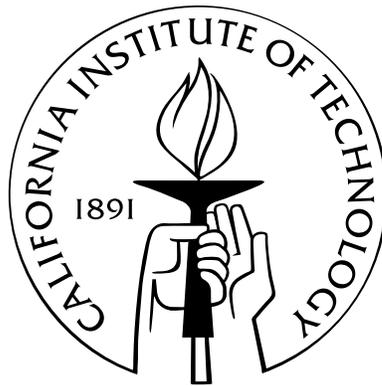


# Simulation Capabilities for Challenging Medical Imaging and Treatment Planning Problems

Thesis by  
Catherine E. Beni

In Partial Fulfillment of the Requirements  
for the Degree of  
Doctor of Philosophy



California Institute of Technology  
Pasadena, California

2011  
(Defended May 24, 2011)



for B, the greatest teacher of them all

# Acknowledgements

I owe my deepest gratitude to my advisor, Prof. Oscar P. Bruno. Working with him has been an honor and I have greatly benefited from his mentorship. This dissertation would not have been possible without his insight, guidance, contagious enthusiasm, and tremendous patience.

I would like to thank Sandia National Laboratories and the National Physical Sciences Consortium for encouraging me to begin my graduate studies and further enabling them by providing me with financial support.

It is a great pleasure for me to thank my friends and student colleagues at Caltech for all the care, laughter, and caffeine we shared over my years here. I am especially grateful to Greg Griffin, Andrew Kositsky, Nadine Dabby, Constantine Evans, Bao Ha, Alex Gittens, and Edwin Jimenez. My time here would not have been the same without them.

I greatly appreciate the care of all the administrative assistance at Caltech for helping the departments run smoothly and assisting me in many different ways. Sheila Shull, Syndey Garstang, and Tess Legaspi deserve special mention.

I am indebted to Riverside Community College, for taking a chance with such a young student and for all the wonderful mentoring and nurturing the faculty provided. I would especially like to thank Sal Rotella (for first letting me enroll at RCC), John “Biff” Pietro (for letting an eight year-old take his calculus class), Mary Legner, Rogelio Ruiz, and Sally Armstrong.

I wish to thank my undergraduate mentors at UCLA, for instilling a love of mathematics in me and for guiding me during my time there. For all their kind mentorship, I would like to thank in particular: Richard Elman (for convincing me to focus on applied mathematics), Chad Topaz (for introducing me to numerical analysis), Andrea Bertozzi, and John Garnett.

I would like to thank my sister, Juliet Beni, for standing by me, caring for me, and for patiently listening to more than her fair share of my long speeches about my thesis work.

Lastly, and most importantly, I wish to thank my parents, Susan Hackwood and Gerardo Beni. No words can truly express my gratitude toward them: for all their love and support over my life, and especially their care over my time here. They have nurtured me, mentored me, supported me, and loved me. They have been patient, kind, and understanding, even through difficult times. They raised me with a love of science and a love of learning by handcrafting a stimulating and fun learning

environment out of their own house for my sister and me to grow and learn in. I could not be more grateful.

# Abstract

Advanced numerical solvers and associated simulation tools, such as, for example, numerical algorithms based on novel spectral methods, efficient time-stepping and domain meshing techniques for solution of Partial Differential Equations (PDEs) (enabling, in particular, effective resolution of extremely steep boundary layers in short computing times), can have a significant impact in the design of medical procedures. In this thesis we present three recently introduced numerical algorithms for medical problems whose performance improves significantly over those of earlier counterparts, and which can thereby provide solutions to a range of challenging computational problems for planning and design of medical treatments.

# Contents

<b>Acknowledgements</b>	<b>iv</b>
<b>Abstract</b>	<b>vi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Magnetic Drug Delivery . . . . .	1
1.1.1 A Vessel-Membrane-Tissue Model . . . . .	2
1.1.2 Magnetically Enhanced Diffusion . . . . .	3
1.2 FC-AD Methodology . . . . .	5
1.3 Positron Emission Tomography . . . . .	7
1.4 Overview of Chapters . . . . .	8
<b>2 Overview of FC Methodology</b>	<b>9</b>
2.1 Fourier Continuation Basics . . . . .	9
2.2 Explicit Solver . . . . .	12
2.3 Alternating Directions Implicit . . . . .	13
2.4 FC-ODE Solver . . . . .	17
2.4.1 Constant Coefficient Solver . . . . .	17
2.4.2 Variable Coefficient Solver . . . . .	18
<b>3 Vessel-Membrane-Tissue Model</b>	<b>21</b>
3.1 Introduction to VMT Model . . . . .	21
3.2 VMT Solvers . . . . .	24
3.2.1 Domain Meshing . . . . .	24
3.2.1.1 Graded Mesh . . . . .	24
3.2.1.2 Multiple Equispaced Meshes . . . . .	25
3.2.2 ADI . . . . .	26
3.2.3 Jump conditions . . . . .	27
3.2.3.1 FD-Based Methods . . . . .	28

3.2.3.2	FC-Based Methods . . . . .	28
3.2.4	Fluid-Freezing Methodology . . . . .	29
3.2.5	Fast Time-Stepping . . . . .	30
3.3	Numerical Results . . . . .	31
<b>4</b>	<b>Magnetically Enhanced Diffusion</b>	<b>36</b>
4.1	Introduction . . . . .	36
4.2	Grids . . . . .	40
4.3	Dynamic Control Solver . . . . .	42
4.3.1	Periodic Boundary Conditions . . . . .	42
4.3.2	Radial Boundary Conditions . . . . .	43
4.3.3	Explicit-Implicit Solver . . . . .	44
4.3.4	Implicit-Implicit Solver . . . . .	45
4.3.5	Interpolation . . . . .	46
4.4	Numerical Results . . . . .	46
4.4.1	Ad Hoc Control Examples . . . . .	47
4.4.2	Future Work . . . . .	48
<b>5</b>	<b>Positron Emission Tomography</b>	<b>52</b>
5.1	Introduction . . . . .	52
5.2	Modified Filtered Back-Projection Algorithms . . . . .	54
5.3	Fejér-mFBP Algorithm . . . . .	57
5.4	Numerical Tests and Results . . . . .	60
5.4.1	Effects of Noise . . . . .	64
5.4.2	Fejér-mFBP and Classical Filtering . . . . .	65
<b>6</b>	<b>Conclusions</b>	<b>68</b>
<b>A</b>	<b>Numerical Computation of Magnetic Fields</b>	<b>69</b>
<b>B</b>	<b>Hilbert Transform Approximations</b>	<b>71</b>
	<b>Bibliography</b>	<b>73</b>

# List of Figures

2.1	Illustration of a periodic extension of $f(x) = \exp(\sin(5.4\pi x - 2.7\pi) - \cos(2\pi x))$ computed using the accelerated FC(Gram) method. . . . .	11
3.1	VMT geometry, as depicted in [42]. . . . .	23
3.2	Meshing of the domain of solution $\Omega$ of the PDE (3.1). . . . .	25
3.3	Grids used by the Graded and Standard VMT solvers. . . . .	26
3.4	Solution of the steady state concentration $C(x, y)$ obtained through use of the FC-VMT solver for a VMT model with for three parameter sets . . . . .	34
4.1	Geometry corresponding to the control setup. . . . .	39
4.2	Grids used by the Explicit-Implicit FC solver. . . . .	41
4.3	Numerical solution of the ad hoc control scheme (4.3) using the parameter choices $D = 0.001, k = 10$ . . . . .	49
4.4	Numerical solution of the ad hoc control scheme (4.3) using the parameter choices $D = 0.001, k = 20$ . . . . .	50
5.1	Variables and geometry associated with the Radon Transform. . . . .	53
5.2	The original $500 \times 500$ pixel Shepp-Logan phantom, and noiseless-data reconstruction resulting from the averaged mFBP and averaged Fejér-mFBP algorithms. . . . .	61
5.3	MATLAB's 'iradon' reconstructions of the $500 \times 500$ pixel Shepp-Logan phantom for noiseless data. . . . .	62
5.4	The original $500 \times 500$ pixel Shepp-Logan phantom and its Fejér-mFBP reconstructions in the presence of 10.52% noise. . . . .	62
5.5	mFBP reconstructions of the $500 \times 500$ pixel Shepp-Logan phantom for noiseless data. . . . .	63
5.6	Fejér-mFBP reconstructions of the $500 \times 500$ pixel Shepp-Logan phantom for noiseless data. . . . .	63
5.7	Comparison of Fejér filtering and previous filtering methodologies (using, in all cases, the new three-fold averaging approach) for the $500 \times 500$ pixel Shepp-Logan phantom with 18.38% noise. . . . .	65

5.8	Shepp-Logan-mFBP and Fejér-mFBP vs. Shepp-Logan-filtered FBP for the $500 \times 500$ pixel Shepp-Logan phantom with 18.38% noise. . . . .	66
B.1	The true values (solid lines) of $C_T(\rho, k)$ for $k = 1, 2, 3$ and 4 (Figures (i)–(iv)) and $S_T(\rho, k)$ for $k = 1, 2, 3$ and 4 (Figures (v)–(viii)) plotted with their corresponding approximations (circles) computed via (5.12) using $T = 4$ . . . . .	72

# List of Tables

- 3.1 Table of coefficients and coordinates used in each layer, including both the standard equispaced and exponentially graded meshes in the vessel. The far left column provides the corresponding notation used in Section 2.3 for the description of the ADI algorithm. 27
- 3.2 Computational times required to obtain various accuracies for the parameter set  $D^v = 10^{-5}$ ,  $M^v = 10^{-4}$  through use of each of the three VMT solvers. The solutions were computed over grids of size  $N_x \times N_y$  with the parameter choices  $\Lambda = 25$ ,  $\sigma_f = 10^{-5}$ , and  $\sigma_s = 10^{-5}$ . The entry marked by \* was obtained using the selection  $\sigma_f = 10^{-6}$ . . . 33
- 3.3 Computational times and errors at the critical midpoint  $\boldsymbol{\rho} = ((x_r - x_l)/2, I^{vm})$  using  $D^v = 10^{-5}$ ,  $M^v = 10^{-4}$  for the three VMT solvers. The solutions were computed over a  $400 \times 400$  grid with the parameter choices  $\Lambda = 25$ ,  $\sigma_f = 10^{-5}$ , and  $\sigma_s = 10^{-5}$ . . . . 35
- 4.1 Table of coefficients and coordinates used over each of the two grid types for both the Explicit-Implicit and Implicit-Implicit FC-Solvers. The far-left column provides the corresponding notation used in Section 2.3 for the description of the ADI algorithm. . . 42

# Chapter 1

## Introduction

Recent advances in medical science have given rise to many challenging computational problems. In some cases the resulting computational demands are so taxing that they simply overwhelm the capabilities of standard numerical methods. In this thesis we introduce three innovative numerical algorithms which, relying on novel spectral approaches, efficient time-stepping and domain meshing techniques for solution of Partial Differential Equations (PDEs) (enabling, in particular, effective resolution of extremely steep boundary layers in short computing times), improve significantly on existing numerical tools and provide, in fact, solutions to previously intractable medically-relevant problems. These methods were developed for use in two specific medical applications: 1) the field of therapeutic magnetic drug delivery, which involves the transport of magnetized particles through a convecting blood vessel and subsequent diffusion into surrounding tissues, and 2) the reconstruction of images obtained from Positron Emission Tomography (PET) scans.

### 1.1 Magnetic Drug Delivery

The goal of magnetic drug delivery is to use magnetic fields to direct and confine magnetically responsive particles (which, containing therapeutic agents, are injected into the bloodstream) to specific regions in a patient's body—thus allowing for focused treatment in an area of interest. Numerous benefits arise from the use of such a site-directed drug delivery system, including the enabled focusing of chemotherapy to tumors, anticoagulants to blood clots, and antibacterial drugs to infection sites, in addition to the limiting of the potentially debilitating effects of overtreatment. In cancer treatment, for example, severe complications may arise from overadministration of chemotherapy to healthy tissues [36]. By instead selectively delivering the necessary treatment, such disease sites may be treated effectively while minimizing damage to surrounding healthy tissues.

### 1.1.1 A Vessel-Membrane-Tissue Model

Several factors are most critical for understanding the efficacy of magnetically controlled drug targeting: 1) the geometry of the vasculature and velocity of the blood flow near the disease site, 2) the susceptibility of the magnetic particles (or ferrofluid) to the externally-applied magnetic forces, and 3) the *in vivo* diffusion of the particles into surrounding tissues. Past animal experiments [37, 38, 3, 53, 41] and phase I human clinical trials [38, 39, 35] have observed the accumulation of magnetic nanoparticles (or ferrofluid) by visual inspection, magnetic resonance imaging (MRI), and histology studies. These studies have shown that magnetic forces can concentrate micro and nanoparticles *in vivo* near the location of the external magnets. However, the details of the accumulation are not visible experimentally: the resolution in both MRI and visual inspection is not high enough to determine where magnetic forces have overcome blood velocity in the blood vessels and, because they must be carried out after the animal has been sacrificed and blood flow stopped, only partial knowledge of the particle behavior is provided through histology studies. Exact determination of the location of the ferrofluid accumulation via these methods is nearly impossible. In the recent publication [42], we proposed the use of numerical simulation of a simplified vessel geometry to better understand the accumulation behavior and evaluate the effects of external magnetic forces on the convection and diffusion of magnetic particles through the bloodstream and in membranes and tissues. While taking into account that the simulations are performed in an idealized geometry, the use of numerical simulation enables preliminary analysis and characterization of the particle behavior under different levels of diffusive and advective forces produced by the blood velocity and magnetic fields to be carried out both systematically and inexpensively. Such an analysis will thus, in turn, better enable the design of a method leading to confinement of the magnetically responsive particles to a particular region of the body.

An effective mathematical model of a blood vessel has been proposed by Grief and Richardson [26] and was extensively analyzed in [42] utilizing solver codes based on the numerical methods described in Chapter 3. The idealized geometry of the Grief and Richardson model is a version of the Krogh tissue cylinder [23] consisting of a lateral cross section of a blood vessel, including the endothelial layer (or membrane) and some surrounding tissue, with an external magnet situated below (see Figure 3.1). The characterizing equation of the resulting Vessel-Membrane-Tissue (VMT) geometry is given by the hyperbolic convection-diffusion PDE,

$$\frac{\partial}{\partial t} C(\vec{r}, t) = -\nabla \cdot \left[ C(\vec{r}, t) \vec{V}_{\text{blood}}(\vec{r}, t) - D(\vec{r}) \nabla C(\vec{r}, t) + k(\vec{r}) C(\vec{r}, t) \nabla \left( |\vec{H}(\vec{r}, t)|^2 \right) \right], \quad (1.1)$$

with diffusion coefficient and advective terms varying over each layer.

Numerical solution of the Grief and Richardson VMT model for realistic values of diffusion coefficients has proven highly challenging to numerical solvers for several reasons, amongst which

figures prominently the strong concentration build-up, or *boundary layer*, that appears at the vessel-membrane interface: due to the discontinuity in diffusion coefficients over each layer, sharp boundary layers occur when the magnetic forces acting on the particles are strong enough to overcome the blood velocity, an effect that is particularly noticeable for the extremely small nondimensional diffusion coefficients present in realistic models of capillaries. Additionally, the advective forces provided by the blood velocity are significantly more powerful than the diffusive forces, thus giving rise to greatly disparate time scales; the development of an efficient time evolution methodology was necessary to produce steady state solutions in a reasonable amount of time.

In analyses prior to the work [42], numerical solution of the VMT problem was performed through use of the finite-element-based commercial software COMSOL Multiphysics ([www.comsol.com](http://www.comsol.com)). While capable of solving the VMT problem for large diffusion coefficients, the COMSOL software encountered many difficulties for the small diffusion coefficients inherent in realistic instances of the VMT model, especially for larger values of the magnetization. For example, some relevant drug absorption studies required solutions of convection-diffusion problems with (dimensionless) diffusion constants of the order of  $10^{-7}$ . In the preliminary studies performed on the COMSOL software, taking the diffusion and magnetization coefficients to equal  $10^{-4}$  and  $10^{-3}$  respectively, a steady state solution was reached in 36 hours of run time on a 3.16 GHz single processor of a quad-core Intel Xeon CPR X5460 computer with 32 GB of memory; the corresponding memory requirements to obtain the steady state solution for values of the diffusion and magnetization equal to  $10^{-5}$  exceeded the amount available on the same computer. Therefore, the medically relevant VMT model with diffusion constant equal to  $10^{-7}$  and magnetization on the order of  $10^{-6}$  lies far outside the domain of applicability of the COMSOL software in the said computer. In contrast, the algorithms described in Chapter 3 produced accurate solutions for cases relevant to the contribution [42] without difficulty. For example, on a computer with a 2.66 GHz Intel Core 2 Duo processor and 4GB of memory, our solvers provide the required numerical solutions to the  $10^{-4}$  and  $10^{-5}$  problems in under five minutes using 3 MB (not GB!) of memory and under fourteen minutes using 15 MB of memory, respectively. One of the most challenging cases considered in [42], in turn, for which the diffusion constant equals  $10^{-7}$ —a case that is very far from feasible for other methods—completed, using 25 MB of memory, in a six hour run.

### 1.1.2 Magnetically Enhanced Diffusion

Magnetic drug delivery inherently suffers from two severe limitations: 1) the inability to focus treatment on targets located deep inside the body with a stationary magnetic field (when stationary external magnets are used, particles can only be moved at a maximum depth of 5 cm) and 2) the fundamental consequence of the classic Samuel Earnshaw theorem [18] that no inverse-square law force (including magnetic force on a single particle) can create a stable equilibrium in the

interior of a domain—only unstable equilibria for ferrofluid particles may be attained with a static magnetic field. Several methods to bypass Earnshaw’s theorem have been proposed, including the implantation of magnetic materials, such as stents or wires, inside the body in order to create a local magnetic field [28, 49, 50], and the use of carefully placed external magnets so as to trap the ferrofluid against certain blood vessel walls [38, 39]. However, these techniques are not always viable solutions: surgical implantation of such objects in a patient may be undesirable or not feasible in a clinical setting and, due to the complex nature of the human blood vasculature network, entrapment of the particles against particular blood vessel walls has proven to be nearly impossible.

To overcome these limitations, Shapiro [51] proposed the development of a *dynamic* feedback-control scheme, where manipulation of the magnetically responsive particles is sought through dynamic adjustment of the magnetic fields. (An example of the use of dynamic control to bypass Earnshaw’s theorem has been shown by the work of Potts et al. [47], wherein dynamic manipulation of a single electromagnet was used to suspend a drop of ferrofluid a distance away from the magnet.)

Development of such a feedback-control scheme depends on evaluation of the effects of external magnetic forces on the convection and diffusion of magnetically responsive particles through the relevant tissues. In order to simplify the complex nature of these effects in standard vasculature geometries, and thus better understand the feasibility of such a control scheme, Shapiro [51] considered the Grief and Richardson convection diffusion PDE (1.1) over an idealized domain: a circular region surrounded by eight electromagnets placed equally in a ring (see Figure 4.1). A zero-flux condition is imposed at the boundary of the domain in order to ensure no ferrofluid leaves the domain. Clearly, the design of adequate control schemes requires efficient numerical solution of such PDEs. This modification of the Grief and Richardson model may thus be utilized as a test bed for the development of numerical algorithms capable of evaluating such solutions accurately and efficiently.

For the parameter values inherent in the medical configurations under consideration, the numerical PDE problems have proven quite challenging. For the small diffusion coefficients typically required to portray a relevant control setup, the imposition of the zero-flux boundary condition combined with the strong convective forces generated by the external magnetic field will cause extremely sharp boundary layers to rapidly appear near the boundary of the domain.

Once again, numerical studies presented in reference [51] were conducted using the commercial software package COMSOL Multiphysics. Similarly to the problems encountered for the VMT configuration, the COMSOL software was incapable of accurately resolving the steep boundary layer occurring for smaller values of diffusion coefficient. For example, numerical solution for the choice of diffusion coefficient  $D = 0.001$  and magnetic drift  $k(\vec{r}) = 1$  using the previously mentioned Intel Xeon CPR X5460 computer with 32 GB of memory was not feasible with the COMSOL software: to do so would require memory allocation greater than the computer’s capacity.

In contrast, taking the diffusion and magnetic drift,  $k(\vec{r})$ , coefficients to be 0.001 and 20 re-

spectively, numerical solution of the ad hoc control scheme proposed in [51] was achieved with a maximum relative error of  $10^{-2}$  in under two hours with 50 MB of memory by using the solvers introduced in Chapter 4 of this thesis.

## 1.2 FC-AD Methodology

The central challenge posed by both of these magnetic drug delivery simulations is the need for accurate and efficient resolution of boundary layers. Accurate resolution of steep boundary layers presents many difficulties for numerical solvers, at the heart of which is the requirement of a very fine spatial step size. This requirement imposes limitations on the type of numerical method efficiently useable: due to the requirement of such a fine mesh, explicit methods are rendered highly inefficient by the restrictive CFL condition  $\Delta t \sim \mathcal{O}(\Delta x^2)$  imposed on the time step.

Further, because numerical solution over a nonrectangular domain is required for the idealized control setup described above and certainly necessary for more complicated models of the vasculature system, a standard finite differences approach over a Cartesian grid is not an efficient method of solution: such a scheme would require either a “staircasing” of the boundary of the domain or challenging, essentially impractical domain mapping strategies. Use of the first, rather simple technique reduces the spatial accuracy of the resulting finite difference method to first order. In addition, due to the absence of solution values outside the computational domain, finite difference stencils are forced to be made increasingly one-sided as the domain boundaries are approached. In general, stability is not achieved by simply using high-order centered difference methods in the interior of the domain and equally high-order biased stencils near the boundary. While there are several techniques to resolve this problem (such as the use of compact schemes [4] or Summation By Parts operators [40, 48]), these approaches are computationally expensive and must sacrifice some accuracy near the boundary to gain stability. Importantly, further, the multidomain strategies associated with these algorithms require the discretizations of neighboring domains to match perfectly at common boundaries; see, e.g., [2] for details.

Numerical solution of PDEs over nonrectangular domains may also be produced by means of finite element and finite volume solvers. These algorithms do not provide an effective method of solution in the presence of boundary layers, however: because the spectral radii of differentiation operators based on nonuniformly spaced structured grids such as those used by finite element and finite volume methods will, in general, grow superlinearly, these techniques must also satisfy a stringent CFL condition for stability. High-order methods based on unstructured meshes give rise to similarly restrictive CFL constraints.

A new methodology for the numerical solution of PDEs in *general domains* has been recently introduced by Bruno and Lyon [16]. Based on use of the the well-known Alternating Direction

Implicit (ADI) methodology first introduced in 1955 by Peaceman and Rachford [45] in conjunction with the pseudospectral Fourier Continuation (FC) method, the resulting FC-AD algorithm can yield high-order accurate solutions with unconditional stability over general spatial domains at an essentially linear computational cost.

Many variants of the finite-difference-based ADI algorithm for the Heat and Laplace Equations have been put forward in the more than 50 years since its introduction, including methods for solution of various linear and nonlinear PDEs and solvers providing high-order spatial and temporal accuracy. Unconditionally stable alternating-direction methods preceding [16] could only achieve high-order accuracy for PDEs over domains representable as a union of a finite number of rectangular regions containing perfectly matched Cartesian discretizations. The few unconditionally stable high-order ADI algorithms that have been applied to nonrectangular geometries rely on use of domain mappings to transform the given problem into one posed over a rectangular geometry. Unfortunately, the inherently laborious construction of such domain mappings prohibits the use of these methods for most problems posed by engineering and scientific applications.

Alternating direction methodologies have also been previously used with spatial differentiation methods that do not depend on finite difference techniques. In particular, alternating direction approaches relying on a Fourier basis for differentiation have been proposed [5, 19, 43, 56]. However, despite efforts seeking to generalize these methods to more general domains, the application of previous Fourier-based techniques has also been restricted to rectangular geometries.

Because it is dependent on the Fourier approximation of nonperiodic functions, the use of Fourier bases in alternating directions methodologies requires resolution of a notorious problem in numerical analysis: the Gibbs phenomenon. A cornucopia of methods have been developed to resolve or reduce the detrimental effects of Gibbs phenomenon (see, e.g., [17, 25, 24, 12]). The Fourier Continuation method first proposed in [14, 13, 11] and accelerated in [16] eliminates the Gibbs phenomenon through the use of a “continuation” of the original function into a periodic extension; due to its periodic nature, the production of such a continuation enables the smooth approximation of the original function through use of Fourier series without difficulty. The use of this FC continuation strategy in conjunction with the alternating directions methodology, gives rise to the so-called FC-AD algorithm.

The high-order accuracy, unconditional stability, and ability to accurately handle boundary conditions over general domains of the FC-AD algorithm provides significant advantages over other methods. In particular, the unconditional stability allows for the use of significantly larger time steps than those required by conditionally stable methods. This characteristic is of the utmost importance in the context of the convection-diffusion equations under consideration in this thesis, especially in the presence of the overwhelmingly restrictive CFL conditions imposed on the time step by the fine spatial step size required for the resolution of steep boundary layers. Addition-

ally, the high-order accuracy and accurate handling of boundary conditions offered by the FC-AD algorithm generally allow for accurate results with much coarser discretizations than otherwise necessary. These qualities make the FC-AD method a prime choice for the numerical solution of the convection-diffusion PDEs used to model magnetic drug delivery.

### 1.3 Positron Emission Tomography

Positron emission tomography (PET) is a nuclear imaging technique relying on the unique positron-emitting decay characteristics of radioactive tracer isotopes, or radiopharmaceuticals [20, 46]. After being introduced into the body (typically through intravenous injection) the radiopharmaceuticals undergo positron emission decay, ultimately leading to the emission of a pair of high-energy photons traveling in approximately opposite directions. A ring of detectors placed around the body, the core component of the PET scanner, enables suitable detection of these high-energy photon pairs. Raw data collected from a PET scanner is simply a list of “coincidence events” representing near-simultaneous detection of the pair of high-energy photons by a pair of detectors. These coincident events, in turn, represent lines in space between the detector pairs along which the positron emission occurred. The collection of coincidence events obtained from a PET scanner is described mathematically by the Radon transform [44, 7, 8, 6]; reconstruction of the desired image follows from appropriate inversion of the Radon transform.

Over the last few decades, two major types of reconstruction methodologies have emerged in the field of positron emission tomography (PET): 1) Iterative methods (which include the ML-EM [52] and OSEM [27] algorithms), and 2) Methodologies based on the Filtered Back-Projection (FBP) approach [20]. While iterative approaches are significantly slower than FBP-based methods, in presence of the types of errors arising from PET scanners they produce images of better quality than their previously existing FBP-type counterparts. As a result, slow iterative approaches are almost exclusively used in commercial PET devices [46]. In Chapter 5 we introduce a new FBP-based method for reconstruction of images from PET scans, the Fejér-mFBP algorithm [9], that incorporates 1) A certain averaging technique that helps moderate the detrimental effects of high frequencies in the reconstruction process, and 2) A new Fejér-based filtering procedure. The resulting Fejér-mFBP algorithm runs at essentially the same cost as previous FBP-based methods and, for the types of noise typical in present day PET devices, it gives rise to images of significantly higher quality than previous approaches—compare, e.g., Figures 5.3(c) and 5.7(f).

## 1.4 Overview of Chapters

The remainder of this thesis is organized as follows. In Chapter 2, an overview of the Fourier continuation algorithm and its use in explicit and implicit solvers is given. This chapter provides a basis for the numerical methodologies used in the solution of the convection-diffusion problems considered in Chapters 3 and 4. Chapter 3 considers the Vessel-Membrane-Tissue (VMT) convection-diffusion problem and introduces three new VMT solvers based on the techniques discussed in Chapter 2. Next, Chapter 4 focuses on the magnetically enhanced diffusion model proposed in [51] and presents a series of numerical tools also based on the methodologies presented in Chapter 2. Various combinations of these tools enable efficient solution of the proposed model for differing values of diffusion and magnetic drift coefficients. In Chapter 5, finally, we consider the problem of reconstruction of images from Positron Emission Tomography (PET) scans and present a new FBP-based method for image reconstruction.

## Chapter 2

# Overview of FC Methodology

### 2.1 Fourier Continuation Basics

The Fourier Continuation (FC) methodology first introduced in [11, 13] and then accelerated in [16] is a central element in a class of unconditionally stable implicit PDE solvers, the FC-AD methods, for linear constant [16] and variable-coefficient [15] PDEs in general domains. At the heart of this methodology is an accelerated periodic-continuation algorithm enabling a smooth Fourier representation of nonperiodic functions without the Gibbs ringing effect inherent in standard Fourier series approximations of nonperiodic functions. Indeed, by constructing a periodic continuation of the function in a domain significantly larger than the original interval, the resulting FC method smoothly resolves the detrimental oscillatory effects of Gibbs phenomenon. In this chapter, we provide the basic outlines for implementation of two Fourier continuation methods: the unaccelerated FC(SVD) method presented in [13] and the accelerated FC(Gram) method first introduced in [16]. To facilitate explanation, we will consider both the FC algorithms over the unit interval  $[0, 1]$ ; application to a general interval easily follows from a simple scaling and translation of the discrete grid.

Consider a smooth function  $f(x)$  for which the point values  $f_j = f(x_j)$  are given over the discrete grid  $x_j$ ,  $j = 0, \dots, N - 1 \in [0, 1]$ ,

$$x_j = j h, \quad j = 0, \dots, N - 1, \quad h = 1/(N - 1).$$

For a given value  $b > 1$ , the Fourier continuation algorithm prescribes a method of construction of a  $b$ -periodic trigonometric polynomial,

$$f^c(x) = \sum_{k \in t(M)} a_k e^{\frac{2\pi i}{b} k x}, \quad (2.1)$$

with

$$f^c(x_j) = f(x_j) \text{ for } j = 0, \dots, N - 1.$$

that approximates  $f(x)$  in the interval  $[0, 1]$ . The index function  $t(M)$  is given by  $t(M) = \{j \in \mathbb{N} : -M/2 + 1 \leq j \leq M/2\}$  for  $M$  even and  $t(M) = \{j \in \mathbb{N} : -(M - 1)/2 \leq j \leq (M - 1)/2\}$  for  $M$  odd. Selection of the bandwidth  $M$  and the period  $b$  depends on the FC method used and will be determined in what follows. It is important to note that for  $b = 1$ , the continuation  $f^c(x)$  reduces to the discrete Fourier transform of  $f(x)$ , and thus typically suffer from Gibbs phenomenon near the endpoints  $x = 0, 1$ . Selecting  $b > 1$ , on the other hand, allows for  $f^c(x)$  to smoothly transition from the values of  $f(x)$  near  $x = 1$  to the values of  $f(x)$  near  $x = 0$  without oscillatory effects.

The basic, unaccelerated FC algorithm presented in [13] obtains the coefficients  $a_k$  through solution of the least-squares system

$$\min_{\{a_k: k \in t(M)\}} \sum_{j=0}^{N-1} \left| \sum_{k \in t(M)} a_k e^{\frac{2\pi i}{b} k x_j} - f(x_j) \right|^2. \quad (2.2)$$

Numerical results [13] have shown that it is advantageous to use a Singular Value Decomposition (SVD) to solve least-squares system (2.2); to better distinguish it from the accelerated FC algorithm presented below, we henceforth refer to this FC method as FC(SVD). Figure 2.1 presents a typical periodic continuation produced by application of the FC(SVD) method to function values of  $f(x) = \exp(\sin(5.4\pi x - 2.7\pi) - \cos(2\pi x))$  given in the interval  $[0, 1]$ .

While adequate for applications requiring a small number of highly-accurate continuations, such as the high-order surface representations in [13], the  $\mathcal{O}(N^3)$  computational cost of one application of the FC(SVD) method is significantly higher than desirable for use in an algorithm dependent on the rapid computation of many periodic continuations, such as the FC-AD PDE solver introduced in [16, 15] and generalized in Section 2.3 of this text. Indeed, because both of these FC-based PDE solvers depend on computation of  $\mathcal{O}(N^2)$  continuations per time-step, the high cost per continuation required by the FC(SVD) algorithm would render such solvers extremely inefficient. Recently, [16] overcame this issue by introducing a significantly accelerated FC method, the FC(Gram) algorithm. To highlight the differences between the FC(SVD) method and the accelerated FC algorithm presented in [16] and used in the numerical solvers introduced this text, a brief outline of the FC(Gram) method is provided in the remainder of this section.

As mentioned earlier in this section, a primary component of the general Fourier continuation method is the generation of a discrete periodic extension of the given function values into a longer interval  $[1, b]$ . Such an extension can be obtained by appending an additional  $n_d > 0$  function values to the original  $N$  function values such that the newly extended function provides a smooth transition from  $f_{N-1}$  back to  $f_0$  on the interval  $[1, b]$ , as shown in Figure 2.1. Once a discrete periodic extension

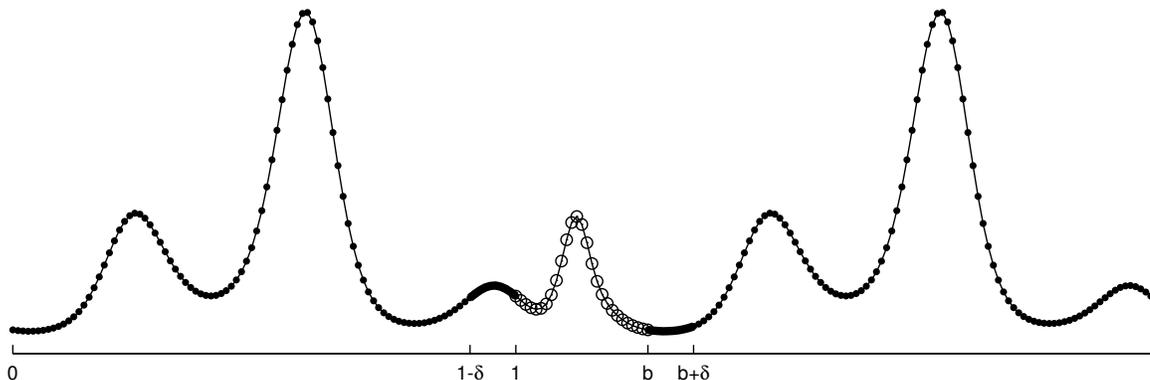


Figure 2.1: Illustration of a periodic extension of  $f(x) = \exp(\sin(5.4\pi x - 2.7\pi) - \cos(2\pi x))$  computed using the accelerated FC(Gram) method, including the original function values (solid circles) and the extended function values (open circles). The thick black lines depict the function values over matching intervals  $S_{\text{left}}$  and  $S_{\text{right}}$ .

is produced, an application of the FFT algorithm to the newly extended discrete function values in the interval  $[0, b]$  yields the coefficients  $a_k$  of the Fourier continuation  $f^c(x)$  (as seen in equation (2.1)); the resulting approximation has bandwidth  $M = N + n_d$  and period  $b = (N + n_d)/(N - 1)$ .

In the FC(Gram) algorithm, the main idea underlying the construction of the  $n_d$  extension values mentioned above involves consideration of both the given function  $f$  defined in the interval  $[0, 1]$  together with the translation  $f(x - b)$  defined in the interval  $[b, 1 + b]$  (depicted as thin black lines with solid circles in Figure 2.1). Selecting a positive integer  $n_\delta$ , we define the “matching intervals”  $S_{\text{right}} = \{x_j | j = N - n_\delta, N - n_\delta + 1, \dots, N - 1\}$  to be the set of  $n_\delta$  right-most discretization points in the interval  $[0, 1]$  and  $S_{\text{left}} = \{b + x_j | j = 0, 1, \dots, n_\delta - 1\}$  to be the set of  $n_\delta$  left-most discretization points in the interval  $[b, 1 + b]$ . Loosely speaking, the  $n_d$  function values sought are obtained as the point values of an auxiliary trigonometric polynomial produced by means of a least-squares fit to the function values of  $f(x)$  over  $S_{\text{right}}$  and the translation  $f(x - b)$  over  $S_{\text{left}}$ .

The construction of such a trigonometric polynomial proceeds by considering two orthonormal bases of Gram polynomials of degree  $< m$ . These bases, denoted by  $P_{\text{right}}$  and  $P_{\text{left}}$ , are generated through application of the Gram-Schmidt orthonormalization process with inner products

$$\begin{aligned} \langle g, h \rangle_{\text{right}} &= \sum_{\{x_j \in S_{\text{right}}\}} g(x_j)h(x_j), \quad \text{and} \\ \langle g, h \rangle_{\text{left}} &= \sum_{\{x_j \in S_{\text{left}}\}} g(x_j - b)h(x_j - b), \end{aligned}$$

respectively, to the standard polynomial basis  $\{1, x, x^2, \dots, x^{m-1}\}$ . As fully detailed in [16], the FC(Gram) algorithm precomputes a set of functions that smoothly transition from a given poly-

nomial in  $P_{\text{right}}$  to a given polynomial in  $P_{\text{left}}$  over the interval  $[1, b]$  by applying the FC(SVD) algorithm to discrete data sets of the form  $\{p(x_j)|x_j \in S_{\text{right}}\} \cup \{q(x_j)|x_j \in S_{\text{left}}\}$  for each polynomial  $p \in P_{\text{right}}$  and  $q \in P_{\text{left}}$ . This set of precomputed transition functions effectively forms a basis of smooth transition functions from function values in  $S_{\text{right}}$  to function values in  $S_{\text{left}}$ . Thus, by noticing that the given function values of  $f(x)$  in  $S_{\text{right}}$  and  $f(x - b)$  in  $S_{\text{left}}$  may be expressed as linear combinations of the Gram polynomials in  $P_{\text{right}}$  and  $P_{\text{left}}$  respectively, it follows that we can quickly compute a smooth transition from  $f(x)$  to  $f(x - b)$  in the interval  $[1, b]$  as an appropriate linear combination of the precomputed smooth transition functions.

An illustration of the full periodic-extension procedure is provided in Figure 2.1. As indicated in Figure 2.1, the  $n_d$  extension values we seek are given simply as the values of this new smooth transition function over the discrete grid  $1 + x_j$ ,  $j = 0, \dots, n_d - 1$ . An application of the discrete Fourier transform in the interval  $[0, b]$  to the function values  $f_j$  appended to the additional  $n_d$  extended function values yields the desired trigonometric continuation polynomial  $f^c(x)$ .

**Remark 2.1.1** Once a continuous continuation function  $f^c(x)$  has been obtained, numerical approximations of the derivatives of  $f(x)$  can easily be computed with spectral accuracy through direct differentiation of  $f^c(x)$ :

$$\frac{df}{dx} \approx \frac{df^c}{dx} = \sum_{k \in t(M)} \frac{2\pi i a_k}{b} e^{\frac{2\pi i}{b} kx}. \quad (2.3)$$

The overall procedure for approximating one-dimensional derivatives of a given vector of function values,  $f(x_j)$ ,  $j = 0, \dots, N - 1$  thus consists of application of the FC method followed by an application of the differential operator (2.3); clearly, derivatives of higher order can be produced similarly. Derivatives in multiple spatial dimensions on a structured mesh are implemented through successive line-by-line applications of the FC algorithm and differential operator in each coordinate direction. The behavior of the two-dimensional algorithm is straightforward, simply sweeping through horizontal and vertical lines of grid points and applying the one-dimensional differentiation algorithm in the corresponding coordinate; generalization to higher dimensions and higher-order derivatives is similarly straightforward.

## 2.2 Explicit Solver

Recalling the accurate FC-based method for the approximation of spatial derivatives presented in Remark 2.1.1 of Section 2.1, it is a straightforward matter to prescribe an explicit PDE solver: the algorithm proceeds, simply, by iterating a time-marching scheme. For the Explicit-Implicit and Implicit-Implicit FC solvers described in Chapter 4, we made use of the second-order Runge-Kutta time-marching scheme known as Heun's method. In detail, denoting  $C^n = n\Delta t$  and  $C^n = C(u, v, t^n)$ , approximate solution to the general-convection diffusion PDE (2.5) is advanced through use of the

second-order Runge-Kutta scheme,

$$\begin{aligned}\tilde{C}^{n+1} &= C^n + \Delta t L[C^n], \\ C^{n+1} &= C^n + \frac{\Delta t}{2} \left( L[C^n] + L[\tilde{C}^{n+1}] \right),\end{aligned}\tag{2.4}$$

with

$$L[C] = \vec{\kappa}(u, v) \cdot \nabla^2 C + \vec{\lambda}(u, v) \cdot \nabla C + \nu(u, v)C + Q(u, v, t).$$

The spatial derivatives present in application of the differential operator  $L$  are approximated by the previously described FC method.

## 2.3 Alternating Directions Implicit

As mentioned in Section 2.2, an explicit method needs to satisfy the CFL condition  $\Delta t \sim F(\Delta x)$  (specifically  $\Delta t \sim \mathcal{O}(\Delta x^2)$  for the problems under consideration in this thesis) to ensure numerical stability; the very fine discretization mesh required to resolve the boundary layers occurring in magnetic drug delivery problems therefore places a severe restriction on the maximum time step useable by an explicit method. In order to overcome this restriction, the solvers presented in this text make use of an efficient FC-AD implicit solver [16, 15]. Based on the Alternating Directions Implicit (ADI) method first developed by Peaceman and Rachford in 1955 [45], the FC-AD solver relies on line-by-line solution of ordinary differential equations (ODEs) that arise from factorization of the partial differential operator into differential operators of each variable; the resulting ODEs are then, in turn, solved for using an efficient FC method. In this section, a description of the FC-AD method for solution of a generalized version of the PDE (1.1) is provided. This generalization is chosen to allow for easy adaptation to the various convection-diffusion equations for the solution of each of the magnetic drug delivery problems presented in later chapters. An outline of the efficient FC-based ODE solvers used will be provided in Section 2.4.

Consider the general convection-diffusion PDE:

$$\begin{aligned}\vec{\kappa}(u, v) \cdot \nabla^2 C + \vec{\lambda}(u, v) \cdot \nabla C + \nu(u, v)C + Q(u, v, t) &= C_t, \quad (u, v, t) \in \Omega \times (0, T], \\ a(u, v) C(u, v, t) + b(u, v) \frac{\partial C(u, v, t)}{\partial \mathbf{n}} &= G(u, v, t), \quad (u, v, t) \in \partial\Omega \times (0, T], \\ C(u, v, 0) &= C_0(u, v), \quad (u, v) \in \Omega,\end{aligned}\tag{2.5}$$

where  $\Omega \in \mathbb{R}^2$  is a smoothly bounded domain and  $\vec{\kappa} = (\kappa^u(u, v), \kappa^v(u, v))$ ,  $\vec{\lambda} = (\lambda^u(u, v), \lambda^v(u, v))$ ,  $\nu$ ,  $Q$ ,  $a$ ,  $b$ ,  $G$ , and  $C_0$  are all smooth functions. Letting  $t^n = n\Delta t$ ,  $C^n = C(u, v, t^n)$ , and  $Q^{n+\frac{1}{2}} = Q(x, y, (n + 1/2)\Delta t)$ , we discretize the time derivative using the Crank-Nicolson centered finite

difference scheme to obtain

$$\frac{C^{n+1} - C^n}{\Delta t} = \frac{1}{2} \left( \vec{\kappa} \cdot \nabla^2 + \vec{\lambda} \cdot \nabla + \nu \right) (C^{n+1} + C^n) + Q^{n+\frac{1}{2}} + E_1(u, v, \Delta t),$$

where

$$E_1(u, v, \Delta t) \sim \mathcal{O}(\Delta t^2)$$

is the truncation error. Grouping together the terms for  $C^n$  and  $C^{n+1}$  yields

$$\begin{aligned} & \left( 1 - \frac{\Delta t}{2} \left( \vec{\kappa} \cdot \nabla^2 + \vec{\lambda} \cdot \nabla + \nu \right) \right) C^{n+1} = \\ & \left( 1 + \frac{\Delta t}{2} \left( \vec{\kappa} \cdot \nabla^2 + \vec{\lambda} \cdot \nabla + \nu \right) \right) C^n + \Delta t Q^{n+\frac{1}{2}} + \Delta t E_1(u, v, t), \end{aligned}$$

which can, in turn, be expressed in the form

$$\begin{aligned} & \left( 1 - \frac{\Delta t}{2} \left( \kappa^u \frac{\partial^2}{\partial u^2} + \lambda^u \frac{\partial}{\partial u} + \nu \right) \right) \left( 1 - \frac{\Delta t}{2} \left( \kappa^v \frac{\partial^2}{\partial v^2} + \lambda^v \frac{\partial}{\partial v} \right) \right) C^{n+1} = \\ & \left( 1 + \frac{\Delta t}{2} \left( \kappa^u \frac{\partial^2}{\partial u^2} + \lambda^u \frac{\partial}{\partial u} + \nu \right) \right) \left( 1 + \frac{\Delta t}{2} \left( \kappa^v \frac{\partial^2}{\partial v^2} + \lambda^v \frac{\partial}{\partial v} \right) \right) C^n \\ & + \Delta t Q^{n+\frac{1}{2}} + \Delta t E_1(u, v, \Delta t) + E_2(u, v, \Delta t), \end{aligned} \quad (2.6)$$

after appropriate factorization of the differential operators. A simple calculation shows that the truncation error  $E_2(u, v, \Delta t)$  that arises from factoring the differential operators is on the order of  $\mathcal{O}(\Delta t^2)$ . For ease of explanation, we introduce the notation

$$\begin{aligned} A_u &= \left( \kappa^u \frac{\partial^2}{\partial u^2} + \lambda^u \frac{\partial}{\partial u} + \nu \right), \\ A_v &= \left( \kappa^v \frac{\partial^2}{\partial v^2} + \lambda^v \frac{\partial}{\partial v} \right), \end{aligned}$$

and rewrite (2.6) in the simpler form

$$\begin{aligned} \left( 1 - \frac{\Delta t}{2} A_u \right) \left( 1 - \frac{\Delta t}{2} A_v \right) C^{n+1} &= \left( 1 + \frac{\Delta t}{2} A_u \right) \left( 1 + \frac{\Delta t}{2} A_v \right) C^n \\ &+ \Delta t Q^{n+\frac{1}{2}} + \mathcal{O}(\Delta t^2). \end{aligned} \quad (2.7)$$

Finally, making use of the approximation

$$Q^{n+\frac{1}{2}} = \frac{1}{2} \left( 1 + \frac{\Delta t}{2} A_u \right) \frac{\Delta t}{2} Q^{n+\frac{1}{4}} + \frac{1}{2} \left( 1 + \frac{\Delta t}{2} A_u \right) \frac{\Delta t}{2} Q^{n+\frac{3}{4}} + E_3(u, v, t) \quad (2.8)$$

and taking into account the commutability of the differential operators, we rearrange (2.7) to obtain

the scheme

$$\begin{aligned} \left(1 - \frac{\Delta t}{2} A_u\right) \tilde{C}^{n+\frac{1}{2}} &= \left(1 + \frac{\Delta t}{2} A_v\right) \tilde{C}^n + \frac{\Delta t}{2} Q^{n+\frac{1}{4}} \\ \left(1 - \frac{\Delta t}{2} A_v\right) \tilde{C}^{n+1} &= \left(1 + \frac{\Delta t}{2} A_u\right) \tilde{C}^{n+\frac{1}{2}} + \frac{\Delta t}{2} Q^{n+\frac{3}{4}}, \end{aligned} \quad (2.9)$$

with  $\tilde{C}^n$  providing an approximation to the exact solution  $C^n$ . Noting that terms  $\frac{\Delta t}{2} Q^{n+\frac{1}{4}}$  and  $\frac{\Delta t}{2} Q^{n+\frac{3}{4}}$  denote  $Q(u, v, (n + 1/4)\Delta t)$  and  $Q(u, v, (n + 3/4)\Delta t)$  respectively, the truncation error,  $E_3(u, v, t)$  generated from the approximation (2.8) can easily be shown to be of order  $\mathcal{O}(\Delta t^2)$ .

From examining (2.9), it is apparent that solution of  $\tilde{C}^{n+1}$  depends on the inversion of the differential operators  $(1 - \frac{\Delta t}{2} A_u)$  and  $(1 - \frac{\Delta t}{2} A_v)$ . Recalling the definitions of  $A_u$  and  $A_v$ , application of the inverse of such operators to a given function  $f$  can be expressed as solution of the one-dimensional variable-coefficient boundary value problem

$$\alpha C'' + \beta C' + \gamma C = f, \quad \begin{cases} a_l C(u_l) + b_l C'(u_l) = B_l, \\ a_r C(u_r) + b_r C'(u_r) = B_r, \end{cases} \quad (2.10)$$

for appropriate functions  $\alpha$ ,  $\beta$ , and  $\gamma$ . A prescription of the coefficients  $a_l$ ,  $b_l$ ,  $a_r$ ,  $b_r$  and boundary values  $B_l$  and  $B_r$  for the specific PDE under consideration is provided in Chapters 3 and 4. The function values of  $\alpha$ ,  $\beta$ , and  $\gamma$  depend on the operator being inverted: application of the inverse of  $(1 - \frac{\Delta t}{2} A_u)$  requires

$$\alpha(u) = -\frac{\Delta t}{2} \kappa^u, \quad \beta(u) = -\frac{\Delta t}{2} \lambda^u, \quad \gamma(u) = 1 - \frac{\Delta t}{2} \nu,$$

while application of the inverse of  $(1 - \frac{\Delta t}{2} A_v)$  requires

$$\alpha(v) = -\frac{\Delta t}{2} \kappa^v, \quad \beta(v) = -\frac{\Delta t}{2} \lambda^v, \quad \gamma(v) = 1.$$

An efficient algorithm (based on the FC method introduced in Section 2.1) for the solution of such ODEs is presented in Section 2.4.

**Remark 2.3.1** In the interest of computational efficiency, it is important to notice that repeated application of the discrete operators  $(1 + \frac{\Delta t}{2} A_v)$  and  $(1 + \frac{\Delta t}{2} A_u)$  in (2.9) does not actually require differentiation with respect to the relevant independent variable. Indeed, by rewriting (2.9) as

$$\tilde{C}^{n+1} = \left(1 - \frac{\Delta t}{2} A_v\right)^{-1} \left(1 + \frac{\Delta t}{2} A_u\right) \left(1 - \frac{\Delta t}{2} A_u\right)^{-1} \left(1 + \frac{\Delta t}{2} A_v\right) \tilde{C}^n,$$

we observe that repeated iteration of the ADI algorithm depends on computation of the combination of application and inversion of the discrete operators, i.e.,  $(1 + \frac{\Delta t}{2} A_u) (1 - \frac{\Delta t}{2} A_u)^{-1}$  and

$(1 + \frac{\Delta t}{2} A_v) (1 - \frac{\Delta t}{2} A_v)^{-1}$ . Further, letting

$$q = \left(1 + \frac{\Delta t}{2} A_u\right) \left(1 - \frac{\Delta t}{2} A_u\right)^{-1} f,$$

we obtain the equivalent system

$$\begin{aligned} -\kappa^u \frac{\Delta t}{2} C'' - \lambda^u \frac{\Delta t}{2} C - \nu \frac{\Delta t}{2} C + C &= f, \\ \kappa^u \frac{\Delta t}{2} C'' + \lambda^u \frac{\Delta t}{2} C + \nu \frac{\Delta t}{2} C + C &= q. \end{aligned} \tag{2.11}$$

Adding these equations yields

$$q = 2C - f, \tag{2.12}$$

where  $C$  is the solution to the first equation in (2.11); a similar result is easily shown for the  $v$ -dependent combination operator  $(1 + \frac{\Delta t}{2} A_v) (1 - \frac{\Delta t}{2} A_v)^{-1}$ . It thus follows that each intermediate step of (2.9) may be computed by solving the first equation in (2.11) followed by use of (2.12) or its  $v$ -dependent counterpart.

All that remains to complete the scheme is a prescription of the boundary conditions for  $\tilde{C}^{n+\frac{1}{2}}$  and  $\tilde{C}^{n+1}$ . The boundary values for  $\tilde{C}^{n+1}$  are given through enforcement of the condition

$$a(u, v) \tilde{C}^{n+1} + b(u, v) \frac{\partial \tilde{C}^{n+1}}{\partial \mathbf{n}} = G(u, v, t^{n+1})$$

for the appropriate boundary points  $(u, v) \in \partial\Omega$ . From examining (2.9), we see that the boundary values for  $\tilde{C}^{n+\frac{1}{2}}$  are given through enforcement of

$$\begin{aligned} a(u, v) \tilde{C}^{n+\frac{1}{2}} + b(u, v) \frac{\partial \tilde{C}^{n+\frac{1}{2}}}{\partial \mathbf{n}} &= \frac{1}{2} \left(1 + \frac{\Delta t}{2} A_v\right) G(u, v, t^n) + \frac{1}{2} \left(1 - \frac{\Delta t}{2} A_v\right) G(u, v, t^{n+1}) \\ &\quad + \frac{\Delta t}{4} \left(Q(u, v, t^{n+\frac{1}{4}}) - Q(u, v, t^{n+\frac{3}{4}})\right), \end{aligned} \tag{2.13}$$

for  $(u, v) \in \partial\Omega$ . Because the spatial derivatives of  $G$  needed for computation of (2.13) are not known a priori for complex domains, the FC-AD algorithm makes use of the approximate boundary condition

$$a(u, v) \tilde{C}^{n+\frac{1}{2}} + b(u, v) \frac{\partial \tilde{C}^{n+\frac{1}{2}}}{\partial \mathbf{n}} = G(x, y, (n + 1/2)\Delta t) + E_4(u, v, t),$$

where the time discretization error  $E_4(u, v, t)$  arising from this approximation can be shown to be of order  $\mathcal{O}(\Delta t^2)$ .

Accounting for the time-discretization and boundary condition errors, the overall error arising from one step of the resulting FC-AD scheme is thus of order  $\mathcal{O}(\Delta t^2)$ . While this bound predicts an overall error of  $\mathcal{O}(\Delta t)$ , we find that, in practice, the overall accuracy of the FC-AD algorithm

remains  $\mathcal{O}(\Delta t^2)$ .

**Remark 2.3.2** Because appropriate choice of the boundary conditions is dependent on the specific problem, discussion on the selection and enforcement of the boundary conditions for each magnetic drug delivery problem is provided in Chapters 3 and 4 below.

## 2.4 FC-ODE Solver

To accurately compute the solutions of the ODEs present in the ADI scheme (2.9), a suitable discrete operator that approximates the inverse of the simple differential operator

$$\alpha(x)\frac{\partial^2}{\partial x^2} + \beta(x)\frac{\partial}{\partial x} + \gamma(x) \quad (2.14)$$

is required. Recently, such discrete-solver operators based on the FC methodology have been developed for both constant-coefficient [16] and variable-coefficient [15] ODEs. Recalling that, for appropriate boundary conditions and coefficients, application of the inverse of (2.14) to a given function  $f(x)$  is equivalent to solving the boundary value problem

$$\alpha C'' + \beta C' + \gamma C = f, \quad \begin{cases} a_l C(u_l) + b_l C'(u_l) = B_l, \\ a_r C(u_r) + b_r C'(u_r) = B_r, \end{cases} \quad (2.15)$$

we will henceforth refer to these discrete-solver operators as FC-ODE solvers. In this section, we provide an outline of the constant and variable coefficient FC-ODE solvers presented in [16] and [15], respectively.

### 2.4.1 Constant Coefficient Solver

Given discrete right-hand side data  $f_j = f(x_j)$ , the constant-coefficient FC-ODE algorithm proceeds by first approximating the discrete data with its corresponding FC(Gram) continuation Fourier series

$$f^c(x) = \sum_{k \in t(M)} a_k e^{\frac{2\pi i}{b(x_r - x_l)} kx},$$

The solution of the approximate ODE,

$$\alpha \tilde{C}'' + \beta \tilde{C}' + \gamma \tilde{C} = f^c(x), \quad \begin{cases} a_l \tilde{C}(u_l) + b_l \tilde{C}'(u_l) = B_l, \\ a_r \tilde{C}(u_r) + b_r \tilde{C}'(u_r) = B_r, \end{cases} \quad (2.16)$$

is then easily obtained as the series

$$\tilde{C}(x) = \sum_{k \in t(M)} \frac{a_k}{\mu_k} e^{\frac{2\pi i}{b(x_r - x_l)} kx} + c_1 h_1(x) + c_2 h_2(x), \quad \mu_k = \gamma + \beta \frac{2\pi i k}{b(x_r - x_l)} + \alpha \left( \frac{2\pi i k}{b(x_r - x_l)} \right)^2. \quad (2.17)$$

As described below, enforcement of the boundary conditions is achieved through appropriate selection of the functions  $h_1(x)$ ,  $h_2(x)$  and their associated coefficients  $c_1$ ,  $c_2$ .

Defining  $\tilde{C}_p(x) = \sum_{k \in t(M)} \frac{a_k}{\mu_k} e^{\frac{2\pi i}{b(x_r - x_l)} kx}$ , we select  $h_1(x)$  and  $h_2(x)$  to be solutions of the corresponding homogeneous problem. That is,

$$h_1(x) = e^{r_1(x-x_l)}, \quad h_2(x) = e^{r_2(x-x_r)}, \quad \text{where} \\ r_1 = \frac{-\beta - \sqrt{\beta^2 - 4\alpha\gamma}}{2\alpha}, \quad r_2 = \frac{-\beta + \sqrt{\beta^2 - 4\alpha\gamma}}{2\alpha}.$$

The coefficients  $c_1$  and  $c_2$  are then obtained through solution of the  $2 \times 2$  system of equations

$$\begin{pmatrix} a_l h_1(x_l) + b_l h_1'(x_l) & a_l h_2(x_l) + b_l h_2'(x_l) \\ a_r h_1(x_r) + b_r h_1'(x_r) & a_r h_2(x_r) + b_r h_2'(x_r) \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} B_l - a_l \tilde{C}_p(x_l) - b_l \tilde{C}_p'(x_l) \\ B_r - a_r \tilde{C}_p(x_r) - b_r \tilde{C}_p'(x_r) \end{pmatrix}; \quad (2.18)$$

clearly the resulting full solution  $\tilde{C}(x) = \tilde{C}_p(x) + c_1 h_1(x) + c_2 h_2(x)$  satisfies the boundary conditions.

## 2.4.2 Variable Coefficient Solver

While solution of such linear ODEs may be obtained in a fairly straightforward manner when the coefficients  $\alpha$ ,  $\beta$ , and  $\gamma$  are constant, this is not the case when the coefficients are variable. Indeed, the simple previous representation of the Fourier coefficients of the solution in terms of the Fourier coefficients of the right-hand side (as seen in (2.17)) now depends on convolutions with the Fourier series of the coefficient functions. Recently, a new FC-based solver has been developed [15] for variable coefficient ODEs; we provide a brief outline of its implementation in what follows.

Consider again the ODE in equation (2.15). A new ODE approximating (2.15) results from replacing each function in (2.15) with its corresponding Fourier continuation:

$$\alpha^c(x) \frac{d^2 C^c(x)}{dx^2} + \beta^c(x) \frac{dC^c(x)}{dx} + \gamma^c(x) C^c(x) = C^c(x). \quad (2.19)$$

Because the Fourier continuation of a function is periodic by construction, the derivatives of  $C^c(x)$  may be represented as operations on their Fourier coefficients  $\hat{C}_k$ . This observation prompts the

re-expression of (2.19) as

$$\alpha^c(x) \sum_{k \in t(M)} \left( \frac{2\pi ik}{b} \right)^2 \hat{C}_k e^{\frac{2\pi ik}{b}x} + \beta^c(x) \sum_{k \in t(M)} \frac{2\pi ik}{b} \hat{C}_k e^{\frac{2\pi ik}{b}x} + \gamma^c(x) \sum_{k \in t(M)} \hat{C}_k e^{\frac{2\pi ik}{b}x} = f^c(x), \quad (2.20)$$

where the bandwidth  $M$  and the period  $b$  are prescribed via the FC method discussed in Section 2.1. From further examination, we see that equation (2.20) applied over a discrete grid is simply a linear system of equations for the Fourier coefficients  $\hat{C}_k$ ; an efficient method of solution can be provided by GMRES with an appropriate preconditioner. As is fully explored in reference [15], such a preconditioner can be obtained from a second-order implicit finite difference solution of (2.19) with periodic boundary conditions. Once the Fourier coefficients  $\hat{C}_k$  are obtained, a solution to (2.19), and subsequently an approximation to the solution of (2.15), is produced through direct application of the IFFT algorithm to  $\hat{C}_k$ ; we denote this solution as  $C_p(x)$ .

Similarly to the constant-coefficient FC-ODE solver, satisfaction of the boundary conditions is enforced through addition of an appropriate linear combination of functions, i.e.,  $C(x) = C_p(x) + c_1 h_1(x) + c_2 h_2(x)$ . Such functions may again be provided by the solutions of the homogenous ODE

$$\alpha^c(x) \frac{d^2 C^c(x)}{dx^2} + \beta^c(x) \frac{d C^c(x)}{dx} + \gamma^c(x) C^c(x) = 0. \quad (2.21)$$

In practice, these functions are found by applying the above GMRES-based solver to the ODE (2.21) after replacing the zero right-hand side with the smooth periodic functions

$$\begin{aligned} f_1(x) &= \begin{cases} 0, & x \in [0, 1] \\ e^{-1/(1+n(x)^2)}, & x \in (1, b) \end{cases} \quad \text{and} \\ f_2(x) &= \begin{cases} 0 & x \in [0, 1] \\ n(x)e^{-1/(1+n(x)^2)}, & x \in (1, b) \end{cases}, \quad \text{where} \\ n(x) &= \frac{2(x(N-1) - (x_N(N-1) + 1))}{n_d - 1}. \end{aligned}$$

This choice of right-hand side functions ensures that the solutions of their corresponding ODEs,  $h_1(x)$  and  $h_2(x)$  respectively, satisfy (2.21) in the original interval  $[0, 1]$ . Once the approximate solutions to the homogeneous ODE are obtained, the coefficients  $c_1$  and  $c_2$  are found through the inversion of the system of equations (2.18), where the function and derivative values at the boundary points  $x_l$  and  $x_r$  are obtained from appropriate Fourier series expansions.

**Remark 2.4.1** In the interest of computational efficiency for use in conjunction with the ADI algorithm, it is important to note that several of the elements vital to the constant and variable coefficient FC-ODE solvers do not need to be computed at every iteration. Indeed, observing that, for a given PDE, the coefficients of the ODEs inherent in the ADI algorithm do not depend on

the iteration-step index  $n$ , the functions  $h_1(x)$ ,  $h_2(x)$  and the matrix in the system (2.18) may be precomputed and stored before initiating the ADI iterations. Further, if the variable coefficient solver is used, the continuations of the coefficient functions  $\alpha^c$ ,  $\beta^c$ ,  $\gamma^c$  and the finite difference GMRES preconditioner matrix may also be precomputed and stored.

## Chapter 3

# Vessel-Membrane-Tissue Model

The goal of magnetic drug delivery is to use magnetic fields to direct and confine magnetically responsive particles (which, containing therapeutic agents, are injected into the bloodstream) to specific regions in a patient’s body—thus allowing for focused treatment in an area of interest. To design a method leading to confinement of the magnetically responsive particles to a particular region of the body, a predictive capability must be used to evaluate the effects of external magnetic forces on the convection and diffusion of magnetic particles through the bloodstream and in membranes and tissue. A simplified, but effective mathematical model of the resulting Vessel-Membrane-Tissue (VMT) convection diffusion problem has been proposed by Grief and Richardson [26] and was recently extensively analyzed in [42]. The aim of the contribution [42] was to determine how the interplay between the advective and diffusive forces generated from the blood velocity and magnetic field affects the flow of magnetically responsive particles through the bloodstream and their diffusion into surrounding tissue without the added difficulties of a complex geometry. As mentioned in [42], the numerical solution of the Grief and Richardson model proved to be very challenging and required the development of a sophisticated solver—which we call the VMT solver; in this chapter, we provide a detailed description of this solver and we briefly describe its application to the problems considered in [42].

### 3.1 Introduction to VMT Model

The idealized geometry of the Grief and Richardson model consists of a lateral cross section of a blood vessel, including the endothelial layer (membrane) and some surrounding tissue. A magnet is situated far below, see Figure 3.1. Each layer has a different diffusion and magnetic advection coefficient, with the relationships between the various coefficients given by the Renkin and Renkin Tissue numbers,

$$\mathcal{R} = \frac{D^m}{D^v} \quad \text{and} \quad \mathcal{R}^t = \frac{D^t}{D^v},$$

respectively. For notational convenience, in this chapter we make use of the superscripts  $v$ ,  $m$ , and  $t$  to denote the value of the quantity in the corresponding vessel ( $v$ ), membrane ( $m$ ), or tissue ( $t$ ) layer.

The characterizing equation of the Grief and Richardson model is the hyperbolic convection-diffusion PDE,

$$\frac{\partial}{\partial t} C(\vec{r}, t) = -\nabla \cdot \left[ C(\vec{r}, t) \vec{V}_{\text{blood}}(\vec{r}, t) - D(\vec{r}) \nabla C(\vec{r}, t) + k(\vec{r}) C(\vec{r}, t) \nabla \left( |\vec{H}(\vec{r}, t)|^2 \right) \right],$$

where  $C(\vec{r}, t)$  is the concentration of magnetic particles in the blood,  $\vec{V}_{\text{blood}}(\vec{r}, t)$  is the blood convection,  $D(\vec{r})$  is the diffusion coefficient of particles within the bloodstream, and  $k(\vec{r})$  is the magnetic drift coefficient. The term  $\nabla \left( |\vec{H}(\vec{r}, t)|^2 \right)$ , where  $\vec{H}(\vec{r}, t)$  represents the externally applied magnetic field, is referred to as the *control*.

For the preliminary analyses performed in [42], several further simplifications were made: the control is set to be constant, the magnetic drift coefficient  $k(\vec{r})$  is simplified to 1, the diffusion term  $D$  is assumed to be constant in each layer, and the blood velocity  $\vec{V}_{\text{blood}}$  in the vessel is assumed to be determined by a parabolic profile. A zero convective flux is required on all the outside boundaries except for the left  $x$ -boundary in vessel layer, where the concentration  $C(x, y, t)$  is set to 1. Initially, we assume the concentration equals 1 throughout the vessel layer and it equals 0 everywhere else. Applying these simplifications gives us the following VMT model:

$$\begin{aligned} C_t &= D\nabla^2 C - v_{\text{blood}} C_x + M C_y, & (x, y, t) &\in \Omega \times (0, T], \\ C &= 1, & x = x_l, y &\geq I^{vm}, t \in (0, T], \\ C_x &= 0, & x = x_l, x_r, y &< I^{vm}, t \in (0, T], \\ C_y &= 0, & y = y_l, y_r, t &\in (0, T], \\ C(x, y, 0) &= \begin{cases} 1; & y \geq I^{vm}, \\ 0; & y < I^{vm}, \end{cases} \end{aligned} \tag{3.1}$$

where  $D$  is the piecewise constant diffusion,  $v_{\text{blood}}$  is the (parabolic) blood velocity,  $M$  is the piecewise constant downward magnetic force,  $I^{vm}$  and  $I^{mt}$  describe the location of the vessel-membrane and membrane-tissue interfaces respectively, and the domain of solution is given by  $\Omega = [x_l, x_r] \times [y_l, y_r]$ . In the vessel layer,  $v_{\text{blood}}$  is given by the blunted parabolic profile,

$$v_{\text{blood}}(y) = \begin{cases} \left( 1 - \frac{|2y - (I^{vm} + I^{mt})|}{I^{vm} - I^{mt}} \right)^g & ; y \geq I^{vm}, \\ 0 & ; y < I^{vm}, \end{cases}$$

where  $g$  is some constant. In all the numerical simulations shown in Section 3.3, we chose  $g = 9$ .

Numerical solution of the VMT model for realistic values of diffusion coefficients has proven

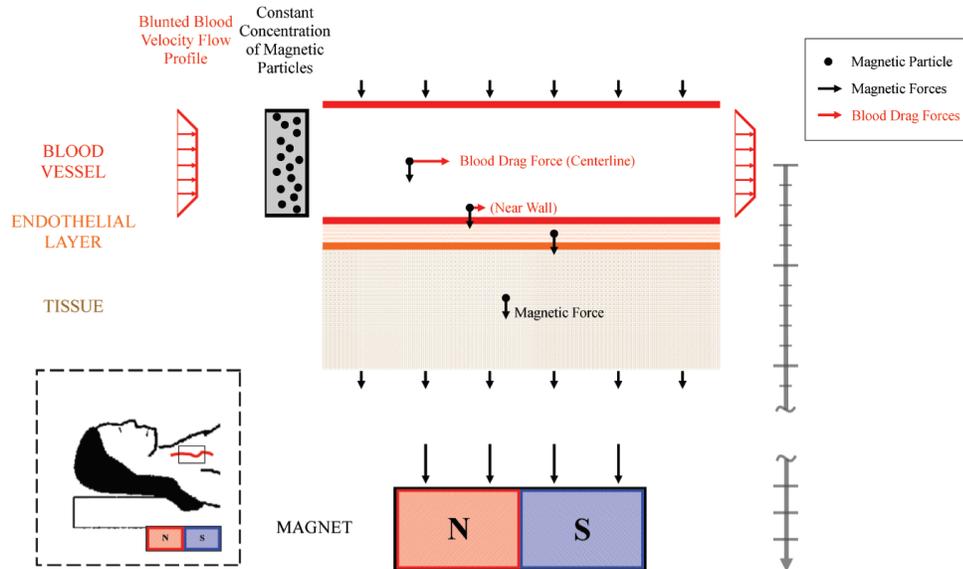


Figure 3.1: VMT geometry, as depicted in [42].

highly challenging to numerical solvers for several reasons, amongst which figures prominently the strong concentration build-up, or *boundary layer*, that appears at the vessel-membrane interface. Indeed, due to the discontinuity in diffusion coefficients over each layer, sharp boundary layers occur when the magnetic forces acting on the particles are strong enough to overcome the blood velocity, an effect that is particularly noticeable for the extremely small nondimensional diffusion coefficients present in realistic models of capillaries. Additionally, the advective forces provided by the blood velocity are significantly more powerful than the diffusive forces, thus giving rise to greatly disparate time scales: in order to produce a steady state solution in a reasonable amount of time, an efficient time evolution methodology must be developed.

To overcome these difficulties, each of the VMT solvers presented in this chapter employs a combination of the following techniques: 1) Use of domain meshes that can adequately resolve boundary layers while controlling computational costs, 2) The Alternating Directions Implicit (ADI) method to overcome the severe CFL condition imposed by the fine spatial discretization required by the previously mentioned meshing scheme, 3) An on-and-off fluid-freezing methodology that allows for efficient treatment of the multiple timescales that coexist in the problem, and 4) A highly accelerated time-stepping procedure that enables evaluation of steady states in tissue and membrane layers. Each of the above four procedures are described in detail in the following subsections.

In analyses prior to the work [42], numerical solution of the VMT problem was performed through use of the finite-element-based commercial software COMSOL Multiphysics. While capable of solving the VMT problem for large diffusion coefficients, the COMSOL software encountered many difficulties for the small diffusion coefficients inherent in realistic instances of the VMT model, es-

pecially for larger values of the magnetization. For example, in the preliminary studies performed on the COMSOL software, taking the diffusion and magnetization coefficients to equal  $10^{-4}$  and  $10^{-3}$  respectively, a steady state solution was reached in 36 hours of run time on a 3.16 GHz single processor of a quad-core Intel Xeon CPR X5460 computer with 32 GB of memory; the corresponding memory requirements to obtain the steady state solution for values of the diffusion and magnetization equal to  $10^{-5}$  exceeded the amount available on the same computer. Clearly, the medically relevant VMT model with diffusion constant equal to  $10^{-7}$  and magnetization on the order of  $10^{-6}$  lies far outside the domain of applicability of the COMSOL software in the said computer. In contrast, the VMT solvers described in this chapter produced accurate solutions for cases relevant to the contribution [42] without difficulty. For example, on a computer with a 2.66 GHz Intel Core 2 Duo processor and 4GB of memory, our solvers provide the required numerical solutions to the  $10^{-4}$  and  $10^{-5}$  problems in under five minutes using 3 MB (not GB!) of memory and under 14 minutes using 15 MB of memory, respectively. One of the most challenging cases considered in [42], in turn, for which the diffusion constant equals  $10^{-7}$ —a case that is very far from feasible for other methods—completed, using 25 MB of memory, in a six hour run.

## 3.2 VMT Solvers

### 3.2.1 Domain Meshing

As mentioned in the previous section, when the magnetic forces acting on the particles are strong enough to overcome the blood velocity, a boundary layer appears at the vessel-membrane interface. To resolve the boundary layer numerically, the VMT solvers presented in this chapter make use of either one of the following meshing techniques, see also Figure 3.2.1.

#### 3.2.1.1 Graded Mesh

As first presented in [42], one method for achieving accurate numerical resolution of the boundary layer is provided through implementation an exponential change of variables in the vessel:

$$\xi = e^{-Gy},$$

where  $G$  is a prescribed constant. Recalling that  $I^{mt}$  and  $I^{vm}$  denote the locations of the tissue-membrane and membrane-vessel interfaces respectively, the discrete grids used for solution in  $y$  are thus given by

$$\begin{aligned} y_t &:= \{y_j = j(I^{mt} - I_t)/(N^t - 1) \mid j = 0, \dots, N^t - 1\}, \\ y_m &:= \{y_j = j(I^{vm} - I^{mt})/(N^m - 1) \mid j = 0, \dots, N^m - 1\}, \\ \xi_v &:= \{\xi_j = j(e^{-GI_r} - e^{-GI^{vm}})/(N^v - 1) \mid j = 0, \dots, N^v - 1\}, \end{aligned} \tag{3.2}$$

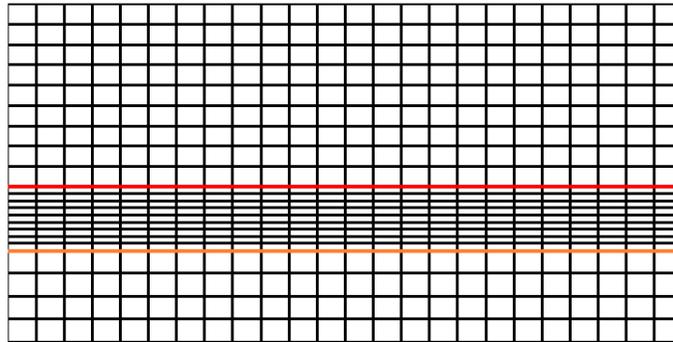


Figure 3.2: Meshing of the domain of solution  $\Omega$  of the PDE (3.1). The red line represents the vessel-membrane interface  $I^{vm}$  and the orange line represents the membrane-tissue interface  $I^{mt}$ . Notice that the spatial resolutions in the  $y$ -direction vary over each layer; further details on the selection of the spatial step size in  $y$  is provided in Section 3.2.1. In all numerical results presented in Section 3.3, we used an equispaced grid in the  $x$ -direction.

where  $N^t$ ,  $N^m$ , and  $N^v$  are the number of grid points used in the  $y$ -direction in the tissue, membrane, and vessel layers respectively. An example of these grids for a particular  $D^v$  and  $M^v$  is displayed in Figure 3.3(a).

Application of this change of variables transforms the PDE (3.1) in the vessel into the variable-coefficient convection-diffusion equation

$$\check{C}_t = D^v(\check{C}_{xx} + G^2\xi^2\check{C}_{\xi\xi}) - v_{\text{blood}}\check{C}_x + G\xi(GD^v + M^v)\check{C}_\xi, \quad (3.3)$$

where  $\check{C}(x, \xi, t) = C(x, y, t)$ .

For clarity, and to highlight the differences between VMT solvers using the above graded mesh and those using alternate meshing strategies, we henceforth refer to any VMT method using an exponential change of variables as a Graded VMT Solver. In all the numerical simulations of the finite differences-based Graded VMT solver shown in Section 3.3, we chose  $G = -D^v/8M^v$  and  $N^v = 2N_y/5$ ,  $N^m = 2N_y/5$ , and  $N^t = N_y/5$ , where  $N_y$  is the number of grid points in the  $y$ -direction.

### 3.2.1.2 Multiple Equispaced Meshes

An alternate approach for the resolution of the boundary layer is provided through the inclusion of two additional equispaced meshes located near the boundary layer, such as the those displayed in Figure 3.3(b). The rationale behind the choice of these meshes lies in computational efficiency: because the modified PDE (3.3) arising from implementation of the exponentially graded mesh has variable coefficients, the FC-AD method our VMT solvers are based on would require use of

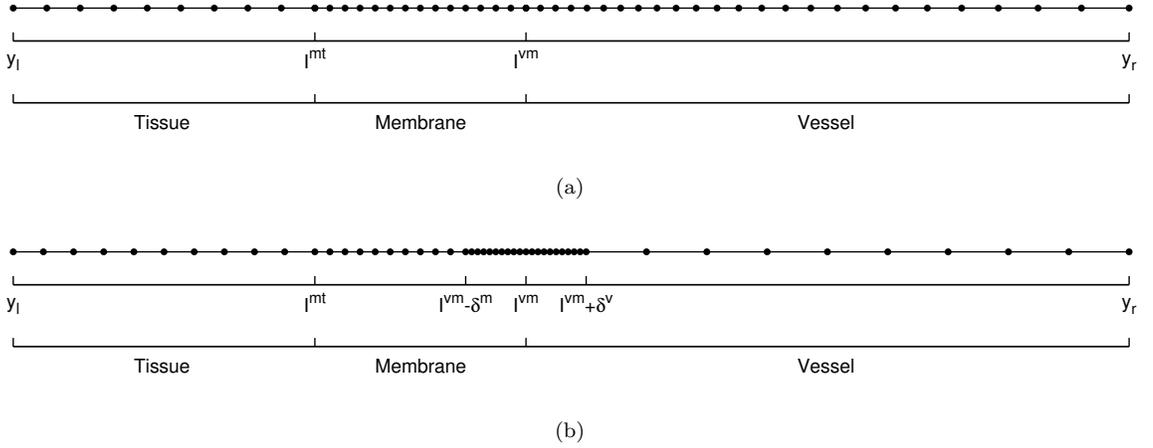


Figure 3.3: Grids used by the Graded and Standard VMT solvers. Figure (a) depicts the exponentially graded mesh given by (3.2) with  $G = -D^v/8M^v$  and Figure (b) displays the multiple equispaced meshes given by (3.4) with  $\delta^m = D^v/M^v$ ,  $\delta^v = D^v/M^v$ .

the variable-coefficient FC-ODE solver; due to the GMRES-step, the variable-coefficient FC-ODE algorithm is more computationally expensive than its constant-coefficient counterpart. In contrast, the use of additional equispaced meshes does not depend on transformation of the constant-coefficient PDE (3.1), and so efficient numerical solution is provided by the FC-AD method with the constant-coefficient FC-ODE solver.

In detail, we make use of the discrete grids

$$\begin{aligned}
 y_t &:= \{y_j = j(I^{mt} - y_l)/(N^t - 1) \mid j = 0, \dots, N^t - 1\}, \\
 y_{cm} &:= \{y_j = j(I^{vm} - \delta^m - I^{mt})/(N^{cm} - 1) \mid j = 0, \dots, N^{cm} - 1\}, \\
 y_{fm} &:= \{y_j = j\delta^m/(N^{fm} - 1) \mid j = 0, \dots, N^{fm} - 1\}, \\
 y_{fv} &:= \{y_j = j\delta^v/(N^{fv} - 1) \mid j = 0, \dots, N^{fv} - 1\}, \\
 y_{cv} &:= \{y_j = j(y_r - I^{vm} + \delta^v)/(N^{cv} - 1) \mid j = 0, \dots, N^{cv} - 1\},
 \end{aligned} \tag{3.4}$$

where  $\delta^m$  and  $\delta^v$  are selected to ensure adequate resolution of the boundary layer. For all the numerical results presented in Section 3.3, we chose  $\delta^m = D^v/M^v$ ,  $\delta^v = D^v/M^v$  and  $N^t = N_y/5$ ,  $N^{cm} = N_y/5$ ,  $N^{fm} = N_y/5$ ,  $N^{fv} = N_y/5$ ,  $N^{cv} = N_y/5$ .

### 3.2.2 ADI

In order for an explicit method for our problem to be numerically stable, it needs to satisfy the well-known diffusive CFL condition:  $\Delta t \sim \mathcal{O}(\Delta x^2)$ . The choice of either of the meshing schemes described above therefore places a severe restriction on the time step used in explicit methods. To

	Vessel		Membrane	Tissue
	Standard	Graded Mesh	Standard	Standard
$(u, v)$	$(x, y)$	$(x, \xi)$	$(x, y)$	$(x, y)$
$\vec{\kappa}(u, v)$	$(D^v, D^v)$	$(D^v, D^v G^2 \xi^2)$	$(D^m, D^m)$	$(D^t, D^t)$
$\vec{\lambda}(u, v)$	$(-v_{\text{blood}}, M^v)$	$(-v_{\text{blood}}, G\xi(GD^v + M^v))$	$(0, M^m)$	$(0, M^t)$
$\nu(u, v)$	0	0	0	0

Table 3.1: Table of coefficients and coordinates used in each layer, including both the standard equispaced and exponentially graded meshes in the vessel. The far left column provides the corresponding notation used in Section 2.3 for the description of the ADI algorithm.

overcome this restriction, we make use of an efficient solver that does not incur a CFL condition, namely, the Alternating Directions Implicit (ADI).

Numerical solvers for the full VMT problem arise from implementation of the ADI methodology in each grid: solution of the ODEs inherent in the ADI algorithm via finite differences yield FD-based VMT solvers, while FC-based VMT solvers result from application of the FC-AD method in each grid to the associated PDE. For all the numerical results presented in Section 3.3 that were obtained from implementation of FD-based VMT solvers, we make use of the standard centered-difference finite difference scheme.

Recalling the ADI methodology previously described in Section 2.3, we note that application of the FC-AD algorithm to the VMT model depends on appropriate definition of the coordinates  $(u, v)$  and functions  $\vec{\kappa}$ ,  $\vec{\lambda}$ ,  $\nu$ ,  $Q$ ,  $a$ ,  $b$ ,  $G$ , and  $C_0$ . From inspection we may immediately note that the inhomogeneity  $Q(u, v, t)$  is simply 0 and the initial condition  $C_0$  is given by (3.1). However, because of the variety of meshes and change of variables available for use in any given VMT solver, definition of the coefficient functions  $\vec{\kappa}$ ,  $\vec{\lambda}$ , and  $\nu$  in each layer depends on the specific combination of techniques employed. A listing of the appropriate choices of the coordinates  $(u, v)$  and coefficient functions  $\vec{\kappa}$ ,  $\vec{\lambda}$ ,  $\nu$  required for various implementations of the VMT solver is provided in Table 3.1.

Due to the various diffusive and advective terms used in each layer, the boundary conditions at each layer interface must be treated with care to ensure a correct physical solution. Section 3.2.3 discusses the enforcement of these boundary conditions, henceforth referred to as “jump conditions”, for FD- and FC-based VMT solvers.

### 3.2.3 Jump conditions

It is important to note that, due to the discontinuous nature of the diffusive and advective terms, additional “jump conditions” must be applied at all layer and mesh interfaces to ensure correct physical solutions. We thus require continuity of the concentration  $C(x, y, t)$  and the flux  $DC_y(x, y, t) + MC(x, y, t)$  to be enforced across each interface.

### 3.2.3.1 FD-Based Methods

For the FD-based ADI method, satisfaction of these jump conditions is achieved through a implementation combination of forward and backward finite-difference schemes at each layer. For example, and for ease of explanation, using the forward and backward Euler discretization schemes the jump conditions at the membrane-tissue interface require

$$\left(-\frac{D^t}{h_y^t}\right)C_{i,I-1}^n + \left(\frac{D^t}{h^t} + \frac{D^m}{h^m} + M^t - M^m\right)C_{i,I}^n + \left(-\frac{D^m}{h^m}\right)C_{i,I+1}^n = 0, \quad (3.5)$$

where  $h^m$  and  $h^t$  are the spatial step sizes in the membrane and tissue respectively, and  $C_{i,I}^n$  is the approximate concentration at  $x = h_x i$ ,  $y = I^m t$ , and time  $t^n = n\Delta t$ . The index  $I$  corresponds to the membrane-tissue index after appropriate discretization. Note that continuity of  $C$  is inherently enforced through the satisfaction of (3.5). The jump conditions at other interfaces are similarly enforced.

Satisfaction of the zero-flux boundary conditions at the domain boundaries is achieved through enforcement of

$$(C_{i,1}^n - C_{i,0}^n)/h^t = 0, \quad (C_{i,N_y}^n - C_{i,N_y-1}^n)/h^v = 0,$$

where we have again approximated  $\partial C/\partial y$  using the forward and backward Euler discretization schemes respectively.

For all implementations of FD-based VMT solvers presented in Section 3.3, we made use of the second-order forward and backward finite difference schemes

$$\begin{aligned} \frac{\partial C_{i,j}^n}{\partial y} &= \frac{1}{2h}(-3C_{i,j}^n + 4C_{i,j+1}^n - C_{i,j+2}^n) + \mathcal{O}(h^2), \quad \text{and} \\ \frac{\partial C_{i,j}^n}{\partial y} &= \frac{1}{2h}(3C_{i,j}^n - 4C_{i,j-1}^n + C_{i,j-2}^n) + \mathcal{O}(h^2), \end{aligned}$$

respectively, to enforce both the boundary conditions in  $x$  and  $y$  and jump conditions at each interface.

### 3.2.3.2 FC-Based Methods

Enforcement of the jump conditions for the FC-based solver requires additional considerations. For ease of explanation, in this section we will consider only the simplest meshing scheme: one equispaced mesh in each layer. Generalization to more complex meshing schemes may be achieved in a fairly straightforward manner.

Returning to Section 2.4, we recall that the FC-ODE method enforces boundary conditions through addition of an appropriate linear combination of solutions to the associated homogeneous ODE. However, because the jump conditions require the continuity of the concentration and the

flux across each interface, the systems determining the corresponding coefficients in each layer may not be treated separately. Indeed, the  $2 \times 2$  system shown in equation (2.18) now becomes a  $6 \times 6$  system relating the coefficients in each layer as follows

$$\begin{pmatrix} (h_1^t)'|_{y_l} & (h_2^t)'|_{y_l} & 0 & 0 & 0 & 0 \\ h_1^t|_{I^{mt}} & h_2^t|_{I^{mt}} & -h_1^m|_{I^{mt}} & -h_2^m|_{I^{mt}} & 0 & 0 \\ \mathcal{F}_{mt}^t(h_1^t) & \mathcal{F}_{mt}^t(h_2^t) & -\mathcal{F}_{mt}^m(h_1^m) & -\mathcal{F}_{mt}^m(h_2^m) & 0 & 0 \\ 0 & 0 & h_1^m|_{I^{vm}} & h_2^m|_{I^{vm}} & -h_1^v|_{I^{vm}} & -h_2^v|_{I^{vm}} \\ 0 & 0 & \mathcal{F}_{vm}^m(h_1^m) & \mathcal{F}_{vm}^m(h_2^m) & -\mathcal{F}_{vm}^v(h_1^v) & -\mathcal{F}_{vm}^v(h_2^v) \\ 0 & 0 & 0 & 0 & (h_1^v)'|_{y_r} & (h_2^v)'|_{y_r} \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \end{pmatrix} = \begin{pmatrix} -(C_p^t)'|_{y_l} \\ C_p^m|_{I^{mt}} - C_p^t|_{I^{mt}} \\ \mathcal{F}_{mt}^m(C_p^m) - \mathcal{F}_{mt}^t(C_p^t) \\ C_p^v|_{I^{vm}} - C_p^m|_{I^{vm}} \\ \mathcal{F}_{vm}^v(C_p^v) - \mathcal{F}_{vm}^m(C_p^m) \\ -(C_p^v)'|_{y_r} \end{pmatrix},$$

where  $\mathcal{F}(f) = Df_y + Mf$  is the flux in the layer specified by the associated superscript and evaluated at the interface specified by the associated subscript.

### 3.2.4 Fluid-Freezing Methodology

To resolve the advection in the vessel, we begin by using a small time step,  $\Delta t = 0.5$ . This presents a problem, however: because the magnitude of the diffusion in the membrane and tissue is typically taken to be extremely small, the concentration will not reach steady state in a reasonable amount of time if the time step is kept small. At the same time, if the time step is taken to be much larger, we risk being unable to resolve the advection in the vessel. To overcome this difficulty, we periodically “freeze” the concentration in the vessel and evolve only in the membrane and tissue using a large time step. Once the freezing approximation is no longer accurate, the vessel is “unfrozen” and the time-step significantly reduced. The entire system is then evolved at the reduced time step until the freezing process can be used again.

In detail, freezing occurs when the concentration in the vessel has reached a near steady state; the concentration is said to have reached a steady state in a specified layer when the maximum relative residue,

$$\frac{|C(x, y, t^{n+1}) - C(x, y, t^n)|}{|C(x, y, t^{n+1})|}, \quad (3.6)$$

over the relevant layer points  $(x, y)$  is smaller than a prescribed tolerance. In the remainder of this text, we denote the steady state tolerance for the concentration in the vessel, or freezing tolerance, by  $\sigma_f$ . The accuracy of the freezing approximation is determined by measuring fulfillment of the jump conditions (see Section 3.2.3)—if the jump conditions are not satisfied, we unfreeze the vessel, reduce the time-step, and evolve the complete system, as described above.

### 3.2.5 Fast Time-Stepping

The final element of the VMT solver is the inclusion of an efficient time-stepping scheme: to efficiently obtain steady states in the membrane and tissue regions for each frozen vessel configuration, we perform a transformation of the differential operators that allows us to take advantage of a fast steady state solver based on selection of adequately chosen, very large time steps. The required transformation is actually a change of unknown,

$$C(x, y, t) = \omega(x, y, t)e^{M^v y/(2D^v)}, \quad y \leq I^{vm},$$

that eliminates the magnetic term in the PDE for the membrane and tissue—actually converting the convection diffusion spatial operator to a spatial operator of Helmholtz type:

$$\omega_t = D\nabla^2\omega - \frac{M}{4}\omega.$$

We then select time steps in a form described in [55], that is

$$\Delta t_n = \frac{(h^t)^2}{b} \left(\frac{b}{a}\right)^{\frac{n+1}{\Lambda}},$$

where

$$a = 4 \sin^2\left(\frac{\pi h^t}{2}\right), \quad b = 4 \cos^2\left(\frac{\pi x h^t}{2}\right),$$

$h = (I^{mt} - y_l)/19$  and  $n$  is the iteration number. The choice of  $\Lambda$  determines how rapidly the time steps increase and depends on the relationship between the diffusion and magnetic force coefficients,  $D$  and  $M$  and on the VMT solver used; further details on the appropriate selection of  $\Lambda$  are provided in Section 3.3. To ensure the solution has reached steady state, once the solution time is greater than time

$$T = \frac{2}{\mathcal{R}^2 \mathcal{R}^t} \left(\frac{M^v}{D^v}\right)^2 \log\left(\frac{\mathcal{R}^t}{\mathcal{R}}\right),$$

the fluid-freezing methodology is terminated and the entire system is evolved using a small time step until the maximum relative residue (3.6) in the membrane and tissue is smaller than a prescribed tolerance,  $\sigma_s$ . The choice of  $T$  arises from a modification of the steady state time for a simple capillary model, the Krogh tissue cylinder model [10, ?], that incorporates the multiple layers and magnetic forces present in the VMT model. For all the simulations presented in Section 3.3, after termination of the freezing methodology at the appropriate time, we use the small time step  $\Delta t = 0.5$  until steady state is reached. For all the numerical results presented in Section 3.3, we selected the steady state tolerance  $\sigma_s = 10^{-5}$ .

While useful for motivating the theory behind the selection of such a time-stepping scheme, im-

plementation of the transformation is not needed in practice. Indeed, as the relevant transformation is simply a multiplication by an exponential, the fast time-stepping scheme is also directly applicable to the concentration  $C(x, y, t)$ .

In summary, the use of this fast time-stepping strategy in conjunction with the freezing methodology described in the previous section gives rise to fast convergence for the VMT model.

**Remark 3.2.1** Returning to the ADI methodology described in Section 2.3, recall that repeated application and inversion of the relevant differential operators  $(1 + \frac{\Delta t}{2}A_x)$  and  $(1 + \frac{\Delta t}{2}A_y)$  in equations (2.9) may be simplified through use of the relation (2.12). However, because the time step  $\Delta t$  is not kept constant with the introduction of the above fast time-stepping scheme, application of the differential operator  $(1 + \frac{\Delta t}{2}A_y)$  must be performed directly with every change in time step, i.e., every time the vessel is frozen or unfrozen. This application maybe performed through either FD or FC methodologies. A simple second-order centered difference FD scheme provides the required operator for our FD-based VMT solvers. The FC operator arises from use of the differential operator (2.3), as described in Remark 2.1.1. It is important to note that for the extremely large time steps required for the VMT model with small diffusion constants, e.g., time steps on the order of  $\Delta t \sim 10^6$  are required for the  $D = 10^{-7}$  example presented in Section 3.3, a filtering procedure is necessary to ensure stability of the FC differential operator in the  $y$ -variable. We make use of the filter presented in [2]:

$$\hat{w}_k = \exp\left(-\alpha\left(\frac{2k}{N_c}\right)^{2p}\right)\hat{f}_k,$$

where  $N_c$  is the length of the vector  $\hat{f}$  of Fourier coefficients and  $\hat{w}$  denotes the filtered Fourier coefficients. Following the parameter selections detailed in [2], for all the numerical simulations in this chapter, we used the filter parameters  $\alpha = 16 \ln 10$  and  $p = \lfloor 3N/5 \rfloor$ , where  $N$  is the number of grid points in the relevant layer.

### 3.3 Numerical Results

As mentioned in Section 3.1, preliminary numerical tests leading to the results demonstrated in reference [42] were conducted using the commercial software COMSOL Multiphysics. While capable of solving the VMT problem (3.1) for large values of  $D^v$ , the COMSOL software encountered many difficulties for small values of this parameter, especially for large values of the magnetization coefficient  $M^v$ . For example, for the choices  $D^v = 10^{-4}$ ,  $M^v = 10^{-3}$ ,  $\mathcal{R} = 10^{-2}$ ,  $\mathcal{R}^t = 0.1$ , a COMSOL steady state solution was obtained in 36 hours of run time on a 3.16 GHz single processor of a quad-core Intel Xeon CPR X5460 computer with 32 GB of memory; the steady state solution using smaller parameter choices such as  $D^v = 10^{-5}$ ,  $M^v = 10^{-5}$ ,  $\mathcal{R} = 10^{-2}$ ,  $\mathcal{R}^t = 0.1$ , on the other hand, would have required a memory allocation beyond the 32GB available in the said computer. In comparison,

some realistic models of capillaries have a diffusion coefficient on the order of  $D^v \sim 10^{-7}$ ; clearly a more specialized solver is required if such cases are to be tackled successfully.

(Henceforth, for clarity, we no longer include the Renkin and Renkin Tissue numbers,  $\mathcal{R} = 10^{-2}$  and  $\mathcal{R}^t = 0.1$  respectively, and instead refer to each of the parameter sets by only their diffusion and magnetization coefficients,  $D^v$  and  $M^v$ .)

The goal of the analyses performed by Nacev et al. [42] was to determine, for a given  $D^v$  and  $M^v$ , whether 1) the concentration forms a boundary layer at the vessel-membrane interface, 2) the concentration remains in the vessel until steady state, i.e., is dominated by the blood velocity, or 3) the concentration simply diffuses in the surrounding tissue, i.e., is dominated by the magnetic forces. To determine if a boundary layer is present, reference [42] makes use of the criterion described in reference [29], that is: a boundary layer is said to be present at a particular point if the value of the concentration at that point is greater than 101% of the inlet concentration. In particular, this criterion was applied to the concentration at the point  $\boldsymbol{\rho} = ((x_r - x_l)/2, I^{vm})$ , the center of the vessel-membrane interface. Because the boundary condition at  $x_l$  requires the concentration at the blood vessel to equal 1, a boundary layer is said to form at the vessel-membrane interface if  $C(\boldsymbol{\rho}, t) \geq 1.01$  at steady state [42]. Thus, a reasonably accurate steady state solution is necessary in order to determine whether a boundary layer has or has not formed. In any case, comparisons with other software and the accuracy of the calculations themselves, which require long-time evaluation of a solution dependent on an extremely wide range of diffusivities, only become credible when a significant convergence pattern is established. In our work we thus performed a convergence analysis that guarantees an overall relative accuracy of  $10^{-5}$ .

Figure 3.3 displays three steady state solutions obtained from C++ implementations of the FC-VMT solver for various parameter sets over the domain  $[0, 24] \times [0, 1.85]$ ; the membrane-tissue and vessel-membrane interfaces are at  $I^{mt} = 0.85$  and  $I^{vm} = 1.85$  respectively. The first two pairs of figures shown in Figures 3.4(a), 3.4(b) and 3.4(c), 3.4(d) respectively correspond to numerical solution of the VMT problem for the above two examples. The third figure pair, as shown in Figures 3.4(e), 3.4(f), was obtained using  $D^v = 10^{-7}$ ,  $M^v = 5 \cdot 10^{-6}$ —a parameter set associated with realistic models of capillaries [42].

Using a computer with a 2.66 GHz Intel Core 2 Duo processor and 4GB of memory, a steady state solution with  $10^{-5}$  absolute error at  $\boldsymbol{\rho}$  for the parameter choices  $D^v = 10^{-4}$ ,  $M^v = 10^{-3}$  was obtained in under five minutes using 3 MB; the steady state solution with parameter choices  $D^v = 10^{-5}$ ,  $M^v = 10^{-4}$  (see Figures 3.4(a) and 3.4(c)), in turn, was reached in under 14 minutes using only 15 MB of memory. Figures 3.4(e) and 3.4(f) demonstrate the capabilities of the VMT solvers for the parameter set  $D^v = 10^{-7}$ ,  $M^v = 5 \cdot 10^{-6}$  mentioned above. Note especially the extremely steep boundary layer present at the vessel-membrane interface (see Figure 3.4(f)) that was accurately captured in the solution. While very far from feasible for other methods, our VMT

(a)			(b)		
FC-VMT Solver			FD-VMT Solver		
$N_x \times N_y$	Rel. $L_2$ Error	Comp. time	$N_x \times N_y$	Rel. $L_2$ Error	Comp. time
$100 \times 100$	$5.0 \cdot 10^{-3}$	238 secs	$100 \times 100$	$4.7 \cdot 10^{-3}$	105 secs
$200 \times 200$	$5.6 \cdot 10^{-4}$	360 secs	$200 \times 200$	$1.1 \cdot 10^{-3}$	366 secs
$400 \times 400$	$1.03 \cdot 10^{-4}$	881 secs	$400 \times 400$	$2.0 \cdot 10^{-4}$	1482 secs
$800 \times 800$	$5.4 \cdot 10^{-5}$	3398 secs	$800 \times 800$	$1.2 \cdot 10^{-4*}$	25,802 secs

(c)		
Graded FD-VMT Solver		
$N_x \times N_y$	Rel. $L_2$ Error	Comp. time
$100 \times 100$	$2.1 \cdot 10^{-2}$	196 secs
$200 \times 200$	$9.3 \cdot 10^{-3}$	605 secs
$400 \times 400$	$2.7 \cdot 10^{-3}$	1841 secs
$800 \times 800$	$5.0 \cdot 10^{-4}$	6666 secs

Table 3.2: Computational times required to obtain various accuracies for the parameter set  $D^v = 10^{-5}$ ,  $M^v = 10^{-4}$  through use of each of the three VMT solvers. The solutions were computed over grids of size  $N_x \times N_y$  with the parameter choices  $\Lambda = 25$ ,  $\sigma_f = 10^{-5}$ , and  $\sigma_s = 10^{-5}$ . The entry marked by \* was obtained using the selection  $\sigma_f = 10^{-6}$ .

solvers (specifically the FC-VMT solver) produced steady state solutions with the required  $10^{-5}$  accuracy at  $\rho$  for the third parameter set using 14.8 MB of memory in a 6.5 hour run on the same 2.66 GHz processor.

For the first and second parameter sets,  $D^v = 10^{-4}$ ,  $M^v = 10^{-3}$  and  $D^v = 10^{-5}$ ,  $M^v = 10^{-4}$ , we chose iteration parameter  $\Lambda = 25$ . This smaller value of  $\Lambda$  was selected to enable rapid computation of the steady state solution; because the boundary layer in both of these cases is not extremely steep, the time steps may be rapidly increased while ensuring an accurate solution. Accurate resolution of the steep boundary layer occurring in the solution for the third parameter set,  $D^v = 10^{-7}$ ,  $M^v = 5 \cdot 10^{-6}$  (see Figure 3.4(f)), requires a larger value of  $\Lambda$ ; for all the numerical simulations of this parameter set, we chose  $\Lambda = 100$ .

Table 3.2 demonstrates convergence through examination of the relative  $L_2$  norm as a function of grid size for the FC-, FD-, and Graded FD-VMT solvers for the parameter set  $D^v = 10^{-5}$ ,  $M^v = 10^{-4}$ , while Table 3.3 displays the absolute errors at the point  $\rho$  for a  $400 \times 400$  grid. We note that the  $10^{-5}$  accuracy we seek at the critical midpoint  $\rho$  is obtainable using only a  $400 \times 400$  grid with the FC-VMT solver.

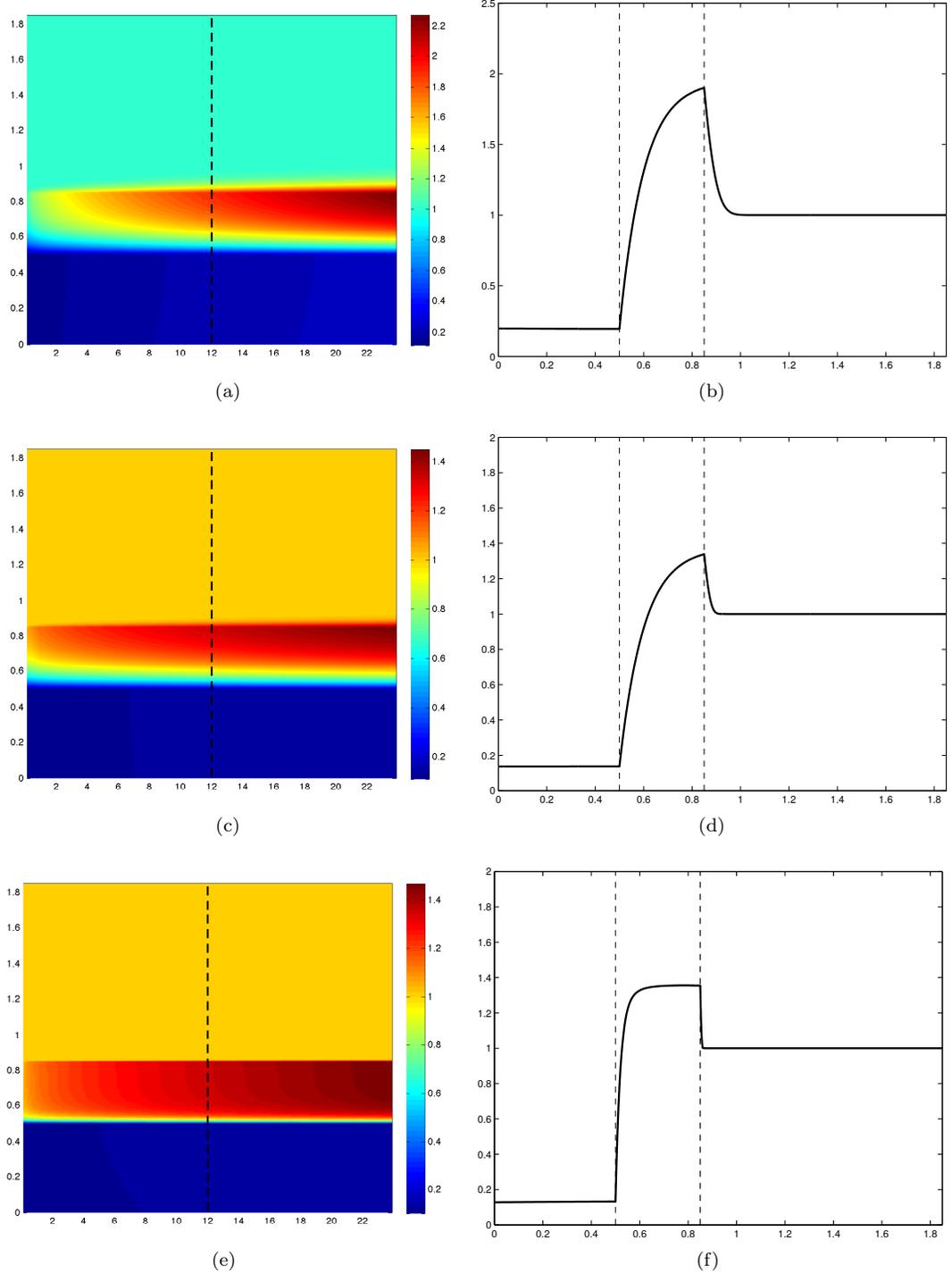


Figure 3.4: Solution of the steady state concentration  $C(x, y)$  obtained through use of the FC-VMT solver for a VMT model with  $I^{vm} = 0.85$  and  $I^{mt} = 0.5$  for three parameter sets. Panels (a), (c), and (e) display the steady state solution over the full domain while Figures (b), (d), and (f) display a cross section of the steady state solution at the midpoint  $x = 12$  (corresponding to the dashed line in Figures (a), (c) and (e)). The dashed lines in Figures (b), (d) and (f) represent the vessel-membrane and membrane-tissue interfaces,  $I^{vm}$  and  $I^{mt}$  respectively. Panels (a) and (b):  $D^v = 10^{-4}$ ,  $M^v = 10^{-3}$ ,  $\mathcal{R} = 10^{-2}$ ,  $\mathcal{R}^t = 0.1$ . Panels (c) and (d):  $D^v = 10^{-5}$ ,  $M^v = 10^{-4}$ ,  $\mathcal{R} = 10^{-2}$ ,  $\mathcal{R}^t = 0.1$ . Panels (e) and (f):  $D^v = 10^{-7}$ ,  $M^v = 5 \cdot 10^{-6}$ ,  $\mathcal{R} = 10^{-2}$ ,  $\mathcal{R}^t = 0.1$ . Note, e.g., the sharp boundary layers to the right of the point  $y = 0.85$  in Figures (b), (d), and (f) that have been resolved by the FC-VMT solver, and that would require use of extremely small time steps if an explicit method were used.

	Abs. Error at $\rho$	Comp. Time
FC-VMT Solver	$5.5 \cdot 10^{-5}$	881 secs
FD-VMT Solver	$2.0 \cdot 10^{-4}$	1482 secs
Graded FD-VMT Solver	$4.3 \cdot 10^{-3}$	1841 secs

Table 3.3: Computational times and errors at the critical midpoint  $\rho = ((x_r - x_l)/2, I^{vm})$  using  $D^v = 10^{-5}$ ,  $M^v = 10^{-4}$  for the three VMT solvers. The solutions were computed over a  $400 \times 400$  grid with the parameter choices  $\Lambda = 25$ ,  $\sigma_f = 10^{-5}$ , and  $\sigma_s = 10^{-5}$ .

## Chapter 4

# Magnetically Enhanced Diffusion

### 4.1 Introduction

As discussed in Chapter 3, the goal of magnetic drug delivery is to use magnetic fields to direct and confine magnetically responsive particles to specific regions in a patient's body, thus allowing for focused treatment in an area of interest. However, magnetic drug delivery inherently suffers from two severe limitations: 1) the inability to focus treatment on targets located deep inside the body with a stationary magnetic field (when stationary external magnets are used, particles can only be moved at a maximum depth of 5 cm) and 2) the fundamental consequence of the classic Samuel Earnshaw theorem [18] that no inverse-square law force (including magnetic force on a single particle) can create a stable equilibrium in the interior of a domain—only unstable equilibria for ferrofluid particles may be attained with a static magnetic field. To overcome these limitations, Shapiro [51] proposed the development of a *dynamic* feedback-control scheme, where manipulation of the magnetically responsive particles is sought through dynamic adjustment of the magnetic fields.

To design such a feedback-control scheme, a predictive capability must be used to evaluate the effects of external magnetic forces on the convection and diffusion of magnetically responsive particles through the relevant tissues. As previously discussed in Chapter 1, a simplified, but effective model describing these effects has been proposed by Grief and Richardson [26]. The characterizing equation of the Grief and Richardson model is the hyperbolic convection-diffusion PDE

$$\frac{\partial}{\partial t} C(\vec{r}, t) = -\nabla \cdot \left[ C(\vec{r}, t) \vec{V}_{\text{blood}}(\vec{r}, t) - D(\vec{r}) \nabla C(\vec{r}, t) + k(\vec{r}) C(\vec{r}, t) \nabla \left( |\vec{H}(\vec{r}, t)|^2 \right) \right],$$

where  $C(\vec{r}, t)$  is the concentration of magnetic particles in the blood,  $\vec{V}_{\text{blood}}(\vec{r}, t)$  is the blood velocity,  $D(\vec{r})$  is the diffusion coefficient of particles within the bloodstream,  $k(\vec{r})$  is the magnetic drift coefficient, and  $\vec{H}(\vec{r}, t)$  represents the externally applied magnetic field. The control of the particles is effected through the term  $\nabla \left( |\vec{H}(\vec{r}, t)|^2 \right)$ .

A test of the feasibility of such a dynamic ad hoc control scheme is presented in [51]. In refer-

ence [51], an idealized control setup is created by making several simplifications to the model: the domain of solution is taken to be the unit circle, the diffusion coefficient,  $D$ , and magnetic drift coefficient,  $k(\vec{r})$ , are assumed to be constant, and the effect of blood velocity,  $\vec{V}_{\text{blood}}(\vec{r}, t)$  is removed. To ensure no fluid leaves the domain, a zero-flux condition is imposed on the boundary. Initially, the fluid concentration is set to 1 throughout the domain. Applying these changes to the Grief and Richardson model yields the control setup:

$$\begin{aligned} C_t &= D\nabla^2 C + M^x(x, y, t)C_x + M^y(x, y, t)C_y + M^u(x, y, t), & (x, y, t) \in \Omega \times (0, T], \\ \frac{\partial C}{\partial \mathbf{n}} &= 0, & (x, y) \in \partial\Omega, \\ C(x, y, 0) &= 1, & (x, y) \in \Omega, \end{aligned} \quad (4.1)$$

where

$$\begin{aligned} (M^x(x, y), M^y(x, y)) &= -k\nabla \left( |\vec{H}(x, y, t)|^2 \right), \text{ and} \\ M^u(x, y) &= -k\nabla^2 \left( |\vec{H}(x, y, t)|^2 \right). \end{aligned}$$

Design of adequate control schemes requires numerical solution of such PDEs. For the parameter values inherent in the medical configurations under consideration, the numerical PDE problems above have proven quite challenging [51]. In this chapter, we utilize the model (4.1) as a test bed for development of numerical algorithms capable of evaluating such numerical solutions accurately and efficiently.

Numerical solution of the control setup (4.1) has proven highly challenging to numerical solvers for several reasons, most prominent of which is the steep build-up of concentration, or *boundary layer*, that occurs near the boundary of the domain. For the small diffusion coefficients typically required to portray a realistic control setup, the imposition of the zero-flux boundary condition combined with the strong convective forces generated by the external magnetic field will cause extremely sharp boundary layers to rapidly appear near the boundary of the domain. Numerical resolution of such boundary layers depends on the use of a very fine spatial step size, thus placing limitations on the type of numerical method efficiently useable; for example, due to the requirement of such a fine mesh, explicit methods are rendered highly inefficient by the restrictive CFL condition imposed on the time step.

Further, because numerical solution over a nonrectangular domain is required for the idealized control setup described above, a standard finite differences approach over a Cartesian grid is not an efficient method of solution: such a scheme would require either a “staircasing” of the boundary of the domain or challenging, essentially impractical domain mapping strategies. Use of the first, rather simple technique reduces the spatial accuracy of the resulting finite difference method to first order. In addition, due to the absence of solution values outside the computational domain, finite difference stencils are forced to be made increasingly one-sided as the domain boundaries are approached. In

general, stability is not achieved by simply using high-order centered difference methods in the interior of the domain and equally high-order biased stencils near the boundary. While there are several techniques to resolve this problem (such as the use of compact schemes [4] or Summation By Parts operators [40, 48]), these approaches are computationally expensive and must sacrifice some accuracy near the boundary to gain stability. Importantly, further, the multidomain strategies associated with these algorithms require the discretizations of neighboring domains to match perfectly at common boundaries; see, e.g., [2] for details.

Numerical solutions of PDEs over nonrectangular domains are frequently obtained through use of finite element and finite volume solvers. However, because the spectral radii of differentiation operators based on nonuniformly spaced structured grids, such as those used by finite element and finite volume methods will, in general, grow superlinearly, these techniques must satisfy a stringent CFL condition for stability. High-order methods based on unstructured meshes give rise to similarly restrictive CFL constraints.

In this chapter, we present two numerical solvers which, through employment of a combination of explicit and implicit Fourier Continuation (FC)-based pseudospectral solvers in appropriately partitioned subdomains, overcome these problems and provide efficient and accurate numerical solutions of the control setup (4.1). These solvers, which we refer to as the Explicit-Implicit and Implicit-Implicit FC solvers, may in turn be used as effective numerical tools for the design of control schemes.

By dividing the domain of solution  $\Omega$  into smaller regions each selected to capture a particular behavior of the concentration, numerical algorithms may be specifically tailored in each subdomain to provide a much more efficient method of solution. In both the Explicit-Implicit and Implicit-Implicit FC solvers, adequate numerical resolution of the steep boundary layers is achieved through partitioning a finely discretized thin annular region toward the boundary; to overcome the extremely restrictive CFL condition such a grid would place on an explicit method, the Explicit-Implicit and Implicit-Implicit FC solvers makes use of an efficient implicit approach, the FC-AD solver [16, 15]. Because the concentration varies little away from the boundary layer, for sufficiently small values of the diffusion coefficient  $D$  an efficient method of solution is provided in the remainder of the domain through an explicit FC-based solver; the resulting combination of explicit and implicit FC methodologies gives rise to the Explicit-Implicit FC solver. However, because the restriction on the time step imposed by the CFL condition is dependent on choice of the diffusion coefficient  $D$ , larger values of  $D$  cause the CFL condition to once again become overwhelmingly restrictive, thus rendering use of an explicit method in any subdomain of  $\Omega$  inefficient; the Implicit-Implicit FC solver overcomes this restriction by making use of the FC-AD method in the full domain  $\Omega$ .

Control of the ferrofluid is performed through adjustment of eight external electromagnets equally situated around ring surrounding the domain of solution, see Figure 4.1; the control term  $\vec{H}(\vec{x}, t)$  is

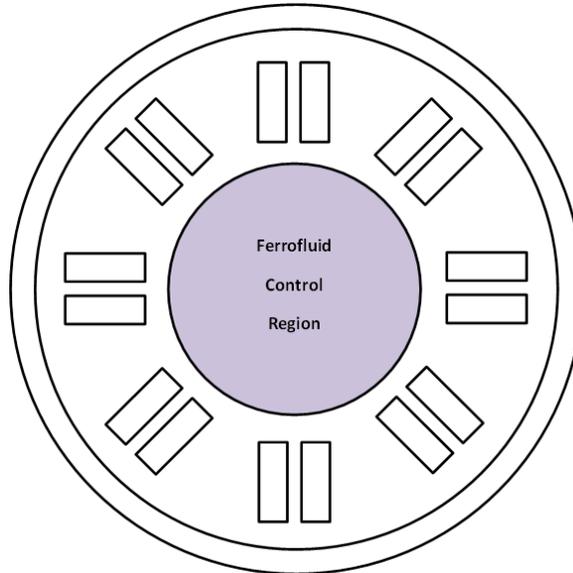


Figure 4.1: Geometry corresponding to the control setup. The domain of solution of the PDE (4.1), denoted by  $\Omega$ , is the innermost circle labeled “Ferrofluid Control Region”. The eight surrounding electromagnets are each represented by two rectangular regions.

simply computed as the magnetic field generated by this ring of magnets. For preliminary analyses of this idealized control setup, we simply seek to direct and confine the particles to near the center of the domain. As detailed in [51], focus of the particles to the near the center of the domain is sought through dynamic adjustment of the magnetic fields as follows: given an initial choice of currents in the electromagnets, a simulation of the solution of the corresponding PDE is run until a particular specified time; upon reaching this time, the currents in the electromagnets are adjusted (either from examination, as done in [51], or by a prescribed control algorithm), and simulation of the newly adjusted PDE is run until another specified time. This last step is then repeated until adequate focus of the particles is obtained. We refer to such a collection of currents and times as a *control scheme*.

In the analyses performed in [51], numerical solutions of the PDE (4.1) for varying magnetic fields were obtained through use of the finite-element-based commercial software COMSOL Multiphysics. While capable of numerically solving (4.1) for weak magnetic fields, the COMSOL software encountered many difficulties for the stronger magnetic fields required to direct the particles near the center of the domain, including an inability to resolve the extremely sharp particle concentration build-up occurring near the boundary in the presence of strong magnetic forces. Further, the computational times required by the COMSOL software to obtain a solution impose additional difficulties for the development of a more precise control scheme. For example, the COMSOL software obtained a solution of the control setup described in [51] in a run time of several days with 32 GB of memory. In contrast, the Explicit-Implicit FC solver presented in this thesis obtains a solution for a similar

control setup, in which extremely sharp boundary layers of height  $10^4$  occur, in a run time of under two hours using 50 MB of memory.

The remainder of this chapter is organized as follows. In Section 4.2, we provide a prescription of the division of the full domain  $\Omega$  into appropriately selected subdomains allowing for efficient resolution of the boundary layer. In Section 4.3, we describe in detail our implementation of the Explicit-Implicit and Implicit-Implicit FC solvers. Finally, in Section 4.4 we present a variety of numerical experiments, including solution for several different control setups, and compare the performance of the Explicit-Implicit and Implicit-Implicit FC solvers to that achieved by COMSOL for the simulations presented in [51].

## 4.2 Grids

As discussed in Section 4.1, when the magnetic forces acting on the particles are strong enough to overcome the effects of diffusion, a sharp build-up of concentration, or boundary layer, appears near the boundary of  $\Omega$ . In order to efficiently resolve this boundary layer numerically, we split  $\Omega$  into three subdomains: a square domain  $S$  near the origin, a wide annular domain  $R_1$  extending from near the center of  $\Omega$  to near the boundary  $\partial\Omega$ , and a thin annular domain  $R_2$  extending from the outer radius of  $R_1$  to the boundary  $\partial\Omega$ . Accurate numerical resolution of the boundary layer is achieved through discretization of the thin annulus  $R_2$  with a very fine polar mesh. The wide annulus  $R_1$  is chosen for computational efficiency: because the concentration of particles closer to the center of the domain is near-constant, adequately accurate numerical solutions can be obtained with a much coarser mesh. Finally, the coarsely discretized square domain  $S$  is implemented near the origin to avoid any singularities use of a polar mesh at  $r = 0$  may generate. As illustrated in Figure 4.2, the Cartesian grid

$$S = \{(x, y) \mid x \in [-a, a], y \in [-a, a]\}$$

is used to discretize  $S$ , while the polar grids

$$R_1 = \{(r, \theta) \mid r \in [b, 1 - \epsilon], \theta \in [0, 2\pi)\}$$
 and

$$R_2 = \{(r, \theta) \mid r \in [1 - \epsilon, 1], \theta \in [0, 2\pi)\}$$

are used to discretize  $R_1$  and  $R_2$  respectively. Note that the square and wide annular domains,  $S$  and  $R_1$ , are chosen to overlap with one another, while the thin annulus  $R_2$  has inner radius equal to the outer radius of the wide annulus  $R_1$ —there is no overlap between the annuli. The grid boundary parameters  $a$ ,  $b$ , and  $\epsilon$  as must be carefully chosen to enable accurate numerical resolution of the boundary layer present at the edge of the domain; appropriate selection of these parameters

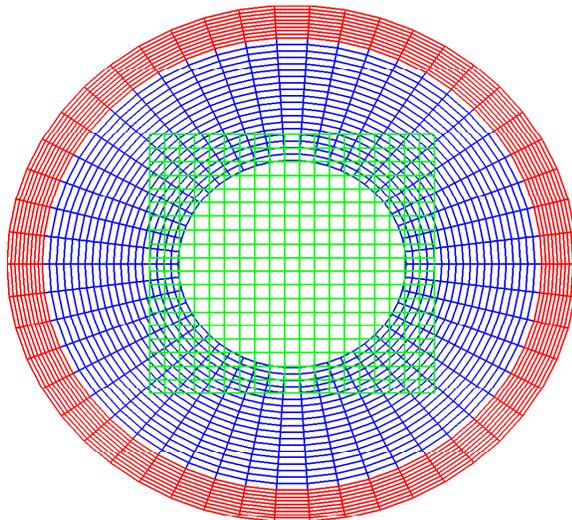


Figure 4.2: Grids used by the Explicit-Implicit FC solver. The Cartesian grid  $S$  is represented by the green grid lines, while the polar grids  $R_1$  and  $R_2$  are represented by the blue and red grid lines respectively. Notice that the Cartesian grid  $S$  and the polar grid  $R_1$  overlap, while the polar grids  $R_1$  and  $R_2$  are chosen to align along their outer and inner radii respectively.

is discussed in detail in Section 4.4.

Because polar grids are used in the annuli  $R_1$  and  $R_2$ , a change of variables needs to be applied to the PDE (4.1). Applying a transformation to polar coordinates to (4.1) yields the modified PDE

$$\begin{aligned}
 DC_{rr} + \frac{D}{r^2}C_{\theta\theta} + \left(\frac{D}{r} + M^x(x, y)\cos\theta + M^y(x, y)\sin\theta\right)C_r \\
 + \left(M^y(x, y)\frac{\cos\theta}{r} - M^x(x, y)\frac{\sin\theta}{r}\right)C_\theta + M^u(x, y)C = C_t, \tag{4.2} \\
 C_r(1, \theta, t) = 0, \quad \theta \in [0, 2\pi).
 \end{aligned}$$

It is useful to note that this new PDE is of the same structure as the original PDE (4.1); only the coefficient functions have changed. Thus, because the solution methods incorporated in the Explicit-Implicit and Implicit-Implicit FC solvers do not depend on prior knowledge of the coefficient functions, the same techniques may be used in all three domains with only minor modifications. Recalling that the explicit and implicit FC methodologies discussed in Sections 2.2 and 2.3 respectively were presented for general coordinates  $(u, v)$ , Table 4.1 provides the relevant details required for numerical solution of the PDEs (4.1) and (4.2) in the corresponding coordinate systems.

	Cartesian Mesh	Polar Mesh
$(u, v)$	$(x, y)$	$(r, \theta)$
$\vec{\kappa}(u, v)$	$(D, D)$	$(D, \frac{D}{r^2})$
$\vec{\lambda}(u, v)$	$(M^x, M^y)$	$(\frac{D}{r} + M^x \cos \theta + M^y \sin \theta, M^y (\frac{\cos \theta}{r}) - M^x (\frac{\sin \theta}{r}))$
$\nu(u, v)$	$M^u$	$M^u$

Table 4.1: Table of coefficients and coordinates used over each of the two grid types for both the Explicit-Implicit and Implicit-Implicit FC-Solvers. The far-left column provides the corresponding notation used in Section 2.3 for the description of the ADI algorithm.

### 4.3 Dynamic Control Solver

Returning to the outline of the general FC-AD methodology presented in Chapter 2, we note that a full description of the application of the FC-AD method to the dynamic control problem (4.2) requires prescription of the boundary conditions. Because of the multiple grids used to discretize the full domain of solution, obtaining these boundary conditions is a nontrivial feat requiring the implementation of several vital details. A description of these details is provided in Sections 4.3.1 through 4.3.4 below.

#### 4.3.1 Periodic Boundary Conditions

While the FC-AD method was presented for general coordinates in Chapter 2, we recall that the Explicit-Implicit and Implicit-Implicit FC methods are based on solving the PDE (4.2) in annular domains. Making the change of notation  $(u, v) \rightarrow (r, \theta)$ , we note that the boundary condition in  $\theta$  is inherently given by periodicity. That is, we require  $C^n(r, 0) = C^n(r, 2\pi)$ . Subsequently, the boundary conditions required for inversion of the operator (2.10) corresponding to  $A_\theta$  are also given by periodicity.

It is also useful to note that the FC-ODE method described in Section 2.4 used to invert the operator  $(1 - \frac{\Delta t}{2} A_\theta)$  may be reduced to a spectral solver. Indeed, because the coefficient functions of the ODE associated with the operator  $(1 - \frac{\Delta t}{2} A_\theta)$  and the solution  $\tilde{C}$  are periodic in  $\theta$ , the Fourier continuations inherent in the FC-ODE solver may be replaced by their corresponding standard Fourier series.

### 4.3.2 Radial Boundary Conditions

From equation (4.2), it is clear that the right boundary condition for  $\tilde{C}^{n+1}$  in the outer annulus  $R_2$  is simply given by the homogeneous Neumann boundary condition

$$\frac{\partial \tilde{C}^{n+1}}{\partial r}(r, \theta) = 0, \quad (u, v) \in \partial_r R_2,$$

where  $\partial_r R_2 = \{(1, \theta) \mid \theta \in [0, 2\pi)\}$  is the right boundary of  $R_2$ .

Upon examination, we see that the condition given by equation (4.2) provides only one boundary condition for  $C^{n+1}$  (and subsequently  $\tilde{C}^{n+1}$ ) over the outer boundary of  $R_2$ ; clearly a boundary condition over the inner radius of  $R_2$  is required to solve for  $\tilde{C}^{n+1}$ . Recalling that the inner and outer radii of  $R_2$  and  $R_1$  respectively are selected to match one another, a natural choice would be to take this missing boundary condition to be a Dirichlet boundary condition with value given by the solution over the coarse annulus,  $R_1$ : once a solution over the  $R_1$  is obtained (possibly via the explicit method as described in Section 2.2), the boundary values at the inner radius of  $R_1$  could simply be taken as the solution values over the outer radius of  $R_2$ .

However, choice of differing spatial step sizes and solvers used over  $R_1$  and  $R_2$  will typically yield a small discrepancy between the numerical solutions over each domain. Upon viewing the solution over the full domain, this discrepancy will present itself as a “seam” near the boundary of two neighboring grids. By obtaining the left boundary condition for  $R_1$  directly from the solution over  $R_2$ , the discrepancy present at the seam will propagate numerical error deeper into each domain with repeated iterations of the FC-AD algorithm.

While it is possible to resolve this “seaming” issue by overlapping the two domains  $R_1$  and  $R_2$  and overwriting (using an interpolation scheme such as, e.g., the one detailed in Section 4.3.5) the less-accurate solution  $R_1$  with the more-accurate solution from  $R_2$ , this is not an efficient method of solution in the presence of the steep boundary layers occurring near the boundary of  $\Omega$ . Indeed, the overlap between the two grids needed to remove the effects of seam requires significantly increasing the width of  $R_2$ , thus, in turn, requiring many more grid points in  $R_2$  to enable accurate resolution of the boundary layer. In practice, this necessary increase in the number of grid points will cause the implicit solver in  $R_2$  and thus the overall solver to become extremely computationally expensive.

An efficient resolution of the detrimental effects of seaming is provided through numerical solution over an auxiliary grid located over the boundary between  $R_1$  and  $R_2$ . In detail, we define the auxiliary annulus

$$A = \{(r, \theta) \mid r \in [1 - \delta, 1 + v], \theta \in [0, 2\pi)\},$$

where  $\delta > \epsilon$  and  $v < \epsilon$  are chosen to allow for sufficient overlap with  $R_1$  and  $R_2$ ; appropriate choice of  $\delta$  and  $v$  is provided in Section 4.4. We proceed by explicitly solving in  $R_1$  and  $R_2$  using the FC-based

time-marching method described in Section 2.2. By interpolating onto the left and right boundaries of  $A$ , the explicit solutions in  $R_1$  and  $R_2$  respectively provide approximate boundary values for  $A$ . Once given both approximate boundary values, a numerical solution in  $A$  is obtained via the FC-AD method with Dirichlet boundary conditions. The left boundary condition for  $R_2$  is then given as a Dirichlet boundary condition with value obtained by interpolation of the solution in  $A$  onto the left boundary of  $R_2$ . Similarly, if an implicit method is to be used, the right boundary condition for  $R_1$  may be given as a Dirichlet boundary condition with value obtained by interpolation of the solution in  $A$  onto the right boundary of  $R_1$ .

Using the notation in Section 2.3, the boundary conditions in  $R_2$  are thus given by the selections

$$\begin{aligned} a_l = 1, \quad b_l = 0, \quad B_l = \tilde{C}_A^{n+1}|_{\partial_l R_2}, \quad \text{and} \\ a_r = 0, \quad b_r = 1, \quad B_r = 0, \end{aligned}$$

where  $\partial_l R_2$  is the left boundary of  $R_2$  and  $\tilde{C}_A^{n+1}$  is the approximate solution at time  $t^{n+1}$  obtained from applying the FC-AD method over  $A$ .

**Remark 4.3.1** From Section 4.3.1 and the above prescription, we note that the Dirichlet left boundary condition in  $r$  is only easily given at the full next step, that is for the computation of  $\tilde{C}^{n+1}$ , while the periodicity boundary conditions in  $\theta$  hold for both  $\tilde{C}^{n+1}$  and the half-step  $\tilde{C}^{n+\frac{1}{2}}$ . Thus, for ease of implementation, we rearrange the scheme (2.9) as

$$\begin{aligned} \left(1 - \frac{\Delta t}{2} A_\theta\right) \tilde{C}^{n+\frac{1}{2}} &= \left(1 + \frac{\Delta t}{2} A_r\right) \tilde{C}^n, \\ \left(1 - \frac{\Delta t}{2} A_r\right) \tilde{C}^{n+1} &= \left(1 + \frac{\Delta t}{2} A_\theta\right) \tilde{C}^{n+\frac{1}{2}}, \end{aligned}$$

where, for the appropriate coefficient functions  $\vec{\kappa}$ ,  $\vec{\lambda}$  and  $\nu$ ,  $A_r$  and  $A_\theta$  correspond to the previously defined  $A_v$  and  $A_u$ , respectively.

### 4.3.3 Explicit-Implicit Solver

The Explicit-Implicit FC solver arises from implementation of the FC-AD method in the thin annulus  $R_2$ , as described above, in conjunction with use of an explicit solver in the coarsely meshed regions  $S$  and  $R_1$ . Such an explicit solver is provided by explicit FC solver described in Section 2.2, with appropriately selected coordinates and coefficient functions (as detailed in Table 4.2).

Several different time-stepping methods were considered for the solution of  $C^n$ , including the forward Euler, second-order Runge-Kutta, and second-order Adams-Bashforth schemes. In comparing the computational speed versus the accuracy of the overall solution in  $\Omega$  for each of these methods, we determined the second-order Runge-Kutta scheme described in Section 2.2 was most efficient for

our numerical simulations. This is, in part, due to the fact that any of the above explicit methods used in the right boundary domains  $S$  and  $R_1$  will be inherently significantly faster than the implicit method used to resolve the boundary layer in the finely meshed annulus  $R_2$ . Further, because the implicit method used in  $R_2$  has second order accuracy in time—thus limiting the maximum accuracy of the overall solution—higher order explicit schemes in  $S$  and  $R_1$  were not considered.

It is important to note that this explicit time-marching scheme is efficiently applicable only for when the diffusion coefficient  $D$  is sufficiently small and the domains  $S$  and  $R_1$  are coarsely discretized: since a large spatial step used, the well-known diffusive CFL condition,  $\Delta t \sim \Delta x^2/D$ , for stability is not overly stringent. Use of an explicit method is not computationally efficient in the domain  $R_2$ , however: the thin width and high grid resolution of  $R_2$  required to resolve the boundary layer force the CFL condition to become overwhelmingly restrictive. To overcome this difficulty, the Explicit-Implicit FC-solver makes use of the implicit FC-AD method in the thin annulus  $R_2$ .

#### 4.3.4 Implicit-Implicit Solver

As mentioned in Sections 4.1 and 4.3.3, the explicit FC-based solver provides an efficient method of solution for sufficiently small values of diffusion coefficient  $D$ . However, when the diffusion coefficient  $D$  is sufficiently large, the CFL condition imposed by an explicit method places a stringent restriction on the maximum value of the time step  $\Delta t$ , even in the coarsely discretized domains  $S$  and  $R_1$ . This condition is overcome by replacing the explicit FC solver with the implicit FC-AD method in both  $S$  and  $R_1$ ; the resulting fully implicit solver is referred to as the Implicit-Implicit FC solver.

Implementation of the FC-AD method in  $S$  and  $R_1$  requires prescription of the left boundary condition in  $R_1$  and both boundary conditions in  $S$ . Analogously to the appropriate selection of boundary conditions at the interface between  $R_1$  and  $R_2$ , the relevant boundary conditions for  $\tilde{C}^{n+1}$  may be obtained without the adverse effects of seaming through inclusion of another auxiliary annulus

$$A_2 = \{(r, \theta) \mid r \in [b - \delta_2, b - v_2], \theta \in [0, 2\pi)\},$$

located over the boundary between  $S$  and  $R_1$ . Selection of  $\delta_2$  and  $v_2$  is provided in Section 4.4.

Because a Cartesian mesh is used to discretize the square domain  $S$ , implementation of the FC-AD method in  $S$  requires boundary conditions to be provided for the half-step solution  $\tilde{C}^{n+\frac{1}{2}}$ . These half-step boundary conditions are obtained by applying the FC-AD method with time step  $\Delta t/2$  in  $A_2$ , where the boundary conditions for solution of  $\tilde{C}^{n+\frac{1}{2}}$  over  $A_2$  are, in turn, provided by application of the explicit FC solver over  $S$  and  $R_1$  with time step  $\Delta t/2$ .

**Remark 4.3.2** It is important to note that, for reasonable values of the magnetic advection, larger values of the diffusion coefficient  $D$  give rise to less steep boundary layers. Thus, in the interest of computational efficiency, the fine annulus  $R_2$  and, in turn, the auxiliary annulus  $A_1$ , specifically

implemented to resolve steep boundary layers are not necessary for the Implicit-Implicit FC solver when  $D$  is sufficiently large. As this is the case for the numerical results presented in Section 4.4, our implementations of the Implicit-Implicit FC solver make use of only the grids  $S$ ,  $R_1$ , and  $A_2$ .

### 4.3.5 Interpolation

Due to the variations in spatial resolution and coordinates between each of the multiple domains, the discrete grids used in the dynamic control solver will not, in general, have perfectly overlapping grid points. To enable the proper communication between grids required for both obtaining the necessary boundary conditions and appropriate overwriting of the solutions, we make use of the following 2D interpolation scheme.

Considering a set of discrete function values  $f(u_i, v_j)$  given over an  $N \times N$  grid  $\{(u_i, v_j) | i, j = 1, \dots, N\}$ , we seek to find an approximation to the function value at the point  $(u_0, v_0)$ . For each grid point  $v_j$ , we proceed by interpolating over the grid points  $(u_i, v_j)$ ,  $i = 1, 2, \dots, N$  to obtain an approximate value of  $f(u_0, v_j)$ . A suitable approximation to  $f(u_0, v_0)$  is then obtained by interpolating over the set  $(u_0, v_j)$  for  $j = 1, \dots, N$ .

All that remains is selection of a one-dimensional interpolation algorithm. For the numerical simulations presented in this chapter, we implemented the one-dimensional Neville's interpolation algorithm typically used in solution of PDEs over multiple overlapping domains. The algorithm is as follows: given a set of  $N$  data points  $(u_i, f_i)$ , the interpolating polynomial  $p$  is obtained through the recursion scheme

$$\begin{aligned} p_{i,i}(u) &= f_i, & 1 \leq i \leq N, \\ p_{i,j}(u) &= \frac{(u - u_j)p_{i,j-1}(u) + (u_i - u)p_{i+1,j}(u)}{u_i - u_j}, & 1 \leq i < j \leq N, \end{aligned}$$

where  $p_{i,j}$  denotes the polynomial of degree  $j - i$  such that  $p(u_k) = f_k$  for  $k = i, \dots, j$ . The polynomial interpolating  $(u_i, f_i)$  for  $i = 1, \dots, N$  is given by  $p_{1,N}(u)$ .

## 4.4 Numerical Results

We present a variety of numerical tests produced by both the Explicit-Implicit and Implicit-Implicit FC solvers for the ferrofluid model (4.1). In Section 4.4.1, we present simulations, obtained from a C++ implementation of the Explicit-Implicit FC solver, of the ad hoc control sequence introduced in [51] for varying diffusion coefficients.

#### 4.4.1 Ad Hoc Control Examples

In this section, we present numerical examples that follow the setup and ad hoc control scheme proposed in reference [51]. We consider the domain of solution to be a circular region  $\Omega$  of radius 1. The externally placed electromagnets are represented by two rectangular domains of opposing current (see Figure 4.1) and taken to have (dimensionless) length 4, inner radius .025 and outer radius .325. The eight electromagnets are placed in an equispaced manner around a ring surrounding  $\Omega$ ; the distance between the center of each electromagnet and the origin was taken to be 1.5. A vertical current of strength  $J$  is applied in the upper half of each magnet. Similarly, an opposing vertical current of strength  $-J$  is applied in the lower half of each magnet. For ease of explanation, the magnets are labeled in a counterclockwise manner starting from the magnet centered at  $(1.5, 0)$ . A detailed discussion of the numerical computation of the relevant magnetic fields is provided in Appendix A. Following the ad hoc control scheme described in [51], we use the following set of currents and times:

$$\begin{aligned}
 \vec{J} &= \{0, 0, .5, 0, 0, 0, -.5, 0\}, & t = 0 \text{ to } t = 3.9, \\
 \vec{J} &= \{.5, .15, 0, -.15, -.5, -.15, 0, .15\}, & t = 3.9 \text{ to } t = 5.4, \\
 \vec{J} &= \{.5, .2, 0, -.2, -.5, -.2, 0, .2\}, & t = 5.4 \text{ to } t = 9.0 \\
 \vec{J} &= \{.5, .25, 0, -.25, -.5, -.25, 0, .25\}, & t = 9.0 \text{ to } t = 11.4, \\
 \vec{J} &= \{.5, 0, 0, 0, -.5, 0, 0, 0\}, & t = 11.4 \text{ to } t = 14.1 \\
 \vec{J} &= \{0, .15, .5, .15, 0, -.15, -.5, -.15\}, & t = 14.1 \text{ to } t = 22.5, \\
 \vec{J} &= \{0, .25, .5, .25, 0, -.25, -.5, -.25\}, & t = 22.5 \text{ to } t = 30.0,
 \end{aligned} \tag{4.3}$$

where  $\vec{J} = (J_1, J_2, J_3, \dots, J_8)$  is a vector containing the currents for each of the eight surrounding electromagnets.

As mentioned in Section 4.1, preliminary numerical tests leading to the contribution [51] were conducted using the commercial software package COMSOL Multiphysics. Although capable of providing adequately accurate numerical solutions of the ad hoc control scheme presented in [51], the COMSOL software was incapable of resolving the steep boundary layer occurring for smaller values of  $D$ . For example, numerical solution for the choice of diffusion coefficient  $D = 0.001$  and magnetic drift  $k = 1$  using the previously mentioned 3.16 GHz single processor of a quad-core Intel Xeon CPR X5460 computer with 32 GB of memory was not feasible with COMSOL: to do so would require memory allocation greater than the computer's capacity.

For the choices  $D = 0.1$ ,  $k = 1$ , numerical solution of the control scheme (4.3) was achieved with maximum relative error  $10^{-3}$  in 8 minutes with 18 MB of memory using a C++ implementation of

the Implicit-Implicit FC solver on the previously mentioned 2.66 GHz Intel Core 2 Duo processor. Numerical solutions of the control scheme (4.3) for the parameter sets  $D = 0.001$ ,  $k = 10$ , and  $D = 0.001$ ,  $k = 20$  were obtained with maximum relative error  $10^{-2}$  using a C++ implementation of the Explicit-Implicit FC-Solver in run times of 40 minutes using 28 MB of memory and two hours using 50 MB of memory, respectively.

Figures 4.3 and 4.4 display numerical solutions obtained from a C++ implementation of the Explicit-Implicit FC solver for the control scheme (4.3) for two different parameter sets:  $D = 0.001$ ,  $k = 20$  and  $D = 0.001$ ,  $k = 10$ . To better display how the control scheme affects the concentration toward the center of the domain, the colormaps in both Figures 4.4 and 4.3 have been truncated to range from .3 to 1.1. The extremely steep boundary layers can be clearly seen in the 3D view of the concentration at each end time; note especially the extremely steep boundary layers on the order of  $10^4$  that have been accurately resolved by the Explicit-Implicit FC solver in solution for the parameter set  $D = 0.001$ ,  $k = 20$ , see Figure 4.4(n).

For the numerical results obtained using the Explicit-Implicit FC solver, we chose the grid parameters  $a = 0.41$ ,  $b = 0.4$ , and

$$\begin{aligned}\delta &= 10 \frac{D}{m_y k}, \\ \epsilon &= 10 \frac{2D}{m_u k + \sqrt{m_y m_u k^2 + 4Dm_u}}, \\ v &= 5 \frac{2D}{m_u k + \sqrt{m_y m_u k^2 + 4Dm_u}},\end{aligned}$$

where  $m_y$  and  $m_u$  are the maximum values of  $M_y$  and  $M_u$  over the domain  $\Omega$ , respectively. For the magnet configuration described in [51] and used in the numerical simulations presented in this section, we have  $m_y = 0.0065$  and  $m_u = 0.0656$ . The Implicit-Implicit FC solver uses the grid parameters  $a = 0.41$ ,  $b = 0.4$ ,  $\delta_2 = 0.1$  and  $v_2 = b - (\sqrt{2}a + 0.05)$ .

#### 4.4.2 Future Work

From examining the numerical simulations of the ad hoc control scheme proposed in [51], we can see that the development of more sophisticated control schemes is required to enable the desired manipulation of the ferrofluid. An example of one such control scheme is provided by [33], wherein a small circle of ferrofluid (modeled by a Gaussian) located at the left  $x$ -boundary is moved along the  $x$ -axis toward the center of the domain. The behavior of the ferrofluid in this example intrinsically differs from that displayed in our previous control setup by two key factors: 1) a much smaller diffusion coefficient on the order of  $D = 10^{-7}$  is used and 2) the combination of a small Gaussian initial condition and the proposed current schematic does not give rise to boundary layer behavior. The combination of these two factors renders direct application of the above Explicit-Implicit and

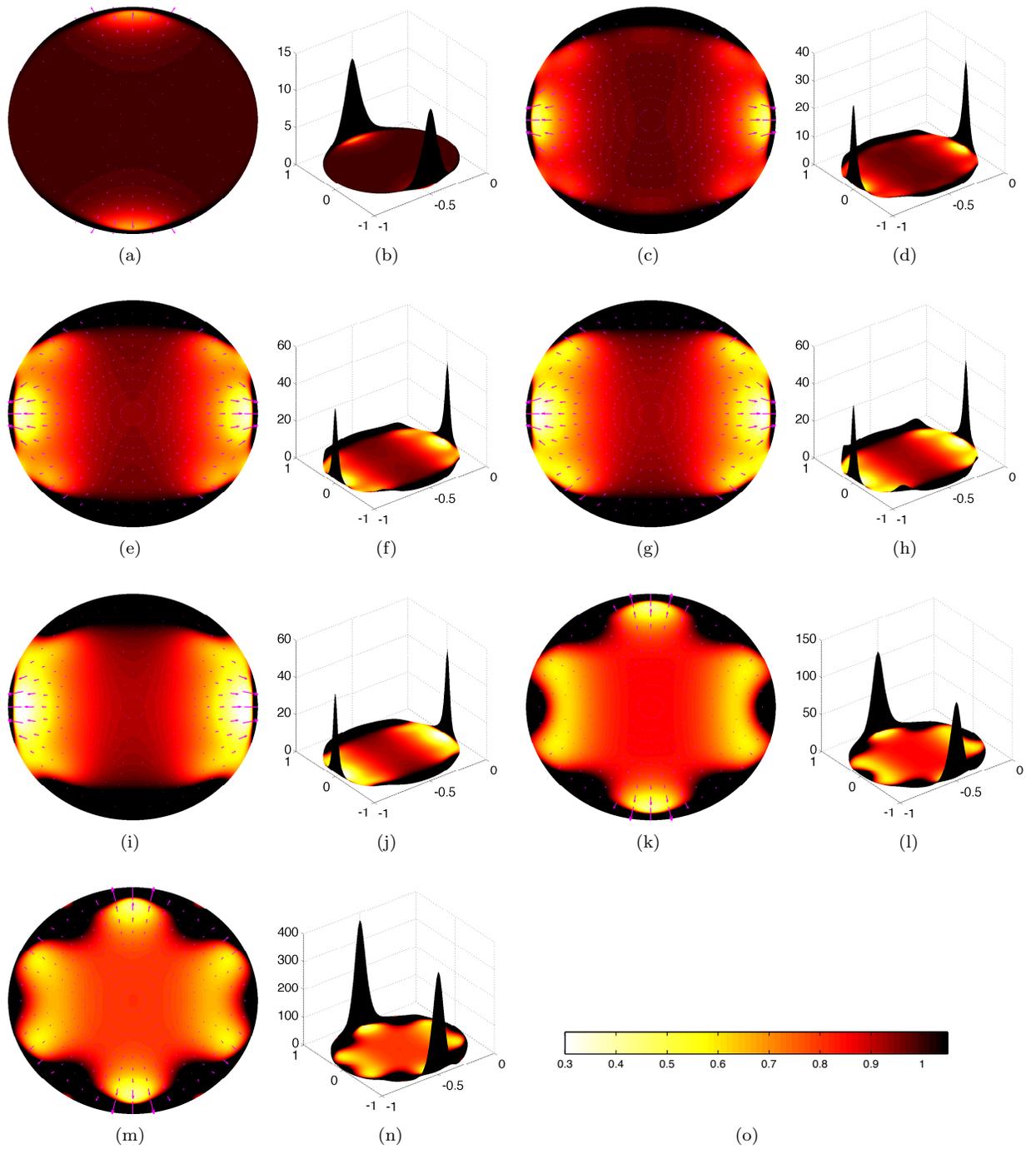


Figure 4.3: Numerical solution of the ad hoc control scheme (4.3) using the parameter choices  $D = 0.001$ ,  $k = 10$ . The first seven figure sets contain the overhead and corresponding 3D views of the concentration values at the end time of each of the seven current and time sets. The pink arrows represent the gradient of the norm of the magnetic field,  $\nabla|\vec{H}(x, y)|^2$ , used to manipulate the ferrofluid. Panel (o) represents the truncated colormap used to better display the movement of the ferrofluid in the center of the domain.

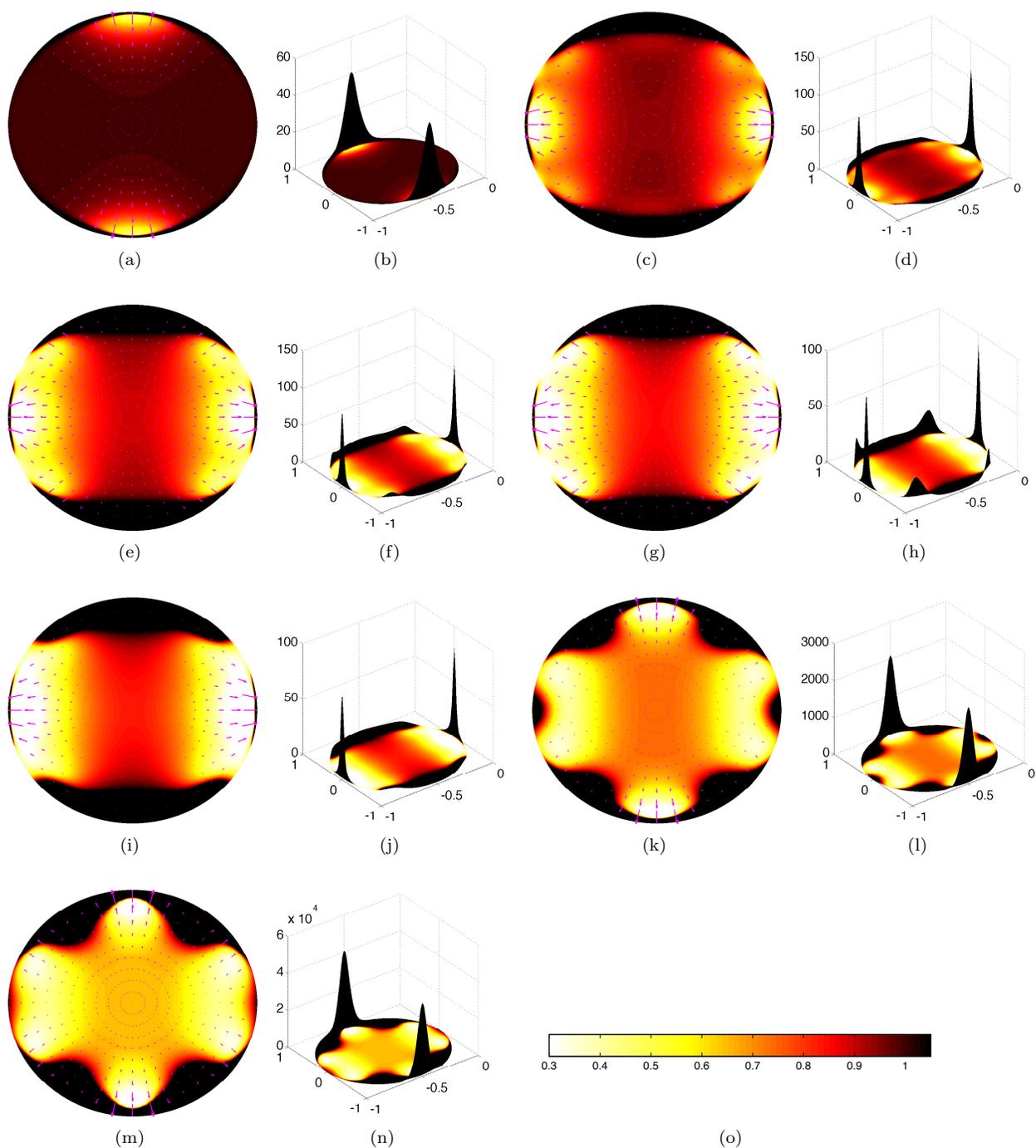


Figure 4.4: Numerical solution of the ad hoc control scheme (4.3) using the parameter choices  $D = 0.001$ ,  $k = 20$ . The first seven figure sets contain the overhead and corresponding 3D views of the concentration values at the end time of each of the seven current and time sets. The pink arrows represent the gradient of the norm of the magnetic field,  $\nabla|\vec{H}(x, y)|^2$ , used to manipulate the ferrofluid. Figure (o) represents the truncated colormap used to better display the movement of the ferrofluid in the center of the domain. Note, e.g., the extremely steep boundary layer of magnitude  $10^4$  that has been accurately captured by the Explicit-Implicit FC solver, see Figure (n).

Implicit-Implicit solvers to be somewhat inefficient. First, due to the lack of boundary layer behavior, the use of the fine annular domain toward the boundary specifically implemented for the accurate resolution of steep boundary layers is no longer necessary. Second, the small diffusion coefficient significantly lessens the restriction imposed on the time step by the CFL condition to ensure stability. This, in turn, enables the efficient use of explicit schemes in the full domain.

To provide accurate numerical solution of such a model, we suggest the development of the Explicit-Explicit FC-Solver. Easily derivable from the set of numerical tools presented in this chapter, the proposed Explicit-Explicit FC method would depend on numerical solution over only two domains: an annular domain with a polar mesh such as  $R = \{(r, \theta)\}$  extending from the outermost boundary into the center of the domain, and an overlapping square grid, such as  $S = \{(x, y)\}$ , located over the center of the domain. Use of the Explicit FC-based solver discussed in Section 2.2 would provide efficient and accurate numerical solution over both domains.

## Chapter 5

# Positron Emission Tomography

### 5.1 Introduction

As mentioned in Chapter 1, two major types of reconstruction methodologies exist in the PET field: 1) Iterative methods, which produce high-quality images but are quite slow, and 2) Methodologies based on the FBP approach [20] which, although fast, do not deliver acceptable image quality. As a result, slow iterative approaches are almost exclusively used in commercial PET devices [46]. In recent work [9] we introduced a new FBP-based methodology that incorporates (i) A certain averaging technique that helps moderate the detrimental effects of high frequencies in the reconstruction process, and (ii) A new Fejér-based filtering procedure. The resulting Fejér-mFBP algorithm runs at essentially the same cost as previous FBP-based methods and, for the types of noise typical in present day PET devices, it gives rise to images of significantly higher quality than previous FBP-based approaches—compare, e.g., Figures 5.3 and 5.6, and see Figure 5.7 and Remarks 5.4.2 and 5.4.3.

In FBP, the Radon transform associated with the PET scanning process is inverted using a discretized version of a well-known closed-form expression [44, 8, 7, 6]. In detail, PET reconstruction results from inversion of the Radon transform

$$\hat{f}(\rho, \theta) = \int_{-\infty}^{\infty} F(\tau, \rho, \theta) d\tau, \quad (5.1)$$

where, letting  $f(x_1, x_2)$  denote the density of the scanned object as a function of Cartesian coordinates  $x_1, x_2$ , the function  $F$  is given by

$$F(\tau, \rho, \theta) \equiv f(\tau \cos \theta - \rho \sin \theta, \tau \sin \theta + \rho \cos \theta),$$

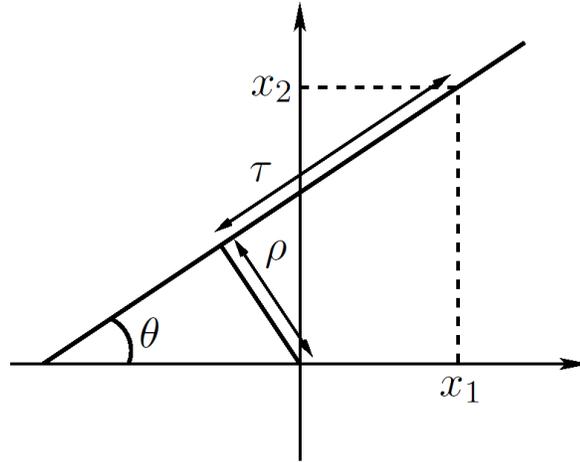


Figure 5.1: Variables and geometry associated with the Radon Transform.

together with the change of variables

$$\rho = x_2 \cos \theta - x_1 \sin \theta,$$

$$\tau = x_2 \sin \theta + x_1 \cos \theta;$$

see Figure 5.1.

Writing  $\partial_{x_i} = \frac{\partial}{\partial x_i}$ , the inverse Radon transform, in turn, is given by the closed-form expression

$$f(x_1, x_2) = \frac{1}{4i\pi^2} (\partial_{x_1} - i\partial_{x_2}) \int_0^{2\pi} e^{i\theta} h(\rho, \theta) d\theta, \quad (5.2)$$

where, using the symbol  $f$  to denote principal value integrals,

$$h(\rho, \theta) = \int_{-\infty}^{\infty} \frac{\hat{f}(\rho', \theta)}{\rho' - \rho} d\rho' \quad (5.3)$$

denotes the Hilbert transform of the function  $\hat{f}(\rho, \theta)$ . Using the relations

$$\partial_{x_1} - i\partial_{x_2} = -\sin \theta \partial_\rho - i \cos \theta \partial_\rho = -ie^{-i\theta} \partial_\rho$$

and taking into account the symmetry of the PET geometry, equation (5.2) becomes

$$f(x_1, x_2) = -\frac{1}{2\pi^2} \int_0^\pi \partial_\rho h(\rho, \theta) d\theta. \quad (5.4)$$

Typically, numerical implementations of the FBP algorithm are based on substitution of a Fourier approximation of the function  $\hat{f}$  (that can be obtained from the discrete approximate values gener-

ated by the scanning procedure) in the inversion formulae (5.3)–(5.4). The fundamental challenge inherent in such approaches relates to the ill-posed character of the Radon-transform inversion: a small perturbation of the Radon transform  $\hat{f}$ , such as those arising from measurement noise, can cause large errors on the reconstructed image [7]. The approach presented in this chapter remedies these difficulties and gives rise to reconstructions of high quality, even in the presence of high levels of noise.

This chapter is organized as follows: in Section 5.2, after a brief description of the standard FBP algorithm, we introduce our modified FBP methodology—which, on the basis of an appropriate averaging scheme, moderates the noise-magnifying character of the FBP procedure. Our use of Fejér means and our overall recommended method, the combined Fejér-mFBP algorithm, are then described in Section 5.3. Finally, the qualities of our algorithm are demonstrated in Section 5.4.

## 5.2 Modified Filtered Back-Projection Algorithms

The basic elements of the FBP algorithm are described easily: using the Fourier transform, the Hilbert transform can be expressed in the form

$$h(\rho, \theta) = \frac{1}{2i} \int_{-\infty}^{\infty} \mathcal{F}_{\rho \rightarrow \omega} \{ \hat{f}(\rho, \theta) \} \operatorname{sgn}(\omega) e^{i\omega\rho} d\omega, \quad (5.5)$$

where  $\mathcal{F}_{\rho \rightarrow \omega} \{ \hat{f}(\rho, \theta) \}$  is the Fourier transform of  $\hat{f}(\rho, \theta)$  with respect to  $\rho$ . Differentiating (5.5) with respect to  $\rho$  yields

$$\partial_{\rho} h(\rho, \theta) = \frac{1}{2} \int_{-\infty}^{\infty} \mathcal{F}_{\rho \rightarrow \omega} \{ \hat{f}(\rho, \theta) \} \omega \operatorname{sgn}(\omega) e^{i\omega\rho} d\omega \quad (5.6)$$

so that, after substitution into (5.4), we obtain

$$f(x_1, x_2) = -\frac{1}{4\pi^2} \int_0^{\pi} \int_{-\infty}^{\infty} \mathcal{F}_{\rho \rightarrow \omega} \{ \hat{f}(\rho, \theta) \} \omega \operatorname{sgn}(\omega) e^{i\omega\rho} d\omega d\theta. \quad (5.7)$$

The FBP algorithm is a direct numerical implementation of this expression based on use of the Fast Fourier Transform (FFT). Recently, use of other algorithms for evaluation of the Hilbert transform in equation (5.4) (that employ Chebyshev expansions or Spline representations of  $\hat{f}$  instead of Fourier approximations), have given rise to new implementations [21, 22] of the FBP algorithm. In the absence of noise, these algorithms are capable of producing accurate reconstructions for realistic phantoms such as the Shepp-Logan phantom (see Figure 5.2(a)).

In the remainder of this section we present a different approach to the evaluation of the Hilbert transform derivative  $\partial_{\rho} h$ . Based on approximation of the Radon transform by Fourier series in conjunction with the use of an appropriate averaging procedure, this approach gives rise, in fact, to a new variant of the FBP method, which we henceforth refer to as the mFBP algorithm. While

in itself a meaningful improvement over unfiltered FBP approaches, the mFBP algorithm can be enhanced further by incorporating Fejér summation; see Section 5.3. As shown in Section 5.4, even in presence of noise, the overall averaged Fejér-mFBP method gives rise to high-quality imaging.

To introduce the mFBP algorithm for evaluation of the derivative of the Hilbert transform, consider a function  $f(x_1, x_2)$  which vanishes outside a certain circle of radius  $p$ , and expand the function  $\hat{f} = \hat{f}(\rho, \theta)$  (which, clearly, vanishes for  $|\rho| > p$  for all angles  $\theta$ ) in a Fourier series in the variable  $\rho$  on the interval  $[-\frac{T}{2}, \frac{T}{2}]$  for some  $T > 2p$ . In other words, write

$$\hat{f}(\rho, \theta) = \frac{1}{2}a_0(\theta) + \sum_{k=1}^{\infty} a_k(\theta) \cos\left(\frac{2\pi}{T}k\rho\right) + \sum_{k=1}^{\infty} b_k(\theta) \sin\left(\frac{2\pi}{T}k\rho\right), \quad (5.8)$$

where the Fourier coefficients are given by

$$\begin{aligned} a_k(\theta) &= \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} \hat{f}(t, \theta) \cos\left(\frac{2\pi}{T}kt\right) dt = \frac{1}{T} \int_{-p}^p \hat{f}(t, \theta) \cos\left(\frac{2\pi}{T}kt\right) dt, \\ b_k(\theta) &= \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} \hat{f}(t, \theta) \sin\left(\frac{2\pi}{T}kt\right) dt = \frac{1}{T} \int_{-p}^p \hat{f}(t, \theta) \sin\left(\frac{2\pi}{T}kt\right) dt. \end{aligned} \quad (5.9)$$

From (5.3) and (5.8) we have

$$\begin{aligned} \partial_\rho h(\rho, \theta) &= \partial_\rho \int_{-\infty}^{\infty} \frac{\hat{f}(\rho', \theta)}{\rho' - \rho} d\rho' = \partial_\rho \int_{-\frac{T}{2}}^{\frac{T}{2}} \frac{\hat{f}(\rho', \theta)}{\rho' - \rho} d\rho' \\ &= -\frac{1}{2}a_0 \frac{T}{\left(\frac{T}{2}\right)^2 - \rho^2} + \sum_{k=1}^{\infty} a_k C_T(\rho, k) + \sum_{k=1}^{\infty} b_k S_T(\rho, k), \end{aligned} \quad (5.10)$$

where

$$\begin{aligned} C_T(\rho, k) &= \partial_\rho \int_{-\frac{T}{2}}^{\frac{T}{2}} \frac{1}{\rho' - \rho} \cos\left(\frac{2\pi}{T}k\rho'\right) d\rho' \quad \text{and,} \\ S_T(\rho, k) &= \partial_\rho \int_{-\frac{T}{2}}^{\frac{T}{2}} \frac{1}{\rho' - \rho} \sin\left(\frac{2\pi}{T}k\rho'\right) d\rho'. \end{aligned}$$

Now, in view of the well-known expressions for the Hilbert transforms of cosine and sine [32], we have

$$\begin{aligned} \partial_\rho \int_{-\infty}^{\infty} \frac{1}{\rho' - \rho} \cos\left(\frac{2\pi}{T}k\rho'\right) d\rho' &= -2\pi^2 \frac{k}{T} \cos\left(\frac{2k\pi\rho}{T}\right), \\ \partial_\rho \int_{-\infty}^{\infty} \frac{1}{\rho' - \rho} \sin\left(\frac{2\pi}{T}k\rho'\right) d\rho' &= -2\pi^2 \frac{k}{T} \sin\left(\frac{2k\pi\rho}{T}\right). \end{aligned} \quad (5.11)$$

As shown in Appendix B, the approximations

$$\begin{aligned} C_T(\rho, k) &\approx -2\pi^2 \frac{k}{T} \cos\left(\frac{2k\pi\rho}{T}\right), \\ S_T(\rho, k) &\approx -2\pi^2 \frac{k}{T} \sin\left(\frac{2k\pi\rho}{T}\right) \end{aligned} \quad (5.12)$$

(which arise from consideration of equation (5.11)) can be used without any noticeable impact in the quality of image reconstructions. Such a substitution is highly advantageous: it eliminates the need to evaluate  $C_T$  and  $S_T$  numerically, and it leads to the new expression

$$\partial_\rho h(\rho, \theta) \approx \frac{1}{2} a_0(\theta) \frac{T}{\left(\frac{T}{2}\right)^2 - \rho^2} - \frac{2\pi^2}{T} \sum_{k=1}^{\infty} k \left[ a_k(\theta) \cos\left(\frac{2k\pi\rho}{T}\right) - b_k(\theta) \sin\left(\frac{2k\pi\rho}{T}\right) \right], \quad (5.13)$$

which can be evaluated efficiently by means of FFTs.

Naturally, a reconstruction algorithm is obtained by truncating the infinite sum in (5.13) to a certain number  $N_c$  of Fourier modes. The choice of  $N_c$  has a tremendous impact on the quality of the reconstructed image, especially in the presence of noise. The numerical evaluation of the infinite sum in (5.13) must therefore be handled carefully: an inappropriately large value of  $N_c$  gives rise to deterioration caused by the noise magnification arising from the factors of  $k$  in equation (5.13), while an inadequately small value of  $N_c$  prevents resolution of key features (such as edges) of the original image.

To address these issues effectively, assuming that for each  $\theta$  the Radon transform is given at points  $\rho_i = \rho_0 + ih$  for  $i = 0, \dots, N_\rho - 1$ , we seek to avoid noise magnification by using a number  $N_c < N_\rho$  of Fourier modes while still employing all the information available in the  $N_\rho$  values of the Radon transform for each value of  $\theta$ . Thus, taking  $N_c$  to be an integer divisor of  $N_\rho$  and arranging the values  $\rho_j$  into several subsets  $S^i = \{\rho_j^i = \rho_i + jhN_\rho/N_c : j = 0 \dots N_c - 1\}$  of size  $N_c$  ( $i = 1, \dots, N_\rho/N_c$ ), we compute  $N_\rho/N_c$  sequences  $(a_k^i, b_k^i)$ ,  $k = 1, \dots, N_c$ , of approximate Fourier coefficients of  $\hat{f}(\rho, \theta)$  by means of FFTs—one for each set  $R^i$ . The sum (5.13) (and, hence,  $\partial_\rho h(\rho, \theta)$ ) is then obtained for any given value of  $\rho$  by averaging the approximate values of this sum resulting from the various groups  $S^i$ .

Our mFBP algorithm thus evaluates  $\partial_\rho h$  by means of the approximate expression

$$\partial_\rho h(\rho, \theta) \approx \frac{N_c}{N_\rho} \sum_{i=1}^{N_\rho/N_c} \left\{ \frac{1}{2} a_0^i(\theta) \frac{T}{\left(\frac{T}{2}\right)^2 - \rho^2} - \frac{2\pi^2}{T} \sum_{k=1}^{N_c} k \left[ a_k^i(\theta) \cos\left(\frac{2k\pi\rho}{T}\right) - b_k^i(\theta) \sin\left(\frac{2k\pi\rho}{T}\right) \right] \right\}. \quad (5.14)$$

**Remark 5.2.1** Because the approximation (5.14) results from linear operations on the coefficients  $(a_k^i, b_k^i)$ , the above averaging procedure can instead be applied directly to these sequences, or equivalently, as it is checked easily, by computing for each  $\theta$  all  $N_\rho$  pairs of Fourier coefficients  $(a_k, b_k)$  via a single FFT over the complete set  $\rho_j$ ,  $j = 0, \dots, N_\rho - 1$ , and simply truncating the sum (5.13) after  $N_c$  terms: the averaged result is thus given by

$$\partial_\rho h(\rho, \theta) \approx \frac{1}{2} a_0(\theta) \frac{T}{\left(\frac{T}{2}\right)^2 - \rho^2} - \frac{2\pi^2}{T} \sum_{k=1}^{N_c} k \left[ a_k(\theta) \cos\left(\frac{2k\pi\rho}{T}\right) - b_k(\theta) \sin\left(\frac{2k\pi\rho}{T}\right) \right]. \quad (5.15)$$

As it eliminates the need to compute multiple truncated sums, this method is somewhat preferable for simplicity.

The mFBP algorithm is now obtained from 1) Evaluation of  $\partial_\rho h(\rho, \theta)$  through equation (5.14) in conjunction with the above averaging/truncation scheme, and 2) Numerical integration of  $\partial_\rho h(\rho, \theta)$  as in (5.4) via the trapezoidal rule. A description of an interpolation procedure that gives rise to an efficient implementation of this algorithm is presented in Remark 5.3.1.

**Remark 5.2.2** Note that the restriction  $T > 2p$  on the period  $T$  of the Fourier series allows us to avoid singularities in the first terms of equations (5.10) and (5.13)—since the range of  $\rho$  values is restricted by the inequality  $|\rho| < p$ . Our numerical tests indicate that the value  $T = 4p$  results in higher visual accuracy than smaller values of this parameter; in all the reconstructions presented in this text the value  $T = 4p$  was thus used.

**Remark 5.2.3** The similarities between the classical FBP and the present mFBP algorithm are apparent from comparison of equations (5.6) and (5.13): note the multiplication of the Fourier modes by the summation index in equation (5.13) and the corresponding application of the ramp filter  $\omega \operatorname{sgn}(\omega)$  in equation (5.6). By using a Fourier series instead of a Fourier transform, however, the mFBP algorithm takes into account the finite support of the function  $\hat{f}(\rho, \theta)$ . This function can thus be approximated in a bounded set by a smaller number of frequencies than those required to adequately perform numerically the Fourier transform  $\mathcal{F}_{\rho \rightarrow \omega}\{\hat{f}(\rho, \theta)\}$  over the extended  $\omega$  interval necessary for approximation of the infinite integral in equation (5.6).

As previously mentioned, a high-quality reconstruction method results if the strategy outlined in this section is used in conjunction with Fejér means, as discussed in Section 5.3.

### 5.3 Fejér-mFBP Algorithm

A major drawback of the FBP algorithm is the noise enhancement that arises from the factor  $\omega$  in the integrand of equation (5.7): any noise existing in the Radon transform  $\hat{f}$  will be significantly amplified if, as is often the case, large values of  $\omega$  must be used. While the mFBP algorithm mitigates the debilitating effects of noise-amplifying high-frequency Fourier modes through implementation of an appropriate averaging procedure, this algorithm may still fail to reconstruct key features of an image in the presence of high levels of noise, see, e.g., Figure 5.5(c).

In addition to noise amplification, the use of Fourier series gives rise to one other inherent difficulty: Fourier series approximations of discontinuous periodic functions converge very poorly—in addition to slow convergence, they suffer from a notorious ringing effect, the “Gibbs phenomenon”, around discontinuity points. While the Radon transform  $\hat{f}$  is not itself discontinuous, the original

image  $f$  typically is. Since equations (5.7) and (5.13) effectively present  $f$  as Fourier series, direct application of such formulae yield reconstructions that suffer from the Gibbs phenomenon.

To address the Gibbs phenomenon and noise amplification problems, the Fejér-mFBP approach introduced in this text approximates the Fourier sums present in the mFBP algorithm by their corresponding Fejér means, as discussed below. The Fejér approach possesses useful Fourier approximation properties [20, 54, 57] which lead to corresponding gains in image quality. Figures 5.5(c) and 5.6(c) demonstrate the quality gains that result as the Fejér means procedure is used instead of direct summation of Fourier series: using the same data and under the same noise level, the Fejér-mFBP reconstruction shown in Figure 5.6(c) is clearly superior to the mFBP reconstruction depicted in Figure 5.5(c), for which Fejér means were not used.

Figure 5.7, in turn, compares the performance of our mFBP approach when used in conjunction with Fejér means vs. some of the best filtering procedures otherwise available. Clearly, the Fejér-mFBP reconstruction 5.7(f) represents a significant improvement over the results, shown in Figures 5.7(a) through 5.7(e), of the algorithms resulting as combinations of the mFBP with previous filtering methodologies, see Section 5.4.2 and, in particular, Remarks 5.4.2 and 5.4.3.

To introduce the Fejér-mFBP algorithm, recall that the  $N$ -th order Fejér mean [20] of a function  $g$  is defined, quite simply, as the average of the first  $N$  partial sums of the Fourier series of  $g$ :

$$\sigma_N(g)(x) = \frac{1}{N} \sum_{k=0}^N S_k(g)(x),$$

where

$$S_k(g) = \sum_{j=0}^k a_j e^{2\pi i j x/T}.$$

The beneficial effect of the Fejér averaging procedure can be appreciated easily by considering the associated Dirichlet and Fejér kernels. As is known, the  $N$ -th partial sum of the Fourier series of a function  $g$  can be represented as the convolution of  $g$  with the Dirichlet kernel

$$D_N(x) = \frac{\sin\left(\left(N - \frac{1}{2}\right)x\right)}{\sin\left(\frac{x}{2}\right)}.$$

The Fejér means  $\sigma_N$  of  $g$ , in turn, is given by convolution with the Fejér kernel  $F_N$ :

$$\sigma_N(g)(x) = F_N * g \quad , \quad \text{where}$$

$$F_N(x) = \frac{1}{N} \frac{\sin^2\left(N\frac{x}{2}\right)}{\sin^2\left(\frac{x}{2}\right)}.$$

We see that, while the Dirichlet kernel takes on both negative and positive values, making convergence of the series dependent on subtle cancellations that only occur for smooth functions (and do

not occur for scanning errors, that are typically not smooth), the Fejér kernel takes on only positive values. Use of Fejér means thus significantly reduces cancellation errors, and therefore, the impact of noise and the Gibbs' phenomenon on reconstructed functions and images.

Using a Fejér series of order  $N_c$  in conjunction with the averaging scheme introduced in Section 5.2 (Remark 5.2.1 and preceding paragraphs) we obtain the approximation

$$\partial_\rho h(\rho, \theta) \approx \frac{N_c}{N_\rho} \sum_{i=1}^{N_\rho/N_c} \left\{ \frac{1}{2} a_0^i(\theta) \frac{T}{\left(\frac{T}{2}\right)^2 - \rho^2} - \frac{2\pi^2}{T} \sum_{k=1}^{N_c} k \frac{N_c - k}{N_c} \left[ a_k^i(\theta) \cos\left(\frac{2k\pi\rho}{T}\right) - b_k^i(\theta) \sin\left(\frac{2k\pi\rho}{T}\right) \right] \right\}.$$

The Fejér-mFBP algorithm results from an efficient (FFT-based) numerical implementation of this expression in conjunction with equation (5.4). As discussed in Remark 5.2.1, however, the equivalent truncated sum

$$\partial_\rho h(\rho, \theta) \approx \frac{1}{2} a_0(\theta) \frac{T}{\left(\frac{T}{2}\right)^2 - \rho^2} - \frac{2\pi^2}{T} \sum_{k=1}^{N_c-1} k \frac{N_c - k}{N_c} \left[ a_k(\theta) \cos\left(\frac{2k\pi\rho}{T}\right) - b_k(\theta) \sin\left(\frac{2k\pi\rho}{T}\right) \right] \quad (5.16)$$

with coefficients  $a_k$  and  $b_k$  computed using all  $N_\rho$  values of  $\hat{f}(\rho, \theta)$  for each  $\theta$  is used in numerical implementation of the Fejér-mFBP algorithm.

**Remark 5.3.1** for computational efficiency our mFBP and Fejér-mFBP algorithms make use of an interpolation scheme. Indeed, in order to produce the derivative of the Hilbert transform (5.15) for a given angle  $\theta$  and each value  $\rho = x_2 \cos \theta - x_1 \sin \theta$  required for various pixel coordinates  $(x_1, x_2)$  (see equation (5.4)), our algorithms first evaluate the quantity  $\partial_\rho h(\rho, \theta)$  for values of  $\rho$  in a fixed set  $R = \{r_i = -1.5 + 3i/(P-1) : i = 0, \dots, P-1\}$ —from which all needed values of  $\partial_\rho h(\rho, \theta)$  can be obtained by interpolation. (Here we are working under the assumption that the image is to be obtained at points  $(x_1, x_2)$  within the unit square, so that  $p$  must be  $\geq \sqrt{2}$ —and, thus, larger than  $\sqrt{x_1^2 + x_2^2}$  for all points  $(x_1, x_2)$  in the image. In our definition above of the set  $R$  we have assumed  $p = 1.5$ .) The details of the interpolation procedure are as follows: taking into account the fact that the  $\theta$ -integration in equation (5.4) is to be performed by means of an application of the trapezoidal rule based on the set  $\Theta = \{\theta_j = \pi j/(N_\theta - 1) : j = 0, \dots, N_\theta - 1\}$  (which we chose to coincide with the set of angles  $\theta$  at which values of the Radon transform are given), for each  $\theta_j \in \Theta$  our algorithms evaluate  $\partial_\rho h(r_i, \theta_j)$  for all  $r_i \in R$  and then approximate the values  $\partial_\rho h(x_2 \cos \theta_j - x_1 \sin \theta_j, \theta_j)$  through nearest-neighbor linear interpolation in  $\rho$  from the set  $R$ . In this manner, the contributions to the trapezoidal-rule approximations to the  $\theta$ -integrals (5.4) for all needed values of  $x_1$  and  $x_2$  can be obtained at a reduced computational cost. In all the numerical examples shown in what follows we used the value  $P = 3073$  which, at the same time provides adequate resolution for the interpolation procedure in the interval  $-p \leq \rho \leq p$ , and it leads to efficient evaluation of (5.15) and (5.16)—since, after the needed zero-padding to the periodicity interval  $-T/2 \leq \rho \leq T/2$ , this

value of  $P$  gives rise to FFTs of size equal to a power of 2:  $(3073 - 1)T/(r_{P-1} - r_0) = 4096 = 2^{12}$ .

In summary, the Fejér-mFBP algorithm proceeds as follows:

1. For each  $\theta \in \Theta = \{\theta_j = \pi j/(N_\theta - 1) : j = 0, \dots, N_\theta - 1\}$  use FFTs to produce approximate values of the Fourier coefficients  $a_k$  and  $b_k$  in equation (5.9) using the  $N$  available values of the Radon transform  $\hat{f}$ :

$$a_k(\theta) = \frac{1}{T} \sum_{i=0}^{N_\rho} \hat{f}(\rho_i, \theta) \cos\left(\frac{2k\pi\rho_i}{T}\right), \quad b_k(\theta) = \frac{1}{T} \sum_{i=0}^{N_\rho} \hat{f}(\rho_i, \theta) \sin\left(\frac{2k\pi\rho_i}{T}\right).$$

2. Selecting an appropriate value of  $N_c$  (see Section 5.2) for each  $\theta_j \in \Theta$  use FFTs to evaluate the right-hand side of equation (5.16) (which provides an approximation of  $\partial_\rho h(\rho, \theta)$ ) at the values  $R = \{r_i = -1.5 + 3i/(P - 1) : i = 0, \dots, P - 1\}$ , as described in Remark 5.3.1.
3. For every pixel point  $(x_1, x_2)$ , use nearest-neighbor linear interpolation from the set  $R$  to obtain approximate values of  $\partial_\rho h(x_2 \cos \theta_j - x_1 \sin \theta_j, \theta_j)$  for each  $\theta_j \in \Theta$ , and include the corresponding contribution to the trapezoidal rule approximation of the integral (5.4).

The performance of our algorithm, including comparisons with results produced by various other approaches, is demonstrated in Section 5.4.

## 5.4 Numerical Tests and Results

We present numerical tests for the  $500 \times 500$  pixel Shepp-Logan phantom depicted in Figure 5.2(a). Unless stated otherwise, the reconstructed images were generated by means of C++ implementations of the mFBP and Fejér-mFBP algorithms. The computing times required by our algorithms are not higher than those required by the standard FBP methods: all of the algorithms presented in this text require computing times of approximately 3.6 seconds to reconstruct  $500 \times 500$  pixel images on a 1.6GHz Intel Core i7 processor.

The Shepp-Logan phantom is a well-known model of a head section constructed from the summation of different sized ellipses of various density values:

$$\begin{aligned} f(x_1, x_2) = & C(1, x_1, x_2, .69, .92, 0, 0, 0) + C(-.98, x_1, x_2, .6624, .9740, 0, -.0184, 0) \\ & + C(-.02, x_1, x_2, .1100, .3100, .22, 0, -\pi/10) + C(-.02, x_1, x_2, .1600, .4100, -.22, 0, \pi/10) \\ & + C(.01, x_1, x_2, .2100, .2500, 0, .35, 0) + C(.01, x_1, x_2, .0460, .0460, 0, .1, 0) \\ & + C(.01, x_1, x_2, .0460, .0460, 0, -.1, 0) + C(.01, x_1, x_2, .0460, .0230, -.08, -.605, 0) \\ & + C(.01, x_1, x_2, .0230, .0230, 0, -.606, 0) + C(.01, x_1, x_2, .0230, .0460, .06, -.605, 0), \end{aligned}$$

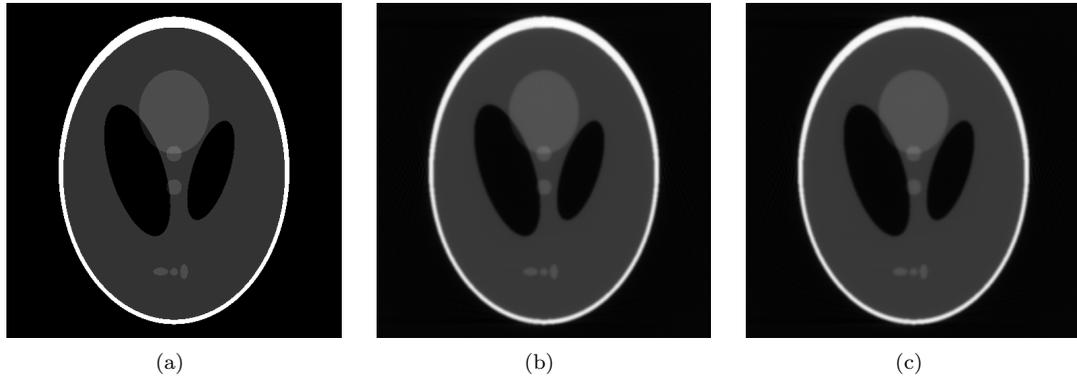


Figure 5.2: The original  $500 \times 500$  pixel Shepp-Logan phantom (a), and noiseless-data reconstructions (b) and (c) resulting from the averaged mFBP and averaged Fejér-mFBP algorithms, respectively. Both reconstructions were obtained using the parameter values  $N_\theta = 1422$ ,  $N_\rho = 711$ , and  $N_c = 237$ .

where  $C(w, x_1, x_2, a, b, x_1^0, x_2^0, \phi)$  takes on the value  $w$  inside the ellipse of horizontal radius  $a$  and vertical radius  $b$ , centered at the point  $(x_1^0, x_2^0)$ , and rotated by the angle  $\phi$ , and zero elsewhere. The data function  $\hat{f}(\rho, \theta)$  was obtained using the Radon transform (5.1) with  $N_\theta = 1422$  and  $N_\rho = 711$ —this corresponds to a PET scanner with 1422 detector pairs (tubes). The choice of  $N_\rho = 711$  was made to facilitate comparisons with reconstructions based on MATLAB’s built-in functions ‘radon’ and ‘iradon’: for a given image, the number of detector pairs  $\rho$  used by ‘radon’ to generate the Radon transform (and therefore reconstruct the image from its Radon transform) is determined by the image size—e.g., the Radon transform of a  $500 \times 500$  pixel image is computed by the MATLAB functions using 711 values of  $\rho$ .

While the main goal of this text is to address the effects of noise on the reconstruction process, it is useful to preface our discussion in these regards by providing an indication of the performance of the mFBP and Fejér-mFBP algorithms when applied to noiseless data. Figures 5.2(b) and 5.2(c) thus present (averaged) mFBP and Fejér-mFBP noiseless-data reconstructions of the Shepp-Logan phantom using with  $N_c = 237$ . Comparison with the original phantom, which is displayed in Figure 5.2(a), shows that both algorithms provide highly accurate noiseless reconstructions: all of the features of the Shepp-Logan phantom are sharply recovered in both cases.

The performance of our various algorithms in presence of noise is described in what follows. In brief, our recommended algorithm is the averaged Fejér-mFBP approach: as shown in Figure 5.6, this algorithm displays an excellent performance even in presence of significant levels of noise. Comparisons of variants of this method (including algorithms, discussed in Section 5.4.2, that incorporate previous filtering methodologies), demonstrate the significant benefits that result from the recommended approach.

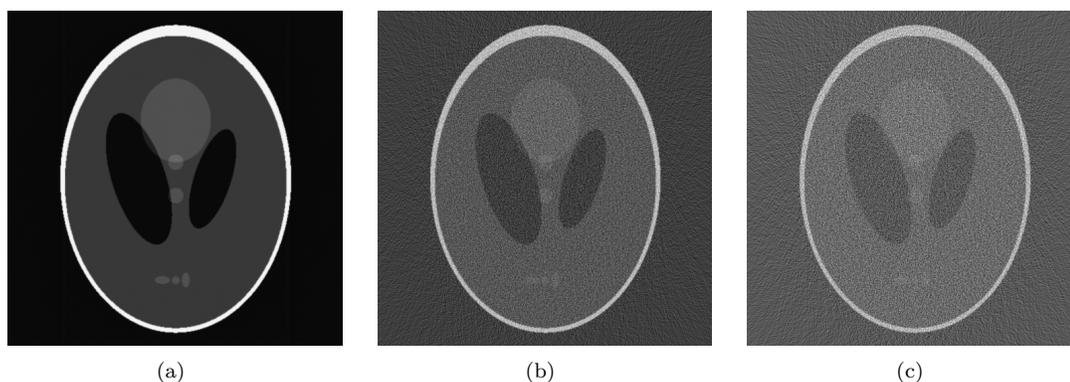


Figure 5.3: MATLAB's 'iradon' reconstructions of the  $500 \times 500$  pixel Shepp-Logan phantom with  $N_\theta = 1422$  and  $N_\rho = 711$  for noiseless data (Figure (a)), as well as 10.52% noise (Figure (b)) and 18.38% noise (Figure (c)).

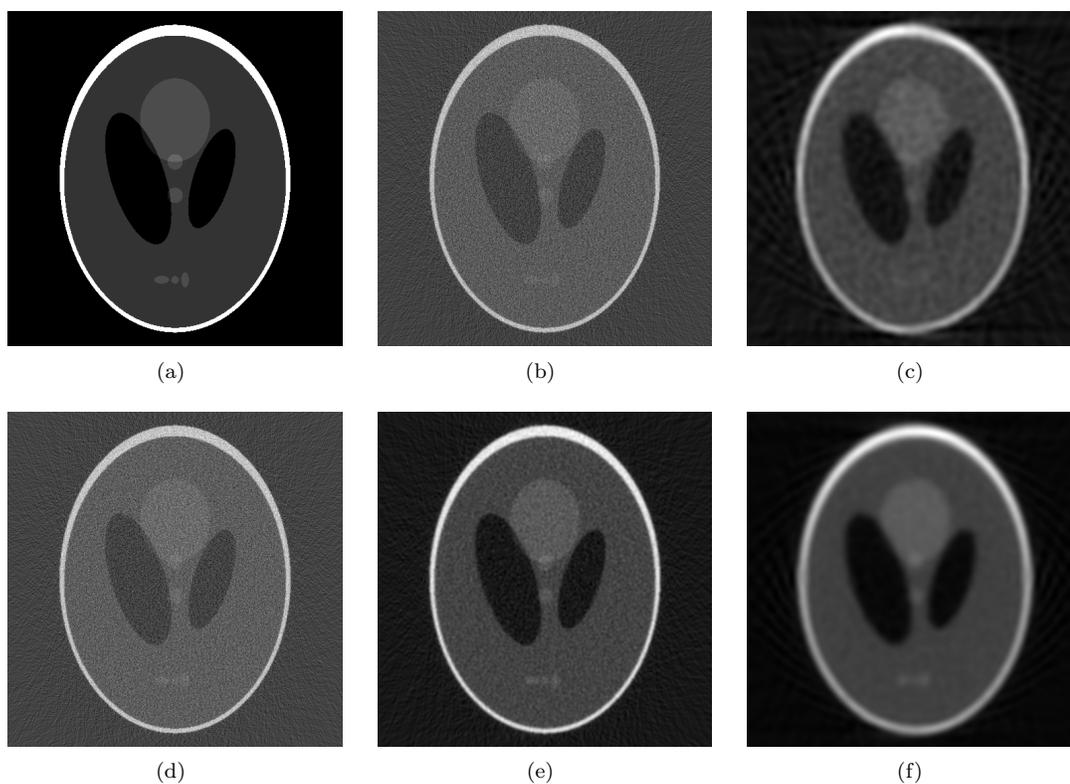


Figure 5.4: (a) The original  $500 \times 500$  pixel Shepp-Logan phantom and its Fejér-mFBP reconstructions (b)–(f) in the presence of 10.52% noise, with  $N_\theta = 1422$ ,  $N_\rho = 711$ , and various values of  $N_c$ : (b)  $N_c = 237$  (c)  $N_c = 79$ , (d)  $N_c = 711$ , (e)  $N_c = 237$ , (f)  $N_c = 79$ .  $N_\rho/N_c$ -fold averaging was used to produce Figures (e) and (f). The three-fold averaged Fejér-mFBP reconstruction shown in Figure (e) is clearly superior to those obtained from use of other parameter values.

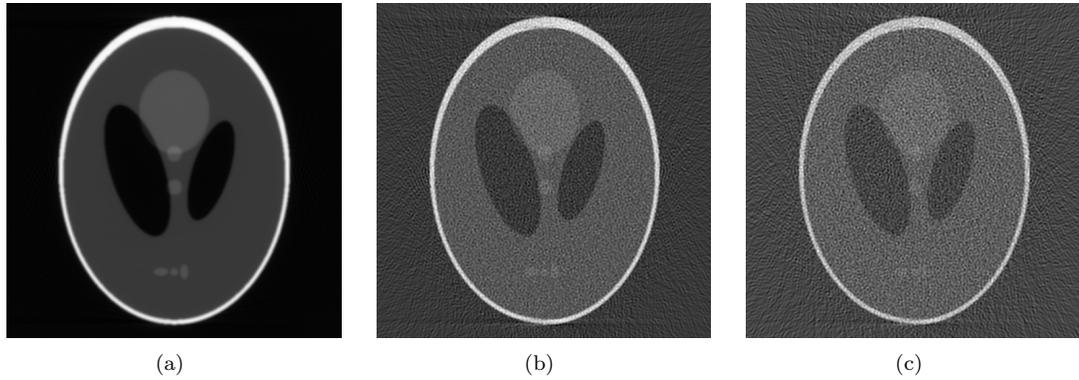


Figure 5.5: mFBP reconstructions of the  $500 \times 500$  pixel Shepp-Logan phantom with  $N_\theta = 1422$  and  $N_\rho = 711$  for noiseless data (Figure (a)), as well as 10.52% noise (Figure (b)) and 18.38% noise (Figure (c)).

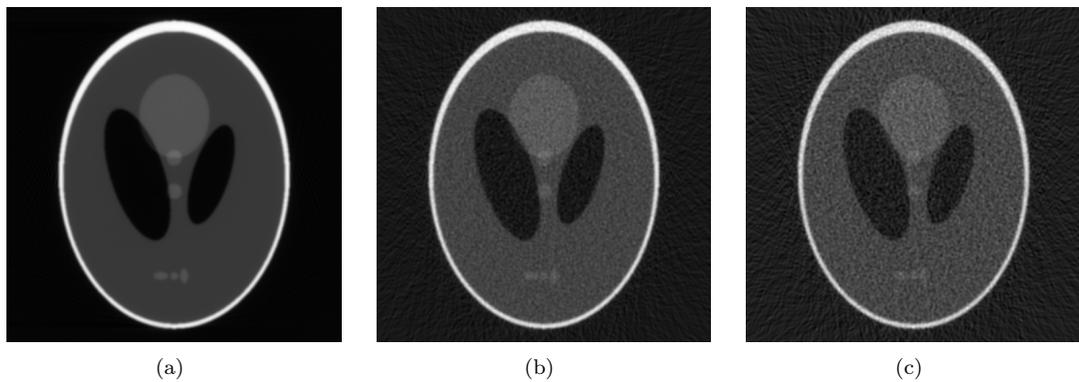


Figure 5.6: fejer-mFBP reconstructions of the  $500 \times 500$  pixel Shepp-Logan phantom with  $N_\theta = 1422$  and  $N_\rho = 711$  for noiseless data (Figure (a)), as well as 10.52% noise (Figure (b)) and 18.38% noise (Figure (c)).

### 5.4.1 Effects of Noise

As mentioned earlier in this text, owing to high frequency amplification, noise generally has a dramatic effect on the quality of the FBP reconstructions. This effect is demonstrated in Figure 5.3, where the Shepp-Logan phantom is reconstructed by the standard MATLAB FBP algorithm (function ‘iradon’) without noise (Figure 5.3(a)) and with noise (amounting to 10.52% and 18.38% of the intensity for Figures 5.3(b) and 5.3(c), respectively, see Remark 5.4.1).

The Fejér-mFBP algorithm introduced in Section 5.3 relies on two main devices to stem the high-frequency noise-amplification effect inherent in the Radon-transform inversion formula: on one hand it limits the number  $N_c$  of Fourier modes used (while making use of all the information contained in the data function  $\hat{f}$ ), and, on the other hand, it uses Fejér summation to neutralize the ill effects associated with the Gibbs phenomenon. Because the number of Fourier modes  $N_c$  has such an important impact on the reconstruction process (see also Section 5.2), a careful selection of this parameter must be made. A sample of an extensive set of experiments we performed in these regards is presented in Figure 5.4—which displays reconstructions of the Shepp-Logan phantom in the presence of 10.52% noise (see Remark 5.4.1) using the Fejér-mFBP algorithm with various values of  $N_c$ . Considering these and other images we determined that, for noise levels of the order of 10% to 20%, and for the numbers of detector pairs of the order of those available in present day PET scanners (we considered  $N_\rho = 711$ ), use of three-fold averaging ( $N_\rho/N_c \sim 3$ , or, in the present case,  $N_c = 237$ ) leads to arguably optimal reconstructions (compare Figure 5.4(e) to other images in Figure 5.4). We thus use  $N_c = 237$  with averaging in all following numerical results presented.

In presence of high levels of noise, the mFBP approach ameliorates reconstructions (compare Figures 5.5(b), 5.5(c) to Figures 5.3(b), 5.3(c)). Clearly, further, the Fejér-mFBP reconstructions with adequately chosen parameters (e.g., Figures 5.6(b), 5.6(c)) are of significantly higher quality than corresponding reconstructions produced by the standard FBP algorithm (such as Figures 5.3(b), 5.3(c)). The Fejér-mFBP reconstructions also improve upon the mFBP reconstructions, as can be seen by comparing Figures 5.5(c) and 5.6(c): some loss of information has resulted in both cases but note, for example, that while some of the smaller, light-grey ellipses (that are present toward the bottom of the phantom) are not clearly distinguishable in the image reconstructed using the mFBP algorithm (Figure 5.5(c)), they are clearly visible in the image reconstructed using the Fejér-mFBP algorithm (Figure 5.6(c)).

Comparisons of the Fejér-mFBP approach with algorithms resulting from other filtering methods are presented in Section 5.4.2.

**Remark 5.4.1** To simulate measurement noise we use the algorithm described in [34]: 1) The function values  $\hat{f}(\rho, \theta)$  are first scaled by a constant multiplicative weight until their maximum value is equal to a prescribed number  $M$ ; 2) Each scaled function value  $S$  is then replaced by a random

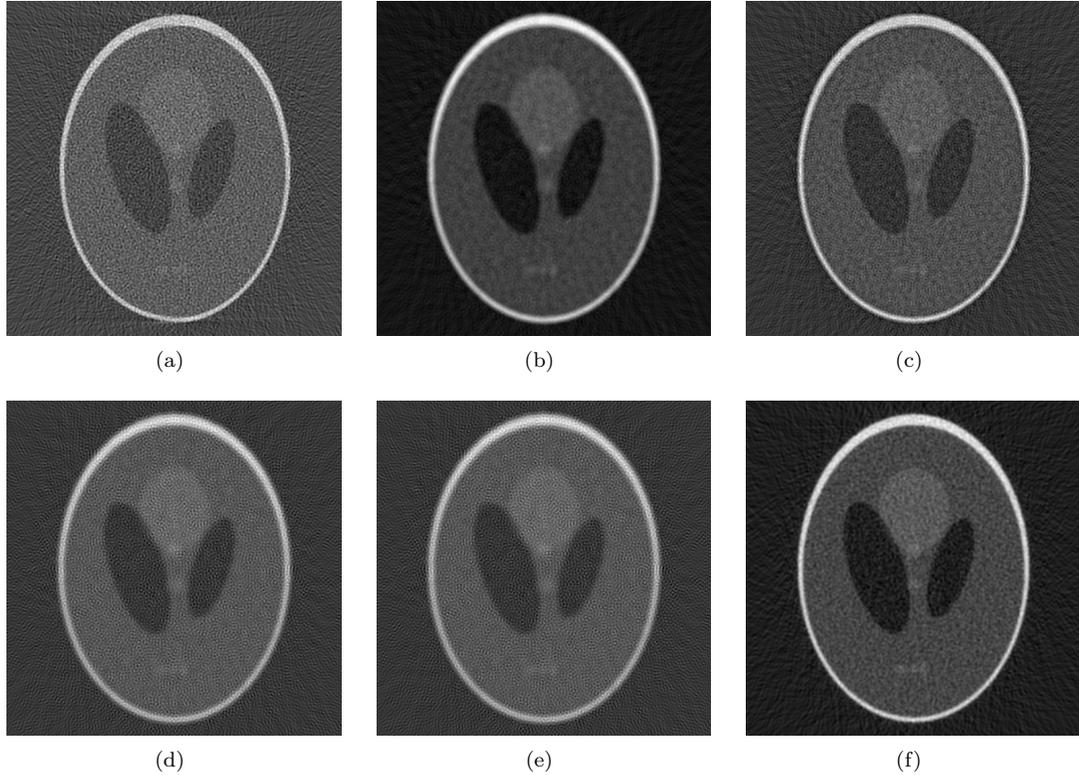


Figure 5.7: Comparison of Fejér filtering and previous filtering methodologies (using, in all cases, the new three-fold averaging approach) for the  $500 \times 500$  pixel Shepp-Logan phantom with 18.38% noise: Figures (a) through (f) demonstrate the performance of the basic mFBP algorithm with  $N_\theta = 1422$ ,  $N_\rho = 711$  and  $N_c = 237$  (three-fold averaging), when used in conjunction with: (a) No filtering, (b) The Shepp-Logan filter (5.17), (c) The cosine filter (5.18), (d) The Hamming window (5.19), (e) The Hann window (5.20), and (f) The Fejér-based filter introduced in Section 5.3. In particular, the image (f) appears to be sharper than the best of its counterparts—which is, arguably, Figure (b): notice, e.g., the parallel array of three ellipses that is more sharply resolved in Figure (f) than it is in any of the other images under consideration. Compare also against Figure 5.3(c).

number generated from a Poisson distribution with mean equal to  $S$ ; and finally 3) The noisy data is divided by the original weight. The amount of noise present in a given Radon transform is evaluated as 100 times the quotient of the  $L^2$  norm of the difference of the given (noisy) data and the original data by the  $L^2$  norm of the original data. For the Shepp-Logan phantom, the choices  $M = 50$  and  $M = 150$  yield data functions with 18.38% and 10.52% noise respectively.

#### 5.4.2 Fejér-mFBP and Classical Filtering

The difficulties arising from the classical FBP reconstruction algorithm are widely recognized and have motivated development of a range of filtering techniques [20]. We note that any such filtering method can be used in conjunction with our mFBP and Fejér-mFBP methods with possible additional benefits. To explore this possibility, in Figure 5.7 we present reconstructions of the

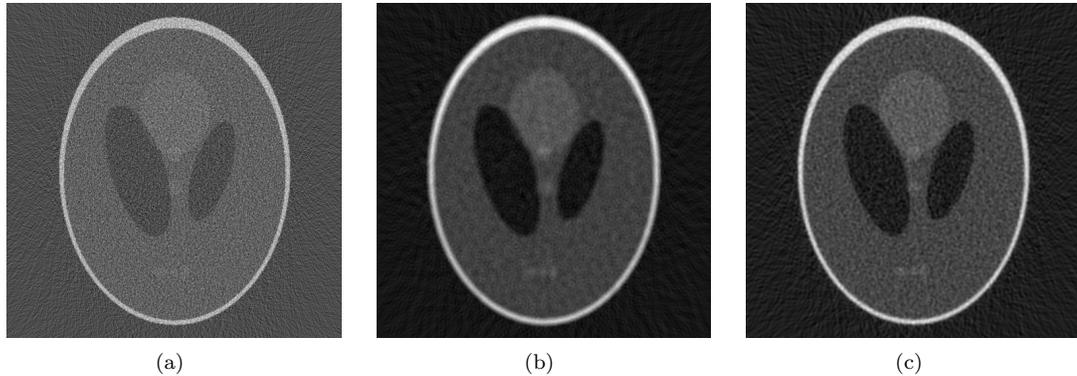


Figure 5.8: Shepp-Logan-mFBP and Fejér-mFBP vs. Shepp-Logan-filtered FBP for the  $500 \times 500$  pixel Shepp-Logan phantom with 18.38% noise. Using the parameters  $N_\theta = 1422$  and  $N_\rho = 711$ , the reconstructions shown in Figures (a)-(c) were obtained from (a) MATLAB’s ‘iradon’ function applied with MATLAB’s Shepp-Logan filter option, (b) The mFBP algorithm with  $N_c = 237$  (three-fold averaging) in conjunction with our implementation of the Shepp-Logan filter, and (c) The Fejér-mFBP algorithm with  $N_c = 237$  (three-fold averaging).

Shepp-Logan phantom produced, in presence of 18.38% noise, by means of the mFBP algorithm in conjunction with various filtering methods—including the Fejér procedure introduced in Section 5.3 (Figure 5.7(f)) as well as several standard filters [31] (Figures 5.7(a)–5.7(e)): the Shepp-Logan filter,

$$w_k = \text{sinc}\left(\frac{2k}{N-1}\right), \quad (5.17)$$

the cosine filter,

$$w_k = \cos\left(\frac{\pi k}{(N-1)} - \frac{\pi}{2} + 1\right), \quad (5.18)$$

the Hamming window,

$$w_k = 0.54 - 0.46 * \cos\left(\frac{2\pi k}{N-1}\right), \quad (5.19)$$

and the Hann window,

$$w_k = \frac{1}{2} \left(1 + \cos\left(\frac{2\pi k}{N-1}\right)\right). \quad (5.20)$$

Note that, while all the reconstructions shown in Figure 5.7 display the prominent features of the Shepp-Logan phantom, the image produced by the Fejér-mFBP algorithm is superior to those produced by the alternatively filtered mFBP algorithms: the Fejér-mFBP image displays the combined benefits of accuracy and sharpness, and it retains the high contrast of the original phantom.

**Remark 5.4.2** Amongst the reconstructions displayed in Figure 5.7, the one that comes closest in quality to that obtained by means of the Fejér-mFBP combination is the one resulting from the Shepp-Logan-mFBP combination. Notice that the Fejér-mFBP image 5.7(f) does represent an improvement over the Shepp-Logan-mFBP image 5.7(b): for example, the upper-right portion of the

large circle appears truncated in the Shepp-Logan-mFBP image, while it is intact in the Fejér-mFBP image. Further, in the mFBP/Shepp-Logan image the lower half of the uppermost small circle is not visible, and the parallel array of three ellipses is not sharply resolved. In comparison, all of these features are crisply reconstructed in the Fejér-mFBP image 5.7(f).

**Remark 5.4.3** In view of the quality, discussed in Remark 5.4.2, of the Shepp-Logan-mFBP reconstruction, the question arises as to whether the classical FBP algorithm combined with the Shepp-Logan filter would give rise to reconstructions of similar quality. To answer this question we performed an additional series of tests, some of whose results are displayed in Figure 5.8. It is clear from these images that the mFBP strategy (and not just the filters used) has a significant impact on the quality of the reconstructions presented in this text.

## Chapter 6

# Conclusions

In this thesis we presented three numerical algorithms designed to overcome the demands of the challenging computational problems inherent in the development of new medical treatment techniques. First, we considered the challenging VMT convection-diffusion problem. Through a combination of exponential meshing, the fast ADI method, and an efficient time-stepping scheme used in conjunction with a novel “fluid-freezing” methodology, the three VMT solvers presented in this thesis effectively resolve flow of magnetically driven therapeutic drug particles through a convecting blood vessel and diffusion into surrounding tissues. The VMT solvers provide steady state solutions for simulations with realistic blood vessel parameters, thus allowing for detailed study of the complex interplay between the convective and diffusive forces present in the VMT model.

Next, we considered the problem of magnetically enhanced diffusion. In Chapter 4, we presented a variety of numerical tools based on explicit FC-based solvers and the implicit FC-AD method used in conjunction with effective domain meshing techniques to enable accurate and efficient capture of solution behaviors. Appropriate combinations of these techniques provide rapid and accurate solution of the control setup proposed in [51], in addition to enabling the development of more sophisticated control schemes.

Based on consideration and adequate treatment of fine spectral properties of discontinuous functions and numerical errors, finally, we introduced the new Fejér-mFBP algorithm for inversion of PET-scan data. This algorithm enables crisp PET reconstructions, even in presence of considerable amounts of noise, at a computing cost comparable to those required by previous lower-quality FBP-based methods.

## Appendix A

# Numerical Computation of Magnetic Fields

The basic laws of magnetostatics are given by

$$\begin{aligned}\nabla \times \vec{B} &= \frac{4\pi}{c} \vec{J}, \\ \nabla \cdot \vec{B} &= 0, \\ \vec{B} &= \mu \frac{4\pi}{c} \vec{H},\end{aligned}$$

where  $\mu$  is the magnetic permeability and  $\vec{J}$  is the current density of a given magnet (see [30] for further details). It is well-known that the magnetic field  $\vec{B}$  and, in turn,  $\vec{H}$ , may be obtained as the curl of some vector field  $\vec{A}(\vec{x})$ , the vector potential,

$$\vec{B}(\vec{x}) = \nabla \times \vec{A}(\vec{x}),$$

where the general form of the vector potential is given by

$$\vec{A}(\vec{x}) = \frac{\mu}{c} \int \frac{\vec{J}(\vec{x}')}{|\vec{x} - \vec{x}'|} d^3x'.$$

In order to ensure relevance in our comparisons, we model the electromagnets as in [51]—that is, by a series of infinite wires along the  $z$ -axis. For a wire of finite length  $2W$  along the  $z$ -axis, the vector potential  $\vec{A}(\vec{x})$  is given by

$$A_x(x) = 0, \quad A_y(x) = 0, \quad A_z(x) = \ln \left[ \frac{\sqrt{x^2 + y^2 + (z - W)^2} - W}{\sqrt{x^2 + y^2 + (z + W)^2} + W} \right]. \quad (\text{A.1})$$

Letting  $z = 0$ , as is the case for the control setup (4.1), and taking the curl of  $\vec{A}(\vec{x})$ , yields the magnetic field components

$$\begin{aligned} H_x(x, y, z) &= \frac{1}{4\pi} \frac{2WJy}{(x^2 + y^2)\sqrt{x^2 + y^2 + W^2}} \\ H_y(x, y, z) &= -\frac{1}{4\pi} \frac{2WJx}{(x^2 + y^2)\sqrt{x^2 + y^2 + W^2}} \\ H_z(x, y, z) &= 0. \end{aligned} \tag{A.2}$$

Taking the limit  $W \rightarrow \infty$ , we have the simplified components

$$\begin{aligned} H_x(x, y) &= \frac{1}{4\pi} \frac{2yJ}{x^2 + y^2}, \\ H_y(x, y) &= -\frac{1}{4\pi} \frac{2xJ}{x^2 + y^2}. \end{aligned} \tag{A.3}$$

An approximation of the magnetic field components for an electromagnet with current density  $J_i$  and centered at the point  $(x_i, y_i)$  may then be obtained from translation and integration of the components in (A.3) as follows:

$$\begin{aligned} H_x^i(x, y) &= \frac{1}{4\pi} \int_{-L/2}^{L/2} \int_{r_i}^{r_o} \frac{2(y - y_i - r)J_i}{(x - x_i - l)^2 + (y - y_i - r)^2} dr dl, \\ &\quad - \frac{1}{4\pi} \int_{-L/2}^{L/2} \int_{-r_o}^{-r_i} \frac{2(y - y_i - r)J_i}{(x - x_i - l)^2 + (y - y_i - r)^2} dr dl, \\ H_y^i(x, y) &= -\frac{1}{4\pi} \int_{-L/2}^{L/2} \int_{r_i}^{r_o} \frac{2(x - x_i - l)J_i}{(x - x_i - l)^2 + (y - y_i - r)^2} dr dl, \\ &\quad + \frac{1}{4\pi} \int_{-L/2}^{L/2} \int_{-r_o}^{-r_i} \frac{2(x - x_i - l)J_i}{(x - x_i - l)^2 + (y - y_i - r)^2} dr dl \end{aligned} \tag{A.4}$$

where  $L$  is the length of the magnet, and  $r_i$  and  $r_o$  are the inner and outer radii, respectively. Computation of the norm of the full magnetic field arises from straightforward summation of the magnetic field components for each magnet:

$$|\vec{H}(x, y)|^2 = \left( \sum_{i=1}^8 H_x^i(x, y) \right)^2 + \left( \sum_{i=1}^8 H_y^i(x, y) \right)^2.$$

In the interest of computational efficiency, the integrals (A.4) and their corresponding first and second derivatives in  $x$  and  $y$  may be precomputed and stored ahead of time. The gradient and Laplacian of the norm of the magnetic field arising from the desired combination of currents, and as required in the solution of (4.1), may then be efficiently obtained as appropriate linear combinations of the precomputed integrals and derivatives.

## Appendix B

# Hilbert Transform Approximations

As mentioned in Section 5.2, the functions  $C_T$  and  $S_T$  can be adequately approximated by corresponding sinusoids; see equation (5.12). Use of these approximations, which generally do not affect the quality of reconstructed images, lead to reductions in computing costs—the image reconstruction algorithm can be based on FFTs and thus be significantly accelerated.

To evaluate the errors implicit in the approximations (5.12), first note that, using the well-known sine and cosine integrals [1]

$$\begin{aligned}\text{Si}(x) &= \int_0^x \frac{\sin(t)}{t} dt, \\ \text{Ci}(x) &= - \int_x^\infty \frac{\cos(t)}{t} dt,\end{aligned}$$

the derivatives of the finite Hilbert transforms (5.11) are given by

$$\begin{aligned}C_T(\rho, k) &= -\frac{T(-1)^k}{T^2 - 4x^2} + \frac{2\pi k}{T} \left( \sin\left(\frac{2\pi kx}{T}\right) \gamma_T(\rho, k) - \cos\left(\frac{2\pi kx}{T}\right) \sigma_T(\rho, k) \right), \quad \text{and} \\ S_T(\rho, k) &= \frac{2\pi k}{T} \left( -\cos\left(\frac{2\pi kx}{T}\right) \gamma_T(\rho, k) + \sin\left(\frac{2\pi kx}{T}\right) \sigma_T(\rho, k) \right),\end{aligned}$$

where

$$\begin{aligned}\gamma_T(\rho, k) &= \text{Ci}\left(\frac{(-T + 2\rho)\pi}{T}\right) - \text{Ci}\left(\frac{(T - 2\rho)\pi}{T}\right), \\ \sigma_T(\rho, k) &= \text{Si}\left(\frac{(-T + 2\rho)\pi}{T}\right) + \text{Si}\left(\frac{(T - 2\rho)\pi}{T}\right).\end{aligned}$$

The approximation errors

$$\begin{aligned}E_C(\rho, k) &= \partial_\rho \int_{-\infty}^{\infty} \frac{1}{\rho' - \rho} \cos\left(\frac{2\pi}{T} k\rho'\right) d\rho' - C_T(\rho, k), \quad \text{and} \\ E_S(\rho, k) &= \partial_\rho \int_{-\infty}^{\infty} \frac{1}{\rho' - \rho} \sin\left(\frac{2\pi}{T} k\rho'\right) d\rho' - S_T(\rho, k),\end{aligned}$$

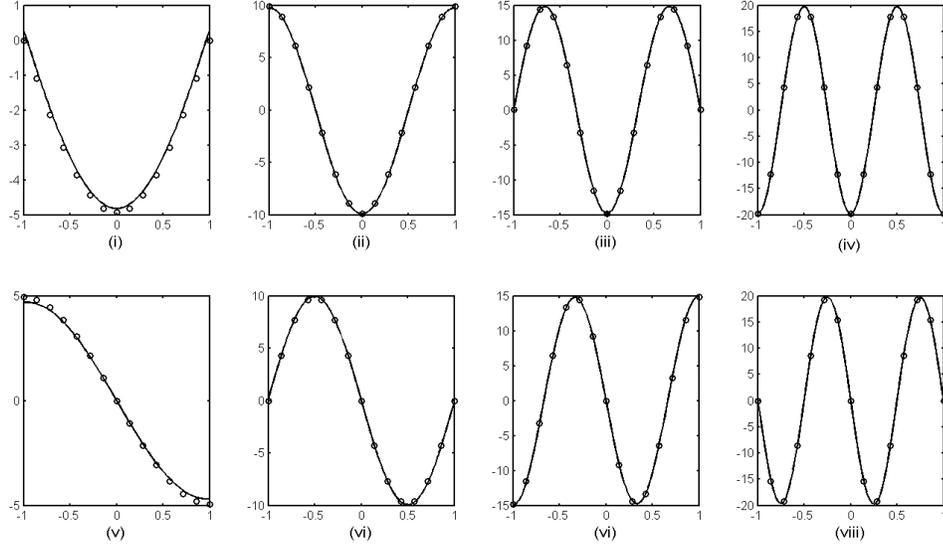


Figure B.1: The true values (solid lines) of  $C_T(\rho, k)$  for  $k = 1, 2, 3$  and  $4$  (Figures (i)–(iv)) and  $S_T(\rho, k)$  for  $k = 1, 2, 3$  and  $4$  (Figures (v)–(viii)) plotted with their corresponding approximations (circles) computed via (5.12) using  $T = 4$ .

which are clearly given by

$$E_C(\rho, k) = \partial_\rho \int_{|\rho'| \geq \frac{T}{2}} \frac{1}{\rho' - \rho} \cos\left(\frac{2\pi}{T} k \rho'\right) d\rho', \quad \text{and}$$

$$E_S(\rho, k) = \partial_\rho \int_{|\rho'| \geq \frac{T}{2}} \frac{1}{\rho' - \rho} \sin\left(\frac{2\pi}{T} k \rho'\right) d\rho,$$

can be estimated through applications of integration by parts: we obtain

$$\begin{aligned} |E_C(\rho, k)| &= \left| \left(\frac{T}{2\pi k}\right)^2 \partial_\rho \left( (-1)^k \left( \frac{1}{(T/2 - \rho)^2} + \frac{1}{(T/2 + \rho)^2} \right) + \int_{|\rho'| \geq \frac{T}{2}} \frac{2}{(\rho' - \rho)^3} \cos\left(\frac{2\pi k}{T} \rho'\right) d\rho' \right) \right| \\ &\leq 4 \left(\frac{T}{2\pi k}\right)^2 \left| \frac{1}{(T/2 - \rho)^3} - \frac{1}{(T/2 + \rho)^3} \right| = \mathcal{O}\left(\frac{1}{k^2}\right) \end{aligned}$$

and

$$\begin{aligned} |E_S(\rho, k)| &= \left| \left(\frac{T}{2\pi k}\right) \partial_\rho \left( (-1)^k \left( \frac{1}{T/2 - \rho} + \frac{1}{T/2 + \rho} \right) + \int_{|\rho'| \geq \frac{T}{2}} \frac{1}{(\rho' - \rho)^2} \sin\left(\frac{2\pi k}{T} \rho'\right) d\rho' \right) \right| \\ &\leq 2 \left(\frac{T}{2\pi k}\right) \left| \frac{1}{(T/2 - \rho)^2} - \frac{1}{(T/2 + \rho)^2} \right| = \mathcal{O}\left(\frac{1}{k}\right). \end{aligned}$$

Clearly the errors decay as  $k$  increases. In fact, the approximations (5.12) are adequately accurate for all values of  $k$ : Figure B.1 compares  $C_T$  and  $S_T$  with their corresponding approximations for  $k = 1, 2, 3$  and  $4$ ; the approximations for higher values of  $k$  are even tighter—as claimed in Section 5.2.

# Bibliography

- [1] M. Abramowitz and I.A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Dover, 1964.
- [2] N. Albin and O.P. Bruno. A spectral FC solver for the compressible Navier-Stokes equations in general domains I: Explicit time-stepping. *J. Comp. Phys.*, In print.
- [3] C. Alexiou, R. Jurgons, C. Seliger, S. Kolb, B. Heubeck, and H. Iro. Distribution of mitoxantrone after magnetic drug targeting: Fluorescence microscopic investigations on VX2 squamous cell carcinoma cells. *Zeitschrift Fur Physikalische Chemie–International Journal of Research in Physical Chemistry & Chemical Physics*, 220:235–240, 2006.
- [4] G. Ashcroft and X. Zhang. Optimized prefactored compact schemes. *J. Comp. Phys.*, 190:459–477, 2003.
- [5] A. Averbuch and L. Vozovoi. Two-dimensional parallel solver for the solution of navier-stokes equations with constant and variable coefficients using adi on cells. *Parallel Comput.*, 2:673–699, 1998.
- [6] D.L. Bailey, D.W. Townsend, P.E. Valk, and M.N. Maisey. *Positron Emission Tomography: Basic Sciences*. Springer, 2005.
- [7] H.H. Barrett and K. Meyers. *Foundations of Image Science*. Wiley Interscience, 2003.
- [8] H.H. Barrett and W. Swindell. *Radiological Imaging: The Theory of Image Formation, Detection, and Processing*. Academic Press, 1996.
- [9] C.E. Beni and O.P. Bruno. Noise-tolerant Fejér-mFBP reconstruction algorithm for positron emission tomography. *Submitted*.
- [10] R.B. Bird, W.E. Steward, and E.N. Lightfoot. *Transport Phenomena*. Wiley, 2001.
- [11] J. Boyd. A comparison of numerical algorithms for Fourier extension of the first, second, and third kinds. *J. Comp. Phys.*, 178:118–160, 2002.

- [12] J.P. Boyd and J.R. Ong. Exponentially-convergent strategies for defeating the Runge Phenomenon for the approximation of non-periodic functions, part I: single-interval schemes. *Commun. Comput. Phys.*, 5:484–497, 2009.
- [13] O. Bruno, Y. Han, and M. Pohlman. Accurate, high-order representation of complex three-dimensional surfaces via Fourier-continuation analysis. *J. Comp. Phys.*, 227:1094–1125, 2007.
- [14] O.P. Bruno. Fast, high-order, high-frequency integral methods for computational acoustics and electromagnetics. In M. Ainsworth, P. Davies, D. Duncan, P. Martin, and B. Rynne, editors, *Topics in Computational Wave Propagation Direct and Inverse Problems Series*, volume 31 of *Lecture Notes in Computational Science and Engineering*, pages 43–82. 2003.
- [15] O.P. Bruno and A. Prieto. Dispersionless and unconditionally stable FC-AD solvers for variable-coefficient PDEs. *In progress*.
- [16] O.P. Bruno and M. Lyon. High-order unconditionally stable FC-AD solvers for general smooth domains I. Basic elements. *J. Comp. Phys.*, 229:2009–2033, 2009.
- [17] C. Canuto, M. Hussaini, A. Quarteroni, and T. Zang. *Spectral Methods: Fundamentals in Single Domains*. Springer, 2006.
- [18] S. Earnshaw. On the nature of the molecular forces which regulate the constitution of the luminiferous ether. *Trans. Cambridge. Philos. Soc.*, 7:97–112, 1842.
- [19] K. Eckhoff. On a high order numerical method for solving Partial Differential Equations in complex geometries. *J. Sci. Comput.*, 12:119–138, 1997.
- [20] C.L. Epstein. *Mathematics of Medical Imaging*. Prentice Hall, 2003.
- [21] A.S. Fokas, A. Iserles, and V. Marinakis. Reconstruction algorithm for single photon emission computed tomography and its numerical implementation. *J. R. Soc. Interface*, 3:45, 2006.
- [22] A.S. Fokas and V. Marinakis. Reconstruction algorithm for the brain imaging techniques of PET and SPECT. *HERMIS*, 4:45–61, 2003.
- [23] R.L. Fournier. *Basic Transport Phenomena in Biomedical Engineering*. Taylor & Francis, 2007.
- [24] A. Gelb and J. Tanner. Robust reprojection methods for the resolution of the Gibbs phenomenon. *Appl. Comput. Harmon. Anal.*, 20:3–15, 2006.
- [25] D. Gottlieb and C.W. Shu. On the Gibbs phenomenon and its resolution. *SIAM Rev.*, 39:644–668, 1997.
- [26] A. Grief and G. Richardson. Mathematical modelling of magnetically targeted drug delivery. *J. Magn. Magn. Mater.*, 293:455–463, 2005.

- [27] H.M. Hudson and R.S. Larkin. Accelerated image reconstruction using ordered subsets of projection data. *IEEE Trans. Med. Imag.*, 4:4, 1994.
- [28] G.H. Iacoba, O. Rotariub, N.J. Strachanband, and U.O. Häfelic. Magnetizable needles and wires—modeling an efficient way to target magnetic microspheres in vivo. *Biorheology*, 41:599–612, 2004.
- [29] F.P. Incropera and D.P. DeWitt. *Fundamentals of Heat and Mass Transfer*. Wiley, 2001.
- [30] J.D. Jackson. *Classical Electrodynamics*. Wiley, 1998.
- [31] A.C. Kak and M. Slaney. *Principles of Computerized Tomographic Imaging*. IEEE Press, 1988.
- [32] F.W. King. *Hilbert Transforms*, volume 2. Cambridge: Cambridge University Press, 2009.
- [33] A. Komae and B. Shapiro. Magnetic steering of a distributed ferrofluid spot towards a deep target with minimal spreading. *In progress*.
- [34] L.A. Kunyansky. A new SPECT reconstruction algorithm based on the Novikov explicit inversion formula. *Inverse Problems.*, 17:293–306, 2001.
- [35] A.J. Lemke, M.I.S. von Pilsach, A. Lubbe, C. Bergemann, H. Riess, and R. Felix. MRI after magnetic drug targeting in patients with advanced solid malignant tumors. *European Radiology*, 14:1949–1955, 2004.
- [36] A.S. Lubbe, C. Alexiou, and C. Bergemann. Clinical applications of magnetic drug targeting. *Journal of Surgical Research*, 95:200–206, 2001.
- [37] A.S. Lubbe, C. Bergemann, J. Brock, and D.G. McClure. Physiological aspects in magnetic drug-targeting. *J. Magn. Magn. Mater.*, 194:149–155, 1999.
- [38] A.S. Lubbe, C. Bergemann, W. Huhnt, T. Fricke, H. Riess, and et al. J.W. Brock. Preclinical experiences with magnetic drug targeting: tolerance and efficacy. *Cancer Research*, 56:4694–4701, 1996.
- [39] A.S. Lubbe, C. Bergemann, H. Riess, F. Schriever, P. Reichardt, and et al. K. Possinger. Clinical experiences with magnetic drug targeting: a phase I study with 4-epidoxorubicin in 14 patients with advanced solid tumors,. *Cancer Research*, 56:46864693, 1996.
- [40] K. Mattson, M. Svård, and M. Shoeybi. Stable and accurate schemes for the compressible Navier-Stokes equations. *J. Comput. Phys.*, 227:2293–2316, 2008.
- [41] Y. Morimoto, M. Okumura, K. Sugibayashi, and Y. Kato. Biomedical applications of magnetic fluids. 2. Preparation and magnetic guidance of magnetic albumin microsphere for site specific drug delivery in vivo. *Journal of Pharmacobio-Dynamics*, 4:624631, 1981.

- [42] A. Nacev, C. Beni, O. Bruno, and B. Shapiro. The behaviors of ferro-magnetic nano-particles in and around blood vessels under applied magnetic fields. *J. Magn. Magn. Mater.*, 2010.
- [43] O. Næss and K. Eckhoff. A modified Fourier Galerkin method for the Poisson and Helmholtz equations. *J. Sci. Comput.*, 17:529–539, 2002.
- [44] F. Natterer. *The Mathematics of Computerized Tomography*. John Wiley & Sons Inc., 1986.
- [45] D.W. Peaceman and H.H. Rachford, Jr. The numerical solution of parabolic and elliptic differential equations. *J. Soc. Ind. Appl. Math*, 3:28–41, 1955.
- [46] M.E. Phelps. *PET: Physics, Implementation, and Scanners*. Springer, 2006.
- [47] H.E. Potts, R.K. Barrett, and D.A. Diver. Dynamics of freely-suspended drops. *Journal of Physics D: Applied Physics*, 34:2626–2636, 2001.
- [48] S. Reddy and L. Trefethen. Stability of the method of lines. *Numer. Math.*, 62:235–267, 1992.
- [49] A.J. Rosengart, M.D. Kaminski, H.T. Chen, P.L. Caviness, A.D. Ebner, and J.A. Ritter. Magnetizable implants and functionalized magnetic carriers: A novel approach for non invasive yet targeted drug delivery. *J. Magn. Magn. Mater.*, 293:633–638, 2005.
- [50] O. Rotariu and N.J.C. Strachan. Modeling magnetic carrier particle targeting in the tumor microvasculature for cancer treatment. *J. Magn. Magn. Mater. - Proceedings of the Fifth International Conference on Scientific and Clinical Applications of Magnetic Carriers*, 293:639–646, 2005.
- [51] B. Shapiro. Towards dynamic control of magnetic fields to focus magnetic carriers to targets deep inside the body. *J. Magn. Magn. Mater.*, 321:1594–1599, 2009.
- [52] L.A. Shepp and Y. Vardi. Maximum Likelihood Reconstruction for Emission Tomography. *IEEE Trans. Med. Imag.*, 1:113–122, 1982.
- [53] C. Simon. Magnetic drug targeting. New paths for the local concentration of drugs for head and neck cancer. *HNO*, 53:600–601, 2005.
- [54] E.M. Stein and R. Shakarchi. *Fourier Analysis: An Introduction*. Princeton University Press, 2003.
- [55] D.M. Young. *Iterative Solutions of Large Linear Systems*. Dover, 2003.
- [56] G. Zhao and Q. Liu. The unconditionally stable pseudospectral time-domain (PSTD) method. *IEEE Mircow. Wireless Comp. Lett.*, 13:475–477, 2003.
- [57] A. Zygmund. *Trigonometric Series*. Cambridge University Press, 2003.