

**THE NUMERICAL SOLUTION OF DIFFERENTIAL EQUATIONS:
GRID SELECTION FOR BOUNDARY VALUE PROBLEMS AND ADAPTIVE
TIME INTEGRATION STRATEGIES**

by

Ronald Dale Haynes

B.Sc. (Hons), Memorial University of Newfoundland, 1996

M.Sc., Simon Fraser University, 1998

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
in the Department
of
Mathematics

© Ronald Dale Haynes 2003
SIMON FRASER UNIVERSITY
March, 2003

All rights reserved. This work may not be
reproduced in whole or in part, by photocopy
or other means, without the permission of the author.

APPROVAL

Name: Ronald Dale Haynes
Degree: Doctor of Philosophy
Title of thesis: The Numerical Solution of Differential Equations: Grid Selection for Boundary Value Problems and Adaptive Time Integration Strategies

Examining Committee: Dr. Rustum Choksi
Chair

Dr. Manfred Trummer, Senior Supervisor

Dr. Robert Russell, Senior Supervisor

Dr. Steven Ruuth

Dr. Mary Catherine Kropinski, SFU Examiner

Dr. Luca Dieci, External Examiner,
Georgia Institute of Technology

Date Approved:

March 31, 2003

PARTIAL COPYRIGHT LICENCE

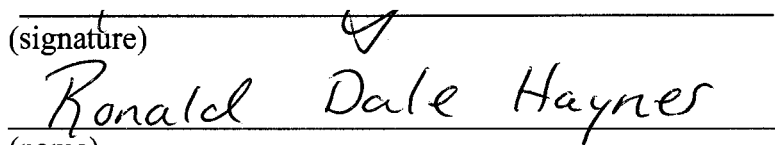
I hereby grant to Simon Fraser University the right to lend my thesis, project or extended essay (the title of which is shown below) to users of the Simon Fraser University Library, and to make partial or single copies only for such users or in response to a request from the library of any other university, or other educational institution, on its own behalf or for one of its users. I further agree that permission for multiple copying of this work for scholarly purposes may be granted by me or the Dean of Graduate Studies. It is understood that copying or publication of this work for financial gain shall not be allowed without my written permission.

Title of Thesis/Project/Extended Essay

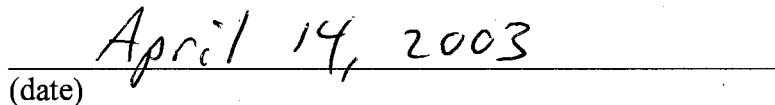
The Numerical Solution of Differential Equations: Grid Selection for Boundary Value Problems and Adaptive Time Integration Strategies

Author:

(signature)

A handwritten signature in cursive script that reads "Ronald Dale Haynes". The signature is written on a horizontal line.

(name)

A handwritten date in cursive script that reads "April 14, 2003". The date is written on a horizontal line.

(date)

Abstract

The numerical solution of differential equations requires selecting an appropriate choice of mesh, spatial and temporal discretization, and algebraic equation solver. No one aspect should be considered in isolation. In the first part of this thesis we consider the issue of appropriate mesh selection for two-point boundary value problems. Specifically, we study how properties of the matrix corresponding to the discrete problem relate to the issue of mesh selection. It is found that the quality of a chosen mesh is identifiable with well-known features of the matrix such as eigenvalues/eigenvectors, and singular values/singular vectors. Moreover, these matrix characteristics may guide us in the construction of more appropriate meshes.

Over the last twenty years there has been much attention paid to numerical methods for differential equations which adapt in either space or time to local features of the computed solution. In the second part of this thesis, we consider the method of lines approach to solving parabolic partial differential equations. Discretizing in space, using either a fixed or moving mesh, results in a system of ordinary differential equations. Traditional implementations solve these equations using classical integration methods with local error control. This approach suffers from an inability to take advantage of the solution evolving at disparate time scales over the spatial domain. To address this issue we consider waveform relaxation and Schwarz waveform relaxation methods which allow individual or groups of solution components to be integrated using different time steps or even entirely different numerical methods. We conclude by proposing a Schwarz Waveform Moving Mesh Method. This implementation combines the robustness of an adaptive spatial mesh with the multi-rate abilities of a relaxation method.

To Angie, Emma, Mom and Dad

Acknowledgments

Completing a PhD is typically not a solitary process, and this thesis has been no exception. My primary thanks has to go to Bob and Manfred, my co-senior supervisors. Their guidance, encouragement and support (financial and otherwise) has made my time at SFU enjoyable and stimulating. To NSERC for two years of funding to allow me to concentrate fully on my research interests. To the Mathematics department staff you have been both extremely helpful and accomodating during my years here. And to those I have mentioned or not *Long may your big jib draw!!* (Live long and prosper!!)

Contents

Approval	ii
Abstract	iii
Dedication	iv
Acknowledgments	v
List of Tables	ix
List of Figures	x
I Boundary Value Problems and Matrix Properties	1
1 Boundary Value Problems	3
1.1 The Continuous Problem	3
1.2 Numerical Solution	4
1.2.1 Finite Difference Solutions on Uniform Grids	4
1.2.2 Finite Difference Solutions on Nonuniform Grids	11
2 Mesh Quality and the Linear System	14
2.1 The Spectrum of the Linear System Matrix	14
2.1.1 Examples	16
2.1.2 Effect of Nonuniform Grids	20
2.1.3 Further Comments	22
2.2 Singular Value Decomposition	23

2.3	Detecting Layers with Iterations	28
2.4	M -Matrices	32
3	A Matrix Inverse Problem	38
3.1	Positivity Subject to a Perturbation	40
3.1.1	Higher Rank Perturbations	43
3.1.2	A Symmetric Perturbation	44
3.1.3	Extensions	45
3.2	An Application	45
II	Numerical Integration, Moving Meshes and Schwarz Waveform	48
4	Moving Mesh Methods	50
4.1	Equidistribution and a Moving Mesh PDE	50
4.2	Discretization and Solution Process	53
4.3	Other Implementation Strategies	54
5	Decoupled Integration and Multirate Methods	56
5.1	ODE methods	57
5.1.1	Decoupled Integration Formulas	57
5.2	Waveform Relaxation methods	60
5.2.1	Partitioning	63
5.3	Schur Decomposition Methods and the Quasi-Steady State Approximation	64
5.3.1	Estimating the QSSA Error	66
5.4	PDE Based Methods	71
5.4.1	Space-Time Adaptive hp -Refinement Methods	71
5.4.2	Schwarz Waveform Relaxation	71
6	Schwarz Waveform Moving Mesh Method	75
6.1	Continuous Algorithm	75
6.2	Solving the Moving Boundary Problems	77
7	Numerical Results	78
7.1	Model Problems	78

7.2	Waveform Relaxation	79
7.2.1	Effect of Overlap and Maximum Number of Times Steps	79
7.2.2	Rate of Convergence	83
7.3	Schwarz Waveform on Fixed Grids	83
7.4	Schwarz Waveform and Moving Meshes	85
7.4.1	Effect of a Fixed Mesh Point	85
7.4.2	Solution of Burgers' Equation with the Schwarz Waveform Moving Mesh method	86
7.4.3	Two Spike Problem	90
8	Conclusions and Future Work	94
	Bibliography	97

List of Tables

7.1	Number of Waveform iterations and CPU time (seconds) for Jacobi Waveform with MAXSTEPS = 1000	81
7.2	Number of Waveform iterations and CPU time (seconds) for Jacobi Waveform with MAXSTEPS = 6000	81
7.3	Number of Waveform iterations and CPU time (seconds) for Gauss-Seidel Waveform with MAXSTEPS = 1000	82
7.4	Number of Waveform iterations and CPU time (seconds) for Gauss-Seidel Waveform with MAXSTEPS = 6000	82
7.5	Number of Waveform iterations and CPU time (seconds) for Schwarz Waveform with MAXSTEPS = 1000	84
7.6	Number of Waveform iterations and CPU time (seconds) for Schwarz Waveform with MAXSTEPS = 6000	85

List of Figures

1.1	Model Problem II: computed solution with $h = 0.05$ and $\epsilon = 0.01$ and a fine grid solution.	7
1.2	Model Problem III: computed solution with $h = 0.05$ and $\epsilon = 5e - 4$ and a fine grid solution.	8
1.3	Model Problem IV: computed solution with $h = 0.05$ and $\epsilon = 5e - 3$ and a fine grid solution.	9
1.4	A solution of a two-dimensional boundary layer problem on a fine grid. . . .	10
1.5	Unresolved boundary layer in a 2-d problem.	10
2.1	Numerical solution (left) and eigenvalue distribution (right) corresponding to an unresolved boundary layer.	17
2.2	Numerical solution (left) and eigenvalue distribution (right) corresponding to a resolved boundary layer.	17
2.3	$ \text{Im } \lambda $ for $h < 2\epsilon$ (left) and $ \text{Re } \lambda $ for $h > 2\epsilon$ (right).	18
2.4	Computed solutions (left) and corresponding eigenvalues (right) for an interior layer problem with decreasing values of h (top to bottom).	19
2.5	Eigenvalue distributions corresponding to two different piecewise uniform grids.	20
2.6	Eigenvalue distributions corresponding to uniform (left) and equidistributed (right) grids.	21
2.7	Equidistributed solution (left) and corresponding eigenvalues (right) for a sharp boundary layer.	23
2.8	The dominant singular vector (left) and α (right) for model problem I.	25
2.9	The three most dominant singular vectors corresponding to a two layer problem on a fine mesh.	25

2.10	The three most dominant singular vectors corresponding to model problem II on a fine mesh.	26
2.11	Under-resolved solution and corresponding dominant singular vector for model problem II.	26
2.12	The dominant singular vectors corresponding to model problem III on a fine (left) and under-resolved (right) mesh.	27
2.13	Solution of model problem I with $\epsilon = 1e - 4$ and $N = 101$ mesh points (left); Approximate solution after 10 CGNR iterations (right).	28
2.14	Difference of approximate solution and filtered approximate solution for model problem I after 10, 20, and 30 CGNR iterations.	29
2.15	Solution of model problem I with $\epsilon = 1e - 4$ and $N = 501$ mesh points (top left); Difference of approximate solution and filtered approximate solution after 10, 20 and 30 iterations of CGNR (top right and bottom).	30
2.16	Difference of approximate solution and filtered approximate solution of model problem IV with $\epsilon = 1e - 4$ after 20 iterations of CGNR for $N = 51, 151,$ and 201 mesh points.	31
2.17	Difference of approximate solution and filtered approximate solution of model problem III with $\epsilon = 1e - 8$ after 20 CGNR iterations for $N = 101, 151$ and 201 mesh points.	31
2.18	After 20 iterations of CGNR with a non-random initial guess.	32
2.19	Solution and local M -matrix structure for model problem I on a uniform (top), piecewise uniform (middle) and equidistributed (bottom) grid.	34
2.20	Solution and local M -matrix structure for model problem IV on a uniform grid (top), a piecewise uniform grid (middle), and a refined uniform grid (bottom).	36
2.21	Solution and local M -matrix structure for a variable coefficient interior cusp problem with $\epsilon = 1e - 2$ (top), $\epsilon = 5e - 4$ (middle) and $\epsilon = 5e - 6$ (bottom).	37
6.1	Sequence of Moving Boundary Problems solved during one iteration of the Moving Mesh Schwarz Waveform method over a time window $[0, T]$	76
7.1	Convergence of waveform relaxation for various spatial mesh sizes and overlap= 8 (left) and tuned overlap (left).	83
7.2	Mesh trajectories for Burgers' equation on one domain	86

7.3	Solutions and errors for Burgers' equation with moving Schwarz waveform method at $t = 0.25, 0.45$ and 1.7	87
7.4	Moving Schwarz Mesh Trajectories for Burgers' Equation with $\epsilon = 1e-4$ and 40 points per domain	88
7.5	Number of time steps taken in each subdomain during each time window.	88
7.6	Length of time windows for moving Schwarz method applied to Burgers' equation with $\epsilon = 1e-4$	89
7.7	Moving Schwarz Mesh Trajectories for Burgers' Equation with $\epsilon = 1e-3$ and 20 points per domain	90
7.8	Exact solutions of two spike problem at $t = 0, 0.6, 1.6$ and 2.7	90
7.9	Solution at $t = 2.7$ and mesh trajectories for the two spike problem with one domain.	91
7.10	Solution at $t = 2.7$ and mesh trajectories for the two spike problem with two subdomains.	92
7.11	Time steps for one domain (left) and two subdomains (right) solution of the two spike problem.	93

Part I

**Boundary Value Problems and
Matrix Properties**

Computing the solution of differential equations requires an appropriate choice of discretization, mesh selection and algebraic equation solver. No aspect should be considered in isolation. A choice for any of these will affect possible options for the other two. To complicate matters further, the hardware and software you choose to compute your solution may affect the structure of linear systems which can be solved efficiently and hence determine the choice of discretization.

The purpose of this work is to investigate connections between the linear system of equations to be solved and the selection of an appropriate mesh. To provide a test suite of problems we study convection-diffusion problems of the form

$$-\epsilon\Delta u + \mathbf{b} \cdot \nabla u + cu = f$$

on a square domain $(x, y) \in [0, 1] \times [0, 1]$ subject to various boundary conditions. For small values of ϵ , corresponding to large Peclet numbers, the problem is convection dominated. Different choices of the problem data can lead to solutions with interesting features such as boundary, interior, and/or corner layers. Capturing these features can be a challenge for discretizations on uniform grids and as such provide an appropriate problem set.

We conclude Part I of the thesis by considering a matrix inverse problem. For symmetric, tridiagonal M -matrices we are able to find a bound on a positive perturbation of the matrix to ensure a positive inverse.

Chapter 1

Boundary Value Problems

1.1 The Continuous Problem

In this chapter, we consider the solution of convection–diffusion problems of the form

$$Lu = -\epsilon\Delta u + \mathbf{b} \cdot \nabla u + cu = f, \quad 0 < \epsilon \ll 1 \quad (1.1)$$

on a square domain $\Omega := \{(x, y) | (x, y) \in [0, 1] \times [0, 1]\}$, subject to Dirichlet boundary conditions. The problem data $b(x)$, $c(x)$, and $f(x)$ are assumed to be continuous or at least bounded on Ω . In 1-d this problem becomes

$$-\epsilon u''(x) + b(x)u'(x) + c(x)u(x) = f(x), \quad u(0) = A, \quad u(1) = B, \quad (1.2)$$

on the interval $x \in [0, 1]$. We begin by commenting on the existence and uniqueness of solutions of (1.1) and then point out features of the solutions which make them difficult to compute.

Existence of solutions for $c(x) \geq 0$ is a classical result which follows from the Fredholm alternative applied to the elliptic operator L . In that case, uniqueness follows directly from a maximum principle. If $c(x) < 0$ then $Lu = f$ will have a unique solution if the homogeneous problem $Lu = 0$, $u = 0$ on $\partial\Omega$, has only the trivial solution, [39]. The Sturm transformation (in 1-d)

$$\bar{u} := \exp\left(\frac{1}{2\epsilon} \int_0^x b(s) ds\right) u,$$

provides a mechanism to determine conditions on $b(x)$, $c(x)$, ϵ , and the boundary values A and B so that that the homogeneous problem has only the trivial solution [93]. In what follows, we will assume that $c(x) \geq 0$ and (1.1) has a unique solution.

Problem (1.1) is a perturbation of the first order differential equation

$$\mathbf{b} \cdot \nabla u_0 + cu_0 = f. \quad (1.3)$$

Since the order of the reduced problem is less than the original differential equation, it is clear that the solution of (1.3) will generally not satisfy the boundary conditions on the whole of $\partial\Omega$. For this reason, regular expansions of the solution will not be valid throughout Ω and (1.1) is referred to as a *singular perturbation problem*. Analytic approximations for solutions of such problems may be obtained using the method of matched asymptotic expansions, see for example [54] and [26], and the Wentzel–Kramer–Brillouin (WKB) method [89] and [98].

Those regions of $\partial\Omega$ on which u_0 does not satisfy the boundary conditions imposed on u are the locations of boundary layers. These are regions of rapid transition in the solution which prove to be a challenge to resolve numerically. Under certain conditions on the coefficient functions, it is possible to determine which boundary conditions u_0 should satisfy and hence the location of layers in the solution. Depending on the functions $b(x)$, $c(x)$ and $f(x)$ and the boundary values A and B , the solution of (1.2) may have one or more regions of rapid transition. A summary for the 1–d case is given in [8] while [66] provides an analysis in 2–d using the characteristics of (1.3).

1.2 Numerical Solution

In this section we point out difficulties with standard finite difference approximations to (1.1). We also provide a brief survey of various numerical approaches which attempt to circumvent these obstacles. Much of the material in this section may be considered classical, and as such, we will only provide sufficient details to introduce notation and highlight the important results. Details can be found in the general references [83] and [8]. Specific non-generic schemes and results will be referenced individually.

1.2.1 Finite Difference Solutions on Uniform Grids

The inability of a uniform mesh to efficiently resolve regions of rapid transition in solutions of differential equations is well-known. After introducing the required notation we will demonstrate the problem with a simple 1–d constant coefficient example.

We replace $\Omega = [0, 1]$ with a finite set of points

$$\Omega^h := \{x_j \mid x_j = jh, \text{ for } j = 0, \dots, M\}$$

where $h = 1/M$ is the spatial step size. We let u_j denote our approximation to $u(x_j)$ for $j = 0, \dots, M$. To approximate the derivatives $u'(x_j)$ and $u''(x_j)$ in (1.2) we introduce the forward and backward differences

$$D^+u_j = \frac{u_{j+1} - u_j}{h} \quad \text{and} \quad D^-u_j = \frac{u_j - u_{j-1}}{h}.$$

The operators $D^1 = (D^+ + D^-)/2$ and $D^2 = D^+D^-$ then give second order approximations to the first and second derivatives, respectively, where

$$D^1u_j = \frac{u_{j+1} - u_{j-1}}{2h} \quad \text{and} \quad D^2u_j = \frac{u_{j+1} - 2u_j + u_{j-1}}{h^2}.$$

We now replace (1.2) with the discrete system of equations

$$\begin{aligned} -\epsilon D^2u_j + b_j D^1u_j + c_j u_j &= f_j, & j = 1, \dots, M-1 \\ u_0 &= A & u_M = B, \end{aligned} \quad (1.4)$$

where $b_j = b(x_j)$, $c_j = c(x_j)$, and $f_j = f(x_j)$. This is equivalent to the system

$$\begin{aligned} \alpha_j u_{j-1} + \beta_j u_j + \gamma_j u_{j+1} &= f_j, & j = 1, \dots, M-1 \\ u_0 &= A & u_M = B, \end{aligned} \quad (1.5)$$

where

$$\alpha_j = -\frac{\epsilon}{h^2} - \frac{b_j}{2h}, \quad \beta_j = \frac{2\epsilon}{h^2} + c_j, \quad \text{and} \quad \gamma_j = -\frac{\epsilon}{h^2} + \frac{b_j}{2h}. \quad (1.6)$$

Written in matrix form we have the system of equations

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & \cdots & 0 \\ \alpha_1 & \beta_1 & \gamma_1 & 0 & 0 & \cdots & 0 \\ 0 & \alpha_2 & \beta_2 & \gamma_2 & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & & \vdots \\ 0 & 0 & \cdots & \alpha_{M-2} & \beta_{M-2} & \gamma_{M-2} & 0 \\ 0 & 0 & \cdots & 0 & \alpha_{M-1} & \beta_{M-1} & \gamma_{M-1} \\ 0 & 0 & \cdots & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \\ u_{M-2} \\ u_{M-1} \\ u_M \end{pmatrix} = \begin{pmatrix} A \\ f_1 \\ f_2 \\ \vdots \\ f_{M-2} \\ f_{M-1} \\ B \end{pmatrix}. \quad (1.7)$$

The scheme (1.4) is second order accurate on uniform grids. It is well known, however, that such 3-point centered difference schemes are not stable for $h \gg \epsilon$. This makes such schemes cost prohibitive for small ϵ , which is indeed the situation of practical interest.

The difficulty is made quite explicit by solving the difference equations which result by discretizing

$$-\epsilon u'' - u' = 0, \quad u(0) = 0, \quad u(1) = 1. \quad (\text{Model Problem I})$$

Solving (1.4) exactly for this example we find

$$u_i = \frac{r^i - 1}{r^M - 1} \quad \text{where} \quad r = \frac{2\epsilon - h}{2\epsilon + h}.$$

The continuous solution is monotonic while u_i clearly oscillates unless $h < 2\epsilon$. This would require an unacceptably large number of nodes for convection dominated equations with $\epsilon \ll 1$.

The stability problems associated with centered differences for the u' term have been remedied in numerous ways. A well-known choice is to use *upwinding*. That is, replace $u'(x_j)$ by D^+u_j if $b_j < 0$ and D^-u_j if $b_j > 0$. This gives uniform stability (with respect to ϵ) and an $O(h)$ uniform approximation outside of the layer. Upwinding is equivalent to adding artificial diffusion to the differential equation, which stabilizes standard discretizations. Unfortunately, upwinding is over-diffusive and has the undesired effect of *smearing* or widening layers in the solution. Upwinding is a particular case of a more general class of schemes which use a fitting factor to add artificial diffusion to a problem. These methods may be written in the form

$$-\epsilon\sigma(q(x_i))D^2u_i + b_iD^1u_i + c_iu_i = f_i, \quad (1.8)$$

where $q(x) = b(x)h/2\epsilon$. We see that centered differences are again used for the u' term and diffusion is added through the term σ . Classical upwinding may be recovered from this class of methods by choosing $\sigma(q) = 1 + q$. The amount of added diffusion may be tuned by introducing a numerical viscosity parameter ξ . For example, by choosing $\sigma(q) = 1 + \xi q$ it is possible to tune ξ so that numerical solution is exact for constant coefficient problems [15]. The choice $\sigma(q) = q \coth q(x)$ gives the Il'in-Allen-Southwell scheme, [2] and [51], which is second order accurate and $O(h)$ uniformly convergent on the entire interval for the variable coefficient problem.

The inadequacy of central difference schemes on uniform grids for convection dominated equations is a well-known issue, and is not restricted to constant coefficient boundary layer problems in one dimension. In fact, similar behaviour exists for problems with multiple boundary and/or interior layers, and for variable coefficient or non-homogeneous terms as the following examples indicate.

Examples

A layer at one end

$$\epsilon u'' + u' = 1, \quad u(0) = 1, \quad u(1) = 1 \quad (\text{Model Problem II})$$

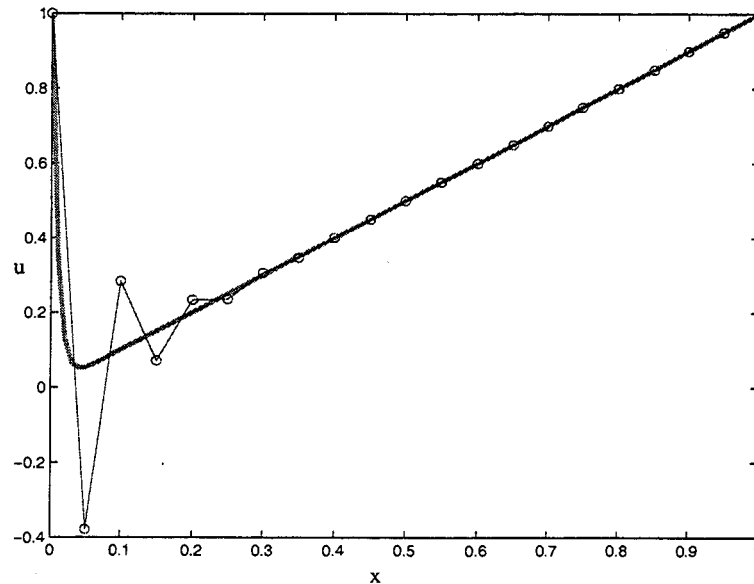


Figure 1.1: Model Problem II: computed solution with $h = 0.05$ and $\epsilon = 0.01$ and a fine grid solution.

An interior layer

$$\epsilon u'' + (x - 1/2)u' = 0, \quad u(0) = 0, u(1) = 1 \quad (\text{Model Problem III})$$

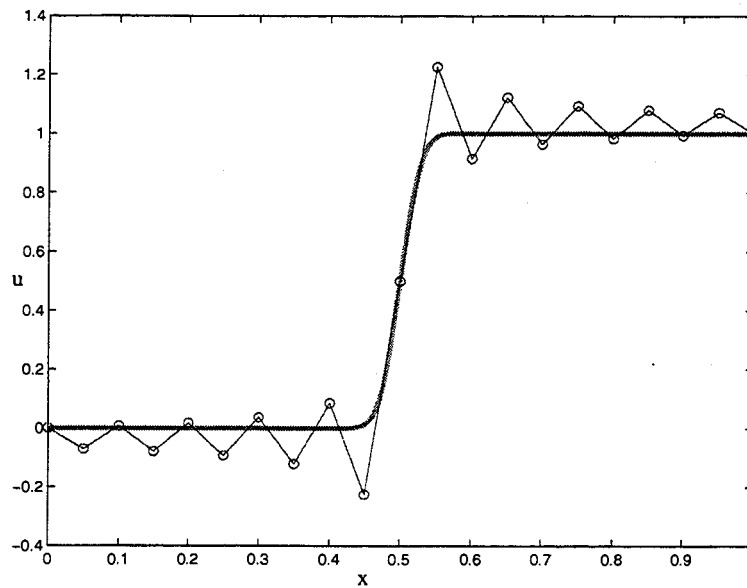


Figure 1.2: Model Problem III: computed solution with $h = 0.05$ and $\epsilon = 5e - 4$ and a fine grid solution.

A layer at both ends

$$-\epsilon u'' + (x - 1/2)u' = 0, \quad u(0) = 0, \quad u(1) = 1 \quad (\text{Model Problem IV})$$

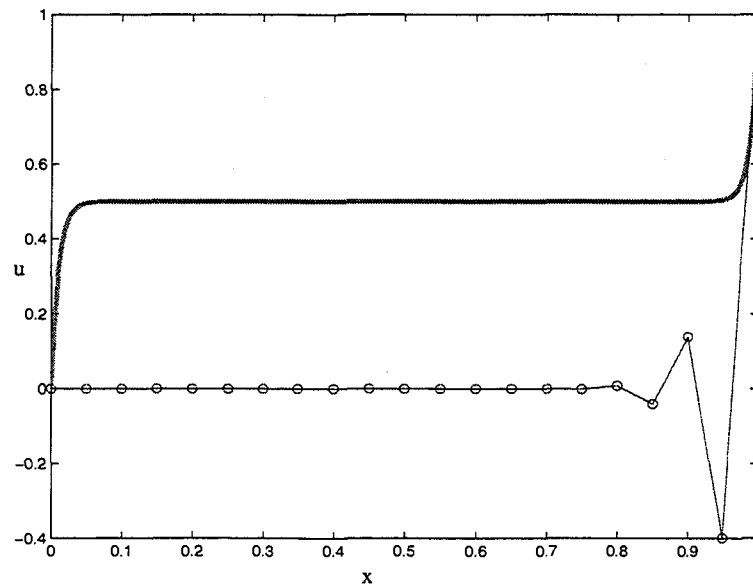


Figure 1.3: Model Problem IV: computed solution with $h = 0.05$ and $\epsilon = 5e - 3$ and a fine grid solution.

A 2-d example

$$-\epsilon \Delta u + (1 + (x^2 + y)^2) u_x = 0$$

$$u(0, 1) = u(x, 1) = 0, \quad u(x, 0) = 64x^3(1-x)^3, \quad u(1, y) = 64y^3(1-y)^3.$$

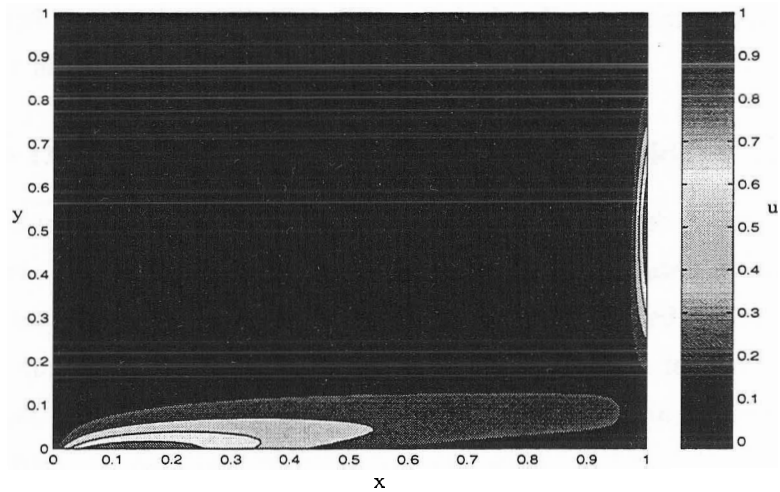


Figure 1.4: A solution of a two-dimensional boundary layer problem on a fine grid.

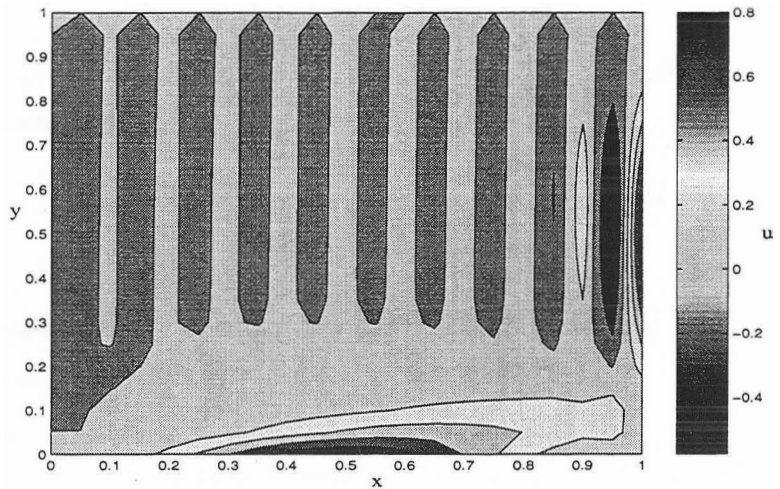


Figure 1.5: Unresolved boundary layer in a 2-d problem.

These examples indicate that a moderate number of equally spaced mesh points fail to resolve boundary or interior layers for convection dominated equations. This is reflected in the mesh scale oscillations in the computed solutions. As we have mentioned there are schemes available which circumvent this problem by adding artificial diffusion. Unfortunately, obtaining high order accuracy independent of ϵ is a challenge, and is very much problem dependent. Another approach is to tune the discrete equations, not by adding diffusion, but by altering the computational mesh.

1.2.2 Finite Difference Solutions on Nonuniform Grids

The general fitting techniques mentioned above work well on relatively simple problems which have layers at one end of the interval. For more complicated problems with more than one boundary layer and/or interior layers more sophisticated methods are required. An alternative approach is to make a selection of nodes which is more appropriate for the problem, that is, concentrating nodes in regions of the domain where the solution has interesting features.

To facilitate nonuniform grid spacing, we discretize our problem using finite differences on an arbitrary mesh

$$\Omega^{\bar{h}} := \{x_i \mid 0 = x_0 < x_1 < \dots < x_M = 1\}.$$

If we let $h_j = x_{j+1} - x_j$, $j = 0, \dots, M-1$ then $\bar{h} = \max h_j$. We replace (1.2) by the discrete system of equations

$$\begin{aligned} -\epsilon D^c u_j + b_j D^0 u_j + c_j u_j &= f_j, & j = 1, \dots, M-1, \\ u_0 &= A & u_M = B. \end{aligned} \quad (1.9)$$

The difference operators $D^0 u_j$ and $D^c u_j$ provide approximations to the first and second derivatives of u at x_j . On $\Omega^{\bar{h}}$, $D^0 u_j$ and $D^c u_j$ are given by

$$D^0 u_j = \frac{1}{h_j h_{j-1} (h_j + h_{j-1})} (h_{j-1}^2 (u_{j+1} - u_j) + h_j^2 (u_j - u_{j-1})), \quad (1.10)$$

$$D^c u_j = \frac{2}{h_{j-1} h_j (h_j + h_{j-1})} (h_{j-1} (u_{j+1} - u_j) - h_j (u_j - u_{j-1})), \quad (1.11)$$

for $j = 1, \dots, M-1$. This gives the linear system of equations

$$\alpha_j u_{j-1} + \beta_j u_j + \gamma_j u_{j+1} = f_j \quad (1.12)$$

for $j = 1, \dots, M - 1$, with boundary conditions

$$u_0 = A \quad \text{and} \quad u_M = B.$$

The coefficients $\alpha_j, \beta_j, \gamma_j$ are given by

$$\alpha_j = \frac{1}{h_j + h_{j-1}} \left(\frac{-2\epsilon}{h_{j-1}} - \frac{b_j h_j}{h_{j-1}} \right), \quad \beta_j = \frac{2\epsilon}{h_j h_{j-1}} + \frac{b_j (h_j - h_{j-1})}{h_j h_{j-1}} + c_j,$$

$$\text{and} \quad \gamma_j = \frac{1}{h_j + h_{j-1}} \left(\frac{-2\epsilon}{h_j} + \frac{b_j h_{j-1}}{h_j} \right).$$

In matrix form we obtain a system identical to (1.7) with α_j, β_j , and γ_j redefined as above.

Equation (1.9) is a 3-point centered difference scheme like (1.4). A Taylor series analysis, however, demonstrates that $D^c u_j$ is only a first order approximation to $u''(x_j)$ when $h_j \neq h_{j-1}(1 + O(h_j))$. Moreover, (1.9) suffers from the same instabilities, which appear as oscillations in the computed solution, unless the nodes are chosen to keep the maximum h_j in the neighborhood of the layer sufficiently small. Pearson [79] used (1.9) along with a basic mesh redistribution strategy and continuation in ϵ to solve various BVP with ϵ as small as 10^{-10} .

Second order accuracy may be recovered by introducing a staggered mesh. Any equation of the form (1.2) is easily rewritten as

$$u'' = (p(x)u)' + r(x)u + q(x), \quad (1.13)$$

where

$$p(x) = \frac{b(x)}{\epsilon}, \quad r(x) = \frac{1}{\epsilon}(c(x) - b'(x)), \quad \text{and} \quad q(x) = -\frac{q(x)}{\epsilon}.$$

To discretize (1.13) we first rewrite the second order differential equation as a system of first order equations

$$u' = p(x)u + v \quad (1.14)$$

$$v' = r(x)u + q(x). \quad (1.15)$$

We now discretize (1.14) using the midpoint rule on $[x_j, x_{j+1}]$ and (1.15) on the staggered interval $[x_{j-1/2}, x_{j+1/2}]$. After a little algebra (see [8] for details), a second order 3-point discretization of (1.13) results. Written in terms of u values we have

$$\frac{2}{h_j + h_{j-1}} \left(\frac{1}{h_j}(u_{j+1} - u_j) - \frac{1}{h_{j-1}}(u_j - u_{j-1}) - \frac{1}{2}p_{i+1/2}(u_{j+1} + u_j) + \frac{1}{2}p_{i-1/2}(u_j + u_{j-1}) \right)$$

$$= r(\tilde{x}_j)\tilde{u}_j + q(\tilde{x}_j). \quad (1.16)$$

In this formula, $p_{j+1/2} = p(x_{j+1/2})$, \tilde{x}_i is the midpoint of $[x_{j-1/2}, x_{j+1/2}]$, and \tilde{u}_j is the value of u at \tilde{x}_j obtained by quadratic interpolation from u_{j-1} , u_j , and u_{j+1} .

Comparing this to (1.12) this difference scheme may be written in matrix form with α_j, β_j and γ_j redefined as

$$\begin{aligned}\alpha_j &= \frac{2}{h_{j-1}(h_j + h_{j-1})} + \frac{p_{j-1/2}}{h_j + h_{j-1}} - r(\tilde{x}_j)\delta_j \\ \beta_j &= \frac{2}{h_{j-1}(h_j + h_{j-1})} + \frac{p_{j+1/2}}{h_j + h_{j-1}} - r(\tilde{x}_j)\eta_j \\ \gamma_j &= -\frac{2(h_j^{-1} + h_{j-1}^{-1})}{h_j + h_{j-1}} + \frac{p_{j-1/2} - p_{j+1/2}}{h_j + h_{j-1}} - r(\tilde{x}_j)\xi_j,\end{aligned}$$

and $f_j = q(\tilde{x}_j)$. The quantities δ_j, η_j and ξ_j are the coefficients of u_{j-1}, u_j and $u_{j+1/2}$ in the interpolation expression for \tilde{u}_j .

Any discretization on a nonuniform mesh must be accompanied with some mesh selection strategy. As a minimum requirement we must choose a mesh so that the computed solution on that mesh is "better" than the solution computed on a uniform mesh of the same size. How do we choose such a mesh? The simplest case occurs when we can be guided by some *a priori* knowledge of the exact solution. For example, in physical problems we may have experimental or theoretical evidence which suggests how the solution will behave. In the finite element literature, *a posteriori* error estimates are used to extract information from a computed solution to choose a better grid. In the second part of this thesis we will review another possibility which allows for the simultaneous calculation of an appropriate mesh and the solution on that mesh. In that case the mesh is chosen so that the computed solution approximately equidistributes some indicator of the solution error.

As we have seen, solving linear two point boundary value problems with finite differences (or finite elements/volumes for that matter) require the solution of a linear system. In the next chapter we begin to ask if the linear system itself contains information which may guide us in the construction of a better mesh.

Chapter 2

Mesh Quality and the Linear System

Upon discretizing a linear boundary value problem, we obtain a linear system of equations $Au^h = f^h$ whose solution $u^h = (u_0, u_1, \dots, u_M)^T$ is an approximation to the solution of the continuous problem on Ω^h . In this section we will highlight how certain properties of the matrix A are related to the appropriateness of the chosen mesh. Ultimately, one would like to be guided in constructing a new, “better” mesh, by these observations.

2.1 The Spectrum of the Linear System Matrix

In this section we provide a derivation of the eigenvalues of the linear system matrix corresponding to a discretization of a linear, constant coefficient two-point BVP on a uniform mesh.

Consider a nonsymmetric tridiagonal matrix $A = \text{tridiag}\{\alpha, \beta, \gamma\} \in \mathbb{C}^{m \times m}$. Assume q is an eigenvector of A with associated eigenvalue λ . Then $Aq = \lambda q$ gives rise to the linear difference equation

$$\alpha q_{l-1} + (\beta - \lambda)q_l + \gamma q_{l+1} = 0,$$

for $l = 1, \dots, m$. To close the system we assign boundary conditions $q_0 = q_{m+1} = 0$. Assuming $q_l = r^l$ we obtain the characteristic roots r_{\pm} as solutions of the quadratic equation

$$\alpha + (\beta - \lambda)r + \gamma r^2 = 0,$$

and this yields

$$r_{\pm} = \frac{\lambda - \alpha \pm \sqrt{(\lambda - \beta)^2 - 4\alpha\lambda}}{2\gamma}.$$

Therefore, each component, q_l , may be written as a linear combination of r_+^l and r_-^l , that is,

$$q_l = c_1 r_+^l + c_2 r_-^l.$$

The boundary condition $q_0 = 0$ implies $c_1 = -c_2$, and

$$q_l = c_1 (r_+^l - r_-^l). \quad (2.1)$$

Now, $q_{m+1} = 0$ gives

$$r_+^{m+1} = r_-^{m+1}$$

or

$$\left(\frac{r_+}{r_-}\right)^{m+1} = 1,$$

which implies

$$r_+ = r_- e^{\frac{2\pi ki}{m+1}}, \quad k = 0, 1, \dots, m. \quad (2.2)$$

By inspection we see that $k = 0$ would imply $r_+ = r_-$ and hence $q = 0$ from (2.1).

Rewriting (2.2) we have

$$r_+ e^{\frac{-\pi ki}{m+1}} = r_- e^{\frac{\pi ki}{m+1}},$$

and substituting the expressions for r_+ and r_- and expanding the complex exponentials we obtain

$$\lambda = \beta \pm 2\sqrt{\alpha\gamma} \cos \frac{\pi k}{m+1}, \quad k = 1, \dots, m. \quad (2.3)$$

The minus sign in the \pm can be ignored since it just repeats the eigenvalues.

Using this expression for λ we find the following expression for the eigenvector components:

$$q_l^{(k)} = 2ci \left(\frac{\alpha}{\gamma}\right)^{l/2} \sin \frac{\pi kl}{m+1},$$

where c is an arbitrary constant.

The nature of the eigenvalues of A depends on the sign of the product $\alpha\gamma$. For problems of the form (1.2) with constant coefficients we have from (1.6)

$$\alpha = -\frac{\epsilon}{h^2} - \frac{b}{2h} \quad \text{and} \quad \gamma = -\frac{\epsilon}{h^2} + \frac{b}{2h}.$$

An easy calculation shows that $\alpha\gamma$ is nonnegative if $h \leq 2\epsilon/|b|$ and negative if $h > 2\epsilon/|b|$. This says that the matrix A will have all real eigenvalues if $h \leq 2\epsilon/|b|$ and complex eigenvalues with constant real part otherwise. Moreover, using the fact that

$$\frac{m-k+1}{m+1} = 1 - \frac{k}{m+1},$$

and the cosine addition formula we see that

$$\cos\left(\frac{(m-k+1)\pi}{m+1}\right) = -\cos\left(\frac{k\pi}{m+1}\right),$$

which implies that

$$\lambda_k = \bar{\lambda}_{m-k+1},$$

that is, complex eigenvalues always occur in conjugate pairs.

2.1.1 Examples

We now consider a concrete example to demonstrate the relationship between the nature of the eigenvalues and the quality of the chosen mesh. As a first example recall model problem I from Chapter 1,

$$-\epsilon u'' - u' = 0, \quad u(0) = 0, \quad u(1) = 1. \quad (2.4)$$

Using the analysis from the previous section we expect real eigenvalues if $h \leq 2\epsilon$ and complex eigenvalues with constant real part if $h > 2\epsilon$. In Figure 2.1 the mesh size h is chosen larger than 2ϵ resulting in a mesh which does not resolve the boundary layer at $x = 0$. The computed solution exhibits mesh scale oscillations as predicted by solving the discrete equations. As anticipated, the eigenvalues of the linear system matrix appear as complex conjugates. There are actually two real eigenvalues at $\lambda = 1$ which result due to the use of non-eliminated boundary conditions. To impose Dirichlet boundary conditions the first and last rows of the linear system matrix A are chosen as $(1, 0, \dots, 0, 0)$ and $(0, 0, \dots, 0, 1)$ with the right hand side vector containing the boundary values. These eigenvalues will appear in all our figures. Figure 2.2 demonstrates the real eigenvalues which result by choosing $h \leq 2\epsilon$. The boundary layer is now resolved resulting in a smooth monotonic solution.

The inhomogeneous boundary value problem (model problem II)

$$-\epsilon u'' - u' = -1, \quad u(0) = 1, \quad u(1) = 1,$$

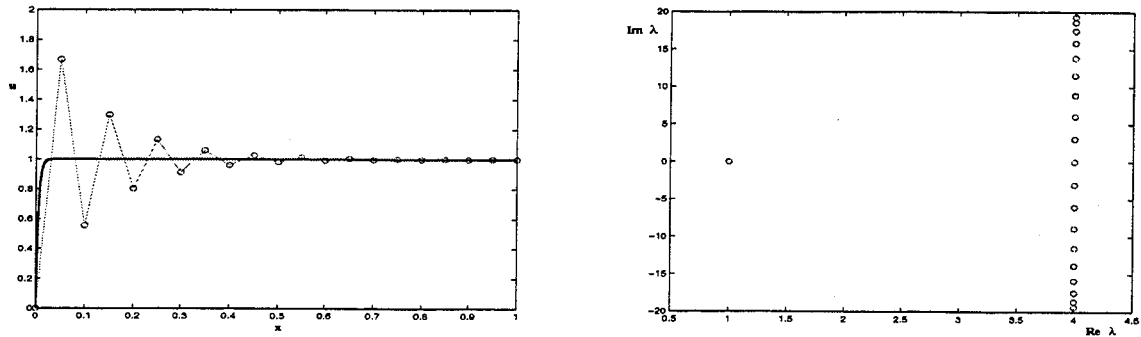


Figure 2.1: Numerical solution (left) and eigenvalue distribution (right) corresponding to an unresolved boundary layer.

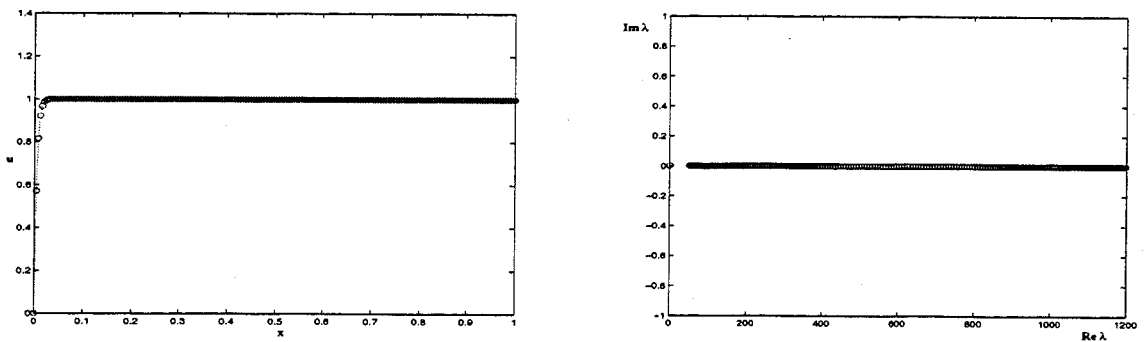


Figure 2.2: Numerical solution (left) and eigenvalue distribution (right) corresponding to a resolved boundary layer.

again has a sharp layer at $x = 0$ for $\epsilon \ll 1$; however, the solution is no longer constant outside of the layer. On the left of Figure 2.3 we illustrate the magnitude of the imaginary part of the largest eigenvalue of A for $h > 2\epsilon$. As shown, the $|\text{Im } \lambda| \rightarrow 0$ as $h \rightarrow 2\epsilon^+$. For $h = 2\epsilon$ all the eigenvalues of A (besides the two eigenvalues at $\lambda = 1$) are given by $\lambda = 1/2\epsilon$. As h decreases past 2ϵ the eigenvalues remain real with $|\lambda| \rightarrow \infty$ as $h \rightarrow 0^+$. The behaviour of the eigenvalues as a function of h for fixed ϵ is easy to see analytically from the expression for the eigenvalues given in equation (2.3).

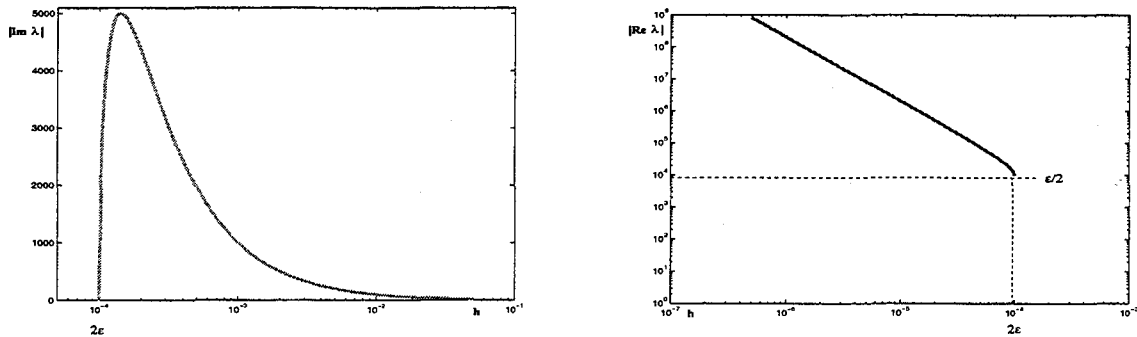


Figure 2.3: $|\text{Im } \lambda|$ for $h < 2\epsilon$ (left) and $|\text{Re } \lambda|$ for $h > 2\epsilon$ (right).

Although our analysis of the eigenvalues of the linear system matrix is restricted to constant coefficient boundary value problems on uniform grids, the next example again illustrates the connection between resolving a region of rapid transition and the emergence of real eigenvalues. Recall model problem III, a variable coefficient interior layer problem,

$$-\epsilon u'' - (x - 1/2)u' = 0, \quad u(0) = 0, \quad u(1) = 1.$$

The solution of this problem, shown in the bottom left of Figure 2.4, has an interior layer of width $O(\sqrt{\epsilon})$ at $x = 1/2$. As the mesh is refined (from top to bottom in the figure), we see the eigenvalues transform from complex with nearly constant real part to a combination of real and complex eigenvalues. Due to the variable coefficient of the u' term, a mesh which sufficiently resolves the layer no longer corresponds to a linear system matrix with all real eigenvalues. The distribution on the left has a larger number of real eigenvalues due to the larger proportion of points with local mesh spacing $\tilde{h} < 2\epsilon$.

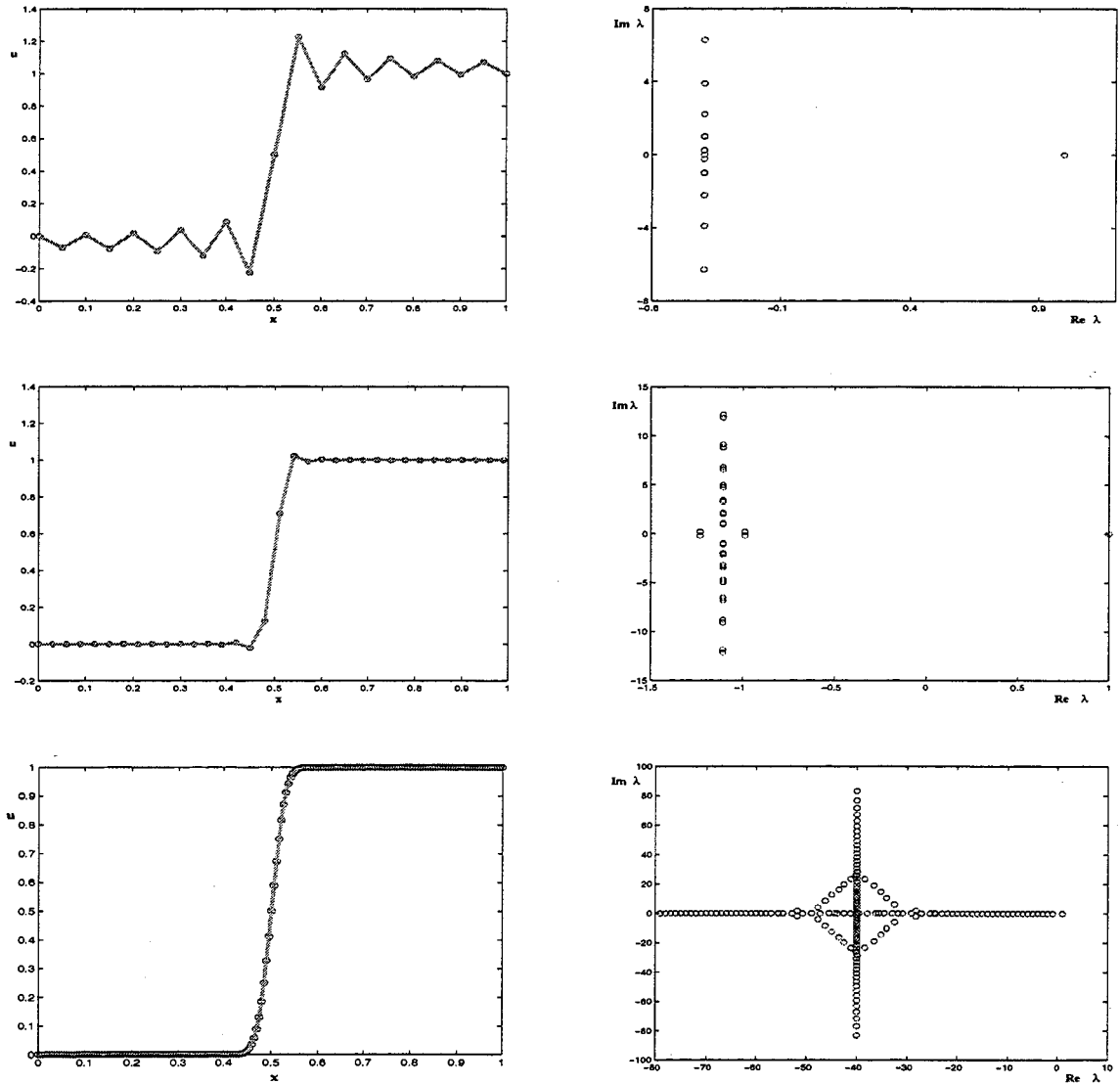


Figure 2.4: Computed solutions (left) and corresponding eigenvalues (right) for an interior layer problem with decreasing values of h (top to bottom).

2.1.2 Effect of Nonuniform Grids

It is quite clear that uniform grids are not sufficient for practical problems involving sharp regions of rapid change in the solution. Unfortunately, as we move away from a uniform mesh we also lose exact expressions for the eigenvalues and eigenvectors of the linear system matrix. In this section we hope to illuminate the effect of the choice of mesh on the spectrum of the matrix through several well-chosen examples. We restrict ourselves to piecewise uniform refinements and equidistributed grids.

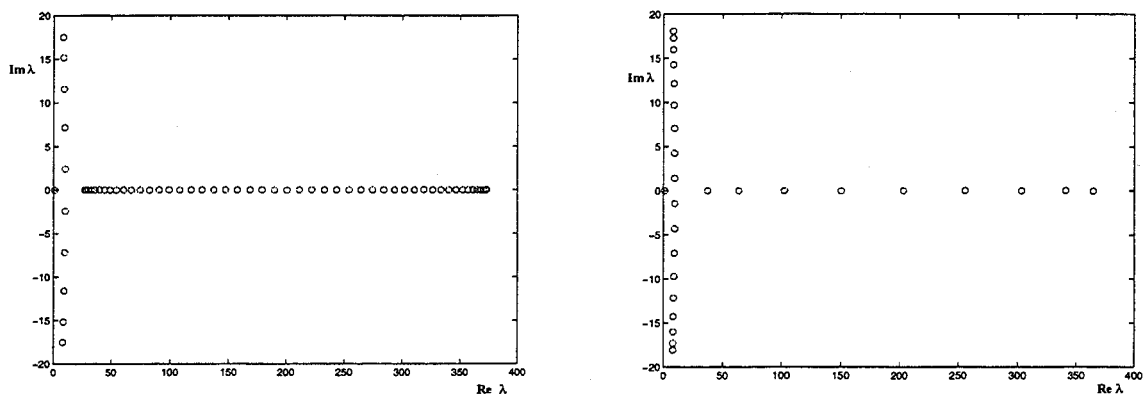


Figure 2.5: Eigenvalue distributions corresponding to two different piecewise uniform grids.

In Figure 2.5 we depict the eigenvalue distributions of the linear system matrix corresponding to model problem I for two different piecewise uniform grids. The plot on the left was generated with a simple grid composed of two uniform sub-grids. For $0 \leq x \leq 1/2$ we choose a spacing $h_1 < 2\epsilon$ and for $1/2 < x \leq 1$ we choose $h_2 > 2\epsilon$. The plot on the right corresponds to a grid with a much smaller region near $x = 0$ with $h_1 < 2\epsilon$. Both grids are chosen so that the boundary layer at $x = 0$ is sufficiently resolved. Both distributions contain real and complex eigenvalues. The real eigenvalues reflect the fact that the grids have local mesh spacings which satisfy the requirement $\tilde{h} \leq 2\epsilon$. Complex eigenvalues arise due to mesh points outside the layer where $\tilde{h} > 2\epsilon$.

We now compare eigenvalue distributions of the linear system matrix for uniform and equidistributed grids. The eigenvalues shown on the left plots in Figure 2.6 correspond to uniform grids while the right plots correspond to equidistributed grids with the same number of points. In the top left we choose 101 equally spaced points, enough to resolve the boundary layer at $x = 0$. As we have already seen the eigenvalues are real indicating that

the local mesh spacing satisfies $\bar{h} \leq 2\epsilon$ across the entire interval. Using an equidistributed grid with the same number of points (top right) we again obtain real eigenvalues. We should note, however, that the largest eigenvalues in this plot are many magnitudes larger than those corresponding to the uniform grid. This is due to the small local mesh spacing for the points in the layer with the equidistributed grid. We have seen in the previous section that the magnitude of the largest real eigenvalue grows large as $h \rightarrow 0$. A cluster of eigenvalues of size $10^1 \sim 10^2$ is quite evident for the equidistributed mesh. These eigenvalues reflect those mesh points outside the layer with local mesh spacing similar to the uniform mesh with the same number of points.

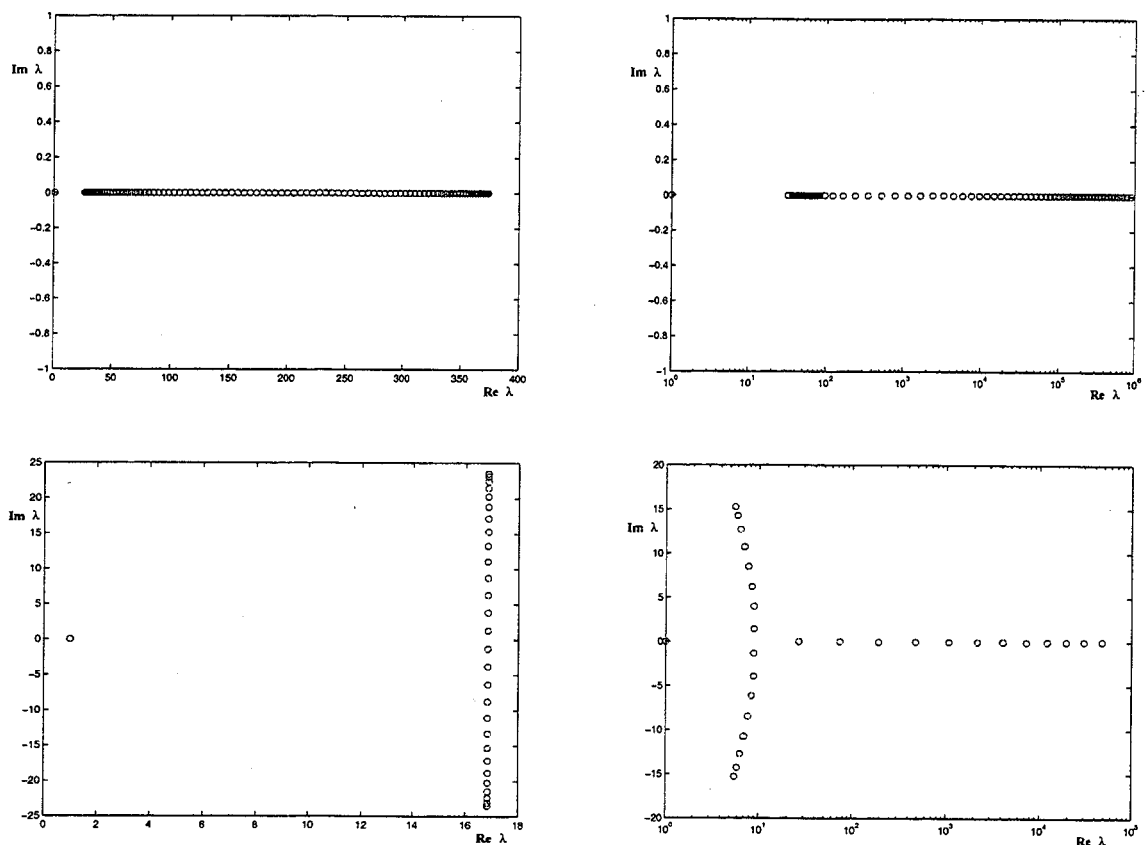


Figure 2.6: Eigenvalue distributions corresponding to uniform (left) and equidistributed (right) grids.

In the bottom two plots of Figure 2.6 we repeat the experiment for the same boundary layer problem but with only 30 points. It is clear that the uniform mesh is not capable of

resolving the layer. The lack of any real eigenvalues suggests the mesh does not achieve a local mesh spacing of $\tilde{h} \leq 2\epsilon$ anywhere on the interval. The equidistributed mesh with the same number of points, however, is able to resolve the layer. The mesh points in the layer satisfy the mesh spacing requirement and real eigenvalues result. In contrast to the equidistributed mesh with 101 points complex eigenvalues are now evident. This is due to those mesh points outside the layer which are unable to satisfy $\tilde{h} \leq 2\epsilon$. With 101 points the equidistributed mesh is able to provide small mesh spacing in the interval containing the boundary layer and keep the mesh spacing relatively small outside of the layer. This is not possible with only 30 equidistributed points for this problem.

We must stress at this point that the presence of real eigenvalues is necessary, but not sufficient evidence that the mesh has resolved a boundary layer for this class of convection dominated BVPs. The real eigenvalues merely indicate that the local mesh spacing requirement is satisfied somewhere in the interval. Indeed, if we repeat the experiments with either the piecewise uniform or equidistributed mesh by reflecting the mesh points in the line $x = 1/2$ (and thus creating inappropriate meshes) we will obtain nearly identical eigenvalue distributions.

As a last example, we consider our model problem I with $\epsilon = 1e - 4$. This would require at least 5000 equally spaced mesh points to resolve the boundary layer. We use an equidistributed grid with only 40 points. The computed solution and associated eigenvalues are illustrated in Figure 2.7. Again we note several important points: the number of real eigenvalues reflect the number of mesh points in the layer, that is, those for which $\tilde{h} \leq 2\epsilon$; the complex eigenvalues indicate those mesh points outside of the layer for which $\tilde{h} > 2\epsilon$; and the size of the largest real eigenvalues reflects the mesh spacing in the layer region.

2.1.3 Further Comments

As we have seen discretizing a constant coefficient singular perturbation problem (1.2) on a uniform mesh with centered differences results in a linear system to solve for the approximate solution. The linear system matrix is composed of a balance of a symmetric, positive definite contribution from the diffusion term and a skew-symmetric contribution from the convective term. Positive definite matrices have real eigenvalues, while skew symmetric matrices have purely imaginary eigenvalues. If h is larger than 2ϵ then the skew-symmetric matrix dominates resulting in complex eigenvalues.

Using upwinding for the convective terms results in a linear system matrix comprised

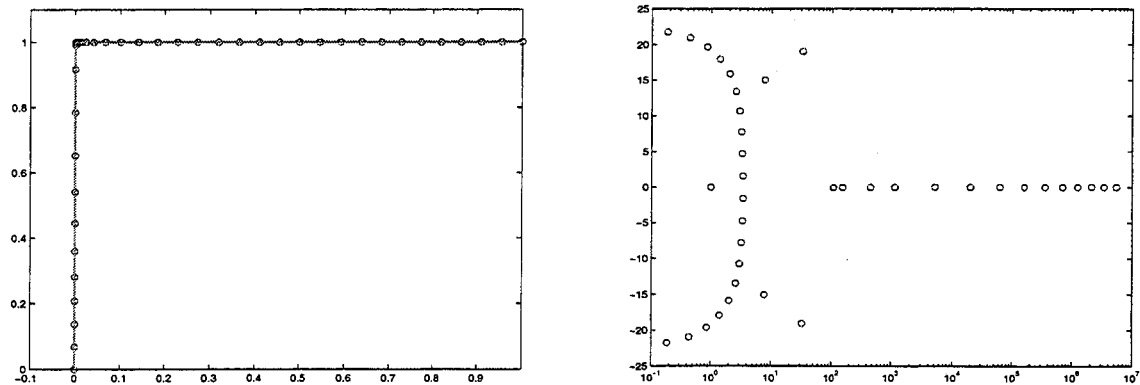


Figure 2.7: Equidistributed solution (left) and corresponding eigenvalues (right) for a sharp boundary layer.

again of a symmetric, positive definite matrix from u'' and a lower triangular matrix from the u' term. In this case, the lower triangular matrix increases the diagonal dominance of the system matrix resulting in real eigenvalues independent of the mesh spacing. In fact, as we have mentioned this leads to a discretization which is overly-diffusive. This may cause excessive widening of boundary layer.

Tuning the diffusion, equation (1.8), allows the user to balance the positive effect of increasing the diagonal dominance of the diffusion contribution to an optimal value, while ensuring the discretization is not over-diffusive.

2.2 Singular Value Decomposition

Although the eigenvalues of the linear system matrix do yield an indication as to the appropriateness of the chosen mesh, we are unable to decide if the refinement is in the correct location spatially. As we will see in this section, the singular vectors of the linear system matrix provide grid information which is spatially relevant.

We now consider the effect of the grid on the singular vectors of the matrix A . Every real, rectangular matrix $A \in \mathbb{R}^{m \times n}$ has a singular value decomposition (SVD) [40]

$$A = U\Sigma V^*.$$

The orthogonal matrices U and V may be written column-wise as

$$U = (u_1, u_2, \dots, u_m) \in \mathbb{R}^{m \times m}, \quad \text{and} \quad V = (v_1, v_2, \dots, v_n) \in \mathbb{R}^{n \times n},$$

and the diagonal matrix Σ written as

$$\Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_p) \in \mathbb{R}^{m \times n} \quad \text{with} \quad \sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_p \geq 0,$$

and $p = \min\{m, n\}$.

If A is square and nonsingular then $\sigma_n > 0$ and the SVD may be used to write the solution of $Au = f$ as

$$u = \sum_{i=1}^n \alpha_i v_i, \tag{2.5}$$

where $\alpha = \Sigma^{-1}U^*f$.

As we have seen, discretizing linear two point boundary value problems gives a linear system of discrete equations $Au = f$. So (2.5) is a representation of the approximate solution of the BVP. Equation (2.5) and the expression for α demonstrate that the singular vectors corresponding to the smallest singular values are dominant in the expansion for u . Of course how well the solution of the BVP resembles these low frequency or smooth singular vectors depends on the distribution of the singular values and the vector α . This is a reflection of the fact that the existence of layers in solution depends not only on the differential operator, represented by the matrix A , but also the boundary values and the inhomogeneity in the differential equation which is stored in the vector f^h and exerts its influence through α .

As we will see from various examples, boundary and interior layer information is contained in the low frequency singular vectors. Furthermore, the smooth singular vectors also indicate important mesh information. When a layer in the continuous solution of a convection dominated problem is not resolved we have seen that centered difference schemes yield oscillations in the computed solution. The oscillations are an important indicator that we have interesting, unresolved behaviour in the continuous problem. Furthermore, the location of the largest oscillations indicates the position of the layer in the continuous solution and hence where higher mesh concentration is required. We will demonstrate that the dominant singular vectors also have oscillations on mesh scale when layers are not resolved.

In Figure 2.8 (left) we illustrate the singular vector corresponding to the smallest singular value of model problem I with $\epsilon = 0.01$ and $N = 201$ uniformly spaced mesh points. The vector α which determines how the solution is represented by the smooth singular vector is shown on the right of the figure. The components of α are essentially zero except for the last entry, which is precisely the contribution of v_n to the solution of the boundary value problem.

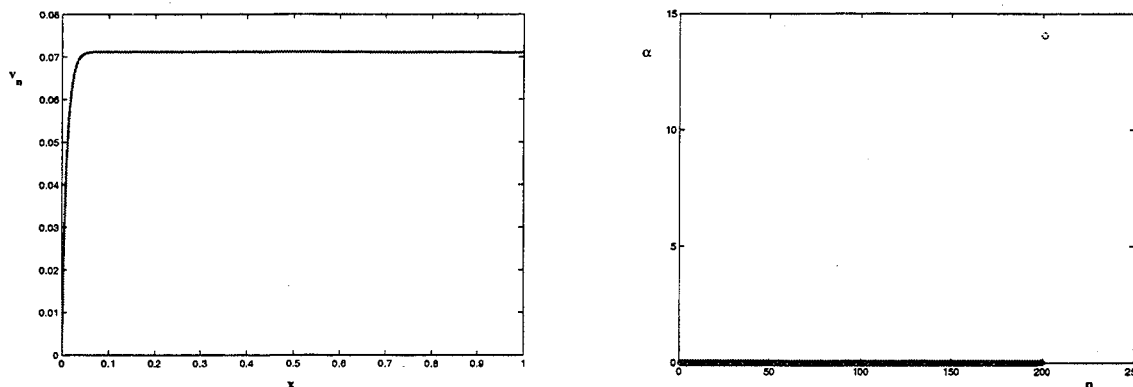


Figure 2.8: The dominant singular vector (left) and α (right) for model problem I.

To begin to understand what information pertaining to the solution of the boundary value problem and appropriateness of the chosen mesh is contained in the SVD of the matrix, we consider a few examples from our selection of model problems. We begin by considering model problem IV which we repeat here for convenience:

$$-\epsilon u'' + (x - 1/2)u' = 0$$

on $[0, 1]$, with $u(0) = 0$ and $u(1) = 1$, and $\epsilon \ll 1$. This problem has two sharp boundary layers at $x = 0$ and $x = 1$.

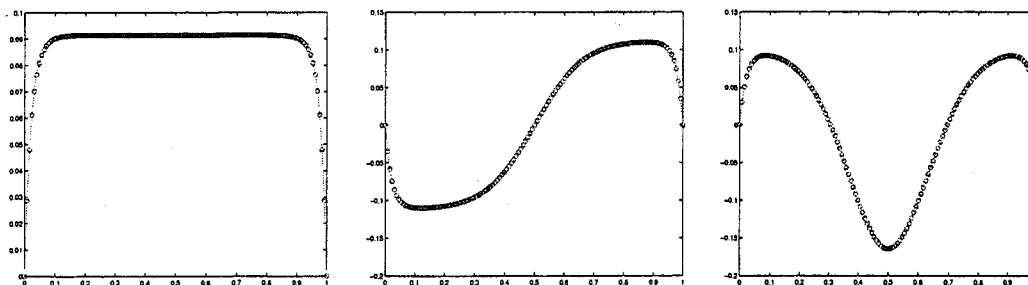


Figure 2.9: The three most dominant singular vectors corresponding to a two layer problem on a fine mesh.

The singular vectors of the linear system matrix of model problem IV are displayed in Figure 2.9. Here we have used a mesh sufficiently fine to resolve the boundary layers. We see that the layer information, most importantly location and steepness, is contained in those low frequency singular vectors.

As a second example, we consider model problem II,

$$\epsilon u'' + u' = 1$$

on $[0, 1]$, with $u(0) = 1$ and $u(1) = 1$. The solution of this problem has a boundary layer at $x = 0$ and an outer solution which looks like $u = x$. Figure 2.10 contains a plot of the three singular vectors corresponding to the smallest three singular values. Again we have chosen a mesh which resolves the layer.

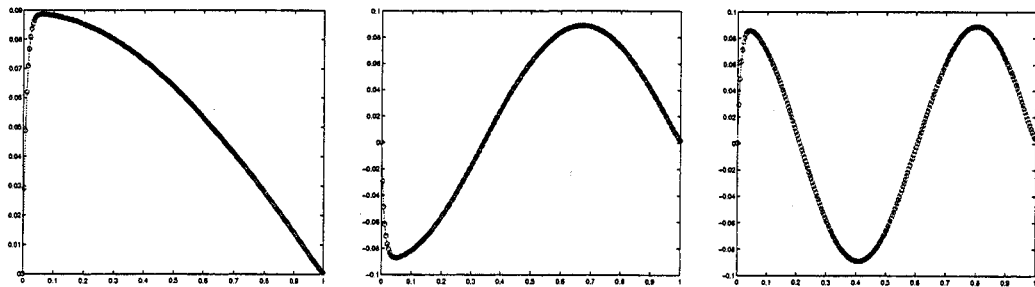


Figure 2.10: The three most dominant singular vectors corresponding to model problem II on a fine mesh.

We now discretize the problem on an inappropriate mesh, one with an insufficient number of points to resolve the boundary layer. The dominant singular vector again resembles the under-resolved solution as shown in Figure 2.11.

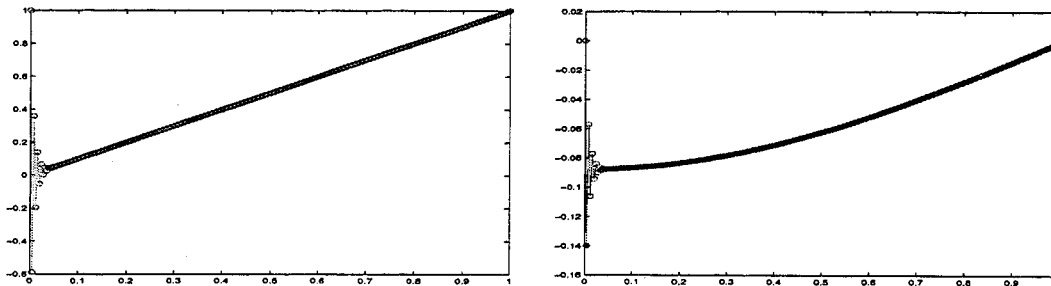


Figure 2.11: Under-resolved solution and corresponding dominant singular vector for model problem II.

Finally, we consider model problem III, a boundary value problem with an interior layer located at $x = 1/2$,

$$\epsilon u'' + (x - 1/2)u' = 0$$

on $[0, 1]$, with $u(0) = 0$ and $u(1) = 1$. The dominant singular vectors of the linear system matrix corresponding to a fine and an under-resolved grid are shown on the left and right of Figure 2.12 respectively.

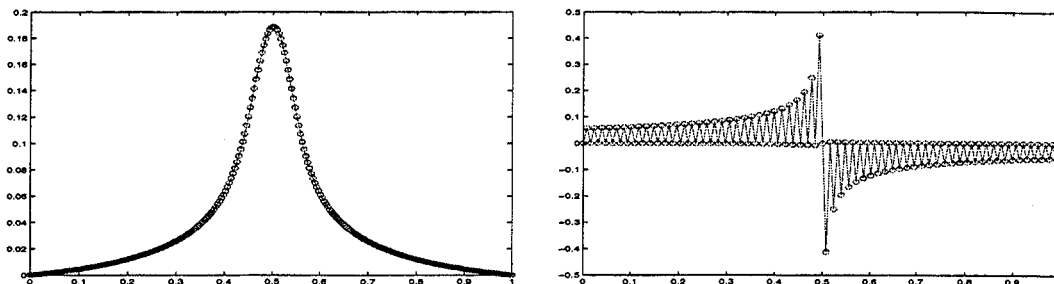


Figure 2.12: The dominant singular vectors corresponding to model problem III on a fine (left) and under-resolved (right) mesh.

The singular vectors not only tell us that we have a mesh resolution issue, they also indicate spatially where further mesh refinement is necessary. This is exactly what the eigenvalues were not able to do.

Unfortunately, computing the singular vector corresponding the smallest singular value is not an easy chore. An obvious choice of techniques would be to use some form of the Lanczos (applied to $A^T A$) or Arnoldi algorithms to compute the singular vectors, see [84] as a general reference. Lanczos and Arnoldi are iterative techniques to spectral decompositions akin to conjugate gradients [43] and GMRES [85], respectively, for solving large system of linear equations. Experiments with the Lanczos algorithm, however, indicate that convergence to the dominant singular vector of A (or eigenvector of $A^T A$) is quite slow. Fast convergence is achieved to the largest eigenvalue and associated eigenvector. This is not the end of the spectrum that we are interested in for this application. Typically, the fastest algorithms to converge to the smallest eigenvalue would involve iterating on A^{-1} . Each iteration coming at a cost roughly equivalent to a linear solve involving A . Since we are trying to ascertain mesh quality without actually solving the BVP this may be too high a price to pay. Some experiments which involve iterating on the original linear system are given in the next section and provide a glimmer of hope.

There is also some recent work by McSherry and Achlioptas [60] concerning acceleration techniques of Lanczos which may be applicable.

2.3 Detecting Layers with Iterations

A poor mesh selection will result in non-physical oscillations in the computed solution and singular vectors of the linear system matrix for convection dominated equations discretized by centered differences. In this section we will illustrate how simple iterations on the linear system may be used to detect boundary and/or interior layers without converging to the solution of the boundary value problem.

In Figure 2.13 we illustrate the solution of model problem I with $\epsilon = 1e - 4$ with a rather crude mesh with uniform spacing $h = 1/100$. Clearly, this mesh is not able to resolve the layer at $x = 0$. In fact, a uniform mesh consisting of several thousands points would be necessary. To the right of the solution we have displayed the result after 10 iterations of CGNR (Conjugate Gradients applied to the Normal Equations) applied to the discrete equations with a random initial guess (more on this later). At first glance, the approximate solution does not resemble the actual solution at all, except for the obvious oscillations. Figure 2.14 once again illustrates the approximate solutions after 10, 20 and 30 iterations. This time, however, we have averaged the results with a simple $[1, 2, 1]$ filter. The data we have pictured then is the absolute value of the difference between the approximate solution and its averaged counterpart. Although we have no grid point in the layer it is clear the approximate solution is having difficulty satisfying the boundary condition at $x = 0$.

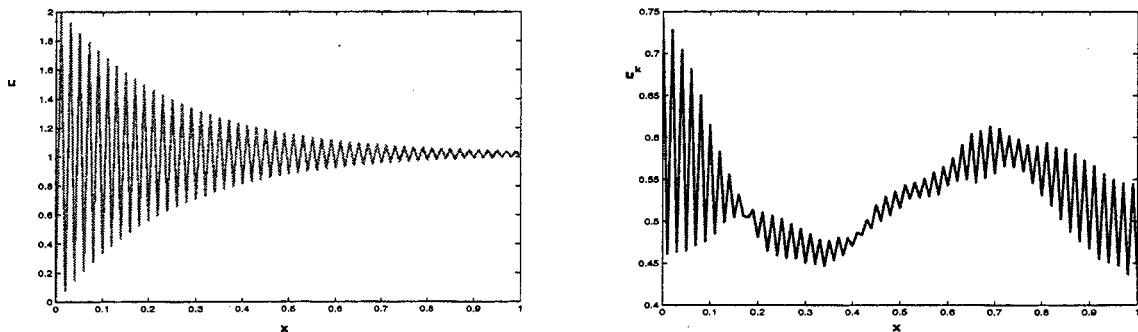


Figure 2.13: Solution of model problem I with $\epsilon = 1e - 4$ and $N = 101$ mesh points (left); Approximate solution after 10 CGNR iterations (right).

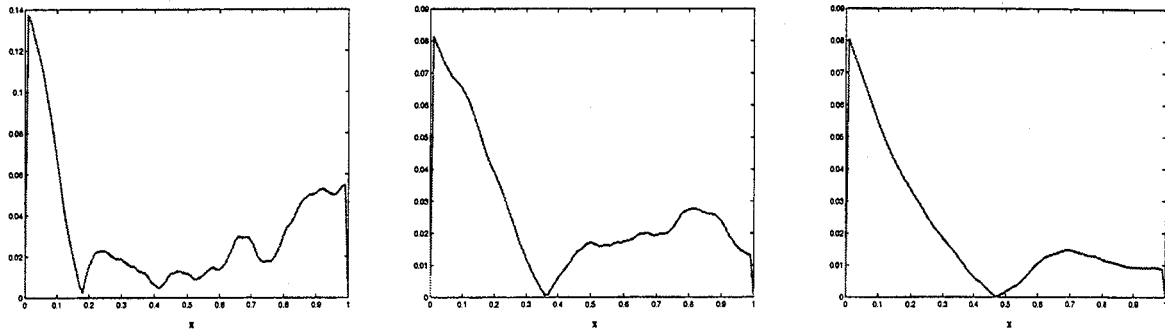


Figure 2.14: Difference of approximate solution and filtered approximate solution for model problem I after 10, 20, and 30 CGNR iterations.

Repeating the experiments on model problem I with a finer mesh consisting of 501 points produces similar results. The numerical solution is shown in the top left of Figure 2.15. Again the mesh is not able to resolve the layer, however, the oscillations are contained in a much smaller region near the boundary. A relatively few CGNR iterations detects the region of interest. The work involved is nominal considering that CGNR would take nearly 400 iterations to obtain the numerical solution to an accuracy of 10^{-6} . It is important to note that after 20 iterations CGNR has not converged, or even obtained a good approximation of the numerical solution, in fact $\|u - u^{20}\|_{\infty} = 1.39$.

Model problem IV is a boundary layer problem with layers at both ends of the interval. An inappropriate choice of mesh for this problem has peculiar results. Not only do centered differences result in oscillations in the computed solution, but the layer at $x = 0$ may be completely missed if the layer at $x = 1$ is unresolved. As shown in Figure 2.16, however, a few CGNR iterations are able to detect the layers at both ends of the interval.

As a final example, we consider model problem III with $\epsilon = 1e - 8$ which results in a relatively sharp interior layer at $x = 1/2$. Figure 2.17 demonstrates the results of 20 CGNR iterations with a varying number of mesh points. Once again, it appears that a moderate number of mesh points in the chosen grid works best. In this case 151 and 201 mesh points are sufficient to quickly indicate the presence of a region of interest in the solution.

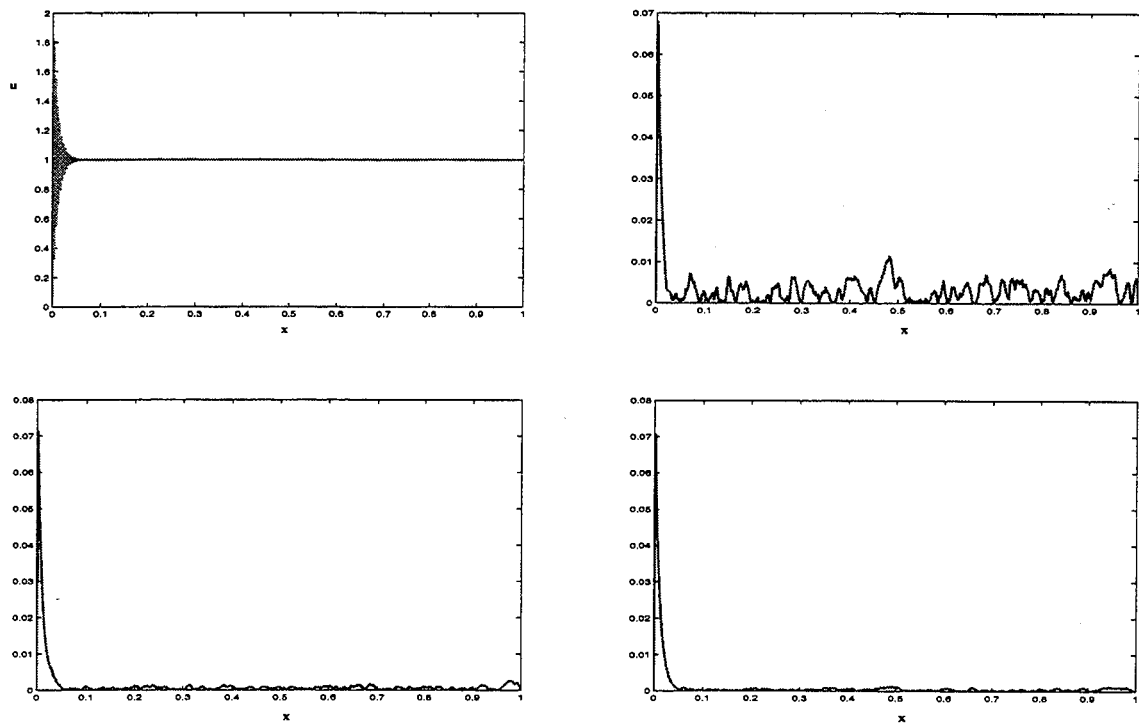


Figure 2.15: Solution of model problem I with $\epsilon = 1e - 4$ and $N = 501$ mesh points (top left); Difference of approximate solution and filtered approximate solution after 10, 20 and 30 iterations of CGNR (top right and bottom).

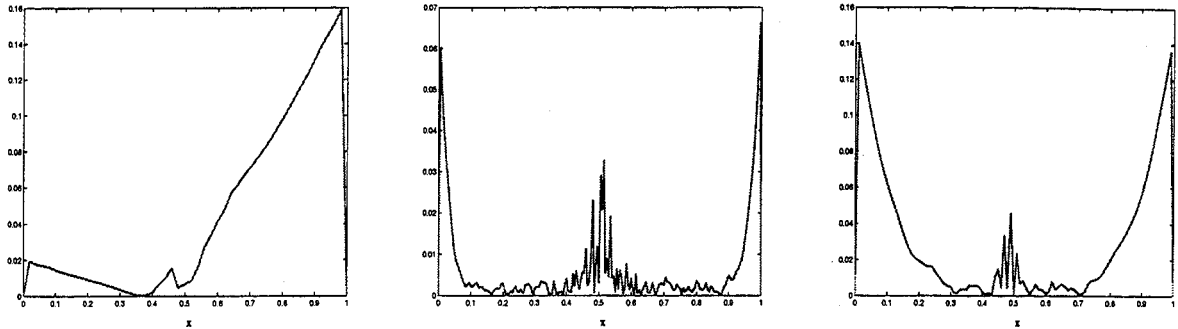


Figure 2.16: Difference of approximate solution and filtered approximate solution of model problem IV with $\epsilon = 1e - 4$ after 20 iterations of CGNR for $N = 51, 151,$ and 201 mesh points.

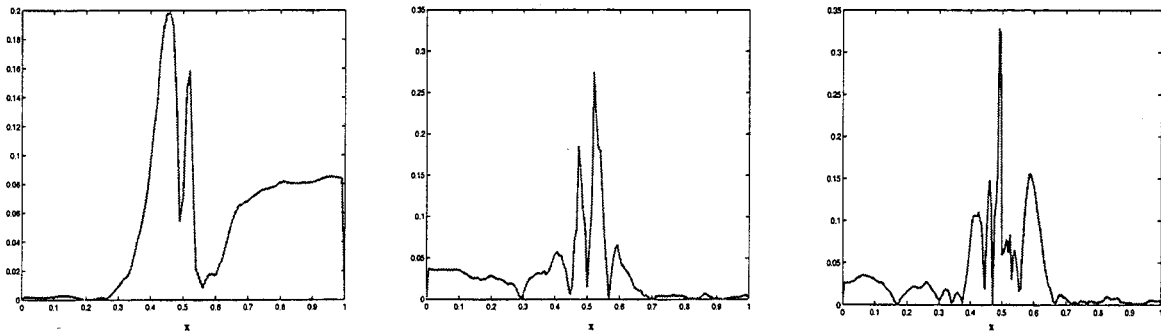


Figure 2.17: Difference of approximate solution and filtered approximate solution of model problem III with $\epsilon = 1e - 8$ after 20 CGNR iterations for $N = 101, 151$ and 201 mesh points.

We end this section with an important comment about the choice an initial guess for the iterative solver. For all the experiments presented here a random initial vector was used. In fact, the results depend heavily on this choice. In Figure 2.18 we repeat the exact same setup used to generate the last picture in Figure 2.16 except we use an initial guess of the constant vector $(1, 1, \dots, 1)^T$. The left plot is of u^{20} while the right plot is of the usual difference between u^{20} and its filtered value. The results are not nearly as impressive. In this case, the choice of initial guess coincides with the boundary value at $x = 1$. The iterations produced by CGNR detects that the early iterates agree with the boundary value and is happy to keep them constant on that end of the interval. The iterations do detect a

problem at $x = 0$ as it tries, unsuccessfully, to satisfy the boundary condition there resulting in the oscillations. With this choice of initial data CGNR would need to iterate almost to convergence to be able to deduce anything definitive from the iterates.

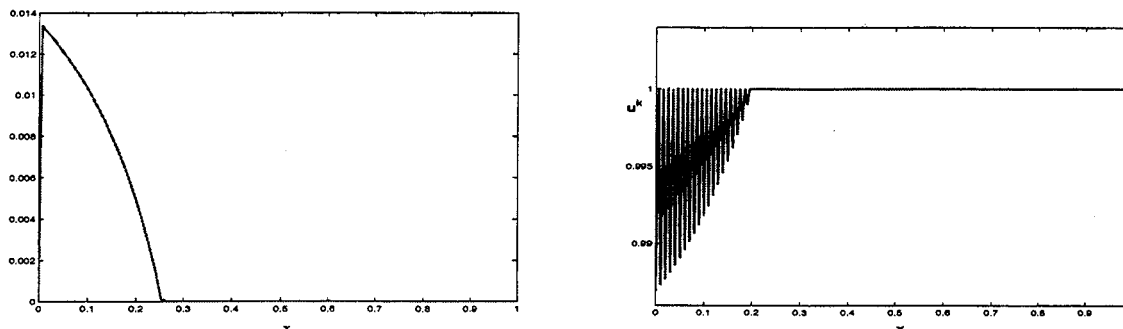


Figure 2.18: After 20 iterations of CGNR with a non-random initial guess.

2.4 M -Matrices

Ostrowski [77] introduced a rich class of matrices known as M -matrices in 1937. A matrix A is a nonsingular M -matrix if and only if A is nonsingular with $a_{ij} \leq 0$ for $i \neq j$ and $A^{-1} \geq 0$. There are many characterizations of M -matrices. Berman and Plemmons [12] give 50 different but equivalent definitions. A condition which is easy to check is that a matrix A is a nonsingular M -matrix if and only if $a_{ij} \leq 0$ for $i \neq j$ and A is generalized strictly diagonally dominant. A matrix is said to be generalized (strictly) diagonally dominant if there exists a diagonal matrix D with positive entries so that AD is (strictly) diagonally dominant¹. It is clear that a sufficient, but not necessary, condition for A to be a M -matrix is that A is strictly diagonally dominant with non-positive off-diagonal entries.

M -matrices have the nice property that if there exists a vector w with $Aw \geq 1$ (component-wise), then $\|A^{-1}\|_{\infty} \leq \|w\|_{\infty}$. With respect to discretizations of boundary value problems

¹A matrix C is diagonally dominant if

$$|c_{ii}| \geq \sum_{\substack{j=1 \\ j \neq i}}^n |c_{ij}| \quad \text{for } i = 1, \dots, n,$$

and strictly diagonally dominant if the equality is removed.

a bounded inverse is sufficient to prove stability of a discretization, that is we can show

$$\|u\| \leq C \|L^h u\|,$$

where L^h is the discrete version of the differential operator which defines the boundary value problem .

For our class of BVPs, equation (1.2) with $c(x) \geq 0$, discretized on a uniform grid with centered differences a sufficient condition to ensure A is a M -matrix is

$$h < \frac{2\epsilon}{\max |b_i|}. \quad (2.6)$$

This agrees, in the constant coefficient case, to the condition which guarantees that A has only real eigenvalues. Moreover, this assumption ensures that the computed solution is oscillation free, that is, all boundary layers are resolved.

In section 2.1.3, we commented on the connection between various discretizations and the eigenvalues that result on uniform grids. The discussion included the effect that upwinding and tuned upwinding had on the diagonal dominance properties of the linear system matrix. Indeed, in light of the definition of a M -matrix, it is clear, and is easily verified, that upwinding provides a linear system matrix which is a M -matrix independent of h and ϵ . Tuned upwinding yields M -matrices under various assumptions on the tuning parameter, see [83] for a nice discussion. In this section we investigate what effect the choice of mesh has on the M -matrix structure of the linear system matrix.

Consider the linear system matrix of model problem I with $\epsilon = 1e-2$. We discretize the problem with three grids; a uniform grid, a piecewise uniform grid and an equidistributed² grid, all of which resolve the boundary layer at $x = 0$, as shown in Figure 2.19. The plots on the left demonstrate the computed solution of the boundary value problem for the chosen grid. On the right of Figure 2.19 we indicate, by solid dots, those rows of the linear system matrix A which satisfy the local M -matrix conditions. That is, we indicate rows which are diagonally dominant and have nonpositive off-diagonal entries. The first two plots correspond to a uniform mesh chosen to resolve the boundary layer. This results in a linear system matrix which is a M -matrix. Therefore, a dot is drawn for each entry of every row of the 100×100 matrix. The quantity "nz" denotes the number of non-zero entries in the matrix.

²The concept of equidistribution was mentioned in section 2.1.2 and will be discussed further in Chapter 4

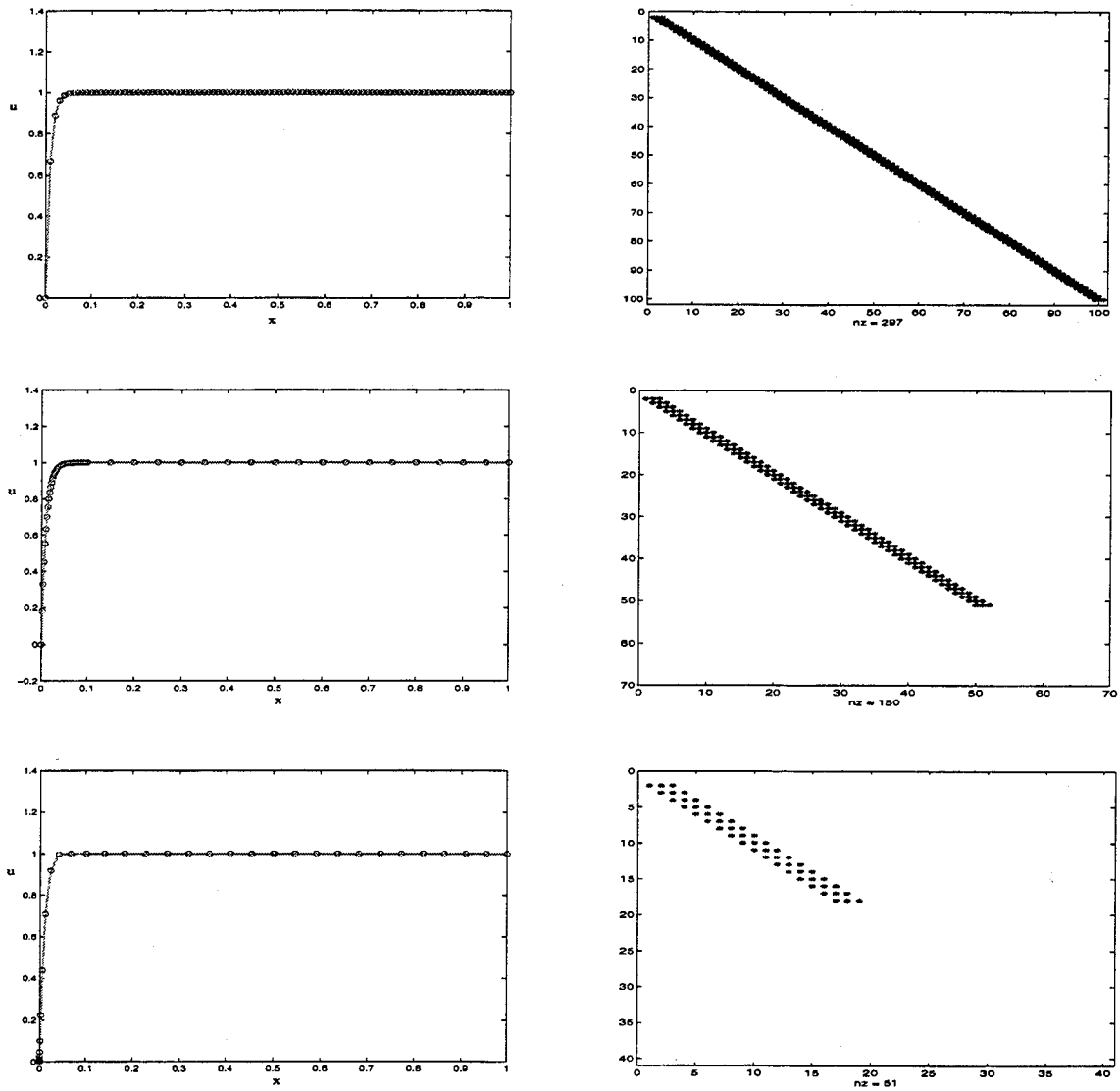


Figure 2.19: Solution and local M -matrix structure for model problem I on a uniform (top), piecewise uniform (middle) and equidistributed (bottom) grid.

Choosing a uniform grid for a constant coefficient BVP of the form (1.2) will result in a matrix where the diagonal dominance condition is either satisfied for all rows or none at all. This is basically saying that on a uniform mesh, if (2.6) is satisfied anywhere on the interval then it is satisfied everywhere. Of course, for nonuniform grids it is possible to have (2.6) satisfied locally, resulting in matrices which are locally like M -matrices. In the middle plots of Figure 2.19 we have used a piecewise uniform grid which satisfies (2.6) for points near the boundary layer. The large number of points in the refined portion of the mesh (relative to the total number of mesh points) results in a large number of rows included in the matrix plot. Moreover, there is a spatial connection between the rows of the matrix and the location of the mesh points on the interval. The equidistributed grid requires fewer points to sufficiently resolve the layer, resulting in fewer rows shown in the matrix plot.

The situation changes slightly for a variable coefficient problem. Figure 2.20 shows the results for model problem IV. Here we use an unresolved uniform grid (top), a piecewise uniform grid which resolves the right layer (middle), and a piecewise uniform grid which resolves both layers (bottom). For this example, we see local M -matrix structure corresponding to a grid which doesn't resolve the layer at all. Indeed, near $x = 1/2$ the coefficient of u' is approximately zero which for this problem relaxes the local requirement on the mesh size. In fact, the matrix appears to be a M -matrix except for rows corresponding to points in the immediate region of the unresolved layers. As we refine the mesh near $x = 0$ the entries in those rows become shaded dots in the matrix plot. And once both layers are resolved the matrix is an M -matrix.

As a final example, we consider another variable coefficient problem whose solution has an interior cusp at $x = 1/2$,

$$-\epsilon u'' - (2x - 1)u' + 0.5u = 0, \quad u(0) = 1/2, \quad u(1) = 1.$$

Here we keep the uniform grid spacing fixed at $h = 1/100$ and vary ϵ . As ϵ is decreased from $1e - 2$ to $5e - 6$, from top to bottom in Figure 2.21, we see that the number of rows of the matrix which satisfy the local M -matrix condition decreases.

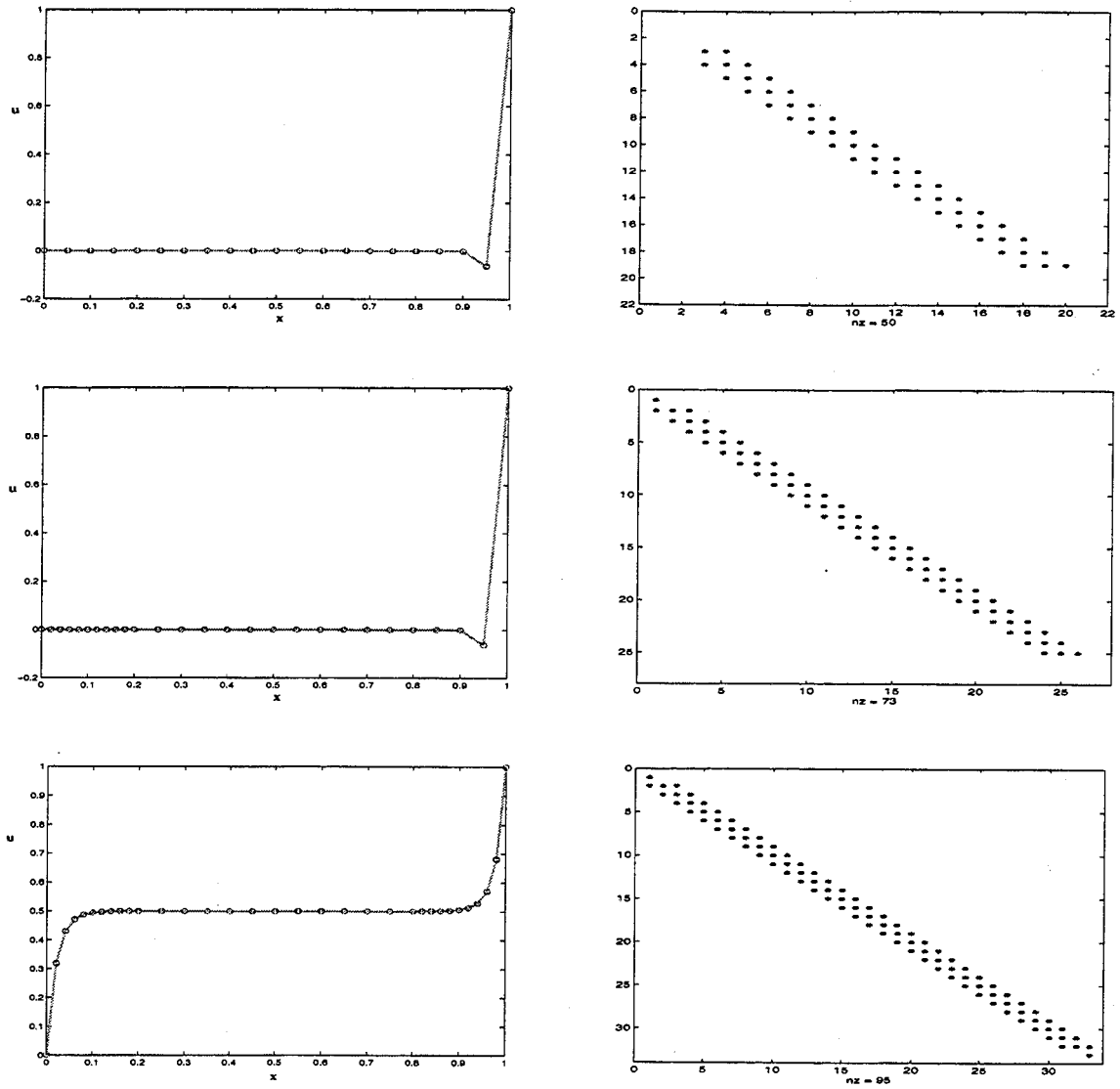


Figure 2.20: Solution and local M -matrix structure for model problem IV on a uniform grid (top), a piecewise uniform grid (middle), and a refined uniform grid (bottom).

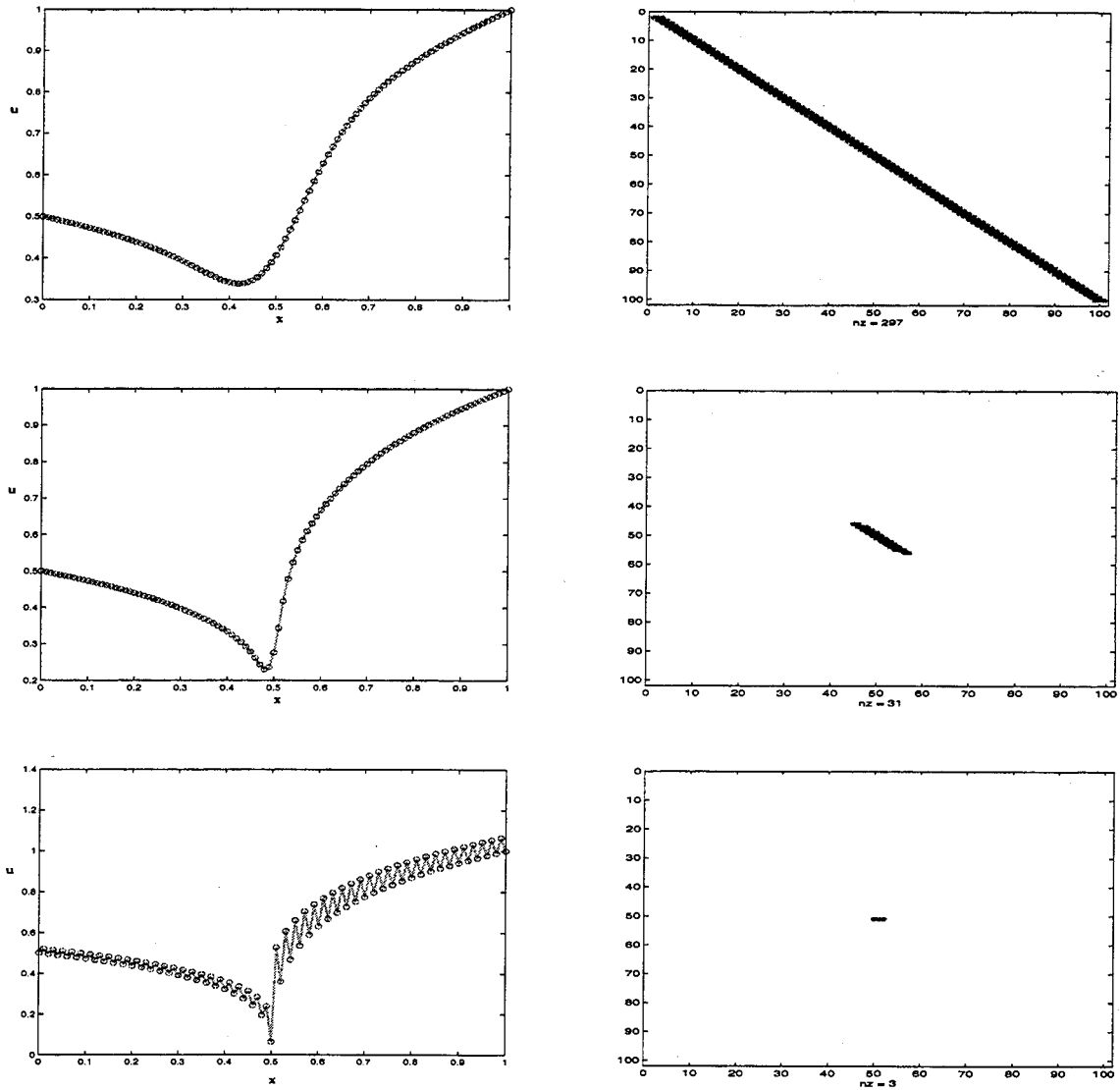


Figure 2.21: Solution and local M -matrix structure for a variable coefficient interior cusp problem with $\epsilon = 1e-2$ (top), $\epsilon = 5e-4$ (middle) and $\epsilon = 5e-6$ (bottom).

Chapter 3

A Matrix Inverse Problem

Consider a tridiagonal symmetric M -matrix

$$T = \begin{pmatrix} a_1 & -b_2 & & & \\ -b_2 & a_2 & -b_3 & & \\ & \ddots & \ddots & \ddots & \\ & & -b_{n-1} & a_{n-1} & -b_n \\ & & & -b_n & a_n \end{pmatrix}.$$

In accordance with the definition of a M -matrix provided in section 2.4 we will assume the entries a_i, b_i nonnegative and $a_i > b_i + b_{i+1}$, i.e. the matrix is strictly diagonally dominant.

The Cholesky factorization of T , given by $T = LD_L^{-1}L^T$ will exist if, for example, T is diagonally dominant. For tridiagonal matrices T it is possible to compute this factorization explicitly. To this end we let

$$L = \begin{pmatrix} \delta_1 & & & & \\ -b_2 & \delta_2 & & & \\ & \ddots & \ddots & & \\ & & -b_{n-1} & \delta_{n-1} & \\ & & & & \delta_n \end{pmatrix}, \text{ and } D_L = \begin{pmatrix} \delta_1 & & & & \\ & \delta_2 & & & \\ & & \ddots & & \\ & & & \delta_{n-1} & \\ & & & & \delta_n \end{pmatrix}.$$

Forming $LD_L^{-1}L^T$ and comparing the entries to those of T we obtain a recurrence for δ_i ,

$$\delta_1 = a_1, \quad \delta_i = a_i - \frac{b_i^2}{\delta_{i-1}}, \quad i = 2, \dots, n.$$

A UL factorization of T , given by $T = UD_U^{-1}U^T$, may be obtained in a similar manner. If we let

$$U = \begin{pmatrix} d_1 & -b_2 & & & \\ & d_2 & -b_3 & & \\ & & \ddots & \ddots & \\ & & & d_{n-1} & -b_n \\ & & & & d_n \end{pmatrix} \text{ and } D_U = \begin{pmatrix} d_1 & & & & \\ & d_2 & & & \\ & & \ddots & & \\ & & & d_{n-1} & \\ & & & & d_n \end{pmatrix},$$

then by simply multiplying and comparing entries to those of T we obtain the following recurrence for the entries of D :

$$d_n = a_n, \quad d_i = a_i - \frac{b_{i+1}^2}{d_{i+1}}, \quad i = n-1, n-2, \dots, 1. \quad (3.1)$$

Meurant [62] has used the Cholesky and UL factorizations of T to study the inverses of symmetric tridiagonal matrices and obtains the following result (see [62] and [63] for details of the proof):

The entries of the inverse of T are given explicitly as

$$T_{ij}^{-1} = b_{i+1} \cdots b_j \frac{d_{j+1} \cdots d_n}{\delta_i \cdots \delta_n}, \quad \text{for all } i, \text{ and } j > i, \quad (3.2)$$

and

$$T_{ii}^{-1} = \frac{d_{i+1} \cdots d_n}{\delta_i \cdots \delta_n}, \quad \text{for all } i.$$

Using (3.2) we now determine the rate of decay of the entries of T^{-1} . Computing directly we have

$$\frac{T_{ij}^{-1}}{T_{i,j+1}^{-1}} = \frac{d_{j+1}}{b_{j+1}}. \quad (3.3)$$

Under the assumption that T is strictly diagonally dominant it is easy to show from the recurrence relation (3.1) and induction that $d_i > b_i$. So we have

$$d_i = a_i - \frac{b_{i+1}^2}{d_{i+1}} > a_i - b_{i+1} > b_i,$$

or

$$\frac{d_i}{b_i} > \frac{a_i - b_{i+1}}{b_i}.$$

This allows us to bound $T_{i,j+1}^{-1}$ in terms of T_{ij}^{-1} as

$$T_{i,j+1}^{-1} < \frac{b_{j+1}}{a_{j+1} - b_{j+2}} T_{ij}^{-1}.$$

Moreover, (3.1) gives us a lower bound as well. We have

$$d_i = a_i - \frac{b_{i+1}^2}{d_{i+1}} < a_i,$$

from which we may show

$$T_{i,j+1}^{-1} > \frac{b_{j+1}}{a_{j+1}} T_{ij}^{-1}.$$

Therefore we have upper and lower bounds for $T_{i,j+1}^{-1}$ in terms of T_{ij}^{-1} :

$$\hat{\rho} T_{ij}^{-1} < T_{i,j+1}^{-1} < \rho T_{ij}^{-1}, \quad (3.4)$$

where

$$\rho = \max \frac{b_{j+1}}{a_{j+1} - b_{j+2}} \text{ and } \hat{\rho} = \min \frac{b_{j+1}}{a_{j+1}}. \quad (3.5)$$

It is clear that this may be extended to compare any off-diagonal entry of T^{-1} to the diagonal entry in that row,

$$\hat{\rho}^{|i-j|} T_{ii}^{-1} < T_{i,j}^{-1} < \rho^{|i-j|} T_{ii}^{-1}. \quad (3.6)$$

To compare T_{ij}^{-1} to an entry closer to the diagonal (along a row) we have

$$\hat{\rho}^{|k-j|} T_{ik}^{-1} \leq T_{ij}^{-1} \leq \rho^{|k-j|} T_{ik}^{-1}.$$

And to bound T_{ij}^{-1} in terms of an entry further from the diagonal (along a row) we have

$$\frac{1}{\rho^{|k-j|}} T_{ik}^{-1} \leq T_{ij}^{-1} \leq \frac{1}{\hat{\rho}^{|k-j|}} T_{ik}^{-1}.$$

Due to symmetry the estimates also work column wise. Care must be taken that the entries that are being compared are on the same side of the diagonal.

3.1 Positivity Subject to a Perturbation

In this section we investigate a positive perturbation of a M -matrix. Specifically we are interested in understanding how large the perturbation can be so that the perturbed matrix

retains the nonnegative inverse. The answer, it turns out, depends on the entries of the matrix M .

Specifically, we consider the matrix B given by

$$B = \begin{pmatrix} a_1 & -b_2 & h & & & & \\ -b_2 & a_2 & -b_3 & h & & & \\ h & -b_3 & a_3 & -b_4 & h & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & h & -b_{n-3} & a_{n-2} & -b_{n-1} & h \\ & & & h & -b_{n-1} & a_{n-1} & -b_n \\ & & & & h & -b_n & a_n \end{pmatrix}, \quad (3.7)$$

where a and b are positive, $h \neq 0$, and the tridiagonal part of B is an M -matrix. For what values of h is the inverse of B nonnegative?

If $h < 0$, then if h is chosen so that B is strictly diagonally dominant (or generalized strictly diagonally dominant), then B is still a M -matrix and will satisfy $B^{-1} \geq 0$. If $h > 0$ then the entries of B no longer satisfy the sign pattern necessary to be a M -matrix. Due to continuity, if h is chosen small enough then we would expect $B^{-1} \geq 0$. Our goal is to find a bound on h which will ensure a nonnegative inverse.

We begin by considering a simpler case. Suppose B is given as $B = M + E$, where M is the tridiagonal M -matrix with entries $\{-b_{i-1}, a_i, -b_i\}$ and $E = uv^T$ is a rank one matrix. We choose u and v so that the $(1, 3)$ entry of B is h . The vectors $u = (h, 0, \dots, 0)^T$ and $v = (0, 0, 1, 0, \dots, 0)^T$ give the correct matrix. Now to get a feeling for what B^{-1} looks like we use the Sherman-Morrison formula,

$$B^{-1} = (M + uv^T)^{-1} = M^{-1} - \frac{M^{-1}uv^T M^{-1}}{1 + v^T M^{-1}u}.$$

A quick calculation shows that $v^T M^{-1}u = hM_{31}^{-1}$ and EM^{-1} is a matrix whose first row is hM_3^{-1} and the rest zeros. The quantity M_3^{-1} denotes the third row of M^{-1} .

So $B^{-1} \geq 0$ if

$$M^{-1} \begin{pmatrix} - & \frac{h}{1+hM_{31}^{-1}}M_3^{-1} & - \\ 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{pmatrix} \leq M^{-1}. \quad (3.8)$$

We will now use the decay estimates (3.6) to find an upper bound on h to ensure $B^{-1} \geq 0$. In addition to the decay estimates we also require a bound on the diagonal entries of M^{-1} . Ostrowski [78] obtained the following bound on M_{ii}^{-1} for strictly row diagonally dominant matrices:

$$M_{ii}^{-1} \leq \frac{1}{|M_{ii}|(1 - \xi_i)},$$

where

$$\xi_i = \frac{1}{M_{ii}} \sum_{i \neq j} |M_{ij}|.$$

These bounds have been tightened by Nabben [72] in the case of nonsymmetric diagonally dominant tridiagonal matrices. For notational convenience in the discussion which follows, we introduce μ as an upper bound on the diagonal entries of M^{-1} .

The (1,1) entry of $M^{-1}EM^{-1}$ is $sM_{11}^{-1}M_{31}^{-1}$ where $s = h/(1 + hM_{31}^{-1})$. So comparing the (1,1) entries of $M^{-1}EM^{-1}$ and M^{-1} we require

$$sM_{11}^{-1}M_{31}^{-1} \leq M_{11}^{-1}.$$

Using the decay estimates we have the following sequence of inequalities,

$$sM_{11}^{-1}M_{31}^{-1} \leq s\rho^2 M_{11}^{-1}M_{33}^{-1} \leq s\rho^2 \mu M_{11}^{-1} \leq M_{11}^{-1}.$$

This indicates that $s \leq 1/\mu\rho^2$ is a sufficient requirement.

For the (1,2) entries we have to show

$$sM_{11}^{-1}M_{32}^{-1} \leq M_{12}^{-1}.$$

In this case, the sequence of inequalities

$$sM_{11}^{-1}M_{32}^{-1} \leq s\rho M_{11}^{-1}M_{33}^{-1} \leq s\rho\mu M_{11}^{-1} \leq \hat{\rho}M_{11}^{-1} \leq M_{12}^{-1}$$

implies that $s \leq \hat{\rho}/\mu\rho$ is sufficient.

To compare the (1, k) entries for $k \geq 3$ we note that $M_{3k}^{-1} > M_{1k}^{-1}$ since the M_{3k}^{-1} entries are closer to the diagonal. Taking advantage of symmetry we use the decay estimates column-wise to obtain

$$M_{3k}^{-1} \leq \frac{M_{1k}^{-1}}{\hat{\rho}^2}$$

and therefore

$$sM_{11}^{-1}M_{3k}^{-1} \leq s\frac{\mu}{\hat{\rho}^2}M_{1k}^{-1} \leq M_{1k}^{-1} \quad \text{if } s \leq \hat{\rho}^2/\mu.$$

Now consider the j -th row, for $j \geq 2$. We want to show

$$sM_{j1}^{-1}M_{3k}^{-1} \leq M_{jk}^{-1}, \quad \text{for all } k = 1, \dots, n.$$

If $k \leq j$ we compare M_{j1}^{-1} to M_{jk}^{-1} which is closer to the diagonal along a row. So we have for $k \leq j$,

$$sM_{j1}^{-1}M_{3k}^{-1} \leq s\mu\rho^{k-1}M_{jk}^{-1} \leq M_{jk}^{-1}, \quad \text{if } s \leq \frac{1}{\mu\rho^{k-1}}.$$

If $k > j$ then we compare M_{3k}^{-1} to M_{jk}^{-1} which is closer to the diagonal along a column. So we have

$$sM_{j1}^{-1}M_{3k}^{-1} \leq s\mu\rho^{j-3}M_{jk}^{-1}, \quad \text{if } s \leq \frac{1}{\mu\rho^{j-3}}.$$

The tightest restriction found on s was that $s \leq \hat{\rho}/\mu\rho$. Therefore, subject to a rank-1 perturbation, a sufficient condition to ensure B^{-1} is nonnegative is that

$$h \leq \frac{\hat{\rho}}{\mu\rho}.$$

3.1.1 Higher Rank Perturbations

To generalize the result of the previous section we consider a perturbation given by $E = UV$ where $U = hI$ and V is a matrix of zeros except for a second superdiagonal of ones. We use an extension of Shermann-Morrison which says, if $I + VM^{-1}U$ is nonsingular then

$$(M + UV)^{-1} = M^{-1} - M^{-1}U(I + VM^{-1}U)^{-1}VM^{-1}.$$

Rows 1 through $n-2$ of VM^{-1} are just the $n-2$ rows of M^{-1} . Rows $n-1$ and n are zeros.

To ensure $B^{-1} = (M + UV)^{-1}$ is nonnegative we require

$$M^{-1}U(I + VM^{-1}U)^{-1}VM^{-1} \leq M^{-1}.$$

Under a suitable assumption on h it possible to show using a Neumann expansion that the inverse of

$$C^{-1} = (I + VM^{-1}U)^{-1} = (I + hVM^{-1})^{-1} \leq I. \quad (3.9)$$

From this we may deduce

$$M^{-1}C^{-1}VM^{-1} \leq M^{-1}VM^{-1}.$$

The bound on h which we obtain below is sufficient to ensure (3.9) is valid.

We wish to find a bound on h so that

$$M^{-1} \begin{pmatrix} - & hM_3^{-1} & - \\ - & hM_4^{-1} & - \\ \vdots & \ddots & \vdots \\ - & hM_{n-2}^{-1} & - \\ 0 & \cdots & 0 \\ 0 & \cdots & 0 \end{pmatrix} \leq M^{-1}. \quad (3.10)$$

Computing directly we find the (ij) -th entry of $P = M^{-1}VM^{-1}$ is given by

$$P_{ij} = \sum_{k=1}^{n-2} hM_{ik}^{-1}M_{k+2,j}^{-1}.$$

Using the bounds from the previous section we have

$$hM_{ik}^{-1}M_{k+2,j}^{-1} \leq \frac{1}{(n-2)}M_{ij}^{-1}, \quad \text{if } h \leq \frac{\hat{\rho}}{(n-2)\mu\rho},$$

from which (3.10) follows directly.

3.1.2 A Symmetric Perturbation

We now consider a symmetric rank 2 perturbation, that is we only perturb the $(1,3)$ and $(3,1)$ entries of M by a quantity h . Let V be the matrix of zeros except for ones in the $(1,3)$ and $(3,1)$ positions. Once again the generalized Sherman–Morrison formula guarantees that $B^{-1} \geq 0$ if

$$hM^{-1}(I + hVM^{-1})^{-1}VM^{-1} \leq M^{-1}. \quad (3.11)$$

The structure of $I + hVM^{-1}$ again allows us to deduce

$$M^{-1}(I + hVM^{-1})^{-1}VM^{-1} \leq M^{-1}VM^{-1}.$$

In this case the (ij) -th entry of $hM^{-1}VM^{-1}$ is given by

$$h(M_{i1}^{-1}M_{3j}^{-1} + M_{i3}^{-1}M_{1j}^{-1}).$$

Therefore, inequality (3.11) will be satisfied if

$$h(M_{i1}^{-1}M_{3j}^{-1} + M_{i3}^{-1}M_{1j}^{-1}) \leq M_{ij}^{-1}.$$

Applying the decay estimates as in the previous sections we deduce that

$$h \leq \frac{\hat{\rho}}{2\mu\rho}$$

is a sufficient bound on the size of the perturbation.

3.1.3 Extensions

We are now in a position to return to the matrix B of (3.7). Writing B as $M + E$ we see that E is a combination of the perturbations discussed in the previous two sections. It will come as no surprise that a bound on h to ensure $B^{-1} \geq 0$ may be derived in a similar way to obtain the following result.

The matrix B from equation (3.7) will have a nonnegative inverse if

$$h \leq \frac{\hat{\rho}}{2(n-2)\mu\rho}.$$

This is a sufficient but not necessary condition. In fact it may be possible to improve the bounds ρ , $\hat{\rho}$ and μ using the improved decay estimates of Nabben [72].

As a final comment, we note that our development does not depend in any way on the symmetry of the matrix M . M is only required to be a tridiagonal M -matrix. In the nonsymmetric case, the decay estimates of Nabben [71] and [72] would assist in extending the result. Although notationally more cumbersome, the arguments may be adapted to consider nonconstant, non-symmetric, positive perturbations of the form

$$E = \begin{pmatrix} 0 & 0 & hf_1 & & & & & \\ 0 & 0 & 0 & hf_2 & & & & \\ hg_1 & 0 & 0 & 0 & hf_3 & & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & & \\ & & hg_{n-4} & 0 & 0 & 0 & hf_{n-2} & \\ & & & hg_{n-3} & 0 & 0 & 0 & \\ & & & & hg_{n-2} & 0 & 0 & \end{pmatrix},$$

where f and g are nonnegative vectors.

3.2 An Application

The bounds developed in the previous sections will now allow us to comment on a time stepping strategy for higher order degenerate diffusion equations.

A model [41] of thin liquid films and fluid interfaces driven by surface tension is given (in 1-D) by the degenerate diffusion equation

$$h_t + (f(h)h_{xxx})_x = 0, \quad (3.12)$$

where $f(h) \sim h^n$ as $h \rightarrow 0$. The power of n is determined by the boundary conditions on the liquid–solid interface. In all applications a physical solution requires h to be nonnegative. It is known that if $n = 0$, in which case (3.12) is the linear fourth order heat equation, negative solutions will result from positive initial data. For larger values of n this is not the case. In 1-D, positive solutions result from positive initial data if $n \geq 3.5$ [14]. Numerical simulations ([14],[3] and [13]) by Bertozzi et al. demonstrate that for smaller values of n the solutions develop singularities of the form $h \rightarrow 0$.

Numerically, one wishes to preserve the positivity of the continuous solution and to resolve any singularities which result. Zhornitskaya and Bertozzi [103] propose discretizing (3.12) in space by

$$y_{i,t} + (a(y_{i-1}, y_i) y_{\bar{x}\bar{x}\bar{x},i})_x = 0, \quad (3.13)$$

for $i = 0, 1, \dots, N - 1$ with $y_i(0) = h_0(x_i)$. The quantities $y_{x,i}$ and $y_{\bar{x},i}$ denote forward and backward differences respectively, and $y_{\bar{x}\bar{x}\bar{x},i}$ is a composition of these differences performed in the usual way.

For $n \geq 2$ the choice

$$a(s_1, s_2) = \begin{cases} \frac{s_1 - s_2}{G'(s_1) - G'(s_2)} & \text{if } s_1 \neq s_2, \\ f(s_1) & \text{if } s_1 = s_2, \end{cases} \quad (3.14)$$

where $G''(s) = 1/f(s)$ is shown to preserve positivity of solutions. For $n < 2$ a positivity-preserving scheme is obtained by discretizing, as above, a suitable regularization of the PDE.

In this section, we propose a time stepping strategy which will retain the positivity-preserving features of the semi-discrete equations (3.13) and (3.14) while only requiring linear solves at each time step. The basic idea is to discretize (3.13) in time by treating the linear parts implicitly and the nonlinear parts explicitly

$$\frac{y_i^{n+1} - y_i^n}{\Delta t} + (a(y_{i-1}^n, y_i^n) y_{\bar{x}\bar{x}\bar{x},i}^{n+1})_x = 0. \quad (3.15)$$

A similar scheme was proposed by Hoff to solve the one-dimensional porous medium equation [46]. Computing y^{n+1} from y^n requires a linear solve

$$By^{n+1} = y^n,$$

where B is a nonsymmetric pentadiagonal matrix of the form (3.7) considered in the previous section. Positivity will be preserved if $B^{-1} \geq 0$. Ensuring that the tridiagonal part of B

is a M -matrix and the perturbation is small enough imposes a constraint on the time step Δt to retain positivity. We note that this constraint is sufficient but not necessary. In fact, y^{n+1} may remain positive even if some entries of B^{-1} are negative. More work is required to consider the effect of boundary conditions on the positivity of B^{-1} .

Part II

Numerical Integration, Moving Meshes and Schwarz Waveform

Discretizing parabolic PDEs spatially result in systems of ordinary differential equations. Integrating in time using ODE software is the well-known method of lines (MOL). Successful implementation requires a good choice of spatial mesh, discretization method, and ordinary differential equation solver.

In problems from chemical kinetics, network analysis, radio frequency applications, and very large scale integrated (VLSI) circuits, many authors have investigated the application of multirate integration techniques. Direct integration methods require that every differential equation be discretized (in time) identically. This results in time-step selection dictated by fastest changing components. Multirate methods attempt to circumvent this restriction by allowing the slow changing components to be integrated using large time steps, while concentrating the computational effort in dealing with the fast components.

Chapter 4

Moving Mesh Methods

Adaptive mesh methods for partial differential equations typically fall into one (or more) of the following broad categories:

- r -refinement: moving a fixed number of mesh points to difficult regions of the physical domain,
- p -refinement: varying the order of the numerical method to adapt to local solution features,
- h -refinement: uniform mesh refinement where resolution is inadequate.

The r -refinement and h -refinement methods mentioned above may be applied in either a static or dynamic fashion. Static methods involve refining/coarsening or redistributing nodes at fixed times during a calculation. Dynamic (or moving mesh) methods solve for the solution and mesh simultaneously. This requires specification of a mesh equation which concentrates nodes in regions of rapid variation of the solution. The equidistribution principle (EP), first introduced by de Boor [22], provides a mechanism for mesh movement. The EP requires that nodes are selected so that some measure of the solution error is equally distributed over all subintervals of the physical domain.

4.1 Equidistribution and a Moving Mesh PDE

We will consider the solution of a PDE of the form

$$u_t = \mathcal{L}(u) \quad 0 < x < 1, \quad t > 0, \quad (4.1)$$

subject to appropriate initial and boundary conditions. In (4.1), \mathcal{L} denotes a spatial differential operator in the physical coordinate x . If this PDE is particularly difficult to solve we may wish to introduce a computational coordinate ξ by a one-to-one (time dependent) coordinate transformation

$$x = x(\xi, t), \quad \xi \in [0, 1], \quad \text{with } x(0, t) = 0, \quad x(1, t) = 1.$$

If this mesh transformation is chosen appropriately then a simple uniform mesh

$$\xi_i = \frac{i}{N}, \quad i = 0, 1, \dots, N \quad (4.2)$$

will suffice in the computational domain. The mesh transformation will then specify the corresponding physical mesh by $x_i(t) = x(\xi_i, t)$ for $i = 0, 1, \dots, N$.

The mesh transformation is specified by the EP written in integral form [99] as

$$\int_0^{x(\xi, t)} M(\bar{x}, t) d\bar{x} = \xi\theta(t), \quad (4.3)$$

where

$$\theta(t) = \int_0^1 M(\bar{x}, t) d\bar{x}, \quad (4.4)$$

and $M(x, t)$ is a chosen monitor function which provides a measure of the error in the numerical solution. The quantity $\theta(t)$ is a measure of the total error in the numerical solution at time t . We may arrive at (4.3) as follows. Requiring the error to be the same on each subinterval $[x_{j-1}(t), x_j(t)]$ is equivalent to

$$\int_{x_{j-1}}^{x_j} M(\bar{x}, t) d\bar{x} = \frac{1}{N}\theta(t).$$

Adding this expression for $j = 1, \dots, i$ gives (4.3) for each value of i .

In [82],[48],[49], Huang, Ren, and Russell develop continuous moving mesh partial differential equations (MMPDEs) based on differentiating (4.3). To avoid the computationally inconvenient $\theta(t)$ we may differentiate twice with respect to ξ to obtain

$$\frac{\partial}{\partial \xi} \left\{ M(x(\xi, t), t) \frac{\partial}{\partial \xi} x(\xi, t) \right\} = 0. \quad (4.5)$$

Since (4.5) does not involve the node speed \dot{x} , it is referred to as a quasi-static EP (QSEP). Requiring the mesh to satisfy this QSEP at some later time $t + \tau$ results in several other

MMPDEs. A particularly useful mesh equation derived in this way, which we use in our calculations, is

$$\frac{\partial}{\partial \xi} \left(M \frac{\partial \dot{x}}{\partial \xi} \right) = -\frac{1}{\tau} \frac{\partial}{\partial \xi} \left(M \frac{\partial x}{\partial \xi} \right). \quad (\text{MMPDE4})$$

A popular choice for $M(x, u, t)$ is the arclength monitor function

$$M(x, u, t) = \sqrt{1 + (\partial u / \partial x)^2}. \quad (4.6)$$

This choice is based on the premise that we expect the error in the numerical solution to be largest in regions where the solution has large gradients. The choice of monitor function is often problem dependent. For example, Budd, Huang, and Russell [16], develop a monitor function which preserves the scaling invariance of the underlying physical PDE. In fact, examples are given for so-called blow-up problems where the arclength monitor function fails to adequately resolve the solution near the singularity time. A regularized monitor function

$$M(x, u, t) = \sqrt{1 + \frac{1}{\alpha} |u_x|^2} \quad (4.7)$$

was introduced by Mackenzie and Beckett [10] to solve singularly perturbed reaction diffusion equations and by Stockie, Mackenzie, and Russell [94] to solve hyperbolic problems. The parameter α may be solution dependent and is used to balance the number of mesh points in the sharp layer with those in the rest of the domain. This works particularly well for one-dimensional conservation laws. In [10] the authors also introduce another regularized monitor function which depends on the curvature of the solution. Moving Finite Element methods [68], [67] use the solution residual, $u_t - \mathcal{L}u$ as a monitor function.

In higher dimensions Huang and Russell [50],[19] develop a moving mesh equation based on minimizing a functional of the mesh coordinate mappings. In this situation the monitor function is replaced by a monitor matrix.

The mesh transformation $x = x(\xi, t)$ suggests that we should rewrite the physical PDE in Lagrangian form. The total time derivative \dot{u} is given as

$$\dot{u} = u_t + u_x \dot{x}.$$

This allows (4.1) to be written as

$$\dot{u} - u_x \dot{x} = \mathcal{L}u. \quad (4.8)$$

Equation (4.8) and (MMPDE4) are solved simultaneously for the mesh and corresponding solution.

4.2 Discretization and Solution Process

As outlined in the previous section, using a moving mesh method to solve a PDE requires solving a coupled system of nonlinear PDEs (even if the original PDE is linear!). For purposes of the development of our Schwarz Waveform Moving Mesh Method in Chapter 6, we will restrict ourselves to the moving mesh equation (MMPDE4).

The resulting system of coupled PDEs to be solved on the uniform mesh (4.2) is given as

$$\begin{aligned} \dot{u} - u_x \dot{x} &= \mathcal{L}(u) \\ (M(x, u, t) \dot{x}_\xi)_\xi &= -\frac{1}{\tau} (M(x, u, t) x_\xi)_\xi. \end{aligned} \quad (4.9)$$

Discretizing spatially with centered differences we obtain the semi-discrete approximation

$$\begin{aligned} \dot{u}_i - \left(\frac{u_{i+1} - u_{i-1}}{x_{i+1} - x_{i-1}} \right) \dot{x}_i &= f_i \\ \frac{M_{i+1} + M_i}{2(1/N)^2} (\dot{x}_{i+1} - \dot{x}_i) - \frac{M_i + M_{i-1}}{2(1/N)^2} (\dot{x}_i - \dot{x}_{i-1}) &= \\ -\frac{1}{\tau} \left\{ \frac{M_{i+1} + M_i}{2(1/N)^2} (x_{i+1} - x_i) - \frac{M_i + M_{i-1}}{2(1/N)^2} (x_i - x_{i-1}) \right\} \end{aligned} \quad (4.10)$$

for $i = 1, \dots, N-1$. The right hand side of the semi-discrete physical PDE, f_i , denotes the discrete approximation to the spatial operator \mathcal{L} at $\xi = \xi_i$. Centered differences are used to discretize \mathcal{L} . The quantity M_i denotes a centered difference approximation to the monitor function at x_i . In actual calculations, M_i is replaced by its smoothed value \tilde{M}_i defined by

$$\tilde{M}_i = \sqrt{\frac{\sum_{k=i-p}^{i+p} (M_k)^2 \left(\frac{\gamma}{\gamma+1} \right)^{|k-i|}}{\sum_{k=i-p}^{i+p} \left(\frac{\gamma}{\gamma+1} \right)^{|k-i|}}}.$$

The role of spatial smoothing is discussed by Dorfi and Drury [25]. For our purposes we choose the smoothing parameters $\gamma = 2$ and $p = 2$. See [48] for comments concerning the choice of γ and p .

To close the system we specify boundary conditions for the mesh and solution. Typically, initial and boundary conditions of the physical PDE come from the problem description. On a fixed interval we specify $\dot{x}_0 = \dot{x}_N = 0$ as boundary conditions for the mesh. If the initial solution $u(x, 0)$ does not have steep layers then an initial uniform mesh for $x(\xi, 0)$ is

normally sufficient. If the initial solution has features which are not resolved on a uniform mesh for the chosen value of N , then an equidistributed initial mesh is required.

One way to accomplish an initial equidistributed mesh is to solve the PDE

$$u_t = u_0(x), \quad u(x, 0) = 0,$$

coupled with a moving mesh equation. Solving this system over the time interval $0 \leq t \leq 1$ with a uniform initial mesh will yield the solution $u(x, 1) = u_0(x)$ and the resulting mesh will equidistribute the initial solution. To avoid propagating any errors from this calculation, the initial solution is re-evaluated at the new initial mesh.

The semi-discrete approximation (4.10) describes a large system of linearly implicit ODEs which may be written in the form

$$V(y)y' = g(y). \quad (4.11)$$

The vector y denotes the unknowns ordered as $(u_0, x_0, u_1, x_1, \dots, u_N, x_N)^T$ and $V(y)$ is a matrix which depends on the unknown solution and mesh.

Solving this system of ODEs in time results in the Moving Method of Lines. Standard implementations solve this stiff system of ODEs with fully implicit time stepping provided by packages such as DASSL [80]. For our purpose, we will solve (4.11) using Backward Euler,

$$V(y_n)(y_n - y_{n-1}) - h_n g(y_n) = 0,$$

with local error control provided by step doubling. The resulting system of nonlinear algebraic equations are solved with modified Newton using strategies of Gustafsson and Söderlind [42] and Alexander [1] to control factorizations and evaluations of the Jacobian matrix. The Jacobians are evaluated using finite difference methods adapted to sparse matrices [21].

4.3 Other Implementation Strategies

Solving for the mesh and solution directly from (4.11) is prohibitively expensive for problems in higher dimensions. In [59],[10] and [47] the authors propose a decoupling procedure which allows the mesh and solution to be computed in an iterative fashion.

Decoupling the mesh equation from the physical PDE has the obvious benefit of reducing the size of the linear algebra problem for the Newton step by half. Furthermore, by

alternating the solves for the x and y grids the structure of the linear algebra problem is made suitable for fast, direct, block tridiagonal solves.

Computing the mesh and solution separately allows the user complete control over the local error tolerances for the ODE integrator, as well as different convergence criteria in the iterative algorithm. This is motivated by the observation that the mesh need not always be computed to the accuracy of the solution. If the iteration between the mesh equation and the physical PDE is continued long enough then grid lag is a serious concern [11]. This is avoided in [10] by allowing up to four iterations in the algorithm. This appears to be sufficient for most examples.

In Chapter 6, we propose and describe the implementation of a Schwarz Waveform Moving Mesh Method in one-dimension. This method decouples (4.11) spatially, and solves the mesh equation and physical PDE over all subdomains using a domain decomposition strategy.

Chapter 5

Decoupled Integration and Multirate Methods

In this chapter we consider novel numerical approaches for solving time dependent differential equations. Of particular interest are strategies which allow solution components or groups of solution components to be integrated with different time steps (multirate methods) or even different numerical methods. The discussion presented here is admittedly incomplete; however, we hope to provide some indication of the methods which are available for ODEs and PDEs.

We begin by describing the decoupled implicit Euler and backward differentiation formulas of Skelboe. Fully implicit decoupled integration methods naturally lead to waveform relaxation. Associated with any decoupled integration method is a partitioning strategy which groups components of the solution together in subsystems. In the context of implicit time stepping we will mention a couple of partitioning methods based on the Jacobian of the nonlinear algebraic equations.

We will then return to PDEs and discuss two classes of methods, *hp*-refinement techniques, and domain decomposition based methods.

5.1 ODE methods

5.1.1 Decoupled Integration Formulas

We will consider the problem of solving first order systems of ordinary differential equations

$$y' = f(y), \quad y \in \mathbb{R}^s. \quad (5.1)$$

Multirate methods attempt to benefit from the inherent time scales in the problem by separating (5.1) into subsystems. Each subsystem is then integrated using possibly different time steps and even numerical methods. We will use the methods of Skelboe to demonstrate the basic idea and conclude the section with some related approaches.

Skelboe ([92], [90],[86],and [91]) considers decoupled implicit Euler and backward differentiation formulas. Suppose a system of the form

$$y' = f(t, y), \quad y(t_0) = y_0, \quad t \geq t_0,$$

where $y : \mathbb{R} \rightarrow \mathbb{R}^s$, and $f : \mathbb{R} \times \mathbb{R}^s \rightarrow \mathbb{R}^s$, is partitioned as

$$\begin{pmatrix} \dot{y}_1 \\ \dot{y}_2 \\ \vdots \\ \dot{y}_q \end{pmatrix} = \begin{pmatrix} f_1(t, y) \\ f_2(t, y) \\ \vdots \\ f_q(t, y) \end{pmatrix}, \quad y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_q \end{pmatrix}.$$

Here $y_r : \mathbb{R} \rightarrow \mathbb{R}^{s_r}$, $f_r : \mathbb{R} \times \mathbb{R}^{s_r} \rightarrow \mathbb{R}^{s_r}$, with $\sum_{i=1}^q s_i = s$ and the initial condition is partitioned as $y(t_0) = (y_{1,0}, y_{2,0}, \dots, y_{q,0})^T$.

The partitioning requires subsystems which consist of strongly interacting components with weak couplings to the other subsystems. This is made precise with the concept of *monotonic max-norm stability* [86]. This condition is necessary to ensure stability of the decoupled integration method.

The decoupled implicit Euler method results by applying backward Euler to each of the q subsystems, and it is written as

$$y_{r,n} = y_{r,n-1} + h_n f_r(t_n, \tilde{y}_{1,n}, \dots, \tilde{y}_{r-1,n}, y_{r,n}, \tilde{y}_{r+1,n}, \dots, \tilde{y}_{q,n}) \quad (5.2)$$

for $n = 1, 2, \dots$ and $r = 1, \dots, q$.

The values $\tilde{y}_{i,n}$ are taken as convex combinations of $\{y_{i,k} \mid k \geq 0\}$ for $i \neq r$, and represent communication between subsystems. Convexity is required to ensure stability. The $\tilde{y}_{i,n}$ are

often chosen by zero order interpolation, $\tilde{y}_{i,n} = y_{i,k}$, or zero order extrapolation $\tilde{y}_{i,n} = y_{i,k+1}$ when $t_{r,n} \in (t_{i,k}, t_{i,k+1})$. Weaker stability but improved accuracy is possible by choosing $\tilde{y}_{i,n}$ by first order extrapolation as

$$\tilde{y}_{i,n} = y_{i,n-1} + \frac{h_n}{h_{n-1}}(y_{i,n-1} - y_{i,n-2}).$$

It is clear that (5.2) may be solved in parallel since communication is only required after each completed step. On a sequential computer, it would be appropriate to use a Gauss-Seidel formulation

$$y_{r,n} = y_{r,n-1} + h_n f_r(t_n, y_{1,n}, \dots, y_{r-1,n}, y_{r,n}, \tilde{y}_{r+1,n}, \dots, \tilde{y}_{q,n}) \quad (5.3)$$

for $n = 1, 2, \dots$ and $r = 1, \dots, q$.

Computing the next step for subsystem r requires the previously computed values on subsystems $1, 2, \dots, r-1$.

Skelboe [90] describes two general categories of decoupled integration formulas, semi-implicit and fully implicit. Equations (5.2) and (5.3) are examples of semi-implicit formulas. The discretization is implicit at the level of the subsystem but explicit with respect to the rest of the system. The method is semi-implicit since the $\tilde{y}_{i,n}$ values are chosen as either convex combinations of previously computed values of $y_{i,k}$ or zero order extrapolated values. This extrapolation can lead to inaccuracies. Fully-implicit decoupled formulas are defined by a compound step $[t_{N-1}, t_N]$ with each subsystem having discretization points at t_{N-1} and t_N . The discretization of all subsystems must be computed simultaneously over a number of time steps, resulting in large system of nonlinear equations. In this case, all the $\tilde{y}_{i,k}$ values are computed by interpolation of solution values within the compound step. The *waveform relaxation* method, discussed in the next section, may be interpreted as an iterative solution method for the large system of nonlinear equations resulting from a fully implicit decoupled integration step.

As mentioned, semi-implicit decoupled integration formulas limit the implicitness to individual subsystems. This reduces the dimension of the system of nonlinear equations for $y_{r,n}$ and the subsequent linear algebra involved in computing the Newton step.

In [92], Skelboe considers more general backward differentiation multirate formulas applied to separated systems of ODEs of the form

$$y' = f(t, y, z) \quad y(t_a) = y_a, \quad y \in \mathbb{R}^N \quad (5.4)$$

$$z' = g(t, y, z) \quad z(t_a) = z_a, \quad z \in \mathbb{R}^M. \quad (5.5)$$

We define the k -step linear multistep operator L_k as

$$L_k[y(t); h] = \sum_{r=0}^k \alpha_r y(t - rh) + h\beta_r y'(t - rh).$$

If y and z solve (5.4) and (5.5), respectively, then we assume

$$\|L_k[y(t); h]\| \gg \|L_k[z(t); h]\|$$

for all $t \in [t_a, t_b]$. This implies we may use L_k to integrate (5.5) using larger time steps than for (5.4). It is appropriate to use this formulation when $N \ll M$ or g is much more expensive to evaluate than f .

The multirate formulation proceeds by integrating the fast system (5.4) with k -step backward differentiation formula with multiple steps of size h :

$$\sum_{r=0}^k \alpha_r y_{m-r} + h\beta_0 f(t_m, y_m, z_m) = 0, \quad \alpha_0 = 1, \quad (5.6)$$

and the slow system (5.5) with step-size qh where q is a positive integer:

$$\sum_{r=0}^k \alpha_r z_{n-qr} + qh\beta_0 g(t_n, y_n, z_n) = 0, \quad \alpha_0 = 1. \quad (5.7)$$

Coupling between the subsystems is taken care of by the interpolation or extrapolation operator

$$\tilde{z}_m = \sum_{r=0}^k \tilde{\alpha}_{r, m-(n-q)} z_{n-rq} \quad \text{with } n - q < m \leq n. \quad (5.8)$$

The quantity \tilde{z}_m represents interpolation if $\tilde{\alpha}_{0,s} \neq 0, \tilde{\alpha}_{k,s} = 0$ and extrapolation if $\tilde{\alpha}_{0,s} = 0, \tilde{\alpha}_{k,s} \neq 0$.

One step of (5.7) and q steps of (5.6) comprises a compound step. The compound step may be computed using either a fastest or slowest first strategy. The *fastest first* multirate method integrates (5.4) with (5.6) from t_{n-q} to t_n using q steps of size h . The required z_m values are provided by \tilde{z}_m , computed by (5.8) using the values z_{n-kq}, \dots, z_{n-q} . Then, the slow system (5.5) is integrated from t_{n-q} to t_n using one step.

The *slowest first* strategy integrates (5.5) using (5.7) with the unknown y_n values approximated by \bar{y}_n computed by extrapolating the values of $y_{n-q-k+1}$ through to y_{n-q} . This is followed by integrating fast system (5.4) from t_{n-q} to t_n by (5.6). The slow solution is then computed from (5.8) using the z -values $z_{n-(k-1)q}, \dots, z_n$. We may then correct the value

of z_n by re-evaluating (5.7) based on y_n computed from (5.6) instead of the extrapolated value \bar{y}_n .

The fully implicit multirate step is obtained by iterating the solution of the slow and fast subsystems until convergence. This corresponds to using waveform relaxation to solve the system of ODEs. Waveform relaxation methods will be the topic of the next section.

Another approach is that of Engstler and Lubich [27], who construct a multirate Richardson Extrapolation method. The extrapolation tableau is used to decide which components are sufficiently accurate and consequently those components which require no further work. The extrapolation is continued for those components which do not satisfy the local error requirement. This provides a dynamic partitioning strategy. A new multirate Runge-Kutta formula, based on embedding low-order methods in the eighth-order method of Dormand and Prince is developed in [28].

For problems which have relatively few equations involving small time-constants Hofer [45] presents a technique which combines implicit and explicit methods. The implicit trapezoidal rule is used to integrate the transient, fast components, while the smooth components are integrated with the explicit modified midpoint rule. Rational extrapolation is implemented to control accuracy and step size. Andrus, [4], considers first order systems which have been partitioned into two subsystems. Subsystems are integrated somewhat independently of one another with different time steps and/or integration method. The slow response system is integrated with a fourth order Runge-Kutta method. Fourth order for the whole system is then achieved by integrating the fast response equation accurately enough so as to contribute terms no larger than fifth order to the slow system. This technique is extended to second order systems in [5]. The absolute stability of decoupled Runge-Kutta methods is considered in [6]. It is shown that if the two subsystems are weakly coupled then the regions of stability are similar to the classical regions of stability. Automatic step-size control based on local truncation error as well as separation error is discussed in [7].

5.2 Waveform Relaxation methods

In this section, we consider the solution of first order systems of ordinary differential equations of the form (5.1) using waveform relaxation methods. The essential idea of the waveform relaxation method is related to the Picard iteration [81] for proving the existence of solutions for initial value problems. The method became popular as a solution technique

for large, stiff systems of ODEs with the paper by Lelarasme et. al. in the VLSI literature [56].

The idea is to partition the ODE into p subsystems

$$\dot{y}_i = f_i(y_1, y_2, \dots, y_i, \dots, y_p), \quad i = 1, 2, \dots, p, \quad (5.9)$$

where

$$y_i \in \mathbb{R}^{n_i}, \quad \text{and} \quad \sum_{i=1}^p n_i = n.$$

Each subset of solution components, $y_i, i = 1, \dots, p$, is then solved in an iterative fashion. For example, a waveform Gauss–Jacobi (WRGJ) method is obtained by iterating

$$\dot{y}_i^{k+1} = f_i(y_1^k, y_2^k, \dots, y_i^{k+1}, \dots, y_p^k), \quad (5.10)$$

for $k = 0, 1, \dots$ and $i = 1, \dots, p$. During each iteration y_i^{k+1} is solved over a time window $[0, T]$ using the possibly interpolated values of the previous waveform for the evaluation of the function f_i . Analogous to the classical iterative techniques for linear systems of algebraic equations, a waveform Gauss–Seidel (WRGS) method is obtained by iterating

$$\dot{y}_i^{k+1} = f_i(y_1^{k+1}, y_2^{k+1}, \dots, y_i^{k+1}, y_{i+1}^k, \dots, y_p^k), \quad \text{for } k = 0, 1, \dots \quad (5.11)$$

Each iteration of either of these waveform methods requires the solution of p systems of first order differential equations. Convergence is achieved if $\|y^{k+1}(t) - y^k(t)\|$ satisfies a termination criteria for all $t \in [0, T]$.

It is clear that during each iteration of WRGJ, the p subsystems may be solved in parallel. Of course there is a rather complicated load balancing issue to deal with.

Convergence Results

For linear systems of ODEs of the form

$$y'(x) + Qy(x) = g(x), \quad (5.12)$$

where Q is a constant matrix, convergence has been studied by Miekkala and Nevanlinna in [64] and [65], and Nevanlinna in [73],[74] and [75].

A general waveform relaxation algorithm for (5.12) may be written as

$$\dot{y}^{k+1} + My^{k+1} = Ny^k + g, \quad y^{k+1}(t_0) = y_0, \quad (5.13)$$

where M and N denote a splitting of the matrix Q .

Using the convolution operator and Laplace transform of the differential system super-linear convergence of (5.13) is obtained on all finite time intervals. Using exponentially weighted norms linear convergence is obtained on unbounded time intervals.

For general nonlinear systems of ODEs of the form

$$c(y)y' = f(y), \quad y(0) = y_0, \quad t \in [0, T], \quad (5.14)$$

convergence results first appeared in [56],[101] and [100]. System (5.14) has a unique solution if $c(y)^{-1}$ exists and is uniformly bounded with respect to y , and f is globally Lipschitz continuous with respect to y .

If c is diagonally dominant then both WRGJ and WRGS both converge uniformly on bounded intervals [56]. This result applies to both Jacobi and Gauss-Seidel waveform techniques. In [17], Burrage extends these results to general waveform splittings and also provides error bounds. The result is summarized in the following theorem:

Theorem 5.2.1 *Consider the first order system of differential equations (5.1) defined on $[0, T]$ and assume there exists a splitting of $f(y)$, denoted $F(y, z)$, where F satisfies*

$$F(y, y) = f(y), \quad F : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^n,$$

and F is Lipschitz continuous with respect to y and z . Then the iteration

$$\dot{y}^{k+1} = F(y^{k+1}, y^k), \quad y^{k+1}(0) = y_0 \quad (5.15)$$

converges uniformly on all finite intervals $[0, T]$. The error, $\epsilon^k = y^k - y$ satisfies

$$\|\epsilon^k\|_T \leq \frac{(L_2 T)^k}{k!} e^{L_1 T} \|\epsilon^0\|_T, \quad (5.16)$$

where L_1 and L_2 are the Lipschitz constants of F with respect to y and z .

The convergence results presented in the previous section do not guarantee the convergence of discretized WR algorithms. Discretized WR algorithms result when the ordinary differential equations which specify the individual waveforms are solved using a numerical method. The interaction between WR algorithms and multistep integration methods is considered in detail in [100]. Convergence is obtained in the case of constant time steps for all subsystems and for the multirate WR algorithm provided the waveforms are interpolated carefully in the evaluation of f .

5.2.1 Partitioning

A suitable partitioning of components is crucial for the practical implementation of waveform relaxation methods. In [27] and [28] the authors use extrapolation methods to compute an error estimate. Components of the solution which satisfy the error requirement by a large margin are classified as inactive. The other components are deemed active and require more computational effort. In the context of waveform relaxation methods an alternate approach is presented in [102]. The authors suggest a multilevel partitioning strategy obtained by analyzing the elements of the Jacobian matrix.

Jacobian Techniques

Waveform relaxation methods have a successful history in the simulation of very large scale integration (VLSI) circuits and power systems. The first published report of the use of relaxation methods in a circuit simulator was the code MOTIS [20] in 1975. This code was up to two orders of magnitude faster than standard circuit simulators of the day.

Partitioning of the components in these applications have traditionally been tightly based on knowledge of the physical system. In [102], the authors describe a partitioning strategy which is independent of knowledge of the physics of the ODE system. When solving (5.1) with implicit time stepping, a system of nonlinear algebraic equations is solved at each iteration. Typically, some modified Newton scheme is used as the nonlinear solver requiring the formation of the Jacobian of the system of equations. Partitioning of the solution components into decoupled subsystems is achieved by performing a ϵ -decomposition ([87],[88], and [97]) of a modified Jacobian.

The ϵ -decomposition algorithm attempts to detect blocks of variables which are strongly coupled. The algorithm proceeds as follows. Given a row-scaled matrix A and an $\epsilon > 0$ if $|a_{ij}| \leq \epsilon$ then set $a_{ij} = 0$. Then, attempt to find a suitable permutation of rows and columns which will permute the matrix to block-diagonal form. The variables within each block are considered strongly coupled. After the deletion of small matrix entries this is just the classic graph theory problem of computing strongly connected components of a graph. This problem has been studied extensively and fast $O(n + e)$ algorithms are available where n is the number of vertices in the graph and e is the number of edges – see [96] and related references. The ϵ decomposition algorithm repeats this procedure for a sequence of ϵ values recording the maximum resulting block size $B_0(\epsilon)$. The value ϵ which gives the largest

decrease in $B_0(\epsilon)$ also yields the partitioning which is chosen for the waveform relaxation.

If backward Euler is used for the time integration then y_n is updated from y_{n-1} through the system of nonlinear equations

$$F(y_n) = y_n - y_{n-1} - h_n f(y_n) = 0.$$

To avoid scaling issues the ϵ -decomposition is not performed on the Jacobian of $F(y)$. Since some of the entries of the Jacobian are functions of the chosen time step small entries may arise even if variables are strongly related. The matrix

$$A = I + \frac{\partial f}{\partial y}$$

is used instead to avoid this scaling problem.

5.3 Schur Decomposition Methods and the Quasi-Steady State Approximation

In [58],[23] and [24], Deuffhard and Maas, develop a decoupled integration technique which involves splitting the solution components into fast and slow subsystems using the Schur decomposition of the Jacobian.

Consider a first order system of ODEs written as

$$\frac{d\phi}{dt} = F(\phi), \quad \phi(0) = \phi_0.$$

We now linearize the system of ODEs about the initial condition ϕ_0 . Letting $\psi = \phi - \phi_0$ then the differential equation implies

$$\frac{d}{dt}(\psi + \phi_0) = F(\psi + \phi_0).$$

Applying Taylor's Theorem to the right side we have

$$\frac{d\psi}{dt} + \frac{d\phi_0}{dt} = F(\phi_0) + F_\phi(\phi_0)\psi. \quad (5.17)$$

Since ϕ_0 is time-independent, $d\phi_0/dt = 0$ so the linearized system becomes

$$\frac{d\psi}{dt} = F(\phi_0) + F_\phi(\phi_0)\psi. \quad (5.18)$$

Assuming F_ϕ is nonsingular, we may solve (5.18) subject to $\psi(0) = 0$ to find

$$\psi(t) = F_\phi^{-1}(e^{F_\phi t} - I)F(\phi_0). \quad (5.19)$$

In [58] the authors suggest a change of variables $\Psi = Q_d\psi$ where

$$Q_d^{-1}\bar{T}Q_d = F_\phi(\phi_0)$$

is the block Schur decomposition [40] of the Jacobian. The matrix

$$\bar{T} = \begin{pmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{pmatrix}$$

is an upper triangular matrix with the eigenvalues of F_ϕ on the main diagonal and Q_d is an orthogonal matrix. In terms of Ψ the solution of the linearized problem (5.19) may be written as

$$\Psi(t) = Q_d F_\phi^{-1}(e^{F_\phi t} - I)F(\phi_0). \quad (5.20)$$

From the change of variables and ODE (5.18) we see that

$$\frac{d\Psi}{dt}(0) = Q_d \frac{d\psi}{dt}(0) = Q_d F(\phi_0).$$

Then using

$$e^{Q_d^{-1}\bar{T}Q_d t} = Q_d^{-1}e^{\bar{T}t}Q_d,$$

and equation (5.20) we see that $\Psi(t)$ can be written in terms of the rates $d\Psi/dt$ at $t = 0$ as

$$\Psi(t) = (e^{\bar{T}t} - I)\bar{T}^{-1} \frac{d\Psi}{dt}(0). \quad (5.21)$$

This representation suggests that small perturbations in the direction of eigenvectors corresponding to large eigenvalues will grow, while those corresponding to small eigenvalues will remain small.

A second decomposition, presented in [24], decouples the system of ODEs by eliminating the S_{12} block. The required similarity transformation is specified by the matrix T_d , where

$$T_d^{-1}F_\phi T_d = S = \begin{pmatrix} S_{11} & 0 \\ 0 & S_{22} \end{pmatrix}, \quad (5.22)$$

where

$$T_d^{-1} = \left(I - \begin{pmatrix} 0 & Z \\ 0 & 0 \end{pmatrix} \right) Q_d^T \quad \text{and} \quad T_d = Q_d \left(I + \begin{pmatrix} 0 & Z \\ 0 & 0 \end{pmatrix} \right),$$

and Z solves the Sylvester equation

$$S_{11}Z - ZS_{22} = -S_{12}.$$

Equation (5.22) suggests grouping the eigenvalues of F_ϕ according to their real parts and defining a splitting parameter μ as

$$\mu = \max_{\lambda \in S_{22}} \Re \lambda < 0 \quad \text{and} \quad \min_{\lambda \in S_{11}} \Re \lambda > \mu.$$

The change of variables

$$\begin{pmatrix} w \\ z \end{pmatrix} = T_d^{-1}y,$$

partitions (5.1) as

$$\begin{aligned} \dot{w} &= f(w, z) \\ \epsilon \dot{z} &= g(w, z) \\ (w, z)(0) &= T_d^{-1}y_0, \end{aligned} \tag{5.23}$$

where the perturbation parameter ϵ is given as $\epsilon = 1/\mu$.

The quasi-steady-state approximation (QSSA) uses (5.23) to produce the differential-algebraic equations (DAEs)

$$\begin{aligned} \dot{w}_0 &= f(w_0, z_0) \\ 0 &= g(w_0, z_0). \end{aligned} \tag{5.24}$$

The assumption here is the fast components z reach steady state quickly, which suggests replacing the differential equation for z with algebraic constraints. With this choice of ϵ , (5.24) is an index 1 DAE and hence may be solved using standard numerical techniques [9]. A linearly implicit Euler discretization is used to solve a chemical combustion problem with this technique in [23].

This procedure is affected by two sources of error. The local truncation error, controllable by usual local error estimates, and the QSSA error. How to estimate and control the QSSA error is discussed in the next section.

5.3.1 Estimating the QSSA Error

Sophisticated ODE packages provide local error control mechanisms which select time steps to keep the local truncation error below some user specified tolerance. Using a decoupled integration strategy imposes another source of error due to the decoupling itself.

The multirate extrapolation method of Engstler and Lubich [28] also gives local error estimates from the extrapolation process itself. These estimates are used to dynamically partition the system of the ordinary differential equations.

We now consider an error estimation technique for the quasi-steady-state approximation. The QSSA involves reducing the dimension of a system of ODEs to a system of DAEs with varying numbers of algebraic constraints, cf. (5.24). Numerical integrators for ODEs and DAEs control local discretization or truncation error. The QSSA error may be estimated and controlled in a similar way. This will allow a monitor on the chosen partitioning and suggest when subsequent Schur decompositions are necessary. This result is outlined in [23]; however, the development is quite involved and we include it here for completeness and future reference.

If (w, z) and (w_0, z_0) denote the solution of the ODE and DAE system respectively, then the QSSA error after one integration step Δt may be written as

$$\alpha = \|(w, z)(\Delta t) - (w_0, z_0)(\Delta t)\|. \quad (5.25)$$

A brief outline of the derivation of an estimate for α is given [23]. The details of this derivation (which is rather involved) are provided below.

If the reduced problem (5.24) has a unique solution (or if the DAE has index one) then the singularly perturbed problem (5.23) may be solved asymptotically [76] as

$$w(t) = W(t, \epsilon) + \epsilon m(\tau, \epsilon) \quad (5.26)$$

$$z(t) = Y(t, \epsilon) + n(\tau, \epsilon), \quad (5.27)$$

where

$$\begin{aligned} W(t, \epsilon) &\sim \sum_{j=0}^{\infty} w_j(t) \epsilon^j, & Z(t, \epsilon) &\sim \sum_{j=0}^{\infty} z_j(t) \epsilon^j \\ m(\tau, \epsilon) &\sim \sum_{j=0}^{\infty} m_j(\tau) \epsilon^j, & n(\tau, \epsilon) &\sim \sum_{j=0}^{\infty} n_j(\tau) \epsilon^j. \end{aligned}$$

The functions $m_j(\tau)$ and $n_j(\tau)$, with $\tau = t/\epsilon$, are exponentially decaying in the sense that

$$|n_j(\tau)| \leq |n_j(0)| e^{-\kappa\tau} \quad \text{and} \quad |m_j(\tau)| \leq |m_j(0)| e^{-\kappa\tau}.$$

If we assume that the fast components (z) are controlled by the system dynamics, then the QSSA error is dominated by the error in the slow components (w), that is, $\alpha \approx \|w(\Delta t) -$

$w_0(\Delta t)\|$. Ignoring $O(\epsilon^2)$ terms from (5.26) we see that α may be approximated as

$$\alpha \approx \epsilon \|w_1(\Delta t)\|.$$

To obtain a bound on $\|w_1(\Delta t)\|$ we substitute the expansions for $w(t)$ and $z(t)$ from (5.26) and (5.27) into the singularly perturbed system of ODEs (5.23). Applying Taylor expansions to the functions f and g and comparing coefficients of ϵ we obtain the $O(1)$ reduced problem (5.24) and the $O(\epsilon)$ problem

$$\dot{w}_1 = f_w(w_0, z_0)w_1 + f_z(w_0, z_0)z_1 \quad (5.28)$$

$$0 = g_w(w_0, z_0)w_1 + g_z(w_0, z_0)z_1 - \dot{z}_0, \quad (5.29)$$

respectively. Note, we have suppressed the t dependence. Differentiating the second equation from the reduced problem (5.24) with respect to time we obtain

$$0 = g_w(w_0, z_0)\dot{w}_0 + g_z(w_0, z_0)\dot{z}_0,$$

or

$$\dot{z}_0 = -\frac{g_w(w_0, z_0)f(w_0, z_0)}{g_z(w_0, z_0)}.$$

Using this value of \dot{z}_0 and substituting z_1 from (5.29) into (5.28) we obtain a linear ODE for w_1 ,

$$\dot{w}_1 = (f_w - f_z g_y^{-1} g_w)w_1 - \frac{f_z g_w f}{g_z^2}. \quad (5.30)$$

Evaluating (5.26) at $t = 0$ and comparing $O(\epsilon)$ terms we see that $w_1(0) + m_0(0) = 0$ or $w_1(0) = -m_0(0)$ (assuming $w(0)$ is ϵ -independent). To find $w_1(0)$ we obtain a solution for $m_0(\tau)$. Using (5.26) we find that to leading order

$$\frac{dw}{dt} = \frac{dW}{dt} + \epsilon \frac{dm}{dt} = \frac{dW}{dt} + \epsilon \frac{dm}{d\tau} \frac{d\tau}{dt} = \frac{dW}{dt} + \frac{dm}{d\tau}.$$

Using the differential equations for $w(t)$ and $W(t)$ we find

$$\frac{dm}{d\tau} = \frac{dw}{dt} - \frac{dW}{dt} = f(w, z) - f(W, Z).$$

Setting $\epsilon = 0$ we have

$$\frac{dm_0}{d\tau} = f(w_0(0), z_0(0) + n_0(\tau)) - f(w_0(0), z_0(0)).$$

Since $m_0 \rightarrow 0$ as $\tau \rightarrow \infty$, we may write $m_0(\tau)$ as

$$m_0(\tau) = \int_{\infty}^{\tau} f(w_0(0), z_0(0) + n_0(s)) - f(w_0(0), z_0(0)) ds. \quad (5.31)$$

Hence, to bound the QSSA error we have to solve (5.30) subject to

$$w_1(0) = \int_0^{\infty} f(w_0(0), z_0(0) + n_0(s)) - f(w_0(0), z_0(0)) ds. \quad (5.32)$$

Solving the first order differential equation for $w_1(t)$ we may bound $w_1(\Delta t)$ as

$$\|w_1(\Delta t)\| \leq e^{L\Delta t} \|w_1(0) + b\|,$$

for some constants L and b . At $t = 0$ we expect, due to our partitioning, that $f_z(w(0), z(0)) = g_w(w(0), z(0)) \approx 0$. This suggests $w_1(0)$ will dominate b in our bound on $w_1(\Delta t)$. Further, we expect the step-size control strategy of the numerical integrator, will choose Δt so that $e^{L\Delta t} \approx 1$. Therefore, we may obtain an approximate bound on $w_1(\Delta t)$ as $\|w_1(\Delta t)\| \leq \|w_1(0)\|$.

To approximate $w_1(0)$ we use the fundamental theorem of calculus, Gauss-Laguerre integration, and another application of the fundamental theorem of calculus. If we let $g(t) = f(w_0(0), z_0(0) + t)$ then

$$g(n_0(s)) - g(0) = \int_0^{n_0(s)} g'(t) dt.$$

Changing from t to θ variables where $t = \theta n_0(s)$, we find

$$g(n_0(s)) - g(0) = \int_0^1 g'(\theta n_0(s)) n_0(s) d\theta = \int_0^1 f_z(w_0(0), z_0(0) + \theta n_0(s)) n_0(s) d\theta.$$

So, from (5.32) and $n_0(s) \sim n_0(0)e^{-\kappa s}$ we have

$$w_1(0) = \int_0^{\infty} \int_0^1 f_z(w_0(0), z_0(0) + \theta n_0(s)) n_0(s) d\theta ds.$$

Interchanging the order of integration and changing variables $s/\kappa \rightarrow s$ we may rewrite $w_1(0)$ as

$$w_1(0) = \frac{1}{\kappa} \int_0^1 \int_0^{\infty} f_z(w_0(0), z_0(0) + \theta n_0(s/\kappa)) n_0(0) e^{-s} ds d\theta.$$

We now use Gauss-Laguerre quadrature [44] to approximate the integral of $f_z n_0(0) e^{-s}$ on the interval $s \in [0, \infty)$. The Gauss-Laguerre quadrature formula is given as

$$\int_0^{\infty} e^{-x} f(x) dx = \sum_{k=1}^m H_k f(x_k) + E, \quad (5.33)$$

where

$$H_i = \frac{(m!)^2}{L'_m(x_i)L_{m+1}(x_i)},$$

L_m is the m -th Laguerre polynomial and x_i is the x_i -th root of L_m . The error, E , is given as

$$E = \frac{(m!)^2}{(2m)!} f^{(2m)}(\xi), \quad \xi \in [0, \infty).$$

For $m = 1$ we have the approximation

$$\int_0^\infty e^{-x} f(x) dx \approx f(1).$$

Here we have used the first two Laguerre polynomials given by $L_1(x) = 1 - x$ and $L_2(x) = 2 - 4x + x^2$.

Applying Gauss-Laguerre quadrature to the integral on the interval $s \in [0, \infty)$ yields

$$\int_0^\infty f_z(w_0(0), z_0(0) + \theta n_0(s/\kappa)) n_0(0) e^{-s} ds = f_z(w_0(0), z_0(0) + n_0(1/\kappa)) n_0(0) + \frac{1}{2} \left[\frac{d^2}{ds^2} f_z(w_0(0), z_0(0) + \theta n_0(s/\kappa)) n_0(0) \right]_{s=\sigma},$$

where $\sigma \in [0, \infty)$. This implies

$$w_1(0) \approx \frac{1}{\kappa} \int_0^1 f_z(w_0(0), z_0(0) + n_0(1/\kappa)) n_0(0) d\theta.$$

We now use the fact that $n_0(0) = n_0(1/\kappa)e^{-1}$ to write $w_1(0)$ in a form for which the fundamental theorem will apply:

$$w_1(0) \approx \frac{e}{\kappa} \int_0^1 f_z(w_0(0), z_0(0) + n_0(1/\kappa)) n_0(1/\kappa) d\theta.$$

Utilizing the fundamental theorem of calculus one more time and $n_0(1/\kappa) = n_0(0)e^{-1}$ we obtain

$$w_1(0) \approx \frac{e}{\kappa} [f(w_0(0), z_0(0) + n_0(0)e^{-1}) - f(w_0(0), z_0(0))].$$

The final result follows by applying linear approximations to the quantity on the right,

$$\begin{aligned} w_1(0) &\approx \frac{1}{\kappa} [f(w_0(0), z_0(0) + n_0(0)) - f(w_0(0), z_0(0))] \\ &\approx \frac{1}{\kappa} [f(w(0), z(0)) - f(w_0(0), z_0(0))]. \end{aligned} \tag{5.34}$$

Finally, using $w_0(0) = w(0)$ we may approximate the QSSA error as

$$\alpha \approx \frac{\epsilon}{\kappa} [f(w(0), z(0)) - f(w_0(0), z_0(0))]. \tag{5.35}$$

This is an intriguing result which expresses the error simply in terms of a difference of the slow system function values.

5.4 PDE Based Methods

5.4.1 Space–Time Adaptive hp –Refinement Methods

Moving Mesh methods described in Chapter 4 allow spatial mesh points to continuously adapt to regions of rapid change in the solution of the physical PDE. Another related approach is so-called h –refinement methods which allow mesh refinement or coarsening. For time dependent PDEs, p –refinement methods are natural to choose temporal order and step size control.

Flaherty and Moore [69], [70] and [30] develop integrated space–time hp –refinement strategies. These methods make mesh and time step/order decisions in a unified manner.

Systems of parabolic PDEs are solved using a Finite Element Galerkin method (in space) and a Singly Implicit Runge–Kutta method (in time). A posteriori temporal and spatial error estimates are used to guide in accepting or rejecting solutions over local space–time domains. Moreover, these estimates provide grid, time step and temporal order for the next step.

These methods are computationally robust; however, the resulting space–time meshes are complicated and require sophisticated data structures for efficient implementation [29].

5.4.2 Schwarz Waveform Relaxation

There have been three general classes of methods discussed in the literature which apply domain decomposition to parabolic problems:

1. Discretize in time and solve the resulting elliptic problems with classical domain decomposition ([18], [61], and [55]).
2. Discretize in space and apply waveform relaxation to the system of ODEs ([57],[53] and [52]).
3. Subdividing the spatial domain, discretizing spatially and apply waveform relaxation across the subdomains ([32] and [38]).

The first approach, which applied domain decomposition to parabolic problem was probably the most natural: utilize the extensive literature concerning domain decomposition applied to elliptic problems. This technique suffers from a couple of difficulties. Since the elliptic problems arise after discretizing in time, we are forced to use the same time step

on each subdomain. Also computed solution information must be exchanged at the end of each time step.

Applying waveform relaxation to the system of ODEs resulting from the semi-discretization in space does allow different time steps in different regions of the spatial domains. Information is transmitted between subdomains after solving a subset of ODEs over a time window, not after each step. Unfortunately, for Jacobi, Gauss-Seidel and SOR Waveform relaxation methods the constants which arise in the error estimates depend negatively on the mesh parameter Δx , if the ODEs result from a spatial discretization of a PDE. The negative dependence on spatial resolution has been overcome for the classical WR methods using multigrid by Lubich and Ostermann [57], [95], and Janssen and Vandewalle [52].

The third class of methods overcome the difficulties associated with waveform relaxation methods. The later technique has become known as Schwarz waveform relaxation methods. These methods were developed by Gander et al. [31],[36], [32],[34] and independently by Giladi in [38]. These methods allow different numerical treatments (time step and integration formula) on different subdomains and convergence independent of the mesh parameter without the added complication of the multigrid framework.

Consider a general parabolic problem

$$u_t = \mathcal{L}(u, x, t) \quad \text{in } \Omega \quad (5.36)$$

subject to appropriate initial and boundary conditions on $\partial\Omega$. The Schwarz waveform relaxation method can be written quite succinctly in two subdomains Ω_0 and Ω_1 as follows: for $j = 0, 1$,

$$\frac{\partial u_j^{(k+1)}}{\partial t} = \mathcal{L}(u_j^{(k+1)}, x, t) \quad x, t \in \Omega_j \quad (5.37)$$

$$u_j^{(k+1)}(x, t) = \begin{cases} u_{1-j}^{(k)}(x, t) & x, t \in \Gamma_j = \partial\Omega_j \cap \Omega_{1-j} \\ \text{given boundary condition} & x, t \in \partial\Omega_j - \Gamma_j \end{cases} \quad (5.38)$$

We make things explicit by considering the Schwarz waveform relaxation method applied to the one-dimensional inhomogeneous heat equation [36]:

$$\begin{aligned} u_t &= u_{xx} + f(x, t) & 0 < x < L, t > 0 \\ u(0, t) &= g_1(t) & t > 0 \\ u(L, t) &= g_2(t) & t > 0 \\ u(x, 0) &= u_0(x) & 0 < x < L. \end{aligned}$$

We assume enough smoothness on the prescribed data to ensure a unique, bounded solution. The spatial domain $\Omega = [0, L] \times [0, \infty)$ is decomposed as $\Omega_1 = [0, \beta L] \times [0, \infty)$ and $\Omega_2 = [\alpha L, L] \times [0, \infty)$, where $0 < \alpha < \beta < L$. We define $v(x, t)$ on Ω_1 and $w(x, t)$ on Ω_2 so that

$$\begin{aligned} v_t &= v_{xx} + f(x, t) & 0 < x < \beta L, t > 0 \\ v(0, t) &= g_1(t) & t > 0 \\ v(\beta L, t) &= w(\beta L, t) & t > 0 \\ v(x, 0) &= u_0(x) & 0 \leq x \leq \beta L, \end{aligned}$$

and

$$\begin{aligned} w_t &= w_{xx} + f(x, t) & \alpha L < x < L, t > 0 \\ w(\alpha L, t) &= v(\alpha L, t) & t > 0 \\ w(L, t) &= g_2(t) & t > 0 \\ w(x, 0) &= u_0(x) & \alpha L \leq x \leq L. \end{aligned}$$

The waveform relaxation iteration proceeds as, for $k = 0, 1, \dots$

$$\begin{aligned} v_t^{(k+1)} &= v_{xx}^{(k+1)} + f(x, t) & 0 < x < \beta L, t > 0 \\ v^{(k+1)}(0, t) &= g_1(t) & t > 0 \\ v^{(k+1)}(\beta L, t) &= w^{(k)}(\beta L, t) & t > 0 \\ v^{(k+1)}(x, 0) &= u_0(x) & 0 \leq x \leq \beta L, \end{aligned} \tag{5.39}$$

and

$$\begin{aligned} w_t^{(k+1)} &= w_{xx}^{(k+1)} + f(x, t) & \alpha L < x < L, t > 0 \\ w^{(k+1)}(\alpha L, t) &= v^{(k)}(\alpha L, t) & t > 0 \\ w^{(k+1)}(L, t) &= g_2(t) & t > 0 \\ w^{(k+1)}(x, 0) &= u_0(x) & \alpha L \leq x \leq L. \end{aligned} \tag{5.40}$$

In [36] superlinear convergence is obtained on bounded time intervals, and linear convergence on unbounded time intervals for the one-dimensional heat equation. Convergence is independent of the mesh parameter (and hence robust with respect to mesh refinement) and the convergence rate improves by increasing the size of the overlap. Results are given for

both the continuous case, equations (5.39) and (5.40), and the semi-discrete case (after spatial discretization). Giladi and Keller [38] prove superlinear convergence on bounded time intervals for a constant coefficient convection diffusion equation. Gander ([31] and [32]) extends these results to the one-dimensional, variable coefficient reaction diffusion equation, and again obtains linear convergence and superlinear convergence on unbounded and bounded time intervals respectively. Gander et al. [34] apply overlapping Schwarz waveform methods to the wave equation and a constant coefficient, linear convection reaction diffusion equation. The effect of the Dirichlet transmission conditions at the subdomain boundaries is studied and found to slow down convergence of the algorithms. Optimal transmission conditions are then derived which lead to non-overlapping Schwarz methods which converge in a finite number of steps. Multi-dimensional extensions of these results may be found in [37] for the heat equation and [33] for convection reaction diffusion problems.

An important question remains: how do we dynamically determine the number and placement of subdomains? The partitioning ideas discussed in a previous section may provide an answer, especially in a dynamic situation where difficult computational regions are changing with time. This is still very much an open research problem.

In the next chapter we propose a new Moving Mesh Schwarz Waveform method which inherits the favourable properties of Schwarz Waveform and the spatial mesh resolution abilities of Moving Mesh PDEs.

Chapter 6

Schwarz Waveform Moving Mesh Method

In this section we propose a new Schwarz Waveform Moving Mesh Method. The basic idea is to apply Schwarz Waveform to the system of ODEs which arise from semi-discretization (in space) of the coupled physical and moving mesh partial differential equation.

6.1 Continuous Algorithm

We propose an overlapping Schwarz Waveform Moving Mesh method to solve parabolic PDEs in one-dimension,

$$u_t = \mathcal{L}(u), \quad \text{on } \Omega = [0, 1],$$

subject to $u(x, 0) = u_0(x)$ and Dirichlet boundary conditions.

We first decompose Ω into D nonoverlapping fixed subdomains $\tilde{\Omega}_j$, $j = 1, \dots, D$. Each subdomain $\tilde{\Omega}_j$ is enlarged by an overlap region consisting of M mesh points, giving overlapping domains $\Omega_1, \Omega_2, \dots, \Omega_D$.

The physical PDE is now discretized along with a moving mesh equation on each subdomain. To avoid mesh crossings from one subdomain to another we fix the mesh points on the boundary of $\tilde{\Omega}_j$, $j = 1, \dots, D$, but allow the mesh points to move within and on the outer boundary of the overlap regions.

Let x_j and ξ_j denote the physical and computational meshes on each subdomain Ω_j . The solution on each subdomain will be denoted as u_j . The Schwarz Waveform Moving

Mesh method requires us to solve, for $k = 1, 2, \dots$, and $j = 1, \dots, D$,

$$\begin{aligned} \dot{u}_j^k - \frac{\partial u_j^k}{\partial x} \dot{x}_j^k &= \mathcal{L}(u_j^k) \\ \left(M(x_j^k, u_j^k, t) \frac{\partial \dot{x}_j^k}{\partial \xi} \right)_\xi &= -\frac{1}{\tau} \left(M(x_j^k, u_j^k, t) \frac{\partial x_j^k}{\partial \xi} \right)_\xi, \end{aligned} \tag{6.1}$$

for $x \in \Omega_j$.

The boundary values for u_j^k and x_j^k come from the values of $u_{j-1}^{k-1}, x_{j-1}^{k-1}$ and $u_{j+1}^{k-1}, x_{j+1}^{k-1}$ on the left and right boundaries of Ω_j respectively, from the previous iteration. Each Schwarz waveform iteration requires the solution of D moving boundary problems as illustrated in Figure 6.1 with $D = 3$.

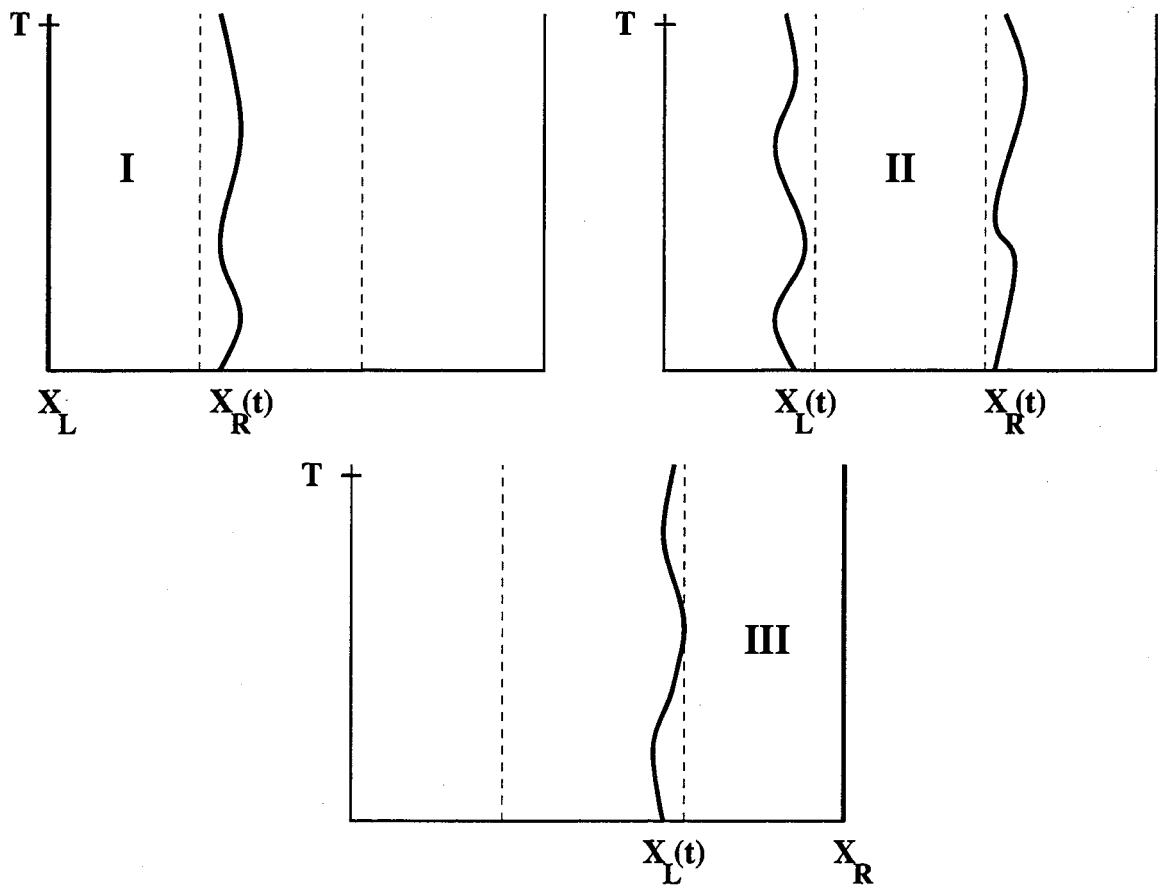


Figure 6.1: Sequence of Moving Boundary Problems solved during one iteration of the Moving Mesh Schwarz Waveform method over a time window $[0, T]$.

6.2 Solving the Moving Boundary Problems

On each subdomain the coupled system of PDEs is discretized in space as described in section 4.2. The Moving Schwarz method described in the previous section requires that the boundary points of $\tilde{\Omega}_j$ be fixed to avoid mesh crossings between subdomains. To fix the boundary points we adjust the moving mesh PDE corresponding to such mesh points to read $\dot{x}_B = 0$ for each boundary index B of $\tilde{\Omega}_j$.

The time dependent ODEs are solved using backward Euler over successive time windows. The time dependent boundary conditions on u_j and x_j are obtained by cubic spline interpolation from the data of the previous iteration. Interpolation is required since generally the data from iteration $k - 1$ is not computed on the same sequence of time steps as iteration k .

For efficiency, the algorithm is designed with an adaptive time windowing strategy. The algorithm begins by solving the sequence of moving boundary problems on a time window $[0, T]$. If u_j or x_j fails to reach $t = T$ in the maximum allowable time steps, the iteration is restarted with a smaller time window. The window is also reduced if the Schwarz Waveform iteration fails to converge in 6 iterations. The time window is enlarged if the iteration converges in less than 4 iterations. The waveform iteration is terminated when $\|y_j^k - y_j^{k-1}\|_\infty \leq \delta$, where y is the vector of unknowns containing both the mesh and solution. A tolerance of $\delta = 1e - 4$ is used in the experiments with local error control tolerance of $1e - 6$.

As a final note, the code provides each subdomain with an initial equidistributed mesh. That is we solve $u_t = u_0(x)$ for $x \in \tilde{\Omega}_j$ and a moving mesh PDE subject to $u(x, 0) = 0$. The provides a grid which equidistributes the initial solution to the PDE over each subdomain.

Chapter 7

Numerical Results

To understand the working details of the Waveform and Schwarz Waveform methods when applied to partial differential equations, we begin our experiments on fixed meshes. This allows us to experiment with the effect of spatial mesh size, overlap and the maximum number of allowable time steps without the complication of moving mesh methods. We will conclude this chapter by applying the Schwarz Waveform moving mesh method to Burgers' equation and a two spike problem.

7.1 Model Problems

As an initial test problem we consider the viscous Burgers' equation,

$$\begin{aligned}u_t &= \epsilon u_{xx} - \frac{1}{2}(u^2)_x \\u(0, t) &= 1, \quad u(1, t) = 0, \\u(x, 0) &= c - \frac{1}{2} \tanh((x - x_0)/4\epsilon),\end{aligned}$$

where $c = 1/2$, $x_0 = 1/4$ or $1/10$, and $\epsilon \ll 1$. The solution is a traveling front of thickness $O(\epsilon)$ which moves to the right from x_0 at speed $c = 0.5$.

We will also use our Moving Schwarz method to solve Burgers' equation subject to the initial condition

$$u(x, 0) = \sin(2\pi x) + \frac{1}{2} \sin(\pi x)$$

with boundary conditions $u(0, t) = u(1, t) = 0$. This results in a solution which develops a sharp front and moves to the right with diminishing amplitude.

As a second test we will consider a two spike problem whose solution is given by

$$u(x, t) = \frac{1}{\epsilon} (\max(0, \tanh(t - t_0)) e^{-(x-x_0)^2/\epsilon} + \max(0, \tanh(t - t_1)) e^{-(x-x_1)^2/\epsilon}).$$

The solution consists of two spikes centered at fixed locations x_0 and x_1 which begin to evolve at different times t_0 and t_1 . We choose x_0 and x_1 to keep the spikes well isolated in the spatial domain. The times t_0 and t_1 are chosen so that $t_1 > t_0$. This allows the first spike to grow to its maximum height before the second spike appears.

7.2 Waveform Relaxation

We will test Jacobi and Gauss-Seidel Waveform Relaxation for fixed grids, applied to Burgers' equation with a moderate value of ϵ . To implement the Waveform Relaxation we divide the solution components into three fixed subsystems. This will allow comparisons with Schwarz Waveform on fixed grids. We note, however, that this may not be the optimal partitioning of components for this problem.

7.2.1 Effect of Overlap and Maximum Number of Times Steps

In Tables 7.1 and 7.2 we record the number of waveform iterations and total CPU time (in seconds) required to solve Burgers' equation with $\epsilon = 1e - 2$ for the Jacobi Waveform iteration method with 1000 and 6000 maximum time steps per iteration. The CPU time and number of iterations are recorded against the number of mesh points and the size of the overlap. Integration is performed with an explicit method. Tables 7.3 and 7.4 repeats these experiments for the Gauss-Seidel Waveform method.

It is clear that even a small amount of overlap produces a tremendous reduction in computational time for both Jacobi and Gauss-Seidel Waveform methods. The relatively improved convergence properties of Gauss-Seidel as compared to Jacobi is evident for small amounts of overlap. Yet moderate overlaps bridge this gap and in fact allow Jacobi to outperform Gauss-Seidel. We also note the inability of larger overlaps to further reduce CPU times. This is due to the increase of work per iteration as the overlap increases, eventually negating any improvements to the convergence rate. The optimal amount of overlap appears to depend on the number of mesh points used. It appears as if a 15-25 percent overlap is optimal for this problem.

Although increasing the maximum number of allowable time steps from 1000 to 6000 per time window has an appreciable effect on the number of waveform iterations, it has a negative effect on the overall CPU time for moderate values of N . As the number of mesh points is increased, increasing the maximum number of time steps is beneficial. This is due to the smaller time requirement imposed by the local error control of the ODE solver for smaller mesh spacings.

		Overlap										
		0	2	4	6	8	16	32	48	64	128	160
257		336	121	113	108	106	111	90	102	111		
		139	59	46	42	41	36	31	34	38		
385		599	190	162	155	160	139	136	138	156		
		323	143	108	93	92	71	70	66	72		
513		777	257	211	188	189	173	183	165	183	219	
		663	284	203	169	165	140	124	115	118	144	
641		912	313	255	236	226	223	216	219	210	261	273
		1102	466	337	285	266	234	195	195	183	206	212
769		1344	387	316	284	274	264	251	267	246	180	177
		1419	739	510	425	383	336	271	282	267	191	176

Table 7.1: Number of Waveform iterations and CPU time (seconds) for Jacobi Waveform with MAXSTEPS = 1000

		Overlap										
		0	2	4	6	8	16	32	48	64	128	160
257		317	55	37	33	30	25	23	21	24		
		188	88	61	50	43	32	26	21	26		
385		452	80	54	45	42	35	32	33	31		
		457	235	164	121	114	83	66	56	50		
513		774	112	63	53	46	45	47	36	39	39	
		754	440	339	265	218	150	126	97	96	91	
641		909	132	88	68	69	55	57	50	42	48	54
		1282	824	569	443	368	254	209	170	147	130	141
769		1339	201	105	83	75	65	65	60	55	33	39
		1723	1223	906	662	580	378	309	277	235	125	123

Table 7.2: Number of Waveform iterations and CPU time (seconds) for Jacobi Waveform with MAXSTEPS = 6000

		Overlap										
		0	2	4	6	8	16	32	48	64	128	160
257		201	95	108	107	99	108	90	102	111		
		<i>119</i>	<i>51</i>	<i>51</i>	<i>49</i>	<i>46</i>	<i>41</i>	<i>37</i>	<i>39</i>	<i>43</i>		
385		302	152	144	152	154	135	135	138	156		
		<i>274</i>	<i>127</i>	<i>109</i>	<i>111</i>	<i>108</i>	<i>83</i>	<i>80</i>	<i>79</i>	<i>81</i>		
513		415	202	182	184	188	161	183	165	183	219	
		<i>534</i>	<i>248</i>	<i>200</i>	<i>200</i>	<i>202</i>	<i>155</i>	<i>151</i>	<i>139</i>	<i>143</i>	<i>152</i>	
641		560	245	209	220	224	206	216	219	210	261	273
		<i>795</i>	<i>395</i>	<i>315</i>	<i>306</i>	<i>312</i>	<i>252</i>	<i>233</i>	<i>224</i>	<i>216</i>	<i>217</i>	<i>228</i>
769		645	306	259	260	268	246	251	267	246	318	315
		<i>1247</i>	<i>605</i>	<i>465</i>	<i>448</i>	<i>446</i>	<i>376</i>	<i>336</i>	<i>332</i>	<i>309</i>	<i>315</i>	<i>310</i>

Table 7.3: Number of Waveform iterations and CPU time (seconds) for Gauss–Seidel Waveform with MAXSTEPS = 1000

		Overlap										
		0	2	4	6	8	16	32	48	64	128	160
257		92	36	30	26	26	24	21	21	24		
		<i>189</i>	<i>66</i>	<i>48</i>	<i>40</i>	<i>40</i>	<i>36</i>	<i>24</i>	<i>22</i>	<i>28</i>		
385		183	52	40	39	35	33	28	30	30		
		<i>451</i>	<i>169</i>	<i>122</i>	<i>115</i>	<i>99</i>	<i>88</i>	<i>60</i>	<i>56</i>	<i>51</i>		
513		312	59	46	40	39	42	43	33	36	39	
		<i>876</i>	<i>347</i>	<i>244</i>	<i>203</i>	<i>195</i>	<i>163</i>	<i>132</i>	<i>95</i>	<i>96</i>	<i>96</i>	
641		387	84	59	52	57	50	52	45	39	48	54
		<i>1360</i>	<i>624</i>	<i>436</i>	<i>348</i>	<i>319</i>	<i>258</i>	<i>221</i>	<i>166</i>	<i>149</i>	<i>139</i>	<i>150</i>
769		434	102	75	65	57	58	63	52	51	57	60
		<i>2233</i>	<i>985</i>	<i>657</i>	<i>537</i>	<i>461</i>	<i>387</i>	<i>375</i>	<i>263</i>	<i>237</i>	<i>207</i>	<i>207</i>

Table 7.4: Number of Waveform iterations and CPU time (seconds) for Gauss–Seidel Waveform with MAXSTEPS = 6000

7.2.2 Rate of Convergence

The tables of the previous section also verify a remark made earlier. The convergence of Waveform Relaxation depends negatively on the size of the mesh spacing. If N is doubled then the number of iterations increase by a factor of approximately two.

Figure 7.1 shows the convergence of Gauss–Seidel Waveform Relaxation over the time window $[0, 0.1]$. The left plot shows the reduction in the difference of successive iterates for an overlap of 8 for all values of N . Linear convergence is achieved; however, it is clear the constant depends on Δx in a negative way. Yet, if we tune the overlap to the best value for a particular N (from the tables) then we obtain convergence (to $1e-12$) in 4 or 5 iterations for all values of N .

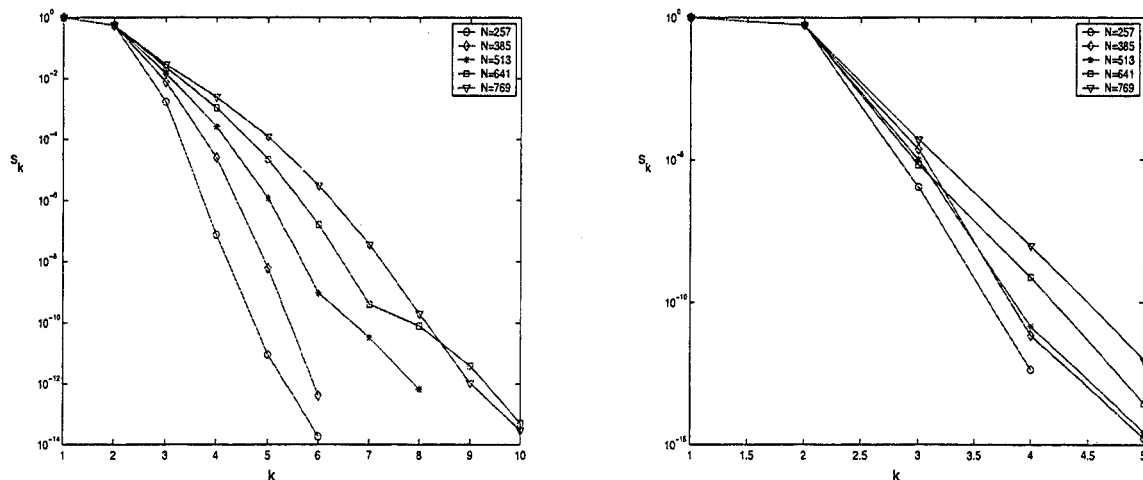


Figure 7.1: Convergence of waveform relaxation for various spatial mesh sizes and overlap= 8 (left) and tuned overlap (left).

7.3 Schwarz Waveform on Fixed Grids

To compare with the results of the previous section we solve Burgers' equation with $\epsilon = 1e-2$ with the Schwarz Waveform method of section 5.4.2. Tables 7.5 and 7.6 record the number of iterations and CPU time required to solve the problem with given values of N and overlap.

An important contrast with the results of the Waveform Relaxation runs is evident. Moderate overlap yields convergence independent of Δx . As N is increased the total CPU

time does increase. This is due to the extra work per iteration by the ODE solver, not the number of Schwarz Waveform iterations. Increasing the maximum number of allowed time steps marginally improves the run times and the number of iterations for larger values of N and overlap.

		Overlap						
		8	16	32	48	64	128	160
257		24	22	18	18	18		
		13	14	13	14	15		
385		28	24	19	18	18		
		25	22	19	20	21		
513		29	24	22	18	18	18	
		37	30	30	26	28	30	
641		33	27	23	21	18	18	18
		56	47	42	40	35	36	38
769		34	28	24	22	19	18	18
		76	65	54	51	46	42	44

Table 7.5: Number of Waveform iterations and CPU time (seconds) for Schwarz Waveform with MAXSTEPS = 1000

		Overlap						
		8	16	32	48	64	128	160
257		13	12	12	10	10		
		14	12	11	13	13		
385		16	12	12	10	10		
		24	19	20	17	18		
N 513		18	13	12	11	10	10	
		39	30	27	26	23	26	
641		20	16	12	12	10	10	10
		63	47	36	37	29	31	33
769		21	16	12	12	12	10	10
		91	63	47	48	48	37	38

Table 7.6: Number of Waveform iterations and CPU time (seconds) for Schwarz Waveform with MAXSTEPS = 6000

7.4 Schwarz Waveform and Moving Meshes

We now provide some brief experiments to demonstrate the Schwarz Waveform Moving Mesh method.

7.4.1 Effect of a Fixed Mesh Point

In the design of the Schwarz Waveform Moving Mesh method we fixed the mesh points at the boundary of the non-overlapping domains $\tilde{\Omega}_j$. This was done to avoid mesh crossings from one subdomain to another. It has the effect of keeping the number of mesh points in each subdomain constant.

To see the effect fixing a node has on mesh movement in the majority of the domain and the simulated overlap region we solve Burgers' equation with a traditional moving mesh method on the entire domain. The mesh point at $x = 0.7$ is kept fixed throughout the computation. The left plot of Figure 7.2 illustrates the typical mesh trajectories obtained by applying a moving mesh partial differential equation to a moving front problem with one spatial domain. The mesh points clearly follow and track the front as it moves from left to right. The right of Figure 7.2 illustrates a simulation involving a moving mesh method

applied to Burgers' equation with $\epsilon = 0.01$ and $N = 60$ mesh points. The mesh point F , initially at $x \approx 0.7$, is held fixed by specifying $\dot{x}_F = 0$ in the specification of the moving mesh PDE. As the front moves from left to right the mesh adapts as expected to the left of the fixed mesh point. It is clear, however, the mesh points to the right of the fixed mesh point do not respond until the front approaches the point F . At that time the points on the right race into layer while the points to the left return quickly to a uniform spacing. Sufficient care must be taken to ensure there are enough points to the left of F to maintain stability and accuracy.

In our Moving Schwarz method the inability of the points to the right of the fixed mesh point to "see" the layer will hopefully be tempered somewhat by overlap.

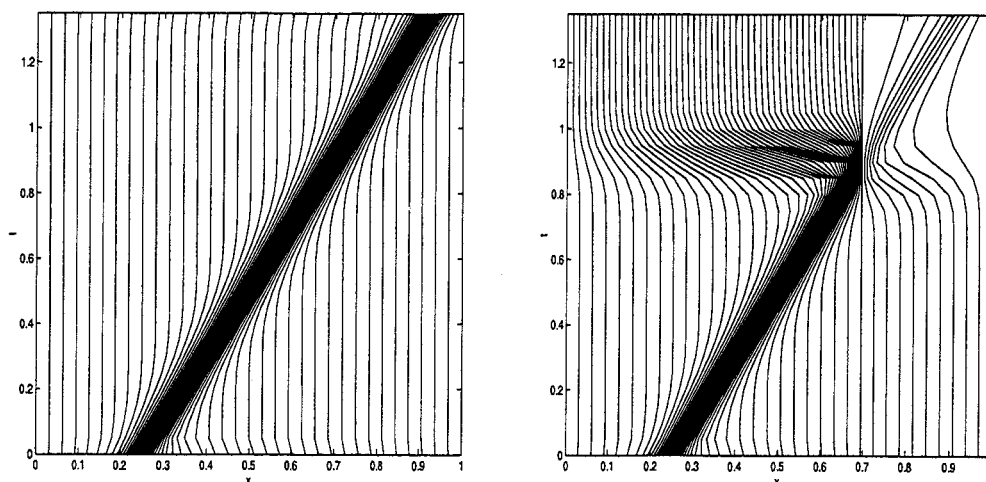


Figure 7.2: Mesh trajectories for Burgers' equation on one domain

7.4.2 Solution of Burgers' Equation with the Schwarz Waveform Moving Mesh method

To provide some initial testing of the Schwarz Waveform Moving Mesh method we solve Burgers' equation with $\epsilon = 1e - 4$ with 40 points in each of the three subdomains. The top row of plots in Figure 7.3 illustrates the computed solution at times $t = 0.25, 0.45$ and $t = 1.7$. The bottom plots show the corresponding error. As expected we see a sharp front moving from left to right. The mesh trajectories as functions of time are displayed in Figure 7.4.

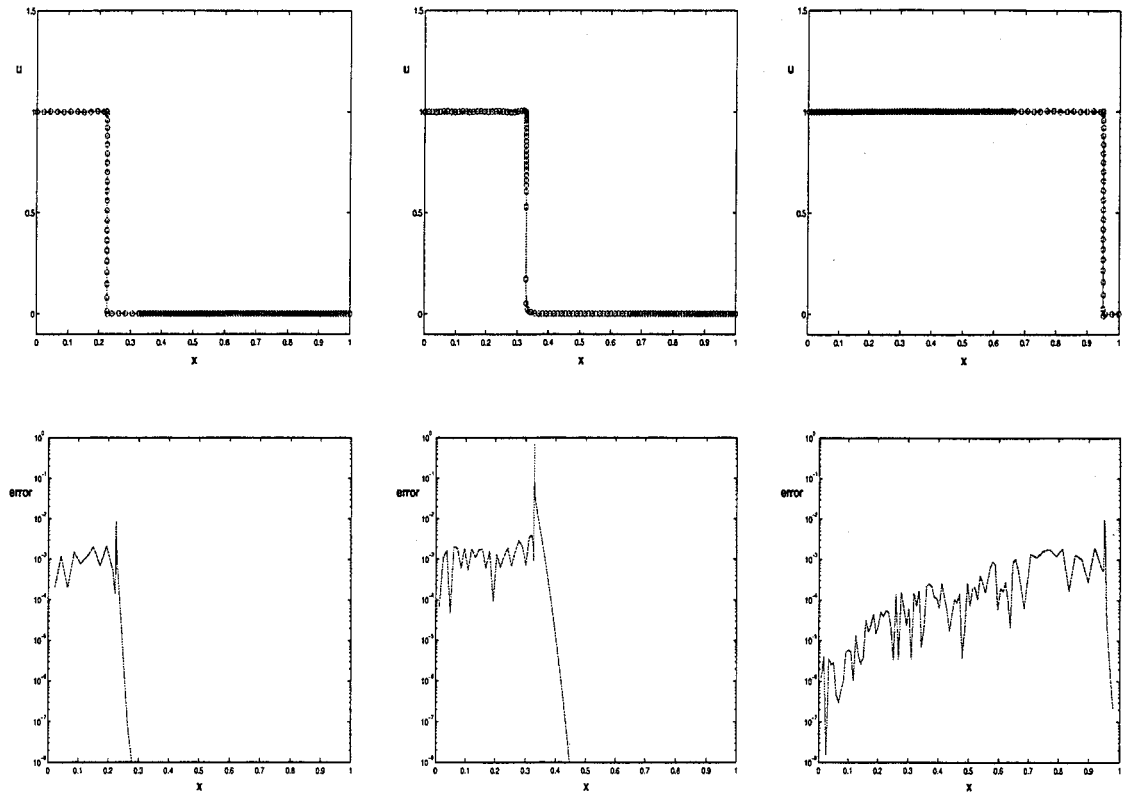


Figure 7.3: Solutions and errors for Burgers' equation with moving Schwarz waveform method at $t = 0.25, 0.45$ and 1.7 .

The solution at $t = 0.25$ and $t = 1.7$ look fine with actual errors on the order of the tolerance controlling the convergence of the Schwarz Waveform iteration. At $t = 0.45$, however, there is an indication of a problem. It is precisely at this time the front encounters the boundary between the first and second subdomains. At that moment we see from the solution that the meshes are having problems communicating. A possible fix may be provided by some of the more sophisticated Schwarz Waveform methods which use “higher-order” transmission conditions at the boundary ([34] and [35]).

Figure 7.5 highlights both the tremendous potential and current difficulties with the Moving Schwarz method. In this plot we have displayed the number of time steps per time window taken by implicit Euler in each subdomain, labeled I, II, and III in the legend of the plot. The dotted vertical lines specify the time at which the layer crosses the subdomain boundary. The data shows that while the layer is completely contained in subdomain I

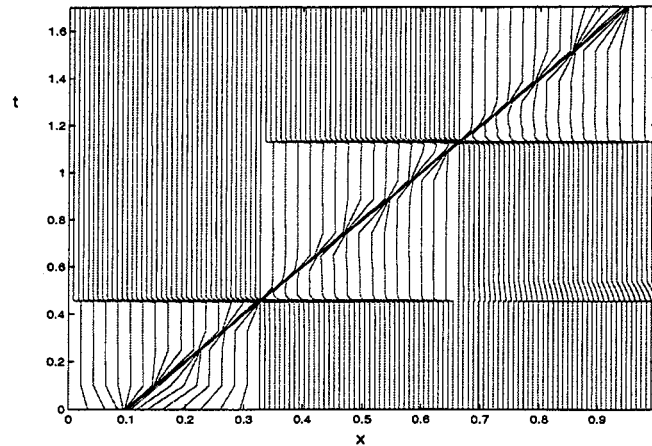


Figure 7.4: Moving Schwarz Mesh Trajectories for Burgers' Equation with $\epsilon = 1e - 4$ and 40 points per domain

($0 \leq t \leq 0.4$), the work involved to integrate over subdomains II and III is negligible. In fact, until the front is well in domain II the time integrator is constantly taking only single digit numbers of time steps. As the front approaches the boundaries the number of time steps in the adjoining domains increases dramatically again indicating the trouble experienced by the Schwarz Waveform.

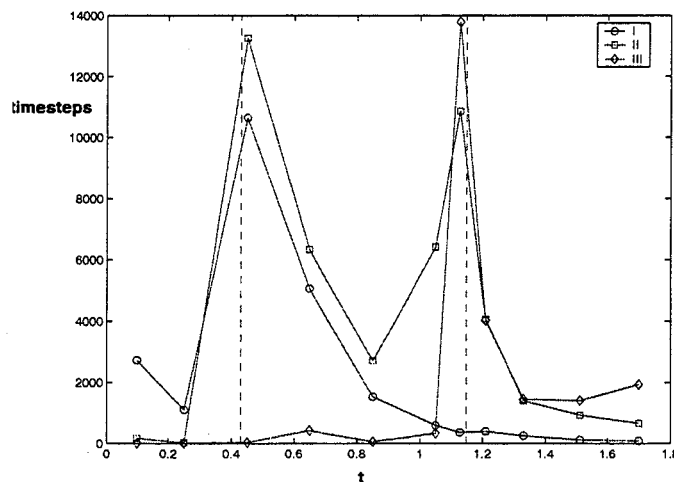


Figure 7.5: Number of time steps taken in each subdomain during each time window.

A final look at what the algorithm is doing as we approach the subdomain boundaries is

given in Figure 7.6. Here we have displayed the size of the time windows used to compute the solution. Again, we see a dramatic decrease in the size of the time window due to the maximum number of allowed time steps being exceeded.

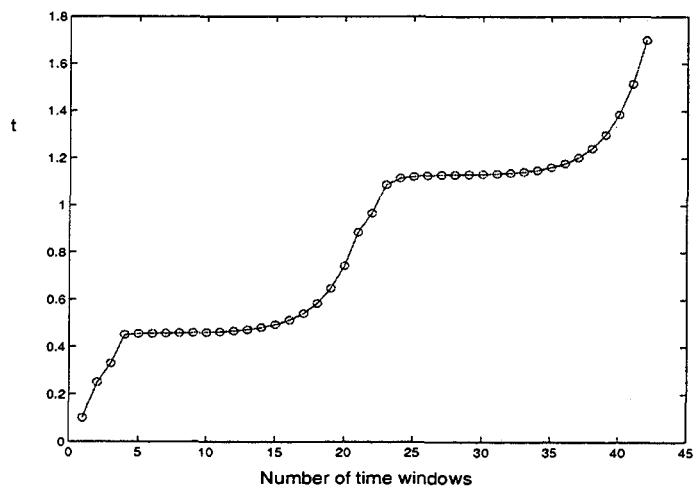


Figure 7.6: Length of time windows for moving Schwarz method applied to Burgers' equation with $\epsilon = 1e - 4$.

Mesh trajectories corresponding to Burgers' equation with the second initial condition are shown in Figure 7.7. In this case, the solution evolves to a front in subdomain two. This results in a nearly uniform mesh in subdomain I for the entire problem. The grids in subdomains two and three, however, react to the evolving solution and carry the front to the boundary at $x = 1$.

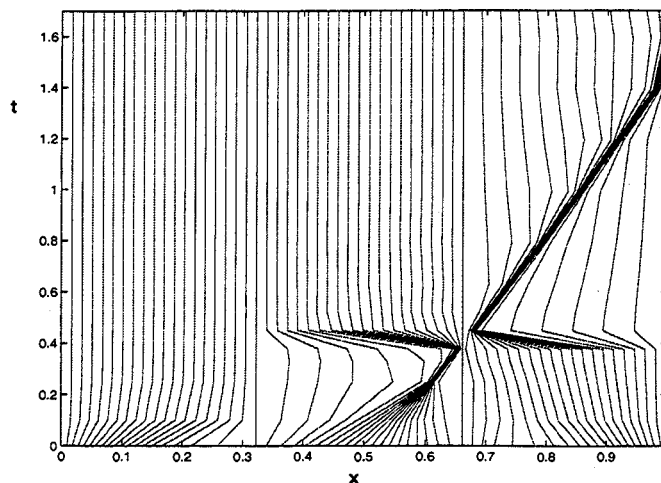


Figure 7.7: Moving Schwarz Mesh Trajectories for Burgers' Equation with $\epsilon = 1e - 3$ and 20 points per domain

7.4.3 Two Spike Problem

Figure 7.8 illustrates the exact solution of the two spike problem for times $t = 0, 0.6, 1.6$ and $t = 2.7$. The solution at $t = 0$ is $u(x, 0) = 0$ and hence is not visible in the plot. At $t = 0.6$ the spike centered at $x_0 = 1/4$ has appeared and continues to grow. The second spike centered at $x_1 = 3/4$ emerges at $t = 3/2$. The final solution shown, at $t = 2.7$ shows both spikes which have nearly reached their maximum height.

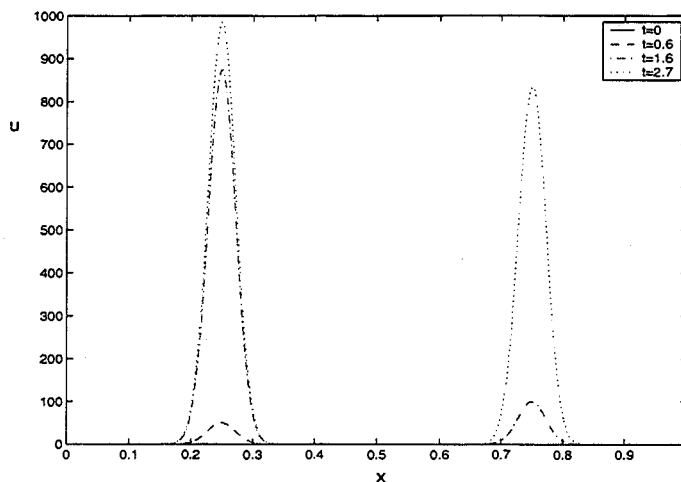


Figure 7.8: Exact solutions of two spike problem at $t = 0, 0.6, 1.6$ and 2.7 .

We solve the two spike problem with 80 mesh points using one and two domains. The final computed solution (at $t = 2.7$) and the corresponding mesh trajectories for the one domain calculation are shown in Figure 7.9. The computed solution is illustrated using solid line and open circle combination. The open circles indicate the location of the mesh points. The exact solution is drawn with a solid line only. Although the moving mesh method does capture the two spikes we do see a loss of accuracy in the region between x_0 and x_1 . Excessive errors arise in the computed solution for $t \geq 3/2$ corresponding to the “birth” of the second spike. Indeed, the method has great difficulty achieving sufficient resolution of the second spike. Attempting to do, by an appropriate movement of mesh points, increases the error in the region of the left spike. From the mesh trajectories we see a relatively quick movement of mesh points from the region of the left spike towards the emerging right spike for $t > 3/2$.

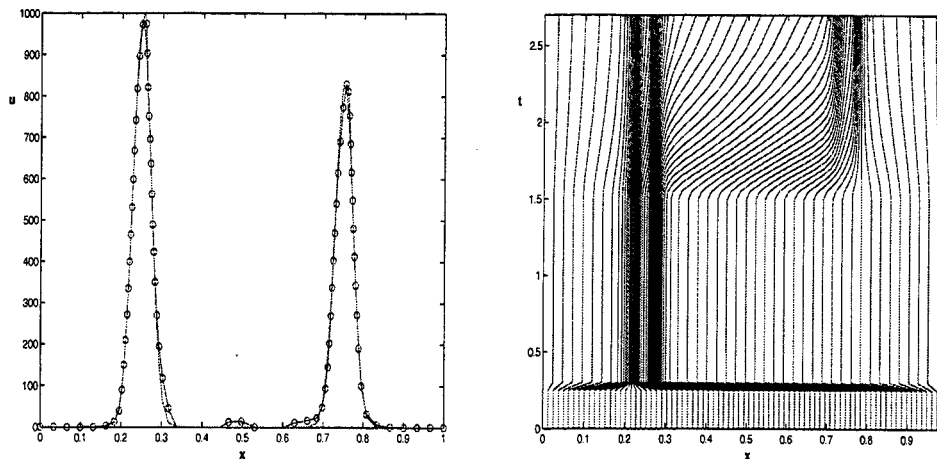


Figure 7.9: Solution at $t = 2.7$ and mesh trajectories for the two spike problem with one domain.

It is important to note that it is not possible to solve this problem to the relatively poor accuracy achieved in Figure 7.9 without a careful choice of monitor function, number of mesh points, and moving mesh parameters (e. g. τ). The number of mesh points and monitor function chosen are vital to allow resolution of the second spike. Using the same number of mesh points and the arclength monitor function it is possible to miss the spike at x_1 completely. We control the proportion of points outside of the initial region of difficulty (by an appropriate choice of α in (4.7)) the second spike is detected and mesh movement

proceeds. The loss of accuracy which occurs as the second spike grows is due in part to the time required for the mesh to adapt to new features in the solution. This is controlled by the moving mesh parameter τ in MMPDE4. Decreasing the value of τ facilitates a quicker mesh movement and hence a quicker return to an equidistributed grid. This will improve the accuracy of the computed solution at the expense, however, of much smaller steps in the time integration.

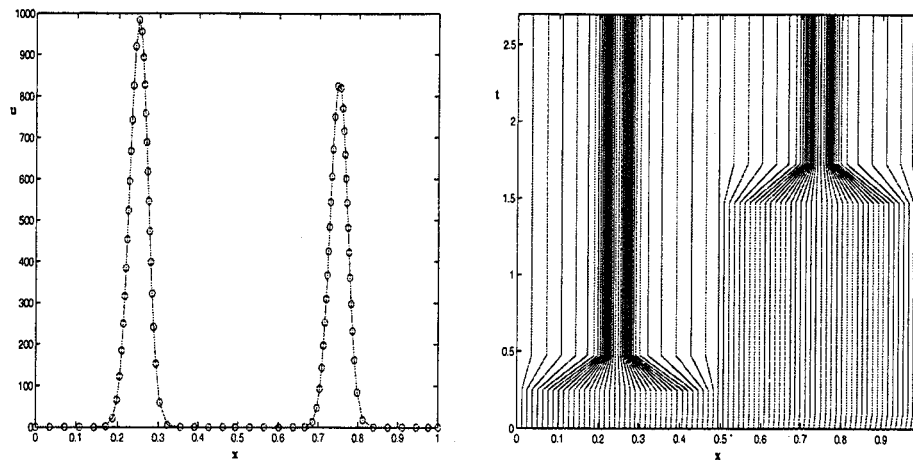


Figure 7.10: Solution at $t = 2.7$ and mesh trajectories for the two spike problem with two subdomains.

This problem is ideal for a two domain simulation of the Schwarz Waveform moving mesh method. The computed and exact solutions and resulting mesh trajectories are shown in Figure 7.10. Compared to the one domain calculation we see improved accuracy between the spikes and in regions of high curvature¹. As expected we see the mesh remains uniform in each subdomain until the spike in that region is activated. At that point the mesh in each subdomain adapts to the evolving features of the solution with a simple arclength monitor function and moderate values of τ . In fact, comparable accuracy is possible with many fewer mesh points than the one domain calculation.

In Figure 7.11 we compare the time steps taken by the ODE solver (backward Euler) for the one domain (left) and two subdomains (right) simulations. In the two subdomain case we have illustrated the time steps for subdomain I (light) and subdomain II (dark)

¹A monitor function which provides a balance between arclength and curvature may improve the one domain calculation in this aspect.

corresponding to the first waveform iteration. We see that the time steps chosen for the one domain calculation are quite large until $t = 1/4$ when the left spike begins to grow. The time steps remain relatively steady at 10^{-4} until the second spike emerges at $t = 3/2$. Immediately the time steps are reduced by an order of magnitude as the mesh points race to adjust to new features in the solution.

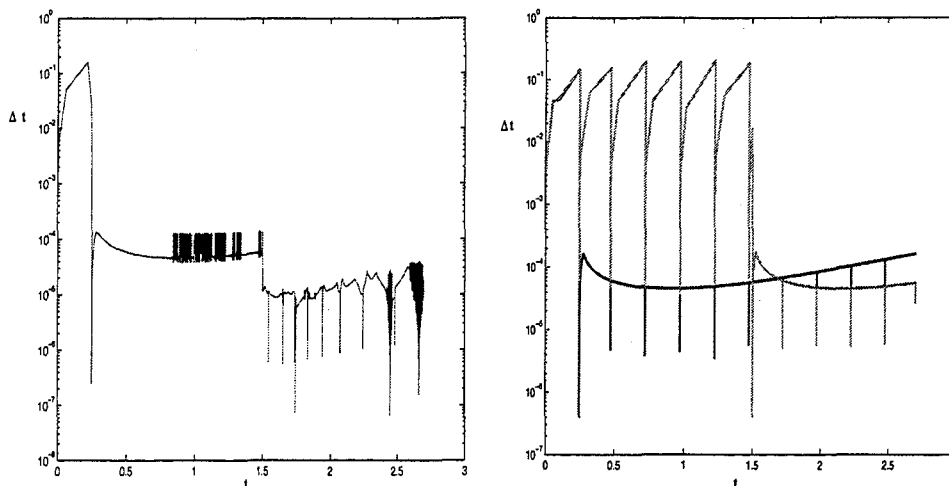


Figure 7.11: Time steps for one domain (left) and two subdomains (right) solution of the two spike problem.

The time steps chosen in each subdomain by the Schwarz Waveform moving mesh method are controlled primarily by the local features of the solution. The time steps in subdomain I react to the spike developing at $t = 1/4$ and reduce to 10^{-4} and remain at this level for the duration of the run. Little work is required to integrate the solution and mesh components in subdomain II until $t = 3/2$. At the time the size of the acceptable time steps decrease to a level similar to those on subdomain I. It is important to note that the time steps in subdomain I are not affected by the development of the second spike. No mesh points move from one subdomain to another. This keeps the time steps an order of magnitude larger than the one domain calculation. This more than compensates for the iteration required in the Schwarz Waveform moving mesh method, resulting in a 40% faster run time.

Chapter 8

Conclusions and Future Work

In this final chapter we summarize the contents of this dissertation, highlighting the most important results and observations and indicating current and future research directions.

Chapter 1 provides a concise overview of finite difference methods for solving two point boundary value problems (BVPs). Difficulties with discretizations on uniform grids are discussed to provide a context for the next chapter.

In Chapter 2 we consider the problem of grid selection for linear BVPs. Specifically, we explore the connection between spectral properties of the discretization matrix A and mesh quality or resolution. The existence of real eigenvalues is found to be correlated to the resolution of boundary or interior layers; however, explicit information pertaining to the location of needed mesh refinement is apparently not available. Using the singular value decomposition of A it is possible to write the discrete solution of the BVP as an expansion of singular vectors. The dominant singular vectors of A indicate mesh quality and provide insight to the spatial location of layers and required refinement. Computing the dominant singular vector(s) efficiently has proven difficult, although recent fast low rank SVD methods [60] may be useful for higher dimensional problems. This exploration is ongoing. An alternate technique to obtain spatially relevant mesh information by simple iterations on the linear system is proposed and tested for a variety of model problems. Further experimentation and theoretical investigation are required to better identify specific iterative methods which are well suited to this application. We expect that model problems which are at least two dimensional are necessary to illustrate significant efficiencies and hence the real potential of the technique. This chapter concludes with a preliminary look at the M -matrix structure of A and connections with grid selection. Experiments indicate

that local M -matrix structure is a sufficient condition for mesh quality. Unfortunately, the investigation has thus far not resulted in any obvious conditions which would ensure an M -matrix structure.

The inverse positivity of a perturbation of a tridiagonal M -matrix is considered in Chapter 3. We obtain a simple bound on the size of the perturbation in terms of the entries and size of the tridiagonal matrix. An extension of these results to include perturbations of more general band matrices is currently underway. An application of this result to time stepping for higher order partial differential equations (PDEs) which require nonnegative physical solutions is presented. More work is required to identify classes of PDEs as well as spatial and temporal discretizations which yield matrices of this form.

Chapters 4 and 5 are used to set the stage for the Schwarz Waveform moving mesh method proposed in Chapter 6. A brief survey of moving mesh methods in one spatial dimension is given in Chapter 4. The concept of equidistribution and the development of the moving mesh PDE is reviewed. Current implementation strategies are indicated. In Chapter 5 we survey decoupled and multirate integration strategies. The idea is to identify solution components which evolve on different time scales. These groups of components are then integrated with different time steps and possibly different integration methods. The main difficulties (and hence potentially fruitful research areas) include the dynamic partitioning of components and the estimation and control of the error associated with the decoupling procedure. Of primary interest for the rest of the thesis is the Schwarz Waveform method for PDEs. Most of the other methods reviewed are developed and tested in the ODE context. Further study of these techniques in the PDE situation is warranted.

In Chapter 6 we propose a new Schwarz Waveform moving mesh method. This is a natural coupling of domain decomposition with moving mesh methods. The method is defined and implementation details are provided. Numerical results for various decoupled integration methods on fixed and moving grids are presented in Chapter 7. The Schwarz Waveform moving mesh method performs quite nicely for a model problem having a "two spike" solution, which serves as a prototype for problems having difficult solution behaviour in more than one region. The numerical solution is found to be not only more accurate, but it is computed more efficiently than for a moving mesh method on one domain. The Schwarz Waveform method is inherently multirate, allowing different time steps for solution and mesh components in different domains. It would be natural in this context to allow for different moving mesh parameters in different subdomains. We intend to develop and test

strategies which take advantage of this flexibility. The ability to partition the unknowns spatially also allows grids to adapt to rapidly changing features of the solution in different regions of the physical domain. Tests on Burgers' equation prove to be more of a challenge, as regions of rapid change in the solution moves from subdomain to subdomain. This causes an increase in the number of time steps in domains which share the difficult regions of the solution. Depending on the details of the implementation this may cause a decrease in the size of the time window and hence an increase in the total number of time windows. The number of waveform iterations required to achieve convergence increases. Many of these difficulties appear related to the changing size of the overlap region. This may be circumvented by posing the problem in the computational domain ξ and discretizing on a uniform grid. Numerical studies using this approach are currently underway. Another possible approach is to develop higher order transmission conditions, particularly suited to moving meshes, to enable the passing of data from one subdomain to another. Finally, research into ways of extending the Schwarz Waveform moving mesh method to problems in higher dimensions is ongoing.

Bibliography

- [1] R. ALEXANDER, *The modified Newton method in the solution of stiff ordinary differential equations*, Math. Comp., 57 (1991), pp. 673–701.
- [2] D. ALLEN AND R. SOUTHWELL, *Relaxation methods applied to determine motion, in two dimensions, of a viscous fluid past a fixed cylinder*, J. Mech. Appl. Math., 8 (1955), pp. 129–145.
- [3] R. ALMGREN, A. BERTOZZI, AND M. P. BRENNER, *Stable and unstable singularities in the unforced Hele-Shaw cell*, Phys. Fluids, 8 (1996), pp. 1356–1370.
- [4] J. F. ANDRUS, *Numerical solution of systems of ordinary differential equations separated into subsystems*, SIAM J. Numer. Anal., 16 (1979), pp. 605–611.
- [5] ———, *Automatic integration of systems of second-order odes separated into subsystems*, SIAM J. Numer. Anal., 20 (1983), pp. 815–827.
- [6] ———, *A Runge-Kutta method with stepsize control for separated systems of first-order ODEs*, Appl. Math. Comput., 59 (1993), pp. 193–214.
- [7] ———, *Stability of a multi-rate method for numerical integration of ODEs*, Comput. Math. Appl., 25 (1993), pp. 3–14.
- [8] U. ASCHER, R. MATTHEIJ, AND R. RUSSELL, *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*, Classics in Applied Mathematics, SIAM, Philadelphia, 1995.
- [9] U. M. ASCHER AND L. R. PETZOLD, *Computer methods for ordinary differential equations and differential-algebraic equations*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1998.
- [10] G. BECKETT AND J. A. MACKENZIE, *Uniformly convergent high order finite element solutions of a singularly perturbed reaction-diffusion equation using mesh equidistribution*, Appl. Numer. Math., 39 (2001), pp. 31–45.
- [11] G. BECKETT, J. A. MACKENZIE, A. RAMAGE, AND D. SLOAN, *On the numerical solution of one-dimensional pdes using adaptive methods based on equidistribution*, J. Comput. Phys., 167 (2001), pp. 372–392.

- [12] P. BERMAN AND R. PLEMMONS, *Nonnegative matrices in the mathematical sciences*, Academic Press, 1979.
- [13] A. L. BERTOZZI, *Symmetric singularity formation in lubrication-type equations for interface motion*, SIAM J. Appl. Math., 56 (1996), pp. 681–714.
- [14] A. L. BERTOZZI, M. P. BRENNER, T. F. DUPONT, AND L. P. KADANOFF, *Singularities and similarities in interface flows*, in Trends and perspectives in applied mathematics, vol. 100 of Appl. Math. Sci., Springer, New York, 1994, pp. 155–208.
- [15] A. N. BROOKS AND T. J. R. HUGHES, *Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations*, Comput. Methods Appl. Mech. Engrg., 32 (1982), pp. 199–259. FENOMECH '81, Part I (Stuttgart, 1981).
- [16] C. J. BUDD, W. HUANG, AND R. D. RUSSELL, *Moving mesh methods for problems with blow-up*, SIAM J. Sci. Comput., 17 (1996), pp. 305–327.
- [17] K. BURRAGE, *Parallel and sequential methods for ordinary differential equations*, The Clarendon Press Oxford University Press, New York, 1995. Oxford Science Publications.
- [18] X. CAI, *Additive schwarz algorithms for parabolic convection-diffusion equations*, Numerische Mathematik, 60 (1991), pp. 41–61.
- [19] W. CAO, W. HUANG, AND R. D. RUSSELL, *A study of monitor functions for two-dimensional adaptive mesh generation*, SIAM J. Sci. Comput., 20 (1999), pp. 1978–1994 (electronic).
- [20] B. CHAWLA, H. GUMMEL, AND P. KOZAH, *Motis—an mos timing simulator*, IEEE Trans. on Circuits and Systems, CAS-22 (1975), pp. 901–909.
- [21] A. CURTIS, M. POWELL, AND J. REID, *On the estimation of sparse Jacobian matrices*, J.I.M.A, 13 (1974), pp. 117–120.
- [22] C. DE BOOR, *Good approximation by splines with variable knots. II*, in Conference on the Numerical Solution of Differential Equations (Univ. Dundee, Dundee, 1973), Springer, Berlin, 1974, pp. 12–20. Lecture Notes in Math., Vol. 363.
- [23] P. DEUFLHARD AND J. HEROTH, *Dynamic dimension reduction in ode models*, in Scientific Computing in Chemical Engineering, F. Keil, W. Mackens, H. Voß, and J. Werther, eds., SV, 1996, pp. 29–43.
- [24] P. DEUFLHARD, J. HEROTH, AND U. MAAS, *Towards dynamic dimension reduction in reactive flow problems*, in Proc. 3rd Workshop on Modelling of Chemical Reaction Systems (CD-Version), Heidelberg, 1996.

- [25] E. DORFI AND L. DRURY, *Simple adaptive grids for 1-d initial value problems*, J. Comput. Phys., 69 (1987), pp. 175–195.
- [26] W. ECKHAUS, *Matched asymptotic expansions and singular perturbations*, North-Holland Publishing Co., Amsterdam, 1973. North-Holland Mathematics Studies, No. 6.
- [27] C. ENGSTLER AND C. LUBICH, *Multirate extrapolation methods for differential equations with different time scales*, Computing, 58 (1997), pp. 173–185.
- [28] C. ENGSTLER AND C. LUBICH, *MUR8: a multirate extension of the eighth-order Dormand-Prince method*, Appl. Numer. Math., 25 (1997), pp. 185–192. Special issue on time integration (Amsterdam, 1996).
- [29] J. FLAHERTY, R. LOY, M. SHEPHARD, B. SZYMANSKI, J. TERESCO, AND L. ZIANTZ, *Adaptive local refinement with octree load balancing for the parallel solution of three-dimensional conservation laws*, J. Parallel Distrib. Comput., 47 (1997), pp. 139–152.
- [30] J. E. FLAHERTY AND P. K. MOORE, *Integrated space-time adaptive hp-refinement methods for parabolic systems*, Appl. Numer. Math., 16 (1995), pp. 317–341.
- [31] M. J. GANDER, *Overlapping Schwarz waveform relaxation for parabolic problems*, in Tenth International Conference on Domain Decomposition Methods, J. Mandel, C. Farhat, and X.-C. Cai, eds., AMS, Contemporary Mathematics 218, 1998.
- [32] ———, *A waveform relaxation algorithm with overlapping splitting for reaction diffusion equations*, Numerical Linear Algebra with Applications, (1998), pp. 125–145.
- [33] M. J. GANDER AND D. DAOUD, *Overlapping schwarz waveform relaxation for convection reaction diffusion problems*, in Thirteenth International Conference on Domain Decomposition Methods, N. Debit, M. Garbey, R. Hoppe, J. Périaux, D. Keyes, and Y. Kuznetsov, eds., Bergen, 2002, Domain Decomposition Press, pp. 227–233.
- [34] M. J. GANDER, L. HALPERN, AND F. NATAF, *Optimal convergence for overlapping and non-overlapping Schwarz waveform relaxation*, in Eleventh international Conference of Domain Decomposition Methods, C.-H. Lai, P. Bjørstad, M. Cross, and O. Widlund, eds., ddm.org, 1999.
- [35] ———, *Optimized Schwarz methods*, in Twelfth International Conference on Domain Decomposition Methods, Chiba, Japan, T. Chan, T. Kako, H. Kawarada, and O. Pironneau, eds., Bergen, 2001, Domain Decomposition Press, pp. 15–28.
- [36] M. J. GANDER AND A. M. STUART, *Space-time continuous analysis of waveform relaxation for the heat equation*, SIAM J. Sci. Comput., 19 (1998), pp. 2014–2031.

- [37] M. J. GANDER AND H. ZHAO, *Overlapping schwarz waveform relaxation for the heat equation in n -dimensions*, preprint, (2000).
- [38] E. GILADI AND H. B. KELLER, *Space-time domain decomposition for parabolic problems*, Numer. Math., 93 (2002), pp. 279–313.
- [39] D. GILBARG AND N. TRUDINGER, *Elliptic Partial Differential Equations of Second Order*, Springer-Verlag, Berlin Heidelberg, second ed., 1983.
- [40] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, John Hopkins Studies in the Mathematical Sciences, John Hopkins University Press, Baltimore, MD, third ed., 1996.
- [41] H. GREENSPAN, *On the motion of a small viscous droplet that wets a surface*, J. Fluid Mech., 84 (1978), pp. 125–143.
- [42] K. GUSTAFSSON AND G. SÖDERLIND, *Control strategies for the iterative solution of nonlinear equations in ODE solvers*, SIAM J. Sci. Comput., 18 (1997), pp. 23–40. Dedicated to C. William Gear on the occasion of his 60th birthday.
- [43] M. HESTENES AND E. STIEFEL, *Methods of conjugate gradients for solving linear systems*, J. Res. Nat. Bur. Stand., 49 (1952), pp. 409–436.
- [44] F. B. HILDEBRAND, *Introduction to numerical analysis*, McGraw-Hill Book Co., New York, second ed., 1974. International Series in Pure and Applied Mathematics.
- [45] E. HOFER, *A partially implicit method for large stiff systems of ode's with only few equations introducing small time-constants*, SIAM J. Numer. Anal., 13 (1976), pp. 645–663.
- [46] D. HOFF, *A linearly implicit finite-difference scheme for the one-dimensional porous medium equation*, Math. Comp., 45 (1985), pp. 23–33.
- [47] W. HUANG, *Practical aspects of formulation and solution of moving mesh partial differential equations*, J. Comput. Phys., 171 (2001), pp. 753–775.
- [48] W. HUANG, Y. REN, AND R. D. RUSSELL, *Moving mesh methods based on moving mesh partial differential equations*, J. Comput. Phys., 113 (1994), pp. 279–290.
- [49] ———, *Moving mesh partial differential equations (MMPDES) based on the equidistribution principle*, SIAM J. Numer. Anal., 31 (1994), pp. 709–730.
- [50] W. HUANG AND R. D. RUSSELL, *Moving mesh strategy based on a gradient flow equation for two-dimensional problems*, SIAM J. Sci. Comput., 20 (1999), pp. 998–1015 (electronic).
- [51] A. IL'IN, *A difference scheme for a differential equation with a small parameter affecting the highest derivative*, Mat. Zametki, 6 (1969), pp. 237–248.

- [52] J. JANSSEN AND S. VANDEWALLE, *Multigrid waveform relaxation on spatial finite-element meshes: The continuous-time case*, SIAM J. Numer. Anal., 33 (1996), pp. 456–474.
- [53] R. JELTSCH AND B. POHL, *Waveform relaxation with overlapping splittings*, SIAM J. Sci. Comput., 16 (1995), pp. 40–49.
- [54] J. KEVORKIAN AND J. COLE, *Perturbation Methods in Applied Mathematics*, Springer-Verlag, Berlin Heidelberg, 1981.
- [55] Y. KUZNETSOV, *Domain decomposition methods for unsteady convection-diffusion problems*, in Proceedings of the Ninth International Conference in Computing Methods in Applied Sciences and Engineering, 1990, pp. 211–227.
- [56] E. LELARASMEE, A. RUEHLI, AND A. SANGIOVANNI-VINCENTELLI, *The waveform relaxation method for time domain analysis of large scale integrated circuits*, IEEE Trans. on CAD of IC and Syst., 1 (1982), pp. 131–145.
- [57] C. LUBICH AND A. OSTERMANN, *Multi-grid dynamic iteration for parabolic equations*, BIT, 27 (1987), pp. 216–234.
- [58] U. MAAS AND S. POPE, *Simplifying chemical kinetics: intrinsic low dimensional manifolds in composition space*, Combustion and Flame, 88 (1992), pp. 239–264.
- [59] J. A. MACKENZIE, *The efficient generation of simple two-dimensional adaptive grids*, SIAM J. Sci. Comput., 19 (1998), pp. 1340–1365.
- [60] F. MCSHERRY AND D. ACHLIOPTAS, *Fast computation of low rank matrix approximations*, Preprint, Department of Computer Science and Engineering, University of Washington, (2002).
- [61] G. A. MEURANT, *A domain decomposition method for parabolic problems*, Applied Numerical Mathematics, 8 (1991), pp. 427–441.
- [62] ———, *A review on the inverse of symmetric tridiagonal and block tridiagonal matrices*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 707–728.
- [63] ———, *Computer solution of large linear systems*, vol. 28 of Studies in Mathematics and its Applications, North-Holland Publishing Co., Amsterdam, 1999.
- [64] U. MIEKKALA AND O. NEVANLINNA, *Convergence of dynamic iteration methods for initial value problem*, SIAM J. Sci. Statist. Comput., 8 (1987), pp. 459–482.
- [65] ———, *Sets of convergence and stability regions*, BIT, 27 (1987), pp. 554–584.
- [66] J. MILLER, E. O'RIORDAN, AND G. SHISHKIN, *Fitted Numerical Methods for Singular Perturbation Problems*, World Scientific, Singapore, 1996.

- [67] K. MILLER, *Moving finite elements. II*, SIAM J. Numer. Anal., 18 (1981), pp. 1033–1057.
- [68] K. MILLER AND R. N. MILLER, *Moving finite elements. I*, SIAM J. Numer. Anal., 18 (1981), pp. 1019–1032.
- [69] P. K. MOORE AND J. E. FLAHERTY, *A local refinement finite-element method for one-dimensional parabolic systems*, SIAM J. Numer. Anal., 27 (1990), pp. 1422–1444.
- [70] ———, *Adaptive local overlapping grid methods for parabolic systems in two space dimensions*, J. Comput. Phys., 98 (1992), pp. 54–63.
- [71] R. NABBEN, *Decay rates of the inverse of nonsymmetric tridiagonal and band matrices*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 820–837 (electronic).
- [72] ———, *Two-sided bounds on the inverses of diagonally dominant tridiagonal matrices*, Linear Algebra & Appl., 287 (1999), pp. 289–305. Special issue celebrating the 60th birthday of Ludwig Elsner.
- [73] O. NEVANLINNA, *Remarks on Picard-Lindelöf iteration. I*, BIT, 29 (1989), pp. 328–346.
- [74] ———, *Remarks on Picard-Lindelöf iteration. II*, BIT, 29 (1989), pp. 535–562.
- [75] ———, *Linear acceleration of Picard-Lindelöf iteration*, Numer. Math., 57 (1990), pp. 147–156.
- [76] R. E. O'MALLEY, JR., *Introduction to singular perturbations*, Academic Press [A subsidiary of Harcourt Brace Jovanovich, Publishers], New York-London, 1974. Applied Mathematics and Mechanics, Vol. 14.
- [77] A. OSTROWSKI, *Über die determinanten mit überwiegender hauptdiagonale*, Comment. Math. Helv., 10 (1937), pp. 69–96.
- [78] A. M. OSTROWSKI, *Note on bounds for determinants with dominant principal diagonal*, Proc. Amer. Math. Soc., 3 (1952), pp. 26–30.
- [79] C. E. PEARSON, *On non-linear ordinary differential equations of boundary layer type.*, J. Math. and Phys., 47 (1968), pp. 351–358.
- [80] L. R. PETZOLD, *A description of DASSL: a differential/algebraic system solver*, in Scientific computing (Montreal, Que., 1982), IMACS Trans. Sci. Comput., I, IMACS, New Brunswick, NJ, 1983, pp. 65–68.
- [81] E. PICARD, *Sur l'application des méthodes d'approximations successives à l'étude de certaines équations différentielles ordinaires*, Journal de Mathématiques Pures et Appliquées, 9 (1893), pp. 217–271.

- [82] Y. REN AND R. D. RUSSELL, *Moving mesh techniques based upon equidistribution, and their stability*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 1265–1286.
- [83] H.-G. ROOS, M. STYNES, AND L. TOBISKA, *Numerical Methods for Singularly Perturbed Differential Equations*, Springer-Verlag, Berlin Heidelberg, 1996.
- [84] Y. SAAD, *Numerical methods for large eigenvalue problems*, Algorithms and Architectures for Advanced Scientific Computing, Manchester University Press, Manchester, 1992.
- [85] Y. SAAD AND M. H. SCHULTZ, *GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.
- [86] J. SAND AND S. SKELBOE, *Stability of backward Euler multirate methods and convergence of waveform relaxation*, BIT, 32 (1992), pp. 350–366.
- [87] M. E. SEZER AND D. D. ŠILJAK, *Nested ϵ -decompositions and clustering of complex systems*, Automatica J. IFAC, 22 (1986), pp. 321–331.
- [88] ———, *Nested epsilon decompositions of linear systems: weakly coupled and overlapping blocks*, SIAM J. Matrix Anal. Appl., 12 (1991), pp. 521–533.
- [89] L. SIROVICH, *Techniques of asymptotic analysis*, Applied Mathematical Sciences, Vol. 2, Springer-Verlag, New York, 1971.
- [90] S. SKELBOE, *Stability properties of backward differentiation multirate formulas*, Appl. Numer. Math., 5 (1989), pp. 151–160. Recent theoretical results in numerical ordinary differential equations.
- [91] ———, *Accuracy of decoupled implicit integration formulas*, SIAM J. Sci. Comput., 21 (2000), pp. 2206–2224 (electronic).
- [92] S. SKELBOE AND P. U. ANDERSEN, *Stability properties of backward Euler multirate formulas*, SIAM J. Sci. Statist. Comput., 10 (1989), pp. 1000–1009.
- [93] D. SMITH, *Singular-perturbation theory*, Cambridge University Press, New York, 1985.
- [94] J. M. STOCKIE, J. A. MACKENZIE, AND R. D. RUSSELL, *A moving mesh method for one-dimensional hyperbolic conservation laws*, SIAM J. Sci. Comput., 22 (2000), pp. 1791–1813 (electronic).
- [95] S. TA'ASAN AND H. ZHANG, *On multigrid waveform relaxation methods*, Rep. Inst. for Comp. Apps. in Science and Engineering, (1994).
- [96] R. TARJAN, *Depth-first search and linear graph algorithms*, SIAM J. Comput., 1 (1972), pp. 146–160.

- [97] D. ŠILJAK, *Decentralized Control of Complex Systems*, vol. 184 of Mathematics in Science and Engineering, Academic Press, New York, 1991.
- [98] W. WASOW, *Asymptotic expansions for ordinary differential equations*, Pure and Applied Mathematics, Vol. XIV, Interscience Publishers John Wiley & Sons, Inc., New York-London-Sydney, 1965.
- [99] A. B. WHITE, JR., *On selection of equidistributing meshes for two-point boundary-value problems*, SIAM J. Numer. Anal., 16 (1979), pp. 472–502.
- [100] J. WHITE AND A. SANGIOVANNI-VINCENTELLI, *Relaxation techniques for the simulation of VLSI circuits*, Kluwer Academic Publishers, Boston, 1987.
- [101] J. WHITE, A. SANGIOVANNI-VINCENTELLI, F. ODEH, AND A. RUEHLI, *Waveform relaxation: theory and practice*, Trans. of Soc. for Computer Simulation, 2 (1985), pp. 95–133.
- [102] A. I. ZEČEVIĆ AND N. GAČIĆ, *A partitioning algorithm for the parallel solution of differential-algebraic equations by waveform relaxation*, IEEE Trans. Circuits Systems I Fund. Theory Appl., 46 (1999), pp. 421–434.
- [103] L. ZHORNITSKAYA AND A. L. BERTOZZI, *Positivity-preserving numerical schemes for lubrication-type equations*, SIAM J. Numer. Anal., 37 (2000), pp. 523–555 (electronic).