

UNIVERSITY OF CALIFORNIA

Los Angeles

**Numerical Methods for Partial Differential
Equations Involving Discontinuities**

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Mathematics

by

Thomas Christopher Cecil

2003

© Copyright by
Thomas Christopher Cecil
2003

The dissertation of Thomas Christopher Cecil is approved.

Ann R. Karagozian

Luminita A. Vese

Christopher R. Anderson

Stanley J. Osher, Committee Chair

University of California, Los Angeles

2003

To Jill

TABLE OF CONTENTS

1	Introduction to this Thesis	1
1.1	Level Set Methods	2
1.2	General Numerical Tools	4
2	Numerical Methods for Hamilton Jacobi Equations Using Radial Basis Functions	7
2.1	Introduction	7
2.2	Function reconstruction using RBFs	9
2.3	Data Access in the Meshless Computational Domain	13
2.4	Monotone Fluxes	14
2.4.1	Introduction to Monotone Schemes	14
2.4.2	Derivation of Schemes	15
2.4.3	Implementation Details of Schemes	19
2.5	A Roe-Fix Scheme	21
2.6	High Order ENO Reconstruction	23
2.7	Outline of Evolution Procedure	26
2.8	Numerical Examples	27
2.9	Conclusion	35
3	Regularization of Certain Ill Posed Differential Equations	40
3.1	Introduction	40
3.2	PDE System	43

3.3	Numerical Methods	45
3.4	Numerical Experiments	47
3.5	Conclusion	50
3.6	Derivation of Euler-Lagrange Equations	50
4	Topology Preserving Level Set Motion	54
4.1	Introduction	54
4.2	A Discrete Method	54
4.3	A PDE Based Method	55
4.4	Numerical Implementation	60
4.5	Numerical Examples	62
4.6	Conclusion	66
5	Fast Numerical Solution to Poisson's Equation for Layered Semi-conductor Devices	70
5.1	Introduction to the Problem	70
5.2	High Order Potential Solver	72
5.2.1	Solution of the One Dimensional BVPs	73
5.2.2	Boundary Conditions	78
5.2.3	Accuracy	80
5.2.4	Implementation Details	81
5.3	Numerical Examples	81
5.4	Conclusion	87
	References	91

LIST OF FIGURES

2.1	2d concentric grid.	28
2.2	$H(p) = - p $	29
2.3	$H(u_x, u_y) = \sin(u_x + u_y)$	30
2.4	$H(u_x, u_y) = 0.5(u_x + u_y + 1)^2$	31
2.5	Points on smaller (top) and larger (bottom) concentric spheres.	32
2.6	$H(p) = - p $, multiple time views.	33
2.7	$H(p) = - p $	34
2.8	Dimension 1 fixed slice.	36
2.9	Dimension 3 fixed slice.	37
2.10	Dimension 4 fixed slice.	38
3.1	$\gamma(\theta) = 1 + \sin(2\theta) $	48
3.2	$\gamma(\theta) = -1$	48
3.3	$\gamma(\theta) = -1$, no regularization.	49
3.4	$\gamma(\theta) = -1$	49
4.1	Segmentation of 2 circles with topology preservation.	63
4.2	Segmentation of 2 circles without topology preservation.	63
4.3	Slit annulus segmentation with topology preservation.	64
4.4	Slit annulus segmentation without topology preservation.	64
4.5	Circle and line under outward normal motion with topology preservation.	65

4.6	Final Jacobian of circle and line under outward normal motion.	65
4.7	Circle and line under outward normal motion without topology preservation.	66
4.8	Hemispheres under outward normal motion with topology preservation.	67
4.9	Hemispheres under outward normal motion without topology preservation.	68
5.1	3d quantum wire and sample mesh.	82
5.2	Level sets of ψ in the upper and lower wells.	83
5.3	Potential level set.	83
5.4	1d potential slice for fixed y and z	84
5.5	Problem 1 computed and exact solution.	84
5.6	Problem 2 computed and exact solution.	85
5.7	Problem 2 computed and exact solution magnified.	86
5.8	Problem 3 exact solution.	87
5.9	Problem 3 Richardson extrapolation, 1^{st} order.	88
5.10	Problem 3 Richardson extrapolation, 2^{nd} order.	88
5.11	Problem 3 Richardson extrapolation, 3^{rd} order.	89
5.12	Problem 3, 3^{rd} order extrapolation with $f/5, p, q$	90

LIST OF TABLES

1.1	TVD RK coefficients.	6
2.1	Sample radial basis functions.	10

ACKNOWLEDGMENTS

I would like to thank the members of my committee and also Dr. Henry Huang who was on my advancement to candidacy committee. My advisors have been a great help to me and have always made time to answer my questions and talk about problems. This, combined with their friendly dispositions has made my time working with them very enjoyable. Their energetic attitude further inspired me always work harder at my studies. I would also like to thank Russel Caflisch for helping support me. Jianliang Qian is to be thanked as well, for together we worked closely on the meshless methods section.

I would also like to thank my fellow students Brian Cotta, Aidas Banaitas, Richard Tsai, and C.Y. Kao. They have made my time at UCLA much more enjoyable and prosperous. Finally, I would like to thank my family for being so supportive in all my schooling, and especially my wife, Jill, who has been very patient and understanding during the completion of this dissertation while also giving birth to our daughter, Solana, in May.

VITA

- 1975 Born, Poughkeepsie, New York, USA.
- 1997 B.A. (Mathematics and English), University of Notre Dame, Notre Dame, Indiana.
- 1997–1999 Teaching Assistant, Mathematics Department, University of California, Los Angeles.
- 1999 M.S. (Mathematics) University of California, Los Angeles.
- 1999–2000 Actuarial Associate, Tillinghast Towers Perrin, San Francisco, California.
- 2000–present Teaching and Research Assistant, Mathematics Department, University of California, Los Angeles.

ABSTRACT OF THE DISSERTATION

Numerical Methods for Partial Differential Equations Involving Discontinuities

by

Thomas Christopher Cecil

Doctor of Philosophy in Mathematics

University of California, Los Angeles, 2003

Professor Stanley J. Osher, Chair

This thesis introduces numerical techniques for solving PDEs involving discontinuities. These discontinuities arise in the form of jumps in solutions and their derivatives, stiff time scale discrepancies, and nonhomogeneous forcing terms. We begin with a new meshless method of resolving viscosity solutions of Hamilton-Jacobi equations. We then move on to two applications of the level set method, surface diffusion and topology preservation, formulating and numerically solving PDEs associated with these concepts. Finally, we discuss the numerical solution of a coupled elliptic/eigenvalue PDE system generated by the goal of constructing isolated quantum bits within a semiconductor device.

CHAPTER 1

Introduction to this Thesis

The underlying theme of this thesis is numerical methods for PDEs involving discontinuities in their solutions, the derivatives of these solutions, the time scales involved or the forcing terms. Care must be taken to resolve these discontinuities sufficiently and efficiently, without amplifying them so as to distort the approximation of the true solution. We encounter hyperbolic, parabolic and elliptic problems, each with its own difficulties and unique characteristics.

Before getting to the new material there is a short review of some of the underlying techniques which are used repeatedly throughout the text. As each section will have its own introduction and discussion of the current state of the art in that specific area, we will end this introduction by simply listing the topics to be covered.

The first section will present a new method of solving time dependent HJ equations on scattered, meshless data sets using radial basis function reconstruction. This numerical solution will converge to the nonclassical vanishing viscosity solution involving discontinuous derivatives. The second section proposes a variational formulation and subsequent minimization procedure of a surface energy involving curvature. In this section we attempt to overcome the inherent stiffness associated with the separation in time scales between the edge diffusion and edge advancement. We then proceed to introduce a method of topology preservation in the context of level set front evolution in the third section. There we must

attempt to resolve the large discontinuities of a delta function. Finally, we discuss a highly accurate numerical method for solving a coupled eigenvalue and potential PDE system in the context of semiconductor device modeling. In this problem we encounter localized forcing terms and jumps in diffusivity which must be resolved accurately.

1.1 Level Set Methods

The impetus for the development of the meshless Hamilton-Jacobi solver, as well as the setting for the topology preserving and edge diffusion sections, is the level set method. By this we mean the framework introduced by Osher and Sethian for implicitly tracking dynamic surfaces [OS88]. The basic idea is to embed an interface Γ , which lies in \mathbb{R}^{n-1} into a surface in dimension \mathbb{R}^n . This embedding is done by defining a suitable function φ such that $\{x|\varphi(x) = 0\} = \Gamma$. If the interface divides \mathbb{R}^n into multiple connected components then we can easily distinguish the inside of one component from its exterior by a change in sign of φ . If we are given a physical velocity v under which Γ moves then by differentiating $\varphi(t, x)$ with respect to t yields

$$\varphi_t + v \cdot \nabla \varphi = 0, \tag{1.1}$$

for the evolution of any level set of φ .

A key advantage of this method over other front tracking methods [ZYK01], [UT92], [TBJ01] involving marker particles on Γ is that topology changes are done "without emotional involvement", which is to say without any special treatment within the algorithm evolving the front. Γ is handled implicitly and only found explicitly when the user desires, which is usually done by a standard interpolation and plotting routine built into the visualization software. However, the ease with

which one can get results using the method have appealed to so many that now the need to prohibit topological changes "without emotional involvement" is being explored. This is a topic on which we will give some initial results.

Writing (1.1) in terms of motion that is strictly in the direction normal to Γ yields the equation

$$\varphi_t + v_N |\nabla \varphi| = 0 \tag{1.2}$$

where $v_N = v \cdot \frac{\nabla \varphi}{|\nabla \varphi|}$. This is a Hamilton-Jacobi equation whose characteristics follow

$$\frac{dx_i}{dt} = H_{p_i},$$

where $p_i = \varphi_{x_i}$ and $H = v_N |\nabla \varphi|$. Because of discontinuous derivatives, classical solutions to (1.2) do not always exist, so we would like to find numerical methods that converge to the unique vanishing viscosity solution [CEL84] that is physically relevant. On uniform Cartesian grids there has been much success in both creating and analyzing numerical methods solving (1.2). Building from the foundations of conservation laws (as H-J equations in one space dimension can be thought of as conservation laws integrated once) high order explicit shock capturing methods have successfully been employed [HOE86], [OS91] in the framework of monotone schemes [CM80], [CL84]. Another important advancement was the introduction of local level set (LLS) methods [PMO99] which evolved φ only within a narrow band near Γ , making the storage and speed comparable to locally defined solvers. With the ever expanding list of problems to which level sets being are applied there will always be someone pushing the computational limits using the current methods. Our meshless method will hopefully bring some relief to those computations that would benefit from escaping the restrictions of fixed width, fixed orientation Cartesian grids.

With the growing interest in higher order variational problems involving inte-

grals of curvature there is a demand to resolve the second order motions previously dismissed from interface dynamics. For these types of problems we introduce a surface diffusion approximation and solution technique that reduces the restrictive timestep normally encountered with these problems.

1.2 General Numerical Tools

Because of the emphasis on solving HJ equations in this thesis we will review some standard numerical tools used repeatedly when solving (1.2). The types of tools we will use are finite differences in space and Runge-Kutta explicit time evolution.

Spatial Derivatives When calculating $\nabla\varphi$ in (1.2) we cannot use standard finite differencing. The reason for this is that the discontinuities in φ_{x_i} will cause nonphysical oscillations. Thus we employ compactly stenciled upwinded differences, following the characteristic directions mentioned above. In order to avoid smoothing the solution unnecessarily we use high order essentially nonoscillatory (ENO) reconstructions for φ which are constructed by beginning with a small upwind stencil, and then increasing the stencil size incrementally, selectively using points which yield the smoothest solution [SO89]. Thus to calculate $\varphi_x(x_i)$ we begin with a 2 point upwind linear interpolant for $\varphi(x)$, and then compare the 2 possible 3 point stencils along the x direction that include the original stencil, choosing the smoothest one. This process is repeated until sufficient accuracy is reached, then the polynomial is differentiated and evaluated at x_i . A clear understanding of this process is helpful when reading the section on ENO methods for meshless evolutions later. Averages of the larger stencil reconstructions weighted by smoothness yield a weighted ENO (WENO) scheme [JP00].

In order to capture the viscosity solution we use a scheme known as Roe with entropy fix (RF), which is basically upwinding when we are not at sonic points, and a replacement of H by $H - D$, where D is an artificial viscosity term, when we are at sonic points. This allows the correct solution to be found without introducing extra diffusion [OS91].

In the section on topology preservation there is a PDE in conservation form which we solve using the conservation law methods [HOE86] on which the Hamilton-Jacobi ENO and RF schemes are based.

Time Derivatives We use total variation diminishing (TVD) Runge-Kutta (RK) methods for time advancement [SO89]. The procedure is as follows: Given a node x_i and function values at time t_n we define the operator

$$L_i = -dt\hat{H}(\varphi^n),$$

where \hat{H} is the numerical Hamiltonian. We then advance the solution using a Runge-Kutta procedure of the form

$$\varphi_i^{(k)} = \sum_{m=0}^{k-1} [\alpha_{km}\varphi_i^{(m)} + \beta_{km}L_i^{(m)}], \quad k = 1, \dots, r,$$

where $\varphi_i^{(0)} = \varphi_i^n$, $\varphi_i^{(r)} = \varphi_i^{n+1}$. If the forward Euler version (i.e. $r = 1, \alpha_{1,0} = 1, \beta_{1,0} = 1$) is TVD under the CFL condition

$$dt/dx \leq \lambda_0,$$

then the RK method can be proven to be TVD under the CFL condition

$$dt/dx \leq C_r \lambda_0.$$

Coefficients for the popular 2^{nd} and 3^{rd} order TVD Runge Kutta methods are shown in Table 1.1.

Order	α_{kl}		β_{kl}		C_r
2	1		1		1
	1/2	1/2	0	1/2	
3	1		1		1
	3/4	1/4	0	1/4	
	1/3	0	2/3	0 0	

Table 1.1: TVD RK coefficients.

Together the ENO and TVD-RK methods give highly accurate solutions and can be quickly adapted to almost any Hamiltonian H .

CHAPTER 2

Numerical Methods for Hamilton Jacobi Equations Using Radial Basis Functions

2.1 Introduction

In the numerical solution of time dependent conservation laws such as

$$u_t + \nabla \cdot f(u) = 0, \quad (2.1)$$

a method for solving the PDE is by dividing the spatial domain into grid cells and solving a small Riemann problem for each cell forward in time. For Hamilton-Jacobi equations of the form

$$\varphi_t + H(\nabla\varphi) = 0, \quad (2.2)$$

we can think of our problem as being a conservation law such as (2.1) in the variable $u = \varphi_x$ in one spatial dimension. In this way there is a direct link between conservation laws and HJ equations, with the solution to (2.1) being a derivative of the solution to (2.2). Although this analogy fails in multiple spatial dimensions it guides us towards the numerical methods of conservation laws when finding solutions to (2.2).

On uniform grids in any dimension [OS91] and [OS88] proposed extending the essentially nonoscillatory (ENO) schemes of [SO89], [HOE86] for conservation laws to HJ equations by reconstructing locally smooth polynomial interpolants

of φ in each individual spatial dimension, x_i , and then taking the derivative, φ_{x_i} of that interpolant for use in $H(\nabla\varphi)$. These methods have shown good results on uniform grids, avoiding the oscillations typically associated with high order methods in the presence of discontinuities.

On nonuniform grids in higher dimensions there has been work done on extending the ENO type of smooth polynomial interpolant reconstruction, see [Abg94], [Fri98], [ZS02]. These methods have shown some success, but have some drawbacks to them. For example, when using divided differences as approximations to the higher derivatives needed to obtain the Newton polynomial in 1D, we see that each time the polynomial is raised by one degree, we need one extra evaluation point. In 2D on an arbitrary triangulated grid there are no Newton divided differences to aid us in reconstruction. If we would like to use polynomial reconstruction we now have the added burden of needing at least $\frac{(n+1)(n+2)}{2}$ nodes to construct a degree n polynomial. In K dimensions we would need at least $\binom{n+K}{K}$ nodes, and even with this many nodes there may still be problems resulting from the ill conditioning of the linear system for the coefficients if the nodes are not well spaced [ZS02], [HS99],[Abg94]. Attempts have been made to rectify these problems, but multidimensional polynomial reconstruction is still far from being a "black box" procedure.

We will begin in section 2.2 by introducing radial basis function interpolation. We then move in section to a brief description of how we handle neighbor access on a meshless computational framework. Next we cover the construction of monotone schemes in section 2.4 including a convergence proof and implementation details. We follow this in section 2.5 by introducing a Roe with entropy fix scheme which minimizes artificial diffusion. In section 2.6 we describe spatial methods of achieving higher order accuracy. Finally, we give a summary of the

implementation procedure in section 2.7.

2.2 Function reconstruction using RBFs

Instead of using polynomial reconstruction for Φ , which has been used successfully in 1-D, we will use a type of multidimensional spline [Har71],[Son98]

$$\Phi(x) := \sum_{j=1}^M \gamma_j \phi(x - y_j) + \sum_{j=1}^Q \beta_j p_j(x) \quad (2.3)$$

where ϕ is the radial basis function (RBF), M is the number of cells in the reconstruction stencil, and the second sum is over polynomials $\{p_j\}$ which form a basis of the kernel of the seminorm $[\cdot, \cdot]$ of the *native* space in which ϕ lives [IS96]. In general a spline, Φ , in a semi-Hilbert space, V , interpolating data, $\{u_i\}$, satisfies $|\Phi|_V = \min_{u \in A} |u|_V$ where $A = \{v \in V | \langle \lambda_i, v \rangle = u_i\}$. So in this norm we are finding an optimal recovery function. The functions ϕ are assumed to have radial symmetry. Φ is forced to have the property that on a given stencil $\{x_i\}_{i=1:M}$,

$$\Phi(x_i) = u(x_i), \text{ and } \sum_{j=1}^M \gamma_j p_s(x_j) = 0, s = 1, \dots, Q. \quad (2.4)$$

So to find $\{\gamma_j\}_{j=1:M}$ we need to solve the linear system

$$A = \begin{bmatrix} M & N \\ N^t & 0_{(Q,Q)} \end{bmatrix} \begin{bmatrix} \gamma \\ \beta \end{bmatrix} = \begin{bmatrix} u \\ 0_{(Q,1)} \end{bmatrix}, \text{ where} \quad (2.5)$$

$$m_{i,j} = \phi(x_i - x_j), \quad n_{i,j} = p_j(x_i).$$

For HJ equations of the form (2.2) we need to calculate $\nabla \Phi(x)$, so we assume the RBF $\phi(x)$ is well behaved and differentiate (2.3).

The RBFs ϕ can be compactly or globally supported, and because we must solve (2.5) it is best if they are positive definite in some sense, implying unique

solvability of (2.5)[Mic86]. In [Mic86] it was shown that there is a direct relationship between ϕ being positive definite and the function $t \rightarrow \phi(\sqrt{t})$ being completely monotone, i.e. $f(t) := \phi(\sqrt{t})$ is smooth and satisfies

$$(-1)^m f^{(m)}(t) \geq 0, m \in \mathbb{N}_0, t > 0.$$

If a function is completely monotone, then it is positive definite, thus it does not need to be augmented by any polynomials in (2.3). However, if we only have that $(-1)^k f^{(k)}(t)$ is completely monotone for some $k > 0$ then ϕ is said to be conditionally positive definite of order k , and requires augmentation by polynomials of degree $k - 1$. Using these types of tools analysts have proven the conditional positivity of many RBFs over the years, and so there are numerous ϕ from which to choose. Table 2.1 shows some useful RBFs and their positive definite order, k .

RBF	$\phi(r)$	k
polynomials	$r^\beta, \beta > 0, \beta \notin 2\mathbb{N}$	$k > \beta/2$
thin plate splines	$r^{2\beta} \log r, \beta \in \mathbb{N}$	$k > \beta$
Gaussians	$e^{-\alpha r^2}, \alpha > 0$	$k \geq 0$
multiquadrics	$(c^2 + r^2)^{\beta/2}, \beta > 0, \beta \notin 2\mathbb{N}$	$k > \beta/2$
inverse multiquadrics	$(c^2 + r^2)^{\beta/2}, \beta < 0$	$k \geq 0$

Table 2.1: Sample radial basis functions.

Wu has also constructed a family of positive definite, compactly supported RBFs [Wu95], [Son98]. All of these functions can be scaled by taking $r \rightarrow \frac{r}{\theta}$ with θ problem dependent. Because of its radial construction, if a basis function ϕ can be used in n dimensions then it can be used just as well in any dimension less than n . This allows for algorithms and theory to be developed and tested in low dimensions with easy extension to problems in higher dimensions.

For our tests so far we have used the positive definite Gaussian and inverse multiquadric RBFs of the form

$$e^{-\alpha r^2} \text{ and } (r^2 + \alpha)^{-1/2},$$

respectively. These do not require augmentation by polynomials when solving (2.3).

In order to achieve better reconstructions we will attempt to optimize the parameter (e.g. α in $e^{-\alpha r^2}$) on each stencil. It has been shown that the accuracy of a RBF interpolant is inversely related to the condition number of the linear system in (2.5)[Sch95]. Our optimization consists of choosing a maximum acceptable condition number, κ_{max} , and performing an iterative method to determine the value of α that yields $0 < \kappa_{max} - \kappa_{\alpha} < \epsilon$ for some tolerance ϵ . This need only be done prior to evolution and allows for optimization on different parts of the domain where mesh spacing may vary greatly. In the case of ENO interpolations it is not feasible to test all possible stencils and store the optimal α on each, so we only store a single α that will be acceptable for all the local stencils near a given data point. As long as the mesh is well behaved this α should work adequately on local stencils.

It should be noted that the procedure for choosing an optimal RBF parameter is an area of current research. There are other ways to optimize the parameter of the radial basis function [CF91], [Rip99], [Har71], and any of these can be incorporated into our framework.

Example of Location Dependent RBF Parameter Optimization Let us take an example to show both the accuracy and method of parameter optimization. To keep things simple we will work on a 2 point stencil, $\{a, b\}$ in 1d. Obviously the best interpolation of our function, $f(x)$, that we can hope for is

linear. In this case the best approximation to $f'(x)$ is

$$\frac{f(b) - f(a)}{b - a},$$

or 2 point finite differencing. Without loss of generality let $a = 0$ and $f(a) = 0$. Using a general RBF interpolation with basis function $\phi(r)$ we would like to find a condition on its parameter α such that for our RBF approximation to $f'(a)$, called app , satisfies

$$app = (\phi_x(0) \ \phi_x(-b)) \begin{pmatrix} \phi(0) & -\phi(-b) \\ -\phi(-b) & \phi(0) \end{pmatrix} / (1 - \phi(-b)^2) \begin{pmatrix} 0 \\ f(b) \end{pmatrix} = \frac{f(b)}{b},$$

using

$$\begin{pmatrix} \phi(0) & -\phi(-b) \\ -\phi(-b) & \phi(0) \end{pmatrix} / (1 - \phi(-b)^2) = A^{-1} \text{ from (2.5),}$$

and differentiating (2.3). By multiplying we see that if we assume $\phi_x(0) = 0$, which it should for all smooth RBF basis functions, and scale ϕ so that $\phi(0) = 1$, then we have

$$app = \frac{-y'y}{1 - y^2} f(b), \tag{2.6}$$

where $y' = \phi_x(-b)$ and $y = \phi(-b)$. Solving the ODE

$$\frac{-y'y}{1 - y^2} = \frac{1}{b} \tag{2.7}$$

will give solutions $y(b) = \phi(b)$ that yield an equivalent RBF interpolation to 2 point finite differencing. If we let $\phi(b) = e^{-\alpha b^2}$ and let $\alpha \rightarrow 0$, then we have a solution to (2.7). If we would like $\phi(r) = \alpha r^2$, then we need

$$\alpha = \sqrt{\frac{1}{3b^4}}.$$

Other basis functions will have different restrictions on α .

Note that for the Gaussian α does not depend on b , which makes its optimization straightforward. In practice if we wanted to optimize α when $\phi = e^{-\alpha r^2}$

all we would need to do is let $\alpha \rightarrow 0$ until we find that $|\kappa_{max} - \kappa_\alpha| < \epsilon$. Even for higher order approximations (with more nodes in the stencil) it can be shown that letting $\alpha \rightarrow 0$ with $\phi = e^{-\alpha r^2}$ approaches the optimal solution.

For other ϕ whose optimal parameter may depend on the data locations, (e.g. b above) we run a root finding method on the equation $\kappa_{max} - \kappa_\alpha = 0$ and iterate in α until $|\kappa_{max} - \kappa_\alpha| < \epsilon$. For many basis functions there exist a priori local error estimates in terms of α which clearly indicate limiting values of α for which to strive during optimization [Sch95], [Sch97].

We also note that a given $\Phi(x)$ can be represented in a Lagrange type fashion

$$\Phi(x) := \sum_{j=1}^M \Psi_j(x)u_j. \quad (2.8)$$

We can find the coefficient $\Psi_i(x_0)$ in (2.8) by setting $u_j = \delta_{i,j}$ for $j = 1, \dots, M$ and solving

$$\Phi(x_0) = \sum_{j=1}^M \Psi_j(x_0)u_j = \Psi_i(x_0)u_i. \quad (2.9)$$

2.3 Data Access in the Meshless Computational Domain

In this section we present the way in which we give some structure to our unstructured data set, allowing access of nodes in a reasonable amount of time. If our data set, X , were to be stored as simply an unordered list of points, then each time we needed to access a neighbor of a given node, x_i , we would need to search through the entire list giving a $O(N)$ algorithm versus $O(1)$ on a uniform grid. Although the storage required for this method is minimal, the access time is much too slow for practical use.

Instead, we use a binning method. This method divides the entire domain, $\Omega \supset X$, into a coarse, structured grid C . Then for each coarse gridcell $c_j \in C$ we

create a list of all the nodes of X that lie inside c_j . When a neighbor of $x_i \in c_j$ needs to be accessed we only need to search the lists of the coarse neighbors of c_j . So the total neighbor access time is $O(1)$ to access the coarse neighbor list times $O(\text{list}_j)$ to search the list and find the neighbor. Of course this procedure can be iterated over multiple coarse levels so that the list sizes are smaller, and other optimizations can be done such as noting which coarse cells are nonempty. Similar ideas have been explored in the context of local level set methods [Str99].

For the evolution procedure we can find all appropriate stencils prior to time evolution if we would like, and then the problem of neighbor access is only relevant in the preprocessing step and does not slow the evolution down.

2.4 Monotone Fluxes

2.4.1 Introduction to Monotone Schemes

In solving equations of the form (2.2) an important class of numerical methods are monotone schemes [CL84]. When they are also consistent these schemes have been shown to converge to the physically correct viscosity solution of (2.2).

For uniform data in 1D there are numerous schemes available [OS91], and these schemes can be generalized for uniform data in higher dimensions. In 2D on triangulated data there has been progress as well [Abg96]. However, in higher dimensions there has not been as much progress for scattered data. One drawback is that as the dimension grows, the triangulation becomes very complex and storage consuming ($O(M^{\lceil d/2 \rceil})$ simplices for M points in d dimensions), and the number of neighbors of a given node grows very large.

In this section we will present some new monotone schemes for scattered data in an arbitrary dimension that is not required to be triangulated. We will also

give a convergence result and discuss some details on implementation.

2.4.2 Derivation of Schemes

Given a Hamilton-Jacobi equation of the form (2.2), i.e.

$$\varphi_t + H(\nabla\varphi) = 0 \tag{2.10}$$

to be solved on a point set, we would like to derive a first order in time monotone scheme. We will use 2 dimensions for simplicity here, and let i and j be 2d multi-indices. The scheme will be of the form

$$\varphi_i^{n+1} = \varphi_i^n - dt \hat{H}_i(\varphi^n) \tag{2.11}$$

where φ_i^n is the numerical approximation to the solution of (2.2) at $(t = t_n, x = x_{i_1}, y = y_{i_2})$, and \hat{H}_i is the numerical Hamiltonian there. The requirement for a method to be monotone [CL84] is that

$$u_i^n \geq v_i^n \forall i, \Rightarrow u_i^{n+1} \geq v_i^{n+1} \forall i.$$

For our scheme of the form (2.11) this means that if we fix an index i_0 , then at x_{i_0}

$$\frac{\partial \hat{H}_{i_0}}{\partial \varphi_j} \leq 0 \text{ and } 0 \leq dt \leq \left(\frac{\partial \hat{H}_{i_0}}{\partial \varphi_{i_0}} \right)^{-1}, \forall j \neq i_0. \tag{2.12}$$

Thus our goal will be to find a numerical Hamiltonian satisfying (2.12).

Guided by the fact that some of the standard monotone schemes on uniform grids, such as Lax-Friedrichs, are approximations to solving the vanishing viscosity equation

$$u_t + H(\nabla u) = \epsilon \Delta u \text{ as } \epsilon \rightarrow 0,$$

we will construct our numerical Hamiltonian as an approximation of

$$H(\nabla u) - \epsilon \Delta u.$$

The procedure will be to reconstruct u near a given point, x_i , using an interpolation method and then differentiate the interpolant to get ∇u and Δu . The interpolation method we will use is radial basis function reconstruction.

Once we have derived the conditions on our RBF reconstruction that guarantee monotonicity we will turn to the issues of consistency and convergence.

Monotonicity If the basic time evolution procedure at node i can be written as

$$u_i^{n+1} = u_i^n - dt \{ H(\nabla u^n) - \epsilon_i \Delta u^n \} = u_i^n - dt G_i(u_{j1}, \dots, u_{Nsten}), \quad (2.13)$$

where $Nsten$ is the number of nodes used in the stencil approximating ∇u^n and Δu^n , then we will need to find an ϵ_i that satisfies all the inequalities in (2.12). Thus for each node i we should be able to calculate a minimal diffusion constant ϵ_i that guarantees monotonicity there. If we decide to evolve our solution using (2.13) with a unique ϵ_i at each node i then the method will be called a Local Lax-Friedrichs scheme. If we decide to take $\epsilon_{max} = \max_i \epsilon_i$ and evolve (2.13) using $\epsilon_i = \epsilon_{max} \forall i$, then the scheme will be called simply Lax-Friedrichs.

To find the appropriate size of ϵ_i we begin by writing our reconstructed partial derivatives in the Lagrange form (2.8). If we know that

$$u_{x_k}(x_i) \approx \sum_{j=1}^{Nsten} c_{k,j} u_j \quad \text{and} \quad \Delta u(x_i) \approx \sum_{j=1}^{Nsten} d_j u_j$$

then

$$\frac{\partial G_i}{\partial u_j} = \nabla H(Z) \cdot (c_{1,j}, \dots, c_{Ndim,j}) - \epsilon_i d_j$$

where $Z_k \in [\min(\nabla u)_k, \max(\nabla u)_k]$.

Note that in the construction of our stencil if we do not have that

$$d_j > 0 \forall j \neq i, \quad \text{and} \quad d_i < 0, \quad (2.14)$$

then we have a bad stencil which cannot yield a monotone scheme. In practice we are much more restrictive than just enforcing (2.14), because it can allow arbitrarily large diffusion terms which smear our solution. So we make sure to enforce restrictions on the relative sizes of c_j and d_j to keep diffusion to a minimum. Details of how this is done will be presented later.

We will now construct an ϵ_1 that satisfies the first $N_{sten} - 1$ inequalities in (2.12), and an ϵ_2 that satisfies the dt inequality, then finally set $\epsilon_i = \max(\epsilon_1, \epsilon_2)$. Thus we need

$$\left(\sum_{k=1}^{Ndim} \max_{x \in \Omega} |H_k(x)| |c_{k,j}| \right) - \epsilon_1 d_j \leq 0, \quad j \neq i.$$

The ϵ_1 for the scheme at x_i satisfies

$$\max_{j \neq i} \left[\frac{\sum_{k=1}^{Ndim} \max_x |H_k(x)| |c_{k,j}|}{d_j} \right] \leq \epsilon_1, \quad (2.15)$$

so to minimize the viscosity we choose ϵ_1 to satisfy the equality in (2.15).

Next we find ϵ_2 that satisfies

$$0 \leq dt \leq \left(\frac{\partial G_i}{\partial u_i} \right)^{-1} \quad (2.16)$$

where

$$\frac{\partial G_i}{\partial u_i} = \nabla H(Z) \cdot (c_{1,i}, \dots, c_{Ndim,i}) - \epsilon_2 d_i.$$

So we need that ϵ_2 satisfies

$$\left[\frac{\sum_{k=1}^{Ndim} \max_x |H_k(x)| |c_{k,i}|}{-d_i} \right] \leq \epsilon_2, \quad (2.17)$$

as $d_i < 0$. As this must hold for all x we choose ϵ_2 to satisfy the equality in (2.17), and finally choose $\epsilon_i = \max(\epsilon_1, \epsilon_2)$.

The CFL condition is then given by

$$dt \leq \frac{1}{\sum_{k=1}^{Ndim} \max_x |H_k(x)| |c_{k,i}| - \epsilon_i d_i}.$$

Consistency and Convergence The consistency of schemes using RBF interpolants has not been fully explored as of yet. There is large amounts of research demonstrating the error bounds of RBF interpolants and their convergence properties, but the strict definition of consistency where we require that

$$\hat{H}(p) = H(p) \text{ if } u(x) \equiv p \cdot x + c_0$$

has not been proven for general basis functions ϕ . However, it is only machine precision that limits us from getting an interpolant, Φ such that

$$|\nabla\Phi - p| < \delta_1 \text{ and } |\Delta\Phi - 0| < \delta_2 \quad (2.18)$$

for arbitrarily small δ_1, δ_2 , given an underlying linear function $u(x) \equiv p \cdot x + c_0$. Therefore our goal of finding a consistent scheme using RBFs may not be attainable, but that does not stop us from proving that our method converges.

All that is needed to be done is follow the proof in [CL84] where the authors derive an estimate of the form

$$|\varphi_i^n - V(x_i, t_n)| \leq c(\sqrt{dt}), \quad (2.19)$$

where V is the exact viscosity solution. At the point where the consistency of \hat{H} is used in their proof we substitute a modified inequality. Namely, in [CL84] in order to derive the estimate (2.19) the authors use a Taylor expansion to get

$$\hat{H}(u_{i,P}^n) \leq H(\nabla u(x_i, t_n)) + C(\sqrt{dt}), \quad (2.20)$$

where $u_{i,P}^n$ is the numerical approximation of $u(x_i, t_n)$ when a linear polynomial reconstruction of u is differentiated, i.e. 2 point finite differencing. [In their proof u is a multiple of a specifically chosen smooth cutoff function which is called β_ϵ .]

Given that for a linear reconstruction u_P we can approximate ∇u_P and Δu_P as in (2.18), then we can attain a convergence proof when δ_1, δ_2 are small enough.

In our case we use a triangle inequality of the form

$$|\hat{H}(u_{i,RBF}^n) - H(\nabla u(x_i, t_n))| \leq |\hat{H}(u_{i,RBF}^n - \hat{H}(u_{i,P}^n)| + |\hat{H}(u_{i,P}^n) - H(\nabla u(x_i, t_n))|,$$

where $u_{i,RBF}^n$ is the RBF reconstruction at $u(x_i, t_n)$. This yields

$$|\hat{H}(u_{i,RBF}^n) - H(\nabla u(x_i, t_n))| \leq [\|\nabla H\|_\infty \delta_1 + \epsilon_i \delta_2] + C(\sqrt{dt}),$$

where the last term is the identical estimate as in (2.20). If

$$\|\nabla H\|_\infty \delta_1 + \epsilon_i \delta_2 \leq C_1(\sqrt{dt}), \quad (2.21)$$

then this allows us to replace (2.20) with

$$\hat{H}(u_{i,RBF}^n) \leq H(\nabla u(x_i, t_n)) + (C + C_1)(\sqrt{dt}), \quad (2.22)$$

yielding a convergence result analogous to (2.19). Here, as in [CL84], the locally Lipschitz property of H must be used.

2.4.3 Implementation Details of Schemes

We can see that there are a few inequalities which must be satisfied by our RBF reconstruction at a point x_i before our scheme is deemed convergent. Firstly, we must satisfy (2.14). Once this is done it is straightforward to calculate the diffusion terms ϵ_i and CFL condition. Also, we must make sure (2.21) holds as well. In practice it is (2.14) which is more difficult to satisfy using meshless methods because the good interpolation properties of RBFs will yield estimates of the form (2.21) without much problem.

Thus we explain how we find a stencil at x_i that satisfies both (2.14) and (2.21). Actually we will tighten the restrictions on (2.14) significantly, as it allows for diffusion terms that are too large. What we require is that (2.14)

holds, but also that $\epsilon_i < \epsilon_{max}$ for a specified ϵ_{max} which is usually dependent on the spacing of the local mesh. Generally what this requires is that for any Lagrange coefficient, d_j , of φ_j in the approximation to $\Delta\varphi$ has magnitude $\approx c_j dx$ as it would if we were using finite difference approximations.

Once we have decided on the bounds for ϵ and d we can begin searching for acceptable stencils at a given node, x_i . The problem that usually arises is that for a given candidate stencil, S_C , one or more of the coefficients d_j of the $\Delta\varphi$ approximation are too small in magnitude or the wrong sign because x_j it is either colinear or almost colinear with another node in S_C . Thus we will try to make our stencil as isotropic as possible. To do this we will decide on a stencil size, $N + 1$ (the stencil will always include the node x_i), define N equispaced rays, $x_i + v_k t$, $t > 0$ emanating from x_i and find the neighbor x_j of x_i that maximizes

$$\frac{x_j - x_i}{\|x_j - x_i\|} \cdot \frac{v_k}{\|v_k\|}. \quad (2.23)$$

For example in 2D the vectors v_k are chosen as

$$(\cos, \sin)(\theta_0 + 2\pi k/N) \text{ for } k = 0 : N - 1.$$

We search over a few different orientations (θ_0 in 2D) of the axes for a fixed stencil size, and stop when we find an acceptable stencil that satisfies our conditions on ϵ and d . As noted in section 2.2 we optimize the RBF interpolation on each candidate stencil by trying to find the best function parameter for the basis function. If none of the candidates satisfy our bounds for ϵ and d then we change the stencil size and repeat the search until we find an acceptable stencil. If we cannot find an acceptable stencil then our mesh is very bad and we must use the best of the candidate stencils we have examined. However, this has not yet occurred our computations.

The only ambiguous point in the description above is the definition of a

neighbor of x_i . Unless a triangulation of the data is constructed we do not have a rigorous definition of what a neighbor is. One method is to search through the coarse cells near x_i for the M closest points, where M is arbitrary but on the order of the stencil size. The neighbors of x_i are then said to be these M closest points. However, this may not work well for stencils with large discrepancies in distances between nodes near x_i . Therefore it is possible to adjust the stencil choosing algorithm so that given an axis v_k as described above we maximize $f(x_j - x_i, v_k)$ instead of (2.23), where f could penalize $\|x_j - x_i\|$ and perhaps place increased weight on the value obtained in (2.23), which we can call v , such as taking $f \approx v^2$ as $\|v\| \leq 1$. This concept of defining and finding a neighbor of a node on a meshless grid without creating a triangulation warrants further research.

Another option is to create a local triangulation of the M closest points to x_i and use this to define neighbors. This method will take more time, but will in general give a smaller number neighbors and more compact candidate stencils. As long as this local triangulation is not stored permanently this method is acceptable. It is when triangulations of large data sets in high dimensions must be stored that we exhaust memory restrictions.

Again it should be noted that this search for acceptable stencils need only be done prior to evolution if we are willing to store the nodes of the stencil at x_i .

2.5 A Roe-Fix Scheme

Given that we are able to construct a prototypical Lax-Friedrichs scheme, we are tempted to push further and find a monotone scheme with even less diffusion. For uniform grids in 1d and even triangulated grids in higher dimensions there are

upwind schemes which can be proved to be monotone [BS98]. These are based specifically on linear reconstructions (standard 2 point upwinding in 1d). When upwinding is used at all non-sonic points, combined with a vanishing viscosity approximation such as LF or LLF at sonic points we have a method known as Roe-Fix or RF. This would be readily implementable were we to have a definition of upwinding that applies to our RBF reconstructions, but unfortunately we do not. However, we can construct a RF method using RBF interpolation and make an argument as to its convergence properties.

If we are advancing the solution at a node x_i the first thing we must determine is whether or not we are at a sonic point. Assume we have constructed a suitable stencil S_i at x_i , adhering to the constraints of Section 2.4.2. At each node $x_j \in S_i$ we calculate $\nabla H(\nabla\varphi(x_j))$ using the stencil S_j . If $\frac{\partial H}{\partial\varphi_{x_k}} \equiv H_k$ changes sign for any $k = 1, \dots, Ndim$ when searching over $j = 1, \dots, |S_i|$ then we are at a sonic point and we advance the solution using either LF or LLF schemes. If H_k does not change sign then we are not at a sonic point, so we would like to use upwinding. Since we do not have a triangulation of the nodes surrounding x_i we cannot choose a triangle, T_c , from which the characteristics are flowing and then use the nodes of T_c to linearly reconstruct the function yielding a monotone, upwind scheme. However, as long as the nodes of S_i surround x_i sufficiently we have encompassed the domain of dependence for φ_i^{n+1} , assuming the CFL condition is small enough. Here, surrounding x_i means that the convex hull of S_i contains x_i . Thus, if φ is smooth near x_i then our RBF reconstruction can be interpreted as a higher order reconstruction extended from the linear interpolant φ_L on T_c . This reconstruction for φ should then only differ from φ_L by terms of order $O(dx^p)$ where $p \geq 2$. Using this interpolant for φ and dropping the artificial diffusion terms of the LF and LLF schemes should then give us a scheme that differs from a monotone scheme by $O(dt dx)$, similar to the argument given in [OS91] for high

order ENO schemes. Because of the lack of the artificial diffusion term we should see better resolution.

Note that we require φ to be smooth near x_i for this argument to be valid. This is usually the case when we are far away from sonic points, but will not be the case when we are at a moving kink, which is a moving discontinuity in 1st derivatives. In that case the RBF interpolant may extend over the discontinuity and differ from φ_L significantly. So our scheme may differ from a monotone scheme by more than $O(dtdx)$ there. Contact discontinuities have presented difficulties to numerical methods before, so this is not a surprise. If a monotone method is desired at points of this type then we can easily insert a check into our algorithm such that when the jump in derivatives of φ near x_i is too large it will trigger the use of LF or LLF schemes even if the signs of H_k indicate upwinding.

2.6 High Order ENO Reconstruction

While the ability of monotone schemes to correctly converge to the viscosity solution of Hamilton-Jacobi equations makes them desirable, they do have an undesirable property: they are at most first order accurate [HHL76],[CM80]. In one dimension on uniform grids this drawback is overcome by taking ENO polynomial function reconstructions that avoid using interpolants which cross discontinuities, causing spurious oscillations. The familiarity of polynomials and the ability to simply construct their derivatives using divided differences made the ENO methods for conservation laws and Hamilton-Jacobi equations very popular [HOE86],[OS91],[SO89]. In multiple dimension on nonuniform grids there has been some progress using polynomials [AA00],[ZS02], and RBFs [IS96]. Here we present an incremental stencil selection method which exploits the LU factorization of the RBF coefficient matrix. We also introduce a self-similar smoothness

indicator that allows the ENO stencil to be chosen.

Our ENO reconstruction will involve extending an existing reconstruction in a smooth fashion. Assuming we have a reconstruction $\Phi_M(x)$ on an existing M point stencil $S_M = \cup_{i=1}^M \{P_i\}$, where $P_i \in X$, that has been constructed starting with $S_r \ni x_k = P_1$ for $r \leq M$, we want to extend it to $M + 1$ nodes in an ENO fashion.

There are two pieces of information we need which are somewhat arbitrary. One is the choice of nodes from which to choose P_{M+1} , and the other is the measure of smoothness of our reconstructed function, $\Phi_{M+1}(x)$. A suggestion for choosing the candidates $\{P_{M+1}^{cand_j}\}$ for P_{M+1} is that we choose the N (again arbitrary, but finite) nodes $P_{M+1}^{cand_j} \in X$ that make the center of gravity of the stencil $S_M \cup P_{M+1}^{cand_j}$ closest to x_k , or closest to the center of gravity of S_M . The number of choices we have for candidates depends on the surface area of the existing stencil, S_M , so in higher dimensions the potential cost incurred by maximizing $card(\{P_{M+1}^{cand_j}\})$ becomes prohibitive. There are many strategies to extend stencils [ZS02], [AA00], [HC91], [Son96], however, these are usually based on polynomial reconstruction and take steps so as to ensure the interpolation coefficient matrix has a good condition number. For RBF reconstruction the condition number depends on ϕ and the stencil, and in practice we have not found any problems with it. Since ϕ is radially symmetric there should not be any directional bias which causes polynomials to have badly conditioned coefficient matrices, see [Abg94] for details about this problem.

For the measure of smoothness we use the self-similar indicator

$$\beta = \sum_{2 \leq |\alpha| \leq s} \int_{P_k} |P_k|^{\frac{2|\alpha|-N}{N}} (D^\alpha \Phi(x))^2 dx, \quad (2.24)$$

where $|P_k|^{\frac{2|\alpha|-N}{N}}$ makes β invariant under grid scaling in N dimensions when $|P_k|$

is the area of a grid cell containing P_k , α is a multi-index, and s is proportional to the size of the stencil [ZS02]. For polynomial interpolants, we can take $s =$ the order of the interpolant, but with RBFs Φ can be a weighted average of C^∞ functions, therefore we take s proportional to stencil size because we cannot expect that $D^\alpha u$ is influencing $D^\alpha \Phi$ for derivatives of order $\geq |\alpha|$ if we are using far fewer than $|\alpha|$ points.

Once we have made the above decisions we can proceed systematically to obtain the $M + 1^{st}$ stencil. When constructing the M^{th} stencil it was necessary to solve the system of equations (2.4), that we will write $A\gamma = \bar{u}$ which is usually small enough to be done by Gaussian Elimination/LU factorization. Noting that A is symmetric we have an $M \times M$ LL^t factorization (here we assume that $Q = 0$ in (2.3)). For the $M + 1^{st}$ cell we must solve a new $A\gamma = \bar{u}$ that can be written as

$$A = \begin{bmatrix} LL^t & \alpha \\ \alpha^t & d \end{bmatrix} = \begin{bmatrix} \bar{u} \\ \bar{u}_{M+1} \end{bmatrix},$$

where α and d are found as in (2.5). To obtain L_{M+1} we compute the Schur complement $S = d - \alpha^t(LL^t)^{-1}\alpha$ of LL^t and get

$$L_{M+1} = \begin{bmatrix} L & 0_{(M,1)} \\ \alpha^t(L^t)^{-1} & \sqrt{S} \end{bmatrix}.$$

Having the new L_{M+1} we can find the new set of $\{\gamma_j\}$ and compute β for each candidate stencil. The stencil with the smallest β is chosen as the $M + 1^{st}$ stencil.

We note that in practice all L 's can be found and stored before the time evolution begins as long as data point set doesn't change during the calculation. Thus, the above Schur complement procedure does not save as much time as when it is applied to an adaptive mesh, where the linear system inversions must

be done at each timestep. However, for large fixed data sets in higher dimensions the storage of the L matrices can become too large to be practical, and any acceleration to the matrix inversion procedure is helpful.

2.7 Outline of Evolution Procedure

In this section we will outline the procedure for solving (2.2) given initial values $\varphi_0(x_j)$ on a dataset $X = \{x_j\}$ of points contained in the computational domain, Ω .

1. Construct a coarse mesh C over Ω and for each coarse gridcell $c_i \in C$ create a list of all the nodes of X that lie within in c_i . If C is uniform then this should take $O(|X|)$ time. An iterated coarse mesh can also be constructed or any other mechanism which allows the user to determine the $M(\ll |X|)$ closest points to a given node in less than $O(|X|)$ time.
2. For each $x_i \in X$
 - do
 - Construct a new candidate stencil S_c , using the guidelines of section 2.4.3.
 - Optimize RBF parameter α on S_c .
 - Determine if d_j 's and ϵ are acceptable for S_c .

while(d_j 's and ϵ are unacceptable).

Set the chosen stencil $S_i = S_c$.

It is actually a matter of memory versus time as to what the user stores here. If memory is abundant and its access is fast then for each stencil

all the Lagrange coefficients (c_j 's, d_j 's) and ϵ_i can be stored, making the evolution procedure faster. If memory is scarce then just the nodes of the stencil and the optimal RBF parameters α should be stored.

3. For $t_n = 0 : T$,

do

- Compute $\nabla\varphi$ at all nodes using stencil from step 2.
- If higher order accuracy of $\nabla\varphi$ is desired then use an ENO reconstruction for φ as described in section 2.6.
- For each node, if using RF scheme determine if sonic fix is necessary using ∇H . If so, or if the scheme is LF or LLF, then compute $\Delta\varphi$ and diffusion weight ϵ .
- For each node, advance solution one step in time using
$$\varphi^{n+1} = \alpha\varphi^n - \beta dt \hat{H}(\varphi^n).$$
 If RK method is being used then go to beginning of this do loop as many times as appropriate.

end do loop.

If adaptive grid is being used repeat stencil finding procedure in step 2, otherwise go to the beginning of this for loop.

2.8 Numerical Examples

Unless otherwise noted the examples are calculated on a domain of $[-1, 1]^d$ in d dimensions.

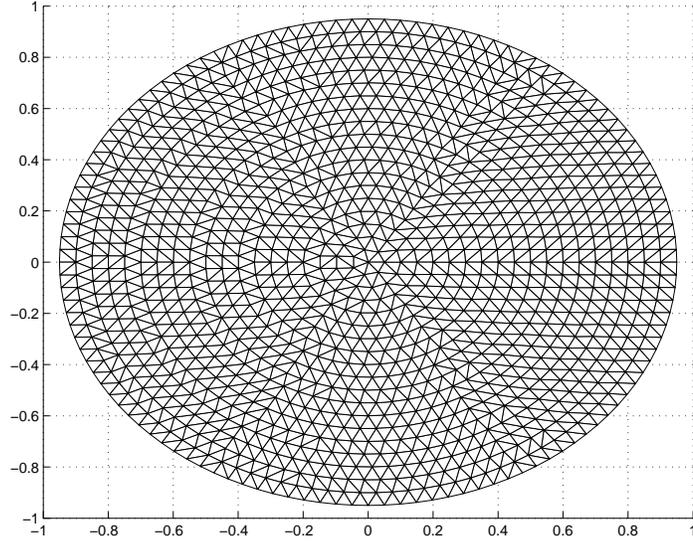


Figure 2.1: 2d concentric grid.

We begin with a level set evolution of the form

$$\varphi_t - |\nabla\varphi| = 0,$$

calculated on a grid of points that lie on concentric circles as in figure 2.1. The nodes used lie at the vertices of the triangulation shown. Note that this triangulation is not necessary for our calculation and is only used in the visualization. As the characteristics flow outward we use "upwind" reconstruction stencils at the boundary consisting of nodes within the domain. Figure 2.2 shows how our method captures the vanishing viscosity solution.

Figure 2.3 shows the solution of

$$u_t + \sin(u_x + u_y) = 0,$$

calculated on a uniform grid, but using our meshless method. Periodic BCs are imposed in both directions.

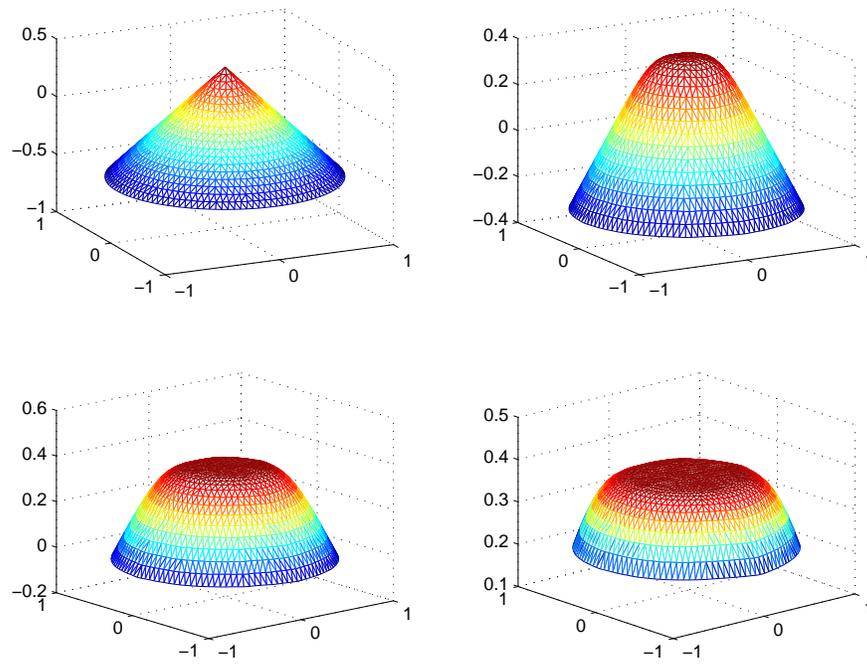


Figure 2.2: $H(p) = -|p|$.

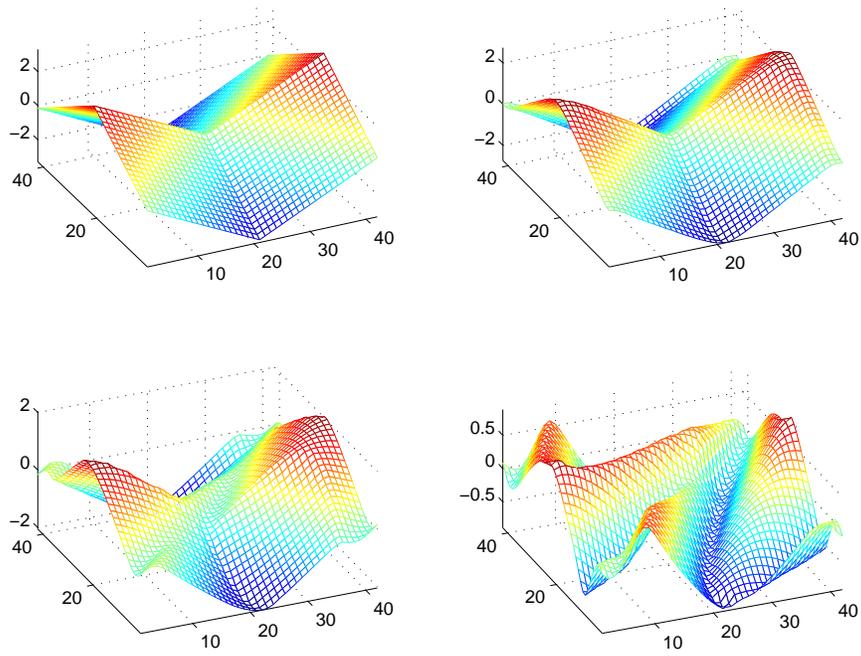


Figure 2.3: $H(u_x, u_y) = \sin(u_x + u_y)$.

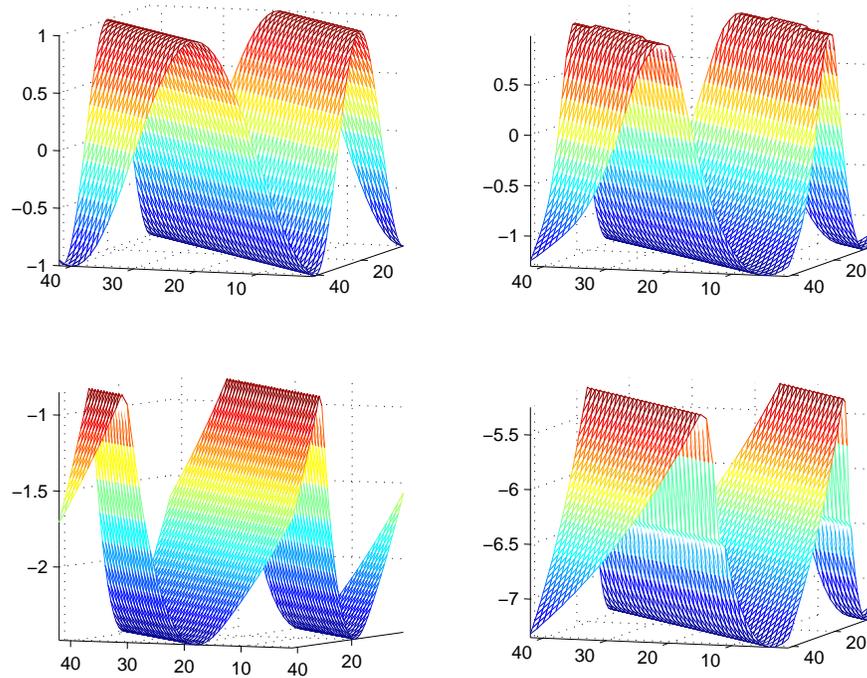


Figure 2.4: $H(u_x, u_y) = 0.5(u_x + u_y + 1)^2$.

In figure 2.4 the solution to Burgers' equation

$$u_t + 0.5(u_x + u_y + 1)^2 = 0,$$

with periodic BCs, is shown.

In figures 2.6 and 2.7 we show level set solutions of

$$\varphi_t - |\nabla\varphi| = 0,$$

for initial conditions of a sphere and torus. Again characteristics flow outward and boundary reconstructions use interior point stencils. The computational domain consists of nodes that are approximately equispaced, lying on concentric spheres as in figure 2.5.

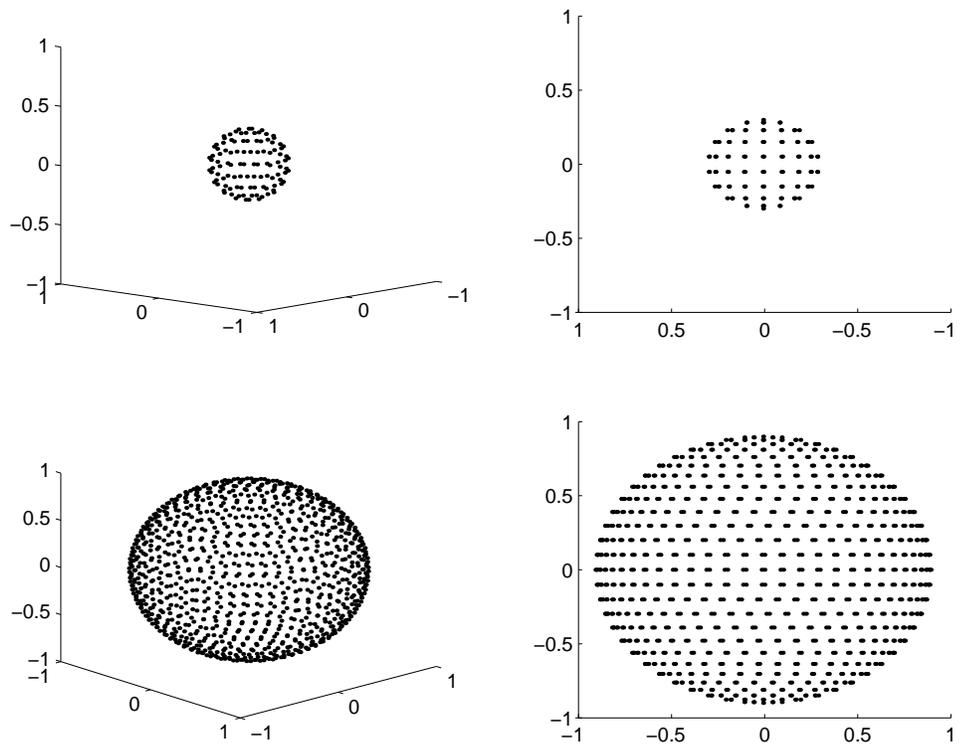


Figure 2.5: Points on smaller (top) and larger (bottom) concentric spheres.

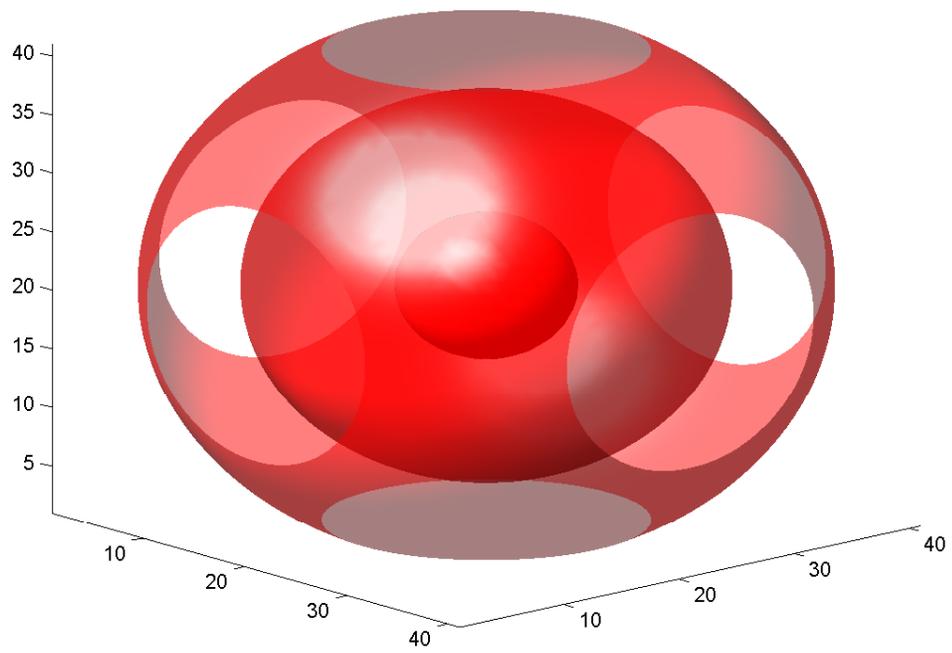


Figure 2.6: $H(p) = -|p|$, multiple time views.

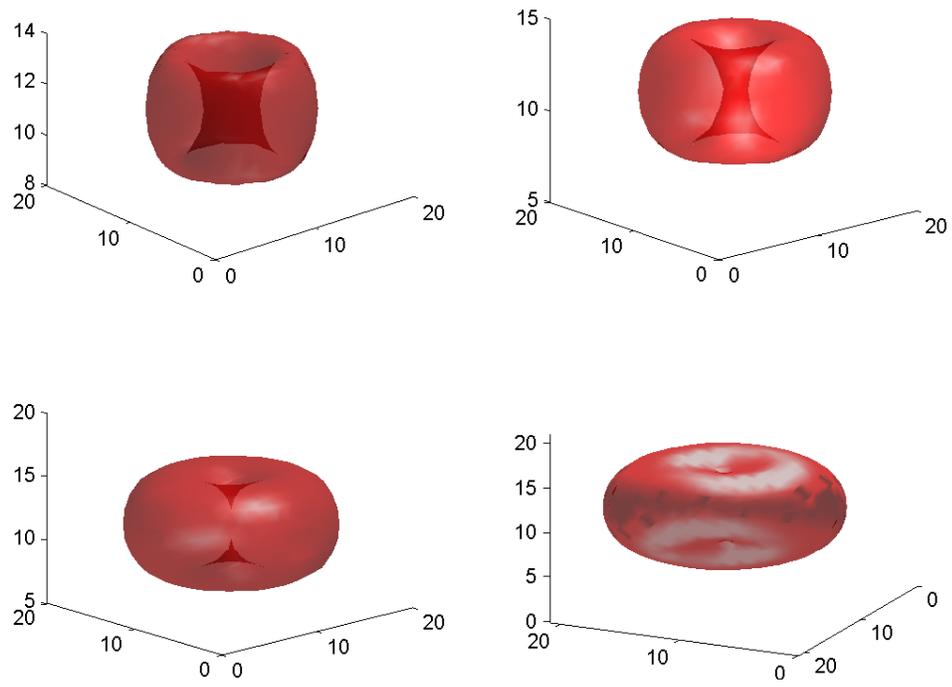


Figure 2.7: $H(p) = -|p|$.

Figures 2.8, 2.9, and 2.10 we show evolution sequences of the level set of a 4 dimensional hypertorus initialized as

$$\varphi(x, y, z, w) = r_3 - \sqrt{w^2 + [\sqrt{z^2 + [\sqrt{y^2 + x^2} - r_1]^2} - r_2]^2},$$

where $r_1 = 0.2, r_2 = 0.4, r_3 = 0.8$, subject to the PDE

$$\varphi_t - |\nabla\varphi| = 0.$$

In each figure we take a 3d slice of the data keeping the coordinate x_i fixed for the indicated i^{th} dimension, and then plot the level set $\{x|\varphi(x) = 0\}$ as a surface in 3d. In the future we will implement a local level set framework allowing for high dimensional computations of this type to be performed without storing gridpoints that are far away from the interface. The storage for this method would then be on the order of the size of the interface (an interface that is of codimension ≥ 1 with respect to the dimension of the computational domain). The speed would then be dependent on whether the user decided to precalculate the Lagrange coefficients of $\nabla\varphi$ or decided to do this at each timestep. In the first case the speed would be comparable to existing level set methods on uniform grids, with a penalty for data access speed only. In the second case the speed would be penalized by the need to invert the coefficient matrix at each data node for each timestep. For small stenciled reconstructions this is not too slow, and is unavoidable if the unstructured mesh is adaptive in time, no matter what reconstruction procedure is used.

2.9 Conclusion

The numerical solution of Hamilton-Jacobi equations on unstructured grids is becoming increasingly important. As higher dimensional problems are encountered

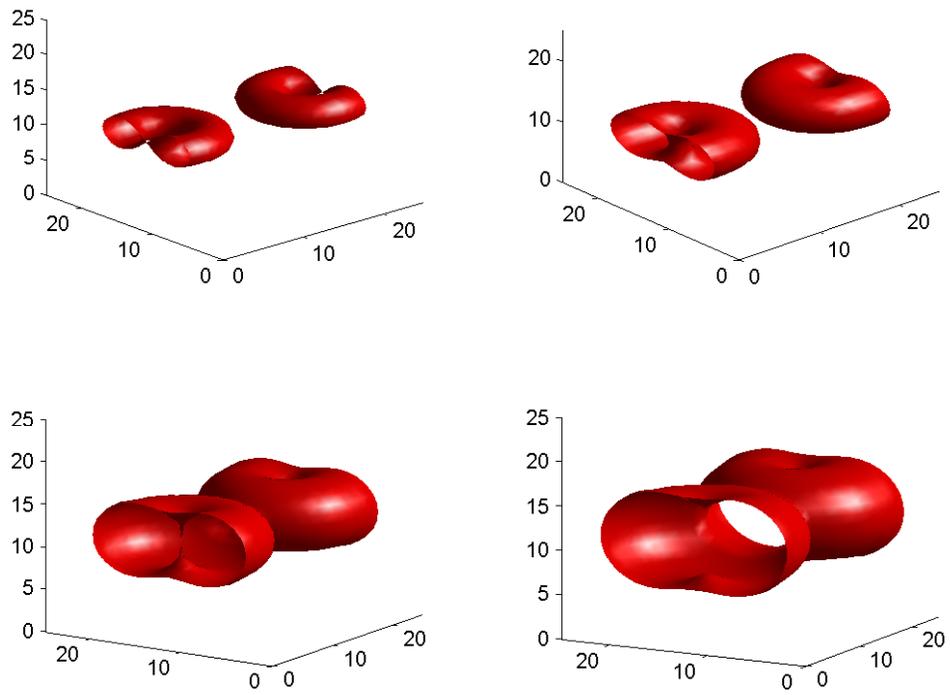


Figure 2.8: Dimension 1 fixed slice.

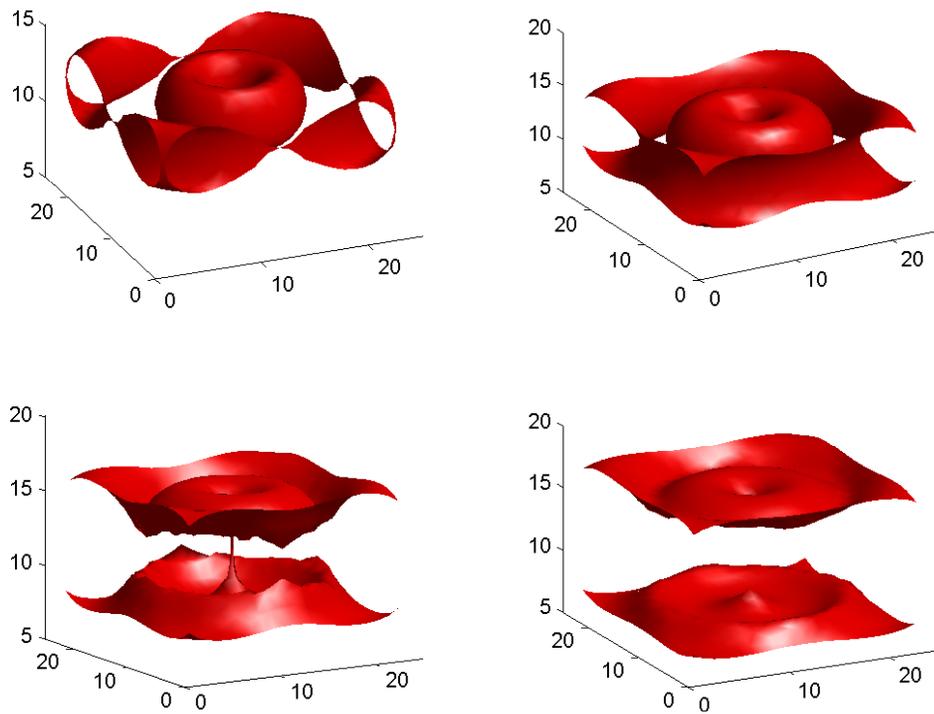


Figure 2.9: Dimension 3 fixed slice.

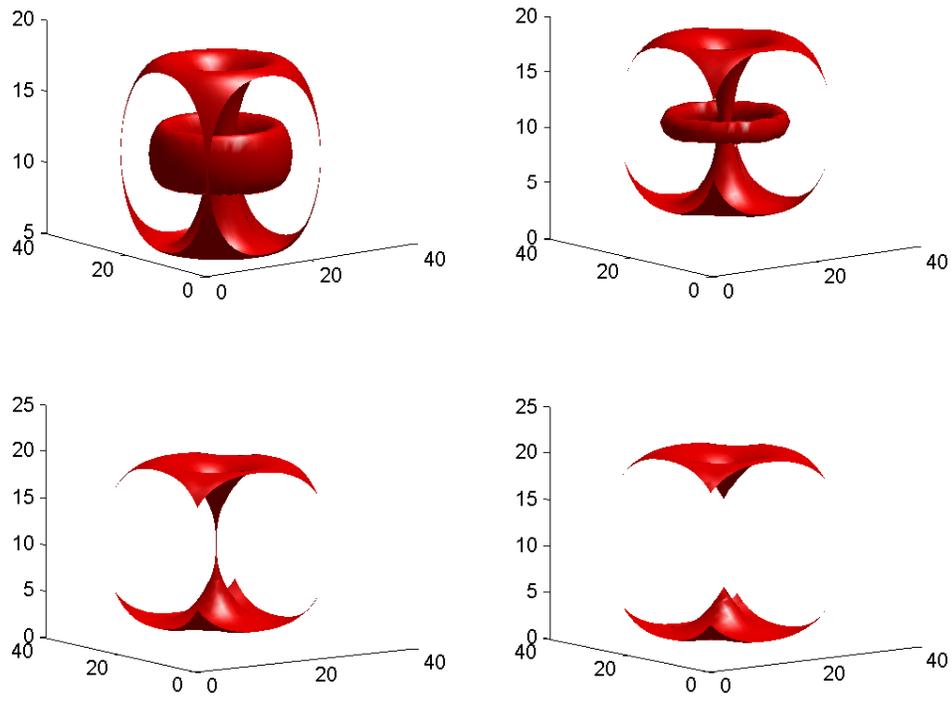


Figure 2.10: Dimension 4 fixed slice.

we would like to isolate the important features of the solution and resolve them locally instead of globally. Examples such as local level set computations are already pushing the computational boundaries on coarse grids in 5d [BCM01]. Minimization and control theory problems on irregularly shaped domains also call for scattered meshes to save space.

The methods presented here yield solutions which converge to the vanishing viscosity solution of HJ equations of the form (2.2). Error estimates for RBF interpolations and estimates on their partial derivatives have been proved, and are an area of current research. Optimal node choice, RBF parameter choice, and RBF basis function form are also areas that are being studied, and demand further theoretical results and a more intuitive description.

It should be noted that the arguments made for monotonicity and convergence of the LF and LLF schemes constructed in section 2.4 can be applied to other interpolation schemes for meshless numerical methods such as moving least squares, kernel based approximations, and partition of unity methods by writing these methods in Lagrange form [BKO96]. However, we cannot hope to apply our monotone construction to global interpolation schemes, as they will not satisfy the restrictions on the signs of d_j .

Other questions to be addressed concern the optimal way to handle neighbor access on a meshless data set, and what is the best way to automate the stencil selection process. We have addressed these problems here, but as they often consume the bulk of the computational time we suggest further study.

CHAPTER 3

Regularization of Certain Ill Posed Differential Equations

3.1 Introduction

In the field of material science one often encounters laws governing the motion of the growth of crystals in 2 dimensions that take the form of

$$\min_{\Omega} \oint_{\partial\Omega} \gamma(\theta) dS, \quad (3.1)$$

where θ is the angle of the normal of the crystal boundary $\partial\Omega$ with respect to some fixed vector [POM99][Gur93]. If we embed the boundary $\partial\Omega = \Gamma$ as a level set of a function φ , then the equation governing the motion of $\partial\Omega$ is

$$\varphi_t = (\gamma(\theta) + \gamma''(\theta))k, \quad (3.2)$$

where

$$k = \nabla \cdot \frac{\nabla\varphi}{|\nabla\varphi|}$$

which is the curvature of the interface, see [POM99] for a derivation of the Euler-Lagrange equation.

The PDE is ill posed when $\gamma(\theta) + \gamma''(\theta) < 0$. It is known that [Gur93] one can convexify γ in a way that will make $\gamma(\theta) + \gamma''(\theta) \geq 0$. This is known as the Frank convexification of the Wulff problem defined by (3.1). When we add an

additional constraint to (3.2) such that

$$\text{area}(\Omega) = C,$$

where C is a fixed constant, then the asymptotic solution of (3.2) is the same as the asymptotic solution of the convexified version of (3.2).

For example, if the surface tension is given by

$$\gamma(\theta) = 1 + |\sin(2\theta)|,$$

we can see that

$$\gamma(\theta) + \gamma''(\theta) = \begin{cases} 1 - 3 \sin(2\theta), & \text{for } \theta \in [0, \pi/2] \cup [\pi, 3\pi/2] \\ 1 + 3 \sin(2\theta), & \text{for } \theta \in [\pi/2, \pi] \cup [3\pi/2, 2\pi], \end{cases}$$

which changes sign. The Frank convexification is

$$\hat{\gamma}(\theta) = |\cos(\theta)| + |\sin(\theta)|,$$

so

$$\gamma(\theta) + \gamma''(\theta) = \sum_{i=0}^3 \delta(\theta - i\pi/2) \geq 0$$

where $\delta(x)$ is a distribution function.

The asymptotic shape these surface evolutions yield is a square, but if we were to numerically evolve the PDE given in (3.2) using γ we would quickly see a blowup in the solution at all points where $\gamma + \gamma'' < 0$. Even the evolution using $\hat{\gamma}$ would require some delicacy as the CFL condition would be restricted by the magnitude of $\delta(x)k$, which could become very large if $|\theta - i\pi/2| \approx 0$ near corners of Γ .

Inspired by the fact that there exist at the same time in nature both nonconvex interfacial energy gradients, γ , as well as physically stable crystal growths governed by these energies, we will attempt to add a regularization term to (3.2)

in the form of surface diffusion which will allow the ill posed PDE to yield results whose behavior asymptotically approaches the expected solution. We note that the intrinsic Laplacian of curvature found in surface diffusion evolutions can be derived by taking the Euler-Lagrange equation of surface energies which are functions of curvature. The physical relevance of this can be traced as far back as [Gib78],[Her51],[Mul57], with more recent derivations in [Gur93],[DGP92]. In [CL02] the authors derived a similar edge diffusion equation from atomistic principles for crystal island dynamics in the context of epitaxial growth. See also [CT94] for an overview.

Adding surface diffusion to a problem involving surface motion generally results in a stiff equation consisting of first or second order derivatives governing the surface advection along with fourth order derivatives found in the surface diffusion term. As these problems are nonlinear, an explicit numerical evolution is usually used, but its CFL condition is determined by the highest order derivative in the equation.

While little is known about the analytic properties of the PDE, there have been some attempts to tackle the numerical solution. In [CS99] the authors demonstrate many of the difficulties involved in modeling surface diffusion with an explicit scheme, the most serious of which is the timestep restriction $dt \leq Cdx^4$. Thus the attempts at numerical solutions are generally aimed at reducing the CFL condition as the fourth order restriction makes it intractable. Smereka has made some contributions to the subject, using a semi-implicit splitting method to make the CFL condition second order, but at the same time reducing the spatial accuracy to first order[Sme]. In [TWB02] the authors used a coupled system of second order PDEs to smooth normals of a surface, but their method is slightly

different from a true energy minimization of

$$\int_D |k|^2 \delta(\varphi) |\nabla \varphi| dx \quad (3.3)$$

(whose Euler-Lagrange equation yields the intrinsic Laplacian of curvature) in that it only minimizes this energy with respect to the normals of φ , not φ itself.

Here, we will present a new surface energy that is a function of both θ and the level set function φ and is to be minimized with respect to both variables. This yields a system of coupled second order PDEs (rather than fourth order) that approximate the regularized version of (3.2) when it is ill-posed.

3.2 PDE System

Given an interface $\Gamma \in \mathbb{R}^2$, which can be thought of as the boundary of a growing crystal, we embed it as the zero levelset of a function $\varphi(x, y)$. If we define $\theta(\varphi) = \arctan(\frac{\varphi_y}{\varphi_x})$, defined appropriately on $[0, 2\pi)$, then our formulation of the problem is to use a coupled set of PDEs derived from the energy

$$\begin{aligned} \min_{\varphi} \int_D \gamma(\theta(\varphi)) \delta(\varphi) |\nabla \varphi| dx + \min_{\varphi, \theta} A \int_D |\nabla_s \theta|^2 \delta(\varphi) |\nabla \varphi| dx \\ + \min_{\varphi, \theta} B \int_D (1 - \vec{n}_{\varphi} \cdot \vec{n}_{\theta}) \delta(\varphi) |\nabla \varphi| dx, \end{aligned} \quad (3.4)$$

where

$$n_{\varphi} = \text{normal of } \{\varphi = \text{constant}\} = \frac{\nabla \varphi}{|\nabla \varphi|},$$

and $n_{\theta} = \text{normal induced by } \theta$, e.g. $(\cos(\theta), \sin(\theta))$.

The first term defines the Wulff motion, the second is the surface diffusion with $A \ll 1$, and as $B \rightarrow \infty$ the third term should force the normals of φ and

θ to align. We note that

$$\begin{aligned} |k| &= \nabla \cdot \frac{\nabla \varphi}{|\nabla \varphi|} = \frac{\varphi_x^2 \varphi_{yy} + \varphi_y^2 \varphi_{xx} - 2\varphi_x \varphi_y \varphi_{xy}}{|\nabla \varphi|^3} \\ &= |\nabla_s \theta| = \frac{|\varphi_x \theta_y - \varphi_y \theta_x|}{|\nabla \varphi|}. \end{aligned}$$

This is the relation that shows us the equivalence of the second integral term in (3.4) with (3.3).

It can be seen that because (3.4) is an integral equation of first order in both θ and φ that when we take the Euler-Lagrange equation we get PDEs of second order, rather than fourth order.

The Euler-Lagrange equations derived from (3.4) (see section 3.6) and implemented with gradient descent are

$$\begin{aligned} \varphi_t &= (\gamma(\arctan(\frac{\varphi_y}{\varphi_x})) + \gamma''(\arctan(\frac{\varphi_y}{\varphi_x})))k|\nabla \varphi| & (3.5) \\ &- A \nabla \cdot \left(\frac{1}{|\nabla \varphi|^3} (\varphi_x^3 \theta_y^2 - 2\varphi_y^3 \theta_x \theta_y + 2\varphi_y^2 \varphi_x \theta_y^2 - \varphi_y^2 \varphi_x \theta_x^2, \right. \\ &\quad \left. \varphi_y^3 \theta_x^2 - 2\varphi_x^3 \theta_y \theta_x + 2\varphi_x^2 \varphi_y \theta_x^2 - \varphi_x^2 \varphi_y \theta_y^2) \right) |\nabla \varphi| \\ &+ B \nabla \cdot \left(\frac{\nabla \varphi}{|\nabla \varphi|} - (\cos \theta, \sin \theta) \right) |\nabla \varphi| \\ \theta_t &= A \nabla \cdot \left(\frac{1}{|\nabla \varphi|^2} (\varphi_y^2 \theta_x - \varphi_x \varphi_y \theta_y, \varphi_x^2 \theta_y - \varphi_y \varphi_x \theta_x) \right) \\ &+ B(-\sin \theta, \cos \theta) \cdot \frac{\nabla \varphi}{|\nabla \varphi|}, \end{aligned}$$

where the sign in front of the matching and smoothing terms depends on the orientation of θ with respect to $\nabla \varphi$. It can be seen that the first term was differentiated with respect to φ only, while the A and B terms were differentiated with respect to both θ and φ . Also, in order to allow all level sets to move we have replaced the δ function in front of all terms and replaced it with $|\nabla \varphi|$ which has now become a standard practice. The B terms are normal matching terms,

and the evolution term in φ ,

$$\varphi_t = B \nabla \cdot \left(\frac{\nabla \varphi}{|\nabla \varphi|} - (\cos \theta, \sin \theta) \right) |\nabla \varphi|$$

can be thought of as the difference in the curvatures of the actual level set function φ and the curvature of a fictitious level set function whose normals are defined by the vector $(\cos \theta, \sin \theta)$.

The highly nonlinear A terms complement each other to approximate the Laplacian of curvature term.

Another related approach is to remove the explicit dependence of θ upon φ in the first integral, and then minimize over both φ and θ .

The surface energy term (the term with γ) becomes

$$\varphi_t = \nabla \cdot \left(\gamma(\theta) \frac{\nabla \varphi}{|\nabla \varphi|} \right) |\nabla \varphi| = (\gamma(\theta)k + \nabla \gamma(\theta) \cdot \frac{\nabla \varphi}{|\nabla \varphi|}) |\nabla \varphi|. \quad (3.6)$$

The θ evolution also gets an additional term,

$$\theta_t = -\gamma'(\theta). \quad (3.7)$$

3.3 Numerical Methods

We treat the PDEs using the method of lines. We leave φ_t PDEs of (3.5) in divergence form and treat each of the three integral minimizations separately as a Hamilton-Jacobi problem of the form

$$\varphi_t + v_n |\nabla \varphi| = 0.$$

All of the derivatives within each v_n (and explicitly stated in (3.5)) are calculated using central differencing, and then we take the divergence (if indicated) using central differencing to find v_n . Thus the φ_t PDE is split into 3 steps using a

different $|\nabla\varphi|$ in each instance, depending on the sign of v_n . Time advancement is done using TVD Runge-Kutta solvers.

For the PDE involving θ_t we apply central differencing to all the explicit derivatives in (3.5) and then take the divergence (if indicated) of this using central differencing again. Time advancement is also done using TVD Runge-Kutta.

There is one important step that is done when taking a spatial derivative of θ at any point in the calculation. In order that we do not smear the discontinuity at 0 (and 2π) when taking finite differences we use the value at a point $(\theta_i) \bmod 2\pi$ which minimizes the discontinuity between the neighboring points in the stencil.

For example if we would like to calculate θ_x using

$$\theta_x(x_i, y_j) \approx \frac{\theta_{i+1,j} - \theta_{i-1,j}}{dx},$$

then we would instead find $k_0 \in \mathbb{Z}$ that minimizes

$$|(\theta_{i+1,j} + 2\pi k_0) \bmod (2\pi) - \theta_{i-1,j}|$$

and then take

$$\theta_x(x_i, y_j) \approx \frac{(\theta_{i+1,j} + 2\pi k_0) \bmod (2\pi) - \theta_{i-1,j}}{dx}.$$

Of course for a surface with only one discontinuity we do not need to search over all of \mathbb{Z} , just the set $\{-1, 0, 1\}$.

Also, the asymptotic limit of these equations is for the interface to shrink to a Wulff shape and then vanish, so we use a projected gradient method [Ros61] to ensure that volume remains fixed in time. This is done by requiring that

$$\left| \int_D H(\varphi) dx - A_0 \right| < \epsilon \tag{3.8}$$

at any given timestep. If this condition is not satisfied then we use Newton's method to find the Lagrange multiplier λ such that when we take the Euler-Lagrange equation of (3.8), apply gradient descent to get

$$\varphi_t + \lambda |\nabla \varphi| = 0,$$

and then advance for one timestep we will satisfy (3.8), see [OS01] for details. All of this is done within a local level set framework[PMO99].

3.4 Numerical Experiments

Figure 3.1 shows the evolution of the PDE system (3.5) with

$$\gamma(\theta) = 1 + |\sin(2\theta)|.$$

The volume is forced to remain constant as well. The level set representing the interface is shown on the left, as well as a plot of θ at the same time on the right.

In figure 3.2 we plot an interface initialized as a circle with $r = 0.3$, and at 3 different times after expanding under the surface energy where $\gamma(\theta) = -1$. We get first order convergence to the analytical solution which is a circle with radius

$$r(T) = \sqrt{2T + (0.3)^2}.$$

Figure 3.3 shows a circle after a few timesteps that has been evolved with

$$\varphi_t = -k |\nabla \varphi|$$

without the extra regularization terms. Note the expected instability quickly occurring.

Figure 3.4 also shows backwards curvature evolution along with plots of θ when the initial interface is a cosine wave. This represents an evolution that attempts to maximize the length of a curve. Note that the discontinuities in θ at $\{0, 2\pi\}$ are not smoothed and do not cause oscillations.

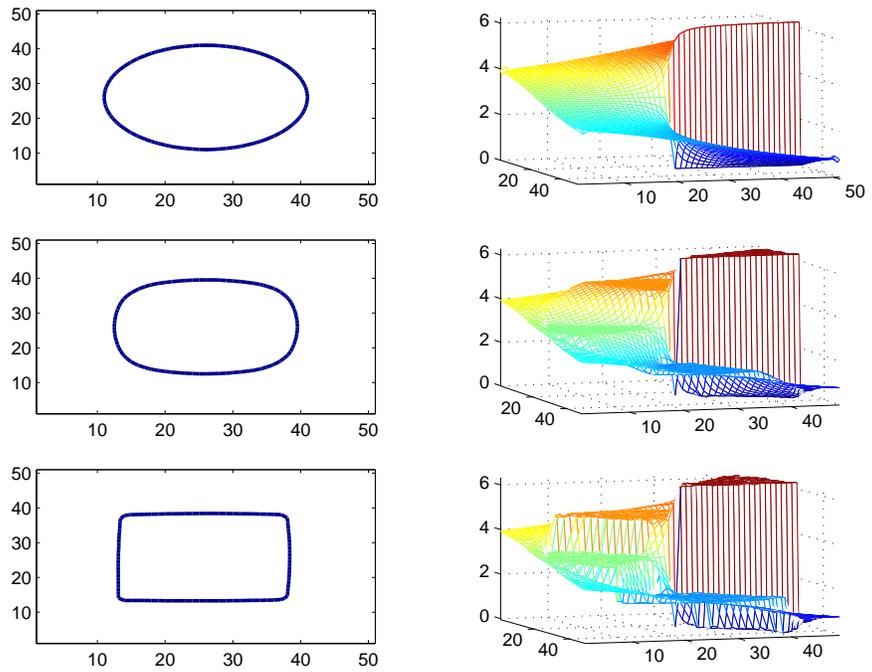


Figure 3.1: $\gamma(\theta) = 1 + |\sin(2\theta)|$.

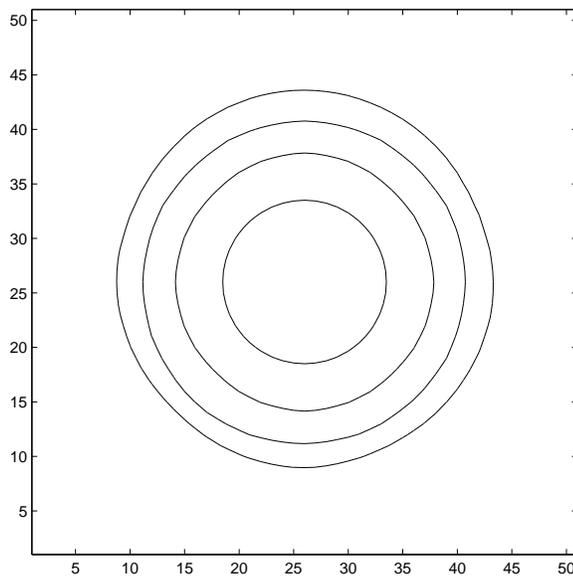


Figure 3.2: $\gamma(\theta) = -1$.

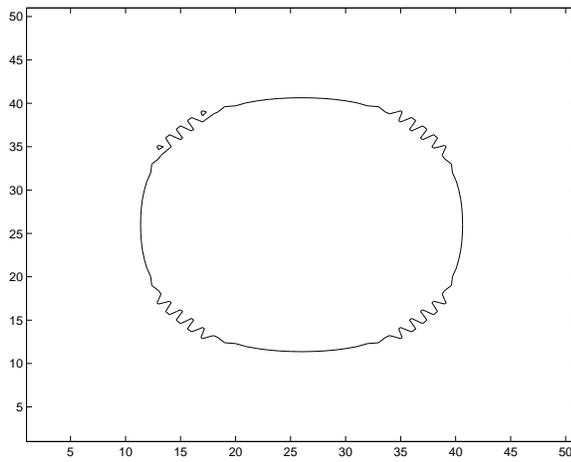


Figure 3.3: $\gamma(\theta) = -1$, no regularization.

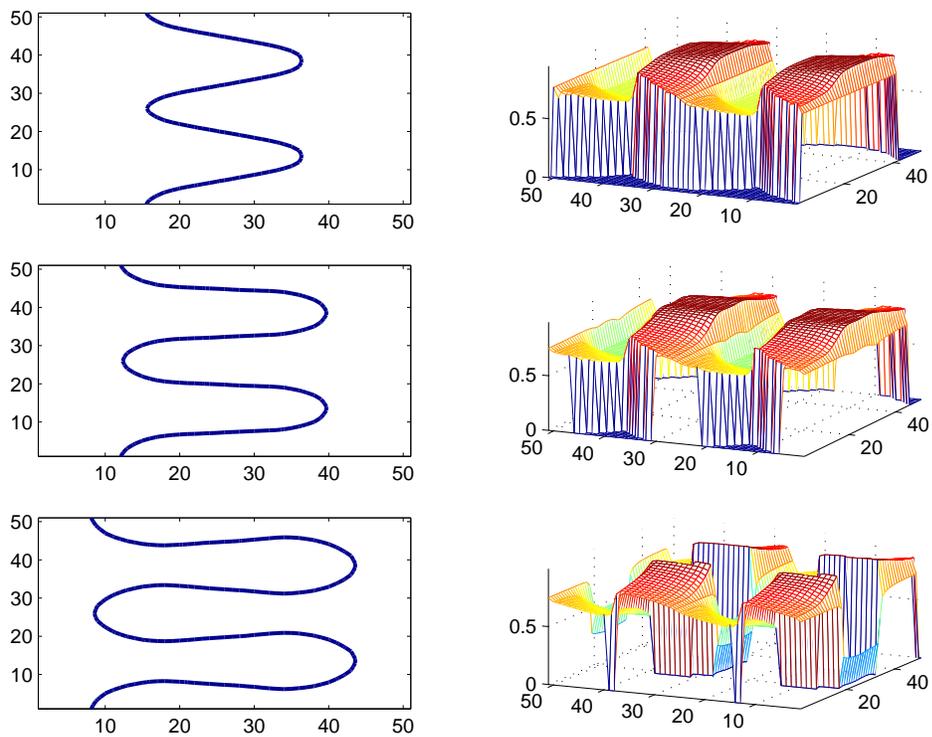


Figure 3.4: $\gamma(\theta) = -1$.

3.5 Conclusion

The applications of surface diffusion in material science are well documented. The authors of [CL02] have recently demonstrated its presence in the theory of epitaxial thin film growth. We have implemented an explicit method within UCLA's thin film evolution code, which already includes an adaptive timestepping routine that calculates the CFL condition at each step, and found the motion to require expectedly small timesteps. Large timesteps are its main advantage over KMC simulations, so it would be better to have a way of keeping the CFL condition as unrestrictive as possible to keep this advantage.

We have implemented a coupled system of nonlinear PDEs exploiting the relationship between curvature of level sets and the angle of their normals with respect to a fixed vector. The analytic properties of these PDE is an area open to further research, as is the intrinsic Laplacian of curvature, on which we currently have few theoretical results.

Generalizing the method to 3d is a possibility, however it presents new problems arising from the singularities of the conjugate functions. Surface processing, the 3d analogue of image processing, involves surface diffusion and would benefit from the reduction in the CFL constraint.

3.6 Derivation of Euler-Lagrange Equations

Here we will go through the derivation of equation (3.5). The main idea is that for solving

$$\min_u \int_D f(u, \nabla u = p) dx$$

we will use the Euler-Lagrange equation to tell us the method of descent towards the solution of a convex problem. This method of gradient descent tells us that we should use

$$u_t = -\frac{\partial}{\partial u}f + \nabla \cdot (\nabla_p f), \quad (3.9)$$

where $\nabla_p G(p) \equiv (G_{p_1}, G_{p_2}, \dots, G_{p_n})$.

For the PDE $\varphi_t = L(\varphi, \theta)$ we note that the Euler-Lagrange equation of the first integral

$$\min_{\varphi} \int_D \gamma(\theta(\varphi)) \delta(\varphi) |\nabla \varphi| dx$$

is derived in [POM99].

For the integral

$$\min_{\varphi} \int_D \left(\frac{|\varphi_x \theta_y - \varphi_y \theta_x|}{|\nabla \varphi|} \right)^2 \delta(\varphi) |\nabla \varphi| dx,$$

if we let

$$F = \left(\frac{|\varphi_x \theta_y - \varphi_y \theta_x|}{|\nabla \varphi|} \right)^2$$

then to find the minimum we must have

$$\delta'(\varphi) |\nabla \varphi| F = \frac{\partial}{\partial x} \left[\frac{\partial}{\partial \varphi_x} [F \delta |\nabla \varphi|] \right] + \frac{\partial}{\partial y} \left[\frac{\partial}{\partial \varphi_y} [F \delta |\nabla \varphi|] \right]. \quad (3.10)$$

It should be noted that because $|\nabla \varphi| F = G$ is homogeneous of degree 1 in $\varphi_{x_i} \forall i$ then all the terms on the right hand side which will have $\delta'(\varphi)$ in them will cancel with the left side of (3.10), as the right side of (3.10) is

$$\nabla \cdot [\nabla_p (G \delta)] = \delta \left[\sum_{i=1}^n F_{p_i p_i} (p_i)_{x_i} \right] + \delta' \left[\sum_{i=1}^n F_{p_i} (p_i) \right]$$

where $p_i = \varphi_{x_i}$. This is because of Euler's relationship $g(x)x = g(x)$ for homogeneous functions of degree 1. Thus we need not worry about the terms with $\delta'(\varphi)$ in them, and do not need to calculate them when they arise in taking the various differentials in (3.10).

Thus we need

$$0 = \nabla \cdot \left[\nabla_p \left(\frac{p_1^2 \theta_y^2 + p_2^2 \theta_x^2 - 2p_1 p_2 \theta_x \theta_y}{|(p_1, p_2)|} \delta(\varphi) \right) \right].$$

Performing ∇_p and $\nabla \cdot$, discarding the $\delta'(\varphi)$ terms and leaving the answer in divergence form we get

$$0 = \delta(\varphi) \nabla \cdot \left[\frac{1}{|(p_1, p_2)|^3} (p_1^3 \theta_y^2 - 2p_2^3 \theta_x \theta_y + 2p_2^2 p_1 \theta_y^2 - p_2^2 p_1 \theta_x^2, \right. \\ \left. p_2^3 \theta_x^2 - 2p_1^3 \theta_y \theta_x + 2p_1^2 p_2 \theta_x^2 - p_1^2 p_2 \theta_y^2) \right]. \quad (3.11)$$

For the integral

$$\min_{\varphi} \int_D \left(1 - (\cos \theta, \sin \theta) \cdot \frac{\nabla \varphi}{|\nabla \varphi|} \right) \delta(\varphi) |\nabla \varphi| dx,$$

if we let

$$F = 1 - (\cos \theta, \sin \theta) \cdot \frac{\nabla \varphi}{|\nabla \varphi|}$$

then use this F in equation (3.10), then after taking the partials $\frac{\partial}{\partial \varphi_{x_i}}$ we see that the first term on the right hand side of (3.10) becomes

$$\frac{\partial}{\partial x} \left(\delta(\varphi) \left(\frac{\varphi_x}{|\nabla \varphi|} - \cos \theta \right) \right) = \delta'(\varphi) \left(\frac{\varphi_x^2}{|\nabla \varphi|} - (\cos \theta) \varphi_x \right) \\ + \delta(\varphi) \left(\frac{\partial}{\partial x} \left(\frac{\varphi_x}{|\nabla \varphi|} \right) + (\sin \theta) \theta_x \right), \quad (3.12)$$

and the second term becomes,

$$\frac{\partial}{\partial y} \left(\delta(\varphi) \left(\frac{\varphi_y}{|\nabla \varphi|} - \sin \theta \right) \right) = \delta'(\varphi) \left(\frac{\varphi_y^2}{|\nabla \varphi|} - (\sin \theta) \varphi_y \right) \\ + \delta(\varphi) \left(\frac{\partial}{\partial y} \left(\frac{\varphi_y}{|\nabla \varphi|} \right) - (\cos \theta) \theta_y \right). \quad (3.13)$$

Here we can explicitly see that the $\delta'(\varphi)$ terms from (3.12) and (3.13) cancel with the left side of (3.10), giving us

$$0 = \delta(\varphi) \nabla \cdot \left(\frac{\nabla \varphi}{|\nabla \varphi|} - (\cos \theta, \sin \theta) \right).$$

For the integral in theta,

$$\min_{\theta} \int_D \left(\frac{|\varphi_x \theta_y - \varphi_y \theta_x|}{|\nabla \varphi|} \right)^2 \delta(\varphi) |\nabla \varphi| dx,$$

the procedure is as with the φ minimizations but where $p \equiv (\theta_{x_1}, \dots, \theta_{x_n})$. Also, we can alter the integral in order to treat all level sets the same,

$$\min_{\theta} \int_D \left(\frac{|\varphi_x \theta_y - \varphi_y \theta_x|}{|\nabla \varphi|} \right)^2 \delta(\varphi) |\nabla \varphi| dx, \Rightarrow \min_{\theta} \int_D \left(\frac{|\varphi_x \theta_y - \varphi_y \theta_x|}{|\nabla \varphi|} \right)^2 dx.$$

From here applying (3.9) is straightforward.

For the minimizations with respect to θ of the integrals for the normal matching and surface tension energies we apply the same alteration and again apply (3.9) in a straightforward manner.

CHAPTER 4

Topology Preserving Level Set Motion

4.1 Introduction

The implicit framework of level set interface motion has yielded many advantages over Lagrangian tracking methods, one of the most notable being its ability to incorporate changes in topology of the front without requiring changes in the algorithm being used to move it.

If this were the only benefit of using level sets for interfacial motion then when the time arose that we did not want changes in topology to occur we would probably decide to use a different method. However, there are other advantages of level set motion which make it desirable[OF01], and the time has come to use level sets to track interfaces whose topology remains fixed. Some examples of this are from medical imaging, where tissues may fold upon each other without joining their interiors[TT02]. Thus, in the spirit of the original level set motion that was able to handle topological changes "without emotional involvement," we would like to find a way for topology to be preserved in an unemotional way.

4.2 A Discrete Method

In [HXP01] the authors introduced a topology preserving method which monitors the digital topology of an evolving interface and prohibits motion if it would in-

roduce topological changes. Thus, at each timestep of discretized time evolution the method computes the change, $\delta\varphi$, in the level set function according to the physical evolution equation, and then decides whether or not $\varphi(x_i, t) + \delta\varphi$ would result in a change in the local topology of the interface by examining the values of the neighbors of x_i . If it does change, then the method sets $\varphi(x_i, t+dt) \rightarrow \varphi(x_i, t)$ and goes on to the next x_i , otherwise it sets $\varphi(x_i, t+dt) \rightarrow \varphi(x_i, t) + \delta\varphi$.

However, there are some drawbacks to the method of [HXP01]. The final solution reached depends on the order in which the points are visited, and using different topology connections will also give different answers. The method is only applicable on lattice structures, not on arbitrary data sets. We can also see that this type of evolution is beginning to stray away from the goal of the use of a simple algorithm to evolve the interface that does not require specialized conditions near points of topological change.

4.3 A PDE Based Method

For many applications the evolution of the level set defining the interface in question can be defined in terms of a single PDE. This can be done when the motion can be described in terms of intrinsic (solely geometry- or position-based) properties. Topology is an example of a nonintrinsic property which requires more than one PDE to describe. In [HO98] the authors presented a coupled system of PDEs to evolve a class of models including the Cauchy-Riemann equations which can be written in a Lagrangian way as:

$$(x_t, y_t) = \vec{v}(t, x, y, x_s, y_s),$$

subject to periodic boundary conditions and initial conditions

$$(x(s, 0), y(s, 0)) = (x_0(s), y_0(s)), \quad 0 \leq s \leq L.$$

Where the parameter s , which may or may not be arclength, is the nonintrinsic variable. In the Eulerian format the two evolution equations must include information about the location of the front, as well as the value of the arclength. So one could define $\varphi(t, x, y)$, whose zero level set defined the interface, and

$$\psi(t, x, y) \text{ such that } \psi(t = 0, x(0, s), y(0, s)) \equiv s,$$

which defines the arclength function. Note that ψ is conjugate to φ in the sense that the two form an orthogonal coordinate system on the zero level set of φ ,

$$\nabla\varphi \cdot (\nabla\psi)^* = \varphi_x\psi_y - \varphi_y\psi_x \neq 0$$

when $t = 0$.

The interface in question, Γ is evolved in time by requiring that

$$\varphi(x(s, t), y(s, t), t) \equiv 0 \quad \forall t > 0, \tag{4.1}$$

and we require the arclength function to obey

$$\psi(x(s, t), y(s, t), t) \equiv s \quad \forall t > 0. \tag{4.2}$$

The PDEs governing the motion of Γ can be found by differentiating (4.1) and (4.2) with respect to t which yields

$$\varphi_t + \vec{w} \cdot \nabla\varphi = 0, \tag{4.3}$$

$$\psi_t + \vec{w} \cdot \nabla\psi = 0. \tag{4.4}$$

If we differentiate(4.1) and (4.2) with respect to s we find that

$$(x_s, y_s) = [(\nabla\varphi) \cdot (\nabla\psi)^*]^{-1}(-\varphi_y, \varphi_x), \tag{4.5}$$

and note that

$$\nabla\varphi \cdot (\nabla\psi)^* = \varphi_x\psi_y - \varphi_y\psi_x = -J \tag{4.6}$$

must not vanish if (4.5) is to be well defined. The quantity $-J$ is the Jacobian of (φ, ψ) , and we use $-J$ in the definition so that $J > 0$.

Given this framework we can see that when an interface begins to change topology we will have either that

$$J \rightarrow 0 \text{ at merge points,}$$

or

$$J \rightarrow \infty \text{ at pinch points.}$$

Note that the definition of the interior of Γ as being $\{x|\varphi(x) > 0\}$ or $\{x|\varphi(x) < 0\}$ will interchange merge and pinch points.

Thus if we could monitor J as it grows towards ∞ or shrinks towards 0 we would be able to also monitor the topology changes of Γ . So we need to be able to determine

$$J(t, x, y) \forall t > 0, (x, y) \in \Gamma.$$

One option would be to evolve φ and ψ using (4.3) and (4.4), and then form J using (4.6). However, this is difficult in our Eulerian level set framework where ψ is multivalued on closed curves and the exact location of Γ is implicitly defined.

Another way to find $J(t, x, y)$ is to derive a PDE for its motion in time and follow this evolution, thus capturing ψ implicitly. To obtain this PDE we follow the method of [HO98]. We begin with $\Phi(t, a) \equiv (x(t, a), y(t, a))$ which is the trajectory of a point referenced by $a \in [0, L]$, such that

$$\varphi(t, \Phi(t, a)) = \varphi(0, a), \psi(t, \Phi(t, a)) = \psi(0, a).$$

Differentiating with respect to a and taking the determinant we have

$$\left| \frac{\partial(\varphi, \psi)}{\partial(x, y)} \right| \left| \frac{\partial(x, y)}{\partial a} \right| = \left| \frac{\partial(\varphi_0, \psi_0)}{\partial a} \right|.$$

Differentiating along the trajectory we obtain

$$D_t J \left| \frac{\partial(x, y)}{\partial a} \right| + J D_t \left| \frac{\partial(x, y)}{\partial a} \right| = 0.$$

A simple derivation often used in fluid mechanics[Mey82] is that

$$D_t \left| \frac{\partial(x, y)}{\partial a} \right| = (\nabla \cdot \vec{v}) \left| \frac{\partial(x, y)}{\partial a} \right|,$$

thus giving

$$J_t + \nabla \cdot (J\vec{v}) = 0. \tag{4.7}$$

It should be noted that this derivation works in n space dimensions for a Jacobian of ϕ and $n - 1$ conjugate functions.

We now have equations (4.3) and (4.7) that govern the variables in which we are interested, and so it is left to determine how to use them to preserve the initial topology of Γ .

If we write \vec{v} in terms of the velocity normal to the interface, v_n , we have $\vec{v} = v_n \frac{\nabla\phi}{|\nabla\phi|}$ and we get:

$$J_t + \vec{v} \cdot (\nabla J) = J(\nabla \cdot v_n \frac{\nabla\phi}{|\nabla\phi|}). \tag{4.8}$$

So if for example $v_n = 1$, then by removing the advection in the equation we see that $J(t) = J_0 e^{\int_0^t k dt}$, where k is the curvature of Γ . This shows that (4.7) is related to the evolution of the curvature along characteristics. This gives some intuition as to why J approaching 0 or ∞ indicates a change in topology.

Therefore, we will force the motion of Γ to stop at (t, x, y) if $J(t, x, y)$ is near those extrema. Initially, $J(0, x, y)$ will be set at 1. Forcing the motion of Γ to stop is done by modifying the physical velocity of the problem, \vec{v} , by multiplying it by a function $p(J)$ such that

$$p(J) \rightarrow 0, \text{ as } J \rightarrow 0 \text{ or } J \rightarrow \infty,$$

and $p(1) = 1$. An example is the linear hat function

$$p(J) \equiv \begin{cases} \frac{J-1}{\delta} + 1, & \delta < J \leq 1 \\ \frac{1-J}{\delta} + 1, & 1 < J \leq 1 + \delta \\ 0 & \text{otherwise,} \end{cases}$$

where $0 < \delta < 1$.

We can see that although topology changes occur when $J \rightarrow \infty$, the above $p(J)$ stops the motion of Γ as $J \rightarrow 1 + \delta$, which is relatively small. One remedy to this situation would be to make a nonsymmetric $p(J)$ such that $p(J) > 0 \forall J < \infty$, but still require that $p(J) \rightarrow 0$ as $J \rightarrow \infty$. Then the singularities at both extrema $\{0, \infty\}$ would be treated similarly. However, this would lead to the necessity to resolving a delta function in J , whose range is infinite, which is a problem that cannot be handled numerically. Thus, if we perform a fractional linear transformation (FLT) $f(J)$ that takes

$$f(0) \rightarrow 0, f(1) \rightarrow 1, f(\infty) \rightarrow 2,$$

we can use a symmetric $p(f(J))$ as described above without worrying about the numerical problems associated with resolving a delta function.

The FLT we use to scale J is

$$f(J) \equiv \frac{2J}{1+J}. \quad (4.9)$$

To follow the evolution of f if we note that

$$f_t = \frac{2}{(1+J)^2} J_t \text{ and } J = \frac{f}{2-f}$$

then we can derive the PDE

$$\frac{f_t}{2} \left(\frac{2}{2-f} \right)^2 + \nabla \cdot \left(\vec{v} \frac{f}{2-f} \right) = 0. \quad (4.10)$$

While this PDE is not in conservation form, if we apply the method of lines we can solve the spatially dependent part of the equation as a conservation law and then treat the ODE as a scaled version of

$$g_t + L(g) = 0, \text{ where } L(g) = \nabla \cdot \left(\vec{v} \frac{g}{2-g} \right). \quad (4.11)$$

4.4 Numerical Implementation

While the PDE framework of our method may be analytically clear, its numerical implementation requires some finesse. Given an initial interface, Γ we form the level set function φ such that $\varphi|_{\Gamma} = 0$. In order to initialize J (which will be used in this section to mean J or f , unless there is a difference in their treatment which will be made clear) we need to extend it off Γ . The best way to extend quantities known only on the interface into the domain near the interface is still a problem that is being researched. One method is to extend the quantity, S , along characteristics in the normal direction by solving a PDE of the form

$$S_t + \text{sign}(\varphi) \vec{N} \cdot \nabla S = 0.$$

However, in our case in order to prevent topology change we need to know the value of J in front of Γ if we are going to stop the motion of Γ prior to the change. So extending J off the interface does not help us. Thus if Γ is defined as the set of points where $\varphi(x) = 0$ then we need to follow J on level sets near 0. However, we do not want to penalize the merging of all levelsets, otherwise our evolution procedure would begin to freeze as soon as level sets far from Γ began to merge or pinch. To fix this we adjust J by smoothing it by

$$J(x) \rightarrow 1 + g(\varphi(x))(J(x) - 1), \quad (4.12)$$

where $g(u)$ is a cutoff function such as

$$g(u) \equiv \begin{cases} 1, & \text{if } |u| < \beta \\ (|u| - \gamma)^2 \frac{2|u| + \gamma - 3\beta}{(\gamma - \beta)^3}, & \text{if } \beta < |u| \leq \gamma \\ 0 & \text{otherwise,} \end{cases}$$

where $0 < \beta < \gamma$ determine the support of the cutoff. This smoothing will leave J unchanged near the interface and taper it off to have value 1 away from the front, thus allowing the level sets of φ which are away from the front to change topology without any effect on the evolution of Γ .

After having initialized J and φ we set up the framework for a local level set (LLS) calculation as described in [PMO99]. However, because J is very sensitive to perturbations it is necessary to set up extra interior tubes in which J will be calculated so that the discontinuity in $\nabla\varphi$ at the edge of the outer LLS tubes will not cause problems if quantities involving second derivatives, such as curvature, are involved in the evolution of J .

Once this LLS framework is established we begin to evolve the PDEs from (4.7) (or (4.10)) and (4.3). After adjusting the physical velocity \vec{v} to be $\vec{v}p(J)$, the equation for φ is rewritten as $\varphi_t + v_n|\nabla\varphi| = 0$ and solved using standard ENO/WENO Roe-Fix Hamilton-Jacobi solvers such as those found in [OS91], etc. with TVD Runge-Kutta timestepping. The equation for f is solved using similar ENO/WENO Roe-Fix methods for conservation laws. The method of lines applied to (4.10) results in using TVD Runge-Kutta timestepping with a timestep scaled pointwise by the factor $2(\frac{2-f}{2})^2$ in solving (4.11). If J is to be used then there is no timestep scaling.

After each timestep we resmooth J or f using (4.12), and periodically we must reinitialize φ making sure not to move the interface (and possibly change topology) when doing so.

In multiple dimensions the evolution procedure generalizes as noted above and with PDEs to be solved are identical.

4.5 Numerical Examples

We begin with 2 examples of topology preserving level set motion applied to the problem of image segmentation. We use the Chan-Vese minimization[CV01] of the Mumford-Shah segmentation energy[MS89] to define the prepenalized evolution velocity. Figure 4.1 shows the segmentation (zoomed in near the conclusion) of 2 circles. Figure 4.2 we show the segmentation process without topology preservation.

Figure 4.3 shows plots at different timesteps (after initialization as a small circle) during the segmentation of an annulus with a small slit in it. In figure 4.4 we turn off the topology preservation and see that the slit is not identified by the segmentation.

In figure 4.5 we track the motion of both a circle and a line under topology preserving outward normal motion

$$\varphi_t - p|\nabla\varphi| = 0,$$

where p is the penalty function. We show the final plot of the Jacobian, J , in figure 4.6, note the large magnitudes of J near points where topology is almost changing. Figure 4.7 shows the same evolution without topology preservation.

Finally, in figure 4.8 we show the evolution of 2 hemispheres in 3d under outward normal motion with topology preservation, and without topology preservation in figure 4.9.

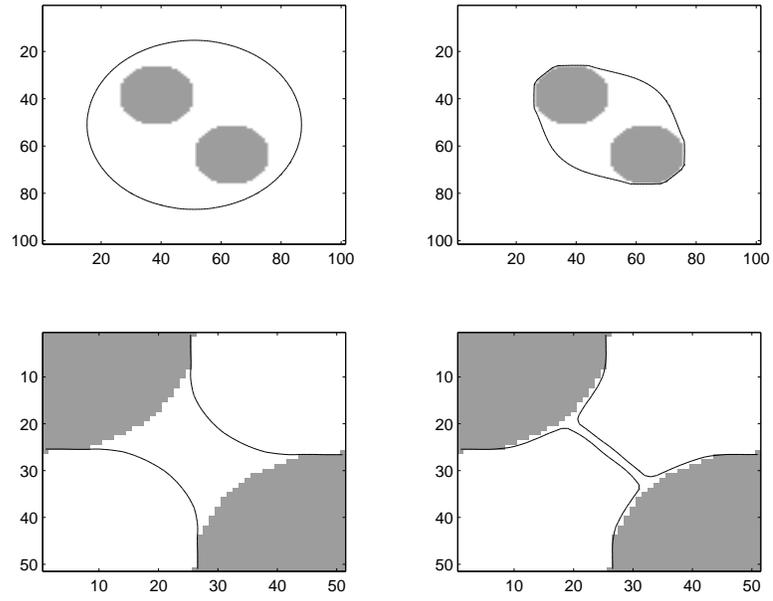


Figure 4.1: Segmentation of 2 circles with topology preservation.

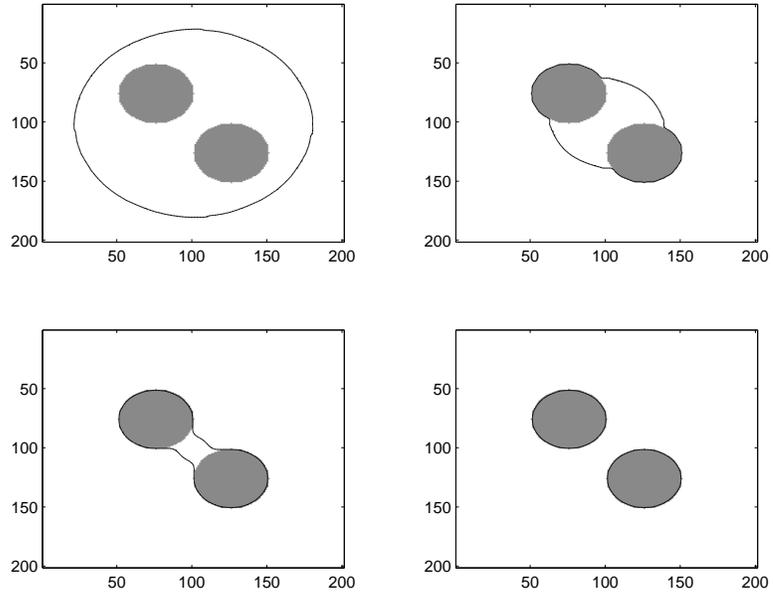


Figure 4.2: Segmentation of 2 circles without topology preservation.

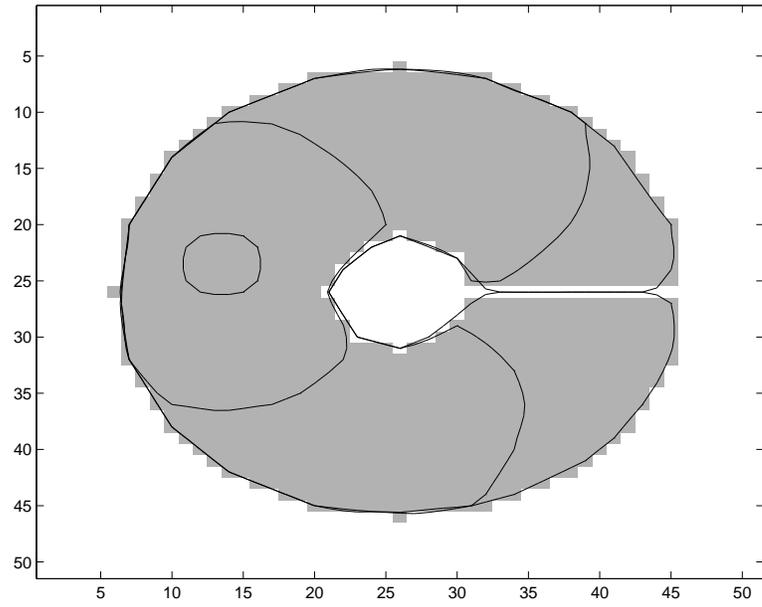


Figure 4.3: Slit annulus segmentation with topology preservation.

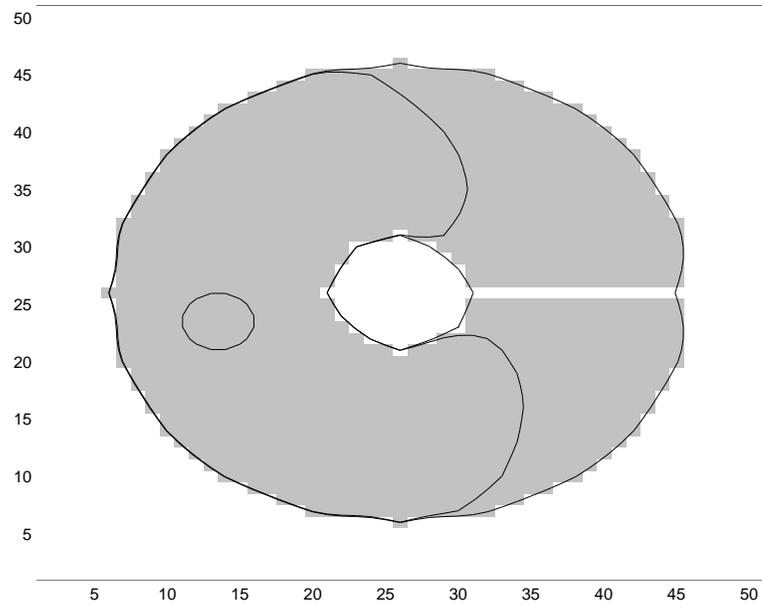


Figure 4.4: Slit annulus segmentation without topology preservation.

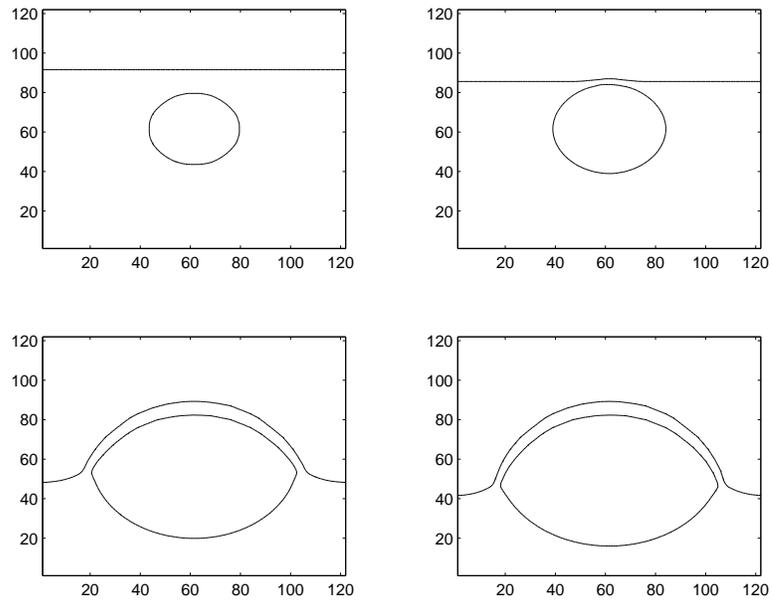


Figure 4.5: Circle and line under outward normal motion with topology preservation.

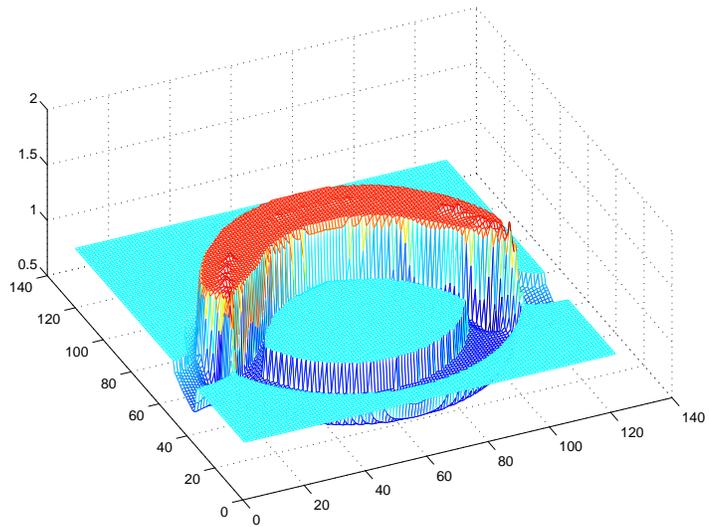


Figure 4.6: Final Jacobian of circle and line under outward normal motion.

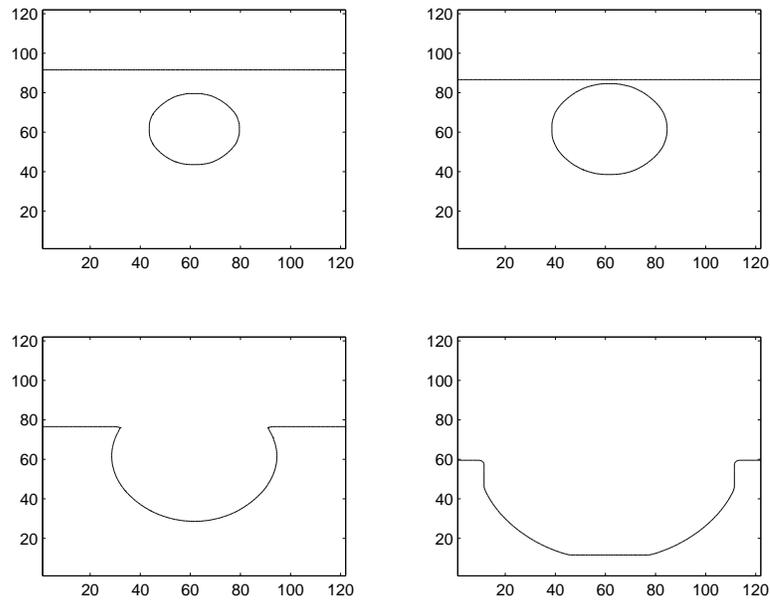


Figure 4.7: Circle and line under outward normal motion without topology preservation.

4.6 Conclusion

The results of this topologically preserving level set method indicate its potential for various applications involving interacting, semi-rigid interface dynamics. While the results are not as resolved as those of [HXP01], they do yield an accurate estimate on the Jacobian of the interface function and its conjugates. This Jacobian has both theoretical and physical significance, and it remains to be seen how it can be used to gain further knowledge of the topological properties of the interface. The local nature of this method as well as that in [HXP01] is the real advantage over all global methods that may require information about the integral of curvature over the interface, or other global properties whose precise measurement may require resolving subgrid phenomena.

In the future we hope to extend our method to 3d applications of brain map-

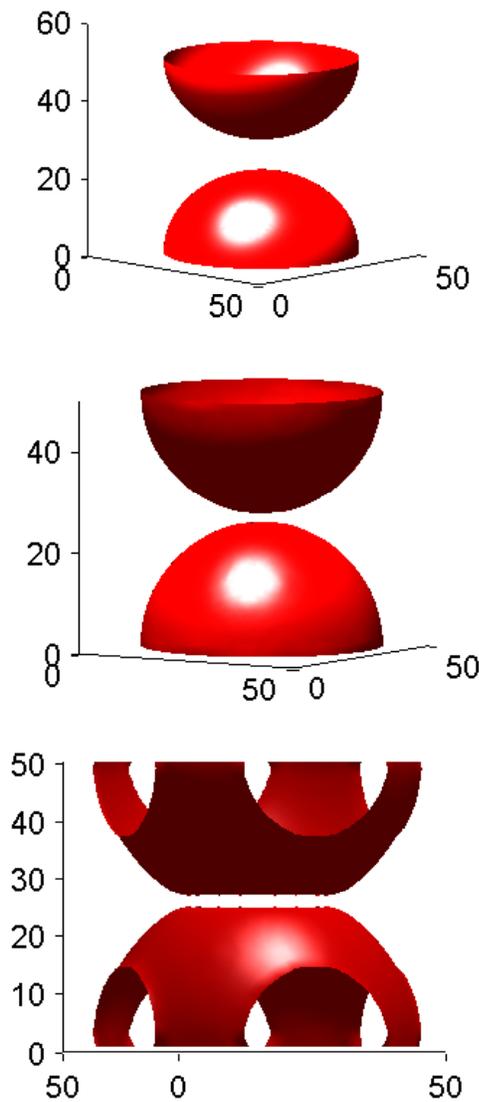


Figure 4.8: Hemispheres under outward normal motion with topology preservation.

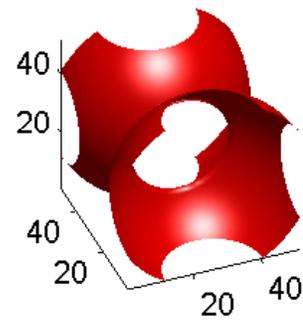
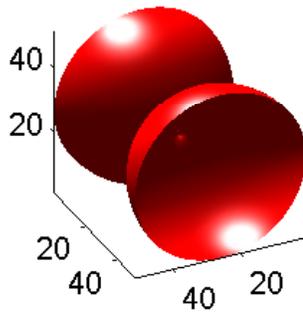
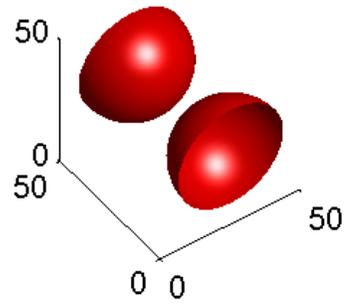


Figure 4.9: Hemispheres under outward normal motion without topology preservation.

ping where a cortex is deformed to a sphere without change in initial topology. Also, this method could be incorporated into sphere or quasi-sphere packing problems where multiple deformable objects are moved in a way that minimizes a functional of volume or surface area.

CHAPTER 5

Fast Numerical Solution to Poisson's Equation for Layered Semiconductor Devices

5.1 Introduction to the Problem

The development of quantum algorithms exploiting quantum parallelism and solving some difficult problems such as prime factorization of large numbers and allowing for rapid searches of databases has led engineers to begin the task of constructing a quantum computer. Eli Yablonovitch's group at UCLA and Mark Gyure's group at HRL have teamed with members of the UCLA math department to work towards the goal of constructing a quantum repeater that can monitor and manipulate the spin of an electron. This in turn will be used for quantum communication. Using a semiconductor hetero-structure the group hopes to build a device with electrostatically controlled potential wells; wells that will trap a single electron and wells that will create 1d quantum wires that can be used to monitor the spin of these electrons[VYW00]. Our job as the mathematicians is to build a numerical simulation of the proposed device that will enable one to determine designs yielding workable devices.

The required equations to be solved are a coupled system determining poten-

tial energy ϕ and wavefunction ψ ,

$$-\nabla \cdot (\epsilon \nabla \phi) = \rho \quad (5.1)$$

$$-(\hbar/2)\nabla \cdot (1/m)\nabla \psi_l + (\tilde{\phi} - E_l)\psi_l = 0, \quad (5.2)$$

where $\rho = f(\{E_j\}, \{\psi_j\})$ which represents a source doping term and a source contribution due to the occupied states ψ_j and energies E_j of trapped electrons, and $\tilde{\phi}$ is the difference between ϕ and the band offset potential. Therefore an elliptic potential problem (5.1) must be solved self consistently with an eigenvalue problem (5.2).

An important part of the solution process is solving (5.1). This solution presents a number of difficulties. Resolving discontinuous coefficients near the very thin layers of the wells is a problem. To capture the behavior of the solution accurately requires a dense set of data points near these wells. In 3d the number of data points required becomes very large, slowing down the solution procedure. Finally, there is the problem of infinite domain boundary conditions which must be imposed within the context of a finite computation.

In section 5.2 we derive a high order solver for (5.1). This solver consists of both standard Fourier methods combined with a new 1d nonhomogeneous Helmholtz solver. The solver can be thought of as a finite element method where the elements are exact solutions of the underlying PDE [SF73],[Cia02]. There have been similar finite element-like methods constructed for homogeneous problems. These are known as partition of unity methods [MB96] or kernel approximations [LJZ95],[Mon82],[BKO96], and have a similar goal of adjusting the elements used to closely match the analytic solution or exactly reproduce a specific class of functions.

This solver will be shown to have arbitrarily high order accuracy (depending only on the smoothness of the wavefunction ψ) in the L^∞ norm, as opposed to

the usual Sobolev norm estimates. The ability of this method to adapt to different boundary conditions and discontinuous material constants without losing accuracy and without substantial change in derivation is also shown. Finally, numerical examples of both the 1d solver and the coupled system (5.1) and (5.2) will be shown, demonstrating the advantage over traditional finite element and finite difference methods in both storage, time, and accuracy.

5.2 High Order Potential Solver

The general idea behind the method of solving (5.1) is to assume a separable solution of the form

$$\phi(x, y, z) = \gamma(x)\eta(y, z)$$

and then solve for ϕ by using a 2d Fourier method for η and a 1d solver for γ . In the y and z directions of the device the dielectric constant ϵ is constant and the solution is not localized as it is in the x direction, so we take a uniform grid in those directions. Assuming an ansatz solution of the form

$$\phi(x, y, z) = \sum_{k_1, k_2} \gamma_{k_1, k_2}(x) e^{ik_1 y} e^{ik_2 z},$$

as well as a nonhomogeneous term that can be expressed in a 2d Fourier basis as

$$f(x, y, z) = \sum_{k_1, k_2} f_{xk_1, k_2}(x) e^{ik_1 y} e^{ik_2 z},$$

then applying (5.1) we get a decoupled system consisting of equations

$$-\gamma''_{k_1, k_2}(x) - (k_1^2 + k_2^2)\gamma(x) = f_{xk_1, k_2}(x) \quad (5.3)$$

for each pair of wavenumbers $\{k_1, k_2\}$. Note that $f(x, y, z) \approx \rho(x, y, z)/\epsilon(x)$ now contains the information about ϵ .

Thus we have a 1d ODE (5.3) to solve for each pair of wavenumbers. If we tried using finite differences to solve this we would reduce the accuracy of the solver to the order of the largest mesh size in x . Away from the wells this mesh size will be many orders of magnitude larger than the mesh size of x in the refined areas inside the wells. Expanding on the ideas of Wachspres [Wac60] we will treat each grid cell as a boundary value problem which can be solved exactly on its interior and then coupled together with its neighbors through continuity conditions. In this way our solver can be viewed as a finite element method using exact solutions as basis functions.

For each grid cell $[x_i, x_{i+1}]$ in x we set up a 2 point Helmholtz BVP derived from (5.3)

$$\frac{d^2 \gamma_{k_1, k_2}^i(x)}{dx^2} + C_{k_1, k_2} \gamma_{k_1, k_2}^i(x) = f_{k_1, k_2}^i(x) \text{ for } x_i \leq x \leq x_{i+1} \quad (5.4a)$$

$$\text{with } \gamma^i \text{ and } \gamma^{i'} \text{ given at } x_i, x_{i+1}, \quad (5.4b)$$

where C_{k_1, k_2} depends on the Fourier modes in y and z . The homogeneous solution of this Helmholtz problem consists of a sum of exponentials which can be coupled using the boundary conditions (5.4b) to form a tridiagonal system of equations giving analytic accuracy when solved.

5.2.1 Solution of the One Dimensional BVPs

In this section we will derive the general solution to a coupled system of linear ODE BVPs of which (5.4a) is a second order Helmholtz example.

General Nonhomogeneous Solution Here we describe the 2 point BVP solution to a nonhomogeneous linear ODE of the form

$$L(u) \equiv \sum_{i=0}^M a_i u^{(i)} = f(x). \quad (5.5)$$

Our goal will be to solve (5.5) for polynomial f , without worrying about the homogeneous solution (which can be added later using superposition). First let us rewrite (5.5) as

$$L(u) = u + \sum_{i=1}^M b_i u^{(i)} = u + \hat{L}(u) = f(x),$$

where $b_i = a_i/a_0$. We assume $a_0 \neq 0$, but if it does then we simply make a change of variables and let $z = u^{(m)}$, where m is the lowest order term with $a_m \neq 0$, then proceed with z in place of u . Say we can approximate $f(x)$ in terms of a polynomial basis, then if we can solve

$$L(u) = x^n,$$

then we can take a superposition of solutions for all n to solve for our polynomial approximation to f .

We begin the trial solution by letting $u = x^n$. If $L(x^n) = x^n$ then we are done. However, if $L(x^n) \neq x^n$ then we must continue to search for a solution. The key point is that $\hat{L}(x^n) = O(x^{m < n})$, so if we alter u to be

$$u = x^n - \hat{L}(x^n),$$

then we can test again to see if

$$L(u) = [x^n + \hat{L}(x^n)] + [-\hat{L}(x^n) + \hat{L}(-\hat{L}(x^n))] = x^n + \hat{L}(-\hat{L}(x^n)) = x^n.$$

If this is not satisfied we repeat the process, setting

$$u = x^n - \hat{L}(x^n) - \hat{L}(-\hat{L}(x^n)).$$

We continue this process until we get a solution for u such that $L(u) = x^n$, so that the final solution can be written as

$$u(x) = x^n + \sum_{i=1}^{\infty} (-\hat{L})^i(x^n), \tag{5.6}$$

where

$$(-\hat{L})^i(g) = \underbrace{(-\hat{L}) \circ \cdots \circ (-\hat{L})}_{i \text{ times}}(g).$$

Note that there exists an $M > 0$ such that $(-\hat{L})^i(g) = 0$ for all $i \geq M$.

Thus the solution for (5.5) with polynomial f is just a superposition of these monomial solutions.

For example, if

$$L(u) = au'' + bu' + u = x^2,$$

then

$$\hat{L}(x^2) = 2a + 2bx, \text{ and } \hat{L}(-\hat{L}(x^2)) = -2b^2.$$

So

$$u(x) = x^2 - (2a + 2bx) - (-2b^2).$$

In the next section we will use this technique applied to a coupled system of Helmholtz BVPs.

Helmholtz Solution In this section we will specifically derive the solution to the coupled system of Helmholtz BVPs of which (5.4a) is an example. For the homogeneous case and the case with constant nonhomogeneous term we follow the derivation in [Wac60].

For a given node x_i we define 2 ODEs on cells adjacent to i . If we define

$$A \equiv [x_{i-1}, x_i], \quad B \equiv [x_i, x_{i+1}]$$

then the ODE in A is

$$-p_{i-1}y'' + q_{i-1}y = f_{i-1}, \tag{5.7}$$

and in B

$$-p_i y'' + q_i y = f_i. \tag{5.8}$$

Note that for the variable $v = p, q$, or f , the notation v_i does not indicate a pointwise value $v(x_i)$, but rather a function $v_i(x)$ on the interval (x_i, x_{i+1}) . For our application we will have $p, q \geq 0$ and constant on intervals $[x_j, x_{j+1}]$, but allow the nonhomogeneous term f to vary. However, for the solution to the ODE, y , we will use the notation that $y_i \equiv y(x_i)$.

The boundary conditions are

$$y(x_{i-1}) = y_{i-1}, \quad y(x_i) = y_i, \quad y(x_{i+1}) = y_{i+1}. \quad (5.9)$$

Let us examine the problem in region B . First we will make the transformation,

$$z = qy - f,$$

so that

$$z' = qy' \text{ and } z'' = qy'',$$

assuming that f is constant. Setting $q/p = \gamma$, (5.8) becomes

$$-z'' + \gamma z = 0. \quad (5.10)$$

This ODE has solutions of the form

$$z_B(x) = ae^{\sqrt{\gamma}x} + be^{-\sqrt{\gamma}x}. \quad (5.11)$$

If $f'' \neq 0$ then we need to adjust z as (5.10) now reads

$$-(z'' + f'') + \gamma z = 0. \quad (5.12)$$

If we make another substitution of

$$w = z - \frac{f''}{\gamma}, \text{ so } w'' = z'' - \frac{f''}{\gamma},$$

and plug this into (5.12) then we get

$$-\left([w'' + \frac{f''}{\gamma}] + f'' \right) + \gamma \left(w + \frac{f''}{\gamma} \right) = 0.$$

If $f^{iv} = 0$ then we solve (5.10) with w in place of z . If $f^{iv} \neq 0$ then we repeat the substitution step again and continue the process until we have that for the M^{th} derivative, $f^M = 0 \forall M > M_0 > 0$. Substituting back to get the solution in terms of y we have

$$\zeta = qy - \sum_{n=0}^{\infty} \frac{f^{(2n)}}{\gamma^n}, \quad (5.13)$$

where ζ solves (5.10). For notational simplicity we will name

$$\hat{f}_i(x) \equiv \sum_{n=0}^{\infty} \frac{f_i^{(2n)}(x)}{\gamma_i^n}.$$

Note that in practice the infinite sum will be truncated as long as we use a finite basis of polynomials to represent $f(x)$. It is this iterative substitution and cancelation of terms that allows us to attain a high order solver by interpolating f using a high order interpolant.

If we let $x_i = 0$ and $x_{i+1} = h_i$ then by matching BCs with (5.11) as our solution for ζ we have

$$\zeta_B(x) = (\zeta_i - \frac{\zeta_{i+1} - \xi_+ \zeta_i}{\xi_- - \xi_+})e^{\sqrt{\gamma_i}x} + (\frac{\zeta_{i+1} - \xi_+ \zeta_i}{\xi_- - \xi_+})e^{-\sqrt{\gamma_i}x}, \quad (5.14)$$

where $\xi_{\pm} = e^{\pm\sqrt{\gamma_i}h_i}$ and $\gamma_i = q_i/p_i$. The solution in region A is found similarly.

Applying the condition of continuity of py' (using $py' = \frac{p}{q}(\zeta' + \hat{f}')$) at x_i we get

$$\lim_{x \rightarrow x_i^-} \frac{p_{i-1}}{q_{i-1}}(\zeta'_A(x) + \hat{f}'_{i-1}(x)) = \lim_{x \rightarrow x_i^+} \frac{p_i}{q_i}(\zeta'_B(x) + \hat{f}'_i(x)). \quad (5.15)$$

Letting $h_{i-1} = x_i - x_{i-1}$ and writing (5.15) in terms of hyperbolic functions we have

$$\begin{aligned} & -\sqrt{\frac{p_i}{q_i}}\zeta_i^+ \coth\left(\sqrt{\frac{q_i}{p_i}}h_i\right) + \sqrt{\frac{p_i}{q_i}}\zeta_{i+1} \operatorname{csch}\left(\sqrt{\frac{q_i}{p_i}}h_i\right) + \frac{p_i}{q_i}\hat{f}'_i(x_i) \\ &= \sqrt{\frac{p_{i-1}}{q_{i-1}}}\zeta_i^- \coth\left(\sqrt{\frac{q_{i-1}}{p_{i-1}}}h_{i-1}\right) - \sqrt{\frac{p_{i-1}}{q_{i-1}}}\zeta_{i-1} \operatorname{csch}\left(\sqrt{\frac{q_{i-1}}{p_{i-1}}}h_{i-1}\right) + \frac{p_{i-1}}{q_{i-1}}\hat{f}'_{i-1}(x_i). \end{aligned} \quad (5.16)$$

Plugging in

$$\begin{aligned}\zeta_i^+ &= q_i y_i - \hat{f}_i(x_i), \zeta_i^- = q_{i-1} y_i - \hat{f}_{i-1}(x_i), \\ \zeta_{i-1} &= q_{i-1} y_{i-1} - \hat{f}_{i-1}(x_{i-1}), \zeta_{i+1} = q_i y_{i+1} - \hat{f}_i(x_{i+1}),\end{aligned}\quad (5.17)$$

and collecting terms we get

$$\begin{aligned}& y_{i-1} \left(\sqrt{\frac{p_{i-1}}{q_{i-1}}} q_{i-1} \operatorname{csch} \left(\sqrt{\frac{q_{i-1}}{p_{i-1}}} h_{i-1} \right) \right) + y_{i+1} \left(\sqrt{\frac{p_i}{q_i}} q_i \operatorname{csch} \left(\sqrt{\frac{q_i}{p_i}} h_i \right) \right) \\ & + y_i \left(-\sqrt{\frac{p_i}{q_i}} q_i \operatorname{coth} \left(\sqrt{\frac{q_i}{p_i}} h_i \right) - \sqrt{\frac{p_{i-1}}{q_{i-1}}} q_{i-1} \operatorname{coth} \left(\sqrt{\frac{q_{i-1}}{p_{i-1}}} h_{i-1} \right) \right) \\ &= -\sqrt{\frac{p_i}{q_i}} \operatorname{coth} \left(\sqrt{\frac{q_i}{p_i}} h_i \right) \hat{f}_i(x_i) + \sqrt{\frac{p_i}{q_i}} \operatorname{csch} \left(\sqrt{\frac{q_i}{p_i}} h_i \right) \hat{f}_i(x_{i+1}) - \frac{p_i}{q_i} \hat{f}'_i(x_i) \\ & \quad - \sqrt{\frac{p_{i-1}}{q_{i-1}}} \operatorname{coth} \left(\sqrt{\frac{q_{i-1}}{p_{i-1}}} h_{i-1} \right) \hat{f}_{i-1}(x_i) \\ & \quad + \sqrt{\frac{p_{i-1}}{q_{i-1}}} \operatorname{csch} \left(\sqrt{\frac{q_{i-1}}{p_{i-1}}} h_{i-1} \right) \hat{f}_{i-1}(x_{i-1}) + \frac{p_{i-1}}{q_{i-1}} \hat{f}'_{i-1}(x_i).\end{aligned}\quad (5.18)$$

It should be noted that if we are on a uniform mesh and f is constant, then by taking the first order Taylor series approximations to the hyperbolic functions we attain the second order central finite difference approximation to the solution.

5.2.2 Boundary Conditions

An added benefit of this type of solution method is the ease with which boundary conditions of varying types can be imposed. The case of infinite BCs is especially relevant for our particular physical problem, a case that can cause difficulty to finite difference approximations.

For infinite BCs if we have the lower domain boundary as $x_i = 0$ with an infinite boundary for $x < x_i$, then the solution to (5.10) is

$$\zeta_{-\infty} = \zeta_i e^{\sqrt{\gamma} x} \quad (5.19)$$

where $\gamma = q_{-\infty}/p_{-\infty}$.

Writing an equation that is analogous to (5.15) in terms of hyperbolic functions we have

$$\begin{aligned}
& -\sqrt{\frac{p_i}{q_i}}\zeta_i^+ \coth\left(\sqrt{\frac{q_i}{p_i}}h_i\right) + \sqrt{\frac{p_i}{q_i}}\zeta_{i+1} \operatorname{csch}\left(\sqrt{\frac{q_i}{p_i}}h_i\right) + \frac{p_i}{q_i}\hat{f}'_i(x_{i+}) \\
& = \sqrt{\frac{p_{-\infty}}{q_{-\infty}}}\zeta_i^- + \frac{p_{-\infty}}{q_{-\infty}}\hat{f}'_{-\infty}(x_{i-}).
\end{aligned} \tag{5.20}$$

In the same way we derived (5.18) by matching py' at x_i we can derive

$$\begin{aligned}
& y_{i+1} \left(\sqrt{\frac{p_i}{q_i}}q_i \operatorname{csch}\left(\sqrt{\frac{q_i}{p_i}}h_i\right) \right) \\
& + y_i \left(-\sqrt{\frac{p_i}{q_i}}q_i \coth\left(\sqrt{\frac{q_i}{p_i}}h_i\right) - \sqrt{\frac{p_{-\infty}}{q_{-\infty}}}q_{-\infty} \right) \\
& = -\sqrt{\frac{p_i}{q_i}} \coth\left(\sqrt{\frac{q_i}{p_i}}h_i\right) \hat{f}_i(x_i) + \sqrt{\frac{p_i}{q_i}} \operatorname{csch}\left(\sqrt{\frac{q_i}{p_i}}h_i\right) \hat{f}_i(x_{i+1}) \\
& \quad - \frac{p_i}{q_i} \hat{f}'_i(x_i) - \sqrt{\frac{p_{-\infty}}{q_{-\infty}}}\hat{f}'_{-\infty}(x_i) + \frac{p_{-\infty}}{q_{-\infty}}\hat{f}'_{-\infty}(x_i).
\end{aligned} \tag{5.21}$$

The boundary equation for infinite BCs at the right side are derived similarly but with (5.19) replaced with

$$\zeta_{+\infty} = \zeta_i e^{-\sqrt{\gamma}x}, \tag{5.22}$$

where $\gamma = q_{+\infty}/p_{+\infty}$.

For Robin BCs (which include Neumann and Dirichlet) we proceed along similar lines. Given left side BCs at $x_i = 0$,

$$\alpha_1 y_L - \alpha_2 p_L y'_L = \alpha_3, \tag{5.23}$$

where $y'_L = \lim_{x \rightarrow x_i+} y'(x)$, then in the region $[x_i, x_{i+1}]$ we have

$$\alpha_1 y_i - \alpha_2 \frac{p_L}{q_i} (\zeta'_i + \hat{f}'_i)|_{x=x_i} = \alpha_3$$

as $y = \frac{\zeta + f}{q}$. Substituting and rearranging yields

$$\begin{aligned}
& y_i \left[\alpha_1 + \alpha_2 \frac{p_L}{q_i} \sqrt{\frac{p_i}{q_i}} q_i \coth \left(\sqrt{\frac{q_i}{p_i}} h_i \right) \right] \\
& + y_{i+1} \left[-\alpha_2 \frac{p_L}{q_i} \sqrt{\frac{p_i}{q_i}} q_i \operatorname{csch} \left(\sqrt{\frac{q_i}{p_i}} h_i \right) \right] \\
& = \alpha_3 - \hat{f}'_i(x_i) + \alpha_2 \frac{p_L}{q_i} \sqrt{\frac{p_i}{q_i}} \hat{f}_i(x_i) \coth \left(\sqrt{\frac{q_i}{p_i}} h_i \right) \\
& \quad - \alpha_2 \frac{p_L}{q_i} \sqrt{\frac{p_i}{q_i}} \hat{f}_i(x_{i+1}) \operatorname{csch} \left(\sqrt{\frac{q_i}{p_i}} h_i \right). \tag{5.24}
\end{aligned}$$

Robin BCs for the right side boundary are derived similarly.

5.2.3 Accuracy

Locally we are exactly solving an ODE of the form

$$L(u_{num}) = g(x),$$

where g is an approximation to the actual nonhomogeneous term $f(x)$, and u_{num} indicates the numerical solution. Thus to get an estimate on $\|u - u_{num}\|$ we can look at

$$L(u(x)) - L(u_{num}(x)) = -(u''(x) - u''_{num}(x)) + \alpha(u(x) - u_{num}(x)) = f(x) - g(x).$$

Say that on a given interval $D = [x_0, x_1]$ we have that $w(x) = u(x) - u_{num}(x)$ for $x \in D$, with BCs given by $w = h(x)$, and that $|f - g|_\infty$ is of order ϵ . Then we have that

$$w'' - \alpha w = g - f \text{ in } D.$$

As long as $\alpha \geq 0$ and w is sufficiently smooth on D then we have an a priori bound on the error [McO95],

$$\max_{x \in D} |w(x)| \leq \max_{x \in \partial D} |h(x)| + C \max_{x \in D} |(g - f)(x)| \leq \epsilon(1 + C), \tag{5.25}$$

where C depends only on the domain, D .

Thus it is only the interpolation error that dictates the error of our method, and this is in turn dictated by the smoothness of f . For our particular quantum modeling problem f represents the Fourier transform of the wave function, ψ , which should be smooth. Also, in regions where $f = 0$ we see that no mesh points are needed to resolve the solution.

5.2.4 Implementation Details

To solve the potential equation we can assume that we have been given the nonhomogeneous term, ρ in (5.1), at all data points. We then take the Fourier transform of (5.1) giving a 1D system of coupled Helmholtz problems of the form (5.4a) for each set of paired wave numbers $\{k_1, k_2\}$ on a set of data points $X \equiv \{x_{j=1:N}\}$.

We will then solve the system on X with BCs specified as in either infinite or as in (5.23). We will set up a tridiagonal system of equations defined by (5.18) for interior points, and either (5.21) or (5.24). To determine $\hat{f}_i(x)$ we will use polynomial interpolation over points $\{x_j\}$ where the diffusivity constant $\epsilon(x_j)$ is the same as $\epsilon(x_i)$. Any suitable polynomial basis can be used.

5.3 Numerical Examples

We begin by showing some results from the full 3d solution to (5.1) and (5.2). Figure 5.1 shows a sample quantum wire position within the well in layer C, as well as the mesh on which we compute the solution. Note the coarse grid in regions A and B away from the well. In figure 5.3 we show a sample level set of the potential ϕ , and in figure 5.2 we show level sets of the wavefunction ψ in

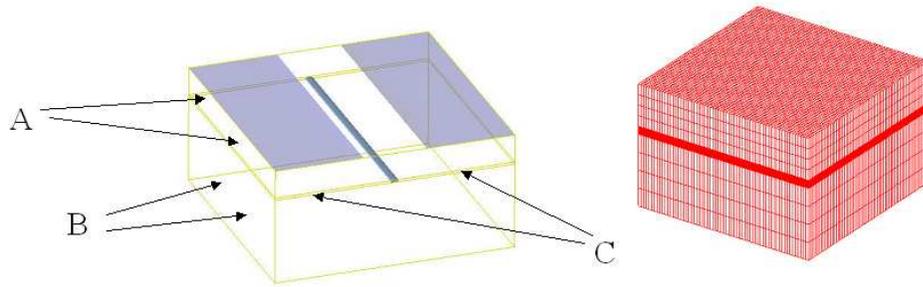


Figure 5.1: 3d quantum wire and sample mesh.

2 different wells, one which will hold the quantum dot and the other which will hold the quantum wire. Figure 5.4 shows a slice of the potential for fixed y and z through the center of the device.

For the 1d Helmholtz solver we will show both error results and convergence order estimates, as well as some sample solutions. The computational domain will be $[0, 1]$ with a uniform grid of 101 points unless noted otherwise.

Problem 1 Figure 5.5 shows the computed and exact solution to the BVP

$$-y'' + y = 0, \quad y(0) = 0, \quad y(1) - y'(1) = 1.$$

The exact solution is

$$y(x) = \frac{e}{2}(e^{-x} - e^x).$$

Note that the approximation and exact solution are indistinguishable, and in fact the L^∞ error $\approx 10^{-12}$ which is approaching machine ϵ .

Problem 2 Figure 5.6 shows the computed and exact solution to the BVP

$$-y'' + y = \sin(2\pi x)(1 + (2\pi)^2), \quad y(0) = 0, \quad y(1) = 0$$

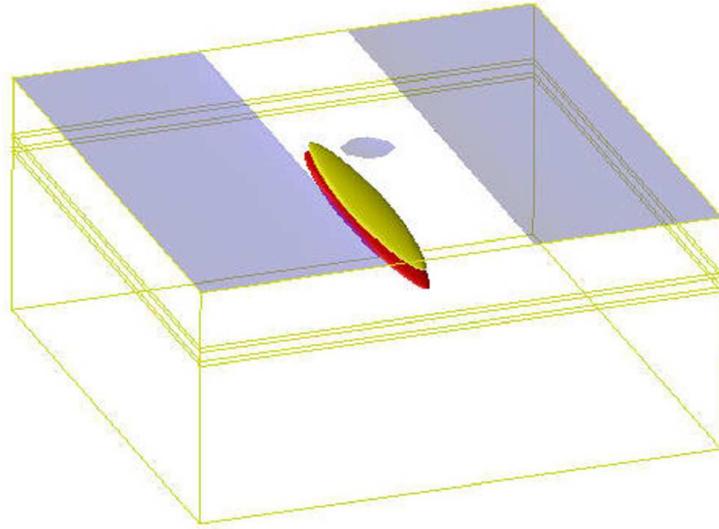


Figure 5.2: Level sets of ψ in the upper and lower wells.

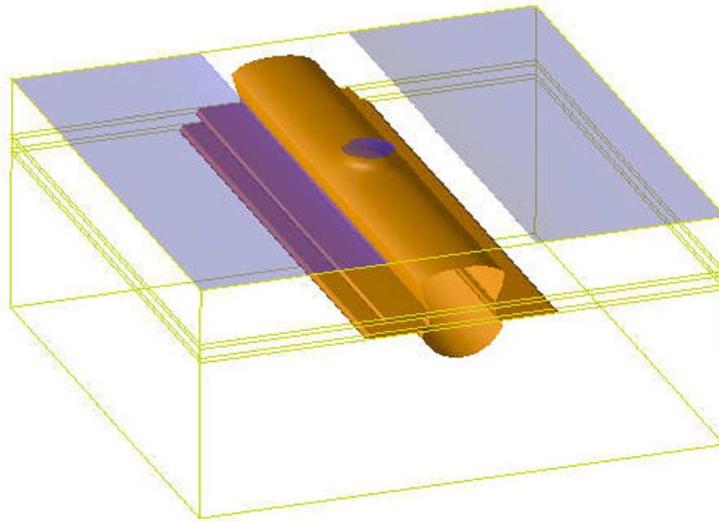


Figure 5.3: Potential level set.

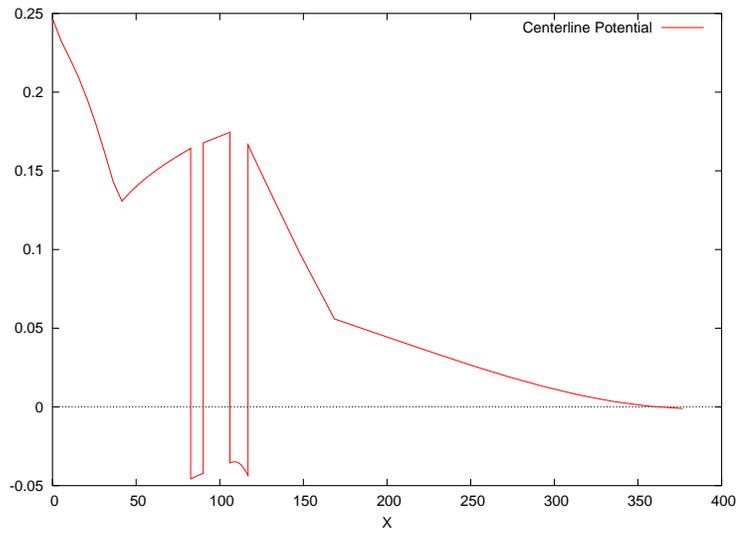


Figure 5.4: 1d potential slice for fixed y and z .

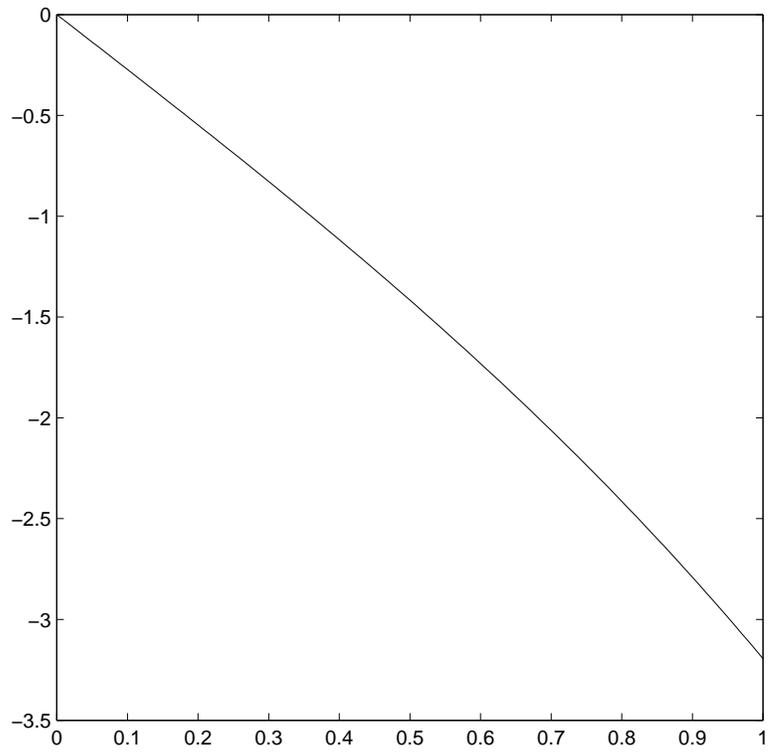


Figure 5.5: Problem 1 computed and exact solution.

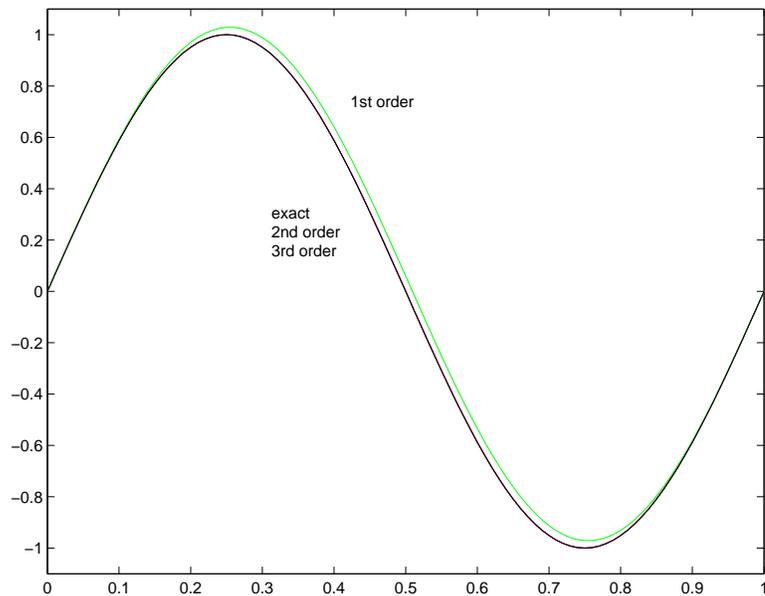


Figure 5.6: Problem 2 computed and exact solution.

using different order polynomial approximations to the nonhomogeneous term.

The exact solution is

$$y(x) = \sin(2\pi x).$$

In figure 5.7 we magnify the region where there is the largest difference in the 2nd and 3rd order solutions to show their accuracy.

Problem 3 Figure 5.8 shows the computed solution using 1601 grid points to the BVP

$$-p(x)y'' + q(x)y = f(x), \quad y(0) = 0, \quad y(1) = 1,$$

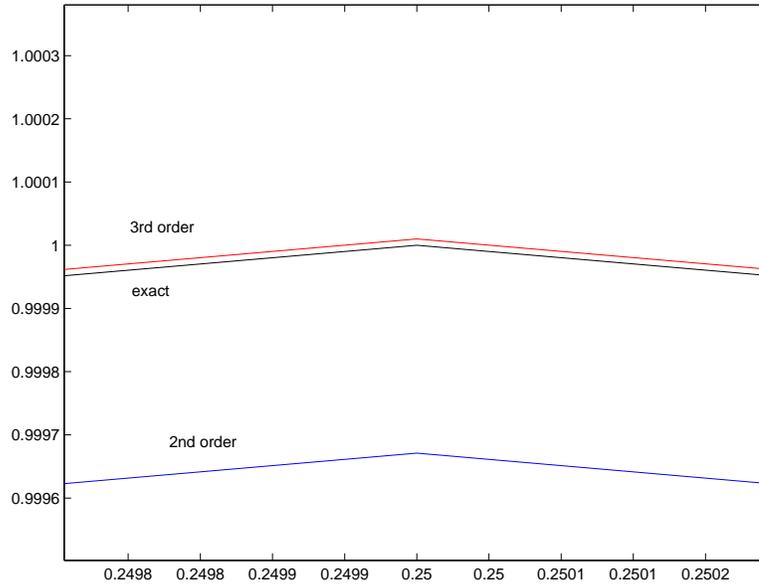


Figure 5.7: Problem 2 computed and exact solution magnified.

where

$$p = 1, q = 10 \text{ on } [0, 0.3], \quad (5.26)$$

$$p = 1, q = 1 \text{ on } (0.3, 0.7], \quad (5.27)$$

$$p = 20, q = 1 \text{ on } (0.7, 1], \quad (5.28)$$

$$f(x) = 100 \cos(6\pi x) \text{ on } [0, 1]. \quad (5.29)$$

Using Richardson extrapolation we can estimate the rate of convergence of the approximations when using different polynomial orders interpolating f . Figures 5.9, 5.10, and 5.11 show the pointwise convergence order estimate when 1st, 2nd and 3rd order interpolation is used, respectively. For smooth regions of f the methods indicate 1st, 2nd and 3rd order convergence. Note that in regions where f is not smooth Richardson extrapolation does not yield a good estimate on the convergence rate. This can be seen in figure 5.12 which shows the 3rd order convergence estimate along with p, q and $f(x)/5$. It becomes obvious that both

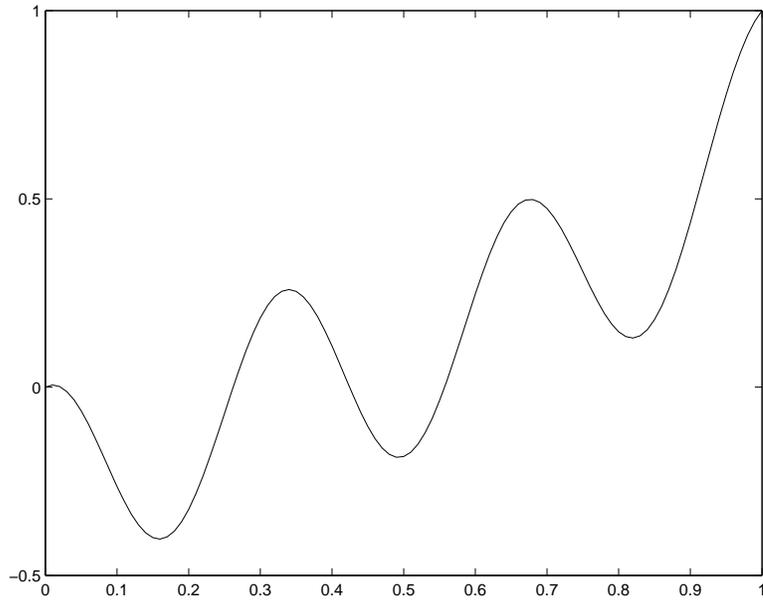


Figure 5.8: Problem 3 exact solution.

the magnitudes and discontinuities of p and q , as well as the nonsmooth regions of f , contribute to losses in accuracy.

5.4 Conclusion

As indicated earlier there is still much work to do on the full coupled system (5.1) and (5.2) as well. The requested design of the devices also changes often and new problems such as 3d gate geometry and curvature effects are being introduced on a regular basis giving the model greater complexity.

The success of the 1d nonhomogeneous solver begs the question of whether other ODE problems can be solved in the same way. ODEs with spatially varying coefficients can also be examined, but their particular solutions for specific nonhomogeneous terms will become more complex and will not always be solvable in

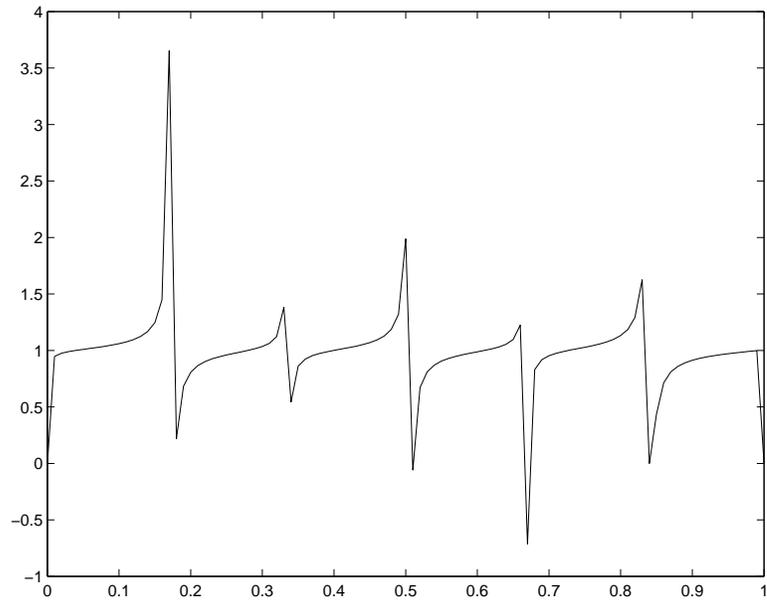


Figure 5.9: Problem 3 Richardson extrapolation, 1st order.

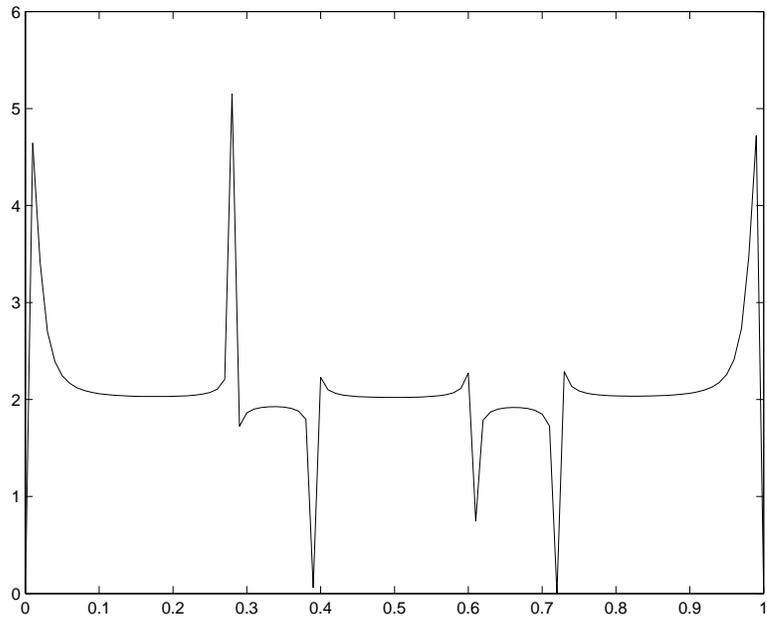


Figure 5.10: Problem 3 Richardson extrapolation, 2nd order.

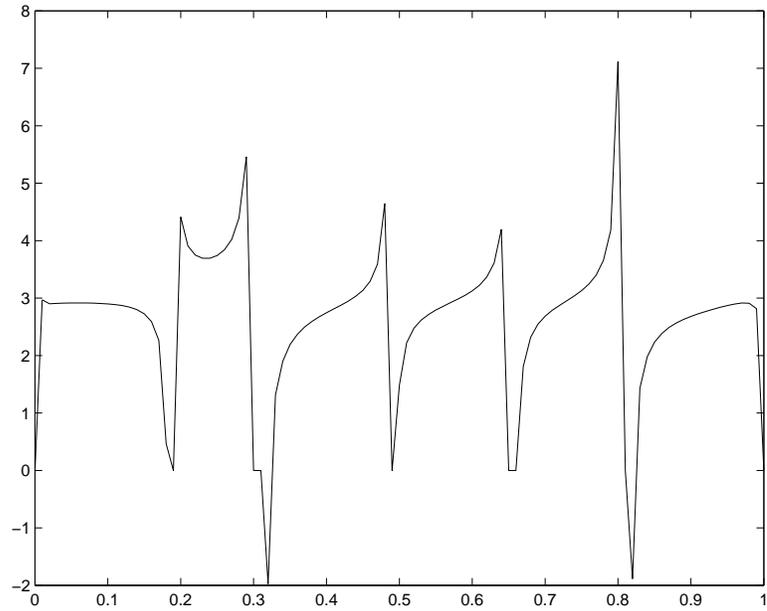


Figure 5.11: Problem 3 Richardson extrapolation, 3^{rd} order.

closed form. For hyperbolic conservation laws the Godunov solvers for Riemann problems are an example of a similar concept, except that the characteristics travel at finite speeds allowing for explicit solvers. The generalization to higher dimensions needs to be explored further, presenting nonhomogeneous boundary value PDE problems to be solved exactly, again calling for analytical manipulation. Also, applying this technique to the eigenvalue problem (5.2) may yield a more accurate solution.

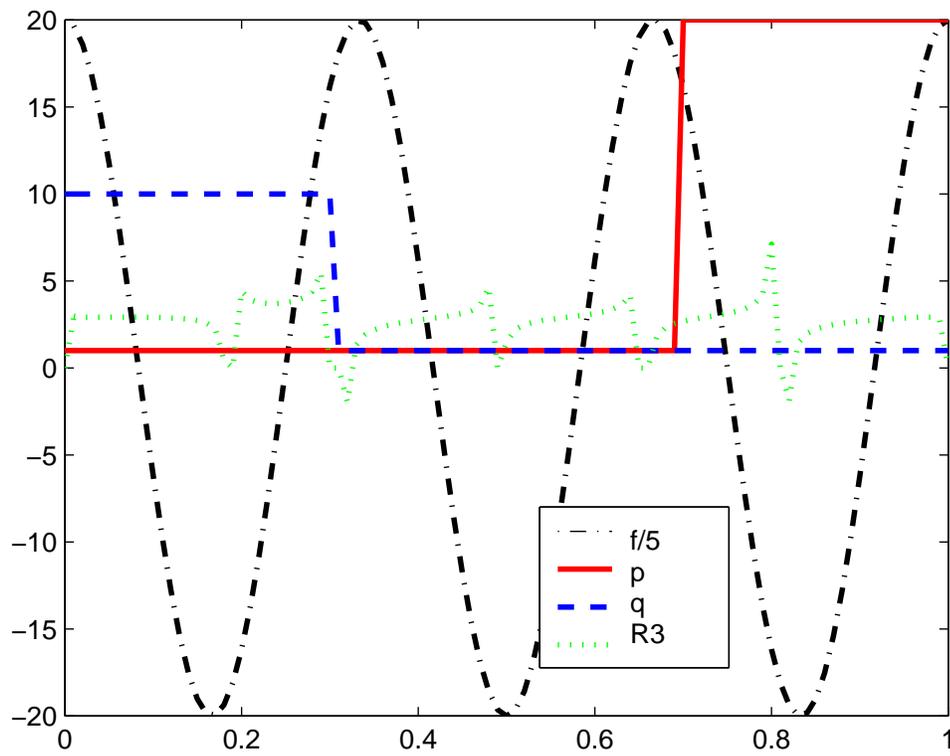


Figure 5.12: Problem 3, 3rd order extrapolation with $f/5, p, q$.

REFERENCES

- [AA00] Steeve Augoula and Rémi Abgrall. “High order numerical discretization for Hamilton-Jacobi equations on triangular meshes.” *J. Sci. Comput.*, **15**(2):197–229, 2000.
- [Abg94] R. Abgrall. “On essentially non-oscillatory schemes on unstructured meshes: analysis and implementation.” *J. Comp. Phys.*, **114**(1):45–58, 1994.
- [Abg96] R. Abgrall. “Numerical discretization of the first-order Hamilton-Jacobi equation on triangular meshes.” *Comm. Pure Appl. Math.*, **49**(12):1339–1373, 1996.
- [BCM01] Paul Burchard, Li-Tien Cheng, Barry Merriman, and Stanley Osher. “Motion of curves in three spatial dimensions using a level set approach.” *J. Comput. Phys.*, **170**(2):720–741, 2001.
- [BKO96] T Belytschko, Y Krongauz, D Organ, M Fleming, and P Krysl. “Meshless methods: An overview and recent developments.” *Comp. Meth. Appl. Mech. Eng.*, **139**:3–47, 1996.
- [BS98] Timothy J. Barth and James A. Sethian. “Numerical schemes for the Hamilton-Jacobi and level set equations on triangulated domains.” *J. Comput. Phys.*, **145**(1):1–40, 1998.
- [CEL84] M. G. Crandall, L. C. Evans, and P.-L. Lions. “Some properties of viscosity solutions of Hamilton-Jacobi equations.” *Trans. Amer. Math. Soc.*, **282**(2):487–502, 1984.
- [CF91] Ralph E. Carlson and Thomas A. Foley. “The parameter R^2 in multi-quadric interpolation.” *Comput. Math. Appl.*, **21**(9):29–42, 1991.
- [Cia02] Philippe G. Ciarlet. *The finite element method for elliptic problems*, volume 40 of *Classics in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2002. Reprint of the 1978 original [North-Holland, Amsterdam; MR 58 #25001].
- [CL84] M. G. Crandall and P.-L. Lions. “Two approximations of solutions of Hamilton-Jacobi equations.” *Math. Comp.*, **43**(167):1–19, 1984.
- [CL02] Russel E. Caffisch and Bo Li. “Analysis of Island Dynamics in Epitaxial Growth of Thin Films.” *UCLA CAM Report*, May 2002.

- [CM80] Michael G. Crandall and Andrew Majda. “Monotone difference approximations for scalar conservation laws.” *Math. Comp.*, **34**(149):1–21, 1980.
- [CS99] David L. Chopp and J. A. Sethian. “Motion by intrinsic Laplacian of curvature.” *Interfaces Free Bound.*, **1**(1):107–123, 1999.
- [CT94] J.W. Cahn and J.E. Taylor. *Acta metall. mater.*, **42**, 1994.
- [CV01] Tony Chan and Luminita Vese. “Active Contours Without Edges.” *IEEE Trans. on Imag. Proc.*, **10**(2), 2001.
- [DGP92] Antonio Di Carlo, Morton E. Gurtin, and Paolo Podio-Guidugli. “A regularized equation for anisotropic motion-by-curvature.” *SIAM J. Appl. Math.*, **52**(4), 1992.
- [Fri98] Oliver Friedrich. “Weighted essentially non-oscillatory schemes for the interpolation of mean values on unstructured grids.” *J. Comput. Phys.*, **144**(1):194–212, 1998.
- [Gib78] J.W. Gibbs. “On the equilibrium of heterogeneous substances.” *Trans. Conn. Acad.*, 1876,1878. Reprinted in *The Scientific Papers of J. Willard Gibbs*, Vol. 1, Dover, New York, 1961.
- [Gur93] Morton Gurtin. *Thermodynamics of Evolving Phase Boundaries in the Plane*. Springer Verlag, 1993.
- [Har71] R.L. Hardy. “Multiquadric equations of topography and other irregular surfaces.” *J. Geophys. Res.*, **76**:1905–1915, 1971. Radial basis functions and their applications.
- [HC91] A. Harten and S. Chakravarthy. “Multi-dimensional ENO schemes for general geometries.” *ICASE Report*, **91-76**, 1991.
- [Her51] C. Herring. “Surface tension as a motivation for sintering.” In W.E. Kingston, editor, *The Physics of Powder Metallurgy*. McGraw-Hill, 1951.
- [HHL76] A. Harten, J. M. Hyman, and P. D. Lax. “On finite-difference approximations and entropy conditions for shocks.” *Comm. Pure Appl. Math.*, **29**(3):297–322, 1976. With an appendix by B. Keyfitz.
- [HO98] Eduard Harabetian and Stanley Osher. “Regularization of ill-posed problems via the level set approach.” *SIAM J. Appl. Math.*, **58**(6):1689–1706 (electronic), 1998.

- [HOE86] Ami Harten, Stanley Osher, Björn Engquist, and Sukumar R. Chakravarthy. “Some results on uniformly high-order accurate essentially nonoscillatory schemes.” *Appl. Numer. Math.*, **2**(3-5):347–377, 1986.
- [HS99] Changqing Hu and Chi-Wang Shu. “Weighted essentially non-oscillatory schemes on triangular meshes.” *J. Comput. Phys.*, **150**(1):97–127, 1999.
- [HXP01] X. Han, C. Xu, and J. Prince. “A Topology Preserving Deformable Model Using Level Sets.” *CVPR2001*, December 2001.
- [IS96] Armin Iske and Thomas Sonar. “On the structure of function spaces in optimal recovery of point functionals for ENO-schemes by radial basis functions.” *Numer. Math.*, **74**(2):177–201, 1996.
- [JP00] Guang-Shan Jiang and Danping Peng. “Weighted ENO schemes for Hamilton-Jacobi equations.” *SIAM J. Sci. Comput.*, **21**(6):2126–2143 (electronic), 2000.
- [LJZ95] Wing Kam Liu, Sukky Jun, and Yi Fei Zhang. “Reproducing kernel particle methods.” *Internat. J. Numer. Methods Fluids*, **20**(8-9):1081–1106, 1995. Finite elements in fluids—new trends and applications (Barcelona, 1993).
- [MB96] J. M. Melenk and I. Babuška. “The partition of unity finite element method: basic theory and applications.” *Comput. Methods Appl. Mech. Engrg.*, **139**(1-4):289–314, 1996.
- [McO95] Robert McOwen. *Partial Differential Equations: methods and applications*. Prentice Hall, Inc., 1995.
- [Mey82] Richard E. Meyer. *Introduction to mathematical fluid dynamics*. Dover Publications Inc., New York, 1982. Corrected reprint of the 1971 original.
- [Mic86] C.A. Micchelli. “Interpolation and scattered data: Distance matrices and conditionally positive definite functions.” *Constr. Approx.*, **2**:11–22, 1986.
- [Mon82] J. J. Monaghan. “Why particle methods work.” *SIAM J. Sci. Statist. Comput.*, **3**(4):422–433, 1982.

- [MS89] David Mumford and Jayant Shah. “Optimal approximations by piecewise smooth functions and associated variational problems.” *Comm. Pure Appl. Math.*, **42**(5):577–685, 1989.
- [Mul57] W.W. Mullins. *J. Appl. Phys.*, **28**, 1957.
- [OF01] Stanley Osher and Ronald P. Fedkiw. “Level set methods: an overview and some recent results.” *J. Comput. Phys.*, **169**(2):463–502, 2001.
- [OS88] S.J. Osher and J.A. Sethian. “Fronts propagating with curvature dependent speed: algorithms based on Hamilton-Jacobi formulations.” *Journal of Computational Physics*, **79**:12–49, 1988.
- [OS91] Stanley Osher and Chi-Wang Shu. “High-order essentially nonoscillatory schemes for Hamilton-Jacobi equations.” *SIAM J. Numer. Anal.*, **28**(4):907–922, 1991.
- [OS01] Stanley J. Osher and Fadil Santosa. “Level set methods for optimization problems involving geometry and constraints. I. Frequencies of a two-density inhomogeneous drum.” *J. Comput. Phys.*, **171**(1):272–288, 2001.
- [PMO99] Danping Peng, Barry Merriman, Stanley Osher, Hongkai Zhao, and Myungjoo Kang. “A PDE-based fast local level set method.” *J. Comput. Phys.*, **155**(2):410–438, 1999.
- [POM99] Danping Peng, Stanley Osher, Barry Merriman, and Hong-Kai Zhao. “The geometry of Wulff crystal shapes and its relations with Riemann problems.” In *Nonlinear partial differential equations (Evanston, IL, 1998)*, volume 238 of *Contemp. Math.*, pp. 251–303. Amer. Math. Soc., Providence, RI, 1999.
- [Rip99] Shmuel Rippa. “An algorithm for selecting a good value for the parameter c in radial basis function interpolation.” *Adv. Comput. Math.*, **11**(2-3):193–210, 1999. Radial basis functions and their applications.
- [Ros61] J. B. Rosen. “The gradient projection method for nonlinear programming. II. Nonlinear constraints.” *J. Soc. Indust. Appl. Math.*, **9**:514–532, 1961.
- [Sch95] Robert Schaback. “Error estimates and condition numbers for radial basis function interpolation.” *Adv. Comput. Math.*, **3**(3):251–264, 1995.

- [Sch97] Robert Schaback. *Reconstruction of Multivariate Functions from Scattered Data*. 1997. To appear, available at <http://www.num.math.uni-goettingen.de/schaback/>.
- [SF73] Gilbert Strang and George J. Fix. *An analysis of the finite element method*. Prentice-Hall Inc., Englewood Cliffs, N. J., 1973. Prentice-Hall Series in Automatic Computation.
- [Sme] Peter Smereka. “Semi-implicit level set methods for motion by mean curvature and surface diffusion.” *To appear*.
- [SO89] Chi-Wang Shu and Stanley Osher. “Efficient implementation of essentially nonoscillatory shock-capturing schemes. II.” *J. Comput. Phys.*, **83**(1):32–78, 1989.
- [Son96] Thomas Sonar. “Optimal recovery using thin plate splines in finite volume methods for the numerical solution of hyperbolic conservation laws.” *IMA J. Numer. Anal.*, **16**(4):549–581, 1996.
- [Son98] Thomas Sonar. “On families of pointwise optimal finite volume ENO approximations.” *SIAM J. Num. Anal.*, **35**(6):2350–2369, 1998.
- [Str99] John Strain. “Tree methods for moving interfaces.” *J. Comput. Phys.*, **151**(2):616–648, 1999.
- [TBJ01] G. Tryggvason, B. Bunner, D. Juric, W. Tauber, S. Nas, J. Han, N. Al-Rawahi, and Y.-J. Jan. “A front tracking method for the computations of multiphase flow.” *J. Comput. Phys.*, **169**:708–759, 12001.
- [TGP97] A. Trellakis, A. Galick, A. Pacelli, and U. Ravaioli. “Iteration scheme for the solution of the two-dimensional Schrodinger-Poisson equations in quantum structures.” *J. Appl. Phys.*, **81**, 1997.
- [TT02] Paul Thompson and Arthur Toga. “A Framework for Computational Anatomy.” *Computing and Visualization in Science*, **5**, 2002.
- [TWB02] T. Tasdizen, R. Whitaker, P. Burchard, and S. Osher. “Geometric Surface Smoothing via Anisotropic Diffusion of Normals.” *UCLA CAM Report*, October 2002.
- [UT92] S.O. Unverdi and G. Tryggvason. “A front tracking method for viscous incompressible, multi-fluid flows.” *J. Comput. Phys.*, **100**:25–37, 1992.

- [VYW00] R. Vrijen, E. Yablonovitch, K. L. Wang, H. W. Jiang, A. Balandin, V. Roychowdhury, T. Mor, and D. DiVincenzo. “Electron Spin Resonance Transistors for Quantum Computing in Silicon-Germanium Hererostructure.” *Phys. Rev. A* **62**, **012306**, 2000.
- [Wac60] Eugene L. Wachspress. “The numerical solution of boundary value problems.” In *Mathematical methods for digital computers*, pp. 121–127. Wiley, New York, 1960.
- [Wu95] Z.-M. Wu. “Multivariate compactly supported positive definite radial basis functions.” *Adv. Comput. Math.*, **4**:389–396, 1995.
- [ZS02] Yong-Tao Zhang and Chi-Wang Shu. “High-order WENO schemes for Hamilton-Jacobi equations on triangular meshes.” *SIAM J. Sci. Comput.*, **24**(3):1005–1030 (electronic), 2002.
- [ZYK01] Y.L. Zhang, K.S. Yeo, B.C. Khoo, and C. Wang. “3d jet impact of toroidal bubbles.” *J. Comput. Phys.*, **166**:336–360, 2001.