J.W. Thomas

Numerical Partial Differential Equations

Conservation Laws and Elliptic Equations

With 99 Illustrations

Springer

J.W. Thomas
Department of Mathematics
Colorado State University
Fort Collins, CO 80543
USA

Series Editors

J.E. Marsden
Control and Dynamical Systems 107-81
California Institute of Technology
Pasadena, CA 91125
USA

L. Sirovich
Division of Applied Mathematics
Brown University
Providence, RI 02912
USA

M. Golubitsky
Department of Mathematics
University of Houston
Houston, TX 77204-3476
USA

W. Jäger
Department of Applied Mathematics
Universität Heidelberg
Im Neuenheimer Feld 294
69120 Heidelberg
Germany

Mathematics Subject Classification(1991): 65-01, 65M06, 65N06

Library of Congress Cataloging-in-Publication Data
Thomas J.W. (James William), 1941–
Numerical partial differential equations : finite difference methods / J.W. Thomas
p. cm. — (Texts in applied mathematics ; 22)
Includes bibliographical references and index.
ISBN 0-387-97999-9 (alk. paper)
ISBN 0-387-98346-5 (alk. paper)
1. Differential equations, Partial—Numerical solutions.
2. Finite differences. I. Title. II. Series.
QA377.T495 1995
315´.353—dc20 95-17143

Printed on acid-free paper.

Production managed by Terry Kornak; manufacturing supervised by Nancy Wu.
Photocomposed copy prepared by the author.
Printed and bound by Maple-Vail Book Manufacturing Group, York, PA.
Printed in the United States of America.

9 8 7 6 5 4 3 2 1

ISBN 0-387-98346-5 Springer-Verlag New York Berlin Heidelberg SPIN 10557067

This book is dedicated to some of the women in my life.

Susan, Shannon, Lindsay

Britt, Stephanie, Karen, Lisa

Megan, Kristen, Lindsay, Amanda, Kirsten

Kelley, Rachel, Kaylee, Lauren, Lindsay

Sara, Brooke, Randi, Ashlee

Kayla, Danielle, Cara, Eryn, Lauren

Caitlin, Alysha, Alex, Tara, Maree, Elizabeth, Caitlin

Preface

This book is the second part of a two part text on the numerical solution of partial differential equations. Part 1 (*TAM 22: Numerical Partial Differential Equations: Finite Difference Methods*) is devoted to the basics and includes consistency, stability and convergence results for one and two dimensional parabolic and hyperbolic partial differential equations—both scalar equations and systems of equations. This volume, subtitled *Conservation Laws and Elliptic Equations*, includes stability results for difference schemes for solving initial–boundary–value problems, analytic and numerical results for both scalar and systems of conservation laws, numerical methods for elliptic equations and an introduction to methods for irregularly shaped regions and for computation problems that need grid refinements. In the Preface to Part 1 (included below), I describe the many ways that I have taught courses out of the material in both Parts 1 and 2. Although I will not repeat those descriptions here, I do emphasize that the two parts of the text are strongly intertwined. Part 1 was used to set up some of the material done in Part 2, and Part 2 uses many results from Part 1. The contribution that I hope Chapter 8 of Part 2 makes to the subject is to give a description of how to use the Gustafsson-Kreiss-Sundström-Osher theory (GKSO theory) to choose numerical boundary conditions when they are necessary. In Chapters 9 and 10 I try to give a reasonably complete coverage of numerical methods for solving conservation laws and elliptic equations. Chapter 11 is meant to introduce the reader to the fact that there are methods for treating irregular regions and for placing refined grids in regions that need them. It is hoped that Parts 1 and 2 help prepare the reader to solve a broad spectrum of problems involving partial

differential equations.

In addition to the people I have already thanked in the Preface to Part 1, I would like to thank Ross Heikes who pushed me to include as much as I did in Chapter 9. I would also like to pay tribute to Amiram Harten. His papers, which were readable and of excellent quality, have made a large contribution to the field of the numerical solution of conservation laws. And finally, I would like to thank my family, Ann, David, Michael, Carrie and Susan, for the patience shown when Part 2 took me much longer to write than I predicted. As before, the mistakes are mine and I would appreciate it if you sould send any mistakes that you find to thomas@math.colostate.edu. Thank you.

J.W. Thomas

Preface to Part 1

This textbook is in two parts. The first part contains Chapters 1–7 and is subtitled *Finite Difference Methods*. The second part contains Chapters 8–11 and is subtitled *Conservation Laws and Elliptic Equations*. This text was developed from material presented in a year long, graduate course on using difference methods for the numerical solution of partial differential equations. Writing this text has been an iterative process, and much like the Jacobi iteration scheme presented in Chapter 10, convergence has been slow. The course at Colorado State University is designed for graduate students in both applied mathematics and engineering. The students are required to have at least one semester of partial differential equations and some programming capability. Generally, the classes include a broad spectrum of types of students, ranging from first year mathematics graduate students with almost no physical intuition into the types of problems we might solve, to third year engineering graduate students who have a lot of physical intuition and know what types of problems they personally want to solve and why they want to solve them. Since the students definitely help shape the class that is taught, they probably have also helped to shape this text.

There are several distinct goals of the courses. One definite goal is to prepare the students to be competent practitioners, capable of solving a large range of problems, evaluating numerical results and understanding how and why results might be bad. Another goal is to prepare the applied mathematics Ph.D. students to take additional courses (the third course in our sequence is a course in computational fluid dynamics which requires

both semesters taught out of this text) and to write theses in applied mathematics.

One of the premises on which this text is based is that in order to understand the numerical solution of partial differential equations the student must solve partial differential equations. The text includes homework problems that implement different aspects of most of the schemes discussed. As a part of the implementation phase of the text, discussions are included on how to implement the various schemes. In later parts of the text, we return to earlier problems to discuss the results obtained (or that should have been obtained) and to explain why the students got the results they did. Throughout the text, the problems sometimes lead to bad numerical results. As I explain to my students, since these types of results are very common in the area of numerical solutions of partial differential equations, they must learn how to recognize them and deal with them. A point of emphasis in my course, which I hope that I convey also in the text, is teaching the students to become experimentalists. I explain that before one runs an experiment, one should know as much as possible about the problem. A complete problem usually includes the physical problem, the mathematical problem, the numerical scheme and the computer. (In this text, the physical problem is often slighted.) I then try to show how to run numerical experiments. As part of the training to be a numerical experimentalist, I include in the Prelude four nonlinear problems. I assume that the students do not generally know much about these problems initially. As we proceed in the text, I suggest that they try generalizations of some of our linear methods on these nonlinear problems. Of course, these methods are not always successful and in these cases I try to explain why we get the results that we get.

The implementation aspect of the text obviously includes a large amount of computing. Another aspect of computing included in the text is symbolic computing. When we introduce the concept of consistency, we show the calculations as being done on paper. However, after we have seen a few of these, we emphasize that a computer with a symbolic manipulator should be doing these computations. When we give algorithms for symbolic computations, we have tried to give it in a pseudo code that can be used by any of the symbolic manipulators. Another aspect of the new technologies that we use extensively is graphics. Of course, we provide plots of our numerical results and ask the students to provide plots of their results. We also use graphics for analysis. For example, for the analyses of dissipation and dispersion, where much of this has traditionally been done analytically (where one obtains only asymptotic results), we emphasize how easy it is to plot these results and interpret the dissipativity and dispersivity properties from the plots.

Though there is a strong emphasis in the text on implementing the schemes, there is also a strong emphasis on theory. Because of the audience, the theory is usually set in what might be called computational space

(where the computations are or might be done) and the convergence is done in $\ell_{2,\Delta x}$ spaces in terms of the energy norm. Though at times these spaces might not be as nice mathematically as some other spaces that might be used, it seems that working in spaces that mimic the computational space is easier for the students to grasp. Throughout the text, we emphasize the meaning of consistency, stability and convergence. In my classes I emphasize that it is dangerous for a person who is using difference methods not to understand what it means for a scheme to converge. In my class and in the text, I emphasize that we sometimes get necessary and sufficient conditions for convergence and sometimes get only necessary conditions (then we must learn to accept that we have only necessary conditions and proceed with caution and numerical experimentation). In the text, not only do we prove the Lax Theorem, but we return to the proof to see how to choose an initialization scheme for multilevel schemes and how we can change the definition of stability when we consider higher order partial differential equations. For several topics (specifically for many results in Chapters 8, 9 and 10) we do not include all of the theory (specifically not all of the proofs) but discuss and present the material in a theoretically logical order. When theorems are used without proof, references are included.

Lastly, it is hoped that the text will become a reference book for the students. In the preparation of the text, I have tried to include as many aspects of the numerical solution of partial differential equations as possible. I do not have time to include some of these topics in my course and might not want to include them even if I had time. I feel that these topics must be available to the students so that they have a reference point when they are confronted with them. One such topic is the derivation of numerical schemes. I personally do not have a preference on whether a given numerical scheme is derived mathematically or based on some physical principles. I feel that it is important for the student to know that they can be derived both ways and that both ways can lead to good schemes and bad schemes. In Chapter 1, we begin by first deriving the basic difference mathematically, and then show how the same difference scheme can be derived by using the integral form of the conservation law. We emphasize in this section that the errors using the latter approach are errors in numerical integration. This is a topic that I discuss and that I want the students to know is there and that it is a possible approach. It is also a topic that I do not develop fully in my class. Throughout the text, we return to this approach to show how it differs when we have two dimensional problems, hyperbolic problems, etc. Also, throughout the text we derive difference schemes purely mathematically (heuristically, by the method of undetermined coefficients or by other methods). It is hoped the readers will understand that if they have to derive their own schemes for a partial differential equation not previously considered, they will know where to find some tools that they can use.

Because of the length of the text, as was stated earlier, the material is

being given in two parts. The first part includes most of the basic material on time dependent equations including parabolic and hyperbolic problems, multi-dimensional problems, systems and dissipation and dispersion. The second part includes chapters on stability theory for initial–boundary value problems (the GKSO theory), numerical schemes for conservation laws, numerical solution of elliptic problems and an introduction to irregular regions and irregular grids. When I teach the course, I usually cover most of the first five chapters during the first semester. During the second semester I usually cover Chapters 6 and 7 (systems and dissipation and dispersion), Chapter 10 (elliptic equations) and selected topics from Chapters 8, 9 and 11. In other instances, I have covered Chapters 8 and 9 during the second semester, and on one occasion, I used a full semester to teach Chapter 9. Other people who have used the notes have covered parts of Chapters 1–7 and Chapter 10 in one semester. In either case, there seems to be sufficient material for at least two semesters of course work.

At the end of most of the chapters of the text and in the middle of several, we include sections which we refer to as "Computational Interludes." The original idea of these sections was to stop working on new methods, take a break from theory and compute for a while. These sections do include this aspect of the material, but as they developed, they also began to include more than just computational material. It is in these sections that we discuss results from previous homework problems. It is also in these sections that we suggest it is time for the students to try one of their new methods on one of the problems HW0.0.1–HW0.0.4 from the Prelude. There are also some topics included in these sections that did not find a home elsewhere. At times a more appropriate title for these sections might have been "etc.".

At this time I would like to acknowledge some people who have helped me with various aspects of this text. I thank Drs. Michael Kirby, Steve McKay, K. McArthur and K. Bowers for teaching parts of the text and providing me with feedback. I also thank Drs. Kirby, McArthur, Jay Bourland, Paul DuChateau and David Zachmann for many discussions about various aspects of the text. Finally, I thank the many students who over the years put up with the dreadfully slow convergence of this material from notes to text. Whatever the result, without their input the result would not be as good. And, finally, though all of the people mentioned above and others have tried to help me, there are surely still some typos and errors of thought (though, hopefully, many mistakes have been corrected for the Second Printing). Though I do so sadly, I take the blame for all of these mistakes. I would appreciate it if you would send any mistakes that you find to thomas@math.colostate.edu. Thank you.

J.W. Thomas

Contents

Contents of Part 1: Finite Difference Methods

8

Stability of Initial–Boundary–Value Schemes

8.1 Introduction

Since early in Chapter 1, we have been computing solutions to initial–boundary–value problems. In Chapter 2 we included some theory that could be used to prove convergence of schemes for solving initial–boundary–value problems. In Example 2.2.2 we used the definition of convergence to prove the convergence of the basic difference scheme for the heat equation with zero Dirichlet boundary conditions. For the same difference scheme, in Section 2.5.2 we noted that the consistency and stability analyses done earlier in the text along with the Lax Theorem for a bounded domain (Theorem 2.5.3) imply convergence. We also pointed out that we could directly apply the definitions of consistency and stability, and Theorem 2.5.3 to obtain convergence for a hyperbolic scheme.

As we started developing tools for proving convergence (via the Lax Theorem), we found that the methods for initial–boundary–value schemes based on Chapter 2 worked nicely when we had Dirichlet or Neumann boundary conditions and symmetric difference operators (Example 3.1.7, Example 3.1.9, Example 4.3.4, Section 4.4.3.3, etc.) but that they cannot be used when either the difference operator is not symmetric (Example 3.1.6, Example 3.1.8, and all of the schemes given in Chapter 5 and 6 for hyperbolic equations) or when the boundary condition makes the finite difference operator nonsymmetric (the example done in Section 3.2 with the mixed boundary condition). Thus, *at the moment we do not have sufficiently good methods for proving stability for initial–boundary–value schemes.*

An additional problem that we encountered in both Chapters 5 and 6 was how to choose a *numerical boundary condition* when one was needed. We saw in HW5.6.10, HW5.6.11 and HW6.4.2 that some numerical boundary conditions seem to work well, while others that appear similar cause instabilities. Whenever we were confronted with the problem of choosing numerical boundary conditions, we promised that we would address this problem in Chapter 8.

In this chapter we will introduce a method for discussing stability of difference schemes in the presence of boundaries that is due to Gustafsson, Kreiss, Sundström, [19] and Osher, ref. [53] and will be referred to as the **GKSO theory.** The theory is difficult and parts of the theory are beyond the scope of this book. We will try to place this theory in the correct setting, outline the results of the theory, and show how to apply this theory to prove stability of schemes for solving initial–boundary value problems. Additional results and insights to the GKSO theory can be found in refs. [33], [63], [65] and [18].

8.2 Stability

We begin by emphasizing that the reason we want stability of a scheme is so that it can be used used with consistency and some sort of Lax Theorem to prove convergence. As we shall see, to be able to use certain analytical tools, we will use a different norm from the one we used in the past. The fact that we must use a different norm should not seem odd to us. The principal reason that we used the $\ell_{2,\Delta x}$ space and norm for convergence, consistency and stability for most of our initial–value schemes in the past was so that we could use the discrete Fourier transform to prove stability. We emphasize that to apply the Lax Theorem, we must use the same norms for consistency and stability, and we get convergence with respect to that norm. Thus, to change norms for this study of stability technically requires that we also redo our consistency proofs with respect to the new norms. We obtained our norm consistency results by assuming that the solutions were sufficiently smooth. Likewise, the difficulties involved with the proofs of consistency with respect to our new norms will again be eliminated by smoothness assumptions (though we must now assume that our solutions are smoother than we assumed earlier). However, the reader should be aware that a part of the process of proving rigorously that a given difference scheme will numerically approximate the solution to a given initial-value or initial–boundary–value problem is to prove that the solutions are sufficiently smooth to apply the necessary consistency results. The proof of smoothness is based on the form of the partial differential equation and the smoothness assumptions made on the nonhomogeneous terms. We will not worry about proving smoothness results. We will assume the necessary

smoothness in the solutions to the problem that we are trying to solve.

The major difference with the norms that we will be using in this chapter is that we will now be measuring functions over the temporal domains as well as over the spatial domains. We will be using several different spatial domains ($\mathbb{R}$, $\mathbb{R}^+$ (positive reals), $(-\infty, 1)$, $(0,1)$ and more). We define the norm over the time domain $[0,\infty)$ that is analogous to the usual ℓ_2 norm as

$$\|\mathbf{u}\|_\alpha^2 = \sum_{n=0}^{\infty} e^{-2\alpha n}|u^n|^2 \tag{8.2.1}$$

where $\mathbf{u} = \{u^n\}_{n=0}^{\infty}$ and α is a free parameter. To define a space-time norm on $(0,1) \times [0,\infty)$, we consider the usual grid on $[0,1]$ defined by $x_k = k\Delta x,\ k = 0, \ldots, M$ and define

$$\|\mathbf{u}\|_{\alpha,(0,1)}^2 = \sum_{n=0}^{\infty} e^{-2\alpha n}\|\mathbf{u}^n\|_2^2 \tag{8.2.2}$$

where $\mathbf{u} = \{u_k^n\},\ n = 0, \ldots,\ k = 1, \ldots, M-1$, $\mathbf{u}^n = \{u_k^n\}_{k=1}^{M-1}$, $\|\cdot\|_2$ is the usual Euclidean norm and α is a free parameter. We should also realize that we need the energy norms associated with the norms defined in (8.2.1) and (8.2.2). The norms $\|\mathbf{u}\|_{\alpha,\Delta t}^2$ and $\|\mathbf{u}\|_{\alpha,\Delta t,(0,1),\Delta x}^2$ are defined by multiplying the right hand sides of (8.2.1) and (8.2.2) by Δt and $\Delta t \Delta x$, respectively. We will introduce other norms as we need them.

Remark 1: We note that to say that a vector $\mathbf{u} = \{u^n\}$ has a finite norm with respect to norm (8.2.1) implies that the vector $\{e^{-\alpha n}u^n\} \in \ell_2$ (with respect to the t or n variable). The same interpretation can be used for the other three norms defined above.

Remark 2: One of the new aspects of the two norms defined above is the $e^{-2\alpha n}$ term in the sum. This exponential term allows our functions to grow exponentially with respect to the n variable, and it is related to the $e^{\beta t}$ term in the definitions of stability given in Section 2.4. This is included since there are times when we want our solutions to be able to grow exponentially with respect to the t variable (especially when the partial differential equation contains a bv term).

Remark 3: In the norm given in (8.2.1) we indicate that the spatial measure is on a vector running from $k = 1$ to $k = M-1$. This is the vector that most commonly denotes the grid points inside of our region. There will be times that we will have to use a vector that also includes the boundary points.

8.2.1 Stability: An Easy Case

We have defined the above norms to help us give a definition of stability that is different from the one used earlier for initial–boundary–value problems.

We begin by considering a model equation of the form

$$v_t + av_x = F(x), \quad x \in (0,1), \quad t > 0 \tag{8.2.3}$$

$$v(1,t) = g(t), \quad t \geq 0 \tag{8.2.4}$$

$$v(x,0) = f(x), \quad x \in [0,1] \tag{8.2.5}$$

where a is chosen to be less than zero (which is why we provided a boundary condition at $x = 1$ and did not provide one at $x = 0$) along with the difference scheme

$$\begin{aligned} a_{-1\,-1}u_{k-1}^{n+1} + a_{0\,-1}u_k^{n+1} + a_{1\,-1}u_{k+1}^{n+1} \\ = a_{-1\,0}u_{k-1}^n + a_{0\,0}u_k^n + a_{1\,0}u_{k+1}^n + \Delta t G_k^n, \\ k = 1, \ldots, M-1 \end{aligned} \tag{8.2.6}$$

$$u_M^{n+1} = g^{n+1} \tag{8.2.7}$$

$$u_k^0 = f_k, \quad k = 0, \ldots, M. \tag{8.2.8}$$

It is clear from the form of the difference equation (8.2.6) that if $a_{-1\,-1}$ or $a_{-1\,0}$ is not equal to 0 (which for now we assume is true), we need a numerical boundary condition at $k = 0$. We include the following numerical boundary condition at $k = 0$ as a part of our model difference scheme.

$$u_0^{n+1} = c_1^0 u_1^{n+1} + c_2^0 u_2^{n+1} + c_1^1 u_1^n + c_2^1 u_2^n + g_0^n. \tag{8.2.9}$$

Remark: Difference equation (8.2.6) is not so general as to include all of our schemes; it does not include for instance the leapfrog scheme. However, it will include most of our favorite schemes, including the Lax-Wendroff scheme, (5.3.8); the Lax-Friedrichs scheme, Table 5.3.1; the BTCS implicit scheme, (5.4.6); and the Crank-Nicolson scheme, (5.4.10). Also, numerical boundary condition (8.2.9) is sufficiently general to include all of the conditions considered in HW5.6.10 and HW6.4.2.

In this section, we are interested in the stability or instability due to the boundary conditions (the real boundary conditions (8.2.7) and the numerical boundary conditions (8.2.9)). The initial condition f and the forcing function F just get in the way. If we take the attitude that solving the Cauchy problem (the initial–value problem with initial condition f and forcing function F) is trivial, there is no problem in assuming that difference equation (8.2.6) and initial condition (8.2.8) are homogeneous (otherwise, we would first solve the Cauchy problem (initial–value problem) and subtract its solution). There are several times that it will be convenient to have nonhomogeneous initial conditions, and at those times, we will allow them, but *generally, throughout this section we will assume that* $f_k = 0$ *and* $G_k^n = 0$ *for all* n *and* k.

We are now ready to define stability for difference scheme (8.2.6)–(8.2.9). We use the definition given in [33], page 76.

Definition 8.2.1 *Difference scheme (8.2.6)–(8.2.9) is stable if there exist positive constants* Δx_0, Δt_0, K *and a nonnegative constant* α_0 *such that for* $\alpha > \alpha_0$

$$(\alpha - \alpha_0)\|\{u_k^n\}\|^2_{\alpha\Delta t,\Delta t,(0,1),\Delta x} \leq K^2\left[\|\{g_0^n\}\|^2_{\alpha\Delta t,\Delta t} + \|\{g^n\}\|^2_{\alpha\Delta t,\Delta t}\right] \tag{8.2.10}$$

for all g_0^n *and* g *with* $\|\{g_0^n\}\|^2_{\alpha\Delta t,\Delta t}$ *and* $\|\{g^n\}\|^2_{\alpha\Delta t,\Delta t}$ *finite and for* $0 < \Delta x \leq \Delta x_0$ *and* $0 < \Delta t \leq \Delta t_0$.

Remark 1: We note specifically with the above definition that the parameter with the norm is $\alpha\Delta t$ instead of just α. This is done to allow for the correct growth with $n\Delta t$ of the functions with respect to the t variable.

Remark 2: We note also that the $\alpha - \alpha_0$ constant in front of the left hand term is different from any of our previous definitions of stability. This is necessary due to the inclusion of the t variable. In ref. [19], page 655, it is proved that if $\alpha_0 = 0$, then the solution will not have exponential growth with respect to t.

Remark 3: One of the very important differences between the above definition and the definitions of stability given in Chapter 2 is that in Definition 8.2.1 above, we sum over the time variable as well as the spatial variable. As we shall see later, the use of this norm allows us to use the discrete Laplace transform in our analysis. The role of the discrete Laplace transform in the GKSO theory for proving stability of initial–boundary–value schemes is analogous to the role of the discrete Fourier transform for proving stability of initial–value schemes. The discrete Laplace transform is a major tool in the stability analysis of initial–boundary–value problems.

Remark 4: There are other definitions available for the stability of difference schemes for initial–boundary–value problems and they are all difficult. In fact, there are two others included in ref. [19]. We chose the above definition because it seems to be a definition that allows us to use most of the available results that we wish to use. If there are times that we must consider a different definition in order to allow us to use a certain result, we will just reference the appropriate definition and use the result. At no time do we actively work in depth with any of the definitions. Our approach in this chapter is to state the results that we want, reference these results and learn to use them. The definition of stability is given largely for completeness and so that we have a somewhat better understanding of the setting in which we are working.

We now have the definition of stability for initial–boundary–value schemes that we wish to use. We should be ready to begin analysis of schemes based on Definition 8.2.1. Instead, we take a detour that will make our stability analyses easier. We state the following theorem from ref. [19], page 660.

Theorem 8.2.2 *Difference scheme (8.2.6)–(8.2.9) is stable if the corresponding initial–value problem and left and right quarter plane problems are stable.*

The idea behind Theorem 8.2.2 is that the basic difference scheme and each of the boundary conditions (8.2.7) and (8.2.9) can be handled separately, and they are separated into the corresponding initial–value problem and the two quarter plane problems that are relatively nice to handle. The assumption that the corresponding initial–value scheme must be stable is not a big assumption, since we saw in Section 3.1.2 that a necessary condition for an initial–boundary–value problem scheme to be stable is that the associated initial–value scheme must be stable. Of course, before we proceed we must describe the quarter plane problems.

We consider a grid on $\mathbb{R}^+ = \{x \in \mathbb{R} : x \geq 0\}$, $x_k = k\Delta x$, $k = 0, 1, \ldots$. On this grid we consider difference scheme

$$\begin{aligned} a_{-1\,-1}u_{k-1}^{n+1} + a_{0\,-1}u_k^{n+1} + a_{1\,-1}u_{k+1}^{n+1} \\ = a_{-1\,0}u_{k-1}^n + a_{0\,0}u_k^n + a_{1\,0}u_{k+1}^n + \Delta t G_k^n, \quad k = 1, 2, \ldots \end{aligned} \tag{8.2.11}$$

along with the initial condition

$$u_k^0 = f_k, \quad k = 0, 1, \ldots \tag{8.2.12}$$

and numerical boundary condition

$$u_0^{n+1} = c_1^0 u_1^{n+1} + c_2^0 u_2^{n+1} + c_1^1 u_1^n + c_2^1 u_2^n + g_0^n. \tag{8.2.13}$$

Problem (8.2.11)–(8.2.13) is referred to as the **right quarter plane problem** associated with problem (8.2.6)–(8.2.9).

If we instead consider the grid on $(-\infty, 1)$, $x_k = k\Delta x$, $k = M, M-1, \ldots$, we refer to difference equation (8.2.11) for $k = M-1, M-2, \ldots$, initial condition (8.2.12) for $k = M, M-1, \ldots$, and boundary condition

$$u_M^{n+1} = g^{n+1} \tag{8.2.14}$$

as the **left quarter plane problem** associated with problem (8.2.6)–(8.2.9).

As we stated before, since we are interested in the stability with respect to the boundary conditions, we assume that f_k and G_k^n are zero for all n and k. The norms analogous to the norm defined in (8.2.2) can be defined on these quarter plane grids as follows:

$$\|\{u_k^n\}\|_{\alpha,\Lambda}^2 = \sum_{n=0}^{\infty} e^{-2\alpha n} \|\mathbf{u}^n\|_2^2 \tag{8.2.15}$$

where the norm $\|\cdot\|_2$ is the usual infinite ℓ_2 norm summing over the appropriate grid depending on whether $\Lambda = \mathbb{R}^+$ or $\Lambda = (-\infty, 1)$. As before, there

are associated energy norms, where we obtain the norms $\|\{u_k^n\}\|^2_{\alpha,\Delta t,\mathbf{R}^+,\Delta x}$ and $\|\{u_k^n\}\|^2_{\alpha,\Delta t,(-\infty,1),\Delta x}$ by multiplying the right side of (8.2.15) by $\Delta t \Delta x$. We then define stability of the right quarter plane problem as follows.

Definition 8.2.3 *The right quarter plane problem is stable if there exist positive constants Δx_0, Δt_0, K and a nonnegative constant α_0 such that for $\alpha > \alpha_0$*

$$(\alpha - \alpha_0)\|\{u_k^n\}\|^2_{\alpha\Delta t,\Delta t,\mathbf{R}^+,\Delta x} \leq K^2\|\{g_0^n\}\|^2_{\alpha\Delta t,\Delta t} \tag{8.2.16}$$

for all g_0^n with $\|\{g_0^n\}\|^2_{\alpha\Delta t,\Delta t}$ finite and for $0 < \Delta x \leq \Delta x_0$ and $0 < \Delta t \leq \Delta t_0$.

Remark: Again, Definition 8.2.3 includes the constant α_0 to allow for exponential growth of the solutions. In ref. [19], page 657, the following result (analogous to Proposition 3.1.8, Part 1) is proved.

Proposition 8.2.4 *If a difference scheme is stable with respect to Definition 8.2.3, then the scheme obtained by perturbing either the difference equation and/or the boundary condition by a term of order Δt will also be stable.*

As was the case in Chapter 3, this result allows us to eliminate the *bv* terms from our equations, and we no longer have to consider solutions that grow exponentially. Thus, for the rest of this chapter,

- *we assume that $\alpha_0 = 0$, i.e., we do not allow our solutions to grow exponentially.*

Of course, the definition of stability for a difference scheme for the left quarter plane problem is the same, except that the norm on the left side is taken over the grid on $(-\infty, 1]$ and the g_0 on the right side is replaced by g. We will consider the stability for the right quarter plane problem first, since this is more interesting than the left quarter plane problem. *Remember, boundary condition (8.2.9) is a numerical boundary condition.*

8.2.1.1 Stability of the Right Quarter Plane Problem

To be specific, we consider difference equation (8.2.11), initial condition $u_k^0 = 0$, $k = 0, 1, \ldots$ and numerical boundary condition (8.2.13). We want to find conditions under which this scheme will be stable based on Definition 8.2.3. Before we proceed, we include the following three assumptions that are necessary for the results contained in this section.

Assump 8.1 The associated initial–value scheme is stable.

Assump 8.2 Equations (8.2.11), (8.2.13) can be solved in $\ell_{2,\Delta x}$ for u_k^{n+1} for any given u_k^n in $\ell_{2,\Delta x}$.

Assump 8.3 The difference scheme is either dissipative or nondissipative.

As we stated earlier in our discussion of Theorem 8.2.2, Assump 8.1 is not a very restrictive assumption since we already know that it is a necessary condition for stability. Likewise, any difference scheme that does not satisfy Assump 8.2 will not be very useful. And finally, as can be seen from the results given throughout Chapter 7, all of the schemes that we have considered are either dissipative or nondissipative, i.e. all of the schemes that we have considered satisfy Assump 8.3.

The approach that we will use is very similar to the approach used for initial–value schemes in Chapter 3. In Chapter 3, we used the discrete Fourier transform to transform out the spatial differences (spatial derivatives) and considered the scheme in transform space as a difference equation in n. In this case we will use the discrete Laplace transform to transform out the temporal differences (time derivatives) and consider the scheme in transform space as a difference scheme in k. Because the difference equations in k that we must handle here are more difficult than those in n that we considered in Chapter 3, the approach here is not as nice and clean as it was in Chapter 3. However, the approach does work and is one of the few approaches available for analyzing the stability of difference schemes for initial–boundary–value problems.

Analogous to our definitions of the discrete Fourier transform made in Section 3.1.1, we define the **discrete Laplace transform** as follows.

Definition 8.2.5 *The discrete Laplace transform of* $\mathbf{u} = \{u^n\}$ *is the function* $\tilde{u} = \mathcal{L}(\{u^n\})$ *defined by*

$$\tilde{u}(s) = \frac{1}{\sqrt{2\pi}} \sum_{n=0}^{\infty} e^{-sn} u^n \tag{8.2.17}$$

where $s = \alpha + i\tau$, $\alpha > 0$ *and* $\tau \in [-\pi, \pi]$.

To norm used to "measure" Laplace transform functions is given by

$$\|\tilde{u}\|_{\alpha,2}^2 = \int_{-\pi}^{\pi} |\tilde{u}(s)|^2 \, d\tau. \tag{8.2.18}$$

Note that the integral in (8.2.18) is taken only with respect to τ, and the subscript in the notation includes an $\alpha 2$ to remind us of the dependence on α. We can then mimic the proofs of Propositions 3.1.2 and 3.1.3 to obtain the following results.

Proposition 8.2.6 *If* $\tilde{u}$ *is the discrete Laplace transform of* $\mathbf{u}$, *then*

$$u^n = \frac{1}{\sqrt{2\pi}} \int_{-\pi}^{\pi} e^{ns} \tilde{u}(s) \, d\tau \tag{8.2.19}$$

where $s = \alpha + i\tau$.

Proposition 8.2.7 *If $\tilde{u}$ is the discrete Laplace transform of* $\mathbf{u}$, *then*

$$\|\tilde{u}\|_{\alpha,2}^2 = \|\mathbf{u}\|_{\alpha}^2 \tag{8.2.20}$$

where $s = \alpha + i\tau$.

We note that we can formally write

$$\begin{aligned}
\|\{u_k^n\}\|_{\alpha\Delta t,\Delta t,\mathbb{R}^+,\Delta x}^2 &= \sum_{n=0}^{\infty} e^{-2n\alpha\Delta t} \sum_{k=1}^{\infty} |u_k^n|^2 \Delta x \Delta t \quad \text{(definition of norm)} \\
&= \Delta t \Delta x \sum_{k=1}^{\infty} \sum_{n=0}^{\infty} e^{-2n\alpha\Delta t} |u_k^n|^2 \\
&= \Delta t \Delta x \sum_{k=1}^{\infty} \|\{u_k^n\}_{n=0}^{\infty}\|_{\alpha\Delta t}^2 \quad \text{(norm (8.2.1))} \\
&= \Delta t \Delta x \sum_{k=1}^{\infty} \int_{-\pi}^{\pi} |\tilde{u}_k(s)|^2 \, d\tau \quad \text{(Proposition 8.2.7)} \\
&= \Delta t \int_{-\pi}^{\pi} \Delta x \sum_{k=1}^{\infty} |\tilde{u}_k(s)|^2 \, d\tau \\
&= \Delta t \int_{-\pi}^{\pi} \|\{\tilde{u}_k(s)\}\|_{2,\Delta x}^2 \, d\tau \\
&= \Delta t \left\| \|\{\tilde{u}_k\}\|_{2,\Delta x}^2 \right\|_{\alpha\Delta t,2}^2 .
\end{aligned}$$

Then, because $\|\{g_0^n\}\|_{\alpha\Delta t,\Delta t} = \Delta t \|\tilde{g}_0\|_{\alpha\Delta t,2}$, inequality (8.2.16) can be rewritten as

$$\alpha \left\| \|\{\tilde{u}_k\}\|_{2,\Delta x}^2 \right\|_{\alpha\Delta t,2}^2 \leq K^2 \|\tilde{g}_0\|_{\alpha\Delta t,2}^2 \tag{8.2.21}$$

(remember that $\alpha_0 = 0$). Thus we see that if we can find a pointwise inequality

$$\alpha \|\{\tilde{u}_k(s)\}\|_{2,\Delta x}^2 \leq K^2 |\tilde{g}_0(s)|^2 \tag{8.2.22}$$

for all $\tau \in [-\pi, \pi]$ and $\alpha > 0$ ($s = \alpha + i\tau$) where both functions are L_2 integrable, we satisfy inequality (8.2.21). We will generally prove stability by proving that our schemes satisfy inequality (8.2.22) in transform space.

We proceed to work with the discrete Laplace transform of equations (8.2.11), (8.2.13). Taking the discrete Laplace transform of the left hand

side of equation (8.2.11), we get

$$\begin{aligned}
&\mathcal{L}\left(a_{-1\,-1}u_{k-1}^{n+1} + a_{0\,-1}u_k^{n+1} + a_{1\,-1}u_{k+1}^{n+1}\right) \\
&= \frac{1}{\sqrt{2\pi}} \sum_{n=0}^{\infty} e^{-sn} \left(a_{-1\,-1}u_{k-1}^{n+1} + a_{0\,-1}u_k^{n+1} + a_{1\,-1}u_{k+1}^{n+1}\right) \\
&= \frac{1}{\sqrt{2\pi}} \sum_{m=1}^{\infty} e^{-s(m-1)} \left(a_{-1\,-1}u_{k-1}^{m} + a_{0\,-1}u_k^{m} + a_{1\,-1}u_{k+1}^{m}\right) \\
&\qquad\qquad \text{(setting } n = m-1\text{)} \\
&= e^{s} \frac{1}{\sqrt{2\pi}} \sum_{m=0}^{\infty} e^{-sm} \left(a_{-1\,-1}u_{k-1}^{m} + a_{0\,-1}u_k^{m} + a_{1\,-1}u_{k+1}^{m}\right) \quad (8.2.23) \\
&\qquad\qquad \text{(since } u_{k-1}^0 = u_k^0 = u_{k+1}^0 = 0\text{)} \\
&= e^{s} \left[a_{-1\,-1}\mathcal{L}(\{u_{k-1}^n\}) + a_{0\,-1}\mathcal{L}(\{u_k^n\}) + a_{1\,-1}\mathcal{L}(\{u_{k+1}^n\})\right].
\end{aligned}$$

Then, since the discrete Laplace transform of the right hand side of equation (8.2.6) can be written as

$$\begin{aligned}
&\mathcal{L}\left(a_{-1\,0}u_{k-1}^n + a_{0\,0}u_k^n + a_{1\,0}u_{k+1}^n\right) \\
&\qquad = a_{-1\,0}\mathcal{L}(\{u_{k-1}^n\}) + a_{0\,0}\mathcal{L}(\{u_k^n\}) + a_{1\,0}\mathcal{L}(\{u_{k+1}^n\}),
\end{aligned}$$

the discrete Laplace transform of equation (8.2.6) is given by

$$z\left(a_{-1\,-1}\tilde{u}_{k-1} + a_{0\,-1}\tilde{u}_k + a_{1\,-1}\tilde{u}_{k+1}\right) = a_{-1\,0}\tilde{u}_{k-1} + a_{0\,0}\tilde{u}_k + a_{1\,0}\tilde{u}_{k+1} \tag{8.2.24}$$

where $z = e^s$ and $s = \alpha\Delta t + i\tau$. We refer to equation (8.2.24) as the **resolvent equation.**

We also see that boundary condition (8.2.13) transforms into

$$z\tilde{u}_0 = z(c_1^0\tilde{u}_1 + c_2^0\tilde{u}_2) + c_1^1\tilde{u}_1 + c_2^1\tilde{u}_2 + \tilde{g}_0. \tag{8.2.25}$$

If we consider the homogeneous version of equation (8.2.25),

$$z\left(\tilde{u}_0 - c_1^0\tilde{u}_1 - c_2^0\tilde{u}_2\right) = c_1^1\tilde{u}_1 + c_2^1\tilde{u}_2, \tag{8.2.26}$$

we can view equations (8.2.24), (8.2.26) as an eigenvalue problem where z is the eigenvalue and $\tilde{\mathbf{u}} = [\tilde{u}_0\ \tilde{u}_1\ \cdots]^T$ is the eigenvector. More specifically, we make the following definition.

Definition 8.2.8 *The complex number z, $|z| > 1$, is an* **eigenvalue** *of equations (8.2.24), (8.2.26) if*
(1) *there exists a vector $\tilde{\mathbf{u}} = [\tilde{u}_0\ \tilde{u}_1\ \cdots]^T$ such that $(z, \tilde{\mathbf{u}})$ satisfies equations (8.2.24), (8.2.26), and*
(2) $\|\tilde{\mathbf{u}}\|_2 < \infty$.

We are then able to state a necessary condition for stability, usually referred to as the **Ryabenkii-Godunov condition.**

Proposition 8.2.9 *If equations (8.2.24), (8.2.26) have an eigenvalue z with $|z| > 1$, then difference scheme (8.2.11)–(8.2.13) is unstable.*

Proof: Assume that $\tilde{u}$ is a nontrivial solution associated with z such that $|z| > 1$. Then the function $u_k^n = z^n \tilde{u}_k$ a solution to the homogeneous version of equations (8.2.11)–(8.2.13). If U_k^n is some solution of equations (8.2.11)–(8.2.13), then $U_k^n + u_k^n$ will also be a solution. This solution will grow exponentially so it will not satisfy the definition of stability.

Before we can really apply Proposition 8.2.9, we must be able to find the eigenvalues and eigenvectors of equations (8.2.24), (8.2.26). To show that we do not get an instability due to Proposition 8.2.9, we must show that equations (8.2.24), (8.2.26) have no eigenvalues. The most direct way to do this is to solve difference equations (8.2.24) and (8.2.26). The standard approach is to look for a solution of the form $\tilde{u}_k = \kappa^k$. (This approach is a standard approach for analytically solving difference equations. Note the similarity with this approach to that used to solve constant coefficient ordinary differential equations.) Substituting $\tilde{u}_k = \kappa^k$ into equation (8.2.24) gives

$$z\left(a_{-1\,-1}\kappa^{k-1} + a_{0\,-1}\kappa^k + a_{1\,-1}\kappa^{k+1}\right) = a_{-1\,0}\kappa^{k-1} + a_{0\,0}\kappa^k + a_{1\,0}\kappa^{k+1} \tag{8.2.27}$$

or

$$(a_{1\,0} - z a_{1\,-1})\kappa^2 + (a_{0\,0} - z a_{0\,-1})\kappa + a_{-1\,0} - z a_{-1\,-1} = 0 \tag{8.2.28}$$

which is called the **characteristic equation** associated with difference equation (8.2.24). Of course, equation (8.2.28) has two roots κ_1, κ_2 (which depend continuously on z). The general solution to equation (8.2.24) can be written as $\tilde{u}_k = \phi_1 \kappa_1^k + \phi_2 \kappa_2^k$ (unless $\kappa_1 = \kappa_2$, in which case the general solution is of the form $\tilde{u}_k = \phi_1 \kappa_1^k + \phi_2 k \kappa_1^k$). We use the fact that $\tilde{u}_k$ must satisfy condition $\|\{\tilde{u}_k\}\|_{2,\Delta x} < \infty$ and boundary condition (8.2.26) to determine the constants ϕ_1 and ϕ_2.

The situation is really much nicer than one might expect. To demonstrate this, we state and prove the following result.

Proposition 8.2.10 *For $|z| > 1$ characteristic equation (8.2.27) (or (8.2.28)) has no solution κ with $|\kappa| = 1$. For $|z| > 1$, one root of equation (8.2.27) will satisfy $|\kappa_1| < 1$, and one root will satisfy $|\kappa_2| > 1$.*

Proof: Suppose $|\kappa| = 1$, i.e., $\kappa = e^{i\theta}$ for some $\theta \in [-\pi, \pi]$. Then equation (8.2.27) becomes

$$\begin{aligned} z\big(a_{-1\,-1}e^{i(k-1)\theta} + a_{0\,-1}e^{ik\theta} + a_{1\,-1}e^{i(k+1)\theta}\big) \\ = a_{-1\,0}e^{i(k-1)\theta} + a_{0\,0}e^{ik\theta} + a_{1\,0}e^{i(k+1)\theta}, \end{aligned}$$

or

$$z = \frac{a_{-1\,0}e^{-i\theta} + a_{0\,0} + a_{1\,0}e^{i\theta}}{a_{-1\,-1}e^{-i\theta} + a_{0\,-1} + a_{1\,-1}e^{i\theta}}.$$

By Assump 8.1, the symbol of difference equation (8.2.6)

$$\rho(\xi) = \frac{a_{-1\,0}e^{-i\xi} + a_{0\,0} + a_{1\,0}e^{i\xi}}{a_{-1\,-1}e^{-i\xi} + a_{0\,-1} + a_{1\,-1}e^{i\xi}}$$

satisfies $|\rho(\xi)| \leq 1$ for all $\xi \in [-\pi, \pi]$. Since $z = \rho(\theta)$, this contradicts the fact that $|z| > 1$.

Since the roots κ are continuous functions of z, for $|z| > 1$, a given root is always outside the unit circle or always inside the unit circle. If equation (8.2.28) has two roots less than one in magnitude, then to solve difference equation (8.2.11), (8.2.13), we must determine both ϕ_1 and ϕ_2. However, since we have only one boundary condition with which to work, boundary condition (8.2.25), it is impossible to determine both ϕ_1 and ϕ_2. Hence, Assump 8.2 cannot be satisfied. Therefore, equation (8.2.28), and hence equation (8.2.27), will have only one root κ_1 satisfying $|\kappa_1| < 1$. Obviously, the other root of equation (8.2.27), κ_2, must satisfy $|\kappa_2| > 1$ for $|z| > 1$.

Remark: Whenever we include both roots in our discussion, we will try to let κ_1 denote the root with $|\kappa_1| < 1$ and κ_2 denote the root with $|\kappa_2| > 1$.

Since we are interested only in solutions of equations (8.2.24), (8.2.26) that satisfy $\|\{\tilde{u}_k\}\|_{2,\Delta x} < \infty$, *we set* $\phi_2 = 0$ *in our general solution and consider solutions of the form* $\tilde{u}_k = \phi_1 \kappa_1^k$. We are now ready to illustrate how we apply Proposition 8.2.9.

Example 8.2.1 Show that the Lax-Friedrichs scheme,

$$u_k^{n+1} = \frac{1}{2}\left(u_{k-1}^n + u_{k+1}^n\right) - \frac{R}{2}\left(u_{k+1}^n - u_{k-1}^n\right), \quad k = 1, 2, \ldots, \tag{8.2.29}$$

along with initial condition $u_k^0 = 0$, $k = 0, 1, \ldots$ and boundary condition

$$u_0^{n+1} = -u_1^{n+1} + 2u_2^{n+1} \tag{8.2.30}$$

is an unstable difference scheme for $-\frac{1}{3} < R < 0$ (recall that $a < 0$).

Solution: We first note that difference scheme (8.2.29)–(8.2.30) is a special case of the right quarter plane problem described in (8.2.11)–(8.2.13). We also note that in HW5.3.2 we showed that $|R| \leq 1$ is a necessary and sufficient condition for the Lax-Friedrichs scheme to be stable for the Cauchy problem. Hence, Assump 8.1 is satisfied if we assume that $|R| \leq 1$. Since $a < 0$, we see that $0 \leq 1 + R \leq 1$ and $1 \leq 1 - R \leq 2$.

If we take the discrete Laplace transform of difference equations (8.2.29) and (8.2.30), we obtain the following eigenvalue problem analogous to equations (8.2.24), (8.2.26).

$$z\tilde{u}_k = \frac{1}{2}(1+R)\tilde{u}_{k-1} + \frac{1}{2}(1-R)\tilde{u}_{k+1}, \quad k = 1, 2, \ldots \tag{8.2.31}$$

$$\tilde{u}_0 + \tilde{u}_1 - 2\tilde{u}_2 = 0 \tag{8.2.32}$$

To solve difference equation (8.2.31)–(8.2.32), we look for a solution of the form $\tilde{u}_k = \kappa^k$ and obtain the characteristic equation

$$\frac{1}{2}(1-R)\kappa^2 - z\kappa + \frac{1}{2}(1+R) = 0. \tag{8.2.33}$$

Substituting $\tilde{u}_k = \phi_1 \kappa_1^k$ into boundary condition (8.2.32), we get $\kappa_1 = 1$ and $\kappa_1 = -\frac{1}{2}$. If we use κ_1 to try to generate an eigenvector, the vector will not satisfy $\|\tilde{u}\|_2 < \infty$. So clearly, the only way that we might get a nontrivial solution to the eigenvalue problem (8.2.31), (8.2.32) ((8.2.24), (8.2.26)) is to use $\kappa_1 = -\frac{1}{2}$. Inserting $\kappa_1 = -\frac{1}{2}$ into the characteristic equation (8.2.33), we see that $z = -\frac{3}{4}R - \frac{5}{4}$. For $|R| < \frac{1}{3}$ (really $R > -\frac{1}{3}$),

$$|z| = \frac{5}{4} - \frac{3}{4}|R| > 1.$$

Thus we have found a solution to equations (8.2.31), (8.2.32), $\tilde{u}_k = (-\frac{1}{2})^k$ for $|z| > 1$, so by Proposition 8.2.9 the difference scheme is unstable.

Example 8.2.2 Analyze the stability of the difference scheme consisting of the Lax-Wendroff difference equation

$$u_k^{n+1} = \frac{1}{2}(R^2 + R)u_{k-1}^n + (1 - R^2)u_k^n + \frac{1}{2}(R^2 - R)u_{k+1}^n \tag{8.2.34}$$

along with initial condition $u_k^0 = 0$, $k = 0, 1, \ldots$ and boundary condition

$$u_0^{n+1} = u_1^{n+1} + u_0^n - u_1^n. \tag{8.2.35}$$

Solution: If we proceed as we did in Example 8.2.1 and take the discrete Laplace transform of equations (8.2.34) and (8.2.35), we obtain the eigenvalue problem

$$z\tilde{u}_k = \frac{1}{2}(R^2 + R)\tilde{u}_{k-1} + (1 - R^2)\tilde{u}_k + \frac{1}{2}(R^2 - R)\tilde{u}_{k+1} \tag{8.2.36}$$

$$z(\tilde{u}_0 - \tilde{u}_1) = \tilde{u}_0 - \tilde{u}_1 \tag{8.2.37}$$

analogous to equations (8.2.24), (8.2.26). If we look for solutions of the form $\tilde{u}_k = \phi_1 \kappa^k$, equation (8.2.37) implies that

$$(z - 1)(\kappa - 1) = 0$$

or $z = 1$ or $\kappa = 1$. Thus, immediately we see that we cannot use the Ryabenkii-Godunov condition to show that the difference scheme is unstable (because if $\kappa = 1$, we know that $|z| \not> 1$, and obviously, if $z = 1$, we know that $|z| \not> 1$). However, *this does not imply that the scheme is stable.*

Remark: In HW5.6.10 we tried to solve the initial–boundary–value problem

$$\begin{aligned} v_t - 2v_x &= 0, \quad x \in (0,1),\ t > 0 \\ v(x,0) &= 1 + \sin 2\pi x, \quad x \in [0,1] \\ v(1,t) &= 1.0, \quad t \geq 0 \end{aligned} \tag{8.2.38}$$

using four different numerical boundary conditions at $x = 0$. One approach now might be to return to the code developed for HW5.6.10, and try to solve the problem using numerical boundary condition (8.2.35). An experiment such as this one can be very useful in that it might show us that the scheme is unstable or that the scheme might be stable. See HW8.2.1.

We see that not only must we develop methods for proving that a scheme is stable, but we also may need stronger results for proving that a scheme is unstable. At this time we return to the problem of solving equations

(8.2.24)–(8.2.25). Let κ_1 denote the root of characteristic equation (8.2.28) that satisfies $|\kappa_1| < 1$ (for $|z| > 1$). We substitute $\tilde{u}_k = \phi_1 \kappa_1^k$ into boundary condition (8.2.25) and get

$$z\phi_1 = z(c_1^0 \phi_1 \kappa_1 + c_2^0 \phi_1 \kappa_1^2) + c_1^1 \phi_1 \kappa_1 + c_2^1 \phi_1 \kappa_1^2 + \tilde{g}_0. \tag{8.2.39}$$

The obvious approach to proving the stability of difference scheme (8.2.11)–(8.2.13) is to solve equation (8.2.39) for ϕ_1, and use $\tilde{u}_k = \phi_1 \kappa_1^k$ to see whether we can verify that inequality (8.2.21) or (8.2.22) is satisfied (or perform the inverse transform and show that inequality (8.2.16) holds).

If we define $D(z)$ by

$$D(z) = z(1 - c_1^0 \kappa_1 - c_2^0 \kappa_1^2) - (c_1^1 \kappa_1 + c_2^1 \kappa_1^2),$$

then the equation we must solve (just a rewrite of equation (8.2.39)) can be written as

$$D(z)\phi_1 = \tilde{g}_0. \tag{8.2.40}$$

Of course, we must decide whether it is possible to solve equation (8.2.40) for $|z| > 1$. We should realize that in this setting, the Ryabenkii-Godunov condition (Proposition 8.2.9) can be written as *if there exists a* z, $|z| > 1$, *for which* $D(z) = 0$; *then difference scheme (8.2.11)–(8.2.13) is unstable* (when z is such that $|z| > 1$ and $D(z) = 0$, then $u_k^n = z^n \phi_1 \kappa_1^k$ will be an unstable solution to the homogeneous version of difference scheme (8.2.11)–(8.2.13)).

When $D(z) \neq 0$ for $|z| > 1$, it is not enough to be able just to solve equation (8.2.40). This solution $\phi_1 = \tilde{g}_0 / D(z)$ must be used in the definition of $\tilde{u} = \phi_1 \kappa_1^k$. We then must show that $\tilde{u}$ satisfies inequality (8.2.22) (or one of the other stability inequalities). The obvious way to satisfy inequality (8.2.22) for all $|z| > 1$ is to find a constant K, independent of z, such that $|1/D(z)| \leq K$ ($1/D(z)$ is uniformly bounded for $|z| > 1$). If $D(z)$ is nonzero for all $|z| > 1$, but approaches zero on the boundary, then $1/D(z)$ will not, in general, be uniformly bounded. One way to ensure that $1/D(z)$ is uniformly bounded near $|z| = 1$ is to *require that* $D(z) \neq 0$ *for all* $|z| \geq 1$. When this condition is satisfied, it is not hard to see that $1/D(z)$ is bounded independent of z and that both inequalities (8.2.21) and (8.2.22) are satisfied.

Proposition 8.2.11 *If* $D(z) \neq 0$ *for all* z, $|z| \geq 1$, *then difference scheme (8.2.11)–(8.2.13) is stable.*

If we return to Example 8.2.2, we see that our problem is that

$$D(z) = (z - 1)(\kappa_1 - 1)$$

and $D(1) = 0$. The fact that $D(1) = 0$ eliminates the use of Proposition 8.2.11 to prove the stability of the scheme. We also emphasize that $D(1) = 0$

does not imply that the scheme is unstable. We must determine how we can determine whether this scheme is stable or unstable. The way we shall proceed at this time is to introduce the results given in ref. [19]. We should realize that because we have limited ourselves to "an easy case," we will use only part of the generality of the results given in ref. [19]. We will return to that result several times throughout this chapter when we wish to extend our results to more difficult cases. In order to account for the zeros of D on $|z| = 1$, we make the following definition.

Definition 8.2.12 *The complex number z is a* **generalized eigenvalue** *of equations (8.2.24), (8.2.26) if*
(1) *there exists a vector $\tilde{\mathbf{u}} = [\tilde{u}_0\ \tilde{u}_1\ \cdots]^T$ such that $(z, \tilde{\mathbf{u}})$ satisfies equations (8.2.24), (8.2.26),*
(2) *$|z| = 1$, and*
(3) *$\tilde{u}_k$ satisfies*

$$\tilde{u}_k(z) = \lim_{w \to z, |w|>1} \tilde{u}_k(w),$$

where $(w, \tilde{\mathbf{u}}(w))$ is a solution to equation (8.2.24).

The result from ref. [19], page 660, is given by the following proposition.

Proposition 8.2.13 *Difference scheme (8.2.11)–(8.2.13) is stable if and only if eigenvalue problem (8.2.24), (8.2.26) has no eigenvalues and no generalized eigenvalues.*

To illustrate the above proposition, we include a series of examples applying Proposition 8.2.13. We begin with the difference scheme considered in Example 8.2.2.

Example 8.2.3 Complete the analysis of the stability of difference scheme (8.2.34)–(8.2.35).

Solution: If we return to Example 8.2.2 and substitute $\tilde{u}_k = \kappa^k$ into the resolvent equation, we obtain the following characteristic equation.

$$\kappa^2 + \frac{2(1 - R^2 - z)}{R^2 - R}\kappa + \frac{R^2 + R}{R^2 - R} = 0. \tag{8.2.41}$$

Substitution of $\tilde{u}_k = \kappa^k$ into the transformed boundary condition (8.2.37) gives us that $z = 1$ or $\kappa = 1$. The argument used in Example 8.2.2 shows that problem (8.2.36), (8.2.37) has no eigenvalues.

If we substitute $\kappa = 1$ in equation (8.2.41), we see that $z = 1$. To get more information, we notice that if we substitute $z = 1$ into equation (8.2.41), we get

$$\kappa = \frac{R^2 \pm |R|}{R^2 - R} = \begin{cases} \frac{R^2+|R|}{R^2-R} = 1 \\ \frac{R^2-|R|}{R^2-R} = \frac{R^2+R}{R^2-R}. \end{cases}$$

Since $z = 1$ is associated with $\kappa_1 = (R^2 + R)/(R^2 - R)$ and

$$\left|\frac{R^2 + R}{R^2 - R}\right| < 1$$

(for $-1 \leq R < 0$), it should be clear that for a sequence $\{z_j\}$ such that $|z_j| > 1$, z_k is near $z = 1$, and $z_j \to 1$, one of the two κ values, κ_j, will be very near κ_1 and satisfy

$|\kappa_j| < 1$. The eigenvectors $\mathbf{u}_j$ associated with κ_j will converge to the vector associated with $z = 1$ and κ_1. Hence, we see that $z = 1$ will be a generalized eigenvalue of equations (8.2.36), (8.2.37) associated with $\kappa_1 = (R^2 + R)/(R^2 - R)$. Therefore, difference scheme (8.2.34)–(8.2.35) is unstable.

Example 8.2.4 Show that the Lax-Wendroff scheme (8.2.34) along with boundary condition

$$u_0^{n+1} = u_1^{n+1} \tag{8.2.42}$$

is stable for $-1 \leq R \leq 0$.

Solution: As we saw in Example 8.2.3, the characteristic equation associated with difference equation (8.2.34) is given by

$$\kappa^2 + \frac{2(1 - R^2 - z)}{R^2 - R}\kappa + \frac{R^2 + R}{R^2 - R} = 0. \tag{8.2.43}$$

If we transform equation (8.2.42), we get $z\tilde{u}_0 = z\tilde{u}_1$ or

$$\tilde{u}_0 = \tilde{u}_1. \tag{8.2.44}$$

Substituting $\tilde{u}_k = \phi_1 \kappa^k$ into equation (8.2.44) gives $\kappa = 1$. From equation (8.2.43), we see that $\kappa = 1$ is associated with $z = 1$. Hence, we already know that there will be no eigenvalues associated with equations (8.2.36), (8.2.44).

We must determine whether $z = 1$ is a generalized eigenvalue of equations (8.2.36), (8.2.44). From Example 8.2.3 we know that it must be the case that if we have $z = 1$, we have $\kappa_2 = 1$ and $\kappa_1 = (R^2 + R)/(R^2 - R)$. For this scheme, κ_1 is not relevant, since κ_1 will not satisfy equation (8.2.44). We note that for $|z| > 1$, κ_2 will satisfy $|\kappa_2| > 1$. This is the case because κ_1 is clearly inside the circle $|z| = 1$, so κ_2 must be outside that circle. Since $z = 1$ is associated with κ_2, the solution at $z = 1$ does not satisfy condition (3) of Definition 8.2.12, $z = 1$ is not a generalized eigenvalue, and difference scheme (8.2.34)–(8.2.42) is stable.

We emphasize that we have already assumed that the difference scheme was stable as an initial–value problem scheme. Hence, the stability proved here will be conditional stability with condition $|R| \leq 1$ (due to the fact that this is the condition required for the Lax-Wendroff scheme to be stable as an initial–value problem scheme). And since we have assumed that $a < 0$, *difference scheme (8.2.34), (8.2.42) will be stable for* $-1 \leq R \leq 0$.

Example 8.2.5 Discuss the stability of the Crank-Nicolson scheme, (5.4.10)

$$-\frac{R}{4}u_{k-1}^{n+1} + u_k^{n+1} + \frac{R}{4}u_{k+1}^{n+1} = \frac{R}{4}u_{k-1}^{n} + u_k^{n} - \frac{R}{4}u_{k+1}^{n}. \tag{8.2.45}$$

along with numerical boundary condition

$$u_0^{n+1} = u_1^{n+1}. \tag{8.2.46}$$

Solution: We note that equations (8.2.45) and (8.2.46) are in the form of our general equation (8.2.6) and our general boundary condition (8.2.9). If we transform equations (8.2.45) and (8.2.46), we obtain the resolvent equation

$$z\left(-\frac{R}{4}\tilde{u}_{k-1} + \tilde{u}_k + \frac{R}{4}\tilde{u}_{k+1}\right) = \frac{R}{4}\tilde{u}_{k-1} + \tilde{u}_k - \frac{R}{4}\tilde{u}_{k+1} \tag{8.2.47}$$

and transformed homogeneous boundary condition

$$\tilde{u}_0 = \tilde{u}_1. \tag{8.2.48}$$

For $|z| > 1$, we know that the general solution of equation (8.2.47) will be of the form $\tilde{u}_k = \phi_1 \kappa_1^k$ where κ_1 is the root of the characteristic equation

$$\frac{R}{4}(z+1)\kappa^2 + (z-1)\kappa - \frac{R}{4}(z+1) = 0 \tag{8.2.49}$$

(obtained by looking for a solution of equation (8.2.47) of the form $\tilde{u}_k = \kappa^k$) that satisfies $|\kappa_1| < 1$. We also know that the "other root" of equation (8.2.49) satisfies $|\kappa_2| > 1$ for $|z| > 1$.

If we look for a solution of equation (8.2.48) of the form $\tilde{u}_k = \phi_1 \kappa^k$, we see that we must have $\kappa = 1$. Since for $|z| > 1$, we have $|\kappa_1| < 1$; it follows that equations (8.2.47), (8.2.48) have no eigenvalues (and cannot be pronounced unstable by Proposition 8.2.9).

Substituting $\kappa = 1$ into equation (8.2.49), we see that $\kappa = 1$ corresponds to $z = 1$. Hence, for $z = 1$, we do have a nontrivial solution to equations (8.2.47) and (8.2.48). We must determine whether this solution corresponding to $\kappa = 1$ is the limit of solutions of equation (8.2.47) associated with $|w| > 1$ (condition (3) of Definition 8.2.12). To help decide whether this is the case, we introduce a technique that is often applicable in this situation. We set $z = 1 + \delta$ and $\kappa = 1 + \eta$, and substitute these values into equation (8.2.49). We get

$$\frac{R}{4}(\delta+2)(1+\eta)^2 + \delta(1+\eta) - \frac{R}{4}(\delta+2) = 0.$$

If we expand this expression, simplify and solve for δ, we get

$$\delta = -R\eta - \frac{R}{2}\eta^2 - (\frac{R}{2}+1)\delta\eta - \frac{R}{4}\delta\eta^2.$$

The linear terms dominate, so solutions to equation (8.2.49) near $z = 1$, $\kappa = 1$ satisfy

$$\delta = -R\eta + \mathcal{O}(\eta^2) + \mathcal{O}(\delta\eta).$$

Since R is negative, when δ is positive ($z > 1$), η is also positive ($\kappa > 1$). Thus, as w approaches $z = 1$ from the outside ($|w| > 1$), then κ is also outside ($|\kappa| > 1$). Therefore, $z = 1$ is associated with $\kappa_2 = 1$ (and because κ_1 and κ_2 must satisfy $\kappa_1\kappa_2 = -1$, $z = 1$ is also associated with $\kappa_1 = -1$, which will not satisfy equation (8.2.46)), and $z = 1$ is not a generalized eigenvalue. *The Crank-Nicolson scheme along with numerical boundary condition (8.2.46) is stable for $R < 0$.*

We should make a special note of the perturbation method introduced in Example 8.2.5 for determining whether z, $|z| = 1$, is a generalized eigenvalue. We will be using variations of this technique often. Using an algebraic manipulator to perform some of these expansions can make the job easier.

Example 8.2.6 Discuss the stability of the Crank-Nicolson scheme along with numerical boundary condition

$$u_0^{n+1} = u_1^n. \tag{8.2.50}$$

Solution: Again we see that this difference scheme is a special case of the general example considered in this section. Since we are using the same difference equation as in the last example, the resolvent equation and characteristic equations are still given by equations (8.2.47) and (8.2.49), respectively. Transforming boundary condition (8.2.50) gives

$$z\tilde{u}_0 = \tilde{u}_1. \tag{8.2.51}$$

Substituting $\tilde{u}_k = \phi_1 \kappa^k$ into equation (8.2.51), we see that κ and z must satisfy $z = \kappa$. Since $|\kappa| \leq 1$ and $|z| \geq 1$, clearly equations (8.2.47), (8.2.51) have no eigenvalues.

To determine whether there are any generalized eigenvalues, we substitute $z = \kappa$ into equation (8.2.49) to get

$$\frac{R}{4}\kappa^3 + \left(\frac{R}{4} + 1\right)\kappa^2 - \left(\frac{R}{4} + 1\right)\kappa - \frac{R}{4} = 0. \tag{8.2.52}$$

Factoring equation (8.2.52) (we should be able to factor (8.2.52) manually, but using maple was much easier), we get

$$\kappa = 1, \quad \kappa_+ = \frac{-2 - R + 2\sqrt{1+R}}{R}, \quad \kappa_- = \frac{-2 - R - 2\sqrt{1+R}}{R}. \tag{8.2.53}$$

Obviously, $z = \kappa = 1$ is a potential generalized eigenvalue. To determine whether $\kappa = 1$ satisfies condition (3) of Definition 8.2.12, we set $z = 1 + \delta$ and $\kappa = 1 + \eta$, substitute z and κ into the characteristic equation (8.2.49) and get

$$\frac{R}{4}(2+\delta)(1+\eta)^2 + \delta(1+\eta) - \frac{R}{4}(2+\delta) = 0.$$

If we expand this expression, simplify and solve for δ, we get

$$\delta = -R\eta + \mathcal{O}(\eta^2) + \mathcal{O}(\delta\eta).$$

Therefore, since $-R > 0$, as w approaches $z = 1$ from outside, κ is also outside, i.e., $z = 1$ is associated with $\kappa_2 = 1$, and $z = 1$ is not a generalized eigenvalue (as was the case given by exactly the same calculation in Example 8.2.5).

We next consider the last two roots given in (8.2.53), κ_+ and κ_-. We discuss them in two stages: when the roots are real and when the roots are complex.

The roots are real if $R \geq -1$. If $\kappa_\pm$ are real and we are interested in generalized eigenvalues, we are interested in the case when $z = \kappa_+ = \pm 1$ and $z = \kappa_- = \pm 1$ (if $|z| = 1$, then $z = \kappa$ and κ must be real, and so the only possibilities are $\kappa = z = \pm 1$). We obtain four equations by setting $\kappa_+ = \pm 1$ and $\kappa_- = \pm 1$ in (8.2.53). It is easy to solve these four equations for R and see that the only solutions are $R = -1$ (a case that we have already considered, $z = \kappa = 1$) and $R = 0$ (a case that is irrelevant). Therefore, there are no generalized eigenvalues for $-1 \leq R < 0$. We should note that a very easy way to see (not prove) that $\kappa_\pm$ is never equal to plus or minus one for $-1 < R < 0$ is to graph the expressions for $\kappa_\pm$ given in (8.2.53) as a function of R defined for R in $(-1, 0)$.

When $R < -1$, we see that

$$\begin{aligned} |\kappa_\pm|^2 &= \left|\frac{-2 - R \pm 2\sqrt{1+R}}{R}\right|^2 \\ &= \left|\frac{-2 - R \pm 2i\sqrt{-1-R}}{R}\right|^2 \\ &= \frac{1}{R^2}\left\{(-2-R)^2 + 4(-1-R)\right\} \\ &= 1. \end{aligned}$$

We begin by considering $z = \kappa_+$. Since we already know that $|z| = 1$, $z = \kappa_+$ and $|\kappa_+| = 1$, we are left to determine whether κ_+ is equal to κ_1 or κ_2. We proceed using a similar approach to that used in the last example. Set

$$z = z_0\frac{1+\delta}{1-\delta}, \quad \kappa = \kappa_0(1+\eta) \tag{8.2.54}$$

(where $z_0 = \kappa_0 = e^{i\theta_0} = \kappa_+$, $\cos\theta_0 = (-2-R)/R$, $\sin\theta_0 = 2\sqrt{-1-R}/R$), substitute these values into the characteristic equation (8.2.49) and get

$$\begin{aligned} 0 &= \kappa_0^2(1+\eta)^2 + \frac{4}{R}\frac{z_0\frac{1+\delta}{1-\delta} - 1}{z_0\frac{1+\delta}{1-\delta} + 1}\kappa_0(1+\eta) - 1 \\ &= \kappa_0^2(1+\eta)^2 + \left(\frac{4}{R}i\tan\frac{\theta_0}{2} + \delta\frac{4}{R}\left[1 + \tan^2\frac{\theta_0}{2}\right] + \mathcal{O}(\delta^2)\right)\kappa_0(1+\eta) - 1. \end{aligned}$$

If we expand this expression, simplify and solve for δ, we get

$$\delta = \frac{-R}{2} \frac{\cos\theta_0}{1+\tan^2\frac{\theta_0}{2}} \eta + \mathcal{O}(\delta^2) + \mathcal{O}(\delta\eta). \tag{8.2.55}$$

When $\cos\theta_0 > 0$, then η is positive when δ is positive, $\kappa_+ = \kappa_2$ and $z = \kappa_+$ is not a generalized eigenvalue. And, when $\cos\theta_0 < 0$, then η is negative when δ is positive, $\kappa_+ = \kappa_1$ and $z = \kappa_+$ is a generalized eigenvalue and the scheme is unstable.

We notice from the definition of z_0 and θ_0 above that

$$\cos\theta_0 = \frac{2}{|R|} - 1.$$

So, when $-2 < R < -1$ $(1 < |R| < 2)$, $0 < \cos\theta_0 = \frac{2}{|R|} - 1 < 1$. Thus there is no generalized eigenvalue for $-2 < R < -1$. When $R < -2$, then $\cos\theta_0 = \frac{2}{|R|} - 1 < 0$, and the scheme is unstable.

If we were to consider $z = \kappa_-$, everything that we have done above is the same, except for the fact that now $\sin\theta_0 = -2\sqrt{-1-R}/R$. Since the relationship between δ and η given by equation (8.2.55) depends only on $\cos\theta_0$ and not on $\sin\theta_0$, the results will be exactly the same. Hence, for $-2 < R < -1$ there will be no generalized eigenvalues, and the scheme will be stable. And, as was stated above, the scheme is unstable for $R < -2$.

And finally, it is easy to see that when $R = -2$, $z = \kappa_\pm = \mp i$. Consider the case when $z = i$. Then $\kappa = i$ is a double root of characteristic equation (8.2.49), i.e. $\kappa_1 = i$ and $\kappa_2 = i$. Hence, $z = i$ must be a generalized eigenvalue and the scheme is unstable. The argument showing that $z = -i$ is a generalized eigenvalue is similar.

Therefore, *the Crank-Nicolson scheme along with numerical boundary condition (8.2.50) is stable for* $-2 < R < 0$ *and unstable for* $R \leq -2$.

Remark 1: We note that the above result is an example of a scheme that is unconditionally stable as an initial-value problem scheme, but only conditionally stable when we include the numerical boundary condition (8.2.50). We also emphasize that numerical boundary condition (8.2.50) is a very popular numerical boundary condition when an implicit scheme (including the Crank-Nicolson scheme) is being used, because the implementation of numerical boundary condition (8.2.50) is the same as if we were given a Dirichlet boundary condition at $x = 0$.

Remark 2: We should note that the perturbation analysis done above was not especially different from the ones that we have performed earlier, except that we made a rather odd choice of perturbation in (8.2.54). We first note that

$$z = z_0 \frac{1+\delta}{1-\delta} = z_0\big(1 + 2\delta + \mathcal{O}(\delta^2)\big),$$

so z does approach z_0 from the outside if δ is positive and approaches zero. It is possible to make the more routine choices for the perturbations, but the calculations become a nightmare. Sometimes we must be very clever when we choose the form of the perturbations.

The examples given above illustrate, we hope, some of the different reasons that schemes for initial-boundary-value problems are unstable and give the necessary tools for proving that a scheme is either stable or unstable. There is an assortment of reasonable standard numerical boundary

conditions that are used, and they can be analyzed by the methods given in these examples.

- The Lax-Wendroff scheme along with the numerical boundary condition

  1.
  $$\delta_+^j u_0^{n+1} = 0 \tag{8.2.56}$$
  is stable if $-1 \leq R \leq 0$. Note that if $j = 1$ or $j = 2$, this numerical boundary condition is of the form given by equation (8.2.9).

  2.
  $$u_0^{n+1} = u_0^n - R(u_1^n - u_0^n) \tag{8.2.57}$$
  is stable for $-1 \leq R \leq 0$.

  3.
  $$u_0^{n+1} + u_1^{n+1} + R(u_1^{n+1} - u_0^{n+1}) = u_0^n + u_1^n - R(u_1^n - u_0^n) \tag{8.2.58}$$
  is stable for $-1 \leq R \leq 0$.

- The Crank-Nicolson scheme along with the numerical boundary condition

  1. $\delta_+^j u_0^{n+1} = 0$ is stable for $R \leq 0$. (Note that only if $j = 1$ and $j = 2$ is this numerical boundary condition of the form given by equation (8.2.9).)
  2. $u_0^{n+1} = u_0^n - R(u_1^n - u_0^n)$ is stable for $-2 \leq R \leq 0$.
  3. $u_0^{n+1} + u_1^{n+1} + R(u_1^{n+1} - u_0^{n+1}) = u_0^n + u_1^n - R(u_1^n - u_0^n)$ is stable for $R \leq 0$.
  4. ${}^-\delta_+^j u_0^{n+1} = 0$ is stable for $-2 < R \leq 0$ where ${}^-\delta_+ u_0^{n+1} = u_1^n - u_0^{n+1}$.

- The Lax-Friedrichs scheme along with numerical boundary condition

  1. $u_0^{n+1} - u_1^{n+1} = 0$ is stable for $-1 \leq R \leq 0$.
  2. $u_0^{n+1} - u_2^{n+1} = 0$ is stable for $-1 \leq R \leq 0$.
  3. $2u_0^{n+1} - u_1^{n+1} - u_2^{n+1} = 0$ is stable for $-1 \leq R \leq 0$.

Remark 1: In HW5.6.10 and HW5.6.11 we used four different numerical boundary conditions along with the Lax-Wendroff and Crank-Nicolson

schemes to try to approximate the solution to the initial–boundary–value problem

$$\begin{aligned} v_t - 2v_x &= 0, \quad x \in (0,1),\ t > 0 \\ v(1,t) &= 1.0, \quad t \geq 0 \\ v(x,0) &= 1 + \sin 2\pi x, \quad x \in [0,1]. \end{aligned}$$

The first numerical boundary condition, (a), was a poor choice (but maybe not at the time) and produced bad results for both schemes. Numerical boundary conditions (b), (c) and (d) were the same as (8.2.46), (8.2.57) and (8.2.58), respectively. We saw, we hope, that both the the Lax-Wendroff scheme and the Crank-Nicolson scheme for the appropriate R values were stable when used with numerical boundary condition (8.2.46). These results corroborate the analytic results found in Examples 8.2.4 and 8.2.5. Likewise, we saw that both schemes were stable when used with numerical boundary condition (8.2.57) and (8.2.58) for the correct ranges of R. The analyses of stability for these schemes are left to the reader in HW8.2.2.

Remark 2: If we were to conduct a numerical experiment as we did in HW5.6.11 for numerical boundary condition (8.2.50), our results would not be quite so satisfying. See HW8.2.3. (However, it may be my code that is at fault.) The problem used in the experiment was the same as that used in HW5.6.11 with $\Delta x = 0.01$. The scheme seems to be nicely stable for values of R in $[-0.7, 0)$. The analysis performed in Example 8.2.6 shows that the scheme is stable for R satisfying $-2 < R < 0$ and unstable for $R \leq -2$. As we choose values of R less than -0.7, an oscillation appears. An oscillation, if it does not blow up, is not necessarily an instability. The oscillation gets worse as R gets nearer to -2.0. The quality of the solution is unacceptable. When $R < -2$, the oscillations get worse, but the solution does not blow up. Here, too, the solutions are unacceptable for use. We must remember that the norms used in the definition of stability for initial–boundary–value schemes involves the time values out to infinity. This is difficult to simulate numerically. Also, when an instability is due to an eigenvalue (z, κ_1), the instability is due to the solution $z^n \kappa_1^k$. When the instability is due to a generalized eigenvalue, $|z| = 1$, we do not see the growth as with an eigenvalue.

In Section 8.2.2, we extend the class of schemes for which we can analyze the stability and give the general results in Section 8.2.3. In each of these sections we will include an overview of some of the available stability results.

HW 8.2.1 Consider the following initial–boundary–value problem.

$$\begin{aligned} v_t - 2v_x &= 0, \quad x \in (0,1),\ t > 0 \\ v(x,0) &= 1 + \sin 2\pi x, \quad x \in [0,1] \\ v(1,t) &= 1.0 \end{aligned}$$

Conduct a numerical experiment that will give you an indication as to whether the Lax-Wendroff scheme along with numerical boundary condition $u_0^{n+1} = u_1^{n+1} + u_0^n - u_1^n$ will be stable or unstable.

HW 8.2.2 (a) Analyze the stability of the Lax-Wendroff scheme used with each of the numerical boundary conditions (8.2.57) and (8.2.58).
(b) Analyze the stability of the Crank-Nicolson scheme used with each of the numerical boundary conditions (8.2.57) and (8.2.58).
(c) Show analytically that both the Lax-Wendroff and Crank-Nicolson schemes used with numerical boundary condition (a) from HW5.6.10 ($u^n + 0 = 1.0$) are unstable.

HW 8.2.3 Conduct a numerical experiment to study the stability of the Crank-Nicolson scheme when used with numerical boundary condition (8.2.50). Use the problem given in HW8.2.1.

8.2.1.2 Stability of the Left Quarter Plane Problem

Based on Theorem 8.2.2, difference scheme (8.2.6)–(8.2.9) is stable if the associated right and left quarter plane problems are stable. In the last section we spent a lot of time and space investigating the stability of the associated right quarter plane problem. In this section we will study the stability of the left quarter plane problem associated with difference scheme (8.2.6)–(8.2.9). Specifically, we will consider the left quarter plane grid on $(-\infty, 1]$ defined in Section 8.2.1 by difference equation (8.2.11), initial condition (8.2.12) and boundary condition

$$u_M^{n+1} = g^{n+1} \tag{8.2.59}$$

(which is the same as boundary condition (8.2.7) and (8.2.14)). Most of the results developed for the right quarter plane problem are also relevant and true for the left quarter plane problem. In particular, we have the following results.

- The definition of stability of the left quarter plane problem will be the same as Definition 8.2.3, except that the sum will be taken over the left quarter plane grid and the boundary condition g_0^n will be replaced by g^n.

- The resolvent equation will have the same form as equation (8.2.24), and the boundary condition (8.2.59) will transform into

$$\tilde{u}_M = \tilde{g}^n.$$

The above transformed boundary condition looks easier than transformed boundary condition (8.2.25) because boundary condition (8.2.9)

is a general form of a numerical boundary condition whereas boundary condition (8.2.59) is a nice mathematical boundary condition. The homogeneous transformed boundary condition is then written as (analogous to equation (8.2.26))

$$\tilde{u}_M = 0. \tag{8.2.60}$$

- The definition of eigenvalue of problem (8.2.24), (8.2.60) will be the same as Definition 8.2.8, except that the eigenvector will now be of the form $\tilde{\mathbf{u}} = [\cdots \ \tilde{u}_{M-1} \ \tilde{u}_M]^T$.

- Propositions 8.2.9 and 8.2.10, Definition 8.2.12 (except that the generalized eigenvector will again be of the form $\tilde{\mathbf{u}} = [\cdots \ \tilde{u}_{M-1} \ \tilde{u}_M]^T$) and Proposition 8.2.13 all are equally appropriate and true for the left quarter plane problem.

- The approach used to solve the resolvent equation, along with the transformed boundary condition, is the same, except that we now consider solutions of the form $\tilde{u}_k = \phi\kappa_2^k$ where $|\kappa_2| > 1$ (since now k will approach $-\infty$).

Thus, as we did for the right quarter plane problem, we solve the characteristic equation (8.2.28) for κ_1 and κ_2. The major difference between the left quarter plane problem and the right quarter plane problem is that we now have negative indices $k = -\infty, \ldots, M$. Thus to satisfy $\|\tilde{\mathbf{u}}\|_{2,\Delta x} < \infty$, *we must choose* $\phi_1 = 0$ *and look for solutions of the form* $\tilde{u}_k = \phi_2\kappa_2^k$. Substituting this into equation (8.2.60) gives $\phi_2 = 0$.

Therefore, *there clearly are no eigenvalues or generalized eigenvalues associated with equations (8.2.24), (8.2.60), and by Proposition 8.2.13 the left quarter plane scheme is stable.*

Obviously, the analysis of the left quarter plane problem was much easier than the analysis of the right quarter plane problem. The left quarter plane problem was easier to analyze because we had an easy mathematical boundary condition at $x = 1$ that was well-posed for the analytic problem (instead of the numerical boundary condition we had at $x = 0$). The stability of the difference scheme for the left quarter plane problem follows from the stability of the difference scheme for the associated initial-value problem. *If a were chosen such that $a > 0$, then we would have a mathematical boundary condition at $x = 0$, a numerical boundary condition at $x = 1$, the analysis of the right quarter plane problem would be easy (if we were given a reasonable mathematical boundary condition) and the analysis of the left quarter plane problem would be similar to that done for the right quarter plane problem in Section 8.2.1.1.*

8.2.2 Stability: Another Easy Case

8.2.2.1 Stability of the Right Quarter Plane Problem

The definitions, propositions and examples given in the last couple of sections have, we hope, given us a view of the types of results we are able to prove and how we must go about proving them. Instead of skipping forward to the general theory now, we take another small step and consider another special case. Specifically, in this section we obtain results for a difference scheme of the same form as that considered in Section 8.2.1, except that we will now consider systems of partial differential equations. As the section title above indicates, we shall proceed immediately to the right quarter plane problem. We do this because, as was the case in Section 8.2.1, we apply a slight variation of Theorem 8.2.2 to see that we can obtain stability for a scheme defined on an interval by obtaining stability for both the right and left quarter plane problems. Specifically, consider the difference scheme

$$\begin{aligned} A_{-1\,-1}\mathbf{u}_{k-1}^{n+1} + A_{0\,-1}\mathbf{u}_{k}^{n+1} + A_{1\,-1}\mathbf{u}_{k+1}^{n+1} \\ = A_{-1\,0}\mathbf{u}_{k-1}^{n} + A_{0\,0}\mathbf{u}_{k}^{n} + A_{1\,0}\mathbf{u}_{k+1}^{n} + \Delta t\mathbf{G}_{k}^{n}, \\ k = 1, \ldots \end{aligned} \tag{8.2.61}$$

$$\mathbf{u}_{k}^{0} = \mathbf{f}_{k} \quad k = 0, \ldots, \tag{8.2.62}$$

$$\mathbf{u}_{0}^{n+1} = C_{1}^{0}\mathbf{u}_{1}^{n+1} + C_{2}^{0}\mathbf{u}_{2}^{n+1} + C_{1}^{1}\mathbf{u}_{1}^{n} + C_{2}^{1}\mathbf{u}_{2}^{n} + \mathbf{g}_{0}^{n}. \tag{8.2.63}$$

As we have in the past, we assume that the vectors $\mathbf{u}_k^n$ are K-vectors and A_{pq} and C_q^p are $K \times K$ matrices. We consider difference scheme (8.2.61)–(8.2.63) to be the result of approximating a system of K partial differential equations defined on $(0, \infty)$. We will generally model our approach based on a hyperbolic system of partial differential equations of the form

$$\mathbf{v}_t = A\mathbf{v}_x, \ x \in (0, \infty), \ t > 0$$

though at times we will add other restrictions. We recall that for a system such as this, the number of mathematical boundary conditions available at $x = 0$ is equal to the number of negative eigenvalues of the matrix A. See Section 6.3.1. Hence, *when we write boundary condition (8.2.63), part of its definition might be due to mathematical boundary conditions and part might be due to numerical boundary conditions.*

To consider the stability of difference scheme (8.2.61)–(8.2.63), we theoretically must define new norms, provide a new definition of stability and prove or state the appropriate theorems that are necessary. To eliminate some of this tedium, we make the following observations.

- The norms used in this section are the obvious extensions of those norms defined in Section 8.2.1.1 (where the appropriate absolute values must be replaced by finite l_2 norms of K-vectors).
- Using these norms, the definition of stability of difference scheme (8.2.61)–(8.2.63) follows immediately from Definition 8.2.3.

- If the definition of the discrete Laplace transform, Definition 8.2.5, is extended to vector valued sequences (of the form $\{\mathbf{u}^n\}$), then vector analogues of Propositions 8.2.6 and 8.2.7 follow immediately. Of course, in this case the discrete Laplace transform of $\{\mathbf{u}^n\}$ will be a vector valued function of the form $\tilde{\mathbf{u}} = \tilde{\mathbf{u}}(s)$, and the L_2 norm used in the analogue to Proposition 8.2.7 will be the L_2 integral of the finite ℓ_2 norm of $\tilde{\mathbf{u}}$.

We should note that these extensions to include vector valued difference equations are very similar to the extensions we made in Chapter 6.

There is a lot of material contained in the previous paragraph. The reader should be careful not to let this change to vector valued equations cause confusion. Everything that we do in this section will be analogous to what we did in Section 8.2.1.1. In addition to the observations made above, we emphasize that we must still satisfy Assump 8.1, Assump 8.2 and Assump 8.3. We must also make the following very strong assumption that we made so often in Chapter 6.

Assump 8.4 The matrices A_{pq}, $p = -1, 0, 1$, $q = -1, 0$ are simultaneously diagonalizable.

As we stated in Chapter 6, Assump 8.4 is a strong assumption. However, in this setting this assumption is not so strong. When we are considering a difference scheme associated with solving a hyperbolic first order system of partial differential equations ($\mathbf{v}_t = A\mathbf{v}_x$), the matrices will most often be simultaneously diagonalizable, because all of the matrices will be identity matrices or multiples of A. In [19], page 659, an alternative to Assump 8.4 (Assumption 5.2) is given. In other cases, we may have to proceed without satisfying this hypothesis, using the results, knowing that they are at least suspect, along with careful experimentation.

We take the discrete Laplace transform of equation (8.2.61) (ignoring the function $\mathbf{F}$) and obtain the resolvent equation

$$z\left(A_{-1\,-1}\tilde{\mathbf{u}}_{k-1} + A_{0\,-1}\tilde{\mathbf{u}}_k + A_{1\,-1}\tilde{\mathbf{u}}_{k+1}\right) = A_{-1\,0}\tilde{\mathbf{u}}_{k-1} + A_{0\,0}\tilde{\mathbf{u}}_k + A_{1\,0}\tilde{\mathbf{u}}_{k+1}. \tag{8.2.64}$$

We then take the discrete Laplace transform of boundary condition (8.2.63) and obtain the transformed boundary condition

$$z\tilde{\mathbf{u}}_0 = z\left(C_1^0\tilde{\mathbf{u}}_1 + C_2^0\tilde{\mathbf{u}}_2\right) + C_1^1\tilde{\mathbf{u}}_1 + C_2^1\tilde{\mathbf{u}}_2 + \tilde{\mathbf{g}}_0, \tag{8.2.65}$$

and write the homogeneous version of equation (8.2.65) as

$$z\left(\tilde{\mathbf{u}}_0 - C_1^0\tilde{\mathbf{u}}_1 - C_2^0\tilde{\mathbf{u}}_2\right) = C_1^1\tilde{\mathbf{u}}_1 + C_2^1\tilde{\mathbf{u}}_2. \tag{8.2.66}$$

As we did in Definitions 8.2.8 and 8.2.12, we say that

(1) z, $|z| > 1$, is an **eigenvalue** of equations (8.2.64), (8.2.66) if $(z, \tilde{\mathbf{u}})$ is a solution to equations (8.2.64), (8.2.66) and $\|\tilde{\mathbf{u}}\|_2 < \infty$, and
(2) z, $|z| = 1$, is a generalized eigenvalue of equations (8.2.64), (8.2.66) if $(z, \tilde{\mathbf{u}})$ is a solution to equations (8.2.64), (8.2.66) and $\tilde{\mathbf{u}}_k$ satisfies

$$\tilde{\mathbf{u}}_k = \lim_{w \to z, |w|>1} \tilde{\mathbf{u}}_k(w),$$

where $(w, \tilde{\mathbf{u}}(w))$ is a solution to equation (8.2.64).

We are then able to state the analogue to Proposition 8.2.9. Instead, we proceed to our main result and state the following analogue to Proposition 8.2.13.

Proposition 8.2.14 *Difference scheme (8.2.61)-(8.2.63) is stable if and only if eigenvalue problem (8.2.64), (8.2.66) has no eigenvalues and no generalized eigenvalues.*

Hence, we now know how to prove that schemes for the right quarter plane problem are stable or unstable. Just as we did in Section 8.2.1.1, we must solve the resolvent equation, (8.2.64) along with the homogeneous transformed boundary condition (8.2.66). If this problem has an eigenvalue (solution for $|z| > 1$), then the scheme is unstable. If there are solutions to (8.2.64), (8.2.66) for $|z| = 1$, then we must determine whether they come from the inside or the outside (whether $\kappa = \kappa_1$ or $\kappa = \kappa_2$; whether condition (3) of Definition 8.2.8 is satisfied). If there is a generalized eigenvalue, the scheme is unstable; and if there is not a generalized eigenvalue, the scheme is stable.

Thus, we must solve equation (8.2.64). We look for solutions of the form $\tilde{\mathbf{u}}_k = \kappa^k \boldsymbol{\phi}$ where $\boldsymbol{\phi}$ is a K-vector. Substituting this form of $\tilde{\mathbf{u}}_k$ into equation (8.2.64), we get

$$\left\{ z \left[\kappa^{-1} A_{-1\,-1} + A_{0\,-1} + \kappa A_{1\,-1} \right] - \kappa^{-1} A_{-1\,0} - A_{0\,0} - \kappa A_{1\,0} \right\} \boldsymbol{\phi} = \boldsymbol{\theta} \tag{8.2.67}$$

where as in Part 1, $\boldsymbol{\theta}$ denotes the zero vector. And of course, equation (8.2.67) has a solution if and only if

$$\det \left\{ z \left[\kappa^{-1} A_{-1\,-1} + A_{0\,-1} + \kappa A_{1\,-1} \right] - \kappa^{-1} A_{-1\,0} - A_{0\,0} - \kappa A_{1\,0} \right\} = 0 \tag{8.2.68}$$

or

$$\det \left\{ (z A_{1\,-1} - A_{1\,0}) \kappa^2 + (z A_{0\,-1} - A_{0\,0}) \kappa + z A_{-1\,-1} - A_{-1\,0} \right\} = 0. \tag{8.2.69}$$

Equation (8.2.69) is referred to as the **characteristic equation**, and it is much nastier than the characteristic equation associated with scalar

schemes. Inspection of equation (8.2.69) shows that it will generally be a polynomial of degree $2K$ with respect to κ. We must be aware that for $K > 1$, solving equation (8.2.69) may be very difficult or impossible. However, we do still get the following analogue to Proposition 8.2.10. (Lemmas 5.1 and 5.2, [19], pages 659–660.)

Proposition 8.2.15 *For $|z| > 1$ characteristic equation (8.2.69) has no solution κ with $|\kappa| = 1$. For $|z| > 1$, K roots of equation (8.2.69) will satisfy $|\kappa_1| < 1$ and K roots will satisfy $|\kappa_2| > 1$.*

To keep our notation consistent with the earlier treatment, we denote the roots less than and greater than one in magnitude by κ_{1_j}, $j = 1, \ldots, K$ and κ_{2_j}, $j = 1, \ldots, K$, respectively. When the roots are distinct, we know that the solution of equation (8.2.64) can be written as

$$\tilde{\mathbf{u}}_k = \sum_{j=1}^{K} \kappa_{1_j}^k \boldsymbol{\phi}_j$$

and equation (8.2.64), along with boundary condition (8.2.66), can be used to determine $\boldsymbol{\phi}_j$, $j = 1, \ldots, K$. When the roots are not distinct, we can write $\tilde{\mathbf{u}}_k$ as

$$\tilde{\mathbf{u}}_k = \sum_{j} P_j(k) \kappa_{1_j}^k \boldsymbol{\phi}_j$$

where the sum is taken over the distinct values of κ_{1_j} and P_j is a polynomial of degree one less than the multiplicity of κ_{1_j}. Another way to view the solution to equation (8.2.64) in the case where we have distinct roots is as

$$\tilde{\mathbf{u}}_k = \sum_{j=1}^{K} c_j \kappa_{1_j}^k \mathbf{u}_j \tag{8.2.70}$$

where κ_{1_j}, $j = 1, \ldots, K$ are the roots of equation (8.2.69) that satisfy $|\kappa| < 1$, $\mathbf{u}_j$ are the vectors in the null space of the coefficient matrix of equation (8.2.67) associated with κ_{1_j}, $j = 1, \ldots, K$ and the constants $c_1, \ldots, c_K$ are determined by the boundary conditions (8.2.66). Insertion of equation (8.2.70) into the homogeneous transformed boundary conditions (8.2.66) leads to an equation of the form

$$D(z) \begin{bmatrix} c_1 \\ \vdots \\ c_K \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \tag{8.2.71}$$

where the j-th column of the matrix $D(z)$ is given by

$$\left\{ z \left(I - \kappa_{1_j} C_1^0 - \kappa_{1_j}^2 C_2^0 \right) - \kappa_{1_j} C_1^1 - \kappa_{1_j}^2 C_2^1 \right\} \mathbf{u}_j. \tag{8.2.72}$$

It is then easy to see that we obtain the following analogue to Proposition 8.2.11.

Proposition 8.2.16 *If* $\det D(z) \neq 0$ *for all* z, $|z| \geq 1$, *then difference scheme (8.2.61)–(8.2.63) is stable.*

The above result is a nice result, but as we shall see, it will often be the case that $\det D(z) = 0$ for some z, $|z| \geq 1$. Clearly, if $\det D(z) = 0$ for z, $|z| > 1$, then equations (8.2.64), (8.2.66) have an eigenvalue and difference scheme (8.2.61)–(8.2.63) is unstable by Proposition 8.2.14. If $\det D(z) = 0$ for z, $|z| = 1$, we must determine whether or not this value of z is a generalized eigenvalue.

We next include several examples that will illustrate the use of the definitions and results described above. These examples, though elementary, include many of the difficulties faced when we must consider stability of systems of difference equations.

Example 8.2.7 Discuss the stability of the Crank-Nicolson difference equation

$$\mathbf{u}_k^{n+1} - \frac{R}{4} A \delta_0 \mathbf{u}_k^{n+1} = \mathbf{u}_k^n + \frac{R}{4} A \delta_0 \mathbf{u}_k^n, \quad k = 1, \ldots \tag{8.2.73}$$

where

$$A = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -2 \end{pmatrix},$$

along with boundary conditions

$$u_{1_0}^{n+1} = u_{1_1}^{n+1} \tag{8.2.74}$$

$$u_{2_0}^{n+1} = u_{2_1}^{n+1} \tag{8.2.75}$$

$$u_{3_0}^{n+1} = -u_{1_0}^{n+1} - u_{2_0}^{n+1} + g_3^{n+1} \tag{8.2.76}$$

$(\mathbf{u}_k^{n+1} = [u_{1_k}^{n+1} \; u_{2_k}^{n+1} \; u_{3_k}^{n+1}]^T)$.

Solution: We note that difference equation (8.2.73) is the result of using the Crank-Nicolson scheme to discretize the system of partial differential equations $\mathbf{v}_t = A\mathbf{v}_x$ where A is the given 3×3 matrix. Since the system has one negative eigenvalue, -2, we get one mathematical boundary condition at $x = 0$ (which we discretize as (8.2.76)). We should note that boundary condition (8.2.76) might be due to a boundary condition for the analytic problem of the form

$$v_3(0,t) = -v_1(0,t) - v_2(0,t) + g_3(t).$$

We recall from Section 6.3.1 that this is an acceptable boundary condition associated with matrix A.

Since we have used the Crank-Nicolson scheme (or any scheme that reaches in both directions), we need two numerical boundary conditions at $x = 0$. We choose numerical boundary conditions of the form (8.2.74)–(8.2.75) because they were found to be stable when considered with the appropriate scalar equation in Example 8.2.5.

Before we begin our work, we should note and admit that this is a trivial problem. Write difference equation (8.2.73) as

$$u_{1_k}^{n+1} - 2\frac{R}{4}\delta_0 u_{1_k}^{n+1} = u_{1_k}^n + 2\frac{R}{4}\delta_0 u_{1_k}^n \tag{8.2.77}$$

$$u_{2_k}^{n+1} - \frac{R}{4}\delta_0 u_{2_k}^{n+1} = u_{2_k}^n + \frac{R}{4}\delta_0 u_{2_k}^n. \tag{8.2.78}$$

$$u_{3_k}^{n+1} + 2\frac{R}{4}\delta_0 u_{3_k}^{n+1} = u_{3_k}^n - 2\frac{R}{4}\delta_0 u_{3_k}^n. \tag{8.2.79}$$

Difference equations (8.2.77) and (8.2.78) along with numerical boundary conditions (8.2.74) and (8.2.75) are stable by the results of Example 8.2.5, independent of the third difference equation and last boundary condition. An easy analysis (such as that discussed in Section 8.2.1.2) will show that difference equation (8.2.79) along with boundary condition (8.2.76) (assuming $u_{1_0}^{n+1}$ and $u_{2_0}^{n+1}$ as known) is also stable. Knowing that this scheme must be stable, we illustrate some of the difficulties caused by considering these two difference equations and boundary conditions as a system.

We begin by taking the discrete Laplace transform of equation (8.2.73) to obtain the resolvent equation

$$z\left[\tilde{\mathbf{u}}_k - \frac{R}{4}A\delta_0\tilde{\mathbf{u}}_k\right] = \tilde{\mathbf{u}}_k + \frac{R}{4}A\delta_0\tilde{\mathbf{u}}_k. \tag{8.2.80}$$

If we rewrite boundary conditions (8.2.74)–(8.2.76) as

$$\mathbf{u}_0^{n+1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & -1 & 0 \end{pmatrix} \mathbf{u}_1^{n+1} + \mathbf{g}^{n+1}$$

(replacing the $u_{1_0}^{n+1}$ and $u_{2_0}^{n+1}$ terms on the right hand side of boundary condition (8.2.76) using boundary conditions (8.2.74) and (8.2.75) and letting $\mathbf{g}^{n+1} = [0\,0\,g_3^{n+1}]^T$), transform and consider the homogeneous transformed boundary conditions, we get

$$\tilde{\mathbf{u}}_0 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & -1 & 0 \end{pmatrix} \tilde{\mathbf{u}}_1. \tag{8.2.81}$$

We look for a solution to equations (8.2.80), (8.2.81) of the form

$$\tilde{\mathbf{u}}_k = \boldsymbol{\phi}\kappa^k$$

where $\boldsymbol{\phi}$ is a 3-vector. Insertion of this form into equation (8.2.80) gives us

$$z\left[\boldsymbol{\phi}\kappa^k - \frac{R}{4}(\kappa^{k+1} - \kappa^{k-1})A\boldsymbol{\phi}\right] = \boldsymbol{\phi}\kappa^k + \frac{R}{4}(\kappa^{k+1} - \kappa^{k-1})A\boldsymbol{\phi}$$

or

$$\left[(z+1)\frac{R}{4}A\kappa^2 - (z-1)I\kappa - (z+1)\frac{R}{4}A\right]\boldsymbol{\phi} = \boldsymbol{\theta}. \tag{8.2.82}$$

Thus the characteristic equation is given by

$$0 = \det\left[(z+1)\frac{R}{4}A\kappa^2 - (z-1)I\kappa - (z+1)\frac{R}{4}A\right] \tag{8.2.83}$$

$$= \left[(z+1)\frac{R}{4}2\kappa^2 - (z-1)\kappa - (z+1)\frac{R}{4}2\right]\left[(z+1)\frac{R}{4}\kappa^2 - (z-1)\kappa - (z+1)\frac{R}{4}\right]\left[(z+1)\frac{R}{4}(-2)\kappa^2 - (z-1)\kappa - (z+1)\frac{R}{4}(-2)\right] \tag{8.2.84}$$

and the roots of the characteristic equation are

$$\kappa = \frac{1}{(z+1)R}\left[z - 1 \pm \sqrt{(z-1)^2 + R^2(z+1)^2}\right] \tag{8.2.85}$$

$$\kappa = \frac{2}{(z+1)R}\left[z - 1 \pm \sqrt{(z-1)^2 + \frac{R^2}{4}(z+1)^2}\right] \tag{8.2.86}$$

$$\kappa = \frac{-1}{(z+1)R}\left[z - 1 \pm \sqrt{(z-1)^2 + R^2(z+1)^2}\right]. \tag{8.2.87}$$

We know from Proposition 8.2.15 that for $|z| > 1$, three of the six roots satisfy $|\kappa| < 1$ and three roots satisfy $|\kappa| > 1$. (Since this is a trivial problem, we know that each of the above expressions will produce one root satisfying $|\kappa| < 1$ and each will produce one root satisfying $|\kappa| > 1$.) Let κ_{1_1}, κ_{1_2} and κ_{1_3} denote the roots given by expressions (8.2.85), (8.2.86) and (8.2.87), respectively, that satisfy $|\kappa| < 1$. In general, the solution to the resolvent equation (8.2.80) can be written as

$$\bar{\mathbf{u}}_k = c_1\kappa_{1_1}^k \mathbf{u}_1 + c_2\kappa_{1_2}^k \mathbf{u}_2 + c_2\kappa_{1_3}^k \mathbf{u}_3 \tag{8.2.88}$$

where c_1, c_2 and c_3 are arbitrary and $\mathbf{u}_1$, $\mathbf{u}_2$ and $\mathbf{u}_3$ are the vectors in the null space of the coefficient matrix in equation (8.2.82) associated with κ_{1_1}, κ_{1_2} and κ_{1_3}, respectively. It is easy to see that in this case the vectors that are in the null space of the coefficient matrix of equation (8.2.82) are the same as the eigenvectors of A and are given by $\mathbf{u}_1 = [1\ 0\ 0]^T$, $\mathbf{u}_2 = [0\ 1\ 0]^T$ and $\mathbf{u}_3 = [0\ 0\ 1]^T$.

If we substitute solution (8.2.88) into the homogeneous transformed boundary condition (8.2.81), we get

$$(c_1\mathbf{u}_1 + c_2\mathbf{u}_2 + c_3\mathbf{u}_3) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & -1 & 0 \end{pmatrix} (c_1\kappa_{1_1}\mathbf{u}_1 + c_2\kappa_{1_2}\mathbf{u}_2 + c_3\kappa_{1_3}\mathbf{u}_3)$$

or

$$D(z)\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{pmatrix} 1-\kappa_{1_1} & 0 & 0 \\ 0 & 1-\kappa_{1_2} & 0 \\ \kappa_{1_1} & \kappa_{1_2} & 1 \end{pmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \tag{8.2.89}$$

We note that equation (8.2.89) is a special case of equation (8.2.71). Clearly, equation (8.2.89) has nontrivial solutions only when the determinant of the coefficient matrix is zero, hence, when $\kappa_{1_1} = 1$ or $\kappa_{1_2} = 1$. Thus *we know that we do not have any eigenvalues* (since when $|z| > 1$, both κ_{1_1} and κ_{1_2} satisfy $|\kappa| < 1$).

We must decide whether $\kappa_{1_1} = 1$ or $\kappa_{1_2} = 1$ is the result of a generalized eigenvalue. By setting $\kappa = 1$ in equation (8.2.84), it is easy to see that $\kappa = 1$ is associated with $z = 1$. Immediately we see that we cannot obtain stability for this difference scheme by using Proposition 8.2.16. We must decide whether the $z = 1$, $\kappa = 1$ solution is a limit of solutions from the inside ($|\kappa| < 1$, which would make it a generalized eigenvalue) or from the outside ($|\kappa| > 1$, in which case it would not be a generalized eigenvalue). One way to see that $z = 1$ *is not a generalized eigenvalue* is to consider the solutions (8.2.85)–(8.2.87) carefully. When $z = e^{i\theta}$, i.e., when $|z| = 1$,

$$\kappa_{1_1} = \frac{i}{R}\tan\frac{\theta}{2} - \sqrt{1 - \frac{\tan^2\frac{\theta}{2}}{R^2}} \tag{8.2.90}$$

$$\kappa_{1_2} = \frac{2i}{R}\tan\frac{\theta}{2} - \sqrt{1 - \frac{4\tan^2\frac{\theta}{2}}{R^2}} \tag{8.2.91}$$

$$\kappa_{1_3} = \frac{-i}{R}\tan\frac{\theta}{2} + \sqrt{1 - \frac{\tan^2\frac{\theta}{2}}{R^2}} \tag{8.2.92}$$

(where the two minus signs and one plus sign are chosen so that κ_{1_j} will satisfy $|\kappa_{1_j}| < 1$ when $|z| > 1$ for $j = 1, 2, 3$). When $\theta = 0$ (when $z = 1$), then $\kappa_{1_1} = -1$ and $\kappa_{1_2} = -1$. Hence, the solutions associated with $z = 1$ and $\kappa = 1$ are limits of solutions from the outside ($|\kappa| > 1$), and *equations (8.2.82), (8.2.81) have no generalized eigenvalues.*

Thus, *difference scheme (8.2.73)–(8.2.76) is stable.*

Remark: We might also note that we could also show that $z = 1$ and $\kappa = 1$ does not produce a generalized eigenvalue in the same way that we showed the scalar equation in Example 8.2.5 did not have a generalized eigenvalue when $z = \kappa = 1$. If we set

$z = 1 + \delta$ and $\kappa = 1 + \eta$, substitute these expressions into equation (8.2.84) and expand and simplify, we get

$$0 = (2R\eta - \delta)(R\eta - \delta)(-2R\eta - \delta) + \text{higher order terms.}$$

We note that we get the above expression (different from results obtained in Section 8.2.1.1) because when $z = 1$, $\kappa = 1$ is a triple root of equation (8.2.84). We also note that if a zero is due to either the first or second term, the root κ will approach 1 from the outside (when δ is positive, η will be positive). A zero due to the third term will produce a root κ that approaches 1 from the inside (when δ is positive, η is negative). We emphasize that this third root does not make $z = 1$ a generalized eigenvalue. We note that the "first, second and third zeros" correspond to κ_{1_1}, κ_{1_2} and κ_{1_3}, respectively. The form of equation (8.2.89) (the fact that $D(z) = 0$ only when $\kappa_{1_1} = 1$ or $\kappa_{1_2} = 1$) makes it clear that the only two roots we care about are the first two. The values $z = 1$, $\kappa_{1_3} = 1$ (without having $\kappa_{1_1} = 1$ or $\kappa_{1_2} = 1$) does not correspond to a solution to equations (8.2.80)–(8.2.81). Since both of these roots approach 1 in κ from the outside, they do not produce a generalized eigenvalue of equations (8.2.80), (8.2.81), and *difference scheme (8.2.73)–(8.2.75) is unconditionally stable.*

HW 8.2.4 Show that for the Crank-Nicolson difference scheme (8.2.73) with an arbitrary diagonalizable matrix A, the vectors that are in the null space of the coefficient matrix of resolvent equation (8.2.82) (or more generally (8.2.67)) are the eigenvectors of the matrix A.

We did a lot of work in the above example to show that an obviously stable scheme was stable. The point is that we must proceed very carefully when we consider systems of difference equations and we hope that the above example will serve as a model as we proceed to do more difficult stability analyses. We next consider the same difference scheme for a more difficult system of partial differential equations.

Example 8.2.8 Discuss the stability of the Crank-Nicolson difference scheme

$$\mathbf{u}_k^{n+1} - \frac{R}{4}A\delta_0\mathbf{u}_k^{n+1} = \mathbf{u}_k^n + \frac{R}{4}A\delta_0\mathbf{u}_k^n, \quad k = 1, \ldots \tag{8.2.93}$$

where

$$A = \begin{pmatrix} 5/3 & 1 & 5/3 \\ -1/3 & -1 & -7/3 \\ 1/3 & -1 & 1/3 \end{pmatrix}, \tag{8.2.94}$$

along with boundary conditions

$$u_{1_0}^{n+1} = u_{1_1}^{n+1} \tag{8.2.95}$$

$$u_{2_0}^{n+1} = u_{2_1}^{n+1} \tag{8.2.96}$$

$$u_{3_0}^{n+1} = g_3^{n+1} \tag{8.2.97}$$

$(\mathbf{u}_k^{n+1} = [u_{1_k}^{n+1}\ u_{2_k}^{n+1}\ u_{3_k}^{n+1}]^T)$.

Solution: We begin by noting that the differences between difference equations (8.2.93)–(8.2.97) and difference equations (8.2.73)–(8.2.76) are that we now have a full matrix A and one different boundary condition. Difference equation (8.2.93) is the result of using the Crank-Nicolson scheme to discretize the system of partial differential equations

$\mathbf{v}_t = A\mathbf{v}_x$, where A is as given above. Since the matrix A has one negative eigenvalue (-2), we get one mathematical boundary condition at $x = 0$. This is discretized as equation (8.2.97). To see that this is an acceptable boundary condition for the above system of partial differential equations, see HW6.3.4(c). As in the last example, since we have chosen a difference scheme that reaches in both directions in all variables, we must provide two numerical boundary conditions at $x = 0$. We have chosen numerical boundary conditions of the form (8.2.95) and (8.2.96) because those are the type of numerical boundary conditions that worked in Examples 8.2.4 and 8.2.7, and because these are boundary conditions that are commonly used.

We now proceed as we did in the last example, taking the discrete Laplace transform of difference equation (8.2.93) and the homogeneous boundary conditions. If we proceed in this manner, we see that the resolvent equation and characteristic equation are exactly the same as those given in (8.2.82) and (8.2.83), except that the matrix A is now as given above. Solving the characteristic equation (and using Maple makes this job easy) gives the same characteristic roots as are given by (8.2.85)–(8.2.87). This can be easily seen if we realize that $S^{-1}AS$ is the diagonal matrix in Example 8.2.7. See HW8.2.5. The general solution to the resolvent equation is given by

$$\tilde{\mathbf{u}}_k = c_1\kappa_{1_1}^k \mathbf{u}_1 + c_2\kappa_{1_2}^k \mathbf{u}_2 + c_2\kappa_{1_3}^k \mathbf{u}_3 \tag{8.2.98}$$

where $\mathbf{u}_1$, $\mathbf{u}_2$ and $\mathbf{u}_3$ are the vectors in the null space of the coefficient matrix in equation (8.2.82) (with the matrix A given by (8.2.94)) and are also the eigenvectors of the matrix A.

The homogeneous transformed boundary conditions can be written as

$$\tilde{\mathbf{u}}_0 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tilde{\mathbf{u}}_1. \tag{8.2.99}$$

If we insert solution (8.2.98) into equation (8.2.99) (and use the fact that the eigenvectors of A are given by $\mathbf{u}_1 = [2\ \ -1\ \ 1]^T$, $\mathbf{u}_2 = [-1\ \ -1\ \ 1]^T$ and $\mathbf{u}_3 = [-1\ \ 2\ \ 1]^T$), we obtain the following system of equations for the coefficients c_1, c_2 and c_3

$$D(z)\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{pmatrix} 2-2\kappa_{1_1} & -1+\kappa_{1_2} & -1+\kappa_{1_3} \\ -1+\kappa_{1_1} & -1+\kappa_{1_2} & 2-2\kappa_{1_3} \\ 1 & 1 & 1 \end{pmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \tag{8.2.100}$$

We again note that this is a special case of equation (8.2.71). There exists a nontrivial solution to equation (8.2.100) if the determinant of the coefficient matrix is zero. Again, letting an algebraic manipulator do the work, we see that there are nontrivial solutions to equation (8.2.100) if κ_{1_1}, κ_{1_2} and κ_{1_3} satisfy

$$\frac{2}{3}\kappa_{1_1} + \frac{2}{3}\kappa_{1_2} + \frac{2}{3}\kappa_{1_3} - \frac{1}{3}\kappa_{1_1}\kappa_{1_2} - \frac{1}{3}\kappa_{1_1}\kappa_{1_3} - \frac{1}{3}\kappa_{1_2}\kappa_{1_3} = 1. \tag{8.2.101}$$

Obviously, this relationship is much more complex than the ones we have dealt with in the past. We must use this relationship to help us determine whether there are any eigenvalues or generalized eigenvalues. It doesn't appear to be possible (or at least obvious how to proceed) to determine analytically whether equation (8.2.101) has any solutions for either $|z| > 1$ or $|z| = 1$. You can look for solutions using an algebraic manipulator, but when you find no solutions, you really do not have enough confidence in the results of the software to conclude that there are no eigenvalues or generalized eigenvalues.

Another approach is to try to use graphics. This technique should be used, but it must be approached with much care. Three reasonably obvious plots to use are the plots of the magnitude of the left hand side of equation (8.2.101) with z replaced by $z = e^{i\theta}$, $z = r$, $r \geq 1$ and $z = r$, $r \leq -1$, respectively. If you plot these functions, there are times that from certain perspectives there appears to be a solution. However, if you look more

closely (zoom in on the alleged solution), you will find that there seem to be no solutions for $|z| = 1$, $z = r \geq 1$ and $z = r \leq -1$.

There is a variety of ways to look further. One way is to look for solutions along rays $z = re^{i\theta_0}$ for a systematic choice of values of θ_0. Another approach is to look for solutions along circles of the form $z = r_0 e^{i\theta}$, $0 \leq \theta \leq 2\pi$, for a systematic choice of values of r_0. All of these plots are difficult to interpret. These plots generally indicate that there are no solutions to equation (8.2.101) but do not make it clear that there are no solutions. See HW8.2.6. Obviously, *none of these plots (or the fact that Maple could not find a solution to equation (8.2.100)) prove that the scheme is stable.* However, this evidence should give us confidence that the scheme might be stable.

We should also add that if the algebra or plots did produce a potential eigenvalue or generalized eigenvalue, then it might be possible to work to find the z value and possibly prove that the scheme is unstable. Generally, the computer algebra and plots will be more useful for proving that a scheme is unstable than for proving that a scheme is stable. When computer algebra or plots are used on a stable scheme, the results will only indicate that the scheme appears to be stable. However, at times, we must use the best information that we are able to obtain.

HW 8.2.5 Verify that if $S^{-1}AS = D$, then

$$\det\left[(z+1)\frac{R}{4}A\kappa^2 - (z-1)I\kappa - (z+1)\frac{R}{4}A\right] = \det\left[(z+1)\frac{R}{4}D\kappa^2 - (z-1)I\kappa - (z+1)\frac{R}{4}D\right].$$

HW 8.2.6 Plot the left hand side of equation (8.2.101) in the following three ways to verify that there appear to be no solutions to this equation equal to one.
(a) Plot (8.2.101) with $z = r$ ($r \geq 1$ and $r \leq 1$) and $z = e^{i\theta}$.
(b) Plot (8.2.101) with $z = re^{i\theta_0}$ for $r \geq 1$ and $\theta_0 = 0, \pi/4, \pi/2, \ldots, 7\pi/4$.
(c) Plot (8.2.101) with $z = r_0 e^{i\theta}$ for $0 \leq \theta \leq 2\pi$ and $r = 1, 2, 3, 4, 5$.

As we saw in the last example, it is not always easy or possible to prove that a given scheme for an initial–boundary value problem is either stable or unstable. However, part of the problem is that the numerical boundary conditions (8.2.95)–(8.2.96) were not chosen wisely. Though these are the numerical boundary conditions that we used often for scalar equations and that we used in Example 8.2.7, the system of partial differential equations used in Example 8.2.8 is much more complicated than the systems solved in the other cases. If we transform difference scheme (8.2.93) to characteristic coordinates using the transformation $\mathbf{U}_k^n = S\mathbf{u}_k^n$ where S is the inverse of the matrix

$$S^{-1} = \begin{pmatrix} 2 & -1 & -1 \\ -1 & -1 & 2 \\ 1 & 1 & 1 \end{pmatrix} \tag{8.2.102}$$

(S^{-1} is the matrix of eigenvectors of A), we get

$$\mathbf{U}_k^{n+1} - \frac{R}{4} D\delta_0 \mathbf{U}_k^{n+1} = \mathbf{U}_k^n + \frac{R}{4} D\delta_0 \mathbf{U}_k^n, \quad k = 1, \ldots \tag{8.2.103}$$

where

$$D = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -2 \end{pmatrix}.$$

We note that this is the same scheme that was used in Example 8.2.7. If we transform the numerical boundary conditions (8.2.95)–(8.2.96) and the boundary condition (8.2.97) in the same manner, we get

$$\mathbf{U}_0^{n+1} = \begin{pmatrix} 2/3 & -1/3 & -1/3 \\ -1/3 & 2/3 & -1/3 \\ -1/3 & -1/3 & 2/3 \end{pmatrix} \mathbf{U}_1^{n+1} + \frac{1}{3} \begin{bmatrix} 0 \\ 0 \\ g_3^{n+1} \end{bmatrix}. \tag{8.2.104}$$

It seems clear that if we were considering using difference scheme (8.2.103) to solve a system of partial differential equations with coefficient matrix D, we would not choose the numerical boundary conditions included in (8.2.104). In addition, it is not at all clear to what sort of analytic boundary condition the third equation corresponds (if any). We close this section with the following example.

Example 8.2.9 Discuss the stability of the Crank-Nicolson difference scheme

$$\mathbf{u}_k^{n+1} - \frac{R}{4} A\delta_0 \mathbf{u}_k^{n+1} = \mathbf{u}_k^n + \frac{R}{4} A\delta_0 \mathbf{u}_k^n, \quad k = 1, \ldots \tag{8.2.105}$$

where A is as given by equation (8.2.94), along with boundary conditions

$$u_{1_0}^{n+1} = u_{1_1}^{n+1} + (u_{3_1}^{n+1} - u_{3_0}^{n+1}) \tag{8.2.106}$$

$$u_{2_0}^{n+1} = u_{2_1}^{n+1} - 2(u_{3_1}^{n+1} - u_{3_0}^{n+1}) \tag{8.2.107}$$

$$u_{3_0}^{n+1} = g_3^{n+1} \tag{8.2.108}$$

($\mathbf{u}_k^{n+1} = [u_{1_k}^{n+1} \; u_{2_k}^{n+1} \; u_{3_k}^{n+1}]^T$).

Solution: Obviously, since this example is very similar to the last example (same difference equation and analytic boundary condition, different numerical boundary conditions), much of the work that we did on Example 8.2.8 will carry over to this example. Specifically, we will have the same resolvent equation (equation (8.2.80) with A given by (8.2.94)), characteristic equation (equation (8.2.83) with A given by (8.2.94)), and characteristic roots (equations (8.2.85)–(8.2.87)). Hence, the general solution to the resolvent equation will again be given by equation (8.2.98).

The difference between this example and the last example is obviously in the boundary conditions. Though it is possible, it is reasonably difficult to rewrite boundary conditions (8.2.106)–(8.2.108) in the form of equation (8.2.63) (so that we can use equation (8.2.72) to write $D(z)$). Instead, we transform equations (8.2.106)–(8.2.108) and write the homogeneous transformed boundary conditions as

$$\tilde{u}_0 = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & -2 \\ 0 & 0 & 0 \end{pmatrix} \tilde{u}_1 + \begin{pmatrix} 0 & 0 & -1 \\ 0 & 0 & 2 \\ 0 & 0 & 0 \end{pmatrix} \tilde{u}_0. \tag{8.2.109}$$

Substituting the form of $\tilde{u}_k$ given by equation (8.2.98) in equation (8.2.109), we see that equation (8.2.109) is equivalent to

$$\begin{pmatrix} 3-3\kappa_{l_1} & 0 & 0 \\ -3+3\kappa_{l_1} & -3+3\kappa_{l_2} & 0 \\ 1 & 1 & 1 \end{pmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \tag{8.2.110}$$

Obviously equation (8.2.110) has a nontrivial solution when the determinant of the coefficient matrix is zero, i.e., when $\kappa_{l_1} = 1$ or when $\kappa_{l_2} = 1$. Since when $|z| > 1$, both κ_{l_1} and κ_{l_2} satisfy, $|\kappa| < 1$, there are no eigenvalues. Using exactly the same calculations used in Example 8.2.7 (either the form of κ_{l_1} and κ_{l_2} when $z = e^{i\theta}$ or the perturbation calculation), we see that neither κ_{l_1} nor κ_{l_2} (along with $z = 1$) corresponds to generalized eigenvalues. Therefore, *difference scheme (8.2.105), along with numerical boundary conditions (8.2.106)-(8.2.107) and analytic boundary condition (8.2.108), is unconditionally stable.*

Remark: It is reasonably clear, we hope, that the scheme considered in Example 8.2.9 is the same as that considered in Example 8.2.7. The scheme considered in Example 8.2.7 can be considered as the characteristic version of the scheme considered in terms of primitive variables in Example 8.2.9. If we want to work with primitive variables, the correct way to choose numerical boundary conditions is to choose numerical boundary conditions for the characteristic variables that we know will be stable and transform these stable, characteristic numerical boundary conditions to the primitive variables. Then the scheme considered in primitive variables will also be stable. Specifically, in Example 8.2.9, we obtained numerical boundary conditions (8.2.106)–(8.2.107) by transforming numerical boundary conditions (8.2.74)–(8.2.75) using the transformation defined by S^{-1}. As we saw in Example 8.2.8, it may be acceptable to naively choose nice boundary conditions in terms of the primitive variables, but we were unable to concretely prove that the resulting scheme was stable.

8.2.2.2 Multilevel Schemes

In Section 6.4 we saw that one of the approaches to the stability of multilevel schemes is to reduce the scheme to a two level system. As we shall see in Section 8.2.3, the general GKSO theory includes multilevel schemes. We include the treatment of multilevel schemes as a system both because the approach is consistent with the approach introduced in Chapter 6 for the study of the stability and convergence of multilevel schemes for initial–value problems and because we do not have a Lax Theorem that applies to multilevel schemes. And sooner or later, when we talk about stability, we will return to some version of the Lax Theorem. The general GKSO theory that we consider in Section 8.2.3 will be sufficiently general to include multilevel schemes. In Section 8.4.3 we state a theorem due to Gustafsson that will be applicable to multilevel schemes.

We introduce the stability of initial–boundary–value, multilevel schemes by considering a specific difference scheme, the leapfrog scheme. We con-

sider applying the leapfrog scheme to the right quarter plane problem

$$v_t + av_x = 0, \quad x \in (0,\infty),\ t > 0 \tag{8.2.111}$$

$$v(x,0) = f(x), \quad x \in [0,\infty) \tag{8.2.112}$$

where a is assumed to be less than zero. Because $a < 0$ and the leapfrog scheme reaches in both directions, we must prescribe a numerical boundary condition at $x = 0$. To illustrate how the stability of such a multilevel scheme can be analyzed, we consider the following example.

Example 8.2.10 Consider the stability of the scheme consisting of

$$u_k^{n+1} = u_k^{n-1} - R\delta_0 u_k^n, \quad k = 1,\ldots \tag{8.2.113}$$

$$u_k^0 = f(k\Delta x), \quad k = 0,\ldots \tag{8.2.114}$$

along with the numerical boundary condition

$$u_0^{n+1} = u_1^{n+1}. \tag{8.2.115}$$

Solution: We proceed as we did in Section 6.4.1, set

$$\mathbf{U}_k^n = \begin{bmatrix} u_k^n \\ u_k^{n-1} \end{bmatrix} \tag{8.2.116}$$

and rewrite difference scheme (8.2.113) as

$$\begin{aligned} \mathbf{U}_k^{n+1} &= \begin{pmatrix} -R\delta_0 & 1 \\ 1 & 0 \end{pmatrix} \mathbf{U}_k^n \\ &= \begin{pmatrix} R & 0 \\ 0 & 0 \end{pmatrix} \mathbf{U}_{k-1}^n + \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \mathbf{U}_k^n + \begin{pmatrix} -R & 0 \\ 0 & 0 \end{pmatrix} \mathbf{U}_{k+1}^n. \end{aligned} \tag{8.2.117}$$

If we consider the form of the numerical boundary condition (8.2.115) along with the form of our vector defined in (8.2.116), we see that we can write the numerical boundary condition as

$$\mathbf{U}_0^{n+1} = \mathbf{U}_1^{n+1}. \tag{8.2.118}$$

Thus, we consider the stability of difference scheme (8.2.113)–(8.2.115) by considering the stability of difference scheme (8.2.117)–(8.2.118).

As we did in Section 8.2.2.1, we take the discrete Laplace transform of equations (8.2.117) and (8.2.118) to get the resolvent equation

$$z\tilde{\mathbf{U}}_k = \begin{pmatrix} R & 0 \\ 0 & 0 \end{pmatrix} \tilde{\mathbf{U}}_{k-1} + \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tilde{\mathbf{U}}_k + \begin{pmatrix} -R & 0 \\ 0 & 0 \end{pmatrix} \tilde{\mathbf{U}}_{k+1} \tag{8.2.119}$$

and the transformed boundary condition

$$\tilde{\mathbf{U}}_0 = \tilde{\mathbf{U}}_1. \tag{8.2.120}$$

If we look for solutions of equation (8.2.119) of the form $\tilde{\mathbf{U}}_k = \kappa^k \boldsymbol{\phi}$, we arrive at the characteristic equation

$$\det \begin{pmatrix} -R\kappa^2 - z\kappa + R & \kappa \\ \kappa & -z\kappa \end{pmatrix} = 0, \tag{8.2.121}$$

or

$$zR\kappa^3 + (z^2 - 1)\kappa^2 - zR\kappa = 0. \tag{8.2.122}$$

Since we know that for $|z| > 1$, one root satisfies $|\kappa| < 1$ (which we call κ_1), one root satisfies $|\kappa| > 1$ (which we call κ_2) and one root $\kappa = 0$ is irrelevant, we look for solutions of the form $\bar{\mathbf{U}}_k = \kappa_1^k \boldsymbol{\phi}$. Inserting this expression into equation (8.2.120) yields $\boldsymbol{\phi} = \kappa_1 \boldsymbol{\phi}$, or $\kappa_1 = 1$. Hence, *we know that there are no eigenvalues of equations (8.2.119), (8.2.120) for this problem.* We must determine whether there are any potential generalized eigenvalues, i.e., whether κ_1 ever satisfies $|\kappa_1| = 1$ for any z satisfying $|z| = 1$. Putting $\kappa = 1$ into equation (8.2.122) implies that $z = \pm 1$. Hence, we must determine whether $\kappa = 1$ is approached from the inside for either $z = \pm 1$.

If we set $z = 1 + \delta$, $\kappa = 1 + \eta$, insert these expressions into equation (8.2.122) and simplify, we see that $\delta = -R\eta +$ higher order terms. Since $R < 0$, as z approaches 1 from the outside ($\delta > 0$), κ approaches 1 from the outside ($\eta > 0$). Hence, $z = 1$ is not a generalized eigenvalue of equations (8.2.119), (8.2.120).

Setting $z = -1 + \delta$, $\kappa = 1 + \eta$, inserting these expressions into equation (8.2.122) and simplifying, we get $\delta = -R\eta +$ higher order terms. Since $R < 0$, as z approaches -1 from the outside ($\delta < 0$), κ approaches 1 from the inside ($\eta < 0$). Hence, $z = -1$ *is a generalized eigenvalue of equations (8.2.119), (8.2.120), and the scheme is unstable.*

Remark: We should note that characteristic equation (8.2.122) has only three roots instead of the usual four, and one of the three roots is trivial (irrelevant). The reason that this characteristic equation has a different number of roots than in the normal case is because of the approach used to rewrite the three level scalar scheme as a two level system. We introduce an artificial difference equation into the system, $U_{1_k}^{n+1} = U_{1_k}^n$. The trivial nature of this difference equation (there are no differences with respect to k) is what causes characteristic equation (8.2.122) to have only two relevant roots rather than the usual four.

Thus, we see that if we use the leapfrog scheme along with a very nice and common numerical boundary condition, we have an unstable scheme. You might recall that numerical boundary condition (8.2.115) was one of the boundary conditions considered in HW6.4.2. We should have found in HW6.4.2 that numerical boundary condition (8.2.115) was unstable when used with the leapfrog scheme. If not, we suggest returning to study the results of that problem or to work that problem for the first time. In the next example, we see that we are able to find numerical boundary conditions that are stable when used with the leapfrog scheme.

Example 8.2.11 Consider the stability of the scheme consisting of difference equations (8.2.113)–(8.2.114) along with the numerical boundary condition

$$u_0^{n+1} = u_0^n - R(u_1^n - u_0^n). \tag{8.2.123}$$

Solution: We note that numerical boundary condition (8.2.123) can viewed as using a one sided difference scheme to solve for u_0^{n+1}.

Proceeding as we did in Example 8.2.10, we obtain the same resolvent equation, (8.2.119), and the same characteristic equation, (8.2.122), which for $|z| > 1$ has roots κ_1 satisfying $|\kappa_1| < 1$, κ_2 satisfying $|\kappa_2| > 1$ and $\kappa = 0$. Also, as in Example 8.2.10, we assume that the numerical boundary condition holds for both n and $n+1$ and write our vector form of the numerical boundary condition (8.2.123) as

$$\mathbf{U}_0^{n+1} = \begin{pmatrix} 1+R & 0 \\ 0 & 1+R \end{pmatrix} \mathbf{U}_0^n + \begin{pmatrix} -R & 0 \\ 0 & -R \end{pmatrix} \mathbf{U}_1^n. \tag{8.2.124}$$

Transforming this boundary condition gives

$$z\tilde{\mathbf{U}}_0 = \begin{pmatrix} 1+R & 0 \\ 0 & 1+R \end{pmatrix} \tilde{\mathbf{U}}_0 + \begin{pmatrix} -R & 0 \\ 0 & -R \end{pmatrix} \tilde{\mathbf{U}}_1. \tag{8.2.125}$$

Substituting $\tilde{\mathbf{U}}_k = \kappa_1^k \boldsymbol{\phi}$ into equation (8.2.125) yields

$$z\boldsymbol{\phi} = \begin{pmatrix} 1+R & 0 \\ 0 & 1+R \end{pmatrix} \boldsymbol{\phi} + \begin{pmatrix} -R & 0 \\ 0 & -R \end{pmatrix} (\kappa_1 \boldsymbol{\phi}),$$

or

$$\begin{pmatrix} z-(1+R)+R\kappa_1 & 0 \\ 0 & z-(1+R)+R\kappa_1 \end{pmatrix} \boldsymbol{\phi} = \boldsymbol{\theta}. \tag{8.2.126}$$

And of course, equation (8.2.126) has a nontrivial solution only if the determinant of the coefficient matrix is zero, or

$$[z-(1+R)+R\kappa_1]^2 = 0. \tag{8.2.127}$$

Solving equation (8.2.122) along with $z-(1+R)+R\kappa_1 = 0$ (with a little help from Maple), we find that either $\kappa_1 = 0$ (which we ignore) or $\kappa_1 = z = 1$. From the computation done in Example 8.2.10, we know that it is impossible to have $\kappa_1 = z = 1$ (the solution is really $\kappa_2 = z = 1$). Hence, since there are no eigenvalues and no generalized eigenvalues of equations (8.2.119), (8.2.124), difference scheme (8.2.113)–(8.2.114), (8.2.123) is stable for $-1 < R_0 \leq R \leq 0$ (the stability conditions are the conditions that the scheme inherits from the assumption that difference scheme (8.2.113) must be stable as an initial–value problem scheme).

In addition to numerical boundary condition (8.2.123), there are other numerical boundary conditions that are stable when considered along with the leapfrog scheme (8.2.113).

- Difference equation (8.2.113) along with the numerical boundary condition

$$u_0^{n+1} + u_1^{n+1} + R\delta_+ u_0^{n+1} = u_0^n + u_1^n - R\delta_+ u_0^n \tag{8.2.128}$$

  is stable for $-1 < R_0 \leq R \leq 0$.

- Difference equation (8.2.113) along with the numerical boundary condition

$$^-\delta_+^j u_0^{n+1} = 0 \tag{8.2.129}$$

  is stable for $-1 < R_0 \leq R \leq 0$ (where as before, $^-\delta_+$ is defined as $^-\delta_+ u_0^{n+1} = u_1^n - u_0^{n+1}$).

Also, we can also obtain a general result that includes the result of Example 8.2.10.

- Difference equation (8.2.113) along with the numerical boundary condition

$$\delta_+^j u_0^{n+1} = 0 \tag{8.2.130}$$

  is unstable.

8.2.3 GKSO: General Theory

We are finally ready to present the general GKSO theory. In presenting the two "easy cases" first, we have included most of the schemes for initial–boundary–value problems that we want to study. One new situation that we will study with the general theory is the stability of some of the higher order initial–boundary–value problem schemes that reach more than one point to the right or left. The general theory will also include multilevel schemes without requiring that we rewrite the scheme as a system. As we have done before, we will present the material for a right quarter plane problem. As we saw in Section 8.2.2, when we consider systems of difference equations for solving problems involving systems of partial differential equations, our boundary conditions become a mixture of analytic boundary conditions, and numerical boundary conditions and the right and left quarter plane problems become very similar.

We begin by considering a difference equation of the form

$$Q_{-1}\mathbf{u}_k^{n+1} = \sum_{j=0}^{s} Q_j \mathbf{u}_k^{n-j} + \Delta t \mathbf{G}_k^n \tag{8.2.131}$$

where $\mathbf{u}_k^n$ is a K–vector, Q_j is a difference operator defined by

$$Q_j = \sum_{m=-r}^{p} A_{j\,m} S_+^m,$$

$A_{j\,m}$ are $K \times K$ matrices, and S_+ is the shift operator $S_+ \mathbf{u}_k = \mathbf{u}_{k+1}$ as defined in Section 6.2. We make special note that difference equation (8.2.131) is an $s+2$-level scheme that reaches r grid points to the left and p grid points to the right. We consider difference equation (8.2.131) along with the initial condition

$$\mathbf{u}_k^m = \mathbf{f}_k^m, \quad m = 0, 1, \ldots, s, \quad k = -r+1, -r+2, \ldots \tag{8.2.132}$$

and boundary conditions

$$\mathbf{u}_k^{n+1} = \sum_{m=-1}^{s} S_k^m \mathbf{u}_1^{n-m} + \mathbf{g}_k^n, \quad k = -r+1, \ldots, 0 \tag{8.2.133}$$

where

$$S_k^m = \sum_{j=0}^{q} C_{j\,k}^m S_+^j$$

(again, $C_{j\,k}^m$ are $K \times K$ matrices).

We note that in (8.2.132) we have provided $s+1$ levels of initial conditions to enable the $s+2$-level difference equation (8.2.131) to get started.

Thus we are assuming that some sort of initialization scheme has been provided. For a discussion of initialization schemes for multilevel schemes, see Section 6.4.2. In addition, note that these initial values are provided on the left out to $k = -r + 1$ so as to initialize the scheme on the $r - 1$ ghost points that are necessary due to the fact that difference equation (8.2.131) reaches r grid points to the left. And finally, the scheme includes r boundary conditions of the form of (8.2.133), which are also necessary due to the fact that difference equation (8.2.131) reaches r grid points to the left. We should be aware that the boundary conditions (8.2.133) will generally be a combination of analytic boundary conditions (equal to the number of negative eigenvalues of the coefficient matrix of the system of partial differential equations), boundary conditions derived from Taylor series expansions of the analytic boundary conditions (which will be illustrated in Example 8.2.14) and numerical boundary conditions.

The definition of stability of the above difference scheme (Definition 3.3, [19], page 654) is the logical generalization of Definition 8.2.1. As in the previous sections, the approach that we use is to take the discrete Laplace transform of equations (8.2.131) and (8.2.133) and obtain the resolvent equation

$$\left(Q_{-1} - \sum_{j=0}^{s} z^{-j-1} Q_j \right) \tilde{\mathbf{u}}_k = \boldsymbol{\theta} \tag{8.2.134}$$

(which is the homogeneous version of the Laplace transform of equation (8.2.131)) and the homogeneous transformed boundary condition

$$\tilde{\mathbf{u}}_k - \sum_{m=-1}^{s} z^{-(m+1)} S_k^m \tilde{\mathbf{u}}_1 = \boldsymbol{\theta}. \tag{8.2.135}$$

Before we can state the theorems relating solutions of equations (8.2.134)–(8.2.135) to the stability of difference scheme (8.2.131)–(8.2.133), we emphasize that *we must again satisfy Assump 8.1–Assump 8.4*. Of course, in this case, Assump 8.2 involves solving equations (8.2.131), (8.2.133) boundedly for $\mathbf{u}_k^{n+1}$, and Assump 8.4 requires that the matrices $A_{j\,m}$, $j = 0, \ldots, s$, $m = -r, \ldots, p$ be simultaneously diagonalizable.

Because of the lack of depth to which we consider the general theory, we will not see how the above assumptions are used or why they are necessary. For further discussions of these assumptions, some alternatives to these assumptions and some of the consequences of these assumptions, see ref. [19] or ref. [18]. These assumptions do not eliminate the consideration of any of the standard difference schemes. One of the very important assumptions is the requirement that the scheme be stable when considered as an initial–value problem scheme. This assumption often gives us a necessary condition for stability that we must include with any stability conditions that we later impose.

As we did in the "easy cases," we must define the concepts of when a complex number z is an eigenvalue or generalized eigenvalue of equations (8.2.134)–(8.2.135).

Definition 8.2.17 *The complex number z, $|z| > 1$, is an eigenvalue of equations (8.2.134)–(8.2.135) if*
(1) *there exists a vector $\tilde{\mathbf{u}} = [\tilde{\mathbf{u}}_{-r+1}\ \tilde{\mathbf{u}}_{-r+2}\ \cdots]^T$ such that $(z, \tilde{\mathbf{u}})$ satisfies equations (8.2.134)–(8.2.135), and*
(2) $\|\tilde{\mathbf{u}}\|_2 < \infty$.

The complex number z is a generalized eigenvalue of equations (8.2.134)–(8.2.135)
(1) *if there exists a vector $\tilde{\mathbf{u}}$ such that $(z, \tilde{\mathbf{u}})$ satisfies equations (8.2.134)–(8.2.135),*
(2) $|z| = 1$, *and*
(3) $\tilde{\mathbf{u}}_k = \lim_{w \to z,\ |w|>1} \tilde{\mathbf{u}}(w)$, *where $(w, \tilde{\mathbf{u}}(w))$ is a solution to equation (8.2.134).*

We can then state the **Ryabenkii-Godunov condition** and the analogue to Proposition 8.2.13 as follows.

Theorem 8.2.18 *If equations (8.2.134)–(8.2.135) have an eigenvalue z, then difference scheme (8.2.131)–(8.2.133) is unstable.*

Theorem 8.2.19 *Difference scheme (8.2.131)–(8.2.133) is stable if and only if eigenvalue problem (8.2.134)–(8.2.135) has no eigenvalues and no generalized eigenvalues.*

To use the above theorems we must, as we did in the "easy cases," determine when we can solve equations (8.2.134)–(8.2.135). Since equations (8.2.134)–(8.2.135) are a set of difference equations in k, we look for solutions of the form $\tilde{\mathbf{u}}_k = \kappa^k \boldsymbol{\phi}$ where $\boldsymbol{\phi}$ is a K-vector. Inserting this expression into equation (8.2.134), we see that

$$\begin{aligned}
\boldsymbol{\theta} &= \left(Q_{-1} - \sum_{j=0}^{s} z^{-j-1} Q_j \right) \kappa^k \boldsymbol{\phi} \\
&= \left(\sum_{m=-r}^{p} A_{-1\,m} S_+^m - \sum_{j=0}^{s} z^{-j-1} \sum_{m=-r}^{p} A_{j\,m} S_+^m \right) \kappa^k \boldsymbol{\phi} \\
&= \left[\sum_{m=-r}^{p} \left(A_{-1\,m} - \sum_{j=0}^{s} z^{-j-1} A_{j\,m} \right) \kappa^{k+m} \right] \boldsymbol{\phi}. \qquad (8.2.136)
\end{aligned}$$

Thus κ must be a solution of the **characteristic equation**

$$0 = \det \left[\sum_{m=-r}^{p} \left(A_{-1\,m} - \sum_{j=0}^{s} z^{-j-1} A_{j\,m} \right) \kappa^m \right]. \qquad (8.2.137)$$

Equation (8.2.137) will be a polynomial of degree less than or equal to $K(p+r+1)$. However, as in the "easy cases," equation (8.2.137) will have the correct number of solutions satisfying $|\kappa| < 1$. We obtain the following result.

Proposition 8.2.20 *For $|z| > 1$ characteristic equation (8.2.137) has no solution that satisfies $|\kappa| = 1$. Also, for $|z| > 1$, equation (8.2.137) will have Kr roots (counting multiplicity) that satisfy $|\kappa| < 1$.*

When the Kr roots of equation (8.2.137) are distinct, the general solution of equation (8.2.134) can be written as

$$\tilde{\mathbf{u}}_k = \sum_{j=1}^{Kr} \kappa_j^k \boldsymbol{\phi}_j \tag{8.2.138}$$

where $|\kappa_j| < 1$, $j = 1, \ldots, Kr$, and the vectors $\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_{Kr}$ contain Kr free parameters (i.e., some of the $K(Kr)$ components of these vectors are determined in terms of the other components). If equation (8.2.137) has repeated roots satisfying $|\kappa| < 1$, the general solution can be written as

$$\tilde{\mathbf{u}}_k = \sum_{|\kappa_j|<1} \mathbf{P}_j(k)\kappa_j^k \tag{8.2.139}$$

where $\mathbf{P}_j(k)$ are polynomials in k of degree one less than the multiplicity of κ_j with K–vector coefficients. As in the previous case, the polynomials contain Kr free parameters.

Thus, in either case, the general solution to equation (8.2.134) depends on Kr free parameters. These parameters must be determined by the boundary conditions (8.2.135). We first note that we have the right number of boundary conditions (r K–vector conditions) to determine the rK parameters. As in the "easy cases," the approach we use is to determine whether there is a nontrivial solution in the form of either equation (8.2.138) or (8.2.139) that satisfies boundary conditions (8.2.135), and if so, determine whether these solutions determine either an eigenvalue or generalized eigenvalue satisfying Definition 8.2.17.

It is clear, we hope, that the above approach was followed in all of the examples considered in Sections 8.2.1.1, 8.2.2.1 and 8.2.2.2. In Section 8.2.1.1, we always had $r = p = 1$, $s = 0$ and $K = 1$; in Section 8.2.2.1, we had $r = p = 1$, $s = 0$ and $K = 3$; and in Section 8.2.2.2, we had $r = p = 1$, $s = 1$ and $K = 1$ (but we proved stability by rewriting the scheme as a system with $r = p = 1$, $s = 0$ and $K = 2$).

We should also note that as in the "easy cases," we can write the solution (8.2.138) as

$$\tilde{\mathbf{u}}_k = \sum_{j=1}^{Kr} c_j \kappa_j^k \mathbf{u}_j \tag{8.2.140}$$

where $\mathbf{u}_j$, $j = 1, \ldots, Kr$, are the vectors in the null space of the coefficient matrix of equation (8.2.136) associated with $\kappa = \kappa_{1_j}$, $j = 1, \ldots, Kr$, respectively. Insertion of solution (8.2.140) into the homogeneous transformed boundary conditions (8.2.135) leaves us again with a system of the form

$$D(z) \begin{bmatrix} c_1 \\ \vdots \\ c_{Kr} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \tag{8.2.141}$$

where the j–th column of the matrix $D(z)$ is given by

$$\left\{ \kappa_j^k I - \sum_{m=-1}^{s} z^{-(m+1)} \sum_{\ell=0}^{q} C_{\ell k}^m \kappa_j^{\ell+1} \right\} \mathbf{u}_j, \quad k = -r+1, \ldots, 0. \tag{8.2.142}$$

We obtain the following analogue to Propositions 8.2.11 and 8.2.16.

Proposition 8.2.21 *If* $\det D(z) \neq 0$ *for all* z, $|z| \geq 1$, *then difference scheme (8.2.131)-(8.2.133) is stable.*

Also, as we have noted earlier, if $D(z) = 0$ for some $|z| > 1$, we know that the difference scheme is unstable. If $D(z) = 0$ for some $|z| = 1$, then the stability of the scheme rests on whether or not z is a generalized eigenvalue.

We next include two examples where we obtain stability results by using the general theory. We include one example where we apply the general theory to a multilevel scheme (to show how different or similar this approach is to rewriting the multilevel scheme as a system) and one example where we apply the theory to a scheme that is fourth order accurate in space (a scheme that reaches further than one point to the right and the left). For examples with K larger than one, see Section 8.2.2.1.

Example 8.2.12 Consider the stability of the scheme consisting of the leapfrog scheme, (8.2.113), along with numerical boundary condition (8.2.129).

Solution: If we write the leapfrog scheme as

$$u_k^{n+1} = u_k^{n-1} - R\delta_0 u_k^n, \tag{8.2.143}$$

it is easy to see that this equation can be written in the form of equation (8.2.131) if we

- choose $K = 1$, $s = 1$ and $p = r = 1$,
- let $Q_{-1} = 1$, $Q_0 = -R\delta_0$ and $Q_1 = 1$, and
- let $A_{0\,-1} = R$, $A_{0\,0} = 0$ and $A_{0\,1} = -R$.

Then we either use this notation in equation (8.2.134) or take the discrete Laplace transform of equation (8.2.143) to obtain the resolvent equation

$$\tilde{u}_k = z^{-2}\tilde{u}_k - Rz^{-1}\delta_0 \tilde{u}_k. \tag{8.2.144}$$

If we look for a solution of equation (8.2.144) of the form $\tilde{u}_k = \kappa^k$ (or use equation (8.2.137)), we obtain the characteristic equation

$$R\kappa^2 + (z - z^{-1})\kappa - R = 0. \tag{8.2.145}$$

If we compare equation (8.2.145) with equation (8.2.122), we see that the only difference is the zero root (which was never relevant anyway and was a figment of the artificial difference equation used to change the scalar leapfrog scheme into a two-level system of difference equations). Thus, the characteristic roots of interest are the same for both approaches.

Rewriting numerical boundary condition (8.2.129) as

$$u_0^{n+1} = -\sum_{m=1}^{j} (-1)^m \binom{j}{m} u_m^{n+1-m}. \tag{8.2.146}$$

we see that numerical boundary condition (8.2.129) can be expressed in the form of equation (8.2.133) if we

- choose $K = 1$, $r = 1$, $s = j - 1$ and $q = j$, and
- let $S_0^{-1} = 0$, $C_{m\,0}^{m} = (-1)^{m+1}\binom{j}{m}$, $m = 1, \ldots, j$ and $C_{k\,0}^{m} = 0$ otherwise.

We can then obtain the homogeneous transformed boundary equation by using equation (8.2.135) or by taking the discrete Laplace transform of equation (8.2.146). We get

$$\tilde{u}_0 = -\sum_{m=1}^{j} (-1)^m \binom{j}{m} z^{-m} \tilde{u}_m. \tag{8.2.147}$$

If we look for solutions to equation (8.2.147) in the form $\tilde{u}_k = \kappa^k \phi$, we get

$$\phi = -\sum_{m=1}^{j} (-1)^m \binom{j}{m} z^{-m} \kappa^m \phi,$$

which is the same as

$$(1 - \kappa/z)^j \phi = 0. \tag{8.2.148}$$

Hence, we see that the boundary condition implies that $\kappa = z$. Substituting $\kappa = z$ into equation (8.2.145) gives $z = \pm 1$. Hence, *there are no eigenvalues of equations (8.2.144), (8.2.147).*

To see whether there are any generalized eigenvalues of equations (8.2.144), (8.2.147), we substitute $z = \pm 1$ into equation (8.2.145) to find that $\kappa = \pm 1$. Since κ must equal z, we must consider $z = 1$, $\kappa = 1$ and $z = -1$, $\kappa = -1$. Returning to the calculation done in Example 8.2.10, we see that as z approaches 1 from the outside, κ approaches 1 from the outside, i.e., $z = 1$ *is not a generalized eigenvalue.*

Setting $z = -1 - \delta$, $\kappa = -1 - \eta$, inserting these expressions into equation (8.2.145) and simplifying, we get $\delta = -R\eta +$ higher order terms. Hence, $z = -1$ *is also not a generalized eigenvalue.* Therefore, *difference scheme (8.2.113) along with numerical boundary condition (8.2.129) is a stable difference scheme when* $-1 < R_0 \leq R < 0$ (which as usual is the condition inherited from the assumption that difference scheme (8.2.113) is stable as an initial-value scheme).

Example 8.2.13 Consider the stability of the difference equation

$$u_k^{n+1} = u_k^n - \frac{R}{2}\Big(1 - \frac{1}{6}\delta^2\Big)\delta_0 u_k^n + \frac{R^2}{2}\Big(\frac{4}{3} + R^2\Big)\delta^2 u_k^n - \frac{R^2}{8}\Big(\frac{1}{3} + R^2\Big)\delta_0^2 u_k^n, \quad k = 1, \ldots \tag{8.2.149}$$

along with the numerical boundary conditions

$$u_0^{n+1} = 2u_1^{n+1} - u_2^{n+1} \tag{8.2.150}$$

$$u_{-1}^{n+1} = 2u_0^{n+1} - u_1^{n+1}. \tag{8.2.151}$$

Solution: We recall that in HW2.3.2 we showed that difference equation (8.2.149) was a $\mathcal{O}(\Delta t^2) + \mathcal{O}(\Delta x^4)$ approximation of the one way wave equation $v_t + av_x = 0$. We see in HW8.2.7 that as an initial–value problem scheme, difference equation (8.2.149) is conditionally stable with condition

$$|R| \leq \sqrt{\frac{\sqrt{17}-1}{6}}.$$

We emphasize here that we are assuming that $a < 0$ ($R < 0$), so that we have no analytic boundary condition at $x = 0$. Because difference equation (8.2.149) reaches two grid points to the left, we need two numerical boundary conditions. And finally, we should realize that the above scheme can be written as our general scheme (8.2.131), (8.2.133) where $K = 1$, $s = 0$, $p = r = 2$ and $q = 1$.

We begin by rewriting difference equation (8.2.149) as

$$\begin{aligned} u_k^{n+1} = & -\frac{1}{24}(3R^4 + R^2 + 2R)u_{k-2}^n + \frac{1}{6}(3R^4 + 4R^2 + 4R)u_{k-1}^n \\ & + \left[1 - \frac{R^2}{4}(3R^2 + 5)\right] u_k^n + \frac{1}{6}(3R^4 + 4R^2 - 4R)u_{k+1}^n \\ & - \frac{1}{24}(3R^4 + R^2 - 2R)u_{k+2}^n. \end{aligned} \tag{8.2.152}$$

If we take the discrete Laplace transform of equation (8.2.152), we get the resolvent equation

$$\begin{aligned} z\tilde{u}_k = & -\frac{1}{24}(3R^4 + R^2 + 2R)\tilde{u}_{k-2} + \frac{1}{6}(3R^4 + 4R^2 + 4R)\tilde{u}_{k-1} \\ & + \left[1 - \frac{R^2}{4}(3R^2 + 5)\right] \tilde{u}_k + \frac{1}{6}(3R^4 + 4R^2 - 4R)\tilde{u}_{k+1} \\ & - \frac{1}{24}(3R^4 + R^2 - 2R)\tilde{u}_{k+2}. \end{aligned} \tag{8.2.153}$$

Looking for solutions of the form $\tilde{u}_k = \kappa^k$, we obtain the characteristic equation

$$\begin{aligned} 0 = & -\frac{1}{24}(3R^4 + R^2 - 2R)\kappa^4 + \frac{1}{6}(3R^4 + 4R^2 - 4R)\kappa^3 \\ & + \left[1 - z - \frac{R^2}{4}(3R^2 + 5)\right] \kappa^2 + \frac{1}{6}(3R^4 + 4R^2 + 4R)\kappa - \frac{1}{24}(3R^4 + R^2 + 2R). \end{aligned} \tag{8.2.154}$$

Clearly, the form of characteristic equation (8.2.154) does not lend itself to be analyzed directly. We know from Proposition 8.2.20 that for $|z| > 1$, equation (8.2.154) has two solutions that satisfy $|\kappa| < 1$, which we will refer to as κ_{1_1} and κ_{1_2}, and two roots that satisfy $|\kappa| > 1$. Hence, the general solution to equation (8.2.153) can be written as

$$\tilde{u}_k = \phi_1 \kappa_{1_1}^k + \phi_2 \kappa_{1_2}^k. \tag{8.2.155}$$

As we have done before, we next transform boundary conditions (8.2.150)–(8.2.151). We get

$$\tilde{u}_0 = 2\tilde{u}_1 - \tilde{u}_2 \tag{8.2.156}$$

$$\tilde{u}_{-1} = 2\tilde{u}_0 - \tilde{u}_1. \tag{8.2.157}$$

If we insert $\tilde{u}_k = \phi_1 \kappa_{1_1}^k + \phi_2 \kappa_{1_2}^k$ into equations (8.2.156)–(8.2.157) and simplify, we get

$$\begin{pmatrix} \kappa_{1_1}^2 - 2\kappa_{1_1} + 1 & \kappa_{1_2}^2 - 2\kappa_{1_2} + 1 \\ \kappa_{1_1} - 2 + \kappa_{1_1}^{-1} & \kappa_{1_2} - 2 + \kappa_{1_2}^{-1} \end{pmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{8.2.158}$$

Equation (8.2.158) has a nontrivial solution if and only if

$$0 = \det \begin{pmatrix} \kappa_{1_1}^2 - 2\kappa_{1_1} + 1 & \kappa_{1_2}^2 - 2\kappa_{1_2} + 1 \\ \kappa_{1_1} - 2 + \kappa_{1_1}^{-1} & \kappa_{1_2} - 2 + \kappa_{1_2}^{-1} \end{pmatrix}$$
$$= (\kappa_{1_1} - 1)^2(\kappa_{1_2} - 1)^2 \frac{(\kappa_{1_1} - \kappa_{1_2})}{\kappa_{1_1}\kappa_{1_2}}. \tag{8.2.159}$$

Thus, a solution of the form (8.2.155) will satisfy equations (8.2.156)–(8.2.157) if $\kappa_{1_1} = 1$, $\kappa_{1_2} = 1$ or $\kappa_{1_1} = \kappa_{1_2}$. Recall that we assumed that κ_{1_1} and κ_{1_2} are distinct. We insert $\kappa = 1$ into equation (8.2.154) to see that z must be equal to one. Hence, *equations (8.2.153), (8.2.156)–(8.2.157) have no eigenvalues.* To see that $z = 1$ is not a generalized eigenvalue, we set $\kappa = 1 + \eta$ and $z = 1 + \delta$, insert these into equation (8.2.154), expand and simplify to see that

$$\delta = -R\eta + \text{higher order terms.}$$

Thus, since $R < 0$, when z satisfies $|z| > 1$, κ satisfies $|\kappa| > 1$, so $z = 1$ *is not a generalized eigenvalue.*

If κ_{1_1} and κ_{1_2} are not distinct, the general solution to equation (8.2.153) can be written as

$$\tilde{u}_k = \phi_1 \kappa_{1_1}^k + \phi_2 k \kappa_{1_1}^k. \tag{8.2.160}$$

Insertion of this expression into equations (8.2.156)–(8.2.157) yields the equation

$$\begin{pmatrix} \kappa_{1_1}^2 - 2\kappa_{1_1} + 1 & 2\kappa_{1_1}(\kappa_{1_1} - 1) \\ \kappa_{1_1} - 2 + \kappa_{1_1}^{-1} & \kappa_{1_1} - \kappa_{1_1}^{-1} \end{pmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{8.2.161}$$

Setting the determinant of the coefficient matrix in equation (8.2.161) equal to zero gives

$$(\kappa_{1_1} - 1)^4 \frac{1}{\kappa_{1_1}} = 0 \tag{8.2.162}$$

or $\kappa_{1_1} = 1$. As in the case where the κ's were distinct, the insertion of $\kappa = 1$ into equation (8.2.154) again gives $z = 1$. The same perturbation analysis used earlier will again show that $z = 1$ is not a generalized eigenvalue.

Therefore, *difference scheme (8.2.149) along with numerical boundary conditions (8.2.150)–(8.2.151) is stable for*

$$0 \geq R \geq -\sqrt{\frac{\sqrt{17} - 1}{6}}$$

(inherited from the assumption that difference scheme (8.2.149) must be stable as an initial–value scheme).

HW 8.2.7 Verify that difference scheme (8.2.149) is stable as an initial–value scheme if

$$|R| \leq \sqrt{\frac{\sqrt{17} - 1}{6}}$$

HW 8.2.8 Show that the difference scheme consisting of equation

$$u_k^{n+1} = u_k^{n-1} - \frac{4R}{3}\delta_0 u_k^n + \frac{R}{6}(u_{k+2}^n - u_{k-2}^n) \tag{8.2.163}$$

along with numerical boundary conditions

$$u_0^{n+1} = u_0^{n-1} + \frac{11R}{6}(u_0^{n+1} + u_0^{n-1}) - 6Ru_1^n + 2u_2^n - \frac{2R}{3}u_3^n \quad (8.2.164)$$

$$u_1^{n+1} = u_1^{n-1} + \frac{2R}{3}u_0^n + 3R(u_1^{n+1} + u_1^{n-1}) - 2Ru_2^n + \frac{R}{3}u_3^n \quad (8.2.165)$$

is a stable scheme.

HW 8.2.9 Repeat the calculations done in HW6.4.2 using (8.2.129) as the numerical boundary condition. Use $j = 1$ and $j = 2$.

HW 8.2.10 Find an approximation of the solution to the problem given in HW5.6.8 using difference scheme (8.2.149)–(8.2.151). Use $u_{M+1}^{n+1} = 1.0$ as the extra boundary condition needed at $x = 1$. See Example 8.2.14 for one explanation why this is a logical choice for the "extra analytic boundary condition."

8.2.4 Left Quarter Plane Problems

By this time we have considered several right quarter plane problems. We emphasized earlier that when we start with an initial–boundary–value problem on a finite domain, if we prove that the scheme is stable as an initial–value problem scheme and both the right and left quarter plane problems are stable, then we can apply Theorem 8.2.2 to show that the initial–boundary–value scheme is stable. And in Sections 8.2.1, 8.2.1.1 and 8.2.1.2, where we discussed the first "easy case," we first considered the right quarter plane problem and then considered the left quarter plane problem. Thus, it would now be logical for us to return to each of the cases considered in Sections 8.2.2, 8.2.2.2 and 8.2.3 to consider the corresponding left quarter plane problems. However, we will not do this, because we assume that if the approach and comments concerning the left quarter plane problem considered in Section 8.2.1.2 are taken into account, the reader will easily be able to adapt the approach of Section 8.2.1.2 to the left quarter plane problems for systems, multilevel schemes and for the general theory.

The left quarter plane problems associated with systems of partial differential equations considered in Section 8.2.2.1 are very similar to the right quarter plane problems associated with the same schemes. The boundary conditions for the left quarter plane problem will generally be a mixture of numerical boundary conditions and analytic boundary conditions (equal to the number of positive eigenvalues of the coefficient matrix). As with the left quarter plane problems considered in Section 8.2.1.2, we find solutions to the resolvent equation in terms of κ_{2_j}, $j = 1, \ldots, K$, where $|\kappa_{2_j}| > 1$ (because k approaches $-\infty$). Otherwise, the analysis is very similar to that done in Section 8.2.2.1.

When $a < 0$, the left quarter plane problems associated with multilevel schemes are very easy. If we consider the left quarter plane problem for the scheme studied in Example 8.2.10 (leapfrog scheme with $a < 0$), we assign one analytic boundary condition at $x = 1$. If we assign $v(1,t) = g(t)$ and consider the definition of $\mathbf{U}_k^n$, (8.2.116), we see that we must consider difference equation (8.2.117) for $k = -\infty, \ldots, M-1$, along with boundary condition

$$\mathbf{U}_M^n = \begin{bmatrix} g(n\Delta t) \\ g((n-1)\Delta t) \end{bmatrix}.$$

With this analytic boundary condition at $k = M$, the analysis of the left quarter plane problem is as easy as that for the left quarter plane problem for a scalar scheme with $a < 0$ and one boundary condition given at $x = 1$.

Probably the most interesting twist that we encounter when we analyze the left quarter plane problems occurs when we have $a < 0$ and use a scheme that reaches more than one point to the right, i.e., like difference scheme (8.2.149). When we use such a scheme, we must obtain additional boundary conditions near $k = M$. To illustrate the method by which we obtain additional boundary conditions from one given analytic boundary condition, we consider the following left quarter plane problem associated with the higher order scheme considered in Example 8.2.13.

Example 8.2.14 Consider the stability of the difference equation

$$u_k^{n+1} = u_k^n - \frac{R}{2}\Big(1 - \frac{1}{6}\delta^2\Big)\delta_0 u_k^n + \frac{R^2}{2}\Big(\frac{4}{3} + R^2\Big)\delta^2 u_k^n - \frac{R^2}{8}\Big(\frac{1}{3} + R^2\Big)\delta_0^2 u_k^n, \quad k = -\infty, \ldots, M-1 \tag{8.2.166}$$

along with the analytic boundary condition

$$v(1,t) = g(t), \quad t \geq 0. \tag{8.2.167}$$

Solution: The analytic boundary condition (8.2.167) gives us the obvious boundary condition for difference equation (8.2.166)

$$u_M^n = g(n\Delta t). \tag{8.2.168}$$

Because difference scheme (8.2.166) reaches two points to the right, we need a boundary condition at $k = M+1$. (Some authors would consider difference equation (8.2.166) only at $k = -\infty, \ldots, M-2$ and then provide boundary conditions at $k = M-1$ and $k = M$. These approaches are clearly equivalent.) We saw in HW2.3.2 that difference scheme (8.2.166) is a $\mathcal{O}(\Delta t^2)+\mathcal{O}(\Delta x^4)$ accurate approximation to the partial differential equation

$$v_t + a v_x = 0. \tag{8.2.169}$$

Hence, we would like to choose a boundary condition for difference equation (8.2.166) at $k = M+1$ that is equally accurate (or as we shall see in Section 8.4.3, we would like to choose a boundary condition that is at most one order less accurate than the difference

scheme). Noting that

$$\begin{aligned} v(1+\Delta x,t) &= \sum_{j=0}^{3} \frac{\Delta x^j}{j!} \frac{\partial^j}{\partial x^j} v(1,t) + \mathcal{O}(\Delta x^4) \\ &= \sum_{j=0}^{3} \frac{\Delta x^j}{j!} \left(\frac{(-1)^j}{a^j} \frac{\partial^j}{\partial t^j} v(1,t) \right) + \mathcal{O}(\Delta x^4) \\ &= \sum_{j=0}^{3} \frac{(-1)^j \Delta x^j}{a^j j!} g^{(j)}(t) + \mathcal{O}(\Delta x^4), \end{aligned} \tag{8.2.170}$$

we see that using

$$u_{M+1}^n = g_1^n = \sum_{j=0}^{3} \frac{(-1)^j \Delta x^j}{a^j j!} g^{(j)}(n\Delta t) \tag{8.2.171}$$

as a boundary condition for difference equation (8.2.166) will provide us with a $\mathcal{O}(\Delta x^4)$ accurate boundary condition.

To perform the stability analysis of difference scheme (8.2.166), (8.2.168), (8.2.171) we proceed much as we did in Example 8.2.13. Taking the transform of equation (8.2.166) will yield a resolvent equation of the form (8.2.153) (except that for this case k ranges from $-\infty$ to $M-1$); the characteristic equation is given by (8.2.154); and the general solution of the resolvent equation will be of the form

$$\tilde{u}_k = \phi_1 \kappa_{2_1}^{k-M} + \phi_2 \kappa_{2_2}^{k-M} \tag{8.2.172}$$

for $k = -\infty, \ldots, M-1$ and where κ_{2_j}, $j = 1, 2$, are chosen so that $|\kappa_{2_j}| > 1$ when $|z| > 1$. The biggest difference between this analysis and that of Example 8.2.13 is that the homogeneous transformed boundary conditions are given by

$$\tilde{u}_M = \tilde{u}_{M+1} = 0. \tag{8.2.173}$$

Combining the solution given by equation (8.2.172) with boundary conditions (8.2.173), we get

$$\begin{aligned} 0 &= \phi_1 + \phi_2 \\ 0 &= \phi_1 \kappa_{2_1} + \phi_2 \kappa_{2_2}. \end{aligned}$$

If we assume that κ_{2_1} and κ_{2_2} are distinct, we find that $\phi_1 = \phi_2 = 0$. Hence, there are no nontrivial solutions to equations (8.2.153) ($k = -\infty, \ldots, M-1$), (8.2.173), and difference scheme (8.2.166), (8.2.168), (8.2.171) is stable.

If $\kappa_{2_1} = \kappa_{2_2}$, then the general solution will be given by

$$\tilde{u}_k = \phi_1 \kappa_{2_1}^{k-M} + \phi_2 k \kappa_{2_1}^{k-M}.$$

Combining this solution with boundary conditions (8.2.173) gives

$$\begin{aligned} 0 &= \phi_1 + M\phi_2 \\ 0 &= \phi_1 \kappa_{2_1} + \phi_2 (M+1) \kappa_{2_1}. \end{aligned}$$

Solving this pair of equations, we again find that $\phi_1 = \phi_2 = 0$ and that difference scheme (8.2.166), (8.2.168), (8.2.171) is stable. And finally, since difference scheme (8.2.166) must be stable as an initial–value scheme, we see that difference scheme (8.2.166), (8.2.168), (8.2.171) is stable if

$$0 \geq R \geq -\sqrt{\frac{\sqrt{17}-1}{6}}.$$

Remark: We should realize that if we have a scheme that reaches three or more points to the right, the same approach as was used in Example 8.2.14 can be used to generate more boundary conditions that are accurate to any desired order. For example, if a boundary condition is needed at $k = M + 2$, we expand u^n_{M+2} about $x = 1$ $(k = M)$ and proceed as we did in Example 8.2.14.

If we consider the one way wave equation with $a > 0$, the approach is the same except for the fact that everything is reversed. Since we have an analytic boundary condition at $x = 0$, if we have a scalar difference equation with $r = 1$, the right quarter plane problem will be easy (if we have a system or a broad stencil, we may have to treat a numerical boundary condition in addition to the analytic boundary condition or use the analytic boundary condition to generate additional numerical boundary conditions), and we will have to treat the numerical boundary condition (if $p = 1$) as a part of the left quarter plane problem. However, the results found earlier for the right quarter plane problem will all transfer over to the left quarter plane problem. The numerical boundary conditions defined analogous to the stable or unstable numerical boundary conditions for the right quarter plane problem for $a < 0$ will be stable or unstable for the left quarter plane problem for $a > 0$. To see how easily the results for $a < 0$ transfer over to the similar problem for $a > 0$, see HW8.2.11.

And finally, we hope that we have provided a sufficient number of examples so that the reader feels comfortable with the GKSO stability analysis. The reader should be aware that the examples were chosen carefully so as to minimize the computational complexity. It should not surprise the reader to be faced with a situation—maybe even only slightly different from a scheme considered in the text—where the necessary analysis is prohibitively difficult. In some of these situations, rather than changing the scheme, the reader may have to become an "experimental analyst." The choice of boundary conditions or numerical boundary conditions may be based on the experience gained from considering the model problems done in the text. Though some of the computations may be impossible, graphics may be used to "indicate" that the desired result is true or false (say that a given equation has no roots for a certain range of values of R). Some experimental numerical calculations may have to be run with the scheme to see whether it can be made to become unstable. And finally, as you proceed to use a scheme that has not been fully analyzed, when the scheme begins to misbehave, you will be at least suspicious that it may be due to the choice of the numerical boundary conditions (and the bad results will probably be due to a bug in your code).

HW 8.2.11 Consider the stability of difference scheme

$$u_k^{n+1} = \frac{1}{2}(R^2 + R)u_{k-1}^n + (1 - R^2)u_k^n + \frac{1}{2}(R^2 - R)u_{k+1}^n, \quad k = 1, \ldots, M-1 \tag{8.2.174}$$

$$u_k^0 = 0, \quad k = 0, \ldots, M$$

$$u_0^{n+1} = g^{n+1}, \quad n \geq 0$$

$$u_M^{n+1} - u_{M-1}^{n+1} = 0, \quad n \geq 0 \tag{8.2.175}$$

where R is assumed to be greater than zero ($a > 0$). (Difference equation (8.2.174) is the Lax-Wendroff difference equation, and equation (8.2.175) is a numerical boundary condition.)

HW 8.2.12 Consider the initial–boundary–value problem

$$v_t + av_x = 0, \quad x \in (0,1), \quad t > 0 \tag{8.2.176}$$

$$v(x,0) = 0, \quad x \in [0,1] \tag{8.2.177}$$

$$v(0,t) = g(t), \quad t \geq 0 \tag{8.2.178}$$

(where $a > 0$) and difference equations (8.2.149), $u_k^0 = 0$, $k = 0, \ldots, M$ and $u_0^{n+1} = g^{n+1}$, $n \geq 0$. Provide a set of numerical boundary conditions so that the given difference equations along with these numerical boundary conditions will be a stable difference scheme for approximating the solution of initial–boundary–value problem (8.2.176)–(8.2.178), and verify that the resulting scheme is stable.

8.3 Constructing Stable Difference Schemes

In this section we shall describe a method for constructing stable difference schemes given in a paper with the same title as this section, ref. [52]. The crucial part of the argument is the treatment of the numerical boundary conditions. The schemes constructed by this technique use "partial differential equation like," or pde-like, numerical boundary conditions in that while serving as numerical boundary conditions, they also approximate the partial differential equation at the given point, e.g., numerical boundary conditions

$$u_0^{n+1} = u_0^n - R(u_1^n - u_0^n) \tag{8.3.1}$$

or

$$u_0^{n+1} + u_1^{n+1} + R(u_1^{n+1} - u_0^{n+1}) = u_0^n + u_1^n - R(u_1^n - u_0^n) \tag{8.3.2}$$

that we used as numerical boundary conditions on page 20.

The setting that we shall consider is that used for the case of the general theory in Section 8.2.3. We should realize that since this is the general case, the method will also apply to the earlier initial–boundary–value problems considered in Sections 8.2.1 and 8.2.2. Hence, we consider the right quarter plane scheme (8.2.131)–(8.2.133). In earlier sections, one of our assumptions was always that difference scheme (8.2.131) was stable as an initial–value problem scheme. To theoretically satisfy this condition, it was necessary for us to be able to extend the difference scheme to be applicable for all k, $-\infty < k < \infty$. In reality, we usually already know that the scheme is at least conditionally stable for the associated initial–value problem and the most that we must do is impose the condition that will ensure the stability of the scheme. For the theorem given below, we want to be able to consider both difference equations (8.2.131) and (8.2.133) as initial–value problem schemes. The difference is in the consideration of difference equation (8.2.133) as an initial–value problem scheme. It should be fairly clear that this process will restrict the form of boundary condition (8.2.133). For difference equation (8.2.133) to be stable as an initial–value problem scheme, it generally must approximate some partial differential equation. One way to get a difference equation that is both stable and consistent is to choose a one sided approximation to the partial differential equation being solved as the numerical boundary condition. As we stated earlier, this is one of the types of numerical boundary conditions that we have been using. This approach will allow us to use numerical boundary conditions that are pde-like and will generally not allow us to use numerical boundary conditions that are extrapolations. We state the following theorem which is stated and proved in ref. [52].

Theorem 8.3.1 *If difference equations (8.2.131) and (8.2.133) are both stable as initial–value problem schemes and difference equation (8.2.133) is dissipative, then the difference scheme (8.2.131)–(8.2.133) is stable.*

If we work at applying this theorem for a while, we will see that the theorem does not give us any stable schemes that could not be proved by the earlier results in this chapter. And since the numerical boundary conditions must be both pde-like and stable as an initial–value scheme, there are some stable schemes that we found earlier that we cannot obtain by this method or that we obtain here with a more restrictive stability condition. However, we will see that this theorem does give us some of those results with little or no work. We illustrate the application of Theorem 8.3.1 with the following examples.

Example 8.3.1 Use Theorem 8.3.1 to obtain a stability result for the scheme consisting of the Lax-Wendroff scheme along with numerical boundary condition (8.3.1).

Solution: We know that the Lax-Wendroff scheme is stable as an initial–value problem

scheme if $|R| \leq 1$. Also, we know that the difference scheme

$$u_k^{n+1} = u_k^n - R(u_{k+1}^n - u_k^n) \tag{8.3.3}$$

is stable if $-1 \leq R \leq 0$ and is dissipative of order 2 (Part 1, page 397). Hence, by Theorem 8.3.1 the difference scheme consisting of the Lax-Wendroff scheme along with numerical boundary condition (8.3.1) is stable if $-1 \leq R \leq 0$. We should note that this result is given on page 20.

Also on page 20 we state that the Crank-Nicolson scheme along with numerical boundary condition (8.3.1) is stable if $-2 \leq R \leq 0$. Of course, we know that the Crank-Nicolson scheme is unconditionally stable as an initial–value problem scheme. As we stated earlier, we know that difference scheme (8.3.3) is stable if $-1 \leq R \leq 0$. And we also know that difference scheme (8.3.3) is dissipative. Hence we see that the Crank-Nicolson scheme along with numerical boundary condition (8.3.1) is stable if $-1 \leq R \leq 0$. We note that though this stability result is more restrictive than the result given on page 20, the result obtained here was very easy to obtain.

Example 8.3.2 Discuss the stability of the Crank-Nicolson scheme along with numerical boundary condition (8.3.2).

Solution: We know that the Crank-Nicolson scheme is unconditionally stable. To apply Theorem 8.3.1, we must analyze the stability of the difference scheme

$$u_k^{n+1} + u_{k+1}^{n+1} + R(u_{k+1}^{n+1} - u_k^{n+1}) = u_k^n + u_{k+1}^n - R(u_{k+1}^n - u_k^n). \tag{8.3.4}$$

This is not a difference scheme that we have seen before. In HW8.3.1 we see that difference scheme (8.3.4) is unconditionally stable as an initial–value problem scheme. However, the result found in HW8.3.1 also shows that difference scheme (8.3.4) is nondissipative. Hence, *we cannot apply Theorem 8.3.1 to prove stability of the Crank-Nicolson scheme along with boundary condition (8.3.2).* To analyze the stability of the Crank-Nicolson scheme along with numerical boundary condition (8.3.2) we must return to the GKSO theory, and as we did in HW8.3.2, find that the Crank-Nicolson scheme along with boundary condition (8.3.2) is stable for $R \leq 0$.

We should be aware that Theorem 8.3.1 holds equally well for systems of equations. However, it is not as easy to construct the dissipative boundary scheme as it is for the scalar case. In the next example, we demonstrate one way that we can use to construct boundary conditions that will give us a stable scheme for solving a hyperbolic system of partial differential equations.

Example 8.3.3 Use Theorem 8.3.1 to construct boundary conditions that along with difference equation

$$\mathbf{u}_k^{n+1} - \frac{R}{4} A\delta_0 \mathbf{u}_k^{n+1} = \mathbf{u}_k^n + \frac{R}{4} A\delta_0 \mathbf{u}_k^n, \quad k = 1, 2, \ldots \tag{8.3.5}$$

will yield a stable difference scheme, where the matrix A is given by

$$A = \begin{pmatrix} 5/3 & 1 & 5/3 \\ -1/3 & -1 & -7/3 \\ 1/3 & -1 & 1/3 \end{pmatrix}$$

Solution: Recall that matrix A is the same matrix used in Examples 8.2.8 and 8.2.9 and that difference equation is designed to find an approximation to the hyperbolic partial differential equation

$$\mathbf{v}_t = A\mathbf{v}_x, \quad x \in (0,\infty),\ t > 0. \tag{8.3.6}$$

Since we know that matrix A has two positive eigenvalues and one negative eigenvalue, we know that we can assign one analytic boundary condition at $x = 0$. Moreover, we know that if we multiply difference equation (8.3.5) and partial differential equation (8.3.6) by S^{-1} (given in (8.2.102)), rewrite the nonexistent identity matrix that sits between the A and the $\mathbf{v}_x$, the A and the $\mathbf{u}_k^{n+1}$, and the A and the $\mathbf{u}_k^n$ as SS^{-1} and let $\mathbf{V} = S^{-1}\mathbf{v}$ and $\mathbf{U}_k^n = S^{-1}\mathbf{u}_k^n$, then partial differential equation (8.3.6) and difference equation (8.3.5) can be rewritten as

$$\mathbf{V}_t = D\mathbf{V}_x \tag{8.3.7}$$

and

$$\mathbf{U}_k^{n+1} - \frac{R}{4}D\delta_0\mathbf{U}_k^{n+1} = \mathbf{U}_k^n + \frac{R}{4}D\delta_0\mathbf{U}_k^n \tag{8.3.8}$$

where

$$D = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -2 \end{pmatrix}.$$

Since the partial differential equation and difference equation have now become completely uncoupled, we know that we can set an analytic boundary condition at $x = 0$ to

$$V_3(0,t) = G_3(t), \tag{8.3.9}$$

and hence we may set the analytic boundary condition to our numerical scheme at $k = 0$ as

$$U_{3_0}^{n+1} = G_3((n+1)\Delta t) = G_3^{n+1}. \tag{8.3.10}$$

We must assign numerical boundary conditions at $k = 0$ to $U_{1_k}^{n+1}$ and $U_{2_k}^{n+1}$. From our previous work done in this section, we know that difference equation (8.3.8) along with analytic boundary condition (8.3.10) and numerical boundary conditions

$$U_{1_0}^{n+1} = U_{1_0}^n + 2R\delta_+U_{1_0}^n \tag{8.3.11}$$

$$U_{2_0}^{n+1} = U_{2_0}^n + R\delta_+U_{2_0}^n \tag{8.3.12}$$

will give us a stable difference scheme for solving partial differential equation (8.3.6) along with boundary condition (8.3.9) for $0 \le R \le \frac{1}{2}$ (R satisfies $-1 \le -2R \le 0$ and $-1 \le -R \le 0$). If we then rewrite boundary conditions (8.3.10)-(8.3.12) as

$$\mathbf{U}_0^{n+1} = \begin{pmatrix} 1-2R & 0 & 0 \\ 0 & 1-R & 0 \\ 0 & 0 & 0 \end{pmatrix}\mathbf{U}_0^n + \begin{pmatrix} 2R & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & 0 \end{pmatrix}\mathbf{U}_1^n + \begin{bmatrix} 0 \\ 0 \\ G_3^{n+1} \end{bmatrix}, \tag{8.3.13}$$

multiply by S, again replace the nonexistent identity matrix in the appropriate places by $S^{-1}S$ and replace $S\mathbf{U}_k^n$ by $\mathbf{u}_k^n$, boundary condition (8.3.13) can be written as

$$\begin{aligned}\mathbf{u}_0^{n+1} =& S\begin{pmatrix} 1-2R & 0 & 0 \\ 0 & 1-R & 0 \\ 0 & 0 & 0 \end{pmatrix}S^{-1}\mathbf{u}_0^n \\ &+ S\begin{pmatrix} 2R & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & 0 \end{pmatrix}S^{-1}\mathbf{u}_1^n + S\begin{bmatrix} 0 \\ 0 \\ G_3^{n+1} \end{bmatrix}.\end{aligned} \tag{8.3.14}$$

By Theorem 8.3.1, difference scheme (8.3.5) along with boundary condition (8.3.14) will provide a stable difference scheme for approximating the solution to partial differential equation (8.3.6) along with boundary condition (8.3.9).

As we see, if it is possible to construct a one sided, dissipative difference scheme approximating the partial differential equation near the boundary, it is possible to link the numerical boundary condition derived from this one sided scheme with any other stable scheme and obtain a scheme for the associated initial–boundary–value problem. We emphasize that when we do this, we must impose the stability conditions on the resulting difference schemes that are associated with both of the component schemes.

When we use this approach, one might ask why we wouldn't just use the one sided scheme that was used to generate the numerical boundary conditions on the entire region. There are times when we want to use a particular scheme as our interior scheme because it has certain properties that we especially like. For example, we might want to use Crank-Nicolson in the interior because it is nondissipative. Also, we might want to use either the Lax-Wendroff scheme or the Crank-Nicolson scheme in the interior because it is second order accurate with respect to both space and time. As we shall see in Section 8.4.3, even though the difference scheme (8.3.3) is only first order accurate with respect to time and space, the difference scheme consisting of the Lax-Wendroff scheme or the Crank-Nicolson scheme along with the numerical boundary condition (8.3.1) will be second order accurate in time and space.

And, as we do so often in this chapter, we remark that the above results will apply equally well for schemes for left quarter problems. In addition, application of Theorem 8.3.1 along with Theorem 8.2.2 enables us to easily construct stable difference schemes for initial–boundary–value problems defined on an interval.

HW 8.3.1 Analyze the stability of difference scheme (8.3.4) as an initial-value problem scheme. Show that difference scheme (8.3.4) is nondissipative.

HW 8.3.2 Use the GKSO theory to analyze the stability of the Crank-Nicolson scheme along with numerical boundary condition (8.3.2).

8.4 Consistency and Convergence

We must always remember that the reason that we have wanted results on stability in the past was so that we can use the Lax Theorem. The Lax Theorem, Theorem 2.5.2, states that consistency plus stability implies convergence and is true when we consider initial–boundary–value problems. The proof of Theorem 2.5.2 had nothing to do with the fact that the grid went from $-\infty$ to ∞. However, the setting in which we proved Theorem 2.5.2 was different from the setting in which we have studied stability in this chapter. In Theorem 2.5.2, we proved convergence at a particular value of t and assumed consistency and stability for values up to that value of t. The

proof of Theorem 2.5.2 started by looking at the norm of $\mathbf{w}^n = \mathbf{u}^n - \mathbf{v}^n$. In this chapter, all of our work consists of summing over all n values, i.e., we prove stability for all of $t > 0$ as well as the given spatial region. The reason we have used this stronger definition of stability is that the definition is compatible with the discrete Laplace transform. In other words, we proved the stability that we proved because we had tools that would allow us to work with this definition of stability.

The Lax Theorem as it is given in Theorem 2.5.2 is not applicable to proving convergence based on the stability of this chapter. It is possible to alter the proof of Theorem 2.5.2 to obtain a convergence result, but this is not the approach we wish to take. In Section 8.4.3 we will state a convergence theorem due to Gustafsson that is superior to the initial–boundary–value scheme application of Theorem 2.5.2. Since we will plan on eventually using Theorem 8.4.1 as our convergence theorem, we do not need a version of Theorem 2.5.2 that is applicable to the schemes analyzed in this chapter. Before we proceed to state the convergence theorem due to Gustafsson, refs. [16] and [17], we will discuss the concepts of consistency and convergence as they apply to this chapter.

8.4.1 Norms and Consistency

The first big difference that we have between the treatment of consistency–stability–convergence in this chapter and that in Chapters 2–6 is that stability in this chapter is proved with respect to the norm used in Definition 8.2.1. It should not surprise us that when we prove convergence based on any of the stability definitions used in this chapter, the convergence will be with respect to the same norm (or a strongly related norm) as is used in the definition of stability. Also, it should be reasonably clear that the consistency must be proved with respect to that same or strongly related norm. The truth of the matter is that we do not generally care with which norm convergence is given (we are just happy to get convergence). The biggest difference is that the consistency must also be proved with respect to this norm. Proving consistency with respect to the norms used in this chapter is not very difficult for the approach to consistency that we have taken in the past (assuming all necessary smoothness). But we must be aware that to obtain consistency with respect to the $\|\cdot\|_{\alpha\Delta t,\Delta t,(0,1),\Delta x}$, $\|\cdot\|_{\alpha\Delta t,\Delta t,[0,\infty),\Delta x}$ or $\|\cdot\|_{\alpha\Delta t,\Delta t,(\infty,1],\Delta x}$ norm, we must generally make stronger assumptions on the smoothness of the solution to the partial differential equation than were necessary when we worked with the $\ell_{2,\Delta x}$ norm. Worse yet, if for some reason you are unable to just assume the necessary smoothness, then you will have to prove that the solution is sufficiently smooth so that the truncation error, $\boldsymbol{\tau}^n$, is small with respect to the appropriate norm.

8.4.2 *Consistency of Numerical Boundary Conditions*

The most important difference in proving consistency is the treatment of any numerical boundary conditions that are present. Though it is not the proof that we care about, recall the proof of Theorem 2.5.2. The accuracy of the scheme (consistency) was used to show that the error $\mathbf{w}^j = \mathbf{v}^j - \mathbf{u}^j$ satisfies

$$\mathbf{w}^{n+1} = Q\mathbf{w}^n + \Delta t \tau^n. \tag{8.4.1}$$

After certain manipulations of the above equation, the growth of Q with respect to n was bounded by the stability of the scheme. Since stability in this chapter includes the numerical boundary conditions, the operator Q must contain the numerical boundary conditions. Specifically, if Q is written as a matrix (finite or infinite), then the numerical boundary condition will either be a row of the matrix or will be included into one of the rows of the matrix. Hence, we must take into account the consistency of the numerical boundary conditions.

The general approach to order of accuracy arguments (consistency) is to expand everything in terms of Taylor series expansions. If we return to the examples done in Section 2.3, we see that the Taylor series expansion of $\Delta t \tau^n = \mathbf{v}^{n+1} - Q\mathbf{v}^n$ usually contains one or more terms that include the partial differential equation and/or the boundary conditions. Because v satisfies the partial differential equation and boundary conditions, these terms are zero when v is a solution to the initial–boundary–value problem, and some of the lower order terms go away.

If we look at the numerical boundary conditions considered in this chapter, we see that there are two distinct types. Some of them look like the partial differential equation that we are solving (usually a one sided scheme that could be used to approximate the solution to the partial differential equation), which we will refer to as **pde-like numerical boundary conditions** , and the rest are **extrapolation numerical boundary conditions** that do not look like anything associated with the analytic problem we are solving. For those numerical boundary conditions that can be derived from the partial differential equation, the consistency follows in the same way that the consistency follows for the difference equation approximating the partial differential equation. The terms are expanded using Taylor series expansions, the result contains a term that includes the partial differential equation and when this term is eliminated (it is zero because we are expanding a solution to the partial differential equation), the result yields the order of approximation. For example, we note that

$$v_0^{n+1} + v_1^{n+1} + R\delta_+ v_0^{n+1} - v_0^n - v_1^n + R\delta_+ v_0^n = \mathcal{O}(\Delta t \Delta x) + \mathcal{O}(\Delta t^3), \tag{8.4.2}$$

page 20 and (8.2.128), and

$$v_0^{n+1} - v_0^n + R\delta_+ v_0^n = \mathcal{O}(\Delta t \Delta x) + \mathcal{O}(\Delta t^2), \tag{8.4.3}$$

page 20, where in both cases we have used the fact that v satisfies $v_t + av_x = 0$. Consistency results (8.4.2), (8.4.3) and other similar results are one of the reasons that difference schemes that include numerical boundary conditions of the type developed in Section 8.3 (pde-like numerical boundary conditions) are so nice to work with.

When we expand the extrapolation numerical boundary conditions in a Taylor series, there is no analytic equation associated with it to help get rid of the lower order terms. Thus we must choose numerical boundary conditions that will rid themselves of the lower order terms. We say that a numerical boundary condition Q_{nbc} is **0-consistent of order** (r, s) if

$$Q_{nbc} V_k^n = \mathcal{O}(\Delta x^r) + \mathcal{O}(\Delta t^s) \tag{8.4.4}$$

for an arbitrary function V. Since there is no function that is special to the extrapolated numerical boundary conditions (as the solution of the partial differential equation is to a pde-like numerical boundary condition), we define 0-consistency for an arbitrary function V. To see that we have been choosing 0-consistent numerical boundary conditions, we note that

$$v_0^{n+1} + v_1^{n+1} - 2v_2^{n+1} = \mathcal{O}(\Delta x) \tag{8.4.5}$$

$$v_0^{n+1} - v_1^{n+1} - v_0^n + v_1^n = \mathcal{O}(\Delta x \Delta t) \tag{8.4.6}$$

$$v_0^{n+1} - v_1^{n+1} = \mathcal{O}(\Delta x) \tag{8.4.7}$$

$$v_0^{n+1} - v_1^n = \mathcal{O}(\Delta x) + \mathcal{O}(\Delta t) \tag{8.4.8}$$

$$v_0^{n+1} - 2v_1^{n+1} + v_2^{n+1} = \mathcal{O}(\Delta x^2). \tag{8.4.9}$$

We see that any numerical boundary condition for which the coefficients sum to zero will be 0-consistent. We emphasize that in the expansions given in (8.4.5)–(8.4.9), we have not used the fact that v satisfies any particular partial differential equation or boundary condition.

One bad boundary condition that we tried in the past deserves particular attention. In HW5.6.10(a) we used the numerical boundary condition $u_0^n = 1.0$. It should be clear that this is a terrible choice for the numerical boundary condition and we obtained terrible numerical results with this choice. However, it is easy to see that the Lax-Wendroff scheme along with boundary condition $u_0^n = 1.0$ is a stable scheme for the right quarter plane problem. The homogeneous transformed boundary condition is $\tilde{u}_0 = 0$ so the only way to obtain a solution of the form $\tilde{u}_k = \phi\kappa^k$ is to choose $\phi = 0$. Thus there can be no eigenvalues or generalized eigenvalues. We note that boundary condition $u_0^n = 1.0$ is neither pde-like nor an extrapolation numerical boundary condition. If we were to do a consistency argument, we would find that $\Delta t \tau_0^n = v_0^n - 1.0 = \mathcal{O}(1)$. Hence,

$\tau_0^n = \mathcal{O}(1/\Delta t)$ and the numerical boundary condition is accurate to order -1. In the case of HW5.6.10(a), it is the order of accuracy of the numerical boundary condition, not the stability, that makes the numerical results so bad.

One last comment should be made with respect to numerical boundary conditions and accuracy. Though the emphasis on the choice of numerical boundary conditions is usually to produce a stable scheme, it is clear that τ^n will also contain the truncated terms of the numerical boundary condition, and τ^n ultimately determines the error in the scheme. As we shall see in the next two sections, just as is the case when we approximate analytic boundary conditions, care must be taken when numerical boundary conditions are chosen so that the order of accuracy of these numerical boundary conditions does not lower the accuracy of the scheme. The possibility of lowering the accuracy of the scheme by the choice of numerical boundary condition is more acute when 0-consistent (extrapolated) numerical boundary conditions are used.

8.4.3 Convergence Theorem: Gustafsson

When we applied the Lax Theorem to prove convergence in Chapter 3, we used norm consistency, and the convergence rate was only as good as the worst accuracy of any of the components of the truncation vector. Specifically, this included any approximations to boundary conditions. In Example 2.3.4, we showed that if we use the norm consistency as our definition of consistency (which is the consistency that we must use to apply Theorem 2.5.2), we were able to show that the first order approximation of the zero Neumann boundary condition was only 0-th order accurate. The first order accuracy of the first order approximation of the zero Neumann boundary condition is due to the fact that norm consistency forces us to write the boundary condition to fit into the form $\mathbf{u}^{n+1} = Q\mathbf{u}^n + \Delta t \mathbf{G}^n$. Hence, we were not able to use the Lax Theorem to prove that difference scheme (2.3.42)–(2.3.44) is convergent. The numerical experiment conducted in HW1.5.9 and HW1.5.10 gives us evidence that the scheme including the first order accurate approximation of the zero Neumann boundary condition does converge.

As we indicated earlier, things are not as bad as they seem. The result proved in ref. [16] and [17] is a convergence result specifically for initial–boundary–value schemes. Other than the fact that the Gustafsson result is designed for norms like those used in this chapter (the convergence is for all x and all t), the result is only slightly different from that of Theorem 2.5.2. However, the difference is very important. Roughly, the result is that *if the order of accuracy of the boundary conditions is $m-1$, the order of accuracy of the difference equation (approximating the partial differential equation) is m and the scheme is stable, then the scheme will be accurate of order m.*

This result should not totally surprise us in that in Section 6.4.2 we saw that we could use an initialization scheme for starting a multilevel scheme that was of lower order than the basic difference scheme and still preserve the order of accuracy of the difference scheme. We note that the analyses performed in the earlier sections of this chapter for proving stability of initial–boundary–value schemes was to treat the spatial boundary conditions (discretization of analytic boundary conditions and numerical boundary conditions) much like the initialization schemes for multilevel schemes. To analyze the stability of a difference scheme containing a spatial boundary condition, we transformed out the time dependence part of the scheme and then solved and/or analyzed the resulting difference equation. The resulting difference equation was generally a multilevel scheme in the spatial index, where the boundary conditions play the part of the initialization scheme. The way that we are able to preserve the order of accuracy of our difference equation while we approximate our boundary conditions to one order less than our approximation of the partial differential equation is the same way that we were able to get by with an initialization scheme in Section 6.4.2 that was of lower order than that of our basic difference scheme.

The setting for the Gustafsson convergence result is that used in Section 8.2.3 for the case of the general GKSO theory. We again consider a scheme for a right quarter plane problem of the form

$$Q_{-1}\mathbf{u}_k^{n+1} = \sum_{j=0}^{s} Q_j \mathbf{u}_k^{n-j} + \Delta t \mathbf{G}_k^n,$$

$$k = 1, 2, \ldots, \; n = s, s+1, \ldots \tag{8.4.10}$$

$$\mathbf{u}_k^m = \mathbf{f}_k^m, \; m = 0, 1, \ldots, s, \; k = -r+1, -r+2, \ldots \tag{8.4.11}$$

$$\mathbf{u}_k^{n+1} = \sum_{m=-1}^{s} S_k^m \mathbf{u}_1^{n-m} + \mathbf{g}_k^n,$$

$$k = -r+1, \ldots, 0 \; n = s, s+1, \ldots \tag{8.4.12}$$

where Q_j, S_k^m, etc. are as in Section 8.2.3. We also assume that we again satisfy Assump 8.1–Assump 8.4. And as we have done in the past to make our results more palatable, *we assume that* $\mathbf{G}_k^n$ *and* $\mathbf{f}_k^m$ *are zero* (if not, we solve the initial–value problem associated with $\mathbf{G}_k^n$ and $\mathbf{f}_k^m$ and subtract off the solution) and that difference equation (8.4.10) does not contain a zeroth order term (if it did, we would eliminate it and apply Proposition 8.2.4). In this setting, we can use Definition 8.2.3 with $\alpha_0 = 0$ as our definition of stability.

To be able to directly use the Gustafsson result, we assume that Δt and Δx are such that $\Delta t/\Delta x$ is constant. This assumption then allows us to eliminate Δx from our conditions and discuss all accuracy relative to Δt. We might add that this approach is reasonably common, but we have not done this often earlier in this text.

The definition of consistency used by Gustafsson is similar to what we have used earlier. Because we want to allow different orders of accuracy for the boundary condition and difference equation, we must describe the consistency pointwise. Specifically, we let $\mathbf{v}$ denote the solution to the given initial–boundary–value problem and define $\boldsymbol{\tau}_k^n$ by

$$\Delta t \boldsymbol{\tau}_k^n = Q_{-1}\mathbf{v}_k^{n+1} - \sum_{j=0}^{s} Q_j \mathbf{u}_k^{n-j} - \Delta t \mathbf{G}_k^n, \quad k = 1, 2, \ldots, \; n = s, s+1, \ldots$$

$$\Delta t \boldsymbol{\tau}_k^n = \mathbf{v}_k^{n+1} - \sum_{m=-1}^{s} S_k^m \mathbf{v}_1^{n-m} - \mathbf{g}_k^n, \; k = -r+1, \ldots, 0, \; n = s, s+1, \ldots .$$

The difference scheme and/or the boundary conditions are said to be of m-th order if $\|\boldsymbol{\tau}_k^n\| = \mathcal{O}(\Delta t^m)$, where we consider the indices $k = 1, 2, \ldots$ when we consider the order of the difference scheme and the indices $k = -r+1, \ldots, 0$ when we consider the order of the boundary condition. We are now able to state the following version of Theorem 2.1, ref. [17], page 399.

Theorem 8.4.1 *Suppose that difference scheme (8.4.10) is an m-th order approximation to a given hyperbolic partial differential equation and that (8.4.12) is an $(m-1)$-st order approximation to the boundary conditions for $m \geq 1$. Then if difference (8.4.10)-(8.4.12) is stable, the solution to the difference scheme converges order m to the solution of the partial differential equation.*

Remark 1: We understand that we have been terribly vague about the appearance of the relevant partial differential equation involved. However, it is assumed that by this time, the reader is very aware that the difference equations and partial differential equations to which the above theorem will apply are the difference equations and partial differential equations that we have been considering throughout this chapter. More importantly, it must be made clear that the boundary conditions on which the order assumption is made in the theorem are generally a combination of analytic boundary conditions that were given as part of the initial–boundary–value problem, boundary conditions that were generated from analytic boundary conditions as we did in Section 8.2.4 and numerical boundary conditions. As usual, it is the numerical boundary conditions that are most difficult to handle.

Remark 2: We should realize that the above theorem is a variation of the statement of the Lax Theorem. As in the Lax Theorem, we have consistency (and order of approximation) and stability which implies convergence.

Remark 3: The result given in refs.. [16] and [17] is given with respect to slightly different norms from those we used earlier in this chapter. As we have stated before, we are not generally too particular as to with which

norm we can get convergence, as long as we can get some sort of convergence. Also, since it is not our desire to consider the proof of Theorem 8.4.1, we will not consider the stability definition used by Gustafsson.

We are now able to apply Theorem 8.4.1 along with the consistency considered in Section 8.4.2 and the stability considered throughout most of this chapter to obtain the following results for the right quarter plane problem for the one way wave equation with $a < 0$.

- The Lax-Wendroff scheme (8.2.34) along with numerical boundary condition (8.2.42) (see also (8.4.7)) is convergent of order one for $-1 \leq R \leq 0$.
- The Lax-Wendroff scheme (8.2.34) along with numerical boundary condition (8.2.57) is convergent of order two for $-1 \leq R \leq 0$.
- The Lax-Wendroff scheme (8.2.34) along with numerical boundary condition (8.2.56), $j = 2$ (see also (8.4.9)), is convergent of order two for $-1 \leq R \leq 0$.
- The Crank-Nicolson scheme (8.2.45) along with numerical boundary condition (8.2.46) (see also (8.4.7)) is convergent of order one for $R \leq 0$.
- The Crank-Nicolson scheme (8.2.45) along with numerical boundary condition (8.2.50) (see also (8.4.8)) is convergent of order one for $-2 < R \leq 0$.
- The Crank-Nicolson scheme (8.2.45) along with numerical boundary condition (8.2.57) is convergent of order two for $-2 \leq R \leq 0$.

And finally, we must realize that we obtain an analogous theorem for left quarter plane problems, and when these are both used in conjunction with Theorem 8.2.2, we obtain a convergence result that applies to initial–boundary–value problems defined on an interval.

8.5 Schemes Without Numerical Boundary Conditions

In this section we give a short discussion of stability (and, when we prove stability along with consistency, we get convergence) of difference schemes for initial–boundary–value problems that we were tempted to refer to as the **easiest case**. In Examples 3.1.6 and 3.1.8 we considered approximating the solution to an initial–boundary–value problem by a one–sided difference scheme. Because the one–sided scheme reached in the correct direction, we did not need a numerical boundary condition. Because the resulting matrix

(in the matrix formulation of the scheme) was not symmetric, we obtained only necessary conditions for convergence.

It should be clear that since we included analysis for analytic boundary conditions in our previous GKSO stability analyses (when the analytic boundary condition was the only boundary condition needed at one end of the interval or when one or more of the boundary conditions was an analytic boundary condition, but we needed more), the GKSO theory will also apply when we have analytic boundary conditions and we do not need any numerical boundary conditions. Also, one might suspect that since treatment of analytic boundary conditions in previous sections was always easy, the analysis here should be easy. And finally, since approximating the partial differential equation and analytic boundary conditions are by now quite routine, consistency and, hence, convergence follow immediately.

As an example, we consider the convergence of the difference scheme considered in Example 3.1.6.

Example 8.5.1 Discuss the convergence of difference scheme

$$u_k^{n+1} = (1+R)u_k^n - Ru_{k+1}^n, \quad k = 0 \ldots, M-1 \tag{8.5.1}$$

$$u_M^{n+1} = 0 \tag{8.5.2}$$

$$u_k^0 = f(k\Delta x), \quad k = 0, \ldots, M, \tag{8.5.3}$$

where a and, hence, R are less than zero.

Solution: Recall that in Section 8.2.1 (page 4) we noted that we can solve the initial–value problem associated with equation (8.5.1) and initial condition (8.5.3) extended to $\mathbb{R}$, subtract off this solution, and be left with the scheme with zero initial conditions. As we have always done in the past, we will assume that this is done, and, hence, consider that $f = 0$. We note that to make it possible to solve the initial–value problem, we must assume that $-1 \leq R \leq 0$. This is no problem, because this is also a necessary assumption for the GKSO theory.

As usual, we consider the left and right quarter plane problems separately. We begin by considering the left quarter plane problem, i.e., we consider the difference scheme

$$u_k^{n+1} = (1+R)u_k^n - Ru_{k+1}^n, \quad k = \ldots, M-2, M-1 \tag{8.5.4}$$

$$u_M^{n+1} = 0 \tag{8.5.5}$$

$$u_k^0 = 0, \quad k = \ldots, M, \tag{8.5.6}$$

We first take the discrete Laplace transform of equations (8.5.4) and (8.5.5) to obtain the resolvent equation

$$z\tilde{u}_k = (1+R)\tilde{u}_k - R\tilde{u}_{k+1}, \quad k = \ldots, M-2, M-1 \tag{8.5.7}$$

and the transformed boundary condition

$$\tilde{u}_M = 0. \tag{8.5.8}$$

If we look for a solution of equation (8.5.7) of the form $\tilde{u}_k = \phi\kappa^k$, we get

$$\{R\kappa + [z - (1+R)]\}\,\phi = 0$$

or

$$\kappa = -\frac{1}{R}\left[z - (1+R)\right]. \tag{8.5.9}$$

If we consider z such that $|z| > 1$, we see that

$$\begin{aligned} |\kappa| &= \frac{1}{|R|}|z - (1+R)| \\ &\geq \frac{1}{|R|}\left[|z| - |(1+R)|\right] \text{ (Recall that } 1+R \geq 0) \\ &> \frac{1}{|R|}\left[1 - (1+R)\right] \\ &= 1. \end{aligned}$$

Since we are considering the left quarter plane problem, κ's associated with any eigenvalue of the form $\tilde{u}_k = \phi\kappa^k$ will satisfy $|t\kappa| > 1$. Substituting $\tilde{u}_k = \phi\kappa^k$ into the transformed boundary condition (8.5.8) gives $\phi\kappa^M = 0$. Therefore, ϕ must be zero and there are no eigenvalues.

From the above argument involving the transformed boundary condition (8.5.8), equations (8.2.117)–(8.5.8) will also not have any solutions with $|z| = 1$, i.e., equations (8.2.117)–(8.5.8) will have no generalized eigenvalues, and *difference scheme (8.5.4)–(8.5.6) is stable.*

We should realize that all of the analyses for schemes that do not require numerical boundary conditions will be as trivial as the one above. However, this approach does allow one to find sufficient conditions for stability for schemes for which in Chapter 5 we could find only necessary conditions for stability.

8.6 Parabolic Initial–Boundary–Value Problems

All of the work done so far in this chapter has been devoted to hyperbolic initial–boundary–value problems. The reasons for this are that we had almost no sufficient conditions for convergence of initial–boundary–value schemes for hyperbolic equations and the fact that the schemes for hyperbolic problems often require numerical boundary conditions that cause many difficulties. However, the GKSO theory can also be used to prove stability for schemes for parabolic initial–boundary–value problems. We will try to provide some of the available results and techniques. For further results regarding GKSO theory for parabolic initial–boundary–value problems, see refs. [69], [56], [65] and [18].

We begin by emphasizing that the setting and forms of difference operators are generally as they were for hyperbolic equations. The results available in the literature use both *sup* norms and some sort of ℓ_2 norm. As we have done in the past, we will first consider the stability of the scheme. The techniques that we will use are essentially the same as those used for hyperbolic equations used in the previous sections of this chapter. Hence, we will take the discrete Laplace transform of our scheme and analyze the resulting difference equation. We will follow a combination of the approaches given in refs. [69], [56] and [65].

We simplify the problems considered here by *considering only difference schemes that require the same number of boundary conditions as do the partial differential equations, i.e., we do not allow for numerical boundary conditions.* We emphasize that this eliminates consideration of higher order schemes for parabolic equations that reach more than one point to the right and/or left. In addition, we make the assumption that *the parabolic initial–boundary–value problem we are considering is well-posed.* As we have done so often before, we consider the stability of the right and left quarter plane problems separately (and also as we have done often before, we give the results for the right quarter plane problem and then state that the left quarter plane results are analogous). There is only one difference between the considerations of parabolic and hyperbolic initial–boundary–value problem schemes. As Theorem 8.6.1 given below shows, when we consider schemes for parabolic initial–boundary–value problems we do not have to worry about $z = 1$ being a generalized eigenvalue.

We consider a difference equation

$$Q_{-1}\mathbf{u}_k^{n+1} = \sum_{j=0}^{s} Q_j \mathbf{u}_k^{n-j} + \Delta t \mathbf{G}_k^n \tag{8.6.1}$$

along with boundary conditions

$$\mathbf{u}_k^{n+1} = \sum_{m=-1}^{s} S_k^m \mathbf{u}_1^{n-m} + \mathbf{g}_k^n, \quad k = -r+1, \ldots, 0, \tag{8.6.2}$$

and initial conditions $\mathbf{u}_k^m = \mathbf{f}_k^m$, $m = 0, 1, \ldots, s$, $k = -r+1, -r+2, \ldots$ (where all of the notation is as in Section 8.2.3). As we stated earlier, we assume that difference equation (8.6.1)–(8.6.2) along with the initial condition is an approximation of a well–posed parabolic initial–boundary–value problem and that (8.6.2) provides the same number of boundary conditions that we are given for our analytic problem.

We proceed as we did in the case of hyperbolic initial–boundary–value schemes, take the discrete Laplace transform of equations (8.6.1) and (8.6.2) and obtain the resolvent equation

$$\left(Q_{-1} - \sum_{j=0}^{s} z^{-j-1} Q_j \right) \tilde{\mathbf{u}}_k = \boldsymbol{\theta} \tag{8.6.3}$$

and homogeneous transformed boundary condition

$$\tilde{\mathbf{u}}_k - \sum_{m=1}^{s} S_k^m \tilde{\mathbf{u}}_1 = \boldsymbol{\theta}. \tag{8.6.4}$$

We then state the following theorem.

Theorem 8.6.1 *Difference scheme (8.6.1)–(8.6.2) is stable if and only if eigenvalue problem (8.6.3)–(8.6.4) has no eigenvalues and no generalized eigenvalues, $z \neq 1$.*

We apply Theorem 8.6.1 in the same way we applied Propositions 8.2.13 and 8.2.14 and Theorem 8.2.19. We first illustrate the application of the above theorem by proving the stability of the right quarter plane problem of the initial–boundary–value scheme already proved stable in HW3.2.2.

Example 8.6.1 Prove that the following difference scheme is stable.

$$u_k^{n+1} = ru_{k-1}^n + (1-2r)u_k^n + ru_{k+1}^n, \quad k = 0, \ldots, \; n = 0, 1, \ldots \tag{8.6.5}$$

$$u_1^{n+1} - u_{-1}^{n+1} = 0, \quad n = 0, 1, \ldots \tag{8.6.6}$$

$$u_k^0, = f(k\Delta x), \quad k = 0, \ldots, \tag{8.6.7}$$

for $r < \frac{1}{2}$.

Solution: We begin by rewriting difference equation (8.6.5), $k = 0$, along with boundary condition (8.6.6) as

$$u_0^{n+1} = (1-2r)u_0^n + 2ru_1^n. \tag{8.6.8}$$

Taking the discrete Laplace transform of equations (8.6.5) and (8.6.8) gives us the resolvent equation

$$z\tilde{u}_k = r\tilde{u}_{k-1} + (1-2r)\tilde{u}_k + r\tilde{u}_{k+1}, \quad k = 1, \ldots \tag{8.6.9}$$

and homogeneous transformed boundary condition

$$z\tilde{u}_0 = (1-2r)\tilde{u}_0 + 2r\tilde{u}_1. \tag{8.6.10}$$

We look for solutions to equations (8.6.9)–(8.6.10) in the form $\tilde{u}_k = \phi\kappa^k$. Substituting this expression into equation (8.6.10) leaves us with

$$z = (1-2r) + 2r\kappa$$

or

$$\kappa = \frac{1}{2r}\left[z - (1-2r)\right]. \tag{8.6.11}$$

Then for $|z| > 1$ (and recalling that we assume that the scheme for the analogous Cauchy problem (equation (8.6.5)) is stable, i.e., $r \leq \frac{1}{2}$), we see that

$$|\kappa| \geq \frac{1}{2r}\left(|z| - |1-2r|\right) > \frac{1}{2r}\left[1 - (1-2r)\right] = 1.$$

Hence, *equations (8.6.9)–(8.6.10) have no eigenvalues.*

We next look for generalized eigenvalues of equations (8.6.9)–(8.6.10). We substitute $\tilde{u}_k = \phi\kappa^k$ into equation (8.6.9), simplify and get the characteristic equation

$$r\kappa^2 + (1-2r-z)\kappa + r = 0. \tag{8.6.12}$$

Replacing κ in equation (8.6.12) by the value of κ given in equation (8.6.11) gives us the following quadratic equation for z.

$$z^2 + 2(2r-1)z + 1 - 4r = 0. \tag{8.6.13}$$

Hence, $z = 1$ or $z = 1 - 4r$. Since we are interested in solutions of the form $z = e^{i\theta}$, we have the solution $z = 1$. By Theorem 8.6.1, we know that we are not concerned about the root $z = 1$. We note that since we require that $r < \frac{1}{2}$, we do not have to concern ourselves with the solution $z = -1$ that occurs when $r = \frac{1}{2}$. Therefore, *difference scheme (8.6.5)–(8.6.7) is conditionally stable with stability condition* $r < \frac{1}{2}$.

Remark 1: We note that in HW3.3.2 we considered a scheme (8.6.5)–(8.6.7) on $[0,1]$ with a Dirichlet boundary condition at $x = 1$. To prove the stability of the difference scheme considered in HW3.2.2, we would next have to consider the left plane problem with the Dirichlet boundary condition at $x = 1$. The analysis of this left plane problem is easy.

Remark 2: In Example 8.6.1 above, we sidestepped the issue of considering the generalized eigenvalue $z = -1$ by requiring that $r < \frac{1}{2}$. This is also what is done in the literature (where I do not know whether they are sidestepping anything). The generalized eigenvalue $z = -1$ is associated with a solution $u_k^n = \phi(-1)^n(-1)^k$. Though this type of solution is irritating, since it is not growing, it is one that we would want to be stable with respect to the *sup* norm. It appears that this solution implies that the scheme is not stable in the ℓ_2 norm for $r = \frac{1}{2}$.

Remark 3: Instead of combining difference equation (8.6.5), $k = 0$, with boundary condition (8.6.6) to obtain boundary condition (8.6.8), we note that it is permissible to consider difference equation (8.6.5) for $k = 0, 1, \ldots$ and boundary condition (8.6.6) directly. If we transform boundary condition (8.6.6), we get $\tilde{u}_{-1} = \tilde{u}_1$. Setting $\tilde{u}_k = \phi\kappa^k$ leaves us with $\kappa^{-1} = \kappa$ or $\kappa = \pm 1$. This is the same result we obtained in the above example, so the analysis will follow exactly as it did above.

We next consider a scheme that again uses difference scheme (8.6.5), but we now use the first order approximation of the Neumann boundary condition,

$$\frac{1}{\Delta x}\delta_+ u_0^{n+1} = 0. \tag{8.6.14}$$

You should recall that we first introduced this scheme (really we considered this scheme along with a Dirichlet boundary condition at $x = 1$) in Section 2.6.4, Part 1, and analyzed the norm consistency and stability in Sections 2.3 and 3.2, Part 1.

Example 8.6.2 Discuss the stability of difference scheme (8.6.5), (8.6.14).

Solution: The characteristic equation associated with difference scheme (8.6.5) was calculated in the previous example. If we take the discrete Laplace transform of boundary condition (8.6.14) and replace $\tilde{u}_k$ by $\phi\kappa^k$, we find that $\kappa = 1$. Substituting $\kappa = 1$ into the characteristic equation (8.6.12), we see that $\kappa = 1$ is associated with $z = 1$. Therefore, there are no eigenvalues and no generalized eigenvalues, $z \neq 1$. Therefore, *difference scheme (8.6.5), (8.6.14) is stable for* $r \leq \frac{1}{2}$.

We next include an example that examines the three popular boundary condition approximations for the Neumann boundary condition used with the Crank-Nicolson scheme: the second order, the first order and the zeroth order boundary conditions.

Example 8.6.3 Discuss the stability of the Crank-Nicolson difference scheme

$$u_k^{n+1} - \frac{r}{2}\delta^2 u_k^{n+1} = u_k^n + \frac{r}{2}\delta^2 u_k^n \tag{8.6.15}$$

along with the given initial condition u_k^0 and three potential boundary conditions at $k = 0$:

$$u_{-1}^{n+1} = u_1^{n+1} \tag{8.6.16}$$

$$u_0^{n+1} = u_1^{n+1} \tag{8.6.17}$$

$$u_0^{n+1} = u_1^n. \tag{8.6.18}$$

Solution: Recall that equations (8.6.16)–(8.6.18) are the second, first and zeroth order approximation of the boundary condition $v_x(0,t) = 0$, respectively. We should note that the results found in this example will apply equally well to a nonhomogeneous boundary condition. When we have a nonhomogeneous boundary condition, we still consider the homogeneous transformed boundary condition. Recall also that the popularity of boundary condition (8.6.18) lies in its ease of implementation, and though it is of zeroth order, boundary condition (8.6.18) is used often. Boundary condition (8.6.18) is implemented much as is a Dirichlet boundary condition.

Boundary condition (8.6.16) Before we analyze boundary condition (8.6.16), we apply equation (8.6.15) at $k = 0$, use boundary condition (8.6.16) to replace the u_{-1}^{n+1} terms and obtain the following boundary condition

$$(1+r)u_0^{n+1} - ru_1^{n+1} = (1-r)u_0^n + ru_1^n. \tag{8.6.19}$$

Hence, we consider boundary condition (8.6.19) along with difference equation (8.6.15) for $k = 1, 2, \ldots$.

We take the discrete Laplace transform of difference equation (8.6.15), set $\tilde{u}_k = \phi\kappa^k$ and simplify to get the characteristic equation

$$\frac{r}{2}(z+1)\kappa^2 + [1 - r - z(r+1)]\kappa + \frac{r}{2}(z+1) = 0. \tag{8.6.20}$$

Transforming boundary condition (8.6.19), setting $\tilde{u}_k = \phi\kappa^k$ and simplifying yields

$$z(1 + r - r\kappa) = 1 - r + r\kappa \tag{8.6.21}$$

or

$$z = \frac{1 - r + r\kappa}{1 + r - r\kappa}. \tag{8.6.22}$$

If we substitute z as given by equation (8.6.22) into equation (8.6.20) and solve the resulting equation for κ, we get $\kappa = \pm 1$. Inserting $\kappa = 1$ into equation (8.6.22) shows that $\kappa = 1$ is associated with $z = 1$. The value $\kappa = -1$ is associated with $z = (1-r)/(1+r)$. It is easy to see that for $r > 0$, $|z| = |(1-r)/(1+r)| < 1$. Hence, there are no eigenvalues and no generalized eigenvalues, $z \neq 1$, and *difference scheme (8.6.15)–(8.6.16) is stable for all* $r > 0$.

Boundary condition (8.6.17) Since we are using the same difference equation as in the previous analysis, we again obtain resolvent equation (8.6.20). If we transform boundary condition (8.6.17), set $\tilde{u}_k = \phi\kappa^k$ and simplify, we get $\kappa = 1$. Since $\kappa = 1$ is associated with $z = 1$, *difference scheme (8.6.15), (8.6.17) is unconditionally stable.*

Boundary condition (8.6.18) If we take the discrete Laplace transform boundary condition (8.6.18), set $\tilde{u}_k = \phi\kappa^k$ and simplify, we get $z = \kappa$. Setting $z = \kappa$ in characteristic equation (8.6.20) yields

$$\frac{r}{2}\kappa^3 - \left(\frac{r}{2} + 1\right)\kappa^2 - \left(\frac{r}{2} - 1\right)\kappa + \frac{r}{2} = 0. \tag{8.6.23}$$

Solving equation (8.6.23) for κ gives $\kappa_1 = 1$ and $\kappa_\pm = \frac{1}{r}[1 \pm \sqrt{1 + r^2}]$.

The value $\kappa_1 = 1$ is again associated with $z = 1$, which we can ignore. The root κ_+ is such that $|\kappa_+| > 1$, so it is not relevant. If you plot $(1/r)[1 - \sqrt{1+r^2}]$ for $r > 0$, it is easy to see that $|z| = |\kappa_-| < 1$ for all $r > 0$. Hence, there are no eigenvalues and no generalized eigenvalues, $z \neq 1$, so *difference scheme (8.6.15), (8.6.18) is stable.*

Remark 1: In the last step of the analysis done in the above example, we stated that we plot $|\kappa_-|$ to verify that $|z| = |\kappa_-| < 1$. Of course, this result can also be done analytically (and hence, rigorously). See HW8.6.1. Often, graphics will provide a quick and dirty way to verify (not prove) such an inequality.

Remark 2: As in Example 8.6.1, instead of combining difference equation (8.6.15), $k = 0$, with boundary condition (8.6.16) to obtain boundary condition (8.6.19), it is permissible to consider difference equation (8.6.15) for $k = 0, 1, \ldots$ and boundary condition (8.6.16) directly. We transform boundary condition (8.6.16) and get $\tilde{u}_{-1} = \tilde{u}_1$. Setting $\tilde{u}_k = \phi\kappa^k$ leaves us with $\kappa^{-1} = \kappa$ or $\kappa = \pm 1$. This is the same result we obtained in the above example, so the analysis will follow exactly as it did above.

We next include two stability results for schemes for which we earlier had only necessary conditions. We first prove stability of the right quarter plane problem for a difference scheme previously considered as an initial–value scheme in Example 3.1.3 and as an initial–boundary–value scheme in HW3.4.1. We then consider the right quarter plane problem for a scheme with a difficult boundary condition considered in Example 3.2.5 (a more general problem was considered in Example 3.3.1).

Example 8.6.4 Discuss the stability of the following difference scheme

$$u_k^{n+1} = u_k^n - \frac{R}{2}\delta_0 u_k^n + r\delta^2 u_k^n, \quad k = 1, \ldots, \; n = 0, 1, \ldots \tag{8.6.24}$$

$$u_0^{n+1} = 0, \quad n = 0, 1, \ldots \tag{8.6.25}$$

$$u_k^0 = f(k\Delta x), \quad k = 0, 1, \ldots .$$

Solution: We begin by taking the discrete Laplace transform of equations (8.6.24)–(8.6.25) to get the resolvent equation

$$z\tilde{u}_k = \tilde{u}_k - \frac{R}{2}\delta_0 \tilde{u}_k + r\delta^2 \tilde{u}_k, \quad k = 1, \ldots \tag{8.6.26}$$

and the transformed boundary condition

$$\tilde{u}_0 = 0. \tag{8.6.27}$$

If we look for a solution to equations (8.6.26)–(8.6.27) in the form $\tilde{u}_k = \phi\kappa^k$, equation (8.6.27) implies that $\phi = 0$. Hence, for $|z| \geq 1$ there are no solutions to equations (8.6.26)–(8.6.27), or no eigenvalues or generalized eigenvalues. Therefore, **difference scheme (8.6.24)–(8.6.25) is conditionally stable with the condition inherited from its analogous initial–value scheme,** $R^2/2 \leq r \leq \frac{1}{2}$.

If we want to consider the entire difference scheme considered in HW3.4.1, we must now consider the appropriate left quarter plane problem. The analysis of the left quarter plane problem will be almost identical to that done above.

Example 8.6.5 Discuss the stability of the following difference scheme.

$$u_k^{n+1} = u_k^n + r\delta^2 u_k^n, \quad k = 0, 2, \ldots, \; n = 0, 1, \ldots \tag{8.6.28}$$

$$\frac{u_1^n - u_{-1}^n}{2\Delta x} = h_1 u_0^n \tag{8.6.29}$$

$$u_k^0 = f(k\Delta x), \quad k = 0, 1, \ldots \tag{8.6.30}$$

where h_1 is constant and $h_1 \geq 0$.

Solution: We begin by using difference equation (8.6.28) at $k = 0$ along with boundary condition (8.6.29) to write

$$u_0^{n+1} = \big[1 - 2(1 + h_1\Delta x)r\big]u_0^n + 2ru_1^n. \tag{8.6.31}$$

We then consider the scheme consisting of difference equation (8.6.28) for $k = 1, 2, \ldots$, equation (8.6.31) and initial condition (8.6.30). We take the discrete Laplace transform of equations (8.6.28) and (8.6.31) to get the resolvent equation

$$z\tilde{u}_k = r\tilde{u}_{k-1} + (1 - 2r)\tilde{u}_k + r\tilde{u}_{k+1} \tag{8.6.32}$$

and the transformed boundary condition

$$z\tilde{u}_0 = \big[1 - 2(2 + h_1\Delta x)\big]\tilde{u}_0 + 2r\tilde{u}_1. \tag{8.6.33}$$

If we look for solutions in the form $\bar{u} = \phi\kappa^k$, equations (8.6.32) and (8.6.33) become

$$r\kappa^2 + (1 - 2r - z)\kappa + r = 0 \tag{8.6.34}$$

and

$$z = \big[1 - 2r(1 + h_1\Delta x)\big] + 2r\kappa, \tag{8.6.35}$$

respectively. We proceed as we did in Example 8.6.1, solve equation (8.6.35) for κ and see that if $|z| > 1$ and

$$r \leq \frac{1}{2(1 + h_1\Delta x)}, \tag{8.6.36}$$

then

$$\begin{aligned} |\kappa| &\geq \frac{1}{2r}\{|z| - |1 - 2r(1 + h_1\Delta x)|\} \\ &> \frac{1}{2r}\{1 - [1 - 2r(1 + h_1\Delta x)]\} \\ &= 1 + h_1\Delta x. \end{aligned}$$

Hence, we see that if r satisfies (8.6.34), then equations (8.6.32)–(8.6.33) have no eigenvalues.

To evaluate the possibility of generalized eigenvalues, we solve equation (8.6.35) for κ, substitute the result into equation (8.6.34) and get

$$z^2 - 2(1 - 2r)z + 1 - 4r - 4r^2h_1^2\Delta x^2 = 0.$$

Solving for z yields

$$z = \begin{cases} 1 - 2r + 2r\sqrt{1 + h_1^2\Delta x^2} \\ 1 - 2r - 2r\sqrt{1 + h_1^2\Delta x^2}. \end{cases}$$

Since we are interested in values of z such that $|z| = 1$ and since z is real, we look for $z = \pm 1$. It is easy to see that the first solution is always greater than 1. We see that the second root, $z = 1 - 2r - 2r\sqrt{1 + h_1^2\Delta x^2}$, equals 1 only when $r = 0$ (which we do not care about) or when $1 + \sqrt{1 + h_1^2\Delta x^2}$ is zero (which is impossible). The second root, $z = 1 - 2r - 2r\sqrt{1 + h_1^2\Delta x^2}$, equals -1 when $r = \frac{1}{1 + \sqrt{1 + h_1^2\Delta x^2}}$. Since

$$\frac{1}{1 + \sqrt{1 + h_1^2\Delta x^2}} > \frac{1}{2(1 + h_1\Delta x)}$$

for $h_1 > 0$ (we considered $h_1 = 0$ already in Example 8.6.1), $z \neq -1$ for $r \leq \frac{1}{2(1 + h_1\Delta x)}$. Hence, there are no generalized eigenvalues and *difference scheme (8.6.28)–(8.6.30) is conditionally stable with condition* $r \leq \frac{1}{2(1 + h_1\Delta x)}$. Note that the sufficient condition found here is different from the necessary condition found in Example 3.2.5 and 3.3.1 (though not much different).

After we obtain stability results as we have done above, we next want a theorem analogous to Theorem 2.5.2, 2.5.3 or 8.4.1 that ensures convergence if we have stability and consistency. More so, we would like a theorem like Theorem 8.4.1 that will allow for one degree of less accuracy for the boundary condition than for the scheme and still give convergence to the accuracy of the scheme. Recall that Theorem 2.5.3 did not give us this result, and consequently, we lost an order of accuracy for the second order approximation of the Neumann boundary condition and could not use Theorem 2.5.3 to prove the convergence of the scheme with the first order accurate approximation of the Neumann boundary condition. See Section 3.1.2, Part 1.

Though we were unable to find the version that we want, the theorem is surely true. The result that comes closest to meeting our needs is Theorem 11.3.1, ref. [18]. The result does not assume that the accuracy of the boundary conditions is one less than the accuracy of the interior scheme. However, the boundary conditions are written separately from the difference equation and do not require that they be written as a part of the equation $\mathbf{u}^{n+1} = Q\mathbf{u}^n$, i.e., they do not use the norm consistency that we used in Definitions 2.3.2 and 2.3.3, Part 1. In this way the first and second order approximations of Neumann boundary conditions are first and second order accurate, respectively (a much nicer result). The only difficulty with the result for our use is that the theorem is for a semi-discrete scheme rather than the fully discrete schemes that we have used. We will not attempt to prove the fully discrete version. We will instead proceed as if we had the appropriate convergence theorem.

HW 8.6.1 Show that $|\frac{1}{r}[1 - \sqrt{1 + r^2}]| < 1$ for all $r > 0$.

9

Conservation Laws

9.1 Introduction

The class of conservation laws is a very important class of partial differential equations because as their name indicates, they include those equations that model conservation laws of physics (mass, momentum, energy, etc.). In Sections 1.6, 4.2.2 and 5.8.1, we used "the conservation law approach" to derive difference equation approximations to certain linear partial differential equations. This approach was related to the subject of this chapter in that we considered the equation as if it had come from some conservation law and proceeded to derive a difference approximation that would respect the conservation principle. The added difficulty that we shall address in this chapter is that conservation laws are generally nonlinear. As we shall see, this strongly affects both the solution's behavior and the numerical solution.

We consider the numerical solution of partial differential equations of the form

$$\frac{\partial \mathbf{v}}{\partial t} + \frac{\partial}{\partial x}\mathbf{F}(\mathbf{v}) = \boldsymbol{\theta} \tag{9.1.1}$$

where $\mathbf{v}$ is a K-vector and $\mathbf{F}$ is a function that maps $\mathbb{R}^K$ into $\mathbb{R}^K$. Unless we specifically state otherwise, we will assume that $\mathbf{F}''(v) \geq 0$, i.e., that $\mathbf{F}$ is **convex**. An equation of the form (9.1.1) is said to be in **conservation form** and is called a **conservation law**. If we integrate equation (9.1.1) with respect to x and t from a to b and t_1 to t_2, respectively, and perform the integration with respect to t for the first term and with respect to x

for the second term, we obtain

$$\int_a^b \mathbf{v}(x,t_2)\,dx - \int_a^b \mathbf{v}(x,t_1)\,dx = -\left(\int_{t_1}^{t_2} \mathbf{F}(\mathbf{v}(b,t))\,dt - \int_{t_1}^{t_2} \mathbf{F}(\mathbf{v}(a,t))\,dt\right). \quad (9.1.2)$$

We should note that a linear version of equation (9.1.2) can be obtained by integrating equation (1.6.2) with respect to t from t_1 to t_2. As we did with equation (1.6.2), we refer to equation (9.1.2) as the **integral form of conservation law** (9.1.1). The physical interpretation of equation (9.1.2) is that the change in the amount of conserved material in the interval $[a,b]$ between times t_1 and t_2 is due to the flux of the material across the boundaries $x = a$ and $x = b$ during the time interval from $t = t_1$ to $t = t_2$.

Though we shall focus on one dimensional problems (in fact, one dimensional scalar problems), multidimensional conservation laws are written in the same form by adding $\frac{\partial}{\partial y}\mathbf{G}(\mathbf{v})$, $\frac{\partial}{\partial z}\mathbf{H}(\mathbf{v})$, etc. terms. We will generally treat conservation laws that are assumed to be strictly hyperbolic, but will occasionally include parabolic equations that are dominated by the hyperbolic part of the equation. **Strictly hyperbolic conservation laws** are partial differential equations of the form of equation (9.1.1) where $\mathbf{F}'$ has distinct, real eigenvalues.

Besides the linear equations mentioned earlier, we have previously been introduced to conservation laws in HW0.0.1, HW0.0.2 and HW0.0.3. To see that Burgers' equation is a conservation law, we rewrite the inviscid Burgers' equation as

$$v_t + \left(\frac{1}{2}v^2\right)_x = 0. \quad (9.1.3)$$

Clearly, the viscous Burgers' equation can be written in a similar form. The Euler equations in HW0.0.3 were given in conservation law form, and in order to use our methods, we re-wrote them in a nonconservation law form. In general, we can always perform the differentiation with respect to x and rewrite equation (9.1.1) as

$$\mathbf{v}_t + \mathbf{F}'(\mathbf{v})\mathbf{v}_x = \boldsymbol{\theta} \quad (9.1.4)$$

where $\mathbf{F}'(\mathbf{v})$ is the Fréchet derivative of $\mathbf{F}$ with respect to $\mathbf{v}$ and is given by the $K \times K$ matrix of partial derivations of $\mathbf{F}$. Equation (9.1.4) is referred to as the nonconservative form of equation (9.1.1). There are times when we will want to use the conservative form of the equation, and there are times when we will want to use the nonconservative form of the equation.

The reason that numerical methods for solving conservation laws are as good as they are is that they take into account properties of the solutions

of conservation laws. For this reason, we will first discuss analytic conservation laws and their solutions. We will not try to teach the complete analytic theory of conservation laws. We will try to include the material necessary for the reader to appreciate the types of equations (and their solutions) that we are trying to solve and to understand the numerical methods for approximating the solutions of conservation laws. After we have some properties of the solutions of conservation laws, we shall introduce numerical methods for solving them. For more information on conservation laws, see ref. [62] or ref. [73]. For more information on the numerical solution of conservation laws, see refs. [37], [74] or [11].

9.2 Theory of Scalar Conservation Laws

We begin our discussion of solutions to conservation laws by considering the scalar version of equation (9.1.1),

$$v_t + \frac{\partial}{\partial x} F(v) = 0 \tag{9.2.1}$$

where we assume that $F'' \geq 0$. The derivation of a conservation law generally begins by considering the amount of substance in an arbitrary interval $[x_0, x_1]$, $\int_{x_0}^{x_1} v(x,t)\,dx$. The conservation law then states that the rate of change of this substance is equal to the flux of substance across the boundaries (plus any of the substance that is created or destroyed in the regions, but since we are interested in homogeneous equations, we will ignore this contribution), i.e.,

$$\frac{d}{dt}\int_{x_0}^{x_1} v(x,t)\,dx = [F]_{x_0}^{x_1} \tag{9.2.2}$$

where for now, F is vaguely defined to be the flux function. Interchanging the differentiation and integration on the left hand side of equation (9.2.2) and writing the right hand side as an integral gives

$$\int_{x_0}^{x_1} (v_t - F_x)\,dx = 0.$$

Then, using the fact that the interval $[x_0, x_1]$ is arbitrary and that the function $v_t - F_x$ is smooth, we obtain equation (9.2.1).

The derivation of the vector form is the same, where v and F are replaced by $\mathbf{v}$ and $\mathbf{F}$, respectively. The derivation of a scalar, multidimensional conservation law (or each component of a vector valued, multidimensional conservation law) proceeds in the same manner, where the flux across the boundary (which is now a curve for two dimensional problems, a surface for three dimensional problems, etc.) is now the normal component of the

flux, and the Divergence Theorem is used to change the surface (line) integral into a volume (area) integral (taking the place of rewriting $[F]_{x_0}^{x_1}$ as an integral).

9.2.1 Shock Formation

We next rewrite equation (9.2.1) in the nonconservative form

$$v_t + F'(v)v_x = 0. \tag{9.2.3}$$

Solutions to equation (9.2.3) behave somewhat like solutions to their linear counterparts, say equation (5.1.4). For example, we define the **characteristic curve** of equation (9.2.1) (or equation (9.2.3)) to be the solution to the differential equation

$$\frac{d}{dt}x(t) = F'(v(x(t),t)). \tag{9.2.4}$$

If we consider a solution of partial differential equation (9.2.3), $v = v(x,t)$, then along any characteristic curve (a curve $x = x(t)$ on which (9.2.4) is satisfied), we have

$$\begin{aligned} \frac{d}{dt}v(x(t),t) &= v_1(x(t),t)\frac{d}{dt}x(t) + v_2(x(t),t) && (9.2.5)\\ &= v_1(x(t),t)F'(v(x(t),t)) + v_2(x(t),t) \\ &= 0 && (9.2.6) \end{aligned}$$

since v is a solution to partial differential equation (9.2.3). We have used the notation v_1 and v_2 to denote the partial derivatives of v with respect to the first and second arguments, respectively. The partial derivatives are most often written as v_x and v_t, but this latter notation might be confusing in this situation. Hence, we have obtained the following result.

Proposition 9.2.1 *Along any characteristic curve defined by equation (9.2.4), the solution to partial differential equation (9.2.3) (or to (9.2.1)) is constant.*

Remark: We note that in the linear case where we consider the equation $v_t + av_x = 0$ where a is constant, equation (9.2.4) reduces to

$$\frac{d}{dt}x(t) = a,$$

so that the characteristic curves for equation (5.1.4), are given by $x(t) = at + C$, which is what we defined them to be in Chapter 5. Also, when we considered linear, hyperbolic initial–value problems

$$\begin{aligned} v_t + av_x &= 0, \quad x \in \mathbb{R},\ t > 0 \\ v(x,0) &= v_0(x), \quad x \in \mathbb{R}, \end{aligned}$$

we were able to write the solution as $v(x,t) = v_0(x - at)$. For nonlinear conservation laws, we obtain the following analogous result.

Proposition 9.2.2 *If v is a sufficiently smooth solution to the initial–value problem defined by conservation law (9.2.1) for $x \in \mathbb{R}$, $t > 0$ (or (9.2.3)) along with initial condition $v(x,0) = v_0(x)$, $x \in \mathbb{R}$, then v will satisfy*

$$v(x,t) = v_0(x - F'(v(x,t))t), \quad x \in \mathbb{R}, \ t > 0. \tag{9.2.7}$$

To prove the above result, it is sufficient to differentiate solution (9.2.7) with respect to t and x and show that $v_t + F'(v)v_x$ equals zero.

Remark: We notice that characteristic curves for the inviscid Burgers' equation are solutions to

$$\frac{d}{dt}x(t) = v(x(t),t). \tag{9.2.8}$$

The fact that the solution is constant along any characteristic will help us to begin to understand the solution behavior of solutions to equation (9.2.1). We first note that since v is constant along any characteristic and the characteristic curve must satisfy

$$\frac{d}{dt}x(t) = F'(v(x(t),t)),$$

the characteristics must be straight lines, $x(t) = F'(v_0)t + \text{constant}$, where v_0 is the constant value of v along the given characteristic. This situation is not too different from the results for the linear case. We recall, or look back at Figure 5.2.1 to see, that the characteristics associated with the linear equation $v_t + av_x = 0$ are straight lines. The difference between the results obtained in Chapter 5 and those that will be obtained here is that in the linear case the slope of all of the characteristics was the same, $1/a$. *In the nonlinear case, the slopes of the characteristics will generally be different, $1/F'(v_0)$ for various values of v_0.*

For example, consider the inviscid Burgers' equation on $[0,1]$,

$$v_t + \left(\frac{1}{2}v^2\right)_x = 0$$

along with the initial condition $v_0(x) = \sin 2\pi x$, $x \in [0,1]$. We have $F(v) = \frac{1}{2}v^2$ and $F'(v) = v$. Hence, the characteristic curves emanating from the $t = 0$ axis at the point x_0 is given by

$$x(t) = F'(v_0)t + \text{constant} = (\sin 2\pi x_0)t + x_0.$$

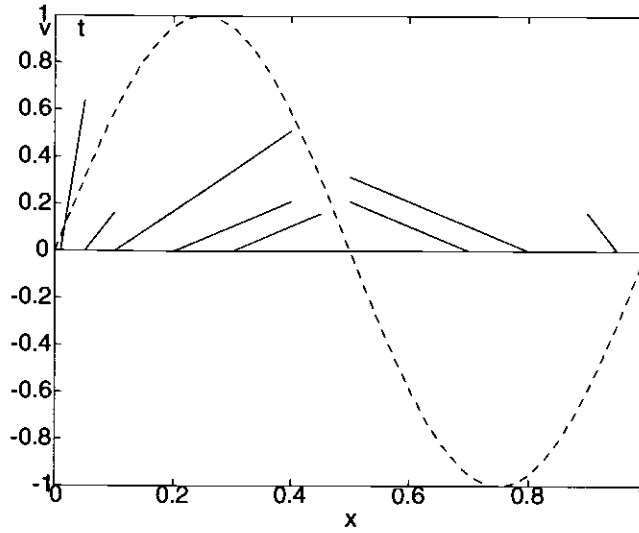

FIGURE 9.2.1. Plots of both the initial condition $v_0 = \sin 2\pi x$ (dotted curve) and the characteristic curves: straight lines with slope $1/v_0(x)$.

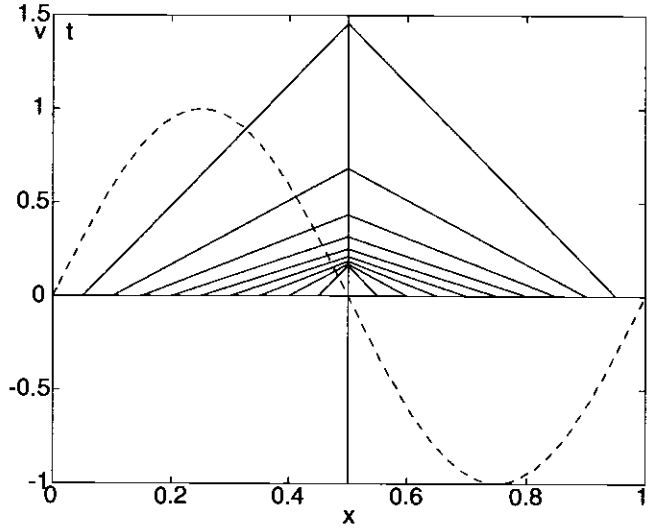

FIGURE 9.2.2. Plots of both the initial condition $v_0 = \sin 2\pi x$ (dotted curve) and the characteristic curves associated with $x = 0.05, 0.1, 0.15, \ldots, 0.95$.

In Figure 9.2.1, we give a rather strange plot, where the vertical axis represents both v_0 and t. This way we are able to plot both v_0 and the characteristics on the same graph. Note that at each point $x_0 \in [0, 1]$ the slope of the characteristic emanating from that point has slope $1/v_0(x_0) = 1/\sin 2\pi x_0$.

The behavior of the characteristic curves is really nicer than is indicated in Figure 9.2.1. If we draw the characteristics given in Figure 9.2.1 a little more carefully and more completely, we obtain the plot shown in Figure 9.2.2. We note that *the characteristic curves intersect.* This will generally be the case for conservation laws (equations of the form (9.2.1)). (It will not generally be the case that the characteristics will intersect along a vertical line as in Figure 9.2.2. In this case, the vertical line $x = 0$ is the characteristic associated with $x_0 = 0$ and all of the characteristic curves intersect along this line because of the symmetry in the initial condition.) What does this mean? When two characteristics $x = F'(v_0)t + x_0$ and $x = F'(v_1) + x_1$ intersect at the point (x, t), the solution at the point wants to be both v_0 (due to the characteristic curve $x = F'(v_0)t + x_0$) and v_1 (due to the characteristic curve $x = F'(v_1)t + x_1$), i.e., *the solution becomes double valued (or more) at points where the characteristics intersect.* As time proceeds past the time of intersection, the solution becomes triple valued and looks like a breaking wave (as is shown in Figure 9.2.3. We denote the time at which the characteristics first intersect by $t = T_b$ and refer to it as the **breaking point.**

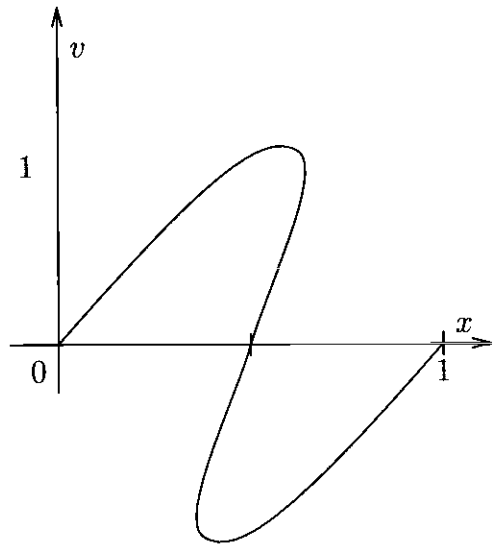

FIGURE 9.2.3. An example of what the solution is trying to become after the characteristics have crossed.

Of course, the above described scenario is not acceptable for a physical situation. We recall that the partial differential equations that we are considering are generally models of a physical situation. The conservation law must approximate the physical situation sufficiently well, or we must obtain a new model equation. Having a solution that is multiple valued and/or that breaks is not acceptable. In the physical situations that we are modeling, the functions do not become multivalued. When the plane flies faster than the speed of sound and we perceive that its characteristics have intersected, they have not. The conservation laws with intersecting characteristics are approximations of the physical system. There is often some sort of viscous term that we have neglected; the solution curves become very steep, but they do not come multivalued.

An explanation of what we are trying to say above is easier if we compare solutions to HW0.0.1 and HW0.0.2. When we finally got good solutions to HW0.0.1 (for small values of ν) and HW0.0.2 in Section 6.10.1. we saw that these solutions were almost identical. Mathematically, the solutions to HW0.0.1 cannot become multiple valued for positive ν. The solutions become very steep, and because we are finding numerical solutions, it is difficult to tell whether the solution should be vertical at $x = \frac{1}{2}$ or just very steep. As we saw in Chapter 7, both of the solution schemes used to get the solutions to HW0.0.1 and HW0.0.2 contained numerical dissipation (viscosity) anyway (sometimes more numerical dissipation than the "real dissipation" was included in the solutions to HW0.0.1). The idea is that the solutions to HW0.0.1 are the more physically relevant solutions (if Burgers' equation described a physical situation). We want solutions of HW0.0.2 that approximate solutions to HW0.0.1 for small values of ν. Hence, in general, *we want solutions to equation (9.2.1) that are limits of solutions to*

$$v_t + F(v)_x = \nu v_{xx}$$

(or $\nu(\beta(v)v_x)_x$, where $\beta(v) > 0$) as $\nu \to 0$. Instead of the breaking wave solution, we will obtain solutions with discontinuities at the breaking point. Clearly, *these solutions will not be classical solutions to equation (9.2.1) in that they will not be differentiable everywhere.*

HW 9.2.1 Determine the breaking point for the problem

$$v_t + \left(\frac{1}{2}v^2\right)_x = 0 \quad x \in \mathbb{R},\ t > 0$$

$$v(x,0) = \sin 2\pi x \quad x \in \mathbb{R}.$$

Hint: Find the characteristics emanating from the points $\frac{1}{2} \pm \delta$ (this choice is used for convenience), set the x values for these two curves equal, solve for t and take the limit as $\delta \to 0$.

9.2.2 *Weak Solutions*

The way that we include the discontinuous solutions to equation (9.2.1) that are so necessary for our discussion is to consider weak solutions to partial differential equation (9.2.1). The topic of weak solutions of partial differential equations is discussed in varying levels of rigor in most books on partial differential equations. Since we do not need any of the difficult results concerning weak solutions, we give only a brief description here. Specifically, we consider the initial–value problem consisting of conservation law (9.2.1) defined for $x \in \mathbb{R}$ and $t > 0$ along with the initial condition

$$v(x,0) = v_0(x), \quad x \in \mathbb{R}. \tag{9.2.9}$$

We define the set of **test functions**, C_0^1, to be the set

$$\{\phi \in C^1 : \{(x,t) \in \mathbb{R} \times [0,\infty) : \phi(x,t) \neq 0\} \subset [a,b] \times [0,T] \text{ for some } a,\ b \text{ and } T\}.$$

Hence, we know that ϕ is once continuously differentiable and zero outside of some rectangle in x-t space. We might mention that functions ϕ that satisfy

$$\{(x,t) \in \mathbb{R} \times [0,\infty) : \phi(x,t) \neq 0\} \subset [a,b] \times [0,T] \text{ for some } a,\ b \text{ and } T$$

are said to have **compact support in** $\mathbb{R} \times [0,\infty)$. The **support of** ϕ, written $\text{supp}(\phi)$, is the set on which $\phi \neq 0$. If we multiply partial differential equation (9.2.1) by $\phi \in C_0^1$ and integrate with respect to x from $-\infty$ to ∞ and with respect to t from 0 to ∞, we get

$$0 = \int_0^\infty \int_{-\infty}^\infty [v_t + F(v)_x]\,\phi(x,t)\,dx\,dt \tag{9.2.10}$$

$$= \int_0^T \int_a^b [v_t + F(v)_x]\,\phi(x,t)\,dx\,dt \quad \text{(because } \phi \in C_0^1\text{)}$$

$$= \int_a^b \int_0^T v_t\phi(x,t)\,dt\,dx + \int_0^T \int_a^b F(v)_x\phi(x,t)\,dx\,dt$$

$$= \int_a^b \left\{ \Big[v\phi\Big]_{t=0}^{t=T} - \int_0^T v\phi_t\,dt \right\} dx \quad \text{(integration by parts)}$$

$$+ \int_0^T \left\{ \Big[F(v)\phi\Big]_{x=a}^{x=b} - \int_a^b F(v)\phi_x\,dx \right\} dt \quad \text{(integration by parts)}$$

$$= -\int_a^b v(x,0)\phi(x,0)\,dx - \int_a^b \int_0^T v\phi_t\,dt\,dx - \int_0^T \int_a^b F(v)\phi_x\,dx\,dt \tag{9.2.11}$$

since $\phi(x,T) = \phi(a,t) = \phi(b,t) = 0$. We can rewrite (9.2.10)–(9.2.11) as

$$0 = \int_0^\infty \int_{-\infty}^\infty [v\phi_t + F(v)\phi_x]\,dx\,dt + \int_{-\infty}^\infty v_0\phi_0\,dx, \tag{9.2.12}$$

where $v_0 = v(x,0)$ is the initial condition and ϕ_0 is a notation for $\phi(x,0)$. We note that since the support of ϕ is contained in $[a,b] \times [0,T]$ and ϕ is defined on $\mathbb{R} \times [0,\infty)$, $\phi(x,0)$ need not be zero.

Hence, we have proved the following proposition.

Proposition 9.2.3 *If v is a classical solution to the initial-value problem (9.2.1), (9.2.9), then v will satisfy equation (9.2.12) for all $\phi \in C_0^1$.*

If we reverse the steps given above until we return to equation (9.2.10) and then use the fact that this must be satisfied for all $\phi \in C_0^1$ to show that v satisfies partial differential equation (9.2.1), we obtain the following result.

Proposition 9.2.4 *If v is continuously differentiable with respect to x and t and satisfies equation (9.2.12) for all $\phi \in C_0^1$, then v is a classical solution of initial-value problem (9.2.1), (9.2.9).*

It should also be clear that *there may be some solutions to equation (9.2.12) that are not classical solutions to initial-value problem (9.2.1), (9.2.9)* (functions that satisfy equation (9.2.12) that may not be differentiable). For this reason, we make the following definition.

Definition 9.2.5 *If v satisfies equation (9.2.12) for all $\phi \in C_0^1$, v is said to be a* **weak solution** *to initial-value problem (9.2.1), (9.2.9).*

Before we proceed, we include several examples of weak solutions.

Example 9.2.1 Show that

$$v(x,t) = \begin{cases} 1 & \text{if } x \le t/2 \\ 0 & \text{if } x > t/2 \end{cases} \tag{9.2.13}$$

is a weak solution to the inviscid Burgers' equation

$$v_t + \left(\frac{1}{2}v^2\right)_x = 0 \tag{9.2.14}$$

with initial condition

$$v_0(x) = \begin{cases} 1 & \text{if } x \le 0 \\ 0 & \text{if } x > 0. \end{cases} \tag{9.2.15}$$

Solution: Let $\phi \in C_0^1$ and let a, b and T be such that $\text{supp}(\phi) \subset [a,b] \times [0,T]$. Then

$$\begin{aligned}
&\int_0^\infty \int_{-\infty}^\infty (v\phi_t + \frac{v^2}{2}\phi_x)\,dx\,dt + \int_{-\infty}^\infty v_0(x)\phi(x,0)\,dx \\
&= \int_0^T \int_a^b (v\phi_t + \frac{v^2}{2}\phi_x)\,dx\,dt + \int_a^b v_0(x)\phi(x,0)\,dx \\
&= \int_0^T \int_a^{t/2} (\phi_t + \frac{1}{2}\phi_x)\,dx\,dt + \int_a^0 \phi(x,0)\,dx \\
&= \int_a^0 \int_0^T \phi_t\,dt\,dx + \int_0^{T/2} \int_{2x}^T \phi_t\,dt\,dx + \frac{1}{2}\int_0^T \int_a^{t/2} \phi_x\,dx\,dt + \int_a^0 \phi(x,0)\,dx \\
&= \int_a^0 [\phi(x,T) - \phi(x,0)]\,dx + \int_0^{T/2} [\phi(x,T) - \phi(x,2x)]\,dx
\end{aligned}$$

$$+\frac{1}{2}\int_0^T [\phi(t/2,t)-\phi(a,t)]\,dt+\int_a^0 \phi(x,0)\,dx \tag{9.2.16}$$

$$= -\frac{1}{2}\int_0^T \phi(y/2,y)dy+\frac{1}{2}\int_0^T \phi(t/2,t)\,dt \tag{9.2.17}$$

$$= 0.$$

(In going from step (9.2.16) to (9.2.17), we use the facts that $\phi(x,T)=\phi(a,t)=0$ to eliminate the first, third and sixth terms, cancel the second and seventh terms, and set $y=2x$ in the fourth term.) Hence, the function v given by (9.2.13) is a weak solution to initial-value problem (9.2.14)–(9.2.15).

Remark 1: Since we see in Figure 9.2.4 that the characteristic curves associated with the above problem intersect, it should not surprise us that we need to consider a weak solution. The example demonstrates that a "solution" that is obviously not a classical solution can still be a weak solution. The weak solution given above is associated with the characteristic curves given in Figure 9.2.5. The solution on the characteristics emanating from x, $x<0$ is different from that on the characteristics emanating from x, $x>0$. Hence, there is a discontinuity along the curve $x=t/2$.

Remark 2: Note that by the form of solution (9.2.13), the discontinuity in the solution propagates along the curve $x=t/2$. Hence, the speed of propagation of the discontinuity is $dx/dt=\frac{1}{2}$.

Remark 3: A discontinuity of a piecewise continuous weak solution is called a **shock** if the characteristics on both sides of the discontinuity impinge on the discontinuity curve in the direction of increasing t, as is the case in Figure 9.2.5. If we let $a_L=F'(v_L)$ and $a_R=F'(v_R)$ where v_L and v_R are the values of v on the left and right sides of the discontinuity, then *a discontinuity will be a shock if*

$$a_L > s > a_R \tag{9.2.18}$$

where s is the speed of propagation of the discontinuity. We see in the case of Example 9.2.1 that $a_L=1$, $a_R=0$ and $s=\frac{1}{2}$, so inequality (9.2.18) is satisfied.

Example 9.2.2 Show that

$$v(x,t)=\begin{cases}0 & \text{if } x\le t/2\\ 1 & \text{if } x>t/2\end{cases} \tag{9.2.19}$$

is a weak solution to the inviscid Burgers' equation (9.2.14) with initial condition

$$v_0(x)=\begin{cases}0 & \text{if } x\le 0\\ 1 & \text{if } x>0.\end{cases} \tag{9.2.20}$$

Solution: Let $\phi\in C_0^1$ and let a, b and T be such that $\text{supp}(\phi)\subset[a,b]\times[0,T]$. Then

$$\int_0^\infty\int_{-\infty}^\infty (v\phi_t+\frac{v^2}{2}\phi_x)\,dx\,dt+\int_{-\infty}^\infty v_0(x)\phi(x,0)\,dx$$

$$=\int_0^T\int_a^b (v\phi_t+\frac{v^2}{2}\phi_x)\,dx\,dt+\int_a^b v_0(x)\phi(x,0)\,dx$$

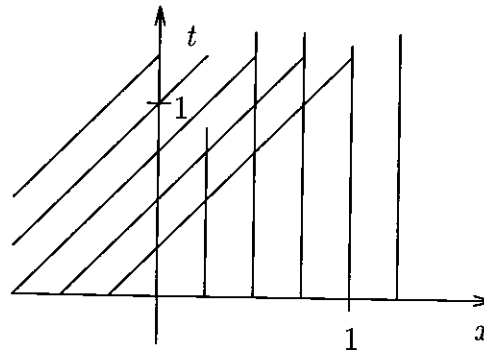

FIGURE 9.2.4. Characteristic curves associated with initial–value problem (9.2.14)–(9.2.15).

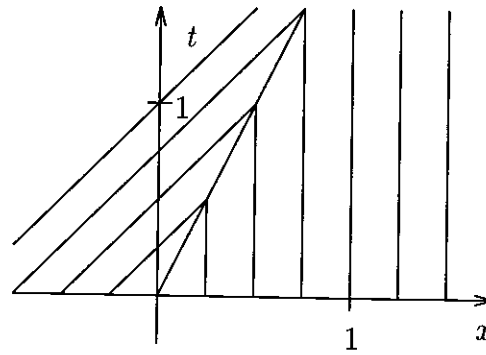

FIGURE 9.2.5. Characteristic curves associated with the solution given in Example 9.2.1 to initial–value problem (9.2.14)–(9.2.15).

$$= \int_0^T \int_{t/2}^b (\phi_t + \frac{1}{2}\phi_x)\,dx\,dt + \int_0^b \phi(x,0)\,dx$$

$$= \int_0^{T/2} \int_0^{2x} \phi_t\,dt\,dx + \int_{T/2}^b \int_0^T \phi_t\,dt\,dx + \frac{1}{2}\int_0^T \int_{t/2}^b \phi_x\,dx\,dt + \int_0^b \phi(x,0)\,dx$$

$$= \int_0^{T/2} [\phi(x,2x) - \phi(x,0)]\,dx + \int_{T/2}^b [\phi(x,T) - \phi(x,0)]\,dx$$

$$+\frac{1}{2}\int_0^T [\phi(b,t) - \phi(t/2,t)]\,dt + \int_0^b \phi(x,0)\,dx(\text{ since } \phi(x,T) = \phi(b,t) = 0, \text{ the}$$

second, fourth and seventh terms cancel and setting $y = t/2$ in the sixth term, we get)

$$= \int_0^{T/2} \phi(x,2x)\,dx - \frac{1}{2}\int_0^{T/2} \phi(y,2y)2dy$$

$$= 0.$$

Hence, v is a weak solution to initial–value problem (9.2.14), (9.2.20).

Remark: As in Example 9.2.1, we see that the speed of propagation of the discontinuity in solution (9.2.19) is $s = \frac{1}{2}$. We notice that since $a_L = 0$ and $a_R = 1$ for this example, inequality (9.2.18) is not satisfied, so the discontinuity in solution (9.2.19) is not a shock.

Example 9.2.3 Show that

$$v(x,t) = \begin{cases} 0 & \text{if } x < 0 \\ x/t & \text{if } 0 \le x \le t \\ 1 & \text{if } x > t \end{cases} \tag{9.2.21}$$

is a weak solution to the inviscid Burgers' equation with initial condition (9.2.20).

Solution: Let $\phi \in C_0^1$ and let a, b and T be such that $\text{supp}(\phi) \subset [a,b] \times [0,T]$. Then

$$\int_0^\infty t \int_{-\infty}^\infty (v\phi_t + \frac{v^2}{2}\phi_x)\,dx\,dt + \int_{-\infty}^\infty v_0(x)\phi(x,0)\,dx$$

$$= \int_0^T \int_a^b (v\phi_t + \frac{v^2}{2}\phi_x)\,dx\,dt + \int_a^b v_0(x)\phi(x,0)\,dx$$

$$= \int_0^T \int_0^t \left(\frac{x}{t}\phi_t + \frac{1}{2}\frac{x^2}{t^2}\phi_x\right)\,dx\,dt + \int_0^T \int_t^b (\phi_t + \frac{1}{2}\phi_x)\,dx\,dt + \int_0^b \phi(x,0)\,dx$$

$$= \int_0^T \int_x^T \frac{x}{t}\phi_t\,dt\,dx + \int_0^T \int_0^t \frac{x^2}{2t^2}\phi_x\,dx\,dt + \int_0^T \int_0^x \phi_t\,dt\,dx + \int_T^b \int_0^T \phi_t\,dt\,dx$$

$$+ \frac{1}{2}\int_0^T \int_t^b \phi_x\,dx\,dt + \int_0^b \phi(x,0)\,dx$$

$$= \int_0^T \left\{ \left[\frac{x}{t}\phi(x,t)\right]_{t=x}^{t=T} - \int_x^T \left(-\frac{x}{t^2}\right)\phi\,dt \right\} dx \quad \text{(integration by parts)}$$

$$+ \int_0^T \left\{ \left[\frac{x^2}{2t^2}\phi(x,t)\right]_{x=0}^{x=t} - \int_0^t \frac{x}{t^2}\phi\,dx \right\} dt \quad \text{(integration by parts)}$$

$$+ \int_0^T [\phi(x,x) - \phi(x,0)]\,dx + \int_T^b [\phi(x,T) - \phi(x,0)]\,dx \quad (\phi(x,T) = 0)$$

$$+ \frac{1}{2}\int_0^T [\phi(b,t) - \phi(t,t)]\,dt + \int_0^b \phi(x,0)\,dx \quad (\phi(b,t) = 0)$$

$$= \int_0^T \left[\frac{x}{T}\phi(x,T) - \phi(x,x)\right]\,dx + \int_0^T \int_x^T \frac{x}{t^2}\phi\,dt\,dx \quad (\phi(x,T) = 0)$$

$$+\int_0^T \frac{1}{2}\phi(t,t)\,dt - \int_0^T \int_0^t \frac{x}{t^2}\phi\,dx\,dt + \int_0^T \phi(x,x)\,dx - \int_0^b \phi(x,0)\,dx$$
$$-\frac{1}{2}\int_0^T \phi(t,t)\,dt + \int_0^b \phi(x,0)\,dx \qquad \text{(since the 2nd and 6th, 3rd and 5th, 4th and 8th and 7th and 9th terms cancel)}$$
$$= 0.$$

Remark 1: From Examples 9.2.2 and 9.2.3 we see that weak solutions to initial–value problems are not unique.

Remark 2: The characteristics associated with the initial–value problem given by Burgers' equation and initial condition (9.2.20) are given in Figure 9.2.6. We see that because the slope of the characteristic curves for $x < 0$ is greater that the slope of the characteristic curves for $x > 0$, there is a region that has no characteristics. The solution in Example 9.2.2 corresponds to filling in this region that has no characteristic curves with characteristics that come out of the curve $t = 2x$, as shown in Figure 9.2.7. We note that since the characteristics on either side of the curve $x = t/2$ emanate from the discontinuity rather than impinge on the discontinuity, *the discontinuity in solution (9.2.19) is not a shock.*

The solution given in Example 9.2.3 corresponds to filling in the region that has no characteristic curves with a **fan** of characteristics as is shown in Figure 9.2.8. We note that we were able to "fill in" the missing characteristics in at least two different ways that are compatible with the weak formulation of the problem. Later, we will have to determine which of these solutions we want. As we shall see, solutions found by filling in a region with a fan are the desired solutions.

Remark 3: We note that the solution given in Example 9.2.3 is continuous. The solution is continuously differentiable and satisfies Burgers' equation for all $t > 0$, except when $x = 0$ and $x = t$. Its behavior at $(x,t) = (0,0)$, $x = 0$ and $x = t$ does not allow v to be a classical solution.

Remark 4: We should note that the solutions to the problems solved in Examples 9.2.1, 9.2.2 and 9.2.3 can all be written in the form $\psi(x/t)$. For example, solution (9.2.13) can be expressed as $v(x,t) = \psi_1(x/t)$ where

$$\psi_1(\xi) = \begin{cases} 1 & \text{if } \xi \le \frac{1}{2} \\ 0 & \text{if } \xi > \frac{1}{2} \end{cases}$$

and solution (9.2.21) can be written as $v(x,t) = \psi_2(x/t)$ where

$$\psi_2(\xi) = \begin{cases} 1 & \text{if } \xi < 0 \\ \xi & \text{if } 0 \le \xi \le 1 \\ 0 & \text{if } \xi > 1 \end{cases}$$

(and solution (9.2.19) can be written in a form very similar to ψ_1, see HW9.2.2).

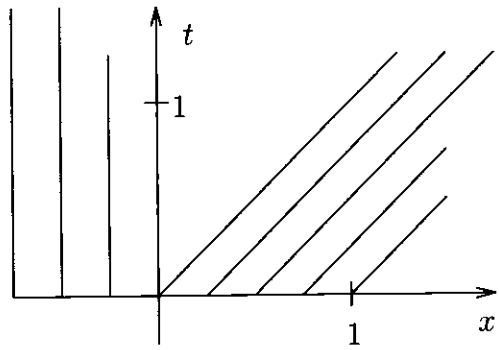

FIGURE 9.2.6. Characteristic curves associated with initial–value problem (9.2.14), (9.2.20).

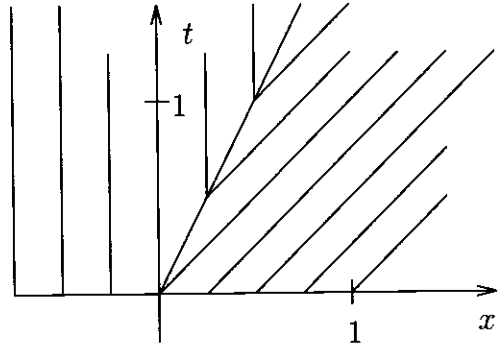

FIGURE 9.2.7. Characteristic curves associated with the solution to initial–value problem (9.2.14), (9.2.20) given in Example 9.2.2.

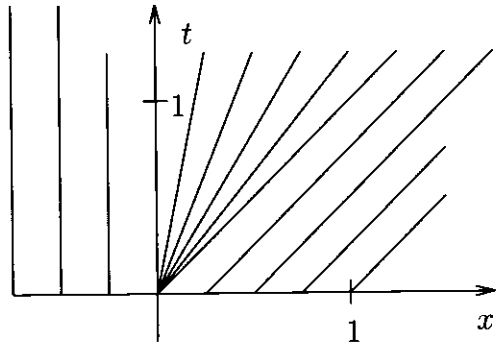

FIGURE 9.2.8. Characteristic curves associated with the solution to initial–value problem (9.2.14), (9.2.20) given in Example 9.2.3.

Remark 5: And finally, we see in HW9.2.3 that for a problem very similar to that solved in Examples 9.2.2 and 9.2.3, we can have an even more complex solution. In Remark 1 we emphasized that weak solutions to initial-value problems need not be unique. Generally, there will be an infinite number of solutions made different only be redefining one or more points that really do not change the character of the solution. However, we should be aware that there are also many non-trivial different solutions to a given initial-value problem. An initial-value problem such as (9.2.14), (9.2.20) will have many interesting solutions that we will not want and that we will have to eliminate from consideration.

HW 9.2.2 Find a function $\psi_3 = \psi_3(\xi)$ such that solution (9.2.19) can be written as $v(x,t) = \psi_3(x/t)$.

HW 9.2.3 Show that

$$v(x,t) = \begin{cases} -1 & \text{if } x < -(1-\alpha)t/2 \\ \alpha & \text{if } -(1-\alpha)t/2 \le x < 0 \\ -\alpha & \text{if } 0 \le x \le (1-\alpha)t/2 \\ 1 & \text{if } x > (1-\alpha)t/2 \end{cases} \tag{9.2.22}$$

$\alpha < 1$, is a weak solution to the inviscid Burgers' equation with initial condition

$$v_0(x) = \begin{cases} -1 & \text{if } x \le 0 \\ 1 & \text{if } x > 0. \end{cases} \tag{9.2.23}$$

HW 9.2.4 Find the fan solutions to the problem consisting of the inviscid Burgers' equation with initial condition

$$v_0(x) = \begin{cases} v_L & x < 0 \\ v_R & x \ge 0 \end{cases}$$

when (a) $0 < v_L < v_R$, (b) $v_L \le 0 \le v_R$ and (c) $v_L < v_R < 0$.

9.2.3 Discontinuous Solutions

We admit that the definition of weak solutions is much more abstract than we have led you to believe. The integration used in equation (9.2.12) is Lebesgue integration, and there could be some very bizarre functions v that satisfy equation (9.2.12). However, let us assure the reader that we are not interested in bizarre solutions. *We are interested in solutions $v = v(x,t)$ that are smooth except across one or more curves in x-t space*—where they have jump discontinuities. When we restrict our interest to solutions of this kind, we severely restrict the types of discontinuities that we allow in our solutions (and we can also use the Riemann integral instead of the Lebesgue integral). We then have the following very important result.

Proposition 9.2.6 *Let C be a smooth curve in x-t space ($\mathbb{R} \times \mathbb{R}^+$), $x_C = x_C(t)$, across which v, a weak solution to initial–value problem (9.2.1), (9.2.9), has a jump discontinuity. Let $P = (x_0, t_0)$, $t_0 > 0$, be any point on C, $s = \frac{dx_C}{dt}(t_0)$, and v_L and v_R be the values of v evaluated to the left and the right of P, respectively. Then*

$$(v_L - v_R)\frac{dx_C}{dt} = F(v_L) - F(v_R). \tag{9.2.24}$$

Proof: Let B be a small ball centered at P that does not contain the point $(x_C(0), 0)$. Let B_1 and B_2 be the parts of B on either side of the curve C. See Figure 9.2.9.

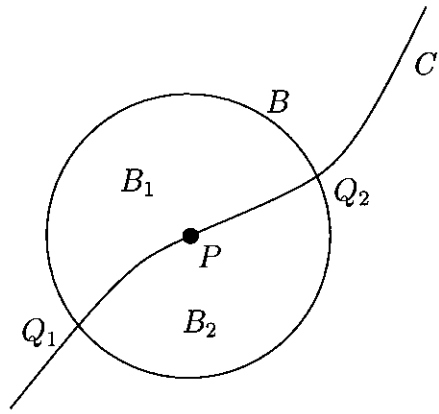

FIGURE 9.2.9.

Since v is a weak solution to problem (9.2.1), (9.2.9), v must satisfy equation (9.2.12) for all $\phi \in C_0^1$. For a function ϕ with support in B, we have

$$0 = \int_0^\infty \int_{-\infty}^\infty (v\phi_t + F(v)\phi_x)\,dx\,dt \tag{9.2.25}$$

$$= \iint_{B_1} (v\phi_t + F(v)\phi_x)\,dx\,dt + \iint_{B_2} (v\phi_t + F(v)\phi_x)\,dx\,dt, \tag{9.2.26}$$

where the $t = 0$ term of equation (9.2.12) is not included because $\phi_0 = 0$ in B. The fact that v is smooth in B_1 and B_2 and v is a weak solution on B_1 and B_2 implies that v is a classical solution in the interior of both B_1 and B_2 (i.e., v satisfies $v_t + F(v)_x = 0$ in the interior of B_1 and B_2). Thus equation (9.2.26) can be rewritten as

$$0 = \iint_{B_1} [(v\phi)_t + (F(v)\phi)_x]\,dx\,dt + \iint_{B_2} [(v\phi)_t + (F(v)\phi)_x]\,dx\,dt. \tag{9.2.27}$$

Applying Green's Theorem, page 348, ref. [38], we write equation (9.2.27) as

$$0 = \int_{\partial B_1} \phi\big[-v\,dx + F(v)\,dt\big] + \int_{\partial B_2} \phi\big[-v\,dx + F(v)\,dt\big]. \quad (9.2.28)$$

If we define

$$v_L(t) = \lim_{\substack{(x,t)\to C \\ (x,t)\in B_1}} v(x,t)$$
$$v_R(t) = \lim_{\substack{(x,t)\to C \\ (x,t)\in B_2}} v(x,t)$$

(remember that $x = x_C(t)$ on C) and use the fact that $\phi = 0$ on ∂B (the outside part of B_1 and B_2), we can rewrite equation (9.2.28) as

$$0 = \int_{Q_1}^{Q_2} \phi\,[-v_L\,dx + F(v_L)\,dt] - \int_{Q_1}^{Q_2} \phi\,[-v_R\,dx + F(v_R)\,dt]\,, \quad (9.2.29)$$

where the minus sign in front of the second integral is due to the fact that the line integral around B_2 in (9.2.28) is going in the opposite direction from that in the first integral. Since $dx = x_C'(t)dt$, equation (9.2.29) can be written as

$$0 = \int_{Q_1}^{Q_2} \phi\left[-(v_L - v_R)\frac{dx_C}{dt} + (F(v_L) - F(v_R))\right] dt.$$

Since ϕ is arbitrary (in C_0^1), we get

$$(v_L - v_R)\frac{dx_C}{dt} = F(v_L) - F(v_R)$$

at each point on C.

We call $s = \frac{dx_C}{dt}$ the **speed of propagation of the discontinuity**. If we define the notation $[\cdot]$ as the jump across C (in general, $[f] = f_L - f_R$), we can write equation (9.2.24) as

$$s\,[v] = [F(v)]\,. \quad (9.2.30)$$

Equation (9.2.24) or (9.2.30) is referred to as the **jump condition** or the **Rankine-Hugoniot condition**. We note that the jump condition (R-H condition) is a condition that any weak solution to an initial–value problem such as (9.2.1), (9.2.9) must satisfy across any jump discontinuity. This condition does not choose for us which weak solution we want. *Physically unacceptable weak solutions will also satisfy the jump condition.*

We next include an example that illustrates the use of the jump condition for Burgers' equation to provide a useful general result concerning the speed of propagation of a discontinuity.

Example 9.2.4 Consider the inviscid Burgers' equation (9.2.14) along with the initial condition

$$v_0(x) = v(x,0) = \begin{cases} v_L & \text{if } x < 0 \\ v_R & \text{if } x \geq 0 \end{cases} \tag{9.2.31}$$

where v_L and v_R are constants. Use the jump condition to determine the speed of propagation of this discontinuity.

Solution: Since for Burgers' equation $F(v) = \frac{1}{2}v^2$, the R-H condition gives

$$s(v_L - v_R) = \frac{1}{2}(v_L^2 - v_R^2)$$

across the jump. Hence $s = (v_L + v_R)/2$, so that the speed of propagation of the discontinuity is the average of the solution values on the left and right.

Remark 1: We see now that since the jump that we computed in HW0.0.2 was skew symmetric with respect to $x = \frac{1}{2}$ and $y = 0$, it follows that $v_R = -v_L$ and $s = 0$ (which is what we saw computationally).

Remark 2: From the results found in Proposition 9.2.6 and Example 9.2.4, we make the following observations.

- The solution given in Example 9.2.1 has a discontinuity with speed of propagation of the discontinuity given by $s = \frac{1}{2}$ and this discontinuity will propagate along the curve $x = t/2$, which is the solution to the differential equation

$$\frac{dx_C}{dt} = s = \frac{1}{2}, \quad x_C(0) = 0.$$

  We note that s can be calculated by the formula $s = (v_L + v_R)/2 = \frac{1}{2}$.

- The solution given in Example 9.2.1 will satisfy the jump condition across the curve $x = t/2$:

$$s(v_L - v_R) = \frac{1}{2} = F(v_L) - F(v_R) = \frac{1}{2}1^2 - \frac{1}{2}0^2 = \frac{1}{2}$$

  (which is the same as $s = (v_L + v_R)/2$).

- The solution given in Example 9.2.2 has a discontinuity with speed of propagation of the discontinuity given by $s = (v_L + v_R)/2 = \frac{1}{2}$. This discontinuity will propagate along the curve that is the solution to the differential equation

$$\frac{dx_C}{dt} = s = \frac{1}{2}, \quad x_C(0) = 0,$$

  or $x = t/2$.

- Since the solution found in Example 9.2.3 is continuous (it has no jump discontinuities), it is not relevant to consider the jump condition with respect to this solution.

- The basic result given in this section, along with the results of Example 9.2.4, show us why the solutions found in Examples 9.2.1 and 9.2.2 depend so strongly on the curve $x = t/2$. It is the solution to the differential equation $dx_C/dt = s$ with initial condition $x_C(0) = 0$ along which the discontinuity in the initial condition is supposed to propagate.

We next include an example where the jump condition is instrumental in helping us find a solution to the initial–value problem.

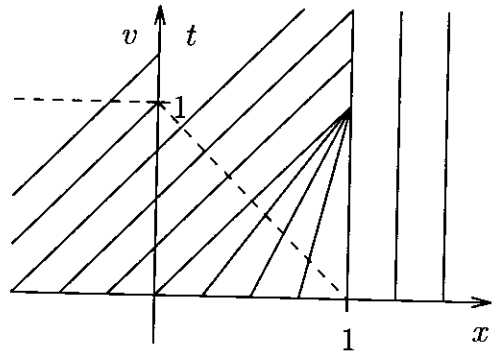

FIGURE 9.2.10. Characteristics associated with the initial–value problem defined by Burgers' equation (9.2.14) along with initial condition (9.2.32) (where we do not allow characteristics to cross).

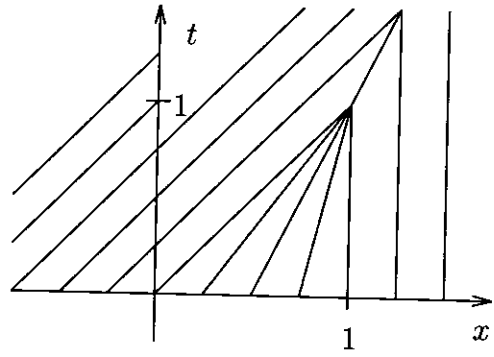

FIGURE 9.2.11. Characteristics associated with solution (9.2.33)–(9.2.35) to the initial–value problem defined by Burgers' equation (9.2.14) along with initial condition (9.2.32).

Example 9.2.5 Discuss the solution to the initial–value problem of Burgers' equation (9.2.14) along with initial condition

$$v_0(x) = \begin{cases} 1 & \text{if } x < 0 \\ 1 - x & \text{if } 0 \le x \le 1 \\ 0 & \text{if } x > 1. \end{cases} \tag{9.2.32}$$

Solution: In Figure 9.2.10 we plot the characteristics associated with this problem, where we draw them in such a way as not to allow them to cross. We find the characteristics emanating from the segment $0 \le x \le 1$ by noting that a characteristic must satisfy

$$t = \frac{1}{F'(v_0)}x + C = \frac{1}{1 - x_0}x + C$$

and determining $C = x_0/(1 - x_0)$ such that $t = 0$ when $x = x_0$. By solving any of the equations

$$t = \frac{1}{1 - x_0}x - \frac{x_0}{1 - x_0}, \quad 0 \le x_0 < 1$$

along with the equation $x = 1$, it is easy to see that the characteristics first intersect at $(x, t) = (1, 1)$, i.e., the breaking point is $T_b = 1$. For $t < 1$, the solution is continuous, determined by its characteristic curves and initial conditions and can be written as

$$v(x,t) = \begin{cases} 1 & \text{if } x < t < 1 \\ \frac{1-x}{1-t} & \text{if } t \le x \le 1 \\ 0 & \text{if } x > 1. \end{cases} \tag{9.2.33}$$

We note that the solution given for $t \le x \le 1$ is found by setting $v(x,t) = v_0(x_0) = 1 - x_0$ along the characteristic

$$t = \frac{1}{1 - x_0}x - \frac{x_0}{1 - x_0} \tag{9.2.34}$$

(using the fact that v is constant along a characteristic curve). We then eliminate x_0 from v by solving the characteristic curve equation (9.2.34) for x_0.

For $t \ge 1$, we have an "initial condition" (occurring at $t = 1$) given by

$$v_0(x) = v(x, 1) = \begin{cases} 1 & \text{if } x < 1 \\ 0 & \text{if } x \ge 1. \end{cases}$$

Hence, using the results found in Example 9.2.4, we know that there will be a solution where this discontinuity will propagate along the characteristic curve defined by

$$\frac{dx_c}{dt} = s = \frac{1}{2}, \quad x_C(1) = 1,$$

or $x_C = (t + 1)/2$. Hence, a solution for $t \ge 1$ is given by

$$v(x,t) = \begin{cases} 1 & \text{if } x < (t+1)/2 \\ 0 & \text{if } x \ge (t+1)/2 \end{cases} \tag{9.2.35}$$

and the characteristic curves associated with this solution are given in Figure 9.2.11.

We showed earlier that weak solutions to initial-value problems are not unique. Not only are they not unique, but we claimed that there are many interesting looking solutions to initial–value problems. As we shall see in the next section, we will be interested in weak solutions like those given in Examples 9.2.1 and 9.2.3 because we like these solutions, and the solution given in Example 9.2.2 because this is a logical solution that we do

not like. What we wish to emphasize here is that there are many solutions to initial–value problems that might look interesting to the untrained eye. As we shall see later, we will sometimes obtain these unwanted solutions numerically. When we obtain these solutions (which are sometimes much more interesting looking than the correct solution) as a part of a complex problem, it may be difficult to see that we should reject the solution and proceed to determine why we obtained the undesirable solution. The bad part is that if we like the (wrong) solution, it is probably possible to rationalize the incorrect behavior and explain (incorrectly) why we think the solution obtained is the correct solution.

We can build these solutions fairly easily based on Proposition 9.2.6. The technique follows from the fact that *if a proposed solution v satisfies the initial condition and if it satisfies the jump condition across all discontinuities, then v will be a weak solution to the initial–value problem.* We use the fact that the result given in Proposition 9.2.6 actually characterizes the weak solutions to initial–value problems for conservation laws. In the next example we will introduce a technique for building a solution to an initial–value problem and discuss what we mean by "v satisfies the initial condition."

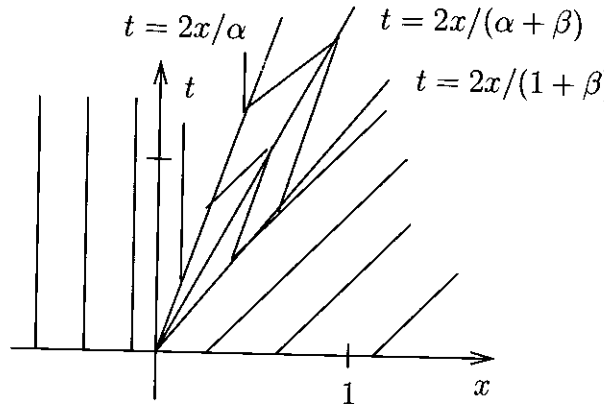

FIGURE 9.2.12. Characteristics associated with solution (9.2.36) (with $\alpha = \frac{3}{4}$ and $\beta = \frac{1}{2}$) to the initial–value problem defined by Burgers' equation (9.2.37) along with initial condition (9.2.20).

Example 9.2.6 Show that the function v defined by

$$v(x,t) = \begin{cases} 0 & x < \alpha t/2 \\ \alpha & \alpha t/2 \le x < (\alpha+\beta)t/2 \\ \beta & (\alpha+\beta)t/2 \le x < (1+\beta)t/2 \\ 1 & (1+\beta)t/2 \le x \end{cases} \tag{9.2.36}$$

for any α, β, $0 < \alpha < 1$, $\beta > 0$, is a weak solution to the initial–value problem consisting

of Burgers' equation

$$v_t + \left(\frac{v^2}{2}\right)_x = 0, \quad t > 0, \ x \in \mathbb{R} \tag{9.2.37}$$

along with initial condition (9.2.20).

Solution: The most obvious way to show that v is a weak solution to initial–value problem (9.2.37), (9.2.20) is to show that v will satisfy equation (9.2.12). This calculation would be gruesome.

If we consider any $x \neq 0$ and let $t \to 0+$ (t approaches 0 from the positive side), it is easy to see that v satisfies initial condition (9.2.20) pointwise. If we consider $(x,t) \to (0,0)$ along any of the different rays along which v is constant, we see that the limit is not defined (we can get either 0, α, β or 1). However, because equation (9.2.12) involves integration, this limit is not relevant, and equation (9.2.12) does not care how v_0 is defined at one point such as $x = 0$. (The Riemann integral will allow you to have at least any finite number of "bad points," and the Lebesgue integral will allow very general sets of "bad points.") Generally, how v_0 or v are defined at the discontinuities is not important. We equate all functions that are equal "almost everywhere." As we saw in the proof of Proposition 9.2.6, the limiting values on each side of the curve, v_L and v_R, are what is important. Hence, since v satisfies initial condition (9.2.20) pointwise for $x \neq 0$, *v will satisfy initial condition (9.2.20) weakly.*

The final step to show that v is a weak solution to initial–value problem (9.2.37), (9.2.20) is to show that v satisfies the jump condition across all discontinuities. This is easy to see if we realize that the discontinuities of v propagate along the three curves $C_1 : x = \alpha t/2$, $C_2 : x = (\alpha+\beta)t/2$ and $C_3 : x = (1+\beta)t/2$. Hence the speed of propagation of the discontinuities along these three curves is given by $s_1 = \alpha/2$, $s_2 = (\alpha+\beta)/2$ and $s_3 = (1+\beta)/2$. If we then evaluate jump condition (9.2.24) across these three curves (or consider the result of Example 9.2.4), we see that the jump condition is satisfied across each of these curves.

The fact that the jump condition is satisfied across each of these curves is obvious to us since we used this condition to define v. In other words, when we want a weak solution to initial–value problem (9.2.37), (9.2.20), we decide approximately where we want to place the curves along which the discontinuities will propagate, set the values we want for the function in each of the wedges defined by the curves (remembering that we must have v equal to 0 and 1 in the two outer wedges so that v will satisfy the initial condition), and then use $s = (v_L + v_R)/2$ (or more generally, the jump condition $s = [F(v_L) - F(v_R)]/(v_L - v_R)$) and $\frac{dx_C}{dt} = s$ to define the curves along which the discontinuities will propagate (placing conditions on α and β so that the curves will stay in the same order that we chose earlier).

Remark: In Figure 9.2.12 we draw the characteristics associated with solution (9.2.36). The characteristics for $x < \alpha t/2$ and for $x > (1+\beta)t/2$ are determined as we have done before from the initial conditions. The characteristics in the other two regions are determined using the fact that the characteristics must still be of the form

$$t = \frac{1}{F'(v_0)}t + C = \frac{1}{v_0}t + C.$$

In the wedge $\alpha t/2 \leq x < (\alpha+\beta)t/2$, the solution is equal to α everywhere, so the characteristics must be of the form $t = \frac{1}{\alpha}x + C$, and in the wedge $(\alpha+\beta)t/2 \leq x < (1+\beta)t/2$, the solution is equal to β everywhere, so the characteristics must be of the form $t = \frac{1}{\beta}x + C$.

We note that since the characteristics are emanating from the curves $t = 2x/\alpha$ and $t = 2x/(1+\beta)$ in the increasing direction of t, the discontinuities across these curves are not shocks. However, since the characteristics are impinging on the curve $t = 2x/(\alpha+\beta)$ in the increasing direction of t, the discontinuity across the curve $t = 2x/(\alpha+\beta)$ is a shock.

We can now, it is to be hoped, use the technique introduced in the above example to define many weak solutions to initial–value problems. For example, if we choose α to be negative and require that β satisfy $0 < \beta < -\alpha$, then the function v given by (9.2.36) will again be a solution where the curves C_1 and C_2 will both be in the second quadrant (and C_3 will again be in the first quadrant).

Also, it is also easy to see that if we require that $0 < \beta$, the function v given by

$$v(x,t) = \begin{cases} 0 & x < 0 \\ x/t & 0 \le x < \beta t \\ \beta & \beta t \le x < (1+\beta)t/2 \\ 1 & (1+\beta)t/2 \le x \end{cases} \tag{9.2.38}$$

will be a weak solution to initial-value problem (9.2.37), (9.2.20).

We can also use Proposition 9.2.6 to build many weak solutions to the initial–value problem consisting of Burgers' equation along with initial condition (9.2.15). For example, it is easy to see that the function v defined by

$$v(x,t) = \begin{cases} 1 & x < -t \\ -3 & -t \le x < 0 \\ 3 & 0 \le x < 3t/2 \\ 0 & 3t/2 \le x \end{cases} \tag{9.2.39}$$

will satisfy both initial condition (9.2.15) (weakly) and the jump conditions across each of the curves $x = -t$, $x = 0$ and $x = 3t/2$.

And finally, we include one last example where we have the continuous initial condition $v_0(x) = 1$ for $x \in \mathbb{R}$. We note that the function v defined by

$$v(x,t) = \begin{cases} 1 & x < -t/2 \\ -2 & -t/2 \le x < 0 \\ 2 & 0 \le x < 3t/2 \\ 1 & 3t/2 \le x \end{cases} \tag{9.2.40}$$

will be a weak solution to the initial–value problem given by Burgers' equation along with the initial condition $v_0(x) = 0$.

HW 9.2.5 (a) Show that the function v defined by

$$v(x,t) = \begin{cases} 0 & x < \alpha t/2 \\ \alpha & \alpha t/2 \le x < 0 \\ -\alpha & 0 \le x < (1-\alpha)t/2 \\ 1 & (1-\alpha)t/2 \le x \end{cases} \tag{9.2.41}$$

where $\alpha < 0$, is a solution to initial–value problem (9.2.37), (9.2.20).
(b) Determine whether or not the discontinuities in the above solution across the curves $x = \alpha t/2$ and $x = (1-\alpha)t/2$ are shocks.

HW 9.2.6 (a) Determine whether the discontinuity in solution (9.2.38) across the curve $x = (1+\beta)t/2$ is a shock.
(b) Determine which of the discontinuities in solutions (9.2.39) and (9.2.40) are shocks and which are not.

9.2.4 The Entropy Condition

As we have seen in the last two sections, weak solutions to conservation laws can contain discontinuities that are due to a discontinuity in the initial condition or to characteristics that cross each other, or may occur reasonably randomly as long as the jump conditions are satisfied across the discontinuities. In addition, we saw that the weak solutions to conservation laws need not be unique. Eventually, we will be trying to compute the solutions (including the discontinuities) to conservation laws, but before we try this, we must find some approach that will help decide which solution we want (in the case of nonunique solutions). We shall not really try to explain how and why the various approaches select the "correct" weak solution. We will try to summarize the results that describe how to choose the correct solution and leave it to the reader to consult the references cited if a more in-depth explanation is desired.

One way of choosing the correct solution is to choose the solutions discussed in Section 9.2.1 that are limits of an associated viscous problem as the viscosity vanishes (which we shall call **vanishing viscosity solutions**). There are various reasons to "want" this solution as our solution. One of the most physically appealing reasons is that many of the equations that we are solving approximate a physical situation that includes some sort of dissipation (a sufficiently small amount, which the modeler assumed was negligible). Hence, the solution that we choose will approximate a solution with a small amount of dissipation. One of the very important attributes of the vanishing viscosity solution is the following result.

Proposition 9.2.7 *If a vanishing viscosity solution exists, it is a weak solution.*

To see that this result might be true, we consider the viscous equation

$$v_t^\epsilon + F(v^\epsilon)_x = \epsilon v_{xx}^\epsilon.$$

If we multiply the above equation by a C_0^2 test function (where ϕ and ϕ_x will be zero outside some closed rectangle $[a,b] \times [0,T]$) and perform the integrations done in (9.2.10)–(9.2.11) (plus two analogous integration by parts operations on the viscous term), we get

$$-\int_0^\infty \int_{-\infty}^\infty [v^\epsilon \phi_t + F(v^\epsilon)\phi_x]\, dx\, dt - \int_{-\infty}^\infty v_0^\epsilon \phi_0\, dx = \epsilon \int_0^\infty \int_{-\infty}^\infty v^\epsilon \phi_{xx}\, dx\, dt.$$

If we formally let $\epsilon \to 0$, then $v^\epsilon \to v$, $F(v^\epsilon) \to F(v)$, and (after we resolve the technical question of the differences between C_0^1 and C_0^2 test functions)

we see that the vanishing viscosity solution is a weak solution to equation (9.2.1).

Another approach is to mimic how the correct solution can be chosen for a physically relevant model problem. For stable physical situations, we know that we must have a unique physical solution. Generally, if our mathematical model does not have a unique solution, some more physics must be included that will close the system and select the correct physical solution. For example, in ref. [57] on page 373 is given a discussion of choosing the "right" solution to the Euler equations based on an entropy argument. Specifically, the author shows that a solution analogous to the one discussed in Example 9.2.2 is unacceptable because entropy decreases across the discontinuity. In addition, he argues that a solution such as (9.2.19) is unstable and will break apart to form a fan solution like that found in Example 9.2.3. He then shows that the solution analogous to (9.2.21) is a solution to the problem for which the entropy does not decrease.

Before we proceed to decide how to choose the correct solutions, we introduce two common entropy conditions. It is not clear whether and/or how the first condition mimics the physical entropy argument. However, the second condition is truly taking an approach that finds a new variable that is to act like the "entropy" for the given system and the condition is designed to imitate the entropy condition of gas dynamics. The first entropy condition that we give was introduced earlier in Remark 3, page 83.

Definition 9.2.8 Entropy Condition I: *The solution to equation (9.2.12), $v = v(x,t)$, containing a discontinuity propagating with speed s is said to satisfy Entropy Condition I if*

$$F'(v_L) > s > F'(v_R) \tag{9.2.42}$$

where v_L and v_R are the solution values to the left and right of the discontinuity, respectively.

Remark 1: It is easy to see that since $F'(v_L) = 1$, $s = \frac{1}{2}$ and $F'(v_R) = 0$, solution (9.2.13) satisfies Entropy Condition I. Likewise, it is easy to see that solution (9.2.19) does not satisfy Entropy Condition I. The solution (9.2.21) satisfies Entropy Condition I vacuously, since there are no discontinuities in the solution. And finally, we note that if we were to choose $\alpha = \frac{3}{4}$ and $\beta = \frac{1}{2}$ in solution (9.2.36), the second discontinuity would satisfy Entropy Condition I, while the first and third discontinuities do not satisfy Entropy Condition I. This can also be seen in Figure 9.2.12, where we see that the characteristics impinge on the second discontinuity and emanate from the first and third discontinuities. Hence, the solution does not satisfy Entropy Condition I.

Remark 2: For Burgers' equation, any jump from v_L to v_R with $v_L > v_R$ that satisfies the jump condition (R-H condition) will satisfy Entropy

Condition I. Also, any jump from v_L to v_R with $v_L < v_R$ that satisfies the jump condition will not satisfy Entropy Condition I. In general, for any convex flux function F ($F'' > 0$, which implies that F' is increasing), any jump from v_L to v_R with $v_L > v_R$ that satisfies the jump condition will satisfy Entropy Condition I, and any jump with $v_L < v_R$ that satisfies the jump condition will not satisfy Entropy Condition I.

Remark 3: Recall that the problem with conservation laws is not the existence of solutions, but rather the uniqueness of solutions. We include the following result, which illustrates how Entropy Condition I implies uniqueness. Based on the relationship between Entropy Condition I and $\mathrm{I_a}$ given on page 104, this result is equivalent to Theorem 16.11, page 283, ref. [62].

Proposition 9.2.9 *Suppose that F is convex and that the solution v to the initial–value problem*

$$v_t + F(v)_x = 0, \quad x \in \mathbb{R}, \ t > 0 \tag{9.2.43}$$

$$v(x,0) = v_0(x), \quad x \in \mathbb{R} \tag{9.2.44}$$

satisfies Entropy Condition I across all jumps. Then the solution v is the unique solution to initial-value problem (9.2.43)–(9.2.44) that satisfies Entropy Condition I and is a vanishing viscosity solution to initial–value problem (9.2.43)–(9.2.44).

We emphasize that the uniqueness that we use above is the same as that used when we discussed how we satisfy initial conditions weakly. As is usually the case, we cannot really care what happens at the discontinuity in the solution.

Remark 4: There are examples of conservation laws that people care about for which the function F is not convex. One such example is the Buckley-Leverett equation, where $F(v) = v^2/\left[v^2 + (1-v)^2/4\right]$ in the area of porous media flow. The nonconvex analogue to Entropy Condition I is as follows.

Definition 9.2.10 Entropy Condition Inc: *The solution to equation (9.2.12) (where F is not necessarily convex), $v = v(x,t)$, containing a discontinuity is said to satisfy Entropy Condition Inc if*

$$\frac{F(v_L) - F(v)}{v_L - v} \geq \frac{F(v_R) - F(v_L)}{v_R - v_L} \tag{9.2.45}$$

for all v between v_L and v_R, where v_L and v_R are the solution values to the left and right of the discontinuity, respectively.

Again, it is not difficult to see that solution (9.2.13) satisfies entropy condition (9.2.45) and solution (9.2.19) does not satisfy entropy condition (9.2.45). As in the case where F is convex, if F is not convex, the solution v is unique and is a vanishing viscosity solution if v satisfies entropy condition (9.2.45) across all jumps.

The second entropy condition is more complex to describe. The approach is to mimic the way that the physical entropy functions enter into and interact with the conservation laws and solutions for physical systems. We let the scalar valued functions $S = S(v)$ and $\Phi = \Phi(v)$ be the **entropy function** and be the **entropy flux function**, respectively. We want to find functions S and Φ that satisfy the conservation law

$$S(v)_t + \Phi(v)_x = 0 \tag{9.2.46}$$

for smooth solutions v to conservation law (9.2.1). *We assume that S satisfies $S'' \geq 0$.* The reason for this assumption will become clear later and is consistent with the physical notion of an entropy function. Hence, we are looking for an additional conservation law that must be satisfied—the conservation of entropy.

In nonconservation form, equation (9.2.46) can be written as

$$S'(v)v_t + \Phi'(v)v_x = 0. \tag{9.2.47}$$

Smooth solutions of conservation law (9.2.1) satisfy

$$v_t + F'(v)v_x = 0. \tag{9.2.48}$$

Multiplying equation (9.2.48) by $S'(v)$, we get

$$S'(v)v_t + S'(v)F'(v)v_x = 0. \tag{9.2.49}$$

Comparing equations (9.2.47) and (9.2.49), we see that S and Φ must satisfy

$$\Phi'(v) = S'(v)F'(v). \tag{9.2.50}$$

For scalar conservation laws, equation (9.2.50) has many solutions.

Remark 1: If equation (9.2.1) is such that $F(v) = H'(v)$ for some H, then we set $S_1(v) = \frac{1}{2}|v|^2$, $\Phi_1(v) = vF(v) - H(v)$ and compute

$$\begin{aligned}\Phi_1'(v) &= F(v) + vF'(v) - H'(v) = H'(v) + vF'(v) - H'(v) \\ &= vF'(v) = S_1'(v)F'(v).\end{aligned}$$

Hence, *if F is such that $F(v) = H'(v)$, then $S_1(v) = \frac{1}{2}|v|^2$ and $\Phi_1(v) = vF(v) - H(v)$ define entropy and entropy flux functions associated with conservation law (9.2.1).*

We should note that if we consider Burgers' equation, then $H(v) = v^3/6$ and the entropy flux function is given by $\Phi_1(v) = v^3/3$.

Remark 2: We notice also that if for any $c \in \mathbb{R}$ we set $S_2(v) = |v - c|$ and $\Phi_2(v) = \frac{v-c}{|v-c|}[F(v) - F(c)]$ for $v \neq c$, then

$$\Phi_2'(v) = \frac{v-c}{|v-c|}F'(v) = S_2'(v)F'(v).$$

Hence, S_2 and Φ_2 define entropy and entropy flux functions associated with conservation law (9.2.1).

We would like to use the entropy function and entropy flux function to ensure that we can select the correct solution to our initial-value problem. The next result is not exactly what we want, but it shows that the vanishing viscosity solution will satisfy an entropy inequality.

Proposition 9.2.11 *Suppose that conservation law (9.2.1) has a related entropy conservation law with S convex and $S \geq 0$. Suppose that v is a vanishing viscosity solution of conservation law (9.2.1) where the convergence is such that v^ϵ and its derivatives with respect to x converge boundedly to v. Then v satisfies*

$$S(v)_t + \Phi(v)_x \leq 0 \tag{9.2.51}$$

weakly.

Proof: See ref. [62], page 400. To satisfy equation (9.2.51) in the weak sense, we must consider only positive test functions. Hence, we define $C^1_{0,+} = C^1_0 \cup \{\phi : \phi(x,t) \geq 0 \text{ for all } (x,t) \in \mathbb{R} \times \mathbb{R}^+\}$ and require that for $\phi \in C^1_{0,+}$, v must satisfy

$$\int_0^\infty \int_{-\infty}^\infty [\phi_t S(v) + \phi_x \Phi(v)]\, dx\, dt + \int_{-\infty}^\infty \phi(x,0) S(v(x,0))\, dx \geq 0. \tag{9.2.52}$$

If we multiply equation

$$v^\epsilon_t + F(v^\epsilon)_x = \epsilon v^\epsilon_{xx}$$

by $S'(v^\epsilon)$, we get

$$S'(v^\epsilon) v^\epsilon_t + S'(v^\epsilon) F(v^\epsilon)_x = \epsilon S'(v^\epsilon) v^\epsilon_{xx},$$

or

$$S(v^\epsilon)_t + S'(v^\epsilon) F'(v^\epsilon) v^\epsilon_x = \epsilon S'(v^\epsilon) v^\epsilon_{xx}. \tag{9.2.53}$$

Since S and Φ satisfy $\Phi'(v^\epsilon) = S'(v^\epsilon) F'(v^\epsilon)$ and $\Phi'(v^\epsilon) v^\epsilon_x = \Phi(v^\epsilon)_x$, equation (9.2.53) can be rewritten as

$$S(v^\epsilon)_t + \Phi(v^\epsilon)_x = \epsilon S'(v^\epsilon) v^\epsilon_{xx}. \tag{9.2.54}$$

We note that $S'(v^\epsilon) v^\epsilon_{xx} = S(v^\epsilon)_{xx} - S''(v^\epsilon)\left(v^\epsilon_x\right)^2$ and rewrite equation (9.2.54) as

$$S(v^\epsilon)_t + \Phi(v^\epsilon)_x = \epsilon \left[S(v^\epsilon)_{xx} - S''(v^\epsilon)\left(v^\epsilon_x\right)^2 \right]. \tag{9.2.55}$$

If we multiply equation (9.2.55) by $\phi \in C^1_{0,+}$, integrate over $\mathbb{R} \times [0, \infty)$ and use the fact that ϕ has compact support, we get

$$\int_{-\infty}^{\infty}\int_{0}^{\infty} \phi \left[S(v^\epsilon)_t + \Phi(v^\epsilon)_x\right] dx\, dt = \int_{-\infty}^{\infty}\int_{0}^{\infty} \epsilon\phi\Big[S(v^\epsilon)_{xx} - S''(v^\epsilon)\left(v^\epsilon_x\right)^2\Big] dx\, dt.$$

Because the test function ϕ has compact support, the above equation can be written as

$$\int_{a}^{b}\int_{0}^{T} \phi\big[\ S(v^\epsilon)_t + \Phi(v^\epsilon)_x\big]\, dx\, dt = \int_{a}^{b}\int_{0}^{T} \epsilon\phi\Big[S(v^\epsilon)_{xx} - S''(v^\epsilon)\left(v^\epsilon_x\right)^2\Big] dx\, dt.$$

If we integrate the first term on the left by parts with respect to t, the second term on the left by parts with respect to x, the first term on the right by parts twice with respect to x and use the fact that ϕ has compact support, we are left with

$$-\int_a^b \phi(x,0)\ S(v^\epsilon(x,0))\, dx - \int_a^b\int_0^T \phi_t S(v^\epsilon)\, dx\, dt - \int_a^b\int_0^T \phi_x \Phi(v^\epsilon)\, dx\, dt = \epsilon\int_a^b\int_0^T \phi_{xx} S(v^\epsilon)\, dx\, dt - \epsilon\int_a^b\int_0^T \phi S''\left(v^\epsilon_x\right)^2 dx\, dt.$$

We then use the fact that

$$-\epsilon\int_a^b\int_0^T \phi S''\left(v^\epsilon_x\right)^2 dx\, dt \le 0$$

and let $\epsilon \to 0$ to get

$$-\int_a^b \phi(x,0) S(v(x,0))\, dx - \int_a^b\int_0^T \phi_t S(v)\, dx\, dt - \int_a^b\int_0^T \phi_x \Phi\, dx\, dt \le 0,$$

which is the same as equation (9.2.52).

Remark: We have not seen how the extra convergence assumed in the hypothesis was necessary. It should not surprise us that the formal manipulation of integrals and limits that we have performed above might need more technical details and hypotheses.

We now define our second entropy condition.

Definition 9.2.12 Entropy Condition II: *The solution to equation (9.2.12), $v = v(x,t)$, is said to satisfy the Entropy Condition II if there exists an entropy function S and an entropy flux function Φ for which v satisfies inequality (9.2.52) (or (9.2.51) in the weak sense).*

The following proposition, which we include without proof, will illustrate how the entropy function and the entropy flux function can be used to select the correct weak solution to our conservation law initial-value problem. See ref. [62], page 401.

Proposition 9.2.13 *Consider conservation law (9.2.1) where F is convex and a solution to equation (9.2.1) that contains a weak shock. Suppose that Entropy Condition II is satisfied for the solution $v = v(x,t)$ where the entropy function S is strictly convex. Then v is the unique solution to conservation law (9.2.1) that satisfies Entropy Condition II and is the vanishing viscosity solution.*

Remark 1: Consider Burgers' equation; the entropy and entropy flux functions for Burgers' equation given in Remark 1, page 100, $S_1(v) = |v|^2/2$ and $\Phi_1(v) = v^3/6$; initial condition

$$v_0(x) = \begin{cases} 1 & \text{if } x \leq 0 \\ 0 & \text{if } x > 0; \end{cases} \tag{9.2.56}$$

and solution

$$v(x,t) = \begin{cases} 1 & \text{if } x \leq t/2 \\ 0 & \text{if } x > t/2. \end{cases} \tag{9.2.57}$$

To show that solution v given by (9.2.57) satisfies Entropy Condition II for the initial–value problem consisting of Burgers' equation along with initial condition (9.2.56), we must show that S_1 and Φ_1 are entropy and entropy flux functions consistent with Burgers' equation (which we did in Remark 1, page 100) and perform the computation

$$\int_0^\infty \int_{-\infty}^\infty \phi_t S_1(v)\,dx\,dt + \int_0^\infty \int_{-\infty}^\infty \phi_x \Phi_1(v)\,dx\,dt + \int_{-\infty}^\infty \phi(x,0) S_1(v(x,0))\,dx$$
$$= \frac{1}{12}\int_0^T \phi(t/2,t)\,dt. \tag{9.2.58}$$

The computation is left to the reader in HW9.2.7 and involves essentially the same steps used in Example 9.2.1 to show that v is a weak solution to Burgers' equation with initial condition (9.2.56). Then, since ϕ is assumed to be positive, the right hand side is greater than or equal to zero, which is what we were to prove. Hence, solution v given by (9.2.56) satisfies Entropy Condition II.

Remark 2: We again consider Burgers' equation, this time along with the initial condition

$$v_0(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ 1 & \text{if } x > 0 \end{cases}$$

and weak solution

$$v(x,t) = \begin{cases} 0 & \text{if } x \le t/2 \\ 1 & \text{if } x > t/2. \end{cases} \tag{9.2.59}$$

As we did in Remark 1 above, we consider the entropy function $S_1(v) = |v|^2$, the entropy flux function $\Phi_1(v) = v^3/6$ and perform the computation

$$\int_0^\infty \int_{-\infty}^\infty \phi_t S_1(v)\,dx\,dt + \int_0^\infty \int_{-\infty}^\infty \phi_x \Phi_1(v)\,dx\,dt + \int_{-\infty}^\infty \phi(x,0) S_1(v(x,0))\,dx$$
$$= -\frac{1}{6}\int_0^T \phi(x,2x)\,dx.$$

If the solution (9.2.59) were to satisfy Entropy Condition II, the above expression would have to be greater than or equal to zero for all $\phi \in C^1_{0,+}$. However, it is not difficult to see that we can choose a function ϕ that is in $C^1_{0,+}$ and is positive in any open neighborhood of $t = 0$, $x = 0$. (It is easy to see that we can draw such a function. Books on distribution theory would show us how to construct such a function analytically.) Hence, we see that solution (9.2.59) does not satisfy the entropy condition with respect to S_1 and Φ_1.

There are other entropy conditions and relationships between these entropy conditions. For a more complete discussion of entropy conditions see refs. [37], [62] or [35]. Which is the best approach (or whether there are other, better, approaches) is not clear. For this reason we will try to summarize some of the most useful results that connect the ideas of weak solutions to conservation laws, vanishing viscosity solutions and entropy solutions. Since many of these results are very technical, we will not present them as theorems, but will instead list them as facts and provide references for some of the more difficult results.

- If F is convex, then Entropy Condition I is equivalent to the following condition:

  Definition 9.2.14 Entropy Condition $\mathbf{I_a}$: *The solution to equation (9.2.12), $v = v(x,t)$, is said to satisfy Entropy Condition I_a if there exists a constant $E > 0$ such that*

  $$\frac{v(x+a,t) - v(x,t)}{a} < \frac{E}{t} \tag{9.2.60}$$

  for all $a > 0$, $t > 0$ and $x \in \mathbb{R}$.

- There exists a unique solution to equation (9.2.12) that satisfies Entropy Condition $\mathrm{I_a}$ (Theorems 16.1, page 266 and 16.11, page 283, ref. [62]). (One of the hypotheses associated with both of these theorems is that F satisfy $F'' > 0$.)

- The solution discussed in the previous item is a vanishing viscosity solution (ref. [51]).
- Suppose conservation law (9.2.1) has a related entropy conservation law with S convex. If v is a piecewise continuous solution, then across each discontinuity v satisfies

$$s\,[S(v_L) - S(v_R)] - [\Phi(v_L) - \Phi(v_R)] \leq 0, \tag{9.2.61}$$

  where s is the speed of propagation of the discontinuity, and v_L and v_R are solution values on the left and right sides of the discontinuity, respectively. (Corollary 20.7, page 401, ref. [62])
- Inequality (9.2.61) can be used as an entropy condition. When S is strictly convex, F is convex and the shock is a weak shock (the difference between v_L and v_R is small), this entropy condition can be seen to be equivalent to Entropy Condition I. (Theorem 20.8, page 401, ref. [62]).

HW 9.2.7 Perform the computation given in equation (9.2.58).

HW 9.2.8 Show that the discontinuity across $x = 0$ in the solution (9.2.22) given in HW9.2.3 satisfies Entropy Condition I (or Entropy Condition II), while the discontinuities across $x = \pm(1-\alpha)t/2$ do not satisfy any of the entropy conditions. Hence, the solution (9.2.22) does not satisfy an entropy condition.

HW 9.2.9 Show that the weak solutions given by (9.2.38)–(9.2.41) are not entropy solutions.

9.2.5 Solution of Scalar Conservation Laws

In the previous sections, we see that if we want interesting solutions to conservation laws (solutions with discontinuities), we must consider weak solutions. We saw that when we do have a discontinuity in our solution, we get some additional information in that we must satisfy a jump condition across the discontinuity. Also, to eliminate the possibilities of nonunique solutions to the weak formulation of the conservation laws, we must add entropy conditions to our equation. To illustrate some of the concepts we were discussing, in Sections 9.2.2 and 9.2.3 we solved several problems with Burgers' equation and different initial conditions. Even though we do not want to make ourselves experts in the analytic theory of scalar conservation laws, we do want to make it clear that the topics discussed in Section 9.2.2 allow us to solve many scalar conservation law problems and to determine information about solutions to other conservation law problems. For example, we have already seen in HW9.2.1 that the shock

in problem HW0.0.2 begins to form when $t = T_b$ where T_b is the breaking point, $T_b = 1/(2\pi)$. It should not surprise us that this will be a shock (a fact that is obvious if we consider the form of the characteristic curves in Figure 9.2.2), since as soon as the shock forms, the solution locally resembles initial condition (9.2.31) with $v_L > v_R$. Also, as we mentioned earlier, the symmetry of the initial condition about $x = \frac{1}{2}$ along with the symmetry of the equation makes it clear that $F(v_R) = \frac{1}{2}v_R^2 = \frac{1}{2}v_L^2 = F(v_L)$, so the speed of propagation of the shock will be $s = 0$.

If we return to Figure 9.2.2 and consider the characteristic emanating from $x = 0.25$, it is clear that the solution along this characteristic will equal 1.0 until the characteristic intersects the characteristic $x = 0.5$, i.e., when

$$t = \frac{1}{\sin(2\pi 0.25)}(0.5 - 0.25) = 0.25.$$

It is at this time that the shock has fully developed. After this time, the solution curve loses is roundness on the top and the bottom (approaches a sawtooth curve) and the sawtooth begins to damp. It would be very easy to assume that the damping we see is numerical dissipation. However, if you performed a good set of numerical experiments, you would see that this is not the case. It can be proved that the solution should approach the zero solution asymptotically proportionally to $1/t$, Theorem 16.15, page 298, ref. [62]. This damping is not easy to find numerically. Since it is a nonlinear effect, it cannot be predicted based on linear model equations. To find results such as this decay that we cannot prove analytically (admitting that it is proved in ref. [62]), one must perform a set of very careful, critical numerical experiments.

To illustrate that we can solve problems for equations other than Burgers' equation, we consider the following examples. In the examples below, when we say to find a solution to a given problem, we find one solution to a problem that usually has many solutions. However, by now we know which solution we want (or sometimes which solution we do not want, but want to discuss at the moment) and we always find that solution. We should always remember that the solution to the weak formulation of these problems is not unique and that we are finding one of the solutions.

Example 9.2.7 Consider conservation law

$$v_t + \left(\frac{1}{3}v^3\right)_x = 0. \tag{9.2.62}$$

Find a solution to conservation law (9.2.62) along with initial condition

$$v_0(x) = \begin{cases} 2 & \text{if } x \le 0 \\ 1 & \text{if } x > 0. \end{cases} \tag{9.2.63}$$

Solution: In Figure 9.2.13 we plot the characteristics associated with conservation law (9.2.62) along with initial condition (9.2.63). Since the characteristics intersect, we expect that a shock will form. Since $F'(v_L) = 4$ and $F'(v_R) = 1$, a solution with two states

$v_L = 2$ and $v_R = 1$ separated by a shock will satisfy Entropy Condition 1. We use the jump condition to determine the speed of propagation of the discontinuity:

$$s = \frac{F(v_L) - F(v_R)}{v_L - v_R} = \frac{7/3}{1} = \frac{7}{3}.$$

Hence, the discontinuity will propagate along the solution of the differential equation

$$\frac{dx_C}{dt} = s = \frac{7}{3}$$

with $x_C(0) = 0$ or $x_C(t) = \frac{7}{3}t$. The plot of the characteristics impinging on the curve $x_C(t) = \frac{7}{3}t$ is given in Figure 9.2.14. Hence, the solution to problem (9.2.62)–(9.2.63) is given by

$$v(x,t) = \begin{cases} 2 & \text{if } x \leq 7t/3 \\ 1 & \text{if } x > 7t/3. \end{cases} \tag{9.2.64}$$

Solution (9.2.64) satisfies Entropy Condition 1. However, since the flux function F is not convex, we cannot use Proposition 9.2.9 to prove the uniqueness of solution (9.2.64). See HW9.2.10.

Another problem we would like to consider is one where $v_L = 1$ and $v_R = 2$. However, before we consider such a problem, we have to get a better understanding of the fan introduced in Example 9.2.3. Consider the following example.

Example 9.2.8 Discuss the solution to conservation law (9.2.1) along with initial condition (9.2.31) where $v_L < v_R$ and F' is increasing.

Solution: We considered a special case of this problem earlier in Examples 9.2.2 and 9.2.3 where we showed that the problem has a nonunique weak solution: with at least one solution with a jump discontinuity and one continuous solution containing what we referred to as a fan. Later, we showed that the discontinuous solution does not satisfy the entropy condition and the solution with the fan does satisfy the entropy condition.

In this example, we will show that this is really a general result. If we let

$$s = \frac{F(v_L) - F(v_R)}{v_L - v_R}$$

and $x = x_C(t)$ be the solution to

$$\begin{aligned} \frac{dx_C}{dt} &= s \\ x_C(0) &= 0, \end{aligned}$$

then it is not difficult (the necessary calculation is very similar to that done in Example 9.2.2) to show that

$$v(x,t) = \begin{cases} v_L & \text{if } x \leq st \\ v_R & \text{if } x > st \end{cases}$$

is a weak solution to initial–value problem (9.2.1), (9.2.31). It is also easy to see that this solution does not satisfy Entropy Condition I (see Remark 2, page 98). Though it is not as easy, an argument similar to that used in Remark 2, page 103, shows that the solution does not satisfy Entropy Condition II.

The characteristics associated with this problem are given by $x(t) = F'(v_0)t + x_0$, where $v_0 = v_0(x_0)$ and x_0 is an arbitrary point on $\mathbb{R}$. There are three different situations that can occur (while we still have the case that $v_L < v_R$ and F' is increasing): both $F'(v_L)$ and $F'(v_R)$ are positive, both $F'(v_L)$ and $F'(v_R)$ are negative, and $F'(v_L)$

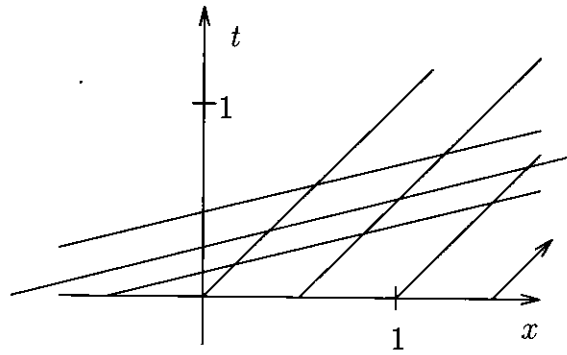

FIGURE 9.2.13. Characteristics associated with the initial–value problem defined by conservation law (9.2.62) along with initial condition (9.2.63).

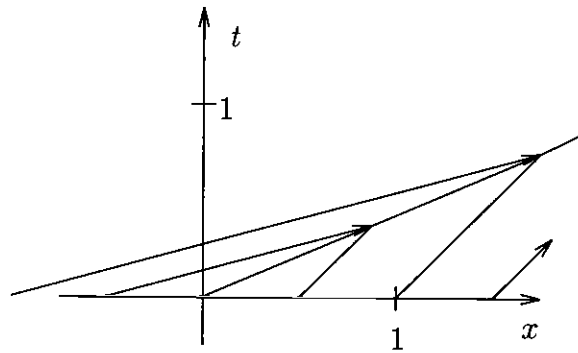

FIGURE 9.2.14. Characteristics associated with solution (9.2.64) of initial–value problem defined by conservation law (9.2.62) along with initial condition (9.2.63).

is negative and $F'(v_R)$ is positive. The plot of the characteristics for the case when $F'(v_L) < 0$ and $F'(v_R) > 0$ is given in Figure 9.2.15. The wedge without any characteristics in this case is due to the fact that the slope of the characteristics to the left of $x = 0$ is negative, while the slope of the characteristics to the right of $x = 0$ is positive. The plots of the characteristics for the other two cases are similar. In those cases the wedge without any characteristics is due to the fact that the slope of the characteristics is given by $1/F'(v_0)$, $v_L < v_R$, and the fact that $1/F'(v_L) > 1/F'(v_R)$.

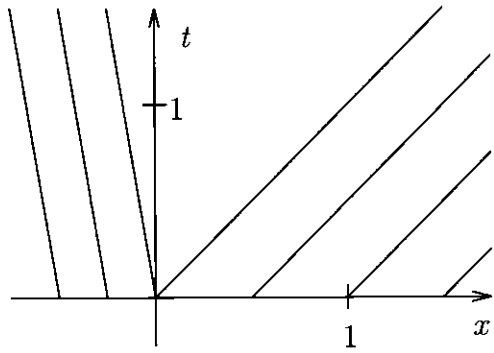

FIGURE 9.2.15. Characteristics associated with the initial-value problem defined by conservation law (9.2.1) along with initial condition (9.2.31) for $v_L < v_R$ when $F'(v_L) < 0 < F'(v_R)$.

The approach to finding a solution is to fill the wedge with a fan. We look for a solution of the form

$$v(x,t) = \psi(x/t), \tag{9.2.65}$$

a solution that will be constant on the characteristic curves in the fan. Substituting solution (9.2.65) into equation (9.2.1) gives

$$-\frac{x}{t^2}\psi'(x/t) + \frac{1}{t}F'(\psi(x/t))\psi'(x/t) = 0$$

or (assuming $\psi'(x/t) \neq 0$)

$$F'(\psi(x/t)) = x/t.$$

Thus, we see that in order for a solution of the form (9.2.65) to satisfy equation (9.2.1), ψ must be the inverse of the function F', i.e., $\psi(x/t) = F'^{-1}(x/t)$.

We note that when $x = F'(v_L)t$ (the characteristic curve on the left side of the fan), we get

$$\psi(x/t) = \psi(F'(v_L)) = F'^{-1}(F'(v_L)) = v_L$$

and when $x = F'(v_R)t$ (the characteristic curve on the right side of the fan), we get

$$\psi(x/t) = \psi(F'(v_R)) = F'^{-1}(F'(v_R)) = v_R.$$

Hence, if we define a solution to be

$$v(x,t) = \begin{cases} v_L & \text{if } x < F'(v_L)t \\ F'^{-1}(x/t) & \text{if } F'(v_L)t \leq x \leq F'(v_R)t \\ v_R & \text{if } x > F'(v_R)t \end{cases} \tag{9.2.66}$$

we have a continuous weak solution to problem (9.2.1), (9.2.31). As was the case for the solution given in Example 9.2.3, solution (9.2.66) will satisfy Entropy Conditions I.

Remark: A justification of trying the use a fan to fill in the wedge can be given as follows. Approximate initial condition (9.2.31) with

$$v_0(x) = \begin{cases} v_L & \text{if } x < -\epsilon \\ v_R + \frac{v_R - v_L}{\epsilon} x & \text{if } -\epsilon \le x < 0 \\ v_R & \text{if } x \ge 0. \end{cases} \tag{9.2.67}$$

In Figure 9.2.16 we give a sequence of plots of the characteristics associated with equation (9.2.1) and initial condition (9.2.67) for various values of ϵ (as we have done before, we let the vertical axis represent both t and v_0 so that we can plot both the characteristics and v_0 on the same graph). It is clear from these plots that as ϵ gets small (approaches zero in plot space?), the plot of characteristics approaches a plot of the usual characteristics plus a fan.

Example 9.2.9 Find a solution to conservation law (9.2.62) along with initial condition

$$v_0(x) = \begin{cases} 1 & \text{if } x \le 0 \\ 2 & \text{if } x > 0. \end{cases} \tag{9.2.68}$$

Solution: Following the approach described in Example 9.2.8, we see that

$$F'(v_L) = F'(1) = 1, \quad F'(v_R) = F'(2) = 4, \quad F'^{-1}(x/t) = \sqrt{x/t}$$

and the solution, similar to solution (9.2.66), can be written as

$$v(x,t) = \begin{cases} 1 & \text{if } x < t \\ \sqrt{x/t} & \text{if } t \le x \le 4t \\ 2 & \text{if } x > 4t. \end{cases}$$

The positive square root is chosen over the negative square root to get the continuity at $x = t$ and $x = 4t$. If we were to choose the negative square root, we see that it would not be a solution because it would not satisfy the jump condition across the discontinuity. We should also note that F' is not increasing. We are saved by the fact that F' is increasing for positive v.

We close this section with an example that shows that the results we have found above are reversed for certain conservation laws (which may not imitate physics as well as the previous examples).

Example 9.2.10 Find a solution to conservation law

$$v_t - \left(\frac{1}{2}v^2\right)_x = 0 \tag{9.2.69}$$

along with initial conditions (9.2.63) and (9.2.68).

Solution: Initial Condition (9.2.63) The characteristic curves associated with conservation law (9.2.69) with initial value (9.2.63) are given by

$$x(t) = F'(v_0)t + x_0 = \begin{cases} -v_L t + x_0 & \text{if } x_0 < 0 \\ -v_R t + x_0 & \text{if } x_0 > 0 \end{cases} = \begin{cases} -2t + x_0 & \text{if } x_0 < 0 \\ -t + x_0 & \text{if } x_0 > 0. \end{cases}$$

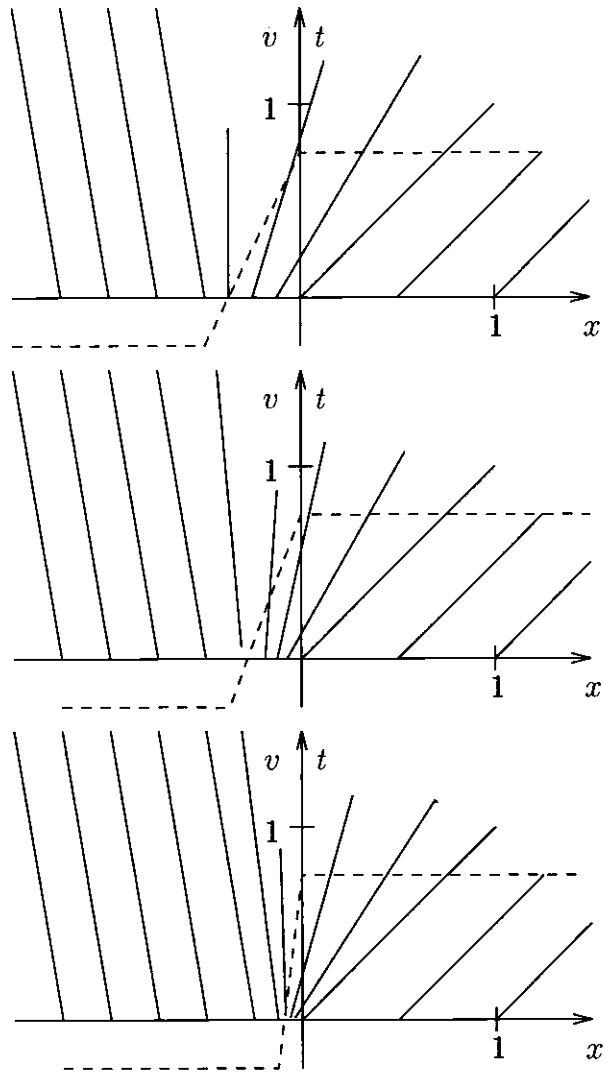

FIGURE 9.2.16. Characteristics associated with the sequence of initial-value problems defined by conservation law (9.2.1) (specifically, Burgers' equation (9.2.14) along with initial conditions (9.2.67) for $v_L < v_R$ and small ϵ).

Plotting these characteristic curves makes it clear that we will have a solution with a fan given by

$$v(x,t) = \begin{cases} 2 & \text{if } x < -2t \\ -x/t & \text{if } -2t \le x \le -t \\ 1 & \text{if } x > -t. \end{cases}$$

Remark: In this example, though F' is not increasing as in Example 9.2.8 (here F' is decreasing), the solution given above can still be written as $v(x,t) = \psi(x,t)$ where

$$\psi(\xi) = \begin{cases} 2 & \text{if } \xi < -2 \\ -\xi & \text{if } -2 \le \xi \le -1 \\ 1 & \text{if } -1 < \xi. \end{cases}$$

If we reconsider the calculation performed in Example 9.2.8, it is easy to see that the important property associated with F is that F' is invertible.

Initial Condition (9.2.68) The characteristic curves associated with conservation law (9.2.69) with initial value (9.2.63) are given by

$$x(t) = \begin{cases} -t + x_0 & \text{if } x_0 < 0 \\ -2t + x_0 & \text{if } x_0 > 0. \end{cases}$$

Plotting these characteristic curves makes it clear that the solution will be the two states $v_L = 1$ and $v_R = 2$ separated by a discontinuity. The discontinuity will propagate along the curve defined by

$$\frac{dx_C}{dt} = s = \frac{F(v_L) - F(v_R)}{v_L - v_R} = -\frac{3}{2}$$

along with $x_C(0) = 0$. Hence, $x_c(t) = -3t/2$. The solution to problem (9.2.69), (9.2.68) is given by

$$v(x,t) = \begin{cases} 1 & \text{if } x \le -3t/2 \\ 2 & \text{if } x > -3t/2. \end{cases} \tag{9.2.70}$$

Remark: We note that v given by

$$v(x,t) = \begin{cases} 2 & \text{if } x \le -3t/2 \\ 1 & \text{if } x > -3t/2 \end{cases}$$

is also a solution to conservation law (9.2.69) along with initial condition (9.2.63). However, since in this case $F'(v_L) = -2$ and $F'(v_R) = -1$, it is clear that v does not satisfy Entropy Condition I. When we consider conservation law (9.2.69) along with initial condition (9.2.68), it is easy to see that the solution given by (9.2.70) will satisfy Entropy Condition I.

Before we leave this section, we want to mention that though we have now solved several initial-value problems associated with nonlinear conservation laws, we have still just solved easy problems. One aspect of problems involving nonlinear conservation laws that we have omitted completely is that of problems that involve multiple shocks. It is possible and rather common to have more than one shock in a problem and such that the shocks interact. We will leave these problems for readers to find in the literature when they are ready to numerically solve more difficult problems.

HW 9.2.10 Show that solution (9.2.64) satisfies Entropy Condition $\mathrm{I_{nc}}$.

HW 9.2.11 Using whichever of your codes solving HW0.0.2 gave you the best results, compute several solutions near $t = 1/(2\pi)$ and $t = 0.25$ to verify that $T_b = 1/(2\pi)$ is the breaking point and that the shock is fully developed at the time $t = 0.25$. Using solutions for larger times, verify that the solution damps proportionally to $1/t$. Support your results by repeating the computation with a smaller Δx and Δt.

HW 9.2.12 Find a solution that contains a fan to conservation law (9.2.62) along with initial condition

$$v_0(x) = \begin{cases} -1 & \text{if } x \leq 0 \\ 1 & \text{if } x > 0. \end{cases}$$

HW 9.2.13 Solve the problem given by conservation law

$$v_t + \left(\frac{1}{2}e^{2v}\right)_x = 0, \quad x \in \mathbb{R}, \ t > 0$$

along with initial conditions (9.2.63) and (9.2.68).

HW 9.2.14 Solve the problem given by conservation law

$$v_t + \left(\frac{1}{4}v^4\right)_x = 0, \quad x \in \mathbb{R}, \ t > 0$$

with initial conditions (9.2.63) and (9.2.68).

9.3 Theory of Systems of Conservation Laws

We now return to the vector version of a conservation law, (9.1.1). Though there will be many similarities between the results for vector and scalar conservation laws, there will be some differences that are technical and sometimes difficult to describe. In this section, we will emphasize the differences between the scalar and vector conservation law results. We begin by noting that the nonconservative form of equation (9.1.1) is given by

$$\mathbf{v}_t + \mathbf{F}'(\mathbf{v})\mathbf{v}_x = \boldsymbol{\theta} \tag{9.3.1}$$

where $\mathbf{F}'(\mathbf{v})$ is the derivative of $\mathbf{F}$ with respect to $\mathbf{v}$ and is given by the $K \times K$ matrix of partial derivatives, $\mathbf{F}'(\mathbf{v}) = [\partial F_i/\partial v_j]_{K\times K}$. Conservation law (9.1.1) is said to be **strictly hyperbolic** if $\mathbf{F}'(\mathbf{v})$ is diagonalizable with K real, distinct eigenvalues. We order the eigenvalues as

$$\nu_1(\mathbf{v}) < \nu_2(\mathbf{v}) < \cdots < \nu_K(\mathbf{v})$$

and denote the basis of the associated eigenvectors as $\mathbf{r}_1(\mathbf{v}), \ldots, \mathbf{r}_K(\mathbf{v})$. We say that the m-th field of conservation law (9.1.1) is **genuinely nonlinear**

if grad $\nu_m(\mathbf{v}) \cdot \mathbf{r}_m(\mathbf{v}) \neq 0$ (where *grad* is the gradient with respect to the components of the $\mathbf{v}$ vector) for all $\mathbf{v} \in \mathbb{R}^K$. If grad $\nu_m(\mathbf{v}) \cdot \mathbf{r}_m(\mathbf{v}) = 0$ for all $\mathbf{v} \in \mathbb{R}^K$, we say that the m-th field of conservation law (9.1.1) is **linearly degenerate**.

An example of a K-system conservation law is the system of equations describing the flow of a gas in a shock tube, equations (0.0.7)–(0.0.9), HW0.0.3. We should recall that in Sections 6.10.2 and 7.9.2 we used the nonconservative form of equations (0.0.7)–(0.0.9) in an attempt to solve HW0.0.3.

To begin work on K-system conservation laws, we return to the beginning of this chapter to see which of the concepts, definitions, results, etc. for scalar conservation laws carry over and which do not carry over to K-system conservation laws. The K-system conservation laws have K families of characteristics. Analogously to what we did in Section 9.2.1, we define the **characteristic curves** of conservation law (9.1.1) to be solutions of the initial-value problem

$$x'(t) = \nu_m(\mathbf{v}(x(t), t)) \tag{9.3.2}$$

$$x(0) = x_0 \tag{9.3.3}$$

for some x_0 and $m = 1, \ldots, K$. Obviously, since ν_m depends on $\mathbf{v}$, the characteristic curves will depend on $\mathbf{v}$, and we cannot easily solve initial-value problem (9.3.2)–(9.3.3). Often, in the literature, plots of the characteristic curves are depicted as straight lines. The reader must be aware that these are linear approximations (tangent lines) of the characteristic curves. More importantly, we see that since we cannot imitate calculation (9.2.5)–(9.2.6), *we do not obtain the result that the solution is constant along a characteristic curve.* As we will see later, we will obtain that result under special circumstances. Also, we may as well realize from the beginning that the technique we used so often for linear systems in Chapter 6—transform the system to characteristic variables, obtain results for each of the equations in characteristic variables, and then transform back to the primitive variables—will not work for nonlinear systems because S and S^{-1} would both depend on $\mathbf{v}$.

We do notice that the computation performed in (9.2.10)–(9.2.11) along with the discussion preceding and following this computation will apply to K-system conservation laws, and we obtain the following **weak formulation of conservation law (9.1.1)**.

$$\boldsymbol{\theta} = \int_0^\infty \int_{-\infty}^\infty [\mathbf{v}\phi_t + \mathbf{F}(\mathbf{v})\phi_x]\, dx\, dt + \int_{-\infty}^\infty \mathbf{v}(x,0)\phi(x,0)\, dx. \tag{9.3.4}$$

In addition, it is not hard to mimic the work done in Section 9.2.3 to see that *a discontinuous solution to equation (9.3.4) must satisfy the jump, or Rankine-Hugoniot, condition*

$$s(\mathbf{v}_L - \mathbf{v}_R) = \mathbf{F}(\mathbf{v}_L) - \mathbf{F}(\mathbf{v}_R) \tag{9.3.5}$$

across the discontinuity. As before, $s = dx_C/dt$ is the speed of propagation of the discontinuity, where the solution to the conservation law is discontinuous across the curve $x = x_C(t)$.

As in the case of scalar conservation laws, the solution to equation (9.3.4) is not unique. We again need the concept of a vanishing viscosity solution or some sort of entropy condition with which to choose our solution. We still want solutions $\mathbf{v}$ that are limits of functions $\mathbf{v}^\epsilon$ that satisfy an equation of the form

$$\mathbf{v}_t^\epsilon + \mathbf{F}(\mathbf{v}^\epsilon)_x = \epsilon \mathbf{v}_{xx}^\epsilon \tag{9.3.6}$$

(or in some cases with a right hand side of the form $\epsilon B\mathbf{v}_{xx}^\epsilon$, where B is a constant matrix that has eigenvalues that have positive real parts). It is not difficult to see formally that *limits of solutions to equation (9.3.6) are weak solutions of conservation law (9.1.1).* As in the scalar case, we want to have conditions that we can impose on the solutions that will guarantee that they are vanishing viscosity solutions.

The K-system analogue to Entropy Condition I is described as follows.

Definition 9.3.1 *Let $x = x_C(t)$ be a curve across which the solution $\mathbf{v}$ of conservation law (9.1.1) is discontinuous and let $\mathbf{v}_L$ and $\mathbf{v}_R$ be the values of $\mathbf{v}$ on the left and right sides of the discontinuity, which moves with speed $s = dx_C/dt$. If for $k = 1$,*

$$\nu_1(\mathbf{v}_L) > s > \nu_1(\mathbf{v}_R) \tag{9.3.7}$$

while

$$s < \nu_2(\mathbf{v}_R); \tag{9.3.8}$$

or for some index k, $2 \le k \le K-1$,

$$\nu_k(\mathbf{v}_L) > s > \nu_k(\mathbf{v}_R) \tag{9.3.9}$$

while

$$\nu_{k-1}(\mathbf{v}_L) < s < \nu_{k+1}(\mathbf{v}_R); \tag{9.3.10}$$

or for $k = K$,

$$\nu_K(\mathbf{v}_L) > s > \nu_K(\mathbf{v}_R) \tag{9.3.11}$$

while

$$\nu_{K-1}(\mathbf{v}_L) < s, \tag{9.3.12}$$

then we call this discontinuity a k **shock** *and say that the solution satisfies* **Entropy Condition $\mathbf{I}_\mathbf{v}$.**

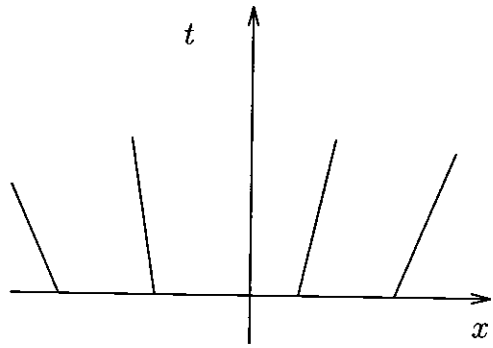

FIGURE 9.3.1. Characteristics associated with the solution to a typical scalar conservation law.

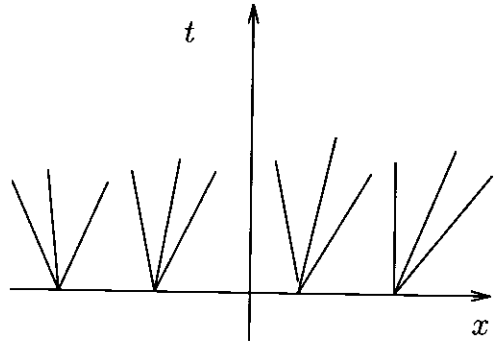

FIGURE 9.3.2. Characteristics associated with the solution to a typical 3-system conservation law.

Remark 1: We should realize that for any solution $\mathbf{v}$ to conservation law (9.1.1), we will have K characteristic curves associated with this solution. In Figure 9.3.1 we plot what the characteristics look like for a solution to a scalar conservation law, and in Figure 9.3.2 we plot the analogous characteristics for a solution to a 3-system conservation law. The point is that whereas in the scalar case we have one characteristic curve emanating from each point, in the 3-system case we have three characteristics emanating from each point (K characteristics for the general K-system case). Analogously to what we did for the scalar case, we must determine our solution based on the characteristic curves. We must realize that we can still have the various characteristic curves intersecting. The intersections will still cause difficulties, and depending on how they intersect, the intersections will determine whether the intersections are due to shocks or fans.

Remark 2: When the m-th field is linearly degenerate, then $s = \nu_m(\mathbf{v}_L) = \nu_m(\mathbf{v}_R)$, and the discontinuity will propagate along the m-characteristic. See [62], page 334. Obviously, since $\nu_m(\mathbf{v}_L) = \nu_m(\mathbf{v}_R)$, a linearly degenerate field cannot satisfy inequality (9.3.9) of Entropy Condition $\mathrm{I_v}$. Such discontinuities are called **contact discontinuities**. Though contact discontinuities are not shocks, when you view a solution that has either contact discontinuities or shocks, you cannot distinguish between the two types of discontinuities.

Remark 3: In order for $\mathbf{v}$ to satisfy Entropy Condition $\mathrm{I_v}$, the characteristic curves must be such that they, along with the jump condition, determine $\mathbf{v}$ at the shock (on both sides) and the speed of propagation s. From inequalities (9.3.9)–(9.3.10) along with our assumption that $\nu_1(\mathbf{v}) < \nu_2(\mathbf{v}) < \cdots < \nu_K(\mathbf{v})$, we see that $K - k + 1$ characteristics impinge on the curve of discontinuity from the left in the direction of increasing t (the characteristics associated with $\nu_k(\mathbf{v}_L)$ as well as those associated with $\nu_{k+1}(\mathbf{v}_L), \ldots, \nu_K(\mathbf{v}_L)$), and k characteristics impinge on the curve of discontinuity from the right in the direction of increasing t (those associated with $\nu_1(\mathbf{v}_R), \ldots, \nu_k(\mathbf{v}_R)$). These $K+1$ pieces of information along with the $K-1$ relations obtained from the jump condition are sufficient to determine the $2K$ values that $\mathbf{v}$ assumes on both sides of the curve of discontinuity and to determine the curve itself. A similar argument holds for inequalities (9.3.7), (9.3.8) and (9.3.11), (9.3.12) with $k = 1$ and $k = K$, respectively. For example, when $k = K$, there will be one characteristic that intersects the curve of discontinuity from the left in the direction of increasing t (the characteristic curve associated with $\nu_K(\mathbf{v}_L)$) and K characteristic curves that intersect the curve of discontinuity from the right in the direction of increasing t (the characteristic curves associated with $\nu_1(\mathbf{v}_R), \ldots, \nu_K(\mathbf{v}_R)$). As in the case for $2 \leq k \leq K-1$, these intersections provide us with $K+1$ pieces of information that along with the jump condition will enable us to determine $\mathbf{v}$ on both sides of the curve and s. If for some k, $\mathbf{v}$ does not satisfy one of the conditions (9.3.7)–(9.3.8), (9.3.9)–(9.3.10) or (9.3.11)–

(9.3.12), then there will not be enough information to determine $\mathbf{v}$ and s.

Remark 4: A variation of Definition 9.3.1 that is common and equivalent to Definition 9.3.1 is to replace inequalities (9.3.9), (9.3.10) by inequalities

$$\nu_k(\mathbf{v}_R) < s < \nu_{k+1}(\mathbf{v}_R) \tag{9.3.13}$$

and

$$\nu_{k-1}(\mathbf{v}_L) < s < \nu_k(\mathbf{v}_L). \tag{9.3.14}$$

See HW9.3.3. Also, there are analogous inequalities that are equivalent to (9.3.7), (9.3.8) and (9.3.11), (9.3.12) when $k = 1$ and $k = K$, respectively. As was the case in Definition 9.3.1, the inequalities for $k = 1$ and $k = K$ must be such that $K + 1$ characteristics will impinge on the curve of discontinuity in the direction of increasing t.

Remark 5: There are at least two ways in which a discontinuity in a solution of a conservation law does not satisfy Entropy Condition $\mathrm{I_v}$. As we stated in Remark 2 above, one way is if a field is linearly degenerate and the eigenvalues are equal on both sides of the discontinuity. Another way that Entropy Condition $\mathrm{I_v}$ might not be satisfied is if the inequalities in the eigenvalues are reversed in direction, for example, if $\nu_k(\mathbf{v}_L) < \nu_k(\mathbf{v}_R)$. In this case, analogous to the solutions to scalar conservation laws, the solution with a fan between the appropriate characteristic curves is the solution that will satisfy Entropy Condition $\mathrm{I_v}$. One of the properties of the fans is that like the solutions to scalar conservation laws, the solution will be constant along the characteristic curves in the fan. See [62], page 323.

Also, as in the case of scalar conservation laws, we can use entropy and entropy flux functions to define an entropy condition analogous to Entropy Condition II. If the scalar valued functions $S = S(\mathbf{v})$ and $\Phi = \Phi(\mathbf{v})$ satisfy (i) S is convex ($S'' > 0$) and (ii) S and Φ satisfy

$$\Phi'(\mathbf{v}) = \mathbf{F}'(\mathbf{v})S'(\mathbf{v}), \tag{9.3.15}$$

then S and Φ are said to be **entropy** and **entropy flux** functions, respectively. Smooth solutions of equation (9.1.1) along with functions S and Φ that satisfy equation (9.3.15) will satisfy

$$S(\mathbf{v})_t + \Phi(\mathbf{v})_x = 0.$$

Definition 9.3.2 *We say that the solution to equation (9.3.4) satisfies* **Entropy Condition $\mathrm{II_v}$** *if* $\mathbf{v}$ *satisfies*

$$S(\mathbf{v})_t + \Phi(\mathbf{v})_x \leq 0 \tag{9.3.16}$$

in the weak sense for some entropy and entropy flux functions S *and* Φ, *i.e., for any* $\phi \in C^1_{0,+}$, $\mathbf{v}$ *satisfies*

$$\int_0^\infty \int_{-\infty}^\infty [S(\mathbf{v})\phi_t + \Phi(\mathbf{v})\phi_x]\, dx\, dt + \int_{-\infty}^\infty \phi(x,0)S(\mathbf{v}(x,0))\, dx \geq 0. \quad (9.3.17)$$

Obviously, Entropy Condition $\mathrm{II_v}$ is very similar to Entropy Condition II.

Remark: It is very common for a scalar conservation to have one or more pairs of entropy and entropy flux functions. *It is common for a* K*-system conservation law not to have an entropy or entropy flux function.* Equation (9.3.15) is a system of K partial differential equations for determining S and Φ. For $k \geq 2$, this system is overdetermined and does not generally have a solution. There are some important (the equations of gas dynamics) and some general (when $\mathbf{F}'(\mathbf{v})$ is symmetric) classes of equations where we are able to find entropy and entropy flux functions. See HW9.3.1 and HW9.3.2.

We next provide results relating weak solutions to conservation law (9.1.1), vanishing viscosity solutions and entropy conditions. As with the analogous list given in Section 9.2.4, these results are very technical, where the technicalities are beyond the scope of this text. For a more precise statement of these results, consult the references provided.

- If $\mathbf{v}$ is a solution with a discontinuity that satisfies Entropy Condition $\mathrm{I_v}$, then $\mathbf{v}$ is the unique solution satisfying Entropy Condition $\mathrm{I_v}$ and is the vanishing viscosity solution (ref. [34]).

- Let (9.1.1) be a conservation law that admits a solution to equation (9.3.15) where S is strictly convex. Let $\mathbf{v}$ be a vanishing viscosity solution of conservation law (9.1.1). Then $\mathbf{v}$ satisfies Entropy Condition $\mathrm{II_v}$ (Theorem 5.6, page 32, ref. [35]).

- If $\mathbf{v}$ is a solution to conservation law (9.1.1) that contains a weak shock and satisfies Entropy Condition $\mathrm{II_v}$ where the entropy function S is strictly convex, then $\mathbf{v}$ is the unique solution satisfying Entropy Condition $\mathrm{II_v}$ and is the vanishing viscosity solution.

  - If $\mathbf{v}$ is piecewise continuous, then across a discontinuity, $\mathbf{v}$ satisfies

    $$s\,[S(\mathbf{v}_L) - S(\mathbf{v}_R)] - [\Phi(\mathbf{v}_L) - \Phi(\mathbf{v}_R)] \leq 0 \qquad (9.3.18)$$

    (Theorem 5.6, page 32, ref. [35]). As was the case with inequality (9.2.61), inequality (9.3.18) can be used as an entropy condition. Also, when the shock is a weak shock, entropy condition (9.3.18) is equivalent to Entropy Condition $\mathrm{I_v}$ (Theorem 5.7, page 32, ref. [35]).

- If $\mathbf{v}$ is a solution of a K-system conservation law that has only contact discontinuities (no shocks or fans), then $\mathbf{v}$ is a limit of continuous solutions (ref. [35], page 44).

Remark: We will find that many results for K-system conservation laws require that we consider **weak shocks**, i.e., shocks for which $\mathbf{v}_L - \mathbf{v}_R$ is sufficiently small. How we measure what it means to be sufficiently small depends on with which norm we are working. This hypothesis is needed at times to eliminate the possibility of undesirable behavior of the solutions. Whether a shock is a weak shock or not is not something that can be determined computationally and will not be something that particularly concerns us.

HW 9.3.1 Find an entropy and entropy flux function for system of equations (0.0.7)–(0.0.9).

HW 9.3.2 Assume that $\mathbf{F}'(\mathbf{v})$ is symmetric and that the function $g = g(\mathbf{v})$ satisfies

$$\frac{\partial g}{\partial v_m} = F_m, \quad m = 1, \ldots, K.$$

Show that

$$S(\mathbf{v}) = \sum_{m=1}^{K} v_m^2 \text{ and } \Phi(\mathbf{v}) = \sum_{m=1}^{K} v_m F_m - g$$

satisfy equation (9.3.15).

HW 9.3.3 Show that the inequalities (9.3.9), (9.3.10) are equivalent to inequalities (9.3.13), (9.3.14).

9.3.1 Solutions of Riemann Problems

The Riemann problem consists in considering either conservation law (9.2.1) or (9.1.1) on $\mathbb{R}$ along with the initial condition

$$v(x,0) = \begin{cases} v_L & \text{for } x < 0 \\ v_R & \text{for } x > 0 \end{cases} \tag{9.3.19}$$

or

$$\mathbf{v}(x,0) = \begin{cases} \mathbf{v}_L & \text{for } x < 0 \\ \mathbf{v}_R & \text{for } x > 0, \end{cases} \tag{9.3.20}$$

respectively. In Sections 9.2.2 and 9.2.3 we found solutions to and obtained results for Riemann problems for Burgers' equation. In Section 9.2.5 we solved several more scalar Riemann problems. In HW0.0.3 we have been attempting to numerically solve a Riemann problem for the gas dynamics

equations. And finally, it is easy to solve Riemann problems for either a scalar or K-system linear conservation law.

It should not be too difficult to understand that it is difficult to obtain solutions for K-system Riemann problems. Most of what we will do in this section is to try to describe to the reader the general form of solutions to Riemann problems. Since we already know how to solve Riemann problems for scalar conservation laws, we will concentrate on K-system Riemann problems. At times the results we obtain can be used to solve a particular Riemann problem. More importantly, these results will show us what to expect when we try to solve a Riemann problem. As we shall see later, the solution to Riemann problems associated with linear K-systems will be important in our discussion of numerical solutions of conservation laws.

The good news is the fact that solutions to Riemann problems for K-system conservation laws are very much like solutions to scalar Riemann problems in that they still can be written in terms of the **similarity variable** $\xi = x/t$. We noted in Section 9.2.2 that the solutions to problems solved in Examples 9.2.1, 9.2.2 and 9.2.3 can be written in the form $v(x,t) = \psi(x/t)$. In Example 9.2.8 we showed that if F' is increasing (if F' is invertible), the solution to a general scalar Riemann problem can be written as $\psi(x/t)$. If we consider a K-system Riemann problem, it is easy to see that by using approximately the same computation used in Example 9.2.8, the solution to the general K-system Riemann problem can be written as $\mathbf{v}(x,t) = \boldsymbol{\psi}(x/t)$. This solution is referred to as the **similarity solution**.

Another way in which the solutions to the K-system Riemann problem are very much like the solutions to the scalar Riemann problems is that the solutions involve shocks and fans. It should not be surprising that solutions to Riemann problems for K-system conservation laws can also involve contact discontinuities. The difficulty is that because the system now has K characteristic curves and the solution at any point has K components, complicated combinations of shocks, fans and contact discontinuities are possible. For a complete understanding of these concepts, one should define the **Hugoniot Locus** associated with the conservation laws. We shall not do that here and refer any reader that wants a better understanding of the solutions of the K-system conservation laws to ref. [37], Chapters 7 and 8, or ref. [11]. We will instead try to describe what we should expect as a solution of a K-system Riemann problem.

Linear K-system We begin by considering the easiest K-system, the initial–value problem

$$\mathbf{v}_t + A\mathbf{v}_x = \boldsymbol{\theta}, \quad x \in \mathbb{R}, \ t > 0 \tag{9.3.21}$$

along with initial condition (9.3.20). We must understand that initial–value problem (9.3.21), (9.3.20) is a very easy Riemann problem, since the partial differential equation (the conservation law) is linear. We also make special

note that the linear partial differential equation (9.3.21) is written differently from the way we wrote these types of problems in Chapter 6. We hate to do this to the reader, but since we will apply our linear results to help solve nonlinear problems, it is most advantageous for us to now write our linear conservation law as we do in (9.3.21).

Of course, we assume that the matrix A is diagonalizable and that the eigenvalues of A satisfy $\nu_1 < \nu_2 < \cdots < \nu_K$. Using a method and notation similar to that used in Chapter 6, we know that the Riemann problem (9.3.21), (9.3.20) can be rewritten as the system of partial differential equations

$$\mathbf{V}_t + D\mathbf{V}_x = \boldsymbol{\theta}, \quad x \in \mathbb{R}, \ t > 0 \tag{9.3.22}$$

with initial condition

$$\mathbf{V}(x,0) = \begin{cases} \mathbf{V}_L & \text{for } x \leq 0 \\ \mathbf{V}_R & \text{for } x > 0 \end{cases} \tag{9.3.23}$$

where $\mathbf{V} = S\mathbf{v}$ and matrix D is the diagonal matrix with $\nu_1, \ldots, \nu_K$ on the diagonal. Of course, the advantage of rewriting problem (9.3.21), (9.3.20) in the form (9.3.22)–(9.3.23) is that the problem has now been uncoupled. If we let $\mathbf{V} = [V_1 \ \cdots \ V_K]^T$, problem (9.3.22)-(9.3.23) can be written as

$$V_{k_t} + \nu_k V_{k_x} = 0, \quad x \in \mathbb{R}, \ t > 0 \tag{9.3.24}$$

with initial condition

$$V_k(x,0) = V_{k_0} = \begin{cases} V_{L_k} & \text{for } x \leq 0 \\ V_{R_k} & \text{for } x > 0 \end{cases} \tag{9.3.25}$$

for $k = 1, \ldots, K$. If for a fixed k we consider initial–value problem (9.3.24)–(9.3.25), we know that the solution is given by $V_k(x,t) = V_{k_0}(x - \nu_k t)$. Hence the solution to Riemann problem (9.3.22)–(9.3.23) is given by

$$\mathbf{V}(x,t) = \begin{bmatrix} V_{1_0}(x - \nu_1 t) \\ \vdots \\ V_{K_0}(x - \nu_K t) \end{bmatrix} = \sum_{j=1}^{K} V_{j_0}(x - \nu_j t)\mathbf{u}_j \tag{9.3.26}$$

(where $\mathbf{u}_j$ is the unit vector with a 1 in the j-th component and 0's elsewhere) and the solution to Riemann problem (9.3.21), (9.3.20) is

$$\mathbf{v}(x,t) = S^{-1}\mathbf{V}(x,t) = \sum_{j=1}^{K} V_{j_0}(x - \nu_j t)\mathrm{r}_j \tag{9.3.27}$$

where r_j, $j = 1, \ldots, K$, are the eigenvectors of matrix A. We note that we have used the fact that $\mathrm{r}_j = S^{-1}\mathbf{u}_j$, $j = 1, \ldots, K$. This fact is true

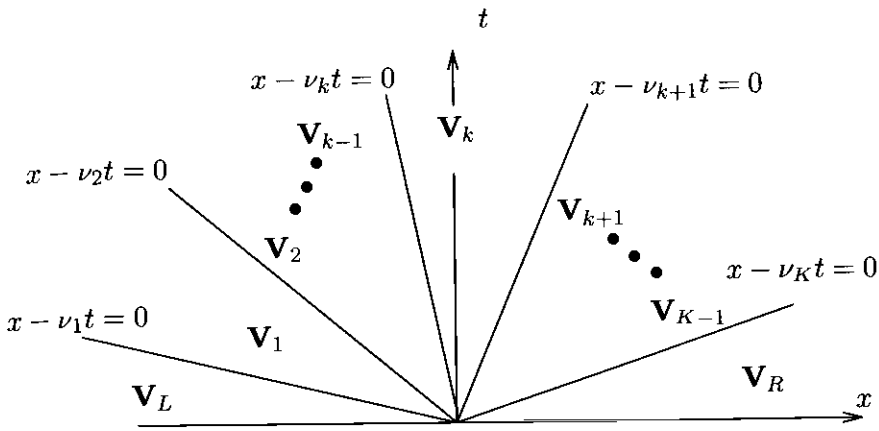

FIGURE 9.3.3. Characteristics and solutions associated with Riemann problem (9.3.22)–(9.3.23).

because we construct the matrix S^{-1} by making the j-th column of S^{-1} be the vector $\mathbf{r}_j$.

To understand the character of solution (9.3.27), consider solution $\mathbf{V}$ given by (9.3.26) and assume that $\mathbf{V}_L = [V_{L_0} \ \cdots \ V_{L_K}]^T$ and $\mathbf{V}_R = [0 \ \cdots \ 0]^T$. In Figure 9.3.3 we have plotted the curves $x - \nu_j t = 0$, $j = 1, \ldots, K$. We see that the way we have drawn Figure 9.3.3, we have assumed that $\nu_1 < \nu_2 < \cdots < \nu_k < 0$ and $0 < \nu_{k+1} < \cdots < \nu_K$. We note the following facts concerning this solution.

1. For x and t values that satisfy $x - \nu_1 t < 0$, we have $x/t < \nu_1$ and $x - \nu_k t < 0$ for all $k = 1, \ldots, K$, each $V_{k_0}(x - \nu_k t) = V_{L_k}$ and $\mathbf{V}(x,t) = \mathbf{V}_L$.

2. As the point (x,t) crosses the characteristic curve $x - \nu_1 t = 0$, only $V_{1_0}(x - \nu_1 t)$ changes in value, and for all (x,t) such that $\nu_1 < x/t < \nu_2$, we have $x - \nu_1 t > 0$ and $x + \nu_k t < 0$ for $k = 2, \ldots, K$, and the solution becomes
$$\mathbf{V}(x,t) = \mathbf{V}_1 = [0 \ V_{L_2} \ \cdots \ V_{L_K}]^T.$$

3. In the wedge $\nu_k < x/t < \nu_{k+1}$ the solution is given by
$$\mathbf{V}(x,t) = \mathbf{V}_k = \sum_{j=1}^{K} V_{j_0}(x - \nu_j t)$$
$$= \sum_{j=k+1}^{K} V_{j_0}(x - \nu_j t) = [0 \ \cdots \ 0 \ V_{L_{k+1}} \ \cdots \ V_{L_K}]^T.$$
As the point (x,t) crosses the characteristic curve $x - \nu_{k+1} t = 0$, we have $x - \nu_j t > 0$ for $j = 1, \ldots, k+1$, $x - \nu_j t < 0$ for $j = k+2, \ldots, K$,

and the solution becomes

$$\mathbf{V}(x,t) = \mathbf{V}_{k+1} = [0 \cdots 0\ V_{L_{k+2}} \cdots V_{L_K}]^T.$$

4. And finally, in the wedge $\nu_{K-1} < x/t < \nu_K$ the solution is given by $\mathbf{V}(x,t) = \mathbf{V}_{K-1} = [0 \cdots 0\ V_{L_K}]^T$. As the point (x,t) crosses the characteristic curve $x - \nu_K t = 0$, the solution becomes $\mathbf{V}(x,t) = \mathbf{V}_R = [0 \cdots 0]^T$.

We note that the solution to Riemann problem (9.3.21), (9.3.20) ($\mathbf{v}$ given in (9.3.27)), behaves in exactly the same way as does solution $\mathbf{V}$, (9.3.26). The solution involves a transition from $\mathbf{v}_L$ on the left through the $K-1$ intermediate states, $\mathbf{v}_1, \ldots, \mathbf{v}_{K-1}$ to $\mathbf{v}_R$. Of course, since $\mathbf{v}$ is more complex than $\mathbf{V}$ (involving $\mathbf{r}_j$ instead of $\mathbf{u}_j$), expressions for the intermediate states are more complex. However, when $\mathbf{v}_R$ is chosen to be $\boldsymbol{\theta}$ as we did above, the intermediate states can be determined by multiplying the expression for $\mathbf{V}_k$ on the left by S^{-1} and can be written as $\mathbf{v}_k = \sum_{j=k+1}^{K} V_{j_0}(x - \nu_j t)\mathbf{r}_j$, $k = 1, \ldots, K-1$.

Remark 1: We note that using 1 through 4 above, the solution to Riemann problem (9.3.22)–(9.3.23) can be written as

$$\mathbf{V}(x,t) = \mathbf{V}_L - \sum_{x/t > \nu_j} V_{L_j}\mathbf{u}_j \tag{9.3.28}$$

$$= \mathbf{V}_R + \sum_{x/t < \nu_j} V_{L_j}\mathbf{u}_j. \tag{9.3.29}$$

If we multiply solutions (9.3.28) and (9.3.29) on the left by S^{-1}, we see that the solution to Riemann problem (9.3.21), (9.3.20) can be written as

$$\mathbf{v}(x,t) = \mathbf{v}_L - \sum_{x/t > \nu_j} V_{L_j}\mathbf{r}_j \tag{9.3.30}$$

$$= \mathbf{v}_R + \sum_{x/t < \nu_j} V_{L_j}\mathbf{r}_j. \tag{9.3.31}$$

Remember that these solutions depend on the fact that $\mathbf{V}_R$, and hence $\mathbf{v}_R$, is zero.

Remark 2: We note especially that at $x = 0$ the solution to the Riemann problem (9.3.22)–(9.3.23), and hence (9.3.21) and (9.3.20) are constant for all t and equal to $\mathbf{V}_k$ (as given in Figure 9.3.3) and $\mathbf{v}_k$, respectively. This follows from the fact that like the solution to the scalar problem, the solutions to (9.3.22)–(9.3.23) and (9.3.21), (9.3.20) are similarity solutions and can be written as $\mathbf{v}(x,t) = \boldsymbol{\psi}(x/t)$, so that $\mathbf{v}(0,t) = \boldsymbol{\psi}(0)$. We also note that we can write the solution as

$$\mathbf{v}_k = \mathbf{v}_R + \sum_{j=k+1}^{K} V_{j_0}(x - \nu_j t)\mathbf{r}_j, \tag{9.3.32}$$

or

$$\mathbf{v}_k = \mathbf{v}_R + \sum_{\nu_k > 0} V_{j_0}(x - \nu_j t)\mathbf{r}_j = \mathbf{v}_L - \sum_{\nu_k < 0} V_{j_0}(x - \nu_j t)\mathbf{r}_j. \qquad (9.3.33)$$

Remark 3: The assumption that $\mathbf{V}_R = \boldsymbol{\theta}$ was only for the convenience of describing the transition of the solution as the point (x,t) crossed the characteristic curves from left to right. If $\mathbf{V}_R \neq \boldsymbol{\theta}$, the problem can be transformed to the problem considered above by solving the Riemann problem with initial condition

$$\mathbf{V}'(x,0) = \begin{cases} \mathbf{V}'_L = \mathbf{V}_L - \mathbf{V}_R & \text{when } x \leq 0 \\ \mathbf{V}'_R = \boldsymbol{\theta} & \text{when } x > 0 \end{cases}$$

and writing the solution as $\mathbf{V}(x,t) = \mathbf{V}'(x,t) + \mathbf{V}_R$. Obviously, the analogous result holds for solutions to Riemann problem (9.3.21), (9.3.20). We note that solutions given by (9.3.28), (9.3.29), (9.3.30), (9.3.31), (9.3.32) and (9.3.33) are correctly written for nonzero $\mathbf{V}_R$ and $\mathbf{v}_R$.

Remark 4: If we consider the Riemann problem (9.3.21), (9.3.20) and write

$$\mathbf{v}_L - \mathbf{v}_R = \sum_{j=1}^{K} \alpha_j \mathbf{r}_j,$$

the problem will be equivalent to Riemann problem (9.3.22)–(9.3.23) where $\mathbf{V}_L = [\alpha_1 \ \cdots \ \alpha_K]^T$ and $\mathbf{V}_R = \boldsymbol{\theta}$. The solution to Riemann problem (9.3.21), (9.3.20) can then be written as

$$\mathbf{v}(x,t) = \mathbf{v}_L - \sum_{x/t > \nu_j} \alpha_j \mathbf{r}_j \qquad (9.3.34)$$

$$= \mathbf{v}_R + \sum_{x/t < \nu_j} \alpha_j \mathbf{r}_j. \qquad (9.3.35)$$

Before we proceed, we emphasize that the solution to general K-systems Riemann problems will behave in a similar manner to the linear K-system Riemann problem. The solution will consist of $K-1$ constant states connecting $\mathbf{v}_L$ to $\mathbf{v}_R$. The two major differences are that (1) more interesting phenomena will appear between consecutive states (shocks, fans and contact discontinuities) and (2) the discontinuities due to shocks will not propagate along a characteristic curve (while the discontinuities due to contact discontinuities will propagate along a characteristic curve). The linear K-system Riemann problem can be considered to be a Riemann problem that has a contact discontinuity across every characteristic curve.

Shock Tube Problem We will begin by describing the solution to the shock tube problem given in HW0.0.3. It is possible to use the various

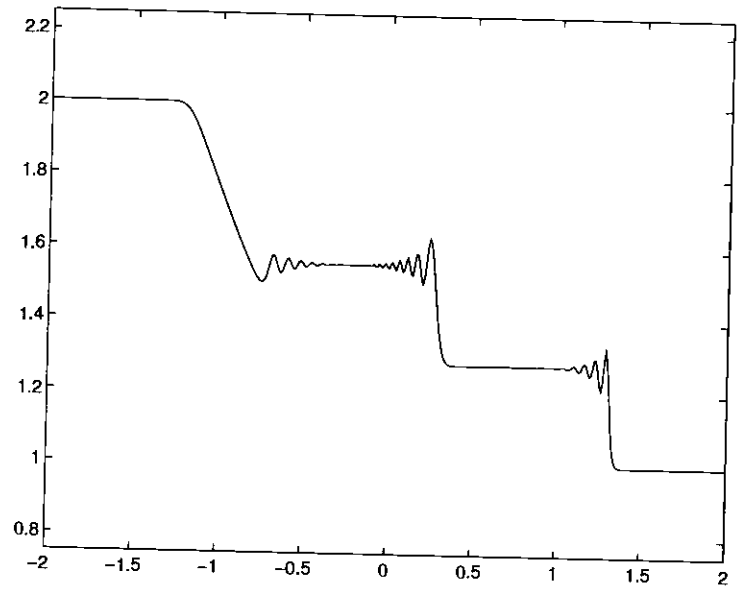

FIGURE 9.3.4. Plot of the density associated with the solution to problem HW0.0.3 at time $t = 1.0$. The solution was found using a linearized Lax-Wendroff scheme, Section 6.10.2, $\Delta x = 0.01$ and $\Delta t = 0.0025$.

results proved in refs. [37] and [62] to solve HW0.0.3 analytically. We will not do that. We hope that in Section 6.10.2 the reader used a rather naive extension of the Lax-Wendroff scheme to solve this problem and found, even though our solution had some unwanted oscillations (at least at the time we hopefully suspected that they were unwanted), we could see the basic form of the solution. In Figures 9.3.4, 9.3.5 and 9.3.6, we show plots of the solution to the shock tube problem at time $t = 1.0$ obtained by using the linearized Lax-Wendroff scheme. We do not claim that this is a sufficiently accurate solution. In fact, if these were sufficiently accurate solutions, we might not be writing Chapter 9. (We note that we have provided plots of ρ and v, and a plot of p instead of E because E is such an unintuitive variable.) However, the plots are of sufficient quality to show the following.

- On the left edge of the plot we still have the given initial values, $\rho_L = 2.0$, $v_L = 0.0$, $p_L = 2.0$ or $\mathrm{v}_L = [\rho \; m \; E]^T = [2.0 \quad 0.0 \quad 5.0]^T$ where $m = \rho v$.

- In the vicinity of $x = -1.0$ each of the variables has a fan.

- Near $x = 0.25$, the density has a jump discontinuity (which looks like a shock but will be a contact discontinuity), while both v and p appear to be continuous.

- Near $x = 1.25$, all three variables have jump discontinuities (which look like shocks, and will be shocks).
- To the right of the last discontinuity, the variables are as given initially, i.e., $\rho_R = 1.0$, $v_R = 0.0$, $p_R = 1.0$ or $\mathbf{v}_R = [1.0 \quad 0.0 \quad 2.5]^T$.

We refer to the constant states from (approximately) -2.0 to -1.25, -0.6 to 0.3, 0.3 to 1.3, and 1.3 to 2.0 as $\mathbf{v}_L$, $\mathbf{v}_1$, $\mathbf{v}_2$, and $\mathbf{v}_R$, respectively. From the plots, we see that **approximately** $\mathbf{v}_1 = [1.55 \quad 0.45 \quad 3.57]^T$ (with $v_1 = 0.29$ and $p_1 = 1.40$) and $\mathbf{v}_2 = [1.28 \quad 0.37 \quad 3.56]^T$ (with $v_2 = 0.29$ and $p_2 = 1.40$).

Since the system is a 3-system and we have a combination of three fans and discontinuities, it is clear that we are getting one fan or discontinuity associated with each of the characteristic curves. Earlier, in Section 6.10.2, we found that for the one dimensional Euler equation, $\mathbf{F}'(\mathbf{v})$ is given by

$$\begin{pmatrix} 0 & 1 & 0 \\ \frac{(\gamma-3)}{2}\frac{m^2}{\rho^2} & -(\gamma-3)\frac{m}{\rho} & \gamma-1 \\ -\frac{m}{\rho^2}\left[\gamma E-(\gamma-1)\frac{m^2}{\rho}\right] & \frac{1}{\rho}\left[\gamma E-\frac{3(\gamma-1)}{2}\frac{m^2}{\rho}\right] & \gamma\frac{m}{\rho} \end{pmatrix} \tag{9.3.36}$$

or

$$\begin{pmatrix} 0 & 1 & 0 \\ \frac{(\gamma-3)}{2}v^2 & -(\gamma-3)v & (\gamma-1) \\ \frac{(\gamma-1)}{2}v^3-\frac{v(E+p)}{\rho} & -(\gamma-1)v^2+\frac{(E+p)}{\rho} & \gamma v \end{pmatrix} \tag{9.3.37}$$

where $m = \rho v$. Also, we saw that the eigenvalues of $\mathbf{F}'(\mathbf{v})$ are given by

$$\nu_1(\mathbf{v}) = v - c < \nu_2(\mathbf{v}) = v < \nu_3(\mathbf{v}) = v + c$$

where c is the speed of sound,

$$c = \sqrt{\frac{\gamma p}{\rho}}.$$

The associated eigenvectors are given by

$$\mathbf{r}_1 = \begin{bmatrix} 1 \\ v-c \\ \frac{1}{2}v^2 - vc + \frac{1}{\gamma-1}c^2 \end{bmatrix}, \qquad \mathbf{r}_2 = \begin{bmatrix} 1 \\ v \\ \frac{1}{2}v^2 \end{bmatrix}$$

and

$$\mathbf{r}_3 = \begin{bmatrix} 1 \\ v+c \\ \frac{1}{2}v^2 + vc + \frac{1}{\gamma-1}c^2 \end{bmatrix}.$$

If we write ν_1, ν_2, ν_3, $\mathbf{r}_1$, $\mathbf{r}_2$ and $\mathbf{r}_3$ in terms of ρ, m and E, we see that

$$\mathbf{r}_1 \cdot \nabla\nu_1 = -\frac{\gamma+1}{2}\frac{c}{\rho}, \quad \mathbf{r}_2 \cdot \nabla\nu_2 = 0, \quad \mathbf{r}_3 \cdot \nabla\nu_3 = \frac{\gamma+1}{2}\frac{c}{\rho}.$$

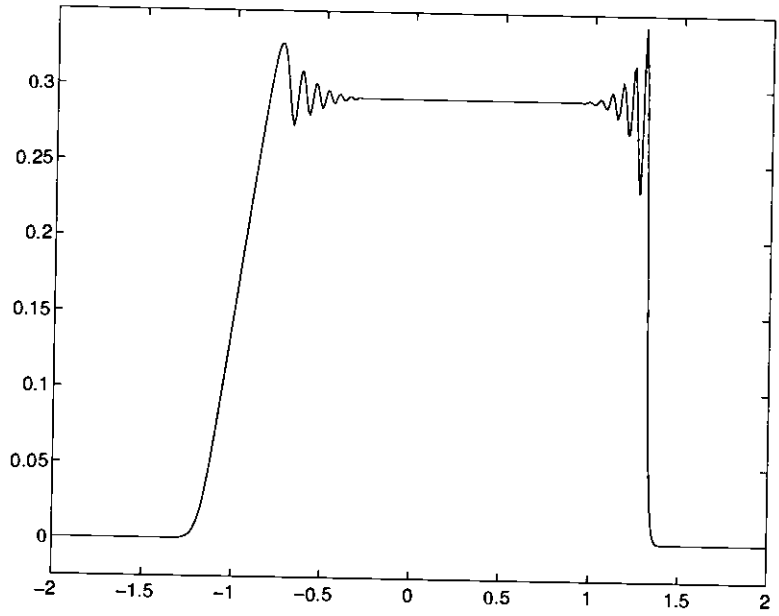

FIGURE 9.3.5. Plot of the velocity associated with the solution to problem HW0.0.3 at time $t = 1.0$. The solution was found using a linearized Lax-Wendroff scheme, Section 6.10.2, $\Delta x = 0.01$ and $\Delta t = 0.0025$.

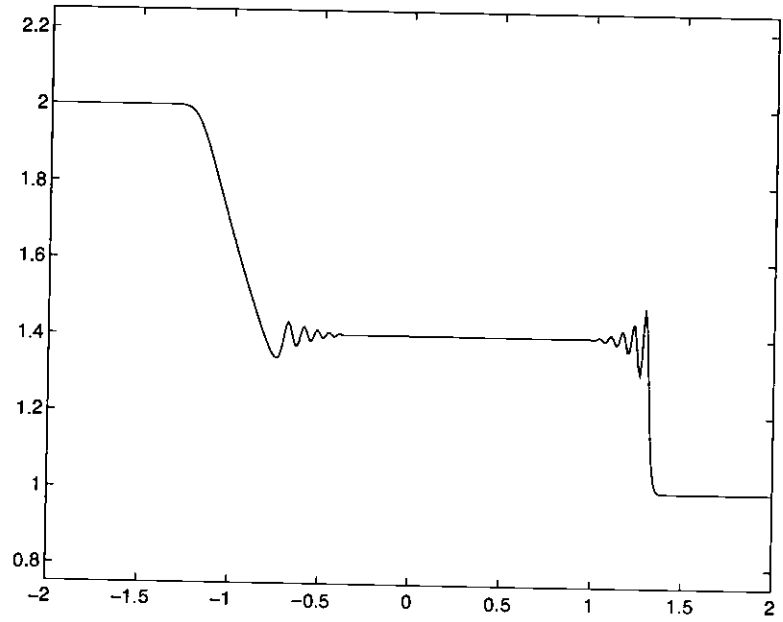

FIGURE 9.3.6. Plot of the pressure associated with the solution to problem HW0.0.3 at time $t = 1.0$. The solution was found using a linearized Lax-Wendroff scheme, Section 6.10.2, $\Delta x = 0.01$ and $\Delta t = 0.0025$.

We note that the second characteristic family is linearly degenerate for any $\mathbf{v}$. Also, since physically $c \neq 0$, the first and third characteristic families are genuinely nonlinear for any choice of $\mathbf{v}$. In our setting, the speed of sound in the four regions is given by

$$c_L = \sqrt{\frac{\gamma p_L}{\rho_L}} \approx 1.18, \quad c_1 = \sqrt{\frac{\gamma p_1}{\rho_1}} \approx 1.12, \quad c_2 = \sqrt{\frac{\gamma p_2}{\rho_2}} \approx 1.24,$$

and

$$c_R = \sqrt{\frac{\gamma p_R}{\rho_R}} \approx 1.18.$$

The eigenvalues associated with states $\mathbf{v}_L$ and $\mathbf{v}_R$ are

$$\nu_1(\mathbf{v}_L) = -\sqrt{1.4} \approx -1.18, \quad \nu_2(\mathbf{v}_L) = 0, \quad \nu_3(\mathbf{v}_L) = \sqrt{1.4} \approx 1.18$$

and

$$\nu_1(\mathbf{v}_R) = -\sqrt{1.4} \approx -1.18, \quad \nu_2(\mathbf{v}_R) = 0, \quad \nu_3(\mathbf{v}_R) = \sqrt{1.4} \approx 1.18,$$

respectively. The families of characteristics associated with states $\mathbf{v}_L$ and $\mathbf{v}_R$ are given by

$$x(t) = -1.18t + x_0, \quad x(t) = x_0, \quad x(t) = 1.18t + x_0$$

and

$$x(t) = -1.18t + x_0, \quad x(t) = x_0, \quad x(t) = 1.18t + x_0,$$

respectively. We would like also to be able to compute the characteristic curves associated with states $\mathbf{v}_1$ and $\mathbf{v}_2$. Generally, we can not a priori compute these curves, because we do not know $\mathbf{v}_1$ and $\mathbf{v}_2$. Because we are trying to describe why our solution looks as it does rather than analytically find the solution, we use the approximate values for $\mathbf{v}_1$ and $\mathbf{v}_2$ found from the plots of the solutions to compute $\nu_j(\mathbf{v}_1)$ and $\nu_j(\mathbf{v}_2)$, $j = 1, \ldots, 3$, and the associated characteristic curves. The (approximate) eigenvalues associated with $\mathbf{v}_1$ and $\mathbf{v}_2$ are given by

$$\nu_1(\mathbf{v}_1) = v_1 - c_1 \approx -0.83, \quad \nu_2(\mathbf{v}_1) = v_1 \approx 0.29, \quad \nu_3(\mathbf{v}_1) = v_1 + c_1 \approx 1.41$$

(where v_1, c_1, etc. denote the values of v, c, etc. in state $\mathbf{v}_1$) and

$$\nu_1(\mathbf{v}_2) \approx -0.95, \quad \nu_2(\mathbf{v}_2) \approx 0.29, \quad \nu_3(\mathbf{v}_2) \approx 1.53.$$

Further, the (approximate) families of characteristic curves associated with $\mathbf{v}_1$ and $\mathbf{v}_2$ are given by

$$x(t) = -0.83t + x_0, \quad x(t) = 0.29t + x_0, \quad x(t) = 1.41t + x_0$$

and

$$x(t) = -0.95t + x_0 =, \quad x(t) = 0.29t + x_0, \quad x(t) = 1.53t + x_0,$$

respectively.

To describe the solution to a problem such as this one, we first use the characteristics associated with the states $\mathbf{v}_L$ and $\mathbf{v}_R$ to show that the solution away from the origin ("away from the origin" being defined very liberally) is equal to $\mathbf{v}_L$ on the left and $\mathbf{v}_R$ on the right. *We want to understand the transition of the solution from state* $\mathbf{v}_L$ *to state* $\mathbf{v}_R$. *Understanding this transition involves understanding the interaction of the discontinuity in the initial condition* $\mathbf{v}_0$ *with the conservation law for* $t > 0$.

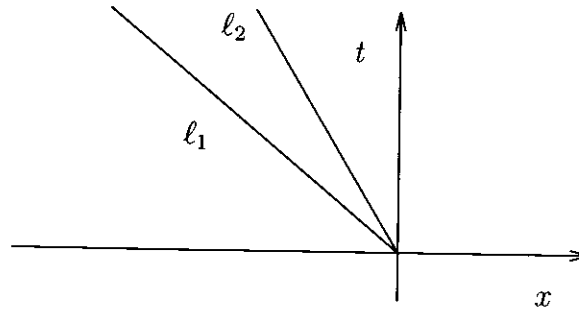

FIGURE 9.3.7. Characteristics due to the smallest eigenvalue ν_1 associated with states $\mathbf{v}_L$ and $\mathbf{v}_1$. The wedge depicted must be "filled in" with a fan of characteristics.

We begin by considering the first characteristic family. Since $\nu_1(\mathbf{v}_L) = -1.18 < \nu_1(\mathbf{v}_1) = -0.83$, the state $\mathbf{v}_L$ is connected to state $\mathbf{v}_1$ by a rarefaction wave or fan in each of the three variables. See Remark 5, page 118. In Figure 9.3.7, we see that there is an approximation of a wedge defined by the characteristic curves

$$\ell_1 : \; x(t) = \nu_1(\mathbf{v}_L)t = -1.18t \quad \text{and } \ell_2 : \; x(t) = \nu_1(\mathbf{v}_1)t = -0.83t$$

that must be filled in with a fan of characteristics (very much like the analogous case for scalar conservation laws). We note that at time $t = 1.0$, the fan would go from $x = -1.18$ to $x = -0.83$. It is a little difficult to determine these values accurately from the plots give in Figures 9.3.4–9.3.6, but the fans clearly spread from approximately $x = -1.2$ to $x = -0.8$. When the solution has evolved from the values in state $\mathbf{v}_L$ to those in state $\mathbf{v}_1$, we say that **state $\mathbf{v}_L$ is connected to state $\mathbf{v}_1$** through the fans plotted in Figures 9.3.4–9.3.6.

After the $\mathbf{v}_L$ state has connected with the $\mathbf{v}_1$ state, we must turn to the second family of characteristics. Since the second family of characteristics is linearly degenerate, the second discontinuity will be a contact discontinuity. Because we have a contact discontinuity, we know that the speed of propagation of the discontinuity is equal to

$$s_2 = \nu_2(\mathbf{v}_1) = \nu_2(\mathbf{v}_2)$$

(of both which equal 0.29). See Remark 2, page 117. For this problem, since $\nu_2(\mathbf{v}) = v$ and $\nu_2(\mathbf{v}_2) = \nu_2(\mathbf{v}_1)$, we find that v is constant across the curve $x(t) = \nu_2(\mathbf{v}_1)t = 0.29t$. Of course, we can see in Figure 9.3.5 that this is the case. Then, using the third jump condition across the contact discontinuity (the third component of the jump condition $\mathbf{F}(\mathbf{v}_2) - \mathbf{F}(\mathbf{v}_1) = s_2(\mathbf{v}_2 - \mathbf{v}_1)$),

$$v_2(E_2 + p_2) - v_1(E_1 + p_1) = s_2(E_2 - E_1)$$

along with the facts that $v_1 = v_2 \neq 0$ and $s_2 = v_1 = v_2$, we see that $p_1 = p_2$. This can be seen in Figure 9.3.6. Thus we see that state $\mathbf{v}_1$ is connected to $\mathbf{v}_2$ through a contact discontinuity where v and p are constant across this contact discontinuity, there is a jump in ρ across this contact discontinuity, and the discontinuity propagates along the characteristic curve $x(t) = 0.29t$ with speed of propagation 0.29. As before, if we consider the situation at time $t = 1.0$, this discontinuity should be at $x = 0.29$. In Figure 9.3.4, we see that the contact discontinuity occurs approximately at $x = 0.29$.

To show that p was constant across the contact discontinuity, we used the third jump condition. The first and second jump conditions must also be satisfied. The first and second jump conditions across the ν_2 contact discontinuity are given by

$$\rho_2 v_2 - \rho_1 v_1 = s_2(\rho_2 - \rho_1)$$

and

$$(\rho_2 v_2^2 + p_2) - (\rho_1 v_1^2 + p_1) = s_2(\rho_2 v_2 - \rho_1 v_1),$$

respectively. Since $v_1 = v_2 = s_2$ and $p_1 = p_2$, both of these jump conditions are trivially satisfied.

And finally, we now turn to the ν_3 characteristic. The transition of the solution through this discontinuity must account for the connection from $\mathbf{v}_2$ to $\mathbf{v}_R$ (because we must get to $\mathbf{v}_R$ and we have no more characteristics over which it might happen). Since

$$\nu_3(\mathbf{v}_2) = 1.53 > \nu_3(\mathbf{v}_R) = 1.18,$$

we potentially have a 3-shock (we must still satisfy inequality (9.3.12)). Through this 3-shock, the state $\mathbf{v}_2$ is connected to state $\mathbf{v}_R$ where the solution has jump discontinuities in each of the three variables. We use the first jump condition (the first component of the jump condition $\mathbf{F}(\mathbf{v}_R) - \mathbf{F}(\mathbf{v}_2) = s_3(\mathbf{v}_R - \mathbf{v}_2)$)

$$\rho_R v_R - \rho_2 v_2 = s_3(\rho_R - \rho_2)$$

along with the given values of $\mathbf{v}_R$ and ρ_R and the approximate values of $\mathbf{v}_2$ and ρ_2 to approximately determine

$$s_3 = \frac{\rho_R v_R - \rho_2 v_2}{\rho_R - \rho_2} = 1.39.$$

We note that s_3 and $\nu(\mathbf{v}_2)$ satisfy the second inequality in the definition of Entropy Condition I_v, (9.3.12),

$$\nu_2(\mathbf{v}_2) = 0.29 < s_3 = 1.39.$$

The shock associated with ν_3 should propagate along the curve $x(t) = s_3 t = 1.39t$. We note in Figures 9.3.4–9.3.6 that the shocks occur at approximately $x = 1.4$.

If we consider the last two jump conditions across this discontinuity, we see that

$$(\rho_R v_R^2 + p_R) - (\rho_2 v_2^2 + p_2) - s_3(\rho_R v_R - \rho_2 v_2) = 0.0043$$

and

$$v_R(E_R + p_R) - v_2(E_2 + p_2) - s_3(E_R - E_2) = -0.034.$$

It seems fairly clear that both the second and third jump conditions are being satisfied (approximately).

Remark: If one looks at Figure 9.3.4 or 9.3.5, one might get the view that the variable will cascade downward from the value given on the left to the value given on the right. However, this is not required. We are not guaranteed that the value of any of the variables given on one side of the Riemann initial condition will change monotonically to the value given on the other side. In fact, if we consider Figure 9.3.6, we see that the velocity component is initially equal on both sides of the origin, increases across the fan and then decreases across the shock. This type of behavior is always a possibility.

K-System Conservation Laws For the shock tube problem we hope that we now have some idea of what the solution should look like and why it looks that way. The good news is that a general solution to a Riemann problem for a K-system conservation law will look similar to the solution of the shock tube problem. There will be fans or rarefactions, shocks and contact discontinuities (which look the same as shocks). Maybe the best theorem describing the situation is the following theorem from ref. [62], page 335.

Theorem 9.3.3 *Suppose the K-system conservation law (9.1.1) is hyperbolic and that each characteristic field is genuinely nonlinear or linearly degenerate in some neighborhood, N_L, of $\mathbf{v}_L$. Then there is a neighborhood $N \subset N_L$ of $\mathbf{v}_L$ such that if $\mathbf{v}_R \in N$, the unique solution $\mathbf{v}$ to problem (9.1.1), (9.3.20) consists of at most $K + 1$ constant states separated by shocks, fans or contact discontinuities.*

Remark 1: The hypothesis requiring that there be a neighborhood N_L such that each characteristic field of the system is either genuinely nonlinear or linearly degenerate in N_L makes the theorem appear more difficult, but is

very nice in that it does not require that the system satisfy this hypothesis everywhere. We should note that since c and ρ are physically nonzero, for system (0.0.7)–(0.0.9) considered in the shock tube problem, the first and the last characteristic fields are genuinely nonlinear and the second characteristic field is linearly degenerate for all $\mathbf{v}$. In general, a field may be genuinely nonlinear in one region of values and linearly degenerate in another region of values.

Remark 2: We also note that the requirement of the existence of the neighborhood N such that $\mathbf{v}_L$ and $\mathbf{v}_R$ are in N is a very precise way of saying that $\mathbf{v}_L$ and $\mathbf{v}_R$ are sufficiently close. Though there are some results that do not require such a hypothesis, most "nice" results do require that $\mathbf{v}_L - \mathbf{v}_R$ be small (where the definition of "small" is generally "small enough to make the proof work."

Remark 3: The computation for K-systems analogous to that done in Example 9.2.8 for scalar conservation laws is as follows. If $\mathbf{v}(x,t) = \psi(x/t)$, then

$$\begin{aligned}\boldsymbol{\theta} &= \mathbf{v}_t + \mathbf{F}'(\mathbf{v})\mathbf{v}_x \\ &= -\frac{x}{t^2}\psi'(x/t) + \mathbf{F}'(\psi(x/t))\frac{1}{t}\psi'(x/t) \\ &= \Big[-\frac{x}{t}I_K + \mathbf{F}'(\psi(x/t))\Big]\frac{1}{t}\psi'(x/t).\end{aligned}$$

Thus if $\mathbf{F}'$ is invertible, $\psi(x/t) = \left(\mathbf{F}'\right)^{-1}\left(\frac{x}{t}I_K\right)$ defines a similarity solution to conservation law (9.1.1).

We next try to describe what the effect of the above theorem on our solutions will be. Solutions to Riemann problems for K-system conservation laws will satisfy the following.

- The $K+1$ states referred to in Theorem 9.3.3 will consist of the $\mathbf{v}_L$ state on the left, the $\mathbf{v}_R$ state on the right, and $K-1$ intermediate states ($\mathbf{v}_1$ and $\mathbf{v}_2$ in the shock tube problem). In accord with the theorem, these states will be separated by shocks, fans and contact discontinuities.

- If associated with a given characteristic field we obtain a fan, we will have a fan or nothing (which could be considered as a very smooth fan) in each of the coordinates of the solution.

  - Having a fan in each of the coordinates of the solution might be thought of as the generic situation. If the fan is due to the k-th characteristic field and a given coordinate of the solution does not depend on the k-th eigenvector of the operator, then there will not be a fan in that coordinate of the solution.

- If associated with a given characteristic field we are to have a shock, we will have a jump discontinuity or nothing (which could be considered as a very small jump) in each of the coordinates of the solution.
  - Again, the generic case is that if there is a shock, each coordinate of the solution will have a jump discontinuity.
- Associated with a given characteristic field, the solution cannot have a fan in one coordinate and a jump discontinuity in another coordinate of the solution at the same time and for the same spatial values.
- When we have a fan associated with the k-th characteristic field, the fan will always be confined between $x = \nu_k(\mathbf{v}_k)t$ and $x = \nu_k(\mathbf{v}_{k+1})t$, where $\mathbf{v}_k$ and $\mathbf{v}_{k+1}$ are the states to the left and right of the fan and we are assuming that the discontinuity occurs at $x = 0$ and $t = 0$. The fans in all of the components of the solution must be confined to this same fan of characteristic directions.
- When we are observing the solutions, a contact discontinuity is recognizable, since the associated discontinuity will be propagating along a characteristic curve.
- The speed of propagation of a shock discontinuity must fall between the characteristic values associated with the states on either side of the shock (the Rankine-Hugoniot or jump conditions).

We now have some idea, we hope, of how a solution to the Riemann problem for a K-system conservation law should look. However, we must remember that we want solutions that satisfy an entropy condition (or are vanishing viscosity solutions). The vanishing viscosity solutions of Riemann problems for K-system conservation laws look like the solutions described above. But, *solutions to Riemann problems for K-system conservation laws that do not satisfy the entropy condition also generally look like the solutions described above. It is not generally possible to look at the form of a solution and decide whether or not it satisfies the entropy condition.*

9.4 Computational Interlude VI

We have been trying to solve either the viscous or inviscid Burgers' equation since Chapter 2, and we have been trying to solve the shock tube problem since Chapter 6. We hope that we have obtained some reasonable results. Also, it is hoped that the material contained in the early sections of this chapter might lead us to a better understanding of these problems.

We give the reader a mild apology for leading you on such a wild goose chase. We recommend that before one tackles a problem, one learn as much

about the problem as possible. Clearly, we did not do this on problems HW0.0.1–HW0.0.3. However, the purpose of these problems has been to help the reader become a competent numerical experimentalist. Hence, you get only a mild apology.

From the earlier sections of this chapter, we see that at least one of the pieces of information that we have not used in our previous computations is the fact that the equations in HW0.0.1, HW0.0.2 and HW0.0.3 are or can be put in conservation law form. We have seen that conservation can be very important when considering problems that contain shocks. If we consider conservation law (9.1.1) and return to some of the explicit schemes derived for numerically approximating the solutions to linear hyperbolic equations in Chapters 5 and 6, we can immediately derive the following explicit difference schemes.

$$\mathbf{u}_k^{n+1} = \mathbf{u}_k^n - R\left(\mathbf{F}_{k+1}^n - \mathbf{F}_k^n\right) \tag{9.4.1}$$

$$\mathbf{u}_k^{n+1} = \mathbf{u}_k^n - R\left(\mathbf{F}_k^n - \mathbf{F}_{k-1}^n\right) \tag{9.4.2}$$

$$\mathbf{u}_k^{n+1} = \frac{1}{2}(\mathbf{u}_{k-1}^n + \mathbf{u}_{k+1}^n) - \frac{R}{2}\left(\mathbf{F}_{k+1}^n - \mathbf{F}_{k-1}^n\right) \tag{9.4.3}$$

$$\mathbf{u}_k^{n+1} = \mathbf{u}_k^{n-1} - R\left(\mathbf{F}_{k+1}^n - \mathbf{F}_{k-1}^n\right) \tag{9.4.4}$$

where $R = \Delta t/\Delta x$. We will refer to these schemes as the FTFS scheme, FTBS scheme, the Lax-Friedrichs scheme and the leapfrog scheme, respectively. It should be reasonably clear that difference schemes (9.4.1)–(9.4.4) take advantage of the fact that we are approximating the solution of a conservation law by differencing the term $\mathbf{F}_x$ rather than differentiating the $\mathbf{F}_x$ term and then differencing it. However, there is more to it than that. All of the above methods could be derived using the conservation approach used in Sections 1.6, 4.2.2 and 5.8.1. More specifically, in Section 9.6 we formally describe conservative difference methods, and difference schemes (9.4.1)–(9.4.3) will be considered as examples. We do not consider multilevel conservative schemes, but the same technique that we use for two-level schemes would work also for multilevel schemes in general and leapfrog in particular.

The methods (9.4.1)–(9.4.4) could be tried on either HW0.0.2 or HW0.0.3. It should not surprise us that these methods might have all of the warts that we associated with their linear counterparts. One of the methods whose linearized version was mildly successful when used in Section 5.9.4 was the Lax-Wendroff scheme. Following the approach used to derive the Lax-Wendroff scheme in Section 5.3.3, we begin by expanding $\mathbf{u}_k^{n+1}$ in a Taylor expansion about $n\Delta t$ (with k fixed)

$$\mathbf{u}_k^{n+1} = \mathbf{u}_k^n + (\mathbf{u}_t)_k^n \Delta t + (\mathbf{u}_{tt})_k^n \frac{\Delta t^2}{2} + \mathcal{O}(\Delta t^3). \tag{9.4.5}$$

We know that we can replace $\mathbf{u}_t$ in equation (9.4.5) by $-\mathbf{F}_x$, i.e. we get

$$\begin{aligned}\mathbf{u}_k^{n+1} &= \mathbf{u}_k^n - (\mathbf{F}_x)_k^n \Delta t + (\mathbf{u}_{tt})_k^n \frac{\Delta t^2}{2} + \mathcal{O}(\Delta t^3) \\ &= \mathbf{u}_k^n - \frac{\Delta t}{2\Delta x}\left(\mathbf{F}_{k+1}^n - \mathbf{F}_{k-1}^n\right) + \mathcal{O}(\Delta x^2) \\ &\quad + (\mathbf{u}_{tt})_k^n \frac{\Delta t^2}{2} + \mathcal{O}(\Delta t^3). \end{aligned} \tag{9.4.6}$$

As with the linear case, we note that

$$\begin{aligned}\mathbf{u}_{tt} &= -(\mathbf{F}_x)_t \\ &= -(\mathbf{F}_t)_x \\ &= -(\mathbf{F}'(\mathbf{u})\mathbf{u}_t)_x \\ &= -(-\mathbf{F}'(\mathbf{u})\mathbf{F}_x)_x\,. \end{aligned} \tag{9.4.7}$$

If we then approximate the term in (9.4.7) by

$$\begin{aligned}[(\mathbf{F}'(\mathbf{u})\mathbf{F}_x)_x]_k^n &\approx \frac{1}{\Delta x}\left[(\mathbf{F}'(\mathbf{u})\mathbf{F}_x)_{k+1/2}^n - (\mathbf{F}'(\mathbf{u})\mathbf{F}_x)_{k-1/2}^n\right] \\ &\approx \frac{1}{\Delta x}\Big[(\mathbf{F}'(\mathbf{u}))_{k+1/2}^n \frac{1}{\Delta x}\left(\mathbf{F}_{k+1}^n - \mathbf{F}_k^n\right) \\ &\qquad - (\mathbf{F}'(\mathbf{u}))_{k-1/2}^n \frac{1}{\Delta x}\left(\mathbf{F}_k^n - \mathbf{F}_{k-1}^n\right)\Big] \\ &= \frac{1}{\Delta x^2}\Big[(\mathbf{F}'(\mathbf{u}))_{k+1/2}^n \left(\mathbf{F}_{k+1}^n - \mathbf{F}_k^n\right) \\ &\qquad - (\mathbf{F}'(\mathbf{u}))_{k-1/2}^n \left(\mathbf{F}_k^n - \mathbf{F}_{k-1}^n\right)\Big], \end{aligned}$$

equation (9.4.6) becomes

$$\begin{aligned}\mathbf{u}_k^{n+1} = \mathbf{u}_k^n - \frac{\Delta t}{2\Delta x}\left(\mathbf{F}_{k+1}^n - \mathbf{F}_{k-1}^n\right) + \frac{\Delta t^2}{2\Delta x^2}\Big[&(\mathbf{F}'(\mathbf{u}))_{k+1/2}^n \left(\mathbf{F}_{k+1}^n - \mathbf{F}_k^n\right) \\ &- (\mathbf{F}'(\mathbf{u}))_{k-1/2}^n \left(\mathbf{F}_k^n - \mathbf{F}_{k-1}^n\right)\Big] + \mathcal{O}(\Delta t^3) + \mathcal{O}(\Delta x^2).\end{aligned}$$

Thus we are left with the Lax-Wendroff scheme that takes into account the fact that we are solving a conservation law.

$$\begin{aligned}\mathbf{u}_k^{n+1} = \mathbf{u}_k^n - \frac{R}{2}\left(\mathbf{F}_{k+1}^n - \mathbf{F}_{k-1}^n\right) + \frac{R^2}{2}\Big[&(\mathbf{F}'(\mathbf{u}))_{k+1/2}^n \left(\mathbf{F}_{k+1}^n - \mathbf{F}_k^n\right) \\ &- (\mathbf{F}'(\mathbf{u}))_{k-1/2}^n \left(\mathbf{F}_k^n - \mathbf{F}_{k-1}^n\right)\Big]\end{aligned} \tag{9.4.8}$$

where $R = \Delta t/\Delta x$. Since we do not have $\mathbf{u}_j^n$ at $j = k \pm \frac{1}{2}$, we have no way of computing $(\mathbf{F}'(\mathbf{u}))_{k\pm 1/2}^n$. We approximate these terms by

$$(\mathbf{F}'(\mathbf{u}))_{k+1/2}^n \approx \mathbf{F}'\Big(\frac{1}{2}(\mathbf{u}_k^n + \mathbf{u}_{k+1}^n)\Big)$$

and

$$(\mathbf{F}'(\mathbf{u}))^n_{k-1/2} \approx \mathbf{F}'\Big(\frac{1}{2}(\mathbf{u}^n_{k-1} + \mathbf{u}^n_k)\Big).$$

One of the difficulties of using the Lax-Wendroff scheme is the fact that we must evaluate the derivatives and compute these derivatives at each time step. For large problems these extra function evaluations can be expensive. (Remember that $\mathbf{F}'$ is a $K \times K$ matrix.) Two schemes designed to circumvent this problem are the Richtmyer two-step scheme and the MacCormack scheme, which are as follows.

Richtmyer Two-Step Scheme

$$\mathbf{u}^{n+1/2}_{k+1/2} = \frac{1}{2}\big[\mathbf{u}^n_k + \mathbf{u}^n_{k+1}\big] - \frac{R}{2}\delta_+\mathbf{F}^n_k \tag{9.4.9}$$

$$\mathbf{u}^{n+1}_k = \mathbf{u}^n_k - R\delta_-\mathbf{F}^{n+1/2}_{k+1/2} \tag{9.4.10}$$

MacCormack Scheme

$$\mathbf{u}^*_k = \mathbf{u}^n_k - R\delta_+\mathbf{F}^n_k \tag{9.4.11}$$

$$\mathbf{u}^{n+1}_k = \frac{1}{2}\big[\mathbf{u}^n_k + \mathbf{u}^*_k\big] - \frac{R}{2}\delta_-\mathbf{F}^*_k. \tag{9.4.12}$$

Both of the above schemes are Lax-Wendroff-like in that if we apply these schemes to a linear problem, they both reduce to the Lax-Wendroff scheme (which is why we have not seen these schemes earlier). See HW9.4.5. True to the notation, the first step in the Richtmyer two-step scheme is a true half time step. The $*$ notation is used in the MacCormack scheme because the first step is a full time step calculation (using the FTFS scheme, which we should suspect will have some unacceptable limitations). Because the first steps of both schemes are partial differential equation-like, it is relatively easy to assign boundary conditions to $\mathbf{u}^{n+1/2}$ and $\mathbf{u}^*$. In addition, in HW9.4.2 we see that both of these schemes are second order accurate.

By switching from the Lax-Wendroff scheme to either the Richtmyer two-step scheme or MacCormack's scheme, we go from $3K + 2K!$ ($\mathbf{F}'$ is symmetric) to $4K$ function evaluations (in all cases the need to perform all of the function evaluations can be eliminated by saving some of the function values from the previous grid point).

Another Lax-Wendroff-like scheme that we will use later is the **Beam-Warming scheme**. We were introduced to the linear version of the Beam-Warming scheme in Table 5.3.1. The Beam-Warming scheme,

$$\mathbf{u}^*_k = \mathbf{u}^n_k - R\delta_-\mathbf{F}^n_k \tag{9.4.13}$$

$$\mathbf{u}^{n+1}_k = \frac{1}{2}\big[\mathbf{u}^n_k + \mathbf{u}^*_k\big] - \frac{R}{2}\delta_-\mathbf{F}^*_k - \frac{R}{2}(\delta_-)^2\mathbf{F}^n_k, \tag{9.4.14}$$

can be described as a MacCormack scheme in which we difference both times in the backward direction (and include the $-\frac{R}{2}(\delta_-)^2\mathbf{F}^n_k$ term to make

the scheme second order accurate. The Beam-Warming scheme is Lax-Wendroff-like in that it can be derived by following the approach of the derivation of the Lax-Wendroff scheme, using one sided differences at all times.

And finally, we recall that in Section 6.10.1 we suggested a method for approximating the solution to problems HW0.0.1 and HW0.0.2 (and gave some of the results in Section 7.9.1) that switched between using forward and backward differencing on the vv_x term depending on whether u_k^n was negative or positive. When we considered what the solutions to HW0.0.1 and HW0.0.2 looked like, it was almost cheating to use such a scheme. It appeared that the scheme was groomed especially for that problem. However, there is a general version of that scheme that is referred to as the **upwind scheme** (because it always looks upwind), and for scalar conservation law (9.2.1) can be written as

$$u_k^{n+1} = \begin{cases} u_k^n - R\delta_+ F_k^n & \text{if } F'(u_k^n) \le 0 \\ u_k^n - R\delta_- F_k^n & \text{if } F'(u_k^n) > 0 \end{cases} \tag{9.4.15}$$

or

$$u_k^{n+1} = \begin{cases} u_k^n - R\delta_+ F_k^n & \text{if } a_{k+1/2}^n \le 0 \\ u_k^n - R\delta_- F_k^n & \text{if } a_{k+1/2}^n > 0 \end{cases} \tag{9.4.16}$$

where

$$a_{k+1/2}^n = \begin{cases} \delta_+ F_k^n / \delta_+ u_k^n & \text{if } \delta_+ u_k^n \ne 0 \\ F'(u_k^n) & \text{if } \delta_+ u_k^n = 0. \end{cases} \tag{9.4.17}$$

We should notice that the two schemes choose the nonlinear FTFS scheme or FTBS scheme depending on whether F' (or in the case of (9.4.16), $a_{k+1/2}^n$) is less than or greater than zero. The difference between schemes (9.4.15) and (9.4.16) is obviously that difference scheme (9.4.16) uses a discrete approximation to the derivative if possible. We mention here that based on the computations done in HW9.4.4 and HW9.4.5, in Section 9.6.2 we note that difference scheme (9.4.15) is not conservative, and in Section 9.6.5 we note that difference scheme (9.4.16) does not approximate the vanishing viscosity solutions. These are important negative aspects of difference schemes (9.4.15) and (9.4.16), and for these reasons, we will hardly ever use these schemes in the form given. However, these schemes with a slight adjustment will be an important tool that we will use for developing high resolution schemes. We might add that there is no problem with these schemes when the conservation law is linear.

It should also be clear that it is not easy to build an upwind scheme for systems. In Section 6.2.1 we developed a flux splitting scheme for a linear system of partial differential equations that was actually an upwind scheme for a linear system. However, the approach used does not work for

nonlinear K-systems, since it is impossible to uncouple the equations as we did in the linear case. The matrices S and S^{-1} will generally depend on $\mathbf{v}$, so we will not be able to take the differentiation inside and outside of the matrix multiplications.

Of course, it is of interest to us to know whether taking advantage of the fact that Burgers' equation can be put into conservation form and that the Euler equations were given to us in conservation form will help us with the numerical solution of these problems. Thus, we suggest that you apply the Lax-Wendroff scheme (9.4.8) to both HW0.0.2 and HW0.0.3. These solutions should be compared and contrasted with the results using the linearized (nonconservative) form of the Lax-Wendroff scheme.

HW 9.4.1 Show that for a linear, hyperbolic K-system of partial differential equations both the Richtmyer two-step scheme, (9.4.9)–(9.4.10), and MacCormack scheme, (9.4.11)–(9.4.12), reduce to the Lax-Wendroff scheme.

HW 9.4.2 Show that the Richtmyer two-step scheme, the MacCormack scheme and the Beam-Warming scheme are second order accurate.

HW 9.4.3 (a) Use the upwind scheme (9.4.16) to approximate the solution to the following initial–boundary–value problem.

$$v_t + \left(\frac{v^2}{2}\right)_x = 0, \quad x \in (-2,2),\ t > 0 \tag{9.4.18}$$
$$v(x,0) = v_0(x) = \begin{cases} 2 & x \le 0 \\ -1 & x > 0 \end{cases}$$
$$v(-2,t) = 2, \qquad v(2,t) = -1$$

See Remark, page 142.

Use $\Delta x = 0.02$, $\Delta t = 0.01$ and plot the solution at time $t = 1.0$.
(b) Repeat part (a) using difference scheme (9.4.15).

HW 9.4.4 (a) Use upwind scheme (9.4.15) to approximate the solution to the inviscid Burgers' equation (9.4.18) with initial condition

$$v(x,0) = v_0(x) = \begin{cases} -1 & x \le 0 \\ 2 & x > 0 \end{cases}$$

on the region $[-2,2]$ with numerical boundary conditions $u_0^n = -1$ and $u_M^n = 2$. See Remark, page 142.

HW 9.4.5 Use upwind scheme (9.4.16) to approximate the solution to the inviscid Burgers' equation (9.4.18) with initial condition

$$v(x,0) = v_0(x) = \begin{cases} -1 & x \leq 0 \\ 2 & x > 0 \end{cases}$$

on the region $[-2,2]$ with numerical boundary conditions $u_0^n = -1$ and $u_M^n = 2$. See Remark, page 142.

9.5 Numerical Solution of Conservation Laws

9.5.1 Introduction

We are now ready to begin our study of the numerical approximation of solutions to conservation laws. Of course, we have previously tried to solve HW0.0.1–HW0.0.3 numerically and were somewhat successful, but that was what we now refer to as our naive approach. In the rest of this chapter we will systematically study numerical schemes for conservation laws that use the analytic and qualitative information that we now have about conservation laws. We might warn you that our goal is not to make you an expert in the numerical solution of conservation laws. (For that you should consult refs. [37] or [11].) We will instead try to make this chapter into an introduction to being an expert in the numerical solution of conservation laws. We will show you the different problems one faces when trying to approximate solutions to conservation laws numerically, introduce you to some of the schemes, and give you some of the relevant definitions and theorems (we will give you proofs of some of the theorems, formal proofs of some of the theorems and some theorems without proof). We will have you apply some of the schemes to some of our favorite problems and, we hope to help you to evaluate the good and bad results that you and other researchers get when numerically solving conservation laws.

We emphasize that the numerical solution of conservation laws will include resolving shocks, contact discontinuities and fans; accurately approximating the speed of propagation of shocks, contact discontinuities and fans; and selecting the entropy (vanishing viscosity) solution from the many weak solutions. Of course, an obvious requirement will be that we use stable, consistent schemes. And finally, we will also still be interested in the dissipation and dispersion contained in the scheme. Some of the properties of linear schemes that we studied so extensively in Part 1 will not be easy to derive, understand and/or apply when we deal with conservation laws (such as studying the dissipation and dispersion contained in the scheme) and some the of the methods used for linear schemes will not be available at all (such as the use of the discrete Fourier transform, which depends heavily on the

linearity). There will be times when we are not able to analyze certain aspects of a nonlinear scheme. Sometimes the best that we will be able to do is to consider the linearization of the conservation law or numerical scheme that we have and use linear methods to obtain some approximate results. Often, these "linearized results" will be good results. Sometimes, these "linearized results" will mislead us. Such "linearized results" should be used only along with a series of carefully designed numerical experiments.

When we try to approximate the solutions to problems involving conservation laws, we will be faced with providing and handling boundary conditions. As we have done so often before, when we are doing any analysis we will often ignore the boundary conditions and consider pure initial–value problems. When we want to compute solutions, we will often use periodic boundary conditions. When it is necessary (or correct) to use Dirichlet boundary conditions, one faces all of the problems considered in Chapter 8, except that we now usually have a nonlinear partial differential equation and a nonlinear difference scheme. The most common approach when considering Dirichlet boundary conditions and any numerical boundary conditions that are necessary is to use the GKSO theory on a linearized model problem to choose the boundary conditions that may be stable for the nonlinear problem. The approach may not be perfect, but there are many times when this is the best that we can do.

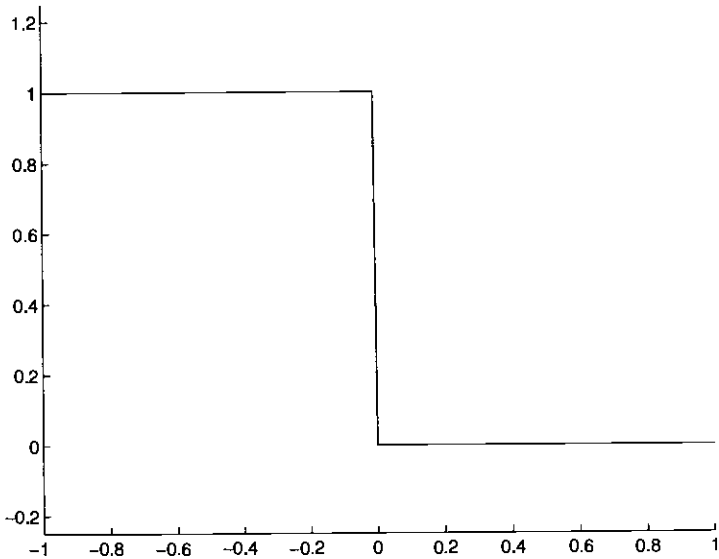

FIGURE 9.5.1. Approximate solution to the inviscid Burgers' equation (9.5.2) with initial condition (9.5.3). The solution was computed using difference scheme (9.5.1), $\Delta x = 0.01$, $\Delta t = 0.005$, and the solution is given at time $t = 0.5$.

Before we proceed, let us consider several sample calculations.

Example 9.5.1 Use the difference scheme

$$u_k^{n+1} = u_k^n - Ru_k^n \delta_- u_k^n \tag{9.5.1}$$

to approximate the solution to the inviscid Burgers' equation

$$v_t + vv_x = 0, \ x \in (-1, 1), \ t > 0 \tag{9.5.2}$$

along with the initial condition

$$v_0(x) = v(x, 0) = \begin{cases} 1.0 & \text{if } x < 0 \\ 0.0 & \text{if } x \geq 0 \end{cases} \tag{9.5.3}$$

and boundary conditions $v(-1, t) = 1.0$ and $v(1, t) = 0.0$.

Solution: We note that difference scheme (9.5.1) is the logical extension of the FTBS scheme to the nonconservative, inviscid Burgers' equation where we construct an explicit scheme by evaluating all of the terms involving spatial derivatives at time t_n. Also, we recall that we used a very similar scheme in Section 2.6.2 to try to solve HW0.0.1.

We should also be aware that we must be somewhat careful using the above boundary conditions, but given the nature of the solution we obtain, the boundary conditions do not cause any problems. In our implementation of difference scheme (9.5.1), we use $\Delta x = 0.01$, $\Delta t = 0.005$ (so $R = \Delta t/\Delta x = 0.5$) and calculate the solution at $t = 0.5$. The numerical solution is given in Figure 9.5.1. The solution to difference scheme (9.5.1) along with the given initial condition and boundary conditions will be the same for all time. The solution to difference scheme (9.5.1), (9.5.3) along with the boundary conditions $v(-1, t) = 1.0$ and $v(1, t) = 0.0$ will converge to the function v_0 given in (9.5.3) as Δx and Δt approach zero.

We should note that $v(x, t) = v_0(x)$ is not a weak solution to problem (9.5.2), (9.5.3), $v(-1, t) = 1.0$, $v(1, t) = 0.0$. If $v(x, t) = v_0(x)$ is to be a solution to the problem, v must satisfy the R-H condition (the jump condition), i.e. across the discontinuity, $v(x, t) = v_0(x)$ must satisfy $s[v] = [F(v)]$. In this case $F(v) = v^2/2$, $[F(v)] = \left(v_L^2 - v_R^2\right)/2 = \frac{1}{2}$, $[v] = v_L - v_R = 1$ and $s = 0$. Obviously, the jump condition is not satisfied.

Remark: We have included two boundary conditions to go along with (9.5.2)–(9.5.3) in defining the above initial–boundary–value problem. We did not address this problem in the sections on analytic solutions because there we were able to consider true initial-value problems. In Section 5.5.2 we saw that for the one way wave equation, we get to assign only one boundary condition. When we considered initial–boundary–value problems for systems in Section 6.3.1.2, we found that we could assign as many boundary conditions at $x = 0$ as there were negative eigenvalues of A (when we wrote the partial differential equation as $\mathbf{v}_t = A\mathbf{v}_x$) and as many boundary conditions at $x = 1$ as there were positive eigenvalues of A. These results were true because the partial differential equations were linear. The results could be restated as follows. We can assign as many boundary conditions at a particular boundary as there are characteristic curves entering the region from that boundary. This latter description is true for nonlinear hyperbolic partial differential equations and is more useful.

In the above example, the above description allows us to assign one boundary condition at $x = -1.0$ because $v_L = 1.0$ and $F'(v_L) = v_L = 1$ at $x = -1.0$. In that case, $x = 1.0$ is a special case, not covered by the description above. If we return to Section 6.3.1.2, we see that we get to assign a boundary condition at $x = 1.0$ because $x = 1.0$ is a characteristic curve, and we can do anything we want on a characteristic curve.

Using the same criteria, we see that for Burgers' equation and the initial condition given in Example 9.5.2, because $v_L = -1.0$, $v_R = 1.0$, $F'(V_L) = -1.0$ and $F'(v_R) = 1.0$, we do not get to assign any analytic boundary conditions.

When we are in the situation where we are doing these little computational examples and we do not have enough analytic boundary conditions, we generally cheat. We might

say that we are assigning numerical boundary conditions at the boundaries where we do not get to assign a boundary condition, but if you analyzed these boundary conditions via the methods given in Chapter 8, you would see that you have chosen bad numerical boundary conditions. We have generally used the principle that since our computations never run long enough to get near the boundaries, using bad boundary conditions will not hurt us.

Example 9.5.2 Use the MacCormack scheme

$$\mathbf{u}_k^* = \mathbf{u}_k^n - R\delta_+\mathbf{F}_k^n \tag{9.5.4}$$

$$\mathbf{u}_k^{n+1} = \frac{1}{2}\left[\mathbf{u}_k^n + \mathbf{u}_k^* - R\delta_-\mathbf{F}_k^*\right] \tag{9.5.5}$$

to approximate the solution of the inviscid Burgers' equation

$$v_t + \left(\frac{1}{2}v^2\right)_x = 0, \quad x \in (-1,1),\ t > 0 \tag{9.5.6}$$

along with the initial condition

$$v_0(x) = v(x,0) = \begin{cases} -1.0 & \text{if } x \le 0 \\ 1.0 & \text{if } x > 0 \end{cases} \tag{9.5.7}$$

and numerical boundary conditions $u_0^n = -1.0$ and $u_M^n = 1.0$. See Remark, page 142.

Solution Of course, here we apply the scalar version of the MacCormack scheme with $F(v) = v^2/2$. We see in Figure 9.5.2 that the solution given by the MacCormack scheme is the same as the initial condition. This is not all bad, since if we compute $s = (F(v_L) - F(v_R))/(v_L - v_R) = 0$, we see that the speed of propagation of the discontinuity should be zero. It is not hard to see that the solution given in Figure 9.5.2 is a weak solution to the above problem (see Example 9.2.2 for a similar example). However, it is also easy to see that the solution does not satisfy Entropy Condition I, i.e., it is not a vanishing viscosity solution. Using the same approach as was used in Example 9.2.3, we see that

$$v(x,t) = \begin{cases} -1 & \text{if } x < t \\ x/t & \text{if } -t \le x \le t \\ 1 & \text{if } x > t \end{cases}$$

is also a weak solution to this problem. In Section 9.2.4 we discussed how such a solution satisfies Entropy Condition I vacuously because there are no discontinuities in the solution. Hence, the vanishing viscosity solution to the problem (9.5.6), (9.5.7) and numerical boundary conditions $u_0^n = -1.0$ and $u_M^n = 1.0$ will contain a fan and will not be the solution given in Figure 9.5.2.

In Section 9.4 we stated and/or derived several difference schemes that took advantage of the fact that we were solving conservation laws. For example, we suggested that the reader return to HW0.0.2 and try using the Lax-Wendroff scheme (9.4.8) to approximate the solution to the inviscid Burgers' equation with the given initial and boundary conditions. We next consider using difference scheme (9.4.8) to approximate the solution to a Riemann problem involving the inviscid Burgers' equation.

Example 9.5.3 Use the Lax-Wendroff scheme (9.4.8) to approximate the solution to the inviscid Burgers' equation (9.5.6) along with the initial condition

$$v_0(x) = v(x,0) = \begin{cases} 1.0 & \text{if } x \le 0 \\ 0.5 & \text{if } x > 0 \end{cases} \tag{9.5.8}$$

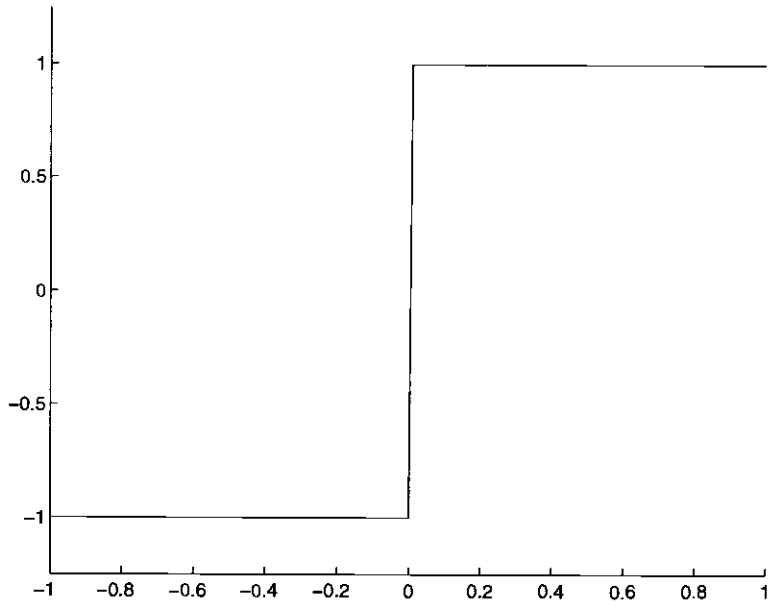

FIGURE 9.5.2. Approximate solution to the inviscid Burgers' equation with initial condition (9.5.7). The solution was computed using the MacCormack scheme (9.5.4)–(9.5.5), $\Delta x = 0.01$, $\Delta t = 0.001$ and the solution is given at time $t = 0.125$.

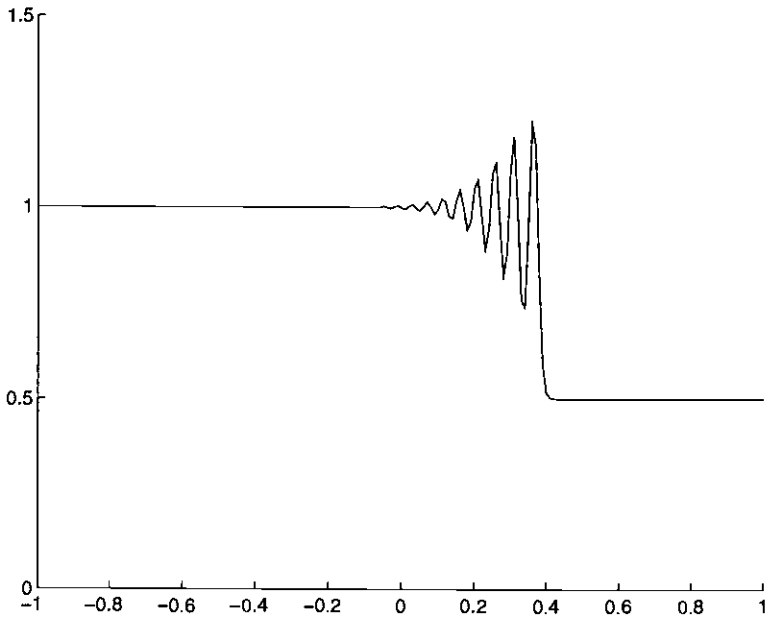

FIGURE 9.5.3. Approximate solution to the inviscid Burgers' equation with initial condition (9.5.8). The solution was computed using the Lax-Wendroff scheme (9.4.8), $\Delta x = 0.01$, $\Delta t = 0.001$ and the solution is given at time $t = 0.5$.

on the region $[-1,1]$ with boundary condition $v(-1,t) = 1.0$ and numerical boundary condition $u_M^n = 0.5$. See Remark, page 142.

Solution: In our implementation of difference scheme (9.4.8), we use $\Delta x = 0.01$, $\Delta t = 0.001$ (so $R = 0.1$) and calculate the solution at $t = 0.5$. We know that the exact vanishing viscosity solution is given by

$$v(x,t) = \begin{cases} 1.0 & \text{if } x \le 3t/4 \\ 0.5 & \text{if } x > 3t/4. \end{cases}$$

Hence, at time $t = 0.5$ we see that the discontinuity should be located at $x = 0.375$. By making a careful measurement on the plot given in Figure 9.5.3 (if that is possible), we see that the jump in the computed solution occurs at approximately the correct spot (if we can decide where the jump occurs). The oscillations that occur should not surprise us. We saw the same types of oscillations in every other occasion where we have tried to use the linear Lax-Wendroff scheme to compute discontinuous solutions, e.g., Sections 5.9.4, 6.10.2 and 7.8. Recall that for linear schemes these oscillations are due to the fact that the leading error term is a dispersive term. The dissipation present is of order four and is not sufficient to dampen these high frequency oscillations. Thus we see that the Lax-Wendroff scheme provides us with approximately the correct speed of propagation of the discontinuity but cannot adequately resolve the discontinuity (or really, the region directly behind the discontinuity). It should not be surprising to us that if we were to compute solutions at a larger time, the solutions would get worse.

Remark: We should note that for scalar schemes we can give several logical approximations to the Lax-Wendroff scheme (9.4.8). For example, we could approximate the derivatives $\left(F'(u)\right)_{k-1/2}^n$ and $\left(F'(u)\right)_{k+1/2}^n$ by $[F(u_{k+1}^n) - F(u_k^n)]/[u_{k+1}^n - u_k^n]$ and $[F(u_k^n) - F(u_{k-1}^n)]/[u_k^n - u_{k-1}^n]$, respectively. Using these differences to approximate the derivatives requires fewer computations than the Lax-Wendroff scheme (9.4.8). Of course, there is the possibility that either $u_{k+1}^n - u_k^n$ or $u_k^n - u_{k-1}^n$ will be zero. We write a version of the Lax-Wendroff scheme as

$$u_k^{n+1} = u_k^n - \frac{R}{2}\delta_0 F_k^n + \frac{R^2}{2}\delta_-\left\{a_{k+1/2}^n \delta_+ F_k^n\right\} \tag{9.5.9}$$

where $a_{k+1/2}^n$ is as defined in equation (9.4.17). Another form given as the Lax-Wendroff scheme that is only slightly different from (9.5.9) (different when $u_k^n = u_{k\pm1}^n$) is as follows.

$$u_k^{n+1} = u_k^n - \frac{R}{2}\delta_0 F_k^n + \frac{R^2}{2}\delta_-\left\{(a_{k+1/2}^n)^2 \delta_+ u_k^n\right\}. \tag{9.5.10}$$

When we refer to the different forms of the Lax-Wendroff scheme, we will try to be careful to refer to them as Lax-Wendroff (9.4.8), etc. We might also mention at this time that in Section 9.6.5 we show on the basis of the computation done in HW9.6.6 that the solution found by difference scheme (9.5.10) will not converge to the vanishing viscosity solution. This fact makes difference scheme (9.5.10) less valuable than it would otherwise be. However, as we shall see later, difference scheme (9.5.10) will be an important part of our definition of high resolution schemes.

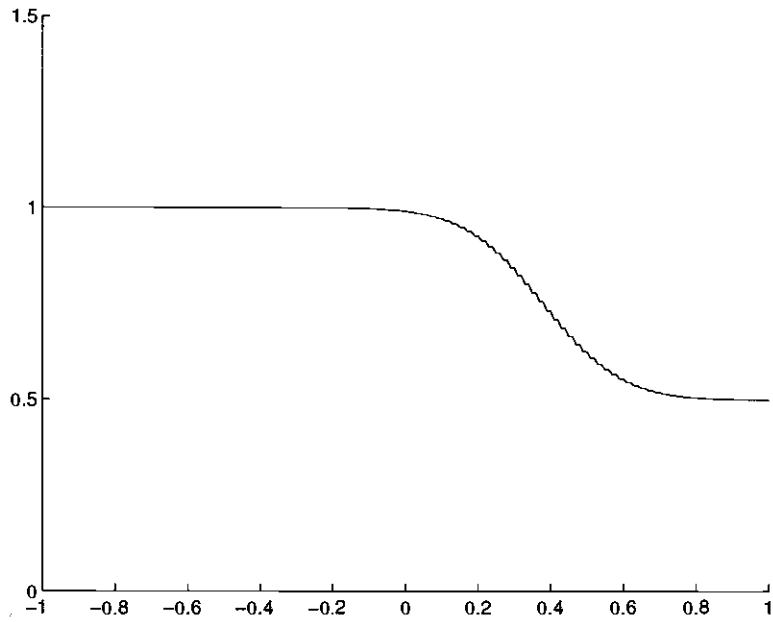

FIGURE 9.5.4. Approximate solution to the inviscid Burgers' equation with initial condition (9.5.8). The solution was computed using the Lax-Friedrichs scheme (9.4.3), $\Delta x = 0.01$, $\Delta t = 0.001$ and the solution is given at time $t = 0.5$.

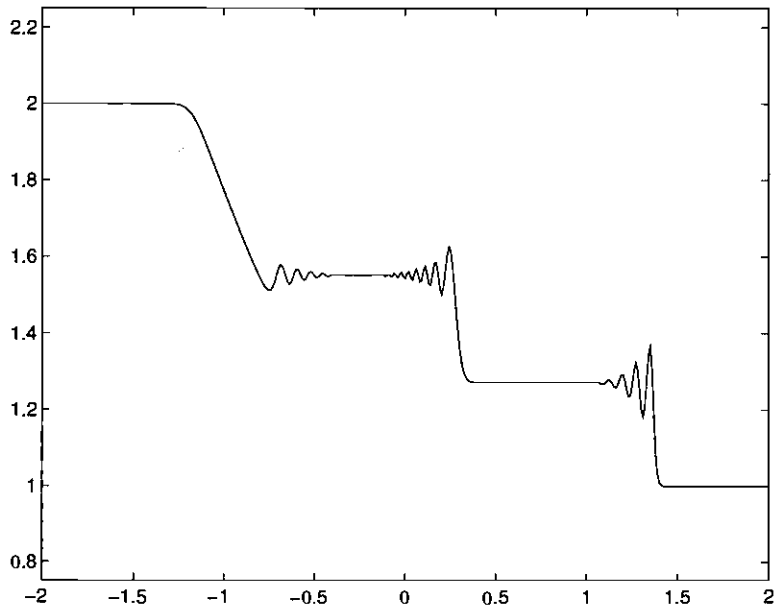

FIGURE 9.5.5. Plot of the density associated with the solution to problem HW0.0.3 at time $t = 1.0$. The solution was found using the Lax-Wendroff scheme, (9.4.8), $\Delta x = 0.01$ and $\Delta t = 0.0025$.

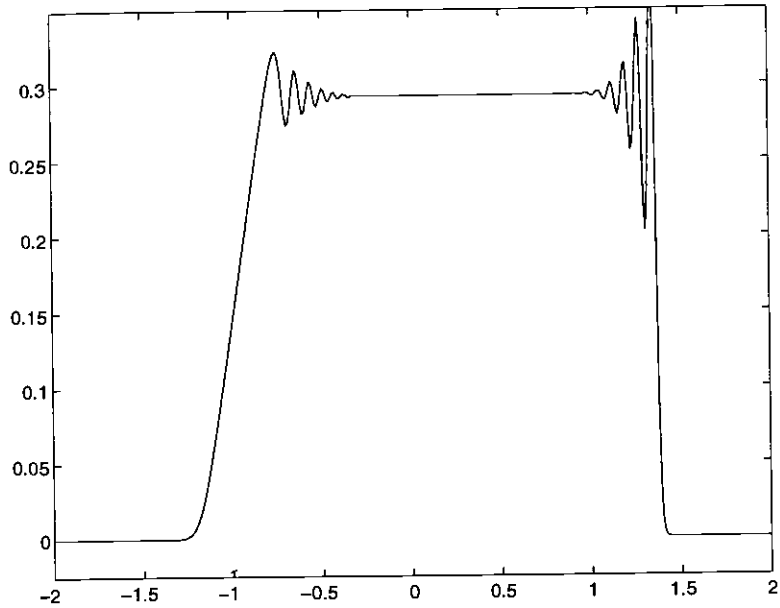

FIGURE 9.5.6. Plot of the velocity associated with the solution to problem HW0.0.3 at time $t = 1.0$. The solution was found using the Lax-Wendroff scheme, (9.4.8), $\Delta x = 0.01$ and $\Delta t = 0.0025$.

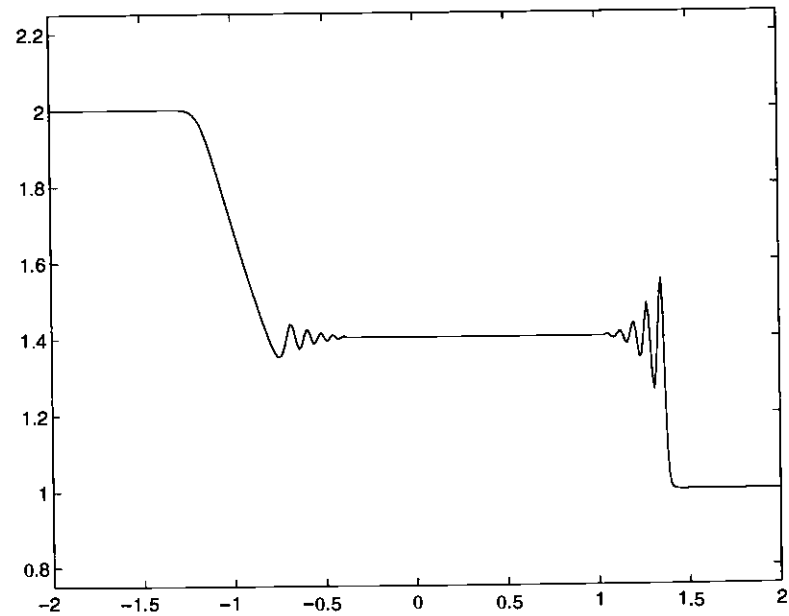

FIGURE 9.5.7. Plot of the pressure associated with the solution to problem HW0.0.3 at time $t = 1.0$. The solution was found using the Lax-Wendroff scheme, (9.4.8), $\Delta x = 0.01$ and $\Delta t = 0.0025$.

Example 9.5.4 Use the Lax-Friedrichs scheme, (9.4.3), to approximate the solution to the problem involving the inviscid Burgers' equation, (9.5.6), along with the initial condition (9.5.8) on the region $[-1, 1]$ along with boundary conditions $v(-1, t) = 1.0$ and numerical boundary condition $u^n_{M+1} = 0.5$. See Remark, page 142.

Solution: We see in Figure 9.5.4 that the solution to the above problem at time $t = 0.5$ places the jump in approximately the correct place, $x = 0.375$. However, we note that the discontinuity is strongly smeared. If we recall our dissipation analysis of the linear Lax-Friedrichs scheme, Section 7.8, this should not surprise us.

Remark: A careful inspection of the front plotted in Figure 9.5.4 shows that like the linear version (Section 7.8), the nonlinear Lax-Friedrichs scheme does not damp out the oscillation that appears in the shortest wave lengths.

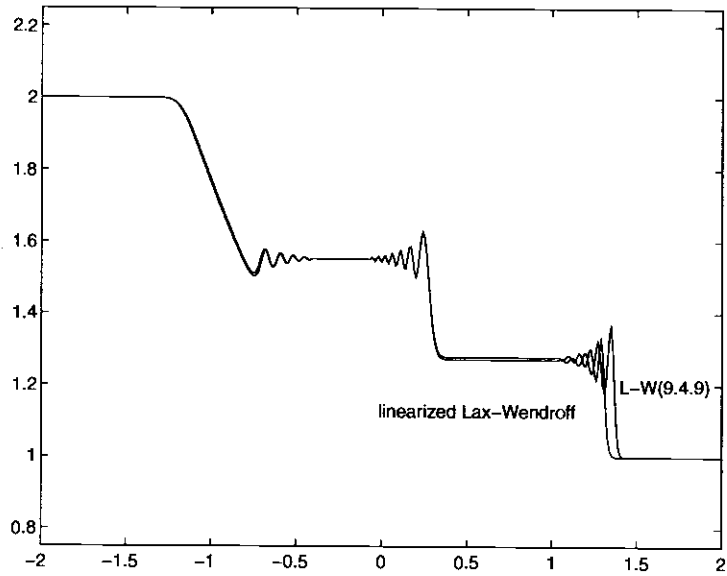

FIGURE 9.5.8. Two plots of the density associated with the solution to problem HW0.0.3 at time $t = 1.0$. One solution was found using the linearized Lax-Wendroff scheme as described in Section 6.10.2, and the other was found using the c onservative form of the Lax-Wendroff scheme, (9.4.8). In both computations we used $\Delta x = 0.01$ and $\Delta t = 0.0025$.

Example 9.5.5 Use the Lax-Wendroff scheme, (9.4.8), to solve the shock tube problem, HW0.0.3.

Solution: We should note that in Section 6.10.2 we suggested that the reader apply the linearized Lax-Wendroff scheme to solve HW0.0.3. In Section 9.3.1 we gave results at time $t = 1.0$ from the linearized computation. And finally, in Section 9.4 we suggested that the reader return to HW0.0.3, using difference scheme (9.4.8). Because we felt that it is important to us, in Figures 9.5.5–9.5.7 we give the results from the last computation at time $t = 1.0$. We note that there is not a great deal of difference between the plots given in Figures 9.3.4–9.3.5 and those given in Figures 9.5.5–9.5.7. The first observation

might be that the oscillations are worse when difference scheme (9.4.8) is used than when the linearized Lax-Wendroff scheme is used.

If a more careful comparison is made in the plots, it is clear that the position of the shock is slightly different for the two computations. In Figure 9.5.8 we see that when the densities from the two schemes are plotted in the same figure, both the fans and the contact discontinuities match up very closely. There is a significant difference between the locations of the shock. There are two questions that appear because of the results plotted in Figure 9.5.8. The first question is, Which result is most accurate? Since we have not given techniques for solving the analytic shock tube problem (and do not want to), we claim that the results obtained using Lax-Wendroff scheme (9.4.8) should produce the more accurate results. For techniques for obtaining the solution to the analytic problem, see ref. [73], page 184. The second question that arises is whether the difference in the location of the shock fronts is because the results using the linearized Lax-Wendroff scheme are less accurate than those using Lax-Wendroff scheme (9.4.8) or whether the two schemes are equally accurate yet produce different shock speeds. It is very difficult to ascertain which of these two possibilities is the case. At $t = 1$ the two schemes do both produce shock speeds that are close enough together so that they are both approximations to the true shock speed, the shock speed produced by Lax-Wendroff scheme (9.4.8) being the more accurate of the two. A larger and more carefully run experiment would be necessary to determine whether the shocks produced by these two schemes continue to separate (i.e., the shocks speeds are actually different, not of equal order Δt or Δx) or whether they stay approximately as near to each other (and the correct position) as they are in Figure 9.5.8. See HW9.5.5.

We should emphasize some of the results obtained above. Examples 9.5.3 and 9.5.5 show that dispersivity is still a problem when we try to approximate solutions to conservation laws, while Example 9.5.4 shows that dissipation can also be a problem. In addition, Examples 9.5.1 and 9.5.2 show that we can use a "logical scheme" to try to approximate the solution to a conservation law and obtain a solution that is not even a weak solution to the problem or obtain a weak solution that is not the vanishing viscosity solution that we want. In the following sections we will try to address some of these problems.

HW 9.5.1 Use difference scheme (9.5.1) to approximate the solution to the inviscid Burgers' equation, (9.5.2), along with initial condition (9.5.8) on the region $[-1, 1]$ with boundary condition $v(-1, t) = 1.0$ and numerical boundary condition $u_M^n = 0.5$. See Remark, page 142.

HW 9.5.2 Use the Lax-Wendroff scheme, (9.4.8) to approximate the solution to the inviscid Burgers' equation, (9.5.6), along with initial condition (9.5.7) on the region $[-1, 1]$ with numerical boundary conditions $u_0^n = -1.0$ and $u_M^n = 1.0$. See Remark, page 142.

HW 9.5.3 Use the MacCormack scheme, (9.5.4)–(9.5.5) to approximate the solution to the inviscid Burgers' equation, (9.5.6), along with initial condition (9.5.8) on the region $[-1, 1]$ with boundary condition $v(-1, t) = 1.0$ and numerical boundary condition $u_{M+1}^n = 0.5$. See Remark, page 142.

HW 9.5.4 Repeat HW9.5.3 with the initial condition

$$v_0(x) = v(x,0) = \begin{cases} 0.5 & \text{if } x \leq 0 \\ 1.0 & \text{if } x > 0 \end{cases}$$

(with boundary condition $v(-1,t) = 0.5$ and numerical boundary condition $u^n_{M+1} = 1.0$). See Remark, page 142.

HW 9.5.5 Repeat the calculations performed in Example 9.5.5 using both the linearized Lax-Wendroff scheme and the Lax-Wendroff scheme (9.4.8). Use the same initial conditions as in HW0.0.3 and computational parameters in Example 9.5.5 except for the length of the domain and the number of grid points. Let the domain of the shock tube go from -8 to 8 (which will require letting $M = 801$). Plot the density obtained from each of the schemes at times $t = 1.0$, $t = 2.0$, $t = 3.0$ and $t = 4.0$. Repeat the above computation using both Δx and Δt half as big. Using these plots, try to determine whether or not the speeds of propagation of the shock are different for the two schemes.

9.6 Difference Schemes for Conservation Laws

We now should have some ideas of the problems that we must face when we try to numerically approximate solutions to problems that involve conservation laws. We should understand that when we used conservation methods to derive difference schemes in Part 1 (Sections 1.6–1.6.4, 2.6.5, 4.2.2 and 5.8), we did so both to teach us how to derive difference schemes that include as much of the physics that is present as is possible and to prepare us for the work in this chapter. We begin by considering the same type of grids on $\mathbb{R}$ or intervals of $\mathbb{R}$ and recall that in Section 1.6.2 we promised that in this chapter we would base our difference equations on cell averages over the control volumes rather than an approximation of the function at cell centers of our grid, i.e., $\mathbf{u}^n_k$ will generally now approximate

$$\int_{x_{k-1/2}}^{x_{k+1/2}} \mathbf{v}(x,t^n)\,dx$$

where $\mathbf{v} = \mathbf{v}(x,t)$ is the solution to appropriate conservation law. We will develop difference schemes in terms of $\mathbf{u}^n_k$ as we did before. We will also consider our approximate solutions to be piecewise constant functions, constant on control volumes $[x_{k-1/2}, x_{k+1/2}]$. One advantage of this approach is that for theoretical purposes our approximate solutions are functions defined on all of $\mathbb{R}$ and can be more easily compared with the solution of the conservation law. The function spaces that we might wish to consider are various spaces of Lebesgue integrable functions such as L_2 (see Section

2.2.3), L_1 or some of the spaces of locally Lebesgue integrable functions, for example $L_{1,loc}$. We denote the piecewise constant function associated with $\mathbf{u}_k^n$ as $\bar{\mathbf{u}}^n$, which is defined as

$$\bar{\mathbf{u}}^n(x) = \mathbf{u}_k^n \text{ for } \left(k - \tfrac{1}{2}\right)\Delta x < x \leq \left(k + \tfrac{1}{2}\right)\Delta x.$$

Then we say that $\mathbf{u}_k^n$ converges to the analytic solution $\mathbf{v} = \mathbf{v}(x, t^n)$ at time $t = t^n$ if $\|\bar{\mathbf{u}}^n - \mathbf{v}(\cdot, t^n)\|_* \to 0$ as $\Delta x \to 0$, where the subscript $*$ denotes the norm in the appropriate space of Lebesgue integrable functions. Since the emphasis here will not be the analysis of the schemes, these spaces will not be especially important for this text. Another advantage of using piecewise constant functions to represent cell averages that we shall see later is that when we consider such approximations, we can adjust these piecewise constant functions slightly to help develop improved schemes.

Remark: We note that the use of piecewise constant functions described above is not that different from the grid centered approach in that if $\mathbf{u}_k^n$ is a grid centered function defined on a grid on $\mathbb{R}$ that is aligned with the centers of our cells (control volumes) and $\bar{\mathbf{u}}^n$ is the piecewise constant analog of $\mathbf{u}_k^{n+1}$, then the L_2 norm of $\bar{\mathbf{u}}^n$ exists if and only if the $\ell_{2,\Delta x}$ norm of $\mathbf{u}_k^n$ exists and

$$\|\bar{\mathbf{u}}^n\|_2 = \|\mathbf{u}_k^n\|_{2,\Delta x}.$$

The same types of identities will be true in other function spaces. This relationship between $\mathbf{u}_k^n$ and $\bar{\mathbf{u}}^n$ allows us to go back and forth between $\mathbf{u}_k^n$ and $\bar{\mathbf{u}}^n$. This relationship also allows us to be sloppy in the use of these two functions (which can be both a good and bad attribute).

9.6.1 Consistency

One basic aspect of difference schemes is that of consistency. As with linear partial differential equations and finite difference schemes, when we try to approximate solutions to problems involving conservation laws, we still want to use difference schemes that are consistent with the appropriate partial differential equation. We hope that by this time the reader has a strong understanding about consistency for the types of linear difference schemes and linear partial differential equations studied in Part 1. As we did for linear schemes, we consider an initial–value problem for a conservation law of the form $\mathbf{v}_t = \mathcal{L}(\mathbf{v}) + \mathcal{F}$, and write our difference scheme both as $L_k^n(\mathbf{u}_k^n) = \mathbf{G}_k^n$ (where this form assumes that we have not yet multiplied through by the Δt) and

$$\mathbf{u}^{n+1} = Q(\mathbf{u}^n) + \Delta t \mathbf{G}^n \tag{9.6.1}$$

(where we now assume that we have multiplied through by Δt and solved for $\mathbf{u}_k^{n+1}$). We note that the parentheses around the $\mathbf{v}$, $\mathbf{u}_k^n$ and $\mathbf{u}^n$ terms in the three equations above indicates that the operators will generally be

nonlinear operators. We also note that we have included a nonhomogeneous term in both the conservation law and the difference equations even though our conservation laws will not generally have a nonhomogeneous term. We include the nonhomogeneous term for convenience in that we will sometimes be faced with a conservation law with a source term and for when we have nonhomogeneous boundary conditions that will make the difference equations nonhomogeneous. And finally, we recall that when we write our difference scheme as we do in (9.6.1), both $\mathbf{u}^n$ and $\mathbf{G}^n$ will be infinite vectors (each component corresponding to a grid point on $\mathbb{R}$) of vectors (since we are considering a K-system conservation law), i.e., $\mathbf{u}^n = [\cdots \ \mathbf{u}_{-1}^n \ \mathbf{u}_0^n \ \mathbf{u}_1^n \ \cdots]^T$ and $\mathbf{G}^n = [\cdots \ \mathbf{G}_{-1}^n \ \mathbf{G}_0^n \ \mathbf{G}_1^n \ \cdots]^T$. We then define the following.

Definition 9.6.1 *The finite difference scheme $L_k^n(\mathbf{u}_k^n) = \mathbf{G}_k^n$ is pointwise consistent with the conservation law $\mathbf{v}_t = \mathcal{L}(\mathbf{v}) + \mathcal{F}$ at point (x,t) if*

$$\boldsymbol{\tau}_k^n = L_k^n(\mathbf{v}(k\Delta x, n\Delta t) - \mathbf{G}_k^n \to 0 \tag{9.6.2}$$

as $\Delta x, \ \Delta t \to 0$ and $(k\Delta x, n\Delta t) \to (x,t)$, where $\mathbf{v}$ is a solution to the conservation law.

Definition 9.6.2 *Difference scheme (9.6.1) is consistent with the conservation law with respect to the norm $\|\cdot\|$ if a solution to the conservation law $\mathbf{v}$ satisfies*

$$\Delta t \boldsymbol{\tau}^n = \mathbf{v}^{n+1} - Q(\mathbf{v}^n) - \Delta t \mathbf{G}^n \tag{9.6.3}$$

and

$$\|\boldsymbol{\tau}^n\| \to 0$$

as $\Delta x, \ \Delta t \to 0$.

Remark 1: We notice in Definition 9.6.1 that the convergence can be considered as pointwise convergence of a sequence of functions. Since $\bar{\mathbf{u}}^n$ (related to $\mathbf{u}_k^n$) is a piecewise constant function, we could consider $\bar{\mathbf{v}}^n$ (related to $\mathbf{v}_k^n$) to be defined as the piecewise constant step function where

$$\bar{\mathbf{v}}^n(\mathbf{x}) = \int_{\mathbf{x_{k-1/2}}}^{\mathbf{x_{k+1/2}}} \mathbf{v}(\mathbf{x}, \mathbf{t}^n)\, \mathbf{dx} \quad \mathbf{x} \in [\mathbf{x_{k-1/2}}, \mathbf{x_{k+1/2}}],$$

and let $\boldsymbol{\tau}_k^n = L_k^n(\mathbf{v}_k^n) - \Delta t \mathbf{G}_k^n$ and let $\bar{\boldsymbol{\tau}}^n$ denote the appropriate piecewise constant function that results from application of the difference operator. Then we could say that the difference scheme and the conservation law are consistent at the point (x,t^n) if the function $\bar{\boldsymbol{\tau}}^n$ converges pointwise to zero at the point x (even though piecewise convergence is generally not useful in spaces of Lebesgue integrable functions). The sequence of functions that we use here is some sequence such that $\Delta x, \ \Delta t \to 0$ and $(k\Delta x, n\Delta t) \to (x,t^n)$.

However, we do not wish to do this. We wish to perform computations as we do below in Remark 2.

Remark 2: There is a slight difference in how the consistency computations must be done in order to take into account the nonlinearity in the conservation law and the difference scheme. For example, consider the inviscid Burgers' equation along with the FTFS scheme, (9.4.1), i.e.,

$$u_k^{n+1} = u_k^n - R\delta_+ F_k^n$$

where $F_k^n = (u_k^n)^2/2$. If we let $v = v(x,t)$ be a solution to the inviscid Burgers' equation, then

$$\begin{aligned}
&\frac{v_k^{n+1} - v_k^n}{\Delta t} + \frac{\frac{1}{2}\left[v_{k+1}^n\right]^2 - \frac{1}{2}\left[v_k^n\right]^2}{\Delta x} \\
&\quad = (v_t)_k^n + \mathcal{O}(\Delta t) + \frac{1}{\Delta x}\left\{ \left(\frac{1}{2}v^2\right)_k^n + \left[\left(\frac{1}{2}v^2\right)_x\right]_k^n \Delta x \right. \\
&\qquad\qquad \left. + \mathcal{O}(\Delta x^2) - \left(\frac{1}{2}v^2\right)_k^n \right\} \\
&\quad = \left[v_t + \left(\frac{1}{2}v^2\right)_x\right]_k^n + \mathcal{O}(\Delta t) + \mathcal{O}(\Delta x) \\
&\quad = \mathcal{O}(\Delta t) + \mathcal{O}(\Delta x),
\end{aligned}$$

(since v is a solution of the conservation law).

It would have been possible to expand v_{k+1}^n in a Taylor series (instead of $(v^2)_{k+1}^n$) and square the result. The only difficulty with that approach is that we then must recognize the nonconservative form of the partial differential equation when we use the fact that v satisfies the partial differential equation.

Remark 3: Another difference between the computation necessary for conservation laws and what we usually did for linear equations is that the above computation is misleading. Recall that we are using cell averaged equations. Our difference operators technically act on functions. However, as we showed in Section 1.6.2, it is easy to go back and forth between the vector $\mathbf{u}^n$ and the piecewise constant function $\bar{\mathbf{u}}^n$. Recall that immediately preceding Definition 9.6.1 we reminded the reader that vectors $\mathbf{u}^n$ and $\mathbf{G}^n$ are given in terms of $\mathbf{u}_k^n$ and $\mathbf{G}_k^n$. We will assume that whenever we need to be more careful, we can perform the computation as done in Section 1.6.2 that will apply specifically to piecewise functions. We will generally not do the computations for piecewise continuous functions here.

Remark 4: The final difference between how Definitions 9.6.1 and 9.6.2 above are used compared to their linear analogues is the choice of norm.

As we mentioned earlier, $\mathbf{u}_k^n$ can be considered either as a sequence of points in $\ell_{2,\Delta x}$ as we did in earlier chapters or as a function defined on $\mathbb{R}$. Because of our discussion in Remarks 1 and 3, Definition 9.6.1 will be applied pointwise, as we have done in the past. However, there will be times when it is easiest (or best) to view Definition 9.6.2 in the appropriate space of Lebesgue integrable functions. The use of these spaces is important for certain theoretical results. Because of the statements made earlier relating the L_2 and $\ell_{2,\Delta x}$ norms (or other norms when necessary), we can apply Definition 9.6.2 to either vectors $\mathbf{v}^n$ or piecewise constant functions $\bar{\mathbf{v}}^{\mathbf{n}}$ (where the constant is defined by the appropriate vector element). Using this approach, the vector form of the truncation error $\boldsymbol{\tau}^n$ and the truncation error function $\bar{\tau}^{\mathbf{n}}$ can be used interchangeably.

Remark 5: The most important point that we must make here is the importance of consistency. Of course, it is important to use difference schemes that are consistent with the conservation law we wish to approximate. In the case of linear equations, we knew that if our difference scheme was consistent with our partial differential equation and stable, then it was convergent. What we mean by "convergent" is that the sequence of approximate solutions converges to the solution (or really, a solution) of the partial differential equation. Consistency and stability will not imply convergence for conservation laws. In Example 9.5.1, we performed a computation with a difference scheme that is consistent with the partial differential equation. The numerical solution is so well behaved that it is clear that the scheme is both stable and convergent. But the approximate solutions converge to a function that is not even a weak solution of the conservation law.

It is easy to understand why consistency and convergence might not ensure that the limiting function is a solution to the analytic problem. In Example 9.5.1, the way that we saw that $v(x,t) = v_0(x)$ was not a weak solution was to show that it did not satisfy the jump condition across the discontinuity. There is nothing in the consistency argument that will force the resulting solution to satisfy a jump condition. We note that the Taylor series expansions used in the consistency argument depend strongly on the smoothness of the solutions. Surely these expansions and, hence, the consistency arguments will give us no information about what happens across a discontinuity.

9.6.2 Conservative Schemes

To imitate the behavior of solutions to conservation laws, it would seem that a logical approach is to require the solutions to our numerical schemes to satisfy some sort of jump condition. Since our solutions to the difference equations are piecewise constant functions and are represented by a series of jumps, this is either very difficult or impossible. If we recall our derivation of the jump conditions, we see that it depended strongly on the fact that our

partial differential equation is in conservation form. Hence, the approach we shall take is to consider conservative difference schemes—difference schemes in conservation form.

We return to the conservation law approach introduced in Section 1.6 and Section 9.1 and consider the conservation law

$$\mathbf{v}_t + \mathbf{F}_x = \boldsymbol{\theta}. \tag{9.6.4}$$

If we integrate partial differential equation (9.6.4) from $x_{k-1/2}$ to $x_{k+1/2}$ with respect to x and from t_n to t_{n+1} with respect to t, we get

$$\int_{t_n}^{t_{n+1}} \int_{x_{k-1/2}}^{x_{k+1/2}} \mathbf{v}_t(x,t)\,dx\,dt + \int_{t_n}^{t_{n+1}} \int_{x_{k-1/2}}^{x_{k+1/2}} \left[\mathbf{F}(\mathbf{v}(x,t))\right]_x dx\,dt = \boldsymbol{\theta}.$$

Reordering the first integral and performing the integration with respect to t and performing the integration with respect to x in the second integral leaves us with

$$\int_{x_{k-1/2}}^{x_{k+1/2}} \mathbf{v}(x,t_{n+1})\,dx - \int_{x_{k-1/2}}^{x_{k+1/2}} \mathbf{v}(x,t_n)\,dx + \int_{t_n}^{t_{n+1}} \mathbf{F}(\mathbf{v}(x_{k+1/2},t))\,dt - \int_{t_n}^{t_{n+1}} \mathbf{F}(\mathbf{v}(x_{k-1/2},t))\,dt = \boldsymbol{\theta}.$$

We should note that the above equation is a special case of equation (9.1.2) derived in Section 9.1. Using the definition of cell averages gives us

$$\Delta x\ (\mathbf{v}_k^{n+1} - \mathbf{v}_k^n) + \left[\int_{t_n}^{t_{n+1}} \mathbf{F}(\mathbf{v}(x_{k+1/2},t))\,dt - \int_{t_n}^{t_{n+1}} \mathbf{F}(\mathbf{v}(x_{k-1/2},t))\,dt\right] = \boldsymbol{\theta}. \tag{9.6.5}$$

We should remember that $\mathbf{v}$ is a solution to partial differential equation (9.6.4) and that equation (9.6.5) is an exact equation. An interpretation of equation (9.6.5) is that the difference in the "amounts of material" (whatever material the conservation law is conserving) entering and/or leaving the control volume $[x_{k-1/2}, x_{k+1/2}] \times [t_n, t_{n+1}]$ across the top and bottom, $t = t_n$ and $t = t_{n+1}$, is balanced by the amount of material entering and/or leaving the sides, $x = x_{k-1/2}$ and $x = x_{k+1/2}$. We should remember that $\mathbf{F}$ is referred to as the "flux function" so that the integrals on the right hand side of equation (9.6.5) give the flux across the sides $x = x_{k-1/2}$ and $x = x_{k+1/2}$.

Based on the above discussion, we consider schemes that approximate equation (9.6.5) (and, hence, approximate equation (9.6.4)) of the form

$$\mathbf{u}_k^{n+1} = \mathbf{u}_k^n - R\left(\mathbf{h}_{k+1/2}^n - \mathbf{h}_{k-1/2}^n\right) \tag{9.6.6}$$

where $R = \Delta t/\Delta x$, and $\Delta t\, h^n_{k-1/2}$ and $\Delta t\, h^n_{k+1/2}$ approximate the flux of material across the sides $x = x_{k-1/2}$ and $x = x_{k+1/2}$, respectively (and approximate the two integrals in equation (9.6.5)). The approximate fluxes are written as

$$\mathbf{h}^n_{k+1/2} = \mathbf{h}(\mathbf{u}^n_{k-p}, \cdots, \mathbf{u}^n_{k+q}) \tag{9.6.7}$$

$$\mathbf{h}^n_{k-1/2} = \mathbf{h}(\mathbf{u}^n_{k-p-1}, \cdots, \mathbf{u}^n_{k+q-1}) \tag{9.6.8}$$

or

$$\mathbf{h}^n_{k+1/2} = \mathbf{h}(\mathbf{u}^n_k, \mathbf{u}^n_{k+1}) \tag{9.6.9}$$

$$\mathbf{h}^n_{k-1/2} = \mathbf{h}(\mathbf{u}^n_{k-1}, \mathbf{u}^n_k) \tag{9.6.10}$$

depending on whether $\mathbf{h}^n_{k\pm1/2}$ depends on $\mathbf{u}$ at $p+q+1$ points or at two points. The function $\mathbf{h}$ is called the **numerical flux function**.

Difference scheme (9.6.6) with $\mathbf{h}^n_{k\pm1/2}$ in either form (9.6.7) and (9.6.8) or (9.6.9) and (9.6.10) is called a **conservative difference scheme**. When $\mathbf{h}^n_{k\pm1/2}$ is given by (9.6.9) and (9.6.10), conservative difference scheme (9.6.6) is called a **three-point scheme**. When it is either clear or we do not care on which values of $\mathbf{u}^n_k$ the scheme depends, we will generally write our schemes as in (9.6.6).

We should note that though we will generally want to use conservative difference schemes, we still want our difference scheme to be consistent with our partial differential equation. If we let $\mathbf{v}$ be a solution to conservation law (9.6.4) and consider the three-point conservative scheme (9.6.6), (9.6.9), (9.6.10), we see that

$$\begin{aligned}
\mathbf{v}^{n+1}_k &- \mathbf{v}^n_k + R\left[\mathbf{h}(\mathbf{v}^n_k, \mathbf{v}^n_{k+1}) - \mathbf{h}(\mathbf{v}^n_{k-1}, \mathbf{v}^n_k)\right] \\
&= \mathbf{v}^n_k + (\mathbf{v}_t)^n_k\,\Delta t + \mathcal{O}(\Delta t^2) - \mathbf{v}^n_k \\
&\quad + R\left[\mathbf{h}(\mathbf{v}^n_k, \mathbf{v}^n_k) + \mathbf{h}_2(\mathbf{v}^n_k, \mathbf{v}^n_k)(\mathbf{v}^n_{k+1} - \mathbf{v}^n_k) + \cdots\right] \\
&\quad - R\left[\mathbf{h}(\mathbf{v}^n_k, \mathbf{v}^n_k) + \mathbf{h}_1(\mathbf{v}^n_k, \mathbf{v}^n_k)(\mathbf{v}^n_{k-1} - \mathbf{v}^n_k) + \cdots\right] \\
&= (\mathbf{v}_t)^n_k\,\Delta t + \mathcal{O}(\Delta t^2) \\
&\quad + R\left[\mathbf{h}_2(\mathbf{v}^n_k, \mathbf{v}^n_k)\left\{\mathbf{v}^n_k + (\mathbf{v}_x)^n_k\,\Delta x + \mathcal{O}(\Delta x^2) - \mathbf{v}^n_k\right\} + \cdots\right] \\
&\quad - R\left[\mathbf{h}_1(\mathbf{v}^n_k, \mathbf{v}^n_k)\left\{\mathbf{v}^n_k + (\mathbf{v}_x)^n_k\,(-\Delta x) + \mathcal{O}(\Delta x^2) - \mathbf{v}^n_k\right\} + \cdots\right] \\
&= (\mathbf{v}_t)^n_k\,\Delta t + R\left[\mathbf{h}_1(\mathbf{v}^n_k, \mathbf{v}^n_k)\,(\mathbf{v}_x)^n_k + \mathbf{h}_2(\mathbf{v}^n_k, \mathbf{v}^n_k)\,(\mathbf{v}_x)^n_k\right]\Delta x \\
&\quad + \mathcal{O}(\Delta t^2) + \mathcal{O}(\Delta t \Delta x) \\
&= (\mathbf{v}_t)^n_k\,\Delta t + R\left[(\mathbf{h}(\mathbf{v}, \mathbf{v}))_x\right]^n_k \Delta x \\
&\quad + \mathcal{O}(\Delta t^2) + \mathcal{O}(\Delta t \Delta x) \\
&= \left[\mathbf{v}_t + \frac{\partial}{\partial x}\mathbf{h}(\mathbf{v}, \mathbf{v})\right]^n_k \Delta t + \mathcal{O}(\Delta t^2) + \mathcal{O}(\Delta t \Delta x)
\end{aligned} \tag{9.6.11}$$

(where $\mathbf{h}_1$ and $\mathbf{h}_2$ represent the partial derivatives of $\mathbf{h}$ with respect to the first and second argument, respectively). Note that the Δx^2 changed to

$\Delta t \Delta x$ in the order terms when we multiplied the $\mathcal{O}(\Delta x^2)$ by R. We should also make a special note that in the consistency argument, the last term represents $\Delta t \tau_k^n$ (Δt times the truncation error). Hence, if the scheme is to be consistent with conservation law (9.6.4), we must be able to eliminate the Δt term in equation (9.6.11), i.e., we must have

$$\left[\mathbf{v}_t + \frac{\partial}{\partial x}\mathbf{h}(\mathbf{v},\mathbf{v})\right]_k^n \Delta t = 0.$$

This will happen if $\mathbf{h}(\mathbf{v},\mathbf{v}) = \mathbf{F}(\mathbf{v})$. We have proved the following result.

Proposition 9.6.3 *If* $\mathbf{h}(\mathbf{v},\mathbf{v}) = \mathbf{F}(\mathbf{v})$, *then difference scheme (9.6.6) will be consistent with conservation law (9.6.4).*

Remark 1: We should note that if we use difference scheme (9.6.6) along with $\mathbf{h}_{k\pm1/2}^n$ given by (9.6.7) and (9.6.8), we have the following analogous result (though it should be clear that the necessary computation would be nastier to write).

Proposition 9.6.4 *If* $\mathbf{h}(\mathbf{v},\ldots,\mathbf{v}) = \mathbf{F}(\mathbf{v})$, *then difference scheme (9.6.6) will be consistent with conservation law (9.6.4).*

Remark 2: In the proof of consistency given above, we used the fact that $\mathbf{h}$ is differentiable. It is possible to prove the same results as above assuming that $\mathbf{h}$ is only Lipschitz continuous in each variable (which is often the case).

We started our discussion of conservative schemes because in Example 9.5.1 we have a solution to a discrete problem (a discrete problem that is consistent with a given analytic problem) that converges to a function that is not a weak solution to the analytic problem. The following theorem shows that if we use conservative schemes, we will get weak solutions in the limit. We assume that we are considering an initial–value problem based on a conservation law of the form (9.6.4) and will obtain approximate solutions to this problem by using a conservative difference scheme of the form (9.6.6).

Theorem 9.6.5 Lax-Wendroff Theorem *If* $\mathbf{u}_k^n$ *is a discrete solution based on a consistent, conservative difference approximation to a given conservation law initial–value problem and if* $\mathbf{u}_k^n \to \mathbf{v}$ *in* $L_{1,loc}$ *as* $\Delta x,\ \Delta t \to 0$, *then* $\mathbf{v} = \mathbf{v}(x,t)$ *is a weak solution to the initial–value problem.*

Proof: We write difference scheme (9.6.6) as

$$\frac{\mathbf{u}_k^{n+1} - \mathbf{u}_k^n}{\Delta t} + \frac{\mathbf{h}_{k+1/2}^n - \mathbf{h}_{k-1/2}^n}{\Delta x} = \boldsymbol{\theta},$$

multiply by ϕ_k^n where $\phi_k^n = \phi(k\Delta x, n\Delta t)$ for $\phi \in C_0^1$ and sum over k and n

to get

$$\sum_{n=0}^{\infty}\sum_{k=-\infty}^{\infty}\left\{\phi_k^n\frac{\mathbf{u}_k^{n+1}-\mathbf{u}_k^n}{\Delta t}+\phi_k^n\frac{\mathbf{h}_{k+1/2}^n-\mathbf{h}_{k-1/2}^n}{\Delta x}\right\}=\boldsymbol{\theta}. \tag{9.6.12}$$

In the continuous analogue we would now use integration by parts to transfer the derivatives from the partial differential equation to the test function. We use the analogous summation by parts rule

$$\sum_{k=q}^{p}a_k\delta_+b_k=a_pb_{p+1}-a_qb_q-\sum_{k=q+1}^{p}b_k\delta_-a_k \tag{9.6.13}$$

(HW9.6.1) and get

$$-\sum_{n=0}^{\infty}\sum_{k=-\infty}^{\infty}\left\{\mathbf{u}_k^{n+1}\frac{\phi_k^{n+1}-\phi_k^n}{\Delta t}+\mathbf{h}_{k+1/2}^n\frac{\phi_{k+1}^n-\phi_k^n}{\Delta x}\right\}-\sum_{k=-\infty}^{\infty}\frac{\mathbf{u}_k^0\phi_k^0}{\Delta t}=\boldsymbol{\theta}. \tag{9.6.14}$$

All of the boundary terms except the term at $n=0$ disappear since the function ϕ has compact support. If we then multiply equation (9.6.14) by $\Delta x\Delta t$ and let $\Delta x,\ \Delta t\to 0$ (using the fact that $\mathbf{u}_k^n\to\mathbf{v}$), we get

$$-\int_0^{\infty}\int_{-\infty}^{\infty}\{\mathbf{v}\phi_t+\mathbf{h}(\mathbf{v},\ldots,\mathbf{v})\phi_x\}\,dx\,dt-\int_{-\infty}^{\infty}\mathbf{v}_0\phi(x,0)\,dx=\boldsymbol{\theta}.$$

Since $\mathbf{h}(\mathbf{v},\ldots,\mathbf{v})=\mathbf{F}(\mathbf{v})$, we have

$$-\int_0^{\infty}\int_{-\infty}^{\infty}\{\mathbf{v}\phi_t+\mathbf{F}(\mathbf{v})\phi_x\}\,dx\,dt-\int_{-\infty}^{\infty}\mathbf{v}_0\phi(x,0)\,dx=\boldsymbol{\theta}. \tag{9.6.15}$$

Hence, by Definition 9.2.5 the limit function $\mathbf{v}$ is a weak solution of the conservation law.

Remark 1: We note that we have not taken the trouble to justify most of the difficult steps given above. Specifically, we surely did not justify the step where the limit is taken that produces the integrals. It is this step that requires the hypothesis that $\mathbf{u}_k^n\to\mathbf{v}$ in $L_{1,loc}$.

Remark 2: We emphasize that this theorem does not claim that the only way to obtain a weak solution is to use a conservative scheme. If we consider the results of Example 9.5.1 along with Theorem 9.6.5, it is clear that difference scheme (9.5.1) is not conservative. The results of HW9.5.1 show that it is possible to obtain reasonably good approximations to weak solutions to given conservation laws using a nonconservative scheme.

We see that to ensure that we obtain weak solutions to our given problems, we will generally want to use conservative schemes. The easiest approach to obtaining conservative schemes is to provide the numerical flux

function associated with the scheme and use difference (9.6.6). Often, the numerical flux function is based on physical principles by approximating the physics that is used to derive the analytic conservation law. We should realize that in Section 5.3.1, we could have derived the FTFS scheme for approximating the solution of the initial–value problem $v_t + av_x = 0, v(x,0) = v_0(x)$ for $a < 0$ by using a numerical flux function $h^n_{k+1/2} = u^n_{k+1}$ (with an analogous result for the FTBS scheme for $a > 0$ using $h^n_{k+1/2} = u^n_k$). We should also emphasize that in Section 5.3.2 we could have used the numerical flux function $h^n_{k+1/2} = (u^n_k + u^n_{k+1})/2$ to derive the FTCS scheme for solving the same equation. Since the FTCS scheme is an unstable scheme for solving the one way wave equation, we emphasize the claim that we made several times in Part 1 that *it is possible to derive bad conservative schemes.*

If we return to the integral form of the conservation law given in equation (9.6.5) and approximate $\mathbf{F}(\mathbf{v}(x_{k+1/2},t))$ and $\mathbf{F}(\mathbf{v}(x_{k-1/2},t))$ in the first and second integrals of equation (9.6.5) by $\mathbf{F}(\mathbf{u}^n_{k+1})$ and $\mathbf{F}(\mathbf{u}^n_k)$, respectively, we obtain the nonlinear FTFS difference scheme (9.4.1), i.e., by choosing $\mathbf{h}^n_{k+1/2} = \mathbf{F}^n_{k+1}$ and $\mathbf{h}^n_{k-1/2} = \mathbf{F}^n_k$. Thus, we see that the FTFS difference scheme (9.4.1) is conservative. To see how well these choices approximate equation (9.6.5) we still must perform a consistency argument. However, by noting that $\mathbf{h}(\mathbf{u},\mathbf{u}) = \mathbf{F}(\mathbf{u})$ and applying Proposition 9.6.3, we can easily see that the FTFS difference scheme (9.4.1) is consistent with conservation law (9.6.4) (in fact, accurate of order $\mathcal{O}(\Delta t) + \mathcal{O}(\Delta x)$ or greater). In a similar manner, we can justify that many of the other schemes with which we have worked are consistent, conservative schemes. It is sometimes more difficult to decide when a scheme is not conservative. In Example 9.5.1, since the approximate solutions converged to a function that is not a weak solution, we know by Theorem 9.6.5 that difference scheme (9.5.1) is not conservative. Likewise, we saw in HW9.4.4 that upwind scheme (9.4.15) will compute and converge to a function that is not a weak solution. Hence, upwind scheme (9.4.15) is not conservative. In these cases, it is easy to describe why difference scheme (9.5.1) cannot be written in the form of (9.6.6) (because if we could write these schemes in the form of (9.6.6), it would contradict Theorem 9.6.5). Without using a result like Theorem 9.6.5, it is difficult to show that a difference scheme cannot be written in the form of difference scheme (9.6.6).

We now review several of the difference schemes with which we are already familiar. We note that if we set

$$\mathbf{h}^n_{k+1/2} = \frac{1}{2}\left[\mathbf{F}^n_{k+1} + \mathbf{F}^n_k\right] - \frac{R}{2}A_{k+1/2}\delta_+\mathbf{F}^n_k \tag{9.6.16}$$

where $A_{k+1/2} = \mathbf{F}'\left(\left(\mathbf{u}^n_{k+1} + \mathbf{u}^n_k\right)/2\right)$, we see that the Lax-Wendroff scheme, (9.4.8), can be written as (9.6.6) and is a conservative scheme. Also, it is not difficult to see that since $\mathbf{h}(\mathbf{u},\mathbf{u}) = \mathbf{F}(\mathbf{u})$, the Lax-Wendroff scheme (9.4.8) is also consistent with conservation law (9.6.4). From the fact that we know

that the Lax-Wendroff scheme is consistent and conservative, along with the results of Example 9.5.3, we see that *using a consistent, conservative scheme is not enough to eliminate the dispersive wiggles* that we saw in the calculation in Example 9.5.3. In addition, it is also not difficult to show that the other forms of the Lax-Wendroff scheme, (9.5.9) and (9.5.10), are also both conservative and consistent with conservation law (9.6.4).

By setting

$$\mathbf{h}^n_{k+1/2} = \frac{1}{2}\left[\mathbf{F}^n_{k+1} + \mathbf{F}^n_k\right] - \frac{1}{2R}\left(\mathbf{u}^n_{k+1} - \mathbf{u}^n_k\right) \tag{9.6.17}$$

we see that the Lax-Friedrichs scheme (9.4.3) is a conservative scheme. Again, it is easy to note that since $\mathbf{h}(\mathbf{u}, \mathbf{u}) = \mathbf{F}(\mathbf{u})$, the Lax-Friedrichs scheme is consistent with conservation law (9.6.4). If we review the results from Example 9.5.4, we see that *using a consistent, conservative scheme (Lax-Friedrichs in this case) is not enough to guarantee that we do not obtain a badly smeared solution.*

Also, if we define

$$h^n_{k+1/2} = \frac{1}{2}\left(F^n_k + F^n_{k+1}\right) - \frac{1}{2}|a^n_{k+1/2}|\delta_+ u^n_k, \tag{9.6.18}$$

we see that upwind scheme (9.4.16) is conservative. Since $h(u, u) = F(u)$, upwind scheme (9.4.16) is consistent with the scalar version of conservation law (9.6.4) (or conservation law (9.2.1)). We note that though the solution found in HW9.4.5) is not the solution that we wanted, the solution is a weak solution.

And finally, if we let

$$\mathbf{h}^n_{k+1/2} = \frac{1}{2}\left[\mathbf{F}^n_{k+1} + \mathbf{F}(\mathbf{u}^*_k)\right] \tag{9.6.19}$$

where $\mathbf{u}^*_k$ is given by (9.5.4), we see that the MacCormack scheme (9.5.4)–(9.5.5) is conservative. We note that *using a consistent, conservative scheme is not sufficient to ensure that the solutions of the scheme will converge to a vanishing viscosity solution of the associated conservation law* (Example 9.5.2).

HW 9.6.1 Derive summation by parts rule (9.6.13).

HW 9.6.2 Verify that the various forms of the Lax-Wendroff scheme, (9.4.8), (9.5.9) and (9.5.10); the Lax-Friedrichs scheme, (9.4.3); and MacCormack's scheme, (9.5.4)–(9.5.5), are consistent with conservation law (9.6.4).

HW 9.6.3 (a) Verify that the Beam-Warming scheme (9.4.13)–(9.4.14) is conservative with numerical flux function

$$\mathbf{u}^*_k = \mathbf{u}^n_k - R\delta_- \mathbf{F}^n_k \tag{9.6.20}$$

$$\mathbf{h}^n_{k+1/2} = \frac{1}{2}\left[\mathbf{F}^n_k + \mathbf{F}^*_k\right] + \frac{1}{2}\delta_- \mathbf{F}^n_k. \tag{9.6.21}$$

(b) Show that the Beam-Warming scheme is consistent with conservation law (9.6.4).

9.6.3 Discrete Conservation

We noted in Section 9.1 that for any region $[a,b] \times [t_1,t_2]$, the solution to conservation law (9.1.1) $\mathbf{v}$ will satisfy the integral form of the conservation law

$$\int_a^b \mathbf{v}(x,t_2)dx - \int_a^b \mathbf{v}(x,t_1)\,dx = -\left(\int_{t_1}^{t_2} \mathbf{F}(\mathbf{v}(b,t))\,dt - \int_{t_1}^{t_2} \mathbf{F}(\mathbf{v}(a,t))\,dt\right). \tag{9.6.22}$$

Since we refer to scheme (9.6.6), with $\mathbf{h}^n_{k\pm1/2}$ written as (9.6.7) and (9.6.8), or (9.6.9) and (9.6.10), as a conservative difference scheme, we might desire and/or expect this scheme to satisfy some analogue of the integral form of the conservation law (9.6.22). If we sum difference scheme (9.6.6) in k and n from $k = k_1$ to $k = k_2$ and from $n = n_1$ to $n = n_2$, we get

$$\sum_{n=n_1}^{n_2}\sum_{k=k_1}^{k_2} \mathbf{u}_k^{n+1} = \sum_{n=n_1}^{n_2}\sum_{k=k_1}^{k_2} \left\{\mathbf{u}_k^n - R\left[\mathbf{h}^n_{k+1/2} - \mathbf{h}^n_{k-1/2}\right]\right\}. \tag{9.6.23}$$

If in equation (9.6.23) we move the $\mathbf{u}_k^n$ term to the left hand side, interchange the order of summation, and use the fact that we have a telescoping sum with respect to n, we get

$$\sum_{k=k_1}^{k_2} \left(\mathbf{u}_k^{n_2+1} - \mathbf{u}_k^{n_1}\right).$$

If we sum the terms in equation (9.6.23) involving $\mathbf{h}$ over k using the fact that this will be a telescoping sum with respect to k and multiply the result by Δx, we can rewrite equation (9.6.23) as

$$\sum_{k=k_1}^{k_2} \left(\mathbf{u}_k^{n_2+1} - \mathbf{u}_k^{n_1}\right)\Delta x = -\sum_{n=n_1}^{n_2} \left(\mathbf{h}^n_{k_2+1/2} - \mathbf{h}^n_{k_1-1/2}\right)\Delta t. \tag{9.6.24}$$

Equation (9.6.24) is referred to as the **summation form of the conservation law**, and the interpretation of its relationship to conservation is almost exactly the same as we described for the integral form of the conservation law in Section 9.1. The left hand side of equation (9.6.24) describes the change in the amount of the conserved material in the interval $[x_{k_1-1/2}, x_{k_2+1/2}]$ between times $t = t_1$ and $t = t_2$. The physical interpretation of the right hand side of equation (9.6.24) is that this change

from $t = t_1$ to $t = t_2$ is due to the flux of material across the boundaries $x = x_{k_1-1/2}$ and $x = x_{k_2+1/2}$ during the time interval from $t = t_1$ to $t = t_2$. The only difference between the integral form of the conservation law and the summation form of the conservation law is that the space-time region considered in the discrete case must be unions of control volumes (the time limits are only times at which the difference equation is defined, and the spatial interval is bounded by the points on which we compute plus or minus $\Delta x/2$), whereas the region in space-time considered in the continuous case is arbitrary. However, when we consider the fact that we are approximating a continuous problem by a discrete problem, the summation form of the conservation law given by equation (9.6.24) is about as arbitrary as is possible. We should also be aware that if we have smoothness assumptions, place a grid on $[a, b] \times [t_1, t_2]$, apply the summation form of the conservation law (9.6.24), and let $\Delta x,\ \Delta t \to 0$, we can derive the integral form of the conservation law (9.6.22) from the summation form of the conservation law.

We emphasize that to parallel the analytic concept of conservative equations, equation (9.6.24) must hold for all k_1, k_2, n_1 and n_2. There are situations in which a scheme can conserve globally (especially when we have boundary conditions that can force global conservation into the system) but not locally. These situations are sometimes the most difficult in that you may get a solution that satisfies certain global properties that you know the solution should have, yet locally the solution can be very inaccurate. However, when we use conservative schemes (9.6.6), the conservation is satisfied both locally and globally.

9.6.4 The Courant-Friedrichs-Lewy Condition

If we reconsider the various discussions we have had about proving that the CFL condition is necessary for convergence of linear difference equations for solving linear, hyperbolic partial differential equations, it should not surprise us that the CFL condition will also be relevant for difference schemes for solving conservation laws. As in the linear case, if for some reason the speed of propagation for the difference scheme is greater than the speed of propagation of the conservation law, at a given point the limiting solution of the difference scheme could be completely unaware of a part of the initial condition on which the solution depends at that point. Hence, it would be impossible for the limiting solution of the difference scheme to be the same as the analytic solution.

To be more specific, consider the scalar conservation law

$$v_t + F(v)_x = 0, \quad x \in \mathbb{R},\ t > 0 \tag{9.6.25}$$

along with initial condition

$$v(x, t) = v_0(x), \quad x \in \mathbb{R} \tag{9.6.26}$$

and a three-point difference scheme

$$u_k^{n+1} = Q(u_{k-1}^n, u_k^n, u_{k+1}^n). \tag{9.6.27}$$

Consider a grid point $(k\Delta x, (n+1)\Delta t)$. As we showed in the linear case, the numerical domain of dependence of difference scheme (9.6.27) is the interval $[(k-n-1)\Delta x, (k+n+1)\Delta x]$. Recall that if the solution v to initial-value problem (9.6.25)–(9.6.26) is sufficiently smooth, v can be written as $v(x,t) = v_0(x - F'(v(x,t))t)$. Again, as we did for the linear CFL condition, we say that the CFL condition is satisfied if the analytic domain of dependence is contained in the numerical domain of dependence, i.e.,

$$(k-n-1)\Delta x \le k\Delta x - F'(v(k\Delta x, (n+1)\Delta t))(n+1)\Delta t \le (k+n+1)\Delta x$$

or

$$-1 \le -RF'(v(k\Delta x, (n+1)\Delta t)) \le 1$$

or

$$R|F'(v(k\Delta x, (n+1)\Delta t))| \le 1. \tag{9.6.28}$$

If we require R to satisfy

$$R \max |F'| \le 1, \tag{9.6.29}$$

then condition (9.6.28) is satisfied for all k and n. CFL conditions for the two point forward and backward difference schemes are given in HW9.6.4. Extensions of the CFL condition to more general equations follows in a similar manner.

There are several differences between the linear and nonlinear CFL conditions. If the linear CFL condition is not satisfied, we can easily describe an initial condition that will show that the scheme is not convergent. It is not as clear that the nonlinear CFL condition is necessary for convergence. Since we do not know v, we cannot decide whether or not condition (9.6.28) is satisfied. If the nonlinear CFL condition (9.6.29) is not satisfied, the values for which it is not satisfied may never be attained by the solution v. However, if this can happen, it is the exception rather than the rule, and we will treat CFL condition (9.6.29) as a necessary condition for convergence, i.e., for just about all of our schemes, we will assume that the CFL condition is satisfied (along with other, possible more stringent conditions) and generally strive to obtain conditions for convergence that are near or equal to the restriction imposed by the CFL condition.

Often, we need and/or want to consider the discrete version of the CFL condition rather than condition (9.6.29). The discrete, nonlinear CFL condition is given by

$$R \max_{k,n} |a_{k+1/2}^n| \le 1 \tag{9.6.30}$$

where $a^n_{k+1/2}$ is defined in (9.4.17). It should be clear that CFL conditions (9.6.29) and (9.6.30) are approximately the same but are definitely not equivalent. There are times when we will assume that either CFL condition (9.6.29) or (9.6.30), but not both, is satisfied, and there will be times when for technical reasons it will be necessary to assume that both CFL conditions (9.6.29) and (9.6.30) are satisfied.

When we consider a K-system conservation law, it is more difficult to motivate the CFL condition. Recall that for linear systems of hyperbolic partial differential equations, the CFL condition is satisfied for a three-point scheme if $R|\nu_j| \leq 1$ for $j = 1, \ldots, K$. For a K-system conservation law, the CFL condition is satisfied if $R \max |\nu_j| \leq 1$, where the maximum is taken for $j = 1, \ldots, K$ and over all $\mathbf{v}$ (remembering that $\nu_j = \nu_j(\mathbf{v})$).

HW 9.6.4 (a) Show that the CFL condition for a difference scheme given by $u^{n+1}_k = \mathcal{Q}(u^n_k, u^n_{k+1})$ is $0 \leq R \max F' \leq 1$.
(b) Show that the CFL condition for a difference scheme given by $u^{n+1}_k = \mathcal{Q}(u^n_{k-1}, u^n_k)$ is $-1 \leq R \max F' \leq 0$.

9.6.5 Entropy

In Section 9.6.2 we found that if we use conservative schemes, then we are assured that if the solutions of our difference schemes converge, they will converge to a weak solution of our conservation law. If we return to Example 9.5.2, we see that even though the MacCormack scheme is a conservative scheme, it converges to a solution that is not the vanishing viscosity solution, i.e., it converges to a solution that does not satisfy the entropy conditions. Hence, it is clear that though we want our schemes to be conservative schemes, we need more.

Recall that in Section 9.3 if the scalar valued, convex function $S = S(\mathbf{v})$ and the scalar valued function $\Phi = \Phi(\mathbf{v})$ satisfied

$$\Phi'(\mathbf{v}) = \mathbf{F}'(\mathbf{v}) S'(\mathbf{v}), \tag{9.6.31}$$

then S and Φ were said to be entropy and entropy flux functions, respectively, consistent with conservation law (9.6.4). Then if a weak solution to conservation law (9.6.4) $\mathbf{v}$ satisfies

$$S(\mathbf{v})_t + \Phi(\mathbf{v})_x \leq 0$$

in the weak sense, $\mathbf{v}$ is said to satisfy Entropy Condition $\mathrm{II_v}$, i.e., if $\mathbf{v}$ is a solution to equation (9.3.4) that satisfies inequality (9.3.17), then $\mathbf{v}$ is said to satisfy Entropy Condition $\mathrm{II_v}$. We also saw in Section 9.6.2 that if the shock is a weak shock, then Entropy Conditions $\mathrm{I_v}$ and $\mathrm{II_v}$ are equivalent, and if $\mathbf{v}$ satisfies either of these conditions, $\mathbf{v}$ will be the vanishing viscosity solution. And finally, we recall from Sections 9.2.4 and 9.6.2 that for scalar

equations it is relatively easy to find entropy and flux functions, whereas for K-system conservation laws, entropy and entropy flux functions may not exist. The situation is not ideal, but we will use the above situation to obtain some nice results.

The approach we shall take is to define the **numerical entropy flux function** $\Psi^n_{k+1/2}$ based on the scalar valued function Ψ (and refer to both $\Psi^n_{k+1/2}$ and Ψ as the numerical entropy flux functions), where $\Psi^n_{k+1/2} = \Psi(\mathbf{u}^n_k, \mathbf{u}^n_{k+1})$ or $\Psi^n_{k+1/2} = \Psi(\mathbf{u}^n_{k-p}, \ldots, \mathbf{u}^n_{k+q})$ depending on whether we have a three-point scheme or a multipoint scheme. We require that the numerical entropy flux function Ψ be consistent with the entropy flux function of the conservation law in that $\Psi(\mathbf{u}, \ldots, \mathbf{u}) = \Phi(\mathbf{u})$. We obtain the following theorem.

Theorem 9.6.6 *Let S and Φ be entropy and entropy flux functions that are consistent with conservation law (9.6.4) and let $\Psi^n_{k+1/2}$ be a numerical entropy flux function that is consistent with the entropy flux function Φ. In addition, suppose that a solution to the difference scheme satisfies the discrete entropy condition*

$$S(\mathbf{u}^{n+1}_k) \leq S(\mathbf{u}^n_k) - R\left[\Psi^n_{k+1/2} - \Psi^n_{k-1/2}\right]. \tag{9.6.32}$$

If $\mathbf{u}^n_k \to \mathbf{v}$ in $L_{1,loc}$ as $\Delta x,\ \Delta t \to 0$, then $\mathbf{v}$ satisfies Entropy Condition II_V.

Proof: We proceed as we did in the proof of the Lax-Wendroff Theorem, multiply inequality (9.6.32) by a nonnegative discretized test function ϕ^n_k and sum over k and n to get

$$\sum_{n=0}^{\infty}\sum_{k=-\infty}^{\infty}\left\{\phi^n_k\left[S(\mathbf{u}^{n+1}_k) - S(\mathbf{u}^n_k) + R\left[\Psi^n_{k+1/2} - \Psi^n_{k-1/2}\right]\right]\right\} \leq 0. \tag{9.6.33}$$

If we now divide inequality (9.6.33) by Δt and use the summation by parts rule (9.6.13), we get

$$-\sum_{n=0}^{\infty}\sum_{k=-\infty}^{\infty}\left\{S(\mathbf{u}^{n+1}_k)\frac{\phi^{n+1}_k - \phi^n_k}{\Delta t} + \Psi^n_{k+1/2}\frac{\phi^n_{k+1} - \phi^n_k}{\Delta x}\right\} - \sum_{k=-\infty}^{\infty}\frac{S(\mathbf{u}^0_k)\phi^0_k}{\Delta t} \leq 0, \tag{9.6.34}$$

where again all of the boundary terms except the term at $n = 0$ disappear because ϕ has compact support. If we multiply equation (9.6.34) by $\Delta x \Delta t$ and let $\Delta x,\ \Delta t \to 0$, we get

$$-\int_0^{\infty}\int_{-\infty}^{\infty}\left\{S(\mathbf{v})\phi_t + \Psi(\mathbf{v}, \ldots, \mathbf{v})\phi_x\right\} dx, dt - \int_{-\infty}^{\infty} S(\mathbf{v}_0)\phi(x, 0)\, dx \leq 0. \tag{9.6.35}$$

Since Ψ was assumed to be consistent with Φ, inequality (9.6.35) gives us the weak form of Entropy Condition $\mathrm{II_v}$

$$-\int_0^\infty \int_{-\infty}^{\infty} \left\{ S(\mathbf{v})\phi_t t + \Phi(\mathbf{v})\phi_x \right\} dx\, dt - \int_{-\infty}^{\infty} S(\mathbf{v}_0)\phi(x,0)\, dx \leq 0, \quad (9.6.36)$$

which we were to prove.

Remark 1: We note that since we saw in Example 9.5.2 that the MacCormack scheme, (9.5.4)–(9.5.5), was unable to break the expansion shock into an expansion wave, the MacCormack scheme does not satisfy the discrete entropy condition for any entropy function S and numerical entropy flux function Ψ. If the scheme satisfied the discrete entropy condition, then Theorem 9.6.6 implies that the limiting solution would satisfy the analytic entropy condition (Entropy Condition II), which the solution in Example 9.5.2 does not.

Remark 2: The computation done in HW9.4.9 is similar to that done in Example 9.5.2 in that we find that the upwind scheme (9.4.16) cannot totally break the expansion fan. The difference is that the scheme partially breaks the fan, and the solution is very interesting looking. Nevertheless, the computation in HW9.4.9 shows that upwind scheme (9.4.16) will converge to a weak solution that is not the entropy solution. Hence, upwind scheme (9.4.16) will not satisfy the discrete entropy condition for any entropy function S and numerical entropy flux function Ψ. We might also add that when we consider a linear conservation law, there is no problem with upwind scheme (9.4.16). This is true because there are no expansion shocks in the linear problems.

Remark 3: Another example of a very common difference scheme that is unable to break the expansion fan is the Lax-Wendroff scheme (9.5.10). As we see in HW9.6.6, the computation does not produce the desired expansion fan and is a very interesting solution. The reader might study the solution to be convinced that the solution is actually a weak solution.

Remark 3: We note that we do not discuss a relationship between the numerical entropy flux function and the difference scheme. The relationship is given implicitly by the fact that the solution to the difference scheme $\mathbf{u}_k^n$ must satisfy inequality (9.6.32).

We will not generally be interested in numerical entropy flux functions for specific schemes. We will attempt to find classes of schemes that will converge to the entropy solution. However, below we include two examples in which we find the numerical entropy flux function associated with the FTBS schemes.

Example 9.6.1 Consider the one-way wave equation $v_t + av_x = 0$, $a > 0$, along with the entropy function and entropy flux functions given in Remark 2, page 100,

$$S(v) = |v - \mathrm{c}| \text{ and } \Phi(v) = \frac{v - c}{|v - c|}[F(v) - F(\mathrm{c})]$$

and the FTBS difference scheme $u_k^{n+1} = u_k^n - aR\delta_- u_k^n$ where $R = \Delta t/\Delta x$. Show that if CFL condition $0 \le aR \le 1$ is satisfied, then the numerical entropy flux function

$$\Psi_{k+1/2}^n = \Psi(u_k^n, u_{k+1}^n) = a\max\{u_k^n, c\} - a\min\{u_k^n, c\} \tag{9.6.37}$$

satisfies inequality (9.6.32) and is consistent with entropy flux function Φ.

Solution: Because for our conservation law we have $F(v) = av$, we see that

$$\Phi(v) = \frac{v-c}{|v-c|}\big[av - ac\big] = a|v-c|.$$

Then, since

$$\begin{aligned}\Psi(u,u) &= a\max\{u,c\} - a\min\{u,c\} \\ &= \begin{cases} au - ac = a|u-c| & \text{if } u > c \\ ac - au = -a(u-c) = a|u-c| & \text{if } u < c, \end{cases}\end{aligned}$$

we see that

$$\Psi(u,u) = a|u-c| = \Phi(u)$$

and Ψ is consistent with Φ.

We next note that $S(u_k^{n+1})$ can be written as

$$S(u_k^{n+1}) = |u_k^{n+1} - c| = \max\{u_k^{n+1}, c\} - \min\{u_k^{n+1}, c\}.$$

We see that

$$\begin{aligned}u_k^{n+1} &= (1-aR)u_k^n + aRu_{k-1}^n \\ &\le (1-aR)\max\{u_k^n, c\} + aR\max\{u_{k-1}^n, c\}\end{aligned} \tag{9.6.38}$$

and

$$\begin{aligned}c &= (1-aR)c + aRc \\ &\le (1-aR)\max\{u_k^n, c\} + aR\max\{u_{k-1}^n, c\}.\end{aligned} \tag{9.6.39}$$

We note that both of the above inequalities depend heavily on the fact that $0 \le aR \le 1$. Using (9.6.38) and (9.6.39), we see that

$$\max\{u_k^{n+1}, c\} \le (1-aR)\max\{u_k^n, c\} + aR\max\{u_{k-1}^n, c\}. \tag{9.6.40}$$

Using the analogous approach with the minimum function, we find that

$$\min\{u_k^{n+1}, c\} \ge (1-aR)\min\{u_k^n, c\} + aR\min\{u_{k-1}^n, c\}. \tag{9.6.41}$$

Then

$$\begin{aligned}S(u_k^{n+1}) &= \max\{u_k^{n+1}, c\} - \min\{u_k^{n+1}, c\} \\ &\le (1-aR)\max\{u_k^n, c\} + aR\max\{u_{k-1}^n, c\} \\ &\quad -\big[(1-aR)\min\{u_k^n, c\} + aR\min\{u_{k-1}^n, c\}\big] \\ &= \max\{u_k^n, c\} - \min\{u_k^n, c\} \\ &\quad - R\Big\{\big[a\max\{u_k^n, c\} - a\min\{u_k^n, c\}\big] - \big[a\max\{u_{k-1}^n, c\} - a\min\{u_{k-1}^n, c\}\big]\Big\} \\ &= S(u_k^n) - R\Big\{\Psi_{k+1/2}^n - \Psi_{k-1/2}^n\Big\}.\end{aligned}$$

Therefore, if $u_k^n \to v$ as $\Delta x,\ \Delta t \to 0$, then v will be the vanishing viscosity solution.

Example 9.6.2 For Burgers' equation along with entropy function and entropy flux function

$$S(v) = |v - c|, \ \Phi(v) = \frac{v - c}{|v - c|}\big[F(v) - F(c)\big],$$

find a numerical entropy flux function Ψ that will be consistent with entropy flux function Φ and satisfy inequality (9.6.32) for the FTBS scheme (9.4.2).

Solution: We want to follow the approach used in Example 9.6.1. If we write difference scheme (9.4.2) as

$$u_k^{n+1} = u_k^n - \frac{R}{2}\big(u_k^n\big)^2 + \frac{R}{2}\big(u_{k-1}^n\big)^2$$

and use HW9.6.5, we see that

$$u_k^{n+1} \le \max\{u_k^n, c\} - \frac{R}{2}\big(\max\{u_k^n, c\}\big)^2 + \frac{R}{2}\big(\max\{u_{k-1}^n, c\}\big)^2 \tag{9.6.42}$$

if $0 \le R\max\{u_k^n, c\} \le 1$ and $0 \le u_{k-1}^n$. We note that these inequalities are satisfied as long as the CFL condition for difference scheme (9.4.2) is satisfied, $0 \le RF' \le 1$, and if we require that c satisfy $0 \le cR \le 1$. We further note that this condition on c along with the CFL condition implies that c satisfies

$$c \le \max\{u_k^n, c\} - \frac{R}{2}\big(\max\{u_k^n, c\}\big)^2 + \frac{R}{2}\big(\max\{u_{k-1}^n, c\}\big)^2. \tag{9.6.43}$$

This condition on c restricts our choice of c in the definition of our entropy and entropy flux functions. As we did in Example 9.6.1, we combine inequalities (9.6.42) and (9.6.43) to get

$$\max\{u_k^{n+1}, c\} \le \max\{u_k^n, c\} - \frac{R}{2}\big(\max\{u_k^n, c\}\big)^2 + \frac{R}{2}\big(\max\{u_{k-1}^n, c\}\big)^2. \tag{9.6.44}$$

And again as we did in Example 9.6.1, we use a similar analysis to get

$$\min\{u_k^{n+1}, c\} \ge \min\{u_k^n, c\} - \frac{R}{2}\big(\min\{u_k^n, c\}\big)^2 + \frac{R}{2}\big(\min\{u_{k-1}^n, c\}\big)^2. \tag{9.6.45}$$

Thus if we define $\Psi_{k+1/2}^n$ by

$$\Psi_{k+1/2}^n = \Psi(u_k^n, u_{k+1}^n) = \frac{1}{2}\big(\max\{u_k^n, c\}\big)^2 - \frac{1}{2}\big(\min\{u_k^n, c\}\big)^2,$$

we see that inequalities (9.6.44) and (9.6.45) can be combined to show that the solution to difference scheme (9.4.2) satisfies

$$S(u_k^{n+1}) \le S(u_k^n) - R\big[\Psi_{k+1/2}^n - \Psi_{k-1/2}^n\big]$$

if $0 \le RF'(u) \le 1$ for $u = c$ or for $u = u_k^n$ for any n and k.

For this problem, Φ is given by $\Phi(v) = |v - c|(v + c)$ and $F(v) = v^2/2$. We see that

$$\begin{aligned}\Psi(u, u) &= F(\max\{u, c\}) - F(\min\{u, c\}) \\ &= \begin{cases} F(u) - F(c) & \text{if } u > c \\ F(c) - F(u) & \text{if } u < c \end{cases} \\ &= \Phi(u).\end{aligned}$$

Hence, the numerical flux function Ψ is consistent with the flux function Φ.

HW 9.6.5 (a) Show that if $a \le b$, then

$$a - \frac{R}{2}a^2 \le b - \frac{R}{2}b^2$$

if $bR \le 1$.

(b) Show that if $a \leq b$, then

$$\frac{R}{2}a^2 \leq \frac{R}{2}b^2$$

if $a \geq 0$.

HW 9.6.6 Use Lax-Wendroff scheme (9.5.10) to approximate the solution to the inviscid Burgers' equation (9.4.18) with initial condition

$$v(x,0) = v_0(x) = \begin{cases} -1 & x \leq 0 \\ 2 & x > 0 \end{cases}$$

and numerical boundary conditions $u_0^n = -1.0$ and $u_M^n = 2.0$.

9.7 Difference Schemes for Scalar Conservation Laws

We should realize that the results given in Section 9.6 apply to K-system conservation laws. We can and will do more concerning K-system conservation laws. However, we now take a break and consider numerical schemes for scalar conservation laws. Of course, we have already considered several schemes for scalar conservation laws. We should emphasize that some of the difficulties with our solution methods that we must still resolve are to ensure that the solutions that we obtain will converge to the vanishing viscosity solution and to find schemes that will give us neither the dispersive wiggles of the Lax-Wendroff scheme nor the severe smearing of the Lax-Friedrichs scheme.

9.7.1 Definitions

We consider the scalar conservation law

$$v_t + F(v)_x = 0 \tag{9.7.1}$$

and will generally have conservative difference schemes of the form

$$u_k^{n+1} = u_k^n - R\left[h_{k+1/2}^n - h_{k-1/2}^n\right], \tag{9.7.2}$$

where $R = \Delta t/\Delta x$. We begin by introducing several definitions concerning conservative difference schemes. We will then follow these definitions with a series of results concerning and relating these ideas. Many of the results are somewhat negative (less than we want, anyway), so we will just state some of the results. The approach will be to systematically continue to

refine the schemes until they produce difference schemes that we want and need.

We begin by introducing the class of difference schemes called **entropy, or E, schemes**, see ref. [54].

Definition 9.7.1 *Difference scheme (9.7.2) is called an E scheme if*

$$\begin{aligned} h_{k+1/2} &\le F(u) \text{ for all } u \in [u_k, u_{k+1}] \text{ if } u_k < u_{k+1} \\ h_{k+1/2} &\ge F(u) \text{ for all } u \in [u_{k+1}, u_k] \text{ if } u_{k+1} < u_k. \end{aligned} \tag{9.7.3}$$

Remark: We should note that condition (9.7.3) is a strong condition in that the condition must be satisfied for all values of u between u_k and u_{k+1}. If F is convex, then condition (9.7.3) will be satisfied for all u between u_k and u_{k+1} if it is satisfied at both u_k and u_{k+1}. If F is not convex, this may not be the case.

Example 9.7.1 Show that the FTFS scheme (9.4.1) is an E scheme.

Solution: For the FTFS scheme (9.4.1), $h^n_{k+1/2} = F^n_{k+1} = F(u^n_{k+1})$. If the scheme is to satisfy the CFL condition, we must have $-1 \le RF' \le 0$. Hence, F' must be negative and F is decreasing. Then if $u_k < u_{k+1}$, for $u \in [u_k, u_{k+1}]$, we have $h(u_k, u_{k+1}) = F(u_{k+1}) \le F(u)$. If $u_{k+1} < u_k$, then for $u \in [u_{k+1}, u_k]$, we have $h(u_k, u_{k+1}) = F(u_{k+1}) \ge F(u)$. Therefore, the FTFS scheme is an E scheme.

Generally, it is difficult to decide whether or not a difference scheme is an E scheme. The advantage of monotone schemes is that it is very easy to determine whether or not a given scheme is monotone. And like E schemes, the class of monotone schemes has some very nice properties. We make the following definition.

Definition 9.7.2 *An difference scheme of the form*

$$u^{n+1}_k = \mathcal{Q}(u^n_{k-p-1}, \cdots, u^n_{k+q}) \tag{9.7.4}$$

is said to be monotone if the function $\mathcal{Q}$ is a monotone increasing function with respect to each of its arguments.

Remark: We should note that if $\mathcal{Q}$ is differentiable with respect to its arguments (which for most difference schemes would be the case), then difference scheme (9.7.4) is monotone if and only if

$$\frac{\partial \mathcal{Q}(u_{-(p+1)}, \cdots, u_q)}{\partial u_j} \ge 0 \text{ for all } j, \ -(p+1) \le j \le q.$$

If we consider the linear FTBS scheme $u^{n+1}_k = \mathcal{Q}(u^n_{k-1}, u^n_k) = u^n_k - aR\delta_- u^n_k$ (where here $R = \Delta t/\Delta x$), then $\partial \mathcal{Q}/\partial u^n_{k-1} = aR$ and $\partial \mathcal{Q}/\partial u^n_k = 1 - aR$. We see that the FTBS scheme will be monotone when $aR \ge 0$ and

$1 - aR \geq 0$. Recall from Section 5.3 that the FTBS scheme is stable for $0 \leq aR \leq 1$. Hence, the FTBS scheme is monotone for $0 \leq aR \leq 1$, and the range of aR over which the FTBS scheme is monotone is the same as its stability range.

If we instead consider the linear Lax-Wendroff scheme, (5.3.8),

$$u_k^{n+1} = \mathcal{Q}(u_{k-1}^n, u_k^n, u_{k+1}^n) = u_k^n - \frac{aR}{2}\delta_0 u_k^n + \frac{a^2R^2}{2}\delta^2 u_k^n,$$

we see that $\partial\mathcal{Q}/\partial u_{k-1}^n = \frac{aR}{2} + \frac{a^2R^2}{2}$, $\partial\mathcal{Q}/\partial u_k^n = 1 - a^2R^2$ and $\partial\mathcal{Q}/\partial u_{k-1}^n = -\frac{aR}{2} + \frac{a^2R^2}{2}$. For the Lax-Wendroff scheme to be monotone, we need

$$\frac{aR}{2} + \frac{a^2R^2}{2} \geq 0,$$

which implies $aR = 0$; $aR > 0$ and $aR \geq -1$; or $aR < 0$ and $aR \leq -1$,

$$1 - a^2R^2 \geq 0,$$

which implies that $-1 \leq aR \leq 1$, and

$$-\frac{aR}{2} + \frac{a^2R^2}{2} \geq 0,$$

which implies $aR = 0$; $aR > 0$ and $aR \geq 1$; or $aR < 0$ and $aR \leq 1$.

In other words, we must have $aR = 0$; $aR > 0$, $0 < aR \leq 1$, and $aR \geq 1$; or $aR < 0$, $aR \leq -1$, and $-1 \leq aR < 0$. Of course, the only one of these that is possible is $aR = 0$ (in which we are not interested) or $aR = \pm 1$. Since there is no range of monotonicity, we say that the linear Lax-Wendroff scheme is not monotone.

If we now consider the Lax-Friedrichs scheme for conservation law (9.7.1),

$$\begin{aligned} u_k^{n+1} &= \mathcal{Q}(u_{k-1}^n, u_k^n, u_{k+1}^n) \\ &= \frac{1}{2}(u_{k-1}^n + u_{k+1}^n) - \frac{R}{2}\left(F_{k+1}^n - F_{k-1}^n\right), \end{aligned} \tag{9.7.5}$$

we see that $\partial\mathcal{Q}/\partial u_{k-1}^n = \frac{1}{2} + \frac{1}{2}RF'(u_{k-1}^n)$, $\partial\mathcal{Q}/\partial u_k^n = 0$ and $\partial\mathcal{Q}/\partial u_{k+1}^n = \frac{1}{2} - \frac{1}{2}RF'(u_{k+1}^n)$. If we assume that the Lax-Friedrichs scheme satisfies the CFL condition $R|F'| \leq 1$ (which is also necessary for convergence), then $\left(1 \pm RF'(u_{k\pm1}^n)\right)/2 \geq 0$, and the Lax-Friedrichs scheme (9.7.5) is monotone.

The next concept that we shall introduce is that of total variation decreasing schemes. It should be clear that in Example 9.5.3 the initial condition was very smooth (except for the jump at the origin) and the wiggles were introduced into the solution by the difference scheme. We can stop this from happening by requiring that the total variation of our solution does not increase from time step to time step (and then try to find a difference scheme that might satisfy that condition). We make the following definition.

	u_0^n	u_1^n	u_2^n	u_3^n	u_4^n	u_5^n	u_6^n	u_7^n	u_8^n
$n=0$	1.0	1.0	1.0	1.0	1.0	0.5	0.5	0.5	0.5
$n=1$	1.0	1.0	1.0	1.0	1.0623	0.7002	0.5	0.5	0.5

TABLE 9.7.1. Initial conditions and solution values at $t = \Delta t$ for the problem solved in Example 9.5.3 found by using the Lax-Wendroff scheme (9.4.8) with $\Delta x = 0.25$ and $\Delta t = 0.124$.

Definition 9.7.3 *A difference scheme is said to be total variation decreasing (TVD) if the solution produced by the scheme satisfies*

$$\sum_{k=-\infty}^{\infty} |\delta_+ u_k^{n+1}| \leq \sum_{k=-\infty}^{\infty} |\delta_+ u_k^n| \tag{9.7.6}$$

for all $n \geq 0$.

If we denote the **total variation** of a grid function by

$$TV(\mathbf{u}) = \sum_{k=-\infty}^{\infty} |\delta_+ u_k|,$$

then the inequality in Definition 9.7.3 can be expressed as $TV(\mathbf{u}^{n+1}) \leq TV(\mathbf{u}^n)$. If we return to Example 9.5.3 and apply the Lax-Wendroff scheme (9.4.8) with $\Delta x = 0.25$ and $\Delta t = 0.125$, at the first time step $t = \Delta t$ we obtain the solution given in Table 9.7.1. It is easy to see that

$$TV(\mathbf{u}^0) = \sum_{k=0}^{9} |\delta_+ u_k^0| = 0.5,$$

while

$$TV(\mathbf{u}^1) = \sum_{k=0}^{9} |\delta_+ u_k^1| = 0.6246.$$

Therefore, the Lax-Wendroff scheme (9.4.8) is not TVD. This should give us hope that TVD schemes will help eliminate the dispersive wiggles that we have seen so often when computing discontinuities.

It is possible to use Definition 9.7.3 above to show that a scheme is TVD. Consider the FTBS for the one-way wave equation with $a > 0$,

$$u_k^{n+1} = u_k^n - aR\delta_- u_k^n.$$

Given that we want to use one of the one sided schemes, we choose the FTBS scheme because it is the one sided scheme that is stable for $a > 0$.

We assume that the CFL condition $0 \le aR \le 1$ (which is also the stability condition) is satisfied. We then note that

$$\begin{aligned}
TV(\mathbf{u}^{n+1}) &= \sum_{k=-\infty}^{\infty} |\delta_+ u_k^{n+1}| \\
&= \sum_{k=-\infty}^{\infty} |\delta_+ u_k^n - aR\delta_+\delta_- u_k^n| \\
&= \sum_{k=-\infty}^{\infty} |\delta_+ u_k^n - aR\delta_-\delta_+ u_k^n| \\
&= \sum_{k=-\infty}^{\infty} |(1-aR)\delta_+ u_k^n + aR\delta_+ u_{k-1}^n| \text{ (use the triangular inequality} \\
&\quad \text{and the fact that } 1-aR \ge 0 \text{ and } aR \ge 0) \\
&\le (1-aR) \sum_{k=-\infty}^{\infty} |\delta_+ u_k^n| + aR \sum_{k=-\infty}^{\infty} |\delta_+ u_{k-1}^n| \\
&= TV(\mathbf{u}^n) \quad (\text{let } j = k-1 \text{ in the second sum}).
\end{aligned}$$

Hence, we see that the FTBS scheme is TVD.

Sometimes, to require our scheme to be TVD is too much. For this reason we include the following definition.

Definition 9.7.4 *A difference scheme is said to be essentially nonoscillatory (ENO) if the solution produced by the scheme satisfies*

$$\sum_{k=-\infty}^{\infty} |\delta_+ u_k^{n+1}| \le \sum_{k=-\infty}^{\infty} |\delta_+ u_k^n| + \mathcal{O}(\Delta x^p) \tag{9.7.7}$$

for all $n \ge 0$ and some p.

Before we proceed to the theorems, we include two more (other than the conservative form given by (9.7.2)) special forms for writing our difference schemes. As we will see, there are different times that these various standard forms are convenient. If we write the difference scheme in the form

$$u_k^{n+1} = u_k^n + C_{k+1/2}^n \delta_+ u_k^n - D_{k-1/2}^n \delta_- u_k^n \tag{9.7.8}$$

where $C_{k+1/2}^n$ and $D_{k-1/2}^n$ depend on u_k^n and its neighboring values, we say that our difference equation is in **incremental form**, or **I-form**. We note that if we have a difference scheme given in incremental form, the scheme is conservative, and the numerical flux function associated with that scheme can be written in terms of C and D by

$$h_{k+1/2}^n = F(u_k^n) - \frac{1}{R} C_{k+1/2}^n \delta_+ u_k^n = F(u_{k+1}^n) - \frac{1}{R} D_{k+1/2}^n \delta_+ u_k^n. \tag{9.7.9}$$

If we have the numerical flux function h associated with a given difference scheme, we can write the scheme in I-form by defining C and D as

$$C^n_{k+1/2} = -R\frac{h^n_{k+1/2} - F(u^n_k)}{\delta_+ u^n_k} \tag{9.7.10}$$

$$D^n_{k+1/2} = -R\frac{h^n_{k+1/2} - F(u^n_{k+1})}{\delta_+ u^n_k}. \tag{9.7.11}$$

We note that since the $C^n_{k+1/2}$ and $D^n_{k-1/2}$ are multiplied by $\delta_+ u^n_k$ and $\delta_- u^n_k$ (which is the same as $\delta_+ u^n_{k-1}$), respectively, the $\delta_+ u^n_k$ term in the denominator does not cause any real problems in equations (9.7.10) and (9.7.11). When $\delta_+ u^n_k$ is zero, $C^n_{k+1/2}$ and $D^n_{k+1/2}$ can be defined to be anything.

Remark: It is easy to see that the linear FTFS scheme, (5.3.1), can be written in I-form by letting $C^n_{k+1/2} = -aR$ and $D^n_{k+1/2} = 0$. The linear Lax-Wendroff scheme, (5.3.8), can be written in I-form by setting $C^n_{k+1/2} = -Ra/2 + a^2R^2/2$ and $D^n_{k+1/2} = aR/2 + a^2R^2/2$, and the nonlinear FTFS scheme (9.4.1) can be written in I-form with $C^n_{k+1/2} = -Ra^n_{k+1/2}$ and $D^n_{k+1/2} = 0$. We should note that the above choices for $C^n_{k+1/2}$ and $D^n_{k+1/2}$ are not the only choices that could be made. As we will see later, the I-form representation of a difference scheme is not unique.

If we write the difference scheme as

$$u^{n+1}_k = u^n_k - \frac{R}{2}\delta_0 F^n_k + \frac{1}{2}\delta_+(Q^n_{k-1/2}\delta_- u^n_k), \tag{9.7.12}$$

where we call Q the **numerical viscosity coefficient**, then the scheme is said to be in **Q-form**. We see that if we are given a numerical viscosity coefficient Q and a difference scheme in Q-form, then the scheme is conservative and the numerical flux function for the scheme can be written as

$$h^n_{k+1/2} = \frac{1}{2}\left(F^n_k + F^n_{k+1}\right) - \frac{1}{2R}Q^n_{k+1/2}\delta_+ u^n_k. \tag{9.7.13}$$

By setting

$$C^n_{k+1/2} = \frac{1}{2}\left(Q^n_{k+1/2} - Ra^n_{k+1/2}\right) \tag{9.7.14}$$

$$D^n_{k+1/2} = \frac{1}{2}\left(Q^n_{k+1/2} + Ra^n_{k+1/2}\right), \tag{9.7.15}$$

where $a^n_{k+1/2}$ is defined in (9.4.17), we transform a difference equation from Q-form to I-form. And if we are given a difference scheme in conservative form or I-form, we can define the numerical viscosity coefficient Q by

$$Q^n_{k+1/2} = R\frac{F(u^n_k) + F(u^n_{k+1}) - 2h^n_{k+1/2}}{\delta_+ u^n_k} \tag{9.7.16}$$

or

$$Q^n_{k+1/2} = C^n_{k+1/2} + D^n_{k+1/2}, \tag{9.7.17}$$

and rewrite the difference scheme in Q-form. Equation (9.7.16) should bother us in that we are dividing by zero when $\delta_+ u^n_k = 0$. When $\delta_+ u^n_k = 0$, the right side of equation (9.7.16) is zero over zero. More importantly, when $\delta_+ u^n_k = 0$, it makes no difference how we define Q, because the Q term in difference scheme (9.7.12) will be zero as long as Q is defined.

Remark: As the Q-form can be considered to be a generalization of the form of the Lax-Wendroff scheme, it is easy to see that the linear Lax-Wendroff scheme, (5.3.8), and the vector form of the nonlinear Lax-Wendroff scheme, (9.4.8), can be expressed in Q-form as

$$u^{n+1}_k = u^n_k - \frac{R}{2}\delta_0 u^n_k + \frac{1}{2}\delta_+\left(a^2 R^2 \delta_- u^n_k\right)$$

and

$$\mathbf{u}^{n+1}_k = \mathbf{u}^n_k - \frac{R}{2}\delta_0 \mathbf{F}^n_k + \frac{1}{2}\delta_+\left(R^2\left(\mathbf{F}'(\mathbf{u})\right)^n_{k-1/2}\delta_- \mathbf{F}^n_k\right),$$

respectively. Also, we see that we can write the scalar Lax-Wendroff scheme, (9.5.10), in Q-form with numerical viscosity coefficient

$$Q^n_{k+1/2} = R^2\left(a^n_{k+1/2}\right)^2.$$

The nonlinear FTFS scheme, (9.4.1), and the upwind scheme, (9.4.16), can be written in Q-form with Q given by

$$Q^n_{k+1/2} = -Ra^n_{k+1/2}$$

and

$$Q^n_{k+1/2} = |Ra^n_{k+1/2}|,$$

respectively. We note above that in all of the examples above, the numerical viscosity Q can be written as a function of $Ra^n_{k+1/2}$, i.e., $Q = Q(Ra^n_{k+1/2})$. We will use this property of Q in Section 9.7.7.

HW 9.7.1 Find conditions that will ensure that the linear FTFS scheme, (5.3.1), will be a monotone scheme.

HW 9.7.2 (a) Show that if the CFL condition $0 \le RF' \le 1$ is satisfied, the nonlinear FTBS scheme (9.4.2) is monotone.
(b) Show that if the CFL condition $-1 \le RF' \le 0$ is satisfied, then the nonlinear FTFS scheme (9.4.1) is monotone.

HW 9.7.3 Show that transformation (9.7.9) will transform a difference scheme in I-form to conservative form.

HW 9.7.4 Verify that transformation (9.7.10)–(9.7.11) will transform a difference scheme from conservative form to I-form.

HW 9.7.5 Verify that transformations (9.7.13) and (9.7.14)–(9.7.15) will transform our difference schemes from Q-form to conservative form and I-form, respectively.

HW 9.7.6 Show that using transformations (9.7.16) and (9.7.17), we can transform our difference schemes from conservative form and I-form, respectively, to Q-form.

HW 9.7.7 Show by example (as we did for the Lax-Wendroff scheme) that the Beam-Warming scheme, (9.4.13)–(9.4.14), is not TVD.

HW 9.7.8 Find the numerical viscosity coefficient Q associated with the Lax-Friedrichs scheme (9.4.3).

9.7.2 Theorems

We are now ready to discuss some results concerning numerical schemes for scalar conservation laws. Specifically, we will give some basic results concerning E schemes, monotone schemes and TVD schemes. We will prove some of these results and just state some of the others. As we shall see, a number of these results are negative results—telling us that certain schemes are not good enough. However, we should appreciate that it is also very important to know what does not work. In addition, knowing where and why certain schemes are not good enough will help us develop better schemes. We begin by stating a proposition that will ensure, via Theorem 9.6.6, that in the limit, E schemes will produce the vanishing viscosity solution. The proof of this proposition along with the numerical entropy flux function (which is difficult) is given in ref. [67], page 377.

Proposition 9.7.5 *Consider conservation law (9.7.1) along with the associated convex entropy function S and entropy flux function Φ. Let u_k^n be an approximation to the solution to an initial-value problem involving conservation law (9.7.1) found by using an E scheme of the form given in (9.7.2). Then if the solution u_k^n satisfies the condition*

$$R|(F_k^n - h_{k+1/2}^n) + (F_{k+1}^n - h_{k+1/2}^n)| \leq \frac{1}{2}|u_{k+1}^n - u_k^n|, \qquad (9.7.18)$$

there exists a numerical entropy flux function $\Psi_{k+1/2}^n$ such that u_k^n satisfies the discrete entropy condition

$$S(u_k^{n+1}) \leq S(u_k^n) - R\left[\Psi_{k+1/2}^n - \Psi_{k-1/2}^n\right]. \qquad (9.7.19)$$

Thus we see that based on Theorems 9.6.5 and 9.6.6, if the solution to the E scheme converges to the function v as $\Delta x,\ \Delta t \to 0$, then v will be both a weak solution and a vanishing viscosity solution to conservation law (9.7.1). We cannot ask for much more.

Remark: We should note that if we consider the E scheme in Q-form, then the CFL condition (9.7.18) is equivalent to the condition $|Q^n_{k+1/2}| \le 1/2$.

In addition to yielding a weak entropy solution in the limit, the E schemes have another nice property in that they will not introduce dispersive wiggles into the solution. Before we can prove this, we state and prove the following result, which gives sufficient conditions for a scheme to be a TVD scheme. This result will be very important, since it will be the principal method for proving that a scheme is TVD.

Proposition 9.7.6 *Consider a difference scheme in I-form such as given in (9.7.8). If*

$$C^n_{k+1/2} \ge 0,\ D^n_{k+1/2} \ge 0 \text{ and } C^n_{k+1/2} + D^n_{k+1/2} \le 1, \tag{9.7.20}$$

then the scheme is TVD.

Proof: We begin by applying δ_+ to both sides of equation (9.7.8), taking the absolute value and summing from $-\infty$ to ∞ to get

$$\begin{aligned}
TV(\mathbf{u}^{n+1}) &= \sum_{k=-\infty}^{\infty} |\delta_+ u_k^{n+1}| = \sum_{k=-\infty}^{\infty} \Big| \delta_+ u_k^n + \delta_+ \left(C^n_{k+1/2} \delta_+ u_k^n \right) \\
&\quad - \delta_+ \left(D^n_{k-1/2} \delta_- u_k^n \right) \Big| \\
&= \sum_{k=-\infty}^{\infty} \Big| \delta_+ u_k^n + C^n_{k+3/2} \delta_+ u_{k+1}^n - C^n_{k+1/2} \delta_+ u_k^n \\
&\quad - D^n_{k+1/2} \delta_- u_{k+1}^n + D^n_{k-1/2} \delta_- u_k^n \Big| \\
&= \sum_{k=-\infty}^{\infty} \Big| \delta_+ u_k^n + C^n_{k+3/2} \delta_+ u_{k+1}^n - C^n_{k+1/2} \delta_+ u_k^n - D^n_{k+1/2} \delta_+ u_k^n \\
&\quad + D^n_{k-1/2} \delta_+ u_{k-1}^n \Big| \text{ (because } \delta_- u_{k+1}^n = \delta_+ u_k^n \text{ and } \delta_- u_k^n = \delta_+ u_{k-1}^n) \\
&\le \sum_{k=-\infty}^{\infty} C^n_{k+3/2} |\delta_+ u_{k+1}^n| + \sum_{k=-\infty}^{\infty} \left(1 - C^n_{k+1/2} - D^n_{k+1/2} \right) |\delta_+ u_k^n| \\
&\quad + \sum_{k=-\infty}^{\infty} D^n_{k-1/2} |\delta_+ u_{k-1}^n| \text{ (by regrouping and using the facts} \\
&\quad \text{that } C^n_{k+1/2} \ge 0,\ D^n_{k+1/2} \ge 0 \text{ and } C^n_{k+1/2} + D^n_{k+1/2} \le 1) \\
&= \sum_{j=-\infty}^{\infty} C^n_{j+1/2} |\delta_+ u_j^n| + \sum_{k=-\infty}^{\infty} \left(1 - C^n_{k+1/2} - D^n_{k+1/2} \right) |\delta_+ u_k^n|
\end{aligned}$$

$$+ \sum_{j=-\infty}^{\infty} D^n_{j+1/2} |\delta_+ u^n_j| \text{ (making a change of index } k = j-1 \text{ in}$$

the first summation and $k = j+1$ in the last summation of the previous line)

$$= \sum_{k=-\infty}^{\infty} |\delta_+ u^n_k| = TV(\mathrm{u}^n).$$

Hence, we see that if condition (9.7.20) is satisfied, difference equation (9.7.8) is TVD.

Remark: If we return to I-form representations of the linear FTFS, linear Lax-Wendroff and nonlinear FTFS (9.4.1) schemes discussed in the Remark on page 174, we obtain the following results.

(1) (linear FTFS, (5.3.1)) The condition $C^n_{k+1/2} = -aR \geq 0$ requires that $a \leq 0$. Obviously, $D^n_{k+1/2} \geq 0$. The condition

$$C^n_{k+1/2} + C^n_{k+1/2} = -aR \leq 1$$

is the same as $-1 \leq aR$. Thus, the condition that Proposition 9.7.6 requires for the linear FTFS scheme to be TVD is that $-1 \leq aR \leq 0$, which is the same as the CFL condition for the FTFS scheme and the same as the stability condition for the scheme.

(2) (linear Lax-Wendroff, (5.3.8)) The conditions

$$C^n_{k+1/2} = -\frac{aR}{2} + \frac{a^2R^2}{2} \geq 0, \quad D^n_{k+1/2} = \frac{aR}{2} + \frac{a^2R^2}{2} \geq 0$$

and

$$C^n_{k+1/2} + D^n_{k+1/2} = a^2R^2 \leq 1$$

are the same conditions that we needed to ensure that the Lax-Wendroff scheme was a monotone scheme, i.e. the scheme will only be TVD only when $R = \pm 1$. We emphasize that this result does not imply that the Lax-Wendroff scheme is not TVD. This result only gives us the fact that Proposition 9.7.6 cannot be used to prove that the Lax-Wendroff scheme is TVD.

(3) (nonlinear FTFS (9.4.1)) Obviously, $D^n_{k+1/2} = 0 \geq 0$. The conditions that $C^n_{k+1/2} = -Ra^n_{k+1/2} \geq 0$ and $C^n_{k+1/2} + D^n_{k+1/2} = -Ra^n_{k+1/2} \leq 1$ are the same as $-1 \leq Ra^n_{k+1/2} \leq 0$. When $\delta_+ u^n_k = 0$, this condition is equivalent to $-1 \leq RF'(u^n_k) \leq 0$, which follows from the fact that we assume that the CFL condition is satisfied. When $\delta_+ u^n_k \neq 0$, this condition is

$$-1 \leq \frac{R\delta_+ F^n_k}{\delta_+ u^n_k} \leq 0.$$

This condition is also satisfied whenever the discrete CFL condition (9.6.30) is satisfied.

We next state and prove the following proposition.

Proposition 9.7.7 *A conservative E scheme that satisfies*

$$R|(h^n_{k+1/2} - F^n_k) + (h^n_{k+1/2} - F^n_{k+1})| \leq |\delta_+ u^n_k| \tag{9.7.21}$$

is TVD.

Proof: Consider an E scheme that is in conservative form (9.7.2). From (9.7.10) and (9.7.11) we see that if we were to write the scheme in I-form, then

$$C^n_{k+1/2} = -R\frac{h^n_{k+1/2} - F(u^n_k)}{\delta_+ u^n_k}$$

and

$$D^n_{k+1/2} = -R\frac{h^n_{k+1/2} - F(u^n_{k+1})}{\delta_+ u^n_k}.$$

If $u^n_k < u^n_{k+1}$, since the scheme is an E scheme, $h^n_{k+1/2} \leq F(u^n_k)$ and $C^n_{k+1/2} \geq 0$. Likewise, $h^n_{k+1/2} \leq F(u^n_{k+1})$ and $D^n_{k+1/2} \geq 0$. Similarly, if $u^n_{k+1} > u^n_k$, then $C^n_{k+1/2} \geq 0$ and $D^n_{k+1/2} \geq 0$. Since

$$\begin{aligned} C^n_{k+1/2} + D^n_{k+1/2} &= -R\frac{h^n_{k+1/2} - F^n_k}{\delta_+ u^n_k} - R\frac{h^n_{k+1/2} - F^n_{k+1}}{\delta_+ u^n_k} \\ &= R\left|\frac{h^n_{k+1/2} - F^n_k}{\delta_+ u^n_k} + \frac{h^n_{k+1/2} - F^n_{k+1}}{\delta_+ u^n_k}\right| \\ &\leq 1, \end{aligned}$$

we see by Proposition 9.7.6 that the scheme is TVD.

The class of E schemes seems to be everything that we might want. However, we state the following result from ref. [54], page 225.

Proposition 9.7.8 *E schemes are at most first order accurate.*

Since it is difficult to live with schemes that are only first order accurate, it would seem logical to look for a different class of difference schemes. We next proceed to consider monotone schemes. We do not consider the class of monotone schemes only because E schemes are first order accurate. Recall that one of the properties of monotone schemes is that they are very easy to identify. It is also easier to prove results for monotone schemes than it is to prove the analogous results for E schemes (which is why we chose to include the proofs for monotone schemes and did not include proofs for E schemes). However, as we see from the following result, many monotone schemes are E schemes (and hence automatically at most first order accurate).

Proposition 9.7.9 *Three-point, conservative, monotone schemes are E schemes.*

Proof: If we write our scheme in conservative form as

$$\begin{aligned} u_k^{n+1} &= u_k^n - R\big[h_{k+1/2}^n - h_{k-1/2}^n\big] \\ &= u_k^n - R\big[h(u_k^n, u_{k+1}^n) - h(u_{k-1}^n, u_k^n)\big], \end{aligned}$$

we can use the monotonicity to show that h will be increasing with respect to its first argument and decreasing with respect to its second argument. Recall also that consistency of the difference scheme implies that $h(u,u) = F(u)$. Suppose now that $u_k < u_{k+1}$ and $u \in [u_k, u_{k+1}]$. Then $u_k \leq u$, $u \leq u_{k+1}$ and

$$\begin{aligned} h_{k+1/2} &= h(u_k, u_{k+1}) \\ &\leq h(u, u_{k+1}) \ (h \text{ is increasing with respect to the first argument}) \\ &\leq h(u,u) = F(u) \ (h \text{ is decreasing with respect to the second} \\ &\qquad \text{argument}). \end{aligned}$$

The argument for the case $u_{k+1} < u_k$ is similar. Hence, the scheme is an E scheme.

Before we proceed, we recall from Section 9.2.4, page 100, that we showed that conservation law (9.7.1) satisfies Entropy Condition II with entropy function $S(v) = |v - c|$ and entropy flux function $\Phi(v) = \text{sign}(v-c)[F(v) - F(c)]$. We then state and prove the following discrete entropy result.

Proposition 9.7.10 *A conservative, monotone scheme satisfies the discrete entropy condition (9.6.32) with entropy function $S(v) = |v - c|$ and the numerical entropy flux function*

$$\Psi_{k+1/2}^n = h_{k+1/2}^n(\tilde{u}_{k-p}^n, \cdots, \tilde{u}_{k+q}^n) - h_{k+1/2}^n(\tilde{\tilde{u}}_{k-p}^n, \cdots, \tilde{\tilde{u}}_{k+q}^n) \quad (9.7.22)$$

where $\tilde{u}_j^n = \max\{c, u_j^n\}$ and $\tilde{\tilde{u}}_j^n = \min\{c, u_j^n\}$. The numerical entropy flux function $\Psi_{k+1/2}^n$ is consistent with the entropy flux function $\Phi = \text{sign}(v - c)\,[F(v) - F(c)]$.

Proof: It should be reasonably clear that the numerical entropy flux function given in (9.7.22) is the general version of the numerical entropy flux functions given in Examples 9.6.1 and 9.6.2. The proof of this proposition is very similar to the analysis performed in those examples. In Examples 9.6.1 and 9.6.2 the inequalities depended on the fact that the CFL condition was satisfied. In this proposition, the inequalities will follow from the fact that we have a monotone operator. For example, the inequality analogous to inequality (9.6.38) follows from the monotonicity of $\mathcal{Q}$.

$$\begin{aligned} u_k^{n+1} &= \mathcal{Q}(u_{k-(p+1)}^n, \cdots, u_{k+q}^n) \\ &\leq \mathcal{Q}(\tilde{u}_{k-(p+1)}^n, \cdots, \tilde{u}_{k+q}^n) \ (\text{since } \mathcal{Q} \text{ is monotone increasing}). \end{aligned}$$

Obviously, the analogue to inequality (9.6.39) and inequalities for $\tilde{\tilde{u}}$ follow in a similar manner. From these inequalities, we obtain inequalities that are analogous to inequalities (9.6.40) and (9.6.41),

$$\tilde{u}_k^{n+1} \leq \mathcal{Q}(\tilde{u}_{k-(p+1)}^n, \cdots, \tilde{u}_{k+q}^n)$$

and

$$\tilde{\tilde{u}}_k^{n+1} \leq \mathcal{Q}(\tilde{\tilde{u}}_{k-(p+1)}^n, \cdots, \tilde{\tilde{u}}_{k+q}^n).$$

The result then follows from the fact that $\mathcal{Q}$ can be written as

$$\mathcal{Q}(u_{k-(p+1)}, \cdots, u_{k+q}) = u_k^n - R\big[h(u_{k-p}, \cdots, u_{k+q}) - h(u_{k-p-1}, \cdots, u_{k+q-1})\big].$$

The consistency is left to the reader in HW9.7.9.

Remark 1: The result of the above proposition along with Theorem 9.6.6 is that if we have a conservative, monotone scheme for which u_k^n converges to v as $\Delta x,\ \Delta t \to 0$ (in $L_{1,loc}$), then v will be the entropy solution, i.e., the scheme will compute approximations to the vanishing viscosity solution.

Remark 2: On page 299, ref. [25], is proved the analogous result to Proposition 9.7.10 and Theorem 9.6.6 for Entropy Condition I, i.e., if we have a conservative, monotone scheme for which u_k^n converges to v as $\Delta x,\ \Delta t \to 0$, then v will satisfy Entropy Condition I.

Hence, as is the case with E schemes, if we compute with a conservative, monotone scheme that converges, we obtain a weak entropy solution. We should note that the above result specifies the particular entropy and entropy flux functions used. We recall from Section 9.3, page 119, that there will generally be several entropy solutions associated with different choices of entropy functions and entropy flux functions. We emphasize that Proposition 9.7.10 holds for the particular entropy function, entropy flux function and numerical entropy flux function given in the hypotheses of the proposition.

Also, as is the case with E schemes, we find that monotone schemes will not produce the dispersive wiggles seen with the Lax-Wendroff scheme.

Proposition 9.7.11 *Monotone conservative schemes are TVD.*

Proof: We note that

$$\begin{aligned} TV(\mathbf{u}^{n+1}) &= \sum_{k=-\infty}^{\infty} |\delta_+ u_k^{n+1}| = \sum_{k=-\infty}^{\infty} |u_{k+1}^{n+1} - u_k^{n+1}| \\ &= \sum_{k=-\infty}^{\infty} |\mathcal{Q}(u_{k+1-(p+1)}^n, \cdots, u_{k+1+q}^n) - \mathcal{Q}(u_{k-(p+1)}^n, \cdots, u_{k+q}^n)| \\ &= \sum_{k=-\infty}^{\infty} |\mathcal{Q}(u_{k-(p+1)}^n + \delta_+ u_{k-(p+1)}^n, \cdots, u_{k+q}^n + \delta_+ u_{k+q}^n) \\ &\qquad - \mathcal{Q}(u_{k-(p+1)}^n, \cdots, u_{k+q}^n)|. \end{aligned}$$

Since

$$\frac{d}{dt}\mathcal{Q}(u^n_{k-(p+1)} + t\delta_+ u^n_{k-(p+1)}, \cdots, u^n_{k+q} + t\delta_+ u^n_{k+q})$$
$$= \sum_{j=-(p+1)}^{q} \mathcal{Q}_j(u^n_{k-(p+1)} + t\delta_+ u^n_{k-(p+1)}, \cdots, u^n_{k+q} + t\delta_+ u^n_{k+q})\delta_+ u^n_{k+j},$$

where $\mathcal{Q}_j$ represents the partial derivative of $\mathcal{Q}$ with respect to the $(p+2+j)$-th variable ($j = -(p+1)$ corresponds to differentiation with respect to the first variable, $j = -p$ corresponds to differentiation with respect to the second variable, $\ldots$, $j = q$ corresponds to differentiation with respect to the $(p+q+2)$-th variable), we can continue the above calculation and get

$$TV(\mathbf{u}^{n+1}) = \sum_{k=-\infty}^{\infty} \Big| \sum_{j=-(p+1)}^{q} \int_0^1 \mathcal{Q}_j(u^n_{k-(p+1)} + t\delta_+ u^n_{k-(p+1)},$$
$$\cdots, u^n_{k+q} + t\delta_+ u^n_{k+q})\delta_+ u^n_{k+j}\, dt \Big|.$$

Then, using the triangular inequality, taking the absolute value inside of the integral, using the fact that the difference scheme is monotone (the appropriate partial derivatives are greater than or equal to zero) and making a change of index $m = k + j$, we get

$$TV(\mathbf{u}^{n+1}) \le \sum_{k=-\infty}^{\infty} \sum_{j=-(p+1)}^{q} \int_0^1 \mathcal{Q}_j(u^n_{k-(p+1)} + t\delta_+ u^n_{k-(p+1)},$$
$$\cdots, u^n_{k+q} + t\delta_+ u^n_{k+q})|\delta_+ u^n_{k+j}|\, dt$$
$$= \sum_{m=-\infty}^{\infty} \sum_{j=-(p+1)}^{q} \int_0^1 \mathcal{Q}_j(u^n_{m-j-(p+1)} + t\delta_+ u^n_{m-j-(p+1)}, \cdots,$$
$$u^n_{m-j+q} + t\delta_+ u^n_{m-j+q})|\delta_+ u^n_m|\, dt. \tag{9.7.23}$$

By the fact that the difference scheme is conservative, we know that we can write

$$\mathcal{Q}(w_{\ell-(p+1)}, \ldots, w_{\ell+q}) = w_\ell - R\Big[h(w_{\ell-p}, \ldots, w_{\ell+q})$$
$$- h(w_{\ell-p-1}, \ldots, w_{\ell+q-1})\Big]. \tag{9.7.24}$$

We get

$$\mathcal{Q}_{-(p+1)} = Rh_1(w_{\ell-p-1}, \dots, w_{\ell+q-1}) \tag{9.7.25}$$

$$\mathcal{Q}_j = \delta_{j\ell} - R\Big[h_{j+p+1}(w_{\ell-p}, \dots, w_{\ell+q}) - h_{j+p+2}(w_{\ell-p-1}, \dots, w_{\ell+q-1})\Big], \quad j = -p, \dots, q-1 \tag{9.7.26}$$

$$\mathcal{Q}_q = -Rh_{q+p+1}(w_{\ell-p}, \dots, w_{\ell+q}). \tag{9.7.27}$$

The notation used above (and during the rest of this proof) is different from what we have used in the past. The function h has no subscripts or superscripts. As long as we include the arguments of h, the usual subscripts and superscripts are unnecessary. The h_j's used refer to partial differentiation with respect to the j-th variable. We apologize for the temporary change in notation (and the different use of subscripts to denote differentiation on the $\mathcal{Q}$'s and the h's), but we felt that it was the best way to make the proof as clear as possible. We note that the $\delta_{j\ell}$ term is the usual Kronecker delta, which provides a one when we differentiate expression (9.7.24) with respect to w_ℓ. Summing (9.7.25)–(9.7.27) gives

$$\begin{aligned}
&\sum_{j=-(p+1)}^{q} \mathcal{Q}_j(w_{m-j-(p+1)}, \dots, w_{m-j+q}) \\
&= Rh_1(w_m, \dots, w_{m+(p+1)+q}) + 1 - R\sum_{j=-p}^{q-1}\Big[h_{j+p+1}(w_{m-j-p}, \dots, w_{m-j+q}) \\
&\quad - h_{j+p+2}(w_{m-j-p-1}, \dots, w_{m-j+q-1})\Big] - Rh_{q+p}(w_{m-q-p}, \dots, w_m) \\
&= Rh_1(w_m, \dots, w_{m+(p+1)+q}) + 1 - R\sum_{j=-p}^{q-1} h_{j+p+1}(w_{m-j-p}, \dots, w_{m-j+q}) \\
&\quad + R\sum_{j=-p}^{q-1} h_{j+p+2}(w_{m-j-p-1}, \dots, w_{m-j+q-1}) - Rh_{q+p}(w_{m-q-p}, \dots, w_m) \\
&= Rh_1(w_m, \dots, w_{m+(p+1)+q}) + 1 - R\sum_{j=-p}^{q-1} h_{j+p+1}(w_{m-j-p}, \dots, w_{m-j+q}) \\
&\quad + R\sum_{\ell=-p+1}^{q} h_{\ell+p-1}(w_{m-\ell-p}, \dots, w_{m-\ell+q}) - Rh_{q+p}(w_{m-q-p}, \dots, w_m) \\
&= Rh_1(w_m, \dots, w_{m+(p+1)+q}) + 1 - Rh_1(w_m, \dots, w_{m+p+q}) \\
&\quad + Rh_{q+p}(w_{m-q-p}, \dots, w_m) - Rh_{q+p}(w_{m-q-p}, \dots, w_m) \\
&= 1.
\end{aligned}$$

Applying this identity to the expression given in (9.7.23) yields

$$TV(\mathbf{u}^{n+1}) \leq \sum_{m=-\infty}^{\infty} \int_0^1 |\delta_+ u_m^n| \, dt = \sum_{m=-\infty}^{\infty} |\delta_+ u_m^n| = TV(\mathbf{u}^n),$$

which is what we were trying to prove.

We next include a result concerning the truncation error of a difference scheme that is helpful for proving Proposition 9.7.13 given below. We should note that this results is a general result that can be used in a variety of ways and will be used later.

Proposition 9.7.12 *The truncation error of a conservative difference scheme of the form*

$$u_k^{n+1} = \mathcal{Q}(u_{k-(p+1)}^n, \ldots, u_{k+q}^n) \tag{9.7.28}$$

is given by

$$\Delta t \tau_k^n = -\Delta t^2 \left\{ [q(u)u_x]_x \right\}_{u=v_k^n} + \mathcal{O}(\Delta t(\Delta t^2 + \Delta x^2)), \tag{9.7.29}$$

where $v = v(x,t)$ is a solution to conservation law (9.7.1) and

$$q(u) = \frac{1}{2}\Big[\frac{1}{R^2} \sum_{j=-(p+1)}^{q} j^2 \mathcal{Q}_j(u, \ldots, u) - \big[F'(u)\big]^2\Big]. \tag{9.7.30}$$

Proof: We begin by expanding $\Delta t \tau_k^n = v_k^{n+1} - \mathcal{Q}(v_{k-(p+1)}^n, \ldots, v_{k+q}^n)$ in a Taylor series, where $v = v(x,t)$ is a solution to conservation law (9.7.1) as

$$\begin{aligned}
\Delta t \tau_k^n &= v_k^{n+1} - \mathcal{Q}(v_{k-(p+1)}^n, \ldots, v_{k+q}^n) \\
&= v_k^n + \Delta t (v_t)_k^n + \frac{\Delta t^2}{2}(v_{tt})_k^n - \mathcal{Q}(v_k^n, \ldots, v_k^n) \\
&\quad - \sum_{j=-(p+1)}^{q} \mathcal{Q}_j(v_k^n, \ldots, v_k^n)(v_{k+j}^n - v_k^n) \\
&\quad - \frac{1}{2} \sum_{j=-(p+1)}^{q} \sum_{m=-(p+1)}^{q} \mathcal{Q}_{km}(v_k^n, \ldots, v_k^n)(v_{k+j}^n - v_k^n)(v_{k+m}^n - v_k^n) \\
&\quad + \mathcal{O}(\Delta t^3) + \mathcal{O}(\Delta t \Delta x^2).
\end{aligned}$$

Expanding v_{k+j}^n and v_{k+m}^n about v_k^n gives

$$\begin{aligned}\Delta t\tau_k^n =& v_k^n + \Delta t(v_t)_k^n + \frac{\Delta t^2}{2}(v_{tt})_k^n - \mathcal{Q}(v_k^n,\dots,v_k^n)\\ &- \Delta x(v_x)_k^n \sum_{j=-(p+1)}^{q} j\mathcal{Q}_j(v_k^n,\dots,v_k^n)\\ &- \frac{\Delta x^2}{2}(v_{xx})_k^n \sum_{j=-(p+1)}^{p} j^2\mathcal{Q}_j(v_k^n,\dots,v_k^n)\\ &- \frac{\Delta x^2}{2}\left[(v_x)_k^n\right]^2 \sum_{j=-(p+1)}^{q}\sum_{m=-(p+1)}^{q} jm\mathcal{Q}_{jm}(v_k^n,\dots,v_k^n)\\ &+ \mathcal{O}\big(\Delta t(\Delta t^2+\Delta x^2)\big).\end{aligned}$$

Since the scheme is conservative, $\mathcal{Q}$ can be written as in (9.7.24), and we see that

$$\mathcal{Q}(v_k^n,\dots,v_k^n) = v_k^n - R[h(v_k^n,\dots,v_k^n) - h(v_k^n,\dots,v_k^n)] = v_k^n.$$

Also, using the conservation form of the scheme, we see that

$$\begin{aligned}&\sum_{j=-(p+1)}^{q}\sum_{m=-(p+1)}^{q} jm\mathcal{Q}_{jm}(v_k^n,\dots,v_k^n)\\ &\qquad = \sum_{j=-(p+1)}^{q}\sum_{m=-(p+1)}^{q} j^2\mathcal{Q}_{jm}(v_k^n,\dots,v_k^n).\end{aligned}\tag{9.7.31}$$

The truncation error can then be written as

$$\begin{aligned}\Delta t\tau_k^n =& \Delta t(v_t)_k^n - \Delta x(v_x)_k^n \sum_{j=-(p+1)}^{q} j\mathcal{Q}_j(v_k^n,\dots,v_k^n) + \frac{\Delta t^2}{2}(v_{tt})_k^n\\ &- \frac{\Delta x^2}{2}(v_{xx})_k^n \sum_{j=-(p+1)}^{p} j^2\mathcal{Q}_j(v_k^n,\dots,v_k^n)\\ &- \frac{\Delta x^2}{2}\left[(v_x)_k^n\right]^2 \sum_{j=-(p+1)}^{q}\sum_{m=-(p+1)}^{q} j^2\mathcal{Q}_{jm}(v_k^n,\dots,v_k^n)\\ &+ \mathcal{O}\big(\Delta t(\Delta t^2+\Delta x^2)\big).\end{aligned}\tag{9.7.32}$$

Using the fact that

$$\sum_{j=-(p+1)}^{q} j\mathcal{Q}_j(v_k^n,\dots,v_k^n) = -R\big[F'(v)\big]_k^n,$$

which is necessary for consistency, and the fact that

$$j^2 \sum_{m=-(p+1)}^{q} \mathcal{Q}_{jm}(v_k^n \dots, v_k^n)(v_x)_k^n = \left[\left(j^2 \mathcal{Q}_j(v, \dots, v)\right)_x\right]_k^n,$$

we can eliminate the first two terms of equation (9.7.32) (because $v_t + F'(v)v_x = 0$) and write the truncation error as

$$\begin{aligned}\Delta t \tau_k^n =& \frac{\Delta t^2}{2}(v_{tt})_k^n - \frac{\Delta x^2}{2}(v_{xx})_k^n \sum_{j=-(p+1)}^{p} j^2 \mathcal{Q}_j(v_k^n, \dots, v_k^n) \\ &- \frac{\Delta x^2}{2}(v_x)_k^n \sum_{j=-(p+1)}^{q} \left[\left(j^2 \mathcal{Q}_j(v, \dots, v)\right)_x\right]_k^n \\ &+ \mathcal{O}(\Delta t(\Delta t^2 + \Delta x^2)). \end{aligned} \tag{9.7.33}$$

If we then use the fact that v is a solution to conservation law (9.7.1), we can derive the identity

$$v_{tt} = \left[\left(F'(v)\right)^2 v_x\right]_x.$$

Eliminating v_{tt} from (9.7.33) leaves us with

$$\begin{aligned}\Delta t \tau_k^n = \frac{\Delta t^2}{2}\Big\{\Big[\Big((F'(v))^2 - \frac{1}{R^2} \sum_{j=-(p+1)}^{q} j^2 \mathcal{Q}_j(v, \dots, v)\Big) v_x\Big]_x\Big\}_k^n \\ + \mathcal{O}(\Delta t(\Delta t^2 + \Delta x^2)), \end{aligned} \tag{9.7.34}$$

which is what we were to prove.

And finally, we state a result that will show us that though monotone schemes give us almost everything we could want and/or need, the schemes will not generally be sufficiently accurate to be used when our solutions include discontinuities.

Proposition 9.7.13 *Monotone schemes are almost always only first order accurate.*

Proof: We begin by noting that

$$RF'(v_k^n) = \sum_{j=-(p+1)}^{q} j \mathcal{Q}_j(v_k^n, \dots, v_k^n).$$

Then

$$R^2\left[F'(v_k^n)\right]^2 = \left(\sum_{j=-(p+1)}^{q} j\mathcal{Q}_j(v_k^n,\ldots,v_k^n)\right)^2$$

$$= \left(\sum_{j=-(p+1)}^{q} j\sqrt{\mathcal{Q}_j(v_k^n,\ldots,v_k^n)}\sqrt{\mathcal{Q}_j(v_k^n,\ldots,v_k^n)}\right)^2 \quad (9.7.35)$$

$$\leq \sum_{j=-(p+1)}^{q} j^2\mathcal{Q}_j(v_k^n,\ldots,v_k^n) \sum_{j=-(p+1)}^{q} \mathcal{Q}_j(v_k^n,\ldots,v_k^n) \quad (9.7.36)$$

$$= \sum_{j=-(p+1)}^{q} j^2\mathcal{Q}_j(v_k^n,\ldots,v_k^n).$$

Thus it follows that $q(v_k^n) \geq 0$.

By the Schwarz inequality, equality in equation (9.7.36) occurs when $j\mathcal{Q}_j(v_k^n,\ldots,v_k^n) = C\mathcal{Q}_j(v_k^n,\ldots,v_k^n)$ for some constant C. Hence,

$$\mathcal{Q}_j(v_k^n,\ldots,v_k^n) = 0$$

for each j, and $\mathcal{Q}$ is the constant function. In this case, the difference scheme will be one of pure translation. We should note that this trivial case is what is referred to as "almost always" in the proposition statement.

The condition that $q > 0$ is technically not enough to ensure that the scheme is of first order. The problem is not that the term $\left[q(u)u_x\right]_x$ is evaluated at v_k^n. If the scheme is to be of second order, the term $\left\{\left[q(u)u_x\right]_x\right\}_{u=v_k^n}$ must hold as $\Delta t,\ \Delta x \to 0$. It is not a problem if the term $\left\{\left[q(u)u_x\right]_x\right\}_{u=v_k^n}$ happens to be zero for some Δx and Δt (some k and n). We must be careful of the fact that $q > 0$ implies that $\left[q(v)v_x\right]_x \neq 0$. If $\left[q(v)v_x\right]_x = 0$, then

$$0 = \int_{-\infty}^{\infty} \phi\left[q(v)v_x\right]_x dx = -\int_{-\infty}^{\infty} \phi_x q(v)v_x\, dx$$

for any $\phi \in C_0^1$. However, it is possible to choose ϕ to be arbitrarily close to v on the support of ϕ. In this case, the last integral will be negative and we have a contradiction.

We emphasize that the monotonicity assumption is used in line (9.7.35) above, where we split $j\mathcal{Q}_j$ into $j\sqrt{\mathcal{Q}_j}\sqrt{\mathcal{Q}_j}$.

We next state and prove several useful results. These results are helpful in that they will show us that in order to obtain a higher order scheme, we should consider neither linear schemes nor three-point schemes. In addition, we see that it is impossible to obtain a TVD scheme (which is what we want) that is second order accurate everywhere. We begin with the following result.

Proposition 9.7.14 *A linear TVD difference scheme is at most of first order.*

Proof: We consider a difference scheme of the form

$$u_k^{n+1} = \sum_{j=-q}^{p} a_j u_{k+j}^n \tag{9.7.37}$$

and the function

$$u_k^n = \begin{cases} 1 & \text{if } j \leq 0 \\ 0 & \text{if } j > 0. \end{cases}$$

Then $TV(\mathbf{u}^n) = 1$, and

$$TV(\mathbf{u}^{n+1}) = \sum_{k=-\infty}^{\infty} |\delta_+ u_k^{n+1}| = \sum_{k=-\infty}^{\infty} \left| \sum_{j=-q}^{p} a_j \delta_+ u_{k+j}^n \right| = \sum_{j=-q}^{p} |a_j|.$$

Difference scheme (9.7.37) is consistent if

$$\sum_{j=-q}^{p} a_j = 1. \tag{9.7.38}$$

Hence, if $a_j < 0$ for some j, then

$$TV(\mathbf{u}^{n+1}) = \sum_{j=-q}^{p} |a_j| > 1 = TV(\mathbf{u}^n)$$

and the scheme is not TVD. Thus we must have $a_j \geq 0$ for all j, $-q \leq j \leq p$. Thus, the scheme is monotone, and by Proposition 9.7.11 the scheme is at most first order accurate.

Of course, we want schemes that are more than first order accurate. One of the methods for obtaining second order accuracy is to use a scheme that is almost the Lax-Wendroff scheme. Let $h_{k+1/2}^{LW}$ denote the numerical flux function associated with the Lax-Wendroff scheme (9.5.10) and let $h_{k+1/2}^n$ denote the numerical flux function for the scheme under consideration. The following result shows how much the scheme must be "almost the Lax-Wendroff scheme" to inherit the second order accuracy.

Proposition 9.7.15 *If*

$$h_{k+1/2}^n - h_{k+1/2}^{LW} = \mathcal{O}(\Delta x^2) \tag{9.7.39}$$

and the leading error term in the $\mathcal{O}(\Delta x^2)$ term is smooth, then the difference scheme associated with numerical flux function $h_{k+1/2}^n$ is second order accurate.

Proof: We know that because the Lax-Wendroff scheme is second order accurate (using the assumption that we have used before that we need include only the Δx term in the error—assuming that Δx and Δt are related linearly),

$$\frac{u_k^{n+1} - u_k^n}{\Delta t} + \frac{\delta_- h_{k+1/2}^{LW}}{\Delta x} - \left[u_t + F(u)_x\right]_k^n = \mathcal{O}(\Delta x^2). \tag{9.7.40}$$

If we add $\delta_- h_{k+1/2}^n/\Delta x$ to both sides of equation (9.7.40) and move the term $\delta_- h_{k+1/2}^{LW}/\Delta x$ to the right hand side, we have

$$\begin{aligned}\frac{u_k^{n+1} - u_k^n}{\Delta t} + \frac{\delta_- h_{k+1/2}^n}{\Delta x} - \left[u_t + F(u)_x\right]_k^n \\ = \mathcal{O}(\Delta x^2) + \frac{\left(h_{k+1/2}^n - h_{k+1/2}^{LW}\right) - \left(h_{k-1/2}^n - h_{k-1/2}^{LW}\right)}{\Delta x}.\end{aligned} \tag{9.7.41}$$

Since $h_{k+1/2}^n - h_{k+1/2}^{LW} = \mathcal{O}(\Delta x^2)$ and the leading term of the $\mathcal{O}(\Delta x^2)$ is smooth, $\delta_-\left(h_{k+1/2}^n - h_{k+1/2}^{LW}\right) = \mathcal{O}(\Delta x^3)$. Dividing by Δx loses one power, so we are left with

$$\frac{\left(h_{k+1/2}^n - h_{k+1/2}^{LW}\right) - \left(h_{k-1/2}^n - h_{k-1/2}^{LW}\right)}{\Delta x} = \mathcal{O}(\Delta x^2).$$

Using this result in (9.7.41) leaves us with

$$\frac{u_k^{n+1} - u_k^n}{\Delta t} + \frac{\delta_- h_{k+1/2}^n}{\Delta x} - \left[u_t + F(u)_x\right]_k^n = \mathcal{O}(\Delta x^2),$$

which is what we were to prove.

As we stated earlier, to eliminate the unwanted oscillations that are so prevalent in higher order schemes, we require that our schemes be TVD. The most common method used to prove that a scheme is TVD (or to force a scheme to be TVD) is to use Proposition 9.7.6. It is a technical difficulty that Proposition 9.7.6 does not provide a necessary condition for a scheme to be TVD. It is at least comforting that we have not been able to find a TVD scheme that does not satisfy the hypotheses of Proposition 9.7.6. To allow us to obtain results that help describe the desired higher order schemes, we define the following subclass of TVD schemes.

Definition 9.7.16 *A difference scheme in incremental form (9.7.8) is said to be incremental TVD if* $C_{k+1/2}^n \geq 0$, $D_{k+1/2}^n \geq 0$ *and* $C_{k+1/2}^n + D_{k+1/2}^n \leq 1$.

Remark: We emphasize that the reason for the above definition is to provide a class of TVD schemes for which we can prove Propositions 9.7.18, 9.7.19 and 9.7.20. There may be TVD schemes that do not satisfy conditions

$C^n_{k+1/2} \geq 0$, $D^n_{k+1/2} \geq 0$ and $C^n_{k+1/2} + D^n_{k+1/2} \leq 1$. Without Definition 9.7.16, we could prove the following variations of Propositions 9.7.18, 9.7.19 and 9.7.20.

- If the numerical viscosity of a conservative scheme satisfies (9.7.43), then the scheme is TVD.
- If the numerical viscosity of a three-point, conservative scheme satisfies (9.7.43), then the difference scheme is at most first order accurate.
- Suppose a difference scheme satisfies the following conditions:

$$C^n_{k+1/2} \geq 0, \quad D^n_{k+1/2} \geq 0 \quad \text{and} \quad C^n_{k+1/2} + D^n_{k+1/2} \leq 1.$$

  Then at smooth extrema that are not sonic points, the scheme is at most first order accurate.

Working with the class of incrementally TVD schemes, we can more easily make the points that we want schemes that are not three-point schemes and that we should not expect second order accuracy everywhere. We should note that by Definition 9.7.16, Proposition 9.7.10 provides a necessary and sufficient condition for a scheme to be incremental TVD. It may be the case that Proposition 9.7.6 is an "iff" in its present form (or in a better approximation to its present form), but the proof would surely be very difficult. For completeness, we state the following "iff" version of Proposition 9.7.6.

Proposition 9.7.17 *Consider a difference scheme in I-form as given in (9.7.8). Then, difference scheme (9.7.8) is incrementally TVD if and only if*

$$C^n_{k+1/2} \geq 0, \quad D^n_{k+1/2} \geq 0 \quad \textit{and} \quad C^n_{k+1/2} + D^n_{k+1/2} \leq 1. \tag{9.7.42}$$

We then prove the following three results.

Proposition 9.7.18 *A conservative scheme is incremental TVD if and only if the numerical viscosity coefficient satisfies*

$$R|a^n_{k+1/2}| \leq Q^n_{k+1/2} \leq 1 \tag{9.7.43}$$

for all k where $a_{k+1/2}$ is the local wave speed defined in (9.4.17).

Proof: Using expressions (9.7.14), (9.7.15), we see that a scheme in Q-form is related to a scheme in I-form via

$$C^n_{k+1/2} = \frac{1}{2}\left(Q^n_{k+1/2} - Ra^n_{k+1/2}\right) \tag{9.7.44}$$

$$D^n_{k+1/2} = \frac{1}{2}\left(Q^n_{k+1/2} + Ra^n_{k+1/2}\right). \tag{9.7.45}$$

We will apply Proposition 9.7.17 and assume that the scheme is incrementally TVD. The conditions that $C^n_{k+1/2} \geq 0$ and $D^n_{k+1/2} \geq 0$ imply that $Q^n_{k+1/2} \geq Ra^n_{k+1/2}$ and $Q^n_{k+1/2} \geq -Ra^n_{k+1/2}$, respectively. In other words, these two conditions imply that $Q^n_{k+1/2} \geq R|a^n_{k+1/2}|$. By adding expressions (9.7.44) and (9.7.45), we see that the condition that $C^n_{k+1/2} + D^n_{k+1/2} \leq 1$ gives $Q^n_{k+1/2} \leq 1$. Therefore, we have $R|a^n_{k+1/2}| \leq Q^n_{k+1/2}| \leq 1$. The proof of the converse is almost identical.

Remark: If we write the Lax-Friedrichs scheme, (9.4.4), in Q-form with $Q^n_{k+1/2} = 1$ and apply Proposition 9.7.18, we see that the Lax-Friedrichs scheme will be TVD if $R|a^n_{k+1/2}| \leq 1$, which will be satisfied if we assume that the discrete CFL condition is satisfied.

Proposition 9.7.18 is a nice result in that condition (9.7.43) characterizes the class of three-point, conservative, incrementally TVD schemes. This characterization can be used to prove the following result.

Proposition 9.7.19 *A three-point, conservative scheme that is incrementally TVD is at most first order accurate.*

Proof: We consider a scheme in Q-form

$$u^{n+1}_k = u^n_k - \frac{R}{2}\delta_0 F^n_k + \frac{1}{2}\delta_+\left(Q^n_{k-1/2}\delta_- u^n_k\right). \tag{9.7.46}$$

Since (9.7.46) is a three-point scheme, we note that the numerical viscosity coefficient $Q_{k+1/2}$ can depend only on u_k and u_{k+1} (so that

$$\delta_+\left(Q^n_{k-1/2}\delta_- u^n_k\right) = Q^n_{k+1/2}\delta_- u^n_{k+1} - Q^n_{k-1/2}\delta_- u^n_k$$

will depend only on u^n_{k-1}, u^n_k and u^n_{k+1}). We return to Proposition 9.7.12 and note that for a scheme of the form (9.7.46),

$$\begin{aligned}
Q_1(u,u) =& \Big\{ -\frac{R}{2}\big(F'(u_{k+1}) \\
&+ \frac{1}{2}\big[Q(u_k,u_{k+1}) + Q_2(u_k,u_{k+1})(u_{k+1}-u_k)\big]\Big\} \\
&\quad \text{(where } u_k = u_{k+1} = u\text{)} \\
=& -\frac{R}{2}F'(u) + \frac{1}{2}Q(u,u) \\
Q_{-1}(u,u) =& \Big\{\frac{R}{2}\big(F'(u_{k-1}) \\
&+ \frac{1}{2}\big[Q(u_{k-1},u_k) + Q_{-1}(u_{k-1},u_k)(u_k-u_{k-1})\big]\Big\} \\
&\quad \text{(where } u_{k-1} = u_k = u\text{)} \\
=& \frac{R}{2}F'(u) + \frac{1}{2}Q(u,u)
\end{aligned}$$

and

$$q(u) = \frac{1}{2}\frac{1}{R^2}Q(u,u) - \frac{1}{2}\big[F'(u)\big]^2.$$

If difference scheme (9.7.46) is to be second order accurate, we must have $q(u) = 0$ or $Q(u,u) = R^2\big[F'(u)\big]^2$. By Proposition 9.7.18, we have

$$R|a_{k+1/2}| \le Q^n_{k+1/2} \le 1.$$

If $a^n_{k+1/2} = F'(u^n_k)$, then we must have $R|F'(v^n_k)| = 1$. If $a^n_{k+1/2} = \delta_+F^n_k/\delta_+u^n_k$, then for sufficiently small $\delta_+u^n_k$ (sufficiently small Δx), we must have

$$R|\delta_+F(v^n_k)/\delta_+u^n_k| = 1$$

where $v = v(x,t)$ is a solution to conservation law (9.7.1). In either case this is a contradiction, since this restricts F to be equal to or approximately equal to $1/R$ times u, which is surely not the case (especially since $1/R$ can vary with Δx and Δt).

Proposition 9.7.20 *At smooth extrema that are not sonic points, an incremental TVD scheme is at most first order accurate.*

Proof: Let $v = v(x,t)$ be a solution to conservation law (9.7.1) and let C and D denote the functions that define $C^n_{k+1/2}$ and $D^n_{k-1/2}$, respectively, i.e.,

$$\begin{aligned} C^n_{k+1/2} &= C(u^n_{k-(p+1)}, \cdots, u^n_{k+q}) \\ D^n_{k-1/2} &= D(u^n_{k-(p+1)}, \cdots, u^n_{k+q}). \end{aligned}$$

We should be aware that $C^n_{k+1/2}$ and $D^n_{k-1/2}$ will not generally both depend on the same terms. i.e., $u^n_{k-(p+1)}, \cdots, u^n_{k+q}$. However, for this proof it is sufficient to let this list contain all of the points on which either $C^n_{k+1/2}$ or $D^n_{k-1/2}$ depend, so that we can easily interface with the notation used in the definition of difference scheme (9.7.4).

We use the truncation error given in Proposition 9.7.12 by equation (9.7.29)–(9.7.30). For a difference scheme in I-form, the operator $\mathcal{Q}$ used in difference scheme (9.7.4) will be given by

$$\mathcal{Q}(u^n_{k-(p+1)}, \cdots, u^n_{k+q}) = u^n_k + C^n_{k+1/2}\delta_+u^n_k - D^n_{k-1/2}\delta_-u^n_k.$$

We note that

$$\sum_{j=-(p+1)}^{q} j^2 \mathcal{Q}_j(u_{k-(p+1)}, \dots, u_{k+q})$$

$$= \sum_{j=-(p+1)}^{q} j^2 \big[C_j(u_{k-(p+1)}, \dots, u_{k+q})(u_{k+1} - u_k)$$

$$- D_j(u_{k-(p+1)}, \dots, u_{k+q})(u_k - u_{k-1})\big]$$

$$+ C(u_{k-(p+1)}, \dots, u_{k+q}) + D(u_{k-(p+1)}, \dots, u_{k+q}).$$

Thus

$$q(u) = \frac{1}{2}\Big[\frac{1}{R^2}\big(C(u, \dots, u) + D(u, \dots, u)\big) - \big(F'(u)\big)\Big] \tag{9.7.47}$$

and

$$\begin{aligned}\Delta t \tau_k^n &= -\Delta t^2 \left\{[q(u)u_x]_x\right\}_{u=v_k^n} + \mathcal{O}(\Delta t(\Delta t^2 + \Delta x^2)) \\ &= -\Delta t^2 \left\{q(u)_x u_x + q(u)u_{xx}\right\}_{u=v_k^n} + \mathcal{O}(\Delta t(\Delta t^2 + \Delta x^2))\end{aligned}$$

Since $v_x = 0$ at extrema, we have

$$\Delta t \tau_k^n = \frac{1}{2}\Big[\frac{1}{R^2}\big[C(v_k^n, \dots, v_k^n) + D(v_k^n, \dots, v_k^n)\big] - \big(F'(v_k^n)\big)^2\Big]. \tag{9.7.48}$$

Thus for the scheme to be second order accurate, we must have

$$C(v_k^n, \dots, v_k^n) + D(v_k^n, \dots, v_k^n) = R^2\big(F'(v_k^n)\big)^2. \tag{9.7.49}$$

Consistency implies that

$$C(v_k^n, \dots, v_k^n) - D(v_k^n, \dots, v_k^n) = -RF'(v_k^n). \tag{9.7.50}$$

Solving equations (9.7.49) and (9.7.50) for C and D gives

$$C(v_k^n, \dots, v_k^n) = \frac{1}{2}\Big[R^2\big(F'(v_k^n)\big) - RF'(v_k^n)\Big] \tag{9.7.51}$$

and

$$D(v_k^n, \dots, v_k^n) = \frac{1}{2}\Big[R^2\big(F'(v_k^n)\big) + RF'(v_k^n)\Big]. \tag{9.7.52}$$

The fact that we have assumed that our schemes all satisfy the CFL condition, $R|F'(v_k^n)| \le 1$, implies that either

$$C(v_k^n, \dots, v_k^n) \le 0 \quad \text{or} \quad D(v_k^n, \dots, v_k^n) \le 0.$$

By (9.7.51) and (9.7.52) we see that the only way that $C(v_k^n, \ldots, v_k^n)$ or $D(v_k^n, \ldots, v_k^n)$ could be zero is if $F'(v_k^n) = 0$ (assuming of course that $R > 0$). Since we are considering only points that are not sonic points, $F'(v_k^n) \neq 0$. Thus either $C(v_k^n, \ldots, v_k^n) < 0$ or $D(v_k^n, \ldots, v_k^n) < 0$.

Using a Taylor series expansion, we see that

$$C_{k+1/2}^n = C(v_k^n, \ldots, v_k^n) + \mathcal{O}(\Delta x) \tag{9.7.53}$$

$$D_{k-1/2}^n = D(v_k^n, \ldots, v_k^n) + \mathcal{O}(\Delta x). \tag{9.7.54}$$

Noting from (9.7.51) and (9.7.52) how $C(v_k^n, \ldots, v_k^n)$ and $D(v_k^n, \ldots, v_k^n)$ depend on F (and not on Δx), whichever one of C and D is negative must be negative to the first order, i.e., when combined with (9.7.53) and (9.7.54), we see that either

$$C_{k+1/2}^n = -\mathcal{O}(1)$$

or

$$D_{k-1/2}^n = -\mathcal{O}(1).$$

This contradicts the hypothesis that the difference scheme is incrementally TVD.

HW 9.7.9 Consider the entropy function S, the entropy flux function Φ and numerical entropy flux function $\Psi_{k+1/2}^n$ in Proposition 9.7.10. Verify that $\Psi_{k+1/2}^n$ is consistent with Φ.

9.7.3 Godunov Scheme

We have found so far that the Lax-Wendroff scheme is too dispersive and does not produce the vanishing viscosity solutions. We have also found that the E schemes and monotone schemes are only first order accurate. Besides the fact that it is useful to know that these schemes are not adequate, we justify their introduction by the fact that they will be part of our high resolution schemes that we introduce later. In this section we introduce another monotone scheme, the Godunov scheme, that will be very important for the development of many of the schemes that we obtain in the remainder of this chapter. In ref. [12], Godunov derived a scheme for solving the gas dynamics equations based on using the exact solution of local Riemann problems. As we shall see, the Godunov scheme is an E scheme (and hence only first order accurate).

We consider the solution to Burgers' equation and the initial condition given in Figure 9.7.1(a). As time proceeds, the part of the solution near points A and B will travel faster than that near C, and the result will be that the solution might eventually look like the solution given in Figure 9.7.1(b).

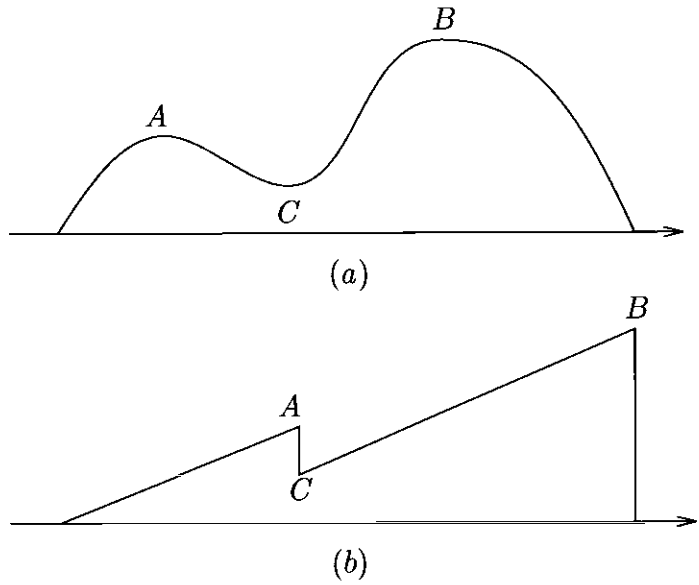

FIGURE 9.7.1. (a) The initial condition. (b) The solution after enough time has passed to form the two shocks (but not enough to eliminate either).

Next we consider the same problem considered above, except that before we start, we approximate the initial condition given in Figure 9.7.1(a) by a piecewise constant step function. The piecewise constant approximation of v_0 given in Figure 9.7.2(a) is based on a uniform grid in the x variable using a reasonably large Δx. The Δx chosen, which is obviously larger than we would use to compute with, was chosen to make the picture a little bit easier to see. We let time proceed using as our initial condition the function plotted in Figure 9.7.2(a). We want to consider this problem as many (eight in our case) local Riemann problems. It is easy to see that some of the jumps will become fans (the jumps with $v_L < v_R$) and some of the jumps will propagate as jumps (when $v_L > v_R$). If we consider the solution after some small time increment, the solution might look like the solution given in Figure 9.7.2(b). The jumps that evolve into fans approximate the portion of the solution given in Figure 9.7.1(b) that slopes upward, and the jumps that evolve as jumps approximate the jumps in the solution given in Figure 9.7.1(b). We see that at this time it takes two jumps (three grid points) in the solution given in Figure 9.7.2(b) to approximate the big jump in the solution given in Figure 9.7.1(b) that is near point B. The point is that the piecewise constant approximation will evolve into a solution that is an approximation of the continuous solution and has the same qualities as that solution.

We want to take our approximation another step. We got the solution

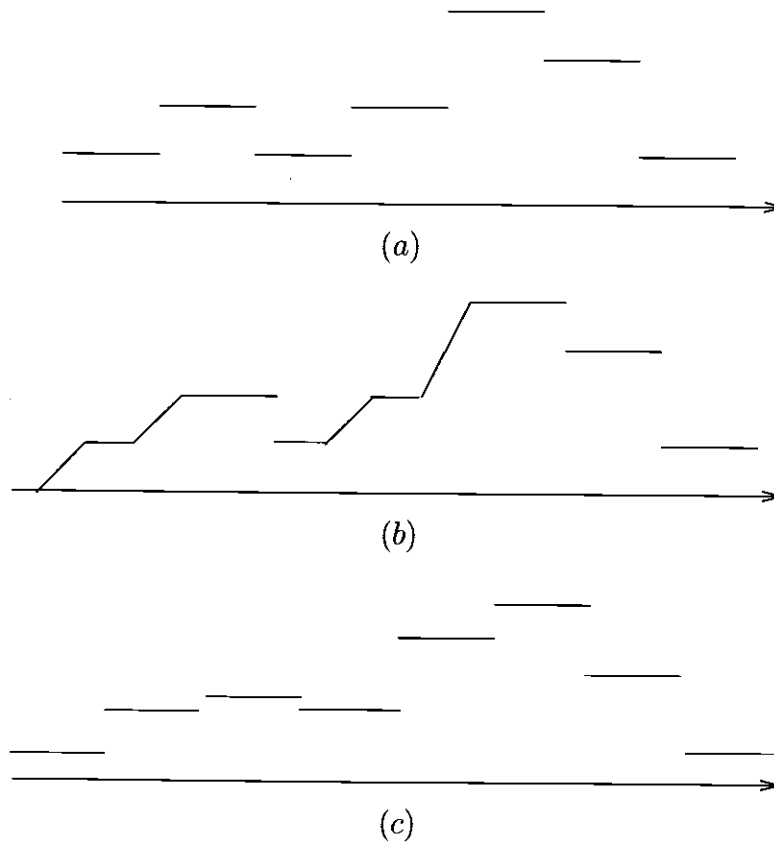

FIGURE 9.7.2. (a) An approximation of the initial condition. (b) The solution obtained by starting with the function given in (a), after enough time has passed to form the two shocks (but not enough to eliminate either). (c) An approximation of the solution given in (b).

pictured in Figure 9.7.2(b) by considering local Riemann problems at each jump. Since they are only Riemann problems locally, we cannot extend the solution given in Figure 9.3.19(b) too far in time. We proceed by approximating the solution given in Figure 9.7.2(b) by a piecewise constant step function. A function that might be this approximation is given in Figure 9.7.2(c). And then, if as before, we let our time proceed again for another small time interval, we obtain a solution with fans "up hill" and jumps "down hill." Again we see that the character of this solution is the same as the character of the continuous solution in that it slopes upward in both increasing parts of the continuous solution and decreases by a series of jump discontinuities in the decreasing parts of the continuous solution. We hope that it is clear that if we repeat this process often with a sufficiently small Δx and Δt, the fans going uphill will approximate the increasing parts of the solution given in Figure 9.7.1(b) and the jump discontinuities will come together to approximate the two jump discontinuities of the solution given in Figure 9.7.1(b).

Of course, there is not enough evidence given above to convince you that all that is claimed is true. The point we wish to make is that it is logical to approximate our initial condition by a piecewise constant function and solve local Riemann problems over small time intervals. We hoped to convince you that the approximate solution retains the same character as the continuous solution. The approach described above is the approach used in what is called the **Godunov scheme.**

We let $\bar{u}^n$ represent the piecewise constant approximate solution to conservation law

$$v_t + F(v)_x = 0, \quad x \in \mathbb{R}, \ t > 0 \tag{9.7.55}$$

along with initial condition $v(x,0) = f(x)$, $x \in \mathbb{R}$ at time $t = n\Delta t$ (the notation introduced at the beginning of Section 9.6). The function $\bar{u}^n$ will be constant on the cells $(x_{k-1/2}, x_{k+1/2})$, and when convenient, we will denote the value of $\bar{u}^n$ at $x = k\Delta x$ (really the value of $\bar{u}$ on the whole interval $(x_{k-1/2}, x_{k+1/2})$) by u_k^n. We denote the solution to conservation law (9.7.55) along with initial condition $v(x,t_n) = \bar{u}^n(x)$, $x \in \mathbb{R}$, by $\bar{U} = \bar{U}(x,t)$. Obviously, we want to use $\bar{U}(x,t)$ at $t = t_{n+1}$ to define our approximation to the solution at time $t = t_{n+1}$, $\bar{u}^{n+1}$, by

$$u_k^{n+1} = \frac{1}{\Delta x} \int_{x_{k-1/2}}^{x_{k+1/2}} \bar{U}(x, t_{n+1})\, dx. \tag{9.7.56}$$

Of course, the solution to conservation law (9.7.55) along with initial condition $\bar{u}^n$ is complex. However, based on the rationale for the Godunov scheme given at the beginning of this section and our discussion of solutions to Riemann problems given in Sections 9.2.2, 9.2.3 and 9.2.5, we know that for sufficiently small Δt the solution will locally look like a series of jumps and fans connected by pieces of constant functions. Some of the jumps and

fans will be moving and some will be stationary. The jumps and fans that are moving will generally move at different speeds and can move in different directions. Of course, this all depends on the flux function F. If we require that Δx and Δt satisfy the CFL condition $R\max|F'(u_k^n)| \leq \frac{1}{2}$ for all k, then the various jumps and fans will not interact with each other. We can reduce this requirement to $R\max|F'(u_k^n)| \leq 1$ for all k, in which case the waves will interact with waves from neighboring Riemann problems, but the interaction will be contained in the control volume.

We continue the procedure for obtaining u_k^{n+1} by solving the Riemann problems associated with each edge of the cells, $x_{k-1/2}$, $k = -\infty, \ldots, \infty$.

$$v_t + F(v)_x = 0, \quad x \in \mathbb{R}, t > t_n \tag{9.7.57}$$

$$v(x, t_n) = \begin{cases} u_{k-1}^n & \text{if } x < x_{k-1/2} \\ u_k^n & \text{if } x \geq x_{k-1/2}. \end{cases} \tag{9.7.58}$$

We integrate conservation law (9.7.57) with respect to x and t from $x_{k-1/2}$ to $x_{k+1/2}$ and t_n to t_{n+1}, respectively. As we have done often before, we perform the integration on the first term with respect to t and on the second term with respect to x. We obtain

$$\int_{x_{k-1/2}}^{x_{k+1/2}} \left[\bar{U}(x, t_{n+1}) - \bar{U}(x, t_n)\right] dx + \int_{t_n}^{t_{n+1}} \left[F(\bar{U}(x_{k+1/2}, t)) - F(\bar{U}(x_{k-1/2}, t))\right] dt = 0 \tag{9.7.59}$$

where we have replaced v in equation (9.7.57) by $\bar{U}$ because we use $\bar{U} = \bar{U}(x, t)$ to denote the solution to initial–value problem (9.7.55) along with initial condition $\bar{U}(x, t_n) = \bar{u}^n(x)$, i.e., the local solution to Riemann problem (9.7.57)–(9.7.58). We note that if we multiply equation (9.7.59) by $1/\Delta x$, use (9.7.56) to identify the first term, and use our initial condition to identify the second term, we can write equation (9.7.59) as

$$u_k^{n+1} = u_k^n - \frac{1}{\Delta x} \int_{t_n}^{t_{n+1}} \left[F(\bar{U}(x_{k+1/2}, t)) - F(\bar{U}(x_{k-1/2}, t))\right] dt. \tag{9.7.60}$$

Since we know that the solution at $x = x_{k+1/2}$ will depend on u_k^n and u_{k+1}^n (in the same way that the solution at $x = x_{k-1/2}$ will depend on u_{k-1}^n and u_k^n), we define the numerical flux function to be

$$h_{k\pm1/2}^n = \frac{1}{\Delta t} \int_{t_n}^{t_{n+1}} F(\bar{U}(x_{k\pm1/2}, t))\, dt, \tag{9.7.61}$$

write equation (9.7.60) as

$$u_k^{n+1} = u_k^n - R\left[h_{k+1/2}^n - h_{k-1/2}^n\right] \tag{9.7.62}$$

and note that the Godunov scheme is a conservative scheme.

If we return to the discussion at the beginning of Section 9.3.1, we recall that locally, the solution $\bar{U}$ is a similarity solution to Riemann problem (9.7.57)–(9.7.58), i.e., $\bar{U}$ an be written as $\bar{U}(x,t) = \psi((x-x_{k-1/2})/(t-t_n))$. Hence, at $x = x_{k-1/2}$, $\bar{U}(x_{k-1/2},t) = \psi(0)$ is constant in t for $t \in [t_n, t_{n+1}]$. Using this fact, the integral in the definition of $h^n_{k-1/2}$, (9.7.61) can be simplified, and we get

$$h^n_{k-1/2} = F(\bar{U}(x_{k-1/2},t)) \tag{9.7.63}$$

(with an analogous simplification in the expression for $h^n_{k+1/2}$).

The next step in our process is to show that *the numerical flux function h is consistent with our flux function F*, i.e., we must show that $h(U,U) = F(U)$. Having the two U's as the argument of $h^n_{k+1/2}$ implies that we should compute the solution to the Riemann problem where both u^n_k and u^n_{k+1} are equal, and equal to U. It is then easy to see that the solution to such a Riemann problem will be $\bar{U}(x,t) = U$, and we have $h^n_{k+1/2}(U,U) = F(\bar{U}(x_{k+1/2},t)) = F(U)$. Hence, $h^n_{k+1/2}$ *is consistent with F.*

Example 9.7.2 Determine the numerical flux function when the Godunov scheme is used to approximate the solution to the inviscid Burgers' equation, $F(v) = v^2/2$.

Solution: we return to Examples 9.2.1, 9.2.3, 9.2.4 and 9.2.8 to see that if $u^n_{k-1} \geq u^n_k$, the solution to Riemann problem (9.7.57)–(9.7.58) is

$$v(x,t) = \begin{cases} u^n_{k-1} & \text{if } (x-x_{k-1/2})/(t-t_n) < s \\ u^n_k & \text{if } (x-x_{k-1/2})/(t-t_n) \geq s \end{cases} \tag{9.7.64}$$

where $s = (u^n_{k-1} + u^n_k)/2$ is the speed of propagation of the discontinuity, and if $u^n_{k-1} < u^n_k$, the solution to Riemann problem (9.7.57)–(9.7.58) is

$$v(x,t) = \begin{cases} u^n_{k-1} & \text{if } \frac{x-x_{k-1/2}}{t-t_n} < u^n_{k-1} \\ \frac{x-x_{k-1/2}}{t-t_n} & \text{if } u^n_{k-1} \leq \frac{x-x_{k-1/2}}{t-t_n} \leq u^n_k \\ u^n_k & \text{if } \frac{x-x_{k-1/2}}{t-t_n} > u^n_k. \end{cases} \tag{9.7.65}$$

Specifically, considering solution (9.7.64), we note that if $u^n_{k-1} \geq u^n_k$, then $\bar{U}(x_{k-1/2},t) = u^n_{k-1}$ if $s > 0$ and $\bar{U}(x_{k-1/2},t) = u^n_k$ if $s \leq 0$. Using solution (9.7.65), we see that if $u^n_{k-1} < u^n_k$, then $\bar{U}(x_{k-1/2},t) = u^n_{k-1}$ if $u^n_{k-1} > 0$, $\bar{U}(x_{k-1/2},t) = 0$ if $u^n_{k-1} \leq 0 \leq u^n_k$, and $\bar{U}(x_{k-1/2},t) = u^n_k$ if $u^n_k < 0$. See HW9.2.4. Hence, we see that for the inviscid Burgers' equation, the numerical flux function associated with the Godunov scheme is given as follows.

If $u^n_{k-1} \geq u^n_k$, then

if $s = \frac{1}{2}\left(u^n_{k-1} + u^n_k\right) > 0$, then $h^n_{k-1/2} = \frac{1}{2}\left(u^n_{k-1}\right)^2$

if $s \leq 0$, then $h^n_{k-1/2} = \frac{1}{2}\left(u^n_k\right)^2$.

If $u^n_{k-1} < u^n_k$, then

if $u^n_{k-1} > 0$, then $h^n_{k-1/2} = \frac{1}{2}\left(u^n_{k-1}\right)^2$

if $u^n_{k-1} \leq 0$ and $u^n_k \geq 0$, then $h^n_{k-1/2} = 0$

if $u_k^n < 0$, then $h_{k-1/2}^n = \frac{1}{2}(u_k^n)^2$.
$h_{k+1/2}^n$ is defined in a similar fashion.

We know from Section 9.2.2 that Riemann problems of the form (9.7.57)–(9.7.58) do not generally have unique solutions. We notice that the solutions that we have given in (9.7.64) and (9.7.65) in Example 9.7.2 are the vanishing viscosity or entropy solutions of the local Riemann problems. The reason for this is that if we want our numerical solution to converge to the entropy solution of the conservation law being solved, it is logical that we should choose the entropy solution to the local Riemann problem (9.7.57)–(9.7.58) as a part of our numerical scheme. We obtain the following result.

Proposition 9.7.21 *If the entropy solutions to the local Riemann problems (9.7.57)–(9.7.58) are chosen in the construction of the Godunov scheme, then there will exist a numerical entropy flux function for which the Godunov scheme will satisfy the discrete entropy condition (9.6.32). If the numerical solution converges to a function v as $\Delta t,\ \Delta x \to 0$, then v will be the entropy solution to the conservation law.*

Proof: We suppose that S is a convex entropy function and Φ is the entropy flux function consistent with conservation law (9.7.55). Since $\bar{U} = \bar{U}(x,t)$, $t_n \le t \le t_{n+1}$, is the entropy satisfying solution to conservation law (9.7.55) with initial condition $\bar{U}(x,t_n) = u^n(x)$, $x \in \mathbb{R}$, $\bar{U}$ must satisfy Entropy Condition II,

$$S(\bar{U})_t + \Phi(\bar{U})_x \le 0, \tag{9.7.66}$$

in the weak sense. We can integrate inequality (9.7.66) (with test function ϕ taken to be unity on $[x_{k-1/2}, x_{k+1/2}] \times [t_n, t_{n+1}]$) from $x = x_{k-1/2}$ to $x = x_{k+1/2}$ with respect to x and from $t = t_n$ to $t = t_{n+1}$ with respect to t (and carry out the obvious integrations) to get

$$\int_{x_{k-1/2}}^{x_{k+1/2}} S(\bar{U}(x,t_{n+1}))\,dx - \int_{x_{k-1/2}}^{x_{k+1/2}} S(\bar{U}(x,t_n))\,dx$$
$$+ \int_{t_n}^{t_{n+1}} \Phi(\bar{U}(x_{k+1/2},t))\,dt - \int_{t_n}^{t_{n+1}} \Phi(\bar{U}(x_{k-1/2},t))\,dt \le 0. \tag{9.7.67}$$

Using the fact that $\bar{U}(x,t_n)$ is constant for $x \in (x_{k-1/2}, x_{k+1/2})$ and $\bar{U}(x_{k\pm 1/2},t)$ are constant for $t \in [t_n, t_{n+1}]$, we can write inequality (9.7.67) as

$$\int_{x_{k-1/2}}^{x_{k+1/2}} S(\bar{U}(x,t_{n+1}))\,dx - \Delta x S(u_k^n) + \Delta t \Phi(\bar{U}(x_{k+1/2},t))$$
$$-\Delta t \Phi(\bar{U}(x_{k-1/2},t)) \le 0. \tag{9.7.68}$$

This is close to what we want. If we define the numerical entropy flux $\Psi^n_{k+1/2} = \Psi^n_{k+1/2}(u^n_k, u^n_{k+1})$ to be

$$\Psi^n_{k+1/2} = \Phi(\bar{U}(x_{k+1/2}, t)),$$

then equation (9.7.68) can be written as

$$\frac{1}{\Delta x}\int_{x_{k-1/2}}^{x_{k+1/2}} S(\bar{U}(x, t_{n+1}))\,dx \leq S(u^n_k) - R\left[\Psi^n_{k+1/2} - \Psi^n_{k-1/2}\right]. \quad (9.7.69)$$

What we need to be able to do is to "take the function S outside of the integral sign." This is a very strange operation, but it is what exactly what Jensen's Inequality allows us to do ([60], page 61). With the hypothesis that S is a convex function, Jensen's Inequality implies that

$$\frac{1}{\Delta x}\int_{x_{k-1/2}}^{x_{k+1/2}} S(\bar{U}(x, t_{n+1}))\,dx \geq S\left(\frac{1}{\Delta x}\int_{x_{k-1/2}}^{x_{k+1/2}} \bar{U}(x, t_{n+1})\,dx\right).$$

We should note that we are doing more than just taking the function S outside of the integral. The fact that we are able to include the $\frac{1}{\Delta x}$ term follows from the hypothesis in Jensen's Inequality that the measure of the interval must be 1. Combining this inequality with inequality (9.7.69) yields

$$S\left(u^{n+1}_k\right) \leq S(u^n_k) - R\left[\Psi^n_{k+1/2} - \Psi^n_{k-1/2}\right].$$

Thus, the solutions to the Godunov scheme satisfy the discrete entropy condition.

The final step we must perform with this entropy argument is to show that Ψ is consistent with Φ. As we had to do when we showed that $h^n_{k+1/2}$ was consistent with F, we must show that $\Psi(U, U) = \Phi(U)$. This is true for the same reason as when we showed the consistency of the numerical flux function. When we take $u^n_k = u^n_{k+1} = U$, the solution to the Riemann problem will be $\bar{U}(x, t) = U$ and $\Psi(U, U) = \Phi(\bar{U}(x_{k+1/2}, t)) = \Phi(U)$. Therefore, *the numerical entropy flux function Ψ is consistent with the entropy flux function Φ.*

Of course, the result that we obtained through this entropy argument is that by Theorem 9.6.6, we know that if the solutions to the Godunov scheme converge as $\Delta t,\ \Delta x \to 0$, then the limiting solution will be an entropy solution to the given conservation law.

As a part of Example 9.7.2, in (9.7.64) and (9.7.65) we give the entropy solution to the Riemann problem

$$\bar{U}_t + F(\bar{U})_x = 0, \quad x \in \mathbb{R},\ t > t_n \quad (9.7.70)$$

$$\bar{U}(x, t_n) = \begin{cases} u^n_{k-1} & \text{if } x - x_{k-1/2} < 0 \\ u^n_k & \text{if } x - x_{k-1/2} \geq 0, \quad x \in \mathbb{R} \end{cases} \quad (9.7.71)$$

for the inviscid Burgers' equation, which we used to define the numerical flux function $h^n_{k\pm1/2}$. For a general convex flux function F (for convenience, take the case when $F'' > 0$), the approach is the same as that used in Example 9.7.2. The entropy solution for Riemann problem (9.7.70)–(9.7.71) is given by

$$\bar{U}(x_{k-1/2}, t_{n+1}) = \begin{cases} u^n_{k-1} & \text{if } F'(u^n_{k-1}),\ F'(u^n_k) \geq 0 \\ u^n_k & \text{if } F'(u^n_{k-1}),\ F'(u^n_k) \leq 0 \\ u^n_{k-1} & \text{if } F'(u^n_{k-1}) \geq 0 \geq F'(u^n_k) \text{ and } s = a^n_{k-1/2} > 0 \\ u^n_k & \text{if } F'(u^n_{k-1}) \geq 0 \geq F'(u^n_k) \text{ and } s = a^n_{k-/2} < 0 \\ u_s & \text{if } F'(u^n_{k-1}) < 0 < F'(u^n_k) \end{cases} \tag{9.7.72}$$

where $a^n_{k-1/2}$ was defined by equation (9.4.17) and u_s is such that $F'(u_s) = 0$.

Remark 1: We note that the first two parts of the solution given in (9.7.72) correspond to either shocks or rarefactions where the shock or rarefaction is to the right or left of the line $x = x_{k-1/2}$ at $t = t_{n+1}$. The third and fourth parts correspond to shocks where the location of the shock is determined by the speed of propagation of the shock, $s = a^n_{k-1/2}$. And finally, the fifth part represents a rarefaction that fans across the line $x = x_{k-1/2}$. The value u_s is that value for which the characteristic speed is zero and is called the **sonic point**.

Remark 2: We note that if we had used

$$\bar{U}(x_{k-1/2}, t_{n+1}) = \begin{cases} u^n_{k-1} & \text{when } a^n_{k-1/2} \geq 0 \\ u^n_k & \text{when } a^n_{k-1/2} < 0 \end{cases} \tag{9.7.73}$$

as the solution to Riemann problem (9.7.70)–(9.7.71) (even though it is not the entropy solution), we would still get the first four parts of solution (9.7.72). The resulting scheme (using (9.7.63) and (9.7.73)) would give upwind scheme (9.4.16), which, as we saw in HW9.4.5, does not produce the entropy solution. This should not surprise us because when $F'(u^n_{k-1}) < 0 < F'(u^n_k)$, equation (9.7.73) is not the entropy solution to Riemann problem (9.7.70)–(9.7.71) at $(x_{k-1/2}, t_{n+1})$. And finally, we see that when we consider the linear one way wave equation, difference scheme (9.7.73) is fine. Since in the linear problem it is not possible to have $F'(u^n_{k-1}) < 0 < F'(U^n_k)$ (because $F' = a$), the first four parts of solution (9.7.72), and hence difference scheme (9.4.16), will handle all possible cases when our conservation law is linear.

Remark 3: We can write flux function (9.7.63) using the solution $\bar{U}$ given

in (9.7.72) as

$$h^n_{k-1/2} = \begin{cases} F^n_{k-1} \text{ if } F'(u^n_{k-1}),\ F'(u^n_k) \geq 0 \\ F^n_k \quad \text{if } F'(u^n_{k-1}),\ F'(u^n_k) \leq 0 \\ F^n_{k-1} \text{ if } F'(u^n_{k-1}) \geq 0 \geq F'(u^n_k) \text{ and } s = a^n_{k-1/2} > 0 \\ F^n_k \quad \text{if } F'(u^n_{k-1}) \geq 0 \geq F'(u^n_k) \text{ and } s = a^n_{k-/2} < 0 \\ F(u_s) \text{if } F'(u^n_{k-1}) < 0 < F'(u^n_k). \end{cases} \tag{9.7.74}$$

The difference scheme (9.7.62), (9.7.74) is the logical upwind scheme that is conservative and produces the entropy solution. When in the future we refer to "the upwind scheme," we will mean the difference scheme associated with numerical flux function (9.7.74). Because of the statement made in Remark 2 above, the easiest way to write numerical flux function (9.7.74) is to write

$$h^n_{k-1/2} = F(u_s) \quad \text{if } F'(u^n_{k-1}) < 0 < F'(u^n_k),$$

otherwise

$$h^n_{k-1/2} = \begin{cases} F^n_{k-1} & \text{if } a^n_{k-1/2} \geq 0 \\ F^n_k & \text{if } a^n_{k-1/2} < 0. \end{cases}$$

Before we leave this section, we note that the Godunov scheme can be written in Q-form by defining the viscosity coefficient

$$Q^n_{k+1/2} = R \max_{(u-u^n_k)(u-u^n_{k+1})\leq 0} \frac{F(u^n_{k+1}) + F(u^n_k) - 2F(u)}{u^n_{k+1} - u^n_k}. \tag{9.7.75}$$

Also, it can be shown that the numerical flux functions associated with the Godunov scheme can be written as

$$h^n_{k+1/2} = \begin{cases} \min_{u^n_k < u < u^n_{k+1}} F(u) & \text{if } u^n_k < u^n_{k+1} \\ \max_{u^n_{k+1} < u < u^n_k} F(u) & \text{if } u^n_k > u^n_{k+1}. \end{cases} \tag{9.7.76}$$

See HW9.7.11. From this formulation of $h^n_{k+1/2}$ we can easily see that the Godunov scheme is an E scheme. In fact, reviewing the definition of E scheme, we see that the Godunov scheme is the limiting E scheme. Since the Godunov scheme is an E scheme, *the Godunov scheme will be TVD and at most first order accurate.*

HW 9.7.10 (a) Write the Godunov scheme for the one way wave equation $v_t + av_x = 0$.
(b) Show that the Godunov scheme written in part (a) is the same as the upwind scheme (9.4.15) or (9.4.16) applied to the one way wave equation.

HW 9.7.11 (a) Show that the Godunov scheme can be written in Q-form with $Q^n_{k+1/2}$ given by (9.7.75).
(b) Show that the Godunov scheme is conservative with the numerical flux function given by (9.7.76).

HW 9.7.12 Use the Godunov scheme to solve the initial-boundary-value problems given in HW9.4.3 and HW9.4.5.

HW 9.7.13 Show that the Godunov scheme is a monotone scheme.

9.7.4 High Resolution Schemes

In previous sections we have studied a variety of types of schemes, none of which satisfy us in that they are conservative, produce a solution that will converge to the vanishing viscosity solution, and resolve shocks. In this section we will derive a class of high resolution difference schemes for scalar conservation laws. We use the term **high resolution schemes** to describe schemes that will generally (i.e., at most places) be second order accurate (or higher, though our emphasis will be on second order accurate schemes) on smooth sections of the solution, nonoscillatory and capable of accurately resolving discontinuities in the solution. We will not attempt to give a complete account of all of the schemes that are available. We will introduce the reader to some of the schemes and some of the approaches used to develop these schemes. From the results presented in Section 9.7.2 we saw that to find a high resolution scheme, (i) we must use a nonlinear scheme, (ii) we need not even try three-point schemes, and (iii) the most we can expect is that our scheme will be second order accurate or higher away from the extreme points of the solution (that are not sonic points). The third of these results is the most disconcerting. It is for this reason that difference scheme developers sometimes relax the TVD requirement (to require that our scheme be an ENO scheme or something else less restrictive than TVD) in an attempt to obtain better accuracy.

The approach that we will use is similar to an approach we introduced in Section 7.6, where we introduced the concept of artificial dissipation and showed how artificial dissipation could be used to stabilize a scheme and how artificial dissipation could be used to eliminate the oscillations that appear in the solution obtained by a stable convergent scheme. The difficulty with the approach used in Section 7.6 is that the same amount of dissipation is added everywhere. With certain test problems we were able to adjust what (second or fourth order dissipation) and how much dissipation we add so that we can obtain a good solution. This approach is very experimental and is difficult when we are solving problems for which we do not know the solution. The methods given in the following sections can be viewed as adding artificial dissipation where it is needed where the solution determines where more dissipation and how much dissipation is needed.

We will introduce three popular approaches for finding high resolution schemes: flux-limiter methods, slope-limiter methods and modified flux

methods. We emphasize that the approaches are not disjoint, and there are other approaches that overlap with the approaches given here and each other. Often a scheme developed by one of these approaches can also be found by one of the other approaches. We present these methods as three approaches that can be used to find high resolution schemes.

9.7.5 Flux-Limiter Methods

The approach that we will use here to construct a high resolution scheme is to define the numerical flux function of the new scheme in terms of numerical flux functions of some of our earlier schemes. In particular, we write the numerical flux function of our scheme as

$$h^n_{k+1/2} = h^n_{L_{k+1/2}} + \phi^n_k \left[h^n_{H_{k+1/2}} - h^n_{L_{k+1/2}} \right] \tag{9.7.77}$$

where h_L and h_H are numerical flux functions of a lower order scheme and a high order scheme, respectively, and ϕ^n_k is a coefficient that is yet to be defined. The idea is to develop a scheme where the numerical flux function has the smoothing capabilities of the lower order scheme when it is necessary and the accuracy of the higher order scheme when it is possible. Thus, we will want to define ϕ^n_k such that ϕ^n_k is near 1 on smooth sections of the solution ($h^n_{k+1/2}$ will be approximately equal to $h^n_{H_{k+1/2}}$) and near 0 on sections of the solution that contain steep gradients or discontinuities ($h^n_{k+1/2}$ will be approximately equal to $h^n_{L_{k+1/2}}$). We should note that when we formulate the scheme by defining the numerical flux function, the resulting scheme will automatically be conservative. We should always remember that we would still like our scheme to be TVD and possess a numerical entropy function that satisfies the discrete entropy condition (9.6.32) (be such that the solution will converge to the entropy or vanishing viscosity solution). Since the scheme with flux given by h_L is considered to be too diffusive, the correction term $\phi^n_k \left[h^n_{H_{k+1/2}} - h^n_{L_{k+1/2}} \right]$ is referred to as the **antidiffusive flux**. We try to add as much antidiffusive flux as possible without increasing the variation of the solution. This approach to developing high resolution schemes is very similar to (if not the same as) the flux corrected transport scheme developed by Boris and Book, ref. [6].

We also note that if h_L and h_H are numerical flux functions associated with three-point schemes (which is surely an easy way to start), the ϕ^n_k term must be used to make the resulting scheme such that we reach to more than three points (recall Proposition 9.7.19). Also, the combination of h_L, h_H and ϕ^n_k must be such that the resulting scheme is nonlinear (Proposition 9.7.14).

And finally, we see that we can just as easily write $h^n_{k+1/2}$ as

$$h^n_{k+1/2} = h^n_{H_{k+1/2}} - \left(1 - \phi^n_k\right) \left[h^n_{H_{k+1/2}} - h^n_{L_{k+1/2}} \right].$$

Thus we see that the flux-limiter method can be viewed either as perturbing the flux of a low order scheme or perturbing the flux of a high order scheme.

9.7.5.1 Flux-Limiter Schemes for the One Way Wave Equation

To make the analysis of the resulting schemes easier, we shall follow Sweby as in ref. [66] and begin by developing a class of high resolution schemes for the linear hyperbolic partial differential equation

$$v_t + av_x = 0. \tag{9.7.78}$$

We begin by writing the numerical flux function associated with the Lax-Wendroff scheme for solving equation (9.7.78) as

$$h^n_{k+1/2} = au^n_k + \frac{1}{2}a(1 - aR)\delta_+ u^n_k. \tag{9.7.79}$$

The numerical flux function given by (9.7.79) can be viewed as $h = h_L + (h_H - h_L)$, where

$$h^n_{L_{k+1/2}} = au^n_k \tag{9.7.80}$$

and

$$h^n_{H_{k+1/2}} = au^n_k + \frac{1}{2}a(1 - aR)\delta_+ u^n_k. \tag{9.7.81}$$

The function h_L is the numerical flux function associated with the first order FTBS scheme, and obviously, h_H is the numerical flux function associated with the Lax-Wendroff scheme. Since the FTBS scheme is unstable when $a < 0$, we will assume for now that $a > 0$. We shall see later that this assumption causes no problems and is not necessary.

We proceed in our endeavor to construct high resolution schemes by replacing the numerical flux function (9.7.79) by

$$h^n_{k+1/2} = h^n_{L_{k+1/2}} + \phi^n_k \left[h^n_{H_{k+1/2}} - h^n_{L_{k+1/2}} \right] = au^n_k + \phi^n_k \frac{1}{2}a(1 - aR)\delta_+ u^n_k \tag{9.7.82}$$

where as before, ϕ^n_k is yet to be defined. The function ϕ^n_k is referred to as the **flux-limiter** and is chosen to be nonnegative so as to maintain the sign of the antidiffusive flux. Clearly, the numerical flux function (9.7.82) is constructed by using $h = h_L + \phi(h_H - h_L)$ where h_L and h_H are the numerical flux functions of the FTBS and Lax-Wendroff scheme, respectively.

In the case of numerical flux function (9.7.82), we want ϕ^n_k to be near 1 on smooth sections of the solution and near 0 near discontinuities. There are many ways that we could try to define ϕ^n_k so as to accomplish our goals.

A common approach is to write $\phi_k^n = \phi(\theta_k^n)$ where θ_k^n is referred to as the **smoothness parameter** and will be defined at this time as

$$\theta_k^n = \frac{\delta_- u_k^n}{\delta_+ u_k^n}. \tag{9.7.83}$$

The function ϕ is still undetermined. We note that θ_k^n as defined in (9.7.83) is used to measure the smoothness of the solution, because when the solution is not changing much, θ_k^n will be approximately 1, and when the solution is changing a significant amount, θ_k^n will be very different from 1. The difference scheme that we will then be considering is

$$\begin{aligned} u_k^{n+1} =& u_k^n - R\left[h_{k+1/2}^n - h_{k-1/2}^n\right] \\ =& u_k^n - aR\delta_- u_k^n - \frac{1}{2}aR(1-aR)\phi_k^n \delta_+ u_k^n \\ &+ \frac{1}{2}aR(1-aR)\phi_{k-1}^n \delta_+ u_{k-1}^n \qquad (9.7.84) \\ =& u_k^n - aR\delta_- u_k^n - \frac{1}{2}aR(1-aR)\delta_-\left(\phi_k^n \delta_+ u_k^n\right). \qquad (9.7.85) \end{aligned}$$

With different choices of ϕ in difference scheme (9.7.85), we get different schemes. Of course, $\phi = 0$ will give just the FTBS scheme. We might also note that $\phi(\theta) = 1$ will give us the Lax-Wendroff scheme and $\phi(\theta) = \theta$ will give us the Beam-Warming scheme.

Thus it now remains for us to choose ϕ in such a way that the resulting difference scheme is TVD and second order (except at the extrema of the solution that are not sonic points—Proposition 9.7.20). The approach that we shall use to ensure that our scheme is TVD is to require that our scheme be incrementally TVD and use Proposition 9.7.17. Of course, this is the same as writing the scheme in incremental form and using Proposition 9.7.6 to force the scheme to be TVD. We rewrite difference scheme (9.7.84) in I-form as

$$u_k^{n+1} = u_k^n - \left[aR - \frac{1}{2}aR(1-aR)\phi_{k-1}^n\right]\delta_- u_k^n - \frac{1}{2}aR(1-aR)\phi_k^n \delta_+ u_k^n \tag{9.7.86}$$

with

$$C_{k+1/2}^n = -\frac{1}{2}aR(1-aR)\phi_k^n$$

and

$$D_{k-1/2}^n = aR - \frac{1}{2}aR(1-aR)\phi_{k-1}^n.$$

It is clear that when $0 \le aR \le 1$ (we must still satisfy the CFL condition) and ϕ_k^n is near 1 (smooth sections of the solution), $C_{k+1/2}^n$ will be less than zero, and hence, difference scheme (9.7.86) will not be incremental TVD.

The above revelation is not a fatal flaw in our approach. We can instead write difference scheme (9.7.84) in I-form as

$$u_k^{n+1} = u_k^n - \left\{ aR - \frac{1}{2}aR(1-aR)\phi_{k-1}^n + \frac{1}{2}aR(1-aR)\phi_k^n \frac{\delta_+ u_k^n}{\delta_- u_k^n} \right\} \delta_- u_k^n. \tag{9.7.87}$$

Then we have

$$C_{k+1/2}^n = 0$$

and

$$D_{k-1/2}^n = aR - \frac{1}{2}aR(1-aR)\phi_{k-1}^n + \frac{1}{2}aR(1-aR)\phi_k^n \frac{\delta_+ u_k^n}{\delta_- u_k^n}. \tag{9.7.88}$$

We are taking advantage of the very important fact that *the I-form representation of a difference scheme is not unique*. Conditions (9.7.20) are then satisfied if

$$0 \le D_{k-1/2}^n \le 1. \tag{9.7.89}$$

If we rewrite $D_{k-1/2}^n$ as

$$D_{k-1/2}^n = aR - \frac{1}{2}aR(1-aR)\phi(\theta_{k-1}^n) + \frac{1}{2}aR(1-aR)\frac{\phi(\theta_k^n)}{\theta_k^n}, \tag{9.7.90}$$

we see that if ϕ satisfies

$$\left| \frac{\phi(\theta_k)}{\theta_k} - \phi(\theta_{k-1}) \right| \le 2 \text{ for all } k, \tag{9.7.91}$$

then inequality (9.7.89) is satisfied. (See HW9.7.14.)

We now use inequality (9.7.91) to help us to choose the form of the limiter function ϕ. In addition to requiring that the function ϕ be nonnegative, we assume that $\phi(\theta) = 0$ for $\theta \le 0$. We note that inequality (9.7.91) will be satisfied if we require that ϕ satisfy

$$0 \le \frac{\phi(\theta)}{\theta} \le 2 \text{ and } 0 \le \phi(\theta) \le 2 \tag{9.7.92}$$

for all θ. See HW9.7.15. Thus we choose the function ϕ such that the graph of ϕ lies within the shaded region given in Figure 9.7.3.

Remark: We should understand that the above argument is difficult. We saw that if we write difference scheme (9.7.85) in I-form as in (9.7.86), the scheme is not incrementally TVD. Rewriting the scheme as in (9.7.88) allows us to prove that the scheme is incrementally TVD. The difference scheme has not changed. This proof emphasizes that the property of being incrementally TVD is dependent on the I-form representation.

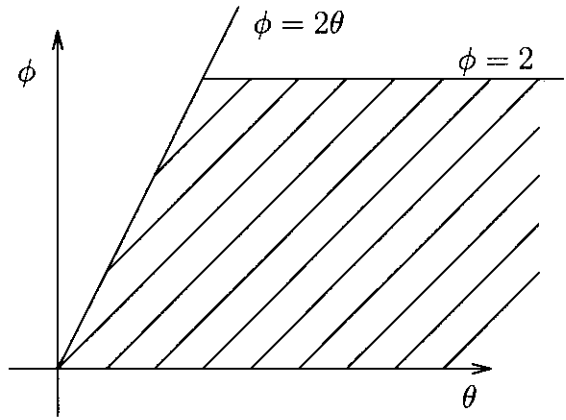

FIGURE 9.7.3. The shaded region represents the region defined by inequality (9.7.92) for $\theta \geq 0$.

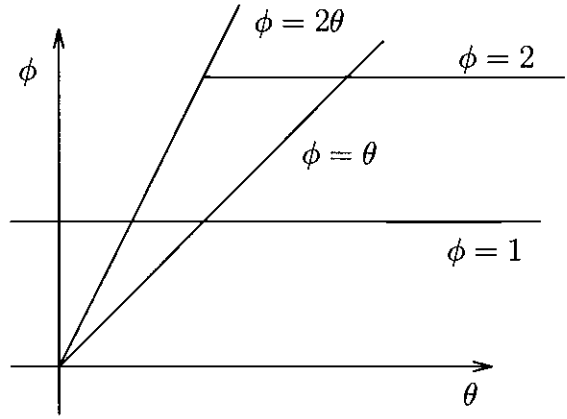

FIGURE 9.7.4. Region shown in Figure 9.7.3 with plots of limiter functions associated with the Lax-Wendroff scheme and the Beam-Warming scheme.

There are obviously many functions that can be chosen that fall into the shaded region given in Figure 9.7.3. The following result gives us some general properties of difference scheme (9.7.85).

Proposition 9.7.22 *If the flux-limiter function ϕ is bounded, then difference scheme (9.7.85) is consistent with the one-way wave equation (9.7.78). If $\phi(1) = 1$ and ϕ is Lipschitz continuous at $\theta = 1$, then difference scheme (9.7.85) is second order accurate on smooth solutions with v_x bounded away from zero.*

Proof: The proof of the first statement above is easy in that the proof is very similar to that for the Lax-Wendroff scheme. The left side along with the first two terms of the right side of equation (9.7.85) will give consistency with partial differential equation (9.7.78). The last term of (9.7.85) can easily be seen to be a second order term (first order when we divide by Δt). Hence, we have consistency.

We prove a slight variation of the second claim. So that we can apply Proposition 9.7.12, we prove the second claim under the assumption that ϕ is differentiable at $\theta = 1$. The proof using the assumption that ϕ is only Lipschitz continuous at $\theta = 1$ is a slight, technical variation of the proof given. For this application of Proposition 9.7.12, we note that we write our difference scheme as

$$\begin{aligned} u_k^{n+1} &= Q(u_{k-2}^n, \cdots, u_{k+1}^n) \\ &= u_k^n - aR\delta_- u_k^n - \frac{1}{2}aR(1 - aR)\delta_-\left(\phi_k^n \delta_+ u_k^n\right), \end{aligned}$$

where $p = q = 1$ and we emphasize the fact that ϕ_k^n and ϕ_{k-1}^n depend on u_{k-1}^n, u_k^n, u_{k+1}^n and u_{k-2}^n, u_{k-1}^n, u_k^n, respectively. An easy calculation shows that

$$q(u) = \frac{1}{2}\left[\frac{1}{R^2}\left(aR - aR(1 - aR)\phi(1)\right) - \left(F'(u)\right)^2\right].$$

Since for conservation law (9.7.78) $F'(u) = a$, it is easy to see that when $\phi(1) = 1$, $q(u) = 0$. Since $q = 0$ for all u, we have $q' = 0$, and by Proposition 9.7.12, the difference scheme is at least of second order.

Remark: We see that condition (9.7.92) already requires that ϕ satisfy $\phi(\theta) \leq 2$, and any difference scheme where ϕ satisfies condition (9.7.92) will be consistent with partial differential equation (9.7.78). The conditions for the scheme to be second order do not limit the choice of the function greatly. The primary requirement is that the graph of ϕ passes through the point $(1, 1)$. We recall that by Proposition 9.7.20 the above results are about the best that we can do. We are unable to produce a scheme that will be second order accurate at extrema that are not sonic points.

Our next step is to construct functions ϕ that lie in the shaded region of Figure 9.7.3 and pass through the point $(1, 1)$. In Figure 9.7.4 we replot

the shaded region along with the graphs of the limiter functions $\phi(\theta) = 1$ associated with the Lax-Wendroff scheme and $\phi(\theta) = \theta$ associated with the Beam-Warming scheme. We note that the limiter functions associated with both of these schemes pass through the point $(1, 1)$ (but we already knew that they were both second order schemes) and are not completely contained in the shaded (TVD) region (we also already knew that neither the Lax-Wendroff nor the Beam-Warming scheme is a TVD scheme). Specifically, we note that as θ gets large ($\delta_+ u_k^n$ is small and $\delta_- u_k^n$ is a reasonable size or large), the Beam-Warming scheme is not TVD. Similarly, we see that when θ is small ($\delta_+ u_k^n$ is small and $\delta_- u_k^n$ is a reasonable size or large or when $\delta_+ u_k^n$ is a reasonable size and $\delta_- u_k^n$ is large), the Lax-Wendroff scheme is not TVD. Since the above-described situations all occur near fronts or discontinuities in the solution, both the Lax-Wendroff and the Beam-Warming schemes fail to be TVD in the vicinity of fronts or discontinuities. The wiggles in the numerical solution that we have seen near discontinuities (most often computing with the Lax-Wendroff scheme) cause the increase in the variation that makes these schemes be not TVD.

The various high resolution schemes have been developed by a variety of means. Rather than trying to motivate the schemes or put the schemes in any historical perspective, the approach that we shall take here is to simply define the appropriate flux-limiter function and reference the scheme. One possible choice for ϕ is to choose ϕ such that we maximize the antidiffusive flux that we add to the first order scheme (subject to the TVD constraints),

$$\phi(\theta) = \min\{2\theta, 2\}, \quad \theta > 0.$$

We should note that the graph of this function is the upper boundary of the TVD region plotted in Figure 9.7.3. However, since $\phi(1) \neq 1$, this limiter will not give us a scheme that is second order accurate. Some of the choices for limiter functions that are in the literature include the following.

Superbee limiter, ref. [59]

$$\phi(\theta) = \max\{0, \min[1, 2\theta], \min[\theta, 2]\} \tag{9.7.93}$$

Van Leer limiter, ref. [36]

$$\phi(\theta) = \frac{|\theta| + \theta}{1 + |\theta|} \tag{9.7.94}$$

C-O limiter, ref. [55]

$$\phi(\theta) = \max\{0, \min[\theta, \psi]\}, \quad 1 \leq \psi \leq 2 \tag{9.7.95}$$

BW-LW limiter

$$\phi(\theta) = \max\{0, \min[\theta, 1]\}. \tag{9.7.96}$$

We should first note that the BW-LW limiter (named because it is a very obvious combination of the Beam-Warming and Lax-Wendroff schemes) is a special case of the C-O limiter ($\psi = 1$). We include this special case separately because it is such an obvious possibility. We should also notice that the limiter functions (9.7.93)–(9.7.96) satisfy conditions (9.7.85) (the graphs of the functions all are included in the shaded region plotted in Figure 9.7.3 and are plotted in Figure 9.7.6). In addition, we note that the functions are all Lipschitz continuous, but they are not all differentiable.

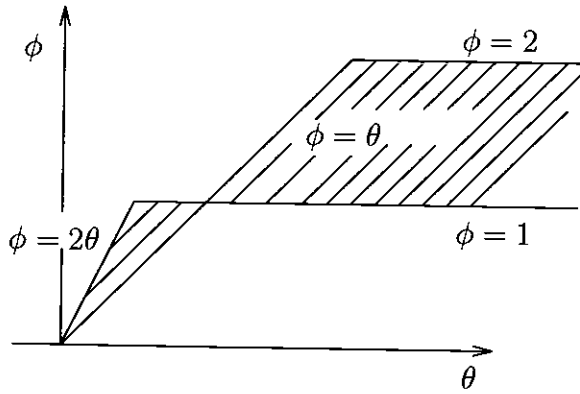

FIGURE 9.7.5. Plot of the region representing limiter functions that are weighted averages of the Lax-Wendroff and Beam-Warming limiter functions.

Of course, there are many other possible limiter functions. In ref. [66], Sweby states that it is best to choose the limiter function so that the new flux function is a weighted average of the Lax-Wendroff and the Beam-Warming flux functions. If this is done, we see that the graph of the limiter function must lie in the shaded region plotted in Figure 9.7.5. We note that this region is such that any limiter function with a graph lying in this region must satisfy $\phi(1) = 1$. In addition, we see in Figure 9.7.6 that the graphs of all of the limiter functions (9.7.93)–(9.7.96) will lie in the region plotted in Figure 9.7.5.

We might note that all of the limiter functions that we have discussed have been monotone increasing functions. Another property that some of these functions have is a symmetry property. We say that the limiter function is **symmetric** if

$$\frac{\phi(\theta)}{\theta} = \phi\left(\frac{1}{\theta}\right). \tag{9.7.97}$$

Symmetry of the limiter function ensures that difference scheme (9.7.85) will treat backward and forward facing gradients in the same way. It is not difficult to see that both the Superbee and Van Leer limiters are symmetric,

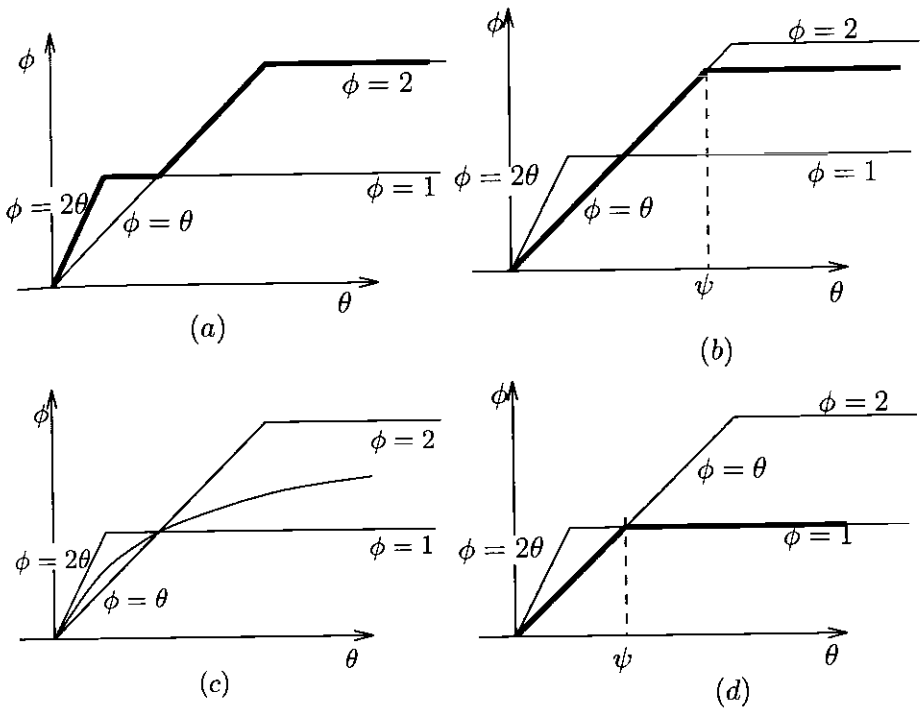

FIGURE 9.7.6. Plots of (a) the Superbee limiter function, (b) the C-O limiter function, (c) the Van Leer limiter function and (d) the BW-LW limiter function.

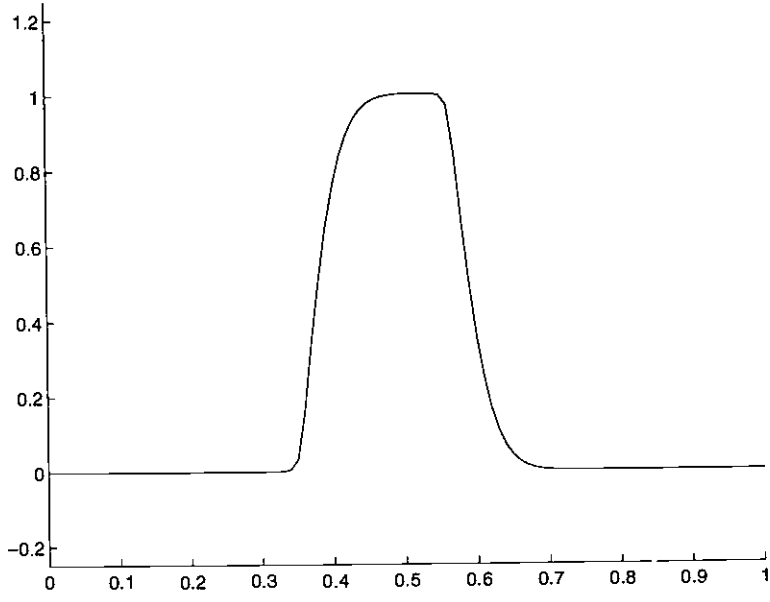

FIGURE 9.7.7. Approximate solution to initial–boundary–value problem (9.7.98)–(9.7.101). The solution was found using the C-O limiter, $\psi = 2.0$,

whereas for $\psi > 1$ the C-O limiter is not symmetric. See HW9.7.16. The effects of the limiter functions being not symmetric can be seen in the numerical results given in Figure 9.7.7. The solution plotted in Figure 9.7.7 is the approximate solution to the initial-boundary value problem

$$v_t - v_x = 0, \quad x \in (0,1), \ t > 0 \tag{9.7.98}$$

$$v(x,0) = f(x), \quad x \in [0,1] \tag{9.7.99}$$

$$v(0,t) = v(1,t), \quad t \geq 0 \tag{9.7.100}$$

where f is defined as

$$f(x) = \begin{cases} 1 & \text{if } 0.4 \leq x \leq 0.6 \\ 0 & \text{otherwise.} \end{cases} \tag{9.7.101}$$

The solution plotted in Figure 9.7.7 was obtained by using difference scheme (9.7.84) using the C-O limiter with $\psi = 2.0$. If we compare the solution on the two sides of the square wave, we see that both sides might be sufficiently accurate, but they are different. If we look carefully (or actually look at the numbers) at the results we get from HW9.7.19, we see that there is no difference between the two sides of the wave for the computations using the Superbee and the Van Leer limiter.

We remind the reader that all we have done so far in this section was based on the assumption that $a > 0$. This is not an acceptable assumption, and as we stated earlier, it is not really necessary (the assumption was made purely for convenience). We could return to the beginning of the section and repeat all we have done using the assumption that $a < 0$. As we see in HW9.7.17, when $a < 0$ we obtain the numerical flux function

$$h^n_{k+1/2} = au^n_{k+1} - \phi^n_k \frac{1}{2} a(1 + aR)\delta_+ u^n_k, \tag{9.7.102}$$

where $\phi^n_k = \phi(\theta^n_k)$ and $\theta^n_k = \delta_+ u^n_{k+1}/\delta_+ u^n_k$. However, the correct way to do the analysis is to not worry about the sign of a and use the general upwind scheme as our low order scheme, i.e., the numerical flux function

$$h^n_{L_{k+1/2}} = \frac{1}{2} a(u^n_k + u^n_{k+1}) - \frac{1}{2}|a|(u^n_{k+1} - u^n_k). \tag{9.7.103}$$

Of course, numerical flux function (9.7.103) reduces to that for the FTBS scheme when $a > 0$ and to that for the FTFS scheme when $a < 0$. Then, using the Lax-Wendroff scheme as the high order scheme, we obtain a high resolution scheme with numerical flux function

$$h^n_{k+1/2} = h^n_{L_{k+1/2}} + \frac{1}{2}\phi^n_k a\big(\text{sign}(a) - aR\big)\delta_+ u^n_k \tag{9.7.104}$$

where $\text{sign}(a)$ denotes the sign of a, $h^n_{L_{k+1/2}}$ is given by (9.7.103), $\phi^n_k = \phi(\theta^n_k)$, and

$$\theta^n_k = \frac{\delta_- u^n_k}{\delta_+ u^n_k} \quad \text{when } a > 0 \tag{9.7.105}$$

and

$$\theta_k^n = \frac{\delta_+ u_{k+1}^n}{\delta_+ u_k^n} \quad \text{when } a < 0. \tag{9.7.106}$$

It is easy to see that numerical flux function (9.7.104) reduces to numerical flux functions (9.7.102) and (9.7.82) when $a < 0$ and $a > 0$, respectively. See HW9.7.18.

Remark 1: Based on the results we obtained in problems HW9.4.4 and HW9.4.5, we must be careful when we use an upwind scheme for any part of our scheme. Since in this case we are considering linear problems, there is no problem caused by using the upwind scheme.

Throughout most of this chapter, as we have developed schemes, we have always strived to obtain schemes that possess three basic properties. We wanted schemes that are conservative, TVD and that give entropy (vanishing viscosity) solutions. At the moment, we have obtained some high resolution schemes that are TVD and conservative. We are not able to prove that the high resolution schemes developed in this section yield entropy solutions. Based on the results obtained when computing with upwind scheme (9.4.16) in HW9.4.5 and the Lax-Wendroff scheme (9.5.10) in HW9.6.6, we should as least be suspicious of any scheme developed using the upwind scheme and the Lax-Wendroff scheme. The difficulties with both of these schemes occurred when computing fans, which occur only when we have nonlinear problems. For this reason, the difficulties that the upwind scheme and the Lax-Wendroff scheme have in obtaining the vanishing viscosity solutions in problems HW9.4.5 and HW9.6.6 are not relevant in this section. Numerical evidence implies that the solutions obtained using the flux-limiter schemes developed in this section will generally yield entropy solutions. In ref. [66], Sweby conjectures that the flux-limiter schemes inherit the property that they give entropy solutions from the corresponding property satisfied by the lower order scheme that is used (provided that by addition of the limited antidiffusive flux, the diffusion at expansions is not decreased). It is, however, possible to obtain entropy violating solutions using flux-limiter schemes. Roe's scheme, which uses the Murman upwind scheme, ref. [49], as the first order scheme along with Roe's Superbee limiter, has been shown to admit entropy violating shocks. By replacing the Murman scheme by an E scheme, these entropy violating solutions are eliminated.

HW 9.7.14 Verify that inequality (9.7.89) follows from condition (9.7.91) (along with the requirement that the CFL condition $0 \leq aR \leq 1$ is satisfied), i.e.,

(a) show that if $0 \leq aR \leq 1$, then

$$aR - \frac{1}{2}aR(1-aR)\left|\frac{\phi(\theta_k)}{\theta_k} - \phi(\theta_{k-1}\right| \leq D_{k-1/2}$$
$$\leq aR + \frac{1}{2}aR(1-aR)\left|\frac{\phi(\theta_k)}{\theta_k} - \phi(\theta_{k-1}\right|, \qquad (9.7.107)$$

and
(b) if ϕ satisfies

$$\left|\frac{\phi(\theta_k)}{\theta_k} - \phi(\theta_{k-1})\right| \leq 2,$$

then inequality (9.7.107) implies that inequality (9.7.89) is satisfied.

HW 9.7.15 Show that if ϕ satisfies

$$0 \leq \frac{\phi(\theta)}{\theta} \leq 2 \quad \text{and} \quad 0 \leq \phi(\theta) \leq 2$$

for all θ, then ϕ satisfies inequality (9.7.91).

HW 9.7.16 Verify that the Superbee and Van Leer limiters are symmetric. In addition, show that when $\psi > 1$, the C-O limiter is not symmetric and when $\psi = 0$ (BW-LW limiter), the C-O limiter is symmetric.

HW 9.7.17 (a) Verify that if we use the FTFS scheme as the low order scheme and the Lax-Wendroff scheme as the high order scheme, we obtain a difference scheme with numerical flux function (9.7.102).
(b) Verify that if $a < 0$ and the CFL condition $-1 \leq aR \leq 0$ is satisfied, the difference scheme with numerical flux function given by (9.7.102) is TVD whenever ϕ satisfies conditions (9.7.92).

HW 9.7.18 Verify that numerical flux function (9.7.104) reduces to flux functions (9.7.103) and (9.7.82) when $a < 0$ and $a > 0$, respectively.

HW 9.7.19 Use the conservative difference scheme defined by numerical flux function (9.7.104) and θ_k^n defined by (9.7.105) and (9.7.106) (which is the same scheme as the scheme defined by numerical flux function (9.7.102) with $\theta_k^n = \delta_+ u_{k+1}^n / \delta_+ u_k^n$) to solve initial–boundary–value problem (9.7.98)–(9.7.101). (Recall that this is a problem that we solved several times in Section 7.8 with limited success.) Use $\Delta x = 0.01$, $\Delta t = 0.005$, (a) the Superbee limiter, (9.7.93), and (b) the Van Leer limiter, (9.7.94). Plot the solution at time $t = 1.0$. Compare and contrast these solutions with each other and with those given in Section 7.8.

9.7.5.2 Flux-Limiter Schemes for Nonlinear Conservation Laws

We now return to the problem of approximating solutions to general scalar conservation laws. An approach we will use is to extend the ideas developed for the one-way wave equation developed in the last section to apply to approximating solutions to conservation law (9.7.1). We follow the approach described in Section 9.7.5 and construct a numerical flux function of the form

$$h^n_{k+1/2} = h^n_{L_{k+1/2}} + \phi^n_k \left[h^n_{H_{k+1/2}} - h^n_{L_{k+1/2}} \right].$$

The most obvious analogy to the scheme developed in Section 9.7.5.1 is to use the nonlinear upwind scheme (difference scheme (9.4.16)) as our low order scheme and the nonlinear Lax-Wendroff scheme (difference scheme (9.5.10)) as our high order scheme. Hence, if we let

$$h^n_{L_{k+1/2}} = \frac{1}{2}\left[F^n_k + F^n_{k+1}\right] - \frac{1}{2}|a^n_{k+1/2}|\delta_+ u^n_k \tag{9.7.108}$$

and

$$h^n_{H_{k+1/2}} = \frac{1}{2}\left[F^n_k + F^n_{k+1}\right] - \frac{R}{2}\left(a^n_{k+1/2}\right)^2 \delta_+ u^n_k, \tag{9.7.109}$$

we obtain a difference scheme with numerical flux function

$$h^n_{k+1/2} = h^n_{L_{k+1/2}} + \phi^n_k \frac{|a^n_{k+1/2}|}{2}\left[1 - R|a^n_{k+1/2}|\right]\delta_+ u^n_k. \tag{9.7.110}$$

Since we are using the general upwind scheme as our low order scheme, we use a definition of the smoothness parameter θ^n_k similar to that given in (9.7.105) and (9.7.106), i.e., we use

$$\theta^n_k = \begin{cases} \frac{\delta_- u^n_k}{\delta_+ u^n_k} & \text{when } a^n_{k+1/2} > 0 \\ \frac{\delta_+ u^n_{k+1}}{\delta_+ u^n_k} & \text{when } a^n_{k+1/2} < 0. \end{cases} \tag{9.7.111}$$

Note that when $a^n_{k+1/2} = 0$, it does not matter how we define ϕ^n_k. Also, when $\delta_+ u^n_k = 0$ (so that there are problems defining θ^n_k and the calculation of $a^n_{k+1/2}$ becomes more difficult), the second term on the right of equation (9.7.110) is zero, and we can define θ^n_k and $a^n_{k+1/2}$ anyway that makes the coding easier.

Of course, the analysis showing that difference scheme (9.7.85) is consistent, second order accurate away from zeros of v_x and TVD performed in the last section does not hold for the difference scheme associated with numerical flux function (9.7.110). We have the choice of trying to extend the analysis performed in the last section to include the difference scheme defined by the numerical flux function $h^n_{k+1/2}$ or to proceed with the new difference scheme with care and numerical experimentation. At the moment, we choose the latter route. We emphasize that we assume that the

new difference scheme will inherit the CFL condition from both the low and high order schemes. For this reason it is good to chose low and high order schemes with compatible CFL conditions. If we used either the non-linear FTFS or FTBS schemes as our low order schemes (and this would surely be entirely possible), we would have a scheme with a restrictive CFL condition.

Remark: As we stated in the previous section, the results of the computations done in problems HW9.4.5 and HW9.6.6 should raise doubt about whether or not we can obtain the vanishing viscosity solution when we use the difference scheme based on the numerical flux function (9.7.110). As we see in HW9.7.23 (the part concerning the problem given in HW9.4.5, i.e., the problem with a transonic rarefaction) our suspicions are well founded. When there is a situation where we do not want to take the chance that the scheme might not resolve the fan, we might want to use the Godunov scheme (9.7.62), (9.7.74). See HW9.7.21.

If we generalized the results of Section 9.7.5.1 even further, we could build schemes where we choose any monotone or E scheme as our low order scheme and any high order scheme (though we do not have many to choose from). For example, if we were to choose the Lax-Friedrichs scheme as our low order scheme and the Beam-Warming scheme as our high order scheme, we would be left with the difference scheme with numerical flux function

$$h^n_{k+1/2} = \frac{1}{2}\left[F^n_k + F^n_{k+1}\right] - \frac{1}{2R}\delta_+ u^n_k + \phi^n_k\left\{\frac{1}{2}\left[F^*_k + F^n_{k+1} + \frac{1}{2}\delta_- F^n_k + \frac{1}{2R}\delta_+ u^n_k\right\}\right. \tag{9.7.112}$$

where $u^*_k = u^n_k - R\delta_- F^n_k$ and the smoothness parameter is defined as in (9.7.111).

One of the weaknesses of the above scheme is that though the limiter looks in both directions to see how the solution is changing, the scheme does not react much to what it sees. The value of ϕ^n_k is the only thing that changes. It is possible and sometimes advantageous to include different terms in the numerical flux function especially designed to treat information coming from the left and the right. Sweby, ref. [66] suggests the scheme with the following numerical flux function, which we see includes both positive and negative antidiffusive flux terms, designed to handle information that is propagating from the right and the left, respectively.

$$h^n_{k+1/2} = h^E_{k+1/2} - \frac{1}{2}\phi(\theta^+_k)\left\{\left(h^E_{k+1/2} - F^n_{k+1}\right) + \frac{R}{\delta_+ u^n_k}\left(h^E_{k+1/2} - F^n_{k+1}\right)^2\right\} - \frac{1}{2}\phi(\theta^-_{k-1})\left\{\left(h^E_{k+1/2} - F^n_k\right) + \frac{R}{\delta_+ u^n_k}\left(h^E_{k+1/2} - F^n_k\right)^2\right\} \tag{9.7.113}$$

where θ_k^+ and θ_{k-1}^- are given by

$$\theta_k^+ = \frac{\delta_+ u_k^n}{\delta_+ u_{k-1}^n} \frac{\left[\delta_+ u_{k-1}^n + R\left(h_{k-1/2}^E - F_k^n\right)\right]\left(h_{k-1/2}^E - F_k^n\right)}{\left[\delta_+ u_k^n + R\left(h_{k+1/2}^E - F_{k+1}^n\right)\right]\left(h_{k+1/2}^E - F_{k+1}^n\right)} \tag{9.7.114}$$

$$\theta_{k-1}^- = \frac{\delta_+ u_{k-2}^n}{\delta_+ u_{k-1}^n} \frac{\left[\delta_+ u_{k-1}^n + R\left(h_{k-1/2}^E - F_{k-1}^n\right)\right]\left(h_{k-1/2}^E - F_{k-1}^n\right)}{\left[\delta_+ u_{k-2}^n + R\left(h_{k-3/2}^E - F_{k-2}^n\right)\right]\left(h_{k-3/2}^E - F_{k-2}^n\right)}. \tag{9.7.115}$$

We see in HW9.7.28 that under the appropriate assumptions, the difference scheme defined by (9.7.113)–(9.7.115) reduces to difference scheme with numerical flux function (9.7.104) along with θ_k^n defined by (9.7.105) and (9.7.106). It is also easy to see that when the low order scheme is chosen to be the nonlinear upwind scheme ($h_{k-1/2}^n = F_k^n$ or $h_{k-1/2}^n = F_{k-1}^n$), one of the smoothness parameters $\theta^{\pm}$ is zero at every point. In this case, at a given grid point the scheme determines from which direction the information is approaching the point and uses the appropriate corrected flux term in (9.7.113) so that the scheme will reach in the correct direction. In ref. [66], Sweby shows with a calculation similar to that performed in Section 9.7.5.1 that *if the CFL condition* $R|F'| \leq \frac{2}{3}$ *is satisfied, then difference scheme (9.7.113)–(9.7.115) is TVD.*

Hence, we have several schemes with which we can experiment. The schemes are nonlinear, conservative schemes that use more than three points. Difference scheme (9.7.113)–(9.7.115) is TVD. Only through experimentation have we been assured that the schemes may choose the vanishing viscosity solution.

HW 9.7.20 Use the difference scheme associated with numerical flux function (9.7.110) along with smoothness parameter (9.7.111) and the Superbee limiter (9.7.93) to approximate the solutions to the initial–boundary–value problems given in HW9.4.3 and HW9.4.5. Compare and contrast your solutions with those found in HW9.4.3, HW9.4.5 and HW9.7.12.

HW 9.7.21 Repeat the part of problem HW9.7.20 associated with HW9.4.9 using the Godunov scheme (9.7.62), (9.7.74) as the low order scheme. Compare and contrast your solution with that found in HW9.7.12.

HW 9.7.22 Use the difference scheme associated with numerical flux function (9.7.110), smoothness parameter (9.7.111), and the Van Leer limiter (9.7.94) to approximate the solution to the problem given in Example 9.5.3. Use $\Delta x = 0.01$, $\Delta t = 0.001$, produce the solution at $t = 0.5$, and compare and contrast the solution with that given in Figure 9.5.3.

HW 9.7.23 Use the difference scheme associated with numerical flux function (9.7.112) along with smoothness parameter (9.7.111) and the Superbee limiter (9.7.93) to approximate the solutions to the initial–boundary–value problems given in HW9.4.3 and HW9.4.5. Compare and contrast your solutions with those found in HW9.4.3 and HW9.4.5, and HW9.7.12.

HW 9.7.24 Use the difference scheme associated with numerical flux function (9.7.110), smoothness parameter (9.7.111), and the C-O limiter (9.4.15) with $\psi = 1.5$ to approximate the solution of Burgers' equation along with the initial condition

$$v_0(x) = v(x,0) = \begin{cases} 1.0 & \text{if } x \leq 0 \\ -0.25 & \text{if } x > 0 \end{cases} \tag{9.7.116}$$

and boundary conditions $v(-1,t) = 1.0$ and $v(1,t) = -0.25$. Use $\Delta x = 0.01$, $\Delta t = 0.005$ and plot the solution when $t = 0.5$.

HW 9.7.25 Solve the problem given in HW9.7.24 using the difference scheme defined by (9.7.113)–(9.7.115) with the C-O limiter (9.4.15) with $\psi = 1.5$. Compare and contrast your solution with that found in HW9.7.24.

HW 9.7.26 Use the difference scheme defined by numerical flux function (9.7.110), smoothness parameter (9.7.111), and the Superbee limiter (9.7.93) to solve the initial–boundary–value problem given in HW9.4.3 and the initial–boundary–value problem given in HW9.4.5.

HW 9.7.27 Use both of the schemes defined by (9.7.110)–(9.7.111) and (9.7.113)–(9.7.115) to solve HW0.0.2.

HW 9.7.28 (a) Show that when $F(v) = av$, $a > 0$, and $h^E_{k+1/2} = au^n_k$, the difference scheme defined by numerical flux function (9.7.113) along with θ defined by (9.7.114) and (9.7.115) reduces to difference scheme (9.7.85). (b) Show that when $f(v) = av$, $a < 0$, and $h^E_{k+1/2} = au^n_{k+1}$, the numerical flux function (9.7.113) along with θ defined by (9.7.114) and (9.7.115) reduces to the numerical flux function (9.7.102) along with $\theta^n_k = \delta_+ u^n_{k+1}/\delta_+ u^n_k$.

HW 9.7.29 Use the difference scheme associated with numerical flux function (9.7.113), smoothness parameters (9.7.114) and (9.7.115) along with the Superbee limiter (9.7.93) and the Van Leer limiter (9.7.94) to approximate the solution to the problem given in Example 9.5.3. Use $\Delta x = 0.01$, $\Delta t = 0.001$, produce the solution at $t = 0.5$, and compare and contrast the solution with that given in Figure 9.5.3 and the results of HW9.7.22.

9.7.6 Slope-Limiter Methods

In Section 9.7.3 we presented the Godunov scheme, where we started at time $t = t_n = n\Delta t$ with an approximation to the solution, u_k^n. We used these data to define an approximation to the solution, $\bar{u}^n$, which was constant over the cells $(x_{k-1/2}, x_{k+1/2})$ (on $(x_{k-1/2}, x_{k+1/2})$, $\bar{u}^n(x) = u_k^n$). We then solved the initial-value problem consisting of the conservation law with initial condition $v(x, t_n) = \bar{u}(x)$ exactly for a small time increment Δt for $\bar{U} = \bar{U}(x, t)$ and found our approximation to the solution at time $t = t_{n+1} = (n+1)\Delta t$ and $x = x_k = k\Delta t$ by setting u_k^{n+1} equal to the cell average of $\bar{U}$.

The slope-limiter method is very similar to the Godunov scheme. Instead of representing our approximation of the solution at time $t = t_n$ as the piecewise constant function as above, we now use the data at time $t = t_n$, u_k^n, to define a piecewise linear approximation to our solution, $\bar{u}^n$, such that $\bar{u}^n(k\Delta x) = u_k^n$ and the slope of $\bar{u}^n$ on the cell $(x_{k-1/2}, x_{k+1/2})$ is given by σ_k^n, which we get to choose. The approximate solution $\bar{u}^n$ might look like the function given in Figure 9.7.8. Hence, on the cell $(x_{k-1/2}, x_{k+1/2})$, $\bar{u}^n(x) = u_k^n + \sigma_k^n(x - x_k)$.

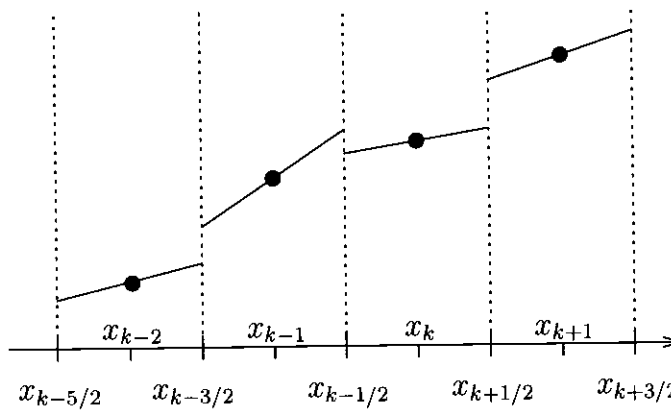

FIGURE 9.7.8. Example of a portion of a piecewise linear approximation to the solution $\bar{u}^n$.

The slope-limiter scheme then consists in solving the initial-value problem given by

$$v_t + F_x = 0, \quad x \in \mathbb{R}, \ t > 0, \tag{9.7.117}$$

$v(x, 0) = \bar{u}^n$ for $\bar{U} = \bar{U}(x, t)$. Let u_k^{n+1} denote the cell average of $\bar{U}$ at $t = \Delta t$ and define the approximate solution $\bar{u}^{n+1}$ on the cell $(x_{k-1/2}, x_{k+1/2})$ as $\bar{u}^{n+1}(x) = u_k^{n+1} + \sigma_k^{n+1}(x - x_k)$. We emphasize that we have not yet discussed how to choose the slopes σ_k^n and σ_k^{n+1}.

9.7.6.1 Slope-Limiter Schemes for the One Way Wave Equation

If we were to consider the linear conservation law, the one way wave equation $v_t + av_x = 0$, because it is very easy to solve the partial differential equation exactly, it is not difficult to see that the above approach will produce the difference scheme

$$u_k^{n+1} = \begin{cases} u_k^n - aR\delta_- u_k^n - \frac{aR}{2}(1-aR)\Delta x \delta_- \sigma_k^n & \text{if } a > 0 \\ u_k^n - aR\delta_+ u_k^n + \frac{aR}{2}(1+aR)\Delta x \delta_+ \sigma_k^n & \text{if } a < 0. \end{cases} \tag{9.7.118}$$

For example, if we consider the case where $a > 0$ (solution propagating to the right) to determine $\bar{U}$ on the control volume $(x_{k-1/2}, x_{k+1/2})$, we must consider the initial condition on both of the intervals $(x_{k-3/2}, x_{k-1/2})$ and $(x_{k-1/2}, x_{k+1/2})$. Hence, we must solve the problem consisting of the conservation law $v_t + av_x = 0$ along with the initial condition

$$\begin{aligned} v(x, t_n) &= v_0(x) \\ &= \begin{cases} u_{k-1}^n + \sigma_{k-1}^n (x - x_{k-1}) & \text{for } x_{k-3/2} \le x \le x_{k-1/2} \\ u_k^n + \sigma_k^n (x - x_k) & \text{for } x_{k-1/2} \le x \le x_{k+1/2}. \end{cases} \end{aligned}$$

Since the solution to this problem at $t = t_{n+1}$ can be written as

$$\begin{aligned} v(x, t_{n+1}) &= v_0(x - a\Delta t) \\ &= \begin{cases} u_{k-1}^n + \sigma_{k-1}^n (x - a\Delta t - x_{k-1}) & \text{for } x_{k-3/2} \le x - a\Delta t \le x_{k-1/2} \\ u_k^n + \sigma_k^n (x - a\Delta t - x_k) & \text{for } x_{k-1/2} \le x - a\Delta t \le x_{k+1/2}, \end{cases} \end{aligned}$$

we can write the solution u_k^{n+1} as

$$\begin{aligned} u_k^{n+1} &= \frac{1}{\Delta x} \int_{x_{k-1/2}}^{x_{k+1/2}} v_0(x - at)\, dx \\ &= \frac{1}{\Delta x} \int_{x_{k-1/2}}^{x_{k-1/2} + a\Delta t} \left[u_{k-1}^n + \sigma_{k-1}^n (x - a\Delta t - x_{k-1}) \right] dx \\ &\quad + \frac{1}{\Delta x} \int_{x_{k-1/2} + a\Delta t}^{x_{k+1/2}} \left[u_k^n + \sigma_k^n (x - a\Delta t - x_k) \right] dx. \end{aligned}$$

Carrying out the indicated integrations, we see that we get the first expression given in (9.7.118). We should be aware that as is usually the case, we must still satisfy the CFL condition $|a|R \le \frac{1}{2}$. This condition can be relaxed to $|a|R \le 1$.

We next note that since we can write difference scheme (9.7.118) in terms of the numerical flux function

$$h_{k+1/2}^n = \begin{cases} au_k^n + \frac{a}{2}(1-aR)\Delta x \sigma_k^n & \text{if } a > 0 \\ au_{k+1}^n - \frac{a}{2}(1+aR)\Delta x \sigma_{k+1}^n & \text{if } a < 0, \end{cases} \tag{9.7.119}$$

difference scheme (9.7.118) is conservative. We also note that $h^n_{k+1/2}$ can be written as

$$h^n_{k+1/2} = \frac{1}{2}a(u^n_k + u^n_{k+1}) - \frac{1}{2}|a|\delta_+ u^n_k + \frac{a}{2}(\operatorname{sign}(a) - aR)\Delta x \sigma^n_{k+\ell}, \tag{9.7.120}$$

where $\ell = 0$ if $a > 0$ and $\ell = 1$ if $a < 0$. If for $a > 0$ we were to choose

$$\sigma^n_k = \frac{\delta_+ u^n_k}{\Delta x}, \tag{9.7.121}$$

it is not difficult to see that we would have the Lax-Wendroff scheme. See HW9.7.30. Hence, it is possible to choose the slopes σ^n_k so that the scheme is both second order accurate and not TVD. To obtain TVD schemes, we must put limits on the slopes that we allow, hence the name **slope-limiter methods**. Since part of the the solution scheme that advances the solution one time step analytically and then takes the average over a cell will not increase the total variation of the solution, we obtain the following result.

Proposition 9.7.23 *If the slopes σ^n_k, $k = -\infty, \ldots, \infty$, are chosen such that the total variation of the piecewise linear approximate solution is less than or equal to the total variation of the piecewise constant approximate solution in the Godunov scheme, then the scheme will be TVD.*

Remark: The hypothesis that the slopes be chosen so as to not increase the total variation of the piecewise approximation over that of the pointwise approximation is obviously very important. Referring to Figure 9.7.8, we see that the variation between points x_{k-2} and x_{k-1} is the same when considered as a piecewise constant function or as a piecewise linear function. The same is true of the variation between points x_k and x_{k+1}. Between points x_{k-1} and x_k, the variation of the data considered as a piecewise linear function is greater than the variation of the data when considered as a piecewise constant function. We will choose our slopes σ^n_k so that the linear pieces are in relation to each other as they are at $x_{k-3/2}$ and $x_{k+1/2}$ and not as they are at $x_{k-1/2}$. For example, the variation between points x_{k-2} and x_{k-1} is not greater than that of the associated piecewise constant function, because

$$\lim_{x \to x_{k-3/2}-} \bar{u}^n(x) \leq \lim_{x \to x_{k-3/2}+} \bar{u}^n(x).$$

Likewise, the variation of $\bar{u}^n$ between x_{k-1} and x_k will be greater than that of the associated piecewise constant function because,

$$\lim_{x \to x_{k-1/2}-} \bar{u}^n(x) > \lim_{x \to x_{k-1/2}+} \bar{u}^n(x).$$

Thus we see that we must choose the slopes carefully. One simple slope limiter is given by the minmod slope limiter,

$$\sigma_k^n = \frac{1}{\Delta x}\,\text{minmod}\{\delta_+ u_k^n, \delta_- u_k^n\}, \tag{9.7.122}$$

where the minmod function is defined by

$$\text{minmod}(a,b) = \begin{cases} a & \text{if } |a| < |b| \text{ and } ab > 0 \\ b & \text{if } |b| < |a| \text{ and } ab > 0 \\ 0 & \text{if } ab \le 0. \end{cases}$$

It is not difficult to see that the above limiter will satisfy the hypotheses of Proposition 9.7.118. Of course, there are other limiters available. It is not difficult to see that σ_k^n can be chosen so that the flux-limiter methods can be considered as special cases of the slope-limiter schemes. See HW9.7.32. It is hoped that this brief introduction to slope-limiter methods for linear equations will make our discussion of slope-limiter methods for nonlinear equations easier to understand.

HW 9.7.30 Show that when σ_k^n is given by (9.7.121) and $a > 0$, difference scheme (9.7.118) reduces to the Lax-Wendroff scheme.

HW 9.7.31 Solve the problem solved in HW9.7.19 using difference scheme (9.7.118) along with the minmod slope limiter (9.7.122).

HW 9.7.32 Show that if we choose the slopes σ_k^n as

$$\sigma_k^n = \phi_k^n \frac{u_{k+1}^n - u_k^n}{\Delta x},$$

then the slope-limiter difference scheme (9.7.118) is the same as the flux-limiter difference scheme associated with numerical flux function (9.7.104).

9.7.6.2 Slope-Limiter Schemes for Nonlinear Conservation Laws

The difference between the case of the slope-limiter scheme for the linear conservation law and the slope-limiter scheme for the nonlinear conservation law is that it is not possible to solve the local problem that consists of conservation law (9.7.117) with initial condition $v(x,t_n) = \bar{u}^n(x) = u_k^n + \sigma_k^n(x - x_k)$. If we return to Section 9.7.3, we note that equation (9.7.59) does not depend on the form of $\bar{u}^n$. In fact, because of the definition of $\bar{u}^n$, equation (9.7.60) will also hold true here. We would like to define the numerical flux functions as we did in equation (9.7.61) as

$$h_{k+1/2}^n = \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} F(\bar{U}(x_{k+1/2},t))\,dt, \tag{9.7.123}$$

where $\bar{U}$ is now the solution to the initial-value problem consisting of conservation law (9.7.117) along with initial condition $v(x,t_n) = \bar{u}^n(x) = u_k^n + \sigma_k^n(x - x_k)$. This numerical flux function would be the natural extension of the Godunov scheme. However, we are unable to compute $h_{k\pm1/2}^n$, since we are unable to compute $\bar{U}$ for a piecewise linear $\bar{u}^n$. The approach that we will describe, given in ref. [15], is to use an approximation of F to compute an approximation of $\bar{U}$. We then use that approximation of $\bar{U}$ to define an approximation of $h_{k+1/2}^n$. Since the description of the algorithm becomes more difficult near extrema of the solution and near sonic points, for our developmental discussion *we assume that the solution has no extrema and F' is not zero in the region we are considering.* In addition, we assume that the CFL condition $R|F'| \leq \frac{1}{2}$ is satisfied. This condition can be relaxed to $R|F'| \leq 1$. We give the complete algorithm at the end of the section. For complete details see ref. [15].

Begin by again defining σ_k^n as we did before as

$$\sigma_k^n = \frac{1}{\Delta x}\,\text{minmod}\{\delta_+ u_k^n, \delta_- u_k^n\}. \tag{9.7.124}$$

This choice of σ_k^n is such that the variation of $\bar{u}^n$ is the same as the variation of u_k^n. (Since $\bar{u}^n$ is a defined on $\mathbb{R}$, the variation of $\bar{u}^n$ is defined as the integral of the absolute value of $\bar{u}^n$ from $-\infty$ to ∞.) Define $U_k^{\pm}$ by

$$U_k^{\pm} = u_k^n \pm \frac{1}{2}\Delta x \sigma_k^n. \tag{9.7.125}$$

We note that in order that the total variation of the piecewise linear ap-

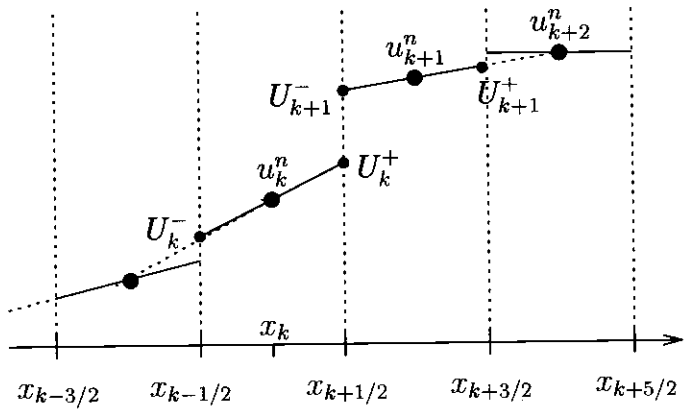

FIGURE 9.7.9. An example of a portion of the approximate solution $\bar{u}^n$. The dashed lines show how the slopes σ_k^n are determined. The points $U_j^{\pm}$ are plotted for $j = k$ and $j = k + 1$.

proximation of the solution be less than or equal to the total variation of

the pointwise approximation of the solution, we must have $U_k^- \le U^+ \le U_{k+1}^-$ or $U_k^+ \le U^+ \le U_{k+1}^+$. In Figure 9.7.9 for several given values of u_k^n we plot $\bar{u}^n$ and $U_j^\pm$, $j = k, k+1$. We let $G = G(u)$ be the piecewise linear function that interpolates F at the four points $U_k^\pm$, $U_{k+1}^\pm$ and set

$$G_k' = \begin{cases} [F(U_k^+) - F(U_k^-)]/(U_k^+ - U_k^-) & \text{if } \sigma_k^n \neq 0 \\ F'(u_k^n) & \text{if } \sigma_k^n = 0. \end{cases} \tag{9.7.126}$$

We note that G_k' is the slope of G between U_k^- and U_k^+, hence, the function G is given by

$$G(u) = \begin{cases} F(U_k^+) + (u - U_k^+)G_k' & \text{for } u \in [U_k^-, U_k^+] \\ F(U_{k+1}^-) + (u - U_{k+1}^-)G_{k+1}' & \text{for } u \in [U_{k+1}^-, U_{k+1}^+]. \end{cases} \tag{9.7.127}$$

An example of an approximate flux function G is given in Figure 9.7.10. We note that we linearly interpolate the values $U_j^\pm$ for $j = k$ and $j = k+1$. We see that these linear portions of the approximations intersect at a value $U = U_k^*$. This value is not used in our treatment (it is used in the proof that the resulting scheme is TVD, which we do not do) but is included for completeness.

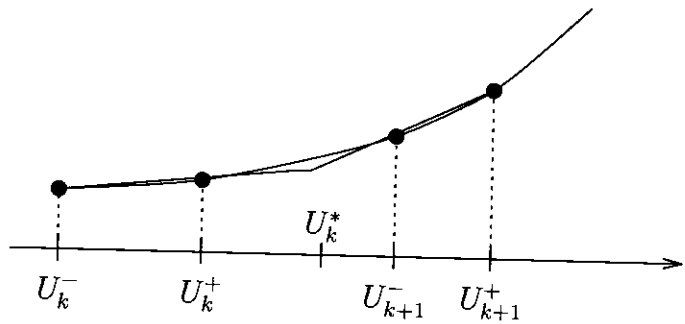

FIGURE 9.7.10. An illustration of the approximate flux function G that is obtained by linearly interpolating F at the points $U_j^\pm$, $j = k$ and $j = k+1$.

We replace the flux function F in equation (9.7.123) by the approximate flux function G. Hence, we must solve the initial-value problem consisting of conservation law

$$v_t + G_x = 0 \tag{9.7.128}$$

along with initial condition

$$v(x, t_n) = \bar{u}^n. \tag{9.7.129}$$

If we write equation (9.7.128) in nonconservative form, we get $v_t + G'v_x = 0$. We must realize that G is not differentiable everywhere, so we must be careful. More importantly, however, is the fact that since G is a piecewise linear function, on the various pieces, G' will be a constant, i.e., we can solve the equation. We need the solution to initial-value problem (9.7.128)–(9.7.129) at $x = x_{k+1/2}$. We begin by considering the case where $F' > 0$. When $F' > 0$, we want G' to be greater than zero also. This will involve choosing Δx sufficiently small so that G is a sufficiently good approximation of F. If $F' > 0$, the solution will propagate to the right. In this case, the solution at $x = x_{k+1/2}$ is determined by $\bar{u}$, F and G' on the interval $(x_{k-1/2}, x_{k+1/2})$. The relevant initial condition will be $v_0(x) = u_k^n + \sigma_k^n(x - x_k)$, the solution will be $v(x,t) = v_0\big(x - G_k'(t - t_n)\big)$, and $\bar{U}(x_{k+1/2}, t)$ is given by

$$\begin{aligned}\bar{U}(x_{k+1/2}, t) &= v_0\big(x_{k+1/2} - G_k'(t - t_n)\big) \\ &= u_k^n + \sigma_k^n\big(x_{k+1/2} - G_k'(t - t_n)\big) \\ &= U_k^+ - G_k'\sigma_k^n(t - t_n). \end{aligned} \tag{9.7.130}$$

If $F' < 0$, the result will be similar but will depend on the solution in the interval $(x_{k+1/2}, x_{k+3/2})$ and G_{k+1}'. Therefore, the solution $\bar{U} = \bar{U}(x,t)$ at $x = x_{k+1/2}$ to initial-value problem (9.7.128)–(9.7.129) is given by

$$\bar{U}(x_{k+1/2}, t) = \begin{cases} U_k^+ - (t - t_n)\sigma_k^n G_k' & \text{if } F' > 0 \\ U_{k+1}^- - (t - t_n)\sigma_{k+1}^n G_{k+1}' & \text{if } F' < 0. \end{cases} \tag{9.7.131}$$

The numerical flux function $\tilde{h}_{k+1/2}^n$ that will approximate numerical flux function (9.7.123) is given by

$$\tilde{h}_{k+1/2}^n = \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} G(\bar{U}(x_{k+1/2}, t))\, dt. \tag{9.7.132}$$

Using G as given in (9.7.127), we see that $G(\bar{U}(x_{k+1/2}, t))$ is given by (again for the case when $F' > 0$)

$$\begin{aligned} G(\bar{U}(x_{k+1/2}, t)) &= F(U_k^+) + (u - U_k^+)G_k' \\ &= F(U_k^+) + \big[(U_k^+ - (t - t_n)\sigma_k^n G_k') = U_k^+\big] \\ &= F(U_k^+) - (t - t_n)\sigma_k^n (G_k')^2. \end{aligned} \tag{9.7.133}$$

Of course, the method and result for the case when $F' < 0$ are very similar. Then, using (9.7.133) in (9.7.132) when $F' > 0$ and the analogous result when $F' < 0$ gives

$$\tilde{h}_{k+1/2}^n = \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} G(\bar{U}(x_{k+1/2}, t))\, dt \tag{9.7.134}$$

$$= \begin{cases} F(U_k^+) - \frac{1}{2}\Delta t \sigma_k^n (G_k')^2 & \text{if } F' > 0 \\ F(U_{k+1}^-) - \frac{1}{2}\Delta t \sigma_{k+1}^n (G_{k+1}')^2 & \text{if } F' < 0. \end{cases} \tag{9.7.135}$$

If there is a sonic point in the region, i.e., if $G'_k G'_{k+1} \le 0$, then we define

$$G'_{k+1/2} = \begin{cases} \left[F(U^-_{k+1}) - F(U^+_k)\right]/[U^-_{k+1} - U^+_k] & \text{if } U^-_{k+1} \neq U^+_k \\ F'(U^+_k) & \text{if } U^-_{k+1} = U^+_k, \end{cases} \tag{9.7.136}$$

and (1) if $G'_k > 0$, $G'_{k+1/2}(U^-_{k+1} - U^+_k) = 0$ and $G'_{k+1} < 0$, set

$$\tilde{h}^n_{k+1/2} = \begin{cases} F(U^+_k) - \frac{1}{2}\Delta t \sigma^n_k (G'_k)^2 & \text{if } \sigma_k (G'_k)^2 \ge \sigma^n_{k+1}(G'_{k+1})^2 \\ F(U^-_{k+1}) - \frac{1}{2}\Delta t \sigma^n_{k+1}(G'_{k+1})^2 & \text{otherwise} \end{cases} \tag{9.7.137}$$

(2) otherwise, set

$$\tilde{h}^n_{k+1/2} = \begin{cases} F(U^+_k) - \frac{1}{2}\Delta t \sigma^n_k (G'_k)^2 & \text{if } G'_k \ge 0 \text{ and } G'_{k+1/2} \ge 0 \\ F(U^-_{k+1}) - \frac{1}{2}\Delta t \sigma^n_{k+1}(G'_{k+1})^2 & \text{if } G'_{k+1} \le 0 \text{ and } G'_{k+1/2} \le 0 \\ F(v_0) & \text{if } G'_k < 0 \text{ and } G'_{k+1} > 0, \end{cases} \tag{9.7.138}$$

where $v_0 = \min\left\{ \max\left\{U^+_k, u_s\right\}, U^-_{k+1}\right\}$ and u_s is the sonic point. For more details involving this development, see ref. [15].

The resulting algorithm is then if $G'_k G'_{k+1} > 0$, define $\tilde{h}$ by equation (9.7.135). Otherwise, use expressions (9.7.136), (9.7.137) and (9.7.138). The following proposition gives some of the very important properties of the slope-limiter scheme developed above. For proofs, see ref. [15].

Proposition 9.7.24 *If $R|F'| \le \frac{1}{2}$, the difference scheme defined by the numerical flux function given in (9.7.134), (9.7.136)–(9.7.138) is second order accurate on smooth sections of the solution that are not local extrema.*

One part of the above proposition that we have not mentioned is the fact that the slope-limiter scheme is second order accurate. Again we consider the case away from a local extremum of the solution (so that $\sigma^n_k \neq 0$) and discuss the proof for $F' > 0$. The case where $F' < 0$ is clearly similar. We see that numerical flux function (9.7.135) is similar to the numerical flux function for the Lax-Wendroff scheme (9.5.10). The numerical flux function for the Lax-Wendroff scheme is given by

$$h^n_{LW_{k+1/2}} = \frac{1}{2}(F^n_k + F^n_{k+1}) - \frac{R}{2}(a^n_{k+1/2})^2 \delta_+ u^n_k.$$

If we expand the expression for $\tilde{h}^n_{k+1/2} - h^n_{LW_{k+1/2}}$ about u^n_k, we get

$$\tilde{h}^n_{k+1/2} - h^n_{LW_{k+1/2}} = \frac{1}{2}F'(u^n_k)\left[RF'(u^n_k) - 1\right]\left[\delta_+ u^n_k - \Delta x \sigma^n_k\right] + \mathcal{O}(\Delta x)^2.$$

We then see that either $\sigma^n_k = \delta_+ u^n_k/\Delta x$, in which case the first term is zero, or $\sigma^n_k = \delta_- u^n_k/\Delta x$, in which case the first term is of order Δx^2. Hence, in either case we have that $\tilde{h}^n_{k+1/2} - h^n_{LW_{k+1/2}} = \mathcal{O}(\Delta x^2)$, and by Proposition 9.7.15, the slope-limiter scheme is second order accurate.

HW 9.7.33 Use the slope-limiter difference scheme defined by (9.7.135)–(9.7.138) to approximate the solutions to the initial–boundary–value problems given in HW9.4.3 and HW9.4.5. Compare and contrast your solutions with those found in HW9.4.3, HW9.4.5 and HW9.7.20.

9.7.7 Modified Flux Method

Another approach for developing high resolution schemes is the modified flux method due to Harten, ref. [23]. The method is similar to both the flux-limiter method and the slope-limiter method in that the flux is adjusted. The difference between the modified flux method and the flux-limiter method is that in the modified flux method, as in the slope-limiter method, the flux function of the conservation law, not the numerical flux function, is modified. When we obtain the final numerical scheme, we may find that the difference is a matter of semantics on which flux is modified. However, the modified flux method provides us with a different approach and, hence, a different view, to obtaining high resolution schemes.

We begin by considering conservation law (9.7.1) along with a conservative, three-point incremental TVD scheme with numerical flux function h^L. One of the monotone or E schemes would be a good candidate for this given scheme. Specifically, the upwind scheme (9.4.15) with $Q^L(x) = |x|$ is a scheme that Harten, [23], uses in his numerical experiments. See Remark, page 235. Let Q^L denote the coefficient of numerical viscosity of the scheme and recall that h^L and Q^L are related by (formula (9.7.13))

$$h^L_{k+1/2} = \frac{1}{2}\left[F^n_k + F^n_{k+1}\right] - \frac{1}{2R}Q^L_{k+1/2}\delta_+ u^n_k. \tag{9.7.139}$$

Also recall that if the scheme we have chosen is incremental TVD, then the numerical viscosity flux function satisfies

$$R|a^n_{k+1/2}| \le Q^L_{k+1/2} \le 1 \tag{9.7.140}$$

(Proposition 9.7.18). We note that we have omitted the usual reference to time in our notation, i.e., the superscript n. (The superscript L refers to the low order scheme.) We have omitted the n because we are using the superscript to label the various numerical flux functions, numerical viscosity coefficients, etc. that we will be using in this development. The omitted n will not, we hope, cause us any problems.

We know from Proposition 9.7.19 that the above-described difference scheme will be at most first order accurate. Surely, we want to choose a scheme that is at least first order accurate. By Proposition 9.7.12 we know that the leading term in the truncation error is given by

$$\Delta t\left\{\left[q(u)u_x\right]_x\right\}_{u=u^n_k}.$$

Though we know that our difference scheme approximates the solution to the conservation law to first order, we see that the difference scheme will approximate the solutions to the modified partial differential equation

$$u_t + F(u)_x = \Delta t\big[q(u)u_x\big]_x \tag{9.7.141}$$

to the second order. The approach taken to obtain the modified flux high resolution scheme is to modify the flux of the conservation law in such a way to eliminate the term on the right side of equation (9.7.141). Hence, we will apply our original first order accurate scheme to approximate the solution to the conservation law

$$v_t + F_x^M = 0 \tag{9.7.142}$$

where F^M has been chosen so that the first order approximate solution to conservation law (9.7.142) will be a second order accurate approximation to the solution of conservation law (9.7.1).

We want our scheme to be of second order whenever possible (which we know is at best of second order away from the local extrema). To be second order, the scheme must be at least "Lax-Wendroff-like" on smooth sections of the solution. Denote the numerical flux function and numerical viscosity coefficient associated with the Lax-Wendroff scheme (9.5.10) by h^{LW} and Q^{LW} and recall that h^{LW} and Q^{LW} are given by

$$h_{k+1/2}^{LW} = \frac{1}{2}\big[F_k^n + F_{k+1}^n\big] - \frac{R}{2}\big(a_{k+1/2}^n\big)^2\delta_+ u_k^n \tag{9.7.143}$$

and

$$Q_{k+1/2}^{LW} = R^2\big(a_{k+1/2}^n\big)^2. \tag{9.7.144}$$

We apply the first order scheme associated with h^L and Q^L with the flux F replaced by F^M where

$$F_k^M = F_k^n + (1/R)g_k \text{ and } g_k = g(u_{k-1}, u_k, u_{k+1})$$

and g is yet to be determined. Let $a_{k+1/2}^M$ denote the modified local speed of propagation given by equation (9.4.17) with F replaced by F^M. The modified numerical flux function is then given by

$$h_{k+1/2}^M = \frac{1}{2}\big[F_{k+1}^M + F_k^M\big] - \frac{1}{2R}Q_{k+1/2}^M\delta_+ u_k^n, \tag{9.7.145}$$

where $Q_{k+1/2}^M$ is Q^L with $a_{k+1/2}^n$ replaced by $a_{k+1/2}^M$, or

$$\begin{aligned} h_{k+1/2}^M =& \frac{1}{2}\big[F_{k+1}^n + F_k^n\big] \\ &+ \frac{1}{2R}\Big[g_{k+1} + g_k - Q^L\big(Ra_{k+1/2}^n + \delta_+ g_k/\delta_+ u_k^n\big)\delta_+ u_k^n\Big]. \end{aligned} \tag{9.7.146}$$

We are then able to prove the following result.

Proposition 9.7.25 *Suppose Q^L is Lipschitz continuous and g_k satisfies*

$$g_k + g_{k+1} = \left[Q^L\left(Ra^n_{k+1/2}\right) - R^2\left(a^n_{k+1/2}\right)^2\right]\delta_+ u^n_k + \mathcal{O}(\Delta x^2) \tag{9.7.147}$$

$$\delta_+ g_k = \mathcal{O}(\Delta x^2). \tag{9.7.148}$$

Then the difference scheme associated with the modified numerical flux function $h^M_{k+1/2}$ is second order accurate on smooth sections of the solution.

Proof: We will prove the second order accuracy by using Proposition 9.7.15. We note that h^M and h^L are related by

$$\begin{aligned} h^M_{k+1/2} =& h^L_{k+1/2} + \frac{1}{2R}\Big\{g_{k+1} + g_k \\ &+ \left[Q^L\left(Ra^n_{k+1/2}\right) - Q^L\left(Ra^n_{k+1/2} + \delta_+ g_k/\delta_+ u^n_k\right)\right]\delta_+ u^n_k\Big\}, \end{aligned} \tag{9.7.149}$$

and h^L and h^{LW} are related by

$$h^L_{k+1/2} = h^{LW}_{k+1/2} - \frac{1}{2R}\left[Q^L\left(Ra^n_{k+1/2}\right) - R^2\left(a^n_{k+1/2}\right)^2\right]\delta_+ u^n_k. \tag{9.7.150}$$

Substituting $h^L_{k+1/2}$ given in equation (9.7.150) into equation (9.7.149) and solving for $h^M_{k+1/2} - h^{LW}_{k+1/2}$ gives us

$$\begin{aligned} & h^M_{k+1/2} - h^{LW}_{k+1/2} \\ =& \frac{1}{2R}\Big\{g_{k+1} + g_k + \left[Q^L\left(Ra^n_{k+1/2}\right) - Q^L\left(Ra^n_{k+1/2} + \delta_+ g_k/\delta_+ u^n_k\right)\right]\delta_+ u^n_k \\ & - \left[Q^L\left(Ra^n_{k+1/2}\right) - R^2\left(a^n_{k+1/2}\right)^2\right]\delta_+ u^n_k\Big\}. \end{aligned} \tag{9.7.151}$$

We can rewrite equation (9.7.151) as

$$\begin{aligned} & h^M_{k+1/2} - h^{LW}_{k+1/2} \\ =& \frac{1}{2R}\Big\{g_{k+1} + g_k - \left[Q^L\left(Ra^n_{k+1/2}\right) - R^2\left(a^n_{k+1/2}\right)^2\right]\delta_+ u^n_k\Big\} \\ & + \frac{1}{2R}\left[Q^L\left(Ra^n_{k+1/2}\right) - Q^L\left(Ra^n_{k+1/2} + \delta_+ g_k/\delta_+ u^n_k\right)\right]\delta_+ u^n_k. \end{aligned} \tag{9.7.152}$$

By assumption (9.7.147), the first term on the right hand side of equation (9.7.151) is $\mathcal{O}(\Delta x^2)$. Since Q^L is Lipschitz continuous, we can write

$$\begin{aligned} & \left|\frac{1}{2R}\left[Q^L\left(Ra^n_{k+1/2}\right) - Q^L\left(Ra^n_{k+1/2} + \delta_+ g_k/\delta_+ u^n_k\right)\right]\delta_+ u^n_k\right| \\ & \quad \le \frac{K}{2R}|\delta_+ g_k/\delta_+ u^n_k||\delta_+ u^n_k| \\ & \quad = \frac{K}{2R}|\delta_+ g_k| \\ & \quad = \mathcal{O}(\Delta x^2) \text{ (by assumption (9.7.148))}. \end{aligned}$$

Hence, the second term on the right hand side of equation (9.7.151) is also $\mathcal{O}(\Delta x^2)$, and Proposition 9.7.19 implies that the difference scheme associated with the modified numerical flux function h^M is second order.

We next define the function g so that the hypotheses of Proposition 9.7.25 are satisfied. We begin by defining

$$\tilde{g}_{k+1/2} = \frac{1}{2}\Big[Q^L\big(Ra^n_{k+1/2}\big) - R^2\big(a^n_{k+1/2}\big)^2\Big]\delta_+ u^n_k. \tag{9.7.153}$$

We then define $g_k = g(u_{k-1}, u_k, u_{k+1})$ as

$$g_k = \begin{cases} \operatorname{sign}\{\tilde{g}_{k+1/2}\}\min\{|\tilde{g}_{k+1/2}|, |\tilde{g}_{k-1/2}|\} & \text{when } \tilde{g}_{k+1/2}\tilde{g}_{k-1/2} \geq 0 \\ 0 & \text{when } \tilde{g}_{k+1/2}\tilde{g}_{k-1/2} < 0. \end{cases} \tag{9.7.154}$$

We then obtain the following result.

Proposition 9.7.26 *Let g_k be defined by equation (9.7.154). Then conditions (9.7.147) and (9.7.148) are satisfied.*

Proof: Begin by assuming that $\tilde{g}_{k+1/2}\tilde{g}_{k-1/2} \geq 0$. Using definition (9.7.154) along with the fact that

$$\min(a,b) = \frac{1}{2}[(a+b) - |a-b|],$$

we see that

$$\begin{aligned} g_k =& \frac{\operatorname{sign}\{\tilde{g}_{k+1/2}\}}{2}\Big[\big(|\tilde{g}_{k+1/2}| + |\tilde{g}_{k-1/2}|\big) - \big||\tilde{g}_{k+1/2}| - |\tilde{g}_{k-1/2}|\big|\Big] \\ =& \frac{1}{2}\Big[\tilde{g}_{k+1/2} + \tilde{g}_{k-1/2} - \operatorname{sign}\{\tilde{g}_{k+1/2}\}\big|\tilde{g}_{k+1/2} - \tilde{g}_{k-1/2}\big|\Big] \\ =& \tilde{g}_{k+1/2} \\ &+ \frac{1}{2}\Big[-\big(\tilde{g}_{k+1/2} - \tilde{g}_{k-1/2}\big) - \operatorname{sign}\{\tilde{g}_{k+1/2}\}\big|\tilde{g}_{k+1/2} - \tilde{g}_{k-1/2}\big|\Big] \end{aligned} \tag{9.7.155}$$

$$\begin{aligned} =& \tilde{g}_{k-1/2} \\ &+ \frac{1}{2}\Big[\big(\tilde{g}_{k+1/2} - \tilde{g}_{k-1/2}\big) - \operatorname{sign}\{\tilde{g}_{k+1/2}\}\big|\tilde{g}_{k+1/2} - \tilde{g}_{k-1/2}\big|\Big]. \end{aligned} \tag{9.7.156}$$

From the definition of $\tilde{g}$, (9.7.153), we see that

$$\begin{aligned} &\tilde{g}_{k+1/2} - \tilde{g}_{k-1/2} \\ &\quad = \frac{1}{2}\Big[Q^L\big(Ra^n_{k+1/2}\big)\delta_+ u^n_k - Q^L\big(Ra^n_{k-1/2}\big)\delta_+ u^n_{k-1} \\ &\qquad - \big[R^2\big(a^n_{k+1/2}\big)^2\delta_+ u^n_k - R^2\big(a^n_{k-1/2}\big)^2\delta_+ u^n_{k-1}\big]\Big]. \end{aligned} \tag{9.7.157}$$

Since $\delta_+ u_k^n = (u_x)_k^n \Delta x + \mathcal{O}(\Delta x^2)$ and $\delta_+ u_{k-1}^n = (u_x)_k^n \Delta x + \mathcal{O}(\Delta x^2)$, we can rewrite equation (9.7.157) as

$$\begin{aligned}\tilde{g}_{k+1/2} - \tilde{g}_{k-1/2} \\ = \frac{1}{2}\Big[Q^L\big(Ra_{k+1/2}^n\big) - Q^L\big(Ra_{k-1/2}^n\big) \\ - R^2\big[\big(a_{k+1/2}^n\big)^2 - \big(a_{k-1/2}^n\big)^2\big]\Big](u_x)_k^n \Delta x + \mathcal{O}(\Delta x^2). \quad (9.7.158)\end{aligned}$$

Then, since Q^L is Lipschitz continuous and $a_{k+1/2}^n - a_{k-1/2}^n = \mathcal{O}(\Delta x)$, we see that

$$\tilde{g}_{k+1/2} - \tilde{g}_{k-1/2} = \mathcal{O}(\Delta x^2). \quad (9.7.159)$$

The above equation along with equation (9.7.155) implies that

$$g_k = \tilde{g}_{k+1/2} + \mathcal{O}(\Delta x^2), \quad (9.7.160)$$

and along with equation (9.7.156) equation (9.7.159) it implies that

$$g_k = \tilde{g}_{k-1/2} + \mathcal{O}(\Delta x^2). \quad (9.7.161)$$

We next show that when $\tilde{g}_{k+1/2}\tilde{g}_{k-1/2} < 0$, then equations (9.7.160) and (9.7.161) are still satisfied. If $\tilde{g}_{k+1/2}\tilde{g}_{k-1/2} < 0$, then $g_k = 0$. Since Q^L satisfies $R|a_{k+1/2}^n| \le Q^L \le 1$, then $Q^L\big(Ra_{k+1/2}^n\big) - R^2\big(a_{k+1/2}^n\big)^2 \ge 0$. Thus, $\tilde{g}_{k+1/2}\tilde{g}_{k-1/2} < 0$ if and only if $\delta_+ u_k^n \delta_+ u_{k-1}^n < 0$. If $\delta_+ u_k^n \delta_+ u_{k-1}^n < 0$, we can expand about the point at which $u_x = 0$ and see that $\delta_+ u_k^n = \mathcal{O}(\Delta x^2)$ and $\delta_+ u_{k-1}^n = \mathcal{O}(\Delta x^2)$ and both (9.7.160) and (9.7.161) are satisfied.

In both cases, g and $\tilde{g}$ satisfy equations (9.7.160) and (9.7.161). Use (9.7.161) to write

$$g_{k+1} = \tilde{g}_{k+1/2} + \mathcal{O}(\Delta x^2). \quad (9.7.162)$$

Then, using (9.7.160) and (9.7.162), it is easy to see that

$$\delta_+ g_k = g_{k+1} - g_k = \mathcal{O}(\Delta x^2),$$

so condition (9.7.148) is satisfied. Also, using (9.7.160) and (9.7.162), we see that

$$g_k + g_{k+1} = 2\tilde{g}_{k+1/2} + \mathcal{O}(\Delta x^2),$$

which shows that condition (9.7.147) is satisfied.

Remark 1: We must be careful of how we interpret the results of Propositions 9.7.25 and 9.7.26. With g_k now defined, assuming that we chose a particular first order scheme, we now have a well defined difference scheme. It might seem that Propositions 9.7.25 and 9.7.26 imply that the scheme is second order accurate. However, we know that this cannot be the case. We

know that the scheme can be at best second order accurate away from the extrema of the solutions that are not sonic points. When we used Proposition 9.7.39 to prove that the scheme was second order, one of the hypotheses that we did not emphasize was the smoothness of the leading term in the $\mathcal{O}(\Delta x^2)$ term. The scheme developed above will be second order accurate whenever this leading term is smooth. The leading term will fail to be smooth when $\text{sign}\{\tilde{g}_{k+1/2}\}$ is discontinuous, i.e., when $\text{sign}\{\tilde{g}_{k+1/2}\}$ switches signs. This will occur at the extrema of the solution. We then obtain the following result.

Proposition 9.7.27 *Difference scheme*

$$u_k^{n+1} = u_k^n - R\delta_- h_{k+1/2}^M$$

with $h_{k+1/2}^M$ defined in (9.7.146) and g defined by (9.7.153)–(9.7.154) is second order accurate away from the extrema of the solutions.

Remark 2: We see that $\tilde{g}_{k+1/2}\tilde{g}_{k-1/2} < 0$ when $\delta_+ u_k^n \delta_+ u_{k-1}^n < 0$. In this case, $g_k = 0$. Hence we see that near an extremum of the solution, we add no correction.

Throughout this section, we have always tried to develop schemes that are TVD schemes. Before we show that the above difference scheme defined above is TVD, we must prove the following lemma.

Lemma 9.7.28 *Let g_k be defined as in (9.7.154). Then*

$$|\delta_+ g_k/\delta_+ u_k^n| \leq \frac{1}{2}\left|Q^L(Ra_{k+1/2}^n) - (a_{k+1/2}^n)^2\right|. \tag{9.7.163}$$

Proof: We note that if $g_k > 0$, then $\tilde{g}_{k+1/2} > 0$. If $\tilde{g}_{k+3/2} > 0$, then $g_{k+1} > 0$. If $\tilde{g}_{k+3/2} \leq 0$, then by the definition of g_{k+1}, (9.7.154), we see that $g_{k+1} = 0$. A similar argument holds if we begin with the assumption that $g_k < 0$ and we see that g_k and g_{k+1} cannot be of different signs. Hence,

$$\begin{aligned} |g_{k+1} - g_k| &\leq \max\{|g_k|, |g_{k+1}|\} \\ &\leq \max\left\{\min\{|\tilde{g}_{k-1/2}|, |\tilde{g}_{k+1/2}|\}, \min\{|\tilde{g}_{k+1/2}|, |\tilde{g}_{k+3/2}|\}\right\} \\ &\leq |\tilde{g}_{k+1/2}|. \end{aligned}$$

Therefore,

$$|\delta_+ g_k/\delta_+ u_k^n| \leq |\tilde{g}_{k+1/2}|/|\delta_+ u_k^n| \leq \frac{1}{2}\left|Q^L(Ra_{k+1/2}^n) - (a_{k+1/2}^n)^2\right|.$$

We now prove that the scheme developed above is a TVD scheme.

Proposition 9.7.29 *Suppose that Q^L satisfies $|Ra_{k+1/2}^n| \leq Q_{k+1/2}^L \leq 1$ for all k and g_k is defined by equation (9.7.154). Then the difference scheme defined by numerical flux function (9.7.146) is TVD.*

Proof: We know that since the low order scheme is incrementally TVD, whenever $|\nu| \le 1$, then $|\nu| \le Q^L(\nu) \le 1$. Also, Q^M is defined as

$$Q^M = Q^L(a^M_{k+1/2}).$$

Then, since

$$\begin{aligned}
|Ra^M_{k+1/2}| &= |Ra^n_{k+1/2} + \delta_+ g_k/\delta_+ u^n_k| \\
&\le R|a^n_{k+1/2}| + |\delta_+ g_k/\delta_+ u^n_k| \\
&\le R|a^n_{k+1/2}| + \frac{1}{2}\left|Q^L - R^2\left(a^n_{k+1/2}\right)^2\right| \\
&= R|a^n_{k+1/2}| + \frac{1}{2}\left[Q^L - R^2\left(a^n_{k+1/2}\right)^2\right] \\
&\le R|a^n_{k+1/2}| + \frac{1}{2}\left[1 - R^2\left(a^n_{k+1/2}\right)^2\right] \\
&= 1 - \frac{1}{2}\left(R|a^n_{k+1/2}| - 1\right)^2 \le 1,
\end{aligned}$$

we get $R|a^M_{k+1/2}| \le Q^M_{k+1/2} \le 1$ for all k.

Remark: As we mentioned earlier, the upwind scheme is often used as the low order scheme in the modified flux scheme. By now, we know enough to be very suspicious of using the upwind scheme, and in HW9.7.34 we see that our suspicions are well founded. In ref. [23] where he introduced the modified flux scheme, Harten discussed how the problem occurs as a result of the numerical viscosity function vanishing at zero. He then suggests using the slight variations of the upwind scheme where Q^L is defined as

$$Q^L(x) = \begin{cases} \left(\frac{x^2}{4\epsilon}\right) + \epsilon & \text{when } |x| < 2\epsilon \\ |x| & \text{when } |x| \ge 2\epsilon \end{cases} \tag{9.7.164}$$

with $\epsilon = 0.1$. We see in HW9.7.34 what can happen if the upwind scheme is used, how the adjusted upwind scheme solves this problem, and how well several other monotone schemes work as the low order scheme.

HW 9.7.34 Use the modified flux scheme defined by the numerical flux function $h^M_{k+1/2}$ given in (9.7.146) with g defined as in (9.7.153)–(9.7.154) to approximate the solutions to the initial–boundary–value problems given in HW9.4.3 and HW9.4.5. Compare and contrast your solutions with those found in HW9.4.3, HW9.4.5, HW9.7.20 and HW9.7.33. Use the upwind scheme (9.4.16), the adjusted upwind scheme with Q^L given in (9.7.164) above, the Godunov scheme (9.7.62), (9.7.74), and the Lax-Friedrichs scheme (9.4.3) as the low order scheme.

HW 9.7.35 (a) Show that $\delta_+ u^n_k = 0$ implies that $g_k = g_{k+1} = 0$.
(b) Show that the difference scheme defined by numerical flux function $h^M_{k+1/2}$ is consistent with the conservation law (9.7.1).

9.8 Difference Schemes for K-System Conservation Laws

We have seen that we are able to numerically resolve discontinuities in the solutions to scalar conservation laws without introducing dispersive wiggles and without overly smearing the solution. It is time to return to K-system conservation laws. We studied some of the most common schemes for linear K-system conservation laws in Chapter 6. We developed one sided schemes that because of stability considerations were not very good. We also developed a Lax-Wendroff scheme for systems of hyperbolic equations. Other than the difficulties caused by the facts that we were dealing with K equations simultaneously, that the signs of the eigenvalues could be different, and that boundary conditions for systems were considerably more complicated than boundary conditions for scalar equations, the results for linear systems of hyperbolic equations are not that different from those for the one way wave equation. It is reasonably clear how we should proceed, in that we want to obtain the success we had for scalar conservation laws by generalizing some of the successful schemes so that they will apply equally successfully to numerically approximating solutions to K-system conservation laws. As is usually the case, treatment of the K-system conservation laws is considerably more difficult than the scalar conservation law counterpart. We will develop schemes for K-system conservation laws by an assortment of methods. We will often use the linear K-system conservation law for motivation and/or development of the schemes. We include some proofs of properties of some of the schemes. At times, we will just introduce the natural generalization of a scalar scheme and proceed with care and experimentation.

9.9 Godunov Schemes

9.9.1 Godunov Schemes for Linear K-System Conservation Laws

We begin by considering a linear hyperbolic system of the form

$$\mathbf{v}_t + A\mathbf{v}_x = \boldsymbol{\theta}. \tag{9.9.1}$$

We emphasize again that when we considered hyperbolic systems in Chapter 6, we wrote the equation as $\mathbf{v}_t = A\mathbf{v}_x$. In Section 9.3.1 we set tradition aside and wrote our system as in (9.9.1). We will see that this is the most convenient for our applications.

As we did in Section 6.2.1, we denote the eigenvalues of A by ν_k, $k = 1, \ldots, K$, and assume that we have K_+ positive eigenvalues, $\nu_1, \ldots, \nu_{K_+}$, and $K_- = K - K_+$ negative eigenvalues, $\nu_{K_+ + 1}, \ldots, \nu_K$. Let D denote the

diagonal matrix with the eigenvalues on the diagonal, let S be the matrix that diagonalizes A ($D = SAS^{-1}$), and set $\mathbf{V} = S\mathbf{v}$. We then multiply equation (9.9.1) through by S to uncouple system (9.9.1) into

$$V_{j_t} + \nu_j V_{j_x} = 0, \quad j = 1, \ldots, K. \tag{9.9.2}$$

If we then apply the scalar Godunov scheme to equation (9.9.2), we get

$$U_{j_k}^{n+1} = U_{j_k}^n - \nu_j R\left[h_{j_{k+1/2}}^n - h_{j_{k-1/2}}^n\right], \tag{9.9.3}$$

where $h_{j_{k-1/2}}^n = \nu_j \bar{U}(x_{k-1/2}, t)$ and $\bar{U}$ is the solution the the Riemann problem

$$v_t + av_x = 0, \quad x \in \mathbb{R}, \ t > t_n$$
$$v(x, t_n) = \begin{cases} U_{k-1}^n & \text{when } x \le x_{k-1/2} \\ U_k^n & \text{when } x > x_{k-1/2} \end{cases}$$

and $h_{j_{k+1/2}}^n$ is defined in an analogous manner. Recall that from HW9.7.10 we know that for a linear, scalar problem, the Godunov scheme is the same as the linear upwind scheme. Hence, we know that

$$h_{j_{k+1/2}}^n = \begin{cases} \nu_j U_k^n & \text{if } \nu_j \ge 0 \\ \nu_j U_{k+1}^n & \text{if } \nu_j < 0. \end{cases}$$

We write D as $D = D_+ + D_-$ where D_+ and D_- are the $K \times K$ diagonal matrices containing the positive and negative eigenvalues of D (in the correct places), respectively. Since for positive ν_j, equation (9.9.3) can be written as

$$U_{j_k}^{n+1} = U_{j_k}^n - \nu_j R\delta_- U_{j_k}^n \tag{9.9.4}$$

and for negative ν_j, equation (9.9.3) can be written as

$$U_{j_k}^{n+1} = U_{j_k}^n - \nu_j R\delta_+ U_{j_k}^n, \tag{9.9.5}$$

equation (9.9.3) can be written as

$$\mathbf{U}_k^{n+1} = \mathbf{U}_k^n - RD_-\delta_+\mathbf{U}_k^n - RD_+\delta_-\mathbf{U}_k^n. \tag{9.9.6}$$

If we multiply equation (9.9.6) on the left by S^{-1} and let $\mathbf{u}_k^n = S^{-1}\mathbf{U}_k^n$, difference scheme (9.9.6) can be written in terms of primitive variables as

$$\mathbf{u}_k^{n+1} = \mathbf{u}_k^n - RA_-\delta_+\mathbf{u}_k^n - RA_+\delta_-\mathbf{u}_k^n \tag{9.9.7}$$

where $A_- = S^{-1}D_-S$ and $A_+ = S^{-1}D_+S$. Hence, we see that for linear K-system conservation laws, the Godunov scheme is the same as the flux splitting scheme introduced in Section 6.2.1 (emphasizing that we have reversed the signs of the eigenvalues by writing the partial differential equation as (9.9.1) instead of as in Chapter 6.

9.9.2 *Godunov Schemes for K-System Conservation Laws*

We might hope that since the Godunov scheme for linear systems was so easy, the Godunov scheme for general K-system conservation laws might be reasonably nice. In one way, that is true. If we return to Section 9.7.3, we note that almost everything that we did holds equally true for K-system conservation laws. Specifically, we assume that we have the solution at the n-th time level, $\mathbf{u}_k^n$, $k = -\infty, \ldots, \infty$. We let $\bar{\mathbf{U}} = \bar{\mathbf{U}}(x,t)$ be the solution to the initial-value problem

$$\mathrm{v}_t + \mathbf{F}(\mathrm{v})_x = \boldsymbol{\theta}, \quad x \in \mathbb{R},\ t > t_n \tag{9.9.8}$$

$$\mathrm{v}(x,t_n) = \mathbf{u}_k^n \quad x \in (x_{k-1/2}, x_{k+1/2}), \quad k = -\infty, \ldots, \infty, \tag{9.9.9}$$

i.e., for a sufficiently small time interval (t_n, t_{n+1}), let $\bar{\mathbf{U}} = \bar{\mathbf{U}}(x,t)$ be made up of local solutions to the following Riemann problems.

$$\mathrm{v}_t + \mathbf{F}(\mathrm{v})_x = \boldsymbol{\theta}, \quad x \in \mathbb{R},\ t > t_n \tag{9.9.10}$$

$$\mathrm{v}(x,t_n) = \begin{cases} \mathbf{u}_k^n & \text{if } x < x_{k+1/2} \\ \mathbf{u}_{k+1}^n & \text{if } x \geq x_{k+1/2}. \end{cases} \tag{9.9.11}$$

If we set

$$\mathbf{u}_k^{n+1} = \frac{1}{\Delta x} \int_{x_{k-1/2}}^{x_{k+1/2}} \bar{\mathbf{U}}(x, t_{n+1})\, dx \tag{9.9.12}$$

and integrate conservation law (9.9.10) with respect to x and t from $x_{k-1/2}$ to $x_{k+1/2}$ and t_n to t_{n+1}, respectively, we obtain a special case of the integral form of the conservation law

$$\begin{aligned} 0 = &\int_{x_{k-1/2}}^{x_{k+1/2}} \bar{\mathbf{U}}(x, t_{n+1})\, dx - \int_{x_{k-1/2}}^{x_{k+1/2}} \bar{\mathbf{U}}(x, t_n)\, dx \\ &+ \int_{t_n}^{t_{n+1}} \left[\mathbf{F}(\bar{\mathbf{U}}(x_{k+1/2}, t)) - \mathbf{F}(\bar{\mathbf{U}}(x_{k-1/2}, t))\right] dt \end{aligned} \tag{9.9.13}$$

or the analogue to equation (9.7.59),

$$\mathbf{u}_k^{n+1} = \mathbf{u}_k^n - \frac{1}{\Delta x} \int_{t_n}^{t_{n+1}} \left[\mathbf{F}(\bar{\mathbf{U}}(x_{k+1/2}, t)) - \mathbf{F}(\bar{\mathbf{U}}(x_{k-1/2}, t))\right] dt. \tag{9.9.14}$$

As was the case for the scalar Godunov scheme, we can set

$$\mathbf{h}_{k+1/2}^n = \frac{1}{\Delta t} \int_{t_n}^{t_{n+1}} \mathbf{F}(\bar{\mathbf{U}}(x_{k+1/2}, t))\, dt, \tag{9.9.15}$$

and because $\bar{\mathbf{U}}(x_{k+1/2}, t)$ does not depend on t (as was the case for scalar equations), $\mathbf{h}_{k+1/2}^n$ can be written as

$$\mathbf{h}_{k+1/2}^n = \mathbf{F}(\bar{\mathbf{U}}(x_{k+1/2}, t)). \tag{9.9.16}$$

Difference scheme (9.9.14) can then be written as

$$\mathbf{u}_k^{n+1} = \mathbf{u}_k^n - R\delta_- \mathbf{h}_{k+1/2}^n. \tag{9.9.17}$$

Difference equation (9.9.16)–(9.9.17) is the Godunov scheme for a general K-system conservation law.

The difficulty that we encounter using difference scheme (9.9.16)–(9.9.17) is that *we are not able to determine the exact solutions to the local Riemann problems.* There are some iterative methods that can be used, but these are too difficult and too expensive computationally. We will now proceed to find an approximate alternative to difference scheme (9.9.16)–(9.9.17).

9.9.2.1 Return to the Godunov Scheme for Linear K-System Conservation Laws

Another way to approach the Godunov scheme for the linear K-system conservation law is to find a solution to the linear analogue to Riemann problem (9.9.10)–(9.9.11) and use (9.9.16) to define the numerical flux function $\mathbf{h}_{k+1/2}^n$, i.e.,

$$\mathbf{h}_{k+1/2}^n = A\bar{\mathbf{U}}(x_{k+1/2}, t). \tag{9.9.18}$$

If the eigenvalues and eigenvectors of the matrix A are given by ν_j, $j = 1, \ldots, K$ and $\mathbf{r}_j$, $j = 1, \ldots, K$, respectively, and we write $\delta_+ \mathbf{u}_k^n$ as

$$\mathbf{u}_{k+1}^n - \mathbf{u}_k^n = \sum_{j=1}^{K} \alpha_{j_k} \mathbf{r}_j, \tag{9.9.19}$$

we see by equations (9.3.34) and (9.3.35) (along with equation (9.3.33)) of Section 9.3.1 that the solution $\bar{\mathbf{U}}$ can be written as

$$\bar{\mathbf{U}}(x_{k+1/2}, t) = \mathbf{u}_k^n + \sum_{\nu_j < 0} \alpha_{j_k} \mathbf{r}_j \tag{9.9.20}$$

and

$$\bar{\mathbf{U}}(x_{k+1/2}, t) = \mathbf{u}_{k+1}^n - \sum_{\nu_j > 0} \alpha_{j_k} \mathbf{r}_j. \tag{9.9.21}$$

We note that the difference in sign before the summation sign here and in Section 9.3.1 is due to the fact that in Section 9.3.1 we expand $\mathbf{v}_L - \mathbf{v}_R$ in terms of the eigenvectors of A, whereas here we expand the equivalent of $\mathbf{v}_R - \mathbf{v}_L$. Then using (9.9.18), we can write $\mathbf{h}_{k+1/2}^n$ as either

$$\begin{aligned} \mathbf{h}_{k+1/2}^n &= A\bar{\mathbf{U}}(x_{k+1/2}, t) = A\mathbf{u}_k^n + \sum_{\nu_j < 0} \alpha_{j_k} A\mathbf{r}_j \\ &= A\mathbf{u}_k^n + \sum_{\nu_j < 0} \alpha_{j_k} \nu_j \mathbf{r}_j \end{aligned} \tag{9.9.22}$$

or

$$\mathbf{h}_{k+1/2}^n = A\bar{\mathbf{U}}(x_{k+1/2},t) = A\mathbf{u}_{k+1}^n - \sum_{\nu_j>0} \alpha_{j_k} A\mathbf{r}_j$$

$$= A\mathbf{u}_{k+1}^n - \sum_{\nu_j>0} \alpha_{j_k}\nu_j\mathbf{r}_j. \tag{9.9.23}$$

If we use (9.9.22) to define $\mathbf{h}_{k+1/2}^n$ and (9.9.23) (at $k-1$)) to define $\mathbf{h}_{k-1/2}^n$, we have the following difference scheme.

$$\mathbf{u}_k^{n+1} = \mathbf{u}_k^n - R\big[\mathbf{h}_{k+1/2}^n - \mathbf{h}_{k-1/2}^n\big]$$

$$= \mathbf{u}_k^n - R\Big[\sum_{\nu_j<0} \alpha_{j_k}\nu_j\mathbf{r}_j + \sum_{\nu_j>0} \alpha'_{j_k}\nu_j\mathbf{r}_j\Big], \tag{9.9.24}$$

where like the α_{j_k}, α'_{j_k} are the coefficients in the expansion of $\delta_+\mathbf{u}_{k-1}^n$. It is not difficult to see that difference scheme (9.9.24) is the same as difference scheme (9.9.7) found in Section 9.9.1.

If instead of choosing $\mathbf{h}_{k\pm1/2}^n$ as described above we set $\mathbf{h}$ to be the average of $\mathbf{h}$ given in (9.9.22) and (9.9.23), we get

$$\mathbf{h}_{k+1/2}^n = \frac{1}{2}A\big(\mathbf{u}_k^n + \mathbf{u}_{k+1}^n\big) + \frac{1}{2}\sum_{\nu_j<0} \alpha_{j_k}\nu_j\mathbf{r}_j - \frac{1}{2}\sum_{\nu_j>0} \alpha_{j_k}\nu_j\mathbf{r}_j \tag{9.9.25}$$

$$= \frac{1}{2}A\big(\mathbf{u}_k^n + \mathbf{u}_{k+1}^n\big) - \frac{1}{2}\big(A_+ - A_-\big)\delta_+\mathbf{u}_k^n \tag{9.9.26}$$

$$= \frac{1}{2}A\big(\mathbf{u}_k^n + \mathbf{u}_{k+1}^n\big) - \frac{1}{2}|A|\delta_+\mathbf{u}_k^n, \tag{9.9.27}$$

where $|A| = A_+ - A_-$ and is also equal to $S^{-1}|D|S$. Both difference scheme (9.9.7) and the scheme associated with numerical flux function (9.9.27) will be referred to as upwind schemes for the linear K-system conservation law.

We should realize that for the linear K-system, it is difficult to see why anyone would want to write the Godunov scheme in the form given in (9.9.24). When A remains the same from grid point to grid point and time step to time step, it might pay to compute A_- and A_+ once, and proceed. However, as we shall see later, when we apply the analogous result to the situation where we have a different matrix at each grid point and time step (the matrix will be the linearization of the nonlinear problem, hence depending generally on $\mathbf{u}_k^n$ and $\mathbf{u}_{k+1}^n$), it would be necessary to recompute A_- and A_+ at each grid point. In that case, it is easier to compute ν_j, $j = 1,\ldots,K$, $\mathbf{r}_j$, $j = 1,\ldots,K$ and α_{j_k}, $j = 1,\ldots,K$ at each grid point and use difference scheme (9.9.24).

Remark: In (9.9.19) we write $\delta_+\mathbf{u}_k^n$ in terms of the eigenvectors $\mathbf{r}_j$, $j = 1,\ldots,K$. The way to compute the coefficients α_{j_k}, $j = 1,\ldots,K$ is to let R be the $K\times K$ matrix $R = \begin{bmatrix}\mathbf{r}_1 & \cdots & \mathbf{r}_K\end{bmatrix}$ and solve the system of equations $R\boldsymbol{\alpha}_k = \delta_+\mathbf{u}_k^n$ where $\boldsymbol{\alpha}_k = \begin{bmatrix}\alpha_{k_1} & \cdots & \alpha_{k_K}\end{bmatrix}^T$.

HW 9.9.1 Show that difference scheme (9.9.24) is the same as difference scheme (9.9.7).

9.9.3 Approximate Riemann Solvers: Theory

For nonlinear conservation laws, we are unable to find h because we cannot compute $\bar{U}$. The obvious approach is to find an approximation to difference scheme (9.9.16)–(9.9.17) by finding an approximation to $\bar{\mathbf{U}}$, $\hat{\mathbf{U}}$; use $\hat{\mathbf{U}}$ in equation (9.9.16) to determine an approximation of $\mathbf{h}^n_{k+1/2}$, $\hat{\mathbf{h}}^n_{k+1/2}$; and use $\hat{\mathbf{h}}$ in place of $\mathbf{h}$ in difference scheme (9.9.17). The difficulty with this approach is that we are unable to do any general analyses that will determine how well our resulting approximate solution to the difference equation approximates either the exact solution to difference scheme (9.9.16)–(9.9.17) or the solution of the original conservation law.

We present an approach where we retreat further in the analysis that enables us to obtain results related to the quality of the resulting solution. We find an approximate solution to the local Riemann problems and use these approximations back in the integral form of the conservation law. Let $\hat{\psi}_k$ be the approximate solution to the local Riemann problem centered at $x = x_{k+1/2}$, (9.9.10)–(9.9.11). We require that the approximate Riemann solution $\hat{\psi}$ be given in the form of a similarity solution $\hat{\psi}_k = \hat{\psi}_k\big((x - x_{k+1/2})/(t - t_n)\big)$ and satisfy

$$\hat{\psi}_k\big((x - x_{k+1/2})/(t - t_n)\big) = \begin{cases} \mathbf{v}_L & \text{when } (x - x_{k+1/2})/(t - t_n) < \nu_{\min} \\ \mathbf{v}_R & \text{when } (x - x_{k+1/2})/(t - t_n) > \nu_{\max} \end{cases} \tag{9.9.28}$$

where $\nu_{\min}$ and $\nu_{\max}$ are the minimum and maximum signal speeds which will be approximately the minimum and maximum eigenvalues of $\mathbf{F}'$ evaluated at $\mathbf{u}^n_k$ and $\mathbf{u}^n_{k+1}$.

We let $\hat{\mathbf{U}}$ denote the approximate solution to initial-value problem (9.9.8)–(9.9.9) obtained by piecing together the approximate solutions to the local Riemann problems $\hat{\psi}_k$, $j = -\infty, \ldots, \infty$. Similarly to what we have done before, we let

$$\mathbf{u}^{n+1}_k = \frac{1}{\Delta x} \int_{x_{k-1/2}}^{x_{k+1/2}} \hat{\mathbf{U}}(x, t_{n+1})\, dx. \tag{9.9.29}$$

We note that the solution given in (9.9.29) can be written in terms of $\hat{\psi}_k$'s as

$$\begin{aligned} \mathbf{u}^{n+1}_k = {} & \frac{1}{\Delta x} \int_{x_{k-1/2}}^{x_k} \hat{\psi}_{k-1}\big((x - x_{k-1/2})/(t_{n+1} - t_n)\big)\, dx \\ & + \frac{1}{\Delta x} \int_{x_k}^{x_{k+1/2}} \hat{\psi}_k\big((x - x_{k+1/2})/(t_{n+1} - t_n)\big)\, dx. \end{aligned} \tag{9.9.30}$$

As is usually the case for Godunov schemes, we must require that the local Riemann problems do not interact, i.e., $R\nu \le \frac{1}{2}$ where

$$\nu = \max\{|\nu_{\min}|, |\nu_{\max}|\}.$$

As is also the case with all Godunov schemes, after the numerical flux function is defined, it will still be consistent with the flux function if we relax the CFL condition to $R\nu \le 1$.

We next proceed to describe how to find approximate Riemann solutions. Obviously, we want to choose our approximate solutions carefully. We have already required that we find similarity solutions. This is not especially difficult, because as we shall see and as we could probably guess, our approximate solutions will generally be solutions to some sort of linear approximation, which will give us similarity solutions. However, we need more. We next state a strong result due to Lax and Harten, Theorem 2.1, ref. [26]. The proposition will be given without proof. (See also, ref. [27].)

Let $S = S(\mathrm{v})$ and $\Phi = \Phi(\mathrm{v})$ denote the entropy and entropy flux functions associated with conservation law (9.9.8), and let $\Psi_{k+1/2} = \Psi(\mathbf{u}_k, \mathbf{u}_{k+1})$ denote a numerical entropy flux function that is consistent with the entropy flux function Φ.

Proposition 9.9.1 *Suppose that the approximate solutions to the Riemann problems (9.9.10)–(9.9.11), $\hat{\psi}_k$, $k = -\infty, \dots, \infty$, satisfy the following conditions.*
(1) *Approximate solution $\hat{\psi}_k$ is consistent with the integral form of the conservation law in that*

$$\int_{x_k}^{x_{k+1}} \hat{\psi}_k\big((x - x_{k+1/2})/(t_{n+1} - t_n)\big)\, dx = \frac{\Delta x}{2}\big(\mathbf{u}_k^n + \mathbf{u}_{k+1}^n\big) - \Delta t\big(\mathbf{F}_{k+1}^n - \mathbf{F}_k^n\big) \qquad (9.9.31)$$

for $R\nu \le \frac{1}{2}$ where $|\nu_j| \le \nu$ for all eigenvalues ν_j of $\mathbf{F}'(\mathbf{u}_k^n)$ and $\mathbf{F}'(\mathbf{u}_{k+1}^n)$.
(2) *Approximate solution $\hat{\psi}_k$ is consistent with the integral form of the entropy condition in that*

$$\int_{x_k}^{x_{k+1}} S\big(\hat{\psi}_k\big((x - x_{k+1/2})/(t_{n+1} - t_n)\big)\big)\, dx \le \frac{\Delta x}{2}\big(S(\mathbf{u}_k^n) + S(\mathbf{u}_{k+1}^n)\big) - \Delta t\big(\Psi_{k+1/2}^n - \Psi_{k-1/2}^n\big) \qquad (9.9.32)$$

for $R\nu \le \frac{1}{2}$ where ν is as in (1). *Then difference scheme (9.9.29) (or (9.9.30)) is conservative, consistent with conservation law (9.9.8) and satisfies the entropy inequality (9.6.32).*

Remark: If we integrate conservation law (9.9.8) with respect to x from x_k to x_{k+1} and with respect to t from t_n to t_{n+1}, we get

$$\boldsymbol{\theta} = \int_{x_k}^{x_{k+1}} \mathbf{v}(x, t_{n+1})\, dx - \int_{x_k}^{x_{k+1}} \mathbf{v}(x, t_n)\, dx + \int_{t_n}^{t_{n+1}} \left[\mathbf{F}(\mathbf{v}(x_{k+1}, t)) - \mathbf{F}(\mathbf{v}(x_k, t))\right] dt. \tag{9.9.33}$$

We know that

$$\mathbf{v}(x, t_n) = \begin{cases} \mathbf{u}_k^n & \text{when } x_k \le x \le x_{k+1/2} \\ \mathbf{u}_{k+1}^n & \text{when } x_{k+1/2} \le x \le x_{k+1}. \end{cases}$$

Also, we have chosen Δx and Δt so that $\mathbf{v}(x_{k+1}, t) = \mathbf{u}_{k+1}^n$ and $\mathbf{v}(x_k, t) = \mathbf{u}_k^n$. Equation (9.9.33) becomes

$$\int_{x_k}^{x_{k+1}} \mathbf{v}(x, t_{n+1})\, dx - \frac{\Delta x}{2}\mathbf{u}_k^n - \frac{\Delta x}{2}\mathbf{u}_{k+1}^n + \Delta t\left[\mathbf{F}(\mathbf{u}_{k+1}^n) - \mathbf{F}(u_k^n)\right]. \tag{9.9.34}$$

Thus we see that assumption (1) (equation (9.9.31)) of Proposition 9.9.1 is equivalent to assuming that the approximate solution $\hat{\mathbf{U}}$ will be conservative on the region $(x_k, x_{k+1}) \times (t_n, t_{n+1})$.

If we apply the integral form of the conservation law on the region $(x_k, x_{k+1/2}) \times (t_n, t_{n+1})$, we get

$$\int_{x_k}^{x_{k+1/2}} \left[\mathbf{v}(x, t_{n+1}) - \mathbf{v}(x, t_n)\right] dx - \int_{t_n}^{t_{n+1}} \left[\mathbf{F}(\mathbf{v}(x_{k+1/2}, t)) - \mathbf{F}(\mathbf{v}(x_k, t))\right] dt = \boldsymbol{\theta}.$$

Replacing $\mathbf{v}(x, t_{n+1})$ by $\hat{\mathbf{U}}(x, t_{n+1})$, $\mathbf{v}(x, t_n)$ by $\mathbf{u}_k^n$, $\mathbf{v}(x_k, t)$ by $\mathbf{u}_k^n$, and setting the flux term at $x = x_{k+1/2}$ equal to $\Delta t\, \mathbf{h}_{k+1/2}^{n-}$ gives

$$\mathbf{h}_{k+1/2}^{n-} = -\frac{1}{\Delta t}\int_{x_k}^{x_{k+1/2}} \hat{\mathbf{U}}(x, t_{n+1})\, dx + \frac{\Delta x}{2\Delta t}\mathbf{u}_k^n + \mathbf{F}_k^n. \tag{9.9.35}$$

We note at this time that we have no special reason for replacing $\mathbf{v}(x, t_{n+1})$ by $\hat{\mathbf{U}}(x, t_{n+1})$ other than that we want to. We can calculate $\hat{\mathbf{U}}$ and cannot calculate $\mathbf{v}$.

In a similar way, if we apply the integral form of the conservation law on the region $(x_{k+1/2}, x_{k+1}) \times (t_n, t_{n+1})$, we get

$$\mathbf{h}_{k+1/2}^{n+} = \frac{1}{\Delta t}\int_{x_{k+1/2}}^{x_{k+1}} \hat{\mathbf{U}}(x, t_{n+1})\, dx - \frac{\Delta x}{2\Delta t}\mathbf{u}_{k+1}^n + \mathbf{F}_{k+1}^n. \tag{9.9.36}$$

We see that $\mathbf{h}_{k+1/2}^{n+} = \mathbf{h}_{k+1/2}^{n-}$ (which is something that we really want) if

$$\frac{1}{\Delta t}\int_{x_{k+1/2}}^{x_{k+1}} \hat{\mathbf{U}}(x, t_{n+1})\, dx - \frac{\Delta x}{2\Delta t}\mathbf{u}_{k+1}^n + \mathbf{F}_{k+1}^n = -\frac{1}{\Delta t}\int_{x_k}^{x_{k+1/2}} \hat{\mathbf{U}}(x, t_{n+1})\, dx + \frac{\Delta x}{2\Delta t}\mathbf{u}_k^n + \mathbf{F}_k^n.$$

But this equality is just a restatement of assumption (9.9.31). This fact is what logically allows us to replace $\mathbf{v}$ by $\hat{\mathbf{U}}$.

We can then use $\mathbf{h}^{n-}_{k+1/2}$ and $\mathbf{h}^{n+}_{k-1/2}$ as our numerical flux functions $\mathbf{h}^n_{k+1/2}$ and $\mathbf{h}^n_{k-1/2}$, respectively, and arrive at the numerical scheme

$$\mathbf{u}_k^{n+1} = \mathbf{u}_k^n - R\big[\mathbf{h}^n_{k+1/2} - \mathbf{h}^n_{k-1/2}\big]. \tag{9.9.37}$$

Obviously, difference scheme (9.9.37) is a conservative scheme.

Since we can write $\mathbf{h}^n_{k\pm1/2}$ as

$$\begin{aligned}\mathbf{h}^n_{k+1/2} &= -\frac{1}{\Delta t}\int_{x_k}^{x_{k+1/2}} \hat{\mathbf{U}}(x,t_{n+1})\,dx + \frac{\Delta x}{2\Delta t}\mathbf{u}_k^n + \mathbf{F}_k^n \\ &= -\frac{1}{\Delta t}\int_{x_k}^{x_{k+1/2}} \hat{\psi}_k\big((x-x_{k+1/2})/(t_{n+1}-t_n)\big)\,dx \\ &\quad + \frac{\Delta x}{2\Delta t}\mathbf{u}_k^n + \mathbf{F}_k^n \end{aligned} \tag{9.9.38}$$

and

$$\begin{aligned}\mathbf{h}^n_{k-1/2} &= \frac{1}{\Delta t}\int_{x_{k-1/2}}^{x_k} \hat{\mathbf{U}}(x,t_{n+1})\,dx - \frac{\Delta x}{2\Delta t}\mathbf{u}_k^n + \mathbf{F}_k^n \\ &= \frac{1}{\Delta t}\int_{x_{k-1/2}}^{x_k} \hat{\psi}_{k-1}\big((x-x_{k-1/2})/(t_{n+1}-t_n)\big)\,dx \\ &\quad - \frac{\Delta x}{2\Delta t}\mathbf{u}_k^n + \mathbf{F}_k^n, \end{aligned} \tag{9.9.39}$$

difference scheme (9.9.37) can be written as

$$\begin{aligned}\mathbf{u}_k^{n+1} =& \mathbf{u}_k^n - R\big[\mathbf{h}^n_{k+1/2} - \mathbf{h}^n_{k-1/2}\big] \\ =& \mathbf{u}_k^n - R\Bigg\{ -\frac{1}{\Delta t}\int_{x_k}^{x_{k+1/2}} \hat{\psi}_k\big((x-x_{k+1/2})/(t_{n+1}-t_n)\big)\,dx + \frac{\Delta x}{2\Delta t}\mathbf{u}_k^n \\ &\quad + \mathbf{F}_k^n - \frac{1}{\Delta t}\int_{x_{k-1/2}}^{x_k} \hat{\psi}_{k-1}\big((x-x_{k-1/2})/(t_{n+1}-t_n)\big)\,dx \\ &\quad + \frac{\Delta x}{2\Delta t}\mathbf{u}_k^n - \mathbf{F}_k^n \Bigg\} \\ =& \frac{1}{\Delta x}\int_{x_k}^{x_{k+1/2}} \hat{\psi}_k\big((x-x_{k+1/2})/(t_{n+1}-t_n)\big)\,dx \\ &+ \frac{1}{\Delta x}\int_{x_{k-1/2}}^{x_k} \hat{\psi}_{k-1}\big((x-x_{k-1/2})/(t_{n+1}-t_n)\big)\,dx. \end{aligned}$$

Thus we see that difference scheme (9.9.37) is the same as difference scheme (9.9.30), and difference scheme (9.9.29) (or (9.9.30)) is conservative.

9.9.4 *Approximate Riemann Solvers: Applications*

One approach to find an approximation to the solution of the Riemann problem is to alter the conservation law (similar to the modified flux method for scalar equations) to a form that makes the approximate Riemann problem solvable. If we consider the approximation to conservation law (9.9.8),

$$\hat{\mathbf{U}}_t + \hat{\mathbf{F}}_x = \boldsymbol{\theta}, \tag{9.9.40}$$

and apply the integral form of the conservation law to conservation law (9.9.40) over the region $(x_k, x_{k+1}) \times (t_n, t_{n+1})$, we get

$$\int_{x_k}^{x_{k+1}} \left[\hat{\mathbf{U}}(x, t_{n+1}) - \hat{\mathbf{U}}(x, t_n)\right] dx + \int_{t_n}^{t_{n+1}} \left[\hat{\mathbf{F}}\left(\hat{\mathbf{U}}(x_{k+1}, t)\right) - \hat{\mathbf{F}}\left(\hat{\mathbf{U}}(x_k, t)\right)\right] dt = \boldsymbol{\theta}. \tag{9.9.41}$$

Since $\hat{\mathbf{U}}(x, t_n) = \mathbf{u}_k^n$ for $x_k \le x \le x_{k+1/2}$, $\hat{\mathbf{U}}(x, t_n) = \mathbf{u}_{k+1}^n$ for $x_{k+1/2} \le x \le x_{k+1}$, and Δx and Δt are chosen sufficiently small so that $\hat{\mathbf{U}}(x_k, t) = \mathbf{u}_k^n$ and $\hat{\mathbf{U}}(x_{k+1}, t) = \mathbf{u}_{k+1}^n$ for $t \in [t_n, t_{n+1}]$ (the bound $R\nu \le \frac{1}{2}$ will now involve the eigenvalues of $\hat{\mathbf{F}}'$), (9.9.41) reduces to

$$\int_{x_k}^{x_{k+1}} \hat{\mathbf{U}}(x, t_{n+1})\, dx = \frac{\Delta x}{2}\left(\mathbf{u}_k^n + \mathbf{u}_k^{n+1}\right) - \Delta t\left(\hat{\mathbf{F}}_{k+1}^n - \hat{\mathbf{F}}_k^n\right). \tag{9.9.42}$$

Thus, it is easy to see that the solution to Riemann problem (9.9.40), (9.9.11) will satisfy condition (9.9.31) if

$$\hat{\mathbf{F}}_{k+1}^n - \hat{\mathbf{F}}_k^n = \mathbf{F}_{k+1}^n - \mathbf{F}_k^n. \tag{9.9.43}$$

It is not difficult to see that satisfying condition (9.9.43) is not enough to make an approximate Riemann problem readily usable. We must be able to solve the approximate Riemann problem. The most obvious way to find an approximation to conservation law (9.9.8) that we can solve is to replace conservation law (9.9.8) by a linear problem. However, this is too drastic. All we need is a solution to a Riemann problem centered at $x = x_{k+1/2}$. There is no reason to expect that some linear operator should work at $x = x_{k+1/2}$ for all k. We will instead replace conservation law (9.9.8) by a locally linearized equation. We consider the local Riemann problem centered at $x = x_{k+1/2}$ and a linear approximation of conservation law (9.9.8) of the form

$$\hat{\mathbf{U}}_t + \hat{A}\hat{\mathbf{U}}_x = \boldsymbol{\theta} \tag{9.9.44}$$

where $\hat{A} = \hat{A}(\mathbf{u}_k^n, \mathbf{u}_{k+1}^n)$. We emphasize that since $\hat{A}$ depends on $\mathbf{u}_k^n$ and $\mathbf{u}_{k+1}^n$ at each interface, our approximate Riemann problem is a very nonlinear problem. Roe, ref. [58], suggested that $\hat{A}$ should be chosen such that $\hat{A}$ will satisfy the following conditions.

(1) $\hat{A}(\mathbf{u}_L, \mathbf{u}_R)(\mathbf{u}_R - \mathbf{u}_L) = \mathbf{F}(\mathbf{u}_R) - \mathbf{F}(\mathbf{u}_L)$ (9.9.45)

(2) $\hat{A}(\mathbf{u}_L, \mathbf{u}_R)$ is diagonalizable with real eigenvalues

(3) $\hat{A}(\mathbf{u}_L, \mathbf{u}_R) \to \mathbf{F}'(\mathbf{u})$ smoothly as $\mathbf{u}_L,\ \mathbf{u}_R \to \mathbf{u}$

where in our case $\mathbf{u}_L = \mathbf{u}_k^n$ and $\mathbf{u}_R = \mathbf{u}_{k+1}^n$. It should be reasonably clear that assumption (2) is required, so that the approximate problem, like the original conservation law, is hyperbolic and so that we have some chance of solving the resulting problem. Assumption (3) is necessary so that as $\Delta x,\ \Delta t \to 0$ and as the numerical solution is converging to the analytic solution, the approximate conservation law will converge to the original conservation law. It would be highly improbable that condition (3) be not satisfied and the resulting difference scheme be consistent.

Assumption (1) above is very important in that assumption (1) implies that condition (1) of Proposition 9.9.1 is satisfied. If we apply the integral form of the conservation law to conservation law (9.9.44), we see that

$$\begin{aligned}\boldsymbol{\theta} &= \int_{x_k}^{x_{k+1}} \left[\hat{\mathbf{U}}(x, t_{n+1}) - \hat{\mathbf{U}}(x, t_n)\right] dx + \int_{t_n}^{t_{n+1}} \hat{A}\left[\hat{\mathbf{U}}(x_{k+1}, t) - \hat{\mathbf{U}}(x_k, t)\right] dt \\ &= \int_{x_k}^{x_{k+1}} \hat{\mathbf{U}}(x, t_{n+1})\, dx - \frac{\Delta x}{2}\left(\mathbf{u}_k^n + \mathbf{u}_{k+1}^n\right) + \int_{t_n}^{t_{n+1}} \hat{A}\left(\mathbf{u}_{k+1}^n - \mathbf{u}_k^n\right) dt \\ &= \int_{x_k}^{x_{k+1}} \hat{\mathbf{U}}(x, t_{n+1})\, dx - \frac{\Delta x}{2}\left(\mathbf{u}_k^n + \mathbf{u}_{k+1}^n\right) + \Delta t\left[\mathbf{F}_{k+1}^n - \mathbf{F}_k^n\right].\end{aligned}$$

The last step is due to the first Roe assumption, (9.9.45). Thus we see that the Roe condition (1) implies that the approximate solution will satisfy condition (9.9.31).

A second property that we obtain from assumption (1) is in the case that $\mathbf{u}_L$ and $\mathbf{u}_R$ are connected by a single shock or contact discontinuity. Since $\mathbf{u}_L$ and $\mathbf{u}_R$ must satisfy the Rankine-Hugoniot condition with respect to conservation law (9.9.44), we get

$$\hat{A}\mathbf{u}_R - \hat{A}\mathbf{u}_L = s(\mathbf{u}_R - \mathbf{u}_L).$$

We see that by condition (9.9.45), $\mathbf{u}_L$ and $\mathbf{u}_R$ will also satisfy the Rankine-Hugoniot condition with respect to conservation law (9.9.8)

$$\mathbf{F}(\mathbf{u}_R) - \mathbf{F}(\mathbf{u}_L) = s(\mathbf{u}_R - \mathbf{u}_L).$$

To solve the Riemann problem consisting of a conservation law (9.9.44) that satisfies the Roe assumptions along with initial condition (9.9.11), we use the first Roe assumption, (9.9.45),

$$\hat{A}\left(\mathbf{u}_{k+1}^n - \mathbf{u}_k^n\right) = \mathbf{F}_{k+1}^n - \mathbf{F}_k^n.$$

Expand $\delta_+ \mathbf{u}_k^n$ in terms of the eigenvectors of $\hat{A}$ (assumed to be $\mathbf{r}_{1_k}, \ldots, \mathbf{r}_{K_k}$) as

$$\delta_+ \mathbf{u}_k^n = \sum_{j=1}^{K} \alpha_{j_k} \mathbf{r}_{j_k}.$$

Then

$$\mathbf{F}_{k+1}^n - \mathbf{F}_k^n = \hat{A}\delta_+ \mathbf{u}_k^n = \sum_{j=1}^{K} \alpha_{j_k} \hat{A}\mathbf{r}_{j_k} = \sum_{j=1}^{K} \alpha_{j_k} \nu_{j_k} \mathbf{r}_{j_k}, \tag{9.9.46}$$

where ν_{j_k}, $j = 1, \ldots, K$, are the eigenvalues of $\hat{A}$. Note that generally both the eigenvectors and eigenvalues of $\hat{A}$ will depend on k.

The solution to Riemann problem (9.9.44), (9.9.11) can be written, as in (9.3.34), as

$$\hat{\mathbf{U}}(x, t_{n+1}) = \mathbf{u}_k^n + \sum_{\nu_{j_k} < (x - x_{k+1/2})/\Delta t} \alpha_{j_k} \mathbf{r}_{j_k}. \tag{9.9.47}$$

Remark: The difference between solution (9.9.47) and solution (9.3.34) is due to the fact that in (9.3.34) we expand $\mathbf{v}_L - \mathbf{v}_R$ in terms of the eigenvectors of A, whereas in (9.9.47) we expand $\delta_+ u_k^n$, the equivalent of $\mathbf{v}_R - \mathbf{v}_L$. This operation changes the sign of α_{j_k}.

If we use solution (9.9.47) in equation (9.9.38), we get

$$\begin{aligned}
\mathbf{h}_{k+1/2}^n &= -\frac{1}{\Delta t} \int_{x_k}^{x_{k+1/2}} \hat{\mathbf{U}}(x, t_{n+1})\, dx + \frac{1}{2R}\mathbf{u}_k^n + \mathbf{F}_k^n \\
&= -\frac{1}{\Delta t}\left\{ \frac{\Delta x}{2}\mathbf{u}_k^n - \Delta t \sum_{\nu_{j_k} < 0} \alpha_{j_k} \nu_{j_k} \mathbf{r}_{j_k} \right\} + \frac{1}{2R}\mathbf{u}_k^n + \mathbf{F}_k^n \qquad (9.9.48) \\
&= \sum_{\nu_{j_k} < 0} \alpha_{j_k} \nu_{j_k} \mathbf{r}_{j_k} + \mathbf{F}_k^n. \qquad (9.9.49)
\end{aligned}$$

We note that the integration necessary to get to equation (9.9.48) is not trivial. The $\mathbf{u}_k^n$ term gets integrated from x_k to $x_{k+1/2}$. The $\alpha_{1_k}\mathbf{r}_{1_k}$ term gets integrated from $x = x_{k+1/2} + \nu_{1_k}\Delta t$ to $x_{k+1/2}$. The value $x_{k+1/2} + \nu_{1_k}\Delta t$ is obtained by evaluating the characteristic $x = x_{k+1/2} - \nu_{1_k}(t - t_n)$ at $t = t_{n+1}$. The $\alpha_{2_k}\mathbf{r}_{2_k}$ term gets integrated from $x = x_{k+1/2} + \nu_{2_k}\Delta t$ to $x_{k+1/2}$. This time the value $x_{k+1/2} + \nu_{2_k}\Delta t$ is the x-coordinate of the intersection of the second characteristic curve with $t = t_{n+1}$. Each time the line $t = t_{n+1}$ intersects another characteristic curve, another term in the summation in solution (9.9.47) kicks in and gets integrated from that point of intersection to $x_{k+1/2}$.

If we solve equation (9.9.46) for $\mathbf{F}_k^n$ and replace half of the $\mathbf{F}_k^n$ term in (9.9.49) with this value, we get

$$\begin{aligned}
\mathbf{h}_{k+1/2}^n &= \frac{1}{2}(\mathbf{F}_k^n + \mathbf{F}_{k+1}^n) + \frac{1}{2}\sum_{\nu_{j_k} < 0} \alpha_{j_k} \nu_{j_k} \mathbf{r}_{j_k} - \frac{1}{2}\sum_{\nu_{j_k} > 0} \alpha_{j_k} \nu_{j_k} \mathbf{r}_{j_k} \\
&= \frac{1}{2}(\mathbf{F}_k^n + \mathbf{F}_{k+1}^n) - \frac{1}{2}\sum_{j=1}^{K} \alpha_{j_k} |\nu_{j_k}| \mathbf{r}_{j_k}. \qquad (9.9.50)
\end{aligned}$$

And finally, we note that using (9.9.50), we can write $\mathbf{h}^n_{k+1/2}$ as

$$\mathbf{h}^n_{k+1/2} = \frac{1}{2}\big(\mathbf{F}^n_k + \mathbf{F}^n_{k+1}\big) - \frac{1}{2}|\hat{A}|\delta_+\mathbf{u}^n_k, \tag{9.9.51}$$

where $|\hat{A}| = \hat{A}_+ - \hat{A}_-$. We might also note that if we solve equation (9.9.46) for $\mathbf{F}^n_k$ and replace the $\mathbf{F}^n_k$ term in (9.9.49) with this value, we get

$$\mathbf{h}^n_{k+1/2} = \sum_{\nu_{j_k}>0} \alpha_{j_k}\nu_{j_k}\mathbf{r}_{j_k} + \mathbf{F}^n_{k+1}. \tag{9.9.52}$$

The numerical flux function (9.9.49) is the result of the exact solution to our approximate Riemann problem (which is the same as the local Godunov scheme associated with the approximate Riemann problem) treated so that it gives us an appropriate approximation to conservation law (9.9.8). For this reason, the difference scheme associated with numerical flux function (9.9.49) is one of our approximate Riemann problem analogues of the upwind scheme. We should understand that the upwind scheme associated with numerical flux function (9.9.49) will generally be dissipative and first order accurate. The difference scheme associated with numerical flux function (9.9.51) is an upwind scheme that takes advantage of the fact that $\hat{A}$ satisfies the Roe assumption (1). It should be clear that numerical flux function (9.9.51) is our approximate Riemann problem analogue of numerical flux function (9.9.27). The difference scheme associated with numerical flux function (9.9.51) will also be dissipative and first order accurate.

One of the common ways people try to define $\hat{A}$ is to set

$$\hat{A}\big(\mathbf{u}^n_k, \mathbf{u}^n_{k+1}\big) = \mathbf{F}'\big((\mathbf{u}^n_k + \mathbf{u}^n_{k+1})/2\big). \tag{9.9.53}$$

This choice of $\hat{A}$ will satisfy conditions (2) and (3). It is not difficult to see that when

$$\hat{A}\big(\mathbf{u}^n_k, \mathbf{u}^n_{k+1}\big)\big(\mathbf{u}^n_{k+1} - \mathbf{u}^n_k\big) = \mathbf{F}'\big(\big(\mathbf{u}^n_k + \mathbf{u}^n_{k+1}\big)/2\big)\big(\mathbf{u}^n_{k+1} - \mathbf{u}^n_k\big), \tag{9.9.54}$$

it will be the exception rather than the rule that the above expression will equal $\mathbf{F}^n_{k+1} - \mathbf{F}^n_k$, i.e., this choice of $\hat{A}$ will not generally satisfy Roe condition (1), (9.9.45). As we will see in HW9.9.2 and HW9.9.14, it is still possible to obtain very good solutions using linearization (9.9.53) along with the difference scheme associated with numerical flux functions (9.9.49) or (9.9.50). Another potential way to define $\hat{A}$ is to set

$$|\hat{A}\big(\mathbf{u}^n_k, \mathbf{u}^n_{k+1}\big)| = \frac{1}{2}\Big[|\mathbf{F}'\big(\mathbf{u}^n_k\big) + |\mathbf{F}'\big(\mathbf{u}^n_{k+1}\big)|\Big]. \tag{9.9.55}$$

There are general approaches for finding appropriate $\hat{A}$'s but, these methods are difficult. See ref. [26]. It is possible to derive the necessary linearizations more easily for special cases. In the next section we present

the appropriate linearization for the one dimensional Euler equations, the shock tube problem. Not only will this provide an example of defining the approximate Riemann problem, it will also provide us with a good setting for solving HW0.0.3.

9.9.4.1 Approximate Riemann Solvers: Euler Equations

It was in the Prelude to Part 1 that we introduced the shock tube problem. Though we wrote them differently, we introduce the Euler equations of gas dynamics,

$$\mathbf{v}_t + \mathbf{F}_x = \boldsymbol{\theta}, \tag{9.9.56}$$

where $\mathbf{v} = [\rho \ \rho v \ E]^T$, $p = (\gamma - 1)[E - \rho v^2/2]$ and

$$\mathbf{F} = \begin{bmatrix} \rho v \\ \rho v^2 + p \\ v(E + p) \end{bmatrix}. \tag{9.9.57}$$

The variables ρ, v, p and E are the density, velocity, pressure and total energy, respectively. We take $\gamma = 1.4$ and write the momentum as $m = \rho v$. In Section 6.10.2 we wrote out the derivative of $\mathbf{F}$, $\mathbf{F}'$, as

$$\mathrm{F}'(\mathbf{v}) = \begin{pmatrix} 0 & 1 & 0 \\ \frac{\gamma-3}{2}v^2 & -(\gamma-3)v & (\gamma-1) \\ \frac{\gamma-1}{2}v^3 - \frac{v}{\rho}(E+p) & \frac{1}{\rho}(E+p) - (\gamma-1)v^2 & \gamma v \end{pmatrix}, \tag{9.9.58}$$

and claimed that the eigenvalues and the associated eigenvectors of $\mathbf{F}'$ are given by

$$\nu_1 = v - c, \ \nu_2 = v \text{ and } \nu_3 = v + c, \tag{9.9.59}$$

$$\mathbf{r}_1 = \begin{bmatrix} 1 \\ v - c \\ \frac{1}{2}v^2 - vc + \frac{1}{\gamma-1}c^2 \end{bmatrix}, \ \mathbf{r}_2 = \begin{bmatrix} 1 \\ v \\ \frac{1}{2}v^2 \end{bmatrix},$$

$$\mathbf{r}_3 = \begin{bmatrix} 1 \\ v + c \\ \frac{1}{2}v^2 + vc + \frac{1}{\gamma-1}c^2 \end{bmatrix}, \tag{9.9.60}$$

respectively, where $c^2 = \gamma p/\rho$.

We now assume that we are solving a problem involving the Euler equations, (9.9.56)–(9.9.57). For example, we might be trying to solve the shock tube problem, HW0.0.3. We assume that we are trying to approximate the solution numerically, that we have a usual grid on $\mathbb{R}$ or some interval, and that we have a solution at time step n, $\mathbf{u}_k^n$. Since we want to use the approximate Riemann solver, we must develop an approximate Riemann problem at each cell interface, $x_{k+1/2}$. For the moment we will fix the index k, and

set $\mathbf{v}_L = \mathbf{u}_k^n$ and $\mathbf{v}_R = \mathbf{u}_{k+1}^n$. To define $\hat{A}$, we let $\mathbf{U} = \mathbf{U}(\mathbf{v}_L, \mathbf{v}_R)$ denote some sort of average of the vectors $\mathbf{v}_L$ and $\mathbf{v}_R$ (i.e., of the vectors $\mathbf{u}_k^n$ and $\mathbf{u}_{k+1}^n$) and set $\hat{A} = \mathbf{F}'(\mathbf{U})$. Denote the velocity and sound speed associated with $\mathbf{U}$ by $\hat{v}$ and $\hat{c}$ and the eigenvectors of $\hat{A}$ by $\mathbf{r}_j(\mathbf{U})$, $j = 1, 2, 3$. To apply the difference scheme associated with numerical flux function (9.9.49) (and the approach would be very similar to use the schemes associated with (9.9.50) or (9.9.52)), we must find α_1, α_2, α_3 so that we can write $\mathbf{v}_R - \mathbf{v}_L = \delta_+ u_k^n$ as

$$\mathbf{v}_R - \mathbf{v}_L = \sum_{j=1}^{3} \alpha_j \mathbf{r}_j(\mathbf{U}),$$

i.e., we must solve the equation $R\boldsymbol{\alpha} = \mathbf{v}_R - \mathbf{v}_L$ where $\boldsymbol{\alpha} = [\alpha_1\ \alpha_2\ \alpha_3]^T$ and R is the 3×3 matrix $R = [\mathbf{r}_1(\mathbf{U})\ \mathbf{r}_2(\mathbf{U})\ \mathbf{r}_3(\mathbf{U})]$. Solving this equation (an algebraic manipulator is nice to use here), we see that

$$\alpha_1 = \frac{1}{2}(\alpha - \beta),\ \alpha_2 = [\rho] - \alpha \text{ and } \alpha_3 = \frac{1}{2}(\alpha + \beta), \tag{9.9.61}$$

where

$$\alpha = (\gamma - 1)\{[E] + \frac{1}{2}\hat{v}^2[\rho] - \hat{v}[m]\}/\hat{c}^2, \tag{9.9.62}$$

$$\beta = \{[m] - \hat{v}[\rho]\}/\hat{c}^2, \tag{9.9.63}$$

and $[\rho] = \rho_R - \rho_L$, etc.

We are next left to determine how we will determine the average value $\mathbf{U}$. Roe's linearization technique is to define

$$\hat{v} = \frac{\sqrt{\rho_L}v_L + \sqrt{\rho_R}v_R}{\sqrt{\rho_L} + \sqrt{\rho_R}} \tag{9.9.64}$$

$$\hat{H} = \frac{\sqrt{\rho_L}H_L + \sqrt{\rho_R}H_R}{\sqrt{\rho_L} + \sqrt{\rho_R}} \tag{9.9.65}$$

$$\hat{c} = \sqrt{(\gamma - 1)(\hat{H} - \frac{1}{2}\hat{v}^2)}, \tag{9.9.66}$$

where $H = (E + p)/\rho$ is the enthalpy. At each k we use $\hat{v}$ and $\hat{c}$ as defined above in (9.9.59) and (9.9.60) to define the eigenvalues and eigenvectors of $\hat{A}$, respectively, and in (9.9.61)–(9.9.63) to define α_j, $j = 1, 2, 3$. We can then apply the difference scheme associated with numerical flux functions (9.9.49), (9.9.50) or (9.9.52) to advance the solution to the shock tube problem (or the Euler equations) one time step.

We note that another approach is to proceed as we mentioned earlier, set $\mathbf{U} = (\mathbf{v}_L + \mathbf{v}_R)/2$ and proceed to use the $\hat{v}$ and $\hat{c}$ defined in this manner to define $\nu_j(\bar{U})$, $j = 1, 2, 3$, $\mathbf{r}_j(\bar{U})$, $j = 1, 2, 3$, α, β, and α_j, $j = 1, 2, 3$.

The difference between these two approaches is that the matrix $\hat{A} = \hat{A}(\bar{U})$ where $\bar{U}$ is defined by (9.9.64)–(9.9.66) will satisfy Roe condition (9.9.45), whereas the matrix $\hat{A} = \hat{A}(\bar{U})$ defined using the straight average will not satisfy Roe condition (9.9.45).

HW 9.9.2 Solve the shock tube problem HW0.0.3 using numerical flux function (9.9.49) and $\hat{A}$ given by (9.9.53).

HW 9.9.3 Solve the shock tube problem HW0.0.3 using numerical flux function (9.9.49) and $\hat{A}$ defined by the Roe linearization (9.9.64)–(9.9.66).

HW 9.9.4 Use the difference scheme defined by numerical flux function (9.9.50) along with $\hat{A}$ as defined in (9.9.53) to solve the shock tube problem HW0.0.3.

HW 9.9.5 Use the difference scheme defined by numerical flux function (9.9.50) along with $\hat{A}$ defined by the Roe linearization (9.9.64)–(9.9.66) to solve the shock tube problem HW0.0.3.

9.9.4.2 Sonic Rarefaction Fix

As we saw in the computations done in HW9.4.5 and HW9.5.2, we must be very careful when computing in the presence of a sonic rarefaction. We have seen that most of the trouble that difference schemes seem to have producing entropy solutions is their inability to break an entropy violating jump in a sonic rarefaction. The problem here is more basic than the inability to break the sonic rarefaction. It would seem that the difference schemes defined by numerical flux functions (9.9.49), (9.9.51) and (9.9.52) will not even try to break a sonic rarefaction. When we use a linear approximation to the problem, even though it is a local linear approximation, we have eliminated fans from our existence. We should understand that it is not as easy as this. The solutions to the various Riemann problems do interact by the way we define our approximation $\mathbf{u}_k^{n+1}$. Those solutions that are interacting change at each grid point and for each time step, generally in a nonlinear way. As we shall see, the schemes are capable of resolving some rarefactions. The type of solution that the approximate Riemann solvers have difficulties resolving are problems involving sonic rarefactions.

As an illustration, consider Burgers' equation $v_t + F_x = 0$ where $F = v^2/2$. We use (9.9.53) as our definition of $\hat{A}$, which in the scalar case we will denote by $\hat{a} = \hat{a}(u_k^n, u_{k+1}^n)$. Hence, we set $\hat{a} = (u_k^n + u_{k+1}^n)/2$ and solve the Riemann problem

$$\hat{U}_t + \hat{a}\hat{U}_x = 0, \quad x \in \mathbb{R}, \ t > t_n \tag{9.9.67}$$

$$\hat{U}(x, t_n) = \begin{cases} u_k^n & \text{for } x \leq x_{k+1/2} \\ u_{k+1}^n & \text{for } x > x_{k+1/2}. \end{cases} \tag{9.9.68}$$

We should understand that in the context of Section 9.9.4, $\hat{a}$ is a 1×1 matrix, $K = 1$, the eigenvalue of $\hat{a}$ is $\nu_{1_k} = \hat{a}$, and the associated eigenvector is $\mathbf{r}_1 = 1$. We expand $\delta_+ u_k^n$ in terms of the eigenvector of $\hat{a}$ and get $\alpha_{1_k} = \delta_+ u_k^n$. The numerical flux function corresponding to numerical flux function (9.9.49) can be written as

$$h_{k+1/2}^n = \begin{cases} F_k^n + \alpha_{1_k}\nu_{1_k} & \text{if } \nu_{1_k} = \hat{a} < 0 \\ F_k^n & \text{if } \nu_{1_k} = \hat{a} \geq 0. \end{cases} \tag{9.9.69}$$

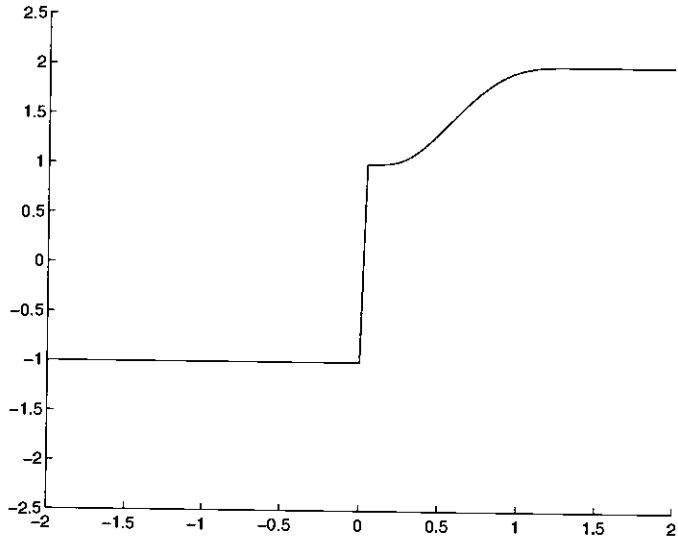

FIGURE 9.9.1. An approximation to the solution of initial–boundary–value problem (9.9.70)–(9.9.72) obtained using the difference scheme associated with numerical flux function (9.9.69).

It is easy to see, HW9.9.6, that the difference scheme associated with numerical flux function (9.9.69) gives the correct solutions for shocks and fans that are not sonic rarefactions. In Figure 9.9.1 we plot the solution obtained by applying the difference scheme associated with numerical flux function (9.9.69) to the initial–boundary–value problem (with the appropriate numerical boundary conditions)

$$v_t + \left(v^2/2\right)_x = 0, \quad x \in (-2, 2), \ t > 0 \tag{9.9.70}$$

$$v(x,0) = \begin{cases} -1 & \text{if } x \leq 0 \\ 2 & \text{if } x > 0 \end{cases} \tag{9.9.71}$$

$$u_0^n = -1.0, \quad u_M^n = 2.0. \tag{9.9.72}$$

Obviously, the difference scheme associated with numerical flux function (9.9.69) cannot produce the sonic rarefaction. We do note, however, that

the solution given in Figure 9.9.1 is a weak solution—it had better be, considering the fact that the difference scheme is conservative.

One of the approaches used to fix numerical flux function (9.9.69) so that it will resolve the sonic rarefaction is due to Harten and Hyman, ref. [24]. The difficulty with the solution provided by numerical flux function (9.9.69) is the fact that the solution to Riemann problem (9.9.67)–(9.9.68) when $u_k^n < u_{k+1}^n$ is given by

$$\hat{U}(x,t) = \begin{cases} u_k^n & \text{when } (x - x_{k+1/2})/(t - t_n) \le s \\ u_{k+1}^n & \text{when } (x - x_{k+1/2})/(t - t_n) > s \end{cases} \tag{9.9.73}$$

where $s = (u_k^n + u_{k+1}^n)/2$, i.e., no fan. We know that solutions to linear Riemann problems do not produce fans. The Harten-Hyman fix to the approximate Riemann solution scheme is as follows. When $u_k^n < 0 < u_{k+1}^n$, replace solution (9.9.73) by

$$\hat{U}(x,t) = \begin{cases} u_k^n & \text{when } (x - x_{k+1/2})/(t - t_n) < F'(u_k^n) \\ u_m & \text{when} \\ & F'(u_k^n) \le (x - x_{k+1/2})/(t - t_n) \le F'(u_{k+1}^n) \\ u_{k+1}^n & \text{when } (x - x_{k+1/2})/(t - t_n) > F'(u_{k+1}^n) \end{cases} \tag{9.9.74}$$

where $u_m = (u_k^n + u_{k+1}^n)/2$. It should be noted that $F'(u_k^n) = u_k^n$, $F'(u_{k+1}^n) = u_{k+1}^n$ and the region on which u_m is defined is chosen as the region on which the fan should form, i.e., $F'(u_k^n) = u_k^n$ and $F'(u_{k+1}^n) = u_{k+1}^n$ represent the eigenvalue of the 1×1 matrix $F'(v)$ evaluated at u_k^n and u_{k+1}^n, respectively, so $(x - x_{k+1/2}) = F'(u_k^n)(t - t_n)$ and $(x - x_{k+1/2}) = F'(u_{k+1}^n)(t - t_n)$ represent the characteristics that define the fan. It should be reasonably clear that a solution such as (9.9.74) will try to break up the jump in the solution given in Figure 9.9.1. In the case where $u_k^n < 0 < u_{k+1}^n$, solution (9.9.74) is used to define the numerical flux function $h_{k+1/2}^n$ via (9.9.38). We then have

$$h_{k+1/2}^n = F_k^n + \alpha_{1_k} \frac{1}{2} u_k^n \text{ when } u_k^n < 0 < u_{k+1}^n$$

otherwise (9.9.75)

$$h_{k+1/2}^n = \begin{cases} F_k^n + \alpha_{1_k} \nu_{1_k} & \text{when } s = (u_k^n + u_{k+1}^n)/2 < 0 \\ F_k^n & \text{when } s = (u_k^n + u_{k+1}^n)/2 \ge 0. \end{cases}$$

The approximate solution to initial–boundary–value problem (9.9.70)–(9.9.72) found by using the difference scheme associated with numerical flux function (9.9.75) is given in Figure 9.9.2. We see that even with the slight "dog leg" as the solution crosses the axis, the solution obtained is a good approximation to the correct fan solution.

The choice of $u_m = (u_k^n + u_{k+1}^n)/2$ is not at all arbitrary. Approximate solution (9.9.74) must satisfy condition (9.9.31) of Proposition 9.9.1, the condition that will ensure that the difference scheme obtained by using the

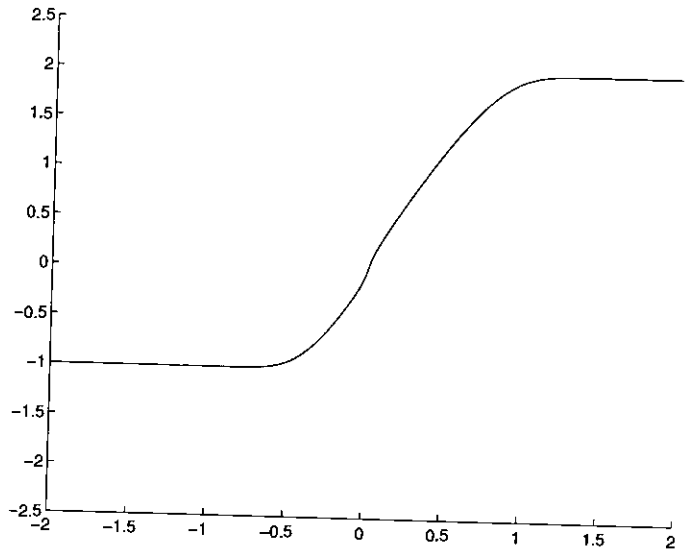

FIGURE 9.9.2. An approximation to the solution of initial–boundary–value problem (9.9.70)–(9.9.72) obtained by using the difference scheme associated with numerical flux function (9.9.75).

approximate Riemann solution to define u_k^{n+1} will be consistent with the original conservation law. We must satisfy

$$\int_{x_k}^{x_{k+1}} \hat{U}(x, t_{n+1})\, dx = \frac{\Delta x}{2}(u_k^n + u_{k+1}^n) - \Delta t(F_k^n + F_{k+1}^n). \quad (9.9.76)$$

Since

$$\int_{x_k}^{x_{k+1}} \hat{U}(x, t_{n+1})dx = \int_{x_k}^{x_{k+1/2}+F'(u_k^n)\Delta t} u_k^n\, dx + \int_{x_{k+1/2}+F'(u_k^n)\Delta t}^{x_{k+1/2}+F'(u_{k+1}^n)\Delta t} u_m\, dx + \int_{x_{k+1/2}+F'(u_{k+1}^n)\Delta t}^{x_{k+1}} u_{k+1}^n\, dx,$$

where $F'(u_k^n) = u_k^n$ and $F'(u_{k+1}^n) = u_{k+1}^n$, we can expand equation (9.9.76) and solve for u_m to get

$$u_m = \frac{1}{2}(u_k^n + u_{k+1}^n).$$

We extend this sonic rarefaction fix to systems of equations in much the same way that we derived the fix for the scalar equations. The situation is that we are trying to approximate solutions to conservation law (9.9.10). We are using an approximate Riemann solver, so we must solve a Riemann problem of the form (9.9.44), (9.9.11) at each interface point $x_{k+1/2}$. When

we derived numerical flux function (9.9.49), we used (9.9.47) as the solution to Riemann problem (9.9.44), (9.9.11) at $t = t_{n+1}$ (which it is). Returning to Section 9.3.1, we recall that the solution of Riemann problem (9.9.44), (9.9.11) is associated with the characteristics as plotted in Figure 9.9.3. We recall also that the solution of Riemann problem (9.9.44), (9.9.11) consists of a series of $K-1$ states, separated by discontinuities, connecting $\mathbf{u}_k^n$ to $\mathbf{u}_{k+1}^n$. We remember that we are using Riemann problem (9.9.44), (9.9.11) to obtain an approximation to the solution of conservation law (9.9.10). Let $\lambda_j = \lambda_j(\mathbf{v})$, $j = 1, \ldots, K$ denote the eigenvalues of the matrix $\mathbf{F}'(\mathbf{v})$. The difficulty arises when for some p, $\lambda_{p_L} = \lambda_p(\mathbf{u}_k^n)$ and $\lambda_{p_R} = \lambda_p(\mathbf{u}_{k+1}^n)$ are such that $\lambda_{p_L} < 0 < \lambda_{p_R}$. Obviously, the states connecting $\mathbf{v}_{p-1}$ to $\mathbf{v}_p$ in the nonlinear problem should be connected by a fan in this case. The analogous discontinuity across ν_p will be connected by a discontinuity (not a fan). And since this is the sonic case, the difference schemes will have problems breaking the jumps into a fan.

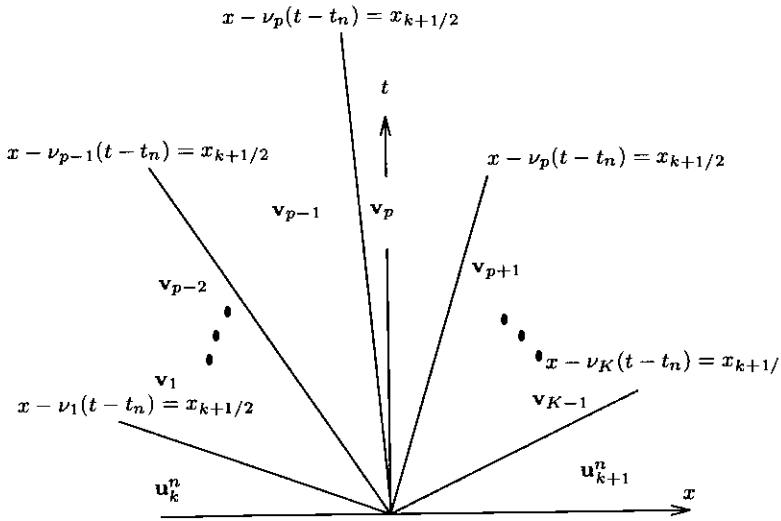

FIGURE 9.9.3. Plot showing the states connecting $\mathbf{u}_k^n$ and $\mathbf{u}_{k+1}^n$ across the characteristics of the matrix $\hat{A}$ for the solution to Riemann problem (9.9.44), (9.9.11).

We let p be the mode for which $\lambda_{p_L} < 0 < \lambda_{p_R}$. The usual solution to the linear Riemann problem (9.9.44), (9.9.11) as given in Section 9.3.1 is to let $\mathbf{r}_{j_k}$, $j = 1, \ldots, K$ denote the eigenvectors of $\hat{A}$, let α_{j_k}, $j = 1, \ldots, K$ be such that $\delta_+ \mathbf{u}_k^n = \sum_{j=1}^K \alpha_{j_k} \mathbf{r}_{j_k}$, and write

$$\mathbf{v}_{p-1} = \mathbf{u}_k^n + \sum_{j=1}^{p-1} \alpha_{j_k} \mathbf{r}_{j_k}$$

to the left of the characteristic $\ell_1 : \; x - \nu_p(t - t_n) = x_{k+1/2}$ and

$$\mathbf{v}_p = \mathbf{u}_k^n + \sum_{j=1}^{p-1} \alpha_{j_k} \mathbf{r}_{j_k} + \alpha_{p_k} \mathbf{r}_{p_k}$$

after the point (x, t) crosses the characteristic ℓ_1.

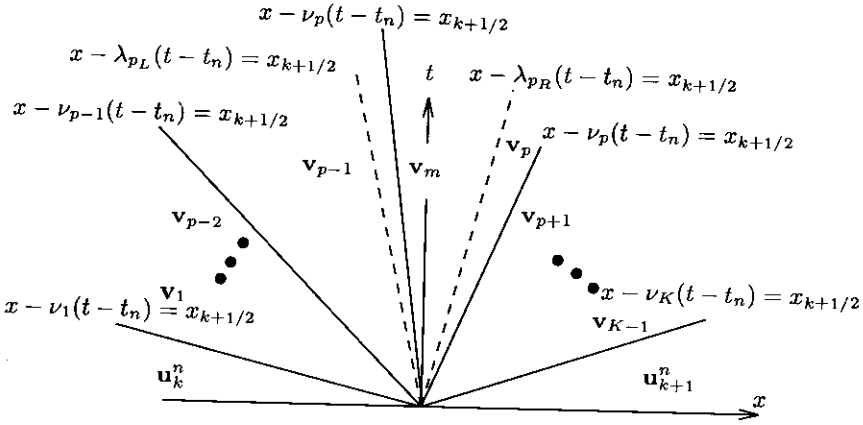

FIGURE 9.9.4. Plot illustrating the approximate solution to Riemann problem (9.9.44), (9.9.11) that we use to define the numerical flux function associated with the sonic rarefaction fix. The approximate solution includes the states connecting $\mathbf{u}_k^n$ and $\mathbf{u}_{k+1}^n$ across the characteristics of the matrix $\hat{A}$ as well as the region $\{(x,t) : \; \lambda_{p_L} \le (x - x_{k+1/2})/(t - t_n) \le \lambda_{p_R}\}$ where we define $\hat{\mathbf{U}}$ to be $\mathbf{v}_m$.

The sonic rarefaction fix in this case is to include two additional "extra characteristic-like curves" $\ell_1' : \; x - \lambda_{p_L}(t - t_n) = x_{k+1/2}$ and $\ell_1'' : \; x - \lambda_{p_R}(t - t_n) = x_{k+1/2}$ and define a new state $\mathbf{v}_m$ between ℓ_1' and ℓ_1''. See Figure 9.9.4. Hence, instead of using the exact solution (9.9.47) to Riemann problem (9.9.44), (9.9.11) to define the numerical flux function (using equation (9.9.38)), we will use

$$\hat{\mathbf{U}}^*(x,t) = \begin{cases} \mathbf{v}_m & \text{when } \lambda_{p_L} \le (x - x_{k+1/2})/(t - t_n) \le \lambda_{p_R} \\ \hat{\mathbf{U}}(x, t_{n+1}) & \text{otherwise,} \end{cases} \tag{9.9.77}$$

where $\hat{\mathbf{U}}(x, t_{n+1})$ is given (9.9.47) and $\mathbf{v}_m$ is yet to be determined.

It might seem a bit odd that we can just change the approximate solution. However, if we return to Proposition 9.9.1, we see that if the approximate solution satisfies equation (9.9.31), then the difference scheme associated with the numerical flux function $\mathbf{h}_{k+1/2}^n$ will be consistent with the given conservation law. We notice that $\hat{\mathbf{U}}^*$ and $\hat{\mathbf{U}}$ are the same except between ℓ_1'

and ℓ_1'' and that $\hat{\mathbf{U}}$ satisfies equation (9.9.31). Then $\hat{\mathbf{U}}$ will satisfy equation (9.9.31) if

$$\theta = \int_{x_{k+1/2}+\lambda_{p_R}\Delta t}^{x_{k+1/2}+\lambda_{p_L}\Delta t} \mathbf{v}_m\,dx - \int_{x_{k+1/2}+\lambda_{p_L}\Delta t}^{x_{k+1/2}+\nu_{p_k}\Delta t} \mathbf{v}_{p-1}\,dx - \int_{x_{k+1/2}+\nu_{p_k}\Delta t}^{x_{k+1/2}+\lambda_{p_R}\Delta t} \mathbf{v}_p\,dx,$$

or

$$\theta = (\lambda_{p_R} - \lambda_{p_L})\Delta t\mathbf{v}_m - (\nu_{p_k} - \lambda_{p_L})\Delta t\mathbf{v}_{p-1} - (\lambda_{p_R} - \nu_{p_k})\Delta t\mathbf{v}_p.$$

Hence, we define $\mathbf{v}_m$ to be

$$\mathbf{v}_m = \frac{\nu_{p_k} - \lambda_{p_L}}{\lambda_{p_R} - \lambda_{p_L}}\mathbf{v}_{p-1} + \frac{\lambda_{p_R} - \nu_{p_k}}{\lambda_{p_R} - \lambda_{p_L}}\mathbf{v}_p. \tag{9.9.78}$$

Now that we have defined a new approximate Riemann solution that we hope might work, we must use $\hat{\mathbf{U}}^*$ along with (9.9.38) to define a numerical flux function. We note that when this is an eigenvalue of $\mathbf{F}'$ that satisfies $\lambda_{p_L} < 0 < \lambda_{p_R}$,

$$\begin{aligned}
&\int_{x_k}^{x_{k+1/2}} \hat{\mathbf{U}}^*(x, t_{n+1})\,dx \\
&\quad = \int_{x_k}^{x_{k+1/2}+\lambda_{p_L}\Delta t} \hat{\mathbf{U}}(x, t_{n+1})\,dx + \int_{x_{k+1/2}+\lambda_{p_L}\Delta t}^{x_{k+1/2}} \mathbf{v}_m\,dx \\
&\quad = \Big(\frac{\Delta x}{2} + \lambda_{p_L}\Delta t\Big)\mathbf{u}_k^n + \Delta t\sum_{j=1}^{p-1}\alpha_{j_k}(\lambda_{p_L} - \nu_{j_k})\mathbf{r}_{j_k} - \lambda_{p_L}\Delta t\mathbf{v}_m \\
&\quad = \frac{\Delta x}{2}\mathbf{u}_k^n + \lambda_{p_L}\Delta t\Big(\mathbf{u}_k^n + \sum_{j=1}^{p-1}\alpha_{j_k}\mathbf{r}_{j_k}\Big) - \Delta t\sum_{j=1}^{p-1}\alpha_{j_k}\nu_{j_k}\mathbf{r}_{j_k} \\
&\qquad - \Delta t\lambda_{p_L}\frac{1}{\lambda_{p_R} - \lambda_{p_L}}\big[(\nu_{p_k} - \lambda_{p_L})\mathbf{v}_{p-1} + (\lambda_{p_R} - \nu_{p_k})\mathbf{v}_p\big].
\end{aligned} \tag{9.9.79}$$

If we recall from Section 9.3.1 that

$$\mathbf{v}_{p-1} = \mathbf{u}_k^n + \sum_{j=1}^{p-1}\alpha_{j_k}\mathbf{r}_{j_k},$$

and

$$\mathbf{v}_p = \mathbf{u}_k^n + \sum_{j=1}^{p}\alpha_{j_k}\mathbf{r}_{j_k},$$

we see that $\mathbf{v}_p = \mathbf{v}_{p-1} + \alpha_{p_k}\mathbf{r}_{p_k}$, the last term of (9.9.79) can be written as

$$\begin{aligned}
&- \Delta t\lambda_{p_L}\frac{1}{\lambda_{p_R} - \lambda_{p_L}}\big[(\nu_{p_k} - \lambda_{p_L})\mathbf{v}_{p-1} + (\lambda_{p_R} - \nu_{p_k})\mathbf{v}_p\big] \\
&\quad = -\Delta t\lambda_{p_L}\frac{1}{\lambda_{p_R} - \lambda_{p_L}}\big[(\lambda_{p_R} - \lambda_{p_L})\mathbf{v}_{p-1} + (\lambda_{p_R} - \nu_{p_k})\alpha_{p_k}\mathbf{r}_{p_k}\big] \\
&\quad = -\Delta t\lambda_{p_L}\mathbf{v}_{p-1} - \Delta t\lambda_{p_L}\frac{\lambda_{p_R} - \nu_{p_k}}{\lambda_{p_R} - \lambda_{p_L}}\alpha_{p_k}\mathbf{r}_{p_k},
\end{aligned}$$

and (9.9.79) becomes

$$\begin{aligned}\frac{\Delta x}{2}\mathbf{u}_k^n + \lambda_{p_L}\Delta t\mathbf{v}_{p-1} - \Delta t\sum_{j=1}^{p-1}\alpha_{j_k}\nu_{j_k}\mathbf{r}_{j_k} - \Delta t\lambda_{p_L}\mathbf{v}_{p-1} \\ - \Delta t\lambda_{p_L}\frac{\lambda_{p_R}-\nu_{p_k}}{\lambda_{p_R}-\lambda_{p_L}}\alpha_{p_k}\mathbf{r}_{p_k} \\ = \frac{\Delta x}{2}\mathbf{u}_k^n - \Delta t\sum_{j=1}^{p-1}\alpha_{j_k}\nu_{j_k}\mathbf{r}_{j_k} - \Delta t\lambda_{p_L}\frac{\lambda_{p_R}-\nu_{p_k}}{\lambda_{p_R}-\lambda_{p_L}}\alpha_{p_k}\mathbf{r}_{p_k}.\end{aligned}$$

Using (9.9.38), we get

$$\mathbf{h}_{k+1/2}^n = \mathbf{F}_k^n + \sum_{j=1}^{p-1}\alpha_{j_k}\nu_{j_k}\mathbf{r}_{j_k} + \lambda_{p_L}\frac{\lambda_{p_R}-\nu_{p_k}}{\lambda_{p_R}-\lambda_{p_L}}\alpha_{p_k}\mathbf{r}_{p_k}. \tag{9.9.80}$$

Hence, the "fixed" numerical flux function becomes

$$\mathbf{h}_{k+1/2}^n = \begin{cases} \mathbf{F}_k^n + \displaystyle\sum_{j=1}^{p-1}\alpha_{j_k}\nu_{j_k}\mathbf{r}_{j_k} + \lambda_{p_L}\frac{\lambda_{p_R}-\nu_{p_k}}{\lambda_{p_R}-\lambda_{p_L}}\alpha_{p_k}\mathbf{r}_{p_k} \\ \qquad\qquad \text{when } \lambda_{p_L} < 0 < \lambda_{p_R} \quad \text{for some } p \\ \text{otherwise} \\ \mathbf{F}_k^n + \displaystyle\sum_{\nu_{j_k}<0}\alpha_{j_k}\nu_{j_k}\mathbf{r}_{j_k}. \end{cases} \tag{9.9.81}$$

HW 9.9.6 Use the difference scheme associated with numerical flux function (9.9.69) to solve the initial–boundary–value problem with Burgers' equation

$$v_t + \left(v^2/2\right)_x = 0 \quad x \in (-2,2),\ t > 0$$

along with the following sets of initial, boundary conditions and numerical boundary conditions. See Remark, page 142.

(a) $v(x,0) = \begin{cases} 2 & \text{if } x \le 0 \\ 1 & \text{if } x > 0 \end{cases}$ $\qquad v(-2,t) = 2$ and $u_M^n = 2.0$

(b) $v(x,0) = \begin{cases} -1 & \text{if } x \le 0 \\ -2 & \text{if } x > 0 \end{cases}$ $\qquad u_0^n = -1.0$ and $v(2,t) = -2$

(c) $$v(x,0) = \begin{cases} 2 & \text{if } x \le 0 \\ -2 & \text{if } x > 0 \end{cases} \qquad v(-2,t) = 2 \text{ and } v(2,t) = -2$$

(d) $$v(x,0) = \begin{cases} 1 & \text{if } x \le 0 \\ 2 & \text{if } x > 0 \end{cases} \qquad v(-2,t) = 1 \text{ and } u_M^n = 2.0$$

(e) $$v(x,0) = \begin{cases} -2 & \text{if } x \le 0 \\ -1 & \text{if } x > 0 \end{cases} \qquad u_0^n = -2.0 \text{ and } v(2,t) = -1$$

Use $\Delta x = 0.01$, $\Delta t = 0.005$ and plot the solutions at times $t = 0.05$ and $t = 0.5$.

HW 9.9.7 Use the "fixed" solution (9.9.74) in equation (9.9.38) to derive the numerical flux function given in (9.9.75).

HW 9.9.8 Show that the numerical viscosity coefficient associated with numerical flux function (9.9.75) is given by

$$Q_{k+1/2}^n = \begin{cases} \frac{R}{2}\delta_+ u_k^n & \text{when } u_k^n < 0 < u_{k+1}^n \\ |Ra_{k+1/2}^n| & \text{otherwise.} \end{cases}$$

HW 9.9.9 Solve the shock tube problem HW0.0.3 using numerical flux function (9.9.81) and $\hat{A}$ defined by the Roe linearization (9.9.64)–(9.9.66). Compare and contrast your solutions with those found in HW9.9.3.

9.10 High Resolution Schemes for Linear K-System Conservation Laws

As we saw when we developed high resolution schemes for scalar conservation laws, both the description and development of the schemes are very difficult. When we developed the high resolution schemes for scalar equations, we often first developed the scheme for the linear scalar equations. For the numerical treatment of K-system conservation laws, we also first treat the linear K-system conservation laws but for another reason. Our high resolution schemes for general K-systems will generally be developed by using a high resolution scheme on an appropriate approximate Riemann problem. For that reason, the results we obtain for linear K-system conservation laws will be very important to our treatment of general K-system conservation laws.

As we have done so often before, we consider a linear conservation law in the form

$$\mathbf{v}_t + A\mathbf{v}_x = \boldsymbol{\theta}, \tag{9.10.1}$$

denote the eigenvalues and eigenvectors of A by ν_j, $j = 1, \ldots, K$, and $\mathbf{r}_j$, $j = 1, \ldots, K$, and let the matrices S and S^{-1} be such that $SAS^{-1} = D$ is a diagonal matrix with the eigenvalues of A on the diagonal. We let $\mathbf{V} = S\mathbf{v}$ and multiply equation (9.10.1) on the left by S to uncouple system (9.10.1) into

$$V_{j_t} + \nu_j V_{j_x} = 0, \quad j = 1, \ldots, K. \tag{9.10.2}$$

We next begin to develop high resolution schemes for linear K-systems much as we have done in the past with most of our work with hyperbolic systems. We will use our scalar, linear results for each of the one way wave equations (9.10.2) and then try to recouple them to give results for our linear K-system.

9.10.1 *Flux-Limiter Schemes for Linear K-System Conservation Laws*

We begin by developing a flux-limiter scheme for each of the equations given in (9.10.2) and use these results to obtain a flux-limiter scheme for equation (9.10.1). If we proceed as in Section 9.7.5.1, we use the upwind scheme (9.9.27) as our low order scheme, i.e., the numerical flux function

$$h^n_{L_{j_{k+1/2}}} = \frac{1}{2}\nu_j\left(U^n_{j_k} + U^n_{j_{k+1}}\right) - \frac{1}{2}|\nu_j|\delta_+ U^n_{j_k}; \tag{9.10.3}$$

the Lax-Wendroff scheme as our high order scheme, i.e., the numerical flux function

$$h^n_{H_{j_{k+1/2}}} = \nu_j U^n_{j_k} + \frac{1}{2}\nu_j(1 - \nu_j R)\delta_+ U^n_{j_k}; \tag{9.10.4}$$

and define the numerical flux function for the flux-limiter scheme to be

$$h^n_{j_{k+1/2}} = h^n_{L_{j_{k+1/2}}} + \phi^n_{j_k}\left(h^n_{H_{j_{k+1/2}}} - h^n_{L_{j_{k+1/2}}}\right).$$

We obtain the numerical flux function

$$h^n_{j_{k+1/2}} = \frac{\nu_j}{2}\left(U^n_{j_k} + U^n_{j_{k+1}}\right) - \frac{1}{2}|\nu_j|\delta_+ U^n_{j_k} + \frac{1}{2}\nu_j\phi^n_{j_k}[\text{sign}(\nu_j) - \nu_j R]\delta_+ U^n_{j_k} \tag{9.10.5}$$

and difference scheme

$$U^{n+1}_{j_k} = U^n_{j_k} - R\left[h^n_{j_{k+1/2}} - h^n_{j_{k-1/2}}\right]. \tag{9.10.6}$$

As in the scalar case, $\phi^n_{j_k} = \phi(\theta^n_{j_k})$, where

$$\theta^n_{j_k} = \begin{cases} \frac{\delta_- U^n_{j_k}}{\delta_+ U^n_{j_k}} & \text{when } \nu_j < 0 \\ \frac{\delta_+ U^n_{j_k+1}}{\delta_+ U^n_{j_k}} & \text{when } \nu_j > 0. \end{cases} \tag{9.10.7}$$

The subscript j is included with the θ notation to emphasize the fact that $\theta^n_{j_k}$ depends on j also.

Because of the $\phi^n_{j_k}$ term, we are not able to combine the terms in (9.10.5) into a vector equation, multiply by S^{-1}, and obtain a nice vector difference scheme in terms of $\mathbf{u}^n_k$, A, etc. We can obviously write $\mathbf{h}^n_{L_{k+1/2}}$ and $\mathbf{h}^n_{H_{k+1/2}} - \mathbf{h}^n_{L_{k+1/2}}$ as

$$\mathbf{h}^n_{L_{k+1/2}} = \frac{1}{2}A(\mathbf{u}^n_k + \mathbf{u}^n_{k+1}) - \frac{1}{2}|A|\delta_+\mathbf{u}^n_k$$

and

$$\mathbf{h}^n_{H_{k+1/2}} - \mathbf{h}^n_{L_{k+1/2}} = \frac{1}{2}(|A| - RA^2)\delta_+\mathbf{u}^n_k.$$

If we expand $\delta_+\mathbf{u}^n_k$ as

$$\delta_+\mathbf{u}^n_k = \sum_{j=1}^{K} \alpha_{j_k}\mathbf{r}_j, \tag{9.10.8}$$

we see that

$$S\delta_+\mathbf{u}^n_k = \delta_+\mathbf{U}^n_k = \sum_{j=1}^{K} \alpha_{j_k} S\mathbf{r}_j = \sum_{j=1}^{K} \alpha_{j_k}\mathbf{u}_j.$$

Thus, $\delta_+ U^n_{j_k} = \alpha_{j_k}$, $j = 1, \ldots, K$. We have used the fact here that $S\mathbf{r}_j = \mathbf{u}_j$, $j = 1, \ldots, K$. This follows because the construction of S^{-1} puts $\mathbf{r}_j$ in the j-th column of S^{-1}, which implies that $S^{-1}\mathbf{u}_j = \mathbf{r}_j$. This allows us to write $\theta^n_{j_k}$ as

$$\theta^n_{j_k} = \begin{cases} \frac{\alpha_{j_k-1}}{\alpha_{j_k}} & \text{when } \nu_j < 0 \\ \frac{\alpha_{j_k+1}}{\alpha_{j_k}} & \text{when } \nu_j > 0. \end{cases} \tag{9.10.9}$$

and the numerical flux function (9.10.5) as

$$h^n_{j_{k+1/2}} = \frac{\nu_j}{2}(U^n_{j_k} + U^n_{j_{k+1}}) - \frac{1}{2}|\nu_j|\delta_+ U^n_{j_k} + \frac{1}{2}\nu_j\phi^n_{j_k}[\operatorname{sign}(\nu_j) - \nu_j R]\alpha_{j_k}. \tag{9.10.10}$$

We next notice that we can write $h^n_{j_{k+1/2}}$, $j = 1, \ldots, K$, as a vector function as

$$\mathbf{H}^n_{k+1/2} = \frac{1}{2}D(\mathbf{U}^n_k + \mathbf{U}^n_{k+1}) - \frac{1}{2}|D|\delta_+\mathbf{U}^n_k + \frac{1}{2}\sum_{j=1}^{K} \nu_j\phi^n_{j_k}[\operatorname{sign}(\nu_j) - \nu_j R]\alpha_{j_k}\mathbf{u}_j. \tag{9.10.11}$$

The first two terms are assembled as we have transformed scalar results to vector results in the past. The last term is found by multiplying the last term in (9.10.10) by the appropriate unit vector $\mathbf{u}_j$ and summing. This procedure has the j-th term in the sum in equation (9.10.11) appearing in only the j-th component of the resulting vector equation. If we then multiply equation (9.10.11) by S^{-1}, we get

$$\mathbf{h}^n_{k+1/2} = \frac{1}{2}A(\mathbf{u}^n_k + \mathbf{u}^n_{k+1}) - \frac{1}{2}|A|\delta_+\mathbf{u}^n_k + \frac{1}{2}\sum_{j=1}^{K}\nu_j\phi^n_{j_k}\left[\operatorname{sign}(\nu_j) - \nu_j R\right]\alpha_{j_k}\mathbf{r}_j. \tag{9.10.12}$$

Numerical flux function (9.10.12) is not in the usual form of our numerical flux functions. The fact that we must compute $|A|$ and have the eigenvalues and eigenvectors of A is not a problem. Since A is a constant matrix here, $|A|$ and ν_j, $j = 1, \ldots, K$, must be computed only once. The difficulty with applying the difference scheme associated with numerical flux function (9.10.12) is that for each time step and each grid point, $\delta_+\mathbf{u}^n_k$ must be expanded in terms of $\mathbf{r}_j$, $j = 1, \ldots, K$, the eigenvectors of A. This expansion is done by solving the $K \times K$ system of equations

$$\mathcal{R}\begin{bmatrix} \alpha_{1_k} \\ \vdots \\ \alpha_{K_k} \end{bmatrix} = \delta_+\mathbf{u}^n_k,$$

where $\mathcal{R}$ is the matrix $\mathcal{R} = \begin{bmatrix}\mathbf{r}_1 & \cdots & \mathbf{r}_K\end{bmatrix}$. When we apply the scheme to a linear problem with nonconstant coefficients, the matrix elements may depend on both space and time, or when we apply the scheme to the linearization of a nonlinear problem (like an approximate Riemann problem), the matrix will depend on $\mathbf{u}^n_k$ and $\mathbf{u}^n_{k+1}$. In each of these cases, the eigenvalues, eigenvectors, and the expansion of $\delta_+\mathbf{u}^n_k$ in terms of the eigenvectors must be computed at each time step and at each grid point. We must understand that to obtain better solutions than we have been able to obtain earlier, we must do more work. Also, we should understand that K is generally not large. The amount of work we are discussing here is not huge. In fact, as we shall see in Section 9.9.4.1 when we consider the Euler equations of gas dynamics, when K is not large, the calculation of eigenvalues, eigenvectors, and coefficients in the expansion of $\delta_+u^n_k$ can be done analytically. Having formulas for these terms greatly reduces the necessary work.

9.10.2 Slope-Limiter Schemes for Linear K-System Conservation Laws

We now want to proceed in a similar way to develop a slope-limiter scheme for equation (9.10.1). Again, we consider each of the equations given in

(9.10.2) and use equation (9.7.120) to write a numerical flux function for a slope-limiter scheme for each one-way wave equation. We get

$$h^n_{j_{k+1/2}} = \frac{\nu_j}{2}\left(U^n_{j_k} + U^n_{j_{k+1}}\right) - \frac{|\nu_j|}{2}\delta_+ U^n_{j_k} + \frac{\nu_j}{2}\left(\operatorname{sign}(\nu_j) - \nu_j R\right)\Delta x \sigma^n_{j_{k+\ell_j}} \tag{9.10.13}$$

where $\ell_j = 0$ when $\nu_j > 0$ and $\ell_j = 1$ when $\nu_j < 0$. As it was in the case with flux-limiter schemes, it is difficult to recouple this system because of the $\sigma^n_{j_k}$ term. We can write the numerical flux functions $h^n_{j_{k+1/2}}$, $j = 1, \ldots, K$, as a vector as

$$\begin{aligned}\mathbf{H}^n_{k+1/2} = {} & \frac{1}{2}D\left(\mathbf{U}^n_k + \mathbf{U}^n_{k+1}\right) - \frac{1}{2}|D|\delta_+ \mathbf{U}^n_k \\ & + \frac{1}{2}\sum_{j=1}^{K} \nu_j\left(\operatorname{sign}(\nu_j) - \nu_j R\right)\Delta x \sigma^n_{j_{k+\ell}}\mathbf{u}_j. \end{aligned} \tag{9.10.14}$$

Multiplying on the left by S^{-1} yields

$$\begin{aligned}\mathbf{h}^n_{k+1/2} = {} & \frac{1}{2}A\left(\mathbf{u}^n_k + \mathbf{u}^n_{k+1}\right) - \frac{1}{2}|A|\delta_+ \mathbf{u}^n_k \\ & + \frac{1}{2}\sum_{j=1}^{K} \nu_j\left(\operatorname{sign}(\nu_j) - \nu_j R\right)\Delta x \sigma^n_{j_{k+\ell}}\mathbf{r}_j, \end{aligned} \tag{9.10.15}$$

$$= \mathbf{h}^n_{L_{k+1/2}} + \frac{1}{2}\sum_{j=1}^{K} \nu_j\left(sgn(\nu_j) - \nu_j R\right)\Delta x \sigma^n_{j_{k+\ell}}\mathbf{r}_j \tag{9.10.16}$$

where $\mathbf{h}^n_{L_{k+1/2}}$ is the numerical flux function for the upwind scheme (9.9.27). Returning to Section 9.7.6, equation (9.7.122), we see that $\sigma^n_{j_k}$ is given by

$$\sigma^n_{j_k} = \frac{1}{\Delta x}\operatorname{minmod}\{\delta_+ U^n_{j_k}, \delta_- U^n_{j_k}\}.$$

If we write $\delta_+ \mathbf{u}^n_k$ as

$$\delta_+ \mathbf{u}^n_k = \sum_{j=1}^{K} \alpha_{j_k}\mathbf{r}_j,$$

then $\delta_+ \mathbf{U}^n_k = \sum_{j=1}^{K} \alpha_{j_k}\mathbf{u}_j$ or $\delta_+ U^n_{j_k} = \alpha_{j_k}$, i.e.,

$$\sigma^n_{j_k} = \frac{1}{\Delta x}\operatorname{minmod}\{\alpha_{j_k}, \alpha_{j_{k-1}}\}. \tag{9.10.17}$$

9.10.3 A Modified Flux Scheme for Linear K-System Conservation Laws

As we have done in each of the last two sections, we again consider conservation law (9.10.1) and work on the uncoupled system of equations (9.10.2).

For a fixed j we can apply equation (9.7.146) to conservation (9.10.2) and get the following numerical flux function

$$h^n_{j_{k+1/2}} = \frac{1}{2}\left[\nu_j U^n_{j_k} + \nu_j U^n_{j_{k+1}}\right] + \frac{1}{2R}\left[g_{j_{k+1}} + g_{j_k} - Q^{L_j}\left(R\nu_j + \delta_+ g_{j_k}/\delta_+ U^n_{j_k}\right)\delta_+ U^n_{j_k}\right]. \quad (9.10.18)$$

As in Section 9.7.7, Q^{L_j} is the coefficient of numerical viscosity of a conservative three-point TVD scheme. the j subscript on the L signifies that we can use different conservative three-point TVD schemes for different components. If we again expand $\delta_+ \mathbf{u}^n_k$ in terms of the eigenvectors of A as

$$\delta_+ \mathbf{u}^n_k = \sum_{j=1}^{K} \alpha_{j_k} \mathbf{r}_{j_k}$$

and note that $\delta_+ U^n_{j_k} = \alpha_{j_k}$, we can write $\tilde{g}_{j_{k+1/2}}$ as

$$\tilde{g}_{j_{k+1/2}} = \frac{1}{2}\left[Q^{L_j}(R\nu_j) - R^2\nu_j^2\right]\alpha_{j_k}. \quad (9.10.19)$$

If we then use the fact that since the scheme associated with the coefficient of numerical viscosity Q^{L_j} is TVD and for all j, $R|\nu_j| \le 1$,

$$Q^{L_j}(R\nu_j) - R^2\nu_j^2 \ge 0$$

(as we saw in the proof to Proposition 9.7.25), we can write $|\tilde{g}_{j_{k+1/2}}|$ as

$$|\tilde{g}_{j_{k+1/2}}| = \frac{1}{2}\left[Q^{L_j}(R\nu_j) - R^2\nu_j^2\right]|\alpha_{j_k}|.$$

Therefore, g_{j_k} is given by

$$g_{j_k} = \begin{cases} \operatorname{sign}\{\tilde{g}_{j_{k+1/2}}\}\min\{|\tilde{g}_{j_{k-1/2}}|, |\tilde{g}_{j_{k+1/2}}|\} & \text{when } \tilde{g}_{j_{k-1/2}}\tilde{g}_{j_{k+1/2}} \ge 0 \\ 0 & \text{when } \tilde{g}_{j_{k-1/2}}\tilde{g}_{j_{k+1/2}} < 0. \end{cases} \quad (9.10.20)$$

We build the vector numerical flux function much as we have for the last two schemes. We write the numerical flux functions given in (9.10.18) as a vector function by writing the first term in terms of the matrix D and multiply the second term by $\mathbf{u}_j$ and sum over j. When we then multiply on the left by S^{-1}, we get

$$\mathbf{h}^M_{k+1/2} = \frac{1}{2}A\left(\mathbf{u}^n_k + \mathbf{u}^n_{k+1}\right) + \frac{1}{2R}\sum_{j=1}^{K}\left[g_{j_{k+1}} + g_{j_k} - Q^{L_j}\left(R\nu_j + \delta_+ g_{j_k}/\alpha_{j_k}\right)\alpha_{j_k}\right]\mathbf{r}_j. \quad (9.10.21)$$

9.10.4 *High Resolution Schemes for K-System Conservation Laws*

Now that we have extended the ideas of high resolution schemes to linear K-systems, we proceed one step further to obtain high resolution schemes for general K-system conservation laws. In this section we will describe the flux-limiter, the slope-limiter and the modified flux schemes for general K-systems. These schemes are done together because the same approach is used for all. As usual, we are considering conservation law (9.9.10). Our approach is to consider the approximate Riemann problem (9.9.44), (9.9.11) where $\hat{A}$ is chosen so that $\hat{A} = \hat{A}(\mathbf{u}_k^n, \mathbf{u}_{k+1}^n)$ and Roe conditions (1)–(3) are satisfied. For each k, denote the eigenvalues and eigenvectors of $\hat{A}$ by ν_{j_k}, $j = 1, \ldots, K$, and $\mathbf{r}_{j_k}$, $j = 1, \ldots, K$, respectively.

9.10.4.1 Flux-Limiter Scheme

To extend the flux-limiter scheme, we use the obvious extension implied by numerical flux function (9.10.12). The first two terms of numerical flux function (9.10.12) are due to the fact that we used an upwind scheme as our low order scheme. We know that we must be more careful when we use an upwind scheme for nonlinear equations. Thus we let $\mathbf{h}_{L_{k+1/2}}^n$ denote numerical flux function (9.9.81). We should understand that $\mathbf{h}_{L_{k+1/2}}^n$ is the approximate Riemann problem analogue of the upwind scheme and thus fulfills the role of the low order scheme.

We then combine $\mathbf{h}_{L_{k+1/2}}^n$ with numerical flux function (9.10.12) to get the following numerical flux function.

$$\mathbf{h}_{k+1/2}^n = \mathbf{h}_{L_{k+1/2}}^n + \frac{1}{2}\sum_{j=1}^{K} \nu_{j_k}\phi_{j_k}^n \left[\operatorname{sign}(\nu_{j_k}) - \nu_{j_k} R\right]\alpha_{j_k}\mathbf{r}_{j_k}, \quad (9.10.22)$$

where $\phi_{j_k}^n = \phi(\theta_{j_k}^n)$ and

$$\theta_{j_k}^n = \begin{cases} \frac{\alpha_{j_{k-1}}}{\alpha_{j_k}} & \text{when } \nu_{j_k} < 0 \\ \frac{\alpha_{j_{k+1}}}{\alpha_{j_k}} & \text{when } \nu_{j_k} > 0. \end{cases} \quad (9.10.23)$$

We note that the switch from the linear upwind scheme used in numerical flux function (9.10.12) to the nonlinear, fixed upwind numerical flux function (9.9.81) used in (9.10.22) follows from the fact that numerical flux function (9.10.22) is based on an approximate Riemann solution to the nonlinear problem, and as we did after the derivation of numerical flux function (9.9.50), we must take into account the sonic rarefaction fix.

9.10.4.2 Slope-Limiter Scheme

To obtain the slope-limiter scheme for a general K-system, we proceed much as we did in the previous section. We obtain the numerical flux function

$$\mathbf{h}^n_{k+1/2} = \mathbf{h}^n_{L_{k+1/2}} + \frac{1}{2}\sum_{j=1}^{K} \nu_{j_k}\left[\operatorname{sign}(\nu_{j_k}) - \nu_{j_k} R\right]\Delta x \sigma^n_{j_{k+\ell}} \mathbf{r}_{j_k}, \quad (9.10.24)$$

where again the numerical flux function $\mathbf{h}^n_{L_{k+1/2}}$ is given by (9.9.81); ν_{j_k}, $\mathbf{r}_{j_k}$, $j = 1, \ldots, K$, are as above; and

$$\sigma^n_{j_k} = \frac{1}{\Delta x}\operatorname{minmod}\{\alpha_{j_k}, \alpha_{j_{k-1}}\}. \quad (9.10.25)$$

9.10.4.3 Modified Flux Scheme

As we did in Section 9.7.7, we present the modified flux scheme for the general K-system conservation law as developed by Harten in ref. [23]. The modified flux scheme for the general K-system conservation law is obtained by the application of the modified flux scheme for the linear K-system to the approximate Riemann problem (9.9.44), (9.9.11). Again, let ν_{j_k}, $\mathbf{r}_{j_k}$, $j = 1, \ldots, K$, be defined as above and let Q^{L_j} denote a numerical viscosity coefficient associated with a conservative three-point TVD scheme—we again allow different schemes to be used on different components, and now Q^{L_j} will be associated with a nonlinear scheme. We define $\tilde{g}_{j_{k+1/2}}$ and g_{j_k} as in (9.10.19) and (9.10.20) with ν_j replaced by ν_{j_k}. We then obtain the following numerical flux function, analogous to numerical flux function (9.10.21).

$$\mathbf{h}^M_{k+1/2} =$$

$$\frac{1}{2}\left[\mathbf{F}^n_k + \mathbf{F}^n_{k+1}\right] \quad + \frac{1}{2R}\sum_{j=1}^{K}\left[g_{j_k} + g_{j_{k+1}} - Q^{L_j}\left(R\nu_{j_k} + \delta_+ g_{j_k}/\delta_+\alpha_{j_k}\right)\right]\mathbf{r}_{j_k}. \quad (9.10.26)$$

If we return to the derivation of the numerical flux function (9.7.146) associated with the modified flux scheme for scalar equations, we see that we replace $\frac{1}{2}A(\mathbf{u}^n_k + \mathbf{u}^n_{k+1})$ by $\frac{1}{2}(\mathbf{F}^n_k + \mathbf{F}^n_{k+1})$ because we now use a nonlinear low order scheme, and the $\frac{1}{2}(\mathbf{F}^n_k + \mathbf{F}^n_{k+1})$ term is a part of the numerical flux function of a difference scheme in Q-form.

9.11 Implicit Schemes

All of the schemes that we have discussed up until this point in this chapter have been explicit schemes. In Chapters 5 and 6 we also developed implicit schemes for linear hyperbolic partial differential equations. It is also possible to develop implicit schemes that can be used to approximate solutions to problems involving hyperbolic conservation laws. Generally,

implicit schemes are not as popular for use with hyperbolic equations as they are with parabolic equations. However, the advantages gained by using implicit schemes still hold when they are used to solve hyperbolic problems. The enhanced stability properties of implicit schemes are often useful. As we saw in Chapter 7, the Crank-Nicolson scheme for linear hyperbolic equations provides a nondissipative second order scheme. One of the common uses of implicit schemes for hyperbolic problems is to use large time steps to obtain the steady solution efficiently. Before we proceed with our work on implicit schemes, we warn the reader as we have done before: If a time accurate solution is required, do not take advantage of the enhanced stability properties of the implicit schemes by taking a time step that is so large as to eliminate the time accuracy.

In this section we will provide only an introduction to implicit schemes for conservation laws. We began by returning to our discussion on conservative difference schemes from Section 9.6.2 and equation (9.6.5)

$$\Delta x\left(\mathbf{v}_k^{n+1} - \mathbf{v}_k^n\right) + \left[\int_{t_n}^{t_{n+1}} \mathbf{F}(\mathbf{v}(x_{k+1/2}, t))\, dt - \int_{t_n}^{t_{n+1}} \mathbf{F}(\mathbf{v}(x_{k-1/2}, t))\, dt\right] = \boldsymbol{\theta}. \tag{9.11.1}$$

In Section 9.6.2 we approximated the integral terms in equation (9.11.1) by $\Delta t \delta\_ \mathbf{h}_{k+1/2}^n$ and proceeded into a discussion of conservative difference schemes. The superscript n implied that we were approximating the flux at the n-th time step. We always did approximate the fluxes at the n-th time step and got explicit difference schemes.

There is no reason that when we consider approximating equation (9.11.1), we approximate the integral terms at the $(n+1)$-st time step instead of the n-th time step. Or we could approximate the fluxes at both the n-th and the $(n+1)$-st time steps (or at additional earlier time steps). One way to consider the approximation of equation (9.11.1) is to approximate the integrals using the rectangular rule and get

$$\Delta x(\mathbf{v}_k^{n+1} - \mathbf{v}_k^n) + \Delta t\left[\mathbf{F}\left(\mathbf{v}(x_{k+1/2}, t_n)\right) - \mathbf{F}\left(\mathbf{v}(x_{k-1/2}, t_n)\right)\right] + \mathcal{O}(\Delta t^2) \tag{9.11.2}$$

and develop the explicit schemes for conservation laws as an approximation to equation (9.11.2). However, we can just as well use the rectangular rule to approximate the integral terms as

$$\Delta x(\mathbf{v}_k^{n+1} - \mathbf{v}_k^n) + \Delta t\left[\mathbf{F}\left(\mathbf{v}(x_{k+1/2}, t_{n+1})\right) - \mathbf{F}\left(\mathbf{v}(x_{k-1/2}, t_{n+1})\right)\right] + \mathcal{O}(\Delta t^2). \tag{9.11.3}$$

We could then proceed and get the pure implicit schemes. Of course, if we

used a trapezoidal rule approximation of the integral terms, we would get

$$\Delta x\left(\mathbf{v}_k^{n+1}-\mathbf{v}_k^n\right)+\frac{\Delta t}{2}\left[\mathbf{F}\left(\mathbf{v}(x_{k+1/2},t_n)\right)-\mathbf{F}\left(\mathbf{v}(x_{k-1/2},t_n)\right)\right]$$
$$+\frac{\Delta t}{2}\left[\mathbf{F}\left(\mathbf{v}(x_{k+1/2},t_{n+1})\right)-\mathbf{F}\left(\mathbf{v}(x_{k-1/2},t_{n+1})\right)\right]+\mathcal{O}(\Delta t^3) \quad (9.11.4)$$

which would lead to Crank-Nicolson-type schemes. We should realize at this time that we can use any numerical integration scheme, using different combinations of the n-th and $(n+1)$-st time levels or more time levels, and all of these would lead to different difference schemes.

We can include the explicit, pure implicit and Crank-Nicolson-type schemes in one approach by approximating equation (9.11.1) by

$$\mathbf{u}_k^{n+1}=\mathbf{u}_k^n-\frac{R}{2}\left(\delta_-\mathbf{h}_{k+1/2}^n+\delta_-\mathbf{h}_{k+1/2}^{n+1}\right), \quad (9.11.5)$$

or

$$\mathbf{u}_k^{n+1}+\frac{R}{2}\delta_-\mathbf{h}_{k+1/2}^{n+1}=\mathbf{u}_k^n-\frac{R}{2}\delta_-\mathbf{h}_{k+1/2}^n. \quad (9.11.6)$$

It should be clear that difference schemes written as (9.11.5) or (9.11.6) will produce conservative difference schemes. For example, if we consider the scalar linear conservation law $v_t+av_x=0$, we see that using (9.11.6) and choosing h as

$$h_{k+1/2}^m=a(u_k^m+u_{k+1}^m) \quad (9.11.7)$$

where m is either n or $n+1$ will give us the traditional Crank-Nicolson scheme. If we use difference scheme (9.11.6) and choose h as

$$h_{k+1/2}^m=F_k^m++F_{k+1}^m, \quad (9.11.8)$$

where again m is either n or $n+1$, we obtain a Crank-Nicolson scheme for approximating the solutions to the nonlinear scalar conservation law (9.7.1). Also, it should be clear that even though we used the same numerical flux function h to define the flux at times steps n and $n+1$, this is not necessary. All that is necessary is that the fluxes that are used at the n-th and $(n+1)$-st time levels sum to approximate equations (9.11.3) or (9.11.4) adequately.

We should understand from the beginning that difference schemes of the form (9.11.6) cause some special difficulties. When we used implicit schemes for linear equations, we found that we had to solve a linear matrix equation at each time step. This difficulty is included in difference scheme (9.11.6). Specifically, if we use difference scheme (9.11.6) along with h defined as (9.11.7), to approximate the solutions to the one way wave equation, we will have to solve a tridiagonal matrix equation at each time step (as we

did often in Part 1). However, when we are considering nonlinear conservation laws, if we approximate the implicit fluxes in a nonlinear way, solving difference equation (9.11.6) for $\mathbf{u}_k^{n+1}$ will involve solving a nonlinear algebraic equation. Since the results for implicit schemes for conservation laws are analogous to the results for explicit schemes for conservation laws, if we want the scheme to be TVD, we have to use a nonlinear difference equation. If we are sufficiently clever, we may be able to make the explicit part of the scheme nonlinear and the implicit part of the scheme linear. If not, solving equation (9.11.6) for $\mathbf{u}_k^{n+1}$ will generally involve something like a Newton's method or some sort of linearization. Recall that in Section 3.4.2 we set up an implicit nonlinear difference scheme for solving the viscous Burgers' equation, HW0.0.1, and solved the resulting equation by three methods, lagging the nonlinear term, linearizing about the previous time step, and Newton's method.

It is hoped that the above discussion makes the reader aware of the fact that it is a possibility to use implicit schemes to approximate the solutions to general conservation laws, both scalar and K-system conservation laws. There are advantages and difficulties to using implicit schemes. There are also theoretical results available concerning the properties of these schemes. The logical approach would be now to include the results available analogous to the results for explicit schemes given in Sections 9.4–9.10.4.3. We are not going to do this. We want the reader to be aware that implicit schemes are available. If the reader decides that implicit schemes are needed, we suggest ref. [74].

9.12 Difference Schemes for Two Dimensional Conservation Laws

In Chapter 5 we developed difference schemes for the two dimensional linear scalar partial differential equation

$$v_t + av_x + bv_y = 0. \tag{9.12.1}$$

In Chapter 6 we developed schemes for approximating the solution to the two dimensional linear system of partial differential equations

$$\mathbf{v}_t = A_1\mathbf{v}_x + A_2\mathbf{v}_y. \tag{9.12.2}$$

Both of these equations are two dimensional hyperbolic conservation laws. In fact, in Section 5.8.1 we use the conservation law approach in the derivation of difference schemes for approximating solutions to equations (9.12.1).

Here we will consider two dimensional scalar conservation laws

$$v_t + F_x + G_y = 0 \tag{9.12.3}$$

and two dimensional K-system conservation laws

$$\mathbf{v}_t + \mathbf{F}_x + \mathbf{G}_y = \boldsymbol{\theta}, \tag{9.12.4}$$

where $F = F(v)$, $G = G(v)$, $\mathbf{F} = \mathbf{F}(\mathbf{v})$ and $\mathbf{G} = \mathbf{G}(\mathbf{v})$. The theory of difference schemes for two dimensional conservation laws is much less developed than the theory for the one dimensional conservation laws. We will not let this concern us. We will present some theoretical results that we feel are useful. We will concentrate our work on a discussion of some of the methods for developing difference schemes for two dimensional conservation laws.

Eventually, we usually want to solve initial–boundary–value problems. As in the earlier sections of this chapter, we consider difference schemes for initial-value problems. Boundary conditions for initial–boundary–value problems involving hyperbolic conservation laws present all of the difficulties that we encountered with boundary conditions for initial–boundary–value problems involving one dimensional hyperbolic partial differential equations (Chapter 8) and the difficulties that we encountered with boundary conditions for initial–boundary–value problems involving two dimensional linear systems of hyperbolic partial differential equations (Chapter 6). We will generally not have enough boundary conditions. We will often have to use numerical boundary conditions (and choose them carefully). In addition, an added difficulty that we have when we consider nonlinear conservation laws is that because of the variability of $\mathbf{F}$ and $\mathbf{G}$, the number of boundary conditions allowed on a given boundary may vary with time and with the solution. Boundary conditions will not cause us difficulties for the problems considered in this section. We will consider initial–boundary–value problems that are sufficiently easy that we can easily choose boundary conditions and/or numerical boundary conditions that will not interfere with our solutions, i.e., that will work. If we must deal with boundary conditions, at this time we must use the results found in Chapter 8 and extend them for use (carefully) for two dimensional conservation laws and their difference schemes.

Of course, as we define conservative difference schemes for two dimensional conservation laws, we would like to find high order, TVD schemes. There are a variety of ways that we could define TVD for two dimensional schemes. Though we will not really use it much, we say that a two dimensional difference scheme is TVD if $TV(\mathbf{u}^{n+1}) \leq TV(\mathbf{u}^n)$, where

$$TV(\mathbf{u}^m) = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} \left[\Delta x \|\mathbf{u}^m_{j+1\,k} - \mathbf{u}^m_{j\,k}\| + \Delta y \|\mathbf{u}^m_{j\,k+1} - \mathbf{u}^m_{j\,k}\|\right]$$

and $\|\cdot\|$ denotes the usual $\mathbb{R}^K$ norm. With this definition, we are able to state the following result due to Goodman and LeVeque, ref. [14], that shows us that we will not be able to obtain second order accurate TVD schemes.

Proposition 9.12.1 *Most two dimensional TVD schemes are at most first order accurate.*

Remark: The first "Most" in the statement of Proposition 9.12.1 could or should read "Except for a small number of trivial cases."

Though the above negative result is frustrating, we remind the reader that for one dimensional schemes, we could not find TVD schemes that were second order accurate everywhere. One of the approaches that researchers seem to take when working with two dimensional schemes is to consider ENO schemes (essentially nonoscillatory, defined for one dimensional schemes in Section 9.7.1) rather than TVD schemes.

We define a two dimensional grid on $\mathbb{R}^2$, (x_j, y_k), $j = -\infty, \ldots, \infty$, $k = -\infty, \ldots, \infty$, where $x_{j+1} - x_j = \Delta x$ and $y_{k+1} - y_k = \Delta y$ for all j and k. We consider the control volume associated with the point (x_j, y_k) (which we will often just refer to as the j-k point), $(x_{j-1/2}, x_{j+1/2}) \times (y_{k-1/2}, y_{k+1/2})$ (where the half indices have the same two dimensional meaning that they had in one dimension in the rest of the chapter). As usual, we consider the time levels $t_n = n\Delta t$, $n = 0, 1, \ldots$ and denote the approximation to the solution $\mathbf{v}$ at (x_j, y_k, t_n) by $\mathbf{u}^n_{jk}$. Analogously to the case with difference schemes for one dimensional conservation laws, $\mathbf{u}^n_{jk}$ will represent the approximation of the average of the solution $\mathbf{v}$ over the cell $(x_{j-1/2}, x_{j+1/2}) \times (y_{k-1/2}, y_{k+1/2})$.

If we integrate equation (9.12.4) from t_n to t_{n+1} with respect to t, from $x_{j-1/2}$ to $x_{j+1/2}$ with respect to x, and from $y_{k-1/2}$ to $y_{k+1/2}$ with respect to y and perform the obvious integrations, we get

$$\begin{aligned}\boldsymbol{\theta} = &\int_{y_{k-1/2}}^{y_{k+1/2}} \int_{x_{j-1/2}}^{x_{j+1/2}} \left[\mathbf{v}(x, y, t_{n+1}) - \mathbf{v}(x, y, t_n)\right] dx\, dy \\ &+ \int_{t_n}^{t_{n+1}} \int_{y_{k-1/2}}^{y_{k+1/2}} \left[\mathbf{F}(\mathbf{v}(x_{j+1/2}, y, t)) - \mathbf{F}(\mathbf{v}(x_{j-1/2}, y, t))\right] dy\, dt \\ &+ \int_{t_n}^{t_{n+1}} \int_{x_{j-1/2}}^{x_{j+1/2}} \left[\mathbf{G}(\mathbf{v}(x, y_{k+1/2}, t)) - \mathbf{G}(\mathbf{v}(x, y_{k-1/2}, t))\right] dx\, dt.\end{aligned} \tag{9.12.5}$$

Defining $\mathbf{v}^n_{jk}$ as

$$\mathbf{v}^n_{jk} = \frac{1}{\Delta x \Delta y} \int_{y_{k-1/2}}^{y_{k+1/2}} \int_{x_{j-1/2}}^{x_{j+1/2}} \mathbf{v}(x, y, t_n)\, dx\, dy \tag{9.12.6}$$

allows us to rewrite equation (9.12.5) as

$$\mathrm{v}_{j\,k}^{n+1} = \mathbf{v}_{j\,k}^{n} - \frac{1}{\Delta x \Delta y}\int_{t_n}^{t_{n+1}}\int_{y_{k-1/2}}^{y_{k+1/2}} \left[\mathbf{F}(\mathrm{v}(x_{j+1/2}, y, t) - F(x_{j-1/2}, y, t))\right] dy\, dt$$
$$-\frac{1}{\Delta x \Delta y}\int_{t_n}^{t_{n+1}}\int_{x_{j-1/2}}^{x_{j+1/2}} \left[\mathbf{G}(\mathrm{v}(x, y_{k+1/2}, t)) - \mathbf{G}(\mathrm{v}(x, y_{k-1/2}, t))\right] dx\, dt. \tag{9.12.7}$$

We should realize that equation (9.12.7) is an exact equation. Equation (9.12.7) states that the change in the amount of conserved quantity over the time interval $[t_n, t_{n+1}]$ is equal to the fluxes of that quantity across the four boundaries of the region. Clearly, this equation is analogous to equation (9.6.5) in Section 9.6.2. To obtain a two dimensional conservative scheme, we approximate equation (9.12.7) by

$$\mathbf{u}_{j\,k}^{n+1} = \mathbf{u}_{j\,k}^{n} - R_x\left[\mathbf{p}_{j+1/2\,k}^{n} - \mathbf{p}_{j-1/2\,k}^{n}\right] - R_y\left[\mathbf{q}_{j\,k+1/2}^{n} - \mathbf{q}_{j\,k-1/2}^{n}\right], \tag{9.12.8}$$

where $R_x = \Delta t/\Delta x$ and $R_y = \Delta t/\Delta y$. The terms $-R_x\delta_{x-}\mathbf{p}_{j+1/2\,k}^{n}$ and $-R_y\delta_{y-}\mathbf{q}_{j\,k+1/2}^{n}$ approximate the second and third terms on the right side of equation (9.12.7), respectively, and these terms can be chosen to approximate the integrals in (9.12.7) as accurately as we want. The functions $\mathbf{p}$ and $\mathbf{q}$ will be referred to as the x-direction and y-direction numerical flux functions, respectively.

It should be clear that as in the one dimlnsional case, $\mathbf{p}$ will depend on $\mathbf{u}_{j-p\,k}^{n}, \ldots, \mathbf{u}_{j+q\,k}^{n}$ for some p and q (where the p and q in the subscripts have no relationship to $\mathbf{p}$ and $\mathbf{q}$). However, we must now consider the dependence of $\mathbf{p}$ on some other $\mathbf{u}$'s. Generally, we will allow $\mathbf{p}$ to depend on $\mathbf{u}_{j^*\,k-r}^{n}, \ldots, \mathbf{u}_{j^*\,k+r}^{n}$ for $j^* = j-p, \ldots, j+q$. The need for this generality will not appear often. Too often, r will be zero. There are situations where it is desirable and/or necessary to use more than the k index to approximate the flux across $x = x_{j+1/2}$, $y_{k-1/2} \le x \le y_{k+1/2}$. We must remember that we are approximating the second term of the right hand side of equation (9.12.7), and that term contains an integral from $y_{k-1/2}$ to $y_{k+1/2}$. Of course, an analogous discussion holds with respect to the dependence of $\mathbf{q}$ on $\mathbf{u}_{j^*\,k^*}^{n}$, $j^* = j-r, \ldots, j+r$, $k^* = k-p, \ldots, k+q$.

As with the one dimensional conservation laws, the consistency of the difference scheme can be related to the relationship between $\mathbf{p}$ and $\mathbf{F}$, and $\mathbf{q}$ and $\mathbf{G}$. We state the following proposition, the proof of which is very similar to the proof of Proposition 9.6.3.

Proposition 9.12.2 *If* $\mathbf{p}(\mathbf{u}, \ldots, \mathbf{u}) = \mathbf{F}(\mathbf{u})$ *and* $\mathbf{q}(\mathbf{u}, \ldots, \mathbf{u}) = \mathbf{G}(\mathbf{u})$, *then difference scheme (9.12.8) will be consistent with conservation law (9.12.4).*

If we review Sections 5.8.1-5.8.2 and Sections 6.7.1.1-6.7.2, we see that difference schemes (5.8.9), (5.8.13), (6.7.3) (with $C_0 = \Theta$), and (6.7.39) can be expressed as conservation laws by defining $\mathbf{p}$ and $\mathbf{q}$ (or p and q) as

$$p^n_{j+1/2\,k} = a u^n_{j\,k} \quad q^n_{j\,k+1/2} = b u^n_{j\,k} \quad \text{(scheme (5.8.9))} \tag{9.12.9}$$

$$p^n_{j+1/2\,k} = \frac{a}{2}\left(u^n_{j\,k} + u^n_{j+1\,k}\right) - \frac{1}{4R_x}\delta_{x+} u^n_{j\,k}$$

$$q^n_{j\,k+1/2} = \frac{b}{2}\left(u^n_{j\,k} + u^n_{j\,k+1}\right) - \frac{1}{4R_y}\delta_{y+} u^n_{j\,k} \quad \text{(scheme (5.8.13))} \tag{9.12.10}$$

$$\mathbf{p}^n_{j+1/2\,k} = -A_1 \mathbf{u}^n_{j+1\,k} \quad \mathbf{q}^n_{j\,k+1/2} = -A_2 \mathbf{u}^n_{j\,k+1} \quad \text{(scheme (6.7.3))} \tag{9.12.11}$$

$$\mathbf{p}^n_{j+1/2\,k} = -\frac{1}{2} A_1\left(\mathbf{u}^n_{j\,k} + \mathbf{u}^n_{j+1\,k}\right) - \frac{1}{4R_x}\delta_{x+}\mathbf{u}^n_{j\,k}$$

$$\mathbf{q}^n_{j\,k+1/2} = -\frac{1}{2} A_2\left(\mathbf{u}^n_{j\,k} + \mathbf{u}^n_{j\,k+1}\right) - \frac{1}{4R_y}\delta_{y+}\mathbf{u}^n_{j\,k} \quad \text{(scheme (6.7.39))}, \tag{9.12.12}$$

respectively. We note that in all cases, the flux function p or $\mathbf{p}$ looks like a one dimensional flux function in j, holding k fixed. This happens because the flux across $x = x_{j+1/2}$, $y_{k-1/2} \le y \le y_{k+1/2}$ is approximated on the entire interval by the flux across $x = x_{j+1/2}$ along the line $y = y_k$. An analogous statement holds true for $x = x_{j-1/2}$ and for q and $\mathbf{q}$ also. To use one dimensional flux functions to define two dimensional flux functions is not the only way and is surely not the best way for all problems, but it happens often and does give us one method for deriving difference schemes for approximating solutions to equations (9.12.3) and (9.12.4). Thus we can return to any of the one dimensional numerical flux functions considered earlier in this chapter and use them to define the following two dimensional numerical flux functions, which are designed to produce difference schemes for approximating the solution to conservation law (9.12.4). For example, we can define

$$\mathbf{p}^n_{j+1/2\,k} = \mathbf{F}^n_{j+1\,k} \quad \mathbf{q}^n_{j\,k+1/2} = \mathbf{G}^n_{j\,k+1} \tag{9.12.13}$$

$$\mathbf{p}^n_{j+1/2\,k} = \frac{1}{2}\left(\mathbf{F}^n_{j\,k} + \mathbf{F}^n_{j+1\,k}\right) - \frac{1}{4R_x}\delta_{x+}\mathbf{u}^n_{j\,k}$$

$$\mathbf{q}^n_{j\,k+1/2} = \frac{1}{2}\left(\mathbf{G}^n_{j\,k} + \mathbf{G}^n_{j\,k+1}\right) - \frac{1}{4R_y}\delta_{y+}\mathbf{u}^n_{j\,k} \tag{9.12.14}$$

$$\mathbf{p}^n_{j+1/2\,k} = \frac{1}{2}\left(\mathbf{F}^n_{j\,k} + \mathbf{F}^n_{j+1\,k}\right) - \frac{R_x}{2} A_{j+1/2\,k}\delta_{x+}\mathbf{F}^n_{j\,k}$$

$$\mathbf{q}^n_{j\,k+1/2} = \frac{1}{2}\left(\mathbf{G}^n_{j\,k} + \mathbf{G}^n_{j\,k+1}\right) - \frac{R_y}{2} B_{j\,k+1/2}\delta_{y+}\mathbf{G}^n_{j\,k}, \tag{9.12.15}$$

where

$$A_{j+1/2\,k} = \mathbf{F}'\left(\left(\mathbf{u}^n_{j\,k} + \mathbf{u}^n_{j+1\,k}\right)/2\right) \text{ and } B_{j\,k+1/2} = \mathbf{G}'\left(\left(\mathbf{u}^n_{j\,k} + \mathbf{u}^n_{j\,k+1}\right)/2\right).$$

Of course, we realize that numerical flux function (9.12.13) is associated with the FTFS scheme. In addition, we could have defined several more schemes similar to (9.12.13) as we did in HW5.8.6, where we use all combinations of forward and backward in space to approximate the spatial derivatives. These schemes have all of the difficulties associated with the one dimensional schemes using one sided differences to approximate the spatial derivatives. For scalar conservation law (9.12.3), instead of using either numerical flux function (9.12.13), we would use the one dimensional upwind numerical flux function to define the following two dimensional upwind numerical flux functions.

$$p^n_{j+1/2\,k} = \frac{1}{2}\left(F^n_{j\,k} + F^n_{j+1\,k}\right) - \frac{1}{2}|a^n_{j+1/2\,k}|\delta_{x+}u^n_{j\,k} \tag{9.12.16}$$

$$q^n_{j\,k+1/2} = \frac{1}{2}\left(G^n_{j\,k} + G^n_{j\,k+1}\right) - \frac{1}{2}|b^n_{j\,k+1/2}|\delta_{y+}u^n_{j\,k}, \tag{9.12.17}$$

where $a^n_{j+1/2\,k}$ is defined with respect to F in the same way as the one dimensional $a^n_{j+1/2}$ is defined with k held fixed and $b^n_{j\,k+1/2}$ is the G analogue of $a^n_{j+1/2\,k}$, where we use G, fix j and difference k. Likewise, we could use the one dimensional numerical flux function (9.9.81) to define an upwind scheme for two dimensional systems that includes the sonic rarefaction fix.

Numerical flux functions (9.12.14) are those associated with the two dimensional Lax-Friedrichs difference scheme. And finally, we should understand that the numerical flux functions (9.12.15) will produce the difference scheme that is the nonlinear version of the difference scheme given in HW5.8.4, which we referred to as the "not Lax-Wendroff scheme." Recall that in HW5.8.4 we showed that the linear version of the difference scheme associated with numerical flux functions (9.12.15) is not second order accurate. Hence, we see that the one dimensional numerical flux functions can be used to define two dimensional flux functions, but we must be careful of any claims of accuracy of the resulting schemes based on the accuracy of the one dimensional schemes. Consistency is relatively easy. The computations necessary to apply Proposition 9.6.3 to obtain consistency for the analogous one dimensional scheme will guarantee consistency of the resulting two dimensional scheme by Proposition 9.12.2. The accuracy of schemes like the Lax-Wendroff scheme where the higher order accuracy is obtained by adding out part of the expansion of the time difference with part of the expansion of the spatial difference must be inspected very carefully when extended to two dimensions.

The approach that we used in Section 5.8.2 and HW6.7.3 to obtain a two dimensional Lax-Wendroff scheme was to use an approximate factorization technique. If we integrate conservation law (9.12.4) with respect to t from t_n to t_{n+1}, we get

$$\int_{t_n}^{t_{n+1}} \mathbf{v}_t\, dt = \mathbf{v}^{n+1} - \mathbf{v}^n = -\int_{t_n}^{t_{n+1}} \left(\mathbf{F}_x + \mathbf{G}_y\right) dt = -\left(\mathbf{F}^n_x + \mathbf{G}^n_y\right)\Delta t + \mathcal{O}(\Delta t^2),$$

or

$$\mathbf{v}^{n+1} = \mathbf{v}^n - (\mathbf{F}_x^n + \mathbf{G}_y^n)\Delta t + \mathcal{O}(\Delta t^2), \tag{9.12.18}$$

where we have used a first order approximate integration scheme on the flux terms. Expanding the flux terms and adding a higher order term, we can write (9.12.18) as

$$\begin{aligned}\mathbf{v}^{n+1} &= \mathbf{v}^n - \Delta t \mathbf{F}'^n(\mathbf{v}^n)\mathbf{v}_x^n - \Delta t \mathbf{G}'^n(\mathbf{v}^n)\mathbf{v}_y^n + \mathcal{O}(\Delta t^2) \\ &= \mathbf{v}^n - \Delta t \mathbf{F}'^n(\mathbf{v}^n)\mathbf{v}_x^n - \Delta t \mathbf{G}'^n(\mathbf{v}^n)\mathbf{v}_y^n + \Delta t^2 \mathbf{F}'^n \frac{\partial}{\partial x}\mathbf{G}'^n \frac{\partial}{\partial y}\mathbf{v}^n \\ &\quad + \mathcal{O}(\Delta t^2),\end{aligned}$$

or

$$\mathbf{v}^{n+1} = \left[I - \Delta t \mathbf{F}'^n \frac{\partial}{\partial x}\right]\left[I - \Delta t \mathbf{G}'^n \frac{\partial}{\partial y}\right]\mathbf{v}^n + \mathcal{O}(\Delta t^2). \tag{9.12.19}$$

Difference scheme (9.12.19) can be written in split form without the $\mathcal{O}$ term as

$$\mathbf{v}^{n+1/2} = \left[I - \Delta t \mathbf{F}'^n(\mathbf{v}^n)\frac{\partial}{\partial x}\right]\mathbf{v}^n \tag{9.12.20}$$

$$\mathbf{v}^{n+1} = \left[I - \Delta t \mathbf{G}'^n(\mathbf{v}^n)\frac{\partial}{\partial y}\right]\mathbf{v}^{n+1/2}. \tag{9.12.21}$$

Of course, at this time it would be necessary to decide how to difference with respect to x and y. Difference schemes such as that given in (9.12.20)–(9.12.21) are used often. In fact, difference scheme (9.12.20)–(9.12.21) is a special case of the Beam-Warming scheme developed in ref. [72]. In ref. [72], Beam and Warming use the approach above with a very general numerical integration scheme with respect to t to get a difference scheme that contains several different logical schemes (pure implicit, Crank-Nicolson, etc.) as well as difference scheme (9.12.20)–(9.12.21) given above. As we shall see later, the above split scheme is not ideal for what we want to do with it. The flux term in equation (9.12.20) can be rewritten as $\mathbf{F}_x^n$, but the flux term in equation (9.12.21) cannot be written in that form. We could just approximate equation (9.12.21) in the way that we want it written, but we shall use a different approach.

The Lax-Wendroff schemes obtained in Section 5.8.2 and HW6.7.3 are explicit locally one dimensional schemes. We studied implicit locally one dimensional schemes in HW4.4.6 and Section 5.8.3. Another way to consider split schemes for two dimensional conservation laws is to consider a locally one dimensional scheme for solving conservation law (9.12.4) of the form

$$\mathbf{u}_{jk}^{n+1/2} = \mathbf{u}_{jk}^n - R_x \delta_{x-} \mathbf{p}_{j+1/2\,k}^n \tag{9.12.22}$$

$$\mathbf{u}_{jk}^{n+1} = \mathbf{u}_{jk}^{n+1/2} - R_y \delta_{y-} \mathbf{q}_{j\,k+1/2}^{n+1/2}, \tag{9.12.23}$$

where $\mathbf{p}$ and $\mathbf{q}$ are numerical flux functions with respect to x and y, respectively, that satisfy $\mathbf{p}(\mathbf{u},\ldots,\mathbf{u}) = \mathbf{F}(\mathbf{u})$ and $\mathbf{q}(\mathbf{u},\ldots,\mathbf{u}) = \mathbf{G}(\mathbf{u})$. We see that difference scheme (9.12.22)–(9.12.23) is not that different from difference scheme (9.12.20)–(9.12.21) other than that the latter scheme assumes that we have already differenced with respect to x and y (though we do not know how) and that the form of equation (9.12.21) would limit the choices of $\mathbf{q}$ more than we would like. If we use an abominable, though convenient, notation and write $\mathbf{q}^{n+1/2}_{j\,k+1/2}$ as

$$\mathbf{q}^{n+1/2}_{j\,k+1/2} = \mathbf{q}\big(\mathbf{u}^{n+1/2}_{j^*\,k^*}\big),$$

where j^* and k^* vary over the values of j and k that are included in the definition of $\mathbf{q}$, we can then use the fact that

$$\mathbf{u}^{n+1/2}_{j^*\,k^*} = \mathbf{u}^{n}_{j^*\,k^*} - R_x\delta_{x-}\mathbf{q}^{n}_{j^*+1/2\,k^*}$$

and expand $\mathbf{q}^{n+1/2}_{j\,k+1/2}$ in a power series about $\mathbf{u}^n_{j\,k}$ to get

$$\mathbf{q}^{n+1/2}_{j\,k+1/2} = \mathbf{q}^{n}_{j\,k+1/2} + \sum_{j^*,k^*}\big\{\partial_{j^*,k^*}\mathbf{q}\big\}\big[\mathbf{u}^{n+1/2}_{j^*\,k^*} - \mathbf{u}^{n}_{j^*\,k^*}\big] + \cdots \tag{9.12.24}$$

$$= \mathbf{q}^{n}_{j\,k+1/2} + \sum_{j^*,k^*}\big\{\partial_{j^*,k^*}\mathbf{q}\big\}\big[-R_x\delta_{x-}\mathbf{p}^{n}_{j^*+1/2\,k^*}\big] + \cdots, \tag{9.12.25}$$

where the sums and partial derivatives in (9.12.24) and (9.12.25) are taken over all of the j^* and k^* such that $\mathbf{u}^{n+1/2}_{j^*\,k^*}$ is an argument of $\mathbf{q}$. From equation (9.12.25) we see that

$$R_y\delta_{y-}\mathbf{q}^{n+1/2}_{j\,k+1/2} = R_y\delta_{y-}\mathbf{q}^{n}_{j\,k+1/2} - \Delta t^2\sum_{j^*,k^*}\frac{1}{\Delta y}\delta_{y-}\big\{\partial_{j^*,k^*}\mathbf{q}\big\}\frac{1}{\Delta x}\delta_{x-}\mathbf{p}^{n}_{j^*+1/2\,k^*} + \cdots,$$

so with the appropriate assumptions on the smoothness of $\mathbf{q}$ and the solutions and the assumption that we are using the same $\mathbf{p}$ and $\mathbf{q}$, split scheme (9.12.22)–(9.12.23) is a $\mathcal{O}(\Delta t^2)$ approximation of difference scheme (9.12.8). Hence, we see that though difference scheme (9.12.22)–(9.12.23) may not be a conservative scheme, it is at least an approximately conservative scheme.

Of course, we must understand that we can use any of the p's, q's, $\mathbf{p}$'s and $\mathbf{q}$'s given in (9.12.9)–(9.12.15) along with (9.12.22)–(9.12.23) and have a potential difference scheme. As we mentioned earlier, by choosing

$$p^{n}_{j+1/2\,k} = au^{n}_{j\,k} + \frac{1}{2}a(1-aR_x)\delta_{x+}u^{n}_{j\,k} \tag{9.12.26}$$

$$q^{n+1/2}_{j\,k+1/2} = bu^{n+1/2}_{j\,k} + \frac{1}{2}b(1-bR_y)\delta_{y+}u^{n+1/2}_{j\,k}, \tag{9.12.27}$$

this approach will give us the linear, two dimensional Lax-Wendroff schemes developed in Section 5.8.2 and HW6.7.3. In addition, defining p and q as

$$p^n_{j+1/2\,k} = \frac{1}{2}\left(F^n_{j\,k} + F^n_{j+1\,k}\right) - \frac{R_x}{2}\left(a^n_{j+1/2\,k}\right)^2 \delta_{x+} u^n_{j\,k} \tag{9.12.28}$$

$$p^{n+1/2}_{j\,k+1/2} = \frac{1}{2}\left(G^{n+1/2}_{j\,k} + G^n_{j\,k+1}\right) - \frac{R_y}{2}\left(b^{n+1/2}_{j\,k+1/2}\right)^2 \delta_{y+} u^{n+1/2}_{j\,k} \tag{9.12.29}$$

gives a Lax-Wendroff scheme for scalar two dimensional nonlinear conservation laws.

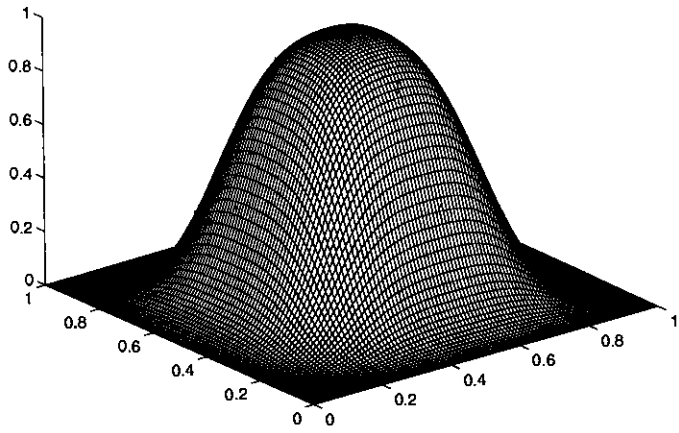

FIGURE 9.12.1. An approximation to the solution of initial–boundary–value problem (9.12.30)–(9.12.32) obtained using the two dimensional upwind difference scheme associated with numerical flux functions (9.12.9) or (9.12.16)–(9.12.17), $a = b = 1.0$, $\Delta x = \Delta y = 0.01$, $\Delta t = 0.002$, and the solution is plotted at time $t = 1.0$.

9.12.1 Some Computational Examples

Now that we have developed some logical approaches for developing two dimensional schemes for approximating the solutions to two dimensional conservation laws, it is time to test some of these schemes. Some of the results shown in this section might be obvious, but are included so as to make the discussion of these schemes complete. We begin by considering

the two dimensional initial–boundary–value problem

$$v_t + av_x + bv_y = 0, \quad (x,y) \in (0,1) \times (0,1), \ t > 0 \tag{9.12.30}$$

$$v(x,0) = \begin{cases} 1 & \text{if } (x,y) \in [0.25, 0.75] \times [0.25, 0.75] \\ 0 & \text{otherwise} \end{cases} \tag{9.12.31}$$

$$v(0,y) = v(1,y), \quad y \in [0,1], \ v(x,0) = v(x,1), \ x \in [0,1]. \tag{9.12.32}$$

We note that conservation law (9.12.30) is both scalar and linear. This problem is enough of a test for the moment. In Figure 9.12.1 we see an approximation to solution to initial–boundary–value problem (9.12.30)–(9.12.32) plotted at time $t = 1.0$. The numerical solution was obtained using the upwind scheme (p and q defined by either (9.12.9) or (9.12.16)–(9.12.17)) with $a = b = 1.0$, $\Delta x = \Delta y = 0.01$ and $\Delta t = 0.002$. We note that the approximate solution is badly smeared, which is what we should expect. As in the one dimensional case, the upwind scheme is not sufficiently accurate to be used by itself as a scheme to resolve discontinuities.

In Figures 9.12.2 and 9.12.3 we include an approximation to the solution of the same problem at time $t = 0.2$, obtained using the split Lax-Wendroff scheme (9.12.22)–(9.12.23) with p and q defined by (9.12.26)–(9.12.27). We see that just as in the one dimensional case, the two dimensional Lax-Wendroff introduces wiggles into the solution (certainly due to numerical dispersion) that limits the use of the solution. You should notice that when trying to view solutions to two dimensional problems, both the three dimensional plot of the solution as in Figure 9.12.2 and the contour plot as in Figure 9.12.3 can give you a perspective that the other plot sometimes cannot give.

9.12.2 Some Two Dimensional High Resolution Schemes

As we saw in the last section, when we try to resolve solutions with discontinuities, we have the same problems with the common schemes that we had when we considered one dimensional problems. As in the one dimensional case, we would like to develop higher order schemes (second or better), conservative TVD schemes that satisfy the appropriate discrete entropy condition. At the moment, this is not possible. What we will do here is use some of the ideas introduced earlier to construct some schemes that give us some of the attributes that we want. As we did in Section 9.12, we can try to build two dimensional high resolution schemes by using difference scheme (9.12.8) or (9.12.22)–(9.12.23), where we define p and q (or even $\mathbf{p}$ and $\mathbf{q}$) analogously to the way we defined h (or $\mathbf{h}$) in Sections 9.7.4–9.10.4. For example, we could define p and q as

$$p^n_{j+1/2\,k} = au^n_{j\,k} + \phi^n_{x_{j\,k}} \frac{1}{2} a(1 - aR_x)\delta_{x+} u^n_{j\,k} \tag{9.12.33}$$

$$q^n_{j\,k+1/2} = bu^n_{j\,k} + \phi^n_{y_{j\,k}} \frac{1}{2} b(1 - bR_y)\delta_{y+} u^n_{j\,k}, \tag{9.12.34}$$

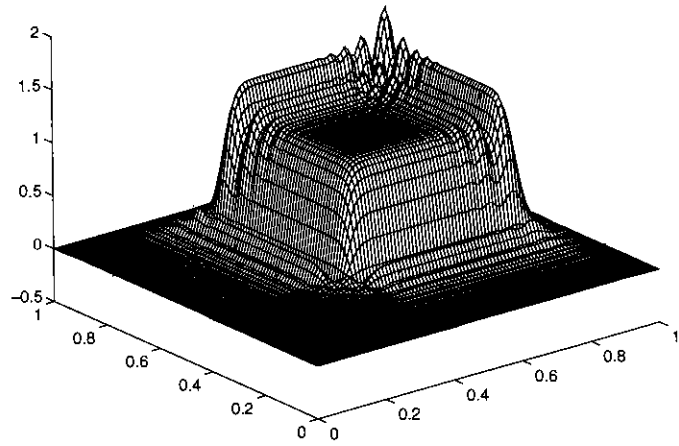

FIGURE 9.12.2. An approximation to the solution of initial–boundary–value problem (9.12.30)–(9.12.32) obtained using the two dimensional split Lax-Wendroff scheme (9.12.22)–(9.12.23) with p and q defined by (9.12.26)–(9.12.27). We use $a = b = 1.0$, $\Delta x = \Delta y = 0.01$, $\Delta t = 0.002$, and the solution is plotted at time $t = 1.0$.

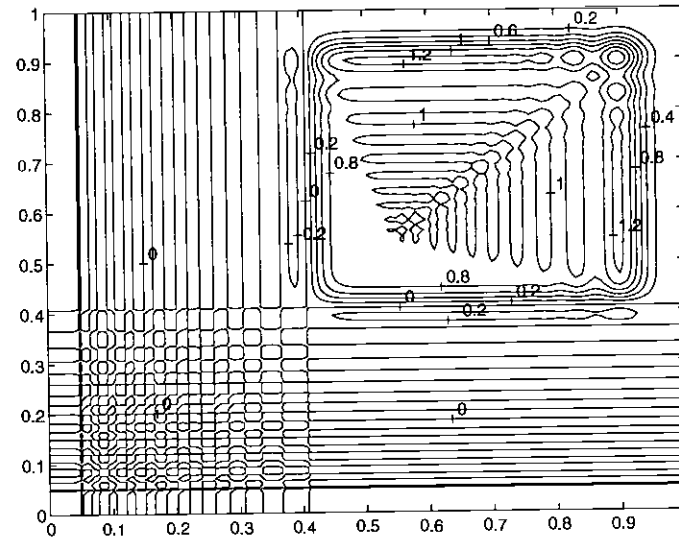

FIGURE 9.12.3. A contour plot of the approximate solution given in Figure 9.12.2.

where $\phi^n_{x_{jk}} = \phi_x(\theta^n_{x_{jk}})$, $\phi_{y_{jk}} = \phi_y(\theta^n_{y_{jk}})$, ϕ_x and ϕ_y are limiter functions, and $\theta^n_{x_{jk}}$ and $\theta^n_{y_{jk}}$ are smoothness parameters in the x and y directions, respectively, that are defined as

$$\theta^n_{x_{jk}} = \frac{\delta_{x-}u^n_{jk}}{\delta_{x+}u^n_{jk}} \tag{9.12.35}$$

$$\theta^n_{y_{jk}} = \frac{\delta_{y-}u^n_{jk}}{\delta_{y+}u^n_{jk}}. \tag{9.12.36}$$

Of course, we must understand that the high resolution fluxes chosen above are two dimensional analogues of those used in Section 9.7.5.1 and for that reason we should hope that difference scheme (9.12.8), (9.12.33)–(9.12.36) will have some of the attributes that we want. We must be aware that we have assumed that $a > 0$ and $b > 0$. These assumptions can easily be eliminated by using the upwind scheme

$$p^n_{L_{j+1/2\,k}} = \frac{1}{2}a(u^n_{jk} + u^n_{j+1\,k}) - \frac{1}{2}|a|\delta_{x+}u^n_{jk} \tag{9.12.37}$$

$$q^n_{L_{j\,k+1/2}} = \frac{1}{2}b(u^n_{jk} + u^n_{j\,k+1}) - \frac{1}{2}|b|\delta_{y+}u^n_{jk} \tag{9.12.38}$$

as the low order scheme when we define the numerical flux functions $p^n_{k+1/2}$ and $q^n_{k+1/2}$, (9.12.33)–(9.12.34). We should also realize that we are using the relatively simple upwind scheme here (i.e., no fixes), since we are considering a linear problem. In Figure 9.12.4 we include a plot of an approximate solution at time $t = 1.0$ to initial–boundary–value problem (9.12.30)–(9.12.32) obtained using $a = b = 1.0$, $\Delta x = \Delta y = 0.01$, $\Delta t = 0.002$. The solution was found using difference scheme (9.12.8) with p and q defined as in (9.12.33)–(9.12.36) and using the obvious two dimensional extension of the Superbee limiter function.

We see from the results given in Figure 9.12.4 that it is possible to obtain very good results using such a scheme. We emphasize, however, that initial-boundary–value problem (9.12.30)–(9.12.32) is a very one dimensional problem, but for the moment we accept the result for what it is. We do notice that the approximate solution is less accurate at the corners than at other parts of the jump. This should not come as a surprise to us, since it is at the corners that the flow is the least one dimensional.

For the above computations, we used a fairly obvious choice of numerical flux function, limiter, and smoothness parameters. The limiter functions can be chosen from those developed earlier, or we might want different limit functions for two dimensional schemes. Likewise, it is surely possible to define different smoothness parameters that might be more appropriate for two dimensional problems.

In Figure 9.12.5 we plot an approximation to the same problem at the same time using the Superbee flux-limiter split scheme, i.e., difference

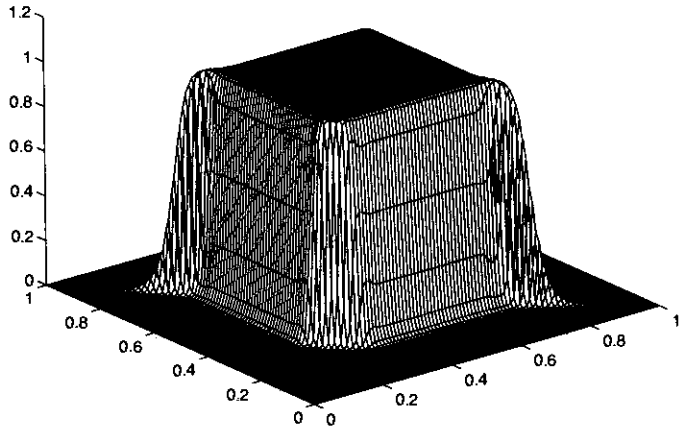

FIGURE 9.12.4. An approximation to the solution of initial–boundary–value problem (9.12.30)–(9.12.32) obtained using the two dimensional scheme defined by (9.12.8), (9.12.33)–(9.12.36), the two dimensional Superbee limiter function, $a = b = 1.0$, $\Delta x = \Delta y = 0.01$ and $\Delta t = 0.002$. The solution is plotted at time $t = 1.0$.

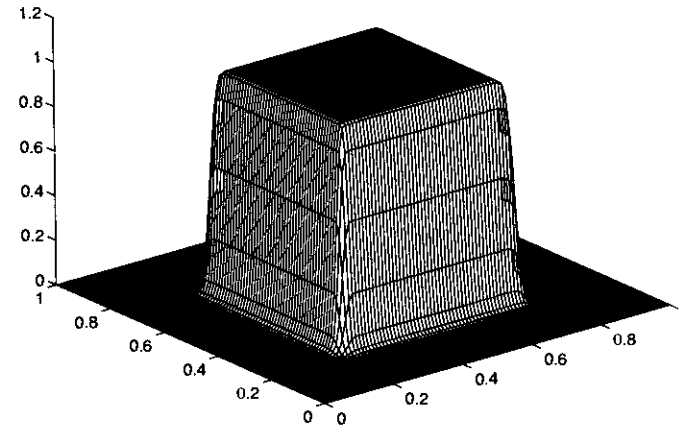

FIGURE 9.12.5. An approximation to the solution of initial–boundary–value problem (9.12.30)–(9.12.32) obtained using the two dimensional scheme defined by (9.12.22)-(9.12.23), (9.12.33)–(9.12.36), the two dimensional Superbee limiter function, $a = b = 1.0$, $\Delta x = \Delta y = 0.01$ and $\Delta t = 0.002$. The solution is plotted

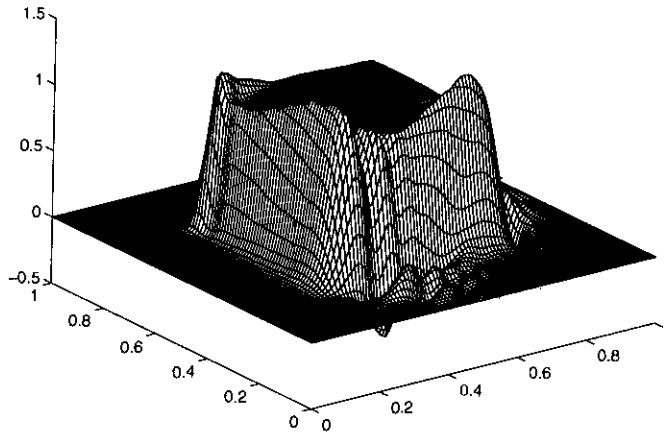

FIGURE 9.12.6. An approximation to the solution of initial–boundary–value problem (9.12.39)–(9.12.40) obtained using the two dimensional scheme defined by (9.12.8), (9.12.33)–(9.12.36), the two dimensional Superbee limiter function, $a(x,y) = \Omega(y-\frac{1}{2})$, $b(x,y) = \Omega(x-\frac{1}{2})$, $\Omega = \sqrt{2}$, $\Delta x = \Delta y = 0.01$ and $\Delta t = 0.002$. The solution is plotted at time $t = 1.11$, which is approximately $t = \pi/(2\Omega)$.

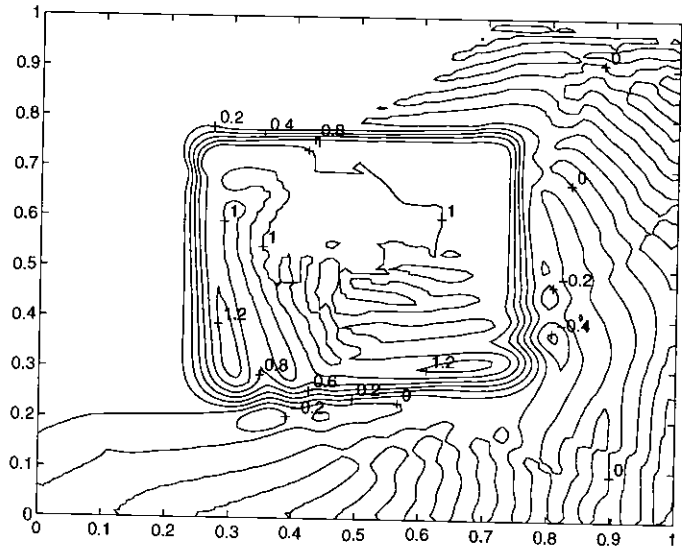

FIGURE 9.12.7. A contour plot of the solution given in Figure 9.12.6.

scheme (9.12.22)–(9.12.23) with p defined as in (9.12.33) above, q defined as in (9.12.34) above with time time step n replaced by $n+\frac{1}{2}$, θ_x and θ_y defined as in (9.12.35) and (9.12.36), and using the Superbee limiter function for both directions. We see that this split scheme can also produce very good results for this problem. Specifically, we see that the split, Superbee scheme resolved the corners of the shock region better than the fully two dimensional Superbee scheme.

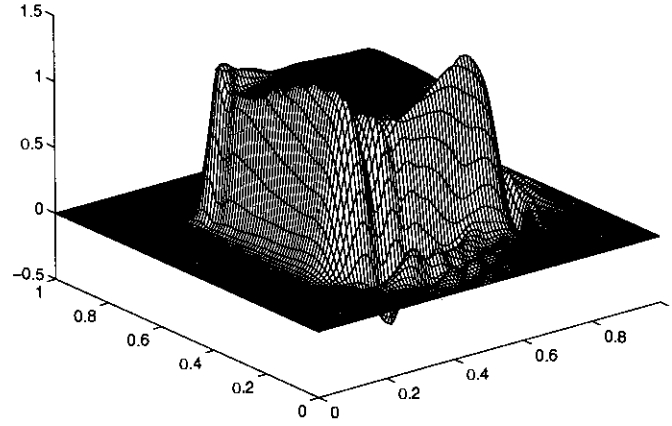

FIGURE 9.12.8. An approximation to the solution of initial–boundary–value problem (9.12.39)–(9.12.40) obtained using the two dimensional scheme defined by (9.12.22)-(9.12.23), (9.12.33)–(9.12.36), the two dimensional Superbee limiter function, $a(x,y) = \Omega(y-\frac{1}{2})$, $b(x,y) = \Omega(x-\frac{1}{2})$, $\Omega = \sqrt{2}$, $\Delta x = \Delta y = 0.01$ and $\Delta t = 0.002$. The solution is plotted at time $t = 1.11$, which is approximately $t = \pi/(2\Omega)$.

Of course we need a more difficult test. In Figures 9.12.6 and 9.12.8 we plot the approximate solution to the initial–boundary–value problem

$$v_t + a(x,y)v_x + b(x,y)v_y = 0 \quad (x,y) \in (0,1)\times(0,1),\ t>0 \tag{9.12.39}$$

$$v(x,y,0) = \begin{cases} 1.0 & \text{when } (x,y) \in [0.25,0.75]\times[0.25,0.75] \\ 0.0 & \text{otherwise} \end{cases} \tag{9.12.40}$$

where the boundary conditions have not been given here but are discussed in HW9.12.1. To make the flow more complex and more difficult to resolve, we have chosen $a(x,y) = -\Omega(y-\frac{1}{2})$, $b(x,y) = \Omega(x-\frac{1}{2})$ and $\Omega = \sqrt{2}$. The plots given in Figures 9.12.6 and 9.12.8 were obtained using difference schemes (9.12.8), (9.12.33)–(9.12.36) and (9.12.22)–(9.12.23), (9.12.33)–(9.12.36), respectively (with n replaced by $n+\frac{1}{2}$ in (9.12.34)).

We should emphasize that though the mathematical boundary conditions allowable for the above problem are difficult to work with, since no action gets near the boundary, we were able to use zero Dirichlet boundary conditions on part of the boundary and zero numerical boundary conditions on the other part of the boundary. We see that both approximate solutions are very similar, probably sufficiently accurate for some applications, surely not accurate enough for all applications and surely less than we would like. In Figure 9.12.7 we include a contour plot of the solution given in Figure 9.12.6. This other view of the solution should give us a little better understanding of how good or bad the solution given in Figure 9.12.6 is.

We might hope that since the split Superbee scheme seemed to work a little better than the nonsplit Superbee scheme for initial–boundary–value problem (9.12.30)–(9.12.32), it might do a better job here. In Figure 9.12.8 we see that the results from using the split scheme on initial–boundary–value problem (9.12.39)–(9.12.40) are no better than those plotted in Figure 9.12.6.

9.12.3 The Zalesak-Smolarkiewicz Scheme

In an attempt to resolve moving discontinuities for multidimensional transport equations, Zalesak and Smolarkiewicz, ref. [77] and [61], devised a multidimensional difference scheme based on the flux corrected transport scheme of Boris and Book, ref. [6]. The Zalesak-Smolarkiewicz scheme, which for reasons that should be obvious we will refer to as the Z-S scheme, can also be developed as either a two dimensional flux-limiter scheme or a two dimensional modified flux scheme. Since we have done more with the flux-limiter schemes and since it is a rather novel approach, we will describe the Z-S scheme as a flux-limiter scheme.

We consider a difference scheme for conservation law (9.12.1), where we allow a and b to be functions of x and y. As we did in Section 9.7.5, we consider numerical flux functions p and q of the form

$$p^n_{j+1/2\,k} = p^n_{L_{j+1/2\,k}} + \phi^n_{x_{j\,k}}\left(p^n_{H_{j+1/2\,k}} - p^n_{L_{j+1/2\,k}}\right) \qquad (9.12.41)$$

$$q^n_{j\,k+1/2} = q^n_{L_{j\,k+1/2}} + \phi^n_{y_{j\,k}}\left(q^n_{H_{j\,k+1/2}} - q^n_{L_{j\,k+1/2}}\right), \qquad (9.12.42)$$

where the L and H subscripts signify that those numerical flux functions are associated with low and high order difference schemes, respectively. Specifically, we will use the upwind numerical flux functions for p_L and q_L as the numerical flux functions for our low order scheme, (9.12.37)–(9.12.38). We should realize that we are really using the nonconstant coefficient version of (9.12.37)–(9.12.38) where we allow a and b to depend on x and y. As we have done so often before, we will use the numerical flux functions associated with the Lax-Wendroff scheme, (9.12.26)–(9.12.27), as the high order numerical flux functions. Again, we will use the version of numerical flux

functions (9.12.26)–(9.12.27) where we assume that a and b may depend on x and y.

We emphasize that at this time we have not yet determined ϕ_x and ϕ_y. As we suggested earlier, we will define ϕ_x and ϕ_y differently from the way they were defined before, trying to capture some of the two dimensional character of the problem. Specifically, we will define ϕ_x so that it depends on more than $u^n_{j^*k}$, $j^* = j-p,\ldots,j+q$ (with the analogous statement being true of ϕ_y). We try to define ϕ_x in this two dimensional way so that it might be better able to resolve the discontinuities in two dimensional problems. In Section 9.7.5 we talked about adding as much antidiffusive flux as possible without increasing the variation of the solution. In the Z-S scheme we add as much antidiffusive flux at each grid point (j,k) as possible so that u^{n+1}_{jk} will still be between two values $u^{\max}_{jk}$ and $u^{\min}_{jk}$, which we get to define.

We begin with the following definitions. We denote the solution due to using the low order scheme by u^L_{jk}, i.e.,

$$u^L_{jk} = u^n_{jk} - R_x\delta_{x-}p^n_{L_{j+1/2\,k}} - R_y\delta_{y-}q^n_{L_{j\,k+1/2}}. \tag{9.12.43}$$

We restrict $u^{\max}_{jk}$ and $u^{\min}_{jk}$ so that $u^{\min}_{jk} \le u^L_{jk} \le u^{\max}_{jk}$. We define the principal part of the antidiffusive flux by the functions

$$p^n_{a_{j+1/2\,k}} = p^n_{H_{j+1/2\,k}} - p^n_{L_{j+1/2\,k}} \tag{9.12.44}$$

$$q^n_{a_{j\,k+1/2}} = q^n_{H_{j\,k+1/2}} - q^n_{L_{j\,k+1/2}} \tag{9.12.45}$$

and the antidiffusive fluxes entering and exiting a grid point by

$$\begin{aligned} a^{in}_{jk} &= \max\left\{0.0, p^n_{a_{j-1/2\,k}}\right\} - \min\left\{0.0, p^n_{a_{j+1/2\,k}}\right\} \\ &\quad + \max\left\{0.0, q^n_{a_{j\,k-1/2}}\right\} - \min\left\{0.0, q^n_{a_{j\,k+1/2}}\right\} \end{aligned} \tag{9.12.46}$$

$$\begin{aligned} a^{out}_{jk} &= \max\left\{0.0, p^n_{a_{j+1/2\,k}}\right\} - \min\left\{0.0, p^n_{a_{j-1/2\,k}}\right\} \\ &\quad + \max\left\{0.0, q^n_{a_{j\,k+1/2}}\right\} - \min\left\{0.0, q^n_{a_{j\,k-1/2}}\right\}, \end{aligned} \tag{9.12.47}$$

respectively. And finally, we define

$$b^{\text{up}}_{jk} = \begin{cases} \min\left\{1.0, \left(u^{\max}_{jk} - u^L_{jk}\right)/a^{\text{in}}_{jk}\right\} & \text{if } a^{\text{in}}_{jk} > 0 \\ 0 & \text{if } a^{\text{in}}_{jk} = 0 \end{cases} \tag{9.12.48}$$

and

$$b^{\text{down}}_{jk} = \begin{cases} \min\left\{1.0, \left(u^L_{jk} - u^{\min}_{jk}\right)/a^{\text{out}}_{jk}\right\} & \text{if } a^{\text{out}}_{jk} > 0 \\ 0 & \text{if } a^{\text{out}}_{jk} = 0 \end{cases} \tag{9.12.49}$$

as the least upper bound and the greatest lower bound of the fraction that must multiply the principal part of the antidiffusive fluxes into and away

from the point (j,k) to guarantee that u_{jk}^{n+1} will not overshoot the interval $[u_{jk}^{\min}, u_{jk}^{\max}]$.

We then define

$$\phi_{x_{jk}}^n = \begin{cases} \min\left\{b_{j+1\,k}^{\text{up}}, b_{jk}^{\text{down}}\right\} & \text{when } p_{a_{j+1/2\,k}}^n \geq 0 \\ \min\left\{b_{jk}^{\text{up}}, b_{j+1\,k}^{\text{down}}\right\} & \text{when } p_{a_{j+1/2\,k}}^n < 0 \end{cases} \tag{9.12.50}$$

and

$$\phi_{y_{jk}}^n = \begin{cases} \min\left\{b_{j\,k+1}^{\text{up}}, b_{jk}^{\text{down}}\right\} & \text{when } q_{a_{j\,k+1/2}}^n \geq 0 \\ \min\left\{b_{jk}^{\text{up}}, b_{j\,k+1}^{\text{down}}\right\} & \text{when } q_{a_{j\,k+1/2}}^n < 0. \end{cases} \tag{9.12.51}$$

We are now done defining the difference scheme based on numerical flux functions given in the form of (9.12.41)–(9.12.42), except that we must first define $u^{\max}$ and $u^{\min}$. One such definition is to define

$$u_{jk}^{\max} = \max\left\{u_{j+1\,k}^n, u_{j\,k+1}^n, u_{j-1\,k}^n, u_{j\,k-1}^n, u_{jk}^n, u_{j+1\,k}^L, u_{j\,k+1}^L, u_{j-1\,k}^L, u_{j\,k-1}^L, u_{jk}^L\right\} \tag{9.12.52}$$

and

$$u_{jk}^{\min} = \min\left\{u_{j+1\,k}^n, u_{j\,k+1}^n, u_{j-1\,k}^n, u_{j\,k-1}^n, u_{jk}^n, u_{j+1\,k}^L, u_{j\,k+1}^L, u_{j-1\,k}^L, u_{j\,k-1}^L, u_{jk}^L\right\}. \tag{9.12.53}$$

We see that this definition of $u^{\max}$ and $u^{\min}$ makes the limiter functions two dimensional. Of course, there are other definitions of $u^{\min}$ and $u^{\max}$ that can be used. In ref. [77], Zalesak mentions that using (9.12.52) and (9.12.53) to define $u^{\max}$ and $u^{\min}$ is better than just taking the maximum over the u^L values. One definition that can be useful is to include the corners of the stencil in the definition of $u_{jk}^{\max}$ and $u_{jk}^{\min}$, i.e., include $u_{j\pm1\,k\pm1}^L$ and $u_{j\pm1\,k\pm1}^n$. Such a definition will stop wiggles from forming in the diagonal direction from the grid point. This definition has no effect on the solution to initial–boundary–value problem (9.12.39)–(9.12.40).

Remark 1: The implication is that now that we have ϕ_x and ϕ_y defined, we can use (9.12.44)–(9.12.45) to define p and q. We really do not have to go through the trouble of building p and q. We can write

$$u_{jk}^{n+1} = u_{jk}^L - R_x\delta_{x-}\phi_{x_{jk}}^n p_{a_{j+1/2\,k}}^n - R_y\delta_{y-}\phi_{y_{jk}}^n q_{a_{j\,k+1/2}}^n. \tag{9.12.54}$$

Remark 2: In ref. [77], Zalesak included the additional restriction that

$$p_{a_{j+1/2\,k}}^n = 0 \quad \text{if } p_{a_{j+1/2\,k}}^n\left(u_{j+1\,k}^L - u_{jk}^L\right) < 0 \tag{9.12.55}$$

and either

$$p^n_{a_{j+1/2\,k}}\left(u^L_{j+2\,k}-u^L_{j+1\,k}\right)<0 \text{ or } p^n_{a_{j+1/2\,k}}\left(u^L_{j\,k}-u^L_{j-1\,k}\right)<0,$$

with an analogous restriction for $q^n_{j\,k+1/2}$. He states that the effect is minimal and is cosmetic in nature. For this reason, we have ignored the restriction.

In Figures 9.12.9 and 9.12.10 we give a plot of the solution and a contour plot of the solution to initial–boundary–value problem (9.12.39)–(9.12.40) obtained using the Z-S scheme described above. We note that though not perfect, the scheme gives much better results than those given in Figures 9.12.6 and 9.12.8.

We should understand that we can also develop a split Z-S scheme. If we consider a difference scheme of the form (9.12.22)–(9.12.23), define p just as we do above, and define $q^{n+1/2}_{j\,k+1/2}$ as we do for the Z-S scheme with $u^n_{j\,k}$ replaced by $u^{n+1/2}_{j\,k}$, i.e., use $q^{n+1/2}_{L_{j\,k+1/2}}$, $q^{n+1/2}_{H_{j\,k+1/2}}$, $q^{n+1/2}_{a_{j\,k}}$, u^L, $u^{\min}$, $u^{\max}$, a^{in}, a^{out}, b^{up} and b^{down} defined at the half step and $\phi^{n+1/2}_{y_{j\,k}}$, then we have a split Z-S scheme. In Figure 9.12.11 we plot the result of using the split Z-S scheme to approximate the solution to initial–boundary–value problem (9.12.39)–(9.12.40) (using zero Dirichlet and numerical boundary conditions). We note that the solution looks as if it might be smoother than the solution plotted in Figure 9.12.9. To be better able to compare the two solutions, in Figure 9.12.12 we include a contour plot of the solution plotted in Figure 9.12.11. Comparing and contrasting Figures 9.12.10 and 9.12.12, we see that though the flat top part of the solution given in Figure 9.12.11 might be smoother than that given in Figure 9.12.9, it is very clear that the discontinuity in the approximate solution given in Figure 9.12.11 is smeared more than that in the solution given in Figure 9.12.9.

We first mention that there are several ways to build the split Z-S scheme. We have chosen one such way, not claiming that it is the best way. Researchers sometimes choose nonsplit schemes over the split schemes, claiming that the split schemes have a directional bias. Split schemes do have a directional bias, as do the standard nonsplit schemes. There are other considerations to be taken into account. Specifically, when we compare the Z-S scheme with the split Z-S scheme, there is good reason to prefer the split scheme (though we admit that the computations here do not make that clear). When ϕ_x and ϕ_y are approximately one, the Z-S scheme reduces to a scheme that in HW5.8.4 we pointed was not the two dimensional Lax-Wendroff scheme and was not second order accurate. Part of the rationale in building flux-limiter schemes was to build a scheme that reduced to a high order scheme when the scheme did not need dissipation for smoothing. The notation used, p_H and q_H, was deceptive in that the subscript H is supposed to be attached to the numerical flux function of a high order scheme. One the other hand, when ϕ_x and ϕ_y are approximately one, the

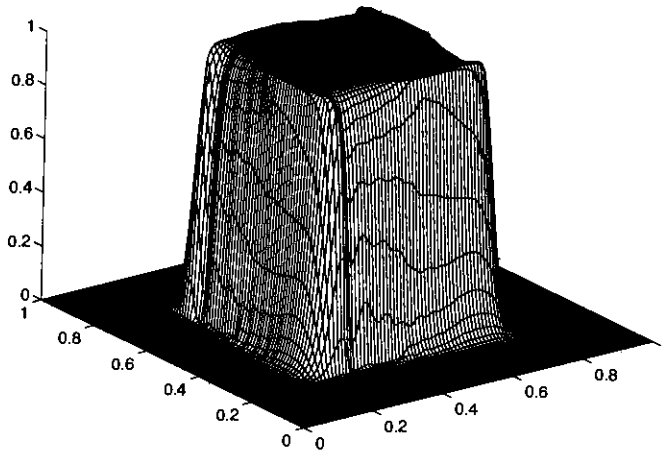

FIGURE 9.12.9. An approximation to the solution of initial–boundary–value problem (9.12.39)–(9.12.40) obtained using the two dimensional Z-S scheme defined by (9.12.8), (9.12.41)–(9.12.42), (9.12.50)–(9.12.51), zero Dirichlet numerical boundary conditions, $a(x,y) = \Omega\left(y - \frac{1}{2}\right)$, $b(x,y) = \Omega\left(x - \frac{1}{2}\right)$, $\Omega = \sqrt{2}$, $\Delta x = \Delta y = 0.01$ and $\Delta t = 0.002$. The solution is plotted at time $t = 1.11$, which is approximately $t = \pi/(2\Omega)$.

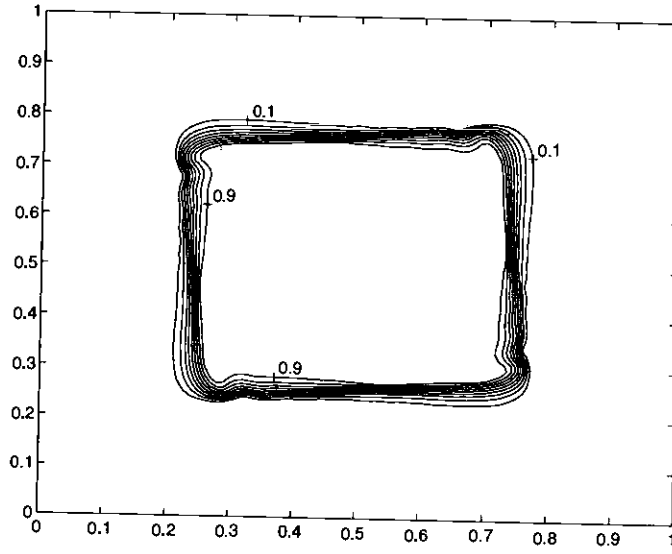

FIGURE 9.12.10. A contour plot of the solution given in Figure 9.12.9.

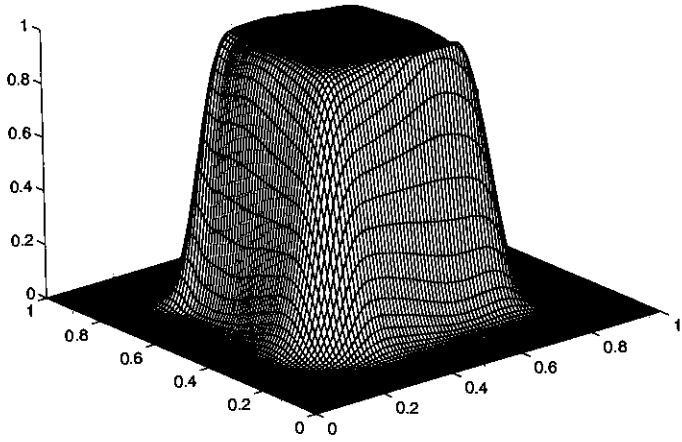

FIGURE 9.12.11. An approximation to the solution of initial–boundary–value problem (9.12.39)–(9.12.40) obtained using the two dimensional split Z-S scheme, zero Dirichlet numerical boundary conditions, $a(x,y) = \Omega\big(y - \frac{1}{2}\big)$, $b(x,y) = \Omega\big(x - \frac{1}{2}\big)$, $\Omega = \sqrt{2}$, $\Delta x = \Delta y = 0.01$ and $\Delta t = 0.002$. The solution is plotted at time $t = 1.11$, which is approximately $t = \pi/\big(2\Omega\big)$.

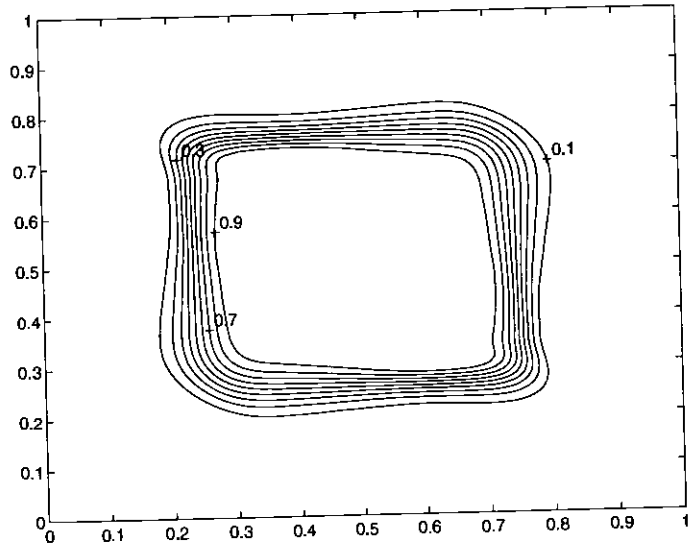

FIGURE 9.12.12. A contour plot of the solution plotted in Figure 9.12.11.

split Z-S scheme reduces to the two dimensional Lax-Wendroff scheme, a second order scheme.

HW 9.12.1 (a) Compute the characteristics associated with partial differential equation (9.12.39) and use these characteristics to determine where it is allowable to place boundary conditions on the domain $[0,1] \times [0,1]$.
(b) Show that

$$v(x,y,t) = v_0\Big(\frac{1}{2} + \Big(x - \frac{1}{2}\Big)\cos(\Omega t) + \Big(y - \frac{1}{2}\Big)\sin(\Omega t),$$
$$\frac{1}{2} - \Big(x - \frac{1}{2}\Big)\sin(\Omega t) + \Big(y - \frac{1}{2}\Big)\cos(\Omega t)\Big)$$

is a solution of partial differential equation (9.12.39), initial-condition $v(x,y,0) = v_0(x,y)$, which is zero on the boundary of $[0,1] \times [0,1]$.

HW 9.12.2 Consider the initial–boundary–value problem

$$v_t + a(x,y)v_x + b(x,y)v_y = 0 \quad (x,y) \in (0,1) \times (0,1),\ t > 0 \tag{9.12.56}$$

$$v(x,y,0) = \begin{cases} 1 & \text{if } \Big(x - \frac{1}{3}\Big)^2 + \Big(y - \frac{1}{3}\Big)^2 \le \frac{1}{36} \\ 0 & \text{otherwise} \end{cases} \tag{9.12.57}$$

with appropriate boundary conditions (see HW9.12.1 for a discussion of the boundary conditions for this problem) where $a(x,y) = -\Omega\Big(y - \frac{1}{2}\Big)$, $b(x,y) = \Omega\Big(x - \frac{1}{2}\Big)$ and $\Omega = \sqrt{2}$.
(a) Use the Z-S scheme, $\Delta x = \Delta y = 0.01$, $\Delta t = 0.002$ along with zero Dirichlet numerical boundary conditions to find an approximate solution to the above initial–boundary–value problem. Plot the solution at times $t = \pi/(2\Omega)$ and $t = 2\pi/\Omega$.
(b) Solve the problem described in part (a) using the split Z-S scheme. Compare and contrast the solutions found in both parts.

9.12.4 A Z-S Scheme for Nonlinear Conservation Laws

After the development of the Z-S and the split Z-S schemes given in Section 9.12.3, we have a reasonably obvious candidate for a two dimensional scheme for nonlinear conservation laws. We consider a conservation law of the form (9.12.3) and will refer to the schemes developed in this section as the nonlinear Z-S scheme and the nonlinear split Z-S scheme.

We return to the basic forms of conservative (and approximately conservative) difference schemes (9.12.8) and (9.12.22)–(9.12.23). We shall develop nonlinear schemes that are based on numerical flux functions of the

form (9.12.41)–(9.12.42). We begin by choosing the numerical flux functions p and q associated with nonlinear low and high order schemes. As we have done so often, we choose the upwind scheme as our low order scheme. We might be tempted to use the upwind numerical flux functions (9.12.16)–(9.12.17). It should be fairly clear that we will have the same problems with the two dimensional upwind scheme that we had with the one dimensional upwind scheme. To be safe, we must use upwind flux functions that include the sonic rarefaction fix from Section 9.9.4.2. We should understand that the numerical flux functions associated with the rarefaction fix from Section 9.9.4.2 depended on the solution to the Godonuv scheme, and we are not claiming that we have any such results for two dimensions. We are merely using the results as fluxes across the boundaries.

We set $\alpha_{x_{jk}} = \delta_{x+}u^n_{jk}$, $\alpha_{y_{jk}} = \delta_{y+}u^n_{jk}$, $\nu_{x_{jk}} = F'\big((u^n_{jk} + u^n_{j+1\,k})/2\big)$, $\nu_{y_{jk}} = G'\big((u^n_{jk} + u^n_{jk+1})/2\big)$, $\lambda^x_{jk_L} = F'\big(u^n_{jk}\big)$, $\lambda^x_{jk_R} = F'\big(u^n_{j+1\,k}\big)$, $\lambda^y_{jk_L} = G'\big(u^n_{jk}\big)$, and $\lambda^y_{jk_R} = G'\big(u^n_{jk+1}\big)$. Then, analogously to numerical flux function (9.9.81), we define

$$p^n_{L_{jk}} = \begin{cases} F^n_{jk} + \lambda^x_{jk_L}\dfrac{\lambda^x_{jk_R} - \nu_{x_{jk}}}{\lambda^x_{jk_R} - \lambda^x_{jk_L}}\alpha_{x_{jk}} & \text{when } \lambda^x_{jk_L} < 0 < \lambda^x_{jk_R} \\ \text{otherwise} & \\ F^n_{jk} + \alpha_{x_{jk}}\nu_{x_{jk}} & \text{when } \nu_{x_{jk}} \le 0 \\ F^n_{jk} & \text{when } \nu_{x_{jk}} > 0 \end{cases} \tag{9.12.58}$$

$$q^n_{L_{jk}} = \begin{cases} G^n_{jk} + \lambda^y_{jk_L}\dfrac{\lambda^y_{jk_R} - \nu_{y_{jk}}}{\lambda^y_{jk_R} - \lambda^y_{jk_L}}\alpha_{y_{jk}} & \text{when } \lambda^y_{jk_L} < 0 < \lambda^y_{jk_R} \\ \text{otherwise} & \\ G^n_{jk} + \alpha_{y_{jk}}\nu_{y_{jk}} & \text{when } \nu_{y_{jk}} \le 0 \\ G^n_{jk} & \text{when } \nu_{y_{jk}} > 0 \end{cases} \tag{9.12.59}$$

As we have done before, we use the numerical flux functions associated with the Lax-Wendroff scheme (9.12.16)–(9.12.17) as our high order numerical flux functions. We do not ignore (or forget) the fact that when ϕ_x and ϕ_y are near 1, the resulting split scheme will be only first order accurate.

Having made these choices, we can use equations (9.12.43)–(9.12.53) to define ϕ_x, ϕ_y and, consequently, u^{n+1}_{jk} and the nonlinear Z-S scheme. Of course, the nonlinear split Z-S scheme follows from the nonlinear Z-S scheme in the same way that the linear split Z-S scheme followed from the linear Z-S scheme. We probably do not need to emphasize the fact that when ϕ_x and ϕ_y are near 1, the split Z-S scheme will reduce to the true two dimensional Lax-Wendroff scheme.

9.12.5 Two Dimensional K-System Conservation Laws

In the previous sections we have developed some two dimensional schemes for both linear and nonlinear scalar conservation laws and hinted that some of them might be high resolution schemes. Of course, we should now follow with schemes for two dimensional K-system conservation laws. We will do very little of this. We mention that the approach that we used to build two dimensional scalar schemes will generally work for building schemes for two dimensional K-system conservation laws. Difference scheme (9.12.8) was developed in the setting that included nonlinear K-system conservation laws. Hence, if we can develop acceptable numerical flux functions, we will have schemes. And very much as in the scalar case, we can obtain some reasonably acceptable numerical flux functions by extending the one dimensional numerical flux functions to two dimensions. There is one very big difference between the scalar and system cases. We might recall that in order to obtain the two dimensional Lax-Wendroff schemes for linear systems (linear K-system conservation laws) in Example 6.7.2, we assumed that the matrices A_1 and A_2 were simultaneously diagonalizable. It should not surprise us that we might have the same problem if we try to use difference scheme (9.12.22)–(9.12.23). However, there is a cure to this problem. In ref. [64], Strang proves that if (9.12.22)–(9.12.23) is replaced by

$$\mathbf{u}_{j\,k}^{n+1/3} = \mathbf{u}_{j\,k}^{n} - \frac{R_x}{2}\delta_{x-}\mathbf{p}_{j+1/2\,k}^{n} \tag{9.12.60}$$

$$\mathbf{u}_{j\,k}^{n+2/3} = \mathbf{u}_{j\,k}^{n+1/3} - R_y\delta_{y-}\mathbf{q}_{j\,k+1/2}^{n+1/3} \tag{9.12.61}$$

$$\mathbf{u}_{j\,k}^{n+1} = \mathbf{u}_{j\,k}^{n+2/3} - \frac{R_x}{2}\delta_{x-}\mathbf{p}_{j+1/2\,k}^{n+2/3}, \tag{9.12.62}$$

and the numerical flux functions are associated with second order accurate schemes, then difference scheme (9.12.60)–(9.12.62) will be second order accurate. Thus we can use our one dimensional numerical flux functions to help generate difference schemes for two dimensional K-system conservation laws just as we did in the scalar case. We will not do that. We will either leave it to the reader to do or leave it to the reader's imagination. It should not surprise us that a scheme such as difference scheme (9.12.60)–(9.12.62) will have a directional bias. If we attempt to approximate the solution of a problem involving systems of equations that is analogous to initial–boundary–value problem (9.12.39)–(9.12.40) with fluxes that are very one dimensional, we will have problems. There are other ways to address this problem, but the method that is consistent with our approach for scalar problems is to build numerical flux functions that are more two dimensional in nature. Of course, it would be possible to use a scheme that is mildly two dimensional such as the Z-S scheme. Or it may be necessary to develop numerical flux functions that are very two dimensional.

Remark: At first glance, it might appear that using difference scheme (9.12.60)–(9.12.62) will cause significantly more work. However, that is not

the case. The last half step in x at any time step can be done together with the first half time step with respect to x for the next time step. In this manner, only a slight amount of extra work is necessary when output at a particular time step is desired.

10
Elliptic Equations

10.1 Introduction

One of the most common classes of partial differential equations is the class of elliptic partial differential equations. We have not delayed the discussion of elliptic equations because we do not feel that they are important. We have done so only to keep the topics on parabolic and hyperbolic equations together (because so many of the topics related to these types of equations are similar). Elliptic partial differential equations are fundamentally different from parabolic and hyperbolic partial differential equations, and the numerical schemes for approximating solutions to elliptic partial differential equations are fundamentally different from the schemes for approximating solutions to parabolic and hyperbolic partial differential equations. The class of problems involving elliptic partial differential equations is a very important class of problems. Simulations of steady heat flows or irrotational flows of an inviscid, incompressible fluid; pressure computations for either the flow through a porous medium or that associated with the flow of a viscous, incompressible fluid; and many others all involve solving elliptic equations.

Consider an operator of the form

$$Lv = \sum_{p,q=1}^{K} a^{pq}(\mathbf{x}) D_{pq} v + \sum_{p=1}^{K} b^{p}(\mathbf{x}) D_{p} v + c(\mathbf{x}) v, \quad a^{pq} = a^{qp}, \quad (10.1.1)$$

where $\mathbf{x} = (x_1, \ldots, x_K) \in R \subset \mathbb{R}^K$, $K \geq 2$.

Definition 10.1.1 *L is* **elliptic at a point** $\mathbf{x} \in R^K$ *if*

$$0 < \lambda(\mathbf{x})|\boldsymbol{\xi}|^2 \leq \sum_{p,q=1}^{K} a^{p\,q}(\mathbf{x})\xi_p\xi_q \leq \Lambda(\mathbf{x})|\boldsymbol{\xi}|^2$$

for all $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_K) \in \mathbb{R}^K - \{\boldsymbol{\theta}\}$, *where* $\lambda(\mathbf{x})$ *and* $\Lambda(\mathbf{x})$ *denote the smallest and largest eigenvalues of the matrix* $[a^{p\,q}(\mathbf{x})]$, *respectively. If* $\lambda(\mathbf{x}) > 0$ *for all* $\mathbf{x} \in R$, *then* L **is elliptic on** R.

We will most often consider constant coefficient elliptic equations, i.e., we will consider $a^{p\,q}$, $p, q = 1, \ldots, K$, b^p, $p = 1, \ldots, K$, and c to be constant. The most common model equation of a linear elliptic partial differential equation is the Poisson equation

$$-\nabla^2 v = F, \tag{10.1.2}$$

where v and F are defined on a subset of $\mathbb{R}^2$ or $\mathbb{R}^3$ and

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \quad \text{and} \quad \nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2},$$

depending on whether our domain is two or three dimensional (called Laplace's equation when $F = 0$). The operator ∇^2 is referred to as "del squared" or "nabla squared," depending on your background. Two other elliptic equations that we shall consider and use as model equations are

$$av_{xx} + cv_{yy} + dv_x + ev_y + fv = F, \tag{10.1.3}$$

where we assume that $a, c < 0$ (so that partial differential equation (10.1.3) is elliptic) and the self-adjoint elliptic equation

$$(\alpha v_x)_x + (\beta v_y)_y - fv = F, \tag{10.1.4}$$

where $\alpha,\ \beta < 0$. We will develop numerical schemes for approximating the solutions to equations (10.1.2), (10.1.3) and (10.1.4) with boundary conditions. As is usually the case, our methods will generally extend nicely to nonconstant coefficient problems and will sometimes be applicable to nonlinear problems. The approach taken in this chapter will be to give solution algorithms, do some analysis when it helps explain the algorithm, and state some theorems (and maybe prove some). The material that will be covered will include

- defining the associated discrete problem and showing that this problem is uniquely solvable (Sections 10.2 and 10.6);
- proving that the solution of the discrete problem converges to the solution of the partial differential equation (Sections 10.3 and 10.6);

and

- developing algorithms for solving the discrete problems and discussing the convergence of these algorithms (most of the rest of the chapter).

The first step will involve approximating the elliptic boundary–value problems by difference equations and showing that the resulting discrete equations have a unique solution. The second step involves determining how well the solution to the discrete problem approximates the solution to the continuous problem, i.e. what happens as Δx and Δy approach zero. For parabolic and hyperbolic equations, the convergence was handled by the Lax Theorem, Theorem 2.5.2, or the Gustafsson result of Section 8.4.3. When we see that the difference equations for elliptic partial differential equations are basically different from those for parabolic and hyperbolic partial differential equations (we miss the time derivative), it will be fairly clear that the methods considered before will not work for elliptic partial differential equations. We will prove that as Δx and Δy approach zero, the solution of our discrete problem converges to the solution of our continuous problem.

The last step involves the study of the methods of solution of the discrete problem. We never considered this aspect for schemes for parabolic and hyperbolic equations. We always found a relatively easy (and some times a very easy) direct method for solving the difference equations and acted as if these solutions were exact. In actuality the stability of our schemes took care of any round-off error we did introduce. For elliptic equations, we will generally use some iterative method for solving the discrete problem rather than a direct method. We must consider whether or not we are solving the discrete problem sufficiently accurately and how accurately should we try to solve the discrete problem. The theory associated with the numerical schemes for solving the difference equations associated with elliptic partial differential equations fits better into a course in numerical analysis or numerical linear algebra. Hence, proofs for many of these methods will not be included, and we generally refer the reader to the books [22], [13], [31] and [76].

10.2 Solvability of Elliptic Difference Equations: Dirichlet Boundary Conditions

We begin by considering a problem that will serve as our basic model problem

$$-\nabla^2 v = F, \quad (x,y) \in R = (0,1) \times (0,1) \tag{10.2.1}$$

$$v = f, \quad (x,y) \in \partial R. \tag{10.2.2}$$

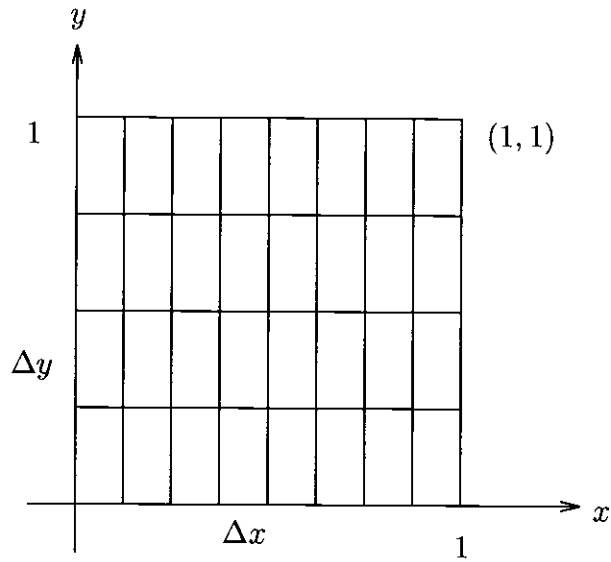

FIGURE 10.1.1. Two dimensional grid on the region $[0,1] \times [0,1]$.

As we did in Chapter 4 when we considered two dimensional problems, we consider a grid G_R as shown in Figure 10.1.1 and approximate problem (10.2.1)–(10.2.2) by

$$-\frac{1}{\Delta x^2}\delta_x^2 u_{jk} - \frac{1}{\Delta y^2}\delta_y^2 u_{jk} = F_{jk}, \quad j = 1, \ldots, M_x - 1,$$
$$k = 1, \ldots, M_y - 1 \tag{10.2.3}$$
$$u_{0k} = f_{0k}, \quad k = 1, \ldots, M_y - 1 \tag{10.2.4}$$
$$u_{M_x k} = f_{M_x k}, \quad k = 1, \ldots, M_y - 1 \tag{10.2.5}$$
$$u_{j0} = f_{j0}, \quad j = 1, \ldots, M_x - 1 \tag{10.2.6}$$
$$u_{jM_y} = f_{jM_y}, \quad j = 1, \ldots, M_x - 1. \tag{10.2.7}$$

Our first step in the consideration of equations (10.2.3)–(10.2.7) is to decide whether or not equations (10.2.3)–(10.2.7) have a solution, and if they do, whether it is unique. We begin by considering a matrix equation of the form

$$A\mathbf{x} = \mathbf{f}$$

where $A = [a_{jk}]_{L \times L}$, $\mathbf{x} = [x_1 \ \cdots \ x_L]^T$ and $\mathbf{f} = [b_1 \ \cdots \ b_L]^T$. We discuss two methods that can be used to ensure that equations (10.2.3)–(10.2.7) have a unique solution. The first result involves the assumption that matrix A is positive definite. We say that A **is positive definite** if $\mathbf{x}^T A\mathbf{x} > 0$ for all $\mathbf{x} \neq 0$. We might add that one of the characterizations of a positive definite matrix is that *A is positive definite if and only if A is symmetric*

and all of the eigenvalues of A are greater than zero. ([31], page 1001.) We then state the following result.

Proposition 10.2.1 *If A is positive definite, then A is invertible.*

Proof: This proposition follows directly from the characterization of positive definite matrices given above. If all of the eigenvalues of A are positive, then since the determinant of A is the product of the eigenvalues, the determinant of A is nonzero and A is invertible.

We note that if we order the unknowns associated with problem (10.2.3)–(10.2.7) in lexicographical order as

$$[u_{11} \cdots u_{M_x-1\,1}\ u_{12} \cdots u_{M_x-1\,M_y-1}]^T$$

(assuming that all of the Dirichlet boundary conditions have been included into the right hand side), the matrix that must be solved is of the form

$$A = \begin{pmatrix} B & -\frac{1}{\Delta y^2}I & \Theta & \cdots & \\ -\frac{1}{\Delta y^2}I & B & -\frac{1}{\Delta y^2}I & \Theta & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ & \cdots & \Theta & -\frac{1}{\Delta y^2}I & B \end{pmatrix}_{(M_y-1)\times(M_y-1)} \tag{10.2.8}$$

where B is the $(M_x-1)\times(M_x-1)$ matrix

$$\begin{pmatrix} 2(\frac{1}{\Delta x^2}+\frac{1}{\Delta y^2}) & -\frac{1}{\Delta x^2} & 0 & \cdots & \\ -\frac{1}{\Delta x^2} & 2(\frac{1}{\Delta x^2}+\frac{1}{\Delta y^2}) & -\frac{1}{\Delta x^2} & 0 & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ & \cdots & 0 & \frac{1}{\Delta x^2} & 2(\frac{1}{\Delta x^2}+\frac{1}{\Delta y^2}) \end{pmatrix}. \tag{10.2.9}$$

As we mentioned in Section 4.3.2, the reader should inspect the form of matrix A above carefully and probably write out a version of it in order to understand what it looks like. The interesting part of the matrix A that is most easily lost from the form given in (10.2.8)–(10.2.9) are the scattered zeros along the super and subdiagonals due to the boundary conditions at $x = 0$ and $x = 1$. In Figure 10.2.1 we give a specific case of the matrix A with $M_x = 5$ and $M_y = 6$. The boldface zeros given in Figure 10.2.1 represent zeros that result from the scheme reaching in the x-direction to boundary points from the $(M_x-1,1)$, $(1,2),\ldots,$ (M_x-1,M_y-2), and $(1,M_y-1)$ grid points.

It is not difficult to see that A is symmetric (we must look at the part of the matrix near the extra zeros carefully). To see that A is positive definite is much more difficult. It is possible to try to show that A is positive definite using the definition or several other of the available characterizations of a positive definite matrix. However, since we are essentially going to compute the eigenvalues of A in Example 10.5.1 anyway, we refer the reader to that example to note that the eigenvalues of A are all positive. Hence, A is

$$
\left(\begin{array}{cccc|cccc|cccc|cccc|cccc}
d & b & 0 & 0 & a & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
b & d & b & 0 & 0 & a & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & b & d & b & 0 & 0 & a & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & b & d & \mathbf{0} & 0 & 0 & a & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
a & 0 & 0 & \mathbf{0} & d & b & 0 & 0 & a & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & a & 0 & 0 & b & d & b & 0 & 0 & a & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & a & 0 & 0 & b & d & b & 0 & 0 & a & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & a & 0 & 0 & b & d & \mathbf{0} & 0 & 0 & a & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & a & 0 & 0 & \mathbf{0} & d & b & 0 & 0 & a & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & a & 0 & 0 & b & d & b & 0 & 0 & a & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & a & 0 & 0 & b & d & b & 0 & 0 & a & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & a & 0 & 0 & b & d & \mathbf{0} & 0 & 0 & a & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & a & 0 & 0 & \mathbf{0} & d & b & 0 & 0 & a & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & a & 0 & 0 & b & d & b & 0 & 0 & a & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & a & 0 & 0 & b & d & b & 0 & 0 & a & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & a & 0 & 0 & b & d & \mathbf{0} & 0 & 0 & a \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & a & 0 & 0 & \mathbf{0} & d & b & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & a & 0 & 0 & b & d & b & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & a & 0 & 0 & b & d & b \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & a & 0 & 0 & b & d
\end{array}\right)
$$

FIGURE 10.2.1. The form of the matrix A described above in (10.2.8)–(10.2.9) with $M_x = 5$, $M_y = 6$, $a = -1/\Delta y^2$, $b = -1/\Delta x^2$ and $d = 2(1/\Delta x^2 + 1/\Delta y^2)$. The boldface zeros represent the zeros that appear because the scheme is reaching for boundary conditions.

positive definite and we can apply Proposition 10.2.1 to see that problem (10.2.3)–(10.2.7) is uniquely solvable.

Another useful approach to proving that equations (10.2.3)–(10.2.7) have a unique solution is to use the diagonal dominance of the matrix. We make the following definition.

Definition 10.2.2 *A is* **diagonally dominant** (**strictly diagonally dominant**) *if*

$$|a_{jj}| \geq \sum_{\substack{k=1 \\ k \neq j}}^{L} |a_{jk}| = \rho_j \quad \textit{for all } j = 1, \ldots, L$$

$$\left(|a_{jj}| > \sum_{\substack{k=1 \\ k \neq j}}^{L} |a_{jk}| = \rho_j \quad \textit{for all } j = 1, \ldots, L\right).$$

We then obtain the following result (the proof of which can be found in [31], page 302).

Proposition 10.2.3 *If A is strictly diagonally dominant, then A is invertible.*

The good news is that the above result is very nice (and the proof is easy). Consider partial differential equation (10.1.3) where a, $c < 0$ and $f > 0$, and difference scheme

$$\frac{a}{\Delta x^2}\delta_x^2 u_{jk} + \frac{c}{\Delta y^2}\delta_y^2 u_{jk} + \frac{d}{2\Delta x}\delta_{x0} u_{jk} + \frac{e}{2\Delta x}\delta_{y0} u_{jk} + f u_{jk} = F_{jk}, \tag{10.2.10}$$

along with the appropriate boundary conditions. The term on the diagonal is

$$-\frac{2a}{\Delta x^2} - \frac{2c}{\Delta y^2} + f, \tag{10.2.11}$$

and the sum of the absolute values of the terms off of the diagonal is

$$\left|\frac{a}{\Delta x^2} + \frac{d}{2\Delta x}\right| + \left|\frac{a}{\Delta x^2} - \frac{d}{2\Delta x}\right| + \left|\frac{c}{\Delta y^2} + \frac{e}{2\Delta y}\right| + \left|\frac{c}{\Delta y^2} - \frac{e}{2\Delta y}\right|. \tag{10.2.12}$$

If Δx and Δy are taken to be sufficiently small so that

$$0 < \Delta x < \frac{-2a}{|d|} \quad \text{and} \quad 0 < \Delta y < \frac{-2c}{|e|},$$

then the terms in expression (10.2.12) involving a and c will dominate, and expression (10.2.12) can be written as (remembering that a and c are < 0)

$$-\frac{a}{\Delta x^2} - \frac{d}{2\Delta x} - \frac{a}{\Delta x^2} + \frac{d}{2\Delta x} - \frac{c}{\Delta y^2} - \frac{e}{2\Delta y} - \frac{c}{\Delta y^2} + \frac{e}{2\Delta y}.$$
$$= -\frac{2a}{\Delta x^2} - \frac{2c}{\Delta y^2}. \tag{10.2.13}$$

If we compare expression (10.2.11) with (10.2.13), we see that the associated matrix will be strictly diagonally dominant (remember that $f > 0$), and hence the coefficient matrix will be invertible, and the problem will be uniquely solvable.

Remark: We note that if we allow the coefficients of equation (10.1.3) to depend on x and y and consider the nonconstant analogue of difference scheme (10.2.10) (with each a, c, etc. replaced by an a_{jk}, c_{jk}, etc.), the same analysis as is used above will again apply. In this case, if we require that Δx and Δy be chosen such that $0 < \Delta x < -2a_{jk}/|d_{jk}|$ and $0 < \Delta y < -2c_{jk}/|e_{jk}|$ for all j and k, the associated matrix will again be strictly diagonally dominant, and the equation will be uniquely solvable.

The bad news is that we cannot use Proposition 10.2.3 on such a basic problem as that associated with matrix (10.2.8)–(10.2.9) (our model problem). In matrix (10.2.8)–(10.2.9), though many of the rows are strictly diagonally dominant, a large number (rows associated with the interior points) are only diagonally dominant. If we return to the analysis done above for difference scheme (10.2.10), we see that the assumption $f > 0$ (instead of allowing $f = 0$) is very important. For this reason we need a stronger result (one that is not nearly as nice). We make the following definition.

Definition 10.2.4 *The $L \times L$ matrix A is* **reducible** *if either*
(a) $L = 1$ *and* $A = \Theta$, *or*
(b) $L \geq 2$ *and there is an $L \times L$ permutation matrix P and some integer r, $1 \leq r \leq L-1$ such that*

$$P^T A P = \begin{pmatrix} B & C \\ \Theta & D \end{pmatrix}$$

where B is $r \times r$, D is $(L-r) \times (L-r)$, C is $r \times (L-r)$, and Θ is the $(L-r) \times r$ zero matrix.

The matrix A is **irreducible** *if it is not reducible.*

We understand that the above definition is not very palatable. Let us assure you that most of the matrices obtained from finite difference equations are irreducible. A convenient characterization of an irreducible matrix A can be given in the context of the system of equations $A\mathbf{u} = \mathbf{f}$. *A matrix A is irreducible if a change in any of the components of* $\mathbf{f}$ *will cause a change*

in the solution **u**. To obtain a reducible matrix in a finite difference setting one would have to either be solving a problem that can be separated into two problems or be using a very strange difference scheme that does not reach to some of the points in the grid.

The result that we then want concerning the solvability of matrix equations is as follows.

Proposition 10.2.5 *If A is an irreducible diagonally dominant matrix for which $|a_{jj}| > \rho_j$ for at least one j, then A is invertible.*

Proof: See [31], page 356 or [76], page 1004.

It is not hard (taking our word for the fact that the matrix will be irreducible) to see that matrix (10.2.8)–(10.2.9) will satisfy Proposition 10.2.5. The row (or rows) that satisfy the hypothesis "for which $|a_{jj} > \rho_j$ for at least one j" are usually the rows associated with points that reach to one of the boundary conditions. Hence, problem (10.2.3)–(10.2.7) is uniquely solvable.

We now return to difference scheme (10.2.10) with the assumption that $a, c < 0$ and $f = 0$ and accept the fact that the associated matrix is irreducible. Then using the same analysis used earlier (except now $f = 0$), we see that for sufficiently small Δx and Δy the associated matrix is diagonally dominant, the inequality will be strict on the rows associated with boundary conditions, and we can apply Proposition 10.2.5 to get that difference scheme (10.2.10) along with Dirichlet boundary conditions is uniquely solvable.

Remark: We should note that in the same way that we were able to extend the application of Proposition 10.2.3 to include the nonconstant coefficient case, the arguments that apply to equation (10.1.3) and difference equation (10.2.10) (along with appropriate boundary conditions) will still apply if $a, b, \ldots, f$ are functions of x and y. The only new condition that must be met is that Δx and Δy must be chosen sufficiently small so as to satisfy $0 < \Delta x < -2a_{jk}/|d_{jk}|$ and $0 < \Delta y < -2c_{jk}/|e_{jk}|$ for all j and k.

10.3 Convergence of Elliptic Difference Schemes: Dirichlet Boundary Conditions

Now that we know that problem (10.2.3)–(10.2.7) has a unique solution, we must decide whether or not the solution to discrete problem (10.2.3)–(10.2.7) will provide a good approximation to the solution to our continuous problem (10.2.1)–(10.2.2). It is easy to use the same Taylor expansions used earlier to show that problem (10.2.3)–(10.2.7) is a $\mathcal{O}(\Delta x^2) + \mathcal{O}(\Delta y^2)$ approximation to problem (10.2.1)–(10.2.2). Below we include three theorems by which we are able to prove that the solution of problem (10.2.3)–(10.2.7)

will also be a $\mathcal{O}(\Delta x^2) + \mathcal{O}(\Delta y^2)$ approximation to the solution of problem (10.2.1)–(10.2.2).

Before we proceed with our results, we include some notation. We let v be the solution to problem (10.2.1)–(10.2.2) and $\mathbf{u} = \{u_{jk}\}$, $= 0, \ldots, M_x$, $k = 0, \ldots, M_y$ be the solution to problem (10.2.3)–(10.2.7). As we did before, we denote the grid on R by G_R and let G_R^0 and ∂G_R denote the grid points in the interior of R and on the boundary of R, respectively. We define the sup-norms of grid functions defined on G_R, G_R^0 and ∂G_R as

$$\|\mathbf{u}\|_\infty = \max_{\substack{0 \le j \le M_x \\ 0 \le k \le M_y}} |u_{jk}| \tag{10.3.1}$$

$$\|\mathbf{u}\|_{\infty 0} = \max_{\substack{1 \le j \le M_x - 1 \\ 1 \le k \le M_y - 1}} |u_{jk}| \tag{10.3.2}$$

$$\|\mathbf{u}\|_{\infty \partial G_R} = \max_{(j,k) \in \partial G_R} |u_{jk}|. \tag{10.3.3}$$

We next include the statement and proof of the discrete maximum principle (of course, the analogous analytic result can be found in almost any textbook on partial differential equations).

Proposition 10.3.1 Discrete Maximum Principle *If*

$$L_{jk}u_{jk} = -\Big(\frac{1}{\Delta x^2}\delta_x^2 + \frac{1}{\Delta y^2}\delta_y^2\Big)u_{jk} \le 0$$

$(L_{jk}u_{jk} \ge 0)$ on G_R^0, then the maximum (minimum) value of u_{jk} on G_R is attained on ∂G_R.

Proof: We begin by showing that u_{jk} cannot have a local maximum in G_R^0. To do this we note that the condition $L_{jk}u_{jk} \le 0$ is equivalent to

$$\Big(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}\Big)u_{jk} \le \frac{1}{2}\Big[\frac{1}{\Delta x^2}(u_{j+1\,k} + u_{j-1\,k}) + \frac{1}{\Delta y^2}(u_{j\,k+1} + u_{j\,k-1})\Big]. \tag{10.3.4}$$

We now suppose that u_{jk} is a local maximum, i.e.,

$$u_{jk} \ge u_{j+1\,k},\ u_{jk} \ge u_{j-1\,k},\ u_{jk} \ge u_{j\,k+1},\ \text{and}\ u_{jk} \ge u_{j\,k-1}.$$

We first use the fact that

$$u_{jk} \ge u_{j-1\,k},\ u_{jk} \ge u_{j\,k+1},\ \text{and}\ u_{jk} \ge u_{j\,k-1}$$

in inequality (10.3.4) and then the fact that $u_{jk} \ge u_{j+1\,k}$ to get the following series of inequalities.

$$\Big(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}\Big)u_{jk} \le \frac{1}{2}\Big[\frac{1}{\Delta x^2}u_{j+1\,k} + \frac{1}{\Delta x^2}u_{jk} + \frac{2}{\Delta y^2}u_{jk}\Big] \tag{10.3.5}$$

$$\le \Big(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}\Big)u_{jk} \tag{10.3.6}$$

Since the left hand side and the end result are the same, we know that all the inequalities must be equalities. Using the fact that (10.3.5) is an equality, we are left with the equation

$$\frac{1}{\Delta x^2}u_{jk} = \frac{1}{2}\Big[\frac{1}{\Delta x^2}u_{j+1\,k} + \frac{1}{\Delta x^2}u_{jk}\Big],$$

or

$$u_{j+1\,k} = u_{jk}.$$

In the same manner, we can apply the appropriate inequalities to see that

$$u_{jk} = u_{j+1\,k} = u_{j-1\,k} = u_{j\,k+1} = u_{j\,k-1}.$$

Hence, u_{jk} can not have a local maximum on G_R^0, and the maximum of u_{jk} on G_R must occur on ∂G_R.

The proof of the discrete minimum principle follows in exactly the same manner.

We next state and prove a discrete regularity result.

Proposition 10.3.2 *Suppose that u_{jk}, $j = 0, \ldots, M_x$, $k = 0, \ldots, M_y$, is a function defined on G_R with $u_{0k} = u_{M_x k} = u_{j0} = u_{j M_y} = 0$. Then*

$$\|\mathbf{u}\|_{\infty} \leq \frac{1}{8}\|L_{jk}u_{jk}\|_{\infty 0}. \tag{10.3.7}$$

Proof: Consider a grid function u_{jk} defined on G_R that satisfies $u_{jk} = 0$ on ∂G_R. Define the grid function G_{jk} on G_R^0 by

$$G_{jk} = L_{jk}u_{jk}.$$

Obviously,

$$-\|\mathbf{G}\|_{\infty 0} \leq L_{jk}u_{jk} \leq \|\mathbf{G}\|_{\infty 0}. \tag{10.3.8}$$

Define

$$w_{jk} = \frac{1}{4}\left[\left(x_j - \frac{1}{2}\right)^2 + \left(y_k - \frac{1}{2}\right)^2\right]$$

and note that

$$L_{jk}w_{jk} = -1.$$

Then by inequality (10.3.8) we get

$$\begin{aligned} L_{jk}(u_{jk} - \|\mathbf{G}\|_{\infty 0}w_{jk}) &= L_{jk}u_{jk} - \|\mathbf{G}\|_{\infty 0}L_{jk}w_{jk} \\ &= L_{jk}u_{jk} + \|\mathbf{G}\|_{\infty 0} \geq 0, \end{aligned}$$

and likewise,

$$L_{jk}(u_{jk} + \|\mathbf{G}\|_{\infty 0}w_{jk}) \leq 0.$$

Thus the minimum of $\mathbf{u} - \|\mathbf{G}\|_{\infty 0}\mathbf{w}$ and the maximum of $\mathbf{u} + \|\mathbf{G}\|_{\infty 0}\mathbf{w}$ occur on the boundary, ∂G_R. Then

$$\begin{aligned}\max_{\partial G_R}\left[\mathbf{u} + \|\mathbf{G}\|_{\infty 0}\mathbf{w}\right] &= \|\mathbf{G}\|_{\infty 0}\|\mathbf{w}\|_{\infty \partial R} \quad (\text{since } u_{jk} = 0 \text{ on } \partial G_R)\\ &\geq u_{jk} + \|\mathbf{G}\|_{\infty 0} w_{jk} \quad (\text{since the maximum of}\\ &\qquad\qquad \mathbf{u} + \|\mathbf{G}\|_{\infty 0}\mathbf{w} \text{ occurs on the boundary})\\ &\geq u_{jk} \quad (\text{since } \|\mathbf{G}\|_{\infty 0} w_{jk} \geq 0)\end{aligned}$$

for any $(j,k) \in G_R$ and

$$\min_{\partial G_R}\left[\mathbf{u} - \|\mathbf{G}\|_{\infty 0}\mathbf{w}\right] = -\|\mathbf{G}\|_{\infty 0}\|\mathbf{w}\|_{\infty \partial R} \leq u_{jk} - \|\mathbf{G}\|_{\infty 0} w_{jk} \leq u_{jk},$$

for any $(j,k) \in G_R$, or (remember that $u_{jk} = 0$ on ∂G_R)

$$-\|\mathbf{G}\|_{\infty 0}\|\mathbf{w}\|_{\infty \partial R} \leq u_{jk} \leq \|\mathbf{G}\|_{\infty 0}\|\mathbf{w}\|_{\infty \partial R}.$$

Since $\|\mathbf{w}\|_{\infty \partial R} = \frac{1}{8}$,

$$\|\mathbf{u}\|_\infty \leq \frac{1}{8}\|L_{jk}u_{jk}\|_{\infty 0},$$

which is what we were to prove.

We next define $\|\partial^4 v\|_{\infty 0}$ to be

$$\|\partial^4 v\|_{\infty 0} = \sup\left\{\left|\frac{\partial^4 v}{\partial x^p \partial y^q}(x,y)\right| : (x,y) \in R^0,\right.$$
$$\left. p+q=4,\ p,q=0,\ldots,4\right\} \tag{10.3.9}$$

(R^0 is the interior of R) and use Propositions 10.3.1 and 10.3.2 to prove the following convergence theorem.

Theorem 10.3.3 *Let $v \in C^4(\overline{R})$ be a solution to problem (10.2.1)–(10.2.2) and u_{jk} a solution to problem (10.2.3)–(10.2.7). Then*

$$\|\mathbf{v} - \mathbf{u}\|_\infty \leq C(\Delta x^2 + \Delta y^2)\|\partial^4 v\|_{\infty 0}. \tag{10.3.10}$$

Proof: We begin by noting that the term $\|\partial^4 v\|_{\infty 0}$ in inequality (10.3.10) is the maximum of all fourth derivatives evaluated in the interior of the domain R. We let v be a solution to partial differential equation (10.2.1)–(10.2.2). As usual with our consistency arguments, we note that v satisfies

$$L_{jk}v_{jk} = F_{jk} + \mathcal{O}(\Delta x^2) + \mathcal{O}(\Delta y^2), \tag{10.3.11}$$

where the constant that is present in the definition of the $\mathcal{O}$ term can be bounded by $C\|\partial^4 v\|_{\infty 0}$ for some constant C. Since u_{jk} satisfies

$$L_{jk}u_{jk} = F_{jk}, \tag{10.3.12}$$

we can take the difference of equations (10.3.11) and (10.3.12) to get

$$L_{j\,k}(v_{j\,k} - u_{j\,k}) = \mathcal{O}(\Delta x^2) + \mathcal{O}(\Delta y^2)$$

or

$$\|\{L_{j\,k}(v_{j\,k} - u_{j\,k})\}\|_{\infty\,0} \leq C(\Delta x^2 + \Delta y^2)\|\partial^4 v\|_{\infty\,0}$$

and $v_{j\,k} - u_{j\,k} = 0$ on ∂G_R. Then by Proposition 10.3.2 we have that

$$\|\mathbf{v} - \mathbf{u}\|_\infty \leq \frac{C}{8}(\Delta x^2 + \Delta y^2)\|\partial^4 v\|_{\infty\,0}.$$

Remark 1: The hypothesis that the solution to problem (10.2.1)–(10.2.2), v, be in C^4 is the assumption that the solution has four derivatives that are continuous. These theorems can be proved with weaker hypotheses, but we do not think that this is necessary for our purposes.

Remark 2: Reviewing the proofs of the above three theorems, we see that the proofs extend directly to the analogous three dimensional result. We might think that since the analogous analytic problem with Neumann boundary conditions on all boundaries is not well posed (the solution is determined only up to an additive constant), it is not necessary to consider whether the solution to the obvious discrete problem converges to the solution of the analytic problem. However, we want to solve this problem, and we will address it later in Section 10.6.

Remark 3: Suppose instead of equations (10.2.1) and (10.2.3) we consider the elliptic partial differential equation (10.1.3) (with boundary condition (10.2.2)) along with difference scheme (10.2.10) (with discrete boundary conditions (10.2.4)–(10.2.7)). We saw at the end of Section 10.2 that for sufficiently small Δx and Δy this difference problem is uniquely solvable. Using straightforward extensions of the proofs given for Propositions 10.3.1 and 10.3.2 and Theorem 10.3.3, analogous results can be proved for elliptic boundary–value problem (10.1.3), (10.2.2) and difference scheme (10.2.10), (10.2.4)–(10.2.7). See HW10.3.1 and/or [76], page 971. Hence, for sufficiently small Δx and Δy,

$$\|\mathbf{v} - \mathbf{u}\|_\infty = \mathcal{O}(\Delta x^2) + \mathcal{O}(\Delta y^2),$$

where $\mathbf{v}$ is the solution to equation (10.1.3) and boundary condition (10.2.2), and $\mathbf{u}$ is a solution to difference scheme (10.2.10) and boundary conditions (10.2.4)–(10.2.7).

HW 10.3.1 Show that we can obtain results analogous to those of Propositions 10.3.1 and 10.3.2 and Theorem 10.3.3 for the elliptic boundary–value problem (10.1.3), (10.2.2) and difference scheme (10.2.10), (10.2.4)–(10.2.7).

10.4 Solution Schemes for Elliptic Difference Equations: Introduction

We see that as Δx and Δy approach zero, the solution to the difference equation problem approaches the solution to the elliptic boundary–value problem. In fact, we see that it approaches it at a rate proportional to Δx^2 and Δy^2. We now consider how to solve equations (10.2.3)–(10.2.7). Again, we consider equations (10.2.3)–(10.2.7) as a system of equations of the form

$$A\mathbf{u} = \mathbf{f}, \tag{10.4.1}$$

where again the variables are ordered in lexicographical order,

$$\mathbf{u} = [u_{11}\ u_{21}\ \cdots\ u_{M_x-1\,1}\ u_{12}\ \cdots\ u_{M_x-1\,M_y-1}]^T,$$

and $\mathbf{f}$ depends on F and, in the appropriate rows, f (when the scheme reaches to the boundary). As we showed earlier, matrix A will be broadly banded just as those discussed in Section 4.4 when we were considering two dimensional implicit schemes for parabolic problems. The situation here is nicer than that in Section 4.4 in that we have to solve equation (10.4.1) once, whereas in Section 4.4 an equation like equation (10.4.1) had to be solved at each time step. However, we do not want to be so wasteful as trying to solve equation (10.4.1) by Gaussian reduction. Often, when we are faced with an elliptic equation, we are in a situation where the equation must be solved many times or where M_x and M_y (and M_z in the case of three dimensional problems) are very large.

One of our first thoughts might be to try to approach the problem as we did in Section 4.4 and use some sort of ADI scheme. A little thought shows that this is not as easy as we might first think. The difficulty is that for elliptic partial differential equations we do not have the time derivative term. Later, in Section 10.12, we do show that it is possible to artifically insert a time derivative term into the equation and use an ADI scheme to solve equations (10.2.3)–(10.2.7). Instead, we will introduce several iterative methods for solving equation (10.4.1). In the next section, we begin discussing relaxation methods. The class of relaxation methods includes a group of methods that are very popular and successful for elliptic problems. We then include the conjugate gradient method and give a short introduction to the multigrid method.

10.5 Residual Correction Methods

We begin our discussion of ways to solve equation (10.4.1) efficiently by introducing a class of methods referred to as residual correction methods. We let $\mathbf{w}$ denote an approximation to $\mathbf{u}$, the solution of equation (10.4.1),

and denote the **algebraic error** by $\mathbf{e} = \mathbf{u} - \mathbf{w}$ and the **residual error** by $\mathbf{r} = \mathbf{f} - A\mathbf{w}$. At different times we will measure both the algebraic and residual errors with respect to either the sup-norm or the ℓ_2 norm defined on R^L, where $L = (M_x - 1)(M_y - 1)$. It is easy to see that the algebraic and residual errors are related by the **residual equation**

$$A\mathbf{e} = A(\mathbf{u} - \mathbf{w}) = \mathbf{f} - A\mathbf{w} = \mathbf{r}. \tag{10.5.1}$$

From equation (10.5.1) we obtain the **correction equation** given by

$$\mathbf{u} = \mathbf{w} + \mathbf{e} = \mathbf{w} + A^{-1}\mathbf{r}. \tag{10.5.2}$$

Thus for a given approximate solution $\mathbf{w}$, an obvious approach to solving equation (10.4.1) is to compute $\mathbf{r}$ and use equation (10.5.2) to obtain $\mathbf{u}$. Clearly, if we could compute $A^{-1}\mathbf{r}$, we could solve equation (10.4.1) directly. The **residual correction method** is to approximate A^{-1} and define the iteration

$$\mathbf{w}_{k+1} = \mathbf{w}_k + B\mathbf{r}_k, \tag{10.5.3}$$

where $\mathbf{r}_k = \mathbf{f} - A\mathbf{w}_k$ and B is some approximation to A^{-1}. For example, if we let $B = I$, we get

- the **Richardson iterative scheme**

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \mathbf{r}_k. \tag{10.5.4}$$

Decompose the matrix A into $A = L+D+U$ where L is the lower triangular matrix consisting of the elements of A below the diagonal, D is a diagonal matrix consisting of the diagonal of A, and U is an upper triangular matrix consisting of the elements of A above the diagonal. Then, if we choose

- $B = D^{-1}$, we get the **Jacobi relaxation scheme**;

if we choose

- $B = (L + D)^{-1}$, we get the **Gauss-Seidel relaxation scheme**;

if we choose

- $B = \omega(I + \omega D^{-1}L)^{-1}D^{-1} = \omega(D + \omega L)^{-1}$, we get the **successive overrelaxation scheme**, where ω is a free parameter;

and if we choose

- $B = \omega(2 - \omega)(D + \omega U)^{-1}D(D + \omega L)^{-1}$, we get the **symmetric successive overrelaxation scheme**, where ω is a free parameter.

It should not surprise us that a large number of iterative schemes can be described as residual correction schemes by the appropriate definition of B.

10.5.1 Analysis of Residual Correction Schemes

The approach that is used often to obtain an approximate solution to equation (10.4.1) is to choose an initial guess, $\mathbf{w}_0$, and use the iterative scheme (10.5.3) (with one of the choices for B). Obviously, the sequence $\{\mathbf{w}_k\}$ will not converge to the solution of equation (10.4.1) for all choices of B and $\mathbf{w}_0$. To make a convergence analysis easier, we note that we can use equation (10.5.1) to rewrite the residual correction scheme as

$$\mathbf{w}_{k+1} = \mathbf{w}_k + B\mathbf{r}_k = \mathbf{w}_k + BA\mathbf{e}_k \tag{10.5.5}$$

where $\mathbf{e}_k = \mathbf{u} - \mathbf{w}_k$. If we multiply equation (10.5.5) by -1 and add $\mathbf{u}$, we get

$$\mathbf{e}_{k+1} = \mathbf{e}_k - BA\mathbf{e}_k = (I - BA)\mathbf{e}_k. \tag{10.5.6}$$

The matrix $R = I - BA$ is called the **error propagation matrix** or the **iteration matrix**. Since

$$\|\mathbf{e}_{k+1}\| \leq \|I - BA\|\|\mathbf{e}_k\| \leq \cdots \leq \|I - BA\|^{k+1}\|\mathbf{e}_0\|,$$

the term $\|R\| = \|I - BA\|$ is referred to as the **convergence factor**. We obtain the following result.

Proposition 10.5.1 *If $\|R\| < 1$, the sequence of approximate solutions to equation (10.4.1), $\{\mathbf{w}_k\}$, will converge in norm $\|\cdot\|$ to the solution to equation (10.4.1), $\mathbf{u}$, for an arbitrary initial guess $\mathbf{w}_0$.*

If we use the sup-norm, it is not difficult to see that

$$\|R\|_\infty = \sup_{\|\mathbf{x}\|_\infty = 1} \|R\mathbf{x}\|_\infty = \max_{1 \leq j \leq L} \Big\{ \sum_{k=1}^{L} |r_{jk}| \Big\},$$

where $R = [r_{jk}]_{L \times L}$ ([31], page 295). Though easy to compute, we shall see that the matrix sup-norm is not very useful for proving convergence.

As we saw earlier, it is generally difficult to compute the norm of a matrix. We recall that $\sigma(R)$, the spectral radius of R, is defined to be

$$\sigma(R) = \max\{|\lambda| : \lambda \text{ an eigenvalue of } R\}.$$

If we consider the ℓ_2 norm and if R is symmetric, we can use the fact that $\|R\|_2 = \sigma(R)$ to determine the convergence factor by computing the eigenvalues of R. As we see below in Proposition 10.5.2, if we compute the eigenvalues of R, we know that the scheme converges without any reference to the norm of R.

Proposition 10.5.2 *The residual correction scheme converges for any initial choice $\mathbf{w}_0$ if and only if $\sigma(R) < 1$.*

Proof: [31], page 298.

The result given in Proposition 10.5.2 is really stronger than the statement given in that we also get an indication of the speed of convergence based on the spectral radius. Consider an $L \times L$ matrix R that has a full independent set of eigenvectors and one eigenvalue that is the largest in magnitude. Suppose that the eigenvalues are ordered such that $|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_L|$, and write the independent set of L eigenvectors as $\mathbf{x}_1, \ldots, \mathbf{x}_L$. We can then write our original error, $\mathbf{e}_0$, in terms of the basis of eigenvectors of A as

$$\mathbf{e}_0 = \sum_{j=1}^{L} a_j \mathbf{x}_j \tag{10.5.7}$$

and note that

$$\mathbf{e}_{k+1} = R\mathbf{e}_k = R^{k+1}\mathbf{e}_0 = \sum_{j=1}^{L} a_j \lambda_j^{k+1} \mathbf{x}_j. \tag{10.5.8}$$

Equation (10.5.8) can be used to write $\mathbf{e}_{k+1}$ as

$$\mathbf{e}_{k+1} = \lambda_1^{k+1}\left[a_1\mathbf{x}_1 + \sum_{j=2}^{L} a_j \left(\frac{\lambda_j}{\lambda_1}\right)^{k+1} \mathbf{x}_j\right]. \tag{10.5.9}$$

Since $|\lambda_j/\lambda_1|^{k+1} \to 0$ as $k \to \infty$, it is clear that eventually the convergence is determined by $|\lambda_1^{k+1}|$. We notice that for large k, $\mathbf{e}_{k+1} \approx a_1\lambda_1^{k+1}\mathbf{x}_1$ and $\|\mathbf{e}_{k+m}\|/\|\mathbf{e}_k\| \approx |\lambda_1|^m$. To reduce the error by a factor of

$$\zeta = \|\mathbf{e}_{k+m}\|/\|\mathbf{e}_k\|,$$

we must iterate approximately $m = \log \zeta / \log |\lambda_1|$ times.

Specifically, to reduce the error one more decimal place, we must iterate approximately $m = \log(0.1)/\log|\lambda_1|$ times. Since the factors ζ by which we wish to reduce our error will generally be given in terms of 10^{-q} (say, 10^{-1}), it is clearly better to use logarithms base 10 here. We then have $m = -1/\log_{10}|\lambda_1|$.

We note that when $|\lambda_1| \approx 1$, m gets large. For example, to reduce the error one decimal place, we must iterate

$$m = \frac{-1}{\log_{10}|\lambda_1|} \approx \begin{cases} 4 & \text{if } |\lambda_1| = 0.5 \\ 22 & \text{if } |\lambda_1| = 0.9 \\ 230 & \text{if } |\lambda_1| = 0.99 \\ 2302 & \text{if } |\lambda_1| = 0.999 \end{cases}$$

times.

The asymptotic analysis done above is more general than is indicated. The result given is based on the assumption that A has a full, independent

set of eigenvectors and that one eigenvalue is larger in magnitude than the others. These assumptions make the analysis given above easy and clear, but they are not needed. Define the **average rate of convergence** to be

$$R_k(R) = -\frac{1}{k}\log\|R^k\|.$$

In ref. [31], page 299, it is proved that

$$\sigma(R) = \lim_{k\to\infty}\|R^k\|^{1/k}.$$

We see that

$$\begin{aligned}\log\sigma(R) &= \log\Big(\lim_{k\to\infty}\|R^k\|^{1/k}\Big)\\ &= \lim_{k\to\infty}\frac{1}{k}\log\|R^k\|\\ &= -\lim_{k\to\infty}R_k(R).\end{aligned}$$

Hence, we define $R_\infty(R) = \lim_{k\to\infty} R_k(R) = -\log\sigma(R)$ to be the **asymptotic rate of convergence**. If $\sigma(R) < 1$, the number of iterations, k, needed to reduce the norm of the initial error vector by a factor of ζ can be approximated by $k \approx -\log\zeta/R_\infty(R)$. We note that in the case described earlier where λ_1 is the eigenvector largest in magnitude,

$$R_\infty(R) = -\log\sigma(R) = -\log|\lambda_1|.$$

In this case, we find that the number of iterations needed to reduce the initial error vector by a factor of ζ is given by

$$k \approx -\log\zeta/R_\infty(R) = -\log\zeta/\log|\lambda_1|,$$

which is what we found earlier.

The above analysis shows that the convergence of the residual correction scheme is dominated by one of its eigenvalues (and other eigenvalues such that $|\lambda_j/\lambda_1| < 1$ but $|\lambda_j/\lambda_1| \approx 1$). Most of the work performed in a residual correction calculation will be to eliminate the error in the component of the eigenvector associated with these "largest eigenvalues." In Section 10.10 we will take advantage of this knowledge as we introduce the multigrid method.

10.5.2 Jacobi Relaxation Scheme

Earlier, we noted that by choosing $B = D^{-1}$ in the residual correction scheme, we obtained the Jacobi relaxation scheme

$$\begin{aligned}\mathbf{w}_{k+1} &= \mathbf{w}_k + D^{-1}\mathbf{r}_k &(10.5.10)\\ &= \mathbf{w}_k + D^{-1}(\mathbf{f} - (L+D+U)\mathbf{w}_k) &(10.5.11)\\ &= D^{-1}\mathbf{f} - D^{-1}(L+U)\mathbf{w}_k. &(10.5.12)\end{aligned}$$

Consider an equation $A\mathbf{u} = \mathbf{f}$ that results from solving a general two dimensional difference equation of the form

$$\beta^1_{j\,k}u_{j+1\,k} + \beta^2_{j\,k}u_{j-1\,k} + \beta^3_{j\,k}u_{j\,k+1} + \beta^4_{j\,k}u_{j\,k-1} - \beta^0_{j\,k}u_{j\,k} = F_{j\,k} \quad (10.5.13)$$

along with Dirichlet boundary conditions (where the $j\,k$ subscript on the β terms imply that the coefficients may be a function of x and y). We should understand that this general difference equation includes both difference equations (10.2.3) and (10.2.10) as well as many other specific difference equations. If we again order the unknowns $u_{j\,k}$ in the j–k ordering used earlier, then the $(M_x - 1)(M_y - 1) \times (M_x - 1)(M_y - 1)$ lower triangular matrix L is given by

$$\begin{pmatrix}
0 & \cdots & & & & & & & \\
\beta^2_{2\,1} & 0 & \cdots & & & & & & \\
0 & \beta^2_{3\,1} & 0 & \cdots & & & & & \\
 & & \ddots & \ddots & \ddots & & & & \\
\beta^4_{1\,2} & 0 & \cdots & 0 & \beta^2_{1\,2} & 0 & \cdots & & \\
 & & \ddots & \ddots & & & \ddots & \ddots & \\
 & \cdots & 0 & \beta^4_{M_x-1\,M_y-1} & 0 & \cdots & 0 & \beta^2_{M_x-1\,M_y-1} & 0
\end{pmatrix} \quad (10.5.14)$$

where $\beta^2_{1\,2},\ \beta^2_{1\,3}, \ldots, \beta^2_{1\,M_y-1}$ are zero due to the fact that when

$$(j,k) = (1,2),(1,3),\ldots,(1,M_y-1),$$

difference scheme (10.5.13) reaches to the boundary instead of the next value given in the $\mathbf{u}$ vector. The $(M_x - 1)(M_y - 1) \times (M_x - 1)(M_y - 1)$ diagonal matrix D is given by

$$\begin{pmatrix}
-\beta^0_{1\,1} & 0 & \cdots & & & & & \\
0 & -\beta^0_{2\,1} & 0 & \cdots & & & & \\
 & \ddots & \ddots & \ddots & & & & \\
 & \cdots & 0 & -\beta^0_{j\,k} & 0 & \cdots & & \\
 & & & & \ddots & \ddots & \ddots & \\
 & & & \cdots & 0 & -\beta^0_{M_x-2\,M_y-1} & 0 & \\
 & & & & \cdots & 0 & -\beta^0_{M_x-1\,M_y-1}
\end{pmatrix} \quad (10.5.15)$$

the $(M_x - 1)(M_y - 1) \times (M_x - 1)(M_y - 1)$ upper triangular matrix U is

given by

$$\begin{pmatrix} 0 & \beta^1_{1\,1} & 0 & \cdots & 0 & \beta^3_{1\,1} & 0 & \cdots \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \\ \cdots & 0 & \beta^1_{M_x-1\,M_y-2} & 0 & \cdots & 0 & \beta^3_{M_x-1\,M_y-2} & \\ & \cdots & 0 & \beta^1_{1\,M_y-1} & 0 & \cdots & & \\ & & & \ddots & \ddots & \ddots & & \\ & & & \cdots & 0 & \beta^1_{M_x-3\,M_y-1} & 0 & \\ & & & & \cdots & 0 & \beta^1_{M_x-2\,M_y-1} & \\ & & & & & \cdots & 0 & \end{pmatrix} \tag{10.5.16}$$

where $\beta^1_{M_x-1\,1}$, $\beta^1_{M_x-1\,2}, \cdots, \beta^1_{M_x-1\,M_y-2}$ are zero (again, difference equation (10.5.13) reaches to the boundary at these points) and **f** includes the contribution due to $F_{j\,k}$ and the boundary conditions for the appropriate terms. *Note specifically that the β's that we claim are zero following matrices (10.5.15) and (10.5.16) account for the fact that the equation at the $(M_x-1,1)$ point does not reach to the $(1,2)$ point, which is the next u value in the ordering; the equation at the $(1,2)$ point does not reach to the $(M_x-1,1)$ point, which was the previous u value in the ordering, etc.*

It is not difficult to see that the Jacobi relaxation scheme, (10.5.12), applied to solving equation (10.5.13) can be written as follows (where the Dirichlet boundary conditions are included by loading them into the $u_{j\,k}$ array for the appropriate $j=0$, $j=M_x$, $k=0$, and $k=M_y$).

Jacobi-10.5.13

For $k=1,\ldots,M_y-1$
For $j=1,\ldots,M_x-1$
$$u'_{j\,k} = -\tfrac{1}{\beta^0_{j\,k}}\left[F_{j\,k} - \beta^1_{j\,k}u_{j+1\,k} - \beta^2_{j\,k}u_{j-1\,k} - \beta^3_{j\,k}u_{j\,k+1} - \beta^4_{j\,k}u_{j\,k-1}\right]$$
Next j
Next k

The $-1/\beta^0_{j\,k}$ term is due to the D^{-1} matrix, the β^1 and β^3 terms are due to the U matrix, and the β^2 and β^4 terms are due to the L matrix. Very specifically, we note that if we apply the above algorithm to solve the difference equations associated with the Poisson equation, (10.2.3)–(10.2.7), the above algorithm can be written as

Jacobi-10.2.3

For $k=1,\ldots,M_y-1$
For $j=1,\ldots,M_x-1$
$$u'_{j\,k} = \tfrac{1}{d}\left[F_{j\,k} + \tfrac{1}{\Delta x^2}(u_{j+1\,k}+u_{j-1\,k}) + \tfrac{1}{\Delta y^2}(u_{j\,k+1}+u_{j\,k-1})\right]$$
Next j
Next k

where $d = 2(1/\Delta x^2 + 1/\Delta y^2)$. We note that both formulations given above make it clear that the Jacobi scheme consists of writing the equations that must be solved and replacing the four neighbors of the (j,k) point by "old" values. Hence, the Jacobi relaxation scheme can be viewed as replacing the (j,k) term by the adjusted average (adjusted by the $F_{j\,k}$) of its four neighbors.

10.5.3 Analysis of the Jacobi Relaxation Scheme

The easiest way to discuss convergence for the Jacobi scheme is to include a result that is true for the Jacobi relaxation scheme (and, as we shall see for the Gauss-Seidel relaxation scheme), but not generally true for other residual correction schemes.

Proposition 10.5.3 *If A is irreducible, diagonally dominant and, for at least one j,*

$$|a_{j\,j}| > \rho_j = \sum_{\substack{k=1 \\ k \neq j}}^{L} |a_{j\,k}|,$$

then the Jacobi iteration scheme converges for all $\mathbf{w}_0$ (where ρ_j is as defined in Definition 10.2.2).

Proof: ([76], page 1013) We note that the proof of Proposition 10.5.3 involves using the hypotheses to show that the spectral radius of the iteration matrix associated with the Jacobi scheme, R_J, is less than one (and then apply Proposition 10.5.2).

Remark 1: By inspection of matrix (10.2.8) (which we already did to see that equations (10.2.3)–(10.2.7) were uniquely solvable), it is easy to see that *the Jacobi iteration for our model problem (10.2.3)–(10.2.7) converges.*

Remark 2: Using the analysis that we did in Section 10.2, we see that *for sufficiently small Δx and Δy, the Jacobi iteration scheme for solving difference equation (10.2.10) along with boundary conditions (10.2.4)–(10.2.7) will converge.*

Remark 3: We should realize that the same result will apply to the nonconstant coefficient case when Δx and Δy are chosen correctly (as we did in Section 10.2 when we proved the unique solvability of the appropriate difference equation with nonconstant coefficients).

We note that we have been able to apply Proposition 10.5.3 to a large class of difference schemes. Proposition 10.5.3 is a strong result.

Remark 4: Reconsider the special case of difference equation (10.2.10) when $a, c < 0$ and $f > 0$ (with either constant or nonconstant coefficients) and the analysis done in Section 10.2 when we showed that the associated

matrix was strictly diagonally dominant. We should realize that this same computation shows that for sufficiently small Δx and Δy, the sup-norm of $R_J = -D^{-1}(L+U)$, $\|R_J\|_\infty$, will be strictly less than one, i.e.,

$$\|R_J\|_\infty = \| - D^{-1}(L+U)\|_\infty = \sup_{j,k} \frac{-\frac{2a_{jk}}{\Delta x^2} - \frac{2c_{jk}}{\Delta y^2}}{-\frac{2a_{jk}}{\Delta x^2} - \frac{2c_{jk}}{\Delta y^2} + f} < 1.$$

Hence, in this special case we can use the sup-norm and apply Proposition 10.5.1 to ensure that the Jacobi relaxation scheme will converge for this problem. In addition, we can use $\|R_J\|_\infty$ along with the asymptotic analysis done in Section 10.5.1 to provide information concerning the speed of convergence of the scheme.

Obviously, another way to prove the convergence of Jacobi relaxation for solving equations (10.2.3)–(10.2.7) is to return to Section 10.5.1 and apply Proposition 10.5.2 directly. Though we were able to ensure the convergence of the Jacobi scheme above using Proposition 10.5.3, we want to apply Proposition 10.5.2 so that we obtain the speed of convergence results discussed in Section 10.5.1. In addition, we will see that the calculation performed will be important for some of our later work. We see that the iteration matrix (error propagation matrix) associated with the Jacobi scheme is given by

$$R_J = I - D^{-1}A = I - D^{-1}(L+D+U) = -D^{-1}(L+U). \quad (10.5.17)$$

To apply Proposition 10.5.2, we must calculate the eigenvalues of the matrix R_J. Clearly, this is impossible to do for an equation as general as equation (10.5.13). To illustrate the convergence result, in Example 10.5.1 below we compute the eigenvalues of R_J associated with solving our model problem (10.2.3)–(10.2.7).

Example 10.5.1 Compute the eigenvalues of the Jacobi iteration matrix $R_J = -D^{-1}(L+U)$ associated with equations (10.2.3)–(10.2.7).

Solution: We must find λ and $\mathbf{X}$ that satisfies

$$R_J\mathbf{X} = \lambda\mathbf{X} \quad (10.5.18)$$

or

$$\frac{1}{d}\Big[\frac{1}{\Delta x^2}(X_{j+1\,k} + X_{j-1\,k}) + \frac{1}{\Delta y^2}(X_{j\,k+1} + X_{j\,k-1})\Big] = \lambda X_{j\,k}, \quad (10.5.19)$$

where $d = 2(1/\Delta x^2 + 1/\Delta y^2)$, and $X_{0\,k} = X_{M_x\,k} = X_{j\,0} = X_{j\,M_y} = 0$ for $j = 1, \ldots, M_x - 1$ and $k = 1, \ldots, M_y - 1$. We should emphasize the difference between equation (10.5.18) and difference equation (10.5.19)—the similarities should be pretty obvious. The matrix equations used to define D, L and U are over the interior of the region $R = [0,1] \times [0,1]$ and do not contain equations associated with the points $j = 0$ or $j = M_x$; or $k = 0$ or $k = M_y$. When a certain row wants to reach to one of these points, it reaches to a different value with a zero coefficient (which were always the boldfaced zeros when we wrote out the matrix). When we consider eigenvalue problem (10.5.18) as a different equation, it is most convenient to allow the difference operators to actually

reach to the boundaries. We allow this—without changing the equation—by defining $X_{0\,k} = X_{M_x\,k} = X_{j\,0} = X_{j\,M_y} = 0$ for $j = 1, \ldots, M_x - 1$ and $k = 1, \ldots, M_y - 1$.

There are numerous ways to find the eigenvalues of matrix R_J. One elementary method is to use separation of variables as we did in Chapters 3 and 7, and as we do for partial differential operators in most elementary courses on partial differential equations. We begin by letting $X_{j\,k} = x_j y_k$ and inserting this form of $X_{j\,k}$ into equation (10.5.19). We get

$$\frac{1}{d}\Big[\frac{1}{\Delta x^2}(x_{j+1}y_k + x_{j-1}y_k) + \frac{1}{\Delta y^2}(x_j y_{k+1} + x_j y_{k-1})\Big] = \lambda x_j y_k,$$

or, when divided by $x_j y_k$ and rearranged,

$$d\lambda - \frac{x_{j+1} + x_{j-1}}{x_j \Delta x^2} = \frac{y_{k+1} + y_{k-1}}{y_k \Delta y^2}. \tag{10.5.20}$$

We use the usual separation argument given for any separation of variable calculation, that the only way that the function on the left hand side (which is a function of j) can equal the function on the right hand side (which is a function of k) is that they both be constant, say μ. Hence, including the separated boundary conditions, we are left with the following pair of equations

$$y_{k+1} + y_{k-1} = \mu \Delta y^2 y_k, \quad k = 1, \ldots, M_y - 1 \tag{10.5.21}$$

$$y_0 = y_{M_y} = 0 \tag{10.5.22}$$

$$x_{j+1} + x_{j-1} = (d\lambda - \mu)\Delta x^2 x_j, \quad j = 1, \ldots, M_x - 1 \tag{10.5.23}$$

$$x_0 = x_{M_x} = 0. \tag{10.5.24}$$

Equation (10.5.21) along with boundary conditions (10.5.22) is equivalent to the eigenvalue problem

$$\begin{pmatrix} 0 & 1 & 0 & \cdots & \\ 1 & 0 & 1 & 0 & \cdots \\ & \ddots & \ddots & \ddots & \\ \cdots & 0 & 1 & 0 & 1 \\ & \cdots & 0 & 1 & 0 \end{pmatrix} \mathbf{Y} = \mu \Delta y^2 \mathbf{Y}. \tag{10.5.25}$$

Using equation (2.2.41), Part 1, we see that the eigenvalues of the matrix given in (10.5.25) are given by

$$\mu_s = \frac{2}{\Delta y^2} \cos \frac{s\pi}{M_y}, \quad s = 1, \ldots, M_y - 1.$$

If we now use μ_s in equation (10.5.23) (so we now have the equation

$$x_{j+1} + x_{j-1} = (d\lambda - \mu_s)\Delta x^2 x_j, \quad s = 1, \ldots, M_y - 1,$$

we see that equation (10.5.23) along with boundary conditions (10.5.24) is equivalent to the eigenvalue problem

$$\begin{pmatrix} 0 & 1 & 0 & \cdots & \\ 1 & 0 & 1 & 0 & \cdots \\ & \ddots & \ddots & \ddots & \\ \cdots & 0 & 1 & 0 & 1 \\ & \cdots & 0 & 1 & 0 \end{pmatrix} \mathbf{X} = \omega_s \mathbf{X}, \quad s = 1, \cdots, M_y - 1, \tag{10.5.26}$$

where $\omega_s = (d\lambda - \mu_s)\Delta x^2$. Again using formula (2.2.41), we know that equation (10.5.26) has $M_x - 1$ eigenvalues for each value of s, and they are given by

$$\omega_s^p = 2\cos\frac{p\pi}{M_x}, \quad p = 1, \ldots, M_x - 1. \tag{10.5.27}$$

Then, solving for λ_s^p (where the notation indicates the fact that λ will depend on both p and s), we get

$$\begin{aligned}\lambda_s^p &= \frac{1}{d}\Big(\frac{1}{\Delta x^2}\omega_s^p + \mu_s\Big) \\ &= \frac{2}{d}\Big(\frac{1}{\Delta x^2}\cos\frac{p\pi}{M_x} + \frac{1}{\Delta y^2}\cos\frac{s\pi}{M_y}\Big), \quad s = 1,\dots,M_y - 1, \\ &\qquad\qquad p = 1,\dots,M_x - 1. \end{aligned} \tag{10.5.28}$$

Clearly, the maximum of $|\lambda_s^p|$ occurs with $s = p = 1$ (and with $s = M_y - 1$ and $p = M_x - 1$), so that

$$\sigma(R_J) = \frac{2}{d}\Big(\frac{1}{\Delta x^2}\cos\frac{\pi}{M_x} + \frac{1}{\Delta y^2}\cos\frac{\pi}{M_y}\Big).$$

Since

$$\sigma(R_J) < \frac{2}{2\big(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}\big)}\Big(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}\Big) \le 1,$$

the Jacobi relaxation scheme converges when used to solve equations (10.2.3)–(10.2.7).

Remark 1: Based on the asymptotic analysis given in Section 10.5.1 we note that to improve our result by one decimal place ($\zeta = 10^{-1}$), we must perform $m = -1/\log_{10}\sigma(R_J)$ iterations. The number of iterations required for various grid sizes is given in Table 10.5.1. We should realize that this is an asymptotic result. Often, early iterations will do much better than the results indicated by Table 10.5.1.

M_x	M_y	$\sigma(R_J)$	$m = -1/\log_{10}\sigma(R_J)$
10	10	0.951057	46
100	100	0.999507	4666
1000	1000	0.999995	466,601
10	100	0.999027	2,365
50	100	0.999211	2,916
50	1000	0.999990	233,922

TABLE 10.5.1. Values of the spectral radius and number of iterations of the Jacobi iteration needed to improve a result by one decimal place for various values of M_x and M_y.

Remark 2: To be able to apply the rate of convergence results of Section 10.5.1, we must know that we have a full set of independent eigenvectors. At least in the special case where $\Delta x = \Delta y$ ($M_x = M_y = M$), it is clear that we have repeated eigenvalues. Specifically, $\lambda_{M-p}^p = 0$ for $p = 1,\dots,M-1$. Return to Section 2.2.3 and apply formula (2.2.42) to both matrix equations (10.5.25) and (10.5.26) to obtain the eigenvectors $\mathbf{Y}$ and $\mathbf{X}$ where $y_k = \sin ks\pi/M$ and $x_j = \sin jp\pi/M$. Thus the eigenvector associated with eigenvalue λ_s^p for problem (10.5.19) is given by

$$\mathbf{u}^{p\,s} = [x_1y_1 \ \cdots \ x_{M-1}y_1 \ x_1y_2 \ \cdots \ x_{M-1}y_2 \ x_1y_3 \ \cdots \ x_{M-1}y_{M-1}]^T$$

for $p = 1, \ldots, M-1$ and $s = 1, \ldots, M-1$. Hence, we see that we do have M^2 independent eigenvectors, so the results given in Section 10.5.1 apply. We also recall that the results related to the asymptotic rate of convergence given in Section 10.5.1 (near the end of Section 10.5.1) do not require that we have independent eigenvectors. These results would apply even with or without the assumption that $\Delta x = \Delta y$ and the calculation of the eigenvectors given above.

Remark 3: In our discussion in Section 10.2 concerning the application of Proposition 10.2.1 to show that the difference equations associated with our model problem (10.2.3)–(10.2.7) are uniquely solvable, we mentioned that we would "essentially" compute the eigenvalues of the matrix A in Example 10.5.1. If we note that the Jacobi iteration matrix R_J and the matrix A are related by $A = d(I - R_J)$, then the eigenvalues of A, λ_A, are related to the eigenvalues of R_J, μ, by $\mu = 1 - \lambda_A/d$ (since $A\mathbf{u} = d(I - R_J)\mathbf{u} = \lambda_A\mathbf{u}$ implies that $R_J\mathbf{u} = ((d - \lambda_A)/d)\mathbf{u}$). Hence, using formula (10.5.28), we see that the eigenvalues of A are

$$\lambda_{A_{sp}} = d - 2\Big(\frac{1}{\Delta x^2}\cos\frac{s\pi}{M_x} + \frac{1}{\Delta y^2}\cos\frac{s\pi}{M_y}\Big), \quad p = 1, \ldots, M_x - 1, \quad s = 1, \ldots, M_y - 1, \quad (10.5.29)$$

which are all positive. We should realize that if we want to find the eigenvalues of A, we could also apply separation of variables directly to the equation $A\mathbf{X} = \lambda_A\mathbf{X}$.

10.5.4 Stopping Criteria

When we consider iterative schemes, it is not enough to describe an iteration of the scheme and analyze the scheme. We must discuss how we will decide to stop iterating. The discussion given here will apply to the Jacobi relaxation scheme that we have just developed and other residual correction schemes that follow. In this section we introduce six of the most common methods used as stopping criteria. Most often, the choice of which method one uses is based on the experience, likes and dislikes of the user.

Error Bounds: We consider solving a problem of the form

$$A\mathbf{u} = \mathbf{f}$$

using an iteration that produces a sequence of iterates $\{\mathbf{w}_k\}$ and would like to be able to measure the error. Since this is usually impossible, we will discuss methods for establishing a bound on the error, $\mathbf{e}_k = \mathbf{u} - \mathbf{w}_k$. One of the most obvious approaches is to measure the rate of convergence as

$$\|\mathbf{w}_{k+1} - \mathbf{w}_k\|. \quad (10.5.30)$$

Another common approach is to consider the residual error,

$$\|\mathbf{f} - A\mathbf{w}_k\|. \tag{10.5.31}$$

Both of these measures of convergence can give misleading information regarding the convergence of the sequence. However, *both of the measures of convergence are used often, and most often used successfully.*

To see how the residual, (10.5.31), measures the convergence of $\{\mathbf{w}_k\}$ to $\mathbf{u}$, we note that

$$\begin{aligned} \mathbf{e}_k &= \mathbf{u} - \mathbf{w}_k \\ &= A^{-1}(A\mathbf{u} - A\mathbf{w}_k) \\ &= A^{-1}(\mathbf{f} - A\mathbf{w}_k) \\ &= A^{-1}\mathbf{r}_k \end{aligned} \tag{10.5.32}$$

and

$$\|\mathbf{e}_k\| \leq \|A^{-1}\|\|\mathbf{r}_k\|. \tag{10.5.33}$$

Thus, we see that although we cannot measure the actual error, if we bound the norm of the residual error, we will bound the norm of the error. Thus it is logical to use the norm of $\|\mathbf{r}_k\|$ as a stopping criterion. As is so often the case, it is only for the nice problems that we can compute $\|A^{-1}\|$ (see HW10.5.1). Most often $\|A^{-1}\|$ is too difficult or impossible to compute and must be considered to be unknown. Most often this factor is of a reasonable size, but there are times when this factor is very large.

Though the difference of two successive iterates looks different from the residual error, the two measures of convergence are really not that different. We recall from our definition of the residual correction scheme that

$$\mathbf{w}_{k+1} - \mathbf{w}_k = B\mathbf{r}_k, \tag{10.5.34}$$

where B is some sort of an approximation of A^{-1}. Hence, the difference between using $\mathbf{w}_{k+1} - \mathbf{w}_k$ and $\mathbf{r}_k$ as our stopping criterion is the effect of B. Specifically, a calculation analogous to the calculation done in (10.5.32)–(10.5.33) above shows that

$$\mathbf{e}_k = A^{-1}\mathbf{r}_k = A^{-1}B^{-1}(\mathbf{w}_{k+1} - \mathbf{w}_k)$$

and

$$\|\mathbf{e}_k\| \leq \|A^{-1}B^{-1}\|\|\mathbf{w}_{k+1} - \mathbf{w}_k\|. \tag{10.5.35}$$

If we use two successive iterates as a bound on the error, then as with the residual, we include a constant that is usually unknown. The unknown constant associated with two successive iterates, $\|A^{-1}B^{-1}\|$, appears as if it should be nicer than the analogous constant associated with the residual,

$\|A^{-1}\|$. As a part of a residual correction scheme, we choose B such that B in some way approximates A^{-1}. Thus, a good choice of B will make BA approximate the identity matrix and, we hope, cause $\|(BA)^{-1}\| = \|A^{-1}B^{-1}\|$ to be near one. We shall see that this is not always the case.

Remark: In some instances, relationship (10.5.34) between the two successive iterates and the residual is very nice. By (10.5.4) we see that if we use the Richardson iterative scheme, $B = I$, then $\mathbf{r}_k = \mathbf{w}_{k+1} - \mathbf{w}_k$. If we use the Jacobi scheme on our model problem (10.2.3)–(10.2.7), then B is the constant diagonal matrix with

$$\frac{1}{d} = \frac{1}{\frac{2}{\Delta x^2} + \frac{2}{\Delta y^2}}$$

on the diagonal. Hence, $\mathbf{w}_{k+1} - \mathbf{w}_k$ is a constant multiple of $\mathbf{r}_k$. Notice also in the case of the Jacobi scheme that when Δx and Δy are small, d is large and $1/d$ is small. In this case, $\mathbf{w}_{k+1} - \mathbf{w}_k$ is a small multiple of $\mathbf{r}_k$. Hence, when d is large, $\mathbf{r}_k$ could be large when $\mathbf{w}_{k+1} - \mathbf{w}_k$ is small.

Another aspect of choosing a stopping criterion is the expense of implementation. Since usually both $\mathbf{w}_k$ and $\mathbf{w}_{k+1}$ are available at the end of an iteration, using $\mathbf{w}_{k+1} - \mathbf{w}_k$ as a stopping criterion does not involve much additional computational expense. It is not as clear that $\mathbf{r}_k$ is also easily available. Since the Jacobi scheme can be written as

$$\mathbf{r}_k = \mathbf{f} - A\mathbf{w}_k \tag{10.5.36}$$

$$\mathbf{w}_{k+1} = \mathbf{w}_k + D^{-1}\mathbf{r}_k, \tag{10.5.37}$$

computation of the residual can be a very inexpensive part of the Jacobi iteration.

Norms: In the beginning of the section we promised six measures of convergence, and we have produced only two. However, we have not yet specified a norm. As usual, it is possible to use a variety of norms. The most common norms to use are the sup-norm, the ℓ_2 norm and the $\ell_{2,\Delta x}$ norm. There are advantages and disadvantages to the use of each of these norms. The ℓ_2 and $\ell_{2,\Delta x}$ norms are clearly very similar in that one is a multiple of the other. The ℓ_2 norm measures $\mathbf{w}_{k+1} - \mathbf{w}_k$ or $\mathbf{r}_k$ as if they were vectors in $\mathbb{R}^L$ space, and the $\ell_{2,\Delta x}$ norm is an approximation of the square root of the integral of a function squared that is represented by these vectors. One of the advantages of using either the ℓ_2 or the $\ell_{2,\Delta x}$ norm is that the error is forced to be uniformly small. However, this can also be one of its disadvantages. These norms do not tell whether the error is due to a small error uniformly spread across the entire region or an error that is zero everywhere except at one point.

If the sup-norm is used, it is easy to see where this maximum of the absolute value of the elements of either $\mathbf{w}_{k+1} - \mathbf{w}_k$ or $\mathbf{r}_k$ occurs. Often this information can give either numerical or physical information (for example,

there should be some reason if the maximum error always occurs at the same point). It may be that the point at which the maximum error occurs is the most active point in the solution, and it is logical to have the greatest error at that point. It may also be that the point at which the maximum error occurs is in some way more poorly approximated than other points (say due to a boundary condition approximation). In either case, it may be possible to change the scheme to better resolve the solution at that point. In any case, it is usually advantageous for the user to know whether there is such an important point in the domain. However, using the sup-norm to measure the error will not tell you whether the error is as large as the tolerance at one or a few points, or whether it is approximately that large at all points.

One approach that is reasonably common and safe is to use both the sup-norm and either the ℓ_2 or $\ell_{2,\Delta x}$ norm. Though this approach is more expensive, you know that you have just about the maximum amount of information available about the error. We should note that if we have an error that is uniform throughout the components of the vector associated with a 100×100 grid, then if the sup-norm of the vector is ϵ, the ℓ_2 norm will be about 100ϵ and the $\ell_{2,\Delta x}$ norm will be about ϵ. This might help decide what size tolerances to use when using different norms.

Remark: Another obvious variation of the above choice of norm is to use either the sup-norm or one of the ℓ_2 norms as a stopping criterion and include the other norm as a part of your output information.

Tolerance: We must discuss how small we should set our tolerance, i.e., how small should we require $\|\mathbf{f} - A\mathbf{w}_k\|$ or $\|\mathbf{w}_{k+1} - \mathbf{w}_k\|$ to be before we stop our iteration procedure. One approach that is all too common is to smash the iteration error down near the accuracy of the machine being used. If this is done, most of the computer time used is wasted. Another approach is to use a tolerance that is sufficiently small so that the iterative scheme is pleasantly fast. If the tolerance is chosen too large, then we accept a bad approximation. In this discussion we will try to illustrate the difficulties involved in choosing the tolerance. We cannot give a deterministic rule for choosing the tolerance. We try to provide some insight into the selection of the tolerance.

We begin by showing by example that it is easy to waste computer time by choosing a tolerance that is smaller than necessary. Consider the problem assigned in HW10.5.2. Applying the Jacobi scheme, using the sup-norm to measure the difference between iterates, and choosing a tolerance of 10^{-4}, we obtain an approximate solution in 36 iterations. The sup-norm of the error (using the fact that it is easy to obtain the exact solution to the given boundary–value problem) associated with this approximate solution is 0.008543. If we redo this calculation using a tolerance of 10^{-5}, the sup-norm of the error associated with this new approximate solution is 0.00909. We note that the error did not get smaller—it actually got neg-

ligibly larger— and the extra 42 iterations that were needed were wasted. Reducing the tolerance more will not reduce the error and the extra iterations used will be wasted work.

In the preceding paragraph, we saw that it took only 36 iterations to obtain an approximate solution to the desired tolerance—a solution that is as accurate as possible with the given difference equation and grid. If we use the Jacobi scheme to resolve that problem on a grid with $M_x = M_y = 100$ and a tolerance equal to 10^{-6}, the solution will require 2709 iterations and the error associated with this solution will be 0.0011815. A simple calculation with a smaller tolerance, say a calculation with a tolerance equal to 10^{-10} that will take 15,989 iterations and produce an error of 0.0012350, will show that the tolerance of 10^{-6} was sufficiently small.

The computation with $M_x = M_y = 100$ took many more iterations than that with $M_x = M_y = 10$. That should be expected. Recall from Section 10.5.3 that the Jacobi scheme should asymptotically require $m \approx -1/\log_{10} \sigma(R_J)$ iterations to improve the result by one decimal place. In Table 10.5.1 we see that for $M_x = M_y = 100$ it should take 4666 iterations per decimal place whereas for $M_x = M_y = 10$, it should take 46 iterations per decimal place. In both cases Jacobi required fewer iterations than the asymptotic analysis predicts. We should always remember that the results given in Table 10.5.1 are asymptotic results based on the largest eigenvalue (in magnitude). If the error does not contain a significant component in the direction of the eigenvector associated with the largest eigenvalue, the results given in Table 10.5.1 will give only an upper bound of the number of iterations that are necessary. It should be clear that for the problem given in HW10.5.2, the error is completely in the direction of one eigenvector, and the associated eigenvalue is far from the largest eigenvalue. However, we specifically include this discussion to emphasize how slowly the difference of two iterates changes on the grid with $M_x = M_y = 100$. When the spectral radius of R_J is very near one (0.999507 in this case), convergence can be very slow. It is very easy to choose a tolerance that is too large.

We next describe the basic mechanisms that control the error. When approximating the solution to an elliptic problem numerically using an iterative solver, we have two distinct steps and two distinct contributions to the error. The **truncation error** is the approximation error due to the fact that we are approximating a partial differential equation by a difference equation. The algebraic error is the error due to the fact that we are using an iterative solver to solve the difference equation approximately. Let $\mathbf{v}$ denote the vector of values of the solution to the analytic boundary–value problem evaluated at the grid points, let $\mathbf{u}$ denote the exact solution to our discrete approximation of the boundary–value problem, and let $\mathbf{w}_k$ denote the approximate solution to the discrete problem after k iterations. The truncation error, algebraic error and the error in the solution can be written as

$$\mathbf{E} = \mathbf{v} - \mathbf{u}, \quad \mathbf{e}_k = \mathbf{u} - \mathbf{w}_k, \quad \mathbf{e} = \mathbf{v} - \mathbf{w}_k.$$

Since

$$\mathbf{e} = \mathbf{v} - \mathbf{w}_k = (\mathbf{v} - \mathbf{u}) - (\mathbf{u} - \mathbf{w}_k) = \mathbf{E} + \mathbf{e}_k,$$

by the triangular inequality, we get

$$\|\mathbf{e}\| \leq \|\mathbf{E}\| + \|\mathbf{e}_k\|. \tag{10.5.38}$$

We first approximate the partial differential equation problem to be solved by a difference equation with some given accuracy, say $\mathcal{O}(\Delta x^2)$. The error due to this approximation is the truncation error $\mathbf{E}$. This error is due to the choice of discrete equation and the size of M_x and M_y (or Δx and Δy). *The error introduced in this step cannot be recovered, no matter how accurately we solve the difference equation.* If we solve the discrete problem exactly (if that is possible), the error in the computed solution will be $\mathbf{E}$ and will still be $\mathcal{O}(\Delta x^2)$. If we solve the discrete problem to an accuracy significantly greater than the truncation error, the error in the computed solution will be approximately the truncation error. If we solve the discrete problem to an accuracy equal to the truncation error, then by (10.5.38) the error in the computed solution is less than or equal to twice the truncation error. If we choose a scheme and M_x and M_y so that we are satisfied with the accuracy of twice the truncation error, solving the difference equation to fifteen or sixteen decimal places will not give us anything extra. All of the extra work done approximating the iterative solution more accurately than the truncation error is wasted.

Clearly, we do not know our truncation error. In Section 10.3, we considered a $\mathcal{O}(\Delta x^2) + \mathcal{O}(\Delta y^2)$ approximation to our model problem (10.2.3)–(10.2.7) and in Theorem 10.3.3 proved that the sup-norm of the truncation error was less than or equal to

$$C(\Delta x^2 + \Delta y^2)\|\partial^4 v\|_{\infty 0},$$

but we do not know C or $\|\partial^4 v\|_{\infty 0}$.

We have one more aspect of the stopping criterion that makes the choice of the tolerance difficult. We remember that we are working with either the difference between successive iterates or the residual and not the actual error. As we showed earlier, there is a factor of A^{-1} or $A^{-1}B^{-1}$ between the residual or the difference between two successive iterates and the error. In the problem considered in HW10.5.7, we see that we have a truncation error of 0.0072. If we use the Jacobi scheme to solve this problem using a stopping criterion consisting of the sup-norm of successive iterates and a tolerance of 0.0072 (the sup-norm of the truncation error), we see that the scheme requires 2388 iterations. The actual error associated with this solution is 17.1988—much greater than twice the truncation error. We cannot forget that we are bounding the error by $\|\mathbf{w}_{k+1} - \mathbf{w}_k\|$. If we had monitored the residual instead (or computed it at the end of our computation), we would see that the sup-norm of the residual is over 471.0, and the large error would

not surprise us. A smaller tolerance and more iterations are necessary to produce an error as small as twice the truncation error.

We try to use all of the information that we have to choose a reasonable tolerance. For example, consider a $\mathcal{O}(\Delta x^2)$ accurate scheme with $\Delta x = 0.1$. Clearly, we start with a factor of 0.01. If C and the fourth derivatives of the analytic solution are large, then the solution to the difference equation will be a bad approximation to the partial differential equation no matter how we solve the discrete problem. The most difficult case is when C and the fourth derivatives are small, i.e. when we have the capability of obtaining a good solution. If we choose our tolerance to be 0.0001 (where 0.01 is due to the Δx^2, and another multiple of 0.01 is included because of our fear that the constants C and $\|\partial^4 v\|_{\infty 0}$ might be small and because we may need a safety factor), we will generally be successful. Choosing a tolerance of 1.0×10^{-10} in this case would generally be wasteful. For example, if we choose the tolerance to be 6.0×10^{-7} (approximately Δx^2 times a safety factor of 0.01), the solution takes 32,128 iterations and produces a solution with an error of 0.0089, which is less than twice the truncation error. We note that in this calculation we obtain a sup-norm of the residual of the final result of 0.0393.

Thus, to design a stopping criterion for our codes, we take the following steps.

- Choose one the ℓ_2 norms or the sup-norm, or both.
- Decide whether to work with the difference of two successive iterates, the residual, or both. If you use only one of these measures, at least evaluate the other with your final result.
- Choose a tolerance, taking into account
  - whether you chose $\mathbf{w}_{k+1} - \mathbf{w}_k$ or $\mathbf{r}_k$,
  - your choice of norm made above, and
  - the truncation error of your difference scheme.

Of course, there are other stopping criteria than those given above. Some of them are rigorous (at least under certain assumptions) and some are not. Usually the more exotic the stopping criterion is, the more computationally expensive it is. The criteria given in this section are generally the easiest and most commonly used criteria.

HW 10.5.1 Let A be the matrix given in (10.8.1) (the matrix associated with the model problem (10.2.3)-(10.2.7) and let B be the diagonal matrix associated with defining the Jacobi scheme as a residual correction scheme. Use the eigenvalues computed in Remark 3, page 319, to compute $\|A^{-1}\|$ and $\|A^{-1}B^{-1}\|$.

Hint: Use the fact that if D is an $L \times L$ symmetric matrix and the eigenvalues of D are λ_j, $j = 1, \ldots, L$, then

$$\|D\| = \max_{1 \le j \le L} |\lambda_j|.$$

The norm given above will be the ℓ_2 matrix norm.

10.5.5 Implementation of the Jacobi Scheme

We now include a short discussion on implementation of the Jacobi scheme. As was the case with the stopping criterion, the implementation discussion given here is equally applicable to other iterative schemes that will be introduced in following sections. We begin by noting that it is very easy to implement a basic Jacobi scheme. Suppose, for example, that we wish to solve a Poisson equation with Dirichlet boundary conditions. It is easy to see that by setting up two arrays, u and *uold*, initialized to include the boundary conditions, it is not hard to write a program applying algorithm Jacobi-10.5.13. The program consists of initializing the variables, applying Jacobi-10.5.13, and checking either the residual error or the rate of convergence error against the tolerance to see whether we set $u = uold$ and apply Jacobi-10.5.13 again or we write out our results and quit.

Instead of the basic approach suggested above, we suggest that the implementation be done in a more flexible way. Specifically, we suggest that the code be written as

```
      Call Initialize
10    Call Jacobi-10.5.13
      Call Tolerance, if not converged go to 10
      Call Output
```

In subroutine Initialize, in addition to setting such constants as Δx, *tolerance*, etc. and initializing both u and *uold*, we define a stencil array $S(j,k,m)$ where $j = 0, \ldots, M_x$, $k = 0, \ldots, M_y$ and $m = 0, \ldots, 4$. The stencil array would contain the β_{jk}^m values given in algorithm Jacobi-10.5.13 (and be numbered in the same manner).

In subroutine Jacobi-10.5.13, algorithm Jacobi-10.5.13 would be implemented in terms of u, *uold* and S. And of course, the Tolerance subroutine would contain our stopping criterion and Output would give us our output along with any convergence values that might be of interest (the point at which the sup-norm of the error is attained, the values of the norms of the different errors, etc.).

For solving a Poisson equation with Dirichlet boundary conditions, this approach would be wasteful. Stencil values would not be needed at $j = k = 0$, $j = M_x$, and $k = M_y$, and values of $S(j,k,m)$ would be either $4/\Delta x^2$ (for $m = 0$) or $1/\Delta x^2$ (for $m = 1,\ 2,\ 3$,or 4). It is not difficult to see that

if we write a program including the stencil and then wish to solve some other two dimensional elliptic problem with Dirichlet boundary conditions (say with nonconstant coefficients), we could do so easily by changing S. As we shall see later, the implementation described above can also be used to solve problems with Neumann or mixed boundary conditions on one or more boundaries and problems with irregular boundaries with a minimum amount of additional work.

Remark: Of course, there are variations of the description of the pseudocode given above. The idea of the algorithm given above is to provide the logical flow of the code and to emphasize the use of the stencil array. One logical variation involves the stopping criterion. If we want to use the residual as our stopping criterion, we might want to use the form of the Jacobi iteration given in (10.5.36)–(10.5.37). We might compute $\mathbf{r}_k$ and $\|\mathbf{r}_k\|$ (using any norm we desire) and then use (10.5.37) to compute $\mathbf{w}_{k+1}$ in the Jacobi subroutine and then proceed to Tolerance to make the decision.

HW 10.5.2 Use the Jacobi relaxation scheme to find an approximate solution to the following elliptic boundary–value problem.

$$\nabla^2 v = \sin \pi x \sin 2\pi y, \quad (x, y) \in R = (0, 1) \times (0, 1) \tag{10.5.39}$$

$$v = 0 \text{ on } \partial R. \tag{10.5.40}$$

Use $M_x = M_y = 10$, both the difference of two successive iterates and the residual as stopping criterion with both the ℓ_2 and sup-norms, and use a tolerance of 0.0001. Compare the iteration number for which each of the above stopping criteria is satisfied and the quality of the solution for each of the criteria.

HW 10.5.3 Repeat the solution done in HW10.5.2 twice, first using $M_x = M_y = 50$ with tolerance 4.0×10^{-6} and then using $M_x = M_y = 100$ with tolerance 1.0×10^{-6}. Use any one of the stopping criteria described in HW10.5.2.

HW 10.5.4 Resolve the problem given in HW10.5.2 using $M_x = M_y = 50$ and the difference between two successive iterates measured with the ℓ_2 norm as a stopping criterion. Solve this problem twice, first using a tolerance of $1.0e^{-5}$ and then using a tolerance of $1.0e^{-15}$ (or whatever the accuracy of your machine allows). Compute the ℓ_2 norm of the true error (computed solution minus the analytic solution) for each of these solutions, and compare and contrast these true errors.

HW 10.5.5 Use the Jacobi relaxation scheme to find an approximate solution to the following elliptic boundary–value problem

$$\begin{aligned} \nabla^2 v &= 0, \quad (x,y) \in R = (0,1) \times (0,1) \\ v(x,0) &= 1 - x^2, \quad x \in (0,1) \\ v(x,1) &= -x^2, \quad x \in (0,1) \\ v(0,y) &= 1 - y^2, \quad y \in (0,1) \\ v(1,y) &= -y^2, \quad y \in (0,1). \end{aligned}$$

Use $M_x = M_y = 10$. Choose one or more of the stopping criteria to be used and an appropriate tolerance.

HW 10.5.6 Use the Jacobi scheme to find an approximate solution to the problem

$$\begin{aligned} \nabla^2 v &= e^{x+y}, \quad (x,y) \in R = (0,1) \times (0,1) \\ v &= -e^{1-x-y} \text{ on } \partial R. \end{aligned}$$

Use $M_x = M_y = 100$ and tolerance $= 1.0 \times 10^{-6}$.

HW 10.5.7 (a) Consider the boundary–value problem

$$\nabla^2 v = 2\pi^2(\sin \pi x \cos \pi y + \cos \pi x \sin \pi y)e^{\pi(x+y)}, \quad (x,y) \in R = (0,1) \times (0,1) \tag{10.5.41}$$

$$v = 0, \quad (x,y) \text{ on } \partial R. \tag{10.5.42}$$

Use the Jacobi scheme with $M_x = M_y = 128$ and a stopping criterion consisting of the sup-norm of the difference between two successive iterates and a tolerance of 1.0×10^{-10} to determine "essentially" the exact solution to the discrete problem. Use the exact solution to the analytic problem,

$$v(x,y) = \sin \pi x \sin \pi y e^{\pi(x+y)},$$

to determine the sup-norm of the truncation error. Repeat the solution using the Jacobi scheme with the sup-norm of the truncation error as the tolerance. Determine the error in this solution and compare this error with the truncation error.
(b) Determine a tolerance (larger than 1.0×10^{-10}) that will produce a solution with error less than or equal to twice the truncation error.

10.5.6 Gauss-Seidel Scheme

We proceed as we did in Section 10.5.2 and consider solving difference equation (10.5.13) along with Dirichlet boundary conditions. The Gauss-Seidel algorithm for solving this problem can be written as follows.

Gauss-Seidel-10.5.13

For $k = 1, \ldots, M_y - 1$
For $j = 1, \ldots, M_x - 1$
$u'_{j\,k} = -\frac{1}{\beta^0_{j\,k}}\left[F_{j\,k} - \beta^1_{j\,k}u_{j+1\,k} - \beta^2_{j\,k}u'_{j-1\,k} - \beta^3_{j\,k}u_{j\,k+1} - \beta^4_{j\,k}u'_{j\,k-1}\right]$
Next j
Next k

We see immediately that the difference between the Gauss-Seidel iteration and the Jacobi iteration is that the Gauss-Seidel uses "new" values when it reaches in the $(j-1,k)$ and $(j,k-1)$ directions. Because of the order in which we have done the calculation, these "new" values have already been computed. Hence, the Gauss-Seidel scheme replaces the (j,k) term by the adjusted average of its four neighbors, using "new" values when they are available.

If we return to the description of the Gauss-Seidel scheme given in Section 10.5, we see that it was given as a residual correction scheme with $B = (L+D)^{-1}$ where L, D, and U are the lower triangular, diagonal, and upper triangular parts of $A = L + D + U$, respectively. Thus the Gauss-Seidel scheme is given by

$$\begin{aligned}\mathbf{w}_{k+1} &= \mathbf{w}_k + (L+D)^{-1}\mathbf{r}_k \\ &= \mathbf{w}_k + (L+D)^{-1}(\mathbf{f} - (L+D+U)\mathbf{w}_k) \\ &= (L+D)^{-1}(\mathbf{f} - U\mathbf{w}_k). \end{aligned} \tag{10.5.43}$$

Given $\mathbf{w}_k$, to find $\mathbf{w}_{k+1}$ using (10.5.43), we must solve

$$(L+D)\mathbf{w}_{k+1} = \mathbf{f} - U\mathbf{w}_k. \tag{10.5.44}$$

The choice of matrix $B = (L+D)^{-1}$ was a good choice because $L+D$ is a lower triangular matrix, and solving equation (10.5.44) is an easy forward Gaussian elimination sweep.

To show that solving equation (10.5.44) is the same as Algorithm Gauss-Seidel-10.5.13, we consider solving an equation of the form of equation (10.5.13) where the Dirichlet boundary conditions are included in the terms $j = 0$, $k = 0$, $j = M_x$ and $k = M_y$ (the same problem solved by Algorithm Gauss-Seidel-10.5.13). Matrices L, D and U are given by (10.5.14)–(10.5.16). We must solve equation (10.5.44) where the $(M_x - 1)(M_y - 1) \times (M_x - 1)(M_y - 1)$ matrix $L + D$ is given in Figure 10.5.1 where $\beta^2_{1\,2}$, $\beta^2_{1\,3}, \ldots, \beta^2_{1\,M_y-1}$ are zero, again because this is where difference equation (10.5.13) reaches to the boundary. The vectors $\mathbf{w}_{k+1}$ and $\mathbf{w}_k$ are given by

$$\mathbf{w}_{k+1} = \begin{bmatrix} u'_{1\,1} \\ \vdots \\ \vdots \\ u'_{M_x-1\,M_y-1} \end{bmatrix} \quad \text{and} \quad \mathbf{w}_k = \begin{bmatrix} u_{1\,1} \\ \vdots \\ \vdots \\ u_{M_x-1\,M_y-1} \end{bmatrix}$$

$$L+D=\begin{pmatrix}
-\beta^0_{1\,1} & 0 & \cdots & & & & & \\
\beta^2_{2\,1} & -\beta^0_{2\,1} & 0 & \cdots & & & & \\
0 & \beta^2_{3\,1} & -\beta^0_{3\,1} & 0 & \cdots & & & \\
 & \ddots & \ddots & \ddots & \ddots & & & \\
 & \cdots & 0 & \beta^2_{M_x-1\,1} & -\beta^0_{M_x-1\,1} & 0 & \cdots & \\
\beta^4_{1\,2} & 0 & \cdots & 0 & \beta^2_{1\,2} & -\beta^0_{1\,2} & 0 & \cdots \\
 & \ddots & & \ddots & \ddots & \ddots & \ddots & \\
\cdots & 0 & \beta^4_{M_x-1\,M_y-1} & 0 & \cdots & 0 & \beta^2_{M_x-1\,M_y-1} & -\beta^0_{M_x-1\,M_y-1}
\end{pmatrix}$$

FIGURE 10.5.1. The $(M_x-1)(M_y-1)\times(M_x-1)(M_y-1)$ matrix $L+D$, where $\beta^2_{1\,2},\ \beta^2_{1\,3},\ldots,\beta^2_{1\,M_y-1}$ are zero.

and the right hand side contains $-U\mathbf{w}_k$ (i.e., in this case the $-\beta^1_{j\,k}u_{j+1\,k} - \beta^3_{j\,k}u_{j\,k+1}$ terms) and $\mathbf{f}$ (which contains the contribution due to F and the boundary conditions in the appropriate rows). It is not hard to see that the forward Gaussian elimination sweep necessary to solve equation (10.5.44) is that given in the Gauss-Seidel algorithm, Gauss-Seidel-10.5.13.

If we specialize the Gauss-Seidel algorithm to solve the Poisson equation, (10.2.3)–(10.2.7), we obtain the following algorithm.

Gauss-Seidel-10.4

For $k = 1, \ldots, M_y - 1$
For $j = 1, \ldots, M_x - 1$
$u'_{j\,k} = \frac{1}{d}\left[F_{j\,k} + \frac{1}{\Delta x^2}(u_{j+1\,k} + u'_{j-1\,k}) + \frac{1}{\Delta y^2}(u_{j\,k+1} + u'_{j\,k-1})\right]$
Next j
Next k

where $d = 2(1/\Delta x^2 + 1/\Delta y^2)$.

Remark: For all of the reasons given for the Jacobi scheme, we recommend that the Gauss-Seidel scheme be implemented using stencils.

HW 10.5.8 Use the Gauss-Seidel relaxation scheme to find an approximate solution to the following elliptic boundary value problem.

$$\nabla^2 v = \sin \pi x \sin 2\pi y, \quad (x, y) \in R = (0,1) \times (0,1) \tag{10.5.45}$$

$$v = 0 \text{ on } \partial R. \tag{10.5.46}$$

Use $M_x = M_y = 10$, both the difference of two successive iterates and the residual as stopping criteria with both the l_2 and sup–norms, and a tolerance of 0.0001. Compare the iteration number for which each of the above stopping criteria is satisfied and the quality of the solution for each of the criteria. Also compare the number of iterations necessary for solution with those found using the Jacobi scheme in HW10.5.2

HW 10.5.9 Use the Gauss-Seidel scheme to find an approximate solution to the following elliptic boundary value problem.

$$\begin{aligned} \nabla^2 v &= 0, \quad (x, y) \in R = (0,1) \times (0,1) \\ v(x,0) &= 1 - x^2, \quad x \in (0,1) \\ v(x,1) &= -x^2, \quad x \in (0,1) \\ v(0,y) &= 1 - y^2, \quad y \in (0,1) \\ v(1,y) &= -y^2, \quad y \in (0,1). \end{aligned}$$

Use $M_x = M_y = 10$. Choose one or more of the stopping criteria to be used and an appropriate tolerance. Compare the number of iterations necessary with those found using the Jacobi scheme in HW10.5.5.

HW 10.5.10 Use the Gauss-Seidel scheme to find an approximate solution to the problem

$$\nabla^2 v = e^{x+y}, \quad (x,y) \in R = (0,1) \times (0,1)$$
$$v = -e^{1-x-y} \text{ on } \partial R.$$

Use $M_x = M_y = 100$ and tolerance $= 1.0 \times 10^{-6}$. Compare the number of iterations necessary to obtain the solution with the number of Jacobi iterations that were necessary to solve the problem in HW10.5.6.

10.5.7 Analysis of the Gauss-Seidel Relaxation Scheme

As is the case with Jacobi relaxation, the easiest way to obtain convergence for the Gauss-Seidel scheme is the following analogue to Proposition 10.5.3.

Proposition 10.5.4 *If A is irreducible, diagonally dominant, and for at least one j, $|a_{jj}| > \rho_j$, then the Gauss-Seidel iteration scheme converges for all $\mathbf{w}_0$.*

Proof: [76], page 1019.

As was the case for the Jacobi scheme, the Gauss-Seidel scheme will converge when used to approximate the solution to our model problem (10.2.3)–(10.2.7) and, for sufficiently small Δx and Δy, will converge when used to approximate the solution to equation (10.2.10) (either the constant or variable coefficient version) along with boundary conditions (10.2.4)–(10.2.7).

Another convergence result we get for the Gauss-Seidel scheme is the following.

Proposition 10.5.5 *If A is a real positive definite matrix, then $\sigma(R_{GS}) < 1$, where R_{GS} is the Gauss-Seidel iteration matrix associated with the matrix A.*

Proof: [76], page 1019.

Proposition 10.5.5 is a very nice result. One of the interesting points about this proposition is that the analogous result does not hold for the Jacobi relaxation scheme. However, Proposition 10.5.5 is not generally easy to apply. To apply Proposition 10.5.4, it usually is enough (accepting the fact that the matrix will be irreducible) to inspect the matrix or restrict Δx and Δy to satisfy the hypotheses. To apply Proposition 10.5.5, it is usually necessary to compute the eigenvalues of the matrix (which we do often but only for model problems) to show that A is positive definite. If we are going to do this, we may as well obtain the convergence by applying Proposition 10.5.2. One of the reasons that we like to use Proposition 10.5.2 if it is convenient is that it is in that setting that we obtained the

asymptotic analysis of the error given in Section 10.5.1. We next analyze the Gauss-Seidel scheme for our model problem (10.2.3)–(10.2.7) directly, via Proposition 10.5.2.

To compute the eigenvalues associated with the Gauss-Seidel iteration matrix

$$R_{GS} = I - BA = I - (L + D)^{-1}(L + D + U) = -(L + D)^{-1}U,$$

we must solve

$$R_{GS}\,\mathbf{x} = -(L + D)^{-1}U\mathbf{x} = \lambda\mathbf{x},$$

or

$$-U\mathbf{x} = \lambda(L + D)\mathbf{x}. \tag{10.5.47}$$

We now consider the eigenvalue analysis of convergence of the Gauss-Seidel scheme for solving problem (10.2.3)–(10.2.7) (for the matrix equation with matrix (10.2.8)–(10.2.9)).

Example 10.5.2 Perform an eigenvalue analysis of the Gauss-Seidel iteration matrix associated with solving problem (10.2.3)–(10.2.7).

Solution: If we consider problem (10.2.3)–(10.2.7) and its associated iteration matrix, the problem analogous to eigenvalue problem (10.5.47) is

$$\frac{1}{\Delta x^2}u_{j+1\,k} + \frac{1}{\Delta y^2}u_{j\,k+1} = \lambda\left[-\frac{1}{\Delta x^2}u_{j-1\,k} + 2\Big(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}\Big)u_{j\,k} - \frac{1}{\Delta y^2}u_{j\,k-1}\right], \tag{10.5.48}$$

where we have again assumed that $u_{0\,k} = u_{M_x\,k} = 0$, $k = 0, \ldots, M_y$, and $u_{j\,0} = u_{j\,M_y} = 0$, $j = 0, \ldots, M_x$. If in equation (10.5.48) we replace $u_{j\,k}$ by $\lambda^{(j+k)/2}w_{j\,k}$, we see that equation (10.5.48) reduces to

$$d\sqrt{\lambda}w_{j\,k} = \frac{1}{\Delta x^2}w_{j-1\,k} + \frac{1}{\Delta y^2}w_{j\,k-1} + \frac{1}{\Delta x^2}w_{j+1\,k} + \frac{1}{\Delta y^2}w_{j\,k+1}. \tag{10.5.49}$$

By comparing equation (10.5.49) with equation (10.5.19), we see that the eigenvalues associated with the Gauss-Seidel iteration matrix for problem (10.2.3)–(10.2.7) are the square of the eigenvalues for the Jacobi iteration matrix for solving the same problem, or

$$\lambda_s^p = \frac{4}{d^2}\Big(\frac{1}{\Delta x^2}\cos\frac{s\pi}{M_x} + \frac{1}{\Delta y^2}\cos\frac{p\pi}{M_y}\Big)^2. \tag{10.5.50}$$

As we did with the Jacobi scheme, we note that the largest eigenvalue is given by

$$\lambda_1^1 = \frac{4}{d^2}\Big(\frac{1}{\Delta x^2}\cos\frac{\pi}{M_x} + \frac{1}{\Delta y^2}\cos\frac{\pi}{M_y}\Big)^2 \tag{10.5.51}$$

and that

$$|\lambda_1^1| < 1.$$

Hence, by Proposition 10.5.2 the Gauss-Seidel scheme for solving equations (10.2.3)–(10.2.7) converges.

Remark 1: We notice that the eigenvalues of R_{GS} are the square of the eigenvalues associated with the Jacobi scheme. Hence, *the Jacobi scheme converges for our model problem (10.2.3)–(10.2.7) if and only if the Gauss-Seidel scheme converges.*

Remark 2: As was the case for the Jacobi scheme, if $\Delta x = \Delta y$ ($M_x = M_y = M$), we will have repeated eigenvalues for the iteration matrix for the Gauss-Seidel scheme. Specifically, we again have the $M-1$ zero eigenvalues that occur because $\cos s\pi/M = -\cos(M-s)\pi/M$ for $s = 1, \ldots, M-1$. We can get all of the eigenvectors for the Gauss-Seidel iteration matrix that are associated with *nonzero eigenvalues* from the corresponding eigenvectors of the Jacobi iteration matrix by using the transformation $u_{j\,k}^{p\,s} = (\lambda_s^p)^{(j+k)/2} v_{j\,k}^{p\,s}$ where $v_{j\,k}^{p\,s}$ and $u_{j\,k}^{p\,s}$ are the components of the eigenvector associated with the (p, s) eigenvalues for the Jacobi and Gauss-Seidel schemes, respectively. Using separation of variables, it is not hard to see that the only eigenvector associated with the zero eigenvalue of the Gauss-Seidel iteration matrix is the vector

$$[1\ 0\ \cdots]^T.$$

Hence, we should realize that the analysis done in equations (10.5.7) to (10.5.9) does not apply to this situation. Yet we can still apply the asymptotic results given near the end of Section 10.5.1 (if $\sigma(R) < 1$, the number of iterations, k, needed to reduce the norm of the initial error vector by a factor of ζ can be approximated by $k \approx -\log\zeta / R_\infty(R))$ to see that the number of iterations needed to reduce the norm of the error is controlled by the magnitude of the largest eigenvalue. Specifically, we note that since

$$\log_{10}(\sigma(R_{GS})) = \log_{10}((\sigma(R_J))^2) = 2\log_{10}(\sigma(R_J)),$$

the number of iterations needed to reduce the norm of the initial error vector by a factor of ζ for the Gauss-Seidel relaxation scheme is given by

$$k \approx \frac{-\log_{10}\zeta}{2\log_{10}(\sigma(R_J))}.$$

Hence, *to reduce the norm of the initial error vector by a factor of ζ by the Gauss-Seidel relaxation scheme it takes half the number of iterations that it would take to reduce the norm of the initial error the same amount using the Jacobi relaxation scheme.* In Section 10.5.3 we saw that many Jacobi iterations were necessary to reduce the error by one decimal place. Though half as many iterations is a nice result, often it is still too many.

The result given above in Remark 2 concerning the comparison of the speeds of convergence of the Jacobi and Gauss-Seidel schemes is a very specific result. The analysis is done for the iteration matrices R_J and R_{GS} associated with solving our model problem (10.2.3)–(10.2.7). There are similar results that hold for much larger classes of matrices (problems). Consider, for example, the following result.

Proposition 10.5.6 *Let R_J and R_{GS} denote the iteration matrices for the Jacobi and Gauss-Seidel schemes for approximating the solution to difference equation (10.5.13) with Dirichlet boundary conditions based on the variables being given in a j-k lexicographical order and suppose that the eigenvalues of R_J are real. If $\lambda \neq 0$ is an eigenvalue of R_{GS}, then $\sqrt{\lambda}$ and $-\sqrt{\lambda}$ are eigenvalues of R_J. If λ is an eigenvalue of R_J, then λ^2 is an eigenvalue of R_{GS}. And finally, $\sigma(R_{GS}) = \big(\sigma(R_J)\big)^2$.*

We do not prove this result at this time. We shall see that this result will be contained in Proposition 10.5.10 proved in Section 10.5.9. The result given in Proposition 10.5.6 can be extended to even a larger class of problems. In Section 10.5.10, we define consistently ordered matrices and state an analogue of Propositions 10.5.6 and 10.5.10 that holds true for the class of consistently ordered matrices.

HW 10.5.11 Compute the eigenvalues found in Example 10.5.2 directly using separation of variables (as we did in Example 10.5.1).

10.5.8 Successive Overrelaxation Scheme

In this section we we make what might appear to be a slight change in the Gauss-Seidel scheme and obtain a huge increase in speed of convergence. This (not so) slight change of the Gauss-Seidel scheme is known as the **successive overrelaxation scheme (SOR).** As we have done before, we consider solving difference equation (10.5.13) along with Dirichlet boundary conditions. The successive overrelaxation scheme for solving this problem can be written as follows.

SOR-10.5.13

For $k = 1, \ldots, M_y - 1$

For $j = 1, \ldots, M_x - 1$

$$\hat{u}_{j\,k} = -\tfrac{1}{\beta^0_{j\,k}}\big[F_{j\,k} - \beta^1_{j\,k}u_{j+1\,k} - \beta^2_{j\,k}u'_{j-1\,k} - \beta^3_{j\,k}u_{j\,k+1} - \beta^4_{j\,k}u'_{j\,k-1}\big]$$

$$u'_{j\,k} = u_{j\,k} + \omega\big[\hat{u}_{j\,k} - u_{j\,k}\big]$$

Next j

Next k

Clearly the difference between the Gauss-Seidel scheme and SOR is that after a Gauss-Seidel-like step, the SOR scheme computes the weighted average of this Gauss-Seidel step and the previous value (where ω is a free parameter).

HW 10.5.12 Repeat the problems given in HW10.5.2 and HW10.5.8 using the SOR scheme with $\omega = 0.5$, $\omega = 1.5$, and $\omega = 1.75$.

HW 10.5.13 Repeat the problems given in HW10.5.5 and HW10.5.9 using the SOR scheme with $\omega = 0.5$, $\omega = 1.5$, and $\omega = 1.75$.

10.5.9 Elementary Analysis of SOR Scheme

We begin by considering a general matrix equation of the form

$$A\mathbf{u} = \mathbf{f}$$

where $A = L + D + U$. For a matrix of this form, the analogue to algorithm SOR-10.5.13 given above can be written as

$$\hat{\mathbf{u}} = D^{-1}\left[\mathbf{f} - L\mathbf{u}' - U\mathbf{u}\right] \tag{10.5.52}$$

$$\mathbf{u}' = (1-\omega)\mathbf{u} + \omega\hat{\mathbf{u}}, \tag{10.5.53}$$

or

$$\mathbf{u}' = (1-\omega)\mathbf{u} + \omega D^{-1}\left[\mathbf{f} - L\mathbf{u}' - U\mathbf{u}\right]. \tag{10.5.54}$$

If we solve equation (10.5.54) for $\mathbf{u}'$, we get

$$\mathbf{u}' = (I + \omega D^{-1}L)^{-1}\left[(1-\omega)I - \omega D^{-1}U\right]\mathbf{u} + \omega(I + \omega D^{-1}L)^{-1}D^{-1}\mathbf{f}. \tag{10.5.55}$$

If we replace $\mathbf{f}$ in equation (10.5.55) by $\mathbf{r} + A\mathbf{u}$ (since the residual $\mathbf{r} = \mathbf{f} - A\mathbf{u}$) and simplify, we get

$$\mathbf{u}' = \mathbf{u} + \omega(I + \omega D^{-1}L)^{-1}D^{-1}\mathbf{r} = \mathbf{u} + \omega(D + \omega L)^{-1}\mathbf{r}. \tag{10.5.56}$$

We note that this is the form of the SOR scheme given in Section 10.5 (where it was given as a residual correction scheme). The iteration matrix associated with the SOR scheme is given by

$$\begin{aligned} R_{SOR} &= I - \omega(I + \omega D^{-1}L)^{-1}D^{-1}A \\ &= (I + \omega D^{-1}L)^{-1}\left[I + \omega D^{-1}L - \omega D^{-1}(L + D + U)\right] \\ &= (I + \omega D^{-1}L)^{-1}\left[(1-\omega)I - \omega D^{-1}U\right]. \end{aligned} \tag{10.5.57}$$

We emphasize that at this point ω is still a free parameter. The goal is to choose ω cleverly so as to "help our convergence." It is not clear which values of ω are permissible or helpful. We do know that if ω is chosen to be 1, SOR reduces to Gauss-Seidel and the scheme is convergent (but that is not much help).

We recall from Proposition 10.5.2 that the SOR scheme will converge if and only if $\sigma(R_{SOR}) < 1$. We use this result to obtain a range for the parameter ω.

Proposition 10.5.7 *If $\sigma(R_{SOR}) < 1$ (if the SOR scheme converges), then $0 < \omega < 2$.*

Proof: We begin by noting that the eigenvalues λ of R_{SOR} satisfy

$$\begin{aligned} 0 &= \det(R_{SOR} - \lambda I) = \det\left[(I + \omega D^{-1}L)^{-1}\{(1-\omega)I - \omega D^{-1}U\} - \lambda I\right] \\ &= \det\left[(I + \omega D^{-1}L)^{-1}\right] \det\left[(1-\omega)I - \omega D^{-1}U - \lambda(I + \omega D^{-1}L)\right] \\ &= a_0\lambda^L + a_1\lambda^{L-1} + \cdots + a_L, \end{aligned}$$

where $L = (M_x - 1)(M_y - 1)$. We note that $a_0 = \pm 1$ and $a_L = \det(R_{SOR})$. But

$$\begin{aligned} \det(R_{SOR}) &= \det\left[(I + \omega D^{-1}L)^{-1})\{(1-\omega)I - \omega D^{-1}U\}\right] \\ &= \det\left[I + \omega D^{-1}L)^{-1}\right] \det\left[(1-\omega)I - \omega D^{-1}U\right] \\ &= (1-\omega)^L. \end{aligned}$$

Since $a_L = (1-\omega)^L$, we know that the product of the eigenvalues of R_{SOR}, $\lambda_1 \cdots \lambda_L = (1-\omega)^L$. Thus at least one of the eigenvalues must satisfy $|\lambda_k| \geq |1-\omega|$. Since $|\lambda_k| < 1$ (since $\sigma(R_{SOR}) < 1$), $|1-\omega| < 1$. This is the same as $0 < \omega < 2$.

Hence, we see that the largest allowable range of ω values (if we want the scheme to be convergent) is over the interval $(0, 2)$. We obtain several other results as a part of the above proof, which we now state.

Corollary 10.5.8

$$\det(R_{SOR}) = (1-\omega)^L.$$

Corollary 10.5.9 *The iteration matrix R_{SOR} is nonsingular for all values of ω except $\omega = 1$ (except when the SOR scheme reduces to the Gauss-Seidel scheme).*

There are a large number of convergence results for the SOR scheme. Some of these results are useful in the application to the numerical solution of elliptic partial differential equations and some are not (or do not apply very often). Of the convergence results, some are easy and some are difficult. To make the analysis easier, for the rest of this section we consider the analysis of the SOR scheme for the matrix equation associated with solving difference equation (10.5.13) along with Dirichlet boundary conditions. Specifically, *we consider the matrix $A = L + D + U$ where L, D, and U are given by (10.5.14)–(10.5.16) and a j-k lexicographical ordering of the variables.*

Now that we have some structure in our matrices, we try to compute $\sigma(R_{SOR})$. It is not difficult to see that

$$R_{SOR}\,\mathbf{u} = \lambda\mathbf{u}$$

is equivalent to

$$\big[(1-\omega)D-\omega U\big]\mathbf{u}=\lambda(D+\omega L)\mathbf{u}. \tag{10.5.58}$$

We are then able to prove the following result.

Proposition 10.5.10 *If $\lambda\neq 0$ is an eigenvalue of R_{SOR}, then there is an eigenvalue λ_J of R_J such that*

$$(1-\omega-\lambda)^2=\lambda_J^2\omega^2\lambda. \tag{10.5.59}$$

Conversely, if λ_J is an eigenvalue of R_J and (10.5.59) is satisfied for some λ, then λ is an eigenvalue of R_{SOR}.

Proof: Using the form of the matrices L, D and U, eigenvalue problem (10.5.58) can be rewritten as

$$\begin{aligned}(1-\omega)\beta^0_{j\,k}u_{j\,k}-\omega\beta^1_{j\,k}u_{j+1\,k}-\omega\beta^3_{j\,k}u_{j\,k+1}\\ =\lambda\big[\beta^0_{j\,k}u_{j\,k}+\omega\beta^2_{j\,k}u_{j-1\,k}+\omega\beta^4_{j\,k}u_{j\,k-1}\big]\end{aligned} \tag{10.5.60}$$

$j=1,\ldots,M_x-1$, $k=1,\ldots,M_y-1$, where $u_{0\,k}=u_{M_x\,k}=0$, $k=1,\ldots,M_y-1$ and $u_{j\,0}=u_{j\,M_y}=0$, $j=1,\ldots,M_x-1$ (because any nonzero Dirichlet boundary conditions have been moved to the right hand side $(\mathbf{f})$ and will not affect the iteration matrix R_{SOR}).

As we did in the Gauss-Seidel scheme, we let $u_{j\,k}=\lambda^{(j+k)/2}w_{j\,k}$ and get

$$\begin{aligned}\frac{(1-\omega)-\lambda}{\omega\lambda^{1/2}}\beta^0_{j\,k}w_{j\,k}\\ =\big[\beta^3_{j\,k}w_{j\,k+1}+\beta^1_{j\,k}w_{j+1\,k}+\beta^2_{j\,k}w_{j-1\,k}+\beta^4_{j\,k}w_{j\,k-1}\big].\end{aligned} \tag{10.5.61}$$

From (10.5.60) we see that if $\omega=0$, all of the eigenvalues of R_{SOR} are equal to 1. Otherwise, any number λ_J that satisfies equation (10.5.59),

$$\lambda_J=\frac{1-\omega-\lambda}{\omega\lambda^{1/2}}, \tag{10.5.62}$$

will be an eigenvalue of R_J (from equation (10.5.61)).

Conversely, if λ_J is an eigenvalue of R_J, $w_{j\,k}$ is the associated eigenvector and equation (10.5.59) holds, then we can solve for

$$\lambda_\pm=\frac{1}{2}\left[\omega^2\lambda_J^2-2\omega+2\pm\sqrt{(2\omega-2-\lambda_J^2\omega^2)^2-4(1-\omega)^2}\right] \tag{10.5.63}$$

and use the transformation $u_{j\,k}=\lambda_+^{(j+k)/2}w_{j\,k}$ to show that λ_+ will be an eigenvalue of R_{SOR}.

Now that we have a representation for the eigenvalues of R_{SOR}, we must decide which value of the parameter ω we should use. To do this, we introduce the alternative notation for R_{SOR}, $R_{SOR,\omega}$, to emphasize the dependence on ω. In addition, we denote the maximum magnitude of the

roots $\lambda_\pm$ given in (10.5.63) by $\lambda_{\max} = \lambda_{\max}(\omega, \lambda_J)$ and let $\bar{\lambda}_J$ be such that $\bar{\lambda}_J = \sigma(R_J)$. And finally, we add two additional assumptions that will make our work more understandable. *We assume that* $\sigma(R_J) < 1$ *and that the eigenvalues of* R_J *are real.* We then prove the following result.

Proposition 10.5.11 $\sigma(R_{SOR,\omega}) = \lambda_{\max}(\omega, \bar{\lambda}_J)$.

Proof: We begin by noting that

$$\sigma(R_{SOR,\omega}) = \max_{\lambda_J} \lambda_{\max}(\omega, \lambda_J).$$

If we then note that by solving equation (10.5.62) as a quadratic equation in $\sqrt{\lambda}$, we find that $\lambda_\pm$ can be written as

$$\lambda_\pm = (\sqrt{\lambda_\pm})^2 = \frac{1}{4}\left[-\omega\lambda_J \pm \sqrt{\omega^2\lambda_J^2 - 4(\omega - 1)}\right]^2 \tag{10.5.64}$$

(which is equivalent to (10.5.63)). Then

$$\lambda_{\max}(\omega, \lambda_J) = \frac{1}{4}\left|\omega|\lambda_J| + \sqrt{\omega^2\lambda_J^2 - 4(\omega - 1)}\right|^2. \tag{10.5.65}$$

If $\omega \leq 1$, then $\omega^2\lambda_J^2 - 4(\omega - 1) \geq 0$ and $\lambda_{\max}(\omega, \lambda_J)$ is an increasing function of $|\lambda_J|$.

If $\omega > 1$, define

$$\lambda_c = \sqrt{\frac{4(\omega - 1)}{\omega^2}}.$$

Then if $|\lambda_J| \leq \lambda_c$, $\lambda_{\max}(\omega, \lambda_J) = \omega - 1$ (so $\lambda_{\max}(\omega, \lambda_J)$ will again be an increasing function of $|\lambda_J|$). And if $|\lambda_J| > \lambda_c$, then $\omega^2\lambda_J^2 - 4(\omega - 1) > 0$ and $\lambda_{\max}(\omega, \lambda_J)$ is an increasing function of $|\lambda_J|$.

Thus in all cases, $\lambda_{\max}(\omega, \lambda_J)$ is a increasing function of $|\lambda_J|$ and

$$\begin{aligned}\sigma(R_{SOR,\omega}) &= \max_{\lambda_J} \lambda_{\max}(\omega, \lambda_J) = \max_{-\bar{\lambda}_J \leq \lambda_J \leq \bar{\lambda}_J} \lambda_{\max}(\omega, \lambda_J) \\ &= \lambda_{\max}(\omega, \bar{\lambda}_J).\end{aligned}$$

This is what we were to prove.

Now that we have a formula for $\sigma(R_{SOR,\omega})$, we are left with the problem of choosing $\omega_b \in (0, 2)$ so that

$$\sigma(R_{SOR,\omega_b}) = \min_{\omega \in (0,2)} \sigma(R_{SOR,\omega}).$$

This is done with the following result.

Proposition 10.5.12 *If*

$$\omega \neq \omega_b = \frac{2}{1+\sqrt{1-\bar{\lambda}_J^2}}, \tag{10.5.66}$$

then

$$\sigma(R_{SOR,\omega}) > \sigma(R_{SOR,\omega_o}) = \omega_b - 1.$$

Before we prove the above proposition, we prove the following lemma.

Lemma 10.5.13

$$\sigma(R_{SOR,\omega}) = \begin{cases} \omega - 1 & \text{if } \omega_b \leq \omega < 2 \\ \frac{1}{4}\left[\omega\bar{\lambda}_J + \sqrt{\omega^2\bar{\lambda}_J^2 - 4(\omega-1)}\right]^2 & \text{if } 0 < \omega \leq \omega_b. \end{cases}$$

Proof: We begin by noting that $\omega^2\bar{\lambda}_J^2 - 4(\omega-1) \geq 0$ for $\omega \in (0,2)$ when

$$\omega \leq \frac{2}{\bar{\lambda}_J^2}\left[1 - \sqrt{1-\bar{\lambda}_J^2}\right] = \frac{2}{1+\sqrt{1-\bar{\lambda}_J^2}} = \omega_b$$

(where the negative sign was chosen in the quadratic formula to ensure that $\omega \in (0,2)$). Otherwise, $\omega^2\bar{\lambda}_J^2 - 4(\omega-1) < 0$. When $\omega^2\bar{\lambda}_J^2 - 4(\omega-1) \geq 0$ $(0 < \omega \leq \omega_b)$,

$$\sigma(R_{SOR,\omega}) = \frac{1}{4}\left[\omega\bar{\lambda}_J + \sqrt{\omega^2\bar{\lambda}_J^2 - 4(\omega-1)}\right]^2,$$

and when $\omega^2\bar{\lambda}_J^2 - 4(\omega-1) < 0$ $(\omega_b < \omega < 2)$,

$$\sigma(R_{SOR,\omega}) = \omega - 1.$$

Proof of Proposition 10.5.12 When $0 < \omega \leq \omega_b$ (so $\omega^2\bar{\lambda}_J^2 - 4(\omega-1) \geq 0$), since

$$\frac{d}{d\omega}\left[\omega\bar{\lambda}_J + \sqrt{\omega^2\bar{\lambda}_J^2 - 4(\omega-1)}\right] = \frac{\bar{\lambda}_J\sqrt{\omega^2\bar{\lambda}_J^2 - 4(\omega-1)} + \omega\bar{\lambda}_J^2 - 2}{\sqrt{\omega^2\bar{\lambda}_J^2 - 4(\omega-1)}},$$

$$\omega\bar{\lambda}_J^2 - 2 < 0$$

and

$$|\omega\bar{\lambda}_J^2 - 2| > \bar{\lambda}_J\sqrt{\omega^2\bar{\lambda}_J^2 - 4(\omega-1)},$$

M_x	M_y	ω_b	$\sigma(R_{SOR,\omega_b})$	$m = -1/\log_{10}\sigma(R_{SOR,\omega_b})$
10	10	1.527864	0.527864	4
100	100	1.939092	0.939092	37
1000	1000	1.993737	0.993737	367
10	100	1.915514	0.915514	27
50	100	1.923583	0.923583	29
50	1000	1.991165	0.991165	260

TABLE 10.5.2. Values of the optimal parameter, spectral radius for optimal SOR, and number of iterations of optimal SOR needed to improve a result by one decimal place for various values of M_x and M_y.

$\sigma(R_{SOR,\omega})$ is decreasing from $\omega = 0$ to $\omega = \omega_b$. Clearly, for $\omega_b \leq \omega < 2$, $\sigma(R_{SOR,\omega}) = \omega - 1$ is increasing. Thus the minimum value of $\sigma(R_{SOR,\omega})$ occurs at $\omega = \omega_b$.

Remark 1: Using the SOR scheme with optimal $\omega = \omega_b$, we make the spectral radius of the iteration matrix $R_{SOR,\omega}$ as small as possible. If we consider our model problem (10.2.3)–(10.2.7), from Example 10.5.1 we see that since

$$\bar{\lambda}_J = \frac{2}{d}\left(\frac{1}{\Delta x^2}\cos\frac{\pi}{M_x} + \frac{1}{\Delta y^2}\cos\frac{\pi}{M_y}\right),$$

the value of the optimal parameter, ω_b, the spectral radius of the optimal SOR iteration matrix, $\sigma(R_{SOR,\omega_b}) = \omega_b - 1$, and the number of iterations of SOR needed to improve a result by one decimal place ($\zeta = 10^{-1}$), $m \approx -1/\log_{10}(\omega_b - 1)$, can all be easily calculated. In Table 10.5.2, we compute the above values for several values of M_x and M_y. If we compare the last column of Table 10.5.2 (the number of iterations of SOR needed to improve a result by one decimal place) with the last column of Table 10.5.1 (the number of iterations of the Jacobi scheme needed to improve a result by one decimal place), we see that optimal SOR converges much faster than Jacobi (and since Gauss-Seidel takes half as many iterations as Jacobi, optimal SOR converges much faster than Gauss-Seidel also). We note that as the grid gets fine (M_x and M_y get large), the number of iterations of optimal SOR begins to get large. Even using optimal SOR, the number of iterations necessary to obtain a highly accurate result can be large. In HW10.5.14, we see that on a 10×10 grid, although the SOR scheme takes many fewer iterations than does the Jacobi scheme, it is nothing like what the comparison of Tables 10.5.2 and 10.5.1 might lead us to believe it should be. As we consider bigger problems (the 50×50 and 100×100 grids) and smaller tolerances (giving both schemes more time in the "asymptotic range"), the results are at least closer to what we would hope they might be.

Remark 2: Consider our model problem (10.2.3)–(10.2.7) with $M_x = M_y = 10$ (and $\bar{\lambda}_J = 0.951057$ and $\omega_b = 1.527865$). The plot of $\sigma(R_{SOR,\omega})$

for $0 < \omega < 2$ is given in Figure 10.5.2. We note that if we have to estimate ω_b, we want to choose a value larger than ω_b if we can (where the change in $\sigma(R_{SOR,\omega})$ is linear and relatively small) rather than a value smaller than ω_b (where the graph is steep). For this reason, when values of ω_b were given in Table 10.5.2, they were always rounded up. Also, we note how much smaller $\sigma(R_{SOR,\omega})$ is at $\omega = \omega_b$ compared to when $\omega = 1$ (Gauss-Seidel). This is why the number of iterates necessary to reduce the error by one decimal place is so much smaller for optimal SOR than for Gauss-Seidel.

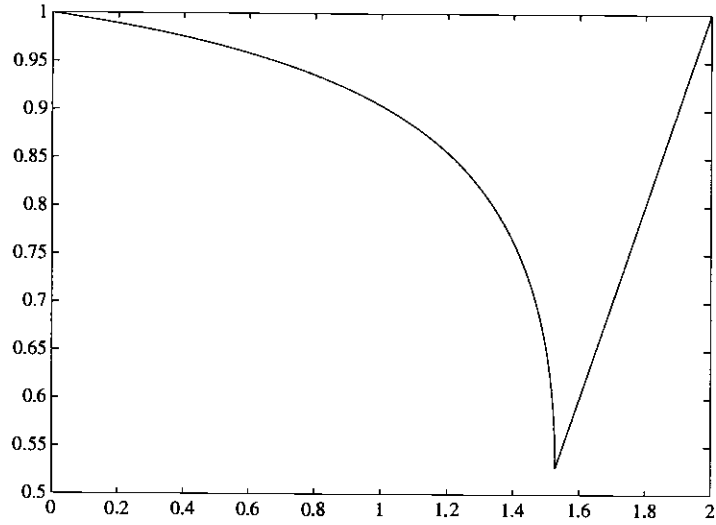

FIGURE 10.5.2. Plot of $\sigma(R_{SOR,\omega})$ for $0 < \omega < 2$ associated with SOR solution scheme for problem (10.2.3)–(10.2.7).

Remark 3: We emphasize that the results given in this section have strong hypotheses (applied to solving difference equation (10.5.13), the variables are ordered in lexicographical order, the eigenvalues of R_J are real, etc.). We shall see in Section 10.5.10 that many of these results are true for a much larger class of problems.

HW 10.5.14 (a) Repeat the solutions done in HW10.5.12 and HW10.5.13 using the optimal value of ω. Compare the number of iterations necessary using ω_b to the number necessary for other values of ω.
(b) Repeat part (a) using $M_x = M_y = 50$ and $M_x = M_y = 100$. In each case, compare the number of iterations necessary for convergence to the number of iterations that were necessary for convergence of the Jacobi scheme (HW10.5.2 and HW10.5.3).

HW 10.5.15 Repeat the solutions done in HW10.5.6 and HW10.5.10 using SOR with the optimal value of ω. Compare the number of iterations necessary to solve the problem using the SOR scheme with the number of iterations that were necessary using the Jacobi and the Gauss-Seidel schemes.

HW 10.5.16 Consider the **weighted Jacobi scheme** for approximating the solution to our model problem (10.2.3)–(10.2.7), which is the scheme that is analogous to the SOR scheme with Gauss-Seidel relaxation replaced by Jacobi relaxation, i.e., consider the scheme

Weighted Jacobi-10.4

For $k = 1, \ldots, M_y - 1$
For $j = 1, \ldots, M_x - 1$

$$\hat{u}_{j\,k} = \frac{1}{d}\left[F_{j\,k} + \frac{1}{\Delta x^2}(u_{j+1\,k} + u_{j-1\,k}) + \frac{1}{\Delta y^2}(u_{j\,k+1} + u_{j\,k-1})\right]$$

$$u'_{j\,k} = u_{j\,k} + \omega\left[\hat{u}_{j\,k} - u_{j\,k}\right]$$

Next j
Next k

(a) Show that the iteration matrix associated with the weighted Jacobi scheme, $R_{J,\omega}$, is given by

$$R_{J,\omega} = (1 - \omega)I + \omega R_J$$

where R_J is the iteration matrix associated with the Jacobi scheme.
(b) Show that the eigenvalues of $R_{J,\omega}$ are give by

$$\lambda_{J,\omega} = \omega\lambda_J + 1 - \omega$$

where λ_J denotes an eigenvalue of R_J.
(c) Show that $|\lambda_{J,\omega}| < 1$ if

$$0 < \omega < \frac{-2}{-1 + \lambda_{M_y-1}^{M_x-1}}$$

where $\lambda_{M_y-1}^{M_x-1}$ is given by (10.5.28), i.e., the weighted Jacobi scheme converges for $0 < \omega \leq 1$.

10.5.9.1 SSOR Scheme

In this section we present a slight variation of the SOR scheme, referred to as the **symmetric, successive overrelaxation scheme**, SSOR. For more information on the SSOR scheme, see [22], page 30, or [13], page 513. One iteration of the SSOR scheme for solving difference equation (10.5.13) along with Dirichlet boundary conditions can be written as follows.

SSOR-10.5.13

For $k = 1, \ldots, M_y - 1$
For $j = 1, \ldots, M_x - 1$
$\hat{u}_{j\,k} = -\frac{1}{\beta^0_{j\,k}}\left[F_{j\,k} - \beta^1_{j\,k}u_{j+1\,k} - \beta^2_{j\,k}u'_{j-1\,k} - \beta^3_{j\,k}u_{j\,k+1} - \beta^4_{j\,k}u'_{j\,k-1}\right]$
$u'_{j\,k} = u_{j\,k} + \omega\left[\hat{u}_{j\,k} - u_{j\,k}\right]$
Next j
Next k

For $k = M_y - 1, \ldots, 1$
For $j = M_x - 1, \ldots, 1$
$\hat{u}'_{j\,k} = -\frac{1}{\beta^0_{j\,k}}\left[F_{j\,k} - \beta^1_{j\,k}u''_{j+1\,k} - \beta^2_{j\,k}u'_{j-1\,k} - \beta^3_{j\,k}u''_{j\,k+1} - \beta^4_{j\,k}u'_{j\,k-1}\right]$
$u''_{j\,k} = u'_{j\,k} + \omega\left[\hat{u}'_{j\,k} - u'_{j\,k}\right]$
Next j
Next k

Inspection of the SSOR scheme described above shows that the SSOR scheme is two SOR iterations, one in the usual j-k order and one in the reverse j-k order. Intuitively, the SSOR scheme is a pleasing scheme in that it reduces the influence of starting each iteration in the $(1,1)$ corner and ending it in the $(M_x - 1, M_y - 1)$ corner. However, there are more explicit advantages to the SSOR scheme. We begin by noting that the first step in the SSOR scheme can be written as

$$\begin{aligned}\hat{\mathbf{w}}_{k+1/2} &= D^{-1}\left[\mathbf{f} - L\mathbf{w}_{k+1/2} - U\mathbf{w}_k\right]\\ \mathbf{w}_{k+1/2} &= (1-\omega)\mathbf{w}_k + \omega\hat{\mathbf{w}}_{k+1/2}.\end{aligned}$$

Following the calculation done in Section 10.5.9 we see that the first step of the SSOR scheme can also be written as

$$\mathbf{w}_{k+1/2} = \mathbf{w}_k + B_1\mathbf{r}_k, \tag{10.5.67}$$

where $B_1 = \omega(D + \omega L)^{-1}$. Analogously, the second step of the SSOR scheme can be written as

$$\begin{aligned}\hat{\mathbf{w}}_{k+1} &= D^{-1}\left[\mathbf{f} - L\mathbf{w}_{k+1/2} - U\mathbf{w}_{k+1}\right]\\ \mathbf{w}_{k+1} &= (1-\omega)\mathbf{w}_{k+1/2} + \omega\hat{\mathbf{w}}_{k+1},\end{aligned}$$

or

$$\mathbf{w}_{k+1} = \mathbf{w}_{k+1/2} + B_2\mathbf{r}_{k+1/2}, \tag{10.5.68}$$

where $B_2 = \omega(D + \omega U)^{-1}$. It is not hard to see that

$$\mathbf{w}_{k+1} = \mathbf{w}_{k+1/2} + B_2\mathbf{r}_{k+1/2}$$

$$
\begin{aligned}
&= \mathbf{w}_{k+1/2} + B_2(\mathbf{f} - A\mathbf{w}_{k+1/2}) \\
&= \mathbf{w}_k + B_1\mathbf{r}_k + B_2\left(\mathbf{f} - A\mathbf{w}_k - AB_1\mathbf{r}_k\right) \\
&= \mathbf{w}_k + \left(B_1 + B_2 - B_2AB_1\right)\mathbf{r}_k \\
&= \mathbf{w}_k + B\mathbf{r}_k,
\end{aligned}
$$

where $B = B_1 + B_2 - B_2AB_1$. Using the form of B_1 and B_2, we see that B can be written as

$$
\begin{aligned}
B &= B_1 + B_2 - B_2AB_1 \\
&= \omega(D + \omega L)^{-1} + \omega(D + \omega U)^{-1} \\
&\quad -\omega^2(D + \omega U)^{-1}(L + D + U)(D + \omega L)^{-1} \\
&= \omega(D + \omega U)^{-1}\Big\{(D + \omega U)(D + \omega L)^{-1} + I \\
&\qquad\qquad - \omega(L + D + U)(D + \omega L)^{-1}\Big\} \\
&= \omega(D + \omega U)^{-1}\left\{(2 - \omega)D\right\}(D + \omega l)^{-1} \\
&= \omega(2 - \omega)(D + \omega U)^{-1}D(D + \omega L)^{-1}. \qquad (10.5.69)
\end{aligned}
$$

We note that this is the same form as when we gave SSOR as a residual correction scheme in Section 10.5. From the representation given above, we see that the iteration matrix for the SSOR scheme can be written as

$$
R_{SSOR} = I - BA = I - (B_1 + B_2 - B_2AB_1)A = (I - B_2A)(I - B_1A).
$$

Hence, we see that the iteration matrix is given as the product of the two iteration matrices of the forward and backward SOR sweeps and is given by

$$
R_{SSOR} = (D + \omega U)^{-1}\left[(1 - \omega)D - \omega L\right](D + \omega L)^{-1}\left[(1 - \omega)D - \omega U\right]. \qquad (10.5.70)
$$

As with the SOR scheme, it would be very helpful to be able to find the value of ω that made $\sigma(R_{SSOR})$ optimally small. However, finding the optimum ω for the SSOR scheme is not as easy as it was for the SOR scheme, and *the rate of convergence of the SSOR scheme is not very sensitive to the exact choice of optimal* ω. Hence, the approach used is that if

$$
\sigma(D^{-1}LD^{-1}U) \leq \frac{1}{4}, \qquad (10.5.71)
$$

then an acceptable approximation to the optimal value of ω is given by

$$
\omega = \frac{2}{1 + \sqrt{2(1 - \bar{\lambda}_J)}}, \qquad (10.5.72)
$$

where $\bar{\lambda}_J$ is the maximum eigenvalue of the Jacobi iteration matrix R_J (see ref. [22], page 31). With this choice of ω, it can be shown that

$$\sigma(R_{SSOR}) \leq \frac{1 - \sqrt{\frac{1-\bar{\lambda}_J}{2}}}{1 + \sqrt{\frac{1-\bar{\lambda}_J}{2}}} \tag{10.5.73}$$

(ref. [22], page 32).

The argument given above is really much nicer than we might first suspect. It is the case that we can find the optimal parameter for the SOR scheme for some easy problems, and we cannot do this for the SSOR scheme. However, when we cannot analytically find the optimal parameter for the SOR scheme (which is often, when we consider problems that people might really be willing to pay us money to solve), we must proceed to approximate ω_b (see Section 10.5.12). Since the SOR scheme is very sensitive to the value of ω_b, it is important that we obtain a good approximation. Since the SSOR scheme is not very sensitive to the value of ω_b, it is sufficient to use the approximation given in (10.5.72).

One of the properties of the SSOR iteration scheme that we will use later involves the fact that if the matrix A is symmetric (i.e., $L^T = U$ and $U^T = L$), then $B = \omega(2-\omega)(D+\omega U)^{-1}D(D+\omega L)^{-1}$ is symmetric. This can be seen by considering

$$\begin{aligned} B^T &= \omega(2-\omega)\left[(D+\omega U)^{-1}D(D+\omega L)^{-1}\right]^T \\ &= \omega(2-\omega)(D^T+\omega L^T)^{-1}D^T(D^T+\omega U^T)^{-1} \\ &= \omega(2-\omega)(D+\omega U)^{-1}D(D+\omega L)^{-1} \\ &= B. \end{aligned}$$

We will take advantage of this property of the SSOR scheme in Section 10.13.2, where use the SSOR scheme as the preconditioner for the conjugate gradient scheme.

HW 10.5.17 Resolve the problem described in HW10.5.14 using SSOR with ω given by formula (10.5.72). Compare the number of iterations necessary to obtain the solution with SSOR to the number of iterations necessary to obtain the solution using optimal SOR (HW10.5.14).

HW 10.5.18 Resolve the problem described in HW10.5.15 using SSOR with ω given by formula (10.5.72). Compare the number of iterations necessary to obtain the solution with SSOR to the number of iterations necessary to obtain the solution using optimal SOR (HW10.5.14 and HW10.5.15).

10.5.9.2 Red-Black Ordering

We saw in the back sweep of the SSOR scheme that by sweeping the data in a different order, we get different properties in our scheme. There are many potential orders in which we can sweep through our data. It should be fairly clear that since the Jacobi scheme relies entirely on old values (values from the previous iteration), *the Jacobi scheme is independent of the order in which the points are processed.*

In this section, we introduced an alternative to the j-k lexicographical ordering, **red-black ordering**. We begin by noting in Figure 10.5.3 that starting with the $(0,0)$ point, every second point in each row is labeled with an R (red). The other points are labeled with a B (black). For the obvious reason, red-black ordering is sometimes referred to as checkerboard ordering. Thus, we will perform our calculations while the points are ordered as

$$(0,0),\ (2,0),\ldots,(M_x,0),(1,1),\ldots,(M_x-1,1),(0,2),\ldots,(M_x,M_y),$$
$$(1,0),\ldots,(M_x-1,0),(0,1),\ldots,(M_x,1),\ldots,(M_x-1,M_y).$$

(We note, for convenience, that we have assumed that M_x and M_y are even. This is not necessary, but it is nice.) Of course, often we will really not process all of these points. For Dirichlet boundary conditions, we will start processing the points beginning with the $(1,1)$ point and not include the boundary points. We might note that the red-black ordering has been around a long time, but it has become very popular and important for implementing iterative schemes on vector and parallel computers.

If we again consider solving difference equation (10.5.13) on the grid given in Figure 10.5.3, we see that the only change in the equations is some sort of change in the numbering (which we will not attempt to make clear). The difference equation will still reach to the point at which it is defined and its four nearest neighbors. And, as we mentioned earlier, if we consider solving this difference equation by a Jacobi iteration with red-black ordering, there will be no difference in the results from the lexicographical ordering because the Jacobi scheme depends only on the old values.

Consider using the Gauss-Seidel iterative scheme with the red-black ordering for solving difference equation (10.5.13). Note that as we sweep through the red points, we have only old information available at all of the four nearest neighbors of the red points. Thus we use all old information on the red sweep. However, when we sweep through the black points, we see that we have new information at all of the neighbors of the black points. Hence, the black sweep uses all new information. The algorithm for one iteration for solving difference equation (10.5.13) using red-black Gauss-Seidel can be given as follows.

R-B Gauss-Seidel-10.5.13

For $k = 1,\ldots,M_y - 1$
For $j = rs_k,\ldots,re_k,2$

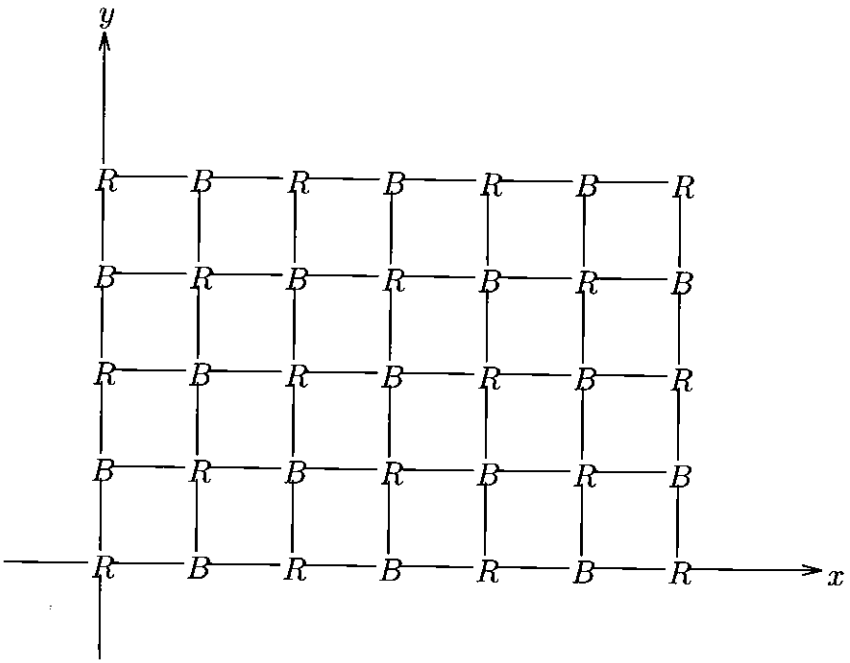

FIGURE 10.5.3. A grid with red-black ordering on $[0,1] \times [0,1]$.

$$u'_{j\,k} = -\tfrac{1}{\beta^0_{j\,k}}\left[F_{j\,k} - \beta^1_{j\,k}u_{j+1\,k} - \beta^2_{j\,k}u_{j-1\,k} - \beta^3_{j\,k}u_{j\,k+1} - \beta^4_{j\,k}u_{j\,k-1}\right]$$

Next j
Next k

For $k = 1, \ldots, M_y - 1$
For $j = bs_k, \ldots, be_k, 2$

$$u'_{j\,k} = -\tfrac{1}{\beta^0_{j\,k}}\left[F_{j\,k} - \beta^1_{j\,k}u'_{j+1\,k} - \beta^2_{j\,k}u'_{j-1\,k} - \beta^3_{j\,k}u'_{j\,k+1} - \beta^4_{j\,k}u'_{j\,k-1}\right]$$

Next j
Next k

If we assume that M_x is even and we are solving a problem with Dirichlet boundary conditions, then $rs_k = 1$ when k is odd, $rs_k = 2$ when k is even, $re_k = M_x - 1$ when k is odd, $re_k = M_x - 2$ when k is even, $bs_k = 2$ when k is odd, $bs_k = 1$ when k is even, $be_k = M_x - 2$ when k is odd, and $be_k = M_x - 1$ when k is even. The 2's in lines two and seven indicate a skip of two. Of course, for problems with Neumann or mixed boundary conditions, an adjustment must be made.

We note that red-black Gauss-Seidel is not too different from the usual Gauss-Seidel with lexicographical ordering. The usual Gauss-Seidel uses

two new values and two old values at each point in the iteration, i.e., half of the information used is old and half of the information used is new. Red-black Gauss-Seidel uses all old information on the red sweep and all new information on the black sweep, i.e., half of the information used is old and half of the information used is new. An intuitive computational difference between the Gauss-Seidel scheme for the two orderings is that the lexicographical ordering tends to sweep the errors into the (M_x, M_y) corner, while the red-black ordering tends to distribute the errors uniformly.

If we write difference equation (10.5.13) along with Dirichlet boundary conditions as a matrix equation using the red-black ordering, we get

$$\begin{pmatrix} D_r & B_1 \\ B_2 & D_b \end{pmatrix} \begin{bmatrix} \mathbf{u}_r \\ \mathbf{u}_b \end{bmatrix} = \begin{bmatrix} \mathbf{f}_r \\ \mathbf{f}_b \end{bmatrix}, \tag{10.5.74}$$

where $\mathbf{u}_r$ and $\mathbf{u}_b$ represent the unknowns, partitioned as red and black,

$$\mathbf{u}_r = \begin{bmatrix} u_{1\,1} \\ u_{3\,1} \\ \vdots \\ u_{M_x-1\,1} \\ u_{2\,2} \\ \vdots \\ u_{M_x-1\,M_y-1} \end{bmatrix} \quad \text{and} \quad \mathbf{u}_b = \begin{bmatrix} u_{2\,1} \\ u_{4\,1} \\ \vdots \\ u_{M_x-2\,1} \\ u_{1\,2} \\ \vdots \\ u_{M_x-2\,M_y-1} \end{bmatrix},$$

D_r and D_b are the diagonal matrices containing the β^0 terms for the red and black points, respectively,

$$D_r = \begin{pmatrix} -\beta^0_{1\,1} & 0 & \cdots & & & & & \\ 0 & -\beta^0_{3\,1} & 0 & \cdots & & & & \\ & \ddots & \ddots & \ddots & & & & \\ 0 & \cdots & 0 & -\beta^0_{M_x-1\,1} & 0 & \cdots & & \\ & & \cdots & 0 & -\beta^0_{2\,2} & 0 & \cdots & \\ & & & & \ddots & \ddots & \ddots & \\ & & & & & \cdots & -\beta^0_{M_x-1\,M_y-1} \end{pmatrix},$$

$$D_b = \begin{pmatrix} -\beta^0_{2\,1} & 0 & \cdots & & & & & \\ 0 & -\beta^0_{4\,1} & 0 & \cdots & & & & \\ & \ddots & \ddots & \ddots & & & & \\ 0 & \cdots & 0 & -\beta^0_{M_x-2\,1} & 0 & \cdots & & \\ & & \cdots & 0 & -\beta^0_{1\,2} & 0 & \cdots & \\ & & & & \ddots & \ddots & \ddots & \\ & & & & & \cdots & -\beta^0_{M_x-2\,M_y-1} \end{pmatrix},$$

B_1 and B_2 represent how difference equation (10.5.13) reaches to its neighbors and are given in Figures 10.5.4 and 10.5.5 (for the starting points and ending points in the vectors and matrices given above and below, M_x and M_y have been assumed even), and $\mathbf{f}_r$ and $\mathbf{f}_b$ are the vectors containing the contribution due to the nonhomogeneous term F and the boundary conditions.

Applying the Gauss-Seidel scheme, (10.5.44), to equation (10.5.74), we see that we must solve

$$\begin{pmatrix} D_r & \Theta \\ B_2 & D_b \end{pmatrix} \begin{bmatrix} \mathbf{u}_{rk+1} \\ \mathbf{u}_{bk+1} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_r \\ \mathbf{f}_b \end{bmatrix} - \begin{pmatrix} \Theta & B_1 \\ \Theta & \Theta \end{pmatrix} \begin{bmatrix} \mathbf{u}_{rk} \\ \mathbf{u}_{bk} \end{bmatrix}. \tag{10.5.75}$$

Multiplying these equations, we see that solving equation (10.5.75) is the same as solving

$$D_r \mathbf{u}_{rk+1} = \mathbf{f}_r - B_1 \mathbf{u}_{bk} \tag{10.5.76}$$

$$B_2 \mathbf{u}_{rk+1} + D_b \mathbf{u}_{bk+1} = \mathbf{f}_b. \tag{10.5.77}$$

Thus we see that if we perform these operations in the correct order (first the red computations and then the black computations), the solution to equation (10.5.75) can be given by

$$\mathbf{u}_{rk+1} = D_r^{-1}\mathbf{f}_r - D_r^{-1}B_1\mathbf{u}_{bk} \tag{10.5.78}$$

$$\mathbf{u}_{bk+1} = D_b^{-1}\mathbf{f}_b - D_b^{-1}B_2\mathbf{u}_{rk+1}. \tag{10.5.79}$$

It is not difficult to see that the solution given by (10.5.78)–(10.5.79) is the same as that given by Algorithm R-B Gauss-Seidel-10.5.13. In HW10.5.20, we see that the solution given by (10.5.78)–(10.5.79) is also the same as that given by inverting

$$L + D = \begin{pmatrix} D_r & \Theta \\ B_2 & D_b \end{pmatrix}$$

in equation (10.5.75).

Since we have a Gauss-Seidel scheme for the red-black ordering, an obvious question is whether it is advantageous to develop a red-black SOR scheme. The red-black SOR scheme is given as follows.

R-B SOR-10.5.13

For $k = 1, \ldots, M_y - 1$

For $j = rs_k, \ldots, re_k, 2$

$\hat{u}_{j\,k} = -\frac{1}{\beta^0_{j\,k}}\left[F_{j\,k} - \beta^1_{j\,k}u_{j+1\,k} - \beta^2_{j\,k}u_{j-1\,k} - \beta^3_{j\,k}u_{j\,k+1} - \beta^4_{j\,k}u_{j\,k-1}\right]$

$u'_{j\,k} = u_{j\,k} + \omega\left[\hat{u}_{j\,k} - u_{j\,k}\right]$

Next j

Next k

For $k = 1, \ldots, M_y - 1$

$$
\begin{pmatrix}
\beta^1_{1\,1} & 0 & \cdots & 0 & \beta^3_{1\,1} & 0 & \cdots & & & & & & \\
\beta^2_{3\,1} & \beta^1_{3\,1} & 0 & \cdots & 0 & \beta^3_{3\,1} & 0 & \cdots & & & & & \\
 & & & & & & \cdots & & & & & & \\
0 & \cdots & 0 & \beta^2_{M_x-3\,1} & \beta^1_{M_x-3\,1} & 0 & \cdots & 0 & \beta^3_{M_x-3\,1} & 0 & \cdots & & \\
0 & & \cdots & 0 & \beta^2_{M_x-1\,1} & \mathbf{0} & 0 & \cdots & 0 & \beta^3_{M_x-1\,1} & 0 & \cdots & \\
\beta^4_{2\,2} & 0 & \cdots & & 0 & \beta^2_{2\,2} & \beta^1_{2\,2} & 0 & \cdots & 0 & \beta^3_{2\,2} & 0 & \cdots \\
 & & & & & & \cdots & & & & & & \\
 & & & & 0 & \cdots & 0 & \beta^4_{M_x-1\,M_y-1} & 0 & \cdots & & 0 & \beta^2_{M_x-1\,M_y-1}
\end{pmatrix}
$$

FIGURE 10.5.4. Matrix B_1 associated with the red-black Gauss-Seidel iteration scheme. The boldface zero represents a zero that is due to a boundary condition term.

$$
\begin{pmatrix}
\beta^2_{2\,1} & \beta^1_{2\,1} & 0 & \cdots & 0 & \beta^3_{2\,1} & 0 & \cdots & & & & & & & \\
0 & \beta^2_{4\,1} & \beta^1_{4\,1} & 0 & \cdots & 0 & \beta^3_{4\,1} & 0 & \cdots & & & & & & \\
 & & & & & & & \cdots & & & & & & & \\
0 & \cdots & 0 & \beta^2_{M_x-4\,1} & \beta^1_{M_x-4\,1} & 0 & \cdots & 0 & \beta^3_{M_x-4\,1} & 0 & \cdots & & & & \\
0 & & \cdots & 0 & \beta^2_{M_x-2\,1} & \beta^1_{M_x-2\,1} & 0 & \cdots & 0 & \beta^3_{M_x-2\,1} & 0 & \cdots & & & \\
\beta^4_{1\,2} & 0 & \cdots & & 0 & \mathbf{0} & \beta^1_{1\,2} & 0 & \cdots & 0 & \beta^3_{1\,2} & 0 & \cdots & & \\
 & & & & & & & \cdots & & & & & & & \\
 & & & 0 & \cdots & 0 & \beta^4_{M_x-2\,M_y-1} & 0 & \cdots & 0 & \beta^2_{M_x-2\,M_y-1} & \beta^1_{M_x-2\,M_y-1} & &
\end{pmatrix}
$$

FIGURE 10.5.5. Matrix B_2 associated with the red-black Gauss-Seidel iteration scheme. The boldface zero represents a zero due to a boundary condition term.

For $j = bs_k, \ldots, be_k, 2$

$\hat{u}_{j\,k} = -\frac{1}{\beta^0_{j\,k}}\left[F_{j\,k} - \beta^1_{j\,k}u'_{j+1\,k} - \beta^2_{j\,k}u'_{j-1\,k} - \beta^3_{j\,k}u'_{j\,k+1} - \beta^4_{j\,k}u'_{j\,k-1}\right]$

$u'_{j\,k} = u_{j\,k} + \omega\left[\hat{u}_{j\,k} - u_{j\,k}\right]$

Next j

Next k

In Section 10.5.10 we will see that the eigenvalues for the iteration matrices for red-black Gauss-Seidel and red-black SOR are the same as those for the lexicographically ordered (j-k ordered) analogues. Hence, the convergence properties of red-black Gauss-Seidel and red-black SOR are the same as those for Gauss-Seidel and SOR in j-k ordering. Thus we see that the red–black Gauss-Seidel converges twice as fast as the Jacobi iteration, and the optimal red-black SOR (where the optimal parameter, ω_b, is still given by formula (10.5.66)) converges much faster than either the Jacobi iteration or the red-black Gauss-Seidel iteration. If the problem is such that the optimal parameter can be found (the eigenvalues of the Jacobi iteration matrix can be found) and we can gain an advantage from the red-black ordering (say to parallelize the code), the red-black SOR can be competitive with the best of schemes.

We might remark that another logical choice of schemes is to consider the red-black ordering with the SSOR scheme. We do not gain any advantage from the red-black SSOR scheme so, we omit any discussion here.

HW 10.5.19 Inverse of a block matrix: Use the block Sherman-Morrison algorithm, Section 6.2.3.3, Part 1, to show that if $\mathcal{A}$ is given by

$$\mathcal{A} = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$$

where A is an $N \times N$ invertible matrix, D is an $M \times M$ invertible matrix, and the inverse of

$$F = I - D^{-1}CA^{-1}B$$

exists, then $\mathcal{A}^{-1}$ exists and is given by

$$\mathcal{A}^{-1} = \begin{pmatrix} A^{-1}\left[I + BF^{-1}D^{-1}CA^{-1}\right] & -A^{-1}BF^{-1}D^{-1} \\ -F^{-1}D^{-1}CA^{-1} & F^{-1}D^{-1} \end{pmatrix}. \quad (10.5.80)$$

HW 10.5.20 (a) Use the formula given in (10.5.80), HW10.5.19, for the inverse of the 2×2 block matrix to show that the solution to equation (10.5.75) can be written as

$$\begin{bmatrix} \mathbf{u}_{r\,k+1} \\ \mathbf{u}_{b\,k+1} \end{bmatrix} = \begin{bmatrix} D_r^{-1}\mathbf{f}_r - D_r^{-1}B_1\mathbf{u}_{b\,k} \\ -D_b^{-1}B_2D_r^{-1}\mathbf{f}_r + D_b^{-1}\mathbf{f}_b + D_b^{-1}B_2D_r^{-1}B_1\mathbf{u}_{b\,k} \end{bmatrix}.$$

(b) Show that the solution found in part (a) is the same as that given in (10.5.78)–(10.5.79).

HW 10.5.21 Solve once again the problem (solved several times by now, we hope) given in HW10.5.2 using both red-black Gauss-Seidel and red-black SOR. For the red-black SOR calculations, use $\omega = 0.5$, 1.5, 1.75 and ω_b as found in HW10.5.14. Compare and contrast your solutions (iteration counts) with the iteration counts found in HW10.5.12 and HW10.5.14.

10.5.10 More on the SOR Scheme

In Section 10.5.9 we stated that it is possible to get results for the SOR scheme for a much larger class of problems. We do not really care about most of the problems in this larger class. However, there are several of the extensions that are very useful for solving partial differential equations numerically. We begin by reintroducing the SOR scheme in a more general setting. We then state some results concerning the convergence of the SOR scheme and the choice of the optimal parameter. And finally, we use these new results to give us some of the useful extensions of the SOR scheme. Several basic references on iterative schemes in general and SOR schemes in particular are [22], [75], and [70].

We consider the $L \times L$ matrix equation

$$A\mathbf{u} = \mathrm{f} \tag{10.5.81}$$

that has been partitioned into the form

$$A = \begin{pmatrix} A_{1\,1} & A_{1\,2} & \cdots & A_{1\,L_1} \\ A_{2\,1} & A_{2\,2} & \cdots & A_{2\,L_1} \\ \vdots & \vdots & & \vdots \\ A_{L_1\,1} & A_{L_1\,2} & \cdots & A_{L_1\,L_1} \end{pmatrix}, \tag{10.5.82}$$

$$\mathbf{u} = \begin{bmatrix} \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_{L_1} \end{bmatrix} \text{ and } \mathrm{f} = \begin{bmatrix} \mathrm{f}_1 \\ \vdots \\ \mathrm{f}_{L_1} \end{bmatrix}, \tag{10.5.83}$$

where $A_{j\,k}$ are $m_j \times m_k$ matrices, $\mathbf{u}_k$ are m_k-vectors, f_j are m_j-vectors, and $\sum_{j=1}^{L_1} m_j = L$. We write A as $L + D + U$ where L, D, and U are the lower block triangular part of A, the block diagonal of A, and the upper block triangular part of A, respectively,

$$L = \begin{pmatrix} \Theta & \Theta & \cdots & \Theta \\ A_{2\,1} & \Theta & \cdots & \Theta \\ \vdots & \vdots & \ddots & \vdots \\ A_{L_1\,1} & \cdots & A_{L_1\,L_1-1} & \Theta \end{pmatrix}, \tag{10.5.84}$$

$$D = \begin{pmatrix} A_{11} & \Theta & \cdots & \\ \Theta & A_{22} & \Theta & \cdots \\ \vdots & \ddots & \ddots & \ddots \\ & \cdots & \Theta & A_{L_1 L_1} \end{pmatrix}, \tag{10.5.85}$$

$$U = \begin{pmatrix} \Theta & A_{12} & A_{13} & \cdots & A_{1 L_1} \\ \Theta & \Theta & A_{23} & \cdots & A_{2 L_1} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \Theta & \cdots & \cdots & \Theta & \Theta \end{pmatrix}. \tag{10.5.86}$$

Using the notation given above, one iteration of the Jacobi, Gauss-Seidel, and SOR schemes can be written as follows.

Block Jacobi-10.5.81

For $k = 1, \ldots, L_1$

solve

$$A_{k\,k}\mathbf{u}'_k = -\sum_{\substack{j=1 \\ j \neq k}}^{L_1} A_{k\,j}\mathbf{u}_j + \mathbf{f}_k$$

Next k

Block Gauss-Seidel-10.5.81

For $k = 1, \ldots, L_1$

solve

$$A_{k\,k}\mathbf{u}'_k = -\sum_{j=1}^{k-1} A_{k\,j}\mathbf{u}'_j - \sum_{j=k+1}^{L_1} A_{j\,k}\mathbf{u}_j + \mathbf{f}_k$$

Next k

Block SOR-10.5.81

For $k = 1, \ldots, L_1$

solve

$$A_{k\,k}\hat{\mathbf{u}}_k = -\sum_{j=1}^{k-1} A_{k\,j}\mathbf{u}'_j - \sum_{j=k+1}^{L_1} A_{k\,j}\mathbf{u}_j + \mathbf{f}_k$$

$$\mathbf{u}'_k = \omega\hat{\mathbf{u}}_k + (1-\omega)\mathbf{u}_k$$

Next k

As before, the iteration matrices for the Jacobi, Gauss-Seidel, and SOR schemes are

$$R_J = -D^{-1}(L+U) \tag{10.5.87}$$

$$R_{GS} = -(D+L)^{-1}U \tag{10.5.88}$$

$$R_{SOR} = (I+\omega D^{-1}L)^{-1}[(1-\omega)I - \omega D^{-1}U]. \tag{10.5.89}$$

Besides conditions on the matrix A and the submatrices A_{jk} that will ensure that the relaxation schemes converge, we are interested in whether the order in which the equations and variables are processed is important. We saw in Section 10.5.9.2 that by using a red–black ordering, we are able to uncouple the red and black calculations to make a scheme that is highly vectorizable and parallelizable. We might ask whether there are other orderings that can give us some advantages and whether there is any reason to believe that these different orderings will converge. We begin with the following two definitions concerning the ordering.

Definition 10.5.14 *The matrix A is said to satisfy* **property** A *if there exist subsets of $\mathbb{N}$ (set of positive integers), S_r and S_b, where $S_r \cup S_b = \{1,\ldots,L_1\}$, such that if $A_{jk} \neq \Theta$ and $j \neq k$, then $j \in S_r$ and $k \in S_b$ or $j \in S_b$ and $k \in S_r$.*

Definition 10.5.15 *The matrix A is consistently ordered if there exist subsets of $\mathbb{N}$, $S_1,\ldots,S_q$, where $\cup_{m=1}^{q} S_m = \{1,\ldots,L_1\}$, such that if $A_{jk} \neq \Theta$, $j \neq k$, and $j \in S_m$, then*
(i) *if $k > j$, $k \in S_{m+1}$;*
(ii) *if $k < j$, $k \in S_{m-1}$.*

Remark 1: We first note that both definitions are made with respect to an initially imposed partitioning of A. In general, a result that is obtained for a given partitioning or ordering of the matrix A may not be true for some other partitioning or ordering of A.

Remark 2: Consider the matrix associated with the five point difference scheme for an elliptic equation with respect to a lexicographical ordering j-k (difference scheme (10.5.13) along with Dirichlet boundary conditions). Renumber the ordering as

$$\begin{array}{ccccccccc} (1,1) & (2,1) & \cdots & (M_x-1,1) & (1,2) & \cdots & (M_x-1, M_y-1) \\ \downarrow & \downarrow & \cdots & \downarrow & \downarrow & \cdots & \downarrow \\ 1 & 2 & \cdots & M_x-1 & M_x & \cdots & (M_x-1)(M_y-1). \end{array} \tag{10.5.90}$$

If we define $s_m = \{\ell : \text{the } j\text{-}k \text{ coordinates associated with the index } \ell \text{ satisfy } j+k=m\}$, $m = 2,\ldots,M_x+M_y-2$, then the matrix associated with the five point difference scheme given in lexicographical ordering can be seen to be consistently ordered. It might be noted that if the points in the domain are given as in Figure 10.1.1, the sets S_m are defined by grouping

points together that lie on diagonals with slope $-\Delta y/\Delta x$ (considering only the interior points).

Remark 3: We also mention that the r and b notation given in Definition 10.5.14 is not accidental. It is intended to look like a red-black ordering. If we we use the numbering of the lexicographical ordering given above in Remark 2, then the red-black ordering can be given as (remember that we are only listing the interior red-black points)

$$1, 3, \ldots, M_x - 1, M_x + 1, \ldots, (M_x - 1)(M_y - 1), 2, \ldots, M_x - 2,$$
$$M_x, \ldots, (M_x - 1)(M_y - 1) - 1.$$

(Note that we are again assuming M_x and M_y to be even in the ordering given above.) The matrix based on this ordering is still that given in (10.5.74). All of the elements that satisfy $j \neq k$ and $a_{jk} \neq 0$ lie in the B_1 and B_2 blocks. Thus if we let $S_r = \{1, 3, M_x - 1, M_x + 1, \ldots, (M_x - 1)(M_y - 1)\}$ and $S_b = \{2, 4, M_x - 2, M_x, \ldots, (M_x - 1)(M_y - 1) - 1\}$, it is easy to see that A satisfies property $\mathcal{A}$. (This is very easy to see if we consider the matrix formulation of red-black SOR given in (10.5.74).) Generally, we might say that a matrix satisfies property $\mathcal{A}$ if it can be transformed into a red-black ordering.

Remark 4: Similar to the relationship between the definition of property $\mathcal{A}$ and the red-black matrix, the definition of a consistently ordered matrix is a generalization of being a tridiagonal matrix. If we consider the block tridiagonal matrix

$$\begin{pmatrix} A_{11} & A_{12} & \Theta & \cdots & & \\ A_{21} & A_{22} & A_{23} & \Theta & \cdots & \\ \Theta & A_{32} & A_{33} & A_{34} & \Theta & \cdots \\ & \ddots & \ddots & \ddots & \ddots & \ddots \\ & \cdots & \Theta & A_{L_1-1\,L_1-2} & A_{L_1-1\,L_1-1} & A_{L_1-1\,L_1} \\ & & \cdots & \Theta & A_{L_1\,L_1-1} & A_{L_1\,L_1} \end{pmatrix},$$

by defining $S_m = \{m\}$, $m = 1, \ldots, L_1$, it is not hard to see that the matrix is consistently ordered.

A very important relationship between property $\mathcal{A}$ and the consistently ordered definitions is the following.

Proposition 10.5.16 *If the matrix A is consistently ordered, then the matrix A satisfies property $\mathcal{A}$.*

Proof: If we assume that the matrix A is consistently ordered (so that we have the sets S_m, $m = 1, \ldots, q$) and let
$S_r = S_1 \cup S_3 \cup \cdots \cup S_{L_1}$, and
$S_b = S_2 \cup S_4 \cup \cdots \cup S_{L_1-1}$ (assuming that L_1 is odd),
it is easy to see that the matrix A will satisfy property $\mathcal{A}$.

Remark 1: To see that the converse of Proposition 10.5.16 is not true, we consider the matrix (a block matrix of this form would work equally well)

$$A = \begin{pmatrix} a_{11} & a_{12} & 0 & a_{14} \\ a_{21} & a_{22} & a_{23} & 0 \\ 0 & a_{32} & a_{33} & a_{34} \\ a_{41} & 0 & a_{43} & a_{44} \end{pmatrix}.$$

Note that this was the sort of matrix that we obtained when we solved partial differential equations with periodic boundary conditions. It is easy to see that if we set $s_r = \{1,3\}$ and $s_b = \{2,4\}$, the matrix A will satisfy property $\mathcal{A}$ (just test each $m = 1,2,3,4$ separately).

We show by contradiction that A is not consistently ordered. Hence, we assume that A is consistently ordered so there are subsets of $\mathbb{N}$, $S_1, \ldots, S_q$, that satisfy Definition 10.5.15. Let m be such that $1 \in S_m$. Then by considering the first row, we see that $2,4 \in S_{m+1}$. Thus we see that $S_m = \{1, \ldots\}$ and $S_{m+1} = \{2,4, \ldots\}$.

Now, since $2 \in S_{m+1}$, considering the second row, we see that $3 \in S_{m+2}$. Thus we know that $S_m = \{1\}$, $S_{m+1} = \{2,4\}$, and $S_{m+2} = \{3\}$. If we finally consider the third row, we see that $4 \in S_{m+3}$. Clearly, this is a contradiction, and we have that A is not consistently ordered.

Remark 2: Earlier, we saw that the *matrix associated with the red-black ordering for the five point scheme satisfies property* $\mathcal{A}$. To see that the red-black ordering also is consistently ordered, we proceed as we did in showing that it satisfies property $\mathcal{A}$, setting $S_1 = S_r$ and $S_2 = S_b$. Then for any $j \in S_1$, the only values of $k \neq j$ for which $A_{jk} \neq \Theta$ are the k's in $S_b = S_2$. Likewise, for j in S_2, the only values of $k \neq j$ for which $A_{jk} \neq \Theta$ are the k's in $S_r = S_1$. Therefore, *the matrix associated with the five point scheme given in red-black ordering is consistently ordered.*

Remark 3: Any block tridiagonal matrix is consistently ordered, so it also satisfies property $\mathcal{A}$.

Remark 4: Another basic approach to obtaining general theorems concerning the SOR scheme involves definitions of when matrices are 2-cyclic and p-cyclic; see [70], page 97. There are relationships between these approaches. One such result is that if $\det(A_{jj}) \neq 0$, $j = 1, \ldots, L_1$, then a matrix satisfies property $\mathcal{A}$ if and only if the matrix is 2-cyclic. We will not develop these topics in this text.

The major result that we obtain concerning ordered matrices and the SOR iteration scheme is the following.

Theorem 10.5.17 *Suppose* (1) *the matrix A is consistently ordered and* (2) *$I - R_J$ is similar to a symmetric positive definite matrix. Then*
(a) *the SOR scheme for solving equation $A\mathbf{x} = \mathbf{b}$ converges for $0 < \omega < 2$ (i.e., $\sigma(R_{SOR}) < 1$),*

(b) *the eigenvalues for the iteration matrix for the SOR scheme,* R_{SOR}, *are given in terms of the eigenvalues of* R_J *by formula (10.5.59), and*
(c) *an optimal value of the parameter* ω *exists, and* ω_b *is given by formula (10.5.66).*

Proof: [22], pages 214–219.

Remark 1: If A is a consistently ordered positive definite matrix, then hypotheses (1) and (2) of Theorem 10.5.17 are satisfied ([76], page 1035).

Remark 2: Since earlier we showed that the matrix associated with difference scheme (10.5.13) (along with Dirichlet boundary conditions) for the unknowns ordered in the j-k order is consistently ordered, if $I - R_J$ is similar to a symmetric positive definite matrix, the results of Theorem 10.5.17 will apply to solving difference scheme (10.5.13). In Propositions 10.5.10 and 10.5.12 of Section 10.5.9, we have already shown that the results of Theorem 10.5.17 are true for the matrix equation associated with difference equation (10.5.13).

Remark 3: Since the matrix associated with difference scheme (10.5.13) given in a red-black ordering is consistently ordered, if $I - R_J$ is similar to a symmetric positive definite matrix, the results of Theorem 10.5.17 will apply to solving difference scheme (10.5.13) by R-B SOR (i.e., Algorithm R-B SOR-10.5.13). However, since R_J is independent of ordering (the Jacobi scheme is independent of ordering), the eigenvalues, optimal parameters, convergence rates, etc. will be the same for both Algorithms SOR-10.5.13 and R-B SOR-10.5.13. Hence, *all of the results concerning convergence rates of Gauss-Seidel and SOR, and optimal parameters for SOR schemes that were given in Section 10.5.9 for the five point scheme given in lexicographical ordering, are also true for the five point scheme using red-black ordering.*

In the Remarks made above, we noted that "if $I - R_J$ is similar to a symmetric positive definite matrix" we get certain results. It is not always easy to check this hypothesis (or other hypotheses that can be given for analogous theorems). Often, when we are forced to solve an elliptic equation that is not a Poisson equation, we must proceed without knowing whether the hypotheses are satisfied. There have been many successful computations performed using this approach. There are theorems that provide convergence results for all three schemes, Jacobi, Gauss-Seidel and SOR, but these schemes will converge (and perform well) for problems that do not satisfy the hypotheses and for problems where it is too hard or impossible to check the hypotheses.

Before we leave this section, we should realize that we have not taken advantage of the fact that the algorithms given in this section were given in block form. When we consider the block relaxation schemes, we must solve an $m_j \times m_j$ matrix equation for $j = 1, \ldots, L_1$ for each iteration. To justify this added computing time, there has to be some gain. Generally,

we will see that the block iterative methods are used when the analogous scalar scheme will not give adequate results. Some of the common block iterative schemes will be given in the next section.

10.5.11 Line Jacobi, Gauss-Seidel and SOR Schemes

10.5.11.1 Line Jacobi Scheme

We return to Section 10.5.2, our introduction to the Jacobi iteration scheme. Specifically, we want to consider difference equation (10.5.13) and Algorithm Jacobi-10.5.13. One interpretation of Algorithm Jacobi-10.5.13 is that we left the u_{jk} term on the left hand side of the equality, moved everything else to the right hand side of the equality, and then lagged the terms on the right hand side (made an iterative scheme by using old values on the right hand side of the equation to determine a new value on the left hand side of the equation). If for some reason we think that the equation is strongly tied together in the x direction, we could instead move only the u_{jk+1} and u_{jk-1} to the right hand side of the equation to get the following algorithm, which is referred to as **line Jacobi**.

Line Jacobi-10.5.13

For $k = 1, \ldots, M_y - 1$

Solve

$$\beta^1_{jk} u'_{j+1\,k} - \beta^0_{jk} u'_{jk} + \beta^2_{jk} u'_{j-1\,k} = F_{jk} - \beta^3_{jk} u_{jk+1} - \beta^4_{jk} u_{jk-1}$$
$$j = 1, \ldots, M_x - 1$$

Next j

Next k

It should be clear that the terms on the left hand side of the equality are all terms on the kth line. Hence, as long as the data are ordered in the j-k ordering, the j loop above can be solved using a tridiagonal solver. If we order the unknowns in the j-k ordering, the matrix equation associated with difference scheme (10.5.13) can be written as

$$A\mathbf{u} = \mathbf{f}, \tag{10.5.91}$$

where

$$A = \begin{pmatrix} B_1 & C_1 & \Theta & \cdots & \\ A_2 & B_2 & C_2 & \Theta & \cdots \\ & \ddots & \ddots & \ddots & \ddots \\ \cdots & \Theta & A_{M_y-2} & B_{M_y-2} & C_{M_y-2} \\ & \cdots & \Theta & A_{M_y-1} & B_{M_y-1} \end{pmatrix}; \tag{10.5.92}$$

for $k = 2, \ldots, M_y - 1$,

$$A_k = \begin{pmatrix} \beta^4_{1\,k} & 0 & \cdots & \\ 0 & \beta^4_{2\,k} & 0 & \cdots \\ & \ddots & \ddots & \ddots \\ & \cdots & 0 & \beta^4_{M_x-1\,k} \end{pmatrix}; \tag{10.5.93}$$

for $k = 1, \ldots, M_y - 1$,

$$B_k = \begin{pmatrix} -\beta^0_{1\,k} & \beta^1_{1\,k} & 0 & \cdots & \\ \beta^2_{2\,k} & -\beta^0_{2\,k} & \beta^1_{2\,k} & 0 & \cdots \\ & \ddots & \ddots & \ddots & \ddots \\ \cdots & 0 & \beta^2_{M_x-2\,k} & -\beta^0_{M_x-2\,k} & \beta^1_{M_x-2\,k} \\ & \cdots & 0 & \beta^2_{M_x-1\,k} & -\beta^0_{M_x-1\,k} \end{pmatrix}; \tag{10.5.94}$$

and for $k = 1, \ldots, M_y - 2$,

$$C_k = \begin{pmatrix} \beta^3_{1\,k} & 0 & \cdots & \\ 0 & \beta^3_{2\,k} & 0 & \cdots \\ & \ddots & \ddots & \ddots \\ & \cdots & 0 & \beta^3_{M_x-1\,k} \end{pmatrix}; \tag{10.5.95}$$

$$\mathbf{f} = \mathbf{F} + \mathbf{b}_x + \mathbf{b}_y; \tag{10.5.96}$$

$$\mathbf{F} = [F_{1\,1} \cdots F_{M_x-1\,1}\, F_{1\,2} \cdots F_{M_x-1\,M_y-1}]^T; \tag{10.5.97}$$

$$\begin{aligned} \mathbf{b}_x = [&-\beta^2_{1\,1} u_{0\,1}\, 0 \cdots 0 - \beta^1_{M_x-1\,1} u_{M_x\,1} \\ &- \beta^2_{1\,2} u_{0\,2} \cdots - \beta^1_{M_x-1\,M_y-1} u_{M_x\,M_y-1}]^T; \end{aligned} \tag{10.5.98}$$

and

$$\begin{aligned} \mathbf{b}_y = [&-\beta^4_{1\,1} u_{1\,0} \cdots - \beta^4_{M_x-1\,1} u_{M_x-1\,0}\, 0 \cdots 0 \\ &- \beta^3_{1\,M_y-1} u_{1\,M_y} \cdots - \beta^3_{M_x-1\,M_y-1} u_{M_x-1\,M_y}]^T. \end{aligned} \tag{10.5.99}$$

We should note that the vector $\mathbf{f}$ contains the contributions due to F (in $\mathbf{F}$) and the contributions due to the boundary conditions on both the x and y boundaries (in $\mathbf{b}_x$ and $\mathbf{b}_y$). It should be clear that the matrix A above is the same as $L + D + U$, where L, D, and U were given earlier in (10.5.14)–(10.5.16). We next partition $\mathbf{u}$, $\mathbf{F}$, and $\mathbf{b}_x$ as $\mathbf{u} = [\mathbf{u}_1 \cdots \mathbf{u}_{M_y-1}]^T$, $\mathbf{F} = [\mathbf{F}_1 \cdots \mathbf{F}_{M_y-1}]^T$, and $\mathbf{b}_x = [\mathbf{b}_{x_1} \cdots \mathbf{b}_{x_{M_y-1}}]^T$, where $\mathbf{u}_k = [u_{1\,k} \cdots u_{M_x-1\,k}]^T$, $\mathbf{F}_k = [F_{1\,k} \cdots F_{M_x-1\,k}]^T$, and $\mathbf{b}_{x_k} = [-\beta^2_{1\,k} u_{0\,k}\, 0 \cdots 0 - \beta^1_{M_x-1\,k} u_{M_x\,k}]^T$, $k = 1, \ldots, M_y - 1$. If we now consider solving equation (10.5.91) using the block iterative form of the Jacobi scheme, Algorithm Jacobi-10.5.81, we obtain the following version of the line Jacobi scheme.

Line Jacobi-10.5.91

For $k = 1, \ldots, M_y - 1$

Solve

$$B_k \mathbf{u}'_k = \mathbf{F}_k + \mathbf{b}_{x_k} - A_k \mathbf{u}_{k-1} - C_k \mathbf{u}_{k+1}$$

Next k

Note that when $k = 1$, the boundary conditions from the $y = 0$ boundary ($k = 0$) are included via the $A_1 \mathbf{u}_0$ term. Likewise, the $k = M_y$ boundary condition is included in the $k = M_y - 1$ equation via the $C_{M_y-1}\mathbf{u}_{M_y}$ term. Though neither A_1 nor C_{M_y-1} is defined in equation (10.5.91), we define these matrices analogously to the other A_k's and C_k's, consistent with the way the $\mathbf{b}_y$ boundary condition vector was included in the right hand side vector $\mathbf{f}$. It is not difficult to see that the line Jacobi-10.5.13 scheme is the same as the block iterative scheme, line Jacobi-10.5.91.

To consider the convergence of the line Jacobi scheme, we begin by decomposing matrix (10.5.92) into $A = L + D + U$ where L, D, and U are the block lower triangular, block diagonal, and the block upper triangular matrices, defined analogously to matrices (10.5.84)–(10.5.86) in the decomposition of matrix (10.5.82). Since the scheme given in Algorithm line Jacobi-10.5.91 is clearly equivalent to solving

$$D\mathbf{u}' = \mathbf{f} - (L + U)\mathbf{u}$$

where $\mathbf{u}' = [\mathbf{u}'_1 \ \cdots \ \mathbf{u}'_{M_y-1}]^T$ and $\mathbf{u} = [\mathbf{u}_1 \ \cdots \ \mathbf{u}_{M_y-1}]^T$ and

$$\begin{aligned} \mathbf{f} - (L + U)\mathbf{u} &= D\mathbf{u} + \mathbf{f} - (L + D + U)\mathbf{u} \\ &= D\mathbf{u} + \mathbf{r} \end{aligned}$$

where $\mathbf{r} = \mathbf{f} - A\mathbf{u}$, we can write the line Jacobi scheme as a residual correction scheme,

$$\mathbf{w}_{k+1} = \mathbf{w}_k + D^{-1}\mathbf{r}_k.$$

We can attempt to obtain convergence results by noting that $R_{LJ} = -D^{-1}(L + U)$ and applying Propositions 10.5.1 or 10.5.2. As usual, it is impossible to compute the eigenvalues of the matrix R_{LJ} coming from a problem as general as that given by difference scheme (10.5.13) with boundary conditions. We instead consider the model problem that we have considered in the past, (10.2.3)–(10.2.7). The line Jacobi scheme applied to solve the difference equation associated with this Poisson equation can be written as

Line Jacobi-10.2.3

For $k = 1, \ldots, M_y - 1$

Solve

$$\begin{aligned} -\tfrac{1}{\Delta x^2} u'_{j-1\,k} + 2\left(\tfrac{1}{\Delta x^2} + \tfrac{1}{\Delta y^2}\right) u'_{j\,k} - \tfrac{1}{\Delta x^2} u'_{j+1\,k} \\ = F_{j\,k} + \tfrac{1}{\Delta y^2}(u_{j\,k+1} + u_{j\,k-1}), \ j = 1, \ldots, M_x - 1 \end{aligned}$$

Next k

As with Algorithm line Jacobi-10.5.13, the j loop in the above algorithm must be solved as a tridiagonal matrix (using TRID, of course). We now proceed to compute the eigenvalues of R_{LJ} associated with applying Algorithm line Jacobi-10.2.3.

Example 10.5.3 Compute the eigenvalues of $R_{LJ} = -D^{-1}(L+U)$ associated with solving equations (10.2.3)–(10.2.7) by the line Jacobi relaxation scheme.

Solution: We must find λ and $\mathbf{X}$ that satisfies

$$R_{LJ}\mathbf{X} = -D^{-1}(L+U)\mathbf{X} = \lambda\mathbf{X}$$

or

$$-(L+U)\mathbf{X} = \lambda D\mathbf{X}.$$

Writing the above matrix equation in terms of elements, we get

$$\frac{1}{\Delta y^2}\left(X_{j\,k-1} + X_{j\,k+1}\right) = \lambda\left[-\frac{1}{\Delta x^2}X_{j-1\,k} + dX_{j\,k} - \frac{1}{\Delta x^2}X_{j+1\,k}\right], \tag{10.5.100}$$

where $d = 2(1/\Delta x^2 + 1/\Delta y^2)$, $X_{0\,k} = X_{M_x\,k} = X_{j\,0} = X_{j\,M_y} = 0$ for $j = 1, \ldots, M_x - 1$ and $k = 1, \ldots, M_y - 1$. We proceed as we did in Example 10.5.1 and attempt to solve equation (10.5.100) by discrete separation of variables. We let $X_{j\,k} = x_j y_k$ in equation (10.5.100), rearrange, divide by $x_j y_k$, and get

$$\frac{y_{k-1} + y_{k+1}}{y_k \Delta y^2} = -\lambda\frac{x_{j-1} + x_{j+1}}{x_j \Delta x^2} + \lambda d. \tag{10.5.101}$$

We note that the function on the left hand side is a function of k alone, and the function on the right hand side is a function of only j. Since these functions are equal, they both must be constant, say equal to μ. Hence, including the separated boundary conditions, we must solve the following pair of equations

$$y_{k+1} + y_{k-1} = \mu\Delta y^2 y_k, \quad k = 1, \ldots, M_y - 1 \tag{10.5.102}$$

$$y_0 = y_{M_y} = 0 \tag{10.5.103}$$

$$x_{j+1} + x_{j-1} = \frac{(d\lambda - \mu)}{\lambda}\Delta x^2 x_j, \quad j = 1, \ldots, M_x - 1 \tag{10.5.104}$$

$$x_0 = x_{M_x} = 0. \tag{10.5.105}$$

Equation (10.5.102) along with boundary conditions (10.5.103) can be written as the eigenvalue problem

$$\begin{pmatrix} 0 & 1 & 0 & \cdots & \\ 1 & 0 & 1 & 0 & \cdots \\ & \ddots & \ddots & \ddots & \\ \cdots & 0 & 1 & 0 & 1 \\ & \cdots & 0 & 1 & 0 \end{pmatrix}\mathbf{Y} = \mu\Delta y^2\mathbf{Y}. \tag{10.5.106}$$

By equation (2.2.41), Part 1, we see that the eigenvalues of the matrix given in (10.5.106) are

$$\mu_s = \frac{2}{\Delta y^2}\cos\frac{s\pi}{M_y}, \quad s = 1, \ldots, M_y - 1.$$

We then see that equation (10.5.104) along with boundary condition (10.5.105) can be written as the eigenvalue problem

$$\begin{pmatrix} 0 & 1 & 0 & \cdots & \\ 1 & 0 & 1 & 0 & \cdots \\ & \ddots & \ddots & \ddots & \\ \cdots & 0 & 1 & 0 & 1 \\ & \cdots & 0 & 1 & 0 \end{pmatrix}\mathbf{X} = \omega_s\mathbf{X}, \quad s = 1, \cdots, M_y - 1, \tag{10.5.107}$$

where $\omega_s = (d\lambda - \mu_s)\Delta x^2/\lambda$. Again using formula (2.2.41), Part 1, we know that equation (10.5.107) has $M_x - 1$ eigenvalues for each value of s, and they are given by

$$\omega_s^p = 2\cos\frac{p\pi}{M_x}, \quad p = 1, \ldots, M_x - 1. \tag{10.5.108}$$

Then, solving for λ_s^p (where the notation again indicates the fact that λ will depend on both p and s), we get

$$\lambda_s^p = \frac{2\frac{\Delta x^2}{\Delta y^2}\cos\frac{s\pi}{M_y}}{d\Delta x^2 - 2\cos\frac{p\pi}{M_x}}, \quad s = 1, \ldots, M_y - 1, \; p = 1, \ldots, M_x - 1. \tag{10.5.109}$$

Since the maximum occurs when $s = p = 1$,

$$\sigma(R_{LJ}) = \frac{2\frac{\Delta x^2}{\Delta y^2}\cos\frac{\pi}{M_y}}{d\Delta x^2 - 2\cos\frac{\pi}{M_x}}. \tag{10.5.110}$$

Then, since

$$\frac{2\frac{\Delta x^2}{\Delta y^2}\cos\frac{\pi}{M_y}}{d\Delta x^2 - 2\cos\frac{\pi}{M_x}} < \frac{2\frac{\Delta x^2}{\Delta y^2}}{d\Delta x^2 - 2} = 1,$$

the line Jacobi scheme converges when used to solve problem (10.2.3)–(10.2.7).

Remark 1: As we have done so often before, we use the analysis near the end of Section 10.5.1 to note that in order improve our result by one decimal place ($\zeta = 10^{-1}$), we must perform $m = -1/\log_{10}\sigma(R_{LJ})$ iterations. The number of iterations required to improve the result by one decimal place using the line Jacobi scheme for various grid sizes is given in Table 10.5.3.

M_x	M_y	$\sigma(R_{LJ})$	$m = -1/\log_{10}\sigma(R_{LJ})$
10	10	0.906680	24
100	100	0.999014	2,334
1000	1000	0.999990	233,301
10	100	0.999018	2,343
50	100	0.999014	2,334
50	1000	0.999990	233,340
100	50	0.996061	584
1000	50	0.996061	584

TABLE 10.5.3. Values of the spectral radius and number of iterations of the line Jacobi iteration needed to improve a result by one decimal place for various values of M_x and M_y.

Remark 2: Comparing the results given in the first three lines of Table 10.5.3 with those given in Table 10.5.1, we see that by using line Jacobi with $\Delta x = \Delta y$ and $M_x = M_y$, asymptotically we can use one half as many iterations as we would have to use with the point Jacobi scheme. This can be seen by noting that the asymptotic convergent rate for the point Jacobi

scheme is

$$
\begin{aligned}
-\log\sigma(R_J) &= -\log\left[\frac{2}{d}\left(\frac{1}{\Delta x^2}\cos\pi\Delta x + \frac{1}{\Delta y^2}\cos\pi\Delta y\right)\right] \\
&= -\log\left[1 - \frac{2}{d}\pi^2 + \cdots\right] \\
&= \frac{2}{d}\pi^2 + \cdots = \frac{\Delta x^2\Delta y^2}{\Delta x^2 + \Delta y^2}\pi^2 + \cdots,
\end{aligned}
$$

and the asymptotic rate of convergence for the line Jacobi scheme is

$$
\begin{aligned}
-\log\sigma(R_{LJ}) &= -\log\frac{2\frac{\Delta x^2}{\Delta y^2}\cos\pi\Delta y}{d\Delta x^2 - 2\cos\pi\Delta x} \\
&= -\log(1 - \pi^2\Delta y^2 + \cdots) = \pi^2\Delta y^2 + \cdots.
\end{aligned}
$$

When $\Delta x = \Delta y$, the asymptotic rate of convergence for the point Jacobi scheme is approximately $\pi^2\Delta x^2/2$, while the asymptotic rate of convergence for the line Jacobi scheme is approximately $\pi^2\Delta x^2$, i.e., the line Jacobi scheme should take one half as many iterations as the point Jacobi scheme. We note that in the cases shown in Table 10.5.3 with $\Delta x \neq \Delta y$, it seems that line Jacobi requires approximately the same number of iterations as point Jacobi. This is not actually the case. The number of line Jacobi iterations needed to improve the answer one decimal place using k lines will be very different for the two cases $M_x > M_y$ and $M_x < M_y$. For example, when $M_x = 50$ and $M_y = 1000$, we see that both point and line Jacobi require over 233,000 iterations to improve the answer one decimal place (from Tables 10.5.1 and 10.5.3). But when $M_x = 1000$ and $M_y = 50$, point Jacobi requires over 233,000 iterations where line Jacobi requires only 584. Of course, an analogous result holds when we compute with j lines.

Remark 3: Comparing lines 5–8 in Table 10.5.3 we see that there is a big difference whether we have $M_x = 50$ and $M_y = 1000$ or $M_x = 1000$ and $M_y = 50$. That should not be too surprising, because the line Jacobi given in algorithm line Jacobi-10.2.3 uses k lines. The data on these k lines are tied together implicitly. In addition, when data off the k lines are used, old values are used. Though we cannot predict how much faster the line Jacobi should converge for $M_x = 1000$, $M_y = 50$ than for $M_x = 50$, $M_y = 1000$, it should be clear that the scheme is not symmetric with respect to direction when M_x and M_y are different.

Remark 4: Everything done so far has been done with a j-k ordering. The result of this is that the lines in the line Jacobi have all been k = constant lines. Everything could equally have been done for a k-j ordering. The lines would then be j = constant lines. Most often, either something physical or the size of M_x and M_y (as was described in Remark 3) determine whether j lines or k lines should be used.

10.5.11.2 Line Gauss-Seidel and Line SOR Schemes

It should be clear that using either the rationale given at the beginning of Section 10.5.11.1 for formulating the line Jacobi scheme or using the block schemes given in Section 10.5.10 (applied to the line Jacobi scheme in Section 10.5.11.1), we can easily develop both a line Gauss-Seidel scheme and a line SOR scheme. In this section we shall consider the line SOR scheme, getting the line Gauss-Seidel scheme as a special case of line SOR with $\omega = 1$.

As in Section 10.5.11.1, we consider solving matrix equation (10.5.91) with the matrix, submatrices, vectors, and partitions as given in equations (10.5.92)–(10.5.99). One iteration of the block iterative SOR algorithm for solving equation (10.5.91) can then be given as (following algorithm SOR-10.5.81))

Line SOR-10.5.91

For $k = 1, \ldots, M_y - 1$

solve

$$B_k \hat{\mathbf{u}}_k = \mathbf{F}_k + \mathbf{b}_{x_k} - A_k \mathbf{u}'_{k-1} - C_k \mathbf{u}_{k+1}$$

$$\mathbf{u}'_k = \omega \hat{\mathbf{u}}_k + (1 - \omega)\mathbf{u}_k$$

Next k

We again note that the boundary conditions for the $k = 0$ and $k = M_y$ boundaries are contained in the $A_1 \mathbf{u}'_0$ and $C_{M_y - 1}\mathbf{u}_{M_y}$ terms, respectively. It should also be clear that we can write this scheme in the following form, analogous to Algorithm Line Jacobi-10.5.13.

Line SOR-10.5.13

For $k = 1, \ldots, M_y - 1$

Solve

$$\beta^1_{j\,k}\hat{u}_{j+1\,k} - \beta^0_{j\,k}\hat{u}_{j\,k} + \beta^2_{j\,k}\hat{u}_{j-1\,k} = F_{j\,k} - \beta^3_{j\,k}u_{j\,k+1} - \beta^4_{j\,k}u'_{j\,k-1}$$
$$j = 1, \ldots, M_x - 1$$

For $j = 1, \ldots, M_x - 1$

$$u'_{j\,k} = \omega \hat{u}_{j\,k} + (1 - \omega)u_{j\,k}$$

Next j

Next k

For convergence results and results on choosing ω optimally for line SOR, we return to Theorem 10.5.17. Since block tridiagonal matrices are consistently ordered, if $I - R_J$ is similar to a symmetric positive definite matrix, Theorem 10.5.17 applies. In this case we know that the scheme converges for $0 < \omega < 2$ and converges optimally if ω is chosen as

$$\omega_b = \frac{2}{1 + \sqrt{1 - \bar{\lambda}^2_{LJ}}}. \tag{10.5.111}$$

M_x	M_y	ω_b	$\sigma(R_{LSOR,\omega_b})$	$m = -1/\log_{10}\sigma(R_{LSOR,\omega_b})$
10	10	1.406650	0.406650	3
100	100	1.914966	0.914966	26
1000	1000	1.991154	0.991154	260
10	100	1.915131	0.915131	26
50	100	1.914971	0.914971	26
50	1000	1.991154	0.991154	258

TABLE 10.5.4. Values of the optimal parameter, spectral radius for optimal line SOR, and number of iterations of optimal line SOR needed to improve a result by one decimal place for various values of M_x and M_y.

In particular, if we consider using line SOR to solve problem (10.2.3)–(10.2.7) and choose ω_b by equation (10.5.111) where $\bar{\lambda}_{LJ} = \sigma(R_{LJ})$ is given by equation (10.5.110), we have optimal line SOR. The spectral radius of the optimal line SOR scheme is given by

$$\sigma(R_{LSOR}) = \omega_b - 1.$$

Remark 1: If we again return to our model problem (10.2.3)–(10.2.7), we can use the spectral radius found for the line Jacobi scheme in Example 10.5.3 to determine the spectral radius of R_{LSOR} and the asymptotic rate of convergence. In Table 10.5.4 we give the optimal parameter, spectral radius of R_{LSOR} and the number of iterations of optimal line SOR necessary to improve a result by one decimal place for the same values of M_x and M_y given in Table 10.5.2. Comparing these tables, we see that line SOR converges faster than point SOR. We also notice that when we are using line SOR with k lines, and M_x is much smaller than M_y, then SOR and line SOR converge approximately equally fast.

Comparing the results of Table 10.5.4 with those given in Table 10.5.3, we see that, as we might expect, line SOR converges much faster than line Jacobi.

Remark 2: As was the case with the line Jacobi scheme, everything done in this section for k lines can also be done for j lines. For the very same reasons as we gave with respect to line Jacobi, there will be times that line SOR with j lines will be preferred over line SOR with k lines.

Remark 3: We mention that we could next consider three dimensional problems and derive both line and plane Jacobi Gauss-Seidel and SOR. The approach would be the same as that used in this section. For three dimensional line methods, we would consider the matrix partitioned so that the blocks correspond to lines in one of the three directions. For three dimensional plane methods, we would consider the matrix partitioned in such a way that each block along the diagonal was associated with solving the equation on a two dimensional plane. Again, in both of these cases

Theorem 10.5.17 would apply. We should note that if we use plane Jacobi, Gauss-Seidel or SOR, the matrix equation that we must solve on each plane will not be tridiagonal. The matrix will be broadly banded, analogous to that found when we solve a two dimensional problem. The problem of solving these matrix equations must be faced when using plane methods.

Remark 4: Both the line Gauss-Seidel and the line SOR schemes have the same difficulty as did the point Gauss-Seidel and point SOR in that they are strongly tied to the previous line. This difficulty can be alleviated by developing either a red-black line Gauss-Seidel or a red-black line SOR (sometimes known as zebra Gauss-Seidel and zebra SOR). It is hoped that by this time, the reader is very capable of developing these schemes.

Remark 5: We notice that the line Jacobi and line SOR converge faster than the point Jacobi and point SOR, respectively. We should remember that we are paying a price to get this faster convergence. To gain this speed, we are solving $M_y - 1$ tridiagonals for each iteration. When a decision is made whether or not to use line Jacobi, Gauss-Seidel, or SOR, the extra cost of the scheme should be one consideration.

10.5.12 Approximating ω_b: Reality

In discussions of point, red–black, and line SOR, we find that the optimal value of ω, ω_b, is given by formula (10.5.111), which involves the spectral radius of the iteration matrix of the associated Jacobi scheme. In Examples 10.5.1 and 10.5.3 we calculate the spectral radius of the iteration matrix for approximating the solution of a Poisson equation by point and line Jacobi. The truth of the matter is that we cannot analytically calculate optimal ω's for many more matrices. The problem must be a very nice problem to allow us to compute ω_b analytically. Hence, *it is very important to be able to approximate* ω_b. There are many different approaches to approximating ω_b. If a particular equation is to be solved many times (with different right hand sides, boundary conditions, etc.), then it is worthwhile to work hard to approximate ω_b accurately.

One approach can be derived using the analysis given in Section 10.5.1. From equation (10.5.9), we see that the error in the $(k+1)$st iteration can be approximated by

$$\mathbf{e}_{k+1} \approx \lambda_1^{k+1} a_1 \mathbf{x}_1,$$

where λ_1 is the eigenvalue of largest magnitude of the iteration matrix, $\mathbf{e}_{k+1} = \mathbf{u} - \mathbf{w}_{k+1}$ is the error in the $(k+1)$st step of the residual correction scheme, and a_1 and $\mathbf{x}_1$ are as in Section 10.5.1. We note that if we compute

$$\begin{aligned} \mathbf{w}_{k+1} - \mathbf{w}_k &= (\mathbf{u} - \mathbf{w}_k) - (\mathbf{u} - \mathbf{w}_{k+1}) = \mathbf{e}_k - \mathbf{e}_{k+1} \\ &\approx \lambda_1^k a_1 \mathbf{x}_1 - \lambda_1^{k+1} a_1 \mathbf{x}_1 = \lambda_1 a_1 \mathbf{x}_1 (\lambda_1^{k-1} - \lambda_1^k) \end{aligned} \quad (10.5.112)$$

and

$$\mathbf{w}_k - \mathbf{w}_{k-1} \approx a_1 \mathbf{x}_1 (\lambda_1^{k-1} - \lambda_1^k), \tag{10.5.113}$$

then

$$\frac{\|\mathbf{w}_{k+1} - \mathbf{w}_k\|}{\|\mathbf{w}_k - \mathbf{w}_{k-1}\|} \approx |\lambda_1|$$

gives an approximation to the magnitude of the largest eigenvalue of the iteration matrix. If instead of taking the norms above, we evaluate the j-th component of equations (10.5.112) and (10.5.113), we get

$$\frac{w_{j_{k+1}} - w_{j_k}}{w_{j_k} - w_{j_{k-1}}} \approx \lambda_1, \tag{10.5.114}$$

an approximation to the eigenvalue with the largest magnitude. We should realize that part of the above analysis is *the assumption (used in Section 10.5.1) that the iteration matrix has a full set of independent eigenvectors.* When this assumption is not satisfied (as is sometimes the case when we apply it to the SOR scheme below), either the analysis must be used as an "indication of what might be true" or a more careful analysis must be done.

The procedure for computing an approximation to the optimal relaxation parameter is the following.

- Choose an initial guess of ω and use the procedure described above to find the largest eigenvalue of the iteration matrix for the SOR scheme.
- Once we have a reasonable value of $\lambda_{1,\omega}$ (which we perceive as being a reasonable approximation to the largest eigenvalue of $R_{SOR,\omega}$), we use equation (10.5.59) to determine the associated eigenvalue of the Jacobi iteration matrix

$$\lambda_{1,J} = \frac{1 - \omega - \lambda_{1,\omega}}{\omega\sqrt{\lambda_{1,\omega}}}.$$

  Since $\lambda_{1,\omega}$ is an approximation for the largest eigenvalue of $R_{SOR,\omega}$, $\lambda_{1,J}$ *will be an approximation to the largest eigenvalue of* R_J, λ_J, *independent of* ω.

- Use this largest eigenvalue of the Jacobi iteration matrix in formula (10.5.66) to compute the optimal iteration parameter ω_b.

Remark 1: As a part of the above procedure, we approximate the largest eigenvalue of the iteration matrix $R_{SOR,\omega}$. We note that if we choose $\omega \leq \omega_b$, then the largest eigenvalue of $R_{SOR,\omega}$ is real and simple. If we choose $\omega > \omega_b$, this procedure does not usually work. Hence *it is very important that when we choose* ω, *we choose* $\omega \leq \omega_b$.

Remark 2: Also, we do not apply the approximation (10.5.114) for just one value of j. We let $\lambda_{1,\omega}$ denote that average value of the approximations given by approximation (10.5.114) over all j (or over a sampling of the j's). If the approximate eigenvalue $\lambda_{1,\omega}$ does not appear to be converging as a function of k, the ω chosen is probably greater than ω_b.

Remark 3: We must realize that this approach depends on an asymptotic result, i.e., k must generally be large before (10.5.114) gives a good approximation of λ_1. As we see below in HW10.5.22, if the calculation is made too early, we get a bad approximation (maybe better than we presently have, but not good enough to warrant all of the work). In parts (a)(iii) and (a)(iv) of HW10.5.22 we see that the approach can give us a very good approximation of ω_b.

Remark 4: We notice in part (b) of HW10.5.22 that if the solution and initial guess (i.e., the initial error) are sufficiently trivial and contain only one of the eigenvectors, the technique will compute that eigenvalue associated with that eigenvector, and will do it well with only three iterations. However, we should understand that we have not found the largest eigenvalue of the Jacobi iterations matrix and cannot find ω_b. In general, if the eigenvector associated with the largest eigenvalue is not present in the eigenvector expansion of the initial error, the above procedure will not find an approximation to the largest eigenvalue (and hence will not compute ω_b). The procedure will find an approximation to the largest eigenvalue associated with one of the eigenvectors present in the eigenvector expansion of the initial error.

Of course, there are other approaches to finding approximations of ω_b. We have included one approach mainly to give a taste of how it might be done and to emphasize that it must be done. For a more complete discussion, which includes some computer programs, see [22], page 223.

HW 10.5.22 Consider the problem

$$\begin{aligned} \nabla^2 v &= F(x,y), \quad (x,y) \in R = (0,1) \times (0,1) \\ v &= 0, \quad (x,y) \text{ on } \partial R. \end{aligned}$$

(a) For $F(x,y) = e^{x+y}$, determine an approximation of ω_b by
  (i) using $\left(\frac{w_{j3}-w_{j2}}{w_{j2}-w_{j1}}\right)$ for one point j.
  (ii) using the average of $\left(\frac{w_{j3}-w_{j2}}{w_{j2}-w_{j1}}\right)$ over 100 points.
  (iii) using $\left(\frac{w_{j10}-w_{j9}}{w_{j9}-w_{j8}}\right)$ for one point j.
  (iv) using the average $\left(\frac{w_{j10}-w_{j9}}{w_{j9}-w_{j8}}\right)$ over 100 points.

(b) Repeat part (a) using $F(x,y) = \sin \pi x \sin 2\pi y$.

(c) Explain why the computation done in part (b) gives a bad approximation of ω_b.

10.6 Elliptic Difference Equations: Neumann Boundary Conditions

To this point in our consideration of elliptic boundary value problems, we have treated only Dirichlet boundary conditions. There are other boundary conditions we could consider other than Dirichlet and Neumann boundary conditions but these are clearly the two most common types of boundary conditions. We consider mixture boundary conditions and Robin boundary conditions in Section 10.8. In this section we will consider the numerical solution of elliptic partial differential equations with Neumann boundary conditions. For further results, analytic and numerical, see [21].

We consider the model problem

$$-\nabla^2 v = F \quad \text{in } R = (0,1) \times (0,1) \tag{10.6.1}$$

$$\frac{\partial v}{\partial n} = g \quad \text{on } \partial R. \tag{10.6.2}$$

It is easy to see that if v is a solution to problem (10.6.1)–(10.6.2), so is $v + c$ for any constant c. Hence, we know that we have a nonunique solution to problem (10.6.1)–(10.6.2). One might be tempted to say that because the problem does not have a unique solution, it cannot be an important problem. This is not the case. Problem (10.6.1)–(10.6.2) is easily as important as the analogous problem with Dirichlet boundary conditions. We must must be able to solve problems of this form and must be very careful to handle the numerical consequences of the nonuniqueness.

To talk about solutions to problem (10.6.1)–(10.6.2), one must prescribe conditions on F and g that will allow solutions to exist. We prove the following proposition.

Proposition 10.6.1 *If R is a Green's region and problem (10.6.1)–(10.6.2) has a solution v, then*

$$-\int_R F(x,y)\,dx\,dy = \int_{\partial R} g(x,y)\,ds. \tag{10.6.3}$$

For any constant c, $v+c$ will also be a solution to problem (10.6.1)–(10.6.2).

Proof: Before we start the proof, we define a **Green's region** to be a region in the plane sufficiently nice to satisfy the hypotheses for the first Green's formula,

$$\int_R v_1 \nabla^2 v_2\,dx\,dy = -\int_R (\nabla v_1, \nabla v_2)\,dx\,dy + \int_{\partial R} v_1 \frac{\partial v_2}{\partial n}\,ds,$$

where $(\nabla v_1, \nabla v_2)$ denotes the dot product of ∇v_1 and ∇v_2. If we let $v_1 = 1$ and $v_2 = v$, we obtain (10.6.3).

Remark 1: The converse of Proposition 10.6.1 is also true. See [21], page 154.

Remark 2: We shall refer to condition (10.6.3) as the **analytic compatibility condition.** It is also a conservation condition. The term on the left represents the amount of the material injected into or pumped out of the region, and the term on the right represents the amount of the material that flows into or out of the boundary of the region.

We wish to approximate the solution to problem (10.6.1)–(10.6.2) numerically. We will proceed much in the same way that we did for the analogous Dirichlet problem. We want an analogue to Theorem 10.3.3, and then we want to study the solution to the discrete problem. For convenience, we consider a grid on the region $R = [0,1] \times [0,1]$, $(x_j, y_k) = (j\Delta x, k\Delta y)$, $j = 0, \ldots, M$, $k = 0, \ldots, M$, where $\Delta y = \Delta x$. We difference the partial differential equation as we have before, and we approximate equation (10.6.1) by

$$-\frac{1}{\Delta x^2}\left(\delta_x^2 + \delta_y^2\right) u_{j\,k} = F_{j\,k} \quad j,k = 1, \ldots, M-1. \tag{10.6.4}$$

It is not as clear how we should approximate the boundary conditions (10.6.2). As was the case when we considered Neumann boundary conditions for parabolic equations, Sections 1.4 and 4.4.3.2, at least two of the obvious choices are to use the first or the second order approximations of the normal derivative. We treat each of these cases in the following two sections.

10.6.1 First Order Approximation

The treatment of the boundary conditions for Dirichlet boundary conditions was very routine and easy. It should not surprise us that the approximation of the boundary conditions might be more difficult and important when we consider Neumann boundary conditions. Using a first order approximation of the derivative in boundary condition (10.6.2) leaves us with the following discrete boundary conditions.

$$-\frac{u_{1\,k} - u_{0\,k}}{\Delta x} = g_{0\,k}, \quad k = 0, \ldots, M \tag{10.6.5}$$

$$\frac{u_{M\,k} - u_{M-1\,k}}{\Delta x} = g_{M\,k}, \quad k = 0, \ldots, M \tag{10.6.6}$$

$$-\frac{u_{j\,1} - u_{j\,0}}{\Delta x} = g_{j\,0}, \quad j = 0, \ldots, M \tag{10.6.7}$$

$$\frac{u_{j\,M} - u_{j\,M-1}}{\Delta x} = g_{j\,M}, \quad j = 0, \ldots, M \tag{10.6.8}$$

We should note that the negative signs in formulas (10.6.5) and (10.6.7) are due to the fact that the normal vector associated with the Neumann boundary condition is assumed to be an outward normal. Thus we must consider solving equations (10.6.4)–(10.6.8). This system of equations can

be written as

$$A\mathbf{u} = \mathbf{f} \tag{10.6.9}$$

where A is the $(M-1)\times(M-1)$ block matrix

$$A = \frac{1}{\Delta x^2}\begin{pmatrix} T_1 & -I & \Theta & \cdots & \\ -I & T & -I & \Theta & \cdots \\ & \ddots & \ddots & \ddots & \\ \cdots & \Theta & -I & T & -I \\ & \cdots & \Theta & -I & T_1 \end{pmatrix}, \tag{10.6.10}$$

T_1 and T are the $(M-1)\times(M-1)$ matrices

$$T_1 = \begin{pmatrix} 2 & -1 & 0 & \cdots & \\ -1 & 3 & -1 & 0 & \cdots \\ & \ddots & \ddots & \ddots & \ddots \\ & \cdots & 0 & -1 & 3 & -1 \\ & & \cdots & 0 & -1 & 2 \end{pmatrix}, \tag{10.6.11}$$

$$T = \begin{pmatrix} 3 & -1 & 0 & \cdots & & \\ -1 & 4 & -1 & 0 & \cdots & \\ & \ddots & \ddots & \ddots & \ddots & \cdots \\ & \cdots & 0 & -1 & 4 & -1 \\ & & \cdots & 0 & -1 & 3 \end{pmatrix}, \tag{10.6.12}$$

I is the $(M-1)\times(M-1)$ identity matrix, Θ is the $(M-1)\times(M-1)$ zero matrix, and $\mathbf{f}$ is given by

$$\mathbf{f} = \mathbf{F} + \mathbf{b}_x + \mathbf{b}_y \tag{10.6.13}$$

where $\mathbf{F}$, $\mathbf{b}_x$, and $\mathbf{b}_y$ are the $L=(M-1)^2$–vectors

$$\mathbf{F} = [F_{1\,1} \cdots F_{M-1\,1}\, F_{1\,2} \cdots F_{M-1\,M-1}]^T, \tag{10.6.14}$$

$$\mathbf{b}_x = \frac{1}{\Delta x}[g_{0\,1}\, 0 \cdots 0\, g_{M\,1}\, g_{0\,2}\, 0 \cdots g_{M\,M-1}]^T \tag{10.6.15}$$

and

$$\mathbf{b}_y = \frac{1}{\Delta x}[g_{1\,0} \cdots g_{M-1\,0}\, 0 \cdots 0\, g_{1\,M} \cdots g_{M-1\,M}]^T. \tag{10.6.16}$$

See Figure 10.6.1 for the full matrix written out for the case of $M = 5$.

We include most of the important properties of matrix A and equation (10.6.9) in the following proposition.

$$\frac{1}{\Delta x^2}\left(\begin{array}{cccc|cccc|cccc|cccc}
2 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 3 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 3 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 2 & \mathbf{0} & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
-1 & 0 & 0 & \mathbf{0} & 3 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & -1 & 3 & \mathbf{0} & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & -1 & 0 & 0 & \mathbf{0} & 3 & -1 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 3 & \mathbf{0} & 0 & 0 & -1 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & \mathbf{0} & 2 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 3 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 3 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 2
\end{array}\right)$$

FIGURE 10.6.1. Matrix A, as in equation (10.6.9), represents the matrix associated with solving difference equation (10.6.4) along with the first order approximation of the Neumann boundary conditions given by (10.6.5)–(10.6.8) with $M = 5$. A boldface zero represents a zero that is due to a boundary condition term.

Proposition 10.6.2 (1) $A\mathbf{1} = \boldsymbol{\theta}$.
(2) $\mathbf{1}$ *is the only vector in the null space of* A.
(3) *If* $\mathbf{u}^1$ *and* $\mathbf{u}^2$ *are any two solutions to equation (10.6.9), then there exists a constant* c *such that* $\mathbf{u}^1 = \mathbf{u}^2 + c\mathbf{1}$, *where* $\mathbf{1} = [1 \cdots 1]^T$.
(4) *Equation (10.6.9) has a solution if and only if*

$$-\Delta x^2 \sum_{k=1}^{M-1}\sum_{j=1}^{M-1} F_{jk} = \Delta x \sum_{j=1}^{M-1} [g_{j0} + g_{jM}] + \Delta x \sum_{k=1}^{M-1} [g_{Mk} + g_{0k}]. \tag{10.6.17}$$

Proof: To see that statement (1) is true, one need only compute $A\mathbf{1}$. The **null space** of a matrix A is defined to be $\{\mathbf{X} : A\mathbf{X} = \boldsymbol{\theta}\}$ and is denoted by $N(A)$. Statement (1) can be expressed as $\mathbf{1} \in N(A)$.

(2) If we eliminate the first row and column of the matrix A, we can apply Proposition 10.2.5 to the resulting submatrix to show that this submatrix is of full rank (the submatrix is invertible). As usual, it is easy to see that the submatrix is diagonally dominant. Likewise, it is easy to see that the first row of the submatrix satisfies the strict inequality necessary for Proposition 10.2.5. Technically, the irreducibility of the submatrix follows from the fact that any two grid points can be connected by a chain of neighboring points. It is also not difficult to use the characterization of irreducibility given immediately preceding Proposition 10.2.5 (if a change in any of the components of the right hand side will cause a change in the solution) to see that the submatrix is irreducible.

Since the submatrix formed by eliminating one row and one column is of full rank, $L - 1$ where $L = (M-1)^2$, then the rank of A is at least $L - 1$. Since we know that $\mathbf{1} \in N(A)$, then the rank of A is $L - 1$, and $\mathbf{1}$ is the only vector in $N(A)$.

(3) When we know that the rank of A is $L - 1$ and $\mathbf{1} \in N(A)$, we know that we can solve equation (10.6.9) for $L - 1$ of the variables in terms of one of the variables, i.e., all solutions of equation (10.6.9) are of the form

$$\mathbf{u} = \mathbf{u}_0 + c\mathbf{1}$$

where $\mathbf{u}_0$ is a fixed vector. Statement (3) then follows easily.

(4) Since $\mathbf{1} \in N(A)$, we know that equation (10.6.9) is solvable if and only if $(\mathbf{f}, \mathbf{1}) = 0$, [31], page 17. If we return to equations (10.6.13)–(10.6.16), we see that

$$\begin{aligned} 0 &= (\mathbf{f}, \mathbf{1}) \\ &= (\mathbf{F}, \mathbf{1}) + (\mathbf{b}_x, \mathbf{1}) + (\mathbf{b}_y, \mathbf{1}) \\ &= \sum_{j=1}^{M-1}\sum_{k=1}^{M-1} F_{jk} + \frac{1}{\Delta x}\sum_{k=1}^{M-1} [g_{0k} + g_{Mk}] + \frac{1}{\Delta x}\sum_{j=1}^{M-1} [g_{j0} + g_{jM}]. \end{aligned}$$

It is easy to see that this is the same as equation (10.6.17).

The results given in Proposition 10.6.2 show that equation (10.6.9) is not a nice equation. Considering the analytic analogue, equations (10.6.1)–(10.6.2), this should not surprise us. In fact, it is clearly good that $\mathbf{1} \in N(A)$ and that all solutions are of the form $\mathbf{u} = \mathbf{u}_0 + c\mathbf{1}$. Both of these facts show that the numerical problem (10.6.9) mimics the analytic problem (10.6.1)–(10.6.2) well. In addition, since the analytic problem had an analytic compatibility condition that must be satisfied for solvability, it is also logical that we have a **discrete compatibility condition**, (10.6.17), that must be satisfied for equation (10.6.9) to be solvable.

Remark 1: Just as the analytic compatibility condition could be described as a conservation property, the discrete compatibility condition (10.6.17) can also be described as a conservation law for difference equations (10.6.4)–(10.6.8).

Remark 2: One problem we face is the fact that if we assume that F and g are nice enough to satisfy the analytic compatibility condition (10.6.3), this is not enough to imply that F and g will satisfy the discrete compatibility condition (10.6.17). If the integrals involved in the analytic compatibility condition are approximated by the appropriate numerical integration scheme, we see that

$$\begin{aligned}
0 &= \int_R F(x,y)\,dx\,dy + \int_{\partial R} g(x,y)\,ds \\
&= \int_R F(x,y)\,dx\,dy + \int_0^1 g(x,0)\,dx + \int_0^1 g(1,y)\,dy \\
&\quad + \int_1^0 g(x,1)\,dx + \int_1^0 g(0,y)\,dy \\
&\approx \sum_{k=1}^{M-1}\sum_{j=1}^{M-1} F_{j\,k}\Delta x\Delta x + \sum_{j=1}^{M-1} g_{j\,0}\Delta x + \sum_{k=1}^{M-1} g_{M\,k}\Delta x \\
&\quad + \sum_{j=1}^{M-1} g_{j\,M}\Delta x + \sum_{k=1}^{M-1} g_{0\,k}\Delta x + \mathcal{O}(\Delta x).
\end{aligned}$$

We note that the numerical approximation is not well done. If we consider rectangular regions centered at the grid points, the area integral omits a strip ($\mathcal{O}(\Delta x)$) around the region. The line integrals skip little chunks in each corner.

More importantly, we see that if F and g satisfy the analytic compatibility condition, F and g will generally only approximately satisfy the discrete compatibility condition. Because of this fact, *part (4) of Proposition 10.6.2 is generally not satisfied.* In addition, if we review our solution methods, we do not have any methods that are designed for solving singular equations. To explain what happens when the discrete compatibility condition is only approximately satisfied and obtain a system of equations that we can solve,

we consider a slight variation of system (10.6.9),

$$\bar{A}\bar{\mathbf{u}} = \bar{\mathbf{f}} \tag{10.6.18}$$

where

$$\bar{A} = \begin{pmatrix} A & \mathbf{1} \\ \mathbf{1}^T & 0 \end{pmatrix}, \quad \bar{\mathbf{u}} = \begin{bmatrix} \mathbf{u} \\ \lambda \end{bmatrix}, \text{ and } \bar{\mathbf{f}} = \begin{bmatrix} \mathbf{f} \\ 0 \end{bmatrix}.$$

We are then able to prove the following proposition related to equation (10.6.18).

Proposition 10.6.3 (1) *System (10.6.18) is solvable.*
(2) *If the solution to system (10.6.18) is of the form* $\bar{\mathbf{u}}_0 = [\mathbf{u}_0 \quad 0]^T$, *then the discrete compatibility condition (10.6.17) is satisfied and* $\mathbf{u}_0$ *is a solution to system (10.6.9) such that* $(\mathbf{u}_0, \mathbf{1}) = 0$.
(3) *If the solution to system (10.6.18) is of the form* $\bar{\mathbf{u}}_0 = [\mathbf{u}_0 \quad \lambda]^T$, *where* $\lambda \neq 0$, *then* $\mathbf{u}_0$ *is a solution to the equation*

$$A\mathbf{u} = \mathbf{f} - \lambda\mathbf{1} \tag{10.6.19}$$

such that $(\mathbf{u}_0, \mathbf{1}) = 0$.

Proof: (1) From [31], page 17, we see that the range of the matrix A is orthogonal to $N(A^T)$. Since A is symmetric, $N(A^T) = N(A)$. The fact that the range of A is the span of the columns of A implies that $\mathbf{1}$ is independent of the columns of A. Then, since the rank of A is $L-1$, the rank of $[A \quad \mathbf{1}]$ is L. Since $[\mathbf{1}^T \quad 0]$ is independent of the rows of $[A \quad \mathbf{1}]$ (for essentially the same reason), the rank of $\bar{A}$ is $L+1$, i.e., $\bar{A}$ is of full rank and, hence, solvable for any right hand side.

(2) If the solution of equation (10.6.18) is in the form $\bar{\mathbf{u}} = [\mathbf{u}_0 \quad 0]^T$, then $\mathbf{u}_0$ satisfies $A\mathbf{u} = \mathbf{f}$ and $(\mathbf{u}_0, \mathbf{1}) = 0$ because of the form of equation (10.6.18). The discrete compatibility condition is satisfied because, as in Proposition 10.6.2, $\mathbf{1} \in N(A)$ implies that $(\mathbf{f}, \mathbf{1}) = 0$ (since equation (10.6.9) is solvable).

The proof of part (3) is a consequence of the hypotheses and the form of $\bar{A}$, $\bar{\mathbf{u}}$ and $\bar{\mathbf{f}}$.

Remark 1: In both of the parts (2) and (3) of Proposition 10.6.3 above, we have constrained our solutions to satisfy $(\mathbf{u}_0, \mathbf{1}) = 0$. The solutions to equation (10.6.9) (part (2) of the proposition) and (10.6.19) are of the form $\mathbf{u}_0 + c\mathbf{1}$ for some constant vector $\mathbf{u}_0$. Requiring that the solution be orthogonal to $\mathbf{1}$ forces c to be zero.

Remark 2: Part (3) of Proposition 10.6.3 first appears to be bad in that we cannot solve the problem that we want to solve. However, instead of being as bad as it first appears, the situation is quite nice. Because the discrete compatibility condition is satisfied only approximately, we know that equation (10.6.9) does not generally have a solution. Part (3) takes

care of the problem and gives the best solution possible—given the approximation of the boundary conditions. The solution to equation (10.6.19) is a solution to equation (10.6.9) where the array F_{jk} (the function F) has been replaced by

$$\tilde{F}_{jk} = F_{jk} - \lambda \mathbf{1}. \tag{10.6.20}$$

In other words, system (10.6.18) is smart enough to realize that we have the wrong F_{jk} (part of it could be due to g) and fixes it for us. As we shall see in the next theorem, λ is small.

Now that we understand what we are solving when we solve the discrete problem associated with problem (10.6.1)–(10.6.2), we state the following analogue of Theorem 10.3.3.

Theorem 10.6.4 *Let $v \in C^4(\overline{R})$ be a solution to Neumann problem (10.6.1)–(10.6.2). Let $\bar{\mathbf{u}} = [\mathbf{u}_0 \quad \lambda]^T$ be a solution to equation (10.6.18). Then*

$$|\lambda| = \mathcal{O}(\Delta x)$$
$$\|\mathbf{u}_0 - \mathbf{v}\|_\infty = \mathcal{O}(\Delta x).$$

Proof: See [21], page 69.

Remark 1: As in Theorem 10.3.3, the hypothesis that $v \in C^4$ implies that v has continuous derivatives up through order four. In addition, as in Theorem 10.3.3, the constants involved in the $\mathcal{O}$ notation depend on the derivatives of v. There is dependence on the fourth derivations due to the fact that difference equation (10.6.4) is a second order approximation to the partial differential equation (10.6.1). There is also dependence on the second derivatives of v because of the first order approximation of the boundary conditions.

Remark 2: From the proof of Theorem 10.6.4, it can be seen that the fact that the convergence above is only of first order in Δx and Δy is due to the treatment of the boundary conditions.

Remark 3: We might recall that in Example 2.3.4 when we used the one dimensional version of the first order approximations for the Neumann boundary condition used here, we found that the difference scheme was not norm consistent (the truncation error was not of first order, as one might expect). (We did find in Example 8.6.2 that we could prove that the scheme was convergent order Δx by the Osher result.) Yet here we find that we obtain first order convergence using these approximations. The difference is due to the time dependence involved in the norm consistency definition, Definition 2.3.2. Since we do not have to define consistency here that is compatible with the Lax Theorem, we have no problem with these boundary conditions.

HW 10.6.1 Use conservation methods similar to those used in Sections 1.6 and 1.6.1 to derive difference equations (10.6.4)–(10.6.8).

10.6.2 Second Order Approximation

Now that we have seen that we obtain first order convergence using the first order approximation of the boundary conditions (while the difference equation was a second order approximation to the partial differential equation), we might hope that if we use a second order approximation to the boundary conditions, we will obtain second order convergence. To show that this is the case, we essentially have to repeat everything done in the last section. We will include as much of the material that we feel is necessary and/or helpful, leaving the rest to your imagination or your own reading.

To use a second order approximation for the boundary conditions, as we did for parabolic equations in Chapter 4, we consider the difference equation at both the interior and boundary grid points, i.e., we consider

$$-\frac{1}{\Delta x^2}\left(\delta_x^2+\delta_y^2\right)u_{j\,k}=F_{j\,k},\quad j,k=0,\ldots,M. \tag{10.6.21}$$

Of course, the problem with difference equation (10.6.21) is that it reaches to fictitious points outside of the domain. These points are eliminated by the use of the following second order approximation of the Neumann boundary condition.

$$-\frac{u_{1\,k}-u_{-1\,k}}{2\Delta x}=g_{0\,k},\quad k=0,\ldots,M \tag{10.6.22}$$

$$\frac{u_{M+1\,k}-u_{M-1\,k}}{2\Delta x}=g_{M\,k},\quad k=0,\ldots,M \tag{10.6.23}$$

$$-\frac{u_{j\,1}-u_{j\,-1}}{2\Delta x}=g_{j\,0},\quad j=0,\ldots,M \tag{10.6.24}$$

$$\frac{u_{j\,M+1}-u_{j\,M-1}}{2\Delta x}=g_{j\,M},\quad j=0,\ldots,M. \tag{10.6.25}$$

Difference equation (10.6.21) along with boundary conditions (10.6.22)–(10.6.25) can be written as

$$A\mathbf{u}=\mathbf{f}, \tag{10.6.26}$$

where A is the $(M+1)\times(M+1)$ block tridiagonal matrix

$$A=\frac{1}{\Delta x^2}\begin{pmatrix} T & -2I & \Theta & \cdots & \\ -I & T & -I & \Theta & \cdots \\ & \ddots & \ddots & \ddots & \\ \cdots & \Theta & -I & T & -I \\ & \cdots & \Theta & -2I & T \end{pmatrix}, \tag{10.6.27}$$

T is the $(M+1)\times(M+1)$ matrix

$$T=\begin{pmatrix} 4 & -2 & 0 & \cdots & & \\ -1 & 4 & -1 & 0 & \cdots & \\ & \ddots & \ddots & \ddots & \ddots & \\ & \cdots & 0 & -1 & 4 & -1 \\ & & \cdots & 0 & -2 & 4 \end{pmatrix}, \tag{10.6.28}$$

I is the $(M+1)\times(M+1)$ identity matrix, Θ is the $(M+1)\times(M+1)$ zero matrix, and $\mathbf{f}$ is given by

$$\mathbf{f}=\mathbf{F}+\mathbf{b}_x+\mathbf{b}_y, \tag{10.6.29}$$

where $\mathbf{F}$, $\mathbf{b}_x$ and $\mathbf{b}_y$ are the $L=(M+1)^2$-vectors

$$\mathbf{F}=[F_{0\,0}\ \cdots\ F_{M\,0}\ F_{0\,1}\ \cdots\ F_{M\,M}]^T, \tag{10.6.30}$$

$$\mathbf{b}_x=\frac{2}{\Delta x}[g_{0\,0}\ 0\ \cdots\ 0\ g_{M\,0}\ g_{0\,1}\ 0\ \cdots\ g_{M\,M}]^T \tag{10.6.31}$$

and

$$\mathbf{b}_y=\frac{2}{\Delta x}[g_{0\,0}\ \cdots\ g_{M\,0}\ 0\ \cdots\ 0\ g_{0\,M}\ \cdots\ g_{M\,M}]^T. \tag{10.6.32}$$

The full matrix A is given in Figure 10.6.2 for $M=4$. Compare the form of A with that given for the scheme using the first order approximation of the Neumann boundary conditions given in Figure 10.6.1 (noting that for Figure 10.6.1 we used $M=5$, whereas here we used $M=4$ so that we could fit it on a page).

Before we get too serious about equation (10.6.26), we notice that as in the previous section, $A\mathbf{1}=\boldsymbol{\theta}$ ($\mathbf{1}\in N(A)$) and $\mathbf{1}$ is the only vector in the null space of A. One of the differences from the case of the first order approximation is that now A is not symmetric. Generally, nonsymmetry tends to cause problems.

We next would like to find the discrete compatibility condition associated with equation (10.6.26). When A is nonsymmetric, we do not get the same result as we did for the case of the first order approximation. (Equation (10.6.9) is solvable if and only if $(\mathbf{f},\mathbf{1})=0$.) In that result, we were using the fact that the matrix A was symmetric. However, the result given in [31], page 17, is that *equation (10.6.26) is solvable if and only if* $(\mathbf{f},\mathbf{u}^*)=0$ *for all* $\mathbf{u}^*\in N(A^T)$. Using this result, we can proceed as we did in the case of the first order approximation and obtain the following proposition.

Proposition 10.6.5 (1) $A\mathbf{1}=\boldsymbol{\theta}$ *and* $N(A)=\{\mathbf{1}\}$.
(2) $A^T\mathbf{u}^*=\boldsymbol{\theta}$ *where* $\mathbf{u}^*$ *is the* $L=(M-1)^2$*-vector*

$$\mathbf{u}^*=[\mathbf{u}^1\ \mathbf{u}^2\ \cdots\ \mathbf{u}^2\ \mathbf{u}^1]^T,$$

$$\frac{1}{\Delta x^2}\left(\begin{array}{ccccc|ccccc|ccccc|ccccc|ccccc}
4 & -2 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 4 & -1 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 4 & -1 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 4 & -1 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -2 & 4 & 0 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
-1 & 0 & 0 & 0 & 0 & 4 & -2 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -2 & 4 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 4 & -2 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -2 & 4 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 4 & -2 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 4 & 0 & 0 & 0 & 0 & -1 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 4 & -2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & -1 & 4 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & -1 & 4 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & -1 & 4 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & -2 & 4
\end{array}\right)$$

FIGURE 10.6.2. Matrix A, as in equation (10.6.26), represents the matrix associated with solving difference equation (10.6.4) along with the second order approximation of the Neumann boundary conditions given by (10.6.22)–(10.6.25) with $M = 4$. A boldface zero represents a zero that is due to a boundary condition term.

made up of the $(M-1)$*-vectors*

$$\mathbf{u}^1 = \left[\frac{1}{4}\ \frac{1}{2}\ \cdots\ \frac{1}{2}\ \frac{1}{4}\right]^T,$$

and

$$\mathbf{u}^2 = \left[\frac{1}{2}\ 1\ \cdots\ 1\ \frac{1}{2}\right]^T,$$

and $N(A^T) = \{\mathbf{u}^*\}$.
(3) *If* $\mathbf{u}^\diamondsuit$ *and* $\mathbf{u}^\heartsuit$ *are any two solutions to equation (10.6.26), then there exists a constant* c *such that* $\mathbf{u}^\diamondsuit = \mathbf{u}^\heartsuit + c\mathbf{1}$.
(4) *Equation (10.6.26) has a solution if and only if*

$$-\Delta x^2 \sum_{k=0}^{M}\sum_{j=0}^{M} s_j s_k F_{j\,k} = \Delta x \sum_{j=0}^{M} \frac{1}{2} s_j \left[g_{j\,0} + g_{j\,M}\right] + \Delta x \sum_{k=0}^{M} \frac{1}{2} s_k \left[g_{M\,k} + g_{0\,k}\right], \tag{10.6.33}$$

where $s_0 = \frac{1}{2}$, $s_M = \frac{1}{2}$ *and* $s_j = 1$ *for* $j = 1, \ldots, M-1$.

Proof: The proofs of (1), (2) and (3) follow the same approach used for parts (1), (2) and (3) of Proposition 10.6.2.

The proof of part (4) follows from the fact mentioned earlier that equation (10.6.26) is solvable if and only if $(\mathbf{f}, \mathbf{u}^*) = 0$ for all $\mathbf{u}^* \in N(A^T)$ and the fact that $N(A^T) = \{\mathbf{u}^*\}$, given in part (2) in this proposition. Equation (10.6.33) is the same as $(\mathbf{f}, \mathbf{u}^*) = 0$, where $\mathbf{u}^*$ is as defined in part (2).

We can then proceed as we did in the last section and consider the variation of equation (10.6.26),

$$\bar{A}\bar{\mathbf{u}} = \bar{\mathbf{f}}, \tag{10.6.34}$$

where

$$\bar{A} = \begin{pmatrix} A & \mathbf{u}^* \\ 1^T & 0 \end{pmatrix}, \quad \bar{\mathbf{u}} = \begin{bmatrix} \mathbf{u} \\ \lambda \end{bmatrix}, \text{ and } \bar{\mathbf{f}} = \begin{bmatrix} \mathbf{f} \\ 0 \end{bmatrix}.$$

We are then able to prove the following proposition.

Proposition 10.6.6 (1) *System (10.6.34) is solvable.*
(2) *If the solution to system (10.6.34) is of the form* $\bar{\mathbf{u}}_0 = [\mathbf{u}_0 \quad 0]^T$, *then the discrete compatibility condition (10.6.33) is satisfied, and* $\mathbf{u}_0$ *is a solution to system (10.6.26) such that* $(\mathbf{u}_0, \mathbf{1}) = 0$.
(3) *If the solution to system (10.6.34) is of the form* $\bar{\mathbf{u}}_0 = [\mathbf{u}_0 \quad \lambda]^T$, *where* $\lambda \neq 0$, *then* $\mathbf{u}_0$ *is a solution to the equation*

$$A\mathbf{u} = \mathbf{f} - \lambda\mathbf{u}^* \tag{10.6.35}$$

such that $(\mathbf{u}_0, \mathbf{1}) = 0$.

Proof: The proof follows much as the proof of Proposition 10.6.3. The only difference is that we must use the complete result (we do not have a symmetric matrix) that $N(A) \oplus R(A^T) = \mathbb{R}^L$ and $N(A^T) \oplus R(A) = \mathbb{R}^L$ (i.e., the null space of A is independent of the range of A^T, the span of the rows of A; and the null space of A^T is independent to the range of A, the span of the columns of A).

We note that as in the case of the first order approximation, if the discrete compatibility condition is not satisfied (as it generally will not be), system (10.6.34) adjusts the right hand side the appropriate amount. We also note that the discrete compatibility condition (10.6.33) is a $\mathcal{O}(\Delta x^2)$ approximation of the analytic compatibility condition (10.6.3).

We now obtain the following convergence result, analogous to Theorems 10.3.3 and 10.6.4.

Theorem 10.6.7 *Let $v \in C^4(\overline{R})$ be a solution to Neumann problem (10.6.1)-(10.6.2). Let $\bar{\mathbf{u}} = [\mathbf{u}_0 \; \lambda]^T$ be a solution to equation (10.6.34). Then*

$$|\lambda| = \mathcal{O}(\Delta x^2)$$
$$\|\mathbf{u}_0 - \mathbf{v}\|_\infty = \mathcal{O}(\Delta x^2).$$

Proof: See [21], page 71.

Remark: We should note that the nonsymmetry in the matrix associated with the second order approximation can be approached by another method. If we define the $(M+1) \times (M+1)$ diagonal matrices

$$D_1 = \begin{pmatrix} \frac{1}{4} & 0 & \cdots & & \\ 0 & \frac{1}{2} & 0 & \cdots & \\ & & \ddots & & \\ & \cdots & 0 & \frac{1}{2} & 0 \\ & & \cdots & 0 & \frac{1}{4} \end{pmatrix},$$

$$D_2 = 2D_1 = \begin{pmatrix} \frac{1}{2} & 0 & \cdots & & \\ 0 & 1 & 0 & \cdots & \\ & & \ddots & & \\ & \cdots & 0 & 1 & 0 \\ & & \cdots & 0 & \frac{1}{2} \end{pmatrix}$$

and the $(M+1) \times (M+1)$ block diagonal matrix

$$D = \begin{pmatrix} D_1 & \Theta & \cdots & & \\ \Theta & D_2 & \Theta & \cdots & \\ & & \ddots & & \\ & \cdots & \Theta & D_2 & \Theta \\ & & \cdots & \Theta & D_1 \end{pmatrix},$$

then the matrix DA is symmetric. Hence, instead of considering equation (10.6.26), we consider equation

$$DA\mathbf{u} = D\mathbf{f}. \tag{10.6.36}$$

The same results used in Section 10.6.1 can be used with equation (10.6.36) to obtain solvability conditions, etc. The results obtained in this manner are equivalent to those given in Propositions 10.6.5 and 10.6.6.

10.6.3 Second Order Approximation on an Offset Grid

In Section 2.3 we showed that when we have Neumann boundary conditions, it is more logical to use an offset grid, i.e., a grid where the boundary points fall halfway between the grid points instead of on the grid points, for example, if we consider the grid

$$\begin{aligned} G = \{(x_j, y_k) : x_j &= (j-1)\Delta x + \Delta x/2,\ j = 0, \ldots, M, \\ y_k &= (k-1)\Delta x + \Delta x/2,\ k = 0, \ldots, M\} \end{aligned} \tag{10.6.37}$$

where $\Delta x = \Delta y = 1/(M-1)$. A picture of the grid is given in Figure 10.6.3. Note that there are no grid points on the boundaries of the region and the points associated with $j = 0$, $j = M$, $k = 0$, and $k = M$ are fictitious points outside of $[0,1] \times [0,1]$. As we have done before, we consider the equation

$$\frac{1}{\Delta x^2}\left(\delta_x^2 + \delta_y^2\right) u_{j\,k} = F_{j\,k}, \quad j,k = 1, \ldots, M-1. \tag{10.6.38}$$

Using a centered approximation for the derivatives applied at the boundary (which are not grid points), we get the following approximations of the Neumann boundary conditions.

$$-\frac{u_{1\,k} - u_{0\,k}}{\Delta x} = g_{1/2\,k}, \quad k = 1, \ldots, M-1 \tag{10.6.39}$$

$$\frac{u_{M\,k} - u_{M-1\,k}}{\Delta x} = g_{M-1/2\,k}, \quad k = 1, \ldots, M-1 \tag{10.6.40}$$

$$-\frac{u_{j\,1} - u_{j\,0}}{\Delta x} = g_{j\,1/2}, \quad j = 1, \ldots, M-1 \tag{10.6.41}$$

$$\frac{u_{j\,M} - u_{j\,M-1}}{\Delta x} = g_{j\,M-1/2}, \quad j = 1, \ldots, M-1. \tag{10.6.42}$$

We note specifically that evaluating g at $j = \frac{1}{2}$, $j = M - \frac{1}{2}$, $k = \frac{1}{2}$, and $k = M - \frac{1}{2}$ centers the normal derivative evaluation on the boundary of the domain.

If we then proceed as we have done in the last two sections, use equations (10.6.39)–(10.6.42) to eliminate $u_{j\,0}$, $u_{j\,M}$, $u_{0\,k}$, and $u_{M\,k}$ from the equations given in (10.6.38), and write this system as a matrix equation, we get the same system that we got in Section 10.6.1. Hence, equations

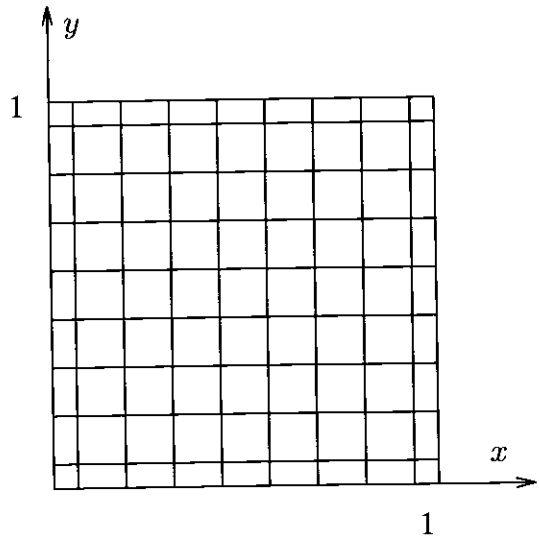

FIGURE 10.6.3. An example of a two dimensional offset grid.

(10.6.38)–(10.6.42) can be written in matrix form as equation (10.6.9). We should recall that this is the same type of situation that we found in Example 2.3.5 (where we found that exactly the same numerical scheme that was not consistent with respect to the usual grid, and hence could not be proved convergent by the Lax Theorem, was consistent with respect to the offset grid, and could be proved convergent by the Lax Theorem).

The next obvious steps that should be taken are to prove analogues of Propositions 10.6.2 and 10.6.3 and Theorem 10.6.4. Clearly, since the system of equations that we consider on the offset grid is the same system considered in Section 10.6.1, it is not necessary to prove analogues to Propositions 10.6.2 and 10.6.3. One difference between the approach taken in Section 10.6.1 and here is that the discrete compatibility condition with respect to the offset grid appears to be a better approximation of the analytic compatibility condition (i.e., we actually integrate over the entire region) than in Section 10.6.1. Of course, because we consider exactly the same matrix equation for the offset grid as we did in Section 10.6.1, the discrete compatibility conditions are the same (the discrete compatibility condition was due to the equation being solved, not the grid).

The major difference between considering equation (10.6.9) as an approximation to our problem on the usual grid versus on the offset grid is the convergence as Δx approaches zero. Since the approximation of the boundary conditions is now a second order approximation, we obtain the following convergence theorem.

Theorem 10.6.8 *Let the function $v \in C^4(\overline{R})$ be a solution to Neumann problem (10.6.1)–(10.6.2). Let $\bar{\mathbf{u}} = [\mathbf{u}_0\ \lambda]^T$ be a solution to equation (10.6.18) with respect to the offset grid (10.6.37). Then*

$$|\lambda| = \mathcal{O}(\Delta x^2)$$
$$\|\mathbf{u}_0 - \mathbf{v}\|_\infty = \mathcal{O}(\Delta x^2).$$

Hence, we see the dichotomy. From Theorem 10.6.4 we see that the solution of equation (10.6.18) converges first order in Δx to the solution of problem (10.6.1)–(10.6.2) when we consider equation (10.6.18) to approximate problem (10.6.1)–(10.6.2) on the usual grid. However, the same solution of equation (10.6.18) converges to the solution of problem (10.6.1)–(10.6.2) second order in Δx when we consider equation (10.6.18) to approximate problem (10.6.1)–(10.6.2) on the offset grid. Clearly, it pays to choose carefully the grid to be used.

10.7 Numerical Solution of Neumann Problems

10.7.1 Introduction

As we did with the Dirichlet problem, now that we know that the solution to the discretized Neumann problem converges to the solution of the analytic Neumann problem and we know that though the discrete problem is not uniquely solvable, it is uniquely solvable to an additive constant, we turn to the task of solving the discrete problems. At this point we know enough about our discrete problems to know that the numerical solution of these problems will not be as straightforward as it was for their Dirichlet counterpart. A summary of our situation is as follows.

- We wish to solve either system (10.6.9) or (10.6.26) but know that neither of these systems has a unique solution.
- Most often—when the discrete compatibility condition is not satisfied because it is only an approximation to the analytic compatibility condition—a solution to systems (10.6.9) and (10.6.26) does not exist.
- The situation is not as bad as it seems. Systems (10.6.18) and (10.6.34) are uniquely solvable and give approximate solutions to the discrete Neumann problem.

Thus, it seems clear that the most obvious approach is to consider solving equations (10.6.18) or (10.6.34). Of course, we do not want to try to solve either equation (10.6.18) or (10.6.34) directly, so we consider iterative solvers. If we inspect both of these systems, we find the following facts.

- If the matrix A is symmetric, then matrix $\bar{A}$ in equation (10.6.18) is symmetric. Even though matrix $\bar{A}$ in equation (10.6.34) is not symmetric, $\bar{A}$ can easily be symmetrized using the matrix D defined in the Remark, page 383.

- Neither of the matrices $\bar{A}$ are positive definite (due to the zero on the diagonal).

- Neither of the matrices $\bar{A}$ are consistently ordered.

- The matrix $\bar{A}$ inherits most of its eigenvalues and eigenvectors from the matrix A. If $\mu \neq 0$ is an eigenvalue of A and $\mathbf{x}$ the eigenvector associated with μ, then μ is an eigenvalue of $\bar{A}$ associated with the eigenvector $[\mathbf{x} \quad 0]^T$.

- The zero eigenvalue of A (and its associated eigenvector $\mathbf{1}$) corresponds to the eigenvalue $\mu = \sqrt{L}$ and eigenvector $\mathbf{x} = [\mathbf{1}^T \quad \sqrt{L}]^T$ of $\bar{A}$ (where $L = (M-1)^2$ and $L = (M+1)^2$ for the cases of first and second order approximation, respectively).

Thus *we cannot apply Jacobi, Gauss-Seidel, or SOR relaxation schemes to solve equations (10.6.18) or (10.6.34)* (if we write $\bar{A}$ as $L+D+U$, D^{-1} will not exist).

10.7.2 Residual Correction Schemes

A different approach is to try to apply the iterative schemes directly to equation (10.6.9) or (10.6.26). It should be clear that we can apply the residual correction algorithms to these problems. Because A is not invertible, it is not clear that the iterations will not converge to an approximate solution of either (10.6.9) or (10.6.26). To be specific, *we consider solving equation (10.6.26)* and assume that $\mathbf{f}$ satisfies $(\mathbf{f}, \mathbf{u}^*) = 0$ (the discrete compatibility condition is satisfied).

It is not difficult to see that since $\mathbf{1}$ is an eigenvector of A associated with the zero eigenvalue,

$$R_J\mathbf{1} = (I - BA)\mathbf{1} = \mathbf{1} - \boldsymbol{\theta} = (1)\mathbf{1},$$

i.e., $\lambda = 1$ is an eigenvalue of the matrix R_J. Hence, *the spectral radius of R_J will not be less than one.*

This is not terrible. *Suppose that R is an iteration matrix associated with a residual correction scheme for solving equation (10.6.26), the eigenvalues of R satisfy*

$$\lambda_1 = 1 > |\lambda_2| \geq \cdots |\lambda_L|,$$

the eigenvector associated with $\lambda_1 = 1$ is $\mathbf{x}_1 = \mathbf{1}$, and the eigenvectors of the matrix R are independent. An analysis similar to that used in Section 10.5.1

will be used to show that the residual correction scheme can still be used to find the solution. Denote the eigenvectors of R by $\mathbf{x}_1 = 1, \mathbf{x}_2, \ldots, \mathbf{x}_L$ and write the initial error, $\mathbf{e}_0$, as

$$\mathbf{e}_0 = a_1\mathbf{x}_1 + \sum_{j=2}^{L} a_j\mathbf{x}_j.$$

Then, after $k+1$ iterations, the error is given by

$$\mathbf{e}_{k+1} = R^{k+1}\mathbf{e}_0 = a_1\mathbf{x}_1 + \sum_{j=2}^{L} \lambda_j^{k+1} a_j\mathbf{x}_j.$$

If k is large enough, the terms in the sum are negligible, and $\mathbf{e}_{k+1} \approx a_1\mathbf{x}_1$ (the error did not and will not converge to zero) and the $(k+1)$-st iterate can be written approximately as

$$\mathbf{w}_{k+1} = \mathbf{u} - \mathbf{e}_{k+1} \approx \mathbf{u} - a_1\mathbf{x}_1.$$

If we want a solution to equation (10.6.9) that satisfies $(\mathbf{u}, 1) = 0$, we see that

$$(\mathbf{w}_{k+1}, 1) = (\mathbf{u}, 1) - a_1(\mathbf{x}_1, 1) = -a_1 L.$$

Hence,

$$a_1 \approx -\frac{1}{L}(\mathbf{w}_{k+1}, 1),$$

and $\mathbf{u} \approx \mathbf{w}_{k+1} + a_1 1$. Thus we see that *if $\mathbf{x}_1 = 1$ is the eigenvector associated with $\lambda_1 = 1$, the effect of this eigenpair $(\lambda_1, \mathbf{x}_1)$ on the iteration can be subtracted off.*

10.7.3 Jacobi and Gauss-Seidel Iteration

The problem with the Jacobi iteration is that R_J does not generally satisfy the conditions required of R above. Since $\lambda_1 = 1$ is an eigenvalue of R_{GS}, by Proposition 10.5.6 we see that both $\lambda_1 = 1$ and $\lambda_{-1} = -1$ will be eigenvalues of R_J. To see what happens if we try to use the Jacobi scheme to solve equation (10.6.26), we denote the eigenvector associated with $\lambda_{-1} = -1$ as $\mathbf{x}_{-1}$, again assume that the eigenvectors are independent, and write the initial error as

$$\mathbf{e}_0 = a_1\mathbf{x}_1 + a_{-1}\mathbf{x}_{-1} + \sum_{j=3}^{L} a_j\mathbf{x}_j.$$

Then, after $k+1$ iterations, we have

$$\mathbf{e}_{k+1} = a_1\mathbf{x}_1 + (-1)^{k+1}a_{-1}\mathbf{x}_{-1} + \sum_{j=3}^{L} \lambda_j^{k+1} a_j\mathbf{x}_j.$$

When k is large enough so that the sum is small, we arrive at a situation very similar to the one discussed above, except that we now have two components of error that do not iterate away. Also, the component of the error associated with $\lambda_{-1} = -1$ will oscillate wildly, so that it is difficult to decide whether k is large enough so that the sum is small. Of course, if $\mathbf{x}_{-1}$ were known, we could proceed as we did above and subtract both components of the error from the iterate. However, since the eigenvector associated with $\lambda_{-1} = -1$ is generally not known, *using the Jacobi iteration for solving equation (10.6.26) is unacceptable.*

If we return to Proposition 10.5.6, we see that there is at least hope that the Gauss-Seidel scheme can be used to solve equation (10.6.26). Since the eigenvalues of R_{GS} are the squares of the eigenvalues of R_J, the problem of the -1 eigenvalue is eliminated. Since $R_{GS} = I - BA$ and $A\mathbf{1} = \boldsymbol{\theta}$, it is easy to see that not only is $\lambda = 1$ an eigenvalue of R_{GS}, but $\mathbf{x}_1 = \mathbf{1}$ is again the eigenvector associated with $\lambda_1 = 1$. As the analysis done earlier shows, it is not a problem that we have one eigenvalue equal to 1 as long as we know the associated eigenvector. We must concern ourselves with the possibility that $\lambda = 1$ may not be a simple eigenvalue. If we consider the matrix equation

$$R_{GS}\mathbf{u} = -(L + D)^{-1}U\mathbf{u} = 1\mathbf{u}$$

(the general eigenvector problem associated with $\lambda = 1$), we see that this is equivalent to

$$-U\mathbf{u} = (L + D)\mathbf{u},$$

or

$$(L + D + U)\mathbf{u} = \boldsymbol{\theta}.$$

Thus the problem associated with finding the eigenvector of R_{GS} associated with $\lambda = 1$ is equivalent to finding the null vector of the matrix A. Using the same argument used in the proof of part (2) of Proposition 10.6.2, we obtain the following result.

Proposition 10.7.1 *$\lambda = 1$ is a simple eigenvalue of R_{GS}, and its associated eigenvector is* $\mathbf{1}$.

Remark: Above we use Proposition 10.5.6. One of the hypotheses of Proposition 10.5.6 is that we have Dirichlet boundary conditions. Since Proposition 10.5.6 involves the very general difference equation (10.5.13), the application of Proposition 10.5.6 is acceptable in this situation. We can reformulate either equations (10.6.9) or equation (10.6.26) as a difference equation of the form of difference equation (10.5.13) with zero Dirichlet boundary conditions. The numerical treatment of the normal derivatives is absorbed into the definition of the stencil and the right hand side.

Before we can claim that the Gauss-Seidel scheme can be used to solve equation (10.6.26), we must show that R_{GS} satisfies the other properties assumed for R in the analysis done earlier in this section. We must first show

that the eigenvalues, other than $\lambda_1 = 1$, satisfy $|\lambda| < 1$. It is not difficult to use the discrete separation of variables approach used in Example 10.5.1 to show that the eigenvalues of the Jacobi iteration matrix R_J associated with solving equation (10.6.26) are given by

$$\lambda_s^p = \frac{1}{2}\left(\cos\frac{p\pi}{M} + \cos\frac{s\pi}{M}\right), \quad p, s = 0, \ldots, M, \tag{10.7.1}$$

and the eigenvector associated with λ_s^p is given by

$$u_{j\,k}^{p\,s} = \cos\frac{p\pi j}{M}\cos\frac{s\pi k}{M}, \quad j, k = 0, \ldots, M. \tag{10.7.2}$$

We note that as we promised earlier, both ± 1 are eigenvalues of R_J. Because the eigenvalues of R_{GS} are squares of the eigenvalues of R_J, we see that all of the eigenvalues of R_{GS}, except $\lambda_1 = 1$, satisfy $|\lambda| < 1$. Again we make special note that even though both ± 1 are eigenvalues of R_J and $(\pm 1)^2 = 1$ is an eigenvalue of R_{GS}, $\lambda_1 = 1$ is a simple eigenvalue of R_{GS}.

We still have one more problem to face in trying to use the Gauss-Seidel iteration scheme to solve equation (10.6.26). One of the assumptions made above is that the matrix R has L independent eigenvectors. The matrix R_{GS} does not have L independent eigenvectors. Though this does not allow us to use the analysis above directly, the result is still true. If you recall, this is the same situation as we had with the rate of convergence results in Section 10.5.1. Though the matrix R_{GS} does not have L independent eigenvectors, generalized eigenvectors can be used in the argument given above to obtain a very similar result. The result is that *the Gauss-Seidel iteration scheme can be used to solve the approximation to the Neumann boundary value problem given by equations (10.6.4)-(10.6.8).*

Remark 1: If we choose an initial guess such that the initial error has no component in the $\mathbf{x}_1 = \mathbf{1}$ direction $((\mathbf{e}_0, \mathbf{1}) = 0$, which will be the case if we choose an initial guess $\mathbf{w}_0 = \boldsymbol{\theta})$, then the iteration $\mathbf{w}_k$ will never have such a component, and it will not have to be subtracted off.

Remark 2: If we are faced with the most common situation where $(\mathbf{f}, \mathbf{u}^*) \neq 0$, we can still use the Gauss-Seidel scheme to approximate the solution to the analytic Neumann problem. If $(\mathbf{f}, \mathbf{u}^*)$ is nonzero because the discrete compatibility condition $(\mathbf{f}, \mathbf{u}^*) = 0$ only approximates the analytic compatibility condition, we must consider equation (10.6.34). When we considered equation (10.6.34), we saw that if the solution was of the form $[\mathbf{u}_0 \quad \lambda]$ where $\lambda \neq 0$, then $\mathbf{u}_0$ satisfies

$$A\mathbf{u}_0 = \mathbf{f} - \lambda\mathbf{u}^*. \tag{10.7.3}$$

If $\mathbf{u}_0$ is a solution to equation (10.7.3), then $\mathbf{f} - \lambda\mathbf{u}^*$ must satisfy $(\mathbf{f} - \lambda\mathbf{u}^*, \mathbf{u}^*) = 0$. Hence,

$$\lambda = \frac{1}{L^*}(\mathbf{f}, \mathbf{u}^*),$$

where $L^* = (\mathbf{u}^*, \mathbf{u}^*) = (M-2)(M-3) + \frac{1}{4}$.

Thus, we see that *when* $(\mathbf{f}, \mathbf{u}^*) \neq 0$, *we replace* $\mathbf{f}$ *by*

$$\mathbf{f} - \frac{(\mathbf{f}, \mathbf{u}^*)}{L^*}\mathbf{u}^*$$

and then proceed to use Gauss-Seidel to solve $A\mathbf{u} = \mathbf{f}$.

Remark 3: There are times when it is impossible to do as we suggest in Remark 2, adjust the right hand side and proceed. The most common such situation is when it is too difficult to compute $\mathbf{u}^*$. It is then best to compute with the "bad" right hand side and adjust the solution, for example, if we consider the equation associated with the second order approximation of the Neumann boundary condition, (10.6.26), with $(\mathbf{f}, \mathbf{u}^*) \neq 0$. If $\mathbf{u}$ is the solution to $A\mathbf{u} = \mathbf{f}$, we set $\mathbf{u}_0 = \mathbf{u} - \lambda^* 1$, where $\lambda^* = (\mathbf{u}, 1)/L$. We see that $\mathbf{u}_0$ satisfies

$$\begin{aligned} A\mathbf{u}_0 &= A(\mathbf{u} - (\mathbf{u}, 1)1) \\ &= A\mathbf{u} - (\mathbf{u}, 1)A1 \\ &= \mathbf{f}, \end{aligned}$$

and $(\mathbf{u}_0, 1) = (\mathbf{u}, 1) - \lambda^*(1, 1) = 0$. The component of 1 can be subtracted off at the end of the iterative procedure (the cheapest way) or can be subtracted off after each iteration (sometimes the most reassuring way).

Remark 4: It should be noted that the same results are true of the first order approximation of the Neumann boundary conditions and equation (10.6.9). The Jacobi iteration scheme is not useful because ± 1 are both eigenvalues of R_J. Since $\lambda_1 = 1$ is a simple eigenvalue of R_{GS}, -1 is not an eigenvalue, and the remaining eigenvalues have magnitude less than one. In the case of equation (10.6.9), it is much more difficult (and maybe impossible) to show that the eigenvalues of R_{GS}, other than $\lambda_1 = 1$, satisfy $|\lambda| < 1$. To convince us that this is true, we introduce another tool that we have at our disposal. Using some convenient software package (we used Matlab), we set $M = 10$, fill a matrix, and compute $R_J = -D^{-1}(L + U)$. It is an easy matter to let a machine compute the eigenvalues for us. We then use the fact that the nonzero eigenvalues of R_{GS} are the squares of the eigenvalues of R_J. (We could have computed the eigenvalues of R_{GS} directly, but the eigenvalues of R_J will be useful to us later.) We can then try several other values of M (whatever the machine will allow). A computation such as this does not provide a proof that the other eigenvalues of R_{GS} are less than one in magnitude. It does, however, give us enough confidence of that fact to let us proceed with our calculations, and it gives us confidence that there is probably an analytic result that will prove that the other eigenvalues of R_{GS} are less than one in magnitude. See HW10.7.1. Thus, the Gauss-Seidel scheme can be used to determine the solution up to an additive multiple of 1 and this additive multiple can be subtracted off.

10.7.4 SOR Scheme

Since the Gauss-Seidel scheme can be used to solve equations (10.6.9) and (10.6.26), we now wonder whether we can obtain better results using the SOR scheme with an ω that is either optimal or at least better than $\omega = 1$. Again, we consider solving equation (10.6.26). If we return to the analysis for the SOR scheme given in Section 10.5.9, we see that since the eigenvalues of R_J satisfy $|\lambda| \leq 1$, the eigenvalues of $R_{SOR,\omega}$ will satisfy $|\lambda| \leq 1$. Using (10.5.63) (with the + sign), we also see that $\lambda_1 = 1$ will be a simple eigenvalue of $R_{SOR,\omega}$, as was the case with the Gauss-Seidel scheme. Thus, if we again are willing to subtract off the component of the error due to the eigenpair $\lambda_1 = 1$ and $\mathbf{x}_1 = \mathbf{1}$, the SOR scheme will converge to the solution of equation (10.6.26). To choose the optimal parameter ω_b, we return to the proofs of Lemma 10.5.13 and Proposition 10.5.12. Since $\bar{\lambda}_J = 1$ is associated with the eigenvalue $\lambda_1 = 1$ of $R_{SOR,\omega}$ and the component of the error associated with $\lambda_1 = 1$ and $\mathbf{x}_1 = \mathbf{1}$ is going to be subtracted off, *we must determine ω_b so as to minimize the value of the second largest eigenvalue.* Following the proofs of Lemma 10.5.13 and Proposition 10.5.12, we see that we must choose

$$\omega_b = \frac{2}{1 + \sqrt{1 - (\lambda_0^1)^2}} = \frac{2}{1 + \sqrt{1 - \frac{1}{4}(1 + \cos\frac{\pi}{M})^2}}. \tag{10.7.4}$$

We note that λ_0^1 was chosen for formula (10.7.4) because of all the eigenvalues of R_J other than $\lambda_{\pm 1} = \pm 1$, λ_0^1 has the largest magnitude (and of course, λ_1^0 would work equally well).

Remark: Of course, everything stated above for using the SOR scheme to solve equation (10.6.26) holds equally well for using the SOR scheme to solve equation (10.6.9) except for the fact that we do not have an expression for the eigenvalues of R_J to use in formula (10.7.4) for determining ω_b. Of course, for $M = 10$, 20, and 40, the results found in HW10.7.1 can be used to determine ω_b.

10.7.5 Approximation of ω_b

We should also realize that the approach used in Section 10.5.12 can be used to approximate the value of ω_b (for the case of solving equation (10.6.9) and for the case of Neumann boundary value problems more general than problem (10.6.1)–(10.6.2)). We note that if we let $\lambda_1 = 1$, let $\lambda_2, \ldots, \lambda_L$ be such that

$$\lambda_1 > |\lambda_2| > |\lambda_3| \geq \cdots \geq |\lambda_L|,$$

and return to equation (10.5.112), we see that in our situation we have

$$\mathbf{w}_{k+1} - \mathbf{w}_k = \mathbf{e}_k - \mathbf{e}_{k+1} \tag{10.7.5}$$

$$= (a_1\mathbf{x}_1 + a_2\lambda_2^k\mathbf{x}_2 + \sum_{j=3}^{L} a_j\lambda_j^k\mathbf{x}_j)$$

$$- (a_1\mathbf{x}_1 + a_2\lambda_2^{k+1}\mathbf{x}_2 + \sum_{j=3}^{L} a_j\lambda_j^{k+1}\mathbf{x}_j)$$

$$\approx \lambda_2 a_2 \mathbf{x}_2(\lambda_2^{k-1} - \lambda_2^k). \tag{10.7.6}$$

We get

$$\frac{\|\mathbf{w}_{k+1} - \mathbf{w}_k\|}{\|\mathbf{w}_k - \mathbf{w}_{k-1}\|} \approx \lambda_2,$$

or

$$\frac{w_{j_{k+1}} - w_{j_k}}{w_{j_k} - w_{j_{k-1}}} \approx \lambda_2,$$

for some j and for sufficiently large k. Hence, exactly the same steps that we used to approximate ω_b for use in the approximate solution of Dirichlet boundary value problems work for approximating ω_b for use in the approximate solutions of Neumann boundary–value problems.

Remark: Expression (10.7.5)–(10.7.6) also indicates that because the contribution from the $a_1\mathbf{x}_1$ term will add out, we can use $\|\mathbf{w}_{k+1} - \mathbf{w}_k\|$ as our stopping criterion. We can also use the residual as a stopping criterion. Since A will annihilate the component of $\mathbf{w}_k$ in the direction of the $\mathbf{1}$ vector, when the residual $\mathbf{r}_k = \mathbf{f} - A\mathbf{w}_k$ is small, we know that the residual $\mathbf{f} - A\mathbf{u}_{k_0}$ will be small, where $\mathbf{u}_{k_0} = \mathbf{w}_k - \lambda^*\mathbf{1}$ and $\lambda^* = (\mathbf{w}_k, \mathbf{1})/L$.

HW 10.7.1 Use some software package to calculate the eigenvalues and the eigenvectors associated with the Jacobi iteration matrix $R_J = -D^{-1}(L+U)$ for the matrix A given in (10.6.10). Use $M = 10$, $M = 20$, and $M = 40$. (2) Prove or disprove that all of the eigenvalues of the matrix A, except $\lambda_1 = 1$, are less than one in magnitude.

HW 10.7.2 Use the discrete separation of variables technique to verify the eigenvalues and eigenvectors given by formulas (10.7.1) and (10.7.2).

HW 10.7.3 Use the discrete separation of variables technique to compute the eigenvalues of R_{GS} associated with the matrix A given in (10.6.10).

10.7.6 Implementation: Neumann Problems

If the suggestions for implementation given in Section 10.5.5 were utilized, implementation of Gauss-Seidel and SOR schemes for approximating the solution of Neumann boundary–value problems should be easy. A part of the implementation suggested in Section 10.5.5 (that was wasteful) was to use a stencil array $S(j,k,m)$. It was suggested that this array be defined for all $j = 0,\ldots,M_x$ and $k = 0,\ldots,M_y$. The stencil values for $j = k = 0$, $j = M_x$ and $k = M_y$ were never used. If the implementation was done in that manner, the code for approximating the solution of Neumann boundary–value problems can be produced by changing the stencil array, the right hand side, and, perhaps changing the range of the loops.

To write a computer program for solving a Neumann boundary–value problem using the first order approximation of the Neumann boundary conditions, say a problem like the problem given in HW10.7.4, it is reasonably easy to start with a program designed to solve an analogous Dirichlet problem, say the problem given in HW10.5.2, and make the following changes (assuming that we are solving a problem written in the form $-\nabla^2 v = F$).

1. The appropriate entries of $S(j,k,m)$ for $j = 1$, $k = 1$, $j = M-1$, and $k = M-1$ must be changed to reflect the 2's and 3's on the diagonal of matrix (10.6.10). For example, when $j = 1$ we set $S(1,k,2) = 0$, $k = 1,\ldots,M-1$, $S(1,k,0) = 3/\Delta x^2$ for $k = 2,\ldots,M-2$, $S(1,k,0) = 2/\Delta x^2$ when $k = 1$ and $k = M-1$, and $S(1,1,4) = S(1,M-1,3) = 0$.

2. The boundary values of u and u' (*unew* and *uold*) must be set equal to zero because the effect of the Neumann boundary conditions are being included in both the stencil and the right hand side (but the algorithms will still reach to $j = 0$, $k = 0$, $j = M$, and $k = M$).

3. The right hand side array must be changed to include the effects of the nonhomogeneous boundary terms; see (10.6.13)–(10.6.16). For example, when $j = 1$ we set $f_{1\,k} = F_{1\,k} + g_{0\,k}/\Delta x$ for $k = 2,\ldots,M-1$, $f_{1\,1} = F_{1\,1} + g_{0\,1}/\Delta x + g_{1\,0}/\Delta x$, and $f_{1\,M-1} = F_{1\,M-1} + g_{0\,M-1}/\Delta x + g_{1\,M}/\Delta x$.

4. Since we have zeroed out the boundary values of u and u', before we output our results (either numerically or graphically), it might be best to use the appropriate approximation of the boundary conditions to compute the values of u on the boundary, i.e., use equations (10.6.5)–(10.6.8).

To alter a program designed to solve a Dirichlet boundary–value problem to solve a Neumann boundary value problem using the second order approximation of the Neumann boundary conditions, we must make the following changes (again assuming that the partial differential equation is written as $-\nabla^2 v = F$).

1. The stencil values of $S(j,k,m)$ for $j = 0$, $k = 0$, $j = M$, and $k = M$ must be defined to reflect the fact that we are now solving on the grid points on the boundaries. In addition, the 2's must be inserted in the appropriate places of the stencil array S in accordance with matrix (10.6.27). For example, when $j = 0$ we define $S(0,k,0) = 4/\Delta x^2$, $S(0,k,1) = -2/\Delta x^2$, $S(0,k,2) = 0$, $k = 0,\ldots,M$, $S(0,k,3) = S(0,k,4) = -1/\Delta x^2$, $k = 1,\ldots,M-1$, and $S(0,0,3) = S(0,M,4) = -2/\Delta x^2$, $S(0,0,4) = S(0,M,3) = 0$.

2. As with the first order approximation of the boundary conditions, the right hand side must be adjusted to include the nonhomogeneous terms in the boundary conditions as given in (10.6.29)–(10.6.32).

3. The loops for the iterative algorithms must be change to "For $k = 0,\ldots,M$" and "For $j = 0,\ldots,M$" because we now solve on the boundary grid points. Also, because the iterative schemes will still reach out further, the dimensions of the arrays for u and u' must be increased to include $j = -1$ and $j = M+1$, and $k = -1$ and $k = M+1$, and these new artificial boundary terms must be filled with zeros so as not to affect the iterative scheme.

HW 10.7.4 Use the Gauss-Seidel and SOR schemes along with both the first and second order approximations of the Neumann boundary conditions to find an approximate solution to the problem

$$\begin{aligned}
\nabla^2 v &= e^{x+y} \quad (x,y) \in R = (0,1)\times(0,1)\\
\frac{\partial v}{\partial x}(0,y) &= \frac{1}{2}e^{y}, \quad y \in (0,1)\\
\frac{\partial v}{\partial x}(1,y) &= \frac{1}{2}e^{1+y}, \quad y \in (0,1)\\
\frac{\partial v}{\partial x}(x,0) &= \frac{1}{2}e^{x}, \quad x \in (0,1)\\
\frac{\partial v}{\partial x}(x,1) &= \frac{1}{2}e^{x+1}, \quad x \in (0,1).
\end{aligned}$$

Use $M = 100$ and tolerance $= 1.0 \times 10^{-6}$. When using the SOR scheme, use the optimal values of ω_b determined by equation (10.7.4).

HW 10.7.5 Verify that the right hand side and boundary conditions of the boundary value problem given in HW10.7.4 satisfy the compatibility conditions.

HW 10.7.6 Use the Gauss-Seidel scheme along with both the first and second order approximations of the Neumann boundary conditions to solve the problem

$$\nabla^2 v = -5\pi^2 \cos \pi x \cos 2\pi y - 2 \sin x \sin y, \quad (x,y) \in R = (0,1) \times (0,1)$$
$$\frac{\partial v}{\partial x}(0,y) = \sin y, \quad y \in (0,1)$$
$$\frac{\partial v}{\partial x}(1,y) = \cos 1 \sin y, \quad y \in (0,1)$$
$$\frac{\partial v}{\partial y}(x,0) = \sin x, \quad x \in (0,1)$$
$$\frac{\partial v}{\partial y}(x,1) = \cos 1 \sin x, \quad x \in (0,1).$$

Use $M_x = 20$ and $M_y = 40$. Note that all of the work done in the sections on Neumann boundary conditions has been done with $M_x = M_y$. This was done for notational convenience, and the change should not be difficult. Verify that the analytic compatibility condition is satisfied.

HW 10.7.7 Verify that the right hand side and boundary conditions of the boundary value problem given in HW10.7.6 satisfy the compatibility conditions.

10.8 Elliptic Difference Equations: Mixed Problems

10.8.1 Introduction

Two other types of elliptic boundary value problems that are of interest and that we have not considered are those with a mixture of Dirichlet and Neumann boundary conditions on different parts of the boundary and those that have a sum of the solution and the normal derivative of the solution prescribed on the boundary. It appears from the literature that the names of these problems are not very well established. We shall refer to the general class of boundary conditions as **mixed boundary conditions**, to those boundary conditions that are a mixture of Dirichlet and Neumann boundary conditions as **mixture boundary conditions**, and to the combination of the two as the **Robin boundary condition** (and the problems with these boundary conditions as mixed problems, mixture problems, and Robin problems, respectively). We write the Robin boundary condition as

$$\alpha v + \beta \frac{\partial v}{\partial n} = g \quad \text{on } \partial R. \tag{10.8.1}$$

We write α and β as if they are constants, but the treatment for nonconstant α and β follows analogously (always being careful when either α or β are zero). Also, since it is important to our results, we specify that we the normal derivative is defined with respect to an outwardly directed normal vector.

Analytically, the mixture problem is very much the same as the Dirichlet problem. If the boundary on which we are given a Dirichlet boundary condition is nonempty (and we would not consider the problem to be a mixture problem if the boundary on which we were given a Dirichlet boundary condition were empty), then we obtain uniqueness and continuous dependence just as in the Dirichlet problem. The analytic Maximum Principle is still true because it does not depend on v on the boundary. The additional result that can be proved for mixture problems is that the maximum must occur on the part of the boundary on which we are given a Dirichlet boundary condition. More explicitly, the maximum cannot occur on the part of the boundary on which we are given the Neumann boundary condition. And unlike the Neumann boundary value problem, we do not need a consistency condition. For convenience, we include the following model mixture problem.

$$-\nabla^2 v = F(x,y), \quad (x,y) \in R = (0,1) \times (0,1) \tag{10.8.2}$$

$$v(1,y) = f_2(y), \quad y \in [0,1] \tag{10.8.3}$$

$$v(x,1) = f_3(x), \quad x \in [0,1] \tag{10.8.4}$$

$$v(0,y) = f_4(y), \quad y \in [0,1] \tag{10.8.5}$$

$$-\frac{\partial v}{\partial y}(x,0) = g(x), \quad x \in [0,1] \tag{10.8.6}$$

The results for the Robin problem are also similar to the results for the Dirichlet problem except that they depend on α and β. Specifically, we find that the solution to the Robin problem is unique if $\beta/\alpha > 0$. Evidently, this is the physically interesting case, and we will see later that it is very convenient for our work. We also might warn the reader that when you find a uniqueness result in the literature for $\beta/\alpha < 0$, the result should use the interior normal. The model Robin problem that we will refer to in this section consists of partial differential equation (10.8.2) along with Robin boundary condition (10.8.1), where we redefine g to be g_1, g_2, g_3 and g_4 on $y = 0$, $x = 1$, $y = 1$ and $x = 0$, respectively.

As we usually do, we next must derive a discrete set of equations that will approximate a mixture problem. The treatment follows much of our previous work. We treat the boundaries with Dirichlet boundary conditions as we did in Section 10.2—using a difference equation like (10.2.3) that reaches to boundary conditions like (10.2.4), (10.2.5), (10.2.6) or (10.2.7). We treat the boundaries that have Neumann boundary conditions as we did in Sections 10.6.1 and 10.6.2. As we did in Sections 10.6.1 and 10.6.2, we must decide whether to use a first order or a second order approximation

of the Neumann boundary condition. Depending on the choice of accuracy, we approximate the Neumann boundary conditions as (10.6.5), (10.6.6), (10.6.7) or (10.6.8); or (10.6.22), (10.6.23), (10.6.24) or (10.6.25), and use these equations to eliminate the terms defined on the boundary for first order approximations and to eliminate the ghost points for the second order approximations. Using the appropriate approximations near the boundaries and a difference equation such as (10.2.3) in the interior, we are left with a system of equations to be solved. Specifically, if we consider a uniform grid on $R = [0,1] \times [0,1]$—for convenience choose $M_x = M_y = M$ ($\Delta x = \Delta y$) and choose the second order approximation of the Neumann boundary condition—we approximate mixture problem (10.8.2–(10.8.6) by

$$A\mathbf{u} = \mathbf{f}, \tag{10.8.7}$$

where A is the $M(M-1) \times M(M-1)$ matrix

$$A = \frac{1}{\Delta x^2}\begin{pmatrix} T & -2I & \Theta & \cdots & \\ -I & T & -I & \Theta & \cdots \\ & \ddots & \ddots & \ddots & \\ \cdots & \Theta & -I & T & -I \\ & \cdots & \Theta & -I & T \end{pmatrix}, \tag{10.8.8}$$

T is the $(M-1) \times (M-1)$ matrix

$$T = \begin{pmatrix} 4 & -1 & 0 & \cdots & \\ -1 & 4 & -1 & 0 & \cdots \\ & \ddots & \ddots & \ddots & \\ \cdots & 0 & -1 & 4 & -1 \\ & \cdots & 0 & -1 & 4 \end{pmatrix}, \tag{10.8.9}$$

I is the $(M-1) \times (M-1)$ identity matrix, $\mathbf{u}$ is the $M(M-1)$-vector

$$\mathbf{u} = [u_{1\,0} \cdots u_{M-1\,0}\ u_{1\,1} \cdots u_{M-1\,M-1}]^T,$$

and $\mathbf{f} = \mathbf{F} + \mathbf{b}_{y_1} + \mathbf{b}_{y_2} + \mathbf{b}_x$, where $\mathbf{F}$, $\mathbf{b}_{y_1}$, $\mathbf{b}_{y_2}$, and $\mathbf{b}_x$ are the $M(M-1)$-vectors

$$\mathbf{F} = [F_{1\,0} \cdots F_{M-1\,0}\ F_{1\,1} \cdots F_{M-1\,M-1}]^T,$$

$$\mathbf{b}_{y_1} = \frac{2}{\Delta y}[g_1 \cdots g_{M-1}\ 0 \cdots 0]^T, \quad \mathbf{b}_{y_2} = \frac{1}{\Delta y^2}[0 \cdots 0\ f_{3_1} \cdots f_{3_{M-1}}]^T,$$

and

$$\mathbf{b}_x = \frac{1}{\Delta x^2}[f_{4_0}\ 0 \cdots 0\ f_{2_0}\ f_{4_1}\ 0 \cdots 0\ f_{2_1}\ f_{4_2} \cdots f_{2_{M-1}}]^T.$$

We leave the representation of the mixture problem using the first order approximation of the Neumann boundary condition to the reader in HW10.8.1.

The treatment of the Robin boundary conditions is very similar to the treatment and implementation of problems involving Neumann boundary conditions (and conceptually a bit easier than that for mixture problems). Consider a Robin boundary condition such as (10.8.1) along the boundary $y = 0$ of the region $R = (0,1) \times (0,1)$ (replacing g by g_1). As with Neumann boundary conditions, we must decide whether we want to use a first order or second order approximation of the derivative in the boundary condition. If we choose to use a first order approximation, we write the approximation of boundary condition (10.8.1) as

$$\alpha u_{j\,0} - \beta \frac{u_{j\,1} - u_{j\,0}}{\Delta y} = g_{1_j}, \tag{10.8.10}$$

where the negative sign in front of the β is due to the fact that the normal derivative to the $y = 0$ boundary is $-\partial/\partial y$. Then as we did for the first order approximations of Neumann boundary conditions, we solve equation (10.8.10) for $u_{j\,0}$, substitute this expression into difference equation (10.2.3) with $k = 1$, and get

$$\begin{aligned} -\frac{1}{\Delta x^2} u_{j+1\,1} - \frac{1}{\Delta y^2} u_{j\,2} + \Big(\frac{2}{\Delta x^2} + \frac{2}{\Delta y^2} - \frac{\beta}{\Delta y^2 (\beta + \alpha \Delta y)} \Big) u_{j\,1} \\ - \frac{1}{\Delta x^2} u_{j-1\,1} = F_{j\,1} + \frac{1}{\Delta y (\beta + \alpha \Delta y)} g_{1_j}. \end{aligned} \tag{10.8.11}$$

Of course, the second order approximation of the mixed boundary condition is similar. We approximate boundary condition (10.8.1) by

$$\alpha u_{j\,0} - \beta \frac{u_{j\,1} - u_{j\,-1}}{2\Delta y} = g_{1_j} \tag{10.8.12}$$

(where again the minus sign in front of β is a part of the outwardly directed normal derivative), solve for $u_{j\,-1}$, and substitute this expression into difference equation (10.2.3). We get

$$\begin{aligned} -\frac{1}{\Delta x^2} u_{j+1\,0} - \frac{2}{\Delta y^2} u_{j\,1} + \Big(\frac{2}{\Delta x^2} + \frac{2}{\Delta y^2} + \frac{2\alpha}{\beta \Delta y} \Big) u_{j\,0} - \frac{1}{\Delta x^2} u_{j-1\,0} \\ = F_{j\,0} + \frac{2}{\beta \Delta y} g_{1_j}. \end{aligned} \tag{10.8.13}$$

Specifically, if we again choose a uniform grid on $R = [0,1] \times [0,1]$, $M_x = M_y = M$ ($\Delta x = \Delta y$), and the second order approximation, we can write our model Robin problem as

$$A\mathbf{u} = \mathbf{f}, \tag{10.8.14}$$

where A is the $(M+1)(M+1) \times (M+1)(M+1)$ matrix

$$A = \frac{1}{\Delta x^2} \begin{pmatrix} T & -2I & \Theta & \cdots & \\ -I & T & -I & \Theta & \cdots \\ & \ddots & \ddots & \ddots & \\ \cdots & \Theta & -I & T & -I \\ & \cdots & \Theta & -2I & T \end{pmatrix}, \tag{10.8.15}$$

T is the $(M+1) \times (M+1)$ matrix

$$T = \begin{pmatrix} 4 + 2\Delta y \frac{\alpha}{\beta} & -2 & 0 & \cdots & \\ -1 & 4 & -1 & 0 & \cdots \\ & \ddots & \ddots & \ddots & \\ \cdots & 0 & -1 & 4 & -1 \\ & \cdots & 0 & -2 & 4 + 2\Delta y \frac{\alpha}{\beta} \end{pmatrix}, \tag{10.8.16}$$

I is the $(M+1) \times (M+1)$ identity matrix, $\mathbf{u}$ is the $(M+1)(M+1)$-vector

$$\mathbf{u} = [u_{00} \ \cdots \ u_{M0} \ u_{01} \ \cdots \ u_{MM}]^T,$$

and $\mathbf{f} = \mathbf{F} + \mathbf{b}_y + \mathbf{b}_x$, where $\mathbf{F}$, $\mathbf{b}_x$, and $\mathbf{b}_y$ are the $(M+1)(M+1)$-vectors

$$\mathbf{F} = [F_{00} \ \cdots \ F_{M0} \ F_{01} \ \cdots \ F_{MM}]^T,$$

$$\mathbf{b}_y = \frac{2}{\beta \Delta y} [g_{1_0} \ \cdots \ g_{1_M} \ 0 \ \cdots \ 0 \ g_{3_0} \ \cdots \ g_{3_M}]^T,$$

and

$$\mathbf{b}_x = \frac{2}{\beta \Delta x} [g_{4_0} \ 0 \ \cdots \ 0 \ g_{2_0} \ g_{4_1} \ 0 \ \cdots \ 0 \ g_{2_1} \ g_{4_2} \ \cdots \ g_{4_M} \ \cdots \ g_{2_M}]^T.$$

Of course, we can write the discrete Robin problem using a first order approximation in the boundary condition in a similar form. See HW10.8.2.

HW 10.8.1 Consider the model mixture problem discussed in Section 10.8.1. Use the first order approximation of the Neumann boundary condition, let $M_x = M_y = M$, and express the discrete problem in matrix form $A\mathbf{u} = \mathbf{f}$.

HW 10.8.2 Consider the model Robin problem discussed in Section 10.8.1. Use the first order approximation of the Robin boundary condition, let $M_x = M_y = M$, and express the discrete problem in matrix form $A\mathbf{u} = \mathbf{f}$.

10.8.2 Mixed Problems: Solvability

When we approach new types of problems as we are doing in this section, we should at least consider the of solvability of the discrete problem. Often, proofs concerning the solvability of our problems are difficult, and we must proceed without them, but we should at least realize that we are then in the experimental range. In this section we will discus the solvability of our problems and briefly discuss some topics related to convergence.

It is especially good to consider the solvability of the mixture problems, because they are generally easy. We might be wary that we might not obtain a unique solution, since we did not get a unique solution for Neumann problems. However, the nonuniqueness of the solution to the numerical approximation of the Neumann problem was good, because the analytic problem had the same difficulty. With analytic mixture problems we generally have uniqueness, hence, we should expect the numerical problems to be uniquely solvable.

If we inspect the matrix A given in (10.8.8), it is easy to see that the matrix is diagonally dominant and that each of the rows associated with Dirichlet boundary conditions satisfy the strict inequality. It is also clear that if we change any entry of $\mathbf{f}$, the solution to equation (10.8.7) will change (A is irreducible). Hence, by Proposition 10.2.5, A is invertible and equation (10.8.7) is uniquely solvable. It should not be too difficult to realize that the same result will be true for a much broader class of mixture problems. The important ingredient is that there is at least one point at which we have a Dirichlet boundary condition.

We use the same approach to show that the mixture problem using the first order approximation of the Neumann boundary condition is uniquely solvable. Inspection of the matrix A found in HW10.8.1, shows that the hypotheses of Proposition 10.2.5 are satisfied and the discrete mixture problem is uniquely solvable.

Solvability for the Robin problem is somewhat more interesting. If we inspect the matrix A given in (10.8.15), we see that the matrix is diagonally dominant and has a strict inequality on each row associated with a point on the boundary. We see that it is the $2\Delta y\alpha/\beta$ term that makes the inequality strict at these points. This is consistent with the assumption that $\alpha/\beta > 0$. We see that if $\alpha = 0$, then the matrix is still diagonally dominant, but it has no rows where the inequality is strict. When $\alpha = 0$, the Robin boundary condition reduces to a Neumann boundary condition, so this should not surprise us. We notice that when $\alpha/\beta < 0$, the matrix is not diagonally dominant. We emphasize that the fact that A is not diagonally dominant (or does not satisfy the rest of the hypotheses of Proposition 10.2.5) does not imply that A is not invertible. It is interesting that the same assumption, $\alpha/\beta > 0$, was necessary both for the uniqueness of the analytic problem and the solvability for the discrete problem. See HW10.8.9.

The solvability of the discrete Robin problem with the first order ap-

proximation of the Robin boundary condition also follows from Proposition 10.2.5. Inspecting the matrix A found in HW10.8.2, we see that the rows associated with points that do not reach to the boundary satisfy the diagonal dominance inequality. The rows of interest are those associated with the boundary condition-like equation (10.8.11). For a row of the matrix associated with equation (10.8.11), we see that

$$\begin{aligned} |a_{\ell\ell}| - \sum_{\substack{k=1 \\ k\neq\ell}}^{(M-1)(M-1)} |a_{\ell k}| &= \Big(\frac{2}{\Delta x^2} + \frac{2}{\Delta y^2} - \frac{\beta}{\Delta y^2(\beta+\alpha\Delta y)}\Big) \\ &\quad - \Big(\frac{2}{\Delta x^2} + \frac{1}{\Delta y^2}\Big) \\ &= \frac{1}{\Delta y^2} - \frac{\beta}{\Delta y^2(\beta+\alpha\Delta y)} = \frac{1}{\Delta y^2}\frac{\alpha\Delta y}{\beta+\alpha\Delta y} > 0. \end{aligned}$$

Hence, the rows associated with the points that reach to the boundary satisfy the strict inequality. (The rows associated with the corner points $(1,1)$, $(M-1,1)$, $(1,M-1)$, and $(M-1,M-1)$ are similar and left to the reader in HW10.8.3.) Then, since the matrix is irreducible, the discrete Robin problem with the first order approximation of the Robin boundary condition is uniquely solvable.

Another important aspect of the scheme that we should at least consider is the convergence of the solution of the discrete problem to the solution of the analytic problem. We will not prove convergence of the schemes for mixed problems. Proofs for our model equations are surely available. However, they will surely be more difficult than the convergence proofs already considered. We emphasize that the cornerstone of the convergence proof given in Section 10.3 is the Maximum Principle. We should understand that Proposition 10.3.1 applies to mixed problems. Proposition 10.3.1 does not care what type of boundary conditions we consider. In HW10.8.4-(a), for the mixture problem with a second order approximation of the Neumann boundary condition, we prove in addition that the maximum cannot occur on the part of the boundary with a Neumann boundary condition.

The Maximum Principle for the mixture problem with a first order approximation of the Neumann boundary condition is given in HW10.8.4-(b). The result given in HW10.8.4-(b) is that if we have a zero Neumann boundary condition on a part of the boundary (which is what we would have in a convergence proof), we use the techniques used in the proof of Proposition 10.3.1 to show that the maximum cannot occur at any of the grid points adjacent to the boundary with a Neumann boundary condition. The zero Neumann boundary condition is then used to show that the maximum cannot occur at the adjacent boundary points either. The results given in HW10.8.4 are given for the setting used in the model mixture problem given in Section 10.8.1. It is clear, we hope, that the results can be made much more general.

HW 10.8.3 Consider the scheme for solving the Robin problem using the first order approximation of the boundary condition. Show that the row of the matrix A associated with the corner points of the domain satisfy the strict inequality

$$|a_{\ell\ell}| - \sum_{\substack{k=1 \\ k\neq\ell}}^{(M-1)(M-1)} |a_{\ell k}| \geq \frac{1}{\Delta x^2}\frac{\alpha\Delta x}{\beta+\alpha\Delta x} + \frac{1}{\Delta y^2}\frac{\alpha\Delta y}{\beta+\alpha\Delta y} > 0.$$

HW 10.8.4 Consider the region R and the types of boundary conditions used in the model mixture problem considered in Section 10.8.1. Consider a uniform grid on R with $M_x = M_y = M$. Let G_R, G_R^0, ∂G_R, ∂G_{R_D}, and ∂G_{R_N} denote the grid points over the entire region, in the interior of the region, on the boundary of the region, on the part of the boundary on which we are given a Dirichlet boundary condition, and on the part of the boundary on which we are given a Neumann boundary condition, respectively.
(a) Define L_{jk}, $j = 1, \ldots, M-1$, $k = 0, \ldots, M-1$, as

$$L_{jk}u_{jk} = \begin{cases} -\frac{1}{\Delta x^2}\left(\delta_x^2 + \delta_y^2\right)u_{jk} & \text{when } j,k = 1,\ldots,M-1 \\ -\frac{1}{\Delta x^2}\left(-4u_{jk} + u_{j+1\,k} + u_{j-1\,k} + 2u_{jk+1}\right) & \text{when } k = 0. \end{cases}$$

Show that if

$$L_{jk}u_{jk} \leq 0 \quad \text{on } G_R^0 \cup \partial G_{R_N},$$

then the maximum value of u_{jk} on $\mathbf{G}_R$ is attained on ∂G_{R_D}.
(b) Define L_{jk}, $j,k = 1, \ldots, M-1$, as

$$L_{jk}u_{jk} = \begin{cases} -\frac{1}{\Delta x^2}\left(\delta_x^2 + \delta_y^2\right)u_{jk} \text{when } j = 1,\ldots,M-1,\ k = 2,\ldots,M-1 \\ -\frac{1}{\Delta x^2}\left(-3u_{jk} + u_{j+1\,k} + u_{j-1\,k} + u_{jk+1}\right) \quad \text{when } k = 1. \end{cases}$$

Show that if $u_{j1} - u_{j0} = 0$, $j = 1, \ldots, M-1$, and

$$L_{jk}u_{jk} \leq 0 \quad \text{on } G_R^0,$$

then the maximum value of u_{jk} on $\mathbf{G}_R$ is attained on ∂G_{R_D}.

HW 10.8.5 Use the discrete separation of variables technique to determine the eigenvalues of the Jacobi iteration matrix for solving the discrete mixture problem (10.8.7). Use these eigenvalues to determine the optimal overrelaxation parameter ω_b.

10.8.3 Mixed Problems: Implementation

When we implemented solution schemes for approximating the solutions to Dirichlet problems in Section 10.5.5 and solution schemes for approximating the solutions to problems with Neumann boundary conditions in Section 10.7.6, we suggested the use of a stencil array that contained the information on how the difference equation reached to its neighbors. In Section 10.5.5, we set up the stencil arrays as $S(j,k,m)$, for $j = 0,\ldots,M_x$ and $k = 1,\ldots,M_y$, and used the arrays only for $j = 1,\ldots,M_x - 1$ and $k = 1,\ldots,M_y - 1$ (and solved only on the interior of the region). We promised at that time that the use of the stencil arrays would make later calculations easier. In Section 10.7.6, for first order approximations we used the same stencil arrays, we still used the arrays only from $j = 1,\ldots,M_x - 1$ and $k = 1,\ldots,M_y - 1$, we solved in the interior and defined the stencil values near the boundaries differently than we did for Dirichlet problems to reflect the use of our first order approximation to the Neumann boundary conditions (and included the nonhomogeneous term on the right hand side). In Section 10.7.6, for second order approximations we used the entire stencil arrays (and solved on the boundaries), this time defining the stencils on the boundary, taking into account that our equation on the boundary is given by difference equation (10.6.21) and boundary conditions (10.6.22), (10.6.23), (10.6.24), or (10.6.25). The stencil on the interior points is defined just as we did for Dirichlet problems.

The implementation of the mixture problems follows from making the observation that we can use all of the above methods at the same time (if possible at different points). At points that neighbor a Dirichlet boundary condition, we do as we did in Section 10.5.5. At points that neighbor a Neumann boundary condition that we want to approximate with a first order approximation, we do as we did in Section 10.7.6 for the first order approximations. For points on the boundary where we have a Neumann boundary condition that we want to approximate with a second order approximation, we treat the points as we did in Section 10.7.6 for the second order approximations. When we are done, we have a system of equations that represents the difference equation along with the appropriate boundary conditions, all expressed in terms of the same stencil array we have used previously. For a good example of the implementation of a mixture of Dirichlet and Neumann boundary conditions, see the $k = 0$ boundary in Example 10.11.1.

The implementation of the methods for solving the discrete Robin problems is very similar to the methods for solving problems involving Neumann boundary conditions. For the first order approximation of the Robin boundary condition, we fill the stencil for $j,k = 1,\ldots,M-1$. At $j = 1$, $j = M-1$, $k = 1$, and $k = M-1$, we must adjust the stencil so that the equation does not reach to the boundary (for example, when $k = 1$, we set $S(j,1,4) = 0$), adjust the stencil to take into account the boundary terms (for example,

when $k = 1$, we set $S(j,1,0) = 2/\Delta x^2 + 2/\Delta y^2 - \beta/(\Delta y^2(\beta + \alpha\Delta y))$, and add the appropriate terms to the right hand side. Remember that there is a double adjustment at the corners. We do the same thing when we implement the scheme for solving the Robin problem with the second order approximation of the Robin boundary condition, i.e., fill the stencil for $j,k = 0,\ldots,M$ with boundary adjustments in $S(j,k,m)$ and on the right hand side when $j = 0$, $j = M$, $k = 0$, and $k = M$. Again remember that there are double adjustments at $(0,0)$, $(M,0)$, $(0,M)$, and (M,M).

At this time, we would generally solve the matrix equations using a residual correction scheme. We will not analyze the convergence of the different relaxation schemes here, but we should be aware that this analysis could be or should be done is some circumstances. By now it should be clear that it would be possible to compute the eigenvalues of the iteration matrix for the various solution methods for nice mixture problems (including the model mixture problem). It is probably impossible to compute the eigenvalues of the iteration matrices for solving the Robin problems. We recall that when we consider the analytic problem and try solving the problem by separation of variables, we are unable to compute the eigenvalues necessary for the solution. When we solve the computational problems given at the end of this section, we must proceed carefully—as a computational experimentalist.

HW 10.8.6 (a) Use the Gauss-Seidel scheme to find an approximate solution to the model mixture problem (10.8.2)–(10.8.6) with $F(x,y) = \sin\pi x \sin 2\pi y$ for $(x,y) \in [0,1] \times [0,1]$; $f_2(y) = f_4(y) = 0$, $y \in [0,1]$; $f_3(x) = 0$, $x \in [0,1]$; and $g(x) = 0$, $x \in [0,1]$. Use $M = 20$ and both the difference of successive iterates and the residual as a stopping criterion.
(b) Repeat part (a) using $M = 100$.
(c) Repeat part (a) using $M = 100$ and the optimal SOR scheme (use the optimal parameter found in HW10.8.5.

HW 10.8.7 Consider the mixture problem

$$
\begin{aligned}
-\nabla^2 v &= 1 - \left|x - \frac{1}{2}\right|\left|y - \frac{1}{2}\right|, \quad (x,y) \in (0,1) \times (0,1) \\
v(1,y) &= 0, \quad y \in [0,1] \\
-\frac{\partial v}{\partial y}(x,1) &= \sin\pi x, \quad x \in [0,1] \\
-\frac{\partial v}{\partial x}(0,y) &= \sin 2\pi y, \quad y \in [0,1] \\
-\frac{\partial v}{\partial y}(x,0) &= \sin\pi x, \quad x \in [0,1].
\end{aligned}
$$

(a) Find an approximate solution to the above problem using $M = 100$, the Gauss-Seidel scheme, and both the difference between successive iterates and the residual as the stopping criteria.

(b) Repeat part (a) using the optimal SOR scheme (by computing ω_b analytically or by the results in Section 10.5.12).

HW 10.8.8 (a) Find an approximate solution to the model Robin problem with $F(x,y) = \sin 2\pi x \sin 2\pi y$, $(x,y) \in R = [0,1] \times [0,1]$; $\alpha = 1.0$, $\beta = 1.0$, and $g = 0$ on ∂R (partial differential equation (10.8.2); and Robin boundary condition (10.8.1)). Use $M = 50$, the Gauss-Seidel scheme, and the difference between successive iterates as a stopping criterion.
(b) Repeat the problem in part (a) using optimal SOR (where you must use the approach given in Section 10.5.12 to approximate the optimal over-relaxation parameter ω_b).

HW 10.8.9 Find an approximate solution to the model Robin problem with $F(x,y) = \sin 2\pi x \sin 2\pi y$, $(x,y) \in R = [0,1] \times [0,1]$; $\alpha = 1.0$, $\beta = -1.0$, and $g = 0$ on ∂R. Use $M = 50$, the Gauss-Seidel scheme, and the difference between successive iterates as a stopping criterion.

10.9 Elliptic Difference Equations: Polar Coordinates

One setting for elliptic partial differential equations that is too often ignored in the numerical literature is that in which an elliptic partial differential equation is expressed with respect to polar or cylindrical coordinates. Since there are many circles (and cylinders) in life, it is important to be able to solve partial differential equations that are written in polar coordinates. In this section we include a brief discussion of the approximate solution of elliptic partial differential equations written in polar coordinates.

We will base our discussion on two model problems: the annulus model problem

$$-\nabla^2 v = F(r,\theta), \quad R_a = \{(r,\theta) : R_i < r < 1, \quad 0 \le \theta < 2\pi\} \tag{10.9.1}$$

$$v(R_i,\theta) = f_1(\theta), \quad \theta \in [0,2\pi] \tag{10.9.2}$$

$$v(1,\theta) = f_2(\theta), \quad \theta \in [0,2\pi] \tag{10.9.3}$$

and the disk model problem

$$-\nabla^2 v = F(r,\theta), \quad R_d = \{(r,\theta) : 0 \le r < 1, \quad 0 \le \theta < 2\pi\} \tag{10.9.4}$$

$$v(1,\theta) = f_2(\theta), \quad \theta \in [0,2\pi] \tag{10.9.5}$$

where ∇^2 is expressed in polar coordinates as

$$\nabla^2 v = \frac{1}{r}(r v_r)_r + \frac{1}{r^2} v_{\theta\theta}. \tag{10.9.6}$$

Obviously, the two model problems are problems in polar coordinates without the origin and with the origin, respectively.

We notice also that we have given Dirichlet boundary conditions on the boundary $r = R_i$ and $r = 1$. We could surely consider Neumann or mixed boundary conditions. The methods that we discuss are easily extended to Neumann and mixed boundary value problems.

We first derive an approximation of boundary–value problems (10.9.1)–(10.9.3) and (10.9.4)–(10.9.5). In Sections 4.5 and 6.10.3, Part 1, we describe grids on the regions R_a and R_d, and discuss the discretization of a two dimensional parabolic equation given in polar coordinates. We will try not to repeat much of the work done in Sections 4.5 and 6.10.3. We place a grid (r_j, θ_k), $j = 0, \ldots, M_r$, $k = 0, \ldots, M_\theta$. When we are considering boundary–value problem (10.9.1)–(10.9.3), we have $r_0 = R_i$, $r_{M_r} = 1$, $\theta_0 = 0$, and $\theta_{M_\theta} = 2\pi$. When we are considering boundary value problem (10.9.4)–(10.9.5), we have $r_0 = 0$, $r_{M_r} = 1$, $\theta_0 = 0$ and $\theta_{M_\theta} = 2\pi$. If we return to equations (4.5.13)–(4.5.17) and realize that we want the steady state version of this set of difference equations, we see that we can approximate boundary–value problem (10.9.1)–(10.9.3) by the following set of difference equations

$$-\frac{1}{r_j \Delta r^2}\left[r_{j+1/2}\left(u_{j+1\,k} - u_{j\,k}\right) - r_{j-1/2}\left(u_{j\,k} - u_{j-1\,k}\right)\right] - \frac{1}{r_j^2 \Delta\theta^2}\delta_\theta^2 u_{j\,k}$$
$$= F_{j\,k}, \quad j = 1, \ldots, M_r - 1, \ k = 1, \ldots, M_\theta - 1 \tag{10.9.7}$$

$$-\frac{1}{r_j \Delta r^2}\left[r_{j+1/2}\left(u_{j+1\,0} - u_{j\,0}\right) - r_{j-1/2}\left(u_{j\,0} - u_{j-1\,0}\right)\right]$$
$$- \frac{1}{r_j^2 \Delta\theta^2}\left(u_{j\,1} - 2u_{j\,0} + u_{j\,M_\theta - 1}\right) = F_{j\,0},$$
$$j = 1, \ldots, M_r - 1 \tag{10.9.8}$$

$$u_{j\,M_\theta} = u_{j\,0}, \quad j = 0, \ldots, M_r \tag{10.9.9}$$

$$u_{0\,k} = f_1(\theta_k), \quad k = 0, \ldots, M_\theta \tag{10.9.10}$$

$$u_{M_r\,k} = f_2(\theta_k), \quad k = 0, \ldots, M_\theta. \tag{10.9.11}$$

It should be clear that equation (10.9.7) is the same as equation (10.9.8) except that we have used the fact that $k = 0$ and $k = M_\theta$ represent the same angle, so when we reach for $k = -1$, we get $k = M_\theta - 1$. The $k = 0$ and $k = M_\theta$ boundaries are treated like a periodic boundary, and the periodic relationship is given by equation (10.9.9).

Equations (4.5.13)–(4.5.17) also help us obtain the approximation to boundary–value problem (10.9.4)–(10.9.5). We understand that when we are considering the case with the origin, the set of grid points associated with $j = 0$, $k = 0, \ldots, M_\theta$ is really only one point. We will use two different notations to denote functions defined at $j = 0$. When it most convenient to use the usual notation, we will denote the approximation of v at the origin by $u_{0\,k}$ for any k, realizing that for the different k values, there is only one

function value. When we want to emphasize that there is only one point associated with $j = 0$, we will denote the approximation of v at the origin by u_0. We get

$$-\frac{1}{r_j\Delta r^2}\left[r_{j+1/2}(u_{j+1\,k}-u_{j\,k})-r_{j-1/2}(u_{j\,k}-u_{j-1\,k})\right]-\frac{1}{r_j^2\Delta\theta^2}\delta_\theta^2 u_{j\,k}$$
$$=F_{j\,k},\quad j=1,\ldots,M_r-1,\ k=1,\ldots,M_\theta-1 \tag{10.9.12}$$

$$-\frac{1}{r_j\Delta r^2}\left[r_{j+1/2}(u_{j+1\,0}-u_{j\,0})-r_{j-1/2}(u_{j\,0}-u_{j-1\,0})\right]$$
$$-\frac{1}{r_j^2\Delta\theta^2}(u_{j\,1}-2u_{j\,0}+u_{j\,M_\theta-1})=F_{j\,0},$$
$$j=1,\ldots,M_r-1 \tag{10.9.13}$$

$$u_{j\,M_\theta}=u_{j\,0},\quad j=0,\ldots,M_r \tag{10.9.14}$$

$$\frac{4}{\Delta r^2}u_0-\frac{2\Delta\theta}{\pi\Delta r^2}\sum_{k=0}^{M_\theta-1}u_{1\,k}=F_0 \tag{10.9.15}$$

$$u_{M_r\,k}=f_2(\theta_k),\quad k=0,\ldots,M_\theta. \tag{10.9.16}$$

Of course, equation (10.9.15) represents the equation centered at $k = 0$. The derivation of equation (10.9.14) is done using a control volume approach the same way that we derived equation (4.5.12). We write difference equations (10.9.12)–(10.9.16) in the form

$$A\mathbf{u}=\mathbf{f}, \tag{10.9.17}$$

where

$$\mathbf{u}=[u_0\ u_{1\,0}\ u_{1\,1}\ \cdots\ u_{1\,M_\theta-1}\ u_{2\,0}\ \cdots\ u_{M_r-1\,M_\theta-1}]^T,$$

$$A=\begin{pmatrix}
\alpha & \mathbf{r}^T & \boldsymbol{\theta}^T & \cdots & & & \\
\mathbf{c} & T_1 & \gamma_1 I & \Theta & \cdots & & \\
\boldsymbol{\theta} & -\alpha_2 I & T_2 & -\gamma_2 I & \Theta & \cdots & \\
\boldsymbol{\theta} & \Theta & -\alpha_3 I & T_3 & -\gamma_3 I & \Theta & \cdots \\
\vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \\
\boldsymbol{\theta} & \Theta & \cdots & \Theta & -\alpha_{M_r-2} I & T_{M_r-2} & -\gamma_{M_r-2} I \\
\boldsymbol{\theta} & \Theta & \cdots & \cdots & \Theta & -\alpha_{M_r-1} I & T_{M_r-1}
\end{pmatrix}, \tag{10.9.18}$$

$\alpha = 4/\Delta r^2$, $\mathbf{r}^T = [a\ \cdots\ a]$ ($\mathbf{r}$ an M_θ-vector), $a = -2\Delta\theta/(\pi\Delta r^2)$, $\mathbf{c} = -[\alpha_1\ \cdots\ \alpha_1]^T$ ($\mathbf{c}$ an M_θ-vector),

$$\alpha_j=r_{j-1/2}/(r_j\Delta r^2),$$
$$\gamma_j=r_{j+1/2}/(r_j\Delta r^2),$$

$$
\begin{aligned}
\beta_j &= ((r_{j-1/2} + r_{j+1/2})/(r_j \Delta r^2)) + (2/(r_j^2 \Delta\theta^2)), \\
\epsilon_j &= 1/(r_j^2 \Delta\theta^2), \\
T_j &= \begin{pmatrix}
\beta_j & -\epsilon_j & 0 & \cdots & 0 & -\epsilon_j \\
-\epsilon_j & \beta_j & -\epsilon_j & 0 & \cdots & \\
0 & -\epsilon_j & \beta_j & -\epsilon_j & 0 & \cdots \\
\cdots & \cdots & \ddots & \ddots & \ddots & \cdots \\
0 & \cdots & 0 & -\epsilon_j & \beta_j & -\epsilon_j \\
-\epsilon_j & 0 & \cdots & 0 & -\epsilon_j & \beta_j
\end{pmatrix}, \qquad (10.9.19) \\
\mathbf{f} &= \mathbf{F} + \mathbf{b}_r, \\
\mathbf{F} &= [F_0 \; F_{1\,0} \; F_{1\,1} \; \cdots \; F_{1\,M_\theta - 1} \; F_{2\,0} \; \cdots \; F_{M_r - 1\,M_\theta - 1}]^T,
\end{aligned}
$$

and

$$
\mathbf{b}_x = \frac{r_{M_r - 1/2}}{r_{M_r - 1} \Delta r^2} [0 \; \cdots \; 0 \; f_{2_0} \; \cdots \; f_{2_{M_\theta - 1}}]^T.
$$

Of course, we could write the system of difference equations associated with the annulus problem, system (10.9.7)–(10.9.11), in matrix form also. The matrix representation of (10.9.7)–(10.9.11) is a part of the above representation and is left to the reader in HW10.9.5.

We see that the matrix problem we must solve for an elliptic equation on a region that includes the origin is like almost nothing we have seen before, except for the implicit scheme for approximating the solution to the parabolic equation given in polar coordinates. We should understand that the first row and first column are "different" because the $(0, k)$ points, $k = 0, \ldots, M_\theta - 1$, are really only one point and that one point is close to all of the points $(1, k)$, $k = 0, \ldots, M_\theta - 1$. The annulus problem will not have the strange first row and first column and will not cause us as much trouble to solve.

Though the matrix for the disk model problem does not look nice, we should still try to show that the discrete problem is uniquely solvable. It is easy to see that for all of the rows except for the first row and those that reach the outer boundary,

$$
\begin{aligned}
|a_{j\,j}| &= \frac{r_{j-1/2} + r_{j+1/2}}{r_j \Delta r^2} + \frac{2}{r_j^2 \Delta\theta^2} \\
&= \frac{r_j - \Delta r/2 + r_j + \Delta r/2}{r_j \Delta r^2} + \frac{2}{r_j^2 \Delta\theta^2} \\
&= \frac{2}{\Delta r^2} + \frac{2}{r_j^2 \Delta\theta^2}
\end{aligned}
$$

and

$$\begin{aligned}\sum_{\substack{k=1\\k\neq j}}^{M_\theta(M_r-1)+1} |a_{jk}| &= \alpha_j + 2\epsilon_j + \gamma_j \\ &= \frac{r_{j-1/2}}{r_j\Delta r^2} + \frac{2}{r_j^2\Delta\theta^2} + \frac{r_{j+1/2}}{r_j\Delta r^2} \\ &= \frac{r_j - \Delta r/2}{r_j\Delta r^2} + \frac{2}{r_j^2\Delta\theta^2} + \frac{r_j + \Delta r/2}{r_j\Delta r^2} \\ &= \frac{2}{r_j^2\Delta\theta^2} + \frac{2}{\Delta r^2}.\end{aligned}$$

Hence, these rows satisfy the diagonal dominance inequality. The only difference between the above calculation and the calculation that would be done for those rows that reach the outer boundary is that one of the off diagonal terms is missing. For this reason, the rows that reach to the outer boundary satisfy the strict diagonal dominance inequality. For the first row, we have $|a_{00}| = 4/\Delta r^2$ and

$$\sum_{\substack{k=1\\k\neq j}}^{M_\theta(M_r-1)+1} |a_{0k}| = M_\theta\frac{2\Delta\theta}{\pi\Delta r^2} = \frac{2M_\theta\Delta\theta}{\pi\Delta r^2} = \frac{4}{\Delta r^2}.$$

Hence, the first row also satisfies the diagonal dominance inequality. Thus we see that by Proposition 10.2.5, discrete problem (10.9.12)–(10.9.16) is uniquely solvable. It should be reasonably clear that these computations also imply that discrete problem (10.9.7)–(10.9.11) is uniquely solvable.

The next step is to solve the discrete problems. These are not problems for which we can develop a lot of theory, but it is easy to see that any of the relaxation schemes are good candidates for solving discrete problem (10.9.7)–(10.9.11). We might emphasize that we have no way of explicitly calculating the optimal relaxation parameter for the SOR scheme. The setting in which we want to use optimal SOR to solve the discrete problem associated with an elliptic boundary–value problem expressed in terms of polar coordinates is a good time to use and test the schemes developed in Section 10.5.12 for finding an approximation of ω_b.

The most obvious way to solve discrete problem (10.9.12)–(10.9.16) is to return to Section 6.10, Part 1, and use the Sherman-Morrison Algorithm. We write A as

$$A = \begin{pmatrix} \alpha & \mathbf{R}^T \\ \mathbf{C} & A' \end{pmatrix}$$

where $\mathbf{R}^T = [\mathbf{r}^T\ 0\cdots\ 0]$, $\mathbf{C} = [\mathbf{c}^T\ 0\ \cdots\ 0]^T$ and A' consists of the second through $\big(M_\theta(M_r-1)+1\big)$-st rows and columns of A. If we partition $\mathbf{f}$ as $\mathbf{f}^T = [f_1\ \mathbf{f}_2^T]^T$ and $\mathbf{u}^T$ as $\mathbf{u}^T = [u_1\ \mathbf{u}_2^T]$, we can write the algorithm given in

Section 6.10 for finding the solution to equation (10.9.17) as

$$A'\mathbf{y}_1 = \mathbf{f}_2 - \frac{f_1}{\alpha}\mathbf{C} \tag{10.9.20}$$

$$A'\mathbf{y}_2 = \frac{1}{\alpha}\mathbf{C} \tag{10.9.21}$$

$$\beta = \frac{\mathbf{R}^T \cdot \mathbf{y}_1}{1 - \mathbf{R}^T \cdot \mathbf{y}_2} \tag{10.9.22}$$

$$\mathbf{u}_2 = \mathbf{y}_1 + \beta\mathbf{y}_2 \tag{10.9.23}$$

$$\mathbf{u}_1 = \frac{1}{\alpha}(f_1 - \mathbf{R}^T \cdot \mathbf{u}_2). \tag{10.9.24}$$

We see that solving equation (10.9.17) by the Sherman-Morrison Algorithm entails the solution of two $\big(M_\theta(M_r - 1) + 1\big) \times \big(M_\theta(M_r - 1) + 1\big)$ matrix problems, (10.9.20) and (10.9.21). The rest of the extra work involved is probably insignificant in comparison to the costs of solving these two systems. Because at this time we would generally solve these equations by iterative methods, there is not much savings from trying to solve the two systems together. The easiest approach is to make the previous Jacobi, Gauss-Seidel, and SOR schemes into subroutines and call them twice. We can and should take advantage of the fact that we use the same stencil for both solutions.

Remark: We should understand that there is a chance that we can apply an iterative scheme directly to equation (10.9.17), i.e., not use the Sherman-Morrison algorithm. See HW10.9.8.

HW 10.9.1 Write the system of difference equations (10.9.7)–(10.9.11) in matrix form

$$A\mathbf{u} = \mathbf{f}.$$

HW 10.9.2 Write the matrix form of difference scheme (10.9.12)–(10.9.16) for the case where $M_r = 5$ and $M_\theta = 6$.

HW 10.9.3 Write the matrix form of difference scheme (10.9.7)–(10.9.11) for the case where $M_r = 5$ and $M_\theta = 6$.

HW 10.9.4 Consider the annulus problem (10.9.1)–(10.9.3) with $R_i = 0.5$, $F(r, \theta) = 0$ for all $(r, \theta) \in R_a$, $f_1(\theta) = \sin 2\theta$, $f_2(\theta) = \sin 3\theta$, $\theta \in [0, 2\pi]$.
(a) Use difference scheme (10.9.7)–(10.9.11) to obtain an approximate solution to the above problem. Use $M_r = 20$, $M_\theta = 20$, and the Gauss-Seidel solution scheme using both the difference of successive iterates and the residual as a stopping criterion.
(b) Repeat part (a) with $M_r = 100$ and $M_\theta = 100$.
(c) Repeat part (a) using optimal SOR.

HW 10.9.5 Find an approximation to the solution of the annulus problem (10.9.1)-(10.9.3) with $R_i = 0.5$, $F(r,\theta) = \exp(r)\sin 2\pi\theta$ for $(r,\theta) \in R_a$, $f_1(\theta) = \sin 4\theta$, $f_2(\theta) = \sin 3\theta$, $\theta \in [0, 2\pi]$. Use difference scheme (10.9.7)-(10.9.11), $M_r = 100$, $M_\theta = 100$, and the optimal SOR scheme with both the difference of successive iterates and the residual as a stopping criterion.

HW 10.9.6 (a) Find an approximate solution of the disk problem (10.9.4)-(10.9.5) with $F(r,\theta) = 0$ for all $(r,\theta) \in R_d$ and $f_2(\theta) = \sin 2\theta$, $\theta \in [0, 2\pi]$. Use difference scheme (10.9.12)-(10.9.16), $M_r = 20$, $M_\theta = 20$, and the Jacobi scheme with the residual as a stopping criterion.

(b) Repeat the solution in part (a) using the optimal SOR scheme with the difference of successive iterates as a stopping criterion.

HW 10.9.7 Find an approximate solution of the disk problem (10.9.4)-(10.9.5) with $F(r,\theta) = \cos \pi r \cos 2\pi\theta$ for $(r,\theta) \in R_d$ and $f_2(\theta) = \sin 4\theta$, $\theta \in [0, 2\pi]$. Use difference scheme (10.9.12)-(10.9.16), $M_r = 100$, $M_\theta = 100$, and optimal SOR with the difference between successive iterates and the residual as a stopping criterion.

HW 10.9.8 Resolve the problem given in HW10.9.7 using the Gauss-Seidel scheme directly on equation (10.9.17).

10.10 Multigrid

10.10.1 Introduction

In Section 10.5.1 we mentioned that in this section we would take advantage of the fact that the components of the error associated with the small eigenvalues were eliminated quickly, and most of the work we did with residual correction schemes was to eliminate the components of the error associated with the large eigenvalues. The basic idea behind multigrid is to eliminate the high frequency components of the error quickly on a fine grid. To accomplish this, the high frequency components of the error will have to correspond to the smallest eigenvalues of the iteration matrix. We then transfer the problem to a coarser grid where high frequency components of the error correspond to some of the lower frequency errors on the previous grid. We can then eliminate these high frequency components of the error on this coarse grid quickly. This process is repeated on yet coarser grids, and the result is finally transferred back to the fine grid. The savings in computational costs are due to both the fact that we are eliminating the errors quickly on the appropriate grid and the fact that the coarse grids

are cheaper to work on. Maybe the most amazing statement about the procedure described above is that it works, and it works well.

In this section we will give an introduction to multigrid. We will include one multigrid algorithm and try to use some computations along with some graphics to illustrate how and why the procedure described above works. For a more in-depth description of the multigrid algorithm and theory, see refs. [8], [20] and [43].

Model Problem

As a part of our discussion, we will return to model problem (10.2.3)–(10.2.7). For convenience, we will set $\Delta x = \Delta y$ and choose $M_x = M_y = M$ so that $M = 2^p$. Obviously, the latter condition restricts our choice of M. It is not a requirement, but as we will see, it is a very convenient assumption. We will consider the model problem both as given in (10.2.3)–(10.2.7) and as the matrix problem $A\mathbf{u} = \mathbf{f}$ as described in equation (10.4.1).

Model Computational Problem

When we use computations to illustrate different aspects of the multigrid scheme, we will use a special case of model problem (10.2.3)–(10.2.7) where we set $f_{jk} = 0$ and $F_{jk} = 0$ for all j and k, and choose $M = 2^4 = 16$. Obviously, the homogeneous difference equation has the unique zero solution and is the approximation to the homogeneous boundary–value problem (10.2.1)–(10.2.2) that will have the unique zero solution. Hence, we see that in our computations, we do not have to be concerned with truncation error. We will concentrate on the convergence of the multigrid scheme. So as to illustrate our point clearly, we will use an initial guess of

$$\mathbf{u}_0 = \sum_{s=1}^{15}\sum_{p=1}^{15} \mathbf{w}^{p\,s} \tag{10.10.1}$$

where $\mathbf{w}^{p\,s}$, $p, s = 1, \ldots, 15$ are the eigenvectors of A. We note that by using such an initial guess, all eigencomponents of the error are present at a significant level and must be eliminated. The initial error is given by $\mathbf{e}_0 = -\mathbf{u}_0$, but for convenience, when we are discussing the error, we will often plot $-\mathbf{e}_0 = \mathbf{u}_0$. Recall from (10.5.29) that the eigenvalues of A are given by

$$\mu_{p\,s} = \frac{2}{\Delta x^2}\left(2 - \cos\frac{p\pi}{M} - \cos\frac{s\pi}{M}\right) \tag{10.10.2}$$

$$= \frac{4}{\Delta x^2}\left(\sin^2\frac{p\pi}{2M} + \sin^2\frac{s\pi}{2M}\right), \tag{10.10.3}$$

and the components of the associated eigenvectors are given by

$$w_{j\,k}^{p\,s} = \sin\frac{jp\pi}{M}\sin\frac{ks\pi}{M}, \quad j, k = 1, \ldots, 15, \quad p, s = 1, \ldots, 15. \tag{10.10.4}$$

We note that these eigenvectors are orthogonal (this makes some of our test computations a bit easier). *We emphasize that from the computational point of view, initial guess (10.10.1) is a terrible initial guess. A little thought (and/or a plot of* u_0 *or its analytic analogue) will reveal that this function is a terrible approximation to the solution of our model problem. We use this initial guess for the convenience of having all of the modes present.*

We notice that the first and last eigenvectors are given by

$$\mathbf{w}^{1\,1} = \Big[\sin\frac{\pi}{16}\sin\frac{\pi}{16}\ \sin\frac{\pi}{16}\sin\frac{2\pi}{16}\ \cdots\ \sin\frac{\pi}{16}\sin\frac{15\pi}{16}\ \sin\frac{2\pi}{16}\sin\frac{\pi}{16} \cdots\ \sin\frac{15\pi}{16}\sin\frac{15\pi}{16}\Big]^T \tag{10.10.5}$$

and

$$\mathbf{w}^{15\,15} = \Big[\sin\frac{15\pi}{16}\sin\frac{15\pi}{16}\ \sin\frac{15\pi}{16}\sin\frac{2\cdot 15\pi}{16}\ \cdots\ \sin\frac{15\pi}{16}\sin\frac{15\cdot 15\pi}{16} \sin\frac{2\cdot 15\pi}{16}\sin\frac{15\pi}{16}\ \cdots\ \sin\frac{15\cdot 15\pi}{16}\sin\frac{15\cdot 15\pi}{16}\Big]^T, \tag{10.10.6}$$

respectively. We should understand that eigenvectors (10.10.5) and (10.10.6) are analogues of the functions $\sin\pi x\sin\pi y$ and $\sin 15\pi x\sin 15\pi y$, respectively. All of the other eigenvectors can be written similarly, and they all have analytic analogues. Inspecting the vectors $\mathbf{w}^{1\,1}$ and $\mathbf{w}^{15\,15}$ carefully (it may be easier to look at the analytic analogues) shows that $\mathbf{w}^{1\,1}$ is slowly varying and $\mathbf{w}^{15\,15}$ is highly oscillatory. The pattern seen here is generally the case for all of the eigenvectors: The eigenvectors $\mathbf{w}^{p\,s}$ where p and s are small change slowly, whereas when p or q are large, the eigenvector is oscillatory. We will generally refer to the slowly varying eigenvectors as **smooth** or **low frequency** vectors and to the highly oscillatory vectors as **oscillatory** or **high frequency** vectors.

If we were to consider a grid with $M_x = M_y = M/2$ ($M_x = M_y = 8$ in our example), we see that the highest frequency eigenvector is given by

$$\mathbf{w}^{7\,7} = \Big[\sin\frac{7\pi}{8}\sin\frac{7\pi}{8}\ \sin\frac{7\pi}{8}\sin\frac{2\cdot 7\pi}{8}\ \cdots\ \sin\frac{7\pi}{8}\sin\frac{7\cdot 7\pi}{8} \sin\frac{2\cdot 7\pi}{8}\sin\frac{7\pi}{8}\ \cdots\ \sin\frac{7\cdot 7\pi}{8}\sin\frac{7\cdot 7\pi}{8}\Big]^T, \tag{10.10.7}$$

i.e., analogous to the analytic function $\sin 7\pi x\sin 7\pi y$. It should be clear that the frequency of the highest frequency eigenvector on the $M/2$-grid is a mid-range frequency on the M-grid. The difference between these "high frequencies" on the two grids is what we will exploit in deriving the multigrid iterative scheme.

Model One Dimensional Problem

There are some aspects of multigrid that we felt are just too gross or impossible to illustrate with two dimensional problems. Hence, we will occa-

sionally consider the one dimensional problem analogous to our two dimensional model computational problem. We will consider the boundary–value problem

$$-v'' = 0, \quad x \in (0,1) \tag{10.10.8}$$

$$v(0) = v(1) = 0, \tag{10.10.9}$$

along with the difference equation approximation to boundary–value problem (10.10.8)–(10.10.9).

$$-\frac{1}{\Delta x^2}u_{k-1} + \frac{2}{\Delta x^2}u_k - \frac{1}{\Delta x^2}u_{k+1} = 0, \quad k = 1, \ldots, M-1 \tag{10.10.10}$$

$$u_0 = u_M = 0. \tag{10.10.11}$$

When we use this example, we will either use a general value of M or set $M = 8$.

If we return to our description of the multigrid algorithm given in the first paragraph of this section, it should be reasonably clear that what we must do is

(i) develop an iterative scheme so that on any given grid, the scheme will eliminate the components of the error associated with the high frequency modes, and

(ii) develop grid transfers so that the (p,s) component of the error on the M-grid is transferred to the (p,s) component of the grid on the $M/2$-grid, and the results can be passed from the $M/2$-grid back to the M-grid without introducing new error.

As we shall see, the first step is relatively easy (because we should know quite a bit about iterative schemes by this time), and the second step is not so easy. We are able, however, to complete the second step sufficiently well to develop a very useful multigrid iterative scheme.

10.10.2 Smoothers

We begin by mentioning that we title this section "smoothers" because that is what we really want out of our iterative scheme. We want the iterative scheme to smoothen the error function, i.e., eliminate the oscillatory modes. We begin with an obvious approach by trying some of our favorite schemes. We consider our model computational problem with the initial guess given by $\mathbf{u}_0$, (10.10.1). We should realize that if we were to plot the coefficients of the initial error expanded with respect to the eigenvalues of A, by our definition of $\mathbf{u}_0$ we would get a constant "one" function (actually a minus one, but we have eliminated the minus sign for convenience).

In Figure 10.10.1 we plot the coefficients of the error for our computational model problem expanded with respect to the eigenvectors $\mathbf{w}^{p\,s}$, $p, s = 1, \ldots, 15$. This plot represents the error after two Jacobi steps with

$\mathbf{u}_0$, (10.10.1), as our initial guess. It should be pretty clear that the Jacobi scheme is eliminating the middle frequency components, not the high frequency components. This should not surprise us, since the eigenvalues of the Jacobi iteration matrix R_J are

$$\lambda_s^p = \frac{1}{2}\Big(\cos\frac{p\pi}{M} + \cos\frac{s\pi}{M}\Big)$$

and the associated eigenvectors are $\mathbf{w}^{p\,s}$, $p, s = 1, \ldots, M$. See example (10.5.1). The error components associated with the smallest eigenvalues of the iteration matrix get smashed down. The smallest eigenvalues of the Jacobi iteration matrix are associated with the mid-range values of p and s. Hence, *the oscillatory components of the error do not get eliminated by the Jacobi iteration scheme.*

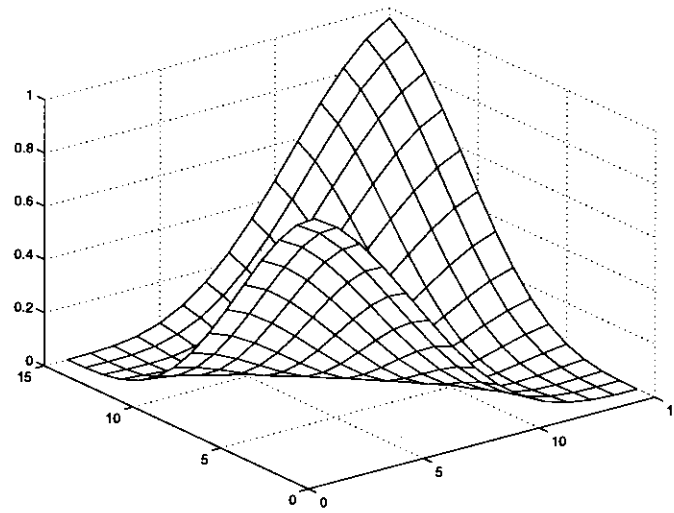

FIGURE 10.10.1. Plot of the coefficients of the error after two Jacobi steps with $\mathbf{u}_0$, (10.10.1), as the starting guess.

Before we proceed with our next candidate, it might be helpful to make it clear how we generate the plot given in Figure 10.10.1. We begin by using a Jacobi scheme where we set *uold* equal to the value defined by $\mathbf{u}_0$ in (10.10.1). After we have performed two Jacobi iterations, we have $\mathbf{u}_2$ and want $a_{p\,s}$, $p, s = 1, \ldots, 15$, such that

$$-\mathbf{e}_2 = \mathbf{u}_2 = \sum_{s=1}^{15}\sum_{p=1}^{15} a_{p\,s}\mathbf{w}^{p\,s}.$$

This is not difficult, since the vectors $\mathbf{w}^{p\,s}$ are orthogonal, i.e., we take the

dot product of both sides with $\mathbf{w}^{p_0 s_0}$ and get

$$a_{p_0 s_0} = \frac{\mathbf{u}_2 \cdot \mathbf{w}^{p_0 s_0}}{\mathbf{w}^{p_0 s_0} \cdot \mathbf{w}^{p_0 s_0}}.$$

Figure 10.10.1 is a plot of the function a_{ps}, for $p, s = 1, \ldots, 15$.

In HW10.5.16 we introduced the weighted Jacobi scheme. We introduced the weighted Jacobi scheme as the Jacobi analogue to the SOR scheme, i.e., the weighted Jacobi is to the Jacobi scheme as the SOR scheme is to the Gauss-Seidel scheme. In HW10.5.16 we found that we could not overrelax the Jacobi scheme. For convergence, we could only underrelax the scheme, i.e., the weighted Jacobi scheme converges for $0 < \omega \leq 1$. Probably at that time, there appeared to be no redeeming characteristics of the weighted Jacobi scheme. In Figure 10.10.2 we plot the coefficients of the error expanded with respect to the eigenvectors $\mathbf{w}^{ps}$, $p, s = 1, \ldots, 15$. The error plotted is after three weighted Jacobi steps with $\omega = \frac{2}{3}$, beginning again with initial guess $\mathbf{u}_0$, (10.10.1). We see in Figure 10.10.2 that *the weighted Jacobi scheme eliminates the high frequency components of the error.*

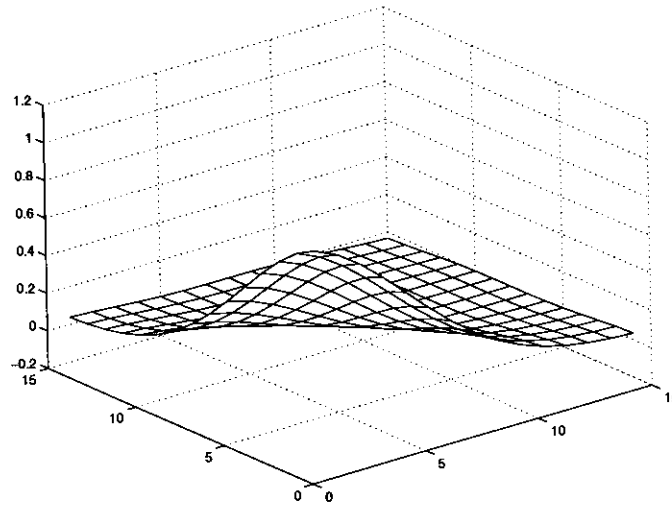

FIGURE 10.10.2. Plot of the coefficients of the error after three weighted Jacobi steps with $\mathbf{u}_0$, (10.10.1), as the starting guess.

It is not difficult to see why the weighted Jacobi scheme eliminates the high frequency components of the error. Using the results of HW10.5.16, we see that the eigenvalues of the iteration matrix for the weighted Jacobi

scheme are given by

$$\begin{aligned}\lambda_s^p &= 1-\omega+\omega\lambda_{J_s}^p \quad (\lambda_{J_s}^p \text{ is the eigenvalue of the Jacobi iteration matrix})\\ &= 1-\omega+\frac{\omega}{2}\Big(\cos\frac{p\pi}{M}+\cos\frac{s\pi}{M}\Big)\\ &= 1-\omega\Big(\sin^2\frac{p\pi}{2M}+\sin^2\frac{s\pi}{2M}\Big), \quad p,s=1,\ldots,M-1. \qquad (10.10.12)\end{aligned}$$

The eigenvalues get smaller as p and s get larger.

When we choose an iterative scheme to use with multigrid, we really want more than the scheme to eliminate some of the high frequency error modes. During the second step of multigrid, we will work on a grid with $M_x = M_y = M/2$. On this grid we will have only the modes analogous to the analytic modes $\sin\pi x \sin\pi y$, $\sin 2\pi x \sin\pi y, \ldots$, $\sin(M/2)\pi x \sin(M/2)\pi y$. For this reason, on the M-grid we would really like to eliminate all of the error associated with the modes from $M/2$ to $M-1$ (or at least eliminate most of the error associated with these modes). It is not difficult to see that the eigenvalues associated with the weighted Jacobi scheme get smaller as ω gets larger. Also, the eigenvalues decrease monotonically with respect to p and s, and for sufficiently large values of ω, the eigenvalues become negative for the larger values of p and s. Hence, we see that if we choose ω so that $\lambda_{M/2}^{M/2} = -\lambda_M^M$, we will damp the modes between $M/2$ and $M-1$ as much as is possible by using the weighted Jacobi scheme. In HW10.10.2 we see that $\lambda_{M/2}^{M/2} = -\lambda_M^M$ if $\omega = \frac{2}{3}$ (which, conveniently, is the value of ω that we used in the calculation given in Figure 10.10.2). We also see from HW10.10.2 that when $\omega = \frac{2}{3}$, then $|\lambda_s^p| \le \frac{1}{3}$ for $p,s = M/2,\ldots,M-1$. Hence, when we use $\omega = \frac{2}{3}$, each iteration reduces each high frequency component of the error (the frequencies between $M/2$ and $M-1$) by at least a multiple of $\frac{1}{3}$. In the above discussion and in HW10.10.2 we use λ_M^M. We should be clear that λ_M^M is not an eigenvalue. We use the logical extension of the notation and definition for the eigenvalues to give λ_M^M, which is a convenient bound.

We must realize that there are other iterative schemes that are effective smoothers. One very obvious choice to try is the Gauss-Seidel scheme. In Figure 10.10.3 we give the plot of the coefficients of the error after two Gauss-Seidel iterations. It is clear that *Gauss-Seidel also eliminates the high frequency components of the error*. If we were to look at the eigenvalues of the iteration matrix associated with the Gauss-Seidel scheme, it would not be clear that the Gauss-Seidel scheme would eliminate the high frequency components of the error. We saw in Example 10.5.2 that the eigenvalues of the Gauss-Seidel iteration matrix are given by

$$\lambda_s^p = \Big(\cos\frac{s\pi}{M}+\cos\frac{p\pi}{M}\Big)^2, \quad p,s=1,\ldots,M-1$$

(the squares of the eigenvalues of the Jacobi iteration matrix), so it should be clear that the mid-range eigenvalues are the smallest. The difference

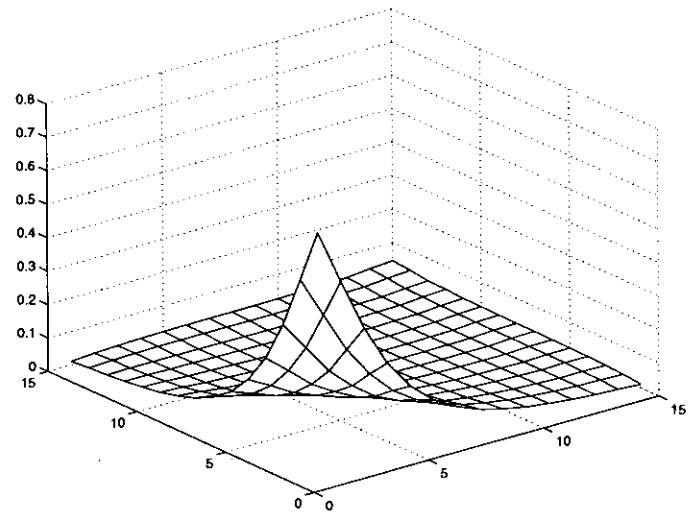

FIGURE 10.10.3. Plot of the coefficients of the error after two Gauss-Seidel steps with $\mathbf{u}_0$, (10.10.1), as the starting guess.

here is that the eigenvectors of the Gauss-Seidel iteration matrix are not the same as the eigenvectors of the matrix A. In fact, in Section 10.5.7 we saw that the Gauss-Seidel iteration matrix does not have a full set of eigenvectors. It is just the case that the mid-range eigenvalues of the Gauss-Seidel iteration matrix (the smallest eigenvalues of the Gauss-Seidel iteration matrix) correspond to the high frequency components given with respect to the eigenvectors of A.

In most of our work on multigrid, we will use the weighted Jacobi scheme. It is not that the weighted Jacobi scheme is the best. We will use it because of the fact that it does work well and because it is very convenient that the eigenvectors of the iteration matrix of the weighted Jacobi scheme and the matrix A are the same.

HW 10.10.1 Consider the one dimensional model problem

$$-v'' = 0, \quad x \in (0,1) \tag{10.10.13}$$

$$v(0) = v(1) = 0 \tag{10.10.14}$$

and the finite difference approximation

$$-\frac{1}{\Delta x^2}u_{k-1} + \frac{2}{\Delta x^2}u_k - \frac{1}{\Delta x^2}u_{k+1} = 0, \quad k = 1, \ldots, M-1 \tag{10.10.15}$$

$$u_0 = u_M = 0. \tag{10.10.16}$$

(a) Show that the eigenvalues and eigenvectors associated with the matrix

of the numerical problem are given by

$$\mu_k = \frac{4}{\Delta x^2}\sin^2\frac{k\pi}{2M}, \quad \mathbf{w}_k = \left[\sin\frac{k\pi}{M}\ \sin\frac{2k\pi}{M}\ \cdots\ \sin\frac{(M-1)k\pi}{M}\right], \quad k = 1,\ldots,M-1. \tag{10.10.17}$$

(b) Let $R_{J\omega}$ denote the iteration matrix of the weighted Jacobi scheme designed to solve difference equations (10.10.15)–(10.10.16). Show that the eigenvalues and eigenvectors of $R_{J\omega}$ are given by

$$\lambda_{J\omega_k} = 1 - 2\omega\sin^2 k\pi/(2M), \quad \mathbf{w}_k, \quad k = 1,\ldots,M-1.$$

(c) Show that if we choose $\omega = \frac{2}{3}$, $|\lambda_{j\omega_k}| \leq \frac{1}{3}$ for $k = M/2,\ldots,M-1$.
(d) Use the weighted Jacobi scheme to solve difference equations (10.10.15)–(10.10.16). Let $M = 16$, $\omega = \frac{2}{3}$ and use a stopping criterion consisting of the sup-norm of two successive iterates and a tolerance of 10^{-4}.

HW 10.10.2 Consider model problem (10.2.3)–(10.2.7) and the weighted Jacobi scheme.
(a) Show that when $\omega = \frac{2}{3}$, $\lambda_{M/2}^{M/2} = -\lambda_M^M$. (Remember that λ_M^M is not an eigenvalue.)
(b) Show that $\omega = \frac{2}{3}$ implies that $|\lambda_s^p| \leq \frac{1}{3}$ for $p, s = M/2,\ldots,M-1$.
(c) Use $M = 16$, $\omega = \frac{2}{3}$ and the weighted Jacobi scheme to solve the two dimensional numerical model problem (10.2.3)–(10.2.7) with F and f taken to be zero and (10.10.1) as the initial guess. Use the sup-norm of the difference between successive iterates as a stopping criterion with a tolerance of 10^{-5}.

10.10.3 Grid Transfers

The uniqueness of the multigrid scheme lies not in the use of the iterative scheme, but in the transfer of information to coarser grids and the complementary nature of the iterative scheme and the grid transfers. The grid transfers are very important aspects of the multigrid schemes and are less understood than the iterative schemes. In this section we will introduce some notation, define the grid transfers in which we have interest and discuss some of the important properties of these grid transfers. Much of the discussion of the grid transfers will be done for the analogous one dimensional grid transfers. The one dimensional discussion of the grid transfers is much easier to follow, and it should be clear that the properties that we discuss with respect to one dimensional grid transfers carry over to two dimensional grid transfers in a fairly obvious (not necessarily clean) manner.

We begin by discussing the different grids that we shall use. We consider the region $R = (0,1) \times (0,1)$. As we mentioned earlier, we will use $M_x =$

$M_y = M$ and choose $M = 2^p$. When we want to choose a particular case, we will usually choose $p = 4$, or in some cases with the one dimensional examples, $p = 3$. We begin by considering a uniform grid on R, with $\Delta x = \Delta y = 1/M$. We will denote this grid by either $\mathcal{G}^h$ or $\mathcal{G}^M$. We should note that earlier we had referred to $\mathcal{G}^M$ and $\mathcal{G}^{M/2}$ as the M-grid and the $M/2$-grid (thinking that the expressions were reasonably obvious and at that time it was not yet necessary to introduce the notation). The notation $\mathcal{G}^h$ is the notation that seems to be used in the multigrid literature (where they denote Δx by h). We then define the subsequent coarsenings of the $\mathcal{G}^h = \mathcal{G}^M$ grid by $\mathcal{G}^{2h}$, $\mathcal{G}^{4h}, \ldots, \mathcal{G}^{1/2}$ or $\mathcal{G}^{2^{p-1}}, \ldots, \mathcal{G}^{2^1}$. We hope that it is clear that the grid $\mathcal{G}^{2h}$ or $\mathcal{G}^{2^{p-1}}$ consists of the uniform grid on R with $\Delta x = 2h = 2/M$. There will be times when the notation $\mathcal{G}^{sh}$ is most convenient and times when the $\mathcal{G}^{2^s}$ notation is most convenient. When we are discussing iterates or vectors defined on the various grids, we will use the notation $\mathbf{u}^h$ to denote a vector defined on $\mathcal{G}^h$, $\mathbf{u}^{2h}$ to denote a vector defined on $\mathcal{G}^{2h}$, etc. We will also include the appropriate superscript when we are writing the components of vectors on the various grids.

We will define our grid transfers from $\mathcal{G}^h$ to $\mathcal{G}^{2h}$ and from $\mathcal{G}^{2h}$ to $\mathcal{G}^h$. It is clear, we hope, that all other transfers to coarser grids or to finer grids are analogous. We begin with the transfer from $\mathcal{G}^h$ to $\mathcal{G}^{2h}$, which we denote by I_h^{2h}, $I_h^{2h} : \mathcal{G}^h \to \mathcal{G}^{2h}$. Note that the notation is similar to tensor notation in that $I_h^{2h}\mathbf{u}^h = \mathbf{u}^{2h}$; the subscript on the I and the superscript on the $\mathbf{u}$ can be thought of as canceling each other, leaving us with the superscript from the I (it might help us remember where we are). The operator I_h^{2h} is referred to as the **restriction operator**. One such restriction operator is called **full weighting**. If we let $I_h^{2h}\mathbf{u}^h = \mathbf{u}^{2h}$, the full weighting operator is defined componentwise as

$$u_{j\,k}^{2h} = \frac{1}{16}\Big[u_{2j-1\,2k-1}^h + u_{2j-1\,2k+1}^h + u_{2j+1\,2k+1}^h + u_{2j+1\,2k-1}^h + 2\big(u_{2j\,2k-1}^h + u_{2j\,2k+1}^h + u_{2j-1\,2k}^h + u_{2j+1\,2k}^h\big) + 4u_{2j\,2k}^h\Big],$$
$$j,k = 1, \ldots \frac{M}{2} - 1. \tag{10.10.18}$$

It should be clear that what we are doing in defining the full weighting grid transfer operator is defining the coarse grid value to be the weighted average of the function at the point and at all of its fine grid neighbors. With such a definition, it should be reasonably clear that full weighting should be a smoothing operator—especially on smooth functions.

Another restriction that is commonly used in many computational problems is the **injection operator** defined by $\mathcal{I}_h^{2h}\mathbf{u}^h = \mathbf{u}^{2h}$, where

$$u_{j\,k}^{2h} = u_{2j\,2k}^h, \quad j,k = 1, \ldots, \frac{M}{2} - 1. \tag{10.10.19}$$

Obviously, the injection operator is easier to implement than the full weighting operator. We will see in our computational tests that the results are

not very different whether injection or full weighting is used. Generally, full weighting becomes necessary for theoretical results and very difficult computational results. For our discussion, we will denote the full weighting restriction operator by I_h^{2h} and the injection restriction operator by $\mathcal{I}_h^{2h}$.

To transfer our coarse grid approximations back to the fine grid, we use a **prolongation** or **interpolation operator**. By far the most common prolongation operator is linear interpolation, where we define $I_{2h}^h : \mathcal{G}^{2h} \to \mathcal{G}^h$ by $I_{2h}^h \mathbf{u}^{2h} = \mathbf{u}^h$, where

$$u^n_{2j\,2k} = u^{2h}_{j\,k} \tag{10.10.20}$$

$$u^n_{2j+1\,2k} = \frac{1}{2}\left(u^{2h}_{j\,k} + u^{2h}_{j+1\,k}\right) \tag{10.10.21}$$

$$u^n_{2j\,2k+1} = \frac{1}{2}\left(u^{2h}_{j\,k} + u^{2h}_{j\,k+1}\right) \tag{10.10.22}$$

$$u^n_{2j+1\,2k+1} = \frac{1}{4}\left(u^{2h}_{j\,k} + u^{2h}_{j+1\,k} + u^{2h}_{j\,k+1} + u^{2h}_{j+1\,k+1}\right) \tag{10.10.23}$$

$$\text{for} \quad j,k = 0,\ldots,\frac{M}{2} - 1. \tag{10.10.24}$$

It would appear that the interpolation operator produces a very smooth solution on $\mathcal{G}^h$. It does. However, as we shall see, if the true error is oscillatory, this smooth interpolated error can be a bad approximation of the true error.

We now turn to our one dimensional model problem to describe what happens or what can happen when we use the restriction and prolongation operators to transfer information between grids. Specifically, we will use full weighting as our restriction operator and linear interpolation as our prolongation operator. We let $M = 8$ and note that the problem to be solved on the $\mathcal{G}^h$ grid can be written as

$$A^h\mathbf{u}^h = \frac{1}{\Delta x^2}\begin{pmatrix} 2 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 2 \end{pmatrix}\begin{bmatrix} u_1^h \\ u_2^h \\ u_2^h \\ u_3^h \\ u_4^h \\ u_5^h \\ u_6^h \\ u_7^h \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \boldsymbol{\theta}^h. \tag{10.10.25}$$

We will write all vectors defined on the $\mathcal{G}^h$ grid expanded with respect to the eigenvectors of A, $\mathbf{w}_j^h$, $j = 1, \ldots, 7$. Recall from HW10.10.1 that the k-th component of $\mathbf{w}_j^h$ is given by $w_{j_k}^h = \sin \frac{jk\pi}{8}$. Whatever we do to our initial guess on the $\mathcal{G}^h$ grid, the vector that we want to pass to the $\mathcal{G}^{2h}$ grid can be written as

$$\mathbf{u}^h = \sum_{j=1}^{M-1} a_j \mathbf{w}_j^h.$$

Since I_h^{2h} is linear, we see that the result on the $\mathcal{G}^{2h}$ grid is

$$\mathbf{u}^{2h} = I_h^{2h}\mathbf{u}^h = \sum_{j=1}^{M-1} a_j I_h^{2h}\mathbf{w}_j.$$

Hence, we must determine how I_h^{2h} transfers $\mathbf{w}_j^h$ to the $\mathcal{G}^{2h}$ grid.

Before we proceed with the task of transforming $\mathbf{w}_j$, we note that the problem to be solved, given on the $\mathcal{G}^{2h}$ grid ($M_x = M_y = 4$), can be written as

$$A^{2h}\mathbf{u}^{2h} = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix} \begin{bmatrix} u_1^{2h} \\ u_2^{2h} \\ u_3^{2h} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = \boldsymbol{\theta}^{2h},$$

and the eigenvectors of A^{2h} are $\mathbf{w}_j^{2h}$, $j = 1, 2, 3$, where $w_{j_k}^{2h} = \sin \frac{jk\pi}{4}$, $k = 1, 2, 3$.

To make the necessary calculations a bit clearer, it is useful to note that I_h^{2h} and I_{2h}^h can be written as matrices

$$I_h^{2h} = \frac{1}{4} \begin{pmatrix} 1 & 2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 2 & 1 \end{pmatrix} \tag{10.10.26}$$

and

$$I_{2h}^h = \frac{1}{2} \begin{pmatrix} 1 & 0 & 0 \\ 2 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 2 \\ 0 & 0 & 1 \end{pmatrix}. \tag{10.10.27}$$

We see that

$$\begin{aligned} I_h^{2h}\mathbf{w}_1^h &= \begin{bmatrix} \sin\frac{\pi}{8} + 2\sin\frac{2\pi}{8} + \sin\frac{3\pi}{8} \\ \sin\frac{3\pi}{8} + 2\sin\frac{4\pi}{8} + \sin\frac{5\pi}{8} \\ \sin\frac{5\pi}{8} + 2\sin\frac{6\pi}{8} + \sin\frac{7\pi}{8} \end{bmatrix} \\ &= \cos^2\frac{\pi}{16}\mathbf{w}_1^{2h}. \end{aligned}$$

The analogous calculations hold for $j = 2$ and 3 and we get

$$I_h^{2h}\mathbf{w}_2^h = \cos^2\frac{2\pi}{16}\mathbf{w}_2^{2h} \text{ and } I_h^{2h}\mathbf{w}_3^h = \cos^2\frac{3\pi}{16}\mathbf{w}_3^{2h}.$$

Similarly, we note that

$$I_h^{2h}\mathbf{w}_{M-j}^h = -\sin^2\frac{j\pi}{16}\mathbf{w}_j^{2h}, \quad j = 1, 2, 3,$$

and

$$I_h^{2h}\mathbf{w}_4^h = \boldsymbol{\theta}^{2h}.$$

We note that the formulas for general M are

$$I_h^{2h}\mathbf{w}_j^h = \cos^2\frac{j\pi}{2M}\mathbf{w}_j^{2h}, \quad j = 1, \ldots, \frac{M}{2} - 1 \tag{10.10.28}$$

$$I_h^{2h}\mathbf{w}_{M/2}^h = \boldsymbol{\theta}^{2h} \tag{10.10.29}$$

$$I_h^{2h}\mathbf{w}_{M-j}^h = -\sin^2\frac{j\pi}{2M}\mathbf{w}_j^{2h}, \quad j = 1, \ldots, \frac{M}{2} - 1. \tag{10.10.30}$$

A similar calculation shows that

$$I_{2h}^h\mathbf{w}_j^{2h} = \cos^2\frac{j\pi}{2M}\mathbf{w}_j^h - \sin^2\frac{j\pi}{2M}\mathbf{w}_{M-j}^h, \quad j = 1, \ldots, \frac{M}{2} - 1. \tag{10.10.31}$$

We are now in a situation to describe reasonably clearly some of the good and bad properties of our grid transfers. We recall in the introduction that we wanted grid transfers that would transform the j-th component on the $\mathcal{G}^h$ grid to the j-th component on the $\mathcal{G}^{2h}$ grid (for $j = 1, \ldots, M/2 - 1$. We see that we get that a multiple of what we wanted (which is surely acceptable), except for the fact that a multiple of the $M - j$ component of the error also gets mapped onto the j-th mode. The $M - j$ mode is referred to as the **complement** of the j-th mode. This is not a great problem. Remember that we are planning on eliminating as much as we can of the $(M/2)$-th through the M-th components of the error on the $\mathcal{G}^h$ grid. In addition, within the set of the $(M/2)$-th to the M-th components of the error, the largest components should be near $M/2$. These components get mapped to the high frequency components on the $\mathcal{G}^{2h}$ grid and should be eliminated fastest on the $\mathcal{G}^{2h}$ grid. Hence, these values that are being included should be small. Though we must be careful of this aliasing, the property should not cause great problems.

The grid transfers from coarse to fine appear to cause bigger problems. The transfers obviously introduce high frequency errors—the types of errors that we claimed that we would get rid of before going to the coarse grid. This difficulty is also not as bad as it might seem. We note from expression (10.10.31) that the highest frequency errors on $\mathcal{G}^h$ come from the lowest frequency errors from the $\mathcal{G}^{2h}$ grid. From (10.10.31) we see that if $j \ll M/2$, then

$$I_{2h}^h\mathbf{w}_j^{2h} = \Big(1 - \mathcal{O}((j/M)^2)\Big)\mathbf{w}_j^h + \mathcal{O}((j/M)^2)\mathbf{w}_{M-j}^h.$$

Hence, we see that the smooth modes are not affected much by the grid transfer I_{2h}^{h}. Also, these low frequency waves on the $\mathcal{G}^{2h}$ grid are the modes that contribute the highest frequency modes on $\mathcal{G}^{h}$. The above result implies that these high frequency modes will be small.

We have seen that we must be careful how we transfer data from grid to grid. In the next section, when we introduce the multigrid algorithm, we will see how we handle the problems caused by the grid transfers. And finally, before we leave this topic, we state again that the properties of the grid transfers presented above have two dimensional analogues when we consider the two dimensional model problem. The same trigonometric identities that gave the above results will give two dimensional results. There will be complementary modes. They will be more complicated, and each low frequency mode will have more than one complementary mode. The interpolation operator I_{2h}^{n} will still introduce oscillatory modes. In two dimensions, a mode on the $\mathcal{G}^{2h}$ grid will map onto the same mode on the $\mathcal{G}^{h}$ grid plus three high frequency modes. The important part is that the good and bad aspects of the grid transfers in one dimension appear in a similar way and for the same reasons in the grid transfers for two dimensions.

10.10.4 Multigrid Algorithm

10.10.4.1 Coarse Grid Correction Scheme

We saw in Section 10.5 that if $\mathbf{w}$ is an approximation to the solution of $A\mathbf{u} = \mathbf{f}$ and $\mathbf{r}$ is the residual, $\mathbf{r} = \mathbf{f} - A\mathbf{w}$, then the solution to $A\mathbf{e} = \mathbf{r}$ is the error and the exact solution is given by $\mathbf{u} = \mathbf{w} + \mathbf{e}$. The residual correction schemes then proceeded to iteratively find approximate solutions to $A\mathbf{e} = \mathbf{r}$ that were added to $\mathbf{w}$. The coarse grid correction scheme is another residual correction method. The coarse grid correction scheme gets its name from the fact that it finds the approximate solution of $A\mathbf{e} = \mathbf{r}$ by solving on a coarse grid. We have separated the coarse grid correction scheme into a subsection of the section on the multigrid algorithm because the coarse grid correction scheme is the cornerstone on which the multigrid algorithm is built. In this section we will present the coarse grid correction scheme and do some calculations that should help the reader understand the scheme.

We begin by introducing the coarse grid correction scheme in a general setting where the problem to be solved is given by $A^h\mathbf{u}^h = \mathbf{f}^h$ on grid $\mathcal{G}^h$, $\mathcal{G}^{2h}$ represents the associated coarse grid, and A^{2h} is the matrix of the problem defined on the coarse grid. We then obtain the following coarse grid correction algorithm.

Coarse Grid Correction Scheme

Relax m_1 times on $A^h\mathbf{u}^h = \mathbf{f}^h$ on $\mathcal{G}^h$ with initial guess $\mathbf{u}_0^h$, result $\mathbf{u}_{m_1}^h$.
Compute $\mathbf{r}^h = \mathbf{f}^h - A^h\mathbf{u}_{m_1}^h$.
Compute $\mathbf{f}^{2h} = I_h^{2h}\mathbf{r}^h$.

Solve $A^{2h}\mathbf{e}^{2h} = \mathbf{f}^{2h}$ on $\mathcal{G}^{2h}$.
Correct fine grid approximation $\hat{\mathbf{u}}^h_{m_1} = \mathbf{u}^h_{m_1} + I^h_{2h}\mathbf{e}^{2h}$.
Relax m_2 times on $A^h\mathbf{u}^h = \mathbf{f}^h$ on $\mathcal{G}^h$ with initial guess $\hat{\mathbf{u}}^h_{m_1}$, result $\hat{\mathbf{u}}^h_{m_2}$.

One Dimensional Model: Coarse Grid Correction Scheme

If we apply the above scheme to our one dimensional model problem, A^h and A^{2h} are given by

$$A^h = \frac{1}{\Delta x^2}\begin{pmatrix} 2 & -1 & 0 & \cdots & \\ -1 & 2 & -1 & 0 & \cdots \\ & \cdots & \cdots & \cdots & \\ \cdots & 0 & -1 & 2 & -1 \\ & \cdots & 0 & -1 & 2 \end{pmatrix}_{M-1\times M-1} \tag{10.10.32}$$

and

$$A^{2h} = \frac{1}{(2\Delta x)^2}\begin{pmatrix} 2 & -1 & 0 & \cdots & \\ -1 & 2 & -1 & 0 & \cdots \\ & \cdots & \cdots & \cdots & \\ \cdots & 0 & -1 & 2 & -1 \\ & \cdots & 0 & -1 & 2 \end{pmatrix}_{M/2-1\times M/2-1} \tag{10.10.33}$$

For this discussion we will choose the problem with $\mathbf{f}^h = \boldsymbol{\theta}$, use full weighting and linear interpolation as our restriction operator and our prolongation operator, respectively, and use

$$\mathbf{u}^h_0 = \sum_{j=1}^{M-1} \mathbf{w}^h_j \tag{10.10.34}$$

as our initial guess, where $\mathbf{w}^h_j$, $j = 1,\ldots,M-1$, are the eigenvectors of A^h. See (10.10.17). We will denote the eigenvectors of A^{2h} by $\mathbf{w}^{2h}_j$, $j = 1\ldots.,M/2-1$ (which are also given by (10.10.17)), the eigenvalues of A^h and A^{2h} by $\mu^h_j = (4/\Delta x^2)\sin^2\frac{j\pi}{2M}$, $j = 1,\ldots,M-1$, and $\mu^{2h}_j = (1/\Delta x^2)\sin^2\frac{j\pi}{M}$, $j = 1,\ldots,M/2-1$, respectively, and the eigenvalues of the iteration matrix $R_{J\omega}$ associated with the weighted Jacobi scheme ($\omega = \frac{2}{3}$) on $\mathcal{G}^h$ by λ^h_j, $j = 1,\ldots,M-1$. See HW10.10.1 and HW10.10.2. Recall that the eigenvectors of $R_{J\omega}$ are given by $\mathbf{w}^h_j$, $j = 1,\ldots,M-1$.

We will approach this problem by solving $M-1$ separate problems and using the linearity of the scheme to give the general solution. We consider each of the eigenvectors in the expression (10.10.34) separately. We feel that this approach is more informative. We begin by considering the initial guess

$$\mathbf{u}^h_{0_j} = \mathbf{w}^h_j, \quad 1 \le j \le M/2-1. \tag{10.10.35}$$

Since the initial error is given by $\mathbf{e}_0 = \boldsymbol{\theta} - \mathbf{u}_{0_j}^h = -\mathbf{u}_{0_j}^h$ and we know that the error after m relaxation steps is given by $\mathbf{e}_m = R_{J\omega}^m \mathbf{e}_0$, after m_1 relaxation steps we have

$$\mathbf{e}_{m_1} = R_{J\omega}^{m_1}(-\mathbf{u}_{0_j}^h) = -R_{J\omega}^{m_1}\mathbf{w}_j^h = -(\lambda_j^h)^{m_1}\mathbf{w}_j^h.$$

Thus the solution after m_1 relaxation steps is given by

$$\mathbf{u}_{m_{1j}}^h = \boldsymbol{\theta} - \mathbf{e}_{m_1} = (\lambda_j^h)^{m_1}\mathbf{w}_j^h. \tag{10.10.36}$$

We can then write $\mathbf{r}_j^h$ as

$$\mathbf{r}_j^h = \mathbf{f}^h - A^h\mathbf{u}_{m_{1j}}^h = -(\lambda_j^h)^{m_1}A^h\mathbf{w}_j^h = -(\lambda_j^h)^{m_1}\mu_j^h\mathbf{w}_j^h. \tag{10.10.37}$$

Using (10.10.28), we write $\mathbf{f}_j^{2h} = I_h^{2h}\mathbf{r}_j^h$ as

$$\begin{aligned}\mathbf{f}_j^{2h} &= -(\lambda_j^h)^{m_1}\mu_j^h I_h^{2h}\mathbf{w}_j^h \\ &= -(\lambda_j^h)^{m_1}\mu_j^h \cos^2\frac{j\pi}{2M}\mathbf{w}_j^{2h}.\end{aligned} \tag{10.10.38}$$

We next must solve $A^{2h}\mathbf{e}_j^{2h} = \mathbf{f}_j^{2h}$. If we write $\mathbf{e}_j^{2h}$ as

$$\mathbf{e}_j^{2h} = \sum_{k=1}^{M/2-1} c_k\mathbf{w}_k^{2h}, \tag{10.10.39}$$

it is not difficult to see that

$$A^{2h}\mathbf{e}_j^{2h} = \sum_{k=1}^{M/2-1} c_k A^{2h}\mathbf{w}_k^{2h} = \sum_{k=1}^{M/2-1} c_k\mu_k^{2h}\mathbf{w}_k^{2h}$$

and that we can solve for c_k as

$$c_j = -(\lambda_j^h)^{m_1}\frac{\mu_j^h}{\mu_j^{2h}}\cos^2\frac{j\pi}{2M}, \text{ and } c_k = 0 \text{ when } k \neq j.$$

Thus

$$\mathbf{e}_j^{2h} = -(\lambda_j^h)^{m_1}\frac{\mu_j^h}{\mu_j^{2h}}\cos^2\frac{j\pi}{2M}\mathbf{w}_j^{2h}. \tag{10.10.40}$$

Using (10.10.31), we transform $\mathbf{e}_j^{2h}$ back to $\mathcal{G}^h$, i.e.,

$$\begin{aligned}\mathbf{e}_j^h &= I_{2h}^h\mathbf{e}_j^{2h} = c_j I_{2h}^h\mathbf{w}_j^{2h} \\ &= c_j\Big[\cos^2\frac{j\pi}{2M}\mathbf{w}_j^h - \sin^2\frac{j\pi}{2M}\mathbf{w}_{M-j}^h\Big].\end{aligned} \tag{10.10.41}$$

Hence, we see that we can write $\hat{\mathbf{u}}^h_{m_1 j}$ as

$$\begin{aligned}\hat{\mathbf{u}}^h_{m_1 j} &= \mathbf{u}^h_{m_1 j} + \mathbf{e}^h_j \\ &= \left[\left(\lambda^h_j\right)^{m_1} + c_j \cos^2 \frac{j\pi}{2M}\right]\mathbf{w}^h_j - c_j \sin^2 \frac{j\pi}{2M}\mathbf{w}^h_{M-j} \qquad (10.10.42)\end{aligned}$$

and $\hat{\mathbf{u}}^h_{m_2 j}$ as

$$\hat{\mathbf{u}}^h_{m_2 j} = \left(\lambda^h_j\right)^{m_2}\left[\left(\lambda^h_j\right)^{m_1} + c_j \cos^2 \frac{j\pi}{2M}\right]\mathbf{w}^h_j - \left(\lambda^h_{M-j}\right)^{m_2} c_j \sin^2 \frac{j\pi}{2M}\mathbf{w}^h_{M-j}. \qquad (10.10.43)$$

We should note carefully that that above calculation applies only to the first $M/2-1$ eigenvectors, $1 \le j \le M/2-1$. More importantly, we should note that if initially we have only a low frequency component of error ($1 \le j \le M/2-1$), the coarse grid correction of the error, (10.10.41), consists of two components. One component is a multiple of the original error (both are multiples of $\mathbf{w}^h_j$). It is hoped that this component of the coarse grid correction of the error will correct the initial solution. The second component of the coarse grid correction of the error is a high frequency component (a multiple of $\mathbf{w}^h_{M-j}$). It is this introduction to the error of the complementary high frequency mode that makes it necessary to relax on grid $\mathcal{G}^h$ the second time. *The purpose of the m_2 relaxation iterations at the end of the coarse grid correction scheme is to get rid of the high frequency modes that have been introduced by I^h_{2h}.*

We next want to consider using the initial guess

$$\mathbf{u}^h_{0_{M/2}} = \mathbf{w}^h_{M/2}. \qquad (10.10.44)$$

Obviously, we want to follow the same steps we used above. As before, we have

$$\mathbf{u}^h_{m_1 M/2} = \left(\lambda^h_{M/2}\right)^{m_1}\mathbf{w}^h_{M/2} \qquad (10.10.45)$$

and

$$\mathbf{r}^h_{M/2} = -\left(\lambda^h_{M/2}\right)^{m_1}\mu^h_{M/2}\mathbf{w}^h_{M/2}. \qquad (10.10.46)$$

By (10.10.29) we see that

$$\mathbf{f}^{2h}_{M/2} = I^{2h}_h \mathbf{r}^h_{M/2} = \boldsymbol{\theta}. \qquad (10.10.47)$$

The rest of this case is easy, but for the sake of completeness, we write

$$c_{M/2} = 0 \qquad (10.10.48)$$

$$\mathbf{e}^h_{M/2} = \boldsymbol{\theta} \qquad (10.10.49)$$

$$\hat{\mathbf{u}}^h_{m_1 M/2} = \left(\lambda^h_{M/2}\right)^{m_1}\mathbf{w}^h_{M/2} \qquad (10.10.50)$$

$$\hat{\mathbf{u}}^h_{m_2 M/2} = \left(\lambda^h_{M/2}\right)^{m_2}\left(\lambda^h_{M/2}\right)^{m_1}\mathbf{w}^h_{M/2}. \qquad (10.10.51)$$

And finally, we consider

$$\mathbf{u}_{0_j}^h = \mathbf{w}_j^h, \quad j = M/2+1, \ldots, M-1. \tag{10.10.52}$$

Again we want to mimic what we did in the case of j between 1 and $M/2-1$. We get

$$\mathbf{u}_{m_1 j}^h = \left(\lambda_j^h\right)^{m_1} \mathbf{w}_j^h \tag{10.10.53}$$

and

$$\mathbf{r}_j^h = -\left(\lambda_j^h\right)^{m_1} \mu_j^h \mathbf{w}_j^h. \tag{10.10.54}$$

Now using (10.10.30), we can write

$$\begin{aligned}
\mathbf{f}_j^{2h} &= I_h^{2h} \mathbf{r}_j^h = -\left(\lambda_j^h\right)^{m_1} \mu_j^h I_h^{2h} \mathbf{w}_j^h \\
&= -\left(\lambda_j^h\right)^{m_1} \mu_j^h \Big[-\sin^2 \frac{(M-j)\pi}{2M} \mathbf{w}_{M-j}^{2h} \Big] \\
&= \left(\lambda_j^h\right)^{m_1} \mu_j^h \cos^2 \frac{j\pi}{2M} \mathbf{w}_{M-j}^{2h}.
\end{aligned} \tag{10.10.55}$$

Again the solution of $A^{2h}\mathbf{e}_j^{2h} = \mathbf{f}_j^{2h}$ can be found by assuming that $\mathbf{e}_j^{2h}$ can be written as in (10.10.39), i.e., as $\sum_{k=1}^{M/2-1} C_k \mathbf{w}_k^{2h}$. As before, we find that $C_k = 0$ for $k \neq M-j$,

$$C_{M-j} = \left(\lambda_j^h\right)^{m_1} \frac{\mu_j^h}{\mu_{M-j}^{2h}} \cos^2 \frac{j\pi}{2M}$$

and

$$\mathbf{e}_j^{2h} = \left(\lambda_j^h\right)^{m_1} \frac{\mu_j^h}{\mu_{M-j}^{2h}} \cos^2 \frac{j\pi}{2M} \mathbf{w}_{M-j}^{2h}. \tag{10.10.56}$$

Again using (10.10.31), we transform $\mathbf{e}_j^{2h}$ back to $\mathcal{G}^h$,

$$\begin{aligned}
\mathbf{e}_j^h &= I_{2h}^h \mathbf{e}_j^{2h} = C_{M-j} I_{2h}^h \mathbf{w}_{M-j}^{2h} \\
&= C_{M-j} \Big[\cos^2 \frac{(M-j)\pi}{2M} \mathbf{w}_{M-j}^h - \sin^2 \frac{(M-j)\pi}{2M} \mathbf{w}_j^h \Big] \\
&= C_{M-j} \Big[\sin^2 \frac{j\pi}{2M} \mathbf{w}_{M-j}^h - \cos^2 \frac{j\pi}{2M} \mathbf{w}_j^h \Big].
\end{aligned} \tag{10.10.57}$$

We can write $\hat{\mathbf{u}}_{m_1 j}^h$ as

$$\hat{\mathbf{u}}_{m_1 j}^h = \Big[\left(\lambda_j^h\right)^{m_1} - C_{M-j} \cos^2 \frac{j\pi}{2M} \Big] \mathbf{w}_j^h + C_{M-j} \sin^2 \frac{j\pi}{2M} \mathbf{w}_{M-j}^h \tag{10.10.58}$$

and $\hat{\mathbf{u}}^h_{m_{2j}}$ as

$$\hat{\mathbf{u}}^h_{m_{2j}} = \left(\lambda^h_j\right)^{m_2}\left[\left(\lambda^h_j\right)^{m_1} - C_{M-j}\cos^2\frac{j\pi}{2M}\right]\mathbf{w}^h_j + \left(\lambda^h_{M-j}\right)^{m_2} C_{M-j}\sin^2\frac{j\pi}{2M}\mathbf{w}^h_{M-j}. \tag{10.10.59}$$

We note that for this last case, $j = M/2+1,\ldots,M-1$, initially be begin with a high frequency component of error. After one iteration of the coarse grid correction scheme, we have introduced a low frequency component into the error. The m_2 relaxation steps performed in (10.10.59) are not very effective at eliminating the low frequency components of the error. The good news is that these low frequency components of the error that have been introduced are small. Recall that the high frequency component of the error that we have initially is associated with an eigenvalue of the iteration matrix $R_{J\omega}$ that is small in magnitude and $C_{M-j} = \left(\lambda^h_j\right)^{m_1}\frac{\mu^h_j}{\mu^{2h}_{M-j}}\cos^2\frac{j\pi}{2M}$. Hence, C_{M-j} is small.

We now return to the consideration of our initial problem with initial guess (10.10.34). Using (10.10.42), (10.10.50), and (10.10.58), we get

$$\begin{aligned}
\hat{\mathbf{u}}^h_{m_1} &= \sum_{j=1}^{M/2-1}\left\{\left[\left(\lambda^h_j\right)^{m_1} + c_j\cos^2\frac{j\pi}{2M}\right]\mathbf{w}^h_j - c_j\sin^2\frac{j\pi}{2M}\mathbf{w}^h_{M-j}\right\} \\
&\quad + \left(\lambda^h_{M/2}\right)^{m_1}\mathbf{w}^h_{M/2} \\
&\quad + \sum_{j=M/2+1}^{M-1}\left\{\left[\left(\lambda^h_j\right)^{m_1} - C_{M-j}\cos^2\frac{j\pi}{2M}\right]\mathbf{w}^h_j + C_{M-j}\sin^2\frac{j\pi}{2M}\mathbf{w}^h_{M-j}\right\} \\
&= \sum_{j=1}^{M/2-1}\left\{\left(\lambda^h_j\right)^{m_1} + \left(c_j + C_j\right)\cos^2\frac{j\pi}{2M}\right\}\mathbf{w}^h_j \\
&\quad + \left(\lambda^h_{M/2}\right)^{m_1}\mathbf{w}^h_{M/2} \\
&\quad + \sum_{j=M/2+1}^{M-1}\left\{\left(\lambda^h_j\right)^{m_1} - \left(C_{M-j} + c_{M-j}\right)\cos^2\frac{j\pi}{2M}\right\}\mathbf{w}^h_j.
\end{aligned} \tag{10.10.60}$$

And using (10.10.43), (10.10.51), and (10.10.59) gives us

$$\begin{aligned}
\hat{\mathbf{u}}^h_{m_2} &= \sum_{j=1}^{M/2-1}\left(\lambda^h_j\right)^{m_2}\left\{\left(\lambda^h_j\right)^{m_1} + \left(c_j + C_j\right)\cos^2\frac{j\pi}{2M}\right\}\mathbf{w}^h_j \\
&\quad + \left(\lambda^h_{M/2}\right)^{m_2}\left(\lambda^h_{M/2}\right)^{m_1}\mathbf{w}^h_{M/2} \\
&\quad + \sum_{j=M/2+1}^{M-1}\left(\lambda^h_j\right)^{m_2}\left\{\left(\lambda^h_j\right)^{m_1} - \left(C_{M-j} + c_{M-j}\right)\cos^2\frac{j\pi}{2M}\right\}\mathbf{w}^h_j.
\end{aligned} \tag{10.10.61}$$

We realize that equation (10.10.61) is not a very nice expression. Besides the fact that what we see is very complex, we must remember that there is more complexity hidden in the coefficients c_j and C_j. We next include a computation using the above analysis that might help clarify the coarse grid correction scheme.

Example 10.10.1 Use equation (10.10.61) to solve the one dimensional model problem with $M = 8$.

Solution: We should realize that we are solving the finite difference problem (10.10.15)–(10.10.16) (whose solution, we hope, we already know) using (10.10.34) as our initial guess. As we have mentioned before, (10.10.34) is not a good initial guess. At this time we are more interested in how we reduce the various modes of the error, not exactly how well we approximate the solution. All we really must do is implement formula (10.10.61). We note that this can be implemented numerically (which we do for the convenience of the graphics) or analytically on Maple or Mathematica.

We perform the computations described above with $m_1 = m_2 = 3$. In Figure 10.10.4-(a), we plot the coefficients of $\mathbf{u}^h_{m_1}$ expanded with respect to the eigenvectors $\mathbf{w}^h_1, \ldots, \mathbf{w}^h_7$. Recall that when we are plotting $\mathbf{u}^h$, $\mathbf{u}^h_{m_1}$, etc., we are essentially plotting the error. We see that the three iterations of the weighted Jacobi scheme have virtually eliminated the high frequency components of the error and do not eliminate the low frequency components of the error. In (c) of Figure 10.10.4, we plot the components of $\hat{\mathbf{u}}_{m_1}$. This represents the solution after we have computed the error on the $\mathcal{G}^{2h}$ grid, $\mathbf{e}^{2h}$, interpolated this error to the $\mathcal{G}^h$ grid, $\mathbf{e}^h$, and added this interpolated error to $\mathbf{u}^h_{m_1}$. We see that the coefficients associated with the low frequency components have gotten much smaller. We also see that the high frequency components of the error appear just as formula (10.10.61a) would predict that they should, i.e., for $j = 5, 6$ and 7, the coefficients have approximately the same magnitude as the coefficients associated with $j = 3, 2$ and 1, respectively. In plot (d) we see how these high frequency components are again eliminated by three iterations on the fine grid. We then perform a second iteration of the coarse grid correction scheme. We see that on the second step after the coarse grid error has been interpolated to the fine grid and added to the previous solution on that grid, the high frequency components of the error dominate. In plot (f) we see that after three iterations of the weighted Jacobi scheme on the fine grid, the magnitudes of the coefficients have been reduced by more than an order of magnitude from their values after the first step.

Remark 1: In the computations performed above, for each iteration of the coarse grid correction scheme, we used three iterations of the weighted Jacobi scheme at the beginning of the step and three iterations of the weighted Jacobi scheme at the end of the step. It would be possible to use fewer than three in either of these places or both of these places. It would also be possible to do an experiment to try to determine how many weighted Jacobi iterations are optimal in each place. We have not done this. We chose three iterations fairly arbitrarily. The intent of this example was to illustrate the reduction of the error in the high and low frequency modes and the reintroduction of the error in the high frequency modes.

Remark 2: We performed the second iteration of the coarse grid correction scheme both to show how the different modes of the error react and to illustrate that we can obtain more accuracy by repeated application of the scheme. It is always important to try to monitor the amount of work necessary to solve using the multigrid and multigrid-like schemes. We are always interested in comparisons between the work necessary to solve a problem by multigrid compared to the work that would be necessary to solve the same problem by other schemes. Of course, a small problem like the one done above is not especially relevant. However, in using two iterations of the coarse grid correction scheme,

we used 12 iterations on the fine grid. To solve the problem to the equivalent degree of accuracy using just the weighted Jacobi scheme would take 120 iterations. However, we must account for the work done on the coarse grid. The coarse grid does not have many points and does not require much work (in this case, the coarse grid had three interior points). We must also account for the work due to the grid transfers. We will return to this topic later.

Remark 3: We saw above that the interpolation back to the fine grid introduces high frequency error. In HW10.10.3 we see that even when there are no high frequency components in the initial error, there are high frequency components in the interpolated error that is added to $\mathbf{u}_{m_1}^h$. The relaxation steps on the fine grid taken after the interpolation are very important in that they eliminate this high frequency interpolation error.

In an attempt to separate the weighted Jacobi iterations from the coarse grid correction, we considered using the scheme with no relaxation, i.e.,

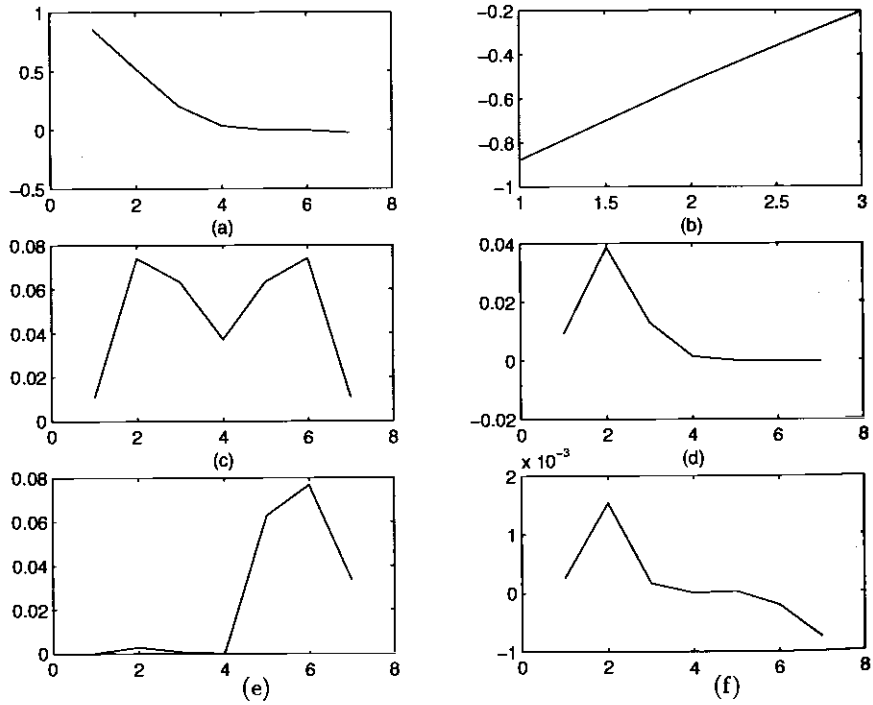

FIGURE 10.10.4. All of the plots given in this figure are plots of the coefficients with respect to the eigenvectors of the matrix A^h or A^{2h}. We have $M = 8$, $m_1 = 3$, and $m_2 = 3$. (a) Plot of the coefficients of $\mathbf{u}_{m_1}^h$. (b) Plot of coefficients of $\mathbf{e}^{2h}$. (c) Plot of the coefficients of $\hat{\mathbf{u}}_{m_1}^h$ after the first iteration of the coarse grid correction scheme. (d) Plot of the coefficients of $\hat{\mathbf{u}}_{m_2}^h$ after the first iteration of the coarse grid correction scheme. (e) Plot of the coefficients of $\hat{\mathbf{u}}_{m_1}^h$ after the second iteration of the coarse grid correction scheme. (d) Plot of the coefficients of $\hat{\mathbf{u}}_{m_2}^h$ after the second iteration of the coarse grid correction scheme.

choose $m_1 = m_2 = 0$. Then

$$c_j = \frac{1}{\mu_j^{2h}}\left(\mu_{M-j}^h \sin^2 \frac{j\pi}{2M} - \mu_j^h \cos^2 \frac{j\pi}{2M}\right)$$

and

$$\hat{\mathbf{u}}^h = \hat{\mathbf{u}}_0^h = \sum_{j=1}^{M/2-1} \left[1 + c_j \cos^2 \frac{j\pi}{2M}\right] \mathbf{w}_j^h + \mathbf{w}_{M/2}^h + \sum_{j=M/2+1}^{M-1} \left[1 - c_{M-j} \cos^2 \frac{j\pi}{2M}\right] \mathbf{w}_j^h. \qquad (10.10.62)$$

An easy computation shows that $c_j = 0$ for $j = 1, \ldots, M/2 - 1$ and $\hat{\mathbf{u}}^h = \mathbf{u}_0^h$. There is no change due to the coarse grid correction. The reason for this result illustrates a potential difficulty associated with the coarse grid correction scheme. We saw earlier that $I_h^{2h}\mathbf{w}_{M/2}^h = \boldsymbol{\theta}^{2h}$. There are more vectors that get mapped onto the zero vector. The operator I_h^{2h} is a 3×7 matrix with rank 3. Hence, the null space of I_h^{2h} has dimension four. Four independent vectors that are annihilated by I_h^{2h} are $\mathbf{w}_{M/2}^h$, $\mathbf{w}_1^h + \mathbf{w}_7^h$, $\mathbf{w}_2^h + \mathbf{w}_6^h$, and $\mathbf{w}_3^h + \mathbf{w}_5^h$. Thus we see that $I_h^{2h}\mathbf{u}_0^h = \boldsymbol{\theta}^{2h}$. If the residual or any part of the residual is in the null space of I_h^{2h}, the error due to that part of the residual will not be corrected on the coarse grid. We should realize that when any number of relaxation steps have been performed, the chances of the residual having significant components contained in the null space is very small.

Two Dimensional Model: Coarse Grid Correction Scheme

It should be clear that we could build an analytic version of the coarse grid correction scheme for our two dimensional model problem analogous to what we have done for the one dimensional problem above. It would be messier and would probably not teach us anything new. It is time to implement the coarse grid correction scheme (see HW10.10.6, HW10.10.7). The multigrid scheme that we will soon be writing will be very similar to the algorithm for the coarse grid correction scheme. When implementing the coarse grid correction scheme, use subroutines for the relaxation schemes, the residual calculation, and both grid transfers that can be used on a general grid. This degree of generality will make it somewhat easier to expand the coarse grid correction algorithm to a multigrid algorithm.

To illustrate that the two dimensional coarse grid correction scheme has the same effect on the low and high frequency components of the error as does the one dimensional scheme, we apply the scheme to solve the two dimensional model computational problem. We use $M = 16$ and three iterations of the weighted Jacobi scheme as our smoother. In Figure 10.10.5 we plot the coefficients of the approximate solution after one iteration of

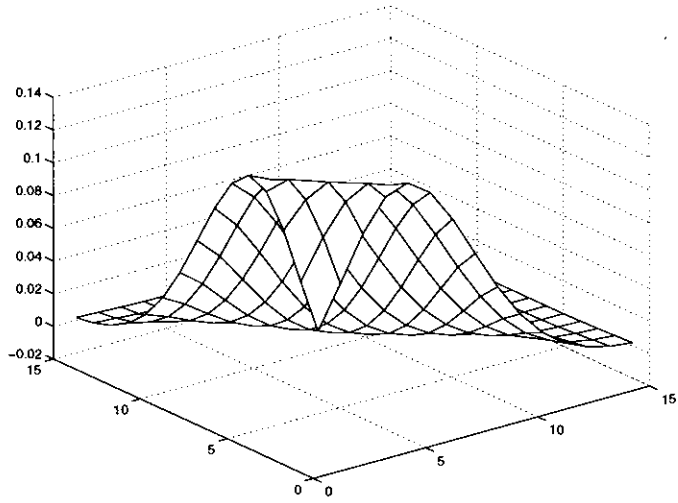

FIGURE 10.10.5. Plot of the coefficients (with respect to an expansion in terms of the eigenvectors of A) of the approximate solution to model problem (10.2.3)–(10.2.7) (with F_{jk} and $f_{jk} = 0$ for all j and k) after one iteration of the coarse grid correction scheme. The scheme uses $\mathbf{u}_0$, (10.10.1), as the initial guess.

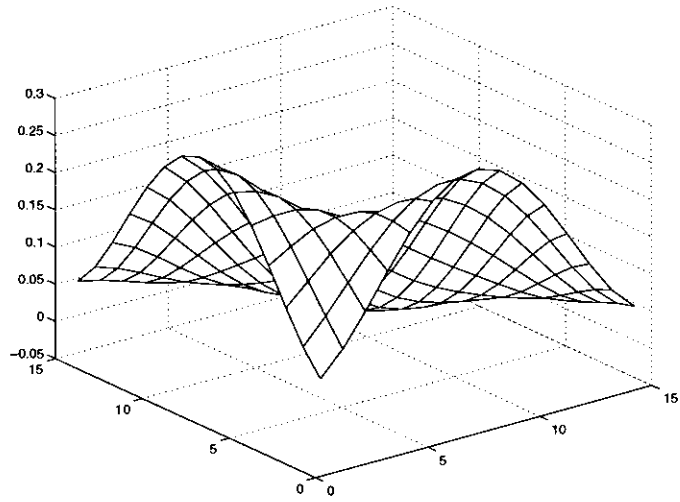

FIGURE 10.10.6. Plot of the coefficients (with respect to an expansion in terms of the eigenvectors of A) of the approximate solution after the coarse grid correction has been interpolated to the fine grid and added to the fine grid solution.

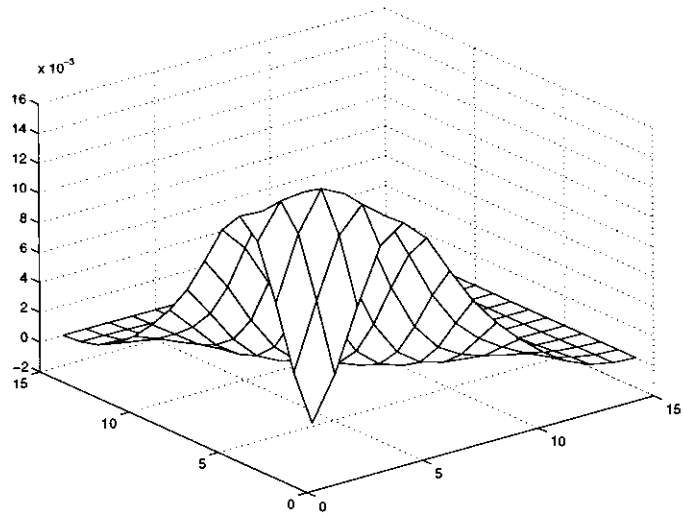

FIGURE 10.10.7. Plot of the coefficients (with respect to an expansion in terms of the eigenvectors of A) of the approximate solution to model problem (10.2.3)–(10.2.7) (with $F_{j\,k}$ and $f_{j\,k} = 0$ for all j and k) after two iterations of the coarse grid correction scheme. The scheme uses $\mathbf{u}_0$, (10.10.1), as the initial guess.

the coarse grid correction scheme. As we have done before, the coefficients are those obtained by expanding the approximate solution in terms of the eigenvectors of A, $\mathbf{w}^{p\,s}$, $p, s = 1, \ldots, 15$. We see that in one coarse grid correction step, the coefficients have been reduced from one to approximately 0.1 (or less). As we should expect, we see that the high frequency components of the error have been eliminated. Earlier, in Figure 10.10.2, we saw how the first three iterations of the weighted Jacobi scheme virtually eliminated the high frequency components of the error and did little damage to the low frequency components of the error. In Figure 10.10.6 we see that, as in the one dimensional case, after the coarse grid correction has been interpolated to the fine grid and added to the solution on that grid, the magnitude of the low frequency components of the error has been reduced, but the high frequency components have been reintroduced. In Figure 10.10.7 we include a plot after the second coarse grid correction iteration. We note that the coefficients have been further reduced significantly. We note that the solution associated with the error given in Figure 10.10.7 required nine relaxations on the fine grid, since we have just relaxed three times on the fine grid at the end of the first iteration of the coarse grid correction scheme, we do not relax three times at the beginning of the second iteration of the coarse grid correction scheme (plus the work necessary to solve the problems on the coarse grid and the work necessary for the intergrid transfers). To solve the model computational problem to the

same degree of accuracy using just the weighted Jacobi scheme, we must use 127 iterations of the weighted Jacobi scheme. Solving the problem to this same degree of accuracy using the Gauss-Seidel scheme requires 27 iterations.

We see that the two dimensional coarse grid correction scheme works very similarly to the one dimensional scheme. Also, we see that we can help solve our problem by working on the coarse grid. We should emphasize that the savings on the two dimensional grids (where the fine grid has 225 interior points and the coarse grid has 49 interior points) are much greater than on the one dimensional problem (where the fine grid has 7 interior points and the coarse grid has 3 interior points). It should be clear that the savings will be greater yet if we apply the coarse grid correction scheme to larger two dimensional problems or three dimensional problems.

HW 10.10.3 Repeat the calculations performed in Example 10.10.1 using $\mathbf{u}_0^h = \mathbf{w}_1^h + \mathbf{w}_2^h + \mathbf{w}_3^h$ as the initial guess.

HW 10.10.4 Repeat the two dimensional coarse grid correction calculation done above, using as many iterations of the coarse grid correction scheme that are necessary to reduce the error below a tolerance of 10^{-5} (measured in the sup-norm).

HW 10.10.5 Repeat the calculation done in HW10.10.4, using the Gauss-Seidel scheme in place of the weighted Jacobi scheme. Compare and contrast your results with those in HW10.10.4.

HW 10.10.6 Use the coarse grid correction scheme to solve the following boundary–value problem.

$$\begin{aligned} \nabla^2 v &= 0, \quad (x,y) \in (0,1) \times (0,1) \\ v(x,0) &= 1 - x^2, \quad x \in [0,1] \\ v(x,1) &= -x^2, \quad x \in [0,1] \\ v(0,y) &= 1 - y^2, \quad y \in [0,1] \\ v(1,y) &= -y^2, \quad y \in [0,1] \end{aligned}$$

Use $M_x = M_y = 16$ and the weighted Jacobi scheme as the iterative scheme. Use the same stopping criterion and tolerance used when this problem was solved by the Jacobi, Gauss-Seidel, and SOR schemes. Rerun the codes used in HW10.5.5, HW10.5.9, and HW10.5.14 using $M_x = M_y = 16$. Compare and contrast these results with those found by the coarse grid correction scheme.

HW 10.10.7 (a) Use the coarse grid correction scheme to find an approximate solution to the problem

$$\nabla^2 v = e^{x+y}, \quad (x,y) \in R = (0,1) \times (0,1)$$
$$v = -e^{1-x-y} \quad \text{on } \partial R.$$

Use $M_x = M_y = 128$, a tolerance of $= 1.0 \times 10^{-6}$, and the weighted Jacobi scheme as your iterative scheme. Compare your results to those found in HW10.5.6, HW10.5.10 and HW10.5.14 (where you rerun your results in HW10.5.6, HW10.5.10 and HW10.5.14 using $M_x = M_y = 128$).
(b) Repeat part (a) using Gauss-Seidel as the iterative scheme.

10.10.4.2 V-Cycle

We are now ready to turn to the major topic of this section, a multigrid scheme. We include only one multigrid scheme, the V-cycle. The emphasis of including multigrid in this chapter is not to make the reader a multigrid expert. The hope is that we can introduce multigrid to give the reader an understanding of how and why the multigrid schemes work. For the reader who needs more, see [8] or the other references mentioned earlier.

To motivate the V-cycle, we return to our discussion of the coarse grid correction scheme. The fourth step of the coarse grid correction scheme is to "Solve $A^{2h}\mathbf{e}^{2h} = \mathbf{f}^{2h}$ on $\mathcal{G}^{2h}$." In our one dimensional example, we were able to solve the coarse grid problem analytically because we were considering a trivial problem. In our two dimensional calculations, we never mentioned how we solved the coarse grid problem. Actually, we used many steps of the weighted Jacobi scheme, using the weighted Jacobi scheme only because it was already there. There are times that it is sufficient to use the coarse grid correction scheme and just solve the problem on the coarse grid, using "the best scheme available" rather than some scheme that happens to be "already there." Generally, it is not sufficient and surely not optimal to proceed in this manner. The coarse grid is much smaller than the fine grid, but the solution on the coarse grid is still expensive.

One approach is to consider using the coarse grid correction scheme to approximate the solution on the $\mathcal{G}^{2h}$ grid. Specifically, we would replace the fourth step of the coarse grid correction scheme by a coarse grid correction iteration and obtain the following algorithm.

Coarse Grid Correction^2 Scheme
Relax m_1 times on $A^h\mathbf{u}^h = \mathbf{f}^h$ on $\mathcal{G}^h$ with initial guess $\mathbf{u}_0^h$, result $\mathbf{u}_{m_1}^h$.
Compute $\mathbf{r}^h = \mathbf{f}^h - A^h\mathbf{u}_{m_1}^h$.
Compute $\mathbf{f}^{2h} = I_h^{2h}\mathbf{r}^h$.
  Relax m_1 times on $A^{2h}\mathbf{e}^{2h} = \mathbf{f}^{2h}$ on $\mathcal{G}^{2h}$ with initial guess $\mathbf{e}_0^{2h} = \boldsymbol{\theta}$, result $\mathbf{e}_{m_1}^{2h}$.
  Compute $\mathbf{r}^{2h} = \mathbf{f}^{2h} - A^{2h}\mathbf{e}_{m_1}^{2h}$.

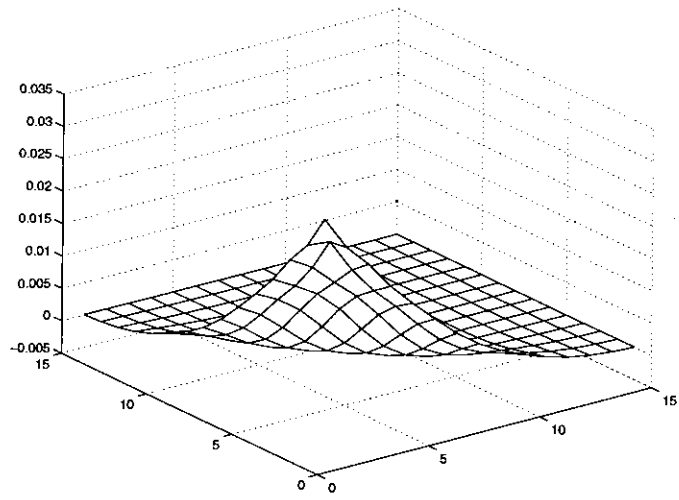

FIGURE 10.10.8. Plot of the coefficients (with respect to an expansion in terms of the eigenvectors of A) of the approximate solution to model problem (10.2.3)–(10.2.7) (with $F_{j\,k}$ and $f_{j\,k} = 0$ for all j and k) after two iterations of the V-cycle scheme. The scheme uses $\mathbf{u}_0$, (10.10.1), as the initial guess, the weighted Jacobi scheme as the smoother, and $m_1 = m_2 = 3$.

Compute $\mathbf{f}^{4h} = I_{2h}^{4h}\mathbf{r}^{2h}$.
  Solve $A^{4h}\mathbf{e}^{4h} = \mathbf{f}^{4h}$ on $\mathcal{G}^{4h}$.
Correct on $\mathcal{G}^{2h}$, $\hat{\mathbf{e}}_{m_1}^{2h} = \mathbf{e}_{m_1}^{2h} + I_{4h}^{2h}\mathbf{e}^{4h}$.
Relax m_2 times on $A^{2h}\mathbf{u}^{2h} = \mathbf{f}^{2h}$ on $\mathcal{G}^{2h}$ with initial guess $\mathbf{e}_0^{2h} = \hat{\mathbf{e}}_{m_1}^{2h}$, result $\hat{\mathbf{e}}_{m_2}^{2h}$.
Correct fine grid approximation $\hat{\mathbf{u}}_{m_1}^{h} = \mathbf{u}_{m_1}^{h} + I_{2h}^{h}\hat{\mathbf{e}}_{m_2}^{2h}$.
Relax m_2 times on $A^h\mathbf{u}^h = \mathbf{f}^h$ on $\mathcal{G}^h$ with initial guess $\hat{\mathbf{u}}_{m_1}^{h}$, result $\hat{\mathbf{u}}_{m_2}^{h}$.

We see that we have replaced a simple step "Solve $A^{2h}\mathbf{e}^{2h} = \mathbf{f}^{2h}$ on $\mathcal{G}^{2h}$" by six steps. This does not make the algorithm appear simpler. More importantly, we have replaced the "simple step" of the coarse grid correction scheme by six steps that will require m_1 relaxations on $\mathcal{G}^{2h}$, a residual calculation on $\mathcal{G}^{2h}$, a grid transfer from $\mathcal{G}^{2h}$ to $\mathcal{G}^{4h}$, a solve on $\mathcal{G}^{4h}$, a grid transfer from $\mathcal{G}^{4h}$ to $\mathcal{G}^{2h}$, and m_2 relaxations on $\mathcal{G}^{2h}$. It should be reasonably clear that if this procedure works sufficiently well (and if it didn't, we wouldn't introduce it), we obtain a cheap way to obtain an approximate solution to the equation $A^{2h}\mathbf{e}^{2h} = \mathbf{f}^{2h}$.

Both the one dimensional model problem and two dimensional model problems considered in the last section can be solved using the coarse grid correction^2 scheme. We leave this to the reader in HW10.10.8 and HW10.10.9. The idea that we want to pursue should now be clear. The difference between the coarse grid correction scheme and the coarse grid

correction[2] scheme is that we replaced solving the problem exactly on the $\mathcal{G}^{2h}$ grid by using the coarse grid correction scheme on $\mathcal{G}^{2h}$ to approximate that solution. The multigrid approach is to return to the original coarse grid correction scheme, replace the solution on the $\mathcal{G}^{2h}$ grid by an application of the coarse grid correction scheme, replace the resulting solution on the $\mathcal{G}^{4h}$ grid by an application of the coarse grid correction scheme, ..., until the grid on which we must solve contains only one interior point. At this time we solve the equation and begin the corrections. We state the following algorithm for the multigrid V-cycle.

V-Cycle

Relax m_1 times on $A^h\mathbf{u}^h = \mathbf{f}^h$ on $\mathcal{G}^h$ with initial guess u_0^h, result $\mathrm{u}_{m_1}^h$.
Compute $\mathbf{r}^h = \mathbf{f}^h - A^h\mathrm{u}_{m_1}^h$.
Compute $\mathbf{f}^{2h} = I_h^{2h}\mathbf{r}^h$.
Relax m_1 times on $A^{2h}\mathbf{e}^{2h} = \mathbf{f}^{2h}$ on $\mathcal{G}^{2h}$ with initial guess $\mathbf{e}_0^{2h} = \boldsymbol{\theta}$, result $\mathbf{e}_{m_1}^{2h}$.
Compute $\mathbf{r}^{2h} = \mathbf{f}^{2h} - A^{2h}\mathbf{e}_{m_1}^{2h}$.
Compute $\mathbf{f}^{4h} = I_{2h}^{4h}\mathbf{r}^{2h}$.
...
...
Compute $\mathbf{f}^{2^{p-1}h} = I_{2^{p-2}h}^{2^{p-1}h}\mathbf{r}^{2^{p-2}h}$.
Solve $A^{2^{p-1}h}\mathbf{e}^{2^{p-1}h} = \mathbf{f}^{2^{p-1}h}$.
Correct on $\mathcal{G}^{2^{p-2}h}$, $\hat{\mathbf{e}}_{m_1}^{2^{p-2}h} = \mathbf{e}_{m_1}^{2^{p-2}h} + I_{2^{p-1}h}^{2^{p-2}h}\mathbf{e}^{2^{p-1}h}$.
Relax m_2 times on $A^{2^{p-2}h}\mathrm{u}^{2^{p-2}h} = \mathbf{f}^{2^{p-2}h}$ on $\mathcal{G}^{2^{p-2}h}$ with initial guess $\mathbf{e}_0^{2^{p-2}h} = \hat{\mathbf{e}}_{m_1}^{2^{p-2}h}$, result $\hat{\mathbf{e}}_{m_2}^{2^{p-2}h}$.
Correct on $\mathcal{G}^{2^{p-3}h}$, $\hat{\mathbf{e}}_{m_1}^{2^{p-3}h} = \mathbf{e}_{m_1}^{2^{p-3}h} + I_{2^{p-2}h}^{2^{p-3}h}\hat{\mathbf{e}}_{m_2}^{2^{p-2}h}$.
...
...
Correct on $\mathcal{G}^{2h}$, $\hat{\mathbf{e}}_{m_1}^{2h} = \mathbf{e}_{m_1}^{2h} + I_{4h}^{2h}\mathbf{e}^{4h}$.
Relax m_2 times on $A^{2h}\mathbf{u}^{2h} = \mathbf{f}^{2h}$ on $\mathcal{G}^{2h}$ with initial guess $\mathbf{e}_0^{2h} = \hat{\mathbf{e}}_{m_1}^{2h}$, result $\hat{\mathbf{e}}_{m_2}^{2h}$.
Correct fine grid approximation $\hat{\mathrm{u}}_{m_1}^h = \mathrm{u}_{m_1}^h + I_{2h}^h\hat{\mathbf{e}}_{m_2}^{2h}$.
Relax m_2 times on $A^h\mathbf{u}^h = \mathbf{f}^h$ on $\mathcal{G}^h$ with initial guess $\hat{\mathbf{u}}_{m_1}^h$, result $\hat{\mathbf{u}}_{m_2}^h$.

We note that when $M = 16 = 2^4$, there are four grids involved, $\mathcal{G}^h$, $\mathcal{G}^{2h}$, $\mathcal{G}^{4h} = \mathcal{G}^{2^{p-2}h} = \mathcal{G}^{2^2h}$, and $\mathcal{G}^{2^{p-1}h} = \mathcal{G}^{2^3h} = \mathcal{G}^{8}$. In Figure 10.10.8 we plot the coefficients of the approximation to our two dimensional model problem after two V-cycles (as usual, the coefficients of the expansion of the approximate solution in terms of the eigenvectors $\mathbf{w}^{p\,s}$, $p, s = 1, \ldots, 16$). Comparing Figure 10.10.8 with Figure 10.10.7, we see that two V-cycles do not do as well as two iterations of the coarse grid correction scheme. Since the V-cycle is an approximation of the coarse grid correction scheme, this should not be surprising. We emphasize that the solution given in Figure 10.10.8 costs 9 iterations on the 15×15 grid, 12 iterations on the 7×7

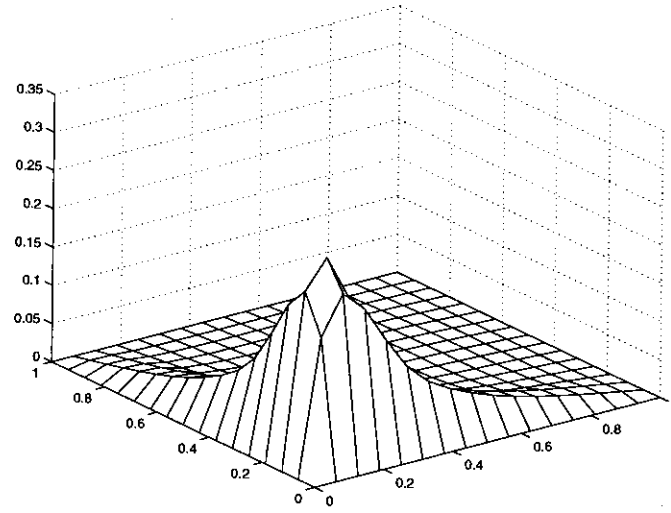

FIGURE 10.10.9. Plot of the approximate solution to model problem (10.2.3)–(10.2.7) (with F_{jk} and $f_{jk} = 0$ for all j and k) after two iterations of the V-cycle scheme. The scheme uses $\mathbf{u}_0$, (10.10.1), as the initial guess, the weighted Jacobi scheme as the smoother, and $m_1 = m_2 = 3$.

grid, 12 iterations on the 3×3 grid (plus the residual calculations and the grid transfers; see Remark, page 441), whereas the solution given in Figure 10.10.7 cost us 12 iterations on the 15×15 grid and hundreds of iterations on the 7×7 grid (plus two residual calculations and four grid transfers). In Figure 10.10.9 we plot the solution after two V-cycles. It is clear that this solution is not good enough yet, but it should be noted that from our terrible starting guess, the error has been reduced by approximately three orders of magnitude in the sup-norm (from over 100 to less than 0.2).

For a comparison, in Figure 10.10.10 we include the approximate solution to the two dimensional model problem after two V-cycles using Gauss-Seidel as the iterative scheme. We see that the V-cycle using Gauss-Seidel gives a much better solution. Since the Gauss-Seidel scheme is a much better iterative scheme than the weighted Jacobi scheme, this should not surprise us. In this case we see that the two V-cycles have reduced the error by approximately five orders of magnitude.

HW 10.10.8 Apply the coarse grid correction^2 scheme to the one dimensional model problem analogously to the way we applied the coarse grid correction scheme to the one dimensional model problem in Section 10.10.4.1.

HW 10.10.9 Apply the coarse grid correction^2 scheme to the two dimensional model problem analogously to the way we applied the coarse grid correction scheme to the one dimensional model problem in Section 10.10.4.1.

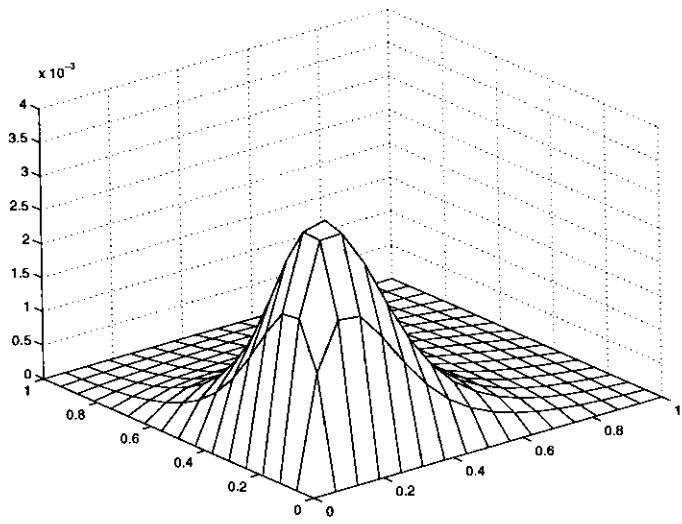

FIGURE 10.10.10. Plot of the approximate solution to model problem (10.2.3)–(10.2.7) (with $F_{j\,k}$ and $f_{j\,k} = 0$ for all j and k) after two iterations of the V-cycle scheme. The scheme uses $\mathbf{u}_0$, (10.10.1) as the initial guess, the Gauss-Seidel scheme as the smoother and $m_1 = m_2 = 3$.

Remark 1: Several times, we have referred to the low cost of the multigrid scheme compared to that of a straight iterative scheme. If we consider that we have $2^j 2^j = 2^{2j}$ points on grid $\mathcal{G}^{2^{p-j}h}$ (we actually have $(2^j+1)(2^j+1)$ points and for the Dirichlet problems we have considered, we work with only $(2^j-1)(2^j-1)$), then we perform $(m_1+m_2)2^{2j}$ relaxations on the $\mathcal{G}^{2^{p-j}h}$ grid and

$$(m_1+m_2)\sum_{j=1}^{p} 2^{2j} = (m_1+m_2)\frac{4}{3}\left(2^{2p}-1\right)$$

relaxation steps on one V-cycle (where we count the exact solution on the $\mathcal{G}^{0.5}$ grid as a relaxation, which it is for a Dirichlet problem). We note that we can logically sum over all of the points, since each relaxation step, no matter on which grid we consider, takes the same number of arithmetic operations. If we use the usual definition of a **work unit** as one relaxation step on the $\mathcal{G}^h$ grid, we see that one V-cycle consists of $(m_1+m_2)\frac{4}{3}(1-2^{-2p})$ work units. This value is often approximated by $(m_1+m_2)\frac{4}{3}$.

Inspecting the full weighting and linear interpolation grid transfers, it is reasonable to approximate the work involved by the grid transfers by a relaxation on the finest level involved. Likewise, the work involved in a residual calculation is also equivalent to the work involved in a relaxation step. Thus we see that the work due to one V-cycle step can be approximated by $\frac{4}{3}(m_1+m_2+3)$ work units. For example, in our calculations

where we used $m_1 = m_2 = 3$, we see that the cost of each V-cycle is approximately 12 relaxation steps on the fine grid.

Remark 2: In the last remark we approximated the work necessary to perform each V-cycle to be 12 work units. We could do some timings to evaluate the accuracy of the estimate. Actually, the problems that we have considered are much too small to use for this purpose. At this time multigrid does not appear to be much more efficient than some of the other iterative schemes. Earlier, we found that it took only 127 iterations of the weighted Jacobi scheme and 27 Gauss-Seidel iterations to solve the two dimensional model problem to the same accuracy as two iterations of the coarse grid correction scheme (which does better than two V-cycles—which using the computation done above would imply that two V-cycles costs the equivalent of 24 iterations on the fine grid). Besides the fact that we have been using a terrible initial guess (the initial guess was terrible for all of the schemes), we must understand that a 16×16 problem is a "pretend" problem. No one will ever pay you to solve such a problem. Consider the boundary–value problem

$$\nabla^2 v = e^{x+y}, \quad (x, y) \in R = (0, 1) \times (0, 1) \tag{10.10.63}$$

$$v = 0, \quad (x, y) \text{ on } \partial R. \tag{10.10.64}$$

This problem is similar to a problem that we solved earlier, except that we now have zero boundary conditions (to make it a bit easier for me to code). If we use $M_x = M_y = 128$ and a stopping criterion consisting of the sup-norm of the difference of successive iterates and a tolerance of 1.0×10^{-6}, we see that it will take 14,128 iterations of the Jacobi scheme, 8,222 iterations of the Gauss-Seidel scheme, and 273 iterations of the SOR scheme with the optimal parameter to find an approximate solution to this problem. We see that to approximate the solution to this problem by our multigrid scheme using the weighted Jacobi scheme with $m_1 = m_2 = 3$ and the same convergence criterion, it takes 3 V-cycles. Returning to the analysis done in Remark 1, we see that even with $M_x = M_y = 128$, it still takes only 12 work units per V-cycle. Hence, these 3 V-cycles are equivalent to the work of 36 iterations of the other schemes. If we repeat the multigrid computation using $m_1 = m_2 = 2$, we see that it still takes only three V-cycles. In this case the cost is only 9.3 work units, which is equivalent to 28 iterations of the other schemes. We see that the multigrid scheme solves the problem much more cheaply than either the Jacobi, Gauss-Seidel, or the optimal SOR schemes (without having to choose an optimal parameter). We might note that we could use a problem this size to check to see whether the work computations done in Remark 1 were reasonably accurate. We leave this to the reader in HW10.10.10.

Remark 3: In Section 10.5.4 we discussed that we should be careful not to over-solve our discrete problem. Return to the boundary–value problem discussed in HW10.5.7. If we use a tolerance of 1.0×10^{-10} and optimal

SOR, it will take 597 iterations to determine our "exact solution." Using the known solution of the analytic problem, we find that the sup-norm of the truncation error is 0.0072. We then set the tolerance equal to this value, 0.0072, and resolve the discrete problem using our multigrid scheme with $m_1 = m_2 = 2$ and the weighted Jacobi scheme. We use the sup-norm of the difference between successive iterates as our stopping criterion (mainly because that is what we used in Section 10.5.4). It takes only two V-cycles to converge to the tolerance of 0.0072. However, we must be careful. As usual, a stopping criterion such as the sup-norm of the difference of two successive iterates is only an approximation. We saw in Section 10.5.4 that in this case using the sup-norm of the truncation error, 0.0072, as the tolerance is not good enough. If we check the error, we find that the error is much larger than twice the truncation error. Repeating the computation several times, we see that it takes four V-cycles before the error in the computed solution is less than or equal to the truncation error. It also takes four V-cycles before the error in the computed solution is less than or equal to twice the truncation error. We might use the excuse that this problem is again difficult because the solution is large and we start with the zero solution, but the problem is a good one, and it is a common approach to choose zero as the initial guess. We do not have an approach to choose a better initial guess. This problem shows that you must be careful how you decide that your multigrid scheme has converged. You can use the idea that you only want to converge to within truncation error, but you still need to use a safety factor when choosing your tolerance.

We note that it takes 346 optimal SOR iterations to solve the discrete problem associated with HW10.5.7 to within truncation error. The four V-cycles discussed above take 9.3 work units per V-cycle, or the equivalent of 37 iterations, to solve to the same accuracy. And if we apply our multigrid scheme with $m_1 = m_2 = 3$, the sup-norm of the difference between successive iterates is less than 0.0072 in one iteration, but it still takes four V-cycles (equivalent to 48 iterations) to reduce the sup-norm of the error to be less than 0.0072.

Remark 4: We next comment on the implementation of the V-cycle multigrid scheme. A naive implementation (and I know because I have done one or more naive multigrid implementations) can make the process very difficult. Probably the best implementation is to use what has become known as the "multigrid storage," which uses two one dimensional arrays and pointers. One array is used for the u values and one array is used for the right hand sides. The arrays must be large enough to include u and F on all of the grids. For example, for the case with $M_x = M_y = 16$, we fill the first 289 elements of the F array with the computed values of the right hand side (and any appropriate boundary data that must be added in) and the first 289 elements of the u array with the values of the solution after m_1 relaxation steps on grid $\mathcal{G}^h$. We should realize that to use a relaxation

scheme such as the weighted Jacobi scheme, we will need a temporary *uold* array.

We next perform the second and third steps of the V-cycle algorithm together and place $I_h^{2h}(f^h - A^h\mathbf{u}^h)$ into slots 290 to 370 of the F array. Then, after m_1 relaxations on the $\mathcal{G}^{2h}$ grid, the result is stored in elements 290 to 370 of the u array. This process is continued until we solve exactly on the $\mathcal{G}^{0.5}$ grid. Each step on the way back to the $\mathcal{G}^h$ grid consists of adding the interpolated solution to the solution on the next grid, zeroing out u on the present grid (in preparation for the next V-cycle), and performing m_2 relaxation steps on the next grid. The $\mathbf{e}_{m_1}^{2^{p-j}h} + I_{2^{p-j-1}h}^{2^{p-j}h}\hat{\mathbf{e}}_{m_2}^{2^{p-j-1}h}$ term must be placed in the temporary array *uold* (as the initial guess); the F array has not been changed, since it was filled earlier, and the result after m_2 iterations is stored in the appropriate area of the u array—overwriting the old u values.

Since we have the mechanism for computing the residual already available, it is most logical to use the residual as the stopping criterion. We emphasize that all of the care suggested in Remark 3 above and in Section 10.5.4 must still be used in choosing the tolerance associated with using the residual as the stopping criterion.

And finally, we mention that in most programming languages if the values of u and F are stored in a j-k ordering in the one dimensional arrays, the arrays can be considered as two dimensional arrays in j-k ordering in the relaxation, residual calculation, and grid transfer routines.

Remark 5: Most of the work done in this section has been devoted to two dimensional problems. Of course, the coarse grid correction scheme and the V-cycle can be used in more than two dimensions. It should not be surprising that the savings are greater and more important when multigrid is used on problems in three or more dimensions. Technically, very little changes between two or three (or more) dimensional problems. Of course, we consider the usual three dimensional discretization on the appropriate grid rather than the two dimensional discrete problems. The full weighting and the linear interpolation are more complex, but they still represent a weighted average and simple linear interpolation, respectively. The difficulty of these grid transfers is a good reason to use injection as the restriction operator when we work in three or more dimensions. And there is a substantial savings in computational costs if multigrid is used rather than some of the other iterative schemes. An analysis similar to that done in Remark 1 can be done to see that a V-cycle in m dimensions with m_1 smoothing iterations on the descent and m_2 smoothing iterations on the ascent will cost approximately $(m_1 + m_2 + 3)2^m/(2^m - 1)$ work units. Since the more traditional iterative methods slow down terribly in three or more dimensions, this value makes the multigrid scheme very efficient for such problems.

Remark 6: When we consider the original problem on the $\mathcal{G}^h$ grid, we

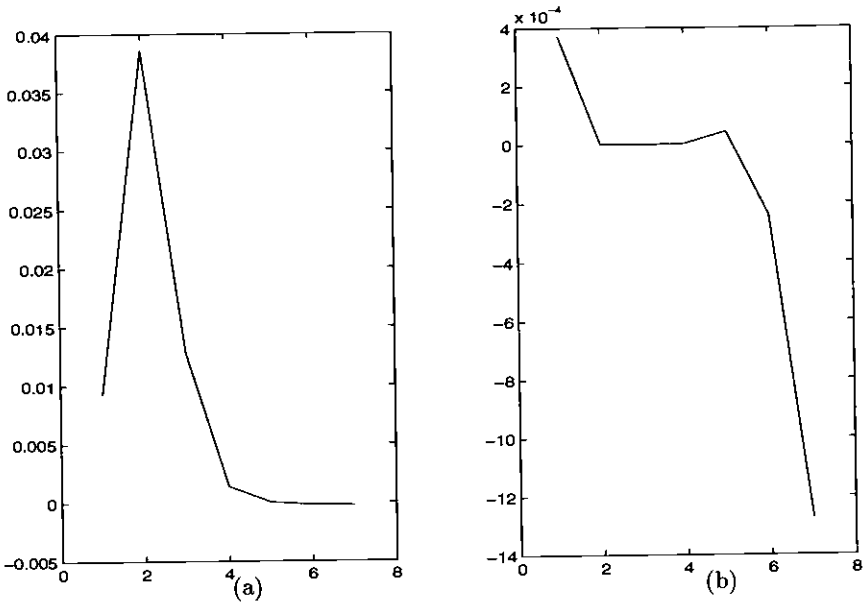

FIGURE 10.10.11. The plots given in this figure are plots of the coefficients with respect the to the eigenvectors of the matrix A^h. We have $M = 8$, $m_1 = 3$ and $m_2 = 3$. (a) Plot of the coefficients of $\hat{\mathbf{u}}^h_{m_2}$ after the first iteration of the coarse grid correction scheme. (b) Plot of the coefficients of $\hat{\mathbf{u}}^h_{m_2}$ after the second iteration of the coarse grid correction scheme.

consider a usual second order discretization (any high order schemes could be used). When we transfered the residual to the $\mathcal{G}^{2h}$ grid, we let A^{2h} be the matrix associated with the usual second order discretization on the $\mathcal{G}^{2h}$ grid. This approach was not wrong but was probably misleading. It is reasonably clear after we write it down that the coarse grid operator should be

$$A^{2h} = I_h^{2h} A^h I_{2h}^h. \tag{10.10.65}$$

If the grid transfers are used to transfer the data, why should they not also be used to transfer the operator? The grid transfer on the right, I_{2h}^h, takes a vector defined on the $\mathcal{G}^{2h}$ grid and transforms it to a vector defined on the $\mathcal{G}^h$ grid. Then A^h operates on this vector, and I_h^{2h} maps the result back to a vector on the $\mathcal{G}^{2h}$ grid. With this definition, the operator A^{2h} is consistent with the operator A^h and the grid transfers. The approach of using (10.10.65) to define the coarse grid operators is referred to the **Galerkin formulation**.

The good news is that if we use (10.10.65) to define our coarse grid operators along with full weighting and linear interpolation, we will obtain the usual approximations of ∇^2 on the coarse grids that we have been using. It is easiest to perform this calculation for the one dimensional problem.

See HW10.10.11.

An easy calculation in the one dimensional case shows that if $\mathcal{I}_h^{2h}$ is injection, A^h is as given in (10.10.32), and I_{2h}^h is linear interpolation, then

$$\mathcal{I}_h^{2h} A^h I_{2h}^h = \mathcal{A}^{2h} = 2A^{2h},$$

where A^{2h} is as given in (10.10.33). It is not clear that using $\mathcal{A}^{2h}$ is bad, but since it does not approximate the differential operator on the $\mathcal{G}^{2h}$ grid, we should at least be suspicious. In Section 10.10.4.1 we analytically applied the coarse grid correction scheme to our one dimensional model problem and arrived at the approximate solution given by (10.10.61). If we repeat the calculations done in Section 10.10.4.1 with the full weighting restriction operator replaced by the injection operator, we obtain the following results. We first note that

$$\mathcal{I}_h^{2h}\mathbf{w}_j^h = \begin{cases} \mathbf{w}_j^{2h} & \text{when } 1 \le j \le M/2-1 \\ \boldsymbol{\theta} & \text{when } j = M/2 \\ -\mathbf{w}_{M-j}^{2h} & \text{when } M/2+1 \le j \le M-1. \end{cases} \tag{10.10.66}$$

These results are not that different from those obtained using the full weighting operator. We now get

$$\begin{aligned} \mathbf{f}_j^{2h} &= -\left(\lambda_j^h\right)^{m_1} \mu_j^h \mathbf{w}_j^{2h} \\ c_j' &= -\left(\lambda_j^h\right)^{m_1} \frac{\mu_j^h}{\mu_j^{2h}} \\ \mathbf{e}_j^{2h} &= -\left(\lambda_j^h\right)^{m_1} \frac{\mu_j^h}{\mu_j^{2h}} \mathbf{w}_j^{2h} \end{aligned}$$

for $j = 1, \ldots, M/2-1$,

$$\mathbf{f}_{M/2}^{2h} = \boldsymbol{\theta}, \quad c_{M/2} = 0, \quad \mathbf{e}_{M/2}^{2h} = \boldsymbol{\theta}$$

and

$$\begin{aligned} \mathbf{f}_j^{2h} &= \left(\lambda_j^h\right)^{m_1} \mu_j^h \mathbf{w}_{M-j}^{2h} \\ C_{M-j}' &= \left(\lambda_j^h\right)^{m_1} \frac{\mu_j^h}{\mu_{M-j}^{2h}} \\ \mathbf{e}_j^{2h} &= -\left(\lambda_j^h\right)^{m_1} \frac{\mu_j^h}{\mu_{M-j}^{2h}} \mathbf{w}_{M-j}^{2h}. \end{aligned}$$

The rest of the computation is the same. Hence, we see that the approximate solution that we obtain using injection and linear interpolation can be written as (10.10.61) with c_j and C_j replaced by c_j' and C_j'. Thus, we

get

$$\begin{aligned}\hat{\mathbf{u}}_{m_2}^h = &\sum_{j=1}^{M/2-1} \left(\lambda_j^h\right)^{m_2}\left\{\left(\lambda_j^h\right)^{m_1} + \left(c_j' + C_j'\right)\cos^2\frac{j\pi}{2M}\right\}\mathbf{w}_j^h \\ &+ \left(\lambda_{M/2}^h\right)^{m_2}\left(\lambda_{M/2}^h\right)^{m_1}\mathbf{w}_{M/2}^h \\ &+ \sum_{j=M/2+1}^{M-1} \left(\lambda_j^h\right)^{m_2}\left\{\left(\lambda_j^h\right)^{m_1} - \left(C_{M-j}' + c_{M-j}'\right)\cos^2\frac{j\pi}{2M}\right\}\mathbf{w}_j^h.\end{aligned} \tag{10.10.67}$$

As a comparison, we repeat the computation done in Example 10.10.1. In Figure 10.10.11 we plot the coefficients of $\hat{\mathbf{u}}_{m_2}^h$ after the first iteration of the coarse grid correction scheme and after two iterations of the coarse grid correction scheme. We see that the first iteration with injection is very comparable with the results using full weighting. We note that after two iterations, the approximate solution using injection is better than that using full weighting. We emphasize that because the result using injection given here was better than that using full weighting, it is not always this way. There are times when the injection result is better and there are times when the full weighting result is better.

We must emphasize that in the results given in Figure 10.10.11, we used the approximation of the differential operator on grid $\mathcal{G}^{2h}$, not $\mathcal{A}^{2h}$. Since A^{2h} and $\mathcal{A}^{2h}$ differ only by a multiple of two, it is easy to see that if we had used $\mathcal{A}^{2h}$ instead of A^{2h}, we would have gotten a result that looks like (10.10.67) with μ_j^{2h} replaced by $2\mu_j^{2h}$ in the definitions of c_j' and C_j'. In HW10.10.12 we show that using $\mathcal{A}^{wh}$ in place of A^{wh} produces poorer results. This shows that we must be very careful when choosing our grid transfers and coarse grid operators. For more information on this topic, see [7].

HW 10.10.10 Write a V-cycle multigrid code for approximating the solution to boundary–value problem (10.10.63)–(10.10.64) with $M_x = M_y = 128$, $m_1 = m_2 = 3$, and the weighted Jacobi scheme. Adjust one of your Jacobi codes from Section 10.5.5 to approximate the solution to boundary–value problem (10.10.63)–(10.10.64). Try to write both schemes so that they are equally efficient. Run 20 V-cycles of your multigrid scheme and 960 iterations of your Jacobi with a timer installed. Compare the time required for one V-cycle with 48 Jacobi iterations.

HW 10.10.11 Consider a uniform grid on $[0, 1]$ with $M = 2^3 = 8$, the operator associated with the one dimensional model problem on $\mathcal{G}^h$ defined by (10.10.32) with $M = 8$ and the full weighting and linear interpolation grid transfer operators given by (10.10.26) and (10.10.27), respectively. Com-

pute the matrix A^{2h} defined by (10.10.65) and show that it is the same as A^{2h} as defined in (10.10.33).

HW 10.10.12 Repeat the approximate solution computation using the coarse grid correction scheme on the one dimensional model problem as is done in Section 10.10.4.1. Use the injection operator as the restriction, linear interpolation as the prolongation, and use

$$\mathcal{A}^{2h} = \mathcal{I}_h^{2h} A^h I_{2h}^h = 2A^{2h}$$

to define the coarse grid problem. Compare your results with those given in Figure 10.10.11.

HW 10.10.13 Use multigrid to obtain an approximate solution to the elliptic boundary–value problem given in HW10.5.6. Use $M = 128$, a tolerance of 1.0×10^{-6}, and both the difference of successive iterates and the residual as a stopping criterion. Compare your results with those found in HW10.5.6. Discuss the efficiency of the multigrid computation compared with the computation using the Jacobi scheme.

HW 10.10.14 Use multigrid to obtain an approximate solution to the boundary–value problem given in HW10.7.6. Use the second order approximation of the Neumann boundary conditions, $M = 128$, a tolerance of 1.0×10^{-6}, and both the difference of successive iterates and the residual as a stopping criterion. Compare your results with those found in HW10.7.6.

HW 10.10.15 Use a multigrid scheme to find an approximate solution to the Robin problem given in HW10.8.8-(a). Use $M = 64$ and the difference between successive iterates as a stopping criterion.

HW 10.10.16 Use a multigrid scheme to find an approximate solution to the boundary–value problem given in HW10.9.6. Use $M_r = M_\theta = 16$ and the residual as a stopping criterion.

10.11 Computational Interlude VII

10.11.1 Blocking Out: Irregular Regions

In this section we introduce the "blocking out" procedure, designed to help solve problems on irregular regions. This subject should really be included in Chapter 11, but since the implementation is strongly related

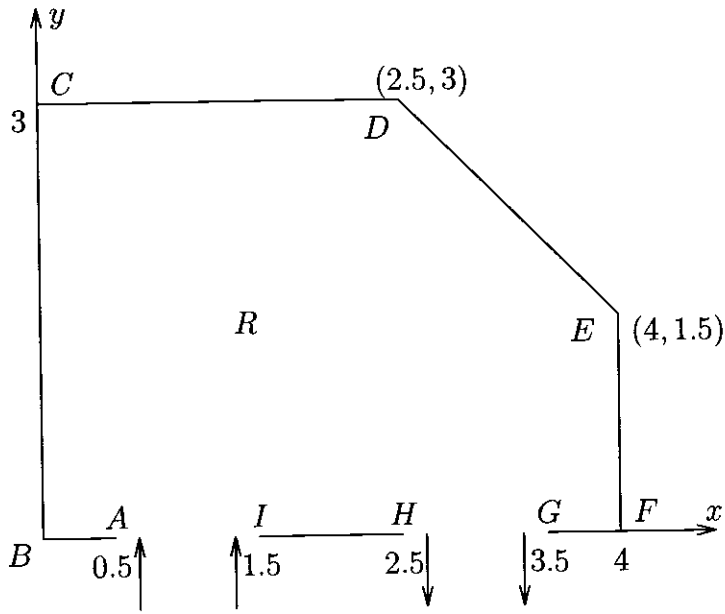

FIGURE 10.11.1. An irregular flow region. The fluid flows into the region through AI and out of the region through HG.

to the implementation given for mixed boundary–value problems described in Section 10.8.3, we thought that it might be appropriate to include it here. The blocking out procedure is a scheme for which stencil arrays are very useful. We illustrate the blocking out procedure by two problems given in Examples 10.11.1 and 10.11.2. The first of these problems involves the irrotational flow of an inviscid, incompressible fluid into and out of the nonrectangular region shown in Figure 10.11.1.

Example 10.11.1 Solve the following mixture boundary–value problem defined on the region R given in Figure 10.11.1.

$$\nabla^2\psi = 0, \quad (x, y) \in R. \tag{10.11.1}$$

$$\psi = 0 \quad \text{on } AB,\ BC,\ CD,\ DE,\ EF \text{ and } FG \tag{10.11.2}$$

$$\psi = 1 \quad \text{on } HI, \tag{10.11.3}$$

$$\psi = x - 0.5 \quad \text{on } AI \tag{10.11.4}$$

$$\frac{\partial \psi}{\partial y} = 0 \quad \text{on } HG. \tag{10.11.5}$$

Solution: Before we begin, we note that the function $\psi = \psi(x, y)$ represents the stream function associated with the flow described earlier. The stream function is such that

$$\frac{\partial \psi}{\partial x} = v \text{ and } \frac{\partial \psi}{\partial y} = -u,$$

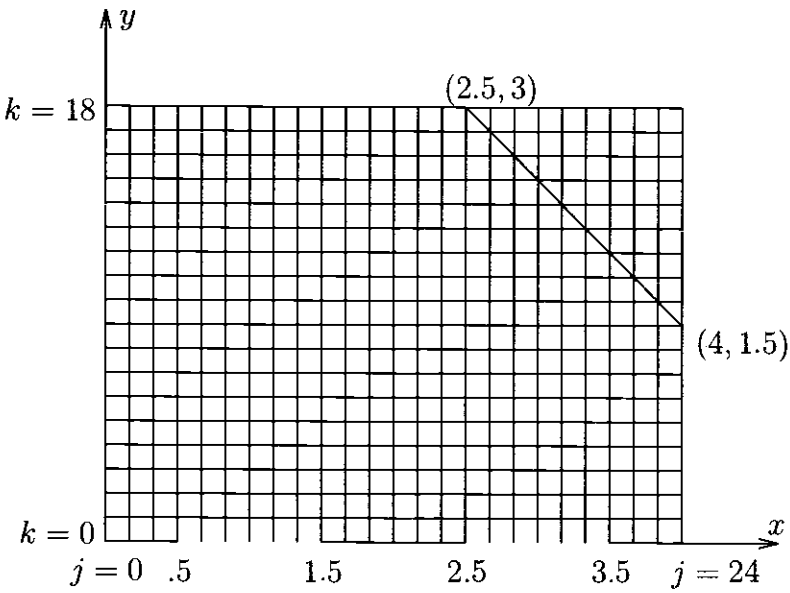

FIGURE 10.11.2. The natural rectangular extension of the flow region covered by a regular grid with $M_x = 24$ and $M_y = 18$.

where u and v are the horizontal and vertical components of the velocity. The curves defined by ψ =constant are called streamlines, and for steady flows they represent flow lines of the fluid particles. Hence, when problem (10.11.1)–(10.11.5) is solved, contour plots will be useful for displaying the results graphically.

The most obvious way to solve a problem like (10.11.1)–(10.11.5) is to place a grid on the region and handle the irregularity of the region by changing the loop variables on the iterative scheme. We solve this problem by a different approach, not that it is necessary for such an easy problem, but because it is a nice problem on which to demonstrate the blocking out technique. We begin by placing a grid on the rectangular region that is the natural extension of the region R. This grid is shown in Figure 10.11.2. We will describe a solution technique that combines the techniques used for mixed problems to handle the mixture of the Dirichlet and Neumann boundary conditions and the technique of "blocking out" a region.

As with most of these implementations, our major task is to fill the stencil array. We will go through the grid points shown in Figure 10.11.2, describing how to fill the stencil at each different kind of point.

Let us begin with the points along the $k = 0$ line. We do nothing with the corner points, since the corner points are neither reached to or solved on. Because the points $j = 16$, 17, 18, 19 and 20 are Neumann boundary points, we must fill the stencil with an equation that approximates the Neumann boundary condition (10.11.5). We assume that we want to use a second order approximation to approximate boundary condition (10.11.5) Hence, at these points we must solve an equation that looks like

$$\frac{-u_{j-1\,0} + 2u_{j\,0} - u_{j+1\,0}}{\Delta x^2} + \frac{2u_{j\,0} - 2u_{j\,1}}{\Delta y^2} = 0, \quad j = 16, \ldots, 20. \tag{10.11.6}$$

We note that when $j = 16$ and $j = 20$, the equation reaches to a boundary condition at

$j-1=15$ and $j+1=21$, respectively.

The other points on the $k=0$ line are regular Dirichlet boundary points, but because of the fact that they are on a grid line with the Neumann boundary points, it pays to treat them differently. We will treat these points as points at which we will solve an equation, but we will set up the equations so that the solution will always be the Dirichlet boundary conditions. At these points we set $S(j,k,0)=1$, $S(j,k,m)=0$ for $m\neq 0$, and the right hand side equal to zero, one, or $(j-3)/6$, depending on whether the point is on either AB or FG, on HI, or on AI, respectively. What we have done with the setup described above is to fix it so that we can solve on the entire $k=0$ line, and when we are solving, we will get the correct equations at the Neumann points and the correct Dirichlet boundary conditions at the other points. We must realize that to allow the equations on the $k=0$ line to reach in all directions, the u and u' arrays should be dimensioned so as to include a $k=-1$ row, and so that they do not affect the results, the u and u' arrays at these points should be filled with zeros.

The points that lie on the diagonal boundary line are very much like the Dirichlet points that lie on the $k=0$ line. These points are Dirichlet boundary points on which we will solve. Hence, the stencil will be set up as a one and four zeros, and the Dirichlet boundary condition 0 will be put in the right hand side.

The points above the diagonal boundary line are the "blocked out" points. They are points that should not be included but that we want to be included so as to give us a "regular data structure." Again, we define the equation at these points by a stencil with a one at the center and zeros on all four sides. The right hand sides associated with these points can be anything, so we set them to be equal to zero. We note that these points are such that they do not reach to any other points and no points reach to them.

There are three kinds of points left. The $j=0$ line and parts of $j=M_x$ and $k=M_y$ are points at which we have a usual Dirichlet boundary condition. The appropriate boundary condition should be placed in the u array for these points; no array need be defined for these points, and they will not be solved on. We should also notice that the blocked out points on the $j=M_x$ and $k=M_y$ lines can be treated in the same manner. Since these points can be treated just as if they were Dirichlet boundary points, we can treat the entire lines $j=0$, $j=M_x$, and $k=M_y$ the same.

And finally, the grid points that are left, the interior points of the region, are the usual interior points. These points get the equation

$$\frac{-u_{j-1\,k}+2u_{j\,k}-u_{j+1\,k}}{\Delta x^2}+\frac{-u_{j\,k-1}+2u_{j\,k}-u_{j\,k+1}}{\Delta y^2}=0. \tag{10.11.7}$$

If you have been keeping score, we have equations associated with all points $j=1,\ldots,M-1$, $k=0,\ldots,M-1$. We have appropriate boundary conditions in the $j=0$, $j=M$ and $k=M$ entries (except at the corners, where we do not need them), and we have the $k=-1$ entries defined and filled with zeros. Hence, we can choose any of our solvers, say for convenience the Jacobi iteration, and solve using an iteration of the form

For $k=0,\ldots,M_y-1$
For $j=1,\ldots,M_x-1$

$$u'_{j\,k}=-\frac{1}{S(j,k,0)}\big[F_{j\,k}-S(j,k,1)u_{j+1\,k}-S(j,k,2)u_{j-1\,k} -S(j,k,3)u_{j\,k+1}-S(j,k,4)u_{j\,k-1}\big]$$

Next j
Next k

We leave the actual implementation to the reader but include a plot of the solution in Figure 10.11.3. We make a special note that if your contour plotter is trying to be friendly and determines the contours by using a reasonably broad interpolation range, the plot can indicate bad results when the numerical procedure gives good results. One reasonably common error that is due to the contour plotter is the existence of small

vortices in the corners of the region. These vortices are not part of the analytic solution and not part of the numerical solution. Such vortices can be due to the interpolation procedure of the contour plotting scheme. One must be careful to understand what a plotting routine is doing with the numbers it is given.

We note that it was very convenient that the grid was chosen so that the grid points were on the slanted part of the boundary. We shall see in the next example that this was not at all necessary. In addition, we emphasize that much of the advantage of the approach introduced in this example, other than that it essentially allows us to solve this problem with the same code used for Dirichlet and Neumann boundary–value problems, is that the problem retains a nice, efficient data structure. For many machines and solving schemes on particular machines, the data structure associated with a problem is very important. Most often, the waste of computer time incurred in using the blocking out technique is made up in the savings in computer time gained from using a very regular data structure.

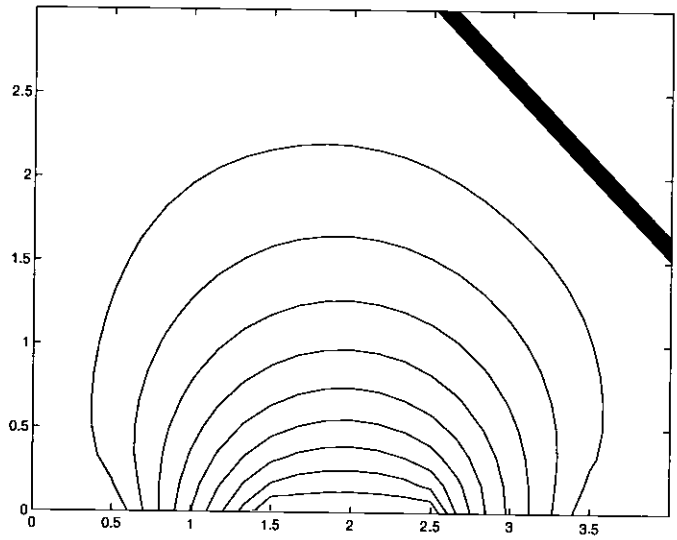

FIGURE 10.11.3. A contour plot of the flow lines associated with problem (10.11.1)–(10.11.5)

The next example that we include is another problem involving blocking out a region of the problem. We include this problem because it is more difficult than the problem done in Example 10.11.1, which causes us to be more clever how we block out the region.

Example 10.11.2 Solve the following Dirichlet boundary–value problem.

$$\nabla^2 v = 0 \text{ for } (x,y) \in D, \tag{10.11.8}$$

where $D = \{(x,y) \in (0,1) \times (0,1) \text{ and } (x-\frac{1}{2})^2 + (y-\frac{1}{2})^2 > \frac{1}{16}\}$ with boundary conditions $v(0,y) = v(1,y) = v(x,0) = v(x,1) = 0$ and $v(x,y) = \sin 4\pi(x-\frac{1}{4})$ when $(x-\frac{1}{2})^2 + (y-\frac{1}{2})^2 = \frac{1}{16}$.

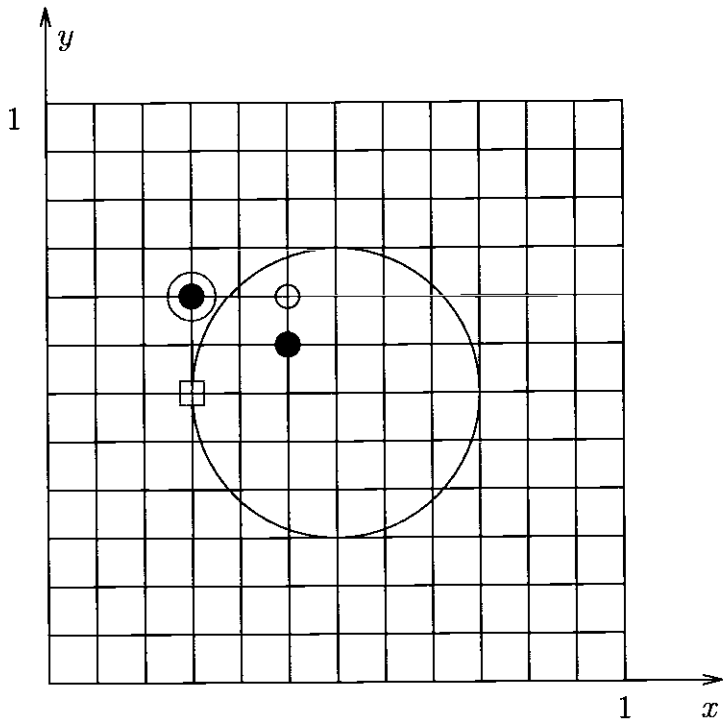

FIGURE 10.11.4. Domain associated with problem (10.11.8) overlaid with a regular grid.

Solution: In Figure 10.11.4 we see the regions on which we must solve overlaid with a uniform grid. Most of the points in the grid can be treated as usual. The boundary points on the edge of the square are reached to but not solved on. The interior points "away" from the circle have the usual difference equation,

$$\frac{-u_{j-1\,k} + 2u_{j\,k} - u_{j+1\,k}}{\Delta x^2} + \frac{-u_{j\,k-1} + 2u_{j\,k} - u_{j\,k+1}}{\Delta y^2} = 0,$$

associated with them.

All of the points inside or on the circle are to be blocked points. However, some of these must be treated differently. All of the stencils for the blocked points will contain a one at the center of the stencil and zeros on the sides. As we shall see later, it is important that the blocked points that neighbor the boundary (that is, those that are inside and get reached to from the outside, such as the one denoted by an open circle) assume the value of one. Hence, a one is placed in the right hand side associated with these points. The points on the circle, such as the one depicted by an open square, are true boundary points. Hence, the appropriate boundary condition is placed in the right hand side associated with these points. It makes no difference what is placed in the right hand side of the nonneighboring blocked points, such as that denoted by a filled circle. To be safe, we place a one in the right hand side associated with these points.

Hence, the only other kind of point that we must decide how to handle are the points that are in the interior of the domain that reach into the circle, such as the point denoted with a bull's-eye. Before we show how to treat the interior boundary points (the bull's-eye points), we must decide what type of difference equation we want at such a point.

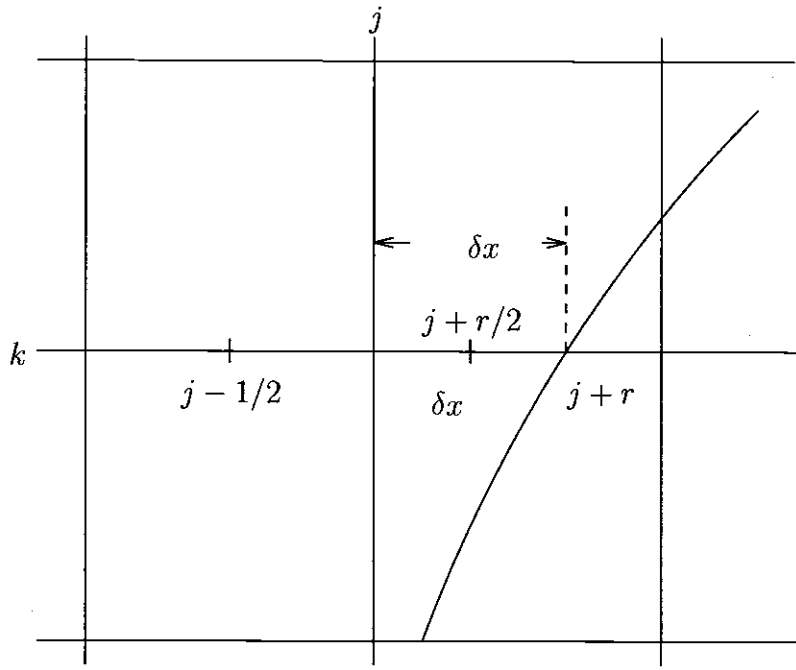

FIGURE 10.11.5. Blown up picture of the grid in the vicinity of the bull's-eye grid point.

In Figure 10.11.5, we show a blown up region near the bull's-eye point. It is not hard to see that if we approximate v_{yy} by

$$\frac{u_{j\,k-1} - 2u_{j\,k} + u_{j\,k+1}}{\Delta y^2},$$

we will get a $\mathcal{O}(\Delta y^2)$ approximation. To help in the approximation of u_{xx}, we set $r = \delta x/\Delta x$, denote the point on the boundary by $(j + r, k)$, the point halfway between the (j, k) point and the boundary by $(j + r/2, k)$, and v defined at the boundary point by $u_{j+r\,k}$. If we then proceed as we did when we derived the approximation of v_{xx} on a regular grid in Section 1.2, we approximate v_{xx} as

$$\begin{aligned}(v_{xx})_{j\,k} &\approx \frac{(v_x)_{j+r/2\,k} - (v_x)_{j-1/2\,k}}{r\frac{\Delta x}{2} + \frac{\Delta x}{2}} \\ &\approx \frac{\frac{u_{j+r\,k} - u_{j\,k}}{r\Delta x} - \frac{u_{j\,k} - u_{j-1\,k}}{\Delta x}}{\Delta x(1+r)/2},\end{aligned}$$

or

$$(v_{xx})_{j\,k} \approx \frac{ru_{j-1\,k} - (1+r)u_{j\,k} + u_{j+r\,k}}{r\Delta x^2(1+r)/2}. \tag{10.11.9}$$

Thus, at a point such as the bull's-eye point, we assign an equation of the form

$$\frac{-ru_{j-1\,k} + (1+r)u_{j\,k} - u_{j+r\,k}}{r\Delta x^2(1+r)/2} + \frac{-u_{j\,k-1} + 2u_{j\,k} - u_{j\,k+1}}{\Delta y^2} = 0. \tag{10.11.10}$$

We now discuss how we implement equation (10.11.10) as a part of our scheme. One might be tempted to place the boundary condition value $u_{j+r\,k}$ in $u_{j+1\,k}$ (which we have already filled with a one), but that does not work, since another point must reach (from above) to the same $u_{j+1\,k}$ point and get a different boundary condition. Hence, the way we approach a point such as the bull's-eye point is to include the boundary condition in the stencil. We define

$$\begin{aligned} S(j,k,3) &= -\frac{1}{\Delta y^2}, \\ S(j,k,4) &= -\frac{1}{\Delta y^2}, \\ S(j,k,0) &= \frac{2}{r\Delta x^2} + \frac{2}{\Delta y^2}, \\ S(j,k,1) &= -\frac{u_{j+r\,k}}{r\Delta x^2(1+r)/2}, \end{aligned}$$

and

$$S(j,k,2) = -\frac{1}{\Delta x^2(1+r)/2}.$$

Thus, we see that $S(j,k,3)$ and $S(j,k,4)$ are defined as usual. They do not even know that the point under consideration is strange. The term $S(j,k,0)$ contains a contribution from both the k direction and the j direction. The left stencil entry, $S(j,k,2)$, contains the term appropriate for reaching to the $u_{j-1\,k}$ term. And $S(j,k,1)$ contains the correct reaching term and the boundary condition. When the scheme reaches to $u_{j+1\,k}$ and multiplies $u_{j+1\,k}$ (which is one) by $S(j,k,1)$, the scheme will obtain the correct term according to equation (10.11.10).

We note that when the point above the (j,k) point reaches down, it will have an adjusted stencil also, and $S(\cdot,\cdot,4)$ will contain the boundary condition. Hence, when this stencil reaches and gets a one from the $(j+1,k)$ point, it too will satisfy the appropriate difference equation.

Above, we have described how we would implement equation (10.11.10). We next describe how we would "really" implement the blocking out for a problem such as the one being considered here. We begin by defining a distance function that will determine whether a point is not in or on the circle (this should be done allowing for a small tolerance so that we do not end up with points outside of the circle but arbitrarily close to the circle). We also need a distance function that will decide whether or not a point is on the circle (again we use a tolerance when defining what it means to be on the circle). When we have these two tools, we proceed as follows.

- Perform a sweep through the region in a j-k ordering
  - If the point and its two j neighbors are not in or on the circle, we set $S(j,k,1) = -1/\Delta x^2$, $S(j,k,2) = -1/\Delta x^2$, and $S(j,k,0) = 2/\Delta x^2$.
  - If the point is in or on the circle, we set the center of the stencil equal to one and its four neighbors equal to zero.
    - If the point is on the circle, we calculate the coordinates of the point on the circle and set the right hand side equal to the appropriate boundary condition.
    - If the point is inside of the circle, we set the right hand side equal to one.
  - If the point is not in or on the circle but one of its j neighbors is in the circle (only one of the j neighbors could possibly be in the circle), then we proceed as we did above, setting $S(j,k,m)$, $m = 0,1,2$, appropriately.

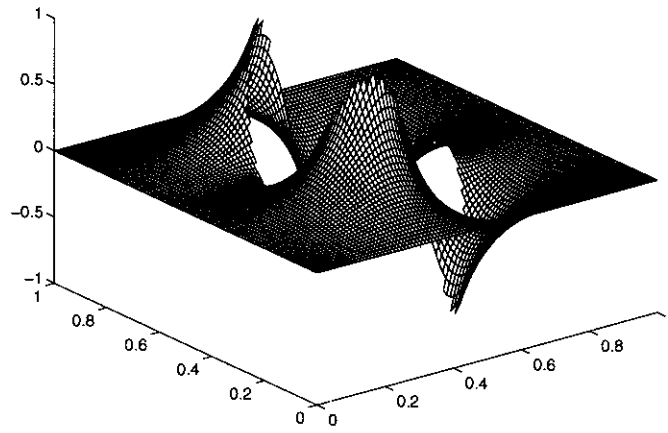

FIGURE 10.11.6. Solution to boundary value problem given in Example 10.11.2. The solution was obtained using $M_x = M_y = 100$ and a Gauss-Seidel iterative scheme. We note that the surface is plotted (using Matlab) only on the grid points that are on or outside of the circle and the circle is left as a hole in the surface.

(Make sure that you understand that we considered a point (j,k) where $(j+1,k)$ was in the circle. We need an analogous formula for a point (j,k) where $(j-1,k)$ is in the circle.)

- Perform a sweep through the region in the k-j ordering.
  - If the point and its two k neighbors are not in or on the circle, we set $S(j,k,3) = -1/\Delta y^2$, $S(j,k,4) = -1/\Delta y^2$, and $S(j,k,0) = S(j,k,0) + 2/\Delta y^2$. We note that the value is added to the center term of the stencil because it already has a contribution from the j direction.
  - If the point is in or on the circle, do nothing (all that is necessary has already been done to these points).
  - If the point is not in or on the circle but one of its k neighbors is in the circle (only one of the k neighbors could possibly be in the circle), then we proceed analogously to the way we did in the j direction. (Though we did not write out the values for $S(j,k,m)$, $m = 0,3,4$, for this case, it should be clear how to proceed. Remember, the contribution from the k direction to $S(j,k,0)$ must be added to the contribution from the j direction.)

The logic necessary to implement the above algorithm is a relatively complex piece of code. However, it should be clear that the same piece of code can be used for any region. The only difference is the "distance functions" that determine whether a point is in or on the region. Hence, the same approach can be used to solve many interesting problems.

We should note that in the grid given in Figure 10.11.4, none of the neighboring points are such that they reach into the circle in both the j and k directions. It should be clear that this is possible with other grids and if regions more complex than circles are considered. However, the algorithm described above will also work when a given point reaches into the circle in both the j and k directions.

Again the implementation is left to the reader. A solution to this problem is given in Figure 10.11.6. We should note the difficulty in plotting such a solution. One approach is to define the function to be some given value on the circle (say 0 or 1). This clearly defines the circle, but also has a steep gradient from the values near the circle to those assigned values on the circle. The plot given in Figure 10.11.6 uses a Matlab plotting routine that does not plot any values for the points in the circle. We should realize, however, that the points plotted include only those nearest or on the circle, i.e., the plot should not define the circle exactly. Surely, when we use $M = 100$, this should not be a problem.

Remark: The matrices associated with Examples 10.11.1 and 10.11.2, and almost any problem using the blocking out procedure, are special matrices. The matrix will be reducible, i.e., the matrix will not be irreducible. This is true because the equations associated with the blocked variables that are nonboundary, blocked grid points, do not interact with any other grid points. Hence, if we use the characterization given of an irreducible matrix immediately following Definition 10.2.2, since the right hand side associated with these nonboundary, blocked variables can be changed without affecting the solution at other points, the matrix is reducible.

We should note that because the matrix is reducible, we do not know that it cannot be solved. It means only that there are several theorems that we cannot apply to this matrix. The equations associated with these examples are easily solved using any of the relaxation techniques.

HW 10.11.1 Implement the blocking out procedure that was described in Example 10.11.1.

HW 10.11.2 Implement the blocking out procedure that was described in Example 10.11.2.

HW 10.11.3 Show that the approximation described in equation (10.11.9) is only a $\mathcal{O}(\Delta x)$ approximation.

10.11.2 HW0.0.4

We are now capable, and maybe ready, to return to the last nonlinear problem given to us in the Prelude, HW0.0.4. We see that HW0.0.4 appears to be elliptic (though we do not really know) with Neumann boundary conditions on all sides. Except for the fact that the equation is nonlinear, we have some tools that can be used to solve this problem. We take this time here to introduce a method that can be used on nonlinear problems.

We begin by placing a grid on the region $[-1, 2] \times [0, 1]$. To be specific, we set $\Delta x = 0.06$, $M_x = 50$, $\Delta y = 0.05$ and $M_y = 20$. If we were to return to the Murman-Cole asymptotic analysis for the derivation of the

thin disturbance transonic flow equations for this problem, ref. [50], we would see that we were fortunate that Δx was chosen such that $(0,0)$ is not a grid point. The point $(0,0)$ is a singular point for this problem.

Using a centered difference for the first and second order derivatives, we obtain the following difference equation approximation of the partial differential equation given in HW0.0.4.

$$\left[1 - M_\infty^2 - (\gamma+1)M_\infty^2 \frac{u_{j+1\,k} - u_{j-1\,k}}{2\Delta x}\right]\left[\frac{u_{j+1\,k} - 2u_{j\,k} + u_{j-1\,k}}{\Delta x^2}\right] + \frac{u_{j\,k+1} - 2u_{j\,k} + u_{j\,k-1}}{\Delta y^2} = 0. \tag{10.11.11}$$

For convenience, we let

$$c_{j\,k} = 1 - M_\infty^2 - (\gamma+1)M_\infty^2\left(\frac{u_{j+1\,k} - u_{j-1\,k}}{2\Delta x}\right),$$

and rewrite equation (10.11.11) as

$$\frac{1}{\Delta y^2}u_{j\,k-1} + \frac{c_{j\,k}}{\Delta x^2}u_{j-1\,k} - \left(\frac{2}{\Delta y^2} + \frac{2c_{j\,k}}{\Delta x^2}\right)u_{j\,k} + \frac{1}{\Delta y^2}u_{j\,k+1} + \frac{c_{j\,k}}{\Delta x^2}u_{j+1\,k} = 0. \tag{10.11.12}$$

If we want second order accurate boundary conditions, we consider

$$\frac{u_{1\,k} - u_{-1\,k}}{2\Delta x} = 0, \quad k = 0, \ldots, M_y \text{ at } j = 0 \tag{10.11.13}$$

$$\frac{u_{M_x+1\,k} - u_{M_x-1\,k}}{2\Delta x} = 0, \quad k = 0, \ldots, M_y \text{ at } j = M_x \tag{10.11.14}$$

$$\frac{u_{j\,1} - u_{j\,-1}}{2\Delta y} = 0, \quad j = 0, \ldots, 16,\ j = 34, \ldots, 50, \text{ at } k = 0 \tag{10.11.15}$$

$$\frac{u_{j\,1} - u_{j\,-1}}{2\Delta y} = \frac{-((-1+j\Delta x) - 0.5)}{\sqrt{6.375625 - ((-1+j\Delta x) - 0.5)^2}}, \quad j = 17, \ldots, 33 \text{ at } k = 0 \tag{10.11.16}$$

$$\frac{u_{j\,M_y+1} - u_{j\,M_y-1}}{2\Delta y} = 0, \quad j = 0, \ldots, M_x \text{ at } k = M_y. \tag{10.11.17}$$

Thus, we must solve equation (10.11.12) for $j = 0, \ldots, M_x$, $k = 0, \ldots, M_y$ along with boundary conditions (10.11.13)–(10.11.17).

At first glance, this problem looks much like the problems solved in the last section. We have a difference equation in the interior and Neumann boundary conditions on all sides. However, we must remember that the difference equation for this problem is nonlinear. Just because we have

used notation in writing equation (10.11.12), do not forget that the $c_{j\,k}$'s depend on the $u_{j\,k}$'s.

As we did in the last section, we should be aware that this problem will not have a unique solution. Again, for any solution ϕ of HW0.0.4, $\phi + c$ will also be a solution. Though we do not forget this fact, we proceed without worrying about it. As with so many potential problems (in this problem ϕ is the **velocity potential**), we need the solution to problem HW0.0.4 only up to an additive constant.

We recall that the major techniques used to try to solve nonlinear problem HW0.0.1 was to (1) lag part of the nonlinearity back to the previous time step, (2) linearize about the previous time step, and (3) use Newton's method. It seems clear that since we do not have time steps (let alone a previous time step), we cannot use the first two approaches. Using Newton's method would surely be a logical (and difficult) approach. We instead use a slight variation of the first approach, lagging part of the nonlinearity. Instead of using time steps, we consider an iterative solver for equation (10.11.12)–(10.11.17), for example, Gauss-Seidel. We begin with an initial guess of zero, assume that we have the n-th iterate, and rewrite equation (10.11.12) as

$$\frac{1}{\Delta y^2}u_{j\,k-1}^{n+1} + \frac{c_{j\,k}^{n+1}}{\Delta x^2}u_{j-1\,k}^{n+1} - \left(\frac{2}{\Delta y^2} + \frac{2c_{j\,k}^{n+1}}{\Delta x^2}\right)u_{j\,k}^{n+1} + \frac{1}{\Delta y^2}u_{j\,k+1}^{n+1} + \frac{c_{j\,k}^{n+1}}{\Delta x^2}u_{j+1\,k}^{n+1} = 0, \tag{10.11.18}$$

where the superscripts on $u_{j\,k}$ indicate the iteration count and the superscript on $c_{j\,k}$ indicates that the $u_{j\,k}$ terms in $c_{j\,k}$ are evaluated with the $(n+1)$-st iterate. A solution technique that is used often for equations such as (10.11.18) is to lag the iteration counter on the $c_{j\,k}^{n+1}$ term and consider the equation

$$\frac{1}{\Delta y^2}u_{j\,k-1}^{n+1} + \frac{c_{j\,k}^{n}}{\Delta x^2}u_{j-1\,k}^{n+1} - \left(\frac{2}{\Delta y^2} + \frac{2c_{j\,k}^{n}}{\Delta x^2}\right)u_{j\,k}^{n+1} + \frac{1}{\Delta y^2}u_{j\,k+1}^{n+1} + \frac{c_{j\,k}^{n}}{\Delta x^2}u_{j+1\,k}^{n+1} = 0. \tag{10.11.19}$$

Thus, *the approach is to consider equation (10.11.19) along with boundary conditions (10.11.13)–(10.11.17), using one of the iterative schemes such as Gauss-Seidel or SOR.* It should be clear that if the solution to equations (10.11.19), (10.11.13)–(10.11.17) converges so that $u_{j\,k}^{n+1} = u_{j\,k}^{n}$ for all j and k, then $u_{j\,k} = u_{j\,k}^{n}$ will satisfy equations (10.11.12)–(10.11.17). We might consider Jacobi, but since Jacobi was unacceptable for the linear problem with Neumann boundary conditions, it is too likely to cause trouble.

At this time, you should be ready to attempt solving HW0.0.4. We suggest that you use either Gauss-Seidel or SOR (try $\omega = 1.8$). If you want

to use SOR, the correct approach is to approximate ω_b using the results from Section 10.5.12. When we did the computation, we used the sup-norm of the difference of two iterates with a tolerance of 10^{-7} as our stopping criterion. We should realize that it may be difficult to know when you have obtained the correct solution to this problem. This will force us to proceed very carefully. We will add, however, that aerodynamicists would be interested in plots of the **reduced pressure coefficient**,

$$\bar{c}_p = -2u = 2\phi_x \approx \frac{u_{j+1\,k} - u_{j-1\,k}}{\Delta x},$$

along "the body," i.e., at $k = 0$.

We emphasize that you are attempting this approach with none of the residual correction theory supporting you. There is some theory available for using iterative techniques for solving nonlinear problems, but it is difficult and restrictive. The approach that you should use in this problem is, as we have done before, to use what we know about the numerical technique for linear problems (and understanding that in this case we do not know as much about the physical and mathematical problem as we should) and proceed carefully.

10.12 ADI Schemes

In Section 10.4 we mentioned that it might be possible to use some sort of ADI scheme for solving problems associated with elliptic partial differential equations. Perhaps the easiest way to see that this is possible is to consider an elliptic problem such as that given in (10.2.1)–(10.2.2) as the steady state equation associated with a time dependent equation of the form

$$v_t - \nabla^2 v = F, \quad (x,y) \in R = (0,1) \times (0,1) \tag{10.12.1}$$

$$v(x,y,t) = f, \quad (x,y) \in \partial R \tag{10.12.2}$$

$$v(x,y,0) = 0, \quad (x,y) \in [0,1] \times [0,1]. \tag{10.12.3}$$

Approximate solutions to problem (10.12.1)–(10.12.2) can be found using any of the ADI schemes developed in Section 4.4. If these ADI solutions are run to steady state, the steady state solution will be an approximate solution to problem (10.2.1)–(10.2.2). This approach has been used often and has been given such interesting names as the "method of false transients" and "relaxing through time." It should also be emphasized that time accuracy is not important (hence the name "the method of false transients"). As long as the scheme converges for large n, the solution will approximate the solution of (10.2.1)–(10.2.2), no matter what type of time steps are chosen. It is often one's first inclination to choose large time steps to try to get to steady state as soon as possible. As we shall see below, this is

not the best approach. The time steps can be chosen so as to minimize the time (computer time) necessary to get to steady state, and we do not have to use a constant time step. We emphasize that by convergence here, we mean that $u_{jk}^{n+1} \approx u_{jk}^{n}$ for all j, k. As is usually the case, we can measure whether $u^{n+1} \approx u^{n}$ using either the ℓ^2 norm or the sup-norm.

In particular, if we chose the Peaceman-Rachford scheme as our ADI scheme of choice, we would find that the difference scheme

$$\frac{u_{jk}^{n+\frac{1}{2}} - u_{jk}^{n}}{\Delta t_n/2} = \frac{1}{\Delta x^2}\delta_x^2 u_{jk}^{n+\frac{1}{2}} + \frac{1}{\Delta y^2}\delta_y^2 u_{jk}^{n} + \frac{1}{2}F_{jk}^{n} \tag{10.12.4}$$

$$\frac{u_{jk}^{n+1} - u_{jk}^{n+\frac{1}{2}}}{\Delta t_n/2} = \frac{1}{\Delta x^2}\delta_x^2 u_{jk}^{n+\frac{1}{2}} + \frac{1}{\Delta y^2}\delta_y^2 u_{jk}^{n+1} + \frac{1}{2}F_{jk}^{n+1} \tag{10.12.5}$$

can be considered to be an iterative scheme in n for solving problem (10.2.3)–(10.2.7) where the boundary conditions are included in the u_{jk} terms where $j = 0$, $k = 0$, $j = M_x$ or $k = M_y$. We have used the notation Δt_n on the time steps to indicate that we may wish to use different Δt's for different time steps.

To implement scheme (10.12.4)–(10.12.5) for solving a particular problem, we proceed as we did in Section 4.4.1 (the codes developed in Section 4.4.1 should be useful here). We recall that if we consider our data being ordered in a j-k ordering, then solving equation (10.12.4) involves solving an $(M_x-1)(M_y-1)\times(M_x-1)(M_y-1)$ tridiagonal matrix equation (or alternatively M_y-1 tridiagonal systems of dimension $(M_x-1)\times(M_x-1)$). Likewise, if we consider our data being ordered in a k-j ordering, then solving equation (10.12.5) involves solving an $(M_x-1)(M_y-1)\times(M_x-1)(M_y-1)$ tridiagonal matrix equation (or alternatively M_x-1 tridiagonal systems of dimension $(M_y-1)\times(M_y-1)$). We must be aware that there is a transpose of data taking place between the solution of these two equations. This must be accounted for as we did for the time dependent Peaceman-Rachford schemes in Section 4.4.1.

As usual for iterative schemes, we must choose a stopping criterion, and the logical choices are those considered in Section 10.5.4. For iterative scheme (10.12.4)–(10.12.5), we must also choose the time steps Δt_n. In Table 10.12.1, for several different choices of Δt_n we give the number of iterations necessary to solve the problem given in HW10.12.1 using $M_x = M_y = 100$ and the sup-norm of the difference between iterates along with a tolerance of 10^{-9} as the stopping criterion. It is clear that the convergence of the Peaceman-Rachford scheme depends strongly on the choice of the parameters Δt_n. We note that we used a problem with M_x and M_y large and a small tolerance to clearly illustrate the fact that the number of iterations necessary is strongly influenced by the choice of time steps and the number of time steps.

To consider the Peaceman-Rachford ADI scheme in general, we consider

number of time steps	time step (steps)	number of iterations necessary
1	5.0	2685
1	0.1	426
1	0.07	359
1	0.062	340
2	0.0032 0.0254	64
3	0.0002 0.0032 0.0508	34
4	0.00014 0.0011 0.0090 0.072	29

TABLE 10.12.1. Number of iterations necessary to solve the problem given in HW10.12.1 approximately. We used $M_x = M_y = 100$ and a stopping criterion consisting in requiring that the maximum of the difference between consecutive iterates be less than 10^{-9}.

a matrix equation of the form

$$A\mathbf{u} = \mathbf{f}, \tag{10.12.6}$$

where A can be written as $A = H + V$. We rewrite equation (10.12.6) twice as

$$(H + rI)\mathbf{u} = \mathbf{f} - (V - rI)\mathbf{u} \tag{10.12.7}$$

$$(V + rI)\mathbf{u} = \mathbf{f} - (H - rI)\mathbf{u}, \tag{10.12.8}$$

obtain an initial estimate for $\mathbf{u}$, $\mathbf{u}^0$, and lag the appropriate terms to rewrite (10.12.7)–(10.12.8) as

$$(H + r_{n+1}I)\mathbf{u}^{n+\frac{1}{2}} = \mathbf{f} - (V - r_{n+1}I)\mathbf{u}^n \tag{10.12.9}$$

$$(V + r_{n+1}I)\mathbf{u}^{n+1} = \mathbf{f} - (H - r_{n+1}I)\mathbf{u}^{n+\frac{1}{2}}, \tag{10.12.10}$$

for $n = 0, 1, \ldots$. The iterative scheme obtained, (10.12.9)–(10.12.10), is referred to as the **Peaceman-Rachford implicit alternating direction iterative scheme** or the Peaceman-Rachford ADI scheme. We must not confuse the Peaceman-Rachford scheme for solving time dependent equations with iterative scheme defined above. Both schemes are generally referred to as the Peaceman-Rachford ADI scheme. In Section 4.4.1 ADI refers to "alternating direction implicit," whereas here ADI refers to "alternating direction iterative." Generally, the parameters r_{n+1} will be chosen

to cycle through a finite set of parameters. When we consider the one parameter case, we refer to the parameter as r, and when we consider the multiple parameter case, we refer to the parameters as $\mathbf{r} = (r_1, \ldots, r_m)$.

Consider using the one parameter Peaceman-Rachford ADI scheme to approximate the solution to the model problem (10.2.3)–(10.2.7). It should be reasonably clear that if we let H denote the matrix associated with the x derivatives and V denote the matrix associated with the y derivatives, then schemes (10.12.4)–(10.12.5) and (10.12.9)–(10.12.10) are the same if we choose $\Delta t_n = \Delta t$ and $r = r_n = 2/\Delta t$ for all n.

If we solve equations (10.12.9)–(10.12.10) for $\mathbf{u}^{n+1}$ in terms of $\mathbf{u}^n$ (eliminating $\mathbf{u}^{n+\frac{1}{2}}$), we get

$$\mathbf{u}^{n+1} = R_{PR_{r_{n+1}}} \mathbf{u}^n + \mathbf{f}'_{r_{n+1}}, \tag{10.12.11}$$

where

$$R_{PR_r} = (V + rI)^{-1}(H - rI)(H + rI)^{-1}(V - rI) \tag{10.12.12}$$

and

$$\mathbf{f}'_r = (V + rI)^{-1} \left\{ I - (H - rI)(H + rI)^{-1} \right\} \mathbf{f}. \tag{10.12.13}$$

If the error is given by $\mathbf{e}^n = \mathbf{u} - \mathbf{u}^n$, then equation (10.12.11) is equivalent to

$$\mathbf{e}^{n+1} = R_{PR_{r_{n+1}}} \mathbf{e}^n. \tag{10.12.14}$$

We cannot use Proposition 10.5.2 to prove that the Peaceman-Rachford ADI scheme, (10.12.9)–(10.12.10), converges, because the iteration matrix $R_{PR_{r_{n+1}}}$ depends on n. Even though this will not cause a problem, to illustrate the convergence of the Peaceman-Rachford ADI scheme we begin by considering the one parameter scheme for solving model problem (10.2.3)–(10.2.7). Hence, we must solve (10.12.4)–(10.12.5) with $\Delta t_n = \Delta t$ for all n.

In Chapter 3 we showed that for Dirichlet boundary conditions, the discrete von Neumann stability analysis actually found the eigenvalues of the iteration matrix (in that case it was a "time" iteration matrix). It is not hard to see that this result will extend also to two dimensional schemes (using a two dimensional Fourier series instead of a one dimensional Fourier series). Hence, we can use the stability analysis for the time dependent Peaceman-Rachford scheme given in Section 4.4.1 to see that the eigenval-

ues of R_{PR_r} are given by

$$\frac{(1-2r_x\sin^2\frac{j\pi\Delta x}{2})(1-2r_y\sin^2\frac{k\pi\Delta y}{2})}{(1+2r_x\sin^2\frac{j\pi\Delta x}{2})(1+2r_y\sin^2\frac{k\pi\Delta y}{2})}t$$
$$= \frac{(\frac{2}{\Delta t}-\frac{4}{\Delta x^2}\sin^2\frac{j\pi\Delta x}{2})(\frac{2}{\Delta t}-\frac{4}{\Delta y^2}\sin^2\frac{k\pi\Delta y}{2})}{(\frac{2}{\Delta t}+\frac{4}{\Delta x^2}\sin^2\frac{j\pi\Delta x}{2})(\frac{2}{\Delta t}+\frac{4}{\Delta y^2}\sin^2\frac{k\pi\Delta y}{2})}$$
$$= \frac{(r-\frac{4}{\Delta x^2}\sin^2\frac{j\pi\Delta x}{2})(r-\frac{4}{\Delta y^2}\sin^2\frac{k\pi\Delta y}{2})}{(r+\frac{4}{\Delta x^2}\sin^2\frac{j\pi\Delta x}{2})(r+\frac{4}{\Delta y^2}\sin^2\frac{k\pi\Delta y}{2})}.$$

It is easy to see (again using the work done in Section 4.4.1) that $\sigma(R_{PR_r}) < 1$ and that the Peaceman-Rachford ADI scheme when applied to solving model equation (10.2.3)–(10.2.7) will converge.

To consider convergence of the multiparameter Peaceman-Rachford ADI scheme, we must use a slight variation of Proposition 10.5.2, because in this case the iteration matrix depends on n. The appropriate result is that *if the iteration matrix R depends on n and satisfies $\sigma(R) < 1$ for all n, then the scheme converges.* Some of the results that are available for the multiparameter Peaceman-Rachford ADI scheme include the following.

Theorem 10.12.1 *(Page 213, [70].) Let H and V be Hermitian nonnegative definite matrices where either H or V is positive definite. Then for any $r > 0$, $\sigma(R_{PR_r}) < 1$ (and hence, the Peaceman-Rachford ADI scheme converges).*

Proposition 10.12.2 *(Page 222, [70].) Suppose H and V are Hermitian positive definite matrices satisfying $HV = VH$. Then if the acceleration parameters $r_1, \ldots, r_{n^2}$ are chosen to be the eigenvalues of either H or V and we consider exact arithmetic, the Peaceman-Rachford ADI scheme converges to the exact solution in n^2 steps (i.e., the Peaceman-Rachford ADI scheme is a direct scheme).*

Theorem 10.12.3 *(Page 222, [70].) Let H and V be Hermitian positive definite matrices that satisfy $HV = VH$. Then the Peaceman-Rachford ADI scheme converges for any choice of acceleration parameters $r_1, \ldots, r_m$.*

Remark 1: In all of the results mentioned above, we must choose the parameters $r_1, \ldots, r_n$. Choosing these parameters poorly can give poor convergence results. These parameters can be chosen so as to give optimal convergence. The choice of the optimal parameters is rather difficult. A careful analysis will show that the optimal one parameter Peaceman-Rachford ADI scheme and the optimal SOR scheme have the same asymptotic rate of convergence. It can also be shown that any optimal Peaceman-Rachford ADI scheme with two or more parameters will have an asymptotic rate of convergence greater than that for optimal SOR. It can be shown that while the

asymptotic rate of convergence of optimal SOR is of order h, the asymptotic rate of convergence of the optimal m-parameter Peaceman-Rachford ADI scheme is of order $h^{1/m}$. However, we must understand that one iterative step of the Peaceman-Rachford ADI scheme is much more expensive than one SOR step. Of course, the difference is machine dependent, but one step of the Peaceman-Rachford ADI scheme will take approximately 10–20 times as much work as one SOR step.

For the case when the eigenvalues of H and V are positive and lie between a and b, Young and Gregory, ref. [76], Volume II, page 1050, give the Peaceman-Rachford parameters

$$r_j^* = b(a/b)^{(2j-1)/2m}, \quad j = 1, \ldots, m.$$

These parameters are nearly as good as the optimal parameters and much easier to compute. For our model problem (10.2.3)–(10.2.7) (and, hence, for the problem considered in HW10.12.1), the Peaceman-Rachford parameters are given by $r_j^* = 4\cos^2\frac{\pi\Delta x}{2}\left(\tan^2\frac{\pi\Delta x}{2}\right)^{(2j-1)/2m}$, $j = 1, \ldots, m$. We should realize that the parameters r_j are related to the time steps Δt_j in scheme (10.12.4)–(10.12.5) by $\Delta t_j = 2\Delta x^2/r_j$, $j = 1, \ldots, m$. It should be easy to see that the time steps given in the last four entries of Table 10.12.1 were determined from the Peaceman-Rachford parameters.

Remark 2: Of course, it should be clear that we can use other parabolic schemes from Chapter 4 to give iterative solvers for elliptic problems. Specifically, we should make special note that the results concerning the three dimensional Peaceman-Rachford scheme given in HW4.4.11 show that *we do not want to use the three dimensional Peaceman-Rachford schemes as an iterative solver for three dimensional elliptic problems.* (Because of the conditional stability, we would have to limit the size of our acceleration parameters.) For three dimensional elliptic problems, one of the common approaches is to use the Douglas-Gunn scheme as an iterative solver.

HW 10.12.1 (a) Use the Peaceman-Rachford ADI scheme to approximate the solution to the following boundary–value problem

$$\begin{aligned} \nabla^2 v &= F(x,y), \quad x \in (0,1) \times (0,1) \\ v(x,0) &= v(x,1) = 0, \quad x \in [0,1] \\ v(0,y) &= v(1,y) = 0, \quad y \in [0,1] \end{aligned}$$

where

$$F(x,y) = \begin{cases} 1.0 & \text{when } 0.2 \le x \le 0.8 \text{ and } 0.2 \le y \le 0.8 \\ 0.0 & \text{otherwise.} \end{cases}$$

Use $M_x = M_y = 100$, a stopping criterion of requiring the sup-norm of the difference of consecutive iterates to be less than 10^{-9}, and $\Delta t = 0.063$.

(b) Repeat part (a) using $\Delta t = 1.0$.
(c) Repeat part (a) using $\Delta t = 5.0$.
(d) Repeat part (a) using $\Delta t_1 = 0.0014$, $\Delta t_2 = 0.0011$, $\Delta t_3 = 0.0090$, and $\Delta t_4 = 0.72$.
(e) Solve the above problem using optimal SOR. Compare the number of iterations necessary for optimal SOR and the optimal Peaceman-Rachford ADI scheme. Compare the amount of computer time necessary for these schemes.

HW 10.12.2 (a) Use the Peaceman-Rachford ADI scheme to approximate the solution to the following boundary–value problem.

$$\nabla^2 v = \sin \pi x \sin 2\pi y, \quad (x, y) \in R = (0, 1) \times (0, 1)$$
$$v = 0 \quad \text{on } \partial R.$$

Use $M_x = M_y = 10$ and compare the number of iterations that are necessary for this solution to the number of iterations that were necessary when we used optimal SOR on this problem. See HW10.5.14.
(b) Repeat the solution in part (a) using
(i) several different fixed values of the time step and
(ii) several sequences of different time steps.
Compare the number of iterations necessary for this solution for the different cases considered in (i) and (ii), and to the number of iterations necessary using optimal SOR.

10.13 Conjugate Gradient Scheme

It should be clear that one of the problems related to the use SOR and ADI methods is the choice of parameters. With the right choice of certain parameters, these methods are very good. However, it can sometimes be very difficult to choose these parameters.

In this section we introduce a method without this difficulty, the conjugate gradient scheme. The conjugate gradient scheme can be regarded as a modification of the method of steepest descent. For a more complete discussion of the conjugate gradient method, see [13], page 516, or [2], page 18.

As usual we consider the equation

$$A\mathbf{u} = \mathbf{f}, \tag{10.13.1}$$

where A is assumed to be an $L \times L$, positive definite matrix. We define the function

$$F(\mathbf{u}) = (A\mathbf{u}, \mathbf{u}) - (\mathbf{f}, \mathbf{u}),$$

where $(\cdot,\cdot)$ is the inner product (dot product) on $\mathbb{R}^L$. It can be shown that F is a uniformly convex quadratic function, so it has a unique minimum. Since

$$F'(\mathbf{u}) = A\mathbf{u} - \mathbf{f}$$

and $F'(\mathbf{u}) = \boldsymbol{\theta}$ at a minimum, finding this unique minimum of F is equivalent to solving equation (10.13.1). The conjugate gradient algorithm for solving equation (10.13.1) is given as follows.

Conjugate Gradient-10.13.1

Step 1: Choose $\mathbf{u}_0$. Set $\mathbf{r}_0 = \mathbf{f} - A\mathbf{u}_0$, $k = 0$, and $\mathbf{q}_0 = \mathbf{r}_0$.

Step 2: $\alpha_k = \frac{(\mathbf{r}_k, \mathbf{r}_k)}{(A\mathbf{q}_k, \mathbf{q}_k)}$

$\mathbf{u}_{k+1} = \mathbf{u}_k + \alpha_k \mathbf{q}_k$

$\mathbf{r}_{k+1} = \mathbf{f} - A\mathbf{u}_{k+1} = \mathbf{r}_k - \alpha_k A\mathbf{q}_k$

Quit if $\mathbf{r}_{k+1}$ is zero or sufficiently small

$\beta_k = -\frac{(\mathbf{r}_{k+1}, \mathbf{r}_{k+1})}{(\mathbf{r}_k, \mathbf{r}_k)}$

$\mathbf{q}_{k+1} = \mathbf{r}_{k+1} + \beta_k \mathbf{q}_k$

Step 3: $k = k + 1$; Go to Step 2

Remark 1: The first step of the conjugate gradient scheme, choosing $\mathbf{u}_1 = \mathbf{u}_0 + \alpha_0 \mathbf{q}_0$ (where $\mathbf{q}_0 = \mathbf{r}_0$) and

$$\alpha_0 = \frac{(\mathbf{r}_0, \mathbf{r}_0)}{(A\mathbf{q}_0, \mathbf{q}_0)}, \tag{10.13.2}$$

is just the first step of the method of steepest descent. We note that we obtain α_0 by minimizing $F(\mathbf{u}_0 + \alpha\mathbf{q}_0)$ as a function of α, i.e., since

$$\begin{aligned} 0 &= \frac{d}{d\alpha} F(\mathbf{u}_0 + \alpha\mathbf{q}_0) && (10.13.3)\\ &= F'(\mathbf{u}_0 + \alpha\mathbf{q}_0) \cdot \mathbf{q}_0 \\ &= (A(\mathbf{u}_0 + \alpha\mathbf{q}_0) - \mathbf{f}, \mathbf{q}_0) \\ &= (-\mathbf{r}_0 + \alpha A\mathbf{q}_0, \mathbf{q}_0) \\ &= -(\mathbf{r}_0, \mathbf{q}_0) + \alpha(A\mathbf{q}_0, \mathbf{q}_0), && (10.13.4) \end{aligned}$$

$F(\mathbf{u}_0 + \alpha\mathbf{q}_0)$ is minimized if α is chosen to be

$$\alpha_0 = \frac{(\mathbf{r}_0, \mathbf{q}_0)}{(A\mathbf{q}_0, \mathbf{q}_0)}.$$

Then, since $\mathbf{q}_0 = \mathbf{r}_0$ and $(\mathbf{r}_0, \mathbf{q}_0) = (\mathbf{r}_0, \mathbf{r}_0)$, we get the expression given in the algorithm.

Remark 2: We note that $\mathbf{r}_0$ and $\mathbf{r}_1$ are orthogonal, since

$$\begin{aligned}(\mathbf{r}_0, \mathbf{r}_1) &= (\mathbf{q}_0, \mathbf{r}_1) \qquad (10.13.5)\\ &= (\mathbf{q}_0, \mathbf{f} - A\mathbf{u}_1)\\ &= (\mathbf{q}_0, \mathbf{f} - A\mathbf{u}_0 - \alpha_0 A\mathbf{q}_0) \qquad \text{(definition of } \mathbf{u}_1)\\ &= (\mathbf{q}_0, \mathbf{r}_0) - \alpha_0(\mathbf{q}_0, A\mathbf{q}_0) \qquad \text{(definition of } \mathbf{r}_0)\\ &= (\mathbf{q}_0, \mathbf{r}_0) - (\mathbf{r}_0, \mathbf{r}_0) \qquad \text{(definition of } \alpha_0)\\ &= 0. \qquad (10.13.6)\end{aligned}$$

Remark 3: Proceed with the algorithm, where at this time $\mathbf{q}_1$ is yet to be determined, and define $\mathbf{u}_2 = \mathbf{u}_1 + \alpha_1\mathbf{q}_1$. If we choose α_1 so as to minimize $F(\mathbf{u}_1 + \alpha\mathbf{q}_1)$, calculation (10.13.3)–(10.13.4) shows that

$$\alpha_1 = \frac{(\mathbf{r}_1, \mathbf{q}_1)}{(A\mathbf{q}_1, \mathbf{q}_1)}.$$

(Note that the expression given above is not the same as that given in the algorithm. We will not be able to show that they are the same until after we determine $\mathbf{q}_1$.) We see that this choice of α_1 makes the residual $\mathbf{r}_2$ orthogonal to the search direction $\mathbf{q}_1$, i.e.,

$$\begin{aligned}(\mathbf{r}_2, \mathbf{q}_1) &= (\mathbf{r}_1, \mathbf{q}_1) - \alpha_1(A\mathbf{q}_1, \mathbf{q}_1)\\ &= (\mathbf{r}_1, \mathbf{q}_1) - \frac{(\mathbf{r}_1, \mathbf{q}_1)}{(A\mathbf{q}_1, \mathbf{q}_1)}(A\mathbf{q}_1, \mathbf{q}_1)\\ &= 0.\end{aligned}$$

We still get to choose $\mathbf{q}_1$. As a part of this choice, we set $\mathbf{q}_1 = \mathbf{r}_1 + \beta_0\mathbf{q}_0$ and require (instead of only minimizing $F(\mathbf{u}_1 + \alpha\mathbf{q}_1)$ as a function of α) that we minimize F over the set $\mathbf{u}_0 + \langle \mathbf{q}_0, \mathbf{q}_1 \rangle$ where $\langle \mathbf{q}_0, \mathbf{q}_1 \rangle$ denotes the span of $\mathbf{q}_0$ and $\mathbf{q}_1$. (We should realize that this could require us to change both α_0 and α_1.) To minimize F over $\mathbf{u}_0 + \langle \mathbf{q}_0, \mathbf{q}_1 \rangle$, we consider the function

$$g(\gamma, \delta) = F(\mathbf{u}_0 + \gamma\mathbf{q}_0 + \delta\mathbf{q}_1)$$

and set

$$\begin{aligned}0 &= \frac{\partial g}{\partial \gamma} \qquad (10.13.7)\\ &= F'(\mathbf{u}_0 + \gamma\mathbf{q}_0 + \delta\mathbf{q}_1) \cdot \mathbf{q}_0\\ &= (A(\mathbf{u}_0 + \gamma\mathbf{q}_0 + \delta\mathbf{q}_1) - \mathbf{f}, \mathbf{q}_0)\\ &= (-\mathbf{r}_0 + \gamma A\mathbf{q}_0 + \delta A\mathbf{q}_1, \mathbf{q}_0)\\ &= -(\mathbf{r}_0, \mathbf{q}_0) + \gamma(A\mathbf{q}_0, \mathbf{q}_0) + \delta(A\mathbf{q}_1, \mathbf{q}_0) \qquad (10.13.8)\end{aligned}$$

and

$$0 = \frac{\partial g}{\partial \delta} \tag{10.13.9}$$

$$= F'(\mathbf{u}_0 + \gamma\mathbf{q}_0 + \delta\mathbf{q}_1) \cdot \mathbf{q}_1$$

$$= (A(\mathbf{u}_0 + \gamma\mathbf{q}_0 + \delta\mathbf{q}_1) - \mathbf{f}, \mathbf{q}_1). \tag{10.13.10}$$

If we choose $\gamma = \alpha_0$ (as given by (10.13.2)), calculation (10.13.7)–(10.13.8) reduces to

$$\delta(A\mathbf{q}_1, \mathbf{q}_0) = 0. \tag{10.13.11}$$

When $\mathbf{q}_1$ and $\mathbf{q}_0$ satisfy $(A\mathbf{q}_1, \mathbf{q}_0) = 0$, $\mathbf{q}_1$ and $\mathbf{q}_0$ are said to be A-**conjugate**. Using the fact that we assumed that $\mathbf{q}_1$ was given by $\mathbf{q}_1 = \mathbf{r}_1 + \beta_0\mathbf{q}_0$, requiring that $\mathbf{q}_1$ and $\mathbf{q}_0$ be A-conjugate yields

$$0 = (A\mathbf{q}_1, \mathbf{q}_0) = (A\mathbf{r}_1, \mathbf{q}_0) + \beta_0(A\mathbf{q}_0, \mathbf{q}_0), \tag{10.13.12}$$

or

$$\beta_0 = -\frac{(A\mathbf{r}_1, \mathbf{q}_0)}{(A\mathbf{q}_0, \mathbf{q}_0)}. \tag{10.13.13}$$

Thus if we choose $\gamma = \alpha_0$, $\mathbf{q}_1 = \mathbf{r}_1 + \beta_0\mathbf{q}_0$ with β_0 given by (10.13.13), the function g will satisfy $g_\gamma = 0$.

With γ chosen to be equal to α_0, calculation (10.13.9)–(10.13.10) reduces to

$$0 = (A\mathbf{u}_1 + \delta A\mathbf{q}_1 - \mathbf{f}, \mathbf{q}_1) \tag{10.13.14}$$

$$= (-\mathbf{r}_1 + \delta A\mathbf{q}_1, \mathbf{q}_1). \tag{10.13.15}$$

Then it is easy to see that we must choose

$$\delta = \alpha_1 = \frac{(\mathbf{r}_1, \mathbf{q}_1)}{(A\mathbf{q}_1, \mathbf{q}_1)}. \tag{10.13.16}$$

(Hence, we see that we do not have to change α_0 or α_1. The only additional requirement needed to minimize F over $\mathbf{u}_0 + \langle \mathbf{q}_0, \mathbf{q}_1 \rangle$ was the choice of $\mathbf{q}_1$.)

And finally, it is not hard to see that our choice of α_0, α_1, β_0, and $\mathbf{q}_1$ also gives us that $\mathbf{r}_2$ *is orthogonal to* $\mathbf{q}_0$, $\mathbf{q}_0$ *and* $\mathbf{q}_1$ *are independent, and* $\mathbf{q}_1$ *is not orthogonal to* $\mathbf{r}_1$. See HW10.13.1.

Remark 4: We note that the expressions given for α_1 and β_0 do not correspond to those given in the conjugate gradient algorithm. Both expressions are true, but those given in the algorithm require fewer calculations than those given in (10.13.13) and (10.13.16). To see that α_1 can be written in either form, we note that

$$(\mathbf{r}_1, \mathbf{q}_1) = (\mathbf{r}_1, \mathbf{r}_1 + \beta_0\mathbf{q}_0)$$

$$= (\mathbf{r}_1, \mathbf{r}_1) + \beta_0(\mathbf{r}_1, \mathbf{q}_0)$$

$$= (\mathbf{r}_1, \mathbf{r}_1). \tag{10.13.17}$$

Also, since

$$(\mathbf{r}_1, \mathbf{r}_1) = (\mathbf{r}_0, \mathbf{r}_1) - \alpha_0 (A\mathbf{q}_0, \mathbf{r}_1)$$
$$= -\frac{(\mathbf{r}_0, \mathbf{r}_0)}{(A\mathbf{q}_0, \mathbf{q}_0)} (A\mathbf{q}_0, \mathbf{r}_1) \qquad ((\mathbf{r}_0, \mathbf{r}_1) = 0 \text{ and definition of } \alpha_0),$$

$$\frac{(\mathbf{r}_1, \mathbf{r}_1)}{(\mathbf{r}_0, \mathbf{r}_0)} = -\frac{(A\mathbf{q}_0, \mathbf{r}_1)}{(A\mathbf{q}_0, \mathbf{q}_0)} = \beta_0 \quad \text{(using the symmetry of } A\text{)}.$$

Remark 5: As we proceed with the algorithm, at the end of Step 2 for any k, we will have

- approximations to the solution $\mathbf{u}_0, \ldots, \mathbf{u}_{k+1}$;
- residuals $\mathbf{r}_0, \ldots, \mathbf{r}_{k+1}$ that satisfy
  $(\mathbf{r}_j, \mathbf{r}_m) = 0$, $j \neq m$, $j, m = 1, \ldots, k+1$, and $(\mathbf{r}_j, \mathbf{r}_j) \neq 0$, $j = 0, \ldots, k+1$;
- search directions $\mathbf{q}_0, \ldots, \mathbf{q}_{k+1}$ that satisfy
  * $(A\mathbf{q}_m, \mathbf{q}_j) = 0$, $j = 1, \ldots, m-1$ ($\mathbf{q}_m$ is A-conjugate to $\mathbf{q}_j$ for $j = 1, \ldots, m-1$) for each $m = 1, \ldots, k+1$;
  * $\mathbf{q}_0, \ldots, \mathbf{q}_{k+1}$ are independent ($\langle \mathbf{q}_0, \ldots, \mathbf{q}_{k+1} \rangle$ defines a $(k+2)$ dimensional space);
  * $(\mathbf{q}_j, \mathbf{r}_m) = 0$ for $j = 0, \ldots, m-1$, $(\mathbf{q}_m, \mathbf{r}_m) \neq 0$ for $m = 0, \ldots, k+1$.

Since $\mathbf{q}_0, \ldots, \mathbf{q}_{k+1}$ are independent, either there exists a k such that $\mathbf{r}_k = \boldsymbol{\theta}$ (and we are done) or we eventually compute $\mathbf{u}_{L-1}$ that minimizes F over $\mathbf{u}_0 + \langle \mathbf{q}_0, \ldots, \mathbf{q}_{L-1} \rangle = \mathbb{R}^L$. In either case (assuming infinite precision arithmetic), we have found the solution to equation (10.13.1), and we have the following result.

Proposition 10.13.1 *The conjugate gradient scheme will converge to the solution of equation (10.13.1) in L or fewer steps.*

The above result is not adequate. Most often, L will be too large to even consider taking L conjugate gradient steps to obtain a solution. In addition, as we see in HW10.13.2, there are times when the m-th iterate is a good approximation to the solution of the problem (where m is much smaller than L), but if we continue to the L-th iterate, because we are working with inexact arithmetic we get nonsense. Though the conjugate gradient scheme converges in L or fewer steps, the conjugate gradient scheme is useful because it (or variations of the conjugate gradient scheme) approximates the solution well in many fewer than L steps. The speed of convergence of the conjugate gradient scheme depends strongly on the **condition number**

of the matrix, $\kappa(A) = |\mu_L|/|\mu_1|$, where μ_L and μ_1 are, respectively, the largest and smallest eigenvalues of A and can be given in terms of the A-norm

$$\|\mathbf{u}\|_A = \sqrt{(A\mathbf{u}, \mathbf{u})}$$

(which is a norm, since A is assumed to be positive definite) by the following result.

Proposition 10.13.2 *Suppose A is a symmetric, positive, $L \times L$ matrix, $\mathbf{u}$ is the solution to equation (10.13.1), $\kappa(A)$ is the condition number of A, and $\mathbf{u}_k$ is the k-th iterate of the conjugate gradient scheme. Then*

$$\|\mathbf{u} - \mathbf{u}_k\|_A \leq 2\left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1}\right)^k \|\mathbf{u} - \mathbf{u}_0\|_A. \tag{10.13.18}$$

Proof: [2], page 26.

Thus we see that if $(\sqrt{\kappa(A)} - 1)/(\sqrt{\kappa(A)} + 1)$ is small, the conjugate gradient scheme converges very fast. When the conjugate gradient scheme is implemented, based on Proposition 10.13.2 it is logical to use the A-norm as a part of our stopping criterion. There is a variety of good stopping criteria for the conjugate gradient scheme, several given in terms of the A-norm. We shall use the approach of monitoring both $\mathbf{r}_k$ and $\alpha\mathbf{q}_k = \mathbf{u}_{k+1} - \mathbf{u}_k$ using either the finite dimensional l_2 or the sup-norm. The advantage of using this approach is that it is elementary, yet effective.

If we were to do some computational experiments, it would be seen that the conjugate gradient scheme will converge in fewer iterations that the SOR scheme (and we do not have to determine any optimal parameters). However, if we timed our runs or did some sort of operation count, we would see that each iteration of the conjugate gradient scheme takes about twice as much work as an iteration of the SOR scheme, so that the SOR scheme actually converges faster than the conjugate gradient scheme. For this reason, we next introduce the variation of the conjugate gradient scheme called the preconditioned conjugate gradient scheme.

HW 10.13.1 Prove that
(a) $\mathbf{r}_2$ is orthogonal to $\mathbf{q}_0$,
(b) $\mathbf{q}_0$ and $\mathbf{q}_1$ are independent, and
(c) $\mathbf{q}_1$ is not orthogonal to $\mathbf{r}_1$.

10.13.1 Preconditioned Conjugate Gradient Scheme

From equation (10.13.18) we see that the convergence of the conjugate gradient scheme gets better if the condition number gets closer to one. Further analysis of the conjugate gradient scheme shows that clustering together the eigenvalues of a matrix makes the convergence of the conjugate

gradient scheme faster. One way to try to effect this change is to multiply equation (10.13.1) by some matrix M^{-1} to get

$$M^{-1}A\mathbf{u} = M^{-1}\mathbf{f}. \tag{10.13.19}$$

The plan would be to choose M so that the condition number of $M^{-1}A$ is nearer to one than the condition number of A and use the conjugate gradient scheme on this new equation. However, unless we are lucky, *even if we choose M to be symmetric, $M^{-1}A$ will not be symmetric (and, hence, the conjugate gradient scheme is not applicable).*

The approach used is to choose a matrix $C = MM^T$ such that

$$\kappa(M^{-1}AM^{-T}) < \kappa(A),$$

where $M^{-T} = \left(M^{-1}\right)^T$. We then apply the conjugate gradient scheme to

$$\tilde{A}\tilde{\mathbf{u}} = \tilde{\mathbf{f}}, \tag{10.13.20}$$

where

$$\tilde{A} = M^{-1}AM^{-T}, \ \tilde{\mathbf{u}} = M^T\mathbf{u}, \quad \text{and } \tilde{\mathbf{f}} = M^{-1}\mathbf{f}.$$

Equation (10.13.20) is algebraically equivalent to equation (10.13.19) (and, hence, equation (10.13.1)), and $\tilde{A}$ is symmetric if A is symmetric. We will discuss how to choose C and M in later sections.

If we apply the conjugate gradient scheme to equation (10.13.20), we have the following version of Conjugate Gradient-10.13.1.

Conjugate Gradient-10.13.20

Step 1: Choose $\tilde{\mathbf{u}}_0$. Set $\tilde{\mathbf{r}}_0 = \tilde{\mathbf{f}} - \tilde{A}\tilde{\mathbf{u}}_0$, $k = 0$, and $\tilde{\mathbf{q}}_0 = \tilde{\mathbf{r}}_0$.

Step 2: $\tilde{\alpha}_k = \frac{(\tilde{\mathbf{r}}_k, \tilde{\mathbf{r}}_k)}{(\tilde{A}\tilde{\mathbf{q}}_k, \tilde{\mathbf{q}}_k)}$

$\tilde{\mathbf{u}}_{k+1} = \tilde{\mathbf{u}}_k + \tilde{\alpha}_k\tilde{\mathbf{q}}_k$

$\tilde{\mathbf{r}}_{k+1} = \tilde{\mathbf{f}} - \tilde{A}\tilde{\mathbf{u}}_{k+1} = \tilde{\mathbf{r}}_k - \tilde{\alpha}_k\tilde{A}\tilde{\mathbf{q}}_k$

Quit if $\tilde{\mathbf{r}}_{k+1}$ is zero or sufficiently small

$\tilde{\beta}_k = -\frac{(\tilde{\mathbf{r}}_{k+1}, \tilde{\mathbf{r}}_{k+1})}{(\tilde{\mathbf{r}}_k, \tilde{\mathbf{r}}_k)}$

$\tilde{\mathbf{q}}_{k+1} = \tilde{\mathbf{r}}_{k+1} + \tilde{\beta}_k\tilde{\mathbf{q}}_k$

Step 3: $k = k + 1$; Go to Step 2

We do not want to proceed in this manner, i.e., actually transform equation (10.13.1) into equation (10.13.20) by computing $\tilde{A}$ and $\tilde{\mathbf{f}}$. We instead want to be able to write algorithm Conjugate Gradient-10.13.20 in terms of the original variables $\mathbf{u}$, $\mathbf{r}$, $\mathbf{q}$, etc. We note that in constructing equation (10.13.20) we have made the following substitutions:

$$\tilde{\mathbf{u}} = M^T\mathbf{u}, \ \tilde{\mathbf{f}} = M^{-1}\mathbf{f}, \quad \text{and } \tilde{A} = M^{-1}AM^{-T}.$$

Using these substitutions, an easy calculation shows that

$$\tilde{\mathbf{r}}_k = \tilde{\mathbf{f}} - \tilde{A}\tilde{\mathbf{u}}_k = M^{-1}\mathbf{f} - \left(M^{-1}AM^{-T}\right)M^T\mathbf{u}_k = M^{-1}\left(\mathbf{f} - A\mathbf{u}_k\right) = M^{-1}\mathbf{r}_k,$$

or

$$\tilde{\mathbf{r}} = M^{-1}\mathbf{r}.$$

One last relationship between the "wigl" variables and the basic variables that is necessary is the relationship between $\tilde{\mathbf{q}}$ and $\mathbf{q}$. The relationship that we use is not one that is forced on us, but is one that gives us a nice preconditioned conjugate gradient algorithm. We let

$$\tilde{\mathbf{q}} = M^T\mathbf{q}.$$

We now return to algorithm Conjugate Gradient-10.13.20. We see that Step 1 is the same as

Choose $\mathbf{u}_0$. (This will then choose $\tilde{\mathbf{u}}_0 = M^T\mathbf{u}_0$.)

Set $\mathbf{r}_0 = \mathbf{f} - A\mathbf{u}_0$, $\tilde{\mathbf{r}}_0 = M^{-1}\mathbf{r}_0$, $\tilde{\mathbf{q}}_0 = \tilde{\mathbf{r}}_0$, and $\mathbf{q}_0 = M^{-T}\tilde{\mathbf{q}}_0$.

In Step 2, $\tilde{\alpha}_k$ can be computed as

$$\begin{aligned}
\tilde{\alpha}_k &= \frac{(\tilde{\mathbf{r}}_k, \tilde{\mathbf{r}}_k)}{(\tilde{A}\tilde{\mathbf{q}}_k, \tilde{\mathbf{q}}_k)} \\
&= \frac{(M^{-1}\mathbf{r}_k, M^{-1}\mathbf{r}_k)}{(M^{-1}AM^{-T}M^T\mathbf{q}_k, M^T\mathbf{q}_k)} \\
&= \frac{(\mathbf{r}_k, M^{-T}M^{-1}\mathbf{r}_k)}{(A\mathbf{q}_k, M^{-T}M^T\mathbf{q}_k)} \\
&= \frac{(\mathbf{r}_k, M^{-T}M^{-1}\mathbf{r}_k)}{(A\mathbf{q}_k, \mathbf{q}_k)}.
\end{aligned} \tag{10.13.21}$$

When we compute the new value of $\tilde{\mathbf{u}}_{k+1}$ in Step 2, it is the same as computing a new value of $\mathbf{u}_{k+1}$ of the form

$$\begin{aligned}
\tilde{\mathbf{u}}_{k+1} &= M^T\mathbf{u}_{k+1} = \tilde{\mathbf{u}}_k + \tilde{\alpha}_k\tilde{\mathbf{q}}_k \\
&= M^T\mathbf{u}_k + \tilde{\alpha}_k M^T\mathbf{q}_k \\
&= M^T\left(\mathbf{u}_k + \tilde{\alpha}_k\mathbf{q}_k\right),
\end{aligned}$$

or

$$\mathbf{u}_{k+1} = \mathbf{u}_k + \tilde{\alpha}_k\mathbf{q}_k. \tag{10.13.22}$$

Likewise, computing $\tilde{\mathbf{r}}_{k+1}$ can be transformed back to the basic variables by

$$\begin{aligned}
\tilde{\mathbf{r}}_{k+1} &= M^{-1}\mathbf{r}_{k+1} = \tilde{\mathbf{r}}_k - \tilde{\alpha}_k\tilde{A}\tilde{\mathbf{q}}_k \\
&= M^{-1}\mathbf{r}_k - \tilde{\alpha}_k M^{-1}AM^{-T}M^T\mathbf{q}_k \\
&= M^{-1}\left(\mathbf{r}_k - \tilde{\alpha}_k A\mathbf{q}_k\right),
\end{aligned}$$

or

$$\mathbf{r}_{k+1} = \mathbf{r}_k + \tilde{\alpha}_k \mathbf{q}_k. \tag{10.13.23}$$

Analogous to the result in the numerator of the $\tilde{\alpha}$ calculation, $\tilde{\beta}_k$ can be written as

$$\begin{aligned}\tilde{\beta}_k &= -\frac{(\tilde{\mathbf{r}}_{k+1}, \tilde{\mathbf{r}}_{k+1})}{(\tilde{\mathbf{r}}_k, \tilde{\mathbf{r}}_k)} \\ &= -\frac{(\mathbf{r}_{k+1}, M^{-T}M^{-1}\mathbf{r}_{k+1})}{(\mathbf{r}_k, M^{-T}M^{-1}\mathbf{r}_k)}.\end{aligned} \tag{10.13.24}$$

And finally, the $\tilde{\mathbf{q}}_{k+1}$ calculation can be transformed by

$$\begin{aligned}\tilde{\mathbf{q}}_{k+1} &= M^T\mathbf{q}_{k+1} = \tilde{\mathbf{r}}_{k+1} + \tilde{\beta}_k\tilde{\mathbf{q}}_k \\ &= M^{-1}\mathbf{r}_{k+1} + \tilde{\beta}_k M^T\mathbf{q}_k, \\ &= M^T\left(M^{-T}M^{-1}\mathbf{r}_{k+1} + \tilde{\beta}_k\mathbf{q}_k,\right.\end{aligned}$$

or

$$\mathbf{q}_{k+1} = M^{-T}M^{-1}\mathbf{r}_{k+1} + \tilde{\beta}_k\mathbf{q}_k. \tag{10.13.25}$$

It should be clear that the computations performed in algorithm Conjugate Gradient-10.13.20 can be rewritten in terms of the basic variables using the changes given in Step 1 and equations (10.13.21)–(10.13.25). In this way we write the preconditioned conjugate gradient algorithm as follows.

Preconditioned Conjugate Gradient-10.13.1

Step 1: Choose $\mathbf{u}_0$. Set $\mathbf{r}_0 = \mathbf{f} - A\mathbf{u}_0$, $\mathbf{r}_0^* = C^{-1}\mathbf{r}_0 = M^{-T}M^{-1}\mathbf{r}_0$, $k = 0$, and $\mathbf{q}_0 = \mathbf{r}_0^*$.

Step 2: $\tilde{\alpha}_k = \frac{(\mathbf{r}_k, \mathbf{r}_k^*)}{(A\mathbf{q}_k, \mathbf{q}_k)}$
$\mathbf{u}_{k+1} = \mathbf{u}_k + \tilde{\alpha}_k\mathbf{q}_k$
$\mathbf{r}_{k+1} = \mathbf{f} - A\mathbf{u}_{k+1} = \mathbf{r}_k - \tilde{\alpha}_k A\mathbf{q}_k$
Quit if $\mathbf{r}_{k+1}$ is zero or sufficiently small
$\mathbf{r}_{k+1}^* = C^{-1}\mathbf{r}_{k+1} = M^{-T}M^{-1}\mathbf{r}_{k+1}$
$\tilde{\beta}_k = -\frac{(\mathbf{r}_{k+1}, \mathbf{r}_{k+1}^*)}{(\mathbf{r}_k, \mathbf{r}_k^*)}$
$\mathbf{q}_{k+1} = \mathbf{r}_{k+1}^* + \tilde{\beta}_k\mathbf{q}_k$

Step 3: $k = k + 1$; Go to Step 2

We note that application of the preconditioned conjugate gradient is not much more work than the application of the conjugate gradient scheme. The difference is that for each step of the preconditioned conjugate gradient scheme, $C^{-1}\mathbf{r}_{k+1}$ must be computed. The obvious approach is to *choose C (and/or M) so that $\kappa(M^{-1}AM^{-T})$ is small and $C^{-1}\mathbf{r}_{k+1}$ is easy to compute*. Clearly, the choice of C is a very important part of obtaining good results with the preconditioned conjugate gradient scheme.

10.13.2 SSOR as a Preconditioner

It should be reasonably clear from the last section that for the preconditioned conjugate gradient scheme to be successful, it is important that we choose the correct preconditioner. We want to choose a preconditioner so that $\kappa(\bar{A}) \approx 1$. Since there is not a "correct" preconditioner, the job of choosing the correct preconditioner is not easy. There are many ways to choose preconditioners. Usually, each of these different preconditioners works best in some situation. All of these preconditioners approximate A^{-1} to some degree. If we could choose C so that $\bar{A} = I$ (we would be done), we would have $\kappa(\bar{A}) = 1$. Of course, we cannot do this. Instead, we choose preconditioners that crudely approximate A^{-1}, which in the process makes $\kappa(\bar{A})$ near one.

As the title of this section should suggest, in this section we will use SSOR as a preconditioner. We choose SSOR as our model preconditioner both because it is a good general preconditioner and because with our approach it is easy to implement. Since the SSOR scheme is a potential solver for the problem, it should be clear that the SSOR scheme will provide us with an approximation of A^{-1}. For information on other preconditioners, see [13], page 527, or [2], page 30.

We begin by emphasizing that we consider a matrix A that is symmetric and positive definite. As usual, we write A as $L + D + U$ and note that since A is positive definite, the elements on the diagonal D are positive.

To find a preconditioner, we must find a matrix $C = MM^T$. If we return to the residual correction form of SSOR, we see from equation (10.5.69) that SSOR can be written as

$$\mathbf{w}_{k+1} = \mathbf{w}_k + B\mathbf{r}_k,$$

where $B = \omega(2-\omega)(D+\omega U)^{-1}D(D+\omega L)^{-1}$. Thus, when we apply SSOR, we are solving the equation

$$B^{-1}(\mathbf{w}_{k+1} - \mathbf{w}_k) = \frac{1}{\omega(2-\omega)}(D+\omega L)D^{-1}(D+\omega U)(\mathbf{w}_{k+1} - \mathbf{w}_k) = \mathbf{r}_k.$$

When A is symmetric, $U = L^T$. When we have symmetry and the elements of D are positive, B^{-1} can be written as $B^{-1} = MM^T$, where

$$M = \frac{1}{\sqrt{\omega(2-\omega)}}(D+\omega L)D^{-1/2} \tag{10.13.26}$$

and $D^{-1/2}$ is the diagonal matrix with the square root of the diagonal elements of D on the diagonal.

The fact that B^{-1} can be written as MM^T is necessary for us to be able to use $C = B^{-1}$ as our preconditioner, but M and M^T are not a necessary part of the solution process. The one step in the preconditioned conjugate gradient scheme that involves the preconditioner is the

step $\mathbf{r}^*_{k+1} = C^{-1}\mathbf{r}_{k+1}$. Hence, we must solve the equation

$$C\mathbf{r}^*_{k+1} = \frac{1}{\omega(2-\omega)}(D+\omega L)D^{-1}(D+\omega L^T)\mathbf{r}^*_{k+1} = \mathbf{r}_{k+1}.$$

The solution to this equation can be expressed as

$$(D+\omega L)\hat{\mathbf{r}}^* = \omega(2-\omega)\mathbf{r}_{k+1} \tag{10.13.27}$$

$$(D+\omega L^T)\mathbf{r}^*_{k+1} = D\hat{\mathbf{r}}^*. \tag{10.13.28}$$

Thus we see that to use SSOR as our preconditioner, we choose ω according to formula (10.5.72) and solve equations (10.13.27)–(10.13.28). We note that since $D+\omega L$ and $D+\omega L^T$ are lower and upper triangular matrices, respectively, solving equation (10.13.27) involves only a forward sweep, and solving equation (10.13.28) involves only a backward sweep (not surprising, considering the two sweeps involved in the implementation of the SSOR scheme).

As we stated earlier, there are many other preconditioners available. The choice of preconditioner is very important. We might mention that as a part of the introduction to the conjugate gradient scheme, we emphasized that one of the nice aspects of the conjugate gradient scheme was that we did not have to calculate any iteration parameters. However, now we see that to be competitive, when using the conjugate gradient scheme we must make a decision on which preconditioner to use. This is an illustration of the idea that we have seen several other times earlier in this text that there is no one best way to solve most of these problems. The choice of which scheme to use depends on the problem, what software is available, the number of times that the particular implementation will be run, the machine being used, and the background of the user. All of these schemes are good schemes.

10.13.3 Implementation

Since the conjugate gradient scheme is included in the preconditioned conjugate gradient scheme, we shall consider only the implementation of the preconditioned conjugate gradient scheme. To change the implementation described below to the conjugate gradient scheme, we skip the preconditioning step (solving $C\mathbf{r}^*_{k+1} = \mathbf{r}_{k+1}$) and replace all of the $\mathbf{r}^*$'s in the formulas by $\mathbf{r}$'s.

We consider solving a system of equations of the form (10.13.1) that results from solving a general two dimensional difference equation of the form (10.5.13) along with Dirichlet boundary conditions. Of course, since A must be symmetric in order to apply the preconditioned conjugate gradient scheme, the appropriate symmetry assumptions must be made on the β terms. To apply the preconditioned conjugate gradient scheme, we must perform the following computations.

1. After choosing our initial guess $\mathbf{u}_0$, we must compute the residual $\mathbf{f} - A\mathbf{u}_0$, solve $C\mathbf{r}_0^* = \mathbf{r}_0$ for $\mathbf{r}_0^*$, and compute $(\mathbf{r}_0, \mathbf{r}_0^*)$.

2. Then for each iteration, we must compute

   (a) the matrix multiplication $A\mathbf{q}_k$,

   (b) the dot products $(A\mathbf{q}_k, \mathbf{q}_k)$ and $(\mathbf{r}_{k+1}, \mathbf{r}_{k+1}^*)$,

   (c) the solution to $C\mathbf{r}_{k+1}^* = \mathbf{r}_{k+1}$,

   (d) two operations that involve a scalar multiplication of a vector and a vector addition, $\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha A\mathbf{q}_k$ and $\mathbf{q}_{k+1} = \mathbf{r}_{k+1}^* + \beta_k \mathbf{q}_k$,

   (e) plus a few scalar multiplications and divisions.

There are at least two obvious ways that we can approach the implementation of the preconditioned conjugate gradient scheme. Often, it depends somewhat on our choice of preconditioner (C) on which approach might be best. The first approach, which we will refer to as the **matrix approach**, is to use the β's given in equation (10.5.13) to build the matrix A. Obviously, we do not fill A as a huge $L \times L$ matrix. Rather, we fill five one dimensional arrays with dimension L, each one representing one of the nonzero diagonals of the A matrix. When filling the super and subdiagonals of A, we must remember to place zeros in the appropriate places due to the difference scheme reaching to the boundary.

To compute $\mathbf{r}_0$, we must also fill one array with our initial guess $\mathbf{u}_0$ and another with the right hand side of our matrix equation, $\mathbf{f}$ (remembering that in $\mathbf{f}$ there are contributions from both the right hand side of the difference equation, $F_{j\,k}$ and the boundary conditions). Of course, the array for $\mathbf{f}$ is a temporary array. We can then define four one dimensional arrays with dimension L for $\mathbf{r}$, $\mathbf{r}^*$, $\mathbf{q}$, and $A\mathbf{q}$ and apply the algorithm.

Another approach to implementation, which we will refer to as the **physical space approach**, is based on the stencil arrays used with the relaxation schemes. We write our initial $\mathbf{u}_0$ and $F_{j\,k}$ as two $(M_x+1) \times (M_y+1)$ arrays (instead of long one dimensional arrays). Note that there is room in the $\mathbf{u}_0$ array for the boundary conditions. We also write $\mathbf{r}$, $\mathbf{r}^*$, $\mathbf{q}$, and $A\mathbf{q}$ as $(M_x+1) \times (M_y+1)$ arrays. As we shall see later, we want the boundary cells for the $\mathbf{r}$, $\mathbf{r}^*$, $\mathbf{q}$, and $A\mathbf{q}$ arrays to be filled with zeros.

The computation of the initial residual involves the following loop.

For $k = 1, M_y - 1$
For $j = 1, M_x - 1$

$$r_{j\,k} = F_{j\,k} - \left[\beta^1_{j\,k} u_{j+1\,k} + \beta^2_{j\,k} u_{j-1\,k} + \beta^3_{j\,k} u_{j\,k+1} + \beta^4_{j\,k} u_{j\,k-1} - \beta^0_{j\,k} u_{j\,k}\right]$$

Next j
Next k

We note that with this approach, the contribution to $\mathbf{f}$ due to the right hand side of the difference equation is included in the F_{jk} term, and the boundary conditions are included when the scheme reaches to the boundary cells.

In a like manner, it is easy to see that the computation of $A\mathbf{q}$ is obtained by the following loop.

For $k = 1, M_y - 1$
For $j = 1, M_x - 1$
$Aq_{jk} = \beta^1_{jk} q_{j+1\,k} + \beta^2_{jk} q_{j-1\,k} + \beta^3_{jk} q_{j\,k+1} + \beta^4_{jk} q_{j\,k-1} - \beta^0_{jk} q_{jk}$
Next j
Next k

We see that for this computation, *it is important that we have included and zeroed out the boundary cells of the* $\mathbf{q}$ *array.* This computation involves just the matrix A: We do not want any contributions from the stencil that reaches to the boundary points. The dot product computations are performed with similar double loops over the indices $k = 1, \ldots, M_y - 1$, $j = 1, \ldots, M_x - 1$.

And finally, in a similar manner we see that when SSOR is used as the preconditioner using the physical space approach, equations (10.13.27) and (10.13.28) can be solved by the following loops.

For $k = 1, \ldots, M_y - 1$
For $j = 1, \ldots, M_x - 1$
$\hat{r}_{jk} = -\frac{1}{\beta^0_{jk}} \left[\omega(2-\omega) r_{jk} - \omega \left(\beta^2_{jk} \hat{r}_{j-1\,k} + \beta^4_{jk} \hat{r}_{j\,k-1} \right) \right]$
Next j
Next k

For $k = M_y - 1, \ldots, 1$
For $j = M_x - 1, \ldots, 1$
$r^*_{jk} = -\frac{1}{\beta^0_{jk}} \left[\beta^0_{jk} \hat{r}_{jk} - \omega \left(\beta^1_{jk} r^*_{j+1\,k} + \beta^3_{jk} r^*_{j\,k+1} \right) \right]$
Next j
Next k

Of the two approaches, the matrix approach is the most common. One of the reasons for this is that the scheme is usually given in terms of matrices, but the real advantage to the matrix approach is that the matrix approach may be better for canned, multipurpose software. The physical space approach is probably better suited for readers of this text, since it uses the same data structures that we have been using for other two dimensional schemes. It should be clear that the physical space approach makes it easier to use SSOR as the preconditioner. With some of the other more commonly used preconditioners, it is more convenient to use the matrix approach.

HW 10.13.2 (a) Use the conjugate gradient scheme to approximate the solution to the boundary value problem

$$\begin{aligned} \nabla^2 v &= 0, \quad (x,y) \in (0,1) \times (0,1) \\ v(0,y) &= 1.0, \quad y \in [0,1] \\ v(1,y) &= 0.0, \quad y \in [0,1] \\ v(x,0) &= 1.0, \quad x \in [0,1] \\ v(x,1) &= 0.0, \quad x \in [0,1] \end{aligned}$$

Use $M_x = M_y = 50$ and the residual with a tolerance of 5.0×10^{-5} as a stopping criterion.
(b) Resolve the problem given in part (a) by doing 2500 conjugate gradient steps (no other stopping criterion—see Proposition 10.13.1). Compare and contrast your results with those found in part (a).

HW 10.13.3 (a) Use the conjugate gradient scheme to approximate the solution to the problem given in HW10.5.2 (and solved by most other methods throughout this chapter). Compare and contrast your results with the solutions (number of iterations, computer time, etc.) obtained by the other methods (HW10.16.1, HW10.5.14, HW10.12.2, etc.).

(b) Apply the conjugate gradient scheme to the problem discussed in part (a) for $(M_x - 1)(M_y - 1)$. Is the result better? As good?

(c) Apply the preconditioned conjugate gradient scheme (preconditioned with SSOR) to the problem discussed in part (a).

HW 10.13.4 Use the preconditioned conjugate gradient scheme (preconditioned with SSOR) to approximate the solution of the problem given in HW10.5.5.

10.14 Using Iterative Methods to Solve Time Dependent Problems

If we review the numerical schemes that we have developed in this chapter and/or refer to some of the references for these schemes, we find that most of the time we are really developing methods for solving a linear equation $A\mathbf{u} = \mathbf{f}$ and applying these methods to solve difference equations associated with elliptic partial differential equations. (Probably the main exception is the ADI scheme developed in Section 10.12, which was an adaptation of a numerical scheme for solving parabolic problems.) It should not be surprising that we would like to try to use these same iterative methods

to solve the equations resulting from implicit schemes for time dependent problems.

If we first consider a two dimensional parabolic equations of the form

$$v_t = v_{xx} + v_{yy}$$

and use a backward time, center space scheme (a Crank-Nicolson scheme would give use the same results), the linear system of equations that we must solve is the broadly banded matrix given by (4.3.27). The difficulty of solving these equations is the reason we developed alternating direction implicit schemes. A comparison of the matrix Q_1 given in (4.3.27) with the matrix A given in (10.2.8) shows that the matrix equation associated with an implicit scheme for a time dependent problem is very similar to that associated with an elliptic equation. A more careful inspection of Q_1 shows that the matrix Q_1 is nicer than the matrix given in (10.2.8). The 1 on the diagonal of Q_1 due to the approximation of v_t makes the matrix strictly diagonally dominant. Hence, we know immediately by Proposition 10.2.3 that the matrix Q_1 is invertible. Likewise, we can use an analysis analogous to that used in Remark 4, page 315, to show that $\|R_J\|_\infty < 1$, where R_J is the Jacobi iteration matrix associated with solving equation (4.3.27). Hence, by Proposition 10.5.1 the Jacobi scheme applied to solve equation (4.3.27) will converge.

Likewise, the Gauss-Seidel scheme can be used to solve equation (4.3.27). In either case, neither the Jacobi nor the Gauss-Seidel scheme will come close to competing with an ADI scheme for approximating the solution of the problem. Using an SOR scheme will come much closer to being competitive with an ADI scheme. As can be seen in HW10.14.1, it is possible to find the optimal parameter for using SOR to solve equation (4.3.27).

And finally, it is also possible to solve equation (4.3.27) using either a conjugate gradient scheme or a preconditioned conjugate gradient scheme.

We should notice that everything that has been said above will also hold for three dimensional equations. If we include some lower order terms in the equation above, i.e., consider approximating the solution to a parabolic partial differential equation of the form

$$v_t + av_x + bv_y = v_{xx} + v_{yy}, \tag{10.14.1}$$

we must be more careful. Again using an argument analogous to that made in Remark 4, page 315, we see that for Δx and Δy sufficiently small, the matrix associated with either the BTCS or Crank-Nicolson scheme will be strictly diagonally dominant, and the sup-norm of the iteration matrix for either the Jacobi or Gauss-Seidel scheme will be less than one. Approximating the solution of equation (10.14.1) by iterative methods is much like using iterative methods to solve difference equations like (10.2.10), except that when we solve difference equation (10.14.1), we must use the iterative scheme at each time step. It should be reasonably clear that it would be

either very difficult or impossible to find the optimal parameter for solving the difference equation associated with equation (10.14.1) by SOR. Hence, if an equation such as (10.14.1) is to be solved by SOR, it may be necessary to use the methods for approximating the optimal parameter ω_b.

And finally, it should be reasonably clear that we cannot use either the conjugate gradient scheme or the preconditioned conjugate gradient scheme to solve a difference equation associated with equation (10.14.1). For the application of the conjugate gradient scheme as it is given in Section 10.13, it is essentially necessary that the matrix be symmetric (some slight asymmetries may be tolerable). However, there are conjugate gradient schemes for nonsymmetric problems. Although these schemes are more expensive than the symmetric conjugate gradient schemes, they can be used to approximate solutions to equations such as (10.14.1).

Iterative methods are not generally well suited for hyperbolic problems. In the first place, many people do not like to use implicit schemes for hyperbolic problems. In addition, when the problems are considered, they are never symmetric (so for example, it would be impossible to use our conjugate gradient scheme for solving hyperbolic problems). However, there are ranges for which the Jacobi and Gauss-Seidel schemes will converge when used to solve equations associated with hyperbolic problems.

One might ask why one should try to use iterative methods for solving time dependent problems when we already have ways for solving these problems that are usually faster than the iterative methods. There are times when using an iterative method is the easiest approach. Even if it might take more computer time, if it takes less time for the user, it may be worthwhile. Other times, especially when a very good preconditioner is available, iterative schemes can be competitive with other methods.

HW 10.14.1 Find the optimal parameter associated with solving equation (4.3.27) by SOR.

10.15 Using FFTs to Solve Elliptic Problems

In this section we introduce another technique that is a common method for solving elliptic boundary value problems (and is used for other types of problems). The abbreviation FFT traditionally stands for the "fast Fourier transform." In this introduction, FFT will denote either finite Fourier transform or fast Fourier transform. Usually, while we develop the method and maybe even for our first implementations of the method, we will use the **finite** Fourier transform. When we get down to serious computing, we will want to use the **fast** Fourier transform. This should not cause a problem.

Before we proceed, we should mention that there is a large amount of work done using spectral methods for approximating the solutions to initial–boundary–value problems and boundary–value problems. See, for

example, ref. [9]. We include this short introduction to using FFTs for approximating the solutions to elliptic boundary–value problems because there is a subset of the people working in finite difference methods who routinely use the methods given here to approximate solutions to certain elliptic boundary–value problems.

We consider the elliptic boundary value problem

$$-\nabla^2 v = F, \quad (x,y) \in R = (0,1) \times (0,1) \tag{10.15.1}$$

$$v = 0 \quad \text{on } \partial R \tag{10.15.2}$$

and the numerical analogue

$$-\left(\frac{1}{\Delta x^2}\delta_x^2 + \frac{1}{\Delta y^2}\delta_y^2\right) u_{jk} = F_{jk}, \quad j = 1, \ldots, M_x - 1,$$

$$k = 1, \ldots, M_y - 1 \tag{10.15.3}$$

$$u_{0k} = u_{M_x k} = 0, \quad k = 0, \ldots, M_y \tag{10.15.4}$$

$$u_{j0} = u_{jM_y} = 0, \quad j = 0, \ldots, M_x. \tag{10.15.5}$$

We recall from Section 3.2 that if we consider a grid G consisting of the points $x_0 = 0$, $x_1 = \Delta x$, $\ldots$, $x_{M_x} = 1$, basis functions $\phi_j = e^{ij\pi x}$ defined on G, and a function f defined on G with $f(0) = f(1) = 0$, then f can be written as

$$f(x) = \sum_{j=-k_0}^{k_0+\theta} c_j e^{ij\pi x}, \tag{10.15.6}$$

where

$$c_j = \frac{1}{M_x}(f, \phi_j) = \frac{1}{M_x} \sum_{m=0}^{M_x-1} f(m\Delta x) e^{-ij\pi m \Delta x}, \tag{10.15.7}$$

and $\theta = 0$ and $k_0 = (M_x - 1)/2$ if M_x is odd, and $\theta = 1$ and $k_0 = (M_x - 2)/2$ if M_x is even. Note that when M_x is odd, the sum goes from $-(M_x - 1)/2$ to $(M_x - 1)/2$, and when M_x is even, the sum goes from $-(M_x - 2)/2$ to $M_x/2$. Since the notation in the odd case is a little bit cleaner, *we will assume for this discussion that M_x and M_y are odd.* The sequence $\{c_{-(M_x-1)/2}, \ldots, c_{(M_x-1)/2}\}$ is referred to as the **finite Fourier transform of** f, and the elements will be written as $\hat{f}_j$, $j = -(M_x - 1)/2, \ldots, (M_x - 1)/2$.

One way that we can use the FFT to solve problem (10.15.1)–(10.15.2) is to write the functions u_{jk} and F_{jk} in terms of finite Fourier series with respect to the j index:

$$u_{jk} = \sum_{\ell=-(M_x-1)/2}^{(M_x-1)/2} \hat{u}_{\ell k} e^{i\ell\pi j \Delta x}, \tag{10.15.8}$$

where

$$\hat{u}_{\ell k} = \frac{1}{M_x}(u_{\cdot k}, \phi_\ell) = \frac{1}{M_x}\sum_{m=0}^{M_x-1} u_{m k} e^{-i\ell\pi m\Delta x} \tag{10.15.9}$$

and

$$F_{j k} = \sum_{\ell=-(M_x-1)/2}^{(M_x-1)/2} \hat{F}_{\ell k} e^{i\ell\pi j\Delta x}, \tag{10.15.10}$$

where

$$\hat{F}_{\ell k} = \frac{1}{M_x}(F_{\cdot k}, \phi_\ell) = \frac{1}{M_x}\sum_{m=0}^{M_x-1} F_{m k} e^{-i\ell\pi m\Delta x}. \tag{10.15.11}$$

We next want to insert the series given in (10.15.8) and (10.15.10) into difference equation (10.15.3). Before doing so, we prove the following result.

Proposition 10.15.1

$$\delta_x^2 u_{j k} = \sum_{\ell=-(M_x-1)/2}^{(M_x-1)/2} -4\sin^2\frac{\ell\pi\Delta x}{2}\hat{u}_{\ell k} e^{i\ell\pi j\Delta x}. \tag{10.15.12}$$

Proof: We note that

$$\begin{aligned}
\delta_x^2 u_{j k} &= \sum_{\ell=-(M_x-1)/2}^{(M_x-1)/2} \hat{u}_{\ell k}\delta_x^2 e^{i\ell\pi j\Delta x} \\
&= \sum_{\ell=-(M_x-1)/2}^{(M_x-1)/2} \hat{u}_{\ell k} e^{i\ell\pi j\Delta x}\left(e^{i\ell\pi\Delta x} - 2 - e^{-i\ell\pi\Delta x}\right) \\
&= \sum_{\ell=-(M_x-1)/2}^{(M_x-1)/2} -4\sin^2\frac{\ell\pi\Delta x}{2}\hat{u}_{\ell k} e^{i\ell\pi j\Delta x}.
\end{aligned}$$

We see that if we insert the series given in (10.15.8) and (10.15.10) into difference equation (10.15.3), we get

$$\left(\frac{1}{\Delta x^2}\delta_x^2 + \frac{1}{\Delta y^2}\delta_y^2\right) u_{j k} = \sum_{\ell=-(M_x-1)/2}^{(M_x-1)/2} e^{i\ell\pi j\Delta x}\left[\frac{-4}{\Delta x^2}\sin^2\frac{\ell\pi\Delta x}{2} + \frac{1}{\Delta y^2}\delta_y^2\right]\hat{u}_{\ell k} \tag{10.15.13}$$

$$= F_{jk}$$
$$= \sum_{\ell=-(M_x-1)/2}^{(M_x-1)/2} e^{i\ell\pi j\Delta x}\hat{F}_{\ell k}. \tag{10.15.14}$$

Using the basic result for finite Fourier series (or multiply both sides by $\overline{\phi_m}$ and sum), we get

$$\left[\frac{-4}{\Delta x^2}\sin^2\frac{\ell\pi\Delta x}{2} + \frac{1}{\Delta y^2}\delta_y^2\right]\hat{u}_{\ell k} = \hat{F}_{\ell k}, \quad k = 1,\ldots,M_y-1,$$
$$\ell = -(M_x-1)/2,\ldots,(M_x-1)/2. \tag{10.15.15}$$

If we also transform the boundary conditions $u_{j0} = u_{jM_y} = 0$, we obtain boundary conditions for $\hat{u}$, $\hat{u}_{\ell 0} = \hat{u}_{\ell M_y} = 0$, $\ell = -(M_x-1)/2,\ldots,(M_x-1)/2$. The transformed problem (equation (10.15.15) along with the transformed boundary conditions) can then be rewritten as follows.
For $\ell = -(M_x-1)/2,\ldots,(M_x-1)/2$,

$$\left[\frac{-4}{\Delta x^2}\sin^2\frac{\ell\pi\Delta x}{2} + \frac{1}{\Delta y^2}\delta_y^2\right]\hat{u}_{\ell k} = \hat{F}_{\ell k}, \ k = 1,\ldots,M_y-1 \tag{10.15.16}$$
$$\hat{u}_{\ell 0} = \hat{u}_{\ell M_y} = 0. \tag{10.15.17}$$

We see that for each ℓ, equations (10.15.16)–(10.15.17) describe a tridiagonal system of equations. Thus, the transformed problem, (10.15.16)–(10.15.17), can be solved as $M_x - 1$ tridiagonal systems of equations. The approach for this solution can be described by the following algorithm.

One Dimensional FFT-10.15.3

For $\ell = -(M_x-1)/2,\ldots,(M_x-1)/2$
For $k = 1,\ldots,M_y-1$

$$r_k = \hat{F}_{\ell k} = \frac{1}{M_x}(F_{\cdot k}, \phi_\ell) \tag{10.15.18}$$
$$a_k = \frac{1}{\Delta y^2} \ (a_1 \text{ is not necessary})$$
$$b_k = \frac{-4}{\Delta x^2}\sin^2\frac{\ell\pi\Delta x}{2} - \frac{2}{\Delta y^2}$$
$$c_k = \frac{1}{\Delta y^2} \ (c_{M_y-1} \text{ is not necessary})$$

Next k
Call Trid (Save the solution in $\hat{u}_{\ell k}$, $\ell = -(M_x-1)/2,\ldots,(M_x-1)/2$, $k = 1,\ldots,M_y-1$)
Next ℓ

For $k = 1, \ldots, M_y - 1$
For $j = 1, \ldots, M_x - 1$

$$u_{jk} = \sum_{\ell=-(M_x-1)/2}^{(M_x-1)/2} \hat{u}_{\ell k} e^{i\ell\pi j\Delta x} \tag{10.15.19}$$

Next j
Next k

We see that the work necessary to solve problem (10.15.3)–(10.15.5) comes to the arithmetic involved in filling the arrays **r**, **a**, **b**, **c**, the Trid call, and the transformation back to u space (the loop about equation (10.15.19)).

For anyone who has used transformations to solve continuous problems, it becomes immediately clear that we did not have to quit transforming when we obtained equation (10.15.15). If we set

$$\hat{u}_{\ell k} = \sum_{m=-(M_y-1)/2}^{(M_y-1)/2} \hat{\hat{u}}_{\ell m} e^{im\pi k\Delta y}, \tag{10.15.20}$$

where

$$\hat{\hat{u}}_{\ell m} = \frac{1}{M_y}(\hat{u}_{\ell\cdot}, \phi_m) = \frac{1}{M_y} \sum_{p=0}^{M_y-1} \hat{u}_{\ell p} e^{-im\pi p\Delta y} \tag{10.15.21}$$

and

$$\hat{F}_{\ell k} = \sum_{m=-(M_y-1)/2}^{(M_y-1)/2} \hat{\hat{F}}_{\ell m} e^{im\pi k\Delta y}, \tag{10.15.22}$$

where

$$\hat{\hat{F}}_{\ell m} = \frac{1}{M_y}(\hat{F}_{\ell\cdot}, \phi_m) = \frac{1}{M_y} \sum_{p=0}^{M_y-1} \hat{F}_{\ell p} e^{-im\pi p\Delta x}, \tag{10.15.23}$$

we can transform equation (10.15.15) into

$$\left[-\frac{4}{\Delta x^2}\sin^2\frac{\ell\pi\Delta x}{2} - \frac{4}{\Delta y^2}\sin^2\frac{m\pi\Delta y}{2}\right]\hat{\hat{u}}_{\ell m} = \hat{\hat{F}}_{\ell m},$$
$$\ell = 1,\ldots,M_x-1,\ m = 1,\ldots,M_y-1. \quad (10.15.24)$$

Solving for $\hat{\hat{u}}_{\ell m}$, we get

$$\hat{\hat{u}}_{\ell m} = -\frac{1}{\frac{4}{\Delta x^2}\sin^2\frac{\ell\pi\Delta x}{2} + \frac{4}{\Delta y^2}\sin^2\frac{m\pi\Delta y}{2}}\hat{\hat{F}}_{\ell m},$$
$$\ell = 1,\ldots,M_x-1,\ m = 1,\ldots,M_y-1. \quad (10.15.25)$$

The algorithm using this double transformation is as follows: Use formulas (10.15.11) and (10.15.23) to find $\hat{\hat{F}}_{\ell m}$, formula (10.15.25) to find $\hat{\hat{u}}_{\ell m}$ and formulas (10.15.20) and (10.15.8) to find u_{jk}. This is a very easy algorithm to use.

Remark 1: We see that we can use a single transformation and reduce the problem to one of solving $M_x - 1$ tridiagonal systems of equations, or we can use a double transformation and not have to solve any system of equations. It seems clear that the latter approach should be the best approach. Since both approaches require the transforms from F to $\hat{F}$ and $\hat{u}$ to u, we must compare costs of performing the two additional transformations (one from $\hat{F}$ to $\hat{\hat{F}}$ and one from $\hat{\hat{u}}$ to $\hat{u}$) required by the double transformation approach to the work of solving $M_x - 1$ tridiagonal systems of equations. These two transformations require applying both formulas (10.15.23) and (10.15.20) $(M_x - 1)(M_y - 1)$ times. This comparison shows that the approaches require a comparable amount of computer work.

Remark 2: The real way to apply the two techniques described above is to use the fast Fourier transform (FFT) software available to perform the transforms performed in both of these techniques. For the single transform algorithm, an FFT package can be used to perform the transforms contained in equation (10.15.18), and an inverse fast Fourier transform (IFFT) package can be used to perform the transforms contained in equation (10.15.19). Likewise, for the two transform technique, either multiple applications of an FFT package or one application of a two dimensional FFT package can be used to find $\hat{\hat{F}}$, and either multiple applications of an IFFT package or one application of a two dimensional IFFT package can be used to find u.

These packages are fast (hence the name) and will beat coding equations (10.15.18) and (10.15.19), or the double sums necessary to find $\hat{\hat{F}}$ and u, by a large amount. We suggest that for your first implementation of these techniques, you code the finite Fourier series formulas. This approach is very easy to code and very transparent to follow. Then, we suggest that you follow by replacing parts of your code with FFT packages.

Remark 3: When you first look at the techniques introduced above, you might think that we do not really need finite difference techniques. We should note that these methods do not work for a very large class of problems. It is possible to relax the requirement of zero Dirichlet boundary conditions. (We worked with that assumption just to make our results easier.) However, it should be clear that there is a big step required to extend these techniques to nonconstant coefficient or nonlinear problems.

Remark 4: We note that the procedure using finite Fourier transforms given in this section is very similar to the use of transforms for analytic problems (Laplace, Fourier, etc.). The basic use of a transform for analytic problems is to eliminate derivatives in a problem (change an ordinary differential equation to an algebraic equation, change a partial differential equation in two variables into an ordinary differential equation, change a partial differential equation in m variables into a partial differential equation in $m-1$ variables). More than one transform can be used, if desired. Each additional transform further reduces the complexity of the problem. The problem is then solved in transform space—where it is easier—and then transformed back—if that is possible. This is exactly what we have done in this section. We have taken the broadly banded matrix problem (10.15.3)–(10.15.5) and transformed the problem into $M_x - 1$ tridiagonal systems (10.15.15) (plus boundary conditions). After we solve the $M_x - 1$ tridiagonal systems (in transform space), we transform the solution back to reality using (10.15.19). We also show that additional transforms can be used to further reduce the complexity of the problem.

HW 10.15.1 (1) Use the fact that $e^{i\ell\pi j\Delta x} = \cos \ell\pi j\Delta x + i \sin \ell\pi j\Delta x$ to show that formulas (10.15.8) and (10.15.10) can be expressed also as a series in sines and cosines.
(2) Find the transformed equations analogous to equation (10.15.15) associated with sine and cosine transforms.

HW 10.15.2 Extend problems (10.15.1)–(10.15.2) and (10.15.3)–(10.15.5) to the interval $[-1, 1]$. Show that the exponential transform series for the extended problems reduces to a problem involving a sine series (as we did in Example 3.2.1).

HW 10.15.3 (1) Use the One Dimensional FFT-10.15.3 algorithm to solve the following elliptic boundary value problem.

$$-\nabla^2 v = e^{x+y}, \quad (x,y) \in R = (0,1) \times (0,1)$$
$$v = 0 \quad \text{on } \partial R.$$

(2) Solve the problem given in part (1) using the double transform FFT scheme. Compare the amount of computer time necessary for these two solution techniques.

HW 10.15.4 Solve the problem given in HW10.15.3 by both the single and double transform schemes using FFT and IFFT packages (and their two dimensional analogues) to perform the appropriate transformations. Compare the times for this approach with the results found in HW10.15.3.

10.16 Computational Interlude VIII

By now, we hope, you have had time to attempt to solve HW0.0.4 using the approach described in Section 10.11. Your results should be that for $M_\infty = 0.7$, the approach worked well. The plot of $\bar{c}_p$ for $k = 0$ should look like the plot given in Figure 10.16.1.

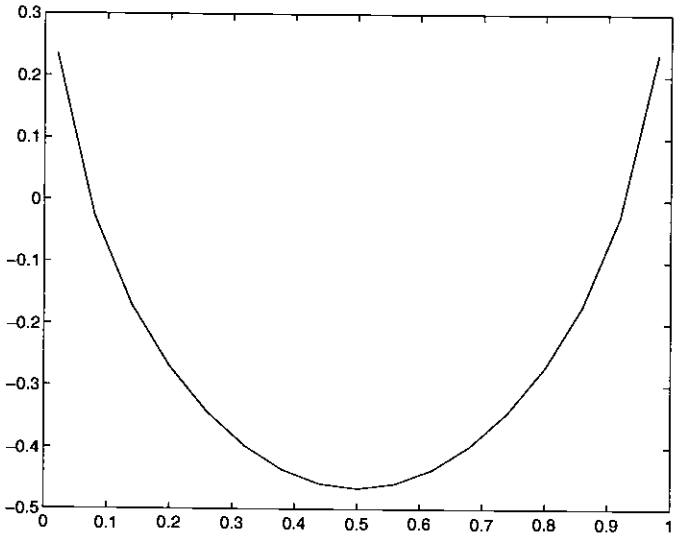

FIGURE 10.16.1. Plot of $\bar{c}_p$ associated with $M_\infty = 0.70$.

However, when you set $M_\infty = 0.78$, your scheme should have gone wild. This problem was included in part to show you how sensitive a numerical scheme can be and that there are times when you must adjust your scheme radically depending on the problem being solved. If you tried too hard to debug your code, you probably noticed that for the case of $M_\infty = 0.78$, just before we ran into trouble, some of the $c^n_{j\,k}$'s turned negative. This should at least be a hint that there is something wrong. For the case $M_\infty = 0.7$, the $c^n_{j\,k}$'s were always positive. Though this is a nonlinear problem, the positive $c^n_{j\,k}$'s gives us evidence that the equation is probably elliptic, and it is. When the $c^n_{j\,k}$'s become negative, the equation becomes locally hyperbolic. We do not know for sure, but we have indications that the differencing used to obtain equation (10.11.12) is not good for hyperbolic equations. The local

domains of dependence must be preserved. The locally hyperbolic region must be influenced only by upstream points.

Hence, for the case $M_\infty = 0.78$, we must come up with an equation and a solution technique. One way to decide (give us a hint?) what should be done is to consider model equations. When we differenced the partial differential equation given in HW0.0.4, we proceeded in the same way we did for the equation $\phi_{xx} + \phi_{yy} = 0$. When $M_\infty = 0.78$, there are points at which the equation looks more like $-\phi_{xx} + \phi_{yy} = 0$. Locally, we must difference our equation more as we would difference $-\phi_{xx} + \phi_{yy} = 0$.

We proceed based on a scheme described in [10] (which is based on results given in [50]). At elliptic points, we continue to use difference scheme 10.11.12. In a locally hyperbolic region, we use an implicit scheme. We use an implicit scheme to ensure stability of the scheme. Using an explicit scheme would require unreasonably small values of Δx in some regions. In the hyperbolic regions, using an upstream differencing with respect to the x variable yields the following difference equation

$$\left[1 - M_\infty^2 - (\gamma+1)M_\infty^2 \frac{u_{j\,k} - u_{j-2\,k}}{2\Delta x}\right]\left[\frac{u_{j\,k} - 2u_{j-1\,k} + u_{j-2\,k}}{\Delta x^2}\right] + \frac{u_{j\,k+1} - 2u_{j\,k} + u_{j\,k-1}}{\Delta y^2} = 0. \tag{10.16.1}$$

Let

$$b_{j\,k} = 1 - M_\infty^2 - (\gamma+1)M_\infty^2\left(\frac{u_{j\,k} - u_{j-2\,k}}{2\Delta x}\right),$$

and rewrite equation (10.16.1 as

$$\frac{b_{j\,k}}{\Delta x^2}u_{j-2\,k} + \frac{1}{\Delta y^2}u_{j\,k-1} - 2\frac{b_{j\,k}}{\Delta x^2}u_{j-1\,k} - \left(\frac{2}{\Delta y^2} - \frac{b_{j\,k}}{\Delta x^2}\right)u_{j\,k} + \frac{1}{\Delta y^2}u_{j\,k+1} = 0. \tag{10.16.2}$$

We want to use difference equation (10.16.2) at points where

$$1 - M_\infty^2 - (\gamma+1)M_\infty^2\phi_x$$

is negative. But the situation is not that nice. We must have some criterion for deciding whether $1 - M_\infty^2 - (\gamma+1)M_\infty^2\phi_x$ is negative. The difficulty comes with whether we use $b_{j\,k}$ or $c_{j\,k}$ to decide whether $1 - M_\infty^2 - (\gamma+1)M_\infty^2\phi_x$ is negative. We will have points where both $b_{j\,k}$ and $c_{j\,k}$ are positive, points where they are both negative, and points where one is positive and one is negative. We shall proceed as follows (see [10]).

- If $b_{j\,k} \geq 0$ and $c_{j\,k} \geq 0$, the point is a **subsonic** point and we use difference equation

$$\frac{1}{\Delta y^2}u_{j\,k-1}+\frac{c_{j\,k}}{\Delta x^2}u_{j-1\,k}-\left(\frac{2}{\Delta y^2}+\frac{2c_{j\,k}}{\Delta x^2}\right)u_{j\,k}+\frac{1}{\Delta y^2}u_{j\,k+1}$$
$$+\frac{c_{j\,k}}{\Delta x^2}u_{j+1\,k}=0. \tag{10.16.3}$$

- If $b_{j\,k}\geq 0$ and $c_{j\,k}<0$, the point is a **sonic** point and we use the difference equation

$$u_{j\,k-1}-2u_{j\,k}+u_{j\,k+1}=0. \tag{10.16.4}$$

- If $b_{j\,k}<0$ and $c_{j\,k}<0$, the point is a **supersonic** point and we use the difference equation

$$\frac{b_{j\,k}}{\Delta x^2}u_{j-2\,k}+\frac{1}{\Delta y^2}u_{j\,k-1}-2\frac{b_{j\,k}}{\Delta x^2}u_{j-1\,k}-\left(\frac{2}{\Delta y^2}-\frac{b_{j\,k}}{\Delta x^2}\right)u_{j\,k}$$
$$+\frac{1}{\Delta y^2}u_{j\,k+1}=0. \tag{10.16.5}$$

- If $b_{j\,k}<0$ and $c_{j\,k}\geq 0$, the point is a **shock** point and we use equation

$$\frac{b_{j\,k}}{\Delta x^2}u_{j-2\,k}+\frac{1}{\Delta y^2}u_{j\,k-1}-\frac{1}{\Delta x^2}\left(2b_{j\,k}-c_{j\,k}\right)u_{j-1\,k}$$
$$-\left(\frac{2}{\Delta y^2}+\frac{2c_{j\,k}}{\Delta x^2}-\frac{b_{j\,k}}{\Delta x^2}\right)u_{j\,k}+\frac{1}{\Delta y^2}u_{j\,k+1}+\frac{c_{j\,k}}{\Delta x^2}u_{j+1\,k}=0. \tag{10.16.6}$$

We note that the subsonic points are treated as we treated all of the points for $M_\infty=0.7$, and the supersonic points are treated as we discussed earlier for points where the equation is hyperbolic. The sonic points are treated very logically. If $b_{j\,k}\geq 0$ (in which case difference scheme (10.16.2) would be unstable) and $c_{j\,k}<0$ (in which case difference scheme (10.11.12) would also be unstable), we treat $1-M_\infty^2-(\gamma+1)M_\infty^2\phi_x$ as if it were zero. This is how we obtain equation (10.16.4).

The treatment of the shock points is much more difficult, and these are very important points. Equation (10.16.6) is formed by taking the sum of the subsonic and supersonic treatments of the x derivative plus the usual treatment of the y derivative. Difference equation (10.16.6) will not be consistent with the partial differential equation, but it is shown in [50] that this strange difference approximation guarantees the correct, unique shock jump.

To be able to solve the difference equations at supersonic and shock points implicitly (as the formulation of difference equation (10.16.2) demands), the reasonable technique is to solve using line Gauss-Seidel (or line Jacobi or line SOR), solving on j-lines.

The approach to use to solve HW0.0.4, $M_\infty=0.78$, is the following.

- Solve on the $j = 0$ line using the subsonic equations (10.16.3) using boundary condition (10.11.13) to define $u_{-1\,k}$.
  - Set up the coefficient matrix and the right hand side matrix.
  - Remember to use boundary conditions (10.11.14), (10.11.16), and (10.11.17) to adjust the coefficients in the top and bottom rows of the tridiagonal matrix.
  - Solve the resulting system of equations using the subroutine TRID.
- Solve on the $j = 1$ line treating all of the points as subsonic points.
  - Except for the fact that no adjustments are necessary for the $j = 0$ boundary condition, the approach is the same as that used on the $j = 0$ line.
- Solve on lines $j = 2, \ldots, M_x - 1$ by using the full system of equations (10.16.3)–(10.16.6) plus boundary conditions (10.11.13)–(10.11.17).
  - Check each point to decide whether they are subsonic, sonic, supersonic, or shock points. Use this information at each point to fill the tridiagonal matrix and the right hand side.
  - Solve using TRID.
- Solve on the $j = M_x$ line using the subsonic equations and boundary condition (10.11.14).

We are now ready to try to solve HW0.0.4 for $M_\infty = 0.78$. If this approach works (and it should), we see that there are times when we must make radical adjustments to our approach. The technique described above should not be considered a standard or obvious approach. This problem is a very difficult problem.

11
Irregular Regions and Grids

11.1 Introduction

Throughout ten chapters of this text we have studied many algorithms, and we hope that we solved many problems. Most of these problems were defined on the interval $[0,1]$, the square $[0,1] \times [0,1]$ or a cube. (There were probably a couple of times when we got adventuresome and used $[-2,1]$ instead of $[0,1]$.) We all know that the world and the objects in the world are not intervals, squares and cubes. It is convenient to treat these easier problems first while we are learning about numerical schemes. It is now time to face some reality and discuss more complicated situations. We will discuss both situations where the geometry is more complex than a square or a cube and the situation where it is best to use a nonuniform grid. We will not discuss these topics in depth. We will try to introduce a variety of the methods that are used to confront problems caused by very irregular solutions and irregular geometries. We hope to provide an overview of some of the difficulties and some of the solutions.

11.2 Irregular Geometries

11.2.1 Blocking Out

Probably the most obvious approaches to solving problems with irregular geometries is to place a grid on the region and use whatever approximations are convenient and necessary to derive a discrete problem that ap-

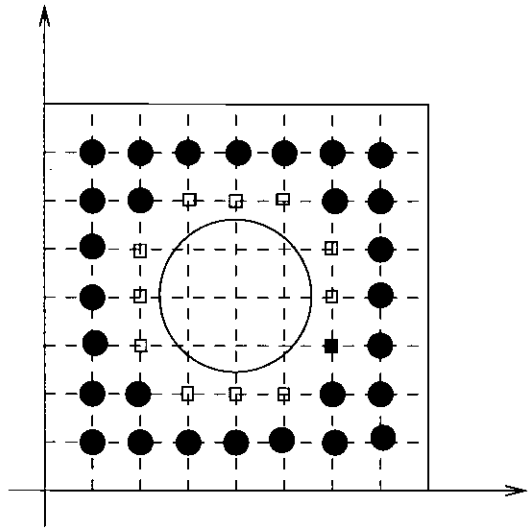

FIGURE 11.1.1. The region $R = \{(x,y) \in (0,1)\times(0,1) : \left(x-\frac{1}{2}\right)^2+\left(y-\frac{1}{2}\right)^2 > \frac{1}{25}\}$ with a grid with $\Delta x = \Delta y = \frac{1}{8}$ over the entire square $[0,1] \times [0,1]$.

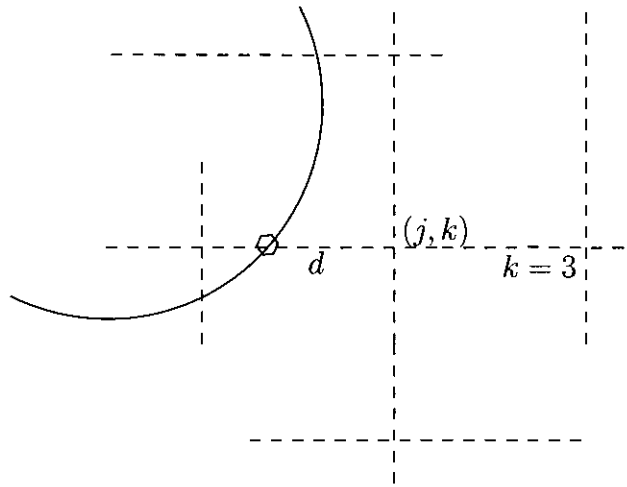

FIGURE 11.1.2. An enlarged picture near the grid point labeled with a filled square in Figure 11.1.1.

proximates the analytic problem to whatever degree of accuracy is desired and/or required. For example, consider the problem

$$v_t = v_{xx} + v_{yy}, \quad t > 0, (x,y) \in R =$$
$$\{(x,y) \in (0,1) \times (0,1) : \left(x - \frac{1}{2}\right)^2 + \left(y - \frac{1}{2}\right)^2 > \frac{1}{25}\}, \tag{11.2.1}$$
$$v(x,y,0) = 0, \quad (x,y) \in R \tag{11.2.2}$$
$$v(x,y,t) = 0 \quad \text{when } x = 0,\ x = 1,\ y = 0 \text{ or } y = 1, \quad t \geq 0 \tag{11.2.3}$$
$$v(x,y,t) = \sin 4\pi\left(x - \frac{1}{2}\right) \quad \text{when } \left(x - \frac{1}{2}\right)^2 + \left(y - \frac{1}{2}\right)^2 = \frac{1}{25}. \tag{11.2.4}$$

We should understand that this is a model problem for an exterior problem. For most of the discussion that follows, it would not be difficult to replace the circle in the interior by other shapes. We should recall that we solved a similar elliptic problem in Section 10.11.1. The solution procedure given here and the discussion of the blocking out procedure will very similar to what we did in Section 10.11.1. We place a grid with $\Delta x = \Delta y = \frac{1}{8}$ over the region as is shown in Figure 11.1.1. The grid is a bit coarse for solving our problem but ideal for illustrating our method. At the points labeled • in Figure 11.1.1, we might approximate the partial differential equation as we have done before by

$$u_{j\,k}^{n+1} = u_{j\,k}^{n} + r(\delta_x^2 + \delta_y^2)u_{j\,k}^{n} \tag{11.2.5}$$

where $r = \Delta t/\Delta x^2 = \Delta t/\Delta y^2$. We note that these are the regular points that are in the interior of the region and reach to four other points in the region or on the exterior boundary. At the points labeled by a small square, we reach to three points outside of the circle and one point on the circle. To illustrate how we derive the difference equations at these points adjacent to the inner boundary, we consider the point labeled by the filled square. In Figure 11.1.2, we given an enlarged view of the grid near the point (j,k) labeled by the filled square in Figure 11.1.1. At such a point, we might approximate the partial differential equation by

$$u_{j\,k}^{n+1} = u_{j\,k}^{n} + r\frac{2}{\rho(1+\rho)}\left(u_{j-\rho\,k}^{n} - (1+\rho)u_{j\,k}^{n} + \rho u_{j+1\,k}^{n}\right) + r\delta_y^2 u_{j\,k}^{n} \tag{11.2.6}$$

where $\rho = d/\Delta x$ and $u_{j-\rho\,k}^{n}$ denotes the value of u where the grid line $k = 3$ intersects the circle nearest the filled square point (labeled with a hexagon in Figure 11.1.2). The partial differential equation at the other unlabeled points neighboring the circle can be approximated in a similar fashion. We should understand that since

$$
\begin{aligned}
&\frac{2}{\Delta x^2\rho(1+\rho)}\left(u_{j-\rho\,k}-(1+\rho)u_{j\,k}+\rho u_{j\,k}\right)\\
&=\frac{2}{\Delta x^2\rho(1+\rho)}\Big\{u_{j\,k}+\left(u_x\right)_{j\,k}(-d)+\left(u_{xx}\right)_{j\,k}\frac{(-d)^2}{2}+\left(u_{xxx}\right)_{j\,k}\frac{(-d)^3}{6}\\
&\quad+\cdots-(1+\rho)u_{j\,k}+\rho\Big[u_{j\,k}+\left(u_x\right)_{j\,k}\Delta x+\left(u_{xx}\right)_{j\,k}\frac{\Delta x^2}{2}\\
&\qquad\qquad+\left(u_{xxx}\right)_{j\,k}\frac{\Delta x^3}{6}+\cdots\Big]\Big\}\\
&=\frac{2}{\Delta x^2\rho(1+\rho)}\Big[\frac{\Delta x^2}{2}\rho(1+\rho)\left(u_{xx}\right)_{j\,k}\\
&\qquad\qquad+\frac{\Delta x^3}{6}\rho(1+\rho)(1-\rho)\left(u_{xxx}\right)_{j\,k}+\cdots\Big]\\
&=\left(u_{xx}\right)_{j\,k}+\frac{\Delta x}{3}(1-\rho)\left(u_{xxx}\right)_{j\,k}+\cdots,
\end{aligned}
$$

the approximation of v_{xx} used to derive difference equation (11.2.6) is only first order accurate. More importantly, the scheme is only first order accurate near the circle where the action in the problem is happening. Hence, the resulting difference scheme will be only first order accurate. If we need and/or want a second order scheme, at the point labeled by the filled square we could use

$$
\begin{aligned}
u_{j\,k}^{n+1}=u_{j\,k}^{n}+r\frac{1}{\rho(\rho-1)(\rho+4)}\Big[&6u_{j-\rho\,k}^{n}+(\rho+2)(\rho^2-3)u_{j\,k}^{n}\\
&+2\rho(\rho-2)(\rho+2)u_{j+1\,k}^{n}+\rho(\rho-1)(\rho+1)u_{j+2\,k}^{n}\Big]+r\delta_y^2u_{j\,k}^{n}\qquad(11.2.7)
\end{aligned}
$$

with similar difference equations used at the other points neighboring the circle. We emphasize that for both the first order equation (11.2.6) and the second order equation (11.2.7), the term $u_{j-\rho\,k}^{n}$ is a boundary condition.

Assuming that we have written a difference equation at the other eleven points neighboring the circle, we could use difference equation (11.2.5), (11.2.6), or (11.2.7) and their eleven analogues, and the boundary conditions and initial conditions to obtain an approximation to the initial–boundary–value problem (11.2.1)–(11.2.4). How we implement the above scheme and how efficiently we implement the above scheme depends on the use the scheme will get. If we were to consider exactly the scheme described above with only approximately 40 interior points, we could just write out the equations using brute force. This is never very satisfying and usually not the desired method of implementation (because we hardly ever have as few as 40 points). The implementation will always be easier if we take the time and computer storage to fill a stencil array associated with each grid point analogous to the stencils we used for elliptic equations. See Section 10.5.5. We must be aware that if we use the second order accurate scheme

at the points neighboring the circle, we must use a larger stencil. The easiest approach, but the most expensive approach when we consider computer storage, is to use a stencil that is $M_x \times M_y \times 9$, giving room for the $j+2$, $j-2$, $k+2$ and $k-2$ stencil values when they are necessary.

If the computer program is going to be used often or a similar program is to be used for other problems, it usually pays to work a bit harder to develop a cleaner scheme. In Section 10.11.1 we introduced the blocking out procedure using the method to solve several elliptic boundary–value problems. The blocking out procedure allows for the inclusion of the points inside the circle, setting the initial values of $u^0_{j\,k}$ inside of the circle equal to one, and using a stencil at these points so that $u^n_{j\,k} = 1$ for all n. The stencils of the points neighboring the circle are adjusted so that the stencil values that reach inside of the circle times the function value inside the circle (one) give the difference equation the correct value. For example, at a point (j,k) that generated difference equation (11.2.6), we would define

$$S(j,k,0) = 1 - \frac{2r}{\rho} - 2r, \quad S(j,k,1) = \frac{2r}{1+\rho}, \quad S(j,k,3) = r,$$

$$S(j,k,4) = r, \quad S(j,k,2) = \frac{2r}{\rho(1+\rho)} u^n_{j-\rho\,k}$$

and use the difference scheme

$$\begin{aligned} u^{n+1}_{j\,k} = {} & S(j,k,0)u^n_{j\,k} + S(j,k,1)u^n_{j+1\,k} + S(j,k,2)u^n_{j-1\,k} \\ & + S(j,k,3)u^n_{j\,k+1} + S(j,k,4)u^n_{j\,k-1}. \end{aligned} \tag{11.2.8}$$

With this scheme, the $S(j,k,2)u^n_{j-1\,k}$ term reaches into the circle and gets $u^n_{j-1\,k} = 1$. Since $S(j,k,2)$ includes the $u^n_{j-\rho\,k}$ term, we obtain the correct difference equation. Of course, all other points neighboring the circle are treated in a similar fashion; for the interior points that do not neighbor the circle we have $S(j,k,0) = 1 - 4r$, $S(j,k,m) = r$ for $m = 1,2,3,4$, and for the points inside of the circle we have $S(j,k,0) = 1$ and $S(j,k,m) = 0$ for $m = 1,2,3,4$. See Section 10.11.1 to see how the blocking out scheme was implemented for a similar elliptic partial differential equation.

Remark: We should be aware that if we use the computer to decide whether a given grid point is inside, on, or outside the circle, we must do so with a tolerance in the computation. If d is very small, then ρ is very small and we will have trouble with the computations. An approach that will work is to consider that a grid point is on the curve if (x_j, y_k) satisfies

$$\frac{1}{25} - tol < \left(x_j - \frac{1}{2}\right)^2 + \left(y_k - \frac{1}{2}\right)^2 < \frac{1}{25} + tol$$

and inside or outside the curve if the point is on the appropriate side of the above annulus. The tolerance tol must be chosen small for accuracy but large enough so that the arithmetic will work. Tolerances around 1.0×10^{-4} are usually sufficient.

We have seen how blocking out can be used for an elliptic boundary–value problem and parabolic initial–boundary–value problem. The scheme developed for the parabolic problem was an explicit scheme. The approach can be used equally well for implicit schemes. Blocking out can also be used for hyperbolic problems. The procedure is more of a convenience than a method. We have a grid over an irregular region. We have difference equations defined at all of the points that we care about. We use the blocking out procedure to give us a more palatable data structure, taking care that our original equations are not affected by filling in the extra points.

11.2.2 Map the Region

Suppose we return to initial–boundary–value problem (11.2.1)–(11.2.4) and suppose that we given functions $\xi = \xi(x,y)$, $\eta = \eta(x,y)$ such that the curves ξ =const, η =const are plotted in Figure 11.2.1. If we assume that the Jacobian

$$J = \frac{\partial(\xi,\eta)}{\partial(x,y)} = \det \begin{pmatrix} \xi_x & \xi_y \\ \eta_x & \eta_y \end{pmatrix} \neq 0 \text{ on } R,$$

then we know that we can solve for x and y as a function of ξ and η and write $x = x(\xi,\eta)$, $y = y(\xi,\eta)$. We should understand that it is very convenient to be able to solve for x and y. However, it is really necessary that we are able to do so. Also, do not get confused by the fact that we define both the functions and coordinates by ξ, η, and x, y. It is nice (though sometimes confusing) notation to define the new coordinate ξ in terms of x and y and denote the function that does so by $\xi(x,y)$, i.e., $\xi = \xi(x,y)$, $\eta = \eta(x,y)$. We note that the curves plotted in Figure 11.2.1 are analogous to plotting $r = \sqrt{x^2+y^2}$ =const, $\theta = \tan^{-1}(y/x)$ =const if we were working with polar coordinates, except that the ξ and η functions are better suited to the region R. The functions ξ and η give a coordinate system on the region R analogous to polar coordinates when the boundaries are circles. The main differences between the polar coordinate system and the ξ–η "coordinate system" is that (1) the ξ–η system is nicer for problem (11.2.1)–(11.2.4) in that both boundaries of the region R are coordinate lines in the ξ–η coordinate system and (2) the ξ–η coordinate system does not have all of the nice geometry that there is in a polar coordinate system. It was this geometry in polar coordinates that we used along with the control volume approach to derive a set of difference equations for problems expressed in polar coordinates.

It might not be clear how we can use the functions ξ and η (or the coordinate system defined by ξ and η) to help solve initial–boundary–value problem (11.2.1)–(11.2.4). When we used polar coordinates in problems where polar coordinates were useful, we transformed our problem into polar coordinates. When we consider problems in polar coordinates, we usually picture them in such a way that we take advantage of r and θ geometrically, as in Figure 4.5.1, Part 1. It is also possible to view polar coordinates in

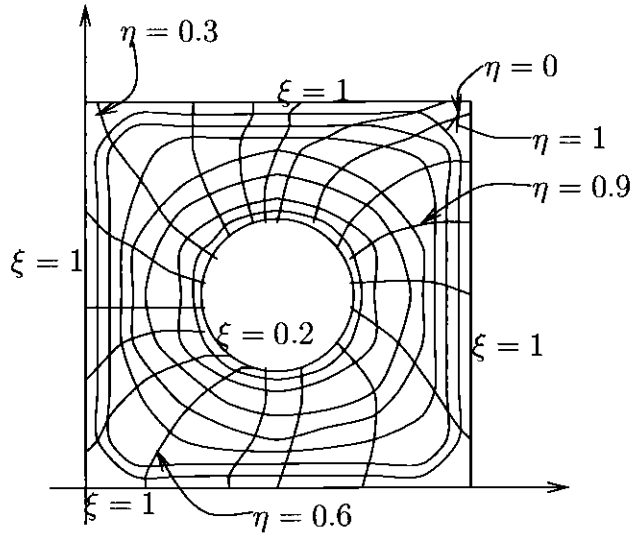

FIGURE 11.2.1. Plots of the curves $\xi(x,y)$ =const, $\eta(x,y)$ =const in the x–y plane.

a rectangular region as we do in Figure 4.5.2, Part 1. In the latter case, polar coordinates make everything more difficult because of the unique nature of the origin in polar coordinates. Most often, we do not have this difficulty and use the idea that we use the functions $\xi(x,y)$, $\eta(x,y)$ to map the region R in the x-y plane, onto a region R' in the ξ-η plane. In Figure 11.2.1 we see that the inner and outer boundaries of the region R correspond to $\xi = 0.2$ and $\xi = 1.0$, respectively. In addition, we see that because of the periodic nature of η (remember that η is similar to the θ variable in polar coordinates), we see that the region is also bounded by $\eta = 0.0$ and $\eta = 1.0$. Hence, we see that when we view our region in the ξ-η plane, the region looks like a rectangle. If we approximate the solution to initial–boundary–value problem (11.2.1)–(11.2.4) in the ξ-η plane, it is easy to place a grid over the region R'. See Figure 11.2.2. If the functions $\xi = \xi(x,y)$, $\eta = \eta(x,y)$ are chosen nicely, the grid lines ξ =const correspond to a series of closed curves in the x-y plane, reminiscent of the polar coordinate grid lines r =const, except for the fact that the curves in this case are distorted so that the $\xi = 1.0$ grid line corresponds to the boundary of the square $[0,1] \times [0,1]$ in the x-y plane and $\xi = 0.2$ corresponds to the circle $\left(x - \frac{1}{2}\right)^2 + \left(y - \frac{1}{2}\right)^2 = \frac{1}{25}$ in the x-y plane. Also, the grid lines η =const correspond to a series of curves from the inner circle to the outer boundary analogous to the θ =const curves in polar coordinates.

Before we can solve initial–boundary–value problem (11.2.1)–(11.2.4) in the ξ-η plane, we must have a partial differential equation defined on R' and boundary conditions defined on $\partial R'$. We transform our equations to the ξ-η plane in much the same way that we transform a problem given

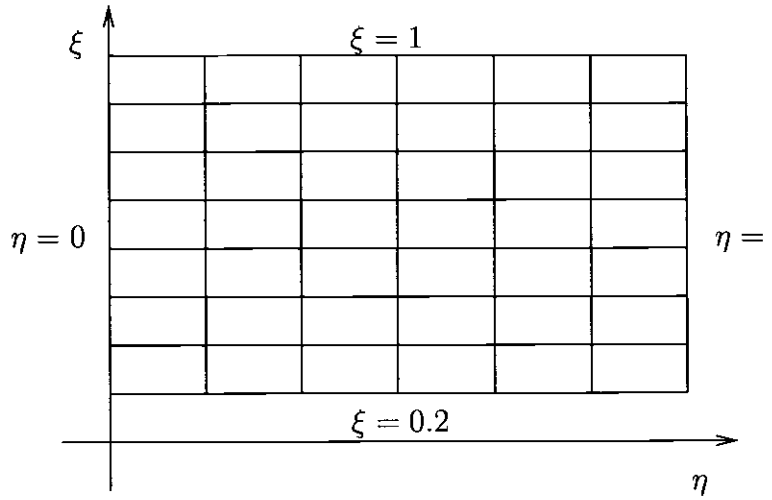

FIGURE 11.2.2. The rectangle R' in the ξ–η plane.

in Cartesian coordinates to a problem given in terms of polar coordinates or any other coordinate system. We define $V = V(\xi, \eta, t)$ by $V(\xi, \eta, t) = v(x(\xi, \eta), y(\xi, \eta), t)$. Clearly, the function V is the equivalent of the function v, connected by the mapping $\xi = \xi(x, y)$, $\eta = \eta(x, y)$ (and its inverse). By many applications of the chain rule, we have

$$\begin{aligned} v_x &= V_\xi \xi_x + V_\eta \eta_x, \quad v_y = V_\xi \xi_y + V_\eta \eta_y, \quad v_t = V_t, \\ v_{xx} &= V_{\xi\xi} \xi_x^2 + 2V_{\xi\eta} \xi_x \eta_x + V_{\eta\eta} \eta_x^2 + V_\xi \xi_{xx} + V_\eta \eta_{xx} \end{aligned}$$

and

$$v_{yy} = V_{\xi\xi} \xi_y^2 + 2V_{\xi\eta} \xi_y \eta_y + V_{\eta\eta} \eta_y^2 + V_\xi \xi_{yy} + V_\eta \eta_{yy}.$$

Then partial differential equation (11.2.1) can be written as

$$\begin{aligned} V_t = {} & (\xi_x^2 + \xi_y^2) V_{\xi\xi} + 2(\xi_x \eta_x + \xi_y \eta_y) V_{\xi\eta} + (\eta_x^2 + \eta_y^2) V_{\eta\eta} \\ & + (\xi_{xx} + \xi_{yy}) V_\xi + (\eta_{xx} + \eta_{yy}) V_\eta. \end{aligned} \tag{11.2.9}$$

However, this is not good enough. The variable V is fine. It is defined on the region R'. Partial differential equation (11.2.9) is a nonconstant coefficient partial differential equation, but the coefficients are defined on the region R. It would be difficult to handle a partial differential equation where the variables and the coefficients are defined on different sets. Also, if we could handle the coefficients defined on R, we could solve the partial differential equation on R, and none of this work would be necessary.

We see that if we take the partial derivative of the equation $\xi = \xi(x, y)$, first with respect to ξ and then with respect to η, we get

$$1 = \xi_x x_\xi + \xi_y y_\xi \tag{11.2.10}$$
$$0 = \xi_x x_\eta + \xi_y y_\eta. \tag{11.2.11}$$

If we then do the same with the equation $\eta = \eta(x, y)$, we get

$$0 = \eta_x x_\xi + \eta_y y_\xi \tag{11.2.12}$$
$$1 = \eta_x x_\eta + \eta_y y_\eta. \tag{11.2.13}$$

If we then solve equations (11.2.10)–(11.2.13) for ξ_x, ξ_y, η_x and η_y, we get

$$\xi_x = \frac{1}{\mathcal{J}} y_\eta \tag{11.2.14}$$
$$\xi_y = -\frac{1}{\mathcal{J}} x_\eta \tag{11.2.15}$$
$$\eta_x = -\frac{1}{\mathcal{J}} y_\xi \tag{11.2.16}$$
$$\eta_y = \frac{1}{\mathcal{J}} x_\xi \tag{11.2.17}$$

where $\mathcal{J}$ is the Jacobian, $\mathcal{J} = x_\xi y_\eta - x_\eta y_\xi = 1/J$.

We must next solve for ξ_{xx}, ξ_{yy}, η_{xx} and η_{yy}. This is more difficult. The approach should be clear. We differentiate equations (11.2.10)–(11.2.13) with respect to ξ and η to obtain eight equations. The equations will involve ξ_{xx}, ξ_{yy}, η_{xx}, η_{yy}, ξ_{xy} and η_{xy}. Two pairs of the equations will be the same (the pairs associated with the cross-derivatives), so we will have six equations and six unknowns. We leave the computation to the reader (use an algebraic manipulator) in HW11.2.1. From HW11.2.1 we find that

$$\xi_{xx} = \frac{1}{\mathcal{J}^3}\left[x_\eta M^{yy} - y_\eta M^{yx}\right] \tag{11.2.18}$$
$$\xi_{yy} = \frac{1}{\mathcal{J}^3}\left[x_\eta M^{xy} - y_\eta M^{xx}\right] \tag{11.2.19}$$
$$\eta_{xx} = \frac{1}{\mathcal{J}^3}\left[-x_\xi M^{yy} + y_\xi M^{yx}\right] \tag{11.2.20}$$
$$\eta_{yy} = \frac{1}{\mathcal{J}^3}\left[-x_\xi M^{xy} + y_\xi M^{xx}\right] \tag{11.2.21}$$

where

$$M^{yy} = y_\eta^2 y_{\xi\xi} - 2y_\xi y_\eta y_{\xi\eta} + y_\xi^2 y_{\eta\eta}$$
$$M^{yx} = y_\eta^2 x_{\xi\xi} - 2y_\xi y_\eta x_{\xi\eta} + y_\xi^2 x_{\eta\eta}$$
$$M^{xy} = x_\eta^2 y_{\xi\xi} - 2x_\xi x_\eta y_{\xi\eta} + x_\xi^2 y_{\eta\eta}$$
$$M^{xx} = x_\eta^2 x_{\xi\xi} - 2x_\xi x_\eta x_{\xi\eta} + x_\xi^2 x_{\eta\eta}.$$

Using (11.2.14)–(11.2.22), we can rewrite partial differential equation (11.2.9) as

$$V_t = \frac{1}{\mathcal{J}^2}(x_\eta^2 + y_\eta^2)V_{\xi\xi} - \frac{2}{\mathcal{J}^2}(x_\xi x_\eta + y_\xi y_\eta)V_{\xi\eta} + \frac{1}{\mathcal{J}^2}(x_\xi^2 + y_\xi^2)V_{\eta\eta}$$
$$+\frac{1}{\mathcal{J}^3}\Big[x_\eta(M^{yy} + M^{xy}) - y_\eta(M^{yx} + M^{xx})\Big]V_\xi$$
$$+\frac{1}{\mathcal{J}^3}\Big[-x_\xi(M^{yy} + M^{xy}) + y_\xi(M^{yx} + M^{xx})\Big]V_\eta. \qquad (11.2.22)$$

We are now almost ready to solve. It is easy to see that initial condition (11.2.2) and boundary conditions (11.2.3)–(11.2.4) become

$$V(\xi, \eta, 0) = 0, \quad (\xi, \eta) \in R' \qquad (11.2.23)$$
$$V(1, \eta, t) = 0, \quad 0 \le \eta \le 1 \qquad (11.2.24)$$
$$V(0.2, \eta, t) = \sin 4\pi\Big(x(0.2, \eta) - \frac{1}{2}\Big) \quad 0 \le \eta \le 1. \qquad (11.2.25)$$

In addition to the natural boundary conditions, we need the following periodicity boundary condition forced on us by the nature of the η variable

$$V(\xi, 0, t) = V(\xi, 1, t). \qquad (11.2.26)$$

If we knew the functions $x = x(\xi, \eta)$, $y = y(\xi, \eta)$, we could use almost any method for approximating the solutions to parabolic partial differential equations to obtain an approximation to initial–boundary–value problem (11.2.22)–(11.2.26). We should realize that the computational problem will be somewhat difficult because partial differential equation (11.2.22) has nonconstant coefficients, but that does not cause big problems. We would then use the definition of V and the functions $\xi(x, y)$, $\eta(x, y)$ to find $v = v(x, y)$.

The real difficulty here is that *we do not know* $x = x(\xi, \eta)$, $y = y(\xi, \eta)$, *or* $\xi = \xi(x, y)$, $\eta = \eta(x, y)$. In addition, it would be very difficult, probably impossible, to determine the analytic form of such functions. The point of this discussion is that if we had such a mapping, we could solve the problem. Hence, it might pay to figure out how to obtain such a mapping—maybe a numerical approximation of the functions.

11.2.3 Grid Generation

We shall continue the discussion from the previous section. Obviously, from the results found in Section 11.2.2, we will try to find the functions ξ and η (or x and y) numerically. This topic could be viewed as "More Mapping the Region" in that we will determine functions ξ and η that will enable us to approximately solve the problem as described in the previous section. However, the topic can also be viewed as obtaining the mapping so that we can use the rectangle in the ξ-η plane to generate a grid in the x-y plane. The topic that we are discussing has become known as "Grid

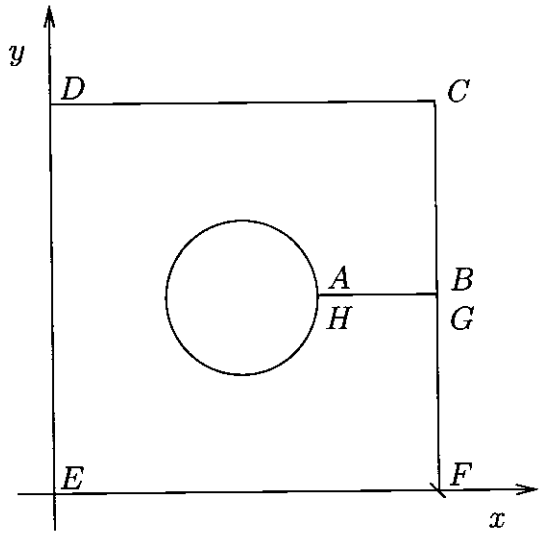

FIGURE 11.2.3. The region R in the x–y plane with the slice and the labeling of the points that we shall use to describe the mapping.

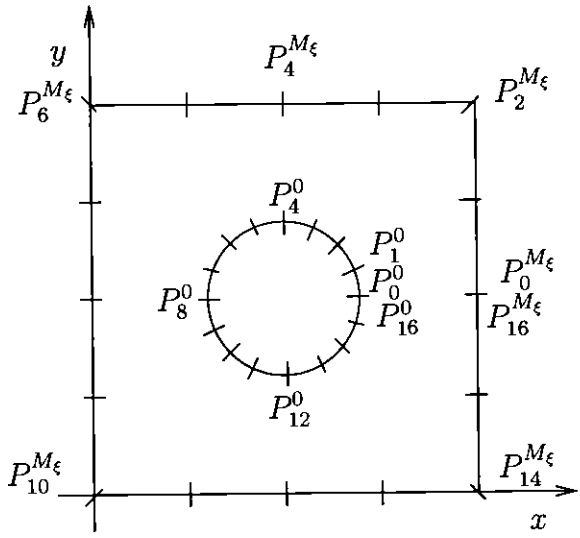

FIGURE 11.2.4. The region R in the x–y plane with points on the boundaries of the region. The points P_k^0 correspond to (x_k^0, y_k^0), $k = 1, \ldots, M$ and the points $P_k^{M_\xi}$ correspond to the $(x_k^{M_\xi}, y_k^{M_\xi})$, $k = 1, \ldots, M$.

Generation," so we refer to this section as such. We shall describe only the most elementary method for finding ξ and η and, hence, generating a grid on R. The subject of grid generation has become huge. Instead of finding a mapping ξ and η that will do the job, the software will allow you to subdivide your region R into a set of smaller regions $R_1, \ldots, R_m$ and find local grids on each of these subregions that join logically at their interfaces. The software will do this in three dimensions. Obviously, for a problem as simple as we have considered, it is not necessary to use such a complex mapping. When you are working in three dimensions and your object in the middle is a space shuttle, it is impossible to accomplish the desired mapping (or generate the desired grid) with just one mapping.

We shall demonstrate the most elementary approach to our problem. For more information on the area of grid generation, see ref. [68] or ref. [32]. We consider what Thompson, et al, ref. [68], refers to as the Laplace system. We begin by slicing our region, for lack of a better place, along the $y = \frac{1}{2}$ line, from $x = \frac{7}{10}$ to $x = 1$. Consider the line segment $y = \frac{1}{2}$, $\frac{7}{10} \leq x \leq 1$ as two line segments drawn very close to each other, one above the other. We label the point where the line segment meets the circle by A and H where the point A refers the point where the top half of the line segment meets the circle and H refers to the point where the bottom half of the line segment meets the circle. We do the same thing where the line meets the $x = 1$ boundary, labeling these points B and G. See Figure 11.2.3. (To help understand what we are doing, we could have literally drawn the line segment $y = \frac{1}{2}$, $\frac{7}{10} \leq x \leq 1$ as two line segments AB, drawn slightly above $y = \frac{1}{2}$, $\frac{7}{10} \leq x \leq 1$, and HG, draw slightly below $y = \frac{1}{2}$ $\frac{7}{10} \leq x \leq 1$.) Our goal is to proceed much as we did in the last section, when we pretended to have the functions ξ and η, and map the region R onto the region R' in the ξ-η plane. We again use a rectangle in the ξ-η plane, this time $R' = [0,1] \times [0,1]$. See Figure 11.2.5.

The approach we use is to place a grid on the circle that consists of $M+1$ equally spaced points, starting with point $A = \left(\frac{7}{10}, \frac{1}{2}\right)$, proceeding in a counterclockwise direction and ending with point $H = \left(\frac{7}{10}, \frac{1}{2}\right)$ (where both points A and H count as grid points). We denote these points by (x_k^0, y_k^0), $k = 0, \ldots, M$.

We then proceed to do the same thing to the outer boundary. We start with point $B = (1, \frac{1}{2}) = (x_0^{M_\epsilon}, y_0^{M_\epsilon})$, proceed in a counterclockwise direction, and end with $G = (1, \frac{1}{2}) = (x_M^{M_\epsilon}, y_M^{M_\epsilon})$. We use equally spaced points, with approximately $M/4$ points on each side. To try to make the implementation as easy as possible, yet nontrivial, in Figure 11.2.4 we have placed only three points in the interior of each side of the outer boundary. In reality, we must choose M sufficiently large to produce a reasonable grid on the outer boundary, and the points would be distributed where we feel they are needed the most.

The next step is to place a grid on the rectangle R' in the ξ-η plane. Analogous to what we discussed in the last section, we want to map R

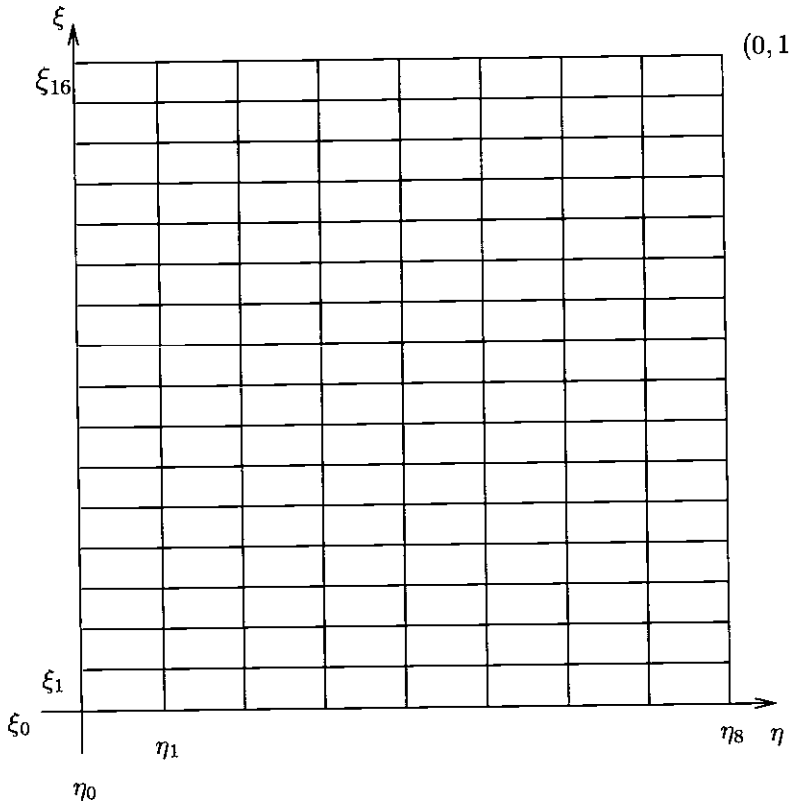

FIGURE 11.2.5. The rectangle $R' = [0,1] \times [0,1]$ in the ξ–η plane.

onto R' in such a way that the outer boundary of the square maps onto $\xi = 1$, the circle maps onto $\xi = 0$, and the two sides of the slice along $y = \frac{1}{2}$ map onto $\eta = 0$ and $\eta = 1$, respectively. We place a uniform grid on R' with $M_\xi + 1$ points in the ξ direction and $M_\eta + 1$ points in the η direction. It will be clear that in this application, M_ξ and M_η should be different (M_η larger than M_ξ) and that M_η should equal M. We will also see that it may be the case that we should not use equally spaced grid points, but we do not concern ourselves with these problems now. We denote these grid points in the usual manner, $(\xi_j, \eta_k) = (j\Delta\xi, k\Delta\eta)$, $j = 0, \ldots, M_\xi$, $k = 0, \ldots, M_\eta$. This grid is illustrated in Figure 11.2.5.

The Laplace system approach is to require that the functions $\xi = \xi(x, y)$, $\eta = \eta(x, y)$ satisfy

$$\nabla^2 \xi = 0 \tag{11.2.27}$$

$$\nabla^2 \eta = 0 \tag{11.2.28}$$

plus some appropriate boundary conditions. We choose the boundary con-

ditions so that the boundary of the region R maps onto the boundary of R' and some sort of boundary condition to help the η variable. There is no special reason to require the functions ξ and η to satisfy (11.2.27) and (11.2.28) rather than some other partial differential equations. Equations (11.2.27) and (11.2.28) are pleasing because they are reminiscent of a conformal map (but we do not get a conformal map); and because the equations have a maximum principle, the Jacobians of the transformations will be nonzero. And lastly, many partial differential equations other than equations (11.2.27), (11.2.28) and techniques other than partial differential equations are used to generate grids.

It should be clear that we cannot solve this problem analytically. Also, solving the problem numerically on the region R is as difficult as solving the original problem on the region R—in which case we would not have needed to discuss mapping the region or generating a grid. Instead, we solve for the functions $x = x(\xi, \eta)$, $y = y(\xi, \eta)$. Using (11.2.18)–(11.2.21), we see that equation (11.2.27) and equation (11.2.28) can be written as

$$\begin{aligned}
0 =& \xi_{xx} + \xi_{yy} \\
=& \frac{1}{\mathcal{J}^3}\Big[(x_\eta M^{yy} - y_\eta M^{yx}) + (x_\eta M^{xy} - y_\eta M^{xx})\Big] \\
=& \frac{1}{\mathcal{J}^3}\Big\{x_\eta\big[(x_\eta^2 + y_\eta^2)y_{\xi\xi} - 2(x_\xi x_\eta + y_\xi y_\eta)y_{\xi\eta} + (x_\xi^2 + y_\xi^2)y_{\eta\eta}\big] \\
& \quad - y_\eta\big[(x_\eta^2 + y_\eta^2)x_{\xi\xi} - 2(x_\xi x_\eta + y_\xi y_\eta)x_{\xi\eta} + (x_\xi^2 + y_\xi^2)x_{\eta\eta}\big]\Big\}
\end{aligned} \tag{11.2.29}$$

and

$$\begin{aligned}
0 =& \eta_{xx} + \eta_{yy} \\
=& \frac{1}{\mathcal{J}^3}\Big[(-x_\xi M^{yy} + y_\xi M^{yx}) + (-x_\xi M^{xy} + y_\xi M^{xx})\Big] \\
=& \frac{1}{\mathcal{J}^3}\Big\{-x_\xi\big[(x_\eta^2 + y_\eta^2)y_{\xi\xi} - 2(x_\xi x_\eta + y_\xi y_\eta)y_{\xi\eta} + (x_\xi^2 + y_\xi^2)y_{\eta\eta}\big] \\
& \quad + y_\xi\big[(x_\eta^2 + y_\eta^2)x_{\xi\xi} - 2(x_\xi x_\eta + y_\xi y_\eta)x_{\xi\eta} + (x_\xi^2 + y_\xi^2)x_{\eta\eta}\big]\Big\}.
\end{aligned} \tag{11.2.30}$$

If we multiply equation (11.2.29) by x_ξ, multiply equation (11.2.30) by x_η and add (and multiply by whatever is needed to clear it all up), we get

$$\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} = 0, \tag{11.2.31}$$

where $\alpha = x_\eta^2 + y_\eta^2$, $\beta = x_\xi x_\eta + y_\xi y_\eta$ and $\gamma = x_\xi^2 + y_\xi^2$. If we instead multiply equation (11.2.29) by y_ξ, multiply equation (11.2.30) by y_η and add, we get

$$\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} = 0. \tag{11.2.32}$$

The reason that it is nicer to solve equations (11.2.31)–(11.2.32) instead of equations (11.2.27)–(11.2.28) is that the domain associated with equations (11.2.31)–(11.2.32) is the rectangle R' in the ξ–η plane pictured in Figure 11.2.5 with a nice uniform grid overlaid on the region. The nasty part of solving equations (11.2.31)–(11.2.32) instead of equations (11.2.27)–(11.2.28) is that equations (11.2.31)–(11.2.32) are very nonlinear. If we compute the discriminant of either equation (11.2.31) or (11.2.32),

$$(-2\beta)^2 - 4\alpha\gamma = -4\left(x_\xi y_\eta - x_\eta y_\xi\right)^2,$$

we see that as long as the Jacobian $\mathcal{J}$ is not zero, partial differential equations (11.2.31) and (11.2.32) are elliptic. At the moment, we are using "the lesser of two evils" approach, but we might add that equations (11.2.31)–(11.2.32) are relatively nice nonlinear partial differential equations.

To determine the boundary conditions associated with equations (11.2.31)–(11.2.32) we begin by placing letters from B through G along $\xi = 1$ as is done in Figure 11.2.5. The outer boundary of the square is mapped to $\xi = 1$ in such a way that the points B through G in the ξ–η plane along $\xi = 1$ map onto points B through G in the x–y plane (see Figure 11.2.3). It is this part of the boundary condition that makes it logical to require that $M_\eta = M$. We will demonstrate this relationship more explicitly below when we give the discrete boundary conditions. The inner boundary of the region R, the circle, is mapped to $\xi = 0$ in such a way that the points A and H in the ξ–η plane map to the points A and H in the x–y plane, and the 15 intermediate points on the circle map onto the 15 points between $(0,0)$ and $(0,1)$ in the ξ-η plane. To allow for the slice in the x–y plane, we map $\eta = 0$ to the top of the slice (the line segment AB) and $\eta = 1$ to the bottom of the slice (line segment GH). Along $\eta = 0$ and $\eta = 1$ we achieve the necessary continuity across the slice by using the periodic boundary conditions $x(\xi,0) = x(\xi,1)$ and $y(\xi,0) = y(\xi,1)$. As we stated earlier, we will make these boundary conditions more explicit when we provide the boundary conditions for the discrete problem.

We are now ready to try to find an approximate solution to equations (11.2.31)–(11.2.32) and the boundary conditions. This boundary–value problem is difficult, but not that different from what we have considered before. Though the partial differential equations are nonlinear, the equations are elliptic, and we will treat them much as we treated linear elliptic partial differential equations. We let x_{jk}, y_{jk}, $j = 0, \ldots, M_\xi$, $k = 0, \ldots, M_\eta$ denote the discrete approximations of $x(j\Delta\xi, k\Delta\eta)$, $y(j\Delta\xi, k\Delta\eta)$, $j = 0, \ldots, M_\xi$, $k = 0, \ldots, M_\eta$, respectively. We replace the derivatives in partial differential equations (11.2.31)–(11.2.32) by differences and obtain the following difference equations.

$$0 = \frac{1}{4\Delta\xi^2\Delta\eta^2}\left[\left(\delta_{\eta 0}x_{j\,k}\right)^2 + \left(\delta_{\eta 0}y_{j\,k}\right)^2\right]\delta_\xi^2 x_{j\,k}$$
$$-\frac{2}{16\Delta\xi^2\Delta\eta^2}\left(\delta_{\xi 0}x_{j\,k}\delta_{\eta 0}x_{j\,k} + \delta_{\xi 0}y_{j\,k}\delta_{\eta 0}y_{j\,k}\right)\delta_{\xi 0}\delta_{\eta 0}x_{j\,k}$$
$$+\frac{1}{4\Delta\xi^2\Delta\eta^2}\left[\left(\delta_{\xi 0}x_{j\,k}\right)^2 + \left(\delta_{\xi 0}y_{j\,k}\right)^2\right]\delta_\eta^2 x_{j\,k} \tag{11.2.33}$$

$$0 = \frac{1}{4\Delta\xi^2\Delta\eta^2}\left[\left(\delta_{\eta 0}x_{j\,k}\right)^2 + \left(\delta_{\eta 0}y_{j\,k}\right)^2\right]\delta_\xi^2 y_{j\,k}$$
$$-\frac{2}{16\Delta\xi^2\Delta\eta^2}\left(\delta_{\xi 0}x_{j\,k}\delta_{\eta 0}x_{j\,k} + \delta_{\xi 0}y_{j\,k}\delta_{\eta 0}y_{j\,k}\right)\delta_{\xi 0}\delta_{\eta 0}y_{j\,k}$$
$$+\frac{1}{4\Delta\xi^2\Delta\eta^2}\left[\left(\delta_{\xi 0}x_{j\,k}\right)^2 + \left(\delta_{\xi 0}y_{j\,k}\right)^2\right]\delta_\eta^2 y_{j\,k}. \tag{11.2.34}$$

Obviously, difference equations (11.2.33)–(11.2.34) are very nonlinear (and this should not surprise us). We note that we have replaced the first order derivatives in α, β and γ by centered differences. We have replaced $x_{\xi\xi}$, $x_{\eta\eta}$, $y_{\xi\xi}$ and $y_{\eta\eta}$ by the usual second order accurate approximations to second derivatives. One of the big differences in this problem from those that we have faced before is the appearance of the cross-derivative terms $x_{\xi\eta}$ and $y_{\xi\eta}$. We have approximated these cross-derivatives in a very elementary fashion by the composition of the two obvious first order difference operators. It should be reasonably clear that the resulting difference is a second order approximation of the derivatives. See HW11.2.2. We notice that the existence of this cross-derivative and the resulting differencing that we have used gives us a stencil different from anything we have seen previously. The stencil is now a nine point stencil. A nine point stencil produces a very different matrix from what we have considered before. See HW11.2.4.

We proceed as we did in HW0.0.4 in Section 10.11.2, to lag the nonlinear coefficients and use a relaxation scheme. From our experience in Section 10.11.2 we know that this procedure can succeed and that it can fail. We let $x_{j\,k}^{\ell}$, $y_{j\,k}^{\ell}$ denote the ℓ-th iterates in the iterative scheme and define

$$\alpha_{j\,k}^{\ell} = \frac{1}{4\Delta\eta^2}\left[\left(\delta_{\eta 0}x_{j\,k}^{\ell}\right)^2 + \left(\delta_{\eta 0}y_{j\,k}^{\ell}\right)^2\right] \tag{11.2.35}$$

$$\beta_{j\,k}^{\ell} = \frac{1}{4\Delta\xi\Delta\eta}\left(\delta_{\xi 0}x_{j\,k}^{\ell}\delta_{\eta 0}x_{j\,k}^{\ell} + \delta_{\xi 0}y_{j\,k}^{\ell}\delta_{\eta 0}y_{j\,k}^{\ell}\right) \tag{11.2.36}$$

$$\gamma_{j\,k}^{\ell} = \frac{1}{4\Delta\xi^2}\left[\left(\delta_{\xi 0}x_{j\,k}^{\ell}\right)^2 + \left(\delta_{\xi 0}y_{j\,k}^{\ell}\right)^2\right]. \tag{11.2.37}$$

We consider the linearizations of difference equations (11.2.33)–(11.2.34),

$$0 = \alpha_{j\,k}^{\ell}\frac{1}{\Delta\xi^2}\delta_\xi^2 x_{j\,k}^{\ell+1} - 2\beta_{j\,k}^{\ell}\frac{1}{4\Delta\xi\Delta\eta}\delta_{\xi 0}\delta_{\eta 0}x_{j\,k}^{\ell+1} + \gamma_{j\,k}^{\ell}\frac{1}{\Delta\eta^2}\delta_\eta^2 x_{j\,k}^{\ell+1} \quad (11.2.38)$$

$$0 = \alpha_{j\,k}^{\ell}\frac{1}{\Delta\xi^2}\delta_\xi^2 y_{j\,k}^{\ell+1} - 2\beta_{j\,k}^{\ell}\frac{1}{4\Delta\xi\Delta\eta}\delta_{\xi 0}\delta_{\eta 0}y_{j\,k}^{\ell+1} + \gamma_{j\,k}^{\ell}\frac{1}{\Delta\eta^2}\delta_\eta^2 y_{j\,k}^{\ell+1} \quad (11.2.39)$$

To obtain boundary conditions to accompany difference equations (11.2.38)–(11.2.39), we recall that we defined 17 grid points along the outer boundary of the square and 17 points along the circle, pictured in Figure 11.2.4, labeling them as (x_k^0, y_k^0) and $(x_k^{M_\xi}, y_k^{M_\xi})$, $k = 0, \ldots, 16$, respectively. We must understand that when we actually use these points, we must write out the coordinate values for the points. For example, $(x_0^{M_\xi}, y_0^{M_\xi})) = (1, \frac{1}{2})$, $(x_1^{M_\xi}, y_1^{M_\xi})) = (1, \frac{3}{4})$, $(x_2^{M_\xi}, y_2^{M_\xi})) = (1, 1)$, $(x_3^{M_\xi}, y_3^{M_\xi})) = (\frac{3}{4}, 1)$, etc., and

$$(x_k^0, y_k^0) = (0.2\cos 2\pi k/16, 0.2\sin 2\pi k/16), \quad k = 0, \ldots, 16.$$

We then set the following boundary conditions for $x_{j\,k}^{\ell+1}$, $y_{j\,k}^{\ell+1}$.

$$x_{M_\xi\,k}^{\ell+1} = x_k^{M_\xi}, \quad k = 0, \ldots, M_\eta \quad (11.2.40)$$

$$y_{M_\xi\,k}^{\ell+1} = y_k^{M_\xi}, \quad k = 0, \ldots, M_\eta \quad (11.2.41)$$

$$x_{0\,k}^{\ell+1} = x_k^0, \quad k = 0, \ldots, M_\eta \quad (11.2.42)$$

$$y_{0\,k}^{\ell+1} = y_k^0, \quad k = 0, \ldots, M_\eta \quad (11.2.43)$$

$$x_{j\,0}^{\ell+1} = x_{j\,M_\eta}^{\ell+1}, \quad j = 0, \ldots, M_\xi \quad (11.2.44)$$

$$y_{j\,0}^{\ell+1} = y_{j\,M_\eta}^{\ell+1}, \quad j = 0, \ldots, M_\xi. \quad (11.2.45)$$

We note that the boundary conditions (11.2.44)-(11.2.45) are periodic boundary conditions due to the fact that we sliced the region R along $y = \frac{1}{2}$, $\frac{7}{10} \le x \le 1$. These "periodic boundary conditions" are really continuity conditions. This treatment is analogous to how we treated the θ-variable in difference implementations involving polar coordinates.

We next solve difference equations (11.2.38)–(11.2.45). A variety of methods are used to solve these equations. Difference equations (11.2.38)–(11.2.45) can be difficult to solve. In Figure 11.2.6 we plot the grid generated by solving equations (11.2.38)–(11.2.45). Notice that the resulting grid looks basically similar to the fictitious grid that we drew free-hand in Figure 11.2.1. This grid was generated using $M_\eta = 17$, $M_\xi = 9$ and using a Gauss-Seidel iteration scheme. We should understand that we would get a finer grid and a better approximation of the circle if we used more grid points in the ξ–η plane. The iteration is very much like Gauss-Seidel iterations that we performed in Chapter 10, except that *we must be careful when choosing an initial guess.* It should be obvious that if we choose the usual zero initial guess, we have a disaster, since all of the α's, β's and γ's will be zero. Depending on your choice of initial guess, you can (1) create an

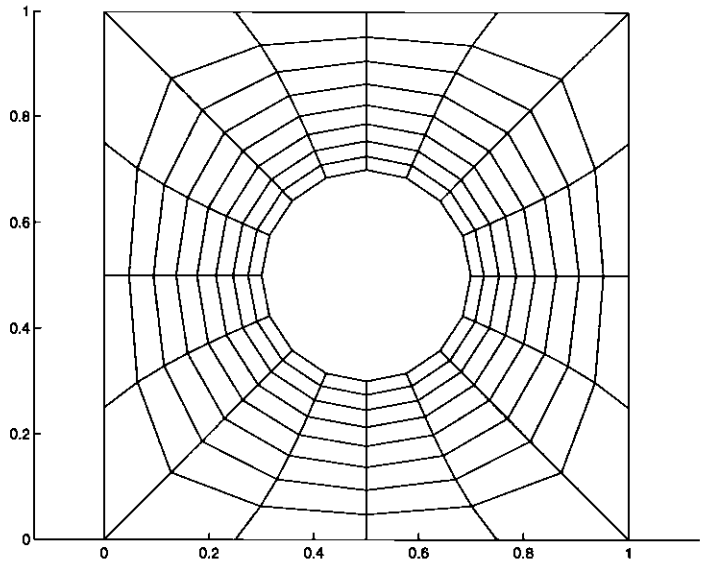

FIGURE 11.2.6. Grid generated by solving difference equations (11.2.38)–(11.2.45).

iteration that will not converge, (2) create an iteration that will converge to a solution that we do not want, or (3) create an iteration that produces the grid given in Figure 11.2.6. To create the solution shown in Figure 11.2.6, we used an initial guess consisting of points on concentric circles about the boundary circle (being careful that our circles stayed in the desired domain). We must remember that the discrete problem we are trying to solve is a nonlinear problem and we are using a solution technique that is very close to a successive approximation scheme (using the Jacobi scheme would result in a successive approximation scheme). We must choose an initial guess in a sufficiently small neighborhood of the solution so that the derivative of the nonlinear operator is less than one in that neighborhood. Hence, to solve difference equations (11.2.38)–(11.2.45), we must work a bit extra to obtain a reasonably nice initial guess.

Remark: We should note that instead of the periodicity boundary condition we used with respect to the η variable, we could have placed a fixed grid on the slice in the region R and used a Dirichlet boundary condition. Surely, this approach would work for an easy geometry such as that given in initial–boundary–value problem (11.2.1)–(11.2.4). However, in general, this is not necessary numerically and constrains our mapping more than we would like. Hence, the periodic boundary condition is preferable.

Now that we have generated a grid on R, really a mapping function

between R and R', it is time to return to the task of approximating the solution to initial–boundary–value problem (11.2.1)–(11.2.4). Theoretically, we would like to solve partial differential equation (11.2.22) along with the appropriate boundary conditions. However, in our case, when we are using a Laplace system to generate the grid (the mapping), we can return to partial differential equation (11.2.9) and note that the lower order terms drop out, since functions ξ and η satisfy $\nabla^2\xi = 0$ and $\nabla^2\eta = 0$. We see that instead of partial differential equation (11.2.22), we can consider the easier partial differential equation

$$V_t = \frac{1}{\mathcal{J}^2}(x_\eta^2 + y_\eta^2)V_{\xi\xi} - \frac{2}{\mathcal{J}^2}(x_\xi x_\eta + y_\xi y_\eta)V_{\xi\eta} + \frac{1}{\mathcal{J}^2}(x_\xi^2 + y_\xi^2)V_{\eta\eta}. \quad (11.2.46)$$

The initial and boundary conditions that we use along with partial differential equation (11.2.46) are

$$V(\xi,\eta,0) = 0, \quad (\xi,\eta) \in R' \quad (11.2.47)$$

$$V(0,\eta,t) = \sin 4\pi\big(x(0,\eta) - 0.5\big), \quad 0 \le \eta \le 1 \quad (11.2.48)$$

$$V(1,\eta,t) = 0, \quad 0 \le \eta \le 1 \quad (11.2.49)$$

$$V(\xi,0,t) = V(\xi,1,t), \quad 0 \le \xi \le 1. \quad (11.2.50)$$

Finding an approximate solution to the transformed initial–boundary–value problem (11.2.46)–(11.2.50) is not that different from the two dimensional problems solved in Chapter 4. Of course, the metric terms x_ξ, x_η, y_ξ, and y_η and the Jacobian $\mathcal{J}$ must be approximated numerically. We use the mapping functions x and y that we have previously computed along with center differencing to approximate these metric terms. The metrics need only be computed once and are used for each time step. Then, one logical scheme for finding an approximate solution to initial–boundary–value problem (11.2.46)–(11.2.50) is to use

$$U_{jk}^{n+1} = U_{jk}^n + \frac{1}{\mathcal{J}_{jk}^2}\Big\{\alpha_{jk}\frac{1}{\Delta\xi^2}\delta_\xi^2 U_{jk}^n - 2\beta_{jk}\frac{1}{4\Delta\xi\Delta\eta}\delta_{\xi 0}\delta_{\eta 0}U_{jk}^n + \gamma_{jk}\frac{1}{\Delta\eta^2}\delta_\eta^2 U_{jk}^n\Big\},$$

$$j = 1,\ldots,M_\xi - 1, \quad k = 1,\ldots,M_\eta \quad (11.2.51)$$

where

$$\alpha_{jk} = \frac{1}{\Delta\eta^2}\Big[\big(\delta_{\eta 0}x_{jk}\big)^2 + \big(\delta_{\eta 0}y_{jk}\big)^2\Big]$$

$$\beta_{jk} = \frac{1}{4\Delta\xi\Delta\eta}\Big[\delta_{\xi 0}x_{jk}\delta_{\eta 0}x_{jk} + \delta_{\xi 0}y_{jk}\delta_{\eta 0}y_{jk}\Big]$$

$$\gamma_{jk} = \frac{1}{\Delta\xi^2}\Big[\big(\delta_{\xi 0}x_{jk}\big)^2 + \big(\delta_{\xi 0}y_{jk}\big)^2\Big]$$

$$\mathcal{J}_{j\,k} = \frac{1}{4\Delta\xi\Delta\eta}\left[\delta_{\xi 0}x_{j\,k}\delta_{\eta 0}y_{j\,k} - \delta_{\eta 0}x_{j\,k}\delta_{\xi 0}y_{j\,k}\right]$$

with initial and boundary conditions

$$U^0_{j\,k} = 0, \quad j = 0, \ldots, M_\xi, \quad k = 0, \ldots, M_\eta, \tag{11.2.52}$$

$$U^n_{0\,k} = \sin 4\pi(x_{0\,k} - 0.5), \quad k = 0, \ldots, M_\eta, \tag{11.2.53}$$

$$U^n_{M_\xi\,k} = 0, \quad k = 0, \ldots, M_\eta, \tag{11.2.54}$$

$$U^n_{j\,0} = U^n_{j\,M_\eta}, \quad j = 0, \ldots, M_\xi. \tag{11.2.55}$$

As we stated earlier, the metrics are computed once and saved, and the scheme is implemented just as we implemented two dimensional parabolic schemes in Chapter 4. It should not be difficult to realize that difference scheme (11.2.51)–(11.2.55) along with the grid generating scheme will provide a second order approximation to the solution of initial–boundary–value problem (11.2.1)–(11.2.4). We should worry about the stability of the scheme. There are some results for equations with nonconstant coefficients, but they are restrictive and difficult. We surely could consider the stability of the constant coefficient analogue to difference scheme (11.2.51). See HW11.2.5. We shall not really worry about these difficulties at this time. We are more interested in the fact that we can use the numerical grid generating method to obtain a grid (mapping) that will allow us to try to solve the problem in the ξ–η plane. Even if the task of solving the problem in the ξ–η plane is difficult, we should realize that it will be possible and may be easier than trying to solve the original problem on R.

Of course, we could just as well formulate an implicit scheme for approximating partial differential equation (11.2.51). As usual, the resulting matrix equation is difficult to solve. We could solve the resulting difference equation using one of the iterative methods given in Chapter 10. See Section 10.14 Recall that the approach we used in Chapter 4 to develop an implicit scheme that was easily solvable was to use an ADI scheme of some sort. It is not obvious how to make an ADI scheme when we have the cross-derivative terms in our equation. However, it is possible to do so, and results concerning ADI schemes for parabolic partial differential equations that include the cross-derivative terms can be found in ref. [48], page 81.

Remark 1: If instead of the Dirichlet boundary conditions that we were given in initial–boundary–value problem (11.2.1)–(11.2.4) we were given Neumann conditions, we would have to be very careful. It is fairly easy to see that the curves $\eta(x, y)$ =constant plotted in the region R do not meet the outer boundary perpendicularly. Hence, it would not be right to consider Neumann boundary conditions in the ξ–η plane. For more information on this topic, see ref. [68].

Remark 2: We should understand that if we use conservation methods to derive our equations in the ξ–η plane, there will be control volumes in the x–y plane analogous to those used in the ξ–η plane. Hence, the resulting

difference equations should be at least approximately conservative in the x–y plane.

The above discussion on grid generation techniques is designed to be only a brief introduction. The area of grid generation techniques is well developed. One of the next steps is to use a Poisson system instead of the Laplace system to generate the grid. The right hand sides of the Poisson system can be chosen in a way to place grid lines where they are needed. There are also the algebraic grid generation techniques. And finally, the ultimate is the complete package that treats the region zonally, using a combination of partial differential equation and algebraic grid generation techniques.

HW 11.2.1 (a) Differentiate equations (11.2.10)–(11.2.13) with respect to ξ and η.
(b) Solve the equations found in part (a) for ξ_{xx}, ξ_{yy}, η_{xx} and η_{yy}.

HW 11.2.2 Show that $\frac{1}{4\Delta x\Delta y}\delta_{x0}\delta_{y0}u_{j\,k}$ is a second order approximation to the derivative v_{xy}.

HW 11.2.3 Repeat the calculation that led to the grid in Figure 11.2.6 using $M_\eta = 33$ and $M_\xi = 17$.

HW 11.2.4 (a) Consider the boundary–value problem

$$\begin{aligned} 0 &= \alpha v_{xx} - 2\beta v_{xy} + \gamma v_{yy}, \quad (x,y) \in (0,1)\times(0,1) \\ v(x,0) &= f_1(x), \quad x \in [0,1] \\ v(1,y) &= f_2(y), \quad y \in [0,1] \\ v(x,1) &= f_3(x), \quad x \in [0,1] \\ v(0,y) &= f_4(y), \quad y \in [0,1] \end{aligned}$$

and the associated difference approximation

$$0 = \alpha\frac{1}{\Delta x^2}\delta_x^2 u_{j\,k} - 2\beta\frac{1}{4\Delta x\Delta y}\delta_{x+}\delta_{y+}u_{j\,k} + \gamma\frac{1}{\Delta y^2}u_{j\,k},$$
$$j = 1,\ldots,M_x - 1, \quad k = 1,\ldots,M_y - 1 \tag{11.2.56}$$
$$u_{j\,0} = f_1(j\Delta x), \quad j = 0,\ldots,M_x \tag{11.2.57}$$
$$u_{M_x\,k} = f_2(k\Delta y), \quad k = 0,\ldots,M_y \tag{11.2.58}$$
$$u_{j\,M_y} = f_3(j\Delta x), \quad j = 0,\ldots,M_x \tag{11.2.59}$$
$$u_{0\,k} = f_4(k\Delta y), \quad k = 0,\ldots,M_y \tag{11.2.60}$$

where α, β are constants and $\beta^2 - \alpha\gamma < 0$. For a grid with $M_x = 6$ and $M_y = 8$, write the set of difference equations (11.2.56)–(11.2.60) in matrix form as $A\mathbf{u} = \mathbf{f}$.

(b) Discuss the symmetry and the diagonal dominance of the matrix A found in part (a).
(c) Let $\alpha = 1.0$, $\beta = 0.5$, $\gamma = 1.0$ and

$$f_1(x) = 0.7 + 0.3x, \quad x \in [0,1]$$

$$f_2(y) = \begin{cases} 1.0 & \text{when } 0 \le y \le 0.125 \\ 1.5 - 4y & \text{when } 0.125 \le y \le 0.375 \\ 0.0 & \text{when } 0.375 \le y \le 0.625 \\ 4(y - 0.625) & \text{when } 0.625 \le y \le 0.875 \\ 1.0 & \text{when } 0.875 \le y \le 1.0 \end{cases}$$

$$f_3(x) = 0.7 + 0.3x, \quad x \in [0,1]$$

$$f_4(y) = 0.5 + 0.2\cos \pi y, \quad y \in [0,1].$$

Use a Gauss-Seidel iteration with $M_x = 16$ and $M_y = 32$ to obtain a solution to difference equations (11.2.56)–(11.2.60).

HW 11.2.5 Consider difference scheme (11.2.51) where we assume that $\mathcal{J}_{jk} = 1$, $\alpha_{jk} = 1$, $\beta = \frac{1}{2}$ and $\gamma = 1$ for all j and k. Discuss the stability of the resulting scheme.

11.3 Grid Refinement

In Section 1.6.4 we discussed how to derive difference equations on nonuniform one dimensional grids. In one dimension everything is nice. It is possible to extend the results given in Section 1.6.4 to two or more dimensions. The resulting grid will still be logically two or three dimensional, so all of the methods work as they do with uniform grids (except that we lose the same order of accuracy, which we discussed in Section 1.6.4). The difficulty with this approach is that it is not very localized. In other words, if we have a problem on $[0,1] \times [0,1]$ (here we go again) with all of the action occurring near the $(0,0)$ corner, we must use a grid like that given in Figure 11.3.1, and the refinement that is included away from the origin is wasted. You can easily look at Figure 11.3.1 and admit that on the grid given there, the number of wasted grid points is not important. But we must understand that Figure 11.3.1 was drawn just to illustrate the problem. In reality, there could be hundreds or thousands of extra grid lines near the origin that would make the extra refinement relevant. And more so, if we considered three or more dimensions, the problem is greater.

Another approach to irregular grids is to allow for a patch of refinement. In Figure 11.3.2, we draw a grid with a large patch near $(0,0)$ and a small patch near $(0,1)$. We refer to the grid placed on $[0,1] \times [0,1]$, the coarse grid plus the patches, as the composite grid. Denote the grid spacing on the coarse and fine parts of the grid by Δx and δx, respectively. Suppose

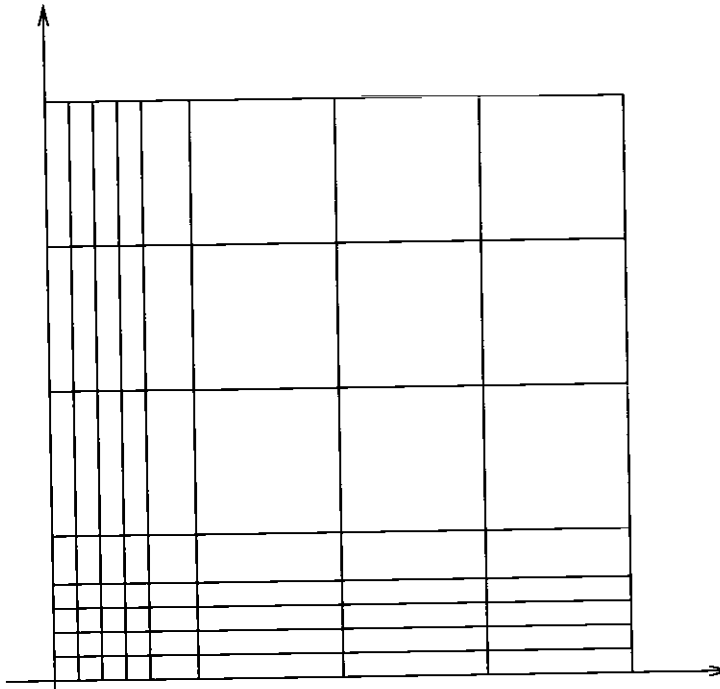

FIGURE 11.3.1. Region with a nonlocal, nonuniform grid.

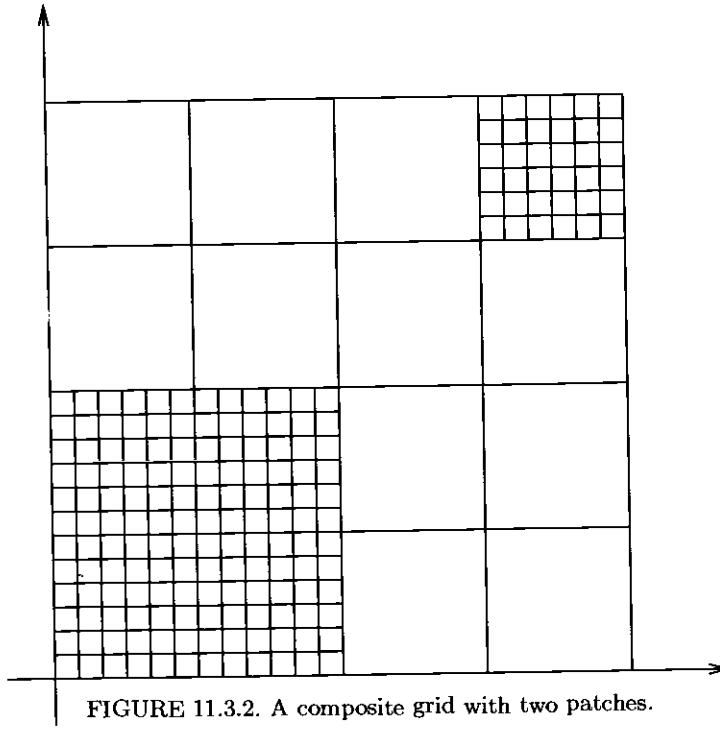

FIGURE 11.3.2. A composite grid with two patches.

the problem we are considering is a simplified version of an injection well at $(1,1)$ and a pumping well at $(0,0)$, and that our modeling describes the flow in the field by the partial differential equation

$$v_t = v_{xx} + v_{yy} + F, \quad (x,y) \in R = (0,1) \times (0,1), \quad t > 0 \qquad (11.3.1)$$

and zero Neumann boundary conditions on the boundary of R. We emphasize that there is no special reason to have different sized patches at the two corners, other than to illustrate that this is possible and sometimes useful to do. In addition, the reduction in grid spacing shown in Figure 11.3.2, $\delta x = \Delta x/6$, is a bit severe. The choice was made to make the fine grid comparable with that given in Figure 11.3.1, and it is often not necessary and most often not prudent to use such a large reduction. Also, we mention that the nonhomogeneous term F associated with the two wells might look something like

$$F(x,y,t) = \delta(x,y) - \delta(x-1,y-1),$$

where the δ's, Dirac delta functions, represent a source and a sink.

We must be careful how we write difference equations to approximate equation (11.3.1) on the grid $\mathcal{G}$ drawn in Figure 11.3.2. We use a conservation law approach, very similar to what we did in Section 4.2.2, Part 1. Instead of working with the grid given in Figure 11.3.2, we will work with a somewhat simpler composite grid given in Figure 11.3.3. We will again denote the region $[0,1] \times [0,1]$ by R and the region where we want a fine grid patch by R_F. Note that in the grid given in Figure 11.3.3, we have chosen a more conservative $\delta x = \Delta x/2$. To aid us in constructing a difference equation approximation of a partial differential equation on the grid $\mathcal{G}$ given in Figure 11.3.3, we partition the grid so that $\mathcal{G} = \mathcal{G}_C \cup \mathcal{G}_I \cup \mathcal{G}_F$, where $\mathcal{G}_C$ represents the coarse grid points outside of R_F, denoted by open squares in Figure 11.3.3; $\mathcal{G}_I$ represents the interface points (the coarse grid points on the boundary of R_F), denoted by filled squares in Figure 11.3.3; and $\mathcal{G}_F$ represents the fine grid points, denoted by open and filled circles in Figure 11.3.3. We separate the grid in this way because each of these types of points must be treated differently. The difference equations on $\mathcal{G}_C$ and $\mathcal{G}_F$ are really quite easy. At the points in either $\mathcal{G}_C$ or $\mathcal{G}_F$, we can use the usual control volumes and obtain the differencing

$$u_{jk}^{n+1} = u_{jk}^n + r_c(\delta_x^2 + \delta_y^2)u_{jk}^n + \Delta t F_{jk}^n \qquad (11.3.2)$$

and

$$u_{jk}^{n+1} = u_{jk}^n + r_f(\delta_x^2 + \delta_y^2)u_{jk}^n + \delta t F_{jk}^n, \qquad (11.3.3)$$

where $r_c = \Delta t/\Delta x^2$ and $r_f = \delta t/\delta x^2$. We should note that at the moment the indexing of the points is being used very loosely. With the patches in there, it is not at all clear how we would index the points. At this time we want only to emphasize that the difference equations obtained at these points are the usual difference equations that we would have obtained if

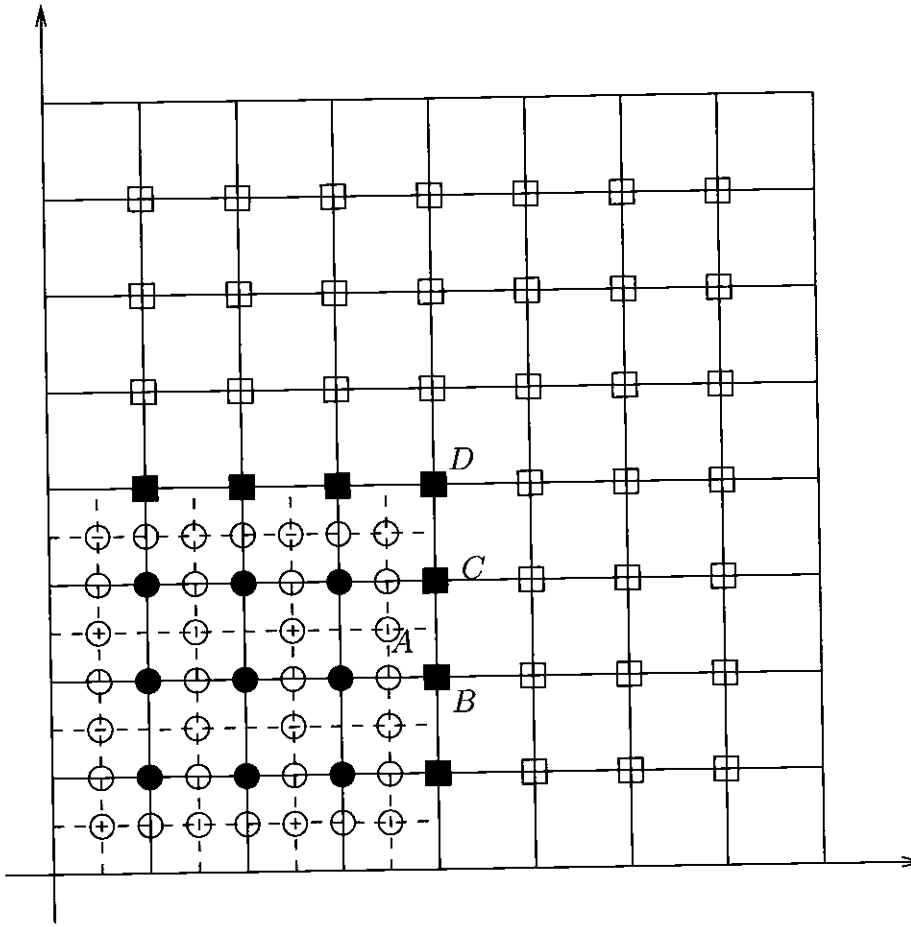

FIGURE 11.3.3. Composite grid. The open squares, □, denote the coarse grid points, $\mathcal{G}_C$, the filled squares, ■, denote the interface points, $\mathcal{G}_I$, and the circles, filled and open, denote the fine grid points, $\mathcal{G}_F$.

we were not using a composite grid. Also, note that we have used different "delta t's" for each equation. We will discuss this later. The only difficulty with either the points in $\mathcal{G}_C$ or in $\mathcal{G}_F$ is with the fine grid points that reach to the boundary of the fine grid region and do not have a value at that point. For example, if we use the usual differencing of a point such as the point labeled A in Figure 11.3.3 or $\left(j-\frac{1}{2},k-\frac{1}{2}\right)$ in Figure 11.3.4, we reach to a nongrid point, half way between points B and C. Here we make a basic assumption that we have constructed the grid so that the boundary of the fine patch is "far enough from the action" to allow us to let the fine grid equation reach to the average of the two appropriate interface grid values. Hence, when the difference equation at point A reaches to the right, we set

$$u^n_{j\,k-1/2} = \left(u^n_{j\,k} + u^n_{j\,k-1}\right)/2. \tag{11.3.4}$$

In other words, at a point such as point $\left(j-\frac{1}{2},k-\frac{1}{2}\right)$, using the notation given in Figure 11.3.4, we use the following difference equation to approximate the partial differential equation.

$$\begin{aligned} u^{n+1}_{j-1/2\,k-1/2} =& u^n_{j-1/2\,k-1/2} + r_f\delta_y^2 u^n_{j-1/2\,k-1/2} + \delta t F^n_{j-1/2\,k-1/2} \\ &+ r_f\Big[0.5\left(u^n_{j\,k} + u^n_{j\,k-1}\right) - 2u^n_{j-1/2\,k-1/2} + u^n_{j-1\,k-1/2}\Big]. \end{aligned} \tag{11.3.5}$$

We next want to discuss how we obtain difference equations at the interface points. There are two types of points of special interest, a point on the interior of the boundary such as point C and a point at a corner such as point D. In Figure 11.3.4, we include a local plot of the grid near the grid points labeled A, B, C and D, including in dotted lines the control volumes associated with the points. In the control volume associated with point C, it is easy to see that we can approximate the flux across the right boundary by

$$\frac{u^n_{j+1\,k} - u^n_{j\,k}}{\Delta x}\Delta y.$$

Noting that the lengths of the top and bottom sides of the control volume are both $\Delta x/2 + \delta x/2$, we can also approximate the flux across the top and bottom of the control volume by

$$\frac{u^n_{j\,k+1} - u^n_{j\,k}}{\Delta y}\left(\frac{\Delta x}{2} + \frac{\delta x}{2}\right)$$

and

$$\frac{u^n_{j\,k} - u^n_{j\,k-1}}{\Delta y}\left(\frac{\Delta x}{2} + \frac{\delta x}{2}\right),$$

respectively. Of course, the most important part of this calculation is how we treat the left part of the boundary. We treat the left boundary in three pieces, marked ab, bc and cd in Figure 11.3.4. We approximate the flux across bc by

$$\frac{u^n_{j\,k} - u^n_{j-1/2\,k}}{\delta x}\delta y$$

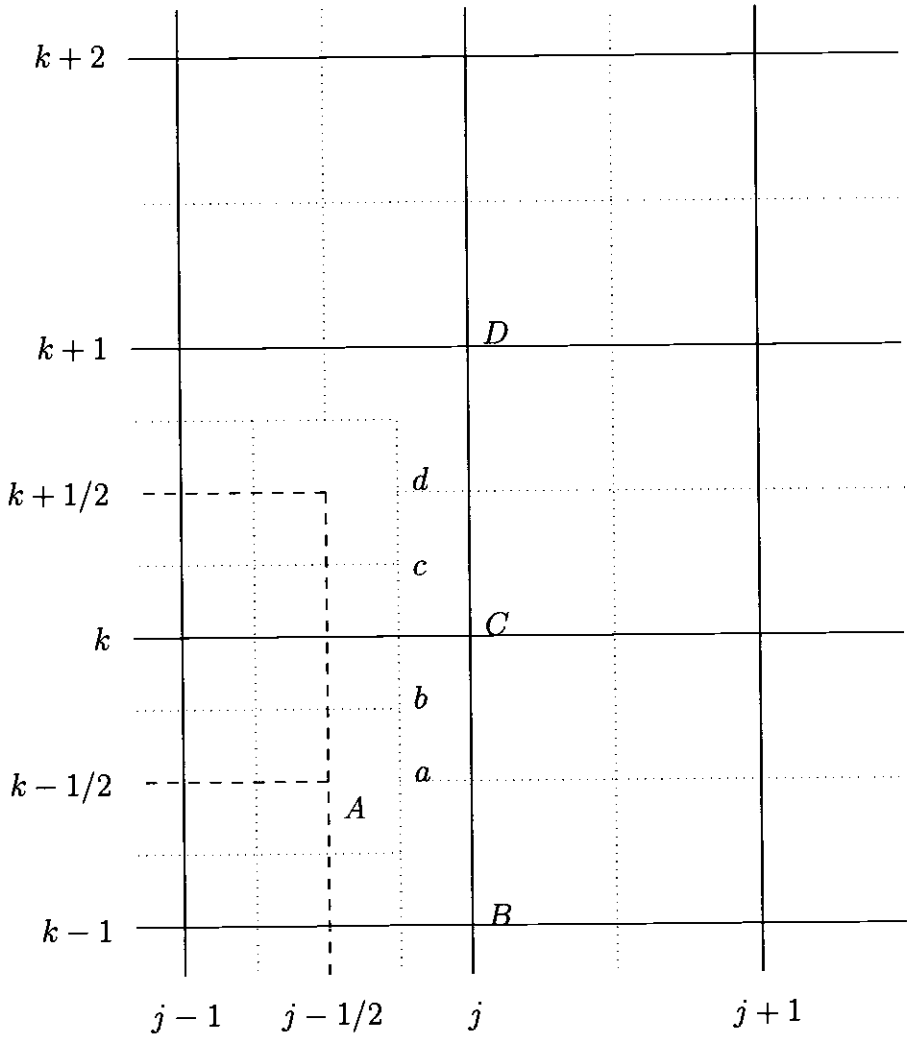

FIGURE 11.3.4. An enlarged version of the region near the points A, B, C and D of the composite grid shown Figure 11.3.3.

and the flux across ab and cd by

$$\frac{(u^n_{jk}+u^n_{jk-1})/2-u^n_{j-1/2\,k-1/2}}{\delta x}\frac{\delta y}{2}$$

and

$$\frac{(u^n_{jk}+u^n_{jk+1})/2-u^n_{j-1/2\,k+1/2}}{\delta x}\frac{\delta y}{2},$$

respectively. We note that we use the values $(u^n_{jk}+u^n_{jk-1})/2$ and $(u^n_{jk}+u^n_{jk+1})/2$ to compute the fluxes across ab and cd. These choices are consistent with the choice used for the flux across the right boundary of the control volume associated with the point $(j-\frac{1}{2},k-\frac{1}{2})$ described earlier and the analogous flux used to define the difference equation associated with the point $(j-\frac{1}{2},k+\frac{1}{2})$. It should be clear that these fluxes give us equal and opposite flows across the boundaries of the interface control volumes, leading to a conservative scheme on the composite grid.

And finally, we now wish to use these approximate fluxes as approximations to the integral form of the conservation law (4.2.26), Part 1. As we approximate the integral form of the conservation law, we must be careful to note that the area of the control volume that we are considering is $(\Delta x/2+\delta x/2)\Delta y$ instead of the usual $\Delta x\Delta y$. Hence, we get

$$\begin{aligned}
&(\Delta x/2+\delta x/2)\Delta y(u^{n+1}_{jk}-u^n_{jk})\\
=&\Delta t\Delta y\frac{u^n_{j+1\,k}-u^n_{jk}}{\Delta x}+\Delta t(\Delta x/2+\delta x/2)\frac{u^n_{jk+1}-u^n_{jk}}{\Delta y}\\
&-\Delta t(\Delta x/2+\delta x/2)\frac{u^n_{jk}-u^n_{jk-1}}{\Delta y}-\Delta t\frac{\delta y}{2}\frac{(u^n_{jk}+u^n_{jk+1})/2-u^n_{j-1/2\,k+1/2}}{\delta x}\\
&-\Delta t\delta y\frac{u^n_{jk}-u^n_{j-1/2\,k}}{\delta x}-\Delta t\frac{\delta y}{2}\frac{(u^n_{jk}+u^n_{jk-1})/2-u^n_{j-1/2\,k-1/2}}{\delta x}\\
&+\Delta t(\Delta x/2+\delta x/2)\Delta y F^n_{jk}
\end{aligned}\tag{11.3.6}$$

or, using the fact that we have chosen $\Delta y=\Delta x$ and $\delta y=\delta x=\Delta x/2$,

$$\begin{aligned}
u^{n+1}_{jk}=&u^n_{jk}+r_c\delta^2_y u^n_{jk}+r_c\frac{4}{3}\Big[-4u^n_{jk}+u^n_{j+1\,k}+\frac{1}{2}u^n_{jk+1}\\
&+\frac{1}{2}u^n_{jk-1}+\frac{1}{2}u^n_{j-1/2\,k+1/2}+u^n_{j-1/2\,k}+\frac{1}{2}u^n_{j-1/2\,k+1/2}\Big].
\end{aligned}\tag{11.3.7}$$

The corner point D is treated just as we treated point C, except that now the control volume about point D is more complex than the control volume about point C. We approximate the flux across the right boundary by

$$\frac{u^n_{j+1\,k+1}-u^n_{jk+1}}{\Delta x}\Delta y,$$

the flux across the larger part of the left boundary by

$$\frac{u^n_{j\,k+1} - u^n_{j-1\,k+1}}{\Delta x}\frac{3}{4}\Delta y,$$

the flux across the small part of the left boundary by

$$\frac{(u^n_{j\,k} + u^n_{j\,k+1})/2 - u^n_{j-1/2\,k+1/2}}{\delta x}\frac{\delta y}{2},$$

the flux across the top boundary by

$$\frac{u^n_{j\,k+2} - u^n_{j\,k+1}}{\Delta y}\Delta x,$$

the flux across the large part of the bottom boundary by

$$\frac{u^n_{j\,k+1} - u^n_{j\,k}}{\Delta y}\frac{3}{4}\Delta x,$$

and the flux across the small part of the bottom boundary by

$$\frac{(u^n_{j-1\,k+1} + u^n_{j\,k+1})/2 - u^n_{j-1/2\,k+1/2}}{\delta y}\frac{\delta x}{2}.$$

If we again use these approximate fluxes to approximate the integral form of the conservation law and do the necessary algebra, we get

$$\begin{aligned} u^{n+1}_{j\,k+1} = u^n_{j\,k+1} + r_c\frac{16}{15}\Big[&-4u^n_{j\,k+1} + \frac{1}{2}u^n_{j\,k} + u^n_{j+1\,k+1} \\ &+ u^n_{j\,k+2} + \frac{1}{2}u^n_{j-1\,k+1} + u^n_{j-1/2\,k+1/2}\Big]. \end{aligned} \tag{11.3.8}$$

If we were now to return to the well problem and use patches with refinement $\delta x = \delta y = \Delta x/2$ (instead of 6), we would like to construct a difference scheme consisting of difference equations that look like equations (11.3.2), (11.3.3), (11.3.5), (11.3.7), and (11.3.8). We use the expression "look like" to emphasize the fact that the equations associated with boundary fine grid points, side interface points, and corner interface points will look different from equations (11.3.5), (11.3.7), and (11.3.8), but they will be derived in the same manner as these equations and will have a similar form when we make allowances for the different orientations. It would be an extremely difficult process to write the different difference equations and cycle through them in a logical manner. To aid in describing what we hope is an easier algorithm, define the coarse grid G to be the grid consisting of $\mathcal{G}_C$ along with the interface points and the fine grid points that would logically be a part of a uniform coarse grid; i.e., in Figure 11.3.3, G is the uniform grid covering $[0,1] \times [0,1]$ with grid spacing $\Delta x = \Delta y$. The grid points on the coarse grid G are those points denoted by squares and filled circles. We then assume that we have a solution given at the n-th time level and proceed as follows.

- Use difference equation (11.3.2) on the coarse grid G using values from the n-th time level on G to produce a solution on G.

- Use difference equation (11.3.3) and equations like equation (11.3.5) on $\mathcal{G}_F$, using values from the n-th time level on $\mathcal{G}_F$ and the appropriate values from the n-th time level on G for the boundary conditions to produce a solution on $\mathcal{G}_F$.

- Use difference equations like equations (11.3.7) and (11.3.8) on the interface points, using values from the n-th time level on G and $\mathcal{G}_F$ to produce a solution on $\mathcal{G}_I$.

- Inject the values found from the fine grid and interface computations into the coarse grid G.

We note that by using this procedure, we do some extra computations. If the region underlying the fine grid patches is not most of the region R (and if the fine patches cover most of R, we should just use a uniform fine grid), these extra computations probably cost less than the logic that would be necessary in a computer program designed to do the computation directly (and much easier to code).

Remark: In Figure 11.3.2, we drew the region with two patches with a refinement factor of six. It should be clear that two or more patches can be handled in the same way that we handled one patch. Also, it should be clear that the conservation law approach to defining the difference scheme will still work where we still interpolate between the adjacent interface points to define the needed intermediate points for defining the difference equations analogous to (11.3.5). Using this approach, the control volumes associated with the interface points will have boundaries of more than three fine grid control volumes adjacent to its inner boundary. And finally, we should understand that if a refinement factor of more than two is necessary, another approach, and probably a more conservative approach, is to have a patch with a refinement factor of two as we described above and have another patch with a refinement factor of two on the fine patch. The algorithm for this approach is very similar to the algorithm given above for one patch, except that there is one more step where the appropriate equations on the "fine-fine grid" must be considered.

HW 11.3.1 Apply the integral form of the conservation law, (4.2.26), Part 1, to the control volume associated with interface point (j, k) to verify equation (11.3.6).

11.3.1 *Grid Refinement: Explicit Schemes for Hyperbolic Problems*

We next want to use the approach that was applied to parabolic equations in Section 11.3 to hyperbolic equations. The approach will work equally well for both types of equations. The only reason that we chose to illustrate the concept of grid refinement for parabolic equations is that it is easier in that setting—the flux across the boundaries is more intuitive.

Before we proceed, let us review the conservation law derivation of difference schemes for two dimensional hyperbolic partial differential equations presented in Section 5.8. We consider the partial differential equation

$$v_t + \alpha v_x + b v_y = 0, \tag{11.3.9}$$

integrate the partial differential equation over the time-space cell

$$[t_n, t_{n+1}] \times [x_{j-1/2}, x_{j+1/2}] \times [y_{k-1/2}, y_{k+1/2}],$$

and perform the integrations that we can. We are left with the following special case of the integral form of the conservation law.

$$\begin{aligned}
&\int_{y_{k-1/2}}^{y_{k+1/2}} \int_{x_{j-1/2}}^{x_{j+1/2}} (v^{n+1} - v^n)\, dx\, dy + \alpha \int_{t_n}^{t_{n+1}} \int_{y_{k-1/2}}^{y_{k+1/2}} v(x_{j+1/2}, y, t)\, dy\, dt \\
&\quad -\alpha \int_{t_n}^{t_{n+1}} \int_{y_{k-1/2}}^{y_{k+1/2}} v(x_{j-1/2}, y, t)\, dy\, dt \\
&\quad +b \int_{t_n}^{t_{n+1}} \int_{x_{j-1/2}}^{x_{j+1/2}} v(x, y_{k+1/2}, t)\, dx\, dt \\
&\quad -b \int_{t_n}^{t_{n+1}} \int_{x_{j-1/2}}^{x_{j+1/2}} v(x, y_{k-1/2}, t)\, dx\, dt = 0.
\end{aligned} \tag{11.3.10}$$

In Section 5.8 we then proceeded to illustrate that we can approximate the fluxes in such a way as to give us the unconditionally unstable centered difference scheme and all of the combinations of conditionally stable, one sided schemes. In Section 9.6.2 we discussed the numerical flux functions associated with the Lax-Wendroff scheme, the Lax-Friedrichs scheme, the MacCormack scheme and the Beam-Warming scheme.

Return to the part of a composite grid pictured in Figure 11.3.4. At the coarse grid points and the interior fine grid points, we proceed as we did in Section 5.8 or Section 9.6.2. Also, as was the case in the previous section, the fine grid points that are adjacent to an interface point (such as point $(j - \frac{1}{2}, k)$ in Figure 11.3.4) also cause no new problems. Thus, as was the case with grid refinement for parabolic equations, we must be careful with three kinds of points: the fine grid points adjacent to the interface that have no interface points to which to reach and the interface points on the sides and the corners of the refinement patch. We will discuss the first two types

of points and assume that the reader can adjust the process to handle the corner interface points.

Before we begin, we emphasize that we will develop refinement schemes by trying to treat the control volumes just as we did when we had uniform grids. As we stated above, we apply the integral form of the conservation law and must approximate the integrals involved. The approximations that we have used in the past involve the function on the boundary (as in the one sided schemes and a part of the flux for the two sided schemes) or a numerical gradient of the function across the boundary (as we had in the parabolic schemes and in the Lax-Wendroff scheme). We should understand that if we want a one sided difference scheme, depending on the signs of coefficients a and b, the difference scheme may not even reach in the direction that causes problems, but of course it may. Also, when we derive one sided schemes on these irregular grids, we do not lose the usual order of accuracy. However, when we use these same approaches to derive grid refinement schemes based on centered difference schemes, we will lose an order of accuracy.

When we consider the point A, $\left(j-\frac{1}{2},k-\frac{1}{2}\right)$, the difficulty is that we may need a value of u^n at points a and $\left(j,k-\frac{1}{2}\right)$. For both of these cases, we use the average of $u^n_{j\,k-1}$ and $u^n_{j\,k}$. This choice is consistent with the common assumption that u^n is a piecewise constant function, constant on each control volume. Once we have function values at points $\left(j-\frac{1}{2},k-\frac{1}{2}\right)$, a, and $\left(j,k-\frac{1}{2}\right)$, the derivation of the difference equation at point A is no different from that at any other of the fine grid points.

When we apply the integral form of the conservation law to the control volume associated with points like point C, we have two special aspects of the derivation with which we must be careful. The first is that the top and bottom sides of the control volumes are only $\Delta x/2+\delta x/2$ long. When the integrals are approximated along the top and bottom boundaries, we must take into account the shorter length of these sides. The second, and probably most difficult, special aspect of the derivation is that the integral along the left hand side of the control volume must be taken in three pieces (as we did in the parabolic case) and done carefully. The major point is that the flux across the parts of the left side, ab, bc and cd should be the same that we used when we applied the integral form of the conservation law to derive difference equations at points $\left(j-\frac{1}{2},k-\frac{1}{2}\right)$, $\left(j-\frac{1}{2},k\right)$ and $\left(j-\frac{1}{2},k+\frac{1}{2}\right)$, respectively. This coordination of fluxes across the boundaries of these control volumes is what produces a conservative scheme.

We generally must be more careful when we include patches of grid in hyperbolic problems than when we do the same for either parabolic or elliptic problems. When we place patches of grid in our computational domain, we always want to place the patch where things are happening. Most often in hyperbolic problems, the interesting phenomena are related to propagating waves. If a wave that is not well resolved on one of the grids passes through

a patch interface, the wave speed will often change. We cannot let this happen. Also, we really do not want the interesting phenomena to be passing out of the patches. M.J. Berger and her co-workers, refs. [4],[5], [3] have developed essentially all of the tools that are needed for grid refinement schemes for hyperbolic problems involving explicit difference schemes. The various schemes allow for self adaptivity, patches on patches, and patches skewed to the principal axes. See ref. [5] for some excellent results obtained using the adaptive grid refinement schemes applied to simulating the flow about an airfoil. For the results presented in ref. [5], a cell centered scheme is used and a variation of interpolation that yields a conservative scheme that retains the second order accuracy of the global scheme.

11.3.2 Grid Refinement for Implicit Schemes

There are times that for some reason we want to or must use an implicit scheme, and we also need to use some sort of grid refinement scheme. It should not be difficult to realize that it will be more difficult to develop a grid refinement scheme for implicit schemes than it was for explicit schemes. We begin by returning to the derivation of difference equations (11.3.2), (11.3.3), (11.3.5), (11.3.7) and (11.3.8). For all of these equations, we approximated the fluxes across the boundaries of the control volumes and used these approximate fluxes in the integral form of the conservation law. When we approximated the time integral in the integral form of the conservation law, we used the rectangle rule, evaluating the functions at $t = t_n$. This application of the rectangle rule is the reason that all of the functions on the right side of the equations have the superscript n. We can also approximate the time integrals by the rectangle rule, evaluating the functions at $t = t_{n+1}$. If we do this, we obtain the following difference equations analogous to difference equations (11.3.2), (11.3.3), (11.3.5), (11.3.7) and (11.3.8).

$$u_{j\,k}^{n+1} = u_{j\,k}^{n} + r_c(\delta_x^2 + \delta_y^2)u_{j\,k}^{n+1} + \Delta t F_{j\,k}^{n+1} \tag{11.3.11}$$

$$u_{j\,k}^{n+1} = u_{j\,k}^{n} + r_f(\delta_x^2 + \delta_y^2)u_{j\,k}^{n+1} + \delta t F_{j\,k}^{n+1} \tag{11.3.12}$$

$$\begin{aligned} u_{j-1/2\,k-1/2}^{n+1} &= u_{j-1/2\,k-1/2}^{n} + r_f\delta_y^2 u_{j-1/2\,k-1/2}^{n+1} + \delta t F_{j-1/2\,k-1/2}^{n+1} \\ &\quad + r_f\left[0.5\left(u_{j\,k}^{n+1} + u_{j\,k-1}^{n+1}\right) - 2u_{j-1/2\,k-1/2}^{n+1} + u_{j-1\,k-1/2}^{n+1}\right] \end{aligned} \tag{11.3.13}$$

$$\begin{aligned} u_{j\,k}^{n+1} &= u_{j\,k}^{n} + r_c\delta_y^2 u_{j\,k}^{n+1} + r_c\frac{4}{3}\Big[-4u_{j\,k}^{n+1} + u_{j+1\,k}^{n+1} + \frac{1}{2}u_{j\,k+1}^{n+1} \\ &\quad + \frac{1}{2}u_{j\,k-1}^{n+1} + \frac{1}{2}u_{j-1/2\,k+1/2}^{n+1} + u_{j-1/2\,k}^{n+1} + \frac{1}{2}u_{j-1/2\,k+1/2}^{n+1}\Big] \end{aligned} \tag{11.3.14}$$

$$\begin{aligned} u_{j\,k+1}^{n+1} &= u_{j\,k+1}^{n} + r_c\frac{16}{15}\Big[-4u_{j\,k+1}^{n+1} + \frac{1}{2}u_{j\,k}^{n+1} + u_{j+1\,k+1}^{n+1} \\ &\quad + u_{j\,k+2}^{n+1} + \frac{1}{2}u_{j-1\,k+1}^{n+1} + u_{j-1/2\,k+1/2}^{n+1}\Big]. \end{aligned} \tag{11.3.15}$$

Obviously, the derivation of the above equations is not very different from that in the explicit case. We should also realize that if we had used the trapezoidal rule to approximate the time integrations, we would obtain a Crank-Nicolson scheme with the average of the fluxes at times $t = t_n$ and $t = t_{n+1}$.

If we were now to try to use the above difference equations, we would have to write a system of equations, one equation for each grid point in the composite grid. We would use difference equation (11.3.11) at the coarse grid points and difference equation (11.3.12) at most of the fine grid points. At the fine grid points that reach to the boundary of the patch to a point that is not a grid point, we use difference equation (11.3.13) or an equation that looks like equation (11.3.13). And we would use difference equations (11.3.14) and (11.3.15) or equations analogous to these equations at the interface points. We write the system of difference equations in matrix form as

$$\mathcal{L}\mathcal{U}^{n+1} = \mathcal{F}^n, \tag{11.3.16}$$

where $\mathcal{U}^{n+1}$ denotes the vector of solution values at the $(n+1)$-st time step; $\mathcal{F}^n$ consists of the boundary conditions, the right hand side values F^n, and u^n; and $\mathcal{L}$ denotes the appropriate matrix. Very little thought is needed to convince one that it is very difficult to arrange the points in any logical order to make it at least civilized to try to write the matrix $\mathcal{L}$, let alone solve system (11.3.16). The time dependent fast adaptive composite grid method (TDFAC), an extension of the fast adaptive composite grid method (FAC), was developed as an efficient scheme for approximating the solution to equation (11.3.16) where the matrix $\mathcal{L}$ is defined by difference equations such as (11.3.11)–(11.3.15).

As we did earlier, in Section 11.3, we partition the composite grid $\mathcal{G}$ as $\mathcal{G} = \mathcal{G}_C \cup \mathcal{G}_I \cup \mathcal{G}_F$. We denote the domain of our problem by R and the region where we will place a fine grid patch (or the fine grid patches) by R_F. Then $\mathcal{G}_C$ will contain the coarse grid points that are outside of the region R_F, $\mathcal{G}_I$ will contain the coarse grid points along the boundary of R_F, and $\mathcal{G}_F$ will contain the fine grid points in R_F. We also use the coarse grid G defined in Section 11.3 as the underlying coarse grid over the region R. We partition G as $G = G_C \cup G_I \cup G_F$, where G_C consists of the coarse grid points outside of R_F ($G_C = \mathcal{G}_C$), G_I consists of the coarse grid points on the boundary of R_F ($G_I = \mathcal{G}_I$), and G_F consists of the coarse grid points in R_F. In Figure 11.3.3 we denoted the points in $\mathcal{G}_C$, $\mathcal{G}_I$, and $\mathcal{G}_F$ by open squares; filled squares; and circles, open and filled, respectively. We see that G_C consists of the points denoted by open squares, G_I consists of the points denoted by filled squares, and G_F consists of the points denoted by filled circles. We describe the TDFAC algorithm in terms of the grids $\mathcal{G}$ and G. In order to pass information between $\mathcal{G}$ and G, we assume that a prolongation or interpolation operator, I, and a restriction operator, I^T,

have been defined such that

$$I : G \to \mathcal{G}$$

and

$$I^T : \mathcal{G} \to G.$$

We note that I and I^T are very much the same as the prolongation and restriction operators used in the multigrid schemes described in Section 10.10, except that in Section 10.10 the prolongation and restriction operators worked on the entire grid. Here, I will be the identity on G_C and G_I and linear interpolation from G_F to $\mathcal{G}_F$. We can use full weighting as our restriction operator, but it might be easier to consider injection. Then I^T will be the identity operator on $\mathcal{G}_C$ and $\mathcal{G}_I$ and injection from $\mathcal{G}_F$ into G_F. We denote the operator on the coarse grid G by L. As we did in Section 10.10, we can define the operator L according to the Galerkin formulation $L = I^T \circ \mathcal{L} \circ I$. Most often, we think of L as and define L as the discrete approximation of the problem on the coarse grid. If this formulation is not the same as the Galerkin definition or at least very similar, there is usually something wrong. We denote the vector of unknowns on the coarse grid at time $t = t_n$ by $\mathbf{u}^n$ and write our coarse grid problem as

$$L\mathbf{u}^n = I^T \mathcal{F}^n = \mathbf{F}^n.$$

Based on the above partitioning of the grids $\mathcal{G}$ and G, we partition $\mathbf{u}$, $\mathcal{U}$, $\mathcal{F}$, $\mathcal{L}$ and L as

$$\mathbf{u} = [\mathbf{u}_C \quad \mathbf{u}_I \quad \mathbf{u}_F]^T, \tag{11.3.17}$$

$$\mathcal{U} = [\mathcal{U}_C \quad \mathcal{U}_I \quad \mathcal{U}_F]^T, \tag{11.3.18}$$

$$\mathcal{F} = [\mathcal{F}_C \quad \mathcal{F}_I \quad \mathcal{F}_F]^T, \tag{11.3.19}$$

$$\mathcal{L} = \begin{pmatrix} \mathcal{L}_{CC} & \mathcal{L}_{CI} & \Theta \\ \mathcal{L}_{IC} & \mathcal{L}_{II} & \mathcal{L}_{IF} \\ \Theta & \mathcal{L}_{FI} & \mathcal{L}_{FF} \end{pmatrix}, \tag{11.3.20}$$

and

$$L = \begin{pmatrix} L_{CC} & L_{CI} & \Theta \\ L_{IC} & L_{II} & L_{IF} \\ \Theta & L_{FI} & L_{FF} \end{pmatrix}. \tag{11.3.21}$$

Begin by noting that the components of $\mathcal{L}$ are defined by difference equations (11.3.11)–(11.3.15), and the components of L are defined by difference

equation (11.3.11) at all of the coarse grid points. Note that $\mathcal{L}_{CC} = L_{CC}$, $\mathcal{L}_{IC} = L_{IC}$ and $\mathcal{L}_{CI} = L_{CI}$. The block $\mathcal{L}_{FF}$ is similar to the block L_{FF} except for the fact that there are more grid points and thus more entries in the $\mathcal{L}_{FF}$ block than in the L_{FF} block. The significant difference between L and $\mathcal{L}$ is in the IF and FI blocks, where L and $\mathcal{L}$ are reaching for grid points in G_I and $\mathcal{G}_I$, and G_F and $\mathcal{G}_F$, respectively. How this is done is what ultimately defines the composite grid operator and makes the difference equations on the composite grid conservative or not conservative.

We can now assume that an approximation of $\mathcal{U}^{n+1}$ is given, $\mathcal{U}^{n+1,k}$, and define a TDFAC cycle as follows.

$$\begin{aligned} &\text{Step 1.} \quad \Delta\mathbf{u}^{n+1,k+1} = L^{-1} I^T \left[\mathcal{F}^n - \mathcal{L}\mathcal{U}^{n+1,k}\right] \\ &\text{Step 2.} \quad \mathcal{U}^{n+1,k+1} = \mathcal{U}^{n+1,k} + I\Delta\mathbf{u}^{n+1,k+1} \\ &\text{Step 3.} \quad \mathcal{U}_F^{n+1,k+1} = \mathcal{L}_{FF}^{-1}\left[\mathcal{F}_F^n - \mathcal{L}_{FI}\mathcal{U}_I^{n+1,k+1}\right] \end{aligned} \tag{11.3.22}$$

where $\Delta\mathbf{u}^{n+1,k+1} = \mathbf{u}^{n+1,k+1} - \mathbf{u}^{n+1,k}$. The most common initial guess used to start the scheme is to set $\mathcal{U}^{n+1,0} = \mathcal{U}^n$ or some extrapolation of the solutions from previous time steps.

In Step 1 the scheme determines a correction to the solution at the coarse grid values. These values are used to update the values of $\mathcal{U}_C$ and $\mathcal{U}_I$ in Step 2. In Step 3 we solve on the patches, using the boundary conditions determined in Step 1 and presented by $\mathcal{L}_{FI}\mathcal{U}_I^{n+1,k+1}$. The definition of $\mathcal{L}_{FI}$ will be such that when the fine grid operator reaches to a point on the boundary of R_F that is not an interface point, the operator will provide the boundary condition that is the average of the neighboring interface points.

Remark: Earlier, we implied that we would use different time steps on the coarse and fine grids. There are times when we can use the same time step on both grids, $\delta t = \Delta t$. Most often, if there is a region in the domain of the problem that requires a finer grid than the rest of the region, the phenomenon being modeled is such that it is also changing in time faster in the refined region than it is on the region that is not refined. In such cases, it is best to use a smaller time step on the fine grid than on the coarse grid. When doing so, it is easiest to choose δt and Δt so that Δt is an integer multiple of δt, say $\Delta t = K\delta t$. The main differences in treating the case of different time steps is that (1) Step 3 now consists of a series of K time steps using δt and (2) the interface conditions must be interpolated in both time and space to provide boundary conditions for the fine grid calculation. For more details, see ref. [28]. This remark is true for both explicit and implicit refinement schemes.

The TDFAC scheme and the related fast adaptive composite grid scheme (FAC) are closely related to a block or adaptive multigrid scheme—where in the FAC-TDFAC schemes, we do not care what type of solver is used on any particular grid. Generally, when two grid levels are present, two cycles, coarse-fine, coarse-fine ($CFCF$), are sufficient to solve the discrete problem

within truncation error. This was shown to be true for a particular case in ref. [30]. When more than two levels are present, a more complicated iteration procedure must be used. For example, when three levels are present, the sequence $CFF^2F - CFF^2FF^2$ has been shown to be effective.

The fast adaptive composite grid methods have been developed for elliptic, parabolic, and hyperbolic problems. In ref. [29] the TDFAC scheme given above is described and proved convergent for quasilinear parabolic problems. See also ref. [28]. In ref. [42] the FAC scheme was first introduced, and it was proved convergent for elliptic boundary–value problems. We should emphasize that in both cases, by convergence we mean that the solutions obtained by the FAC-TDFAC iteration will converge to the solution to the composite grid problem. We are assuming that the composite grid problem has been chosen so as to give a sufficiently accurate approximation of the solution to the problem being considered and that in the case of time dependent problems, the composite grid problem will be a stable scheme. And finally, the TDFAC scheme is applied to hyperbolic initial–boundary–value problems in [1]. Like the explicit grid refinement schemes, the FAC-TDFAC schemes can be designed to be self adaptive. In refs. [47], [44], [45] and [46] the self adaptive FAC-TDFAC schemes are described, tested and applied.

There is a large variety of ways that the FAC-TDFAC schemes can be used other than we have described above. Other than the restriction on the time step for stability, there is no reason that an explicit scheme cannot be used on either the coarse grid or the fine grid. Generally, as long as care is taken to conserve at the patch interfaces, different difference schemes can be used on the coarse grids and the fine grid patches. This technique can be very helpful when we might want to treat the far field calculations less accurately (or just differently) than the main computational region of the problem. Other methods, such as analytic methods or spectral methods, can also logically be used on the patches. The FAC-TDFAC scheme can be viewed as a communication scheme between the coarse grid and the patches. However, it is important that the iteration converge to the appropriate composite grid solution.

It should also be noted that the FAC-TDFAC schemes can be used easily and logically with the grid generation schemes of Section 11.2.3. For example, it is easy to see that one approach to solving initial–boundary–value problem (11.2.1)–(11.2.4) would be to use the approximate method introduced in Section 11.2 as the coarse grid solution and use a generated grid on the patch covering the region

$$\left\{(x,y) : (x,y) \in \left(\frac{1}{4},\frac{3}{4}\right) \times \left(\frac{1}{4},\frac{3}{4}\right), \left(x-\frac{1}{2}\right)^2 + \left(y-\frac{1}{2}\right)^2 > \frac{1}{25}\right\}.$$

Theoretically, this would allow us to work with a smaller generated grid—using a uniform grid for the rest of the problem. Of course, this is not necessary on such an easy problem. It should be clear that for a more

complex geometry the combination of the FAC-TDFAC grid generation approach allows us to introduce intricate generated grids in precisely the region where they are needed.

In addition, it should also be clear that refinement patches can also be used along with the globally generated grid. A generated grid can be used as the coarse grid. Upon inspection of the coarse grid, patches of the generated grid can be chosen where a finer grid is needed. The same grid generation scheme can be used to generate a grid on the patches.

11.4 Unstructured Grids

We will conclude this chapter on irregular regions and grids with a short discussion of a relatively new technique, which we will refer to as the unstructured grid technique. Other than the fact that it is an intriguing technique, it is specially fitting that we end this chapter with the discussion of unstructured grids in that the unstructured grids accommodate both irregular regions and irregular grids very naturally.

Let us return to the region R considered in initial–boundary–value problem (11.2.1)–(11.2.4). The unstructured grid laid over the region R pictured in Figure 11.4.1 consists of a series of points connected by a triangular structure. Even though the grid is too coarse (it is so much easier to draw a coarse grid to illustrate the point), we notice that there are more points near the circular region, where they are needed to describe the circular boundary and to resolve the solution near the circular boundary, where the solution is more complex.

The reader will probably recognize that the grid given in Figure 11.4.1 looks a lot like a finite element grid. If this were a finite element text, we would next define the basis functions on each element and proceed to solve the appropriate variational problem or Galerkin equation. See ref. [71] or [2]. The difficulty with this approach is that we have no control or feel as if we have no control over the conservation principles that are so important for our problems.

Suppose associated with each point in the grid we have a polygonal control volume such that the side of the control volume that intersects the line between any two adjacent grid points is the perpendicular bisector of that line. A portion of an unstructured grid along with control volumes of the type that we have tried to describe is pictured in Figure 11.4.2. If it is hard to see that the portion of the grid drawn in Figure 11.4.2 is a part of the grid drawn in Figure 11.4.1, do not worry about it. The triangular grids along with the appropriate control volumes are difficult to draw (and hence, define). This is a problem that must be faced with unstructured grids and will be discussed later. It should not be difficult to understand that if we have control volumes as drawn in Figure 11.4.2, we can apply the integral form of the conservation law to each of these control volumes and, as we

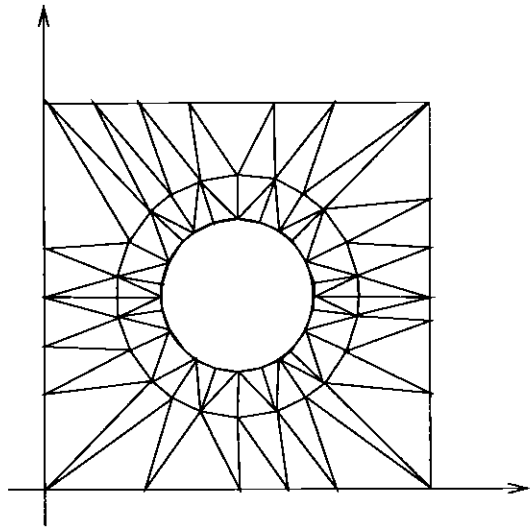

FIGURE 11.4.1. The region $R = \{(x,y) \in (0,1)\times(0,1) : (x-\frac{1}{2})^2+(y-\frac{1}{2})^2 > \frac{1}{25}\}$ with an unstructured grid connected by triangles.

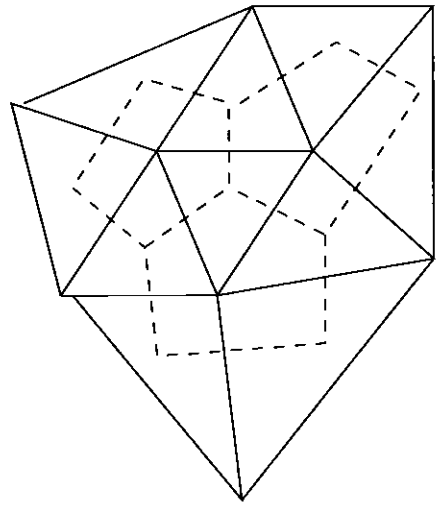

FIGURE 11.4.2. A portion of an unstructured grid along with control volumes about each point.

have always done before, obtain difference equations that approximate the given partial differential equation and/or conservation law. When we applied the integral form of the conservation law in Section 5.8, we applied it to one representative control volume, and this gave us equations at each point. In Section 11.2, we applied the integral form of the conservation law to about four different control volumes and at least claimed that we could then obtain equations at each point. We should understand that the application of the integral form of the conservation law will be more difficult for unstructured grids than it has been in these other cases, because these control volumes may very well be very irregular. However, in principle, the approach is the same as in these other cases: We obtain a discrete equation at each point and solve these discrete equations. We should be aware that since the grid is no longer directionally uniform, it will be very difficult if not impossible to use any type of up-winding schemes for hyperbolic equations. After we have obtained the difference equations, we then must decide how to sequence through the points and how to identify the neighbors of each of the points. These are programming problems.

It should be clear that there is promise in the unstructured grid technique. We must remember that there is some "up front" work that must be done for all of the methods for irregular regions and irregular grids.

The most important problem related to the unstructured grid technique is how to find the triangulation and the associated control volumes. Fortunately, much of this work has been done by the computational geometers. Given a set of points in a two dimensional region, a **Delaunay triangulation** represents a construction that joins these points together to form a set of nonoverlapping triangles. The Delauney triangulation appears to be a perfect triangulation for our work in that it will produce the most equiangular triangles possible. Also, since the Delaunay meshing techniques may be formulated as a sequential and local process, the Delaunay triangulation is particularly well suited for adaptive meshing techniques. From the point of view of our discussion of control volumes given above, the most important property of the Delaunay triangulation is the existence of the geometric dual of the Delaunay triangulation known as the Voronoi tessalation. The Voronoi tessalation is the graph obtained by drawing the median line segments that divide the plane into regions that are closer to a given grid point than to any of the other grid points. These regions are exactly the control volumes described earlier.

It would appear that the problems of constructing an unstructured grid and the associated group of control volumes is solved. There are good algorithms and even programs available for constructing the Delaunay triangulations. However, one of the properties of the Delaunay triangulation that we implied was especially nice, the fact that the Delaunay triangulation produces the most equiangular triangles possible, is not well suited for computational situations where directionally refined meshes are required. There has been a lot of work in obtaining unstructured grids suitable for compu-

tational problems, especially in the area of computational aerodynamics. See ref. [40], [41] and [39]. The Delaunay triangulations are constructed, the grids are stretched, the grids are refined, and solution information is taken into account in the adjustment of the grids. For some remarkable computational results applying unstructured grids to compressible flow problems about complex geometries see refs. [39] and [41].

As we leave the topic of irregular regions and irregular grids, we should emphasize that the problem is not finished. All of the methods discussed in this chapter can be and have been used successfully. There are difficulties with all of the methods. If you are faced with solving a problem involving a complex solution in a very irregular region (worse yet, a irregular three dimensional region), the problem will be difficult using any of the techniques discussed above. And of course, we must realize that we have tried only to introduce the reader to these techniques. For more information on any of these techniques, see the references listed and the references given in those references.

References

[1] John Arvidson and J.W. Thomas. The fast adaptive composite grid method for hyperbolic problems, Preprint.

[2] O. Axelsson and V.A. Barker. *Finite Element Solution of Boundary Value Problems.* Academic Press, New York, 1984.

[3] M. Berger and J. Oliger. Adaptive mesh refinement for hyperbolic partial differential equations. *J. of Comp. Physics*, 53:484–512, 1984.

[4] Marsha J. Berger. On conservation at grid interfaces. Technical Report 84-43, ICASE, NASA, Langley, VA, 1984.

[5] M.J. Berger and A. Jameson. Automatic adaptive grid refinement for the euler equations. *AIAA J.*, 23(4):561–568, 1985.

[6] J.P. Boris and D.L. Book. Flux corrected transport I, SHASTA, a fluid transport algorithm that works. *J. of Comp. Physics*, 11:38, 1973.

[7] Achi Brandt. Guide to multigrid development. In W. Hackbusch and U. Trottenberg, editors, *Multigrid Methods.* Springer-Verlag, Berlin, 1982.

[8] William L. Briggs. *A Multigrid Tutorial.* Society for Industrial and Applied Mathematics, Philadelphia, 1987.

[9] C. Canuto, M.Y. Hussaini, A. Quarteroni, and T.A. Zang. *Spectral Methods in Fluid Dynamics.* Springer-Verlag, New York, 1987.

[10] Julian D. Cole. Modern developments in transonic flow. *SIAM J. Appl. Math.*, 29(4):763–787, 1975.

[11] Edwige Godlewski and Pierre-Arnaud Raviart. *Numerical Approximation of Hyperbolic Systems of Conservation Laws.* Springer-Verlag, New York, 1996.

[12] S.K. Godunov. Finite difference method for numerical computation of discontinuous solutions of the equations of fluid dynamics. *Mat. Sbornik.*, 47:271, 1959.

[13] Gene H. Golub and Charles F. Van Loan. *Matrix Computations.* The Johns Hopkins University Press, Baltimore, 1989.

[14] J.B. Goodman and R.J. LeVeque. On the accuracy of stable schemes for 2D scalar conservation laws. *Mathematics of Computation*, 45(2):15–21, 1985.

[15] J.B. Goodman and R.J. LeVeque. A geometric approach to high resolution TVD schemes. *SIAM J. Numer. Anal*, 25(2):268–284, 1988.

[16] Bertil Gustafsson. On the convergence rate for difference approximations to a mixed initial boundary value problem. Technical Report 33, Department of Computer Science, Uppsala University, Uppsala, Sweden, 1971.

[17] Bertil Gustafsson. The convergence rate for difference approximation to mixed initial boundary value problems. *Mathematics of Computation*, 29(130):396, 1975.

[18] Bertil Gustafsson, Heinz-Otto Kreiss, and Joseph Oliger. *Time Dependent Problems and Difference Methods.* John Wiley & Sons, Inc., New York, 1995.

[19] Bertil Gustafsson, Heinz-Otto Kreiss, and Arne Sundström. Stability theory of difference approximations for mixed initial boundary value problems. II. *Mathematics of Computation*, 26(119):649, 1972.

[20] W. Hackbusch. *Multi-Grid Methods and Applications.* Springer-Verlag, Berlin, 1985.

[21] W. Hackbusch. *Elliptic Differential Equations.* Springer-Verlag, Berlin, 1992.

[22] Louis A. Hageman and David M. Young. *Applied Iterative Methods.* Academic Press, New York, 1981.

[23] A. Harten. High resolution schemes for hyperbolic conservation laws. *J. of Comp. Physics*, 49:357, 1983.

[24] A. Harten and J.M. Hyman. Self-adjusting grid methods for one dimensional hyperbolic conservation laws. *J. of Comp. Physics*, 50:235, 1983.

[25] A. Harten, J.M. Hyman, and P.D. Lax. On finite-difference approximations and entropy conditions for shocks. *Comm. Pure Appl. Math.*, 29:297, 1976.

[26] A. Harten and P.D. Lax. A random choice finite-difference scheme for hyperbolic conservation laws. *SIAM J. Numer. Anal.*,, 18:289, 1981.

[27] Amiran Harten, Peter D. Lax, and Bram Van Leer. On upstream differencing and Godunov–type schemes for hyperbolic conservation laws. *SIAM Review*, 25(1):35–61, 1983.

[28] M. Heroux and J.W. Thomas. TDFAC: A composite grid method for time dependent problems. In *Proceedings of the Fourth Copper Mountain Conference on Multigrid Methods*, pages 273–285, Philadelphia, 1989. SIAM.

[29] Michael A. Heroux. *The Fast Adaptive Composite Grid Method for Time Dependent Problems.* PhD thesis, Colorado State University, 1989.

[30] Michael A. Heroux and J.W. Thomas. A comparison of FAC and PCG methods for solving composite grid problems. *Communications in Applied Numerical Methods*, 8(9):573–584, 1992.

[31] Roger A. Horn and Charles R. Johnson. *Matrix Analysis.* Cambridge University Press, Cambridge, 1985.

[32] P.M. Knupp and S. Steinberg. *The Fundamentals of Grid Generation.* CRC Press, Boca Raton, 1993.

[33] H.O. Kreiss and J. Oliger. Methods for the approximate solution of time dependent problems. Technical Report 10, Global Atmospheric Research Programme, 1973.

[34] P.D. Lax. Shock waves and entropy. In E.A. Zarantonello, editor, *Contributions to Nonlinear Functional Analysis*, pages 603–634. Academic Press, 1971.

[35] Peter D. Lax. *Hyperbolic Systems of Conservation Laws and the Mathematical Theory of Shock Waves.* SIAM, Philadelphia, PA, 1973.

[36] B. Van Leer. Towards the ultimate conservative difference scheme, ii. monotonicity and conservation combined in a second order scheme. *J. Comput. Phys.*, 14:361, 1974.

[37] Randall J. LeVeque. *Numerical Methods for Conservation Laws.* Birkhäuser Verlag, Basel, 1990.

[38] Jerrold E. Marsden and Anthony J. Tromba. *Vector Calculus.* W.H. Freeman and Company, San Francisco, 1976.

[39] D. Mavriplis and A. Jameson. Multigrid solution of the Navier-Stokes equations on triangular meshes. *AIAA J.*, 28:1415–1425, 1990.

[40] D.J. Mavriplis. Adaptive mesh refinement for viscous flows using delaunay triangulation. *J. of Comp. Physics*, 90:271–291, 1990.

[41] D.J. Mavriplis. Unstructured mesh generation and adaptivity. Technical Report 95-26, ICASE, NASA, Langley, VA, 1995.

[42] S. McCormick and J. Thomas. The fast adaptive composite grid (FAC) method for elliptic equations. *Math. Comp.*, 46:439–456, 1986.

[43] S.F. McCormick. *Multigrid Methods.* Society for Industrial and Applied Mathematics, Philadelphia, 1987.

[44] S. McKay and J.W. Thomas. Application of the self adaptive time dependent fast adaptive compos ite grid method. In *Proceedings of the Fourth Copper Mountain Conference on Multigrid Methods*, pages 338–347. SIAM, Philadelphia, 1989.

[45] S. McKay and J.W. Thomas. Application of the fast adaptive composite grid method to nonlinear partial differential equations. *Lectures in Applied Mathematics*, 26:413–428, 1990.

[46] S. McKay and J.W. Thomas. Resolution of moving fronts using the self adaptive time dependent composite grid method. *Communications in Applied Numerical Methods*, 8(9):651–660, 1992.

[47] Steven M. McKay. *Adaptive Methods Applied To The Fast Adaptive Composite Grid Method.* PhD thesis, Colorado State University, 1990.

[48] A.R. Mitchell and D.F. Griffiths. *The Finite Difference Method in Partial Differential Equations.* John Wiley and Sons, New York, 1980.

[49] E.M. Murman. Analysis of embedded shock waves calculated by relaxation methods. *AIAA J.*, 12:626–633, 1974.

[50] E.M. Murman and Julian D. Cole. Calculation of plane steady transonic flow. *AIAA J.*, 9:114–121, 1971.

[51] O. Oleinik. Discontinuous solutions of nonlinear differential equations. *Amer. Math. Soc. Transl. Ser. 2*, 26:95, 1957.

[52] Joseph Oliger. Constructing stable difference methods for hyperbolic equations. In Seymour V. Parter, editor, *Numerical Methods for Partial Differential Equations*, pages 255–271. Academic Press, 1978.

[53] S. Osher. Stability of difference approximations of dissipative type for mixed initial-boundary value problems, I. *Mathematics of Computation*, 23:335, 1969.

[54] S. Osher. Riemann solvers, the entropy condition, and difference approximations. *SIAM J. Numer. Anal*, 21(2):217–235, 1984.

[55] S. Osher and S. Chakravarthy. High resolution schemes and the entropy condition. *SIAM J. Numer. Anal*, 22:995–984, 1984.

[56] S.J Osher. Maximum norm stability for parabolic difference schemes in half-space. In *Hyperbolic Equations and Waves*, pages 61–75, New York, 1970. Springer-Verlag.

[57] Robert D. Richtmyer. *Principles of Advanced Mathematical Physics.* Springer-Verlag, New York, 1978.

[58] P.L. Roe. Approximate Riemann solvers, parameter vectors and difference schemes. *J. of Comp. Physics*, 43:357, 1981.

[59] P.L. Roe. Some contributions to the modeling of discontinuous flows. *Lect. Notes Appl. Math.*, 22:163, 1985.

[60] Walter Rudin. *Real and Complex Analysis.* McGraw-Hill, Inc., New York, 1966.

[61] Piotr K. Smolarkiewicz. A fully multidimensional positive definite advection transport algorithm with small implicit diffusion. *J. of Comp. Physics*, 54(2):325, 1984.

[62] Joel Smoller. *Shock Waves and Reaction-Diffusion Equations.* Springer-Verlag, New York, 1983.

[63] Gary A. Sod. *Numerical Methods in Fluid Dynamics.* Cambridge University Press, Cambridge, 1985.

[64] G. Strang. On the construction and comparison of difference schemes. *SIAM J. Numer. Anal.*, 5:506, 1968.

[65] John C. Strikwerda. *Finite Difference Schemes and Partial Differential Equations.* Wadsworth & Brooks/Cole Advanced Books & Software, Pacific Grove, California, 1989.

[66] P. K. Sweby. High resolution schemes using flux limiters for hyperbolic conservation laws. *SIAM J. Numer. Anal.*, 21(5):995–1010, 1984.

[67] Eitan Tadmor. Numerical viscosity and the entropy condition for conservative difference schemes. *Mathematics of Computation*, 43(168):369, 1984.

[68] Joe F. Thompson, Z.U.A. Warsi, and C. Wayne Mastin. *Numerical Grid Generation.* North-Holland, New York, 1985.

[69] J.M. Varah. Maximum norm stability of difference approximations to the mixed initial boundary-value problem for the heat equation. *Mathematics of Computation*, 2v:31, 1970.

[70] Richard S. Varga. *Matrix Iterative Analysis.* Prentice–Hall, Inc., Englewood Cliffs, New Jersey, 1962.

[71] R. Wait and A. R. Mitchell. *Finite Element Analysis and Applications.* John Wiley & Sons, New York, 1985.

[72] R.F. Warming and Richard M. Beam. On the construction and application of implicit factored schemes for conservation laws. In *Symposium on Computational Fluid Dynamics.* SIAM-AMS, April 1977.

[73] G. Whitham. *Linear and Nonlinear Waves.* Wiley-Interscience, New York, 1974.

[74] H.C. Yee. A class of high-resolution explicit and implicit shock-capturing methods. Technical Report 101088, Ames Research Center, Moffett Field, CA, 1989.

[75] David M. Young. *Iterative Solution of Large Linear Systems.* Academic Press, New York, 1971.

[76] David M. Young and Robert Todd Gregory. *A Survey of Numerical Mathematics, Volumes I and II.* Dover, New York, 1973.

[77] Steven T. Zalesak. Fully multidimensional flux-corrected transport algorithms for fluids. *J. of Comp. Physics*, 31:335, 1979.

Index