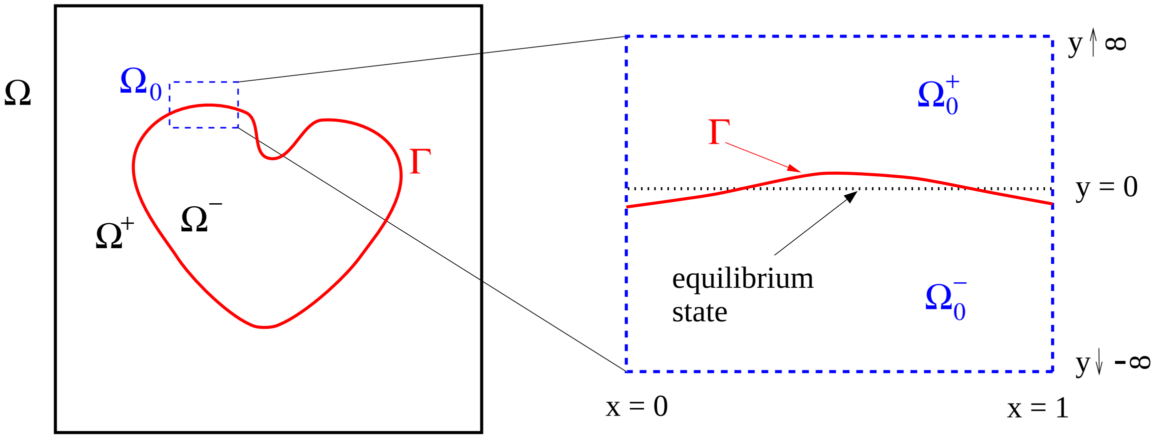


Figure 3.1: The two–dimensional fluid domain Ω with a subdomain Ω_0 on which the fibre is approximately flat.

1. the important dynamics that distinguish fluids with immersed fibres from those without should occur in the region near the fibre;

2. there are no non–trivial discrete modes of Stokes' equations without an immersed fibre on a domain of infinite extent, and so we expect to be able to pinpoint modes associated solely with the fibre;

3. the solution modes that we are most concerned with (that is, which have the most effect on

stability) are those with the largest wavenumber, and these are precisely those modes that are least affected by the presence of boundaries.

A common form of the fibre tension used in immersed boundary computations [TP92, PP93] is $T = T(|\partial \boldsymbol{\mathcal{X}}/\partial s| - 1)$ with $T(0) = 0$, corresponding to a fibre which is slack in the reference configuration $|\partial \boldsymbol{\mathcal{X}}/\partial s| = 1$. In actual computations, however, the fibre is almost always under stress except for possibly isolated instants of time. Hence, we choose an equilibrium state defined by $|\partial \boldsymbol{\mathcal{X}}/\partial s| = \theta \geq 1$, around which the solution is linearised by supposing a perturbation of the form

$$\boldsymbol{\mathcal{X}}(s,t) = (\theta s + \xi(s,t),\, \eta(s,t)) \tag{3.1}$$

(refer to Figure 3.2). We also make the linearity assumptions that ξ, η, $\boldsymbol{u}$ and their derivatives

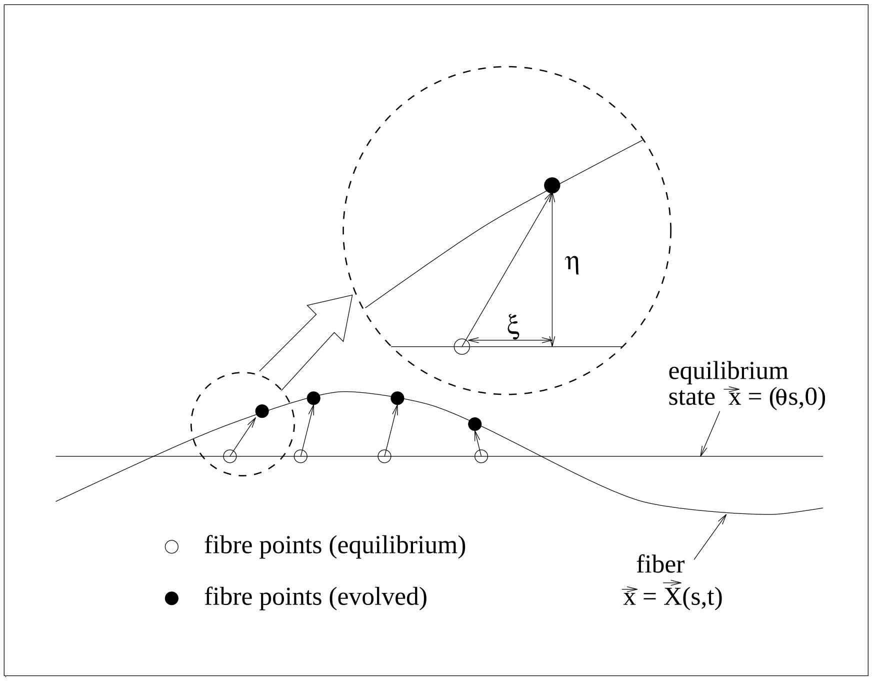


Figure 3.2: Fibre configuration for the linear stability analysis.

are small, at least for some finite time.

The linear versions of equations (2.14a) and (2.14b) are simply Stokes' equations

$$\rho \frac{\partial \boldsymbol{u}}{\partial t} = \mu \Delta \boldsymbol{u} - \nabla p \tag{3.2}$$

$$\nabla \cdot \boldsymbol{u} = 0, \tag{3.3}$$

while the fibre evolution equation (2.14c) becomes

$$\frac{\partial \boldsymbol{\mathcal{X}}}{\partial t} \approx \boldsymbol{u}(\theta s, 0, t). \tag{3.4}$$

Differentiating (3.1) with respect to s and dropping the nonlinear terms yields

$$\begin{aligned} \frac{\partial \boldsymbol{\mathcal{X}}}{\partial s} &\approx \left(\theta + \frac{\partial \xi}{\partial s}, \frac{\partial \eta}{\partial s} \right), \\ \left| \frac{\partial \boldsymbol{\mathcal{X}}}{\partial s} \right| &\approx \sqrt{\left(\theta + \frac{\partial \xi}{\partial s} \right)^2 + \left(\frac{\partial \eta}{\partial s} \right)^2}, \\ &\approx \theta + \frac{\partial \xi}{\partial s} \end{aligned}$$

which may then be used to obtain the linearised tangent vector (2.6)

$$\begin{aligned} \boldsymbol{\tau} &\approx \left(\theta + \frac{\partial \xi}{\partial s}, \frac{\partial \eta}{\partial s} \right) \Big/ \left(\theta + \frac{\partial \xi}{\partial s} \right) \\ &\approx \left(1, \frac{1}{\theta} \frac{\partial \eta}{\partial s} \right). \end{aligned}$$

Expand the tension T from (2.4) in a Taylor series about the equilibrium state $|\partial \boldsymbol{\mathcal{X}} / \partial s| = \theta$ to get

$$\begin{aligned} T &= T(\theta) + T'(\theta) \left(\left| \frac{\partial \mathcal{X}}{\partial s} \right| - \theta \right) + \cdots \\ &\approx T(\theta) + T'(\theta) \frac{\partial \xi}{\partial s}. \end{aligned}$$

The above expressions for T and $\boldsymbol{\tau}$ may be substituted into (2.5) to obtain the linear force density

$$\boldsymbol{f} = \left(\sigma_t \frac{\partial^2 \xi}{\partial s^2}, \sigma_n \frac{\partial^2 \eta}{\partial s^2} \right), \tag{3.5}$$

where $\sigma_n := T(\theta)/\theta$ and $\sigma_t := T'(\theta)$. We make the physically reasonable assumption that the fibre tension is an increasing function of the fibre strain, which for the linear force function amounts to taking $\sigma_n > 0$ and $\sigma_t > 0$. The following physical interpretation may be given to the two tension parameter values:

- σ_n, the *normal stress coefficient*, represents a constant tension in the fibre which (because of its positive sign) acts to restore the fibre to the horizontal whenever any portion is displaced vertically from its equilibrium state. Taking $\sigma_n = 0$ corresponds to a fibre which is slack in its reference state.

- σ_t, the *tangential stress coefficient*, measures the effect that changes in the length of the fibre have on the tension; this parameter is also positive, since stretching ($\frac{\partial^2 \xi}{\partial s^2} > 0$) or compressing ($\frac{\partial^2 \xi}{\partial s^2} < 0$) the fibre amounts to increasing or decreasing the tension.

The jump conditions (2.14d)–(2.14f) then reduce to

$$[\![u]\!] = 0, \tag{3.6}$$

$$[\![v]\!] = 0, \tag{3.7}$$

$$\left[\!\!\left[\mu \frac{\partial u}{\partial y}\right]\!\!\right] = -\sigma_t \frac{\partial^2 \xi}{\partial s^2}, \tag{3.8}$$

$$-[\![p]\!] = -\sigma_n \frac{\partial^2 \eta}{\partial s^2}. \tag{3.9}$$

The linearised version of the immersed fibre problem is now given by equations (3.2)–(3.9).

3.1.1 Derivation of the dispersion relation

To isolate the solution modes associated with the immersed fibre, we look for two separable solutions to the linearised problem of the form

$$\left\{\begin{array}{c} u \\ v \\ p \\ \xi \\ \eta \end{array}\right\}^{\pm} = e^{\lambda t + \hat{\imath}\alpha x} \left\{\begin{array}{c} \widehat{u}(y) \\ \widehat{v}(y) \\ \widehat{p}(y) \\ \widehat{\xi} \\ \widehat{\eta} \end{array}\right\}^{\pm}, \tag{3.10}$$

one on each of the subdomains Ω_0^+ and Ω_0^-. The wavenumber α is a real, positive number,[a] and $\hat{\imath} = \sqrt{-1}$ is the imaginary unit. By substituting (3.10) into (3.2)–(3.3) we obtain

$$\rho\lambda\widehat{u} = \mu\left(\frac{d^2}{dy^2} - \alpha^2\right)\widehat{u} - \hat{\imath}\alpha\widehat{p}, \tag{3.11}$$

$$\rho\lambda\widehat{v} = \mu\left(\frac{d^2}{dy^2} - \alpha^2\right)\widehat{v} - \frac{d\widehat{p}}{dy}, \tag{3.12}$$

$$\hat{\imath}\alpha\widehat{u} + \frac{d\widehat{v}}{dy} = 0. \tag{3.13}$$

We can then form the linear combination $\hat{\imath}\alpha \cdot (3.11) + \frac{d}{dy}(3.12)$ to get

$$\left(\rho\lambda + \mu\alpha^2 - \mu\frac{d^2}{dy^2}\right)\left(\hat{\imath}\alpha\widehat{u} + \frac{d\widehat{v}}{dy}\right) = \alpha^2\widehat{p} - \frac{d^2\widehat{p}}{dy^2},$$

the left hand side of which is zero by (3.13), leaving a Poisson equation for the pressure

$$\frac{d^2\widehat{p}}{dy^2} - \alpha^2\widehat{p} = 0. \tag{3.14}$$

After imposing the requirement that $\widehat{p}$ be bounded as $y \to \pm\infty$, the solution is determined on either side ("±") of the fibre to be

$$\widehat{p}^{\pm}(y) = A^{\pm}e^{\mp\alpha y} \tag{3.15}$$

where $A^{\pm}$ are as yet undetermined constants. We then substitute this expression for the pressure into the velocity equations (3.11) and (3.12) to get

$$\widehat{u}^{\pm}(y) = B^{\pm}e^{\mp\beta y} - \frac{\hat{\imath}\alpha}{\rho\lambda}A^{\pm}e^{\mp\alpha y}, \tag{3.16}$$

$$\widehat{v}^{\pm}(y) = C^{\pm}e^{\mp\beta y} \pm \frac{\alpha}{\rho\lambda}A^{\pm}e^{\mp\alpha y}, \tag{3.17}$$

where we have introduced the additional parameter β defined by

$$\beta^2 := \alpha^2 + \frac{\rho\lambda}{\mu} \tag{3.18}$$

[a] The $\alpha = 0$ case can be ruled out as it leads to the trivial solution. Furthermore, a symmetry argument can be used to show that $\alpha < 0$ leads to the same dispersion relation as for positive wavenumbers.

for convenience of notation. We assume that $Re(\beta) \geq 0$ and $\beta \neq \alpha$ in the above derivation.[b] The fibre unknowns $\widehat{\xi}$ and $\widehat{\eta}$ can then be found by substituting (3.16) and (3.17) into the expressions for the interface position

$$\widehat{\xi} = \frac{1}{\lambda}B^+ - \frac{\hat{\imath}\alpha}{\rho\lambda^2}A^+, \tag{3.19}$$

$$\widehat{\eta} = \frac{1}{\lambda}C^+ + \frac{\alpha}{\rho\lambda^2}A^+. \tag{3.20}$$

We are now in a position to find an explicit form of the solution, provided that the coefficients $A^{\pm}$, $B^{\pm}$ and $C^{\pm}$ can be determined. After applying the incompressibility condition (3.3) to the velocities on either side of the fibre, as well as the four jump conditions (3.6)–(3.9), we obtain a homogeneous system of six linear equations in the six unknown coefficients:

$$\begin{bmatrix} \frac{-\hat{\imath}\alpha}{\rho\lambda} & \frac{\hat{\imath}\alpha}{\rho\lambda} & 1 & -1 & 0 & 0 \\ \frac{\alpha}{\rho\lambda} & \frac{\alpha}{\rho\lambda} & 0 & 0 & 1 & -1 \\ \left(\frac{\hat{\imath}\mu\alpha^2}{\rho\lambda} + \frac{\hat{\imath}\sigma_t\alpha^3}{\rho\lambda^2}\right) & \frac{\hat{\imath}\mu\alpha^2}{\rho\lambda} & \left(-\mu\beta - \frac{\sigma_t\alpha^2}{\lambda}\right) & -\mu\beta & 0 & 0 \\ \left(-1 - \frac{\sigma_n\alpha^3}{\rho\lambda^2}\right) & 1 & 0 & 0 & -\frac{\sigma_n\alpha^2}{\lambda} & 0 \\ 0 & 0 & \hat{\imath}\alpha & 0 & -\beta & 0 \\ 0 & 0 & 0 & \hat{\imath}\alpha & 0 & \beta \end{bmatrix} \cdot \begin{bmatrix} A^+ \\ A^- \\ B^+ \\ B^- \\ C^+ \\ C^- \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{3.21}$$

To ensure that there exists a non-trivial solution, we require that the determinant of the system is zero, which after some manipulation reduces to the following expression

$$\underbrace{\left(\beta^4 + \alpha\beta^3 - \alpha^2\beta^2 - \alpha^3\beta + \frac{\rho\sigma_n}{2\mu^2}\alpha^3\right)}_{S_n(\beta)\ \text{— } \textit{normal modes}} \underbrace{\left(\beta^3 + \alpha\beta^2 - \alpha^2\beta - \alpha^3 + \frac{\rho\sigma_t}{2\mu^2}\alpha^2\right)}_{S_t(\beta)\ \text{— } \textit{tangential modes}} = 0. \tag{3.22}$$

This is a *dispersion relation*, that gives values of the *exponential time constant* λ (via (3.18)) in terms of the wavenumber α and the other parameters.

An important consequence of the linearisation process is that the modes corresponding to the normal and tangential motion of the fibre are decoupled. Of the two factors in equation (3.22),

[b] We exclude the case $\beta = \alpha$ (that is, $\lambda = 0$) because then $B^{\pm} = C^{\pm} = 0$ and we are left with the trivial solution.

the first, $S_n(\beta)$, depends on σ_n only and thus encompasses the normal motion of the fibre; $S_t(\beta)$ on the other hand corresponds to tangential motion. To illustrate this decoupling, we can fix the parameter values and solve explicitly for the coefficients $A^{\pm}$, $B^{\pm}$ and $C^{\pm}$. We choose $\mu = \rho = 1$, $\alpha = 2\pi$ and $\sigma_n = \sigma_t = 20\pi$, for which there are *four admissible roots* of the dispersion relation[c]: that is, four roots with $Re(\beta) > 0$, corresponding to two complex conjugate pairs. The four solution modes are pictured at time $t = 0$ in Figures 3.3 and 3.4, which include for each mode a vector plot of velocity and a surface plot of the pressure. Based on the mode plots, we can make the following observations:

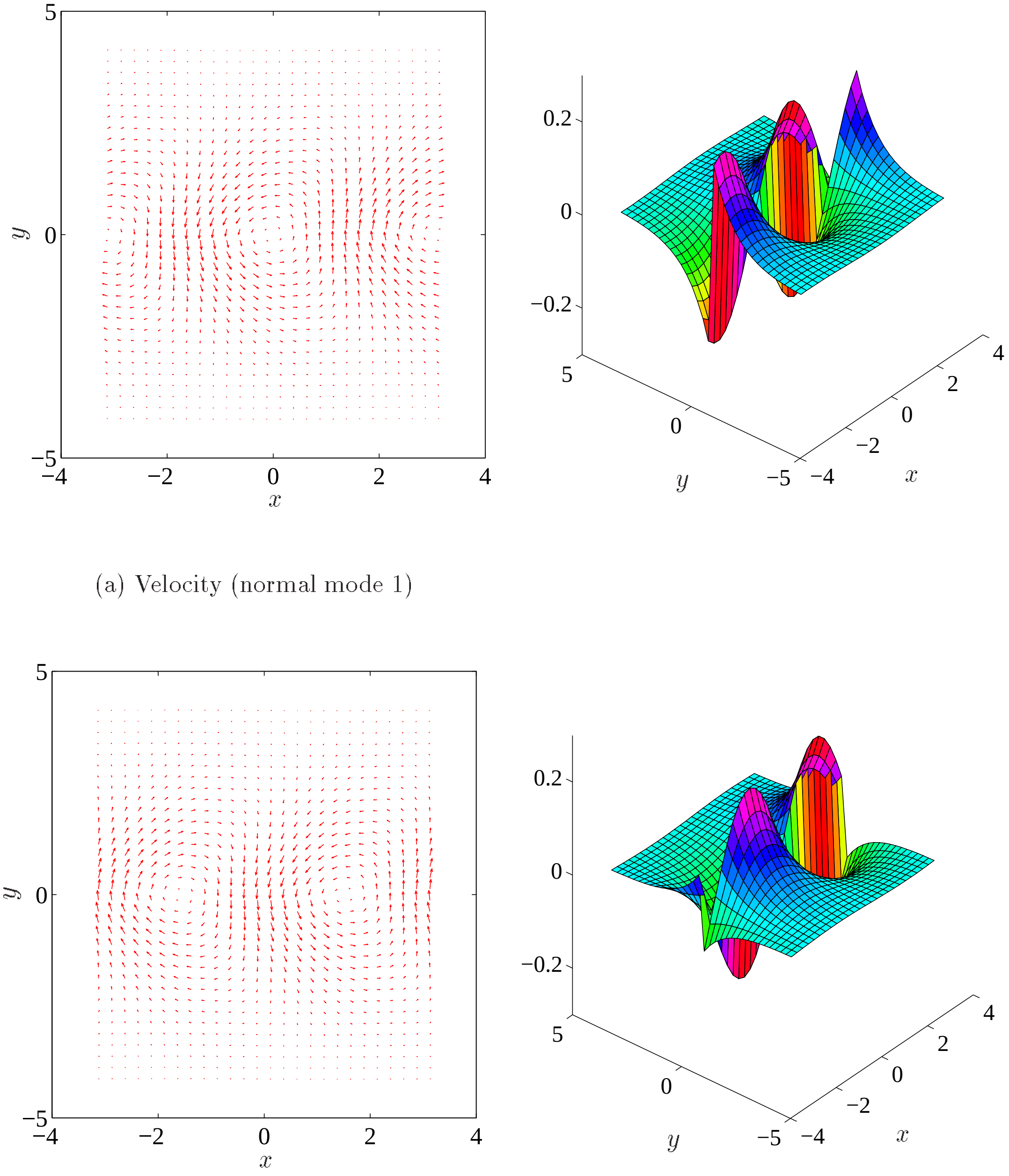
- The velocity vector plots of the two normal modes clearly illustrate a tendency for the fluid near $y = 0$ to move in the vertical (or normal) direction. Furthermore, the velocity appears to be smooth, while the corresponding pressure plots have a discontinuity at the fibre location $y \approx 0$. This is physically reasonable, as it is the pressure jump across the fibre which generates the normal motion.

- The second set of mode plots exhibit a velocity that moves tangentially to the fibre near $y = 0$. The pressure for these two modes is continuous, whereas the velocity appears to be non–smooth near the fibre. Hence, these are the tangential modes which arise from the jump in the tangential component of the normal derivative.

3.1.2 Stability results

To investigate the stability of the solution modes, we need to solve the dispersion relation for β given the wavenumber and the other parameters. We can then easily compute the corresponding values of the exponential time constant $\lambda = \mu/\rho(\beta^2 - \alpha^2)$, which embodies the behaviour of the solution in time. The growth or decay character of solutions is given by the real part of λ: if $Re(\lambda) \leq 0$ for all fibre modes, then the modes decay in time and are stable; otherwise, solutions are unstable. The oscillatory behaviour of the modes, on the other hand, is governed by the

[c] We will see in Section 3.1.4 that these parameter values are typical for physical problems, and so the four solution mode plots are representative of what is seen in actual computations.

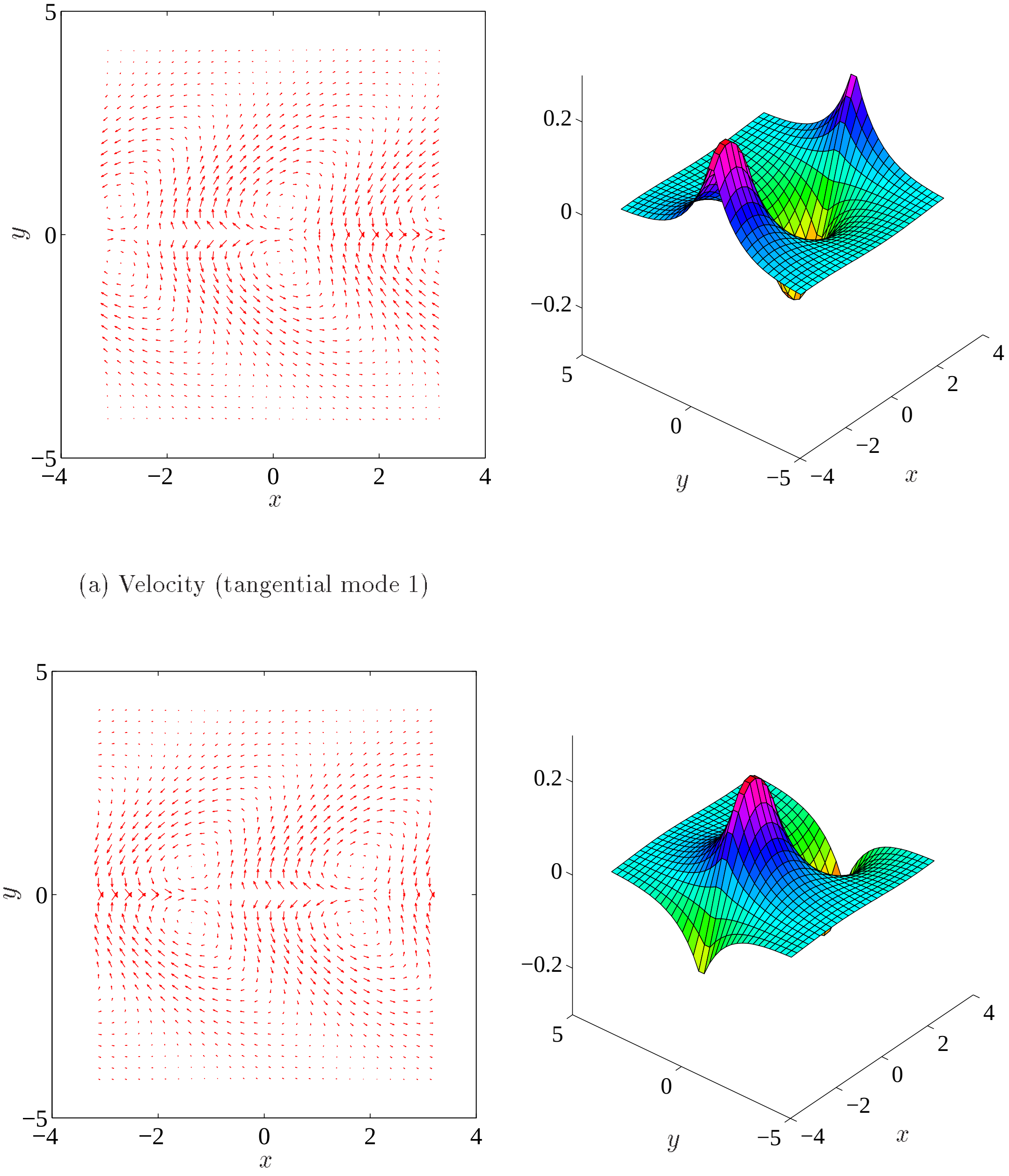


(a) Velocity (normal mode 1)

(c) Velocity (normal mode 2)

(d) Pressure (normal mode 2)

Figure 3.3: Plots of normal solution modes for the parameter values $\mu = \rho = 1$, $\sigma_n = 20\pi$, $\alpha = 2\pi$, $t = 0$.



(a) Velocity (tangential mode 1)

(c) Velocity (tangential mode 2)

(d) Pressure (tangential mode 2)

Figure 3.4: Plots of tangential solution modes for the parameter values $\mu = \rho = 1$, $\sigma_t = 20\pi$, $\alpha = 2\pi$, $t = 0$.

imaginary part of λ. The question of whether or not the fibre modes are stable is equivalent to showing that *all roots* of the dispersion relation satisfy $Re(\lambda) \leq 0$.

Before moving on to a discussion of stability, it will prove helpful to write the dispersion relation in non–dimensional form. We define the dimensionless parameters

$$\begin{aligned}
\widetilde{\beta} &= \beta \Big/ \left(\frac{\rho\sigma_t}{\mu^2}\right) \\
\widetilde{\alpha} &= \alpha \Big/ \left(\frac{\rho\sigma_t}{\mu^2}\right) \\
\widetilde{\lambda} &= \lambda \Big/ \left(\frac{\rho\sigma_t^2}{\mu^3}\right) \\
\gamma &= \frac{\sigma_n}{\sigma_t}
\end{aligned}$$

and substitute into (3.22) to get

$$\left(\widetilde{\beta}^4 + \widetilde{\alpha}\,\widetilde{\beta}^3 - \widetilde{\alpha}^2\widetilde{\beta}^2 - \widetilde{\alpha}^3\widetilde{\beta} + \frac{\gamma}{2}\widetilde{\alpha}^3\right) \cdot \left(\widetilde{\beta}^3 + \widetilde{\alpha}\widetilde{\beta}^2 - \widetilde{\alpha}^2\widetilde{\beta} - \widetilde{\alpha}^3 + \frac{1}{2}\widetilde{\alpha}^2\right) = 0. \tag{3.23}$$

From (3.18), it is easy to show that $\widetilde{\lambda} = \widetilde{\beta}^2 - \widetilde{\alpha}^2$. The use of dimensionless variables will make it easier for us to present the solution modes in terms of the wavenumber and the *single dimensionless parameter* γ.

We now present the following proposition, which is a stability proof for the fibre modes arising from the linearised problem.

Proposition 1. *If $\widetilde{\alpha} > 0$ and $\gamma > 0$ are real and positive, then all $\widetilde{\beta} \in \mathbb{C}$ satisfying the dispersion relation (3.23) with $Re(\widetilde{\beta}) > 0$ have the property*

$$Re(\widetilde{\beta}^2 - \widetilde{\alpha}^2) < 0.$$

Proof. The proof gives a geometric representation of the roots of the dispersion relation and stresses that the result holds for any positive values of the parameters $\widetilde{\alpha}$ and γ.

Let us first denote the two polynomial factors in (3.23) by

$$f(z) = z^4 + z^3 - z^2 - z + \frac{\gamma}{2\widetilde{\alpha}}, \text{ and}$$
$$g(z) = z^3 + z^2 - z - 1 + \frac{1}{2\widetilde{\alpha}},$$

where we have used the substitution $\widetilde{\beta} = \widetilde{\alpha} z$, $z = x + \hat{\imath} y \in \mathbb{C}$, and $x = Re(z) > 0$. We need to show that all roots of f and g satisfy the condition

$$Re(\widetilde{\alpha}^2 (x + \hat{\imath} y)^2 - \widetilde{\alpha}^2) < 0 \quad \text{or simply} \quad x^2 < y^2 + 1. \tag{3.24}$$

We proceed with the proof in two cases, first for the quartic factor $f(z)$.

Quartic factor: Any root of f must satisfy $Re(f) = 0$ and $Im(f) = 0$. We show that for $y = Im(z) > 0$, we have $Im(f(z)) > 0$ on the region $\mathcal{D} := \{z = x + \hat{\imath} y \,|\, x^2 \geq y^2 + 1 \text{ and } x, y > 0\}$, which is plotted in Figure 3.5. The situations $y = 0$ and $y < 0$ are discussed below.

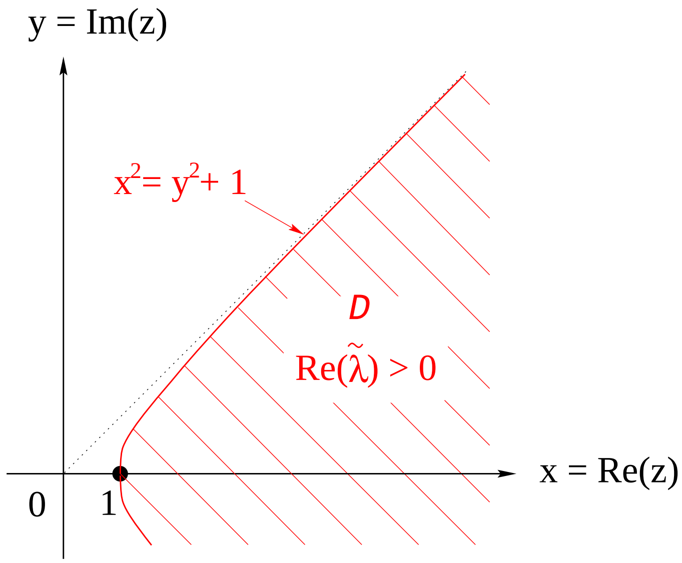


Figure 3.5: The domain D from Proposition 1, corresponding to the unstable solutions $Re(\widetilde{\lambda}) > 0$.

We proceed by simplifying $Im(f)$ on the region $\mathcal{D}$ where $-y^2 \geq 1 - x^2$ and $y > 0$:

$$\begin{aligned} Im(f) &= y\,\left(4x^3 + 3x^2 - 2x - 1 - y^2\,(4x+1)\right) \\ &\geq y\,\left(4x^3 + 3x^2 - 2x - 1 + (1-x^2)\,(4x+1)\right) \\ &= 2xy(x+1) \\ &> 0 \quad \text{when } x, y > 0. \end{aligned}$$

Therefore, there are *no roots* of f on the region $\mathcal{D}$ and hence $Re(\widetilde{\lambda}) < 0$ for all roots $z = x + \hat{\imath}y$, $y > 0$, of $f(z)$.

When $y < 0$, the argument proceeds in an identical fashion (for the reflection of the region $\mathcal{D}$ across the x–axis) except that

$$\begin{aligned} Im(f) &= y\,\left(4x^3 + 3x^2 - 2x - 1 - y^2\,(4x+1)\right) \\ &\leq 2xy(x+1) \\ &< 0 \quad \text{when } x > 0 \text{ and } y < 0. \end{aligned}$$

If $y = 0$, then $Im(f) = 0$ and we instead consider $Re(f(z))$ on $\mathcal{D}$ where we now have $x^2 - 1 \geq 0$, so that:

$$\begin{aligned} Re(f) &= x^2(x^2-1) + x(x^2-1) + \frac{\gamma}{2\widetilde{\alpha}} \\ &> 0 \quad \text{since } \gamma, \widetilde{\alpha} > 0. \end{aligned}$$

Again there are no roots of f when $Re(\widetilde{\lambda}) > 0$. This completes the proof for the quartic polynomial.

Cubic factor: Using a similar argument, we can show that $Re(\widetilde{\lambda}) < 0$ for all roots of g:

$$\begin{aligned} Im(g) &= y\,\left(3x^2 + 2x - 1 - y^2\right) \\ &\geq y\,\left(3x^2 + 2x - 1 + (1-x^2)\right) \\ &= 2xy(x+1) \\ &> 0 \quad \text{for } x, y > 0. \end{aligned}$$

The results for $y < 0$ and $y = 0$ follow using a similar argument used for f above.

$\blacksquare$

An immediate consequence of the proposition is that the discrete modes associated with the linear immersed fibre are stable for all physically reasonable parameter values. It does not prove, however, that the immersed fibre problem is stable, though we expect that this result is true. Such a statement would require a full non-linear analysis of the *continuous spectrum* of the problem, which is beyond the scope of this work.

We can now demonstrate our theoretical result by plotting the exponential time constants. Since (3.23) is a polynomial equation in $\widetilde{\beta}$ with factors of degree 3 and 4, analytic expressions for the roots are easily derived using a symbolic algebra package such as Maple [C+91]. A contour plot of the largest growth/decay rate, $Re(\widetilde{\lambda})$, is given in Figure 3.6 for a range of $\widetilde{\alpha}$ and γ, with the region of stability (where $Re(\widetilde{\lambda}) < 0$) lying above and to the right of the zero contour. It is not surprising that this is precisely the region where $\gamma > 0$ (corresponding to $\sigma_n > 0$ and $\sigma_t > 0$); that is, the fibre modes are stable when the tension force acts to oppose any stretching or normal displacement of the fibre.

3.1.3 Asymptotic expansions of decay rates

Now that the modes associated with the fibre are known to be linearly stable, we will attack the question "How rapidly do perturbations in the fibre die out in time?" and the related issue "How stiff are the fibre modes?" The answer to both questions is embodied in the dependence of λ on the wavenumber and the other physical parameters.

Even though we have explicit formulas for the roots of the dispersion relation, the expressions are too complicated to exhibit any clear functional dependence. We will take an alternative approach, and derive asymptotic expansions of λ for large and small non–dimensionalised wavenumbers. The small $\widetilde{\alpha}$ case is the one we are actually interested in (since the physical parameters lie in this range), but we include the case of large and intermediate wavenumbers for the sake of

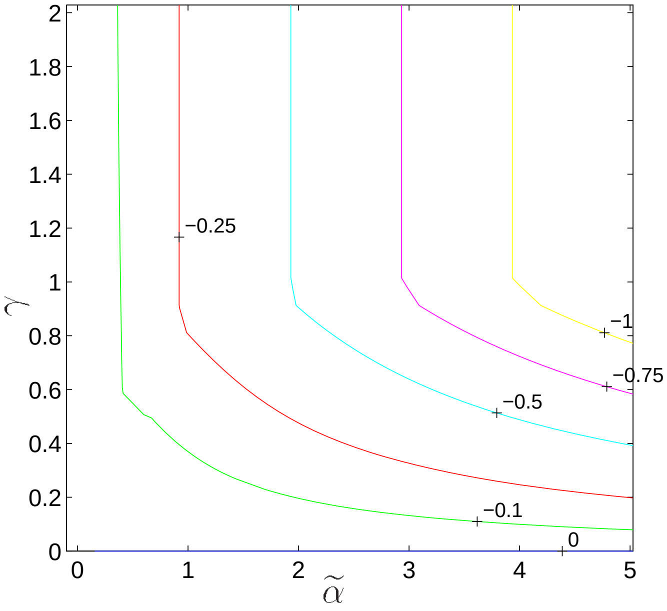


Figure 3.6: Contours of the maximum scaled growth/decay rate $Re(\widetilde{\lambda})$ plotted versus γ and scaled wavenumber $\widetilde{\alpha}$. The contours are all negative and so the fibre modes are *stable*.

completeness.

Small wavenumber

Equation (3.23) was solved for $\gamma = 1$, and all *admissible roots* (that is, roots for which $Re(\widetilde{\beta}) > 0$) are plotted in Figures 3.7 for values of $\widetilde{\alpha} \in [0, 0.6]$. An explicit form for the dependence of the exponential time constant on the wavenumber may be determined by computing a regular asymptotic expansion for each root in powers of the non-dimensionalised wavenumber $\widetilde{\alpha}$, as $\widetilde{\alpha} \to 0$. There are in fact two complex conjugate pairs of roots, given in terms of dimensional variables as

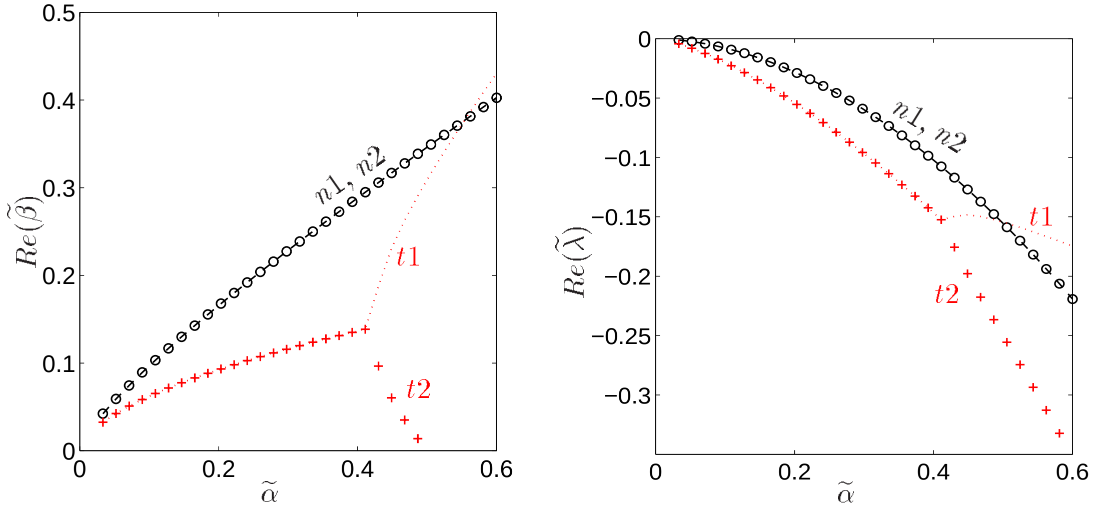


Figure 3.7: Plots of the real parts of $\widetilde{\beta}$ and $\widetilde{\lambda}$ versus wavenumber $\widetilde{\alpha}$ ($\gamma = 1.0$) in the small wavenumber regime.

$$\begin{aligned}\lambda_0^{n1},\ \lambda_0^{n2} &\sim \pm\frac{\hat{\imath}\mu}{\rho}\left(\frac{\rho\sigma_n}{2\mu^2}\right)^{\frac{1}{2}}\alpha^{\frac{3}{2}} - \frac{\mu(1\pm\hat{\imath})}{2\sqrt{2}\rho}\left(\frac{\rho\sigma_n}{2\mu^2}\right)^{\frac{1}{4}}\alpha^{\frac{7}{4}} - \frac{\mu}{4\rho}\alpha^2 + \mathcal{O}(\widetilde{\alpha}^{\frac{9}{4}}),\\ \lambda_0^{t1},\ \lambda_0^{t2} &\sim -\frac{\mu(1\pm\hat{\imath}\sqrt{3})}{2\rho}\left(\frac{\rho\sigma_t}{2\mu^2}\right)^{\frac{2}{3}}\alpha^{\frac{4}{3}} - \frac{\mu(1\mp\hat{\imath}\sqrt{3})}{3\rho}\left(\frac{\rho\sigma_t}{2\mu^2}\right)^{\frac{1}{3}}\alpha^{\frac{5}{3}} + \mathcal{O}(\widetilde{\alpha}^{\frac{7}{3}}),\end{aligned} \tag{3.25}$$

which substantiates the results in Figure 3.7. Note that beyond a value of $\widetilde{\alpha} \approx 0.42$, two of the roots merge and split into a pair of real roots, and shortly thereafter (at $\widetilde{\alpha} \approx 0.5$), one of those roots ($\lambda^{t2}$) is inadmissible after $Re(\beta^{t2})$ becomes negative. These two complex conjugate pairs of roots are what give rise to the solution modes pictured earlier in Figures 3.3 and 3.4.

Intermediate wavenumber

Plots of the roots $\widetilde{\beta}$ and exponential time constants $\widetilde{\lambda}$ in the intermediate wavenumber range $\widetilde{\alpha} \in [0, 1.5]$ are given in Figure 3.8 We can see another bifurcation point at $\widetilde{\alpha} \approx 0.8$ where the complex conjugate pair of normal roots merges and splits into two distinct real solutions.

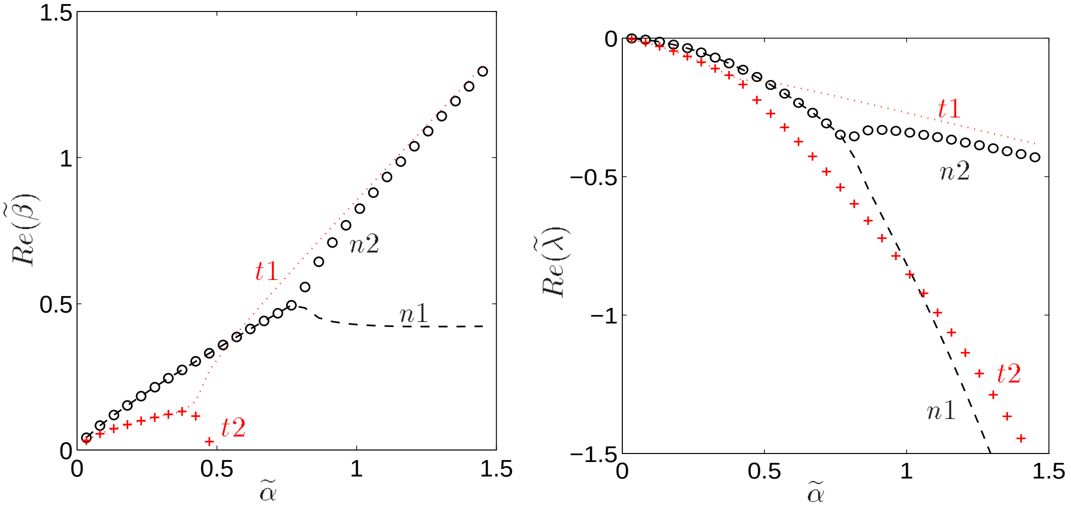


Figure 3.8: Plots of $\widetilde{\beta}$ and $\widetilde{\lambda}$ for intermediate values of the wavenumber. $(\gamma = 1.0)$.

Large wavenumber

Figure 3.9 contains a plot of the scaled $\widetilde{\lambda}$ for values of $\widetilde{\alpha}$ in the intermediate range $[0, 4]$. For values of $\widetilde{\alpha} \gtrsim 0.8$, there are three real roots, corresponding to two normal modes and one tangential mode. We can see investigate more closely the dependence of λ on α for large α in this regime by once again performing an asymptotic expansion of the roots as $\widetilde{\alpha} \to \infty$, from which we obtain the following expressions:

$$\lambda_\infty^{n1} \sim -\frac{\mu}{\rho}\alpha^2 + \frac{\rho\sigma_n^2}{4\mu^3} - \frac{\rho^2\sigma_n^3}{4\mu^5}\alpha^{-1} + \mathcal{O}(\widetilde{\alpha}^{-2}),$$

$$\lambda_\infty^{n2} \sim -\frac{\sigma_n}{4\mu}\alpha - \frac{3\rho\sigma_n^2}{64\mu^3} - \frac{19\rho^2\sigma_n^3}{1024\mu^5}\alpha^{-1} + \mathcal{O}(\widetilde{\alpha}^{-2}),$$

$$\lambda_\infty^{t1} \sim -\frac{\sigma_t}{4\mu}\alpha - \frac{\rho\sigma_t^2}{64\mu^3} - \frac{3\rho^2\sigma_t^3}{1024\mu^5}\alpha^{-1} + \mathcal{O}(\widetilde{\alpha}^{-2}).$$

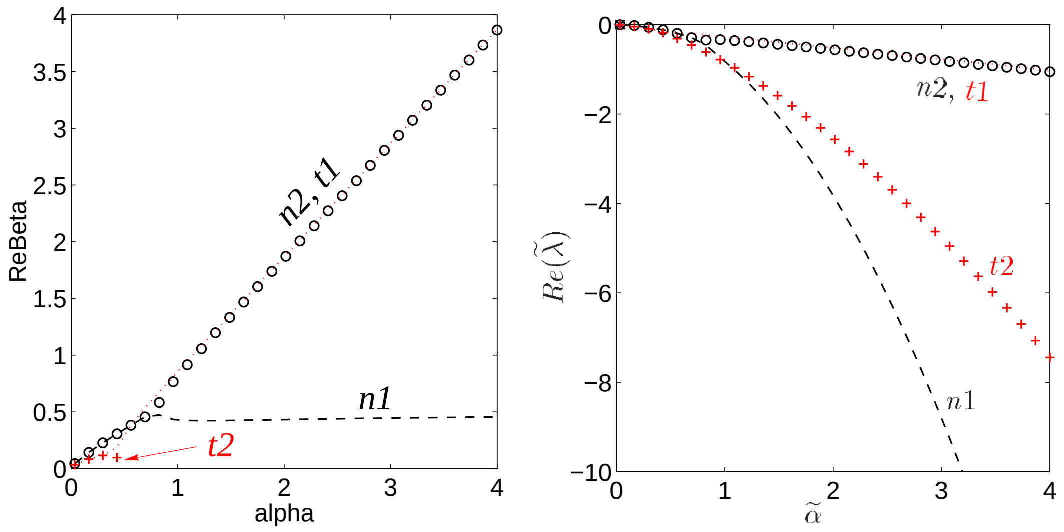


Figure 3.9: Plots of the real parts of $\widetilde{\beta}$ and the scaled decay rate versus wavenumber for each of the admissible roots in the large wavenumber regime.

By comparing the plots of the solution modes in Figure 3.9 to the corresponding asymptotic formulas, it is easy to see that two of the three modes (λ^{n2} and λ^{t1}) are nearly identical for large $\widetilde{\alpha}$ (when $\sigma_n = \sigma_t$) and depend linearly on the wavenumber.

The quadratic dependence of the remaining normal mode on α can also be seen on the graph — to lowest order in α this mode behaves like a Stokes mode. Similar to solutions of Stokes equations without an immersed fibre, the mode represented by λ^{n1}_{∞} gives rise to stiffness in the problem, for as the wavenumber varies, the exponential time constants take on widely disparate values.

3.1.4 Comparison with computations

Many researchers have observed in the course of immersed boundary computations [PM89, Rom96, DFFG96] that a very small time step is required to achieve stability. This seems to suggest that the problem is stiff, and we'd like to find an explanation for this stiffness in terms of the analytical results above. Typical computations, such as those in [MP93], are performed on a domain $\Omega = [0,1]^2$ with a grid spacing of $h = 1/64$ (all measurements being in cm). On this grid

<table>
<tr><th rowspan="2">Mode</th><th rowspan="2">Normal/
Tangential</th><th colspan="4">Interval of wavenumber $\widetilde{\alpha}$ (approximate)</th></tr>
<tr><th>$[0, 0.42]$</th><th>$[0.42, 0.50]$</th><th>$[0.50, 0.80]$</th><th>$[0.80, \infty]$</th></tr>
<tr><td>λ^{n1}</td><td>normal</td><td colspan="3">$\mathbb{C}$</td><td>$\mathbb{R}$</td></tr>
<tr><td>λ^{n2}</td><td>normal</td><td colspan="3">$\mathbb{C}$</td><td>$\mathbb{R}$</td></tr>
<tr><td>λ^{t1}</td><td>tangential</td><td>$\mathbb{C}$</td><td>$\mathbb{R}$</td><td colspan="2">$\mathbb{R}$</td></tr>
<tr><td>λ^{t2}</td><td>tangential</td><td>$\mathbb{C}$</td><td>$\mathbb{R}$</td><td colspan="2">$\mathbb{R}$</td></tr>
<tr><td>λ^{t3}</td><td>tangential</td><td colspan="4">$\mathbb{R}$</td></tr>
<tr><td>λ^{n3}</td><td>normal</td><td colspan="4">$\mathbb{C}$</td></tr>
<tr><td>λ^{n4}</td><td>normal</td><td colspan="4">$\mathbb{C}$</td></tr>
</table>

Table 3.1: Summary of the character of $\widetilde{\lambda}$ for various intervals of scaled wavenumber, with $\mathbb{R}$ representing a real root, and $\mathbb{C}$ a complex root. The shaded entries correspond to the inadmissible roots of the dispersion relation.

the method can resolve modes with a maximum wavenumber of $\alpha_{max} = 2\pi/h = 128\pi$, commonly referred to as the *Nyquist frequency.* We can therefore assess the stiffness of the method by considering the decay of solution modes with wavenumbers $\alpha = 2\pi n$, for $n = 1, 2, \ldots, 64$.

As mentioned earlier, it is the small wavenumber regime which is of most interest, because this is where the scaled wavenumbers are found to lie, based on the parameters values used in typical computations. Using the information provided in [MP93], we select the parameters to be $\mu = 1\ g/cm \cdot s$, $\rho = 1\ g/cm^3$ and $\sigma_n = \sigma_t = 100,000\ g/s^2$ (that is, $\gamma = 1$). We then restrict ourselves to a discrete set of wavenumbers, $\alpha = 2\pi \cdot n$, where $n = 1, 2, \ldots, 64$, which can be thought of as an *idealised discretisation* of the problem (since these are the wavenumbers that can be resolved in computations on a regular 64×64 grid of points). The non–dimensionalised wavenumber then lies in the range $\widetilde{\alpha} \in [6.3 \times 10^{-5}, 4.0 \times 10^{-3}]$, which clearly places the numerical examples in the small wavenumber regime.

Examining the asymptotic expressions for the decay rates as $\widetilde{\alpha} \to 0$, we find that for the parameters used in the previous paragraph, the fibre modes satisfy:

$$\left.\begin{array}{r} -2.3 \times 10^5 \leq Re(\lambda_0^{n1,n2}) \leq -1.4 \times 10^2 \\ -2.3 \times 10^6 \leq Re(\lambda_0^{t1,t2}) \leq -8.1 \times 10^3 \end{array}\right\} \quad \Longrightarrow \quad -2.3 \times 10^6 \leq Re(\lambda_0) \leq -1.4 \times 10^2$$

$$\left.\begin{array}{r} 3.4 \times 10^3 \leq |Im(\lambda_0^{n1,n2})| \leq 1.6 \times 10^6 \\ 1.3 \times 10^4 \leq |Im(\lambda_0^{t1,t2})| \leq 3.0 \times 10^6 \end{array}\right\} \quad \Longrightarrow \quad 3.4 \times 10^3 \leq |Im(\lambda_0)| \leq 3.0 \times 10^6$$

The modes for Stokes equations without an immersed fibre are simply $\lambda^s = -\mu\alpha^2/\rho$, which lie in the range

$$-1.6 \times 10^5 \leq \lambda^s \leq -39.$$

The decay rates for the fibre modes vary over a range that is an order of magnitude larger, and hence the immersed fibre problem is considerably more stiff than Stokes flow without a fibre. Furthermore, it is the tangential modes which make the problem more stiff, since the normal growth rates are approximately the same size as the Stokes modes.

To help account for the severe limitation on the Reynolds number that can be handled by immersed boundary computations, we consider the decay rates arising from parameters that correspond to a more realistic situation. By choosing $\mu = 0.01\, g/cm \cdot s$, which is more representative of the viscosity of blood in the human heart, we find

$$-9.4 \times 10^6 \leq Re(\lambda_0) \leq -13 \quad \text{and} \quad 3.5 \times 10^3 \leq |Im(\lambda_0)| \leq 1.6 \times 10^7$$

and for Stokes modes

$$-1.6 \times 10^3 \leq Re(\lambda^s) \leq -0.4.$$

It is evident that the fibre modes are now more than three orders of magnitude larger than Stokes' modes, which clearly identifies one source of the problems encountered with the Immersed Boundary Method at high Reynolds numbers. Furthermore, from (3.25) we can see that the leading order terms in the decay rate expansions are

$$\begin{aligned} Re(\lambda_0^{n1,n2}) &\sim \mathcal{O}\left(\sigma_n^{1/4} \cdot \rho^{-3/4} \cdot \mu^{1/2}\right), \\ Re(\lambda_0^{t1,t2}) &\sim \mathcal{O}\left(\sigma_t^{2/3} \cdot \rho^{-1/3} \cdot \mu^{-1/3}\right). \end{aligned}$$

The normal modes depend less strongly on σ_n and decrease in magnitude as the viscosity is reduced; hence, they can be characterised as *perturbed Stokes modes.* The tangential modes, on the other hand, are the major source of stiffness in the problem as they depend more strongly on the fibre forcing parameter and also grow with decreasing viscosity. Hence, the problems encountered with immersed boundary computations at high Reynolds numbers are due neither to the stiffness of Stokes modes nor the onset of turbulence. Rather it is the tangential modes of oscillation in the fibre that introduce stiffness through the combination of a large fibre force and small viscosity.

In terms of numerical computations, the presence of stiff modes suggests the use of an implicit time-stepping scheme. The analytical justification given here backs up the conclusion of Tu & Peskin in [TP92] that by applying a fully implicit scheme, a considerable improvement in numerical stability can be realised. They also observe that *"in its present form, the fully implicit scheme is* *probably too expensive for practical application"* [TP92, p. 1376], and suggest that a more efficient implementation be developed.

To further test the analytical results we have implemented the Immersed Boundary Method algorithm as stated in Section 2.3, and compared decay rates of the computed solution to the predicted values. Since no exact solutions to the immersed fibre problem are known, our analytical expressions for the decay rate of the lowest wavenumber mode present the first opportunity for comparison to be made to an analytical solution.

We have set up a test problem pictured in Figure 3.10, which is specially tailored to verify our analytical results and which has incidentally not appeared in the literature to date. The fluid domain is a unit square with periodic boundary conditions in the x and y directions. The initial fibre position is a sinusoidal curve with equation

$$\boldsymbol{\mathcal{X}}(s,0) = \left(s,\, \frac{1}{2} + A \sin\left(2\pi s\right) \right),$$

which is also required to satisfy periodic boundary conditions at $x = 0$ and 1. The constant A is set to 0.05 in the examples computed here. The force density is taken to be the linear function $\boldsymbol{f} = \sigma\, \partial^2 \boldsymbol{\mathcal{X}} / \partial s^2$ from (2.7), where $\sigma := \sigma_n = \sigma_t$. This form of the initial fibre position excites the normal mode of oscillation, which we will see is the dominant solution mode for this problem.

To compare the analytical and computed results, we examine the solution mode with the smallest decay rate (for $\alpha = 2\pi$), which corresponds to the mode of oscillation that will dominate the solution after a short time. The quantity which serves as the simplest basis for computing the decay rate is the maximum height of the immersed fibre. Figure 3.11 provides a sample plot of the computed maximum height of the immersed fibre as a function of time, which oscillates very regularly and has an amplitude that decays with time. There are two quantities that can easily be obtained from this information in order to draw comparisons with the analytical results:

- the decay rate, $Re(\lambda)$, for the $\alpha = 2\pi$ mode which can be determined by measuring the rate at which the maximum fibre height decays to zero (from the diagram, $\frac{1}{t_2 - t_1} \ell n[H_2/H_1]$) ; and

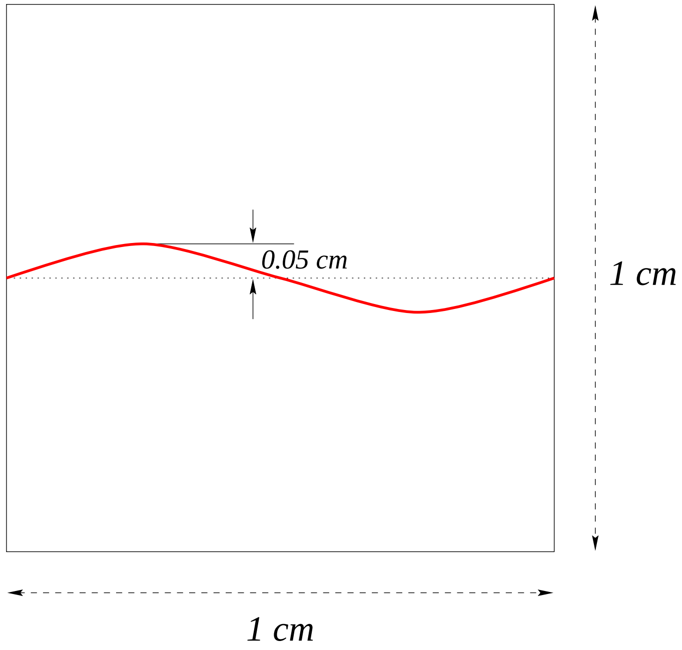


Figure 3.10: The test problem for the linear stability results.

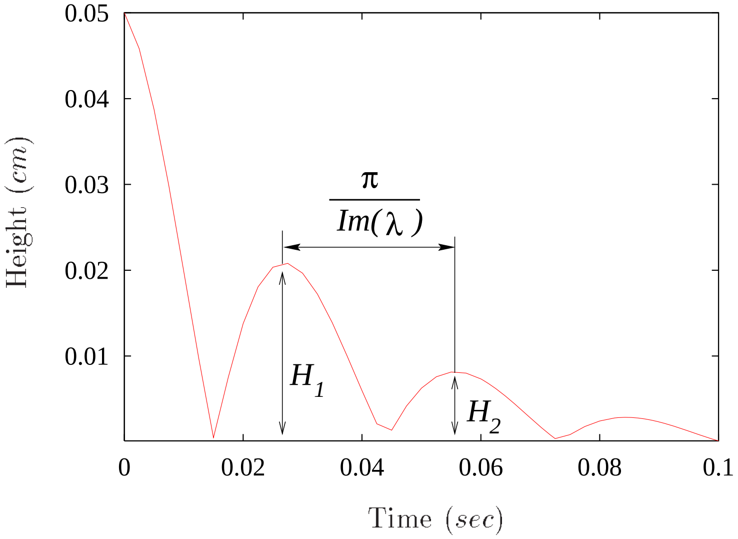


Figure 3.11: Plot of computed maximum fibre height versus time.

- the frequency $Im(\lambda)$, which can be calculated from the period of the fibre oscillations (which is equal to $\pi/Im(\lambda)$ on the diagram).

The results are summarised in Table 3.2 for a the spatial resolution $N = 64$ and $N_b = 192$ (so that $h_b \lesssim \frac{1}{2}h$) and values of the fibre stress parameter $\sigma \in \{1, 20, 100, 1000, 10000, 100000\}$.[d] The "analytical" values are found by taking the root of the dispersion relation whose decay rate $Re(\lambda)$ is smallest in magnitude; the frequency of this dominant mode is then given by $Im(\lambda)$. The precision of the "computed" results is limited to only two significant digits because of the size of the time step. The computed frequency shows extremely good agreement with the analytical results, and the decay rate likewise matches quite well.

	Smallest Decay Rate $Re(\lambda)$		Frequency $Im(\lambda)$	
σ	Analytical	Computed	Analytical	Computed
1	–1.6	–1.5	0	0
20	–26	–24	28	30
100	–33	–32	86	85
1,000	–51	–46	310	310
10,000	–84	–75	1039	1030
100,000	–142	–131	3390	3360

Table 3.2: Analytical and computed values of λ for the solution mode with the smallest wavenumber, $\alpha = 2\pi$ ($N = 64$, $N_b = 192$).

The correspondence between analytical and computed results seems reasonably close, with the relative difference being within 10% for all values of σ. To measure the effect of the spatial discretisation on the solution, we have computed the flat fibre problem on successively finer grids, choosing h as small as $\frac{1}{256}$. Table 3.3 lists a series of computations for $\sigma = 100,000$, at which the

[d] It might seem odd to choose the second stress value to be 20 instead of 10 in this sequence of σ. The reason for this choice is that there are bifurcations of the roots of the dispersion relation near the value 10 (actually at $\sigma \approx 15, 13$ and 8). To avoid problems with the numerical root finding routine, we will avoid the bifurcation points and choose $\sigma = 20$ instead.

largest discrepancy between predicted and computed decay rates occurred in Table 3.2. The error, e_N, is defined to be the magnitude of the difference between the decay rate predicted from the jump dispersion relation (which can be thought of as the "exact" result) and the computed decay rate. The difference between the "jump" and "computed" results decreases with N, and while the convergence rate, $\log_2(e_N/e_{2N})$, does not settle down to any clearly–defined value, it does appear to be reasonably close to the value 1.0 consistent with first order spatial accuracy.[e] The decay rates thus provide a measure of the convergence rate of the numerical scheme to the solution of the original delta function (or jump) problem.

N	$Re(\lambda)$	$Im(\lambda)$	Error	Rate
16	–73	2960	69	—
32	–100	3260	42	0.7
64	–131	3360	11	1.9
128	–147	3370	5	1.1
256	–140	3370	2	1.3

Table 3.3: Convergence of λ to the analytical value ($\lambda \approx -142 + 3390\,\imath$) as the computational grid is refined. The max norm errors and convergence rates are based on the comparison between $Re(\lambda)$ and the predicted decay rate of –142 ($k = 2.5 \times 10^{-6}$, $\sigma = 100,000$). First order convergence is indicated by a convergence rate of 1.0.

Remark 3.1. *By cutting off the fluid domain and imposing periodic boundary conditions at $y = 0$ and 1, we have introduced some error between the computed and analytical results. We can identify the significance of this error by increasing the extent of the fluid domain in the y–direction. By computing on domains of size 1×2, 1×4 and 1×8, we found that the results were practically identical, with the values of λ varying only by a few percent. This gives credence to our earlier assumption on page 31 that it is appropriate to concentrate on the region near the fibre and that*

[e] Computational evidence in the literature (in [LL97], for example) suggests that the Immersed Boundary Method is first order accurate in space.

our use of periodic boundary conditions has minimal effect on the solution, even for the dominant (or lowest wavenumber) modes.

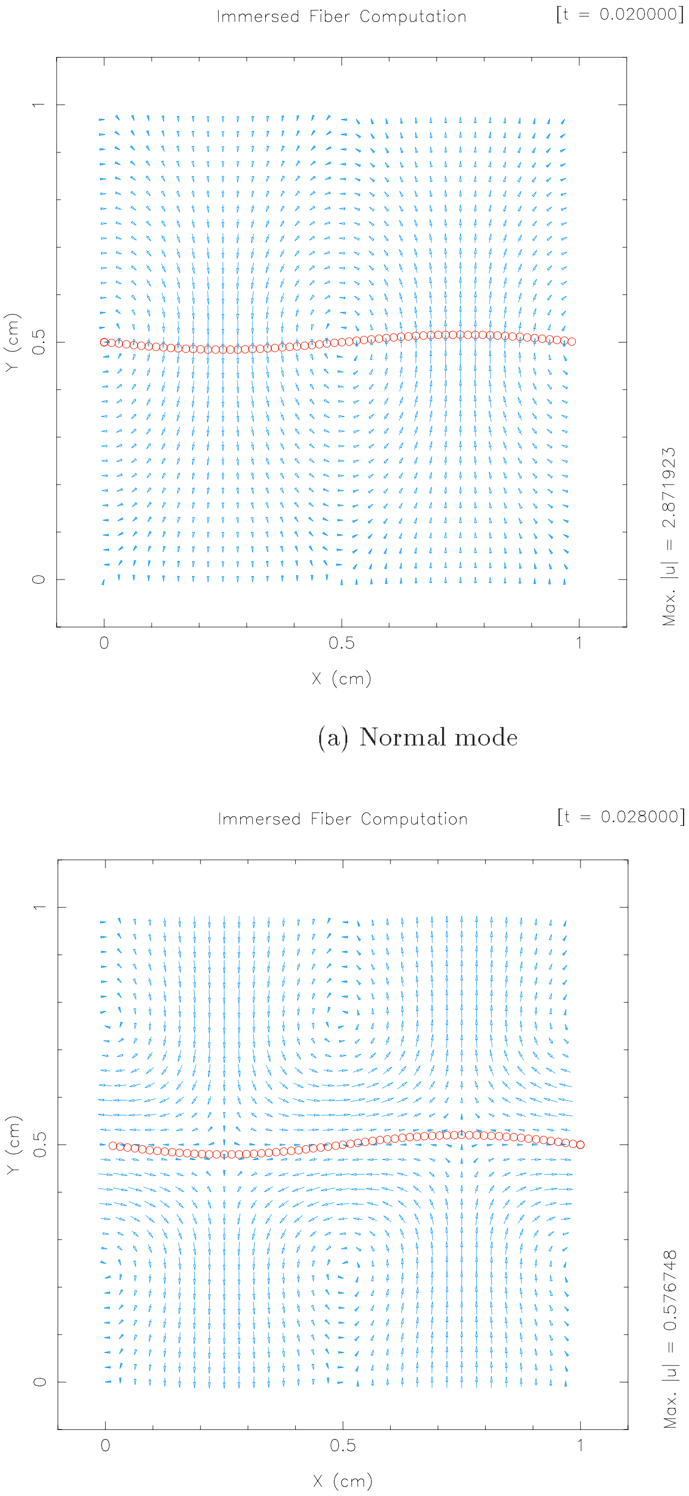
Let us return once more to the distinction made earlier between the normal and tangential motions of the fibre. The linear analysis clearly shows a decoupling between the normal and tangential modes of oscillation. Though we can't expect this to hold exactly in the fully non-linear problem, we can still expect a reasonable agreement with the theory when the displacement and velocity of the fibre are small. In fact, computations based on our "flat fibre" test problem show this decoupling quite clearly. Figure 3.12 contains velocity vector plots which demonstrate that both the normal and tangential motions are present at various points in the cycle of oscillations of the fibre. These pictures should be compared to the analytical solution plots from Figures 3.3 and 3.4. The normal mode of oscillation shown in Figure 3.12(a) dominates the flow for most times, except near instants when the maximum amplitude is reached (in Figure 3.12(b)) and the tangential mode expresses itself. We have also provided a plot of the transitional phase in Figure 3.13, where the flow is a combination of the two modes.

The one thing we have not mentioned is the smoothing effects arising from replacing the delta functions (or jumps) in the exact problem with approximate delta function forces. This is the subject of the following section.

3.2 Linear Stability of the Smoothed Problem

The results of the preceding section are based on the jump formulation of the immersed fibre problem, and so neglect the influence of smoothing from approximating the delta function. From the point of view of formal spatial accuracy, this is not an issue as we can simply let the resolution become arbitrarily fine by letting $N \to \infty$, and the solution will converge. However, this is not practical in computations, and when we select a spatial mesh of dimensions $N \times N$, the stiffness characteristics of the numerical method should be quite different from the original jump problem.

In this section we extend our previous results to an approximation of the delta function



(a) Normal mode

(b) Tangential mode

Figure 3.12: The normal (a) and tangential (b) modes of oscillation at two instants of time.

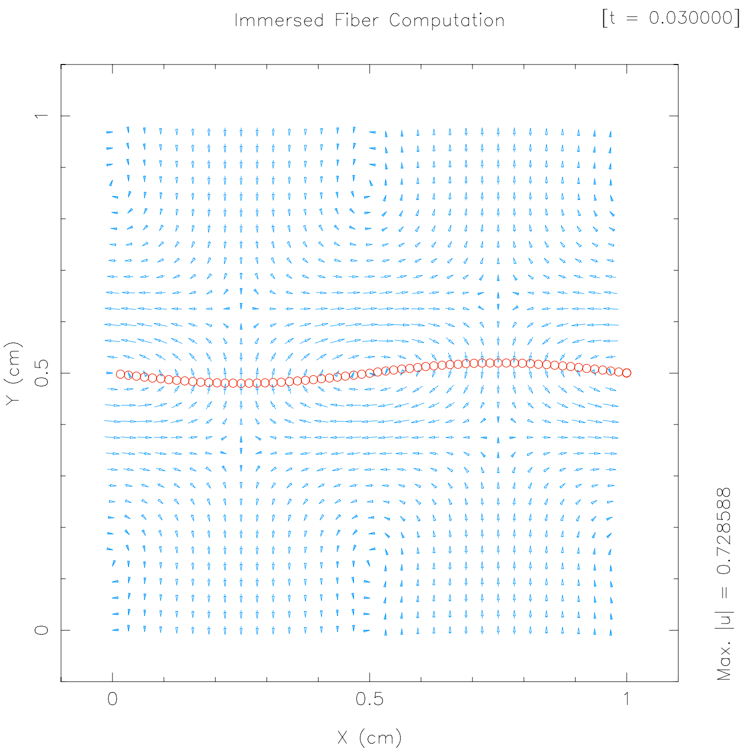


Figure 3.13: The transitional phase between the normal and tangential modes of oscillation.

formulation of the problem, where the force is smoothed over a finite interval. Since the choice of d_h is so important to the Immersed Boundary Method, we will be able to make even more accurate predictions regarding the stiffness and decay characteristics of immersed fibres. We expect that the behaviour of the solution to the discrete problem should be much closer to what is predicted by this smoothed version of the problem.

We use an approach very similar to that in Section 3.1 with the major modification being to introduce a strip — called the *smoothing region* — of width ϵ on either side of the fibre, where ϵ represents the radius of support of the approximate delta function. The region of interest is now divided into three subregions, $\Omega_0^{\pm}$ and Ω_0^{ϵ}, as pictured in Figure 3.14.

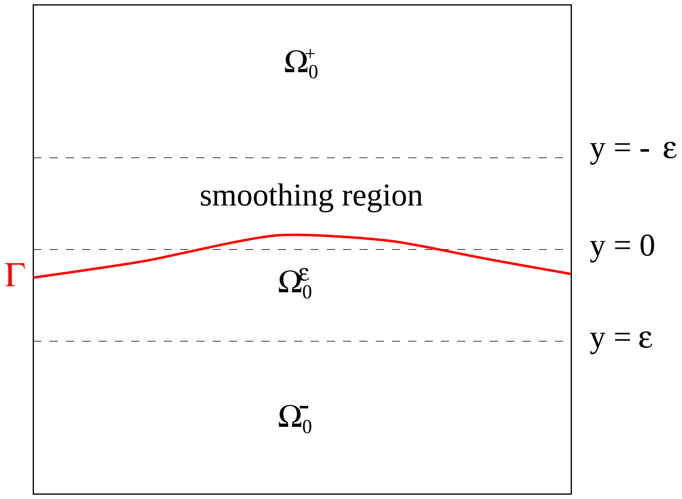


Figure 3.14: The fibre at equilibrium along $y = 0$, with a smoothing region, Ω_0^ϵ, of width 2ϵ.

3.2.1 Derivation of the dispersion relation

The form of the linearised force density function $\boldsymbol{f}$ is identical to that for the jump problem given in (3.5). Furthermore, on the subdomains Ω_0^+ and Ω_0^- the fibre force is zero, and hence the fluid obeys the same linearised equations as in the previous section. Consequently, we can write the solution on these regions immediately as

$$\widehat{p}^{\pm}(y) = A^{\pm}e^{\mp\alpha y}, \tag{3.26}$$

$$\widehat{u}^{\pm}(y) = B^{\pm}e^{\mp\beta y} - \frac{\hat{\imath}\alpha}{\rho\lambda}A^{\pm}e^{\mp\alpha y}, \tag{3.27}$$

$$\widehat{v}^{\pm}(y) = C^{\pm}e^{\mp\beta y} \pm \frac{\alpha}{\rho\lambda}A^{\pm}e^{\mp\alpha y}, \tag{3.28}$$

which are identical to (3.15)–(3.17). However, rather than linking these two solutions with jump conditions at the fibre ($y = 0$), we must first find the solution on Ω_0^ϵ and then perform matching at the boundaries $y = \pm\epsilon$ of the smoothing region.

On Ω_0^ϵ, the linearised fluid and fibre obey Stokes' equations with a *non-zero forcing function*

$$\rho\frac{\partial \boldsymbol{u}}{\partial t} = \mu\Delta\boldsymbol{u} - \nabla p + \int_\Gamma \boldsymbol{f}(s,t)\cdot d_\epsilon(\boldsymbol{x} - \boldsymbol{X}(s,t))\,ds, \tag{3.29}$$

$$\nabla\cdot\boldsymbol{u} = 0, \tag{3.30}$$

and the fibre evolution equation

$$\frac{\partial \boldsymbol{X}}{\partial t} = \int_{\Omega_0^\epsilon} \boldsymbol{u}(\boldsymbol{x},t)\cdot d_\epsilon(\boldsymbol{x} - \boldsymbol{X}(s,t))\,d\boldsymbol{x}. \tag{3.31}$$

The exact delta function $\delta(\boldsymbol{x})$ in (2.9a) and (2.9c) has been replaced by the smoothed version $d_\epsilon(\boldsymbol{x}) = d_\epsilon(x)\cdot d_\epsilon(y)$, which for the time being we choose to be the cosine approximation

$$d_\epsilon^c(x) = \begin{cases} \dfrac{1}{2\epsilon}\left(1+\cos\dfrac{\pi x}{\epsilon}\right), & \text{if } |x|\le\epsilon, \\ 0, & \text{if } |x|>\epsilon. \end{cases} \tag{3.32}$$

We now proceed to linearise the additional integral terms in the equations of motion, beginning with the forcing term on the right hand side of (3.29). After substituting the expressions (3.1) and (3.5) for the fibre position and force density, the fluid force may be written as

$$\begin{aligned} \boldsymbol{F}(x,y,t) &\approx \int_\Gamma \left(-\sigma_t\frac{\partial^2\xi}{\partial s^2}, -\sigma_n\frac{\partial^2\eta}{\partial s^2}\right) d_\epsilon\left(x - \theta s - \xi(s,t)\right) d_\epsilon\left(y - \eta(s,t)\right) ds, \\ &\approx -\int_{(x-\epsilon)/\theta}^{(x+\epsilon)/\theta} \left(\sigma_t\frac{\partial^2\xi}{\partial s^2}, \sigma_n\frac{\partial^2\eta}{\partial s^2}\right) d_\epsilon(x-\theta s)\, d_\epsilon(y)\, ds, \end{aligned}$$

since the ξ and η terms inside the approximate delta functions contribute only to higher order terms. We then substitute the separable form for the fibre position from (3.10) to get

$$\begin{aligned}
\boldsymbol{F}(x,y,t) &\approx -\int_{(x-\epsilon)/\theta}^{(x+\epsilon)/\theta} e^{\lambda t+\hat{\imath}\alpha\theta s}\left(\sigma_t\alpha^2\widehat{\xi},\,\sigma_n\alpha^2\widehat{\eta}\right)\,d_\epsilon(x-\theta s)\,d_\epsilon(y)\,ds,\\
&= -\alpha^2 e^{\lambda t}\,d_\epsilon(y)\,\left(\sigma_t\widehat{\xi},\,\sigma_n\widehat{\eta}\right)\int_{-\epsilon}^{\epsilon}\theta\,e^{\hat{\imath}\alpha(x-r)}d_\epsilon(r)\,dr,\\
&= -\theta\alpha^2 e^{\lambda t+\hat{\imath}\alpha x}\,d_\epsilon(y)\,\left(\sigma_t\widehat{\xi},\,\sigma_n\widehat{\eta}\right)\int_{-\epsilon}^{\epsilon}e^{-\hat{\imath}\alpha r}d_\epsilon(r)\,dr,
\end{aligned}\tag{3.33}$$

where we have performed the change of variables $r = x - \theta s$. The right hand side of the fibre evolution equation is linearised in a similar manner

$$\begin{aligned}
\frac{\partial\boldsymbol{\mathcal{X}}}{\partial t} &\approx \int_{\Omega_0^\epsilon}\left(\widehat{u}(y),\,\widehat{v}(y)\right)\,e^{\lambda t+\hat{\imath}\alpha x}\,d_\epsilon(x-\theta s)\,d_\epsilon(y)\,dy\,dx\\
&\approx e^{\lambda t}\int_{\theta s-\epsilon}^{\theta s+\epsilon}\int_{-\epsilon}^{\epsilon}\left(\widehat{u}(y),\,\widehat{v}(y)\right)\,e^{\hat{\imath}\alpha x}\,d_\epsilon(x-\theta s)\,d_\epsilon(y)\,dy\,dx\\
&= e^{\lambda t}\int_{\theta s-\epsilon}^{\theta s+\epsilon}e^{\hat{\imath}\alpha x}\,d_\epsilon(x-\theta s)\,dx\int_{-\epsilon}^{\epsilon}\left(\widehat{u}(y),\,\widehat{v}(y)\right)\,d_\epsilon(y)\,dy\\
&= e^{\lambda t+\hat{\imath}\alpha\theta s}\int_{-\epsilon}^{\epsilon}e^{\hat{\imath}\alpha r}\,d_\epsilon(r)\,dr\int_{-\epsilon}^{\epsilon}\left(\widehat{u}(y),\,\widehat{v}(y)\right)\,d_\epsilon(y)\,dy.
\end{aligned}\tag{3.34}$$

Notice the presence in both (3.33) and (3.34) of the Fourier transform of the approximate delta function which we define as

$$\widehat{D}_\alpha^\epsilon := \int_{-\epsilon}^{\epsilon}e^{\pm\hat{\imath}\alpha r}d_\epsilon(r)\,dr.\tag{3.35}$$

For the cosine delta function, this transform evaluates to

$$\widehat{D}_\alpha^\epsilon = \frac{\pi^2\sin\left(\alpha\epsilon\right)}{\alpha\epsilon\left(\pi^2-\alpha^2\epsilon^2\right)}.$$

The "±" forms of the transform are equivalent because of the symmetry properties of $d_\epsilon(r)$. Using this definition, we can write the system of integro–differential equations on the strip Ω_0^ϵ as follows:

$$\rho\lambda\widehat{u}^\epsilon = \mu\left(\frac{d^2}{dy^2} - \alpha^2\right)\widehat{u}^\epsilon - \hat{\imath}\alpha\widehat{p}^\epsilon - \sigma_t\theta\alpha^2\widehat{D}_\alpha^\epsilon\widehat{\xi}\,d_\epsilon(y), \tag{3.36}$$

$$\rho\lambda\widehat{v}^\epsilon = \mu\left(\frac{d^2}{dy^2} - \alpha^2\right)\widehat{v}^\epsilon - \frac{d\widehat{p}^\epsilon}{dy} - \sigma_n\theta\alpha^2\widehat{D}_\alpha^\epsilon\widehat{\eta}\,d_\epsilon(y), \tag{3.37}$$

$$\hat{\imath}\alpha\widehat{u}^\epsilon + \frac{d\widehat{v}^\epsilon}{dy} = 0, \tag{3.38}$$

$$\lambda\widehat{\xi} = \widehat{D}_\alpha^\epsilon\int_{-\epsilon}^{\epsilon}\widehat{u}^\epsilon(y)\,d_\epsilon(y)\,dy, \tag{3.39}$$

$$\lambda\widehat{\eta} = \widehat{D}_\alpha^\epsilon\int_{-\epsilon}^{\epsilon}\widehat{v}^\epsilon(y)\,d_\epsilon(y)\,dy. \tag{3.40}$$

At first glance, it would appear that this coupled system of equations is difficult to solve analytically, since the fibre positions are integrals of the velocity components, while $\widehat{u}^\epsilon$ and $\widehat{v}^\epsilon$ are in turn found by solving a differential equation with $\widehat{\xi}$ and $\widehat{\eta}$ on the right hand side. Fortunately, $\widehat{\xi}$ and $\widehat{\eta}$ are *constants* and therefore (3.36)–(3.38) may be solved for the velocity and pressure first, without knowing the fibre positions *a priori*. The resulting $\widehat{u}^\epsilon$ and $\widehat{v}^\epsilon$ can then be used in (3.39) and (3.40) to find expressions for $\widehat{\xi}$ and $\widehat{\eta}$, which are substituted back into the velocity solutions. This procedure involves extensive algebraic manipulations, and is tractable only through the use of MAPLE.[f] The final expressions for $\widehat{u}^\epsilon$, $\widehat{v}^\epsilon$, $\widehat{p}^\epsilon$, $\widehat{\xi}$ and $\widehat{\eta}$ are so large that they are not presented here.

At this point, we have expressions for the solutions on three regions, each involving several unknown constants of integration. On $\Omega_0^\pm$ we have (3.26)–(3.28), which involve the six coefficients $A^\pm$, $B^\pm$ and $C^\pm$. The solution on the strip Ω_0^ϵ introduces an additional six constants of integration: two from the pressure Poisson equation, and another four from the solution of the velocity equations. Consequently, we must come up with a total of 12 equations in order to uniquely determine the values of the 12 constants. We proceed as we did for the jump formulation of the problem and apply the incompressibility condition along with matching conditions at the interfaces $y = \pm\epsilon$:

[f] MAPLE requires approximately two hours of computing time on a HP Apollo 9000/735 (99 MHz PA–RISC 7100) with 80 Megabytes of RAM to solve the integro–differential equations and compute the determinant, using the cosine approximation to the delta function.

- enforcing the incompressibility condition leads to one equation on each of $\Omega_0^\pm$, and two on Ω_0^ϵ (one each for the real and imaginary parts of the equation), for a total of four equations;
- another four conditions arise from the requirement that the pressure, velocity and normal derivative, $d\widehat{u}/dy$, be continuous at the interface $y=\epsilon$;
- the final four matching conditions come from enforcing continuity at $y=-\epsilon$.

The resulting system of equations is homogeneous, and so there is a non-trivial solution only if the determinant is zero. The dispersion relation is too large to include here, but we can write it symbolically as

$$S_n^\epsilon(\beta)\cdot S_t^\epsilon(\beta)=0 \tag{3.41}$$

where

- $S_t^\epsilon(\beta)$ and $S_n^\epsilon(\beta)$ are functions of the parameters α, μ and ρ in addition to β. The parameter σ_n appears only in $S_n^\epsilon(\beta)$ and σ_t only in $S_t^\epsilon(\beta)$.
- The structure of the dispersion relation is very similar to that from the jump formulation (3.22) in that there is a decoupling between the normal and tangential fibre modes.
- On the other hand, the dispersion relation is no longer a polynomial (since it now involves trigonometric and exponential functions of the parameters) and we have been unable to generate an analytical expression for the solutions β. Consequently, our only recourse is to apply a numerical root–finding technique such as Newton's method (modified for complex functions). This is facilitated by Maple's `C()` and `fortran()` functions for generating code from the analytical expression for the determinant.
- We have also found that the presence of the exponential terms in (3.41) make the equation very ill–conditioned, and requires the use of quadruple precision arithmetic in the Newton solver. Even then, very delicate calculations are necessary to find some, let alone all of the roots of the dispersion relation. As a result, we have been unable to reproduce contour plots

of the exponential time constants (for ranges of the parameter values) as we did for the jump problem.

3.2.2 Accuracy and the choice of approximate delta function

Before discussing the stability of the smoothed delta function problem in terms of solutions to the dispersion relation (3.41), we would first like to verify that this expression is compatible with our results from the jump formulation. A simple check for consistency is to see that the results for the smoothed problem match with those from the jump problem in the limit as ϵ approaches zero. This corresponds to the physically reasonable assumption that as the smoothing radius shrinks to zero, the matching conditions across the smoothed region Ω_0^ϵ should reduce to jump conditions.

To perform this comparison, we expand $\widehat{D}_\alpha^\epsilon$ and the exponential terms appearing in (3.41) as Taylor series in ϵ. Omitting the details, we find that the factors $S_n^\epsilon(\beta)$ and $S_t^\epsilon(\beta)$ are series in ϵ whose two leading order coefficients can be written in terms of $S_n(\beta)$ and $S_t(\beta)$ (the factors of the *jump dispersion relation*):

$$S_n^\epsilon(\beta) = S_n(\beta) + \epsilon\, S_n(\beta) + \mathcal{O}(\epsilon^2), \tag{3.42a}$$

$$S_t^\epsilon(\beta) = S_t(\beta) + \epsilon\, S_t(\beta) - \kappa\epsilon \left(\frac{\rho\sigma_t\alpha^2}{2\mu^2}\right) + \mathcal{O}(\epsilon^2), \tag{3.42b}$$

where $\kappa = \frac{1}{6} - \frac{5}{8\pi^2}$. It is clear that any root β that satisfies either of the jump dispersion relations $S_n(\beta) = 0$ or $S_t(\beta) = 0$ must also satisfy the corresponding smoothed version of the formula above *to first order in* ϵ. Notice also that the normal modes match to second order in ϵ, while the tangential factor (which incidentally corresponds to the stiffest modes in typical computations) only matches to first order. We can thus conclude for the fibre modes, that the use of a smoothed approximation to the delta function leads to a solution which is consistent with the jumps in pressure and normal derivative of velocity in the exact delta function formulation of the problem.

At this point it is worthwhile mentioning how these results apply to a discrete version of the immersed fibre problem. One can think of the smoothing radius ϵ in terms of an *idealised discretisation* which holds when the grid spacing $h \ll \epsilon$. The *accuracy* with which the dispersion relation for the smoothed problem approximates the modes from the jump problem is embodied in the asymptotic formulas (3.42a) and (3.42b). We use the term "idealised" since in practice ϵ is typically never chosen larger than $2h$.

There is nothing in the derivation above that restricts us to the use of any particular choice of d_ϵ (except, perhaps, limitations on the size of expressions that MAPLE can handle). Thinking back to the discussion in Section 2.3.1 of moment conditions, it may prove enlightening from a theoretical standpoint to compare the asymptotic results in ϵ for various choices of delta function approximation. The results will of course not be directly applicable to the discrete problem, since they hold only in the limiting case $h \ll \epsilon$; however, it will still prove useful to draw some comparisons to the motivation of delta function choice using discrete compatibility conditions.

We have derived a series of dispersion relations for various polynomial (and piecewise polynomial) approximations $d_\epsilon(r)$ that satisfy one or more of the following continuous analogues of the discrete moment conditions:

$$\begin{aligned}
\text{Zeroth moment:} \quad & \int_{-\epsilon}^{\epsilon} d_\epsilon(r)\,dr = 1, \\
\text{First moment:} \quad & \int_{-\epsilon}^{\epsilon} r\,d_\epsilon(r)\,dr = 0, \\
\text{Second moment:} \quad & \int_{-\epsilon}^{\epsilon} r^2\,d_\epsilon(r)\,dr = 0.
\end{aligned}$$

We consider the following approximations, which are summarised in Table 3.4 along with the continuous moment conditions that they satisfy:

- a piecewise constant function, d_ϵ^0, which is the simplest that satisfies the zeroth order moment condition.
- the cosine approximation, d_ϵ^c, currently the most commonly used function in immersed boundary computations, which satisfies the zeroth and first order continuous moment conditions (as well as the discrete compatibility conditions described in Section 2.3.1).
- a piecewise linear approximation, d_ϵ^1, which was the original delta function used by Pe-

skin [Pes72].

- a quadratic polynomial, d_ϵ^2, which satisfies the zeroth and first order moment conditions. This approximation is motivated by the work of Beyer & LeVeque [BL92], who derived a piecewise quadratic delta function approximation that increased the spatial accuracy of a one–dimensional version of the scheme to second order.[g]
- finally, a sixth degree polynomial, d_ϵ^6, which is the simplest smooth function we have found that satisfies in addition the second order moment condition given above. The inclusion of the higher order moment was motivated in Section 2.3.1. Even though we can show that the delta function based on $\phi_{new}(r)$ from equation (2.18) satisfies the first three *continuous* moment conditions as well as their discrete analogues, we have chosen to use this simpler polynomial function, here in the continuous setting, so that there is some hope of deriving the dispersion relation.

The approximations d_ϵ^0, d_ϵ^2 and d_ϵ^6 have not appeared in the literature in reference to the Immersed Boundary Method. This is not surprising, since these delta functions do not satisfy the discrete moment conditions which computational evidence suggest are required in the method. Nevertheless, they do satisfy the corresponding continuous moment conditions and have a simple enough functional form that the dispersion relations can realistically be derived. Hence, it will prove useful for us to consider all of the above functions so that we can compare the *formal accuracy* of the various approximations.

For each of the approximations listed in Table 3.4 (except for d_ϵ^6), the dispersion relation can be derived and written in the same form (3.41)–(3.42) we had for the cosine approximation. The sixth degree polynomial leads to a determinant that is too large for MAPLE to compute (see Remark 3.2 below). The last column contains the "order constant" κ from the asymptotic expansions of the

[g] To date, this one–dimensional convergence result has not been extended to higher dimensions. Furthermore, all the computational evidence that has appeared in the literature to date suggests that the delta function interpolation limits the Immersed Boundary Method to first order spatial accuracy in two or three dimensions.

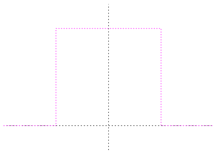
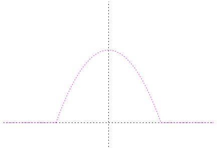
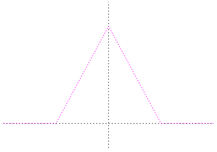
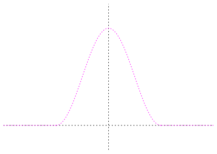
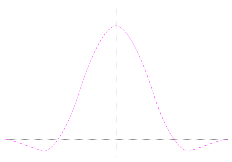
Delta function approximation	Continuous moments	$\mathcal{O}(\epsilon)$ constant κ
 d_ϵ^0 (piecewise constant)	0	$-\frac{1}{6} \approx -0.167$
 d_ϵ^2 (quadratic)	0, 1	$\frac{9}{70} \approx 0.129$
 d_ϵ^1 (piecewise linear)	0, 1	$\frac{7}{60} \approx 0.117$
 d_ϵ^c (cosine)	0, 1	$\frac{1}{6} - \frac{5}{8\pi^2} \approx 0.103$
 d_ϵ^6 (sixth degree polynomial)	0, 1, 2	see Remark 3.2

Table 3.4: Comparison of the constants κ in the $\mathcal{O}(\epsilon)$ term of the dispersion relation for various approximate delta functions (in order of decreasing κ).

dispersion relation for each of the approximate delta functions. κ may be considered as a measure of how accurately the scheme for a given delta function will approximate the original problem. The results in the table seem to indicate that satisfying the higher moment conditions leads to an improvement in the formal accuracy of the scheme. Furthermore, the cosine approximation leads to the best asymptotic results for all of the approximations that we were able to compute results for.

Remark 3.2. *Our inability to push through the dispersion relation calculation for d_ϵ^6 stems from a limitation in* MAPLE *on the number of terms in expressions. According to the technical staff at* Waterloo Maple Inc.*, versions of the program for machines that have 16– or 32–bit addressing (which are the only machines that we can run* MAPLE *on at the moment) have a limit of 65,535 terms. However, on 64–bit address machines, the limit goes up to 2×10^9 terms. We are currently attempting to obtain an 64–bit binary version of* MAPLE*, which we expect will allow us to compute the dispersion relation for the sixth order polynomial in order to show that $\kappa = 0$.*

Unfortunately, it is the delta function d_ϵ^6 that satisfies the second moment condition for which we have been unable to derive a dispersion relation. We expect that this approximation will have $\kappa = 0$, which corresponds to a smoothing that is formally second order accurate. In the absence of any further analytical evidence to support this claim, we perform a computational investigation of the delta function d_h^{new} (introduced in Section 2.3.1) that satisfies the *discrete second moment condition*. We follow the approach of LeVeque & Li [LL97] who considered an elliptical–shaped interface, such as that pictured in Figure 4.9 with semi–axes 0.4 cm and 0.2 cm, which is immersed in a 1 $cm \times 1$ cm periodic box. We compute the solution on a sequence of successively finer grids with $\{N, N_b\} = \{32, 96\}$, $\{64, 192\}$, $\{128, 384\}$, $\{256, 768\}$, $\{512, 1536\}$, and $\{1024, 3072\}$, using parameters $\mu = \rho = 1$, $\sigma = 1,000$ and take 20 time steps of size $k = 1.0 \times 10^{-5}$ using the Immersed Boundary Method described in Section 2.3. We modify LeVeque & Li's error measure slightly, and compute convergence rates based on an $L2$–norm difference, e_{N_b}, between interface positions on successive grids

$$e_{N_b} = \sqrt{\frac{1}{N_b} \sum_{\ell=0}^{N_b-1} \left[\left(X_\ell^{N_b} - X_{2\ell}^{N_b/2} \right)^2 + \left(Y_\ell^{N_b} - Y_{2\ell}^{N_b/2} \right)^2 \right]} \,,$$

where $(X_\ell^{N_b}, Y_\ell^{N_b})$ is the interface position at the end of the computation, using N_b boundary points. The convergence rate can then be estimated using the formula

$$\textit{Convergence rate} \approx \log_2 \left(\frac{e_{N_b}}{e_{2N_b}} \right).$$

The errors and computed rates for the two delta functions are listed in Table 3.5, from which it is clear that our "new" delta function performs better than the cosine approximation. It is not clear that the new approximation leads to second order spatial convergence, and in the absence of further analytical backing we can only say that there is evidence to suggest that this might be true.

N, N_b	d_h^c (cosine)		d_h^{new} (second moment)	
	e_{N_b}	Conv. rate	e_{N_b}	Conv. rate
32,96	—	—	—	—
64,192	7.19×10^{-4}	—	9.45×10^{-4}	—
128,384	8.52×10^{-4}	–0.24	8.43×10^{-4}	0.16
256,768	6.71×10^{-4}	0.34	4.68×10^{-4}	0.85
512,1536	3.86×10^{-4}	0.80	1.85×10^{-4}	1.34
1024,3072	1.72×10^{-4}	1.17	5.60×10^{-5}	1.72

Table 3.5: Comparison of convergence rates for the cosine and "new" delta functions.

We can expect the calculation of d_h^{new} to be considerably more expensive than d_h^{cos}, due to the complexity of the expression in (2.18) and more significantly, the wider stencil of points. To see whether the extra cost is worthwhile, we have listed in Table 3.6 the CPU time required for the two delta functions used in the previous calculation. The use of the d_h^{new} clearly increases the cost of the interpolation routines by up to 200%. However, when compared to the *total cost* of

N, N_b	CPU time required		%
	d_h^c	d_h^{new}	increase
32,96	0.18 (0.55)	0.53 (0.88)	190 (60)
64,192	0.33 (1.85)	1.03 (2.52)	210 (36)
128,384	0.78 (7.68)	2.12 (9.10)	170 (18)
256,768	2.11 (36.5)	4.85 (39.2)	130 (7)
512,1536	6.12 (159.0)	11.5 (164.3)	90 (3)
1024,3072	18.9 (1047)	29.9 (1058)	60 (1)

Table 3.6: Comparison of CPU times required by the delta function interpolation routines using the two approximations, with the total CPU time for the entire calculation given in parentheses. Timings were performed on an SGI Origin 2000 (4 × 195 MHz R10000 processors, 512 Mb RAM).[i]

the immersed boundary computation (the values given in parentheses), this increase is much less significant. For grids of size 64×64, which are typically used in computations, the total CPU time increases by only 36%, with the percentage decreasing as the grid is refined. We have made very little effort to optimise the calculation of d_h^{new}, and so it is our opinion that with some additional work, the cost can be reduced even further. Consequently, we believe that this new delta function may be a useful improvement to immersed boundary computations. Clearly more study is required, and the utility of the new interpolation scheme will be proven in three–dimensional calculations.

3.2.3 Effects of smoothing on stiffness of solution modes

We now investigate two aspects of the smoothing process: the consistency of the smoothed problem with the original delta function or jump formulation; and more importantly, how approximations to the delta function affect the stiffness of the problem.

[i] One is reminded at this point of the immortal words of Bill Gates (1981): *"640 K ought to be enough for anybody."*

We begin by examining the the decay rate and frequency of oscillation for the lowest wavenumber mode, which is the dominant solution mode. We have solved the dispersion relation for various σ values and summarised the results in Table 3.7, with the "jump" results from Table 3.2 repeated for easy comparison. The effect of smoothing on the analytical solution modes is negligible for small σ. However it is clear that smoothing has a significant effect on the decay rates for values of σ in the physical range of 10,000–100,000. The frequency of the dominant solution mode, on the other hand, remains comparatively unaffected over the entire range of fibre stress coefficient.

σ	Smallest decay rate $Re(\lambda)$			Frequency $Im(\lambda)$		
	Analytical (jump)	Analytical (smooth)	Computed	Analytical (jump)	Analytical (smooth)	Computed
1	–1.6	–1.3	–1.5	0	0	0
20	–26	–26	–24	28	29	30
100	–33	–33	–32	86	85	85
1,000	–51	–50	–46	310	307	310
10,000	–84	–76	–75	1039	1025	1030
100,000	–142	–108	–131	3390	3321	3360

Table 3.7: A comparison of the decay rates and frequencies for the lowest wavenumber modes ($\alpha = 2\pi$) for the jump and smoothed delta function formulations, and the computed results ($N = 64$, $\epsilon = \frac{2}{64}$).

The "computed" results from Table 3.7 show reasonable agreement with the analytical results for both the jump and smoothed problems when $\sigma \leq 10,000$. However, there is a considerable difference when σ is taken any larger. If we refer back to the convergence results of Table 3.3, it is clear that the computed decay rates do differ from those of the original jump problem when the grid is not fine enough.

If we think of the smoothed problem as an *idealised discretisation* of the delta function formulation of the immersed fibre problem, then we should observe convergence of the smoothed solution to the jump solution as the smoothing radius goes to zero. As we saw earlier in Table 3.7, the *computed decay rate* converged to the analytical value as the computational grid was refined. We would expect the same behaviour from our smoothed delta–function problem: that is, the dominant decay rate for the smoothed problem should converge to that of the original jump problem as the smoothing radius goes to zero. This is consistent with our asymptotic matching of the dispersion relations from the jump and smoothed problems in Section 3.2.2. Table 3.8 lists the largest decay rate for various values of ϵ, from which is it evident that the results do correspond as ϵ is reduced (as before, the correspondence worsens as the fibre force parameter σ increases). This table provides us with a "consistency check" which verifies that the dominant mode in our idealised ϵ–discretisation matches the delta function solution.

	Smallest Decay Rate							
σ	Jumps	$\epsilon = \frac{1}{64}$	$\epsilon = \frac{2}{64}$	$\epsilon = \frac{3}{64}$	$\epsilon = \frac{4}{64}$	$\epsilon = \frac{5}{64}$	$\epsilon = \frac{6}{64}$	$\epsilon = \frac{7}{64}$
1	–1.6	–1.5	–1.3	–1.2	–1.1	–1.0	–0.9	–0.8
20	–26	–26	–26	–26	–26	–25	–25	–25
100	–33	–33	–33	–33	–32	–32	–31	–31
1,000	–51	–51	–50	–48	–46	–44	–42	–40
10,000	–84	–81	–76	–69	–62	–57	–52	–48
100,000	–142	–129	–108	–89	–75	–64	–57	–51

Table 3.8: A comparison of the analytical decay rates for the lowest wavenumber mode ($\alpha = 2\pi$) for varying smoothing radius ($N = 64$).

Nonetheless, our main interest is not the accuracy of the delta function approximation but rather how the smoothing of the delta function affects the stiffness of the problem. Up to this point, we have considered only the influence of smoothing on the *lowest wavenumber* solution modes, which appear as the dominant solution features (after the higher frequency components have been damped out). We will now consider the impact of smoothing on the small scale features of the solution by looking at the entire range of discrete wavenumbers $\alpha = 2\pi \cdot n$, for $n = 1, 2, \ldots, N$.

Plots of the decay rate $Re(\lambda)$ and frequency $Im(\lambda)$ for wavenumbers $2\pi \leq \alpha \leq 2\pi \cdot 21$ are given in Figure 3.15 for the following values of the parameters: $\sigma_n = \sigma_t = 1000$, $\mu = \rho = 1.0$ and $\epsilon = \frac{2}{64}$. The ill–conditioned nature of the smoothed dispersion relation limits the range of wavenumbers for which roots can be computed. However, by reducing the viscosity to $\mu = 0.01$, we were able to compute solutions over nearly the entire range of $\alpha \in [2\pi, 128\pi]$, which we present in Figure 3.16. The jump problem modes are cut off so that the detail of the smooth modes can be seen; however, if the vertical axis were extended, these modes would increase monotonically over the range of α, with $-5249 \leq Re(\lambda) \leq -4$ and $5080 \leq |Im(\lambda)| \leq 146,622$. It is clear from the plots, that the jump modes for $\alpha > 2\pi$ are not captured well at all in the smoothed problem. However, the good news for the method is that the solution modes are considerably less stiff. This can be interpreted as a "regularising effect" on the higher wave number modes. However, there is still a wide range of sizes in the exponential time constants, and hence a considerable degree of stiffness. It is important to note that it is the imaginary part of λ that is the dominant source of stiffness — this point will come up again in Section 4.1 in the context of choosing an appropriate

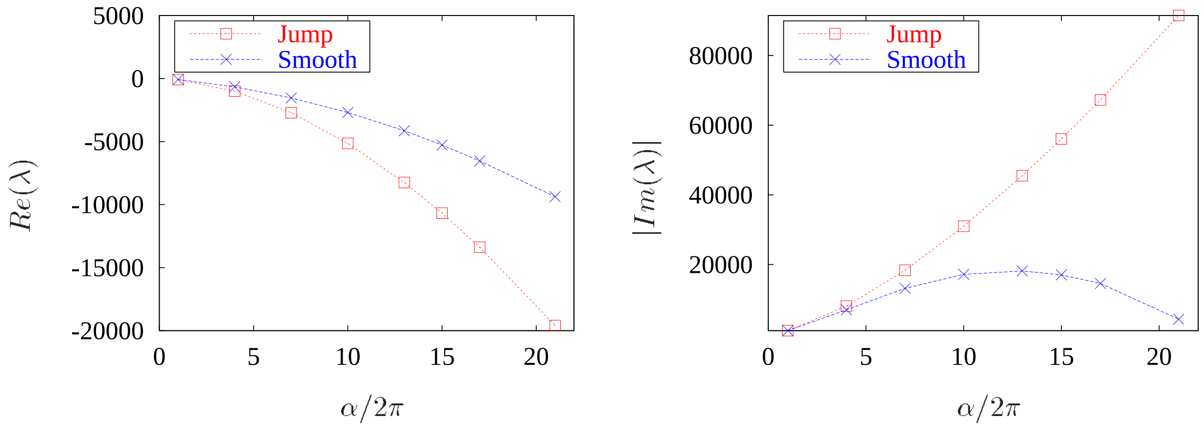


Figure 3.15: A comparison of $Re(\lambda)$ and $Im(\lambda)$ for the jump and smoothed dispersion relations with $\epsilon = \frac{2}{64}$. Here we take $\mu = \rho = 1.0$ and $\sigma_n = \sigma_t = 1000$, but restrict ourselves to $\alpha \leq 42\pi$.

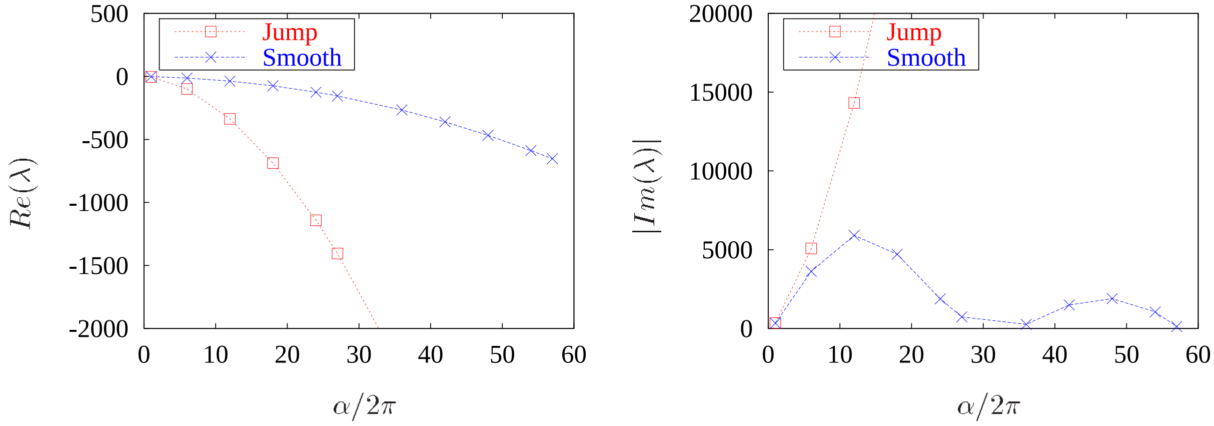


Figure 3.16: A comparison of $Re(\lambda)$ and $Im(\lambda)$ for the jump and smoothed dispersion relations with $\alpha \in [2\pi, 128\pi]$ and $\epsilon = \frac{2}{64}$. We have had choose $\mu = 0.01$, $\rho = 1$ and $\sigma_n = \sigma_t = 1000$ to obtain roots for the whole range of wavenumbers.

explicit time–stepping scheme for the Immersed Boundary Method.

The main conclusion we can draw from this discussion is that the smoothed problem, while still suffering from a significant degree of stiffness, is much less stiff than what is suggested by the dispersion relation from the jump problem. We will see in the next chapter, by comparing time step restrictions, that the smoothed analysis is much better at predicting the behaviour of the computed solution.

Chapter 4
Analysis of Time Discrete Schemes

> *I have tried to avoid long numerical computations, thereby following Riemann's postulate that proofs should be given through ideas and not voluminous computations.*
>
> — David Hilbert.

In this chapter, the analytical solution technique we have just described is extended to a semi–discrete formulation of the immersed fibre problem. We will discretise in time only and leave the fluid force and fibre velocity written in terms of smoothed delta functions, as we did in Section 3.2. By restricting the wavenumber α to a range of integers $[1, N]$ and selecting an appropriate smoothing radius ϵ, we will be able to make further conclusions about *idealised spatial discretisations* of the problem. Our purpose is to determine the convergence properties of iterative schemes that attempt to handle the force implicitly.

We will consider two time–stepping schemes that lead to a non–linear fixed point iteration on the fibre position within each time step. One scheme is a Crank–Nicholson type splitting in time (with diffusion coupled implicitly with the force, and explicit convection), and the other is a method proposed by Mayo & Peskin [MP93], which uses an ADI step for convection and diffusion and the force is coupled only with the interface position. We will see that for both of these schemes, the pressure and velocity can be eliminated from the semi–discrete equations and the method rewritten as an iteration on the fibre position alone. The behaviour of the two methods is discussed in terms of the stability and convergence of the iteration, which is then verified in computations.

However, before investigating implicit discretisations, we will begin with a more thorough examination of the linear modal results of the previous chapter and their application to time–stepping schemes where the force is handled *explicitly.* We will often refer to these schemes as "explicit," even though computations will sometimes include the ADI step in the usual Immersed Boundary Method algorithm. These results are of particular interest since many immersed boundary computations involve a non–linear force for which no effective implicit solvers are available at the current time.

4.1 Explicit Schemes

In Section 3.1.4 we found that for a typical choice of parameters, the linearised fibre modes have magnitudes that differ over a very wide range. This disparity is especially pronounced when the fibre force is large, and can be further compounded by small viscosity in high Reynolds number flows. The analytical results of Section 3.2.3 showed that smoothing the force with an approximate delta function reduces the stiffness considerably, and yet numerical evidence testifies that the method still suffers from severe time step restrictions. We will now investigate the stiff nature of solution modes and how it relates to the behaviour of various explicit time–stepping schemes. Initially, we consider a Forward Euler discretisation of the problem, which differs from the *FE/ADI* scheme (from Section 2.3) in that there is no ADI step and convection and diffusion are handled explicitly.

It is essential in the following discussion that we distinguish clearly between the solution modes arising from the fibre, and those from Stokes flow without an immersed boundary, since the time step in a discretisation of the immersed fibre problem is limited by both. One thing to keep in mind is that Stokes modes (which satisfy $\lambda = -\mu\alpha^2/\rho$) lie entirely on the real axis, while the fibre modes have an imaginary part that is typically large in comparison to the real part (as we saw in Section 3.1.4). To understand whether diffusive effects or the fibre forces are the limiting factor in computations, we begin by plotting the two sets of modes together in the λk–plane along with the region of absolute stability for the Forward Euler scheme. Figure 4.1 depicts the stability region, which is the interior of the circle $|1 + \lambda k| = 1$, along with the two sets of modes (smoothed fibre modes are marked "$\circ$" and Stokes modes "$*$"). We have used the same set of parameters from the example pictured in Figure 3.15: namely, $\sigma = 1,000$, $\mu = \rho = 1$ and smoothing radius $\epsilon = \frac{2}{64}$. It is clear that in this situation the Stokes modes determine the time step restriction. The maximum allowable k is determined by the circled point on the far left, which gives rise to the *stability condition* for an explicit Forward Euler discretisation of diffusion, using centered differences in space:

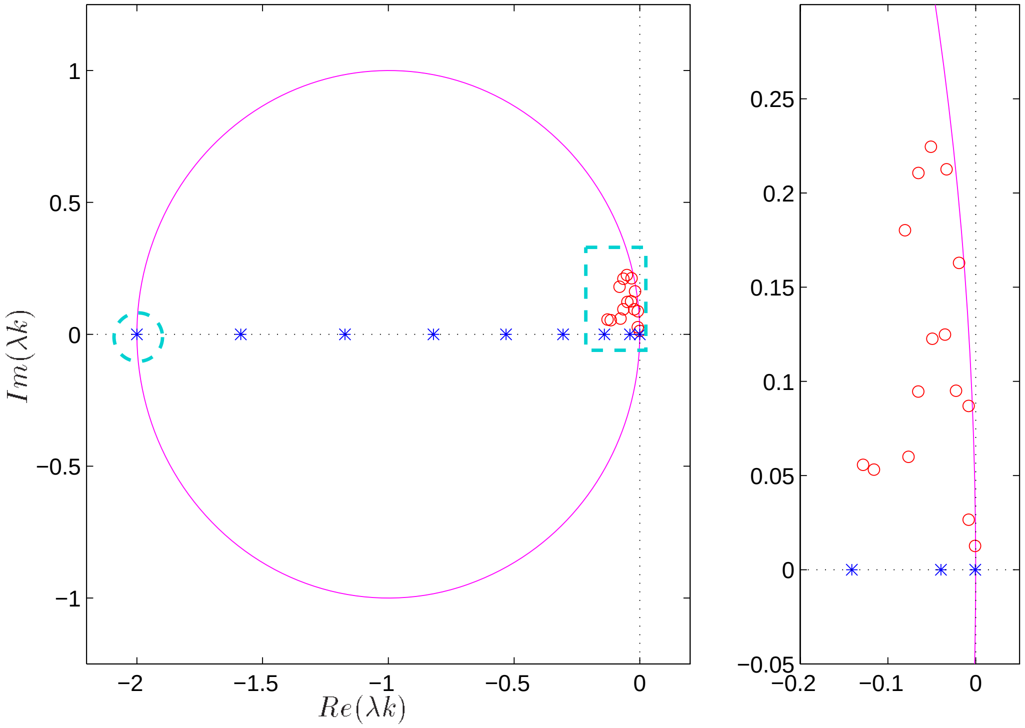


Figure 4.1: Region of absolute stability for the Forward Euler scheme, along with the smoothed fibre modes (plotted as $\circ$) and Stokes (with $*$). The plot on the right is a blow–up of the region containing the fibre modes. The time step is taken to be $k = 6.1 \times 10^{-5}$, which is the largest afforded by the method ($\sigma = 1,000$).

$$k \leq \frac{\rho h^2}{4\mu} \approx 6.1 \times 10^{-5}. \tag{4.1}$$

The *FE/ADI* scheme, on the other hand, avoids the limitation that arises from Stokes modes in the straight Forward Euler scheme by treating diffusion implicitly, but we would still expect this method to be limited by its explicit handling of the fibre force. The right hand plot in Figure 4.1 shows that the fibre modes are already very close to the stability limit, and so we can take the time step only 1.5 times larger than that allowed for explicit diffusion: $k_{max} \approx 6.6 \times 10^{-5}$. This is at odds with the actual maximum time step $k_{max} = 5 \times 10^{-4}$ that we observe in computations. It thus appears that the ADI step confers some additional advantage with regard to the stiffness of fibre modes. Recall that the fibre modes embody the interaction of the fibre force with a viscous fluid. Thus, handling even just the viscous terms implicitly can help deal with the stiffness of the fibre modes.

The forcing parameter $\sigma = 1,000$ is actually somewhat small from a physical standpoint, but we have been unable to solve the smooth dispersion relation (over the entire range of α) for values of σ any larger because of the ill-conditioned nature of the equations.[a] However, we do know from the previous chapter that as σ increases, the smoothed fibre modes increase in magnitude; consequently, we expect that if σ is taken large enough, then the stiffness from the fibre modes will become the limiting factor in the time step. Not only does the magnitude of λ increase as the force is strengthened, but so does the ratio $|Im(\lambda)/Re(\lambda)|$: we can see this effect in Figure 4.2 which depicts this ratio for the lowest wavenumber mode as σ varies. The same behaviour is observed for other values of α,[b] and so we see that the as the force increases, the fibre modes tend to cluster near the imaginary axis further out from the origin. In terms of the stability of the Forward Euler scheme, this is the worst case scenario: k must be taken extremely small to deform the circular region enough to encompass these eigenvalues. This leads us to believe that the stiffness of the fibre modes will eventually begin to dominate the Forward Euler scheme as σ is increased. We will investigate the effects of varying the force parameter shortly.

[a] For the same reason, we cannot solve the dispersion relation with a smaller viscosity, which would also have reduced the effect of the Stokes modes.

[b] The $\alpha = 2\pi$ mode is not the one that limits the time step in calculations, but reliable solutions for λ could not be computed for large σ when $\alpha > 2\pi$. Nonetheless, this mode is representative of how the ratio varies with σ.

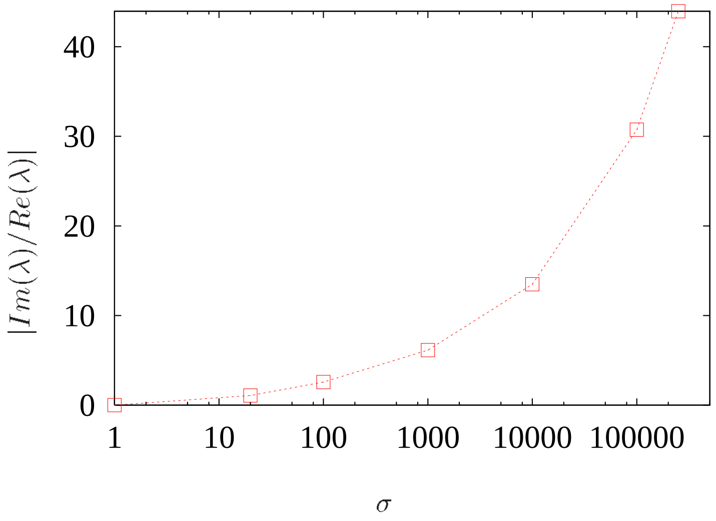


Figure 4.2: Plot of the ratio $Im(\lambda)/Re(\lambda)$ versus σ for the dominant wavenumber mode ($\alpha = 2\pi$). There is a clear tendency for the modes to approach the imaginary axis (that is, $Im(\lambda) \gg Re(\lambda)$) as the force parameter increases ($\mu = 1.0$).

An obvious question to ask at this point is: *Is there another explicit time-stepping scheme that does a better job of covering the imaginary λk-axis?* There are actually many possibilities, but the simplest and most obvious choice is the Runge–Kutta (RK) family of schemes, for which the Forward Euler method is the first order member (and so we denote this scheme by *"RK1"*). We also consider three other schemes: a second order Runge–Kutta method, *RK2*, also known as the *midpoint scheme*; the third order *RK3* scheme, which is a method attributed to Heun; and the standard fourth order formula *RK4* (all of which are described in [But96]). The stability region for each of the four methods is plotted in Figure 4.3, from which it is clear that better coverage is obtained at points near the imaginary axis as the order of the scheme is increased to 4. If we ignore Stokes modes for the moment, then for the particular choice of parameters in the example considered earlier, we may take $k \approx 6.0 \times 10^{-4}$ for the fourth order scheme, which is approximately 10 times larger than the time step allowed by Forward Euler. This is a considerable improvement, which we would expect gets even better as the force parameter σ is increased (again, we can only compare with the analytical results at this smaller value of σ, for which the dispersion relation

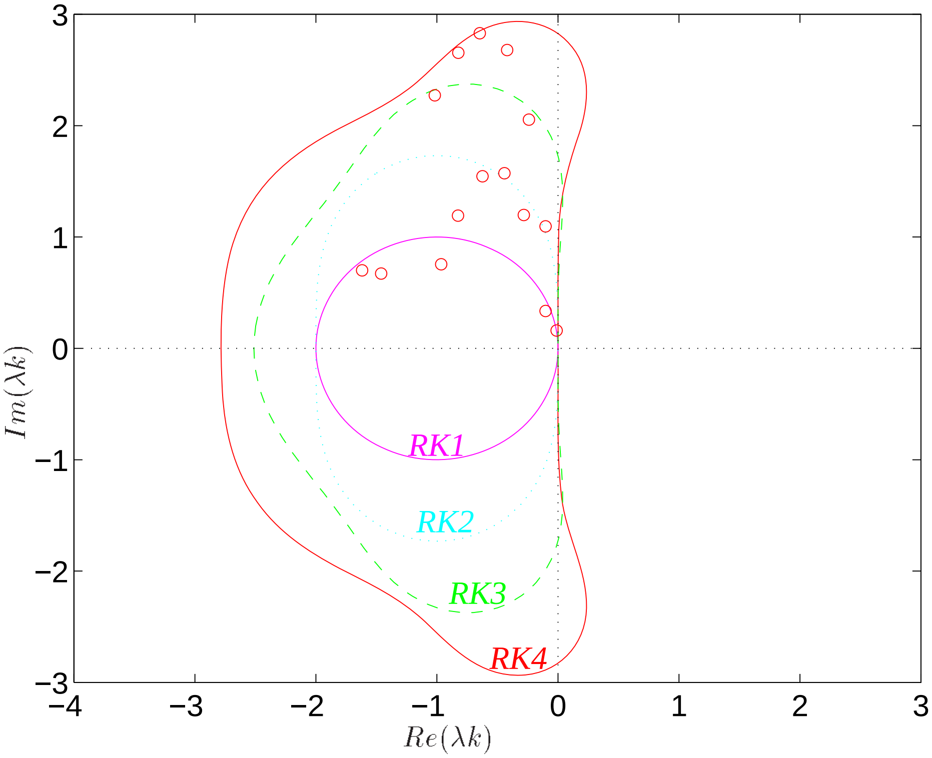


Figure 4.3: Region of absolute stability for the Runge–Kutta schemes of order 1, 2, 3, and 4. A selection of representative eigenvalues for the example problem are plotted on the same axes (using $\circ$), with the value of $k = 6.0 \times 10^{-4}$ chosen to be the best afforded by the fourth order scheme ($\sigma = 1,000$).

can be solved). The results for all four explicit methods are summarised in Table 4.1. The Stokes prediction is slightly different for *RK3* and *RK4* because of a modification in the constant 4 lying in the denominator of the stability condition (4.1) for each of these methods. The maximum time step observed in computations matches almost exactly with the Stokes prediction for each of the RK schemes. Therefore, in this parameter regime, it is diffusion and not the fibre force that governs the time step, which is what we observed earlier for the Forward Euler (*RK1*) scheme.

Scheme	Time step restrictions		
	Stokes prediction (diffusion)	Smooth prediction (fibre modes)	Computed
RK1	6.1×10^{-5}	6.6×10^{-5}	6×10^{-5}
RK2	6.1×10^{-5}	2.7×10^{-4}	6×10^{-5}
RK3	7.7×10^{-5}	4.8×10^{-4}	7×10^{-5}
RK4	8.5×10^{-5}	6.0×10^{-4}	8×10^{-5}
FE/ADI	—	—	5×10^{-4}

Table 4.1: Comparison of the predicted and computed time step restrictions for four fully explicit schemes, based on forcing alone ($\sigma = 1,000$, $\mu = 1$).

While the analytical dispersion relation can only be solved for small σ, we may still use computations to compare the effect of increasing the force parameter. In Figure 4.4, the maximum allowable time step k_{max} is plotted against σ for each scheme. The flat nature of the RK curves at lower σ values corresponds to the parameter regime where Stokes modes dominate and the stability restriction (4.1) holds. Here, there is nothing to be gained by using *RK4* because the negative real parts of the eigenvalues (arising from diffusion) are the limiting factor in explicit computations.

As the force is increased, the comparatively large imaginary parts of the fibre modes begin to dominate the computations and the advantage of the higher order RK methods over the Forward Euler scheme becomes evident. At $\sigma = 100,000$, the time step allowed for *RK4* is approximately

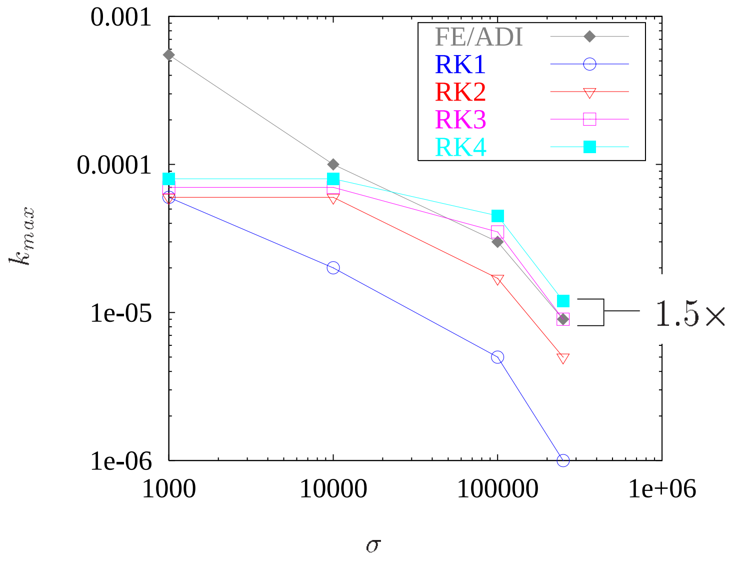


Figure 4.4: Comparison of computed time step restrictions for the Runge–Kutta schemes and the ADI implementation of the Immersed Boundary Method ($\mu = 1.0$).

15 times that for *RK1*. We have also plotted k_{max} for the Forward Euler/ADI implementation of the Immersed Boundary Method. The *RK4* method allows a time step only slightly larger, which is not enough to justify the added expense of the three extra stages.

Since implicit discretisation of diffusion seems to benefit the *FE/ADI* scheme so much, it seems worthwhile to investigate the use of semi–implicit Runge–Kutta methods, such as those described by Ascher, Ruuth & Spiteri [ARS97]. These methods combine implicit handling of diffusion along with all of the advantages of explicit Runge–Kutta stability regions, and we are currently investigating their application to immersed boundary computations.

Owing to the results of Section 3.1.4 where the dominance of fibre modes over Stokes modes asserts itself at higher Reynolds numbers, we performed the same set of computations with $\mu = 0.01$, which are plotted in Figure 4.5. As we might expect from our previous discussion, the advantage of the higher order Runge–Kutta methods becomes more pronounced as the Reynolds

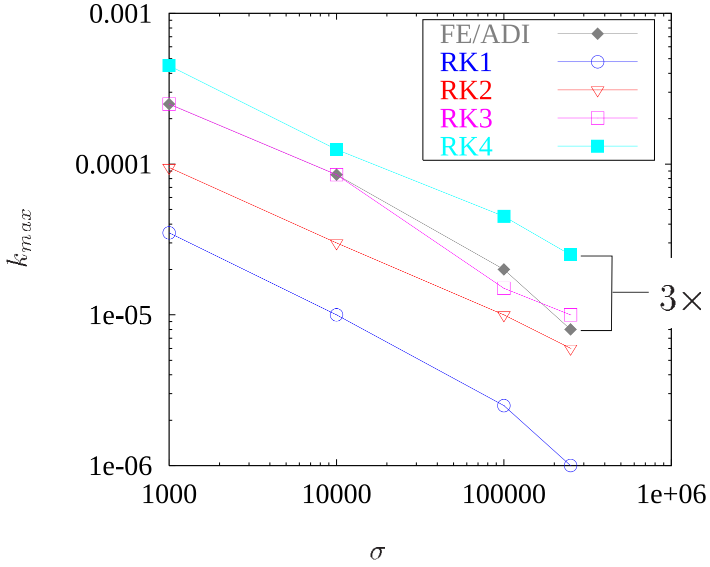


Figure 4.5: Comparison of computed time step restrictions for the Runge–Kutta schemes and the ADI implementation of the Immersed Boundary Method ($\mu = 0.01$).

number increases and the fibre modes dominate. The fourth order *RK4* scheme allows a time step 25 times larger than Forward Euler when $\sigma \geq 10,000$. A further comparison with the *FE/ADI* scheme demonstrates that the time step may be taken three times larger with the fully explicit *RK4* method: in this situation, the fourth order scheme is now much more competitive. The cost of the various schemes is compared in Table 4.2 in terms of the CPU time required for a time step just within the stability boundary.

While the time step requirements shown in Figures 4.4 and 4.5 for the fully explicit *RK1* and *RK4* schemes are consistent with the behaviour of the fibre modes with increasing force (see Figure 4.2), the *FE/ADI* scheme (which treats the fibre force in an explicit fashion) performs much better than we would expect. It is clear that the implicit treatment of viscous terms in the ADI approach offers some advantage, but further analysis is required to determine exactly how the scheme avoids the time step limitations present in fully explicit methods.

Scheme	CPU Time Required (sec)	
	$\sigma = 250,000$, $\mu = 1.0$	$\sigma = 250,000$, $\mu = 0.01$
RK1	370.59	1658.21
RK2	138.12	143.02
RK3	110.87	99.44
RK4	109.76	52.64
FE/ADI	53.96	54.12

Table 4.2: Comparison of the CPU times required for the various RK schemes when the time step is chosen just within the stability region. Timings were performed on an SGI Origin 2000 (4 × 195 MHz R10000 processors, 512 Mb RAM).

In conclusion, explicit Runge–Kutta methods provide no advantage over the *FE/ADI* method at low Reynolds numbers, since the Stokes modes and fibre modes are the same order of magnitude. However, semi–implicit Runge–Kutta schemes may offer a significant benefit, by allowing us to combine implicit discretisation of diffusion with the better treatment of stiff fibre modes afforded by higher order Runge–Kutta schemes. As $\mathcal{R}e$ is increased and the fibre modes begin to dominate the problem, the fourth order Runge–Kutta method is far superior to the *RK1* scheme. Furthermore, it is on an equal footing with the implicit *FE/ADI* scheme in terms of computational cost.

4.2 Implicit and Semi–Implicit Schemes

Before presenting the two semi–implicit schemes that we will investigate in the remainder of this chapter, we will first write a fully implicit version of the Immersed Boundary Method given in Section 2.3. Since the ensuing analysis deals only with solutions that are discrete in time and continuous in space, we will drop the subscripts $(\cdot)_{i,j}$ referring to the spatial discretisation and think of the solutions as depending on $\boldsymbol{x}$. Furthermore, the analysis is linear and so we will also leave out the convection terms from the momentum equations and consider the solution of Stokes equations instead.

A fully implicit Backward Euler discretisation of the problem can be written as the following system of equations to be solved simultaneously for $\vec{F}^n$, $\vec{U}^n$ and $\vec{X}^n$ within each time step:

$$\vec{F}^n(\boldsymbol{x}) = \int_\Gamma \sigma \frac{d^2\vec{X}^n(s)}{ds^2} \cdot \delta_h(\boldsymbol{x} - \vec{X}^n(s))\, ds \tag{4.2a}$$

$$\vec{U}^n(\boldsymbol{x}) = \vec{U}^{n-1}(\boldsymbol{x}) + \frac{k}{\rho}\mathcal{P}\left\{\mu\Delta\vec{U}^n(\boldsymbol{x}) + \vec{F}^n(\boldsymbol{x})\right\} \tag{4.2b}$$

$$\vec{X}^n(s) = \vec{X}^{n-1} + k\int_\Omega \vec{U}^n(\boldsymbol{x}) \cdot \delta_h(\boldsymbol{x} - \vec{X}^n(s))\, d\boldsymbol{x} \tag{4.2c}$$

where $\mathcal{P}$ is the projection operator defined in Section 2.3. In the fully discrete method, the integrals are replaced by sums as was done in (2.16b) and (2.16h). An implementation of the fully implicit scheme using a Newton iteration based on a Green's function solution for Stokes flow was proposed by Tu & Peskin [TP92], but was shown to be far too expensive to be of any practical use. Nonetheless, some form of semi–implicit discretisation is needed to couple the fibre force calculation with the fluid equations to overcome the stiffness recognised in the mathematical problem and in computations.

There have been many efforts to design a version of the scheme that handles the force implicitly, which can be distinguished from each other by identifying which terms in (4.2) are coupled implicitly with the force. The first attempt was the *approximate implicit* method [Pes77], which operates in two steps, like a predictor–corrector scheme:

- First, a prediction of the fibre position is computed by neglecting the coupling between fluid and fibre (by dropping the projection step and diffusion terms in (4.2b)) and applying the following fixed point iteration for $\vec{X}^*$:

  $$\vec{X}^* = \vec{X}^{n-1} + k\,\vec{U}^{n-1} + \frac{k^2}{\rho}\int_\Omega\int_\Gamma \sigma\frac{d^2\vec{X}^*}{ds^2}\,\delta_h(\boldsymbol{x} - \vec{X}(r,t))\,\delta_h(\boldsymbol{y} - \vec{X}(s,t))\,dr\,d\boldsymbol{y}. \tag{4.3}$$

  The resulting fibre configuration $\vec{X}^*$ is used to compute the force.

- The fluid velocity and fibre position are then computed using the standard ADI step (for convection and diffusion) followed by a projection step for the pressure.

In this version of the method, only the force and interface position have been coupled together using a fixed point iteration, and so it is not truly implicit. Though this approach has been taken in many simulations [PM89, FF93], it has been shown [MP93] to suffer from a degree of stiffness comparable to the fully explicit method.

We now present two semi–implicit schemes which lead to iterations that couple the fluid and fibre together. The major distinction between the methods (and the approximate implicit scheme, for that matter) is the choice of terms in (4.2) to treat explicitly. We begin with a simple fixed point iteration which uses a Crank–Nicholson–type discretisation in time and a Stokes solver rather than the split–step projection scheme. This approach has not appeared in the literature and is considered here for comparison purposes because of its simple structure.

Remark 4.1. *A fully coupled Stokes solver is identical to the split–step projection scheme in the discrete setting when the computational domain has periodic boundary conditions.*

This first scheme is formulated as an iteration embedded within each time step, with the solution at the mth iteration written with a second superscript $(\cdot)^{\cdot,m}$. Assuming that the values of the pressure, velocity and fibre position are known at time level $n-1$, the solution at time t_n is computed using the following algorithm.

Crank–Nicholson (CN) Scheme

Step CN1: Set the iteration counter $m = 0$ and the initial guesses

$$\vec{U}^{n,0} = \vec{U}^{n-1}, \qquad P^{n,0} = P^{n-1} \qquad \text{and} \qquad \vec{X}^{n,0} = \vec{X}^{n-1},$$

where $\nabla \cdot \vec{U}^{n-1} = 0$.

Step CN2: Compute the fibre force using

$$\vec{F}^{n,m} = \int_\Gamma \sigma \frac{d^2}{ds^2} \vec{X}^{n,m-1} \cdot \delta_h(\boldsymbol{x} - \vec{X}^{n-1})\, ds$$

Step CN3: Solve for the pressure and velocity simultaneously using Stokes equations with $\vec{F}^{m,n}$ in the right hand side:

$$\vec{U}^{n,m} = \vec{U}^{n-1} + \frac{k}{2\rho}\left\{\mu\Delta\left(\vec{U}^{n,m} + \vec{U}^{n-1}\right) - \nabla\left(P^{n,m} + P^{n-1}\right) + \left(\vec{F}^{n,m} + \vec{F}^{n-1}\right)\right\}$$

$$\nabla \cdot \boldsymbol{U}^{n,m} = 0$$

Step CN4: Update the fibre position for the next iteration

$$\vec{X}^{n,m} = \vec{X}^{n-1} + \frac{k}{2}\int_{\Omega}\left(\vec{U}^{n,m} + \vec{U}^{n-1}\right) \cdot \delta_h(\boldsymbol{x} - \vec{X}^{n-1})\, d\boldsymbol{x}$$

Step CN5: If the iteration has converged (that is, if $\|\vec{X}^{n,m} - \vec{X}^{n,m-1}\| < TOL$) then increment n and go to the next time step (Step 1).

Step CN6: Otherwise, increment m and iterate again starting from Step 2.

The next algorithm, proposed in [MP93], proceeds through the same main steps as the *CN* scheme, but is closer in spirit to the Immersed Boundary Method (2.16). The major differences from the *CN* method are:

- we revert to Backward Euler time discretisation;
- the diffusion terms are handled implicitly with an ADI step as in (2.16). However, there is one major point of departure from the original scheme in that the force is taken out of the step (2.16c);
- a preconditioner is used to accelerate convergence.

Before stating the algorithm, we introduce some additional notation to simplify the presentation of the iterative scheme. Let the delta function interpolation be written in terms of the operators $\mathcal{S}^n$ and $\overline{\mathcal{S}}^n$, where

$$\mathcal{S}^n\vec{V}(\boldsymbol{x}) = \int_{\Omega}\vec{V}(\boldsymbol{x}) \cdot \delta_h\left(\boldsymbol{x} - \vec{X}^n(s)\right)\, d\boldsymbol{x} \tag{4.4}$$

which interpolates fluid quantities $\vec{V}$ onto fibre points and

$$\overline{\mathcal{S}}^n \vec{W}(s) = \int_\Gamma \vec{W}(s) \cdot \delta_h \left(\boldsymbol{x} - \vec{X}^n(s) \right) \, ds \tag{4.5}$$

interpolates fibre quantities $\vec{W}$ onto fluid points.

Mayo–Peskin (MP) Scheme

Step MP1: Compute the intermediate velocity $\vec{U}^{n,*}$ using the ADI procedure described in (2.16c)–(2.16e), except that the fibre force in the right hand side of (2.16c) is zero. Here, $\vec{U}^{n,2}$ has been relabeled $\vec{U}^{n,*}$ to avoid confusion with the iteration subscript m.

Step MP2: Set the iteration counter $m = 0$ and the initial guesses

$$\vec{U}^{n,0} = \vec{U}^{n-1} \text{ or } \vec{U}^{n,*} \qquad \text{and} \qquad \vec{X}^{n,0} = \vec{X}^{n-1},$$

where $\nabla \cdot \vec{U}^{n-1} = 0$.

Step MP3: Compute the force using

$$\begin{aligned} \vec{F}^{n,m} &= \int_\Gamma \sigma \frac{d^2}{ds^2} \vec{X}^{n,m-1} \cdot \delta_h(\boldsymbol{x} - \vec{X}^{n-1}) \, ds \\ &= \overline{\mathcal{S}} \sigma \frac{d^2}{ds^2} \vec{X}^{n,m-1} \end{aligned}$$

where we drop the superscript $(\cdot)^{n-1}$ on $\overline{\mathcal{S}}$.

Step MP4: Perform the projection step

$$\vec{U}^{n,m} = \mathcal{P} \left\{ \vec{U}^{n,*} + \frac{k}{\rho} \vec{F}^{n,m} \right\}$$

Step MP5: Update the fibre position for the next iteration. This step requires some additional explanation, since it is here that the preconditioner is introduced. If we substitute the expressions from Steps 3 and 4 into the backward Euler formula for $\vec{X}^{n,m+1}$, we obtain

$$\begin{aligned}\vec{X}^{n,m} &= \vec{X}^{n-1} + k\,\mathcal{S}\left(\mathcal{P}\vec{U}^{n,*} + \frac{k}{\rho}\mathcal{P}\overline{\mathcal{S}}\sigma\frac{d^2}{ds^2}\vec{X}^{n,m-1}\right) \\ &= \underbrace{\left(\vec{X}^{n-1} + k\mathcal{S}\mathcal{P}\vec{U}^{n,*}\right)}_{\vec{Z}^{n-1}} + \mathcal{S}\mathcal{P}\overline{\mathcal{S}}\underbrace{\frac{\sigma k^2}{\rho}\frac{d^2}{ds^2}}_{\mathcal{A}}\vec{X}^{n,m-1}\end{aligned}$$

which can be written more compactly as

$$\vec{X}^{n,m} = \vec{Z}^{n-1} + \mathcal{S}\mathcal{P}\overline{\mathcal{S}}\mathcal{A}\vec{X}^{n,m-1}.$$

As it stands, this iteration also converges very slowly. The convergence can be speeded considerably by writing an modified iteration which has the same solution:

$$\underbrace{(I-\Lambda\mathcal{A})}_{tridiagonal}(\vec{X}^{n,m} - \vec{X}^{n,m-1}) = \vec{Z}^{n-1} - \underbrace{(I-\mathcal{S}\mathcal{P}\overline{\mathcal{S}}\mathcal{A})}_{dense}\vec{X}^{n,m-1}$$

In the fully discrete setting, $\Lambda = \mathcal{S}\overline{\mathcal{S}}$ is a diagonal matrix, and so the *preconditioning matrix* $(I - \Lambda\mathcal{A})$ is a block tridiagonal matrix, and consequently very easy to invert.

Step MP6: If the iteration has converged (that is, if $\|\vec{X}^{n,m} - \vec{X}^{n,m-1}\| < TOL$) then increment n and go to the next time step (Step 1).

Step MP7: Otherwise, increment m and iterate again starting from Step 2.

4.3 Linear Analysis of the Two Iterative Schemes

The main objective in this section is to determine the conditions under which the two iterative schemes described in the previous section converge. We are still in the semi–discrete setting (that is, continuous in space) and so we may use a similar approach to that employed for the continuous problem in Section 3.2. We will look for solutions of the linearised problem that take the form

$$\left\{ \begin{matrix} U \\ V \\ P \\ X \\ Y \end{matrix} \right\}^{n,m} = e^{\hat{\imath}\alpha x} \left\{ \begin{matrix} \widehat{U}(y) \\ \widehat{V}(y) \\ \widehat{P}(y) \\ \widehat{X} \\ \widehat{Y} \end{matrix} \right\}^{n,m} , \tag{4.6}$$

on each of the subdomains $\Omega_0^{\pm}$ and Ω_0^{ϵ}. The delta function is replaced by the cosine approximation, and the equations of motion can be solved as before.

For both schemes described above, the iteration can be reduced to one on $\vec{\Xi}^{n,m} = (\hat{X}^{n,m}, \hat{Y}^{n,m})$ only, and written as

$$\mathbf{B}\vec{\Xi}^{n,m} = \mathbf{C}\vec{\Xi}^{n,m-1} + \vec{R}^{n-1},$$

where $\mathbf{B}$ and $\mathbf{C}$ are 2×2 matrices and $\vec{R}^{n-1}$ is a 2–vector with entries evaluated at the previous time step. We are only interested in the convergence of the iteration, and so we consider the difference between successive iterates

$$\vec{E}^{n,m} := \vec{\Xi}^{n,m} - \vec{\Xi}^{n,m-1}$$

which satisfies the equation

$$\vec{E}^{n,m} = \mathbf{M}\vec{E}^{n,m-1},$$

where $\mathbf{M} = \mathbf{B}^{-1}\mathbf{C}$ is the *iteration matrix*. In the analysis to follow, the linearisation process decouples the normal and tangential motions of the fibre and so one eigenvalue of $\mathbf{M}$ depends on σ_t only, and the second on σ_n only. The convergence properties of each scheme are manifested in the eigenvalues of $\mathbf{M}$, which are easy to compute. The magnitude of the largest eigenvalue of $\mathbf{M}$, which we denote by ϱ_{max}, is a measure of the rate of convergence of the iteration. In particular, if $\varrho_{max} < 1$, then the scheme converges; otherwise it diverges.

Another consequence of the linearisation process and reformulation of the solution as an iteration on the fibre position only is that the effects of the ADI step are neglected. The *CN* scheme does not use ADI at all, and in the Mayo–Peskin scheme, the split–step nature of the method leaves the ADI step for diffusion outside of the iteration.

4.3.1 Crank–Nicholson convergence rates

The solution procedure parallels that for the smoothed delta function problem in Section 3.2. The solutions on the regions $\Omega_0^{\pm}$ are given by

$$\begin{aligned}\widehat{p}^{\pm}(y) &= A^{\pm}e^{\mp\alpha y},\\ \widehat{u}^{\pm}(y) &= B^{\pm}e^{\mp\beta y} - \frac{\hat{\imath}\alpha k}{2\rho}A^{\pm}e^{\mp\alpha y},\\ \widehat{v}^{\pm}(y) &= C^{\pm}e^{\mp\beta y} \pm \frac{\alpha k}{2\rho}A^{\pm}e^{\mp\alpha y},\end{aligned}$$

where we have defined

$$\beta^2 := \alpha^2 + \frac{2\rho}{\mu k},$$

and dropped the superscripts $(\cdot)^{n,m}$ for ease of notation. These expressions are identical to (3.26)–(3.28), except that the continuous time parameter λ is replaced by $\frac{2}{k}$.

The solution on the smoothing region Ω_0^{ϵ} is derived in a similar manner, and the equations are unwieldy so we leave out the details. However, it is important to realise that there is one very significant difference in the dependence of the solution on the fibre unknowns $\widehat{X}$ and $\widehat{Y}$ from the continuous problem. Rather than the fluid force being defined implicitly in terms of the fibre position, we have in Step 2 of the algorithm that the force is computed *based on the fibre position from the previous iteration.* Consequently, the semi–discrete analogues of the fibre evolution equations (3.39) and (3.40), are explicit formulas for $\vec{\Xi}^{n,m}$ in terms of $\vec{\Xi}^{n,m-1}$.

The system of integro–partial differential equations is solved using Maple, after which we can apply the same matching conditions at the interfaces $y = \pm\epsilon$ to obtain a system of linear equations relating the unknown solution coefficients (the constants of integration on Ω_0^{ϵ}, along with $A^{\pm}$, $B^{\pm}$ $C^{\pm}$). Here, our approach diverges somewhat from that of the continuous problem, in that we need to solve this system of equations to obtain the iteration matrix. Fortunately, the simplification mentioned above (namely, that in the discrete case the fibre positions are no longer coupled to the fluid unknowns) makes it possible to solve explicitly for the coefficients. The system of equations now has a non–zero right hand side (corresponding to the entries from the previous iteration that appear in the expressions for the fibre positions). The coefficient matrix this time has determinant

$$\frac{8k\alpha^2}{\mu\rho\beta^2}\left(\beta^2-\alpha^2\right)$$

which is never zero, provided $\beta \neq \alpha$, thus guaranteeing that the system is always solvable.

After a lengthy Maple computation, we can derive the iteration matrix for the fibre position, which has the very special form

$$\mathbf{M} = \begin{bmatrix} \varrho_t & 0 \\ 0 & \varrho_n \end{bmatrix}, \tag{4.7}$$

where ϱ_t depends on σ_t only, and ϱ_n depends on σ_n only (the explicit forms are long and complicated expressions involving the problem parameters, and so they are not given here). We can make two important observations regarding the convergence of the scheme:

- there is a decoupling between the tangential and normal fibre motions, and the convergence for each of the two modes of oscillation is governed by a rate;
- the convergence rates of the scheme, ϱ_n and ϱ_t, depend *linearly* on the fibre stress parameters σ_n and σ_t.

A contour plot of the convergence rate $\varrho_{max} = \max\left(|\varrho_t|, |\varrho_n|\right)$ is given in Figure 4.6, with parameter values $\sigma = 10,000$ and $N = 64$, $\epsilon = \frac{2}{64}$ and over a range of k and $\alpha \in [2\pi, 128\pi]$. From these results, it is clear that the *CN* scheme is only conditionally convergent. Furthermore, the contours decay quite rapidly to zero from the boundary of the shaded region — this indicates that if the time step is chosen so that the iterations are convergent, then they should converge quite rapidly (in practice, within a couple of iterations). These predictions will be compared with computations in Section 4.4.

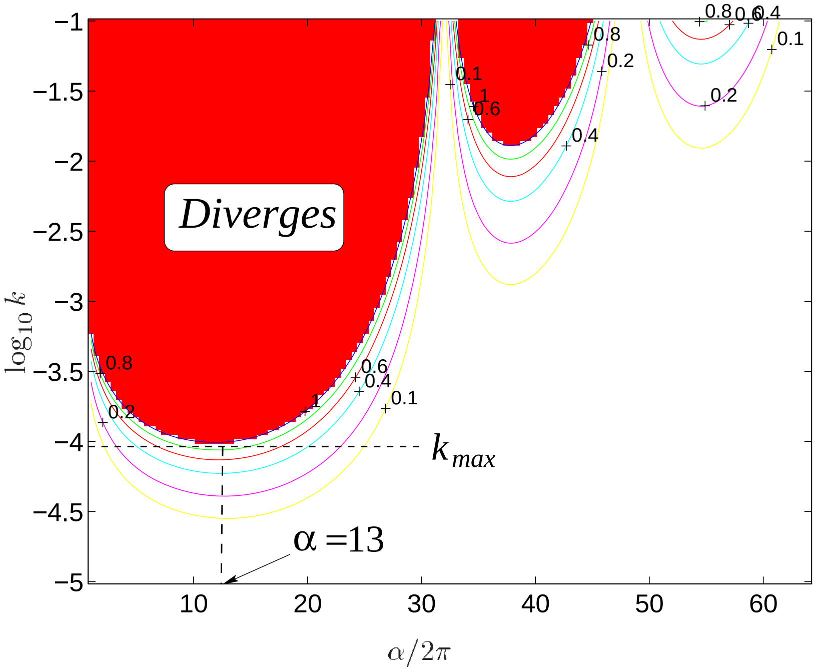


Figure 4.6: Convergence rate contours for the *CN* scheme with the region of divergence $\varrho_{\max} > 1$ shaded. Here, all plotted points derive from the tangential mode, which always has the largest convergence rate ($\sigma = 10,000,\ N = 64$).

4.3.2 Mayo–Peskin convergence rates

The iteration matrix for the second scheme is not diagonal (it is a full 2×2 matrix) but the structure is quite similar to that of the first scheme. This time, the expressions for the eigenvalues of the matrix are much simpler, and so we given them below:

$$\varrho_t = \frac{\pi^4 \sigma_t k^2 \sin^2(\alpha\epsilon)\left(-\pi^4 + \pi^4 e^{-2\alpha\epsilon} + 5\epsilon^3\pi^2\alpha^3 + 3\epsilon^5\alpha^5 + 2\pi^4\epsilon\alpha\right)}{\alpha\epsilon\left(\alpha^2\epsilon^2+\pi^2\right)^2\left(-4\epsilon^7\alpha^4 + 8\epsilon^5\alpha^2\pi^2 - 4\epsilon^3\pi^4 + 3\pi^4\sigma_t k^2 \sin^2(\alpha\epsilon)\right)} \tag{4.8a}$$

$$\varrho_n = \frac{\pi^6 \sigma_n k^2 \sin^2(\alpha\epsilon)\left(-\pi^2 e^{-2\alpha\epsilon} + \epsilon^3\alpha^3 + \pi^2\epsilon\alpha + \pi^2\right)}{\alpha\epsilon\left(\alpha^2\epsilon^2+\pi^2\right)^2\left(-4\epsilon^7\alpha^4 + 8\epsilon^5\alpha^2\pi^2 - 4\epsilon^3\pi^4 + 3\pi^4\sigma_n k^2 \sin^2(\alpha\epsilon)\right)} \tag{4.8b}$$

A contour plot of the convergence rate $\varrho_{max} = \max\left(|\varrho_t|, |\varrho_n|\right)$ is given in Figure 4.7, with parameter values $\mu = \rho = 1$, $\sigma = 10,000$ and $N = 64$ and over a range of k and $\alpha \in [2\pi, 128\pi]$. Based on the convergence rates for the linearised problem, we can make the following observations:

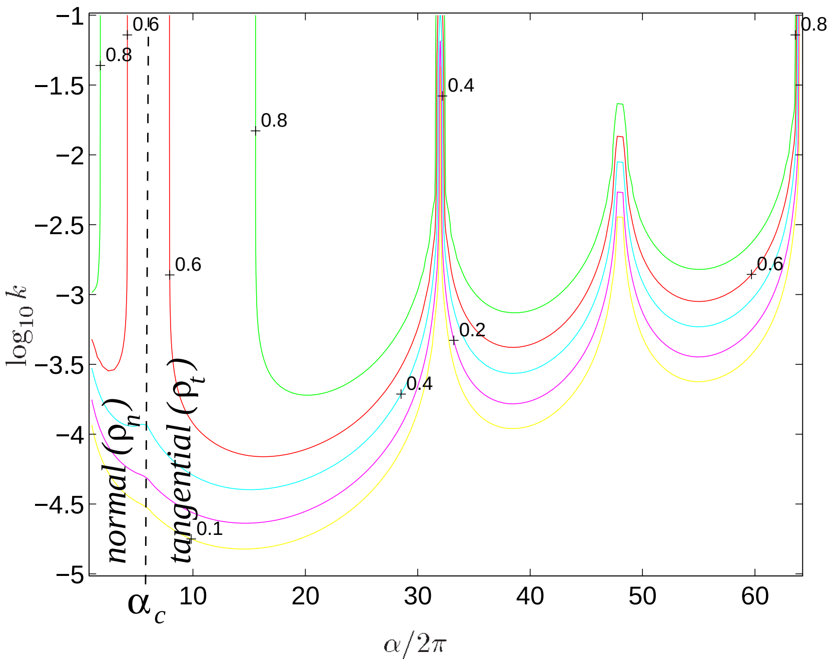


Figure 4.7: Convergence rate contours for the *MP* scheme. The vertical dotted line separates the parameter space into regions where the convergence rate for the normal mode (left) or the tangential mode (right) dominate ($\sigma = 10,000$, $N = 64$).

- The iteration is unconditionally convergent. This is to be expected, since Mayo & Peskin prove this result in [MP93].

- There is a critical value of α given by

$$\alpha_c \approx (0.09057685940 \cdot N) \cdot 2\pi$$

at which $\varrho_n = \varrho_t$ (we have found this expression by equating the two eigenvalues in (4.8), setting $\sigma_n = \sigma_t$ and $\epsilon = \frac{2}{N}$, and solving for α). For the example given in the contour plot, we have $\alpha_c \approx 5.7969 \cdot 2\pi$, which is indicated in Figure 4.7 as a dashed vertical line. For $\alpha < \alpha_c$, the normal convergence rate ϱ_n is the largest, while for $\alpha > \alpha_c$ the convergence of the tangential modes (ϱ_t) dominates the calculation.

Remark 4.2. *There are actually two versions of the MP iterative scheme, one which bases the delta function interpolation on the fibre position from the previous time step (using $\mathcal{S}^{n-1}$ and $\overline{\mathcal{S}}^{n-1}$) while the other uses $\vec{X}^{n,m}$ for the interpolation (that is, $\mathcal{S}^{n,m}$ and $\overline{\mathcal{S}}^{n,m}$). This second version of the scheme (also proposed in [MP93]) is a more stable alternative to the first, since the interpolation is implicit in the fibre position as well. However, it is also considerably more expensive than the first, since the preconditioner must be evaluated at every iteration rather than just once every time step. As it turns out, the preconditioner is an extremely expensive part of the calculation, and can even outstrip the cost of the fluid solver if it is recomputed many times in each time step.*

The linear analysis is unable to distinguish between these two alternatives, since any terms in the interpolation that involve the interface position appear at a higher order and are thus dropped in the linearisation process. However, this is not a serious limitation, since the computational results in the following section are not affected appreciably whether we base the interpolation on $\mathcal{S}^{n-1}$ or $\mathcal{S}^{n,m}$.

4.4 Computational Results

To test the accuracy of the predicted convergence region for the first method (CN), we performed several numerical experiments on the "flat fibre" test problem depicted in Figure 3.10. The results are summarised in Table 4.3. The requirement on k in computations is *very sharp*, which matches with the steep contours in Figure 4.6 — that is, either the scheme diverges, or it converges within

σ	Maximum k		
	Predicted	Computed (CN)	Computed (FE/ADI)
100	0.0018	0.0010	0.008
1,000	0.00039	0.0003	0.0006
10,000	0.000098	0.00009	0.0001
100,000	0.000028	0.00002	0.00003

Table 4.3: Predicted and computed stability boundaries for k using the CN scheme on the "flat fibre" problem ($N = 64$).

one or two iterations. The predicted and computed stability boundaries match quite closely in the range $\sigma = 10^2$–10^5 (corresponding to "physical" values).

The third column gives the maximum time step allowed for the Forward Euler/ADI scheme for comparison. It is clear that there is *no advantage* to using the CN scheme over the original algorithm. Since the force is not handled in a truly implicit fashion, but rather using a fixed point iteration, these results suggest that we must look for another approach which has a better implicit treatment of the stiff forcing term. We will see next that the MP scheme does a good job in this respect.

We can capture the individual solution modes that cause the onset of instability by calculating with a time step only slightly inside the divergence boundary and observing the shape of the fibre as the iteration diverges. Figure 4.8 depicts the fibre configuration (for two values of σ) at an intermediate stage in a divergent iteration, where it is easy to see the instability that develops. There is a clearly defined mode that is excited in each case: $\alpha = 11 \cdot 2\pi$ for $\sigma = 1,000$, and $\alpha = 13 \cdot 2\pi$ for $\sigma = 10,000$. These wavenumbers are exactly those that are predicted by the analysis — namely, the tangential modes having the largest convergence rate, or in other words, the α that has $\varrho_{max} \gtrsim 1$ as we approach the convergence boundary. Referring to Figure 4.6 (for

$\sigma = 10,000$), we see that the wavenumber that intersects the k_{max} boundary is in fact very close to 13.

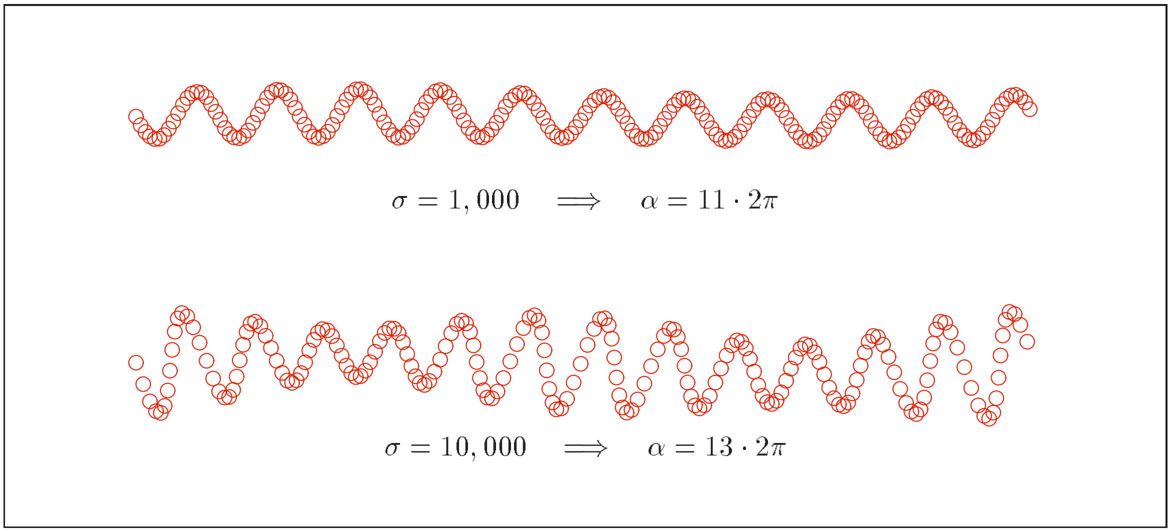


Figure 4.8: Snapshots of the instabilities arising in divergent iterations for the *CN* scheme. The wavenumber of the excited unstable mode matches exactly with the mode having the largest convergence rate for simulations with $\sigma = 1,000$ and $10,000$.

We believe that these unstable tangential modes have also been observed in immersed boundary computations involving two–dimensional heart simulations performed by McCracken & Peskin using a vortex method [MP80]. At high Reynolds numbers ($\mathcal{R}e \approx 300$), they observed instabilities in their computations which appear as "wiggles" in the fibres comprising valve leaflets and the heart wall, accompanied by small vortices in the adjacent fluid. These features appear to be very similar to the medium wavenumber tangential modes excited in our model computations when the time step is taken very close to the convergence limit. McCracken & Peskin explain the instability as follows [MP80, p. 203]: *" . . . because of the very large forces generated at the boundary during ventricular systole, we are unable to complete the runs that we have made at higher Reynolds numbers."* We can provide a more satisfactory explanation using our understanding of the behaviour of the solution modes. It is the combination of a large forcing parameter *and small viscosity* that limits the time step in their high Reynolds number computations.

k	$\sigma = 1,000$		$\sigma = 10,000$		$\sigma = 100,000$	
	Pred.	Comp.	Pred.	Comp.	Pred.	Comp.
0.0001	0.01	0.01	0.08	0.08	0.43	0.43
0.0025	0.05	0.05	0.33	0.33	0.75	0.76
0.0005	0.17	0.18	0.62	0.62	0.84	—
0.001	0.43	0.43	0.79	0.79	0.87	—
0.0025	0.75	0.73	0.86	—	0.88	—
0.005	0.84	0.84	0.88	—	0.88	—

Table 4.4: Predicted and computed convergence rates for the *MP* scheme applied to the "flat fibre" problem ($N = 64$). The "—" entries correspond to instances where the scheme went unstable.

We also performed several computations to verify the convergence rates for the *MP* scheme from Figure 4.7, and the results are summarised in Table 4.4. The convergence rate was computed from the numerical results using the formula

$$Rate = \frac{Res^{m+1}}{Res^m},$$

where Res^m is the residual at iteration level m computed as follows:

$$Res^m = \left[\frac{1}{N_b} \sum_{\ell=0}^{N_b-1} \left\| \vec{X}_\ell^m - \vec{X}_\ell^{m-1} \right\|_2^2 \right]^{1/2}$$

and where $\|\cdot\|_2$ is the standard L_2–norm on vectors. The predicted convergence rates were found by reading off ϱ_{max} for the $\alpha = 2\pi$ mode from the contour plot in Figure 4.7, for which we always have $\varrho_{max} = \varrho_n$ (that is, the normal mode dominates the calculation at the lowest wavenumber), even though ϱ_t is always the largest convergence rate when the full range of α is considered. While intermediate wavenumber modes have the largest ϱ in a given computation, and hence will dominate the convergence rate after a large number of iterations, they are also modes whose amplitude decays much more rapidly in time. Within every time step, however, only ten or so iterations were typically required to satisfy the residual tolerance, and so we expect that the lowest wavenumber modes will dominate the actual convergence rate observed in computations. This is verified by the results of Table 4.4 which compare the $\alpha = 2\pi$ prediction to the computed convergence rates.

The blank entries in the table correspond to instances where the iteration failed to converge, which seems to go against our analytical predictions of unconditional convergence. However, we believe that this arises from a time instability which affects the numerical scheme when the time step is taken too large, perhaps due to the fibre crossing multiple mesh lines. In fact, Mayo & Peskin identify [MP93, p. 269] that even though the iteration scheme is convergent and is *more* stable in time than the fully explicit method, it is not *always* stable.

We now consider another test problem, which is more typical of that seen in the literature to date. By considering this second example, we hope to be able to test the applicability of our analysis to problems that are more strongly nonlinear. In [TP92], [MP93] and [LL97] for example, an elliptical fibre such as that pictured in Figure 4.9 is used to test various aspects of their numerical methods. We take the semi–axes of the ellipse to have length 0.2 cm and 0.4 cm, and use the same linear force density function with stiffness constant σ. The ellipse will tend toward an equilibrium state that is a circle with the same area as the original ellipse, because the fluid is incompressible — the radius of this final circle is approximately equal to 0.2828 cm.

The time step restrictions for the *CN* scheme are compared to the values predicted by our analysis in Table 4.5. We can justify applying the results for a sinusoidally–perturbed flat fibre to the ellipse using the same physical parameters as follows. The modes which govern the convergence behaviour of the scheme lie at medium wavenumbers. We expect that all but the lowest wavenumber modes will be essentially unchanged whether they are located on a flat fibre or on a curved ellipse.

There is, in fact, very little difference between the time restrictions for the elliptical interface and those for the "flat fibre" example given in Table 4.3, particularly for the more representative

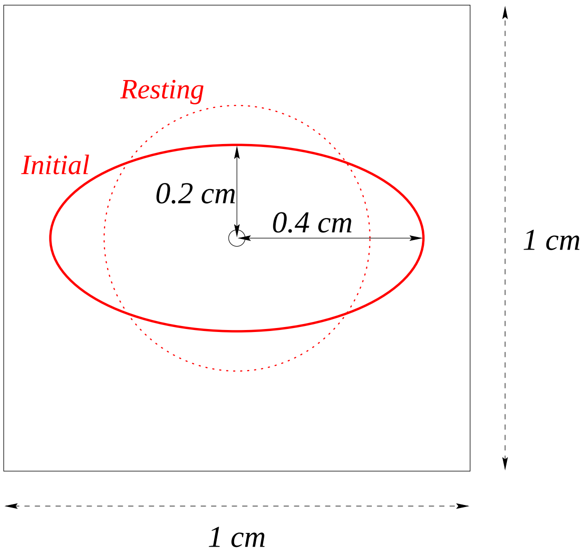


Figure 4.9: The "ellipse" test problem: the initial fibre position is an ellipse with semi–axes 0.4 *cm* and 0.2 *cm*. The equilibrium state is a circle with radius approximately 0.2828 *cm*.

σ	Maximum k	
	Predicted	Computed (CN)
100	0.0018	0.0006
1,000	0.00039	0.00020
10,000	0.000098	0.000060
100,000	0.000028	0.000020

Table 4.5: Predicted and computed stability boundaries for k using the CN scheme on the "ellipse" problem ($N = 64$).

forcing parameters in the range 10,000–100,000. Consequently, we are encouraged that our analytical predictions can be applied to a wider range of non–linear problems, rather than simply the specialised flat fibre test problem tailored to the linear analysis.

Before completing our discussion of time discretisations, we will draw a comparison between the explicit schemes studied in Section 4.1 and the two semi–implicit iterative approaches just considered. For the ellipse test problem, we applied the *RK1*, *RK4*, and *FE/ADI* schemes that are explicit in the force, and the *CN* and *MP* methods that couple the force to the fluid within a fixed point iteration. Table 4.6 lists the maximum time steps and CPU times required for each method for two sets of computations with $\sigma = 10^4$ and 10^5, $\mu = \rho = 1$, $N = 64$ and $N_b = 192$. Notice that the *CN* scheme offers no advantage over the fully explicit *RK4*, particularly in terms of CPU time, which is not surprising from our previous comparisons of the *CN* and explicit stability restrictions. Furthermore, the *RK4* method is almost 10 times more efficient than the Forward

Scheme	$\sigma = 10,000,\ t_{end} = 0.020$			$\sigma = 100,000,\ t_{end} = 0.005$		
	k_{max}	Vol. loss	CPU	k_{max}	Vol. loss	CPU
RK1	1.3×10^{-5}	0.028	114.31	1.0×10^{-6}	0.044	372.49
RK4	8.0×10^{-5}	0.024	66.51	3.0×10^{-5}	0.044	44.16
FE/ADI	7.0×10^{-5}	0.044	28.45	1.0×10^{-5}	0.052	49.00
CN	6.0×10^{-5}	0.076	65.99	1.0×10^{-5}	0.068	98.30
MP	8.0×10^{-5}	0.084	56.62	2.5×10^{-5}	0.068	44.00
	1.6×10^{-4}	0.131	29.99	5.0×10^{-5}	0.119	26.72

Table 4.6: Comparison of computational cost for several explicit and semi–implicit schemes. The time step k_{max} was chosen to be the largest allowed by the method for stability, except for the *MP* scheme (which always converged), for which we chose two representative time steps to compare the volume leakage. The "Vol. loss" is computed relative to the equilibrium value of 0.251 cm^2. CPU timings were taken on an SGI Origin 2000 (4 × 195 MHz R10000 processors, 512 Mb RAM).

Euler (*RK1*) scheme.

Before discussing the relation of the Mayo–Peskin scheme to the others, we must introduce the very important issue of *volume loss*: immersed boundary computations are known to experience loss of volume which becomes significant during more extreme flow conditions (large fibre force, pressure or velocity) such as those we are considering here. In our 2D ellipse example, this manifests itself as a steady loss of area enclosed within the immersed fibre. The volume loss problem was identified by Peskin & Printz in [PP93][c] and shown to arise not from fluid passing physically through the immersed boundary (since the fibre points move along streamlines), but rather to the fact that the interpolated velocity field through which the immersed boundary moves is not discretely divergence–free. LeVeque & Li showed in [LL97] that the volume loss in the Immersed Boundary Method for a problem nearly identical to our ellipse example grows linearly in time. Peskin & Printz proposed a modified divergence stencil which reduces the volume loss significantly at the expense of an increase in the cost of delta function interpolation.

We have not implemented this modified stencil in our code, since the main point we wish to make is that the effect of volume loss is significant for the semi–implicit schemes (in particular, the *MP* scheme) when the time step is taken relatively large. While the iterative method does allow a much larger time step to be taken than for explicit schemes, there is a corresponding increase in the rate of volume loss. This is clearly indicated in the "Vol. loss" column from Table 4.6, which gives the change in area enclosed by the fibre between the beginning and end of the run, relative to the initial area of 0.251 cm^2. Two different time steps are used for the *MP* scheme, which shows that while a larger time step may be taken to reduce the computational cost, it also leads to a much greater loss of volume.

[c] The flow conditions under which the volume loss occurs again lie in a more extreme range of parameters, about which they remark [PP93, p. 33]: *" ... [the volume loss effect] was small enough to be tolerable in the applications described above. In the cardiac research, however, this was only because the computational experiments were primarily concerned with diastole, during which the heart walls are relaxed and the pressure in the cardiac chambers is low. In more recent work concerning ventricular systole, we have found that the volume loss is too large to ignore."*

The explicit methods (*RK1*, *RK4* and *FE/ADI*) have a significantly lower rate of volume loss. Moreover, the *RK4* method is actually quite competitive with the *FE/ADI* and *MP* in terms of computational cost while at the same time experiencing smaller volume errors. We can conclude from these results that while the *MP* iteration may be unconditionally convergent and allow significantly larger time steps to be taken, the time step is still limited by the spatial error in the incompressibility condition. Clearly, there is a need for more work to be done on developing new time–stepping strategies to treat the force implicitly in some type of iteration, while at the same time controlling the volume error.

Chapter 5
Application: Pulp Fibres

A theory has only the alternative of being right or wrong. A model has a third possibility: it may be right, but irrelevant.

— Manfred Eigen.

We will now depart somewhat from the mathematical and numerical analyses of the previous two chapters and concentrate on a novel application of the Immersed Boundary Method to motion of pulp fibres. The main purpose in this chapter is to demonstrate the applicability of the method through comparison with experiments, theory and other computations. Our eventual goal is to use the pulp fibre problem as a benchmark for testing modifications to the numerical scheme that are suggested by our previous analytical work.

We begin with a brief introduction to the experimental, analytical and computational research that has been done on pulp fibres to date. In no way is this intended to be an exhaustive review of the literature; instead, we highlight the main properties of pulp fibres and motivate the importance of numerical simulations. Since the behaviour of fibres depends on so many parameters, we derive a non–dimensional measure of flexibility that can be used to classify fibre motion. Following that, we will illustrate why the immersed fibre framework is so well suited to modeling the dynamics of pulp suspensions. The modifications to the Immersed Boundary Method that are necessary in order to incorporate the geometry and physics particular to this problem are also outlined.

Finally, a series of simulations is performed to validate the immersed boundary model. While the 2D simulations are unable to capture some of the 3D aspects of fibre motion, we will show that qualitative features of planar motions are correctly reproduced. Comparisons made on the basis of our dimensionless flexibility parameter will allow us to compare fibre motions over a large range of physical parameters.

5.1 Physical Background: Pulp Fibres in Shear Flow

A basic understanding of the behaviour of pulp fibres in suspension is extremely important to the pulp and paper industry in many stages of the paper–making process. Of particular interest is the study of fibres of varying length and flexibility, suspended in a shear flow. Take for example the filtering process, where there is a need to separate fibres based on physical properties. Moderately flexible fibres are more desirable than rigid fibres because they have a higher relative bonding area and thus form paper with higher tensile stress [DK82] and better printability; hence, the ability to control the separation process based on flexibility is a prime factor in forming high quality paper. It is also important from the standpoint of quality to obtain pulp consisting of fairly uniform length fibres. Consequently, a knowledge of the hydrodynamic behaviour of fibres with different length is equally essential.

Pulp consists of roughly cylindrical fibres of length 0.1–0.3 cm and aspect ratios ranging from 60 up to 400. A considerable amount of theoretical work has been done on modeling fibres, since fibre suspensions appear in many applications other than paper formation. Much of the theory centres around the motion of rigid cylindrical rods immersed in low Reynolds number or Stokes flows. Attempts have been made to add a small degree of flexibility, but these results are usually fairly limited in their application. Accordingly, much of the work on flexible fibres has been experimental, though more recently numerical simulations have begun to be used.

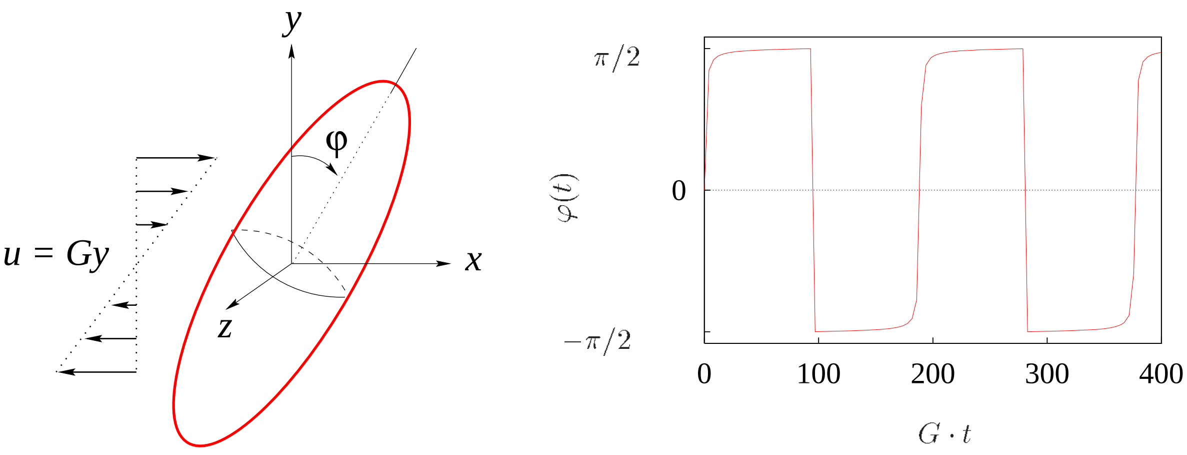
In the following two sections, we will describe all three approaches — analytical, experimental, and computational. The paper of Wherrett et al. [WGSO97] provides an excellent review of the literature on the subject, and we have drawn much of the material in the following sections from that paper and the references therein.

5.1.1 Theory and experiments

As early as 1922, Jeffery studied the motion of a rigid, neutrally–buoyant, elliptical particle in a homogeneous Stokes flow [Jef22]. He proved that the centre of the particle follows streamlines, and that when subjected to a Couette flow, it rotates about its centre according to

$$\varphi(t) = \tan^{-1}\left[r_e \tan\left(\frac{G r_e t}{r_e^2+1}\right)\right], \tag{5.1}$$

where ϕ is the angle the major axis of the ellipse makes with the vertical, G is the shear rate, and r_e is the ratio of the lengths of the major and minor axes of the ellipsoid (refer to Figure 5.1). Two things can be deduced from this formula: first, the particle has a non–uniform angular velocity which is largest at $\varphi = 0°$ and slows to a minimum near $\varphi = 90°$; and second, the period of motion is a constant, given by



(a) Jeffery's ellipsoidal particle immersed in a linear shear flow.

(b) A plot of the angular displacement of the rotating ellipsoid in the $x-y$ plane versus non–dimensionalised time (for $r_e = 60$).

Figure 5.1: An ellipsoidal particle in a shear flow, moving according to Jeffery's equation.

$$T = \frac{2\pi}{G}\left(r_e + \frac{1}{r_e}\right), \tag{5.2}$$

which is approximately $T \approx 2\pi r_e/G$ for long, thin ellipsoids (when $r_e \gg 1$).

Anczurowski & Mason [AM67] showed that Jeffery's equation (5.2) could be used to describe the motion of rigid, cylindrical fibres by replacing r_e with an *equivalent ellipsoidal axis ratio* r_e^*, that is chosen by matching periods from experiments. While Jeffery's equation is a good approximation for rigid fibres, experiments establish that it cannot be applied to fibres that experience significant bending [Mas54]. As a consequence, much of the work on flexible fibres has focused on experimental observations of the periods and types of motion.

Forgacs et al. observed [FRM58] in experiments involving very dilute suspensions (with concentrations $\lesssim 0.01\%$) that fibres are essentially isolated. When subjected to laminar shear, fibres tend to orient themselves in the direction of the shear flow, and when in motion they either rotate in very well–defined orbits, or bend. Experiments by Mason and co–workers [Mas54, AFM58] identified a wide range of fibre behaviours, which they separated into distinct *orbit classes* based on the flexibility of the fibre. We have summarised the orbits in Table 5.1 which are two–dimensional in nature, since these are the ones that relate to our 2D fibre model.[a]

Rigid fibres (class I) rotate as solid cylinders, with an angular velocity that reaches a maximum when the fibre is aligned at right angles to the direction of the flow. Flexible fibres have several possible modes of rotation, the simplest of which is called a *springy rotation* (class II), where there fibre still revolves but deforms into the shape of an arc during the spin. In the *loop* or *S–turn* (IIIA) and *snake turn* (IIIB), the fibre is deformed into a more intricate curved intermediate shape, after which it straightens out once again.[b] The final class IV orbit indicates a fibre that performs a snake–like turn but never straightens out, continuing to loop over itself; this is called a *complex rotation*. Forgacs et al. [FRM58] used measurements of fibre flexibility to show that the various orbit classes occurred for different fibre stiffness values, with the stiffness decreasing as we

[a] There are several other types of orbit involving non–planar motions (such as spinning in the axial direction) that we haven't included here.

[b] The S–turn (class IIIA) is rarely observed in experiments except for very carefully chosen initial configurations and a fibre with a high degree of symmetry [AFM58].

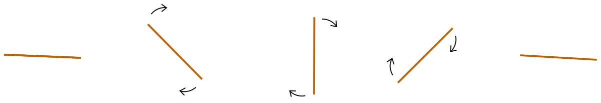
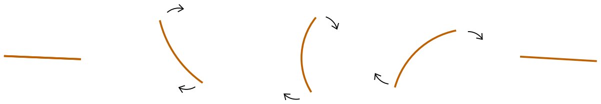
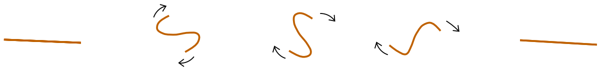
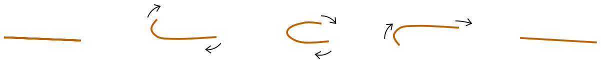
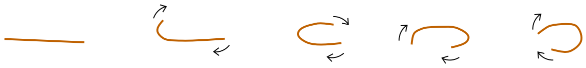
Orbit Class			
I	Rigid rotation		↓ decreasing stiffness increasing length & shear rate ↓
II	Springy rotation		
IIIA	Loop or S turn		
IIIB	Snake turn		
IV	Complex rotation		

Table 5.1: Typical orbit classes for rigid and flexible fibres immersed in a two–dimensional shear flow. Adapted from [FRM58, p. 124].

move down in the table. Differences also arise in orbital motion of fibres having the same physical properties when the shear rate and the length of the fibres is varied [FM59b].

Flexibility is thus a function of shear rate, bending stiffness and length. The effect of varying flexibility on orbits is easily compared (for moderately flexible fibres) by plotting the endpoints of a fibre on a polar plot, as illustrated in Figure 5.2. This plot is taken from experiments where the shear rate was varied, though the same behaviour has been observed in computations as bending stiffness is reduced [YM93]. Fibres that rotate as rigid cylinders generate a circular locus of points, while flexible fibres have orbits that are deformed.

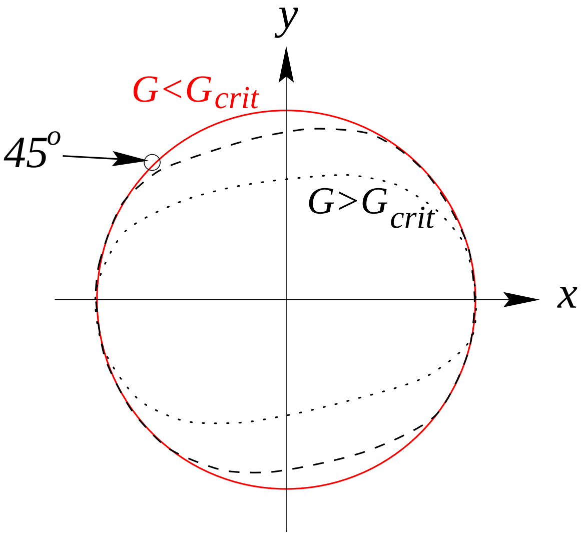


Figure 5.2: Polar plot of the endpoints of a fibre taken from experiments with varying shear rates. The outer circle corresponds to a rigid rod; the dotted line to a rod undergoing "springy" rotation; and the inner dotted curve to an even more flexible fibre. The deviation from the rigid rotation does not agree exactly with the predicted critical angle of 45°. Adapted from [FM59a].

A theoretical justification for this behaviour is provided by Forgacs & Mason [FM59a], who used Burger's theory for small disturbances in thin, slightly flexible rods to study the buckling of fibres. They showed that there is a critical shear rate,

$$G_{crit} = \frac{E\rho}{2\mu r_c^4}\left(\ell n(2r_c) - \frac{7}{4}\right),$$

where $r_c = \frac{L}{D}$ is the ratio of length L to diameter D of the cylinder, E is Young's modulus for the material, and μ is the fluid viscosity. From this, it is possible to derive an expression for the axial force on the rod as a function of the orientation angle φ:

$$F = \frac{\pi G \mu L^2 \sin\varphi \cos\varphi}{\rho\left(\ell n(2r_c) - \frac{7}{4}\right)}. \tag{5.3}$$

The compressive force on the fibre is a maximum at $\varphi = -45°$, which is where the onset of buckling can be expected to occur as $G \to G_{crit}$. Figure 5.2 shows that the predicted angle may be a reasonable approximation for shear rates near the critical value, but worsens as G is increased.

5.1.2 Computational approaches

The motion of flexible fibres in shear can be quite complicated, and the analytical results cannot capture the full range of complexity of observed orbits. Furthermore, due to the small size of the fibres and the difficult and time–consuming process of accumulating accurate flow measurements, there are considerable restrictions placed on the information that can be culled from experiments. Hence, numerical simulations present an ideal opportunity to gain a deeper understanding of flexible fibre motion by studying the fine structure of the fluid and fibre behaviour.

There have been several recent efforts to simulate fibre motion numerically. Yamamoto & Matsuoka [YM93] model a fibre as a chain of bonded spheres that are free to stretch, bend and twist relative to each other. In this model, there is no hydrodynamic coupling between fluid and fibre: the fluid undergoes a given linear shear, and the motion of the fibre is determined by solving a set of dynamic equations with a given applied fluid force. Links between the spherical elements are governed by three stiffness constants (for stretching, bending and twisting motions) whose values depend on the radii of the spheres and Young's modulus for the material. This work has since been extended to simulate large systems of particles [YM94] and also incorporates forces of attraction and repulsion between individual fibres.

Wherrett et al. [WGSO97] implemented a slightly modified version of the Yamamoto–Matsuoka model, which uses cylindrical elements instead of spheres. The stretching and bending stiffnesses are modified to include the aspect ratio of the elements, and the simulations are two–dimensional so that torsional motions are ignored. They derive a dimensionless *bending number*, related to the critical shearing force from (5.3), which is used to relate the changes in computed periods of revolution to fibre flexibility.

The work of Ross & Klingenberg [RK97] introduced another similar mechanical model, consisting of linked prolate spheroids. They eliminate axial stretching by linking the elements with *ball and socket joints* — real fibres do not stretch appreciably, even in highly sheared flows, and so this aspect of the model seems particularly advantageous.

In all of the work previously mentioned, the influence of the fibres on the fluid has been neglected. Another approach has been to treat the fibres as simple rigid rods and concentrate instead on the hydrodynamic coupling. Fibres have been treated using a rheological model for non–dilute fibre suspensions in [RA93] and [RDK90]. These models are able to compute changes in the velocity field and relative viscosity of the fluid due to the presence of many fibres. However, this approach captures only the averaged properties of the suspended particles, whereas the focus of our work is simulating the motion of individual fibres.

Our main purpose in reviewing the theoretical and experimental results above was to introduce several bases for comparison with our immersed boundary simulations of pulp fibres. Let us now summarise the results that are particularly useful to us. Rigid pulp fibres orbit as cylindrical rods with period given by (5.2), where r_e is replaced with an r_e^* calibrated with experiments. Flexible fibres do not obey Jeffery's equation, and have a lower period than their rigid counterparts. Orbits and periods of rotation vary depending on flexibility, which is a function of the fibre length and bending stiffness, and the fluid shear rate. Because there are three independent parameters parameters involved in determining fibre flexibility, we next derive a single dimensionless measuring of fibre flexibility, which permits meaningful comparisons to be made between the various behaviours of flexible fibres.

5.2 A Non–dimensional Measure of Fibre Flexibility

Consider a flexible, cylindrical fibre immersed in a shear flow, which causes the fibre to deform because of a *gradient* in the shear force, F_s, applied by the fluid along the fibre. The strength of the force gradient determines the fibre deflection, d (refer to Figure 5.3). The extent of bending is also affected by the length of the fibre, L, since the *applied bending moment* is proportional to $F_s \cdot L$. Deflection and fibre length are therefore central to the flexibility of the fibre, and so we consider the following dimensionless quantity to compare the behaviour of flexible fibres under various conditions:

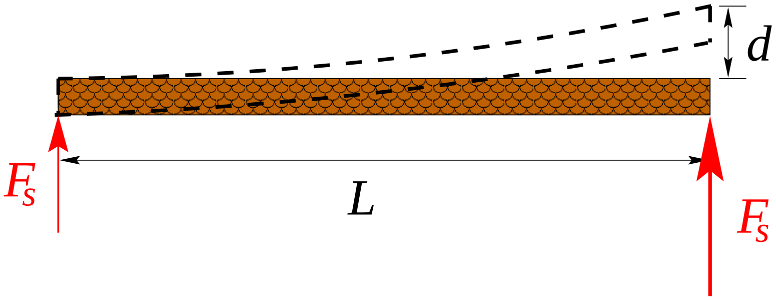


Figure 5.3: Deflection of fibre by a shearing force.

$$\chi = \frac{\textit{deflection of fibre}}{\textit{length of fibre}} = \frac{d}{L}. \tag{5.4}$$

This choice of parameter seems reasonable since fibres with the same flexibility should also have similar geometry (that is, the same deflection to length ratio).

We now need to express the deflection of the fibre in terms of quantities that can be measured in simulations. In the previous section, we saw that flexibility is dependent on fibre length and stiffness, and shear rate, which we may write as $\chi = f(L, EI, G)$. In order to determine the functional dependence of χ on these parameters, we appeal to basic structural mechanics [Hea77] in which the deflection of a rod of length L, clamped at one end and subjected to a force F_s at the free end, is given by

$$d \propto \frac{F_s L^3}{EI},$$

where E is Young's modulus for the material (units $g/cm \cdot s^2$) and I is the moment of area (units cm^4) in the plane of bending. The product EI (with units $g\,cm^3/s^2$) has appeared in the pulp and paper literature [DK81] as the quantity S, and is called the *stiffness* of the material.[c] We can now write

$$\chi \propto \frac{F_s L^2}{EI};$$

since it is the shear force gradient (which we denote ∇F_s) that gives rise to the bending moment in the fibre, we rewrite this expression as

$$\chi \propto \frac{\nabla F_s \cdot L^3}{EI}. \tag{5.5}$$

The above formula now contains all the information that is required about the elastic properties of the fibre material; we now need to determine the dependence on the hydrodynamic parameters through ∇F_s. To do so, we appeal to the derivation in Batchelor [Bat70] of the drag force on a solid sphere (or infinite cylinder) immersed in fluid. The geometry is clearly not the same, but since we are only interested in the dependence of F_s on the parameters and not the constants of proportionality, this will prove sufficient for our purposes. The drag force on a body with cross–sectional area D^2 is given by the equation

$$F_s \propto \rho U^2 D^2 C_D,$$

where C_D is the drag coefficient and U is the fluid velocity. The drag coefficient behaves very differently depending on whether flow is at high or low Reynolds number. If $\mathcal{R}e \lesssim 1$ then $C_D \propto \mathcal{R}e^{-1}$, while for large Reynolds number ($\mathcal{R}e \gtrsim 100$) C_D is effectively constant (see Figure 5.4). If we consider parameters that are typical of the paper–making process, we can compute a maximum Reynolds number of approximately 2.0, which places pulp within this low Reynolds number regime.

[c] The fibre force parameter σ appearing in the Immersed Boundary Method has the same interpretation as Young's modulus.

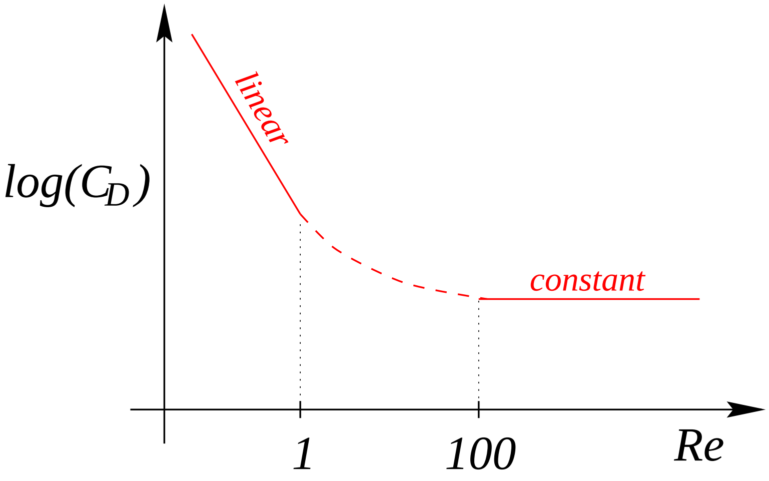


Figure 5.4: The dependence of the drag coefficient on Reynolds number for low ($\mathcal{R}e \lesssim 1$) and large ($\mathcal{R}e \gtrsim 100$).

When the Reynolds number is small, the fluid undergoes what is known as a *creeping flow* (see [Whi74]), in which the drag coefficient varies inversely with $\mathcal{R}e$:

$$\begin{aligned} F_s &\propto \rho U^2 D^2 \left(\frac{\rho U D}{\mu}\right)^{-1} \\ &\propto \mu D U. \end{aligned}$$

We are only interested in the force gradient, and so we divide this expression by L to obtain

$$\nabla F_s \propto \mu D G,$$

where $G = U/L$ is the velocity shear rate (units s^{-1}). Finally, we substitute the expression for ∇F_s into (5.5) to get

$$\boxed{\chi \propto \frac{\mu D G L^3}{EI} \qquad \text{low } \mathcal{R}e\text{, 3D}} \tag{5.6a}$$

The moment of area used above is a three–dimensional quantity; we can derive an expression for χ that is relevant to our two–dimensional fibre by using the 2D equivalent of the moment of area, I_2 (which absorbs the factor of D from the numerator), giving

$$\boxed{\chi \propto \frac{\mu G L^3}{EI_2} \qquad \text{low } \mathcal{R}e\text{, 2D}} \tag{5.6b}$$

We can apply a similar argument for the high Reynolds number case, where a boundary layer forms on the forward side of the body and the drag coefficient subsequently drops off to a constant value. In this flow regime,

$$F_s \propto \rho U^2 D^2 \qquad \text{and} \qquad \nabla F_s \propto \rho D^2 L G^2,$$

which after substituting into (5.5) yields

$$\boxed{\chi \propto \frac{\rho G^2 D^2 L^4}{EI} \qquad \text{high } \mathcal{R}e, \text{ 3D}} \tag{5.6c}$$

and

$$\boxed{\chi \propto \frac{\rho G^2 L^5}{EI_2} \qquad \text{high } \mathcal{R}e, \text{ 2D}} \tag{5.6d}$$

The quantity (5.6b) has appeared before as a *dimensionless shear rate* in [RK97], and its reciprocal as a *bending number* in [WGSO97]. The latter utilised the bending number to compare qualitative behaviour of fibres, and we will draw a similar comparison for the situation where hydrodynamic interactions between fluid and fibre are included. The two–dimensional versions of χ will be utilised in the pulp fibre simulations later in this chapter to separate between the various regimes of fibre motion.

5.3 The Immersed Boundary Method Applied to Pulp Fibres

From the discussion of Section 5.1.2, there is an obvious gap in the computational work on pulp fibres; namely, in the simulation of the hydrodynamic interaction between individual pulp fibres with the surrounding fluid. There is good agreement between theory and experiment for rigid fibres, and so it is unlikely that hydrodynamic coupling has a significant effect. However, the same cannot be said of flexible fibres, and it is here that the Immersed Boundary Method can make a significant contribution.

The method seems particularly well–suited to the simulation of flexible pulp fibres. The typical assumptions made in analytical and numerical investigations of pulp fibres are that the fluid is Newtonian and incompressible, and that the fibres are massless and neutrally–buoyant. The flow conditions under which individual fibres are considered typically correspond to very low Reynolds number. Furthermore, aspect ratios are very large, so that fibres are nearly one–dimensional structures. Taken together, these are precisely the assumptions we made in Chapter 2 for immersed boundaries.

The Immersed Boundary Method has several advantages over the other approaches described in Section 5.1. The action of the fibre through the fluid force term is actually quite simple in comparison to some of the mechanical models of flexible fibres. In addition, the method handles the coupling of fluid and fibre interactions very efficiently using a fast solver.

While the remainder of this chapter will concentrate on two–dimensional simulations of isolated pulp fibres, the Immersed Boundary Method also presents great potential for future applications in many other aspects of fibre motion. The extension of the fluid solver to three dimensions is straightforward. Many three–dimensional immersed boundaries (such as heart valves, arteries, etc.) require elaborate constructions of interwoven fibres. In our application, we have the advantage that in three dimensions, pulp fibres can be described very naturally as isolated immersed fibres, and no such complicated fibre constructions are necessary. Extensive immersed boundary computations of multi–particle systems have already been performed by Peskin & Fogelson, who remarked that they could perform simulations of 1000 or so particles, with the advantage of the Immersed Boundary Method being that the computational work increases only *linearly* with the number of particles [FP88b]. It should be possible to apply a similar technique to dilute suspensions of pulp fibres.

We now move on to a description of some of the details of implementing pulp fibres in the immersed boundary framework. There are essentially three main differences specific to the pulp fibre model from what we have seen in previous chapters:

1. the simple linear force considered earlier is modified to handle stretching[d] and bending stiff-

[d] While actual pulp fibres *do not* stretch appreciably, the Immersed Boundary Method does not easily gen-

ness;

2. the non–linearity subsequently introduced into the forcing function precludes the use of existing implicit solvers (such as the Mayo–Peskin scheme), and so limits us for the moment to explicit time stepping; and

3. the new geometry particular to a shear flow requires modifications to the fluid solver.

We will model the fibre force using a framework that has already been employed by Fauci and others [FP88a, FF93, DFG95] to simulate the motion of biological structures, such as cell walls, that resist bending. The force density function that we have seen so far resists axial stretching and compression, and now must be modified to take into account bending–resistant forces. An alternate way of specifying the force density at a particular fibre point $\vec{f}_\ell$ is to write it as the gradient of a potential function $\mathscr{E}(\ldots, \vec{X}_\ell, \vec{X}_{\ell+1}, \ldots)$

$$\vec{f}_\ell = -\frac{\partial \mathscr{E}}{\partial \vec{X}_\ell}.$$

Contributions to the force arising from stretching–resistant links between successive fibre points can be considered as arising from the following potential:

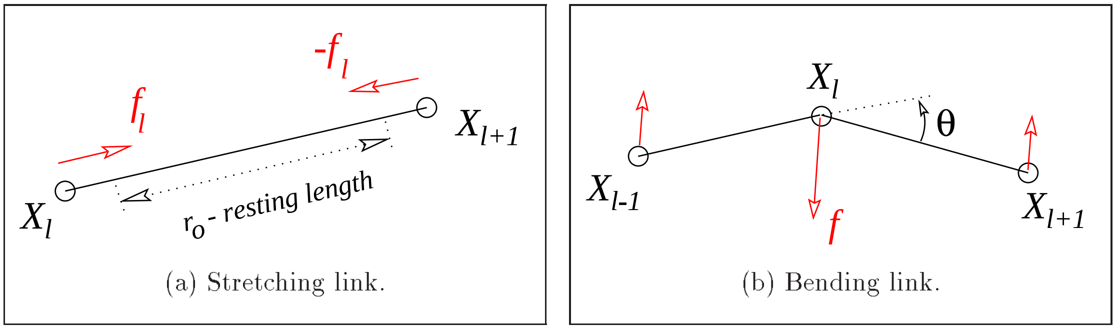
$$\mathscr{E}_s = \frac{1}{2} \sum_{\ell=0}^{N_b-1} \sigma_s \left(\|\vec{X}_{\ell+1} - \vec{X}_\ell\| - r_o \right)^2, \tag{5.7}$$

where σ_s is the stretching stiffness, and r_o is the resting length of the link joining each pair of points (we have chosen r_o equal to the fibre mesh spacing, h_b). Each term in the sum represents a spring–like "link" between two neighbouring points on the fibre. This can be seen by differentiating the sum at $\vec{X}_\ell$, which leads to two contributions to the force at point ℓ of the form:

$$\sigma_s \left(\|\vec{X}_{\ell+1} - \vec{X}_\ell\| - r_o \right) \frac{(\vec{X}_{\ell+1} - \vec{X}_\ell)}{\|\vec{X}_{\ell+1} - \vec{X}_\ell\|}.$$

Written in this manner, the force is clearly like that of a spring with resting length r_o and stiffness σ_s, directed along the vector joining $\vec{X}_\ell$ and $\vec{X}_{\ell+1}$. Figure 5.5(a) pictures a link of this type and the forces arising at each of the two points involved.

eralise to allow fibres to have a fixed distance between points. We have instead employed an approach that has been used in modeling biological filaments [FP88a] where fixed length structures are allowed to stretch, but the axial deformation is kept to a minimum by specifying a relatively large stretching stiffness constant.



(a) Stretching link.

(b) Bending link.

Figure 5.5: Two types of links are used to model flexible fibres: stretching/compression–resistant links between pairs of points; and bending–resistant links between triplets of points.

The bending–resistant links, on the other hand, can be incorporated using a force that drives the angle between successive triplets of points to a given *equilibrium angle* θ_0. An energy function that accomplishes this is the following

$$\mathscr{E}_b = \frac{1}{2}\sum_{\ell=1}^{N_b-1} \sigma_b \underbrace{\left[\hat{z}\cdot\left(\vec{X}_\ell - \vec{X}_{\ell-1}\right)\times\left(\vec{X}_{\ell+1}-\vec{X}_\ell\right) - r_o^2\sin\theta_o\right]}_{*}^2, \tag{5.8}$$

where $\hat{z} = (0,0,1)$, and $r_o^2 \sin\theta_o$ is related to the *equilibrium curvature* of the fibre[e]. To model a rigid rod, we select $\theta_o = 0$ for each link. The cross product term labeled "$*$" in the equation above may be rewritten as

$$\|\vec{X}_\ell - \vec{X}_{\ell-1}\| \cdot \|\vec{X}_{\ell+1} - \vec{X}_\ell\| \, \sin\theta - r_o^2\sin\theta_o,$$

which is approximately $r_o^2(\theta - \theta_o)$ when the fibre is close to equilibrium; hence, this contribution to the energy function serves to drive the angles between neighbouring links to θ_o. The stretching and bending forces given in (5.7) and (5.8) are very similar to that used in the mechanical models mentioned in the previous section.

The energy function describing a flexible fibre is now given by

$$\mathscr{E} = \mathscr{E}_s + \mathscr{E}_b.$$

[e] It is actually the quantity $\sin\theta_o/r_o$ that has the interpretation of curvature (see [FP88a, p. 90-92] for a full discussion).

The force densities that are needed in the Immersed Boundary Method are programmed within the numerical code as derivatives of the above expressions. Calculating the force at a given fibre point requires that the contributions from each link involving that point be added together. The number of links contributing to the force density at a particular fibre point $\mathcal{X}_\ell$, $\ell = 0, 1, 2, \ldots, N_b$, depends on whether the point is in the middle of the fibre or near the endpoints (see Figure 5.6):

- the force at an endpoint ($\ell = 0$ or N_b) receives contributions from only one stretching link and one bending link;
- the points next to endpoints ($\ell = 1$ or $\ell = N_b - 1$) each belong to two stretching links and two bending links;
- all remaining points have the force computed from two stretching links and three bending links.

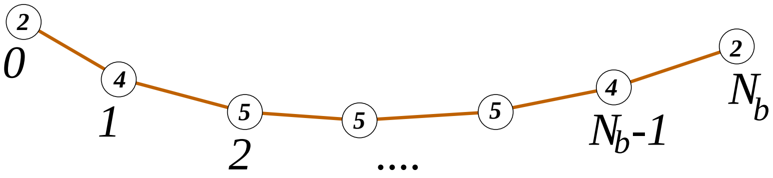


Figure 5.6: Labeling of discrete fibre points along with the total number of links at each point (circled).

An important consequence of this choice of energy function is that the force is now a *non–linear function* of the fibre positions $\mathcal{X}_\ell$. As a result, the Mayo–Peskin iterative scheme cannot be applied to these problems, since it is built around the assumption that the force takes on the simple linear form (2.7). Unfortunately, this means that we are reduced to handling the force explicitly, or at the very best, we can use the *approximate implicit* approach, wherein only the fibre position and force are coupled together in a fixed point iteration, as described on page 84.

The final aspect of the immersed boundary implementation for pulp fibres is the modification of the fluid solver to handle a shear flow. We modify the domain and boundary conditions for the

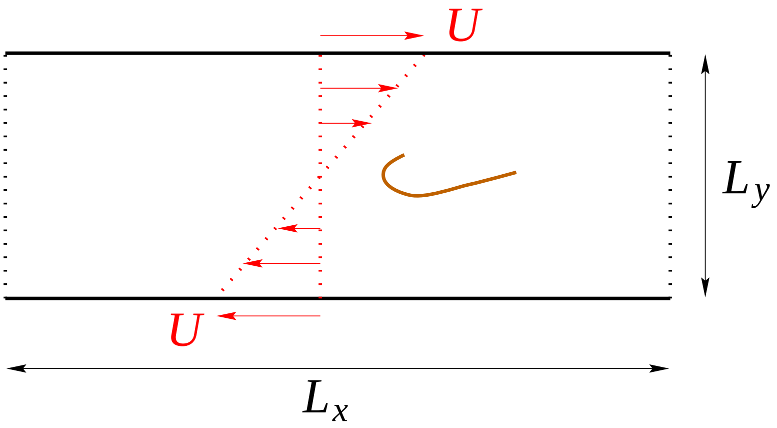


Figure 5.7: The periodic channel domain for the pulp fibre simulations, with shearing motion induced by moving top and bottom walls.

doubly–periodic box to the channel pictured in Figure 5.7. The channel has dimensions $L_x \times L_y$, and is periodic in the x–direction. L_x is chosen larger than L_y so that the effects of periodic boundaries can be minimised. The top and bottom walls are moved with constant velocities U in opposite directions, so that the shear rate is $G = 2U/L_y$. We discretise the domain as before, choosing L_x and L_y so that the fluid mesh spacing $h = \frac{L_x}{N_x} = \frac{L_y}{N_y}$ in each direction, and $h_b = \frac{L}{N_b}$ for the fibre. If we further restrict the dimensions of the domain so that N_x is an integer power of 2, then an FFT algorithm may still be applied to solve the pressure Poisson equation. The method must be modified somewhat to account for the change in aspect ratio and non–zero boundary conditions at the top and bottom of the domain. Essentially, the modified method involves an FFT in the x–direction only, which gives rise to a banded system of equations to be solved for the transformed variables in the y–direction. The pressure is then found by transforming back to real variables by an inverse FFT. The details are described in Appendix A.

5.4 Numerical Simulations

Our main purpose in this chapter is to demonstrate that the Immersed Boundary Method is a useful tool for simulating the motion of pulp fibres. To this end, we present comparisons with experimental and theoretical results — both qualitative and quantitative — to illustrate that the computed results are physically reasonable. We also use the non–dimensional fibre flexibility measure χ to show that the qualitative behaviour of fibres for low $\mathcal{R}e$ is captured well over a large range of parameters.

Experiments are often performed on synthetic fibres made of rayon or dacron, immersed in highly viscous fluids such as corn syrup or castor oil [FM59b]. Representative values of parameters in experiments are listed in Table 5.2, with references to the literature where appropriate. While the physical parameters corresponding to some experiments differ significantly from those for actual pulp fibres, the observed behaviour is very similar. Therefore, we will perform simulations on parameters for both situations whenever possible in order to cover as wide a range of physics as we can, within the stability constraints set by the numerical scheme. The Immersed Boundary Method is limited to low Reynolds numbers ($\mathcal{R}e \lesssim 100$), and so we will perform most simulations for highly

Parameter	*Values*	*Units*	*References*
ρ (density)	1.0	g/cm^3	
μ (viscosity)	10–90 (castor oil/corn syrup) 0.01 (water)	$g/cm \cdot s$	[FM59a], [FM59b]
G (shear rate)	1–100 (experiment)	s^{-1}	[FM59a], [FM59b]
EI (bending stiffness)	0.001 – 0.07 (paper pulp) 0.6 (nylon)	$g\,cm^3/s^2$	[DK81], [DK82], [Sam63]
L (fibre length)	0.1 – 0.3	cm	[DK81], [WGSO97], [FM59a]
r_c (aspect ratio)	10–60 (natural) 40–400 (synthetic)		[WGSO97] [FM59a],[FM59b]
$\mathcal{R}e$ (Reynolds number)	0.01–50	—	

Table 5.2: Parameter values used in simulations, derived from a range of sources both experimental and computational.

viscous fluids and moderate shear rates, which are typical of the experiments just mentioned.

Our computational test chamber was taken in almost all cases to be a rectangle of dimensions $2 \times \frac{1}{2}$ cm, within which was suspended a fiber of length $0.1 - 0.2$ cm. We concentrate mainly on the effects of shear rate (which has typically been the variable quantity in experiments) and bending stiffness, since both can be changed easily without modifying the computational domain. The problem was discretised with a mesh spacing of $h = \frac{1}{64}$ cm (that is, 128×32 fluid grid points) and either 40 or 80 fibre points (depending on whether the fibre is 0.1 or 0.2 cm long)[f]. The time step k required for stability, using the fourth order Runge–Kutta time-stepping scheme, lies in the range $2.0 - 5.0 \times 10^{-5}$. The bending stress parameter σ_b has the same interpretation as Young's modulus E; this quantity is chosen so that when scaled by an appropriate moment of area, the resulting product $\sigma_b \cdot I$ lies in the range 0.001–1.0 $g\, cm^3/s^2$. There is no physical equivalent for the stretching stiffness σ_s, since pulp fibres do not stretch appreciably; consequently, we chose a value large enough (typically from 5,000–10,000) so that the fibre length was held to within 2% of its initial value throughout most simulations.

We begin by comparing the qualitative behaviour of solutions for four choices of bending stiffness that reproduce the orbit classes pictured earlier in Table 5.1. Time sequences from the simulations are given in Figure 5.8 for EI lying between 0.006 and 0.5. The other parameters were chosen to be $G = 10$, $L = 0.1$, and $k = 5 \times 10^{-5}$, except for the first set of images where the stretching stiffness restricted the time step to half that size. The fibre position was initially specified as an arc of a circle — we started with a slight curvature so that the various orbits would develop within a reasonable amount of time (though this was not necessary, if we were willing to wait long enough).

By comparing the images up to time $t = 0.09$ s, we can see that the flexible fibres complete

[f] The mesh spacing and domain size were chosen so as to minimise the effect of boundaries on the solution while at the same time keeping computational cost to a minimum. We performed various tests that showed for $h = \frac{1}{64}$, the domain could be taken as small as $2 \times \frac{1}{2}$ without appreciably changing the qualitative behaviour of the computed solution for a fibre of length $0.1 - 0.2$ cm.

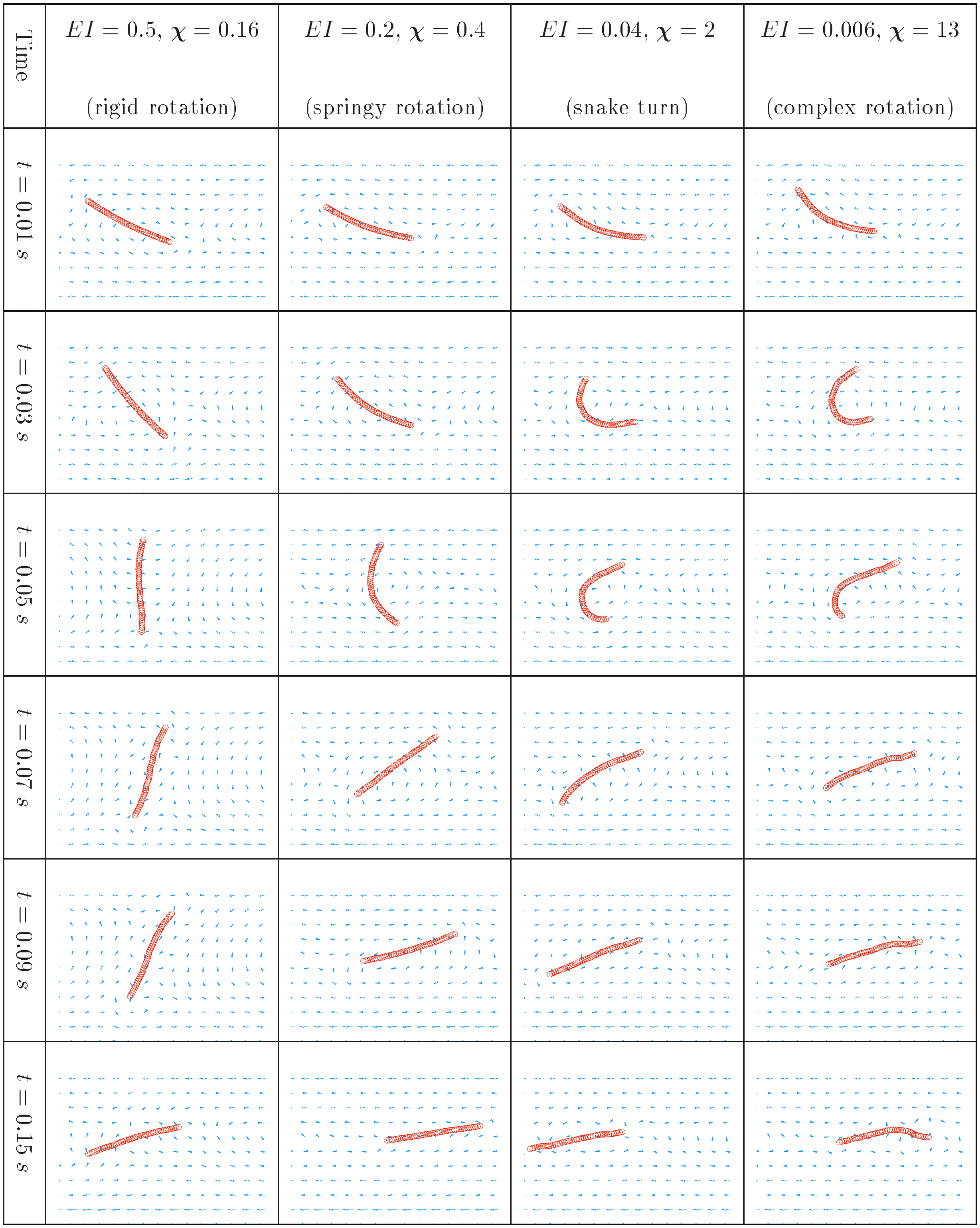


Figure 5.8: Time sequences of orbits at times 0.01, 0.03, 0.05, 0.07, 0.09 and 0.15.

their first half–rotation in a significantly shorter time than the rigid fibre. This is something that has been observed in experiments [AFM58].

Something which is not apparent from these images is that after completing the loop, the fibres in the first three orbits spend a great deal of time near the horizontal. This is consistent with the theoretical orbits for rigid fibres given by Jeffery's equation (5.1); plots of the orientation angle (the angle between the endpoints) versus time look very similar to that pictured in Figure 5.1 for rigid ellipsoids. The fourth fibre never straightens out, and hence its classification as a "complex rotation" — the period of rotation is significantly smaller and the fibre begins another turn very shortly after $t = 0.15$ s. The other fibres eventually pass through $\varphi = 90°$ as well, and begin a second loop that is essentially a mirror image of the first, with the period of rotation decreasing as the fibre stiffness decreases.

A very useful way to compare orbits is to plot the endpoints of the fibre in a reference frame where the centre of the fibre is fixed; Figure 5.9 compares the four orbits we just discussed. The shape of the orbits appears very similar to that observed in experiments, where varying shear rate was used to change the fibre flexibility (refer to the plot in Figure 5.2). Just as was observed in experiments, the fibre begins to buckle before the theoretically–predicted critical bending angle of $\varphi = 45°$.

We can draw a more quantitative comparison with the theoretical predictions in terms of the amount of time the fibre spends at each angle φ. We ran another series of computations with bending stiffness fixed at $\sigma_b \cdot I = 0.01$ and the shear rate taken between 50 and 80. All fibres underwent snake turns, and we computed for a period of time comprising at least four complete rotations. The probability distribution of φ is plotted in Figure 5.11 at open points. We used the average computed period of rotation, along with the formula in (5.2), to come up with an equivalent ellipsoidal axis ratio, r_e^*, for each of the four cases. We then calculated the corresponding predicted distributions of φ from (5.1), which are plotted as solid curves on the same set of axes. From this, we can draw the following comparisons:

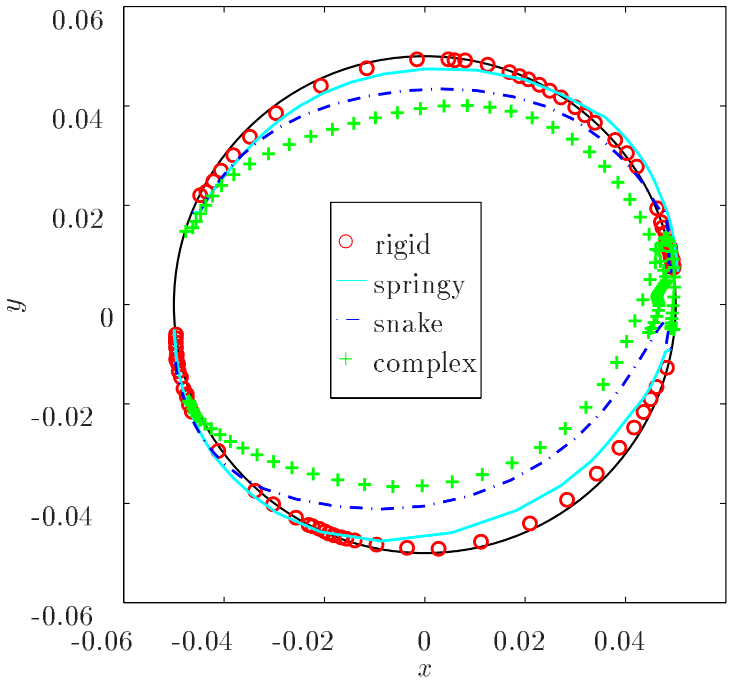


Figure 5.9: Comparison of the curves traced by the endpoints of the fibre for the various orbits classes. Rigid fibres trace a circular orbit, with the curves deforming more as the fibre flexibility is increased. These are the same orbits as pictured in Figure 5.8.

- The fibre spends the majority of its time near the horizontal, with the proportion of time increasing as the shear rate is decreased (that is, for more rigid fibres).
- Unlike the theoretical distributions and simulations which ignore hydrodynamic interactions (such as [YM93, Fig. 10]), the distribution *is not symmetrical about* $\varphi = 90°$. Rather, there is a tendency for the fibre to remain at an angle slightly above the horizontal plane; we believe that this is due to the interaction between fibre and fluid which is not included in either previous computations or the analytical formulae. Though the fibre remains approximately flat when stalled in the stream–wise direction, it continually undergoes very small flexing motions which cause the streamlines to curve slightly upward into the upper half of the channel before the fibre reaches $\varphi = 90°$ (see Figure 5.10). This appears to be enough to cause the slight skewness in angle distribution observed here, and is something that we observe in all simulations over a wide range of parameter values.

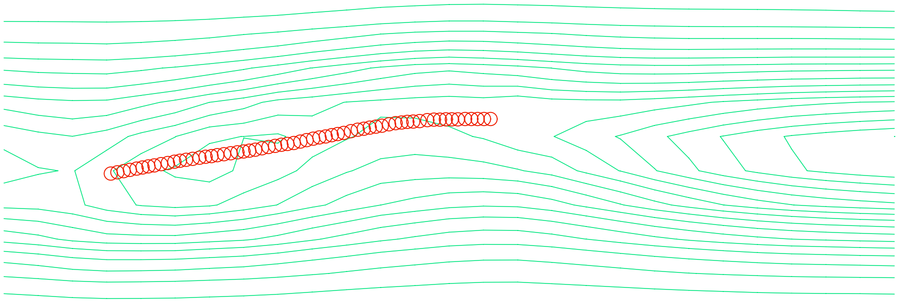


Figure 5.10: Flow streamlines for a fibre stalled at an angle $\varphi > 0$. The streamlines are deformed near the fibre, and there are narrow zones of recirculation to the front and rear. Note that even though the instantaneous streamlines cross the fibre, no fluid flows passes through since the fibre is moving with the fluid.

- Disregarding the slight offset of the curve near the peak, the actual size and shape of the distribution is very similar between the computed and predicted curves.

We have run a large number of simulations with varying fibre length, bending stiffness, shear rate and viscosity. The resulting orbit classifications have been plotted in Figure 5.12 in terms

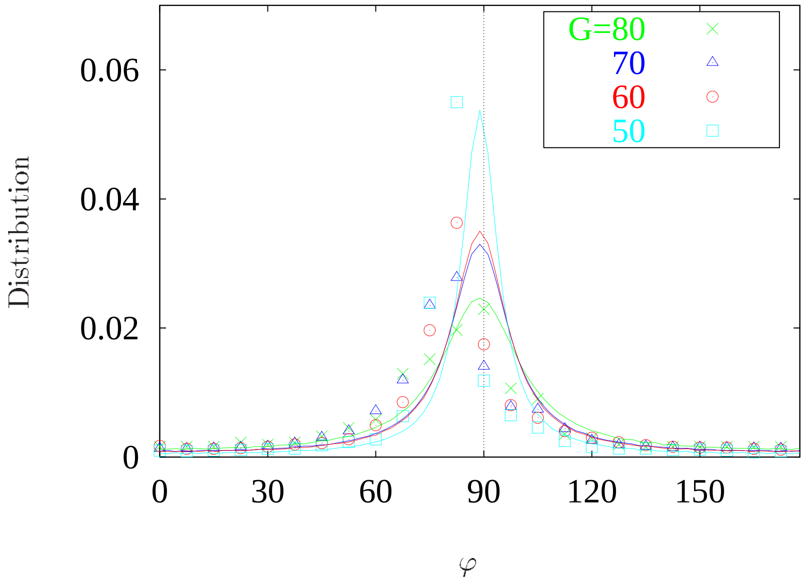


Figure 5.11: Distribution of time spent at various angles throughout the motion of a fibre undergoing springy rotation. The shear rate is varied from 50 to 80. The points represent the computed angle distributions, while the solid lines are the corresponding theoretical predictions from Jeffery's equation (5.1) (with an axis ratio r_e^* chosen to match the observed average period).

of the non–dimensional flexibility measure χ and the bending stiffness EI. We have classified

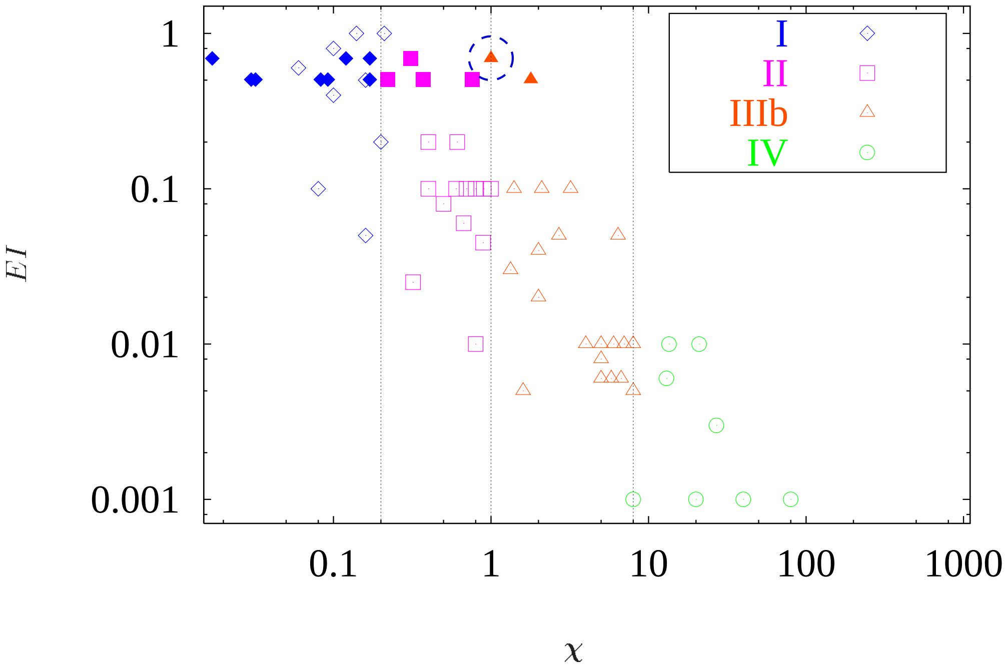


Figure 5.12: Comparison of the orbit class with bending stiffness and χ. The fibre length, shear rate and viscosity are also varied, which accounts for the spread of the data from a straight line. The computed orbits are plotted with open points; experiments from [FM59b, Table III] are plotted as solid points for comparison.

each computed orbit as belonging to either class I, II, IIIB or IV, using a different shape of open point for each (class IIIA was never observed in computations). Our criteria for judging the orbit class was based on the *exterior angle*, α, between the tangents at the endpoints of the fibre (see Figure 5.13):

I: If $175^\circ < \alpha < 180^\circ$, then the fibre was essentially rigid.

II: For $90^\circ < \alpha < 175^\circ$, the ends of the fibre always deformed in unison to induce a springy rotation.

IIIb: When $\alpha < 90°$, the ends of the fibre tended to begin moving independently of each other, leading to a snake turn. This independence of the motion of fibre ends was the same criterion used in [FM59b] to identify snake turns, although the observation that the division occurred at an angle of approximately 90° was not.

IV: When the fibre never straightened out, the orbit was classified as a complex rotation.

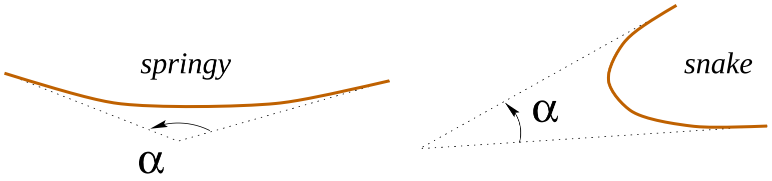


Figure 5.13: Definition of the exterior angle α, measured between the ends of a flexible fibre.

There is a clear division of the orbit classes, which have been drawn as vertical lines at values of $\chi \approx 0.2$, 1.0 and 8. This is very strong evidence of our premise that χ is a useful measure of fibre flexibility.

To push the comparison even further, we have included on the same set of axes a sequence of solid points which were taken from experiments by Forgacs & Mason [FM59b], performed with dacron and rayon filaments suspended in corn syrup or castor oil. In order to ensure that the scaling between experimental and computational results is the same, we have adjusted the parameter χ based on a single experimental data point (circled in Figure 5.12) which was classified as lying on the borderline between a springy rotation and a snake turn: the value of χ was set to equal 1.0 for this experiment, and all other experimental points were scaled by the same factor. The line $\chi = 0.25$ captures the division of experimental values between rigid and springy orbits very well, and so it appears that the computational model predicts quite well the qualitative behaviour of fibre orbits observed in experiments.

These results verify that the Immersed Boundary Method can indeed be used to simulate the motion of flexible fibres at low Reynolds number. The qualitative behaviour of fibre orbits is very similar to what is observed in experiments, both in terms of the orbit classification and the distribution of angular displacement throughout the orbital period.

Chapter 6
Conclusions and Future Research

The outcome of any serious research can only be to make two questions grow where only one grew before.

— Thorstein Veblen.

While the focus of much of the first part of this work was analytical, the results were always interpreted in terms of their application to the Immersed Boundary Method. In the process, we gained a deeper understanding of the behaviour of solutions to the equations of motion governing immersed fibres, while at the same time making suggestions for improvements to the numerical method that handle stiffness and increase spatial and temporal accuracy. In the end, we contend that we have an analytical tool that can be used to develop improved iterative solution schemes and test their convergence behaviour *a priori*.

We will first summarise the main conclusions that were drawn from our linearised analysis. The highlights of the two–dimensional pulp fibre simulations will then be given, which demonstrate great potential as a starting point for computations of three–dimensional pulp suspensions. Finally, we conclude with a description of several avenues of future research that have been opened up by this work.

6.1 Conclusions

We began with a linear modal analysis of two systems of equations for the motion for an immersed fibre: first, for the *jump formulation* of the original problem; and second, for a smoothed version of the *delta function formulation*, in which the delta functions are replaced by suitable approximations. The dispersion relation derived from both problems exhibited a set of discrete fibre modes, which arise due to the presence of the fibre. Furthermore, there is a clear separation in both cases between normal and tangential modes of oscillation; numerical experiments were performed to verify the presence of these decoupled modes in immersed boundary computations.

For the jump problem, we were able to push the analysis further and prove that the fibre modes are stable in time for all physically reasonable values of the parameters. We also showed that the decay rates and frequencies corresponding to these modes vary in magnitude over a much greater range of values than the modes of Stokes' equations without an immersed fibre. An asymptotic analysis of the decay rates identified precisely how the decay rates depend on the parameters, thereby recognising the tangential modes as the principal source of stiffness, while the normal modes are actually "perturbed" Stokes modes.

To see how well the smoothed problem approximates the original one, we compared the modes from the two problems, and found that while the range of decays rates is much smaller, the problem is still stiff. Furthermore, the lowest wavenumber mode (corresponding to the dominant solution features) matches quite well between the two problems, except when the force or the smoothing radius, ϵ, are very large. We employed careful numerical experiments on an example specially–tailored to the linear problem to verify the predicted solution behaviour.

For large forces, the smoothed problem exhibits oscillations with a frequency that is very large in relation to the decay rate — this corresponds to eigenvalues that cluster near the imaginary axis. We used stability diagrams to show analytically that the Runge–Kutta method of order four is a much better alternative among the class of fully explicit schemes, and leads to significant improvements in efficiency for explicit calculations at a high Reynolds number. Furthermore, the RK4 method is also very competitive with all of the implicit schemes that have been proposed to date.

The smoothed problem yielded one more piece of information of particular significance for the

Immersed Boundary Method. By looking at an asymptotic expansion of the dispersion relation as $\epsilon \rightarrow 0$, we evaluated the formal accuracy of various approximations to the delta function. In the setting of this "idealised discretisation," we investigated the link between satisfaction of moment conditions and the formal accuracy of the resulting interpolation. We derived a new delta function that satisfies the discrete second moment condition, for which a proof of formal second order accuracy has eluded us, but which exhibits improved accuracy in finely–resolved computations.

The high degree of stiffness inherent in immersed fibres severely limits the allowable time step, and so implicit methods must be considered if there is any hope of increasing the time step to a level that will allow finely resolved calculations. We extended our analysis to semi–discretisations of the immersed fibre problem, and examined two implicit schemes. One is a Crank–Nicholson–type scheme, which differs from the typical Immersed Boundary Method in that diffusion effects are coupled implicitly to the force within a fixed point iteration. The second is a method introduced by Mayo & Peskin, which uses an ADI step for diffusion, followed by a fixed point iteration on the fibre position, with a judicious choice of preconditioner.

In the time–discrete case, the analysis leads to predictions of the convergence rate of the fixed point iteration embedded within each time step. The theoretical predictions match extremely well with the convergence behaviour observed in calculations. Numerical experiments verify that the Crank–Nicholson scheme offers no advantage over a method that treats the force explicitly; this is not surprising, as all of our work to this point has shown that it is the fibre force that gives rise to the stiffness in the problem rather than the diffusive effects which are dominant in Stokes flow. The predicted convergence rates for the Mayo–Peskin scheme are virtually identical to those observed in computations over the entire physical range of forces.

One issue that has remained in the background throughout this thesis is the use of symbolic computation. MAPLE has proven to be an indispensable tool in the derivation of most of the results beyond Section 3.1. We have used it in its more mundane capacity as an algebraic manipulator, but also for generating C and FORTRAN code. Hence, our approach has really been a three–pronged one, combining mathematical analysis, symbolic computation and numerical experiments in an effort to deepen our understanding of the immersed fibre problem and the behaviour of numerical methods based on it.

The final chapter introduced a new application of the Immersed Boundary Method to simulating the flow of pulp fibres in two dimensions. This work is of particular interest to the paper–making industry as it is one of the first attempts to compute the hydrodynamic coupling between a flexible fibre and an incompressible fluid. We demonstrate that the method reproduces the tumbling motions of fibres observed experimentally in shear flows for reasonable physical parameters. Comparisons of rotation rates with theoretical predictions and experimental observations are also in very close agreement. The Immersed Boundary Method shows great promise as a quantitative tool in pulp fibre modeling.

6.2 Future Work

We hope to make a contribution to the development of more efficient iterative schemes that do a better job of combating the stiffness inherent in the problem. A fully non–linear Newton solver is far too expensive to implement (as demonstrated by Tu & Peskin [TP92]), while the most common approach of building a semi–implicit scheme around a fixed point iteration for the force does not go far enough in handling the stiffness effectively.

Our analysis clearly indicates a decoupling between normal and tangential modes of oscillation, with the primary contribution to the stiffness coming from the tangential modes. We believe that a “local linearisation” of the problem that singles out the stiff interfacial modes will lead to a better iterative technique, with a more effective preconditioning strategy. Once a fast and effective solver has been devised for coupling the fibre and fluid, we hope to incorporate it into a semi–implicit Runge–Kutta scheme (such as those described in [ARS97]), which will give us all the advantages of the better stability properties of Runge–Kutta schemes near the imaginary axis.

Further analysis can be performed on the Forward Euler/ADI scheme, since the results in

Section 4.1 show that the time step restrictions cannot be determined in a straightforward way from the effects of Stokes modes and fibre modes. This might be addressed, at least in part, by incorporating the effect of the ADI step into the analysis. Also, a further investigation of the split–step time discretisation underlying the Mayo–Peskin iterative scheme would show whether the failure of the iteration to converge is due to time–instability of the discretisation or spatially discrete effects, such as fibre mesh crossings.

Another interesting result in this thesis is the suggestion that the accuracy of the Immersed Boundary Method may be increased by a suitable choice of delta function. This is not a new idea, but the search for a better approximate smoothing function was seemingly abandoned after the one–dimensional analysis of Beyer & LeVeque [BL92]. We plan to continue our attempts to derive a dispersion relation for our "new" delta function that would lead to a formal proof of the accuracy of the interpolation. This improved delta function would also have application to many other numerical schemes that use delta function smoothing to handle immersed interfaces, including the level set method for incompressible flows [Hou95]; the particle–in–cell method [SB91, LIB95]; spectral and pseudo-spectral methods applied to particle suspensions [Yus96] and arterial flow [Art96]; and finite element simulations of fluid droplets [Tor96, TMSB97].

Our work on pulp fibres was really only a first step in modeling pulp suspensions, since the motion of fibres in shear flow is fundamentally three–dimensional. We plan to extend our immersed boundary code to 3D and include torsional stiffness in the fibre in order to capture a wider range of complex fibre motions.

Our pulp fibre simulations to this point have neglected fibre inertia, which in some situations is considered an important factor. The mass of particles that are not neutrally–buoyant can be accounted for in the immersed boundary model by including a variable density in the momentum equations [PM93a]. Each fibre contributes a singular mass distribution to the fluid of the following form

$$\rho(\boldsymbol{x}, t) = \rho_o + \int_\Gamma m(s) \cdot \delta(\boldsymbol{x} - \boldsymbol{X}(s,t))\, ds,$$

where $m(s)$ is the *additional mass per unit length* of the fibre (which can be negative), and ρ_o is the constant fluid density in the absence of the fibres. A variable density precludes the use of an FFT solver, and so this extension will require development of an alternate fast fluid solver.

The Immersed Boundary Method has proven very effective in modeling the flow of suspensions that contain on the order of 100–1000 particles in the platelet aggregation studies of Fogelson & Peskin [FP88b]. These authors also incorporate particle–particle interactions using appropriate modifications to the force in the fluid equations. By modifying the inter–particle force to conform with the physics of pulp fibre interaction using the previous work of Doi & Chen [DC89] and Yamamoto & Matsuoka [YM94], we plan to investigate "semi–dilute" suspensions where aggregation of fibres (known as *"flocculation"*) is an important factor. This will significantly increase the range of flow regimes in the paper–making process that can be investigated using the Immersed Boundary Method.

Bibliography

[AFM58] A. P. Arlov, O. L. Forgacs, and S. G. Mason. Particle motions in sheared suspensions 4. General behaviour of wood pulp fibres. *Svensk Papperstidning*, 61(3):61–67, 1958.

[AM67] E. Anczurowski and S. G. Mason. The kinetics of flowing dispersions. II. Equilibrium orientations of rods and discs (experimental). *Journal of Colloid Science*, 23:533–546, 1967.

[ARS97] Uri M. Ascher, Steven J. Ruuth, and Raymond J. Spiteri. Implicit–explicit Runge–Kutta methods for time–dependent partial differential equations. *Applied Numerical Mathematics, Special Issue on Innovative Time Integrators*, 1997. To appear.

[Art96] Kayne Marie Arthurs. *Flow regulation in the afferent arteriole: An application of the Immersed Boundary Method*. PhD thesis, Duke University, 1996.

[Bat70] G. K. Batchelor. *An Introduction to Fluid Dynamics*. Cambridge University Press, Cambridge, 1970.

[Ber82] Marsha J. Berger. *Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations*. PhD thesis, Stanford University, 1982.

[Bey92] Richard P. Beyer, Jr. A computational model of the cochlea using the immersed boundary method. *Journal of Computational Physics*, 98:145–162, 1992.

[BL92] Richard P. Beyer and Randall J. LeVeque. Analysis of a one-dimensional model for the immersed boundary method. *SIAM Journal on Numerical Analysis*, 29(2):332–364, 1992.

[Bot96] Dean C. Bottino. *An Immersed Boundary Model of Ameboid Deformation and Locomotion*. PhD thesis, Tulane University, 1996. [`http://www.math.utah.edu/~bottino/bottino.thesis.ps`].

[But96] J. C. Butcher. A history of Runge–Kutta methods. *Applied Numerical Mathematics*, 20:247–260, 1996.

[C^+91] Bruce W. Char et al. *Maple V Language Reference Manual*. Springer-Verlag, 1991.

[Cho68] Alexandre J. Chorin. Numerical solution of the Navier–Stokes equations. *Mathematics of Computation*, 22:745–762, 1968.

[DC89] M. Doi and D. Chen. Simulation of aggregating colloids in shear flow. *Journal of Chemical Physics*, 90(10):5271–5279, 1989.

[DFFG96] Robert Dillon, Lisa J. Fauci, Aaron L. Fogelson, and Donald Gaver III. Modeling biofilm processes using the immersed boundary method. *Journal of Computational Physics*, 129(1):57–73, 1996.

[DFG95] Robert Dillon, Lisa Fauci, and Donald Gaver III. A microscale model of bacterial swimming, chemotaxis and substrate transport. *Journal of Theoretical Biology*, 177:325–340, 1995.

[DK81] P. A. Tam Doo and R. J. Kerekes. A method to measure we fiber flexibility. *Tappi Journal*, 64(3):113–116, 1981.

[DK82] P. A. Tam Doo and R. J. Kerekes. The flexibility of wet pulp fibres. *Pulp and Paper Canada*, 83(2):46–50, 1982.

[ES86] David G. Ebin and Ralph A. Saxton. The initial value problem for elastodynamics of incompressible bodies. *Archive for Rational Mechanics and Analysis*, 94:15–38, 1986.

[ES87] David G. Ebin and Ralph A. Saxton. The equations of incompressible elasticity. In Barbara Lee Keyfitz and Herbert C. Kranzer, editors, *Nonstrictly Hyperbolic Conservation Laws, Proceedings of an AMS Special Session (Contemporary Mathematics, vol. 60)*, pages 25–34. American Mathematical Society, 1987.

[Fau93] Lisa J. Fauci. Computational modeling of the swimming of biflagellated algal cells. In A. Y. Cheer and C. P. van Dam, editors, *Fluid Dynamics in Biology: Proceedings of the AMS-IMS-SIAM Joint Summer Research Conference on Biofluiddynamics (Contemporary Mathematics, vol. 141)*, pages 91–102. American Mathematical Society, 1993.

[FF93] Lisa J. Fauci and Aaron L. Fogelson. Truncated Newton methods and the modeling of complex immersed elastic structures. *Communications on Pure and Applied Mathematics*, 46:787–818, 1993.

[FM59a] O. L. Forgacs and S. G. Mason. Particle motions in sheared suspensions: IX. Spin and deformation of threadlike particles. *Journal of Colloid Science*, 14:457–472, 1959.

[FM59b] O. L. Forgacs and S. G. Mason. Particle motions in sheared suspensions: X. Orbits of flexible threadlike particles. *Journal of Colloid Science*, 14:473–491, 1959.

[FM95] Lisa J. Fauci and Amy McDonald. Sperm motility in the presence of boundaries. *Bulletin of Mathematical Biology*, 57(5):679–699, 1995.

[Fog84] Aaron L. Fogelson. A mathematical model and numerical method for studying platelet adhesion and aggregation during blood clotting. *Journal of Computational Physics*, 56:111–134, 1984.

[Fog92] Aaron L. Fogelson. Continuum models of platelet aggregation: Formulation and mechanical properties. *SIAM Journal on Applied Mathematics*, 52(4):1089–1110, 1992.

[FP88a] Lisa J. Fauci and Charles S. Peskin. A computational model of aquatic animal locomotion. *Journal of Computational Physics*, 77:85–108, 1988.

[FP88b] Aaron L. Fogelson and Charles S. Peskin. A fast numerical method for solving the three-dimensional Stokes' equations in the presence of suspended particles. *Journal of Computational Physics*, 79:50–69, 1988.

[FRM58] O. L. Forgacs, A. A. Robertson, and S. G. Mason. The hydrodynamic behaviour of paper-making fibres. *Pulp and Paper Magazine of Canada*, 59(5):117–128, 1958.

[HB97] L. H. Howell and J. B. Bell. An adaptive-mesh projection method for viscous incompressible flow. *SIAM Journal on Scientific Computing*, 18(4), 1997. To appear. [`http://www.nersc.gov/research/CCSE/publications/hb96/hb96_abs.html`].

[Hea77] Edwin John Hearn. *Mechanics of Materials*, volume 19 of *International Series in Materials Science and Technology*. Pergamon Press, 1977.

[Hou95] Thomas Y. Hou. Numerical solutions to free boundary problems. In *Acta Numerica 1995*, pages 335–415. Cambridge University Press, 1995.

[Jef22] G. B. Jeffery. The motion of ellipsoidal particles immersed in a viscous fluid. *Proceedings of the Royal Society*, A102:161–179, 1922.

[Li97a] Zhilin Li. Immersed interface methods for moving interface problems. Preprint. [`ftp://www.math.ucla.edu/pub/zhilin/Papers/moving.ps.Z`], 1997.

[Li97b] Zhilin Li. A note on immersed interface method for three dimensional elliptic equations. Preprint. [`ftp://www.math.ucla.edu/pub/zhilin/Papers/3d.ps.Z`], 1997.

[LIB95] Giovanni Lapenta, Fujio Iinoya, and J. U. Brackbill. Particle-in-cell simulation of glow discharges in complex geometries. *IEEE Transactions on Plasma Science*, 23(4):769–779, August 1995.

[LL94] Randall J. LeVeque and Zhilin Li. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM Journal on Numerical Analysis*, 31(3):1019–1044, 1994.

[LL97] Randall J. LeVeque and Zhilin Li. Immersed interface methods for Stokes flow with elastic boundaries or surface tension. *SIAM Journal on Scientific Computing*, 18(3):709–735, 1997. [`ftp://amath.washington.edu/pub/rjl/papers/rjl-li:stokes.ps.Z`].

[LPL85] Randall J. LeVeque, Charles S. Peskin, and Peter D. Lax. Solution of a two-dimensional cochlea model using transform techniques. *SIAM Journal on Applied Mathematics*, 45:450–464, 1985.

[LPL88] Randall J. LeVeque, Charles S. Peskin, and Peter D. Lax. Solution of a two-dimensional cochlea model with fluid viscosity. *SIAM Journal on Applied Mathematics*, 48(1):191–213, 1988.

[Mas54] S. G. Mason. Fiber motions and flocculation. *Tappi Journal*, 37(11):494–501, 1954.

[MP80] M. F. McCracken and C. S. Peskin. A vortex method for blood flow through heart valves. *Journal of Computational Physics*, 35:183–205, 1980.

[MP90] David M. McQueen and Charles S. Peskin. A heart valve prosthesis. European Patent Publication Number EP 0 211 576 B1, 1990.

[MP91] David M. McQueen and Charles S. Peskin. Curved butterfly bileaflet prosthetic cardiac valve. U.S. Patent Number 5026391, 1991.

[MP93] Anita A. Mayo and Charles S. Peskin. An implicit numerical method for fluid dynamics problems with immersed elastic boundaries. In A. Y. Cheer and C. P. van Dam, editors, *Fluid Dynamics in Biology: Proceedings of the AMS–IMS–SIAM Joint Summer Research Conference on Biofluiddynamics (Contemporary Mathematics, vol. 141)*, pages 261–277. American Mathematical Society, 1993.

[OL89] Samuel Ohring and Hans J. Lugt. Two counter-rotating vortices approaching a free surface in a viscous fluid. Technical Report DTRC–89/013, David Taylor Research Center, Bethesda, MD, June 1989.

[OS88] Stanley Osher and James Sethian. Fronts propagating with curvature–dependent speed: Algorithms based on Hamilton–Jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.

[Pes72] Charles S. Peskin. Flow patters around heart valves: A numerical method. *Journal of Computational Physics*, 10:252–271, 1972.

[Pes77] Charles S. Peskin. Numerical analysis of blood flow in the heart. *Journal of Computational Physics*, 25:220–252, 1977.

[PM89] Charles S. Peskin and David M. McQueen. A three-dimensional computational model for blood flow in the heart. I. Immersed elastic fibers in a viscous incompressible fluid. *Journal of Computational Physics*, 81:372–405, 1989.

[PM92] Charles S. Peskin and David M. McQueen. Cardiac fluid dynamics. *Critical Reviews in Biomedical Engineering*, 20(5,6):451–459, 1992.

[PM93a] Charles S. Peskin and David M. McQueen. Computational biofluid dynamics. In A. Y. Cheer and C. P. van Dam, editors, *Fluid Dynamics in Biology: Proceedings of the AMS–IMS–SIAM Joint Summer Research Conference on Biofluiddynamics (Contemporary Mathematics, vol. 141)*, pages 161–185. American Mathematical Society, 1993.

[PM93b] Charles S. Peskin and David M. McQueen. Heart throb. Technical report, Pittsburgh Supercomputing Center, 1993. In *Projects in Scientific Computing*, a PSC research report. [`http://pscinfo.psc.edu/MetaCenter/MetaScience/Articles/Peskin/Peskin.h%tml`].

[PM95] Charles S. Peskin and David M. McQueen. A general method for the computer simulation of biological systems interacting with fluids. In C. P. Ellington and T. J. Pedley, editors, *Biological Fluid Dynamics*, volume 49 of *Symposia of the Society for Experimental Biology*, pages 265–276, Leeds, England, 1995.

[PP93] Charles S. Peskin and Beth Feller Printz. Improved volume conservation in the computation of flows with immersed elastic boundaries. *Journal of Computational Physics*, 105:33–46, 1993.

[PT83] Roger Peyret and Thomas D. Taylor. *Computational Methods for Fluid Flow.* Springer-Verlag, New York, 1983.

[PTVF92] William H. Press, Saul A Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge University Press, 2nd edition, 1992.

[RA93] Sridhar Ranganathan and Suresh G. Advani. A simultaneous solution for flow and fiber orientation in axisymmetric diverging radial flow. *Journal of Non-Newtonian Fluid Mechanics*, 47:107–136, 1993.

[RDK90] Joseph Rosenberg, Morton Denn, and Roland Keunings. Simulations of non-recirculating flows of dilute fiber suspensions. *Journal of Non-Newtonian Fluid Mechanics*, 37:317–345, 1990.

[RK97] Russell F. Ross and Daniel J. Klingenberg. Dynamic simulation of flexible fibers composed of linked rigid bodies. *Journal of Chemical Physics*, 106(7):2949–2960, February 1997.

[Rom96] Alexandre M. Roma. *A Multilevel Self Adaptive Version of the Immersed Boundary Method.* PhD thesis, New York University, 1996.

[Ros94] Madeleine E. Rosar. *A three dimensional model for fluid flow through a collapsible tube.* PhD thesis, New York University, 1994.

[Sam63] Lars-Gunnar Samuelsson. Measurement of the stiffness of fibres. *Svensk Papperstidning*, 66(15):541–546, 1963.

[SB91] Deborah Sulsky and J. U. Brackbill. A numerical method for suspension flow. *Journal of Computational Physics*, 96:339–368, 1991.

[ST82] Richard Skalak and Hüsnü Tözeren. Finite elements in bio-fluid mechanics. In R. H. Gallagher et al., editors, *Finite Elements in Biomechanics*, pages 23–38. John Wiley & Sons, 1982.

[SW95] John M. Stockie and Brian T. R. Wetton. Stability analysis for the immersed fiber problem. *SIAM Journal on Applied Mathematics*, 55(6):1577–1591, 1995.

[TMSB97] Anna-Karin Tornberg, Ralph W. Metcalfe, L. Ridgway Scott, and Babak Bagheri. A front-tracking method for simulating fluid particle motion using high-order finite element methods. In *Proceedings of the 1997 ASME Fluids Engineering Division Summer Meeting*, Vancouver, Canada, June 22-26, 1997.

[Tor96] Anna-Karin Tornberg. A front-tracking method for simulating bubble motion using high-order finite element methods. Technical Report 216, Department of Mathematics, University of Houston, Texas, 1996.

[TP92] Cheng Tu and Charles S. Peskin. Stability and instability in the computation of flows with moving immersed boundaries: A comparison of three methods. *SIAM Journal on Scientific and Statistical Computing*, 13(6):1361–1376, 1992.

[VY92] C. Cockerham Vesier and A. P. Yoganathan. A computer method for simulation of cardiovascular flow fields: Validation of approach. *Journal of Computational Physics*, 99:271–287, 1992.

[Wet97] Brian R. Wetton. Analysis of the spatial error for a class of finite difference methods for viscous incompressible flow. *SIAM Journal on Numerical Analysis*, 34(2):699–722, 1997.

[WGSO97] Geoffrey Wherrett, Ian Gartshore, Martha Salcudean, and James Olson. A numerical model of fibre motion in shear. In *Proceedings of the 1997 ASME Fluids Engineering Division Summer Meeting*, Vancouver, Canada, June 22–26, 1997.

[Whi74] Frank M. White. *Viscous Fluid Flow*. McGraw–Hill, New York, 1974.

[YM93] Satoru Yamamoto and Takaaki Matsuoka. A method for dynamic simulation of rigid and flexible fibers in a flow field. *Journal of Chemical Physics*, 98(1):644–650, January 1993.

[YM94] Satoru Yamamoto and Takaaki Matsuoka. Viscosity of dilute suspensions of rodlike particles: A numerical simulation method. *Journal of Chemical Physics*, 100(4):3317–3324, February 1994.

[Yus96] Jamaludin Mohd Yusof. *Interaction of massive particles with turbulence*. PhD thesis, Cornell University, 1996.

Appendix A
FFT Solver for Pressure on a Periodic Channel

In this Appendix, we outline the Fast Fourier Transform (FFT) technique used to solve the discrete pressure Poisson equation

$$D_h \cdot G_h \, P_{i,j} = R_{i,j},$$

on the domain $[0, L_x] \times [0, L_y]$, where L_x and L_y are chosen so that we can take an $N \times M$ grid with spacing $h = \frac{L_x}{N} = \frac{L_y}{M}$ that is identical in both directions. We must also ensure that M is an integer power of 2 so that an FFT can be used to solved the problem.

The pressure is periodic in the x–direction. Updating the velocity requires values of the pressure gradient at all interior points ($i = 0, 1, \ldots, N-1$ and $j = 1, 2, \ldots, M-1$) and so the pressure must be computed at all points including $j = 0$ and $j = M$.

Since the pressure is periodic in x, the basic solution procedure involves performing a discrete Fourier transform in the x–direction, and then solving the remaining coupled equations in the y–direction. Before describing the actual procedure, we will formulate the discrete equations.

A.1 The discrete equations

We have written the discrete Laplacian as the product $D_h \cdot G_h$, where

- G_h is the standard centered difference approximation to the gradient operator:

$$G_h P_{i,j} = \left(\frac{P_{i+1,j} - P_{i-1,j}}{2h}, \frac{P_{i,j+1} - P_{i,j+1}}{2h} \right);$$

- D_h^* is an approximation to the divergence operator, based on the technique proposed by Chorin [Cho68] and described more fully by Wetton [Wet97]. The "reduced divergence operator" D_h^* is based on simple centered differences at least two grid points away from the boundary (that is, $j = 2, 3, \ldots, M-2$):

$$D_h^* \cdot \vec{U}_{i,j} = \frac{U_{i+1,1} - U_{i-1,1}}{2h} + \frac{V_{i,j+1} - V_{i,j-1}}{2h}.$$

The stencils at the remaining points are derived using second order one–sided differences at points next to the boundary along with homogeneous boundary conditions:

$$\begin{aligned} D_h^* \cdot \vec{U}_{i,0} &= \frac{4V_{i,1} - V_{i,2}}{2h} \\ D_h^* \cdot \vec{U}_{i,1} &= \frac{U_{i+1,1} - U_{i-1,1}}{2h} + \frac{V_{i,2}}{2h} \\ D_h^* \cdot \vec{U}_{i,M-1} &= \frac{U_{i+1,M-1} - U_{i-1,M-1}}{2h} - \frac{V_{i,M-2}}{2h} \\ D_h^* \cdot \vec{U}_{i,M} &= \frac{-4V_{i,M-1} + V_{i,M-2}}{2h} \end{aligned}$$

This ensures that the computed pressure will result in a velocity that satisfies the discrete divergence–free condition with second order accuracy.

If we substitute the expression for $G_h P_{i,j}$ into the divergence formulae above, we obtain

$$D_h^* \cdot G_h P_{i,j} = \begin{cases} \frac{1}{4h^2} \left(-4P_{i,0} + P_{i,1} + 4P_{i,2} - P_{i,3} \right), & \text{if } j = 0, \\ \frac{1}{4h^2} \left(-4P_{i+2,1} + P_{i-2,1} + P_{i,3} - 3P_{i,1} \right), & \text{if } j = 1, \\ \frac{1}{4h^2} \left(P_{i+2,j} + P_{i-2,j} + P_{i,j+2} + P_{i,j-2} - 4P_{i,j} \right), & \text{if } 2 \leq j \leq M-2, \\ \frac{1}{4h^2} \left(P_{i+1,M-1} + P_{i-2,M-1} + P_{i,M-3} - 3P_{i,M-1} \right), & \text{if } j = M-1, \\ \frac{1}{4h^2} \left(-P_{i,M-3} + 4P_{i,M-2} + P_{i,M-1} - 4P_{i,M} \right), & \text{if } j = M. \end{cases} \quad \text{(A.1)}$$

A.2 The discrete Fourier transform

We now proceed to apply the discrete Fourier transform to equations (A.1), which amounts to substituting $P_{i,j} = e^{\hat{\imath}2\pi\alpha i/M}\,\hat{P}_{\alpha,j}$ into the discrete equations $D_h^* \cdot G_h P_{i,j} = R_{i,j}$ If we do so, we obtain

$$\begin{array}{ll}
\text{for } j=0: & -4\hat{P}_{\alpha,0} + \hat{P}_{\alpha,1} + 4\hat{P}_{\alpha,2} - \hat{P}_{\alpha,3} = 4h^2\,\hat{R}_{\alpha,0}, \\
\text{for } j=1: & \left(2\cos\left(\frac{4\pi\alpha}{M}\right) - 3\right)\hat{P}_{\alpha,1} + \hat{P}_{\alpha,3} = 4h^2\,\hat{R}_{\alpha,1}, \\
\text{for } 2 \le j \le M-2: & \left(2\cos\left(\frac{4\pi\alpha}{M}\right) - 4\right)\hat{P}_{\alpha,j} + \hat{P}_{\alpha,j+2} + \hat{P}_{\alpha,j-2} = 4h^2\,\hat{R}_{\alpha,j}, \\
\text{for } j=M-1: & \left(2\cos\left(\frac{4\pi\alpha}{M}\right) - 3\right)\hat{P}_{\alpha,M-1} + \hat{P}_{\alpha,M-3} = 4h^2\,\hat{R}_{\alpha,M-1}, \\
\text{for } j=M: & -\hat{P}_{\alpha,M-3} + 4\hat{P}_{\alpha,M-2} + \hat{P}_{\alpha,M-1} - 4\hat{P}_{\alpha,M} = 4h^2\,\hat{R}_{\alpha,M}.
\end{array}$$

For each α, these equations form a banded system of $M+1$ equations in $M+1$ unknowns which can be written in matrix form as

$$\begin{bmatrix}
-4 & 1 & 4 & -1 & 0 & 0 & 0 & \cdots & 0 \\
0 & 1+\hat{m}_\alpha & 0 & 1 & 0 & 0 & 0 & \cdots & 0 \\
1 & 0 & \hat{m}_\alpha & 0 & 1 & 0 & 0 & \cdots & 0 \\
0 & 1 & 0 & \hat{m}_\alpha & 0 & 1 & 0 & \cdots & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\
0 & \cdots & 0 & 1 & 0 & \hat{m}_\alpha & 0 & 1 & 0 \\
0 & \cdots & & 0 & 1 & 0 & \hat{m}_\alpha & 0 & 1 \\
0 & \cdots & & & 0 & 1 & 0 & 1+\hat{m}_\alpha & 0 \\
0 & \cdots & & & 0 & -1 & 4 & 1 & -4
\end{bmatrix}
\cdot
\begin{bmatrix}
\hat{P}_{\alpha,0} \\ \hat{P}_{\alpha,1} \\ \hat{P}_{\alpha,2} \\ \hat{P}_{\alpha,3} \\ \vdots \\ \hat{P}_{\alpha,M-2} \\ \hat{P}_{\alpha,M-1} \\ \hat{P}_{\alpha,M}
\end{bmatrix}
= 4h^2
\begin{bmatrix}
\hat{R}_{\alpha,0} \\ \hat{R}_{\alpha,1} \\ \hat{R}_{\alpha,2} \\ \hat{R}_{\alpha,3} \\ \vdots \\ \hat{R}_{\alpha,M-2} \\ \hat{R}_{\alpha,M-1} \\ \hat{R}_{\alpha,M}
\end{bmatrix}$$

where $\hat{m}_\alpha := 2\cos\left(\frac{4\pi\alpha}{M}\right) - 4$.

A.3 The solution procedure

The basic outline of the solution procedure is given below. Subroutine names are identified in `typewriter font`, and are adapted from the *Numerical Recipes* code in [PTVF92].

1. Set up the matrix coefficients and perform the *LU* decomposition using `bandec` (need only be done once).

2. Transform the data: $R_{i,j} \longrightarrow \hat{R}_{\alpha,j}$, using `realft( ... ,+1)`.

3. Solve the banded system for each α, using `bandsol`.

4. Apply the inverse transform: $\hat{P}_{\alpha,j} \longrightarrow P_{i,j}$, using `realft( ... ,-1)` for each α.

The quantities $\hat{P}_{i,j}$ and $\hat{R}_{i,j}$ are complex, while the entries of the matrix are real, and so the system above is a set of two coupled banded systems involving the *same real–valued matrix* for the real and imaginary parts. The matrix is non–singular for $\alpha = 1, 2, \ldots, \frac{N}{2} - 1$, but has two null modes for each of the sets of equations corresponding to $\alpha = 0$ and $\alpha = \frac{N}{2}$. The null modes can be eliminated by setting the four transform coefficients $\hat{P}_{0,M-1}$, $\hat{P}_{0,M}$, $\hat{P}_{N/2,M-1}$ and $\hat{P}_{N/2,M}$ to zero before performing the inverse transform.