

AD-A119 417

STANFORD UNIV CA DEPT OF COMPUTER SCIENCE
TIME-SPLIT METHODS FOR PARTIAL DIFFERENTIAL EQUATIONS. (U)
APR 82 R J LEVEQUE
STAN-CS-82-904

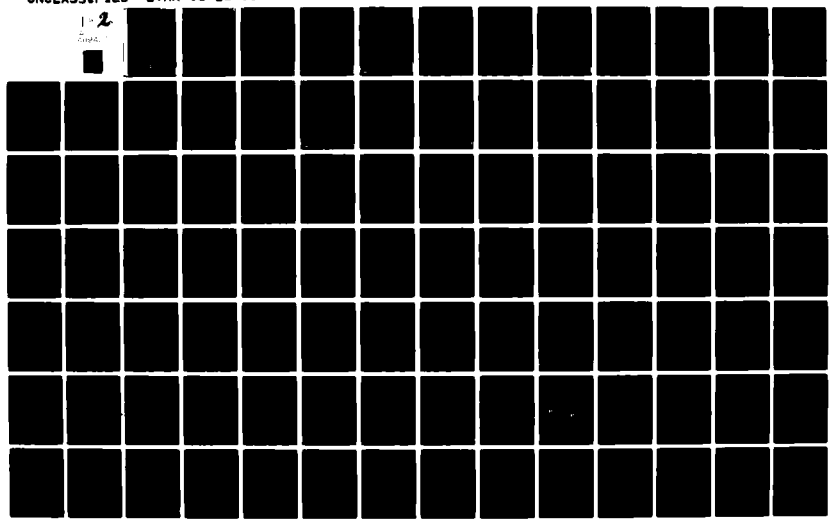
F/6 12/1

N00014-75-C-1132
NL

UNCLASSIFIED

1-2

5040



April 1982

Report No. STAN-CS-82-904

11

AD A119417

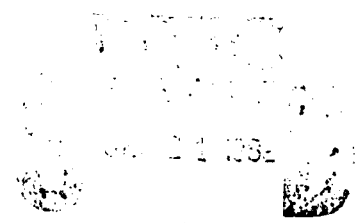
Time-Split Methods for Partial Differential Equations

by

Randall John LeVeque

Department of Computer Science

Stanford University
Stanford, CA 94305



A

DTIC FILE COPY

APPROVED FOR PUBLIC RELEASE, DISTRIBUTION UNLIMITED



82 09 21 007

TIME-SPLIT METHODS FOR PARTIAL DIFFERENTIAL EQUATIONS

A DISSERTATION

SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE

AND THE COMMITTEE ON GRADUATE STUDIES

OF STANFORD UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

N00014-75-c-1132

By

Randall John LeVeque

April 1972



Accession For	
Author	
Title	
Series	
Indexing	<input checked="" type="checkbox"/>
Abstract	<input type="checkbox"/>
Notes	<input type="checkbox"/>
Subject	
Classification Code	
Availability	
Periodical	
Price	
Availability	
Notes	

Time-split methods for partial differential equations

Randall John LeVeque

Abstract. This thesis concerns the use of time-split methods for the numerical solution of time-dependent partial differential equations. Frequently the differential operator splits additively into two or more pieces such that the corresponding subproblems are each easier to solve than the original equation, or are best handled by different techniques. In the time-split method the solution to the original equation is advanced by alternately solving the subproblems. In this thesis a unified approach to splitting methods is developed which simplifies their analysis. Particular emphasis is given to splittings of hyperbolic problems into subproblems with disparate wave speeds.

Three main aspects of the method are considered. The first is the *accuracy* and *efficiency* of the time-split method relative to unsplit methods. We derive a general expression for the splitting error and use it to compute the overall truncation error for the time-split method. This is then used to analyze its efficiency, measured by the amount of work required to obtain a given accuracy.

The second topic is *stability* for split methods. After a demonstration that in general the product of two stable operators need not be stable, some important classes of hyperbolic splittings are identified for which the product of stable approximate solution operators is in fact stable.

The final topic is the proper specification of *boundary data* for the intermediate solutions, e.g., the solution obtained after solving only one of the subproblems. A procedure is described which, for many problems, can be used to transform the given boundary conditions for the original equation into arbitrarily accurate boundary conditions for the intermediate solutions. *Stability* of the initial-boundary value problem is also discussed.

The main emphasis is on hyperbolic problems, and the one-dimensional shallow water equations are used as a specific example throughout. The final chapter is devoted to some other applications of the theory. Two-dimensional hyperbolic problems, convection-diffusion equations, and the Peaceman-Rachford ADI method for the heat equation are considered.

Acknowledgments. I am grateful to my adviser, Joseph Oliger, for suggesting this line of research and for his valuable help and guidance along the way.

It is also a pleasure to acknowledge the influence of Gene Golub and Germund Dahlquist, who have greatly contributed to my education through their knowledge and enthusiasm.

Robert Schreiber and Robert Warming served on my committee and suggested numerous improvements in the content and presentation of this thesis. I have also benefitted from conversations with many other people, including Gerald Browning, Jonathan Goodman, David Gottlieb, and William Gropp.

Special thanks are due to Marsha Berger and Lloyd Trefethen, who have been a source of constructive feedback and unfailing friendship for the past five years.

Thanks also to my parents, for their encouragement and support, and to Sandi Ball, with whom I shared many wonderful experiences, for her love and friendship.

This work has been supported in part by a National Science Foundation Graduate Fellowship, a Hertz Foundation Graduate Fellowship, the Office of Naval Research under contract N00014-75-C-1132 and the National Science Foundation under grant MCS77-02082. Computer time was provided by the Stanford Linear Accelerator Center of the U. S. Department of Energy. The manuscript was produced using \TeX , a computer typesetting system created by Donald Knuth at Stanford.

Contents

1. Introduction	1
1.1. Overview	1
1.2. Some partial differential equations and finite difference methods	2
1.3. The time-split method	6
1.4. Hyperbolic splittings with disparate wave speeds	8
1.5. The shallow water equations	13
1.6. A brief history of splitting methods	14
1.7. Outline and summary of results	17
2. Accuracy and efficiency	20
2.1. Introduction	20
2.2. Truncation error of the time-split method	20
2.3. The splitting error	21
2.4. Computing the splitting error for quasilinear systems	24
2.5. Efficiency analysis for the time-split method on hyperbolic problems	27
2.6. Phase errors	33
2.7. Block triangular systems	37
2.8. Reducing the splitting error	37
2.9. The shallow water equations	39
3. Cauchy stability	47
3.1. Introduction to stability theory	47
3.2. Stability of the time-split method	49
3.3. Simultaneously normalizable splittings	50
3.4. Block triangular systems	52
3.5. The shallow water equations	53
4. Boundary conditions for the intermediate solutions	55
4.1. Introduction and a simple example	55
4.2. Constant coefficient systems—inflow boundaries	62
4.3. Variable coefficient systems—inflow boundaries	68
4.4. Inflow-outflow boundaries	69
4.6. Stability of the initial-boundary value problem	74
5. Other applications of the theory	78
5.1. Introduction	78
5.2. Hyperbolic problems in two space dimensions	78
5.3. Convection-diffusion equations	84
5.4. The Peaceman-Rachford ADI method for parabolic problems	93
References	96

1. Introduction

1.1. Overview.

Splitting methods of one form or another are frequently used in computing numerical solutions to partial differential equations. This thesis concerns one wide class of splitting methods which will be referred to as *time-split methods*. Such methods are also known as *fractional step methods*. These methods apply to time-dependent equations of the form $u_t = A(u)$ for which the differential operator A splits additively into two or more pieces, say $A(u) = A_1(u) + A_2(u)$, such that the subproblems

$$u_t = A_1(u)$$

and

$$u_t = A_2(u)$$

are each easier to solve than the original problem, or are best handled by different techniques. In the time-split method, the solution to the original problem is advanced by alternating between (approximately) solving each of the two subproblems. For example, a *multi-dimensional* problem may be split into one-dimensional subproblems, convection-diffusion or the Navier-Stokes equations may be split into hyperbolic and parabolic subproblems, or a purely hyperbolic problem may be split into subproblems with disparate wave speeds.

The aims of this thesis are twofold. The first is to present a unified framework for studying various aspects of time-split methods. The main idea is to decompose the derivation of a time-split method into two steps. First the exact solution operator for the original problem is approximated to second order accuracy by a product of exact solution operators for the subproblems. Then these exact solution operators are replaced by second order accurate numerical approximations. Many commonly-used splitting methods can be viewed in this manner (see Section 1.6). This viewpoint is not new, but some of its consequences have not been fully exploited.

One advantage of this approach is that the errors in the resulting numerical approximation can be decomposed into errors due to splitting the exact solution operator and errors due to numerically solving the subproblems. The latter errors are well understood when standard numerical methods are applied. Section 2.3 contains some general expressions for the splitting error. This decomposition of errors aids in analyzing the efficiency of the time-split method, defined as the amount of work required to obtain a given accuracy. The size of the splitting error relative to the truncation errors of the numerical methods employed plays a critical role in determining the optimal choice of mesh ratio for the time-split method and in determining how efficient the resulting method will be relative to unsplit methods. This is investigated in detail in Chapter 2.

Another advantage of this viewpoint is that the intermediate solutions (e.g., the solution obtained after solving only one of the subproblems) take on physical meaning. They are second order accurate approximate solutions to some differential equation (though not to the original equation). This is an important realization, particularly when we attempt to specify boundary conditions for the intermediate solutions. Such boundary conditions are often required to implement the time-split method and have frequently been specified in an ad hoc manner, e.g., the boundary conditions from the original equation are imposed on the intermediate solutions as well. More sophisticated approaches, such as the method of undetermined functions[56], derive correct boundary conditions based on the finite difference equations. However, by viewing the intermediate solution as an approximation to a differential equation, it is often possible to derive appropriate boundary conditions without regard to the finite difference methods employed. The given boundary conditions for the original equation are transformed into boundary conditions appropriate for the subproblems. This is the subject of Chapter 4.

The second aim of this thesis is to investigate the applicability of the time-split method to one particular class of problems, namely to hyperbolic systems of equations which are split into subproblems with disparate wave speeds. The original problem either has all fast waves or some fast waves and some slow waves. This splitting may be advantageous if the "fast" subproblem can be solved more efficiently than the full system. The remaining subproblem can also be solved more efficiently than the full system since only slow waves are present. Such problems are described in detail in Section 1.4.

Time-split methods for hyperbolic problems have not been studied extensively in the past, but the results presented here indicate that in many situations they are quite valuable.

Hyperbolic problems also provide specific examples for the general theory being developed. For example, both the efficiency analysis of the time-split method and the procedure for specifying intermediate boundary conditions are introduced by considering hyperbolic examples. A few other applications are treated in Chapter 5.

1.2. Some partial differential equations and finite difference methods.

Time-dependent partial differential equations arise in modeling a wide variety of physical phenomena. Simple examples in two space dimensions include the parabolic *heat equation*

$$u_t = u_{xx} + u_{yy} \quad (1.1)$$

and the hyperbolic *wave equation*

$$u_{tt} = u_{xx} + u_{yy}. \quad (1.2)$$

The latter equation can be rewritten as a first order hyperbolic system of equations in the variables u_t , u_x , and u_y . A general first order hyperbolic system has the form

$$u_t = Au_x + Bu_y \quad (1.3)$$

where u is now a vector and A and B are diagonalizable matrices with real eigenvalues. For the wave equation (1.2) A and B are constant, but in a more general *variable coefficient problem*, A and B could depend on x , y , and t . If A and B also depend

on the solution u then the problem is said to be *quasilinear*. The inviscid Euler equations of gas dynamics are of this form, for example.

Practical problems often include both first and second order spatial derivatives. The simplest example is the scalar *convection-diffusion equation* in one space dimension, which has the form

$$u_t = cu_x + \nu u_{xx} \quad (1.4)$$

for some constants c and $\nu > 0$. More realistically, the compressible *Navier-Stokes equations* for viscous flow in two dimensions constitute a quasilinear system of the form

$$u_t = Au_x + Bu_y + Cu_{xx} + Du_{yy} + Eu_{xy} \quad (1.5)$$

where each of the matrices is a function of u .

Inhomogeneous terms can also arise in practice. For example, the primitive equations of atmospheric flow (the *shallow water equations*) are a quasilinear system of the form (1.3) with an undifferentiated vector $F(u)$ representing Coriolis forces added to the right hand side.

Lower-order terms also occur in *reaction-diffusion equations* of the form

$$u_t = \nu(u_{xx} + u_{yy}) + F(u). \quad (1.6)$$

Here F represents the chemical kinetics of a reacting system with diffusivity $\nu > 0$.

All of the examples given above are of the general form

$$u_t = A(u) \quad (1.7)$$

where $A(u)$ depends on u and its spatial derivatives. It may also depend on t and the spatial variables although this dependence is not explicitly shown.

Initial boundary value problems will be discussed in detail later in this thesis, but for now we restrict our attention to the *Cauchy problem*, which consists of the equation (1.7) on the unbounded spatial domain $-\infty < x < \infty$, $-\infty < y < \infty$ (in two dimensions) together with initial data $u(x, y, t_0) = f(x, y)$.

If the problem is *well-posed* (as all of the examples above are) then the initial conditions uniquely determine the solution at any later time t_1 . We write

$$u(t_1) = S(t_1, t_0)u(t_0). \quad (1.8)$$

In general the *solution operator* $S(t_1, t_0)$ is nonlinear, but satisfies the semigroup property

$$S(t_2, t_0) = S(t_2, t_1)S(t_1, t_0) \quad (1.9)$$

if $t_0 \leq t_1 \leq t_2$.

If t does not appear explicitly in the coefficients of the differential equation, then the equation is said to be *autonomous* and the solution operator depends only on the time elapsed:

$$S(t_1, t_0) = S(t_1 - t_0).$$

For notational convenience we will assume that this is so unless otherwise stated.

Most practical problems cannot be solved exactly. Instead the solution must be approximated numerically. We will be concerned only with finite difference approximations. For such methods a grid is laid out over the spatial domain and an approximate solution at all gridpoints is obtained at each of a sequence of times t_0, t_1, \dots . In general we assume that $t_0 = 0$ and that $t_n = nk$ for some timestep k . For convenience we assume that the grid is uniform with equal mesh spacing h in all spatial coordinate directions, although this is not necessary. We will always use λ to denote the *mesh ratio*:

$$\lambda = k/h.$$

Numerical approximations are denoted by capital letters. In one space dimension U_m^n is the approximation to $u(x_m, t_n)$ where $x_m = mh$. In higher dimensions more subscripts are added.

We will restrict our attention to two-level difference schemes. This means that U^{n+1} is determined solely by U^n via some relation

$$U^{n+1} = Q(k)U^n. \quad (1.10)$$

This is the difference analogue to

$$u(t_{n+1}) = S(k)u(t_n)$$

and the finite difference operator $Q(k)$ is an approximation to the solution operator $S(k)$. The method is said to be *accurate of order p* if, for smooth functions u , the *local truncation error* $(Q(k) - S(k))u$ is $O(k^{p+1})$ as $k \rightarrow 0$ with some fixed relation between k and h .

As an example, consider a one-dimensional constant coefficient hyperbolic equation

$$u_t = Au_x.$$

Here $u \in \mathbb{R}^r$ is a vector and $A \in \mathbb{R}^{r \times r}$ is a square matrix. By Taylor series expansions we find that the exact solution satisfies

$$\begin{aligned} u(x, t+k) &= u(x, t) + ku_t(x, t) + \frac{1}{2}k^2 u_{tt}(x, t) + \dots \\ &= u(x, t) + kAu_x(x, t) + \frac{1}{2}k^2 A^2 u_{xx}(x, t) + \dots \\ &= (I + kA\partial_x + \frac{1}{2}k^2 A^2 \partial_x^2 + \dots)u(x, t) \\ &= \exp(kA\partial_x)u(x, t). \end{aligned}$$

We thus have $S(k) = \exp(kA\partial_x)$, as defined by the series expansion for the exponential. It is convenient to use this exponential notation for the solution operators of constant coefficient problems.

The Lax-Wendroff method. If the expansion for $\exp(kA\partial_x)$ is truncated after the first three terms and the differential operators ∂_x and ∂_x^2 are replaced by appropriate finite difference operators, we obtain the familiar *Lax-Wendroff method*:

$$U_m^{n+1} = (I + kAD_0 + \frac{1}{2}k^2 A^2 D_+ D_-)U_m^n \quad (1.11)$$

where

$$\begin{aligned}
 D_0 U_m &= \frac{1}{2h}(U_{m+1} - U_{m-1}), \\
 D_+ U_m &= \frac{1}{h}(U_{m+1} - U_m), \\
 D_- U_m &= \frac{1}{h}(U_m - U_{m-1}), \\
 D_+ D_- U_m &= \frac{1}{h^2}(U_{m+1} - 2U_m + U_{m-1}).
 \end{aligned}$$

The value U_m^{n+1} is thus determined by the values U_{m-1}^n , U_m^n , and U_{m+1}^n . This is conveniently denoted by showing the *stencil* of the method as in Figure 1.1.

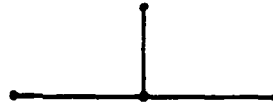


FIG. 1.1. Stencil for Lax-Wendroff.

The numerical operator $Q(k)$ is defined by equation (1.11). This Lax-Wendroff operator appears so often in the sequel that we will introduce the following notation for it, which shows the dependence on the coefficient matrix A explicitly:

$$LW(A, k) = I + kAD_0 + \frac{1}{2}k^2 A^2 D_+ D_- \quad (1.12)$$

Strictly speaking, this operator also depends on h , or, equivalently, on the mesh ratio $\lambda = k/h$, but λ will be assumed to be fixed. Analogous methods can be defined for variable coefficient or quasilinear hyperbolic systems. The same generic symbol $LW(A, k)$ will be used for all of these methods although in general they will be more complicated than in (1.12).

The Lax-Wendroff method is second order accurate since the local truncation error is $O(k^3)$:

$$[LW(A, k) - \exp(kA\partial_x)]u(x, t) = -\frac{1}{6}k^3(A^3 - \frac{1}{\lambda^2}A)u_{xxx} + O(k^4). \quad (1.13)$$

The Crank-Nicolson method. As another example, consider the one dimensional heat equation

$$u_t = u_{xx}. \quad (1.14)$$

The solution operator for this problem is $S(k) = \exp(k\partial_x^2)$. Explicit methods for parabolic problems are generally stable only if the timestep k is very small relative to

h. For this reason implicit methods are often used instead. One popular method is the second order accurate *Crank-Nicolson method*,

$$(1 - \frac{1}{2}kD_+D_-)U_m^{n+1} = (1 + \frac{1}{2}kD_+D_-)U_m^n, \quad (1.15)$$

for which

$$Q(k) = (1 - \frac{1}{2}kD_+D_-)^{-1}(1 + \frac{1}{2}kD_+D_-).$$

This corresponds to using a rational approximation to the exponential solution operator. To implement this method a tridiagonal system of linear equations must be solved at each iteration. This can be done quite efficiently. Because all of the U_m^{n+1} must be determined simultaneously, the method is said to be *implicit*. Lax-Wendroff, by comparison, is an *explicit* method. The stencil for Crank-Nicolson is shown in Figure 1.2.

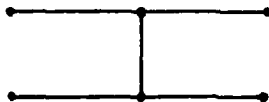


FIG. 1.2. Stencil for Crank-Nicolson.

In two space dimensions the heat equation (1.1) can be solved by a similar method:

$$(1 - \frac{1}{2}k(D_{+x}D_{-x} + D_{+y}D_{-y}))U_{m,j}^{n+1} = (1 + \frac{1}{2}k(D_{+x}D_{-x} + D_{+y}D_{-y}))U_{m,j}^n, \quad (1.16)$$

where, for example, D_{+x} is the forward difference operator in the x -direction. Unfortunately, this no longer leads to a tridiagonal system of equations but rather to a more complicated system which cannot be solved nearly as efficiently. It was this problem which led to the introduction of some of the first splitting methods. One such method is the *locally one-dimensional (LOD) method* in which the solution to (1.1) is advanced by first solving $u_t = u_{xx}$ approximately using (1.15) and then solving $u_t = u_{yy}$ approximately using the same method in the y -direction. In this manner only one-dimensional problems need be solved. The LOD method is one special case of the *time-split method*, which will now be described more generally.

1.3. The time-split method.

Consider again the general problem (1.7) and suppose that the function $A(u)$ splits additively into two or more pieces which are most naturally handled separately. Restricting our attention to two pieces, suppose A is of the form

$$A(u) = A_1(u) + A_2(u), \quad (1.17)$$

where each of the *subproblems*

$$u_t = A_1(u) \quad (1.18a)$$

and

$$u_t = \mathcal{A}_2(u) \quad (1.18b)$$

is easier to solve than the full problem (1.7). As we have already seen, this is the case for the heat equation (1.1) when $\mathcal{A}_1(u) = u_{xx}$ and $\mathcal{A}_2(u) = u_{yy}$. It may also prove useful to handle the different space dimensions in the hyperbolic system (1.3) separately. For other equations the natural splitting is between terms describing different physical processes. In the convection-diffusion equation (1.4) we may take $\mathcal{A}_1(u) = cu_x$ and $\mathcal{A}_2(u) = \nu u_{xx}$, thus splitting the mixed problem up into separate hyperbolic and parabolic equations. The reaction-diffusion equation (1.6) might be handled similarly. The Navier-Stokes equation (1.5) could well be split into five separate pieces.

Splittings have long been used for all of these problems and in many other contexts as well. Some history and references are given in Section 1.6.

We now discuss in more detail the implementation of the time-split method once a splitting of the form (1.17) has been decided upon. The subproblems (1.18a) and (1.18b) have corresponding solution operators $S_1(k)$ and $S_2(k)$. The basic assumption is that these operators are easier to approximate than $S(k)$ is. The time-split method is based on the fact that

$$S(k) \approx S_2(k)S_1(k) \quad (1.19)$$

when k is small. In some cases this splitting is in fact exact. For the heat equation (1.1) with the LOD splitting, for example, we have $S(k) = \exp(k(\partial_x^2 + \partial_y^2))$ while $S_1(k) = \exp(k\partial_x^2)$, $S_2(k) = \exp(k\partial_y^2)$. Since the differential operators ∂_x^2 and ∂_y^2 commute, we find that $S(k) = S_2(k)S_1(k)$. For variable coefficient problems, or systems of equations, the splitting (1.19) is not exact in general. For example, the same LOD splitting on the constant coefficient hyperbolic system (1.3) has an error

$$\begin{aligned} S_2(k)S_1(k) - S(k) &= \exp(kB\partial_y) \exp(kA\partial_x) - \exp(k(A\partial_x + B\partial_x)) \\ &= \frac{1}{2}k^2(BA - AB)\partial_x\partial_y + O(k^3) \end{aligned} \quad (1.20)$$

as can be verified by expanding the exponentials. In this case the splitting is exact only if the matrices A and B commute. Otherwise the local error on smooth solutions is $O(k^2)$ and hence the splitting is only first order accurate.

A simple second order splitting was introduced for this problem by Strang[49] who noted that

$$\exp(k(A\partial_x + B\partial_y)) = \exp(\frac{1}{2}kA\partial_x) \exp(kB\partial_y) \exp(\frac{1}{2}kA\partial_x) + O(k^3).$$

In fact the same type of splitting is second order accurate (on smooth solutions) for general problems of the form (1.7). The general *Strang splitting* is

$$S(k) \approx S_1(k/2)S_2(k)S_1(k/2). \quad (1.21)$$

If the equation depends explicitly on t , then the appropriate form of the splitting is

$$S(t+k, t) \approx S_1(t+k, t + \frac{1}{2}k)S_2(t+k, t)S_1(t + \frac{1}{2}k, t).$$

By the semigroup property (1.9), this can be written as

$$\begin{aligned} S(t+k, t) &\approx [S_1(t+k, t + \frac{1}{2}k)S_2(t+k, t + \frac{1}{2}k)] \\ &\quad \times [S_2(t + \frac{1}{2}k, t)S_1(t + \frac{1}{2}k, t)]. \end{aligned} \quad (1.22)$$

When viewed in this way it is apparent that second order accuracy may also be retained by using a splitting of the form (1.19) but reversing the order of S_1 and S_2 in alternate time steps.

Strang[49] proves that this splitting is second order accurate on a general nonlinear problem. This proof is repeated in Section 2.3, where we also compute a general expression for the error in the splitting.

Once the appropriate splitting of the exact solution operator has been chosen, the *time-split method* results from replacing the exact solution operators $S_1(k)$ and $S_2(k)$ by approximations $Q_1(k)$ and $Q_2(k)$. A numerical method based on the splitting (1.21) would thus be

$$U_m^{n+1} = Q_1(k/2)Q_2(k)Q_1(k/2)U_m^n. \quad (1.23)$$

In practice U_m^{n+1} is computed via the sequence

$$\begin{aligned} U_m^* &= Q_1(k/2)U_m^n \\ U_m^{**} &= Q_2(k)U_m^* \\ U_m^{n+1} &= Q_1(k/2)U_m^{**} \end{aligned} \quad (1.24)$$

where we have introduced nonphysical *intermediate solutions* U^* and U^{**} . When several steps of (1.23) are applied successively the adjacent $Q_1(k/2)$ operators can be combined into $Q_1(k)$, and the half-step operators need only be applied at the beginning and immediately before printout, i.e.,

$$U_m^n = Q_1(k/2)Q_2(k)Q_1(k) \cdots Q_1(k)Q_2(k)Q_1(k/2)U_m^0.$$

When the original problem is split into more than two pieces, say

$$A(u) = A_1(u) + A_2(u) + \cdots + A_p(u),$$

the following splitting is second order accurate:

$$S(k) \approx S_1(k/2)S_2(k/2) \cdots S_{p-1}(k/2)S_p(k)S_{p-1}(k/2) \cdots S_2(k/2)S_1(k/2).$$

This is easily proved by induction (see Gottlieb[23]).

1.4. Hyperbolic splittings with disparate wave speeds.

This thesis is mainly an investigation into the applicability of time-split methods to pure hyperbolic systems whose solutions consist of waves traveling at disparate speeds. Consider the one-dimensional hyperbolic constant coefficient system

$$u_t = Au_x. \quad (1.25)$$

The $r \times r$ matrix A is assumed to be diagonalizable with real eigenvalues $\mu_1, \mu_2, \dots, \mu_r$. If X is the matrix of right eigenvectors of A , then

$$A = XMX^{-1}$$

where $M = \text{diag}(\mu_1, \mu_2, \dots, \mu_r)$ is a diagonal matrix. The solution to (1.25) with initial conditions

$$u(x, 0) = f(x)$$

is given by

$$\begin{aligned} u(x, t) &= \exp(tA\partial_x)u(x, 0) \\ &= X \exp(tM\partial_x)X^{-1}f(x). \end{aligned}$$

Set $v(x) = X^{-1}f(x)$. Then

$$u(x, t) = X \begin{bmatrix} v_1(x + t\mu_1) \\ v_2(x + t\mu_2) \\ \vdots \\ v_n(x + t\mu_r) \end{bmatrix}.$$

In general each component $u_j(x, t)$ of the solution is a linear combination of waves traveling at the various speeds $\mu_1, \mu_2, \dots, \mu_r$. Eigenvalues μ_j with large amplitude give rise to fast waves, those with small amplitude, to slow waves.

Suppose now that the eigenvalues are ordered by magnitude, and that some of them are much larger than others:

$$|\mu_1| \leq |\mu_2| \leq \dots \leq |\mu_p| \ll |\mu_{p+1}| \leq \dots \leq |\mu_r|. \quad (1.26)$$

Now consider the use of a finite difference scheme for solving (1.25). Throughout this thesis we will restrict our attention to the Lax-Wendroff method for hyperbolic problems, both in computational examples and in some of the theory (for example in Section 2.5). The same sort of analysis can be applied to other methods with similar results, but it seems most instructive to concentrate on one particular method.

The Lax-Wendroff method, like all explicit methods, is only conditionally stable. This places a restriction on the size of the time step that can be used. For Lax-Wendroff this stability condition is

$$\frac{k}{h} \rho(A) \leq 1. \quad (1.27)$$

where $\rho(A) = |\mu_r|$ is the *spectral radius* of A . The fastest waves thus dictate the size of the timestep that can be taken. Accuracy considerations also influence the size of the timestep. In fact the fastest waves are computed most efficiently (in the sense that the least work is required to achieve a given accuracy) if the mesh ratio $k/h \approx 1/\rho(A)$ is used. This will be shown in Section 2.5.

Slow waves, on the other hand, can be accurately (and more efficiently) computed using much larger timesteps. The question is whether a split method can be used to compute accurate overall solutions more efficiently.

If the matrix A is diagonal, then the system decouples into r separate scalar equations, each of which can be solved independently using the appropriate mesh ratio. More generally, we can split the matrix A into pieces A_s and A_f corresponding to slow waves and fast waves,

$$A_s = X M_s X^{-1}, \quad A_f = X M_f X^{-1} \quad (1.28)$$

where

$$M_s = \text{diag}(\mu_1, \dots, \mu_p, 0, \dots, 0)$$

$$M_f = \text{diag}(0, \dots, 0, \mu_{p+1}, \dots, \mu_r).$$

This essentially decouples the system into slow and fast parts. Since the matrices A_s and A_f commute, splittings of the form (1.19) or (1.21) are exact and nearly optimal mesh ratios can be used for each part.

Realistic problems can never be split so easily. For variable coefficient or quasilinear systems there will almost always be a splitting error to contend with. It is also generally undesirable or even impossible to use a splitting of the form (1.28), since the eigenvectors are themselves variable.

However, it is not necessary to split by characteristic variables as in (1.28), and the time-split method is often advantageous even when the splitting error is nonzero. Suppose, for example, that r is large but that the matrix A has only a few large eigenvalues. It may be the case that relatively few elements of A contribute to the fast waves. We could then split A as $A = A_f + A_s$ in such a way that A_f is sparse compared to A while A_s has only small eigenvalues. Because of its sparsity, taking small timesteps on A_f requires less work than taking small timesteps with the full matrix A . The matrix A_s can be handled more efficiently using larger timesteps. We could thus consider using a scheme of the form

$$U_m^{n+1} = Q_f(k/2)Q_s(k)Q_f(k/2)U_m^n \quad (1.29)$$

with

$$Q_s(k) = LW(A_s, k)$$

$$Q_f(k/2) = (LW(A_f, k/m))^{m/2} \quad (1.30)$$

for some even integer m . The accuracy and the efficiency of such a method relative to an unsplit method, say $LW(A, k/m)$, depends greatly on the nature of the splitting error. This will be studied in detail in Chapter 2.

Example 1.1. An interesting model system for problems of this form is a block triangular system with

$$A = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}. \quad (1.31)$$

Suppose the eigenvalues of A_{11} are large relative to those of A_{22} and consider the splitting

$$A_f = \begin{bmatrix} A_{11} & 0 \\ 0 & 0 \end{bmatrix}, \quad A_s = \begin{bmatrix} 0 & A_{12} \\ 0 & A_{22} \end{bmatrix}. \quad (1.32)$$

The effectiveness of the split method depends greatly on the coupling A_{12} between the different time scales. This is analyzed in Section 2.7. In Section 2.8 we present a simple procedure for changing variables to reduce the coupling.

Perturbed problems. The time-split method can also be very effective on equations which are small perturbations of some equation for which the exact solution operator is known. We will refer to such problems as *perturbed problems*. For example, consider a variable coefficient problem in which the coefficients have large mean values and small variation. It may then be possible to split off a constant coefficient problem $u_t = A_f u_x$ that can be solved exactly, leaving behind the small perturbations for A_s . We can then

use (1.20) and take $Q_f(k, 2) = \exp(\frac{1}{2}kA_f\partial_x)$ with no error. Since the dominant part of the operator is being handled exactly, substantial increases in efficiency are possible. The one-dimensional shallow water equations introduced in the next section are of this form.

More generally we may divide the computational domain into subintervals and split out a different constant matrix A_f on each subinterval. This might be appropriate if the coefficients are slowly (but widely) varying so that perturbations about the local mean value are small. In this case A_f would be piecewise constant. Alternatively we can view this as a hybrid method in which a different scheme is used on each subinterval.

In other cases the matrix A_f may be variable, but of a special form such that the problem $u_t = A_f u_x$ can be solved exactly.

We continue to use the "fast" and "slow" notation even though for such perturbed problems all of the eigenvalues of A may be roughly the same size. Nevertheless in the splitting $A = A_f + A_s$ we assume that A_f has eigenvalues much larger than those of A_s , and so the subproblem $u_t = A_f u_x$ has waves which are fast relative to those occurring in the subproblem $u_t = A_s u_x$.

Example 1.2. A simple example is the scalar problem

$$u_t = (1 + \alpha(x))u_x \quad (1.33)$$

where $|\alpha(x)| \ll 1$ with the splitting

$$A_f = 1, \quad A_s = \alpha(x).$$

Take $k = 2ph$ for some integer p . The operator $\exp(\frac{1}{2}kA_f\partial_x)$ is known exactly:

$$\exp(\frac{1}{2}kA_f\partial_x)u(x, t) = \exp(ph\partial_x)u(x, t) = u(x + ph, t).$$

If Lax-Wendroff is used for the remaining subproblem $u_t = \alpha(x)u_x$ then the method (1.24) becomes

$$\begin{aligned} U_m^* &= U_{m+p}^n \\ U_m^{**} &= LW(\alpha(x), k)U_m^* \\ &= U_m^* + p\alpha(x_m)(U_{m+1}^* - U_{m-1}^*) + p^2\alpha(x_m) \{(\alpha(x_{m+1}) + \alpha(x_m))(U_{m+1}^* - U_m^*) \\ &\quad - (\alpha(x_m) - \alpha(x_{m-1}))(U_m^* - U_{m-1}^*)\} \\ U_m^{n+1} &= U_{m+p}^{**}. \end{aligned}$$

Notice that even though this is a scalar problem, the operators ∂_x and $\alpha(x)\partial_x$ do not commute and so the Strang splitting must be used to achieve a second order method. This sequence is shown schematically in Figure 1.3 for $p = 3$.

Eliminating the intermediate solutions U^* and U^{**} , we can rewrite this as a one-step method:

$$\begin{aligned} U_m^{n+1} &= U_{m+2p}^n + p\alpha(x_{m+p})(U_{m+2p+1}^n - U_{m+2p-1}^n) + p^2\alpha(x_{m+p}) \\ &\quad \times [(\alpha(x_{m+p+1}) + \alpha(x_{m+p}))(U_{m+2p+1}^n - U_{m+2p}^n) \\ &\quad - (\alpha(x_{m+p}) + \alpha(x_{m+p-1}))(U_{m+2p}^n - U_{m+2p-1}^n)]. \end{aligned} \quad (1.34)$$

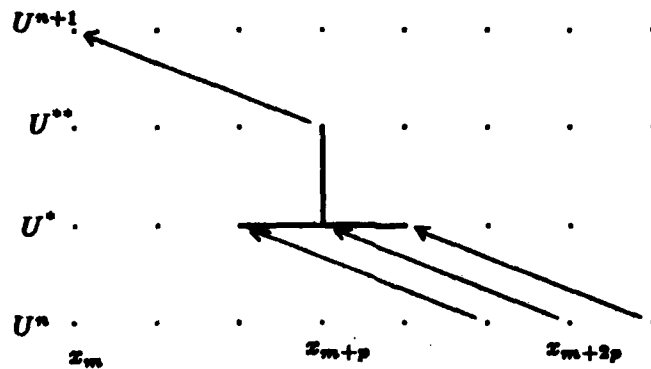


FIG. 1.3. Schematic diagram of the method (1.34) in split form.

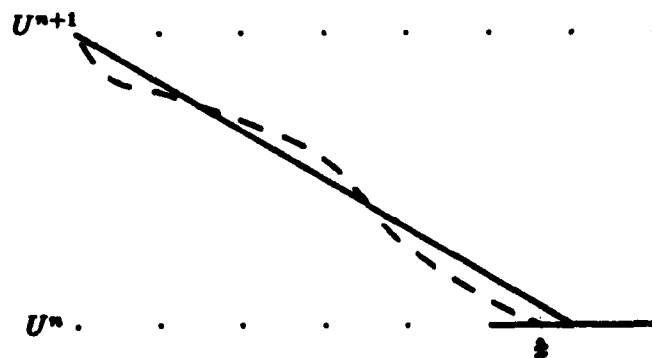


FIG. 1.4 Stencil for the method (1.34) viewed as a one-step method which approximately follows the characteristic of the problem (1.33) (shown, e.g., by the dotted line). Note that values of $\alpha(z)$ are used from near the middle of the interval.

The stencil for this method is shown in Figure 1.4. The value U_m^{n+1} is determined by the values of U^n at x_{m+2p-1} , x_{m+2p} , and x_{m+2p+1} . This scheme can be interpreted as a "skewed Lax-Wendroff" method whose stencil approximately follows the characteristic of the equation, which has slope $-(1 + \alpha(x))$ at each point x . The value of $u(x_m, t_{n+1})$ should thus be equal to the value of $u(\hat{x}, t_n)$ for some point \hat{x} near x_{m+2p} . The exact location depends on the values of $\alpha(x)$ for all x between x_m and \hat{x} . We thus expect such a skewed method to be quite good if $\alpha(x)$ is small. Just how good it is depends on the size of $\alpha(x)$ and also on how rapidly $\alpha(x)$ varies. Note that in the split method (1.34) only values of $\alpha(x)$ near the middle of this interval are used. It turns out that the splitting error for this problem depends on derivatives of $\alpha(x)$. As we will see in Chapter 2, when $\alpha(x)$ is rapidly varying it is most efficient to use small values of p , but the resulting method is still more efficient than using Lax-Wendroff on the unsplit problem.

1.5. The shallow water equations.

Throughout this thesis the shallow water equations will be used as an example to illustrate the general theory being developed. The theory applies to this system in a fairly straightforward but nontrivial manner, and thus studying these equations provides some insight into the issues which arise when splitting methods are applied to other practical problems.

In one space dimension the shallow water equations are

$$\begin{bmatrix} u \\ h \end{bmatrix}_t = - \begin{bmatrix} u & g \\ h & u \end{bmatrix} \begin{bmatrix} u \\ h \end{bmatrix}_x. \quad (1.35)$$

These equations model flow in a channel where g is the gravitational constant, h is the height of the water and u its velocity. This system can be symmetrized by setting $\phi = 2\sqrt{gh}$ to obtain

$$\begin{bmatrix} u \\ \phi \end{bmatrix}_t = - \begin{bmatrix} u & \phi/2 \\ \phi/2 & u \end{bmatrix} \begin{bmatrix} u \\ \phi \end{bmatrix}_x. \quad (1.36)$$

We will make the realistic assumption that u is small compared to ϕ and that variations in ϕ are small compared to some mean value ϕ_0 :

$$\begin{aligned} |\phi - \phi_0| &\leq \epsilon \phi_0, \\ |u| &\leq \epsilon \phi_0 \end{aligned} \quad (1.37)$$

with $\epsilon \ll 1$. Moreover we assume that u_x , ϕ_x and higher derivatives are also $O(\epsilon \phi_0)$. We split the system (1.36) by taking

$$A_f = - \begin{bmatrix} 0 & \phi_0/2 \\ \phi_0/2 & 0 \end{bmatrix}, \quad A_s = - \begin{bmatrix} u & (\phi - \phi_0)/2 \\ (\phi - \phi_0)/2 & u \end{bmatrix}. \quad (1.38)$$

The eigenvalues of A_f are $\pm \phi_0/2$. The exact solution operator $\exp(\frac{1}{2} k A_f \partial_x)$ can be used on the grid provided that

$$\frac{k}{h} = \frac{4p}{\phi_0} \quad (1.39)$$

for some integer $p \geq 1$. The matrix A_s has eigenvalues $u \pm (\phi - \phi_0)/2$ which are smaller by a factor of ϵ than those of A_f .

We will see in Section 2.9 that using a time-split method on this problem reduces the errors by a factor of ϵ (using the same amount of work). The method is stable for the frozen-coefficient problem, as seen in Section 3.5, and in practice is stable for the nonlinear system. The proper specifications of boundary conditions for this problem is discussed in Section 4.5.

1.6. A brief history of splitting methods.

Now that the basic form of the time-split method and a wide variety of possible splittings have been introduced, we pause briefly to review some of the extensive work that has been done on splitting methods. This survey is far from complete, but it provides some historical perspective and references, particularly to the sources which have had the most impact on this thesis.

Splitting methods have been most extensively studied in the context of spatial splittings of multidimensional problems. The first splittings were of implicit methods for solving parabolic problems and were also used as iteration procedures for solving steady-state elliptic problems.

The locally one-dimensional methods were developed primarily by D'Yakonov[11], Marchuk[38], Samarskii[48] and Yanenko[55]. The basic form of such methods has already been indicated in Section 1.2. For the heat equation (1.1) using Crank-Nicolson, for example, the scheme is

$$\begin{aligned} (1 - \frac{1}{2}kD_{+z}D_{-z})U_m^* &= (1 + \frac{1}{2}kD_{+z}D_{-z})U_m^n \\ (1 - \frac{1}{2}kD_{+y}D_{-y})U_m^{n+1} &= (1 + \frac{1}{2}kD_{+y}D_{-y})U_m^*. \end{aligned} \quad (1.40)$$

This clearly fits into the general framework introduced in the Section 1.3 with the splitting

$$A_1(u) = u_{zz}, \quad A_2(u) = u_{yy}. \quad (1.41)$$

The LOD method, however, was not the first such splitting method to be used. In the mid 1950's the *Alternating Direction Implicit (ADI) method* was introduced by Peaceman & Rachford[45] and Douglas[6]. On the equation (1.1) this method, known as the *Peaceman-Rachford method*, has the form

$$\begin{aligned} (1 - \frac{1}{2}kD_{+z}D_{-z})U_m^* &= (1 + \frac{1}{2}kD_{+y}D_{-y})U_m^n \\ (1 - \frac{1}{2}kD_{+y}D_{-y})U_m^{n+1} &= (1 + \frac{1}{2}kD_{+z}D_{-z})U_m^*. \end{aligned} \quad (1.42)$$

The philosophy behind the ADI method is somewhat different from that behind the LOD method. Each equation of (1.42) is, by itself, a first order accurate scheme for solving the original equation (1.1) on a timestep of length $k/2$. The combination provides a second order accurate solution on a step of length k . In some sense it is thus a more natural approach to solving the problem than the LOD method, since the individual equations composing (1.40) do not, by themselves, provide consistent approximations to the original system. On the other hand, the LOD method can be viewed more naturally as a time-split method of the form discussed in Section 1.3, since each equation of (1.40) is a second order accurate approximation to one of the subproblems determined by (1.41) on a timestep of length k .

In fact, the Peaceman-Rachford method can also be viewed as a time-split method of this form, but with a different splitting. Instead of (1.41) suppose we split the operator $A(u) = u_{xx} + u_{yy}$ as

$$\begin{aligned} A_1(u) &= \frac{1}{2}(u_{xx} + u_{yy}) + \frac{1}{8}k(u_{xxxx} - u_{yyyy}), \\ A_2(u) &= \frac{1}{2}(u_{xx} + u_{yy}) - \frac{1}{8}k(u_{xxxx} - u_{yyyy}). \end{aligned} \quad (1.43)$$

Then the equations of (1.42) are second order accurate approximations to $u_t = A_1(u)$ and $u_t = A_2(u)$ on timesteps of length k .

There are many other ways of relating the ADI and LOD methods, see for example Gourlay & Mitchell[25] or Morris & Gourlay[42]. One advantage of viewing ADI as a time-split method is that, in some cases, appropriate boundary conditions for the intermediate solution U^* can then be easily determined using the general procedure described in Chapter 4. This is discussed in Section 5.4.

Numerous variations on the Peaceman-Rachford method have been proposed over the years, for example by Douglas & Rachford[8], Fairweather & Mitchell[19], Douglas & Gunn[7], and D'Yakonov[12]. The last of these is particularly interesting since it is based on approximate factorization, an approach that is currently quite popular in computational fluid dynamics. D'Yakonov's method, which he calls the *method of disintegrating operators*, results from the approximations

$$\begin{aligned} 1 - \frac{1}{2}k(D_{+x}D_{-x} + D_{+y}D_{-y}) &\approx (1 - \frac{1}{2}kD_{+x}D_{-x})(1 - \frac{1}{2}kD_{+y}D_{-y}), \\ 1 + \frac{1}{2}k(D_{+x}D_{-x} + D_{+y}D_{-y}) &\approx (1 + \frac{1}{2}kD_{+x}D_{-x})(1 + \frac{1}{2}kD_{+y}D_{-y}). \end{aligned}$$

Each of these has an error $\frac{1}{4}k^2D_{+x}D_{-x}D_{+y}D_{-y}$. When both approximations are used in (1.16), the resulting error is $O(k^3)$. This leads to the split method

$$\begin{aligned} (1 - \frac{1}{2}kD_{+x}D_{-x})U_m^* &= (1 + \frac{1}{2}kD_{+x}D_{-x})(1 + \frac{1}{2}kD_{+y}D_{-y})U_m^n \\ (1 - \frac{1}{2}kD_{+y}D_{-y})U_m^{n+1} &= U_m^* \end{aligned}$$

which can also be viewed as a time-split method with the splitting

$$\begin{aligned} A_1(u) &= u_{xx} + \frac{1}{2}u_{yy} - \frac{1}{8}ku_{yyyy} \\ A_2(u) &= \frac{1}{2}u_{yy} + \frac{1}{8}ku_{yyyy}. \end{aligned}$$

A great deal of work has gone into the proper specification of intermediate boundary conditions for such splitting methods. See, for example, Lawson & Morris[34], Fairweather & Mitchell[19], or D'Yakonov[13]. General discussions of splitting methods for parabolic problems can be found in many places, including Yanenko[56], Marchuk[40], and Mitchell & Griffiths[41].

As opposed to parabolic problems, many hyperbolic systems of equations are solved using explicit methods. As we saw for the one-dimensional system (1.25), the stability limit frequently allows timesteps that are reasonable from the standpoint of efficiency, and so there is no need to use implicit methods. In more space dimensions, however, the stability limit is often severely reduced. (For example, the stability limit for 2D Lax-Wendroff on (1.3) is $\lambda \max(\rho(A), \rho(B)) \leq 1/\sqrt{8}$.) Strang[49] showed that if the locally one-dimensional method is used on (1.3), then the stability limit is more reasonable (for

Lax-Wendroff, $\lambda \max(\rho(A), \rho(B)) \leq 1$). Thus the LOD method has the use, for explicit methods, of increasing the stability limit.

Implicit methods are often used for certain classes of hyperbolic systems. Recall that the timestep for an explicit method is restricted by the fastest wave speed. For certain systems of equations with disparate wave speeds the physically meaningful solutions contain no fast-wave components, or at least the fast waves have small amplitude compared to the slower waves. For an explicit method applied to such a problem, stability considerations limit the timestep to a value much smaller than that required for accuracy. For this reason, implicit methods are frequently used instead. In more than one space dimension LOD, ADI or approximate factorization methods again prove useful.

Such problems arise, for example, in modeling atmospheric flows. The simplest such system is the two-dimensional shallow water equations. The general solution to these equations includes both fast "gravity waves" and much slower "Rossby waves". In practice, however, the gravity waves contain little energy and, it is thought, have little effect on the weather. Gustafsson[29] has studied an ADI method for this problem.

Another approach to such problems has been taken by Kreiss[32],[33] and Browning, Kasahara & Kreiss[3]. They properly prepare the data so that fast wave components are eliminated. Majda[37] has considered using filters to suppress the fast waves in the same context.

Approximate factorization methods for hyperbolic problems have been studied by Warming & Beam[54], primarily for the Euler equations of gas dynamics and for mixed hyperbolic-parabolic problems such as the Navier-Stokes equations. Again they are dealing with problems where the fast waves have little effect on the solutions of interest.

Another possible approach for such problems is to split the coefficients into fast and slow terms and use an implicit method only on the fast part. This can be quite efficient if, for example, the fast part is sparse. The splitting between implicit and explicit methods can be effected in various ways. For the problem $u_t = Au_x = (A_f + A_s)u_x$, a time-split method of the form (1.29) could be used with $Q_f(k/2)$ implicit and $Q_s(k)$ explicit. Alternatively, one-step methods can be derived that are implicit only in A_f . For example, the trapezoidal formula and leapfrog can be combined into the hybrid method

$$(I - kA_f D_0)U_m^{n+1} = (I + kA_f D_0)U_m^{n-1} + 2kA_s D_0 U_m^n. \quad (1.44)$$

Such methods are called *semi-implicit methods* or *explicit-implicit methods*. Elvius & Sundström[16] have analyzed the two-dimensional analogue of (1.44) for the shallow water equations. Harlow & Amsden[31] have applied a similar method to the Euler equations.

The idea of using different timesteps on various parts of the system has been used in one form or another by several authors, including Engquist, Gustafsson & Vreeburg[17], Gadd[20], and Turkel & Zwas[52].

Many nonlinear hyperbolic systems have solutions involving *shock waves* -- discontinuous solutions which can arise even from smooth initial data. For such problems a wide variety of special methods have been devised. Often these methods are directly applicable only in one space dimension. For higher dimensional problems, LOD splittings are again frequently used. Since the solutions are not smooth, splittings are more difficult to analyze in this context. Crandall & Majda[5] have proved that the splittings (1.19) and (1.21) do give convergent methods when applied to scalar conservation laws.

For mixed problems such as the Navier-Stokes equations the time-split method has been used to split between hyperbolic and parabolic parts. Abarbanel & Gottlieb[1] split the full three-dimensional Navier-Stokes equations into nine pieces—the hyperbolic and parabolic terms in each space dimension and three cross-derivative terms. They then use the time-split method to derive an explicit method with good stability properties. MacCormack[35][36], Strikwerda[50], and Dwoyer & Thames[10] have studied similar methods.

Approximate factorization methods for this problem have been proposed by Beam and Warming[2][54]. This approach appears to have certain advantages in steady state calculations. The numerical steady state is independent of the timestep k used to compute it, and the calculations can be performed in an "increment form" that is computationally efficient.

Convection-diffusion equations similar to (1.4) arise in transport problems that include diffusion, for example in multi-phase miscible flow or in modeling heat flow in a moving material. When the problem is convection dominated ($\nu \ll |c|$ in (1.4)), the propagation of sharp fronts is often of interest. These are difficult to handle numerically. It is often advantageous to again split between the hyperbolic and parabolic parts and handle the hyperbolic part using characteristics. This is studied in Section 5.3. Similar methods have been proposed by MacCormack[36] and Douglas & Russell[9]. Another possibility is to use the finite element method for the parabolic part[9][15][47][53].

1.7. Outline and summary of results.

There are three main issues to be dealt with when considering the use of a time-split method for any differential equation. These may be summarized as *efficiency*, *stability*, and the proper choice of *boundary conditions*. These are, of course, major issues in the choice of any finite difference scheme, but the use of time-split methods introduces new complications into each area.

Efficiency. The first quantity to compute in the analysis of any finite difference scheme is its truncation error. In Section 2.2 we show that for the time-split method the truncation error is simply the sum of the splitting error and the truncation errors for the approximate solution operators Q_1 and Q_2 (plus higher order terms). The splitting error thus plays a fundamental role and techniques for computing this error for general splittings are discussed in Section 2.3.

In comparing methods, however, it is not in general sufficient to compare their truncation errors, since one scheme may require much more computation than another. This is particularly true where time-split methods are concerned. Instead we must compare some measure of the efficiency of the methods such as the amount of work required to achieve a given accuracy.

Since split methods generally involve the conjunction of several different numerical methods, there may be several free parameters, such as stepsizes, to be chosen. For the method (1.30), for example, we must choose both k/h and m . We can essentially choose the mesh ratios for the two time scales independently. As we will see in Section 2.5, the optimal choice depends on the size of the splitting error and is not always obvious a priori. In particular, the optimal mesh ratio k/h is often far from the stability limit of the method used on the slow problem.

Stability. For some time-split methods applied to certain problems, the operator $Q_1(k)Q_2(k)$ is stable whenever the operators $Q_1(k)$ and $Q_2(k)$ are each stable operators on their own. Unfortunately, this is not true in general; the product of two stable operators may be unstable. An example of this is given in Chapter 3.

Of course the stability of $Q_1(k)Q_2(k)$ can always be determined directly by eliminating all intermediate variables and viewing the split method as a one-step method. However, the resulting method is generally quite complicated, making direct analysis difficult. For this reason it is useful to identify special classes of problems for which the individual stability of $Q_1(k)$ and $Q_2(k)$ does guarantee the stability of $Q_1(k)Q_2(k)$. Several such classes of hyperbolic splittings are identified in Chapter 3.

Boundary conditions. All practical calculations are performed on finite domains. If *periodic boundary conditions* are used (e.g., $u(0, t) = u(1, t) \quad \forall t$ on the strip $0 \leq x \leq 1$), then the same finite difference scheme can be used at all points, simply by wrapping around at the boundaries. Otherwise, one or more points at each boundary will have to be determined in some alternative manner (unless a one-sided scheme is used). Some boundary values will be provided as part of the problem, but frequently finite difference approximations require more boundary conditions than the original differential equation. The remaining boundary values must be determined by some other procedure. A variety of techniques are used for this purpose, depending on the context. The easiest approach is often to extrapolate the interior solution at time t_{n+1} out to the boundary. Alternatively, one-sided (or lopsided) finite difference schemes can be used to compute the solution at points on (or near) the boundary. At some boundaries other desirable properties of the solutions, such as nonreflection of outgoing waves, may be used to determine the proper boundary values.

For time-split methods the choice of boundary values is complicated by the need to supply boundary data for the intermediate solutions, such as U^* . These solutions are obtained not by solving the original differential equation but rather by solving one of the subproblems. Because of this, appropriate boundary data for the intermediate solutions is never available directly. Extrapolation from the interior can still be used, but is generally undesirable both for reasons of stability and accuracy.

The generation of boundary data for the intermediate solutions is discussed in Chapter 4. We describe a general procedure for transforming given boundary data for the original equation into appropriate data for the intermediate solutions. This procedure is based on the following idea. We introduce a new function u^* which satisfies the subproblem that is actually being solved in the first step of the splitting. We then expand the desired boundary value for u^* in a Taylor series about the initial time t_n at which $u^* \equiv u$. Using the differential equations for u^* and u we then reexpress this as a series expansion involving only the function u and its time derivatives along the boundary. This can then be evaluated from the given boundary conditions for u .

Each of the next three chapters is devoted to one of these issues. The emphasis is on splittings of hyperbolic problems into subproblems with disparate wave speeds, as discussed in Section 1.3. However, many of the techniques used are also applicable to other splittings of the form $u_t = A_1(u) + A_2(u)$. Whenever possible, the discussion is in terms of the more general splitting to facilitate application to other problems. Hyperbolic splittings are always used as concrete examples in these chapters, and most of the specific

results are for such problems. In particular, the one-dimensional shallow water equations are frequently used as an example.

In Chapter 5 we discuss several other applications of the time-split method using the theory developed in previous chapters. We first consider applications of the time-split method to hyperbolic problems in two space dimensions. The main intent is still to split between different wave speeds, but in conjunction with this spatial splittings may also be used.

Finally we consider two applications of the theory of time-split methods to non-hyperbolic splittings. In Section 5.3 the simple convection-diffusion equation (1.4) is split and solved as a perturbed problem with a skewed Crank-Nicolson method analogous to the skewed Lax-Wendroff method (1.34). The efficiency of this method can be analyzed using the techniques of Chapter 2. Intermediate boundary conditions at the inflow boundary can be specified using the procedure of Chapter 4. When the diffusive parameter ν in (1.4) is small the equation becomes a singular perturbation equation with a boundary layer at the outflow boundary that causes additional difficulties.

In Section 5.4 the Peaceman-Rachford method (1.42) is viewed as a time-split method with the splitting (1.43). For a rectangular region the boundary condition procedure of Chapter 4 can be used to derive appropriate boundary conditions for the intermediate solution U^* . These are seen to agree with the classical boundary conditions of Fairweather and Mitchell[19].

Chapters 2-4 are essentially independent of one another and can be read in any order. The sections of Chapter 5, which deal with other applications, are disjoint from one another, but build upon the results of the previous chapters, particularly Chapters 2 and 4.

2. Accuracy and efficiency

2.1. Introduction.

This chapter begins with a computation of the truncation error for a general time-split method. Neglecting higher order terms, this is simply the sum of the error committed in splitting the exact solution operator (the *splitting error*) and the truncation errors of the schemes used for the subproblems.

In Section 2.3 we present general expressions for the splitting error in both the first order splitting (1.19) and the Strang splitting (1.21). The splitting error is explicitly computed for some model problems, including the one-dimensional shallow water equations.

For the type of splitting with which this thesis is most concerned, namely where $\|A_2(u)\| \leq \epsilon \|A_1(u)\|$ with $\epsilon \ll 1$, the error in the Strang splitting is seen to be $O(\epsilon k^3)$. A simple modification of this splitting is proposed with $O(\epsilon^2 k^3 + \epsilon k^4)$ splitting error.

Once we are able to compute the splitting error for specific problems, we can analyze the efficiency of the split method relative to unsplit methods. It turns out that the size of the splitting error greatly affects what size timesteps should be used in the split method and what increase in efficiency can then be expected. This analysis is presented in Section 2.5 and continues in Section 2.6 where phase errors are computed.

In Section 2.7 these results are interpreted for a block triangular system of the form considered in Example 1.1. For this problem (and also for more general partitioned systems) the splitting error can be reduced by the use of a simple change of variables. This is discussed in Section 2.8.

In Section 2.9 the one-dimensional shallow water equations are studied. The theory developed in Section 2.5 is confirmed numerically for this system.

2.2. Truncation error of the time-split method.

In order to compute the truncation error for the time-split method we first introduce the *truncation error operators* $E_j(k)$ for the approximate solution operators $Q_j(k)$. These are defined by

$$E_j(k) = Q_j(k) - S_j(k), \quad k = 1, 2.$$

We will assume throughout that Q_1 and Q_2 are at least second order accurate. Then $E_j(k)u$ is $O(k^3)$ for smooth u . For shorthand we sometimes write $E_j(k) = O(k^3)$. Similarly, we introduce the splitting error operator $E_{\text{split}}(k)$ defined by

$$E_{\text{split}}(k) = S_1(k/2)S_2(k)S_1(k/2) - S(k).$$

This is also $O(k^3)$ for smooth u .

The truncation error operator for the time-split method is

$$E^{TSM}(k) = Q_1(k/2)Q_2(k)Q_1(k/2) - S(k).$$

If the operators $A_1(u)$ and $A_2(u)$ are linear, this can be easily computed in terms of E_1 , E_2 and E_{split} using the fact that $S_j(k) = I + O(k)$ and $Q_j(k) = I + O(k)$:

$$\begin{aligned} E^{TSM}(k) &= (S_1(k/2) + E_1(k/2)) (S_2(k) + E_2(k)) (S_1(k/2) + E_1(k/2)) - S(k) \\ &= S_1(k/2)S_2(k)S_1(k/2) - S(k) + 2E_1(k/2) + E_2(k) + O(k^4) \\ &= E_{split}(k) + 2E_1(k/2) + E_2(k) + O(k^4). \end{aligned} \quad (2.1)$$

If the operators $A_1(u)$ and $A_2(u)$ are nonlinear, then the Q , S , and E operators will also be nonlinear. More care must then be used in deriving $E^{TSM}(k)$, but the $O(k^3)$ term of the result is exactly the same as above, and the expression (2.1) holds in general.

The truncation error for the time-split method is thus seen to be essentially the sum of the splitting error for the problem and the truncation errors for the finite difference operators. This allows us to easily compute the accuracy and investigate the efficiency of the time-split method relative to unsplit schemes. This will be done in Section 2.5. First we must be able to compute the splitting error $E_{split}(k)$.

2.3. The splitting error.

We will first prove the assertions made in Chapter 1 regarding the accuracy of the splittings (1.19) and (1.21) when applied to the solution operator for a general equation of the form

$$u_t = A(u, t). \quad (2.2)$$

The operator A may also depend on spatial variables, but this dependence will not be explicitly shown. The proofs are completely independent of the number of space dimensions.

We denote by $A'(u, t)$ the total time derivative of A assuming u satisfies (2.2). This is given by

$$\begin{aligned} A'(u, t) &= A_t(u, t) + A_u(u, t)u_t \\ &= A_t(u, t) + A_u(u, t)A(u, t). \end{aligned} \quad (2.3)$$

In the latter form this depends only on u at the time t and does not depend explicitly on u_t . This is crucial in the proofs that follow, where we will be switching between solving different differential equations, which means that time derivatives of u become ambiguous.

A few words should be said about the quantity $A_u(u, t)$. The vector-valued function A generally depends both on u and on one or more spatial derivatives of u . In one space dimension, for example, we could write $A = A(u, u_x, u_{xx}, \dots, t)$. The derivative A_u is then defined as

$$A_u = \frac{\partial A}{\partial u} + \frac{\partial A}{\partial u_x} \partial_x + \frac{\partial A}{\partial u_{xx}} \partial_x^2 + \dots \quad (2.4)$$

where $\partial A/\partial u$, $\partial A/\partial u_x$, etc. are ordinary Jacobian matrices with respect to the appropriate vectors u , u_x , etc. More will be said about evaluating these expressions later in this section.

Now suppose that A is split as $A(u, t) = A_1(u, t) + A_2(u, t)$. One consequence of (2.3) is that $A'(u, t) \neq A'_1(u, t) + A'_2(u, t)$. This is because $A'_j(u, t)$ is the time-derivative of A_j assuming u satisfies $u_t = A_j(u, t)$ rather than (2.2). We find instead that

$$\begin{aligned} A' &= A_t + A_u A \\ &= (A_{1t} + A_{2t}) + (A_{1u} + A_{2u})(A_1 + A_2) \\ &= (A_{1t} + A_{1u} A_1) + (A_{2t} + A_{2u} A_2) + A_{1u} A_2 + A_{2u} A_1 \\ &= A'_1 + A'_2 + A_{1u} A_2 + A_{2u} A_1. \end{aligned} \tag{2.5}$$

We are now ready to prove the results indicated earlier, beginning with a standard proof that the splitting (1.19) is first order accurate (i.e., that the local error is $O(k^2)$).

THEOREM 2.1. *Suppose that $u(t_0)$ is a C^∞ function of all spatial variables and that A , A_1 , and A_2 are smooth functions of u and t related by (1.17). Then the corresponding solution operators S , S_1 , and S_2 satisfy*

$$\begin{aligned} S_2(t_0 + k, t_0)S_1(t_0 + k, t_0)u(t_0) - S(t_0 + k, t_0)u(t_0) \\ = \frac{1}{2}k^2[A_{2u}(u(t_0), t_0)A_1(u(t_0), t_0) - A_{1u}(u(t_0), t_0)A_2(u(t_0), t_0))] + O(k^3) \end{aligned} \tag{2.6}$$

as $k \rightarrow 0$.

Proof. We begin by computing $S(t_0 + k, t_0)u(t_0)$. If u satisfies (2.2) then this is simply $u(t_0 + k)$ and expanding in a Taylor series gives

$$\begin{aligned} S(t_0 + k, t_0)u(t_0) &= u(t_0) + ku_t(t_0) + \frac{1}{2}k^2u_{tt}(t_0) + O(k^3) \\ &= u(t_0) + kA + \frac{1}{2}k^2A' + O(k^3). \end{aligned} \tag{2.7}$$

Here and below, when no arguments are shown for A we mean $A(u(t_0), t_0)$ (similarly for A_1 and A_2). We now compute the solution using the split operator. After the first step we have

$$S_1(t_0 + k, t_0)u(t_0) = u(t_0) + kA_1 + \frac{1}{2}k^2A'_1 + O(k^3).$$

Set $u^* = S_1(t_0 + k, t_0)u(t_0)$. Then

$$\begin{aligned} S_2(t_0 + k, t_0)u^* &= u^* + kA_2(u^*, t_0) + \frac{1}{2}k^2A'_2(u^*, t_0) + O(k^3) \\ &= u^* + k[A_2(u(t_0), t_0) + A_{2u}(u(t_0), t_0)(u^* - u(t_0)) + O(k^2)] \\ &\quad + k^2[A'_2(u(t_0), t_0) + O(k)] + O(k^3) \\ &= [u(t_0) + kA_1 + \frac{1}{2}k^2A'_1 + O(k^3)] \\ &\quad + k[A_2 + A_{2u}(kA_1 + O(k^2)) + O(k^2)] \\ &\quad + \frac{1}{2}k^2[A'_2 + O(k)] + O(k^3) \\ &= u(t_0) + k(A_1 + A_2) + \frac{1}{2}k^2(A'_1 + 2A_{2u}A_1 + A'_2) + O(k^3). \end{aligned}$$

Using (2.5) and (2.7) we find that

$$\begin{aligned} S_2(t_0 + k, t_0)S_1(t_0 + k, t_0)u(t_0) - S(t_0 + k, t_0)u(t_0) \\ = \frac{1}{2}k^2(A_{2u}A_1 - A_{1u}A_2) + O(k^3). \end{aligned}$$

This proves the theorem. ■

Example 2.1. The formula (2.6) can be used to compute the $O(k^2)$ term of the splitting error for any particular problem. Consider, for example, the problems

(a) $u_t = Au_x + Bu_y$ with $A_1(u) = Au_x$, $A_2(u) = Bu_y$ and A and B constant. The solution operators are simply exponentials, so the splitting error can be computed directly as in (1.20). We get the same result by (2.6) since $A_{1u} = A\partial_x$ and $A_{2u} = B\partial_y$.

(b) $u_t = (1 + \alpha(x))u_x$ with $A_1(u) = u_x$ and $A_2(u) = \alpha(x)u_x$. From (2.6) the splitting error is

$$\begin{aligned} & \frac{1}{2}k^2[\alpha(x)\partial_x u_x - \partial_x(\alpha(x)u_x)] + O(k^3) \\ & = -\frac{1}{2}k^2\alpha'(x)u_x + O(k^3). \end{aligned}$$

For this problem the solution operators are again exponentials, $S_1(k) = \exp(k\partial_x)$ and $S_2(k) = \exp(k\alpha(x)\partial_x)$, so this can also be checked directly.

(c) $u_t = (c+u)u_x$ with c constant and u scalar. Take $A_1(u) = cu_x$ and $A_2(u) = uu_x$. Then $A_{1u}A_2 = A_{2u}A_1 = c(u_x^2 + uu_{xx})$ and the $O(k^2)$ term in the splitting error is zero. In fact, for this problem the splitting error is identically zero. This is intuitively clear since solving the subproblem $u_t = cu_x$ simply translates the solution in x . The remaining subproblem $u_t = uu_x$ does not depend explicitly on x , and so solving this problem and shifting the result is equivalent to solving the original problem.

Note that if u is a vector this is no longer true, since in general different eigencomponents of u propagate at different speeds and hence move relative to one another. The splitting error for a system of equations $u_t = [A_f + A_s(u)]u_x$ will be computed later in this section.

The next theorem asserts that the Strang splitting (1.21) is in general second order accurate.

THEOREM 2.2. (Strang[49]) Suppose that $u(t_0)$ is a C^∞ function of all spatial variables and that A , A_1 , and A_2 are smooth functions of u and t related by (1.17). Then the corresponding solution operators S , S_1 , and S_2 satisfy

$$S_1(t_0 + k, t_0 + \frac{1}{2}k)S_2(t_0 + k, t_0)S_1(t_0 + \frac{1}{2}k, t_0)u(t_0) - S(t_0 + k, t_0)u(t_0) = O(k^3)$$

as $k \rightarrow 0$.

Proof. Proceeding as in the proof of Theorem 2.1,

$$S_1(t_0 + \frac{1}{2}k, t_0)u(t_0) = u(t_0) + \frac{1}{2}kA_1 + \frac{1}{8}k^2A_1' + O(k^3).$$

Again denote this by u^* . Then

$$\begin{aligned} S_2(t_0 + k, t_0)u^* &= u^* + kA_2(u^*, t_0) + \frac{1}{2}k^2A_2'(u^*, t_0) + O(k^3) \\ &= u(t_0) + k(\frac{1}{2}A_1 + A_2) + \frac{1}{2}k^2(\frac{1}{4}A_1' + A_{2u}A_1 + A_2') + O(k^3). \end{aligned}$$

Call this quantity u^{**} . Then

$$S_1(t_0 + k, t_0 + \frac{1}{2}k)u^{**} = u^{**} + \frac{1}{2}kA_1(u^{**}, t_0 + \frac{1}{2}k) + \frac{1}{8}k^2A_1'(u^{**}, t_0 + \frac{1}{2}k) + O(k^3).$$

Expanding A_1 and A_1' in both u and t about $(u(t_0), t_0)$ and collecting terms, we find that

$$\begin{aligned} & S_1(t_0 + k, t_0 + \frac{1}{2}k)S_2(t_0 + k, t_0)S_1(t_0 + \frac{1}{2}k, t_0) \\ & = u(t_0) + k(A_1 + A_2) + \frac{1}{2}k^2[\frac{1}{2}A_1' + \frac{1}{2}(A_{1t} + A_{1u}A_1) + A_2' \\ & \quad + A_{1u}A_2 + A_{2u}A_1] + O(k^3) \\ & = u(t_0) + kA + \frac{1}{2}k^2A' + O(k^3) \end{aligned}$$

in view of (2.3) and (2.5). Comparing this with (2.7) shows that the error is indeed $O(k^3)$. ■

Keeping the $O(k^3)$ term everywhere in the proof would have given us a formula analogous to (2.6) for the k^3 term of the error. In general this is quite complicated. For the relatively simple autonomous case where A is a function of u alone, the splitting error is found to be

$$-\frac{1}{8}k^3\left[\frac{1}{4}(A_{1u}A_1)_u A_2 - \frac{1}{2}(A_{1u}A_2)_u A_1 + \frac{1}{4}(A_{2u}A_1)_u A_1 - \frac{1}{2}(A_{2u}A_2)_u A_1 + (A_{2u}A_1)_u A_2 - \frac{1}{2}(A_{1u}A_2)_u A_2\right] + O(k^4). \quad (2.8)$$

Example 2.2. The errors in the Strang splitting for the problems considered in Example 2.1 are relatively easy to compute:

(a) $u_t = Au_x + Bu_y$ with A and B constant. By expanding the exponential solution operators the splitting error is seen to be

$$-\frac{1}{8}k^3\left[\left(\frac{1}{4}A^2B - \frac{1}{2}ABA + \frac{1}{4}BA^2\right)\partial_x^2\partial_y - \left(\frac{1}{2}B^2A - BAB + \frac{1}{2}AB^2\right)\partial_x\partial_y^2\right]u(t_0) + O(k^4). \quad (2.9)$$

The splitting error is zero only if A and B commute.

(b) $u_t = (1 + \alpha(x))u_x$. Again expanding the exponential solution operators shows that the splitting error is

$$-\frac{1}{12}k^3\left[\left(\frac{1}{2} + \alpha(x)\right)\alpha''(x) - (\alpha'(x))^2\right]u_x(t_0) + O(k^4).$$

A higher order splitting. The fact that the Strang splitting is second order accurate can be seen more directly by viewing the Strang splitting, as in (1.22), as two applications of the first order splitting with S_1 and S_2 applied in the opposite order in the second application. By Theorem 2.1 the truncation error in the first step is

$$\frac{1}{8}k^2(A_{2u}(u(t_0), t_0)A_1(u(t_0), t_0) - A_{1u}(u(t_0), t_0)A_2(u(t_0), t_0))) + O(k^3) \quad (2.10)$$

and in the second step:

$$\frac{1}{8}k^2(A_{1u}(u^*, t_0 + \frac{1}{2}k)A_2(u^*, t_0 + \frac{1}{2}k) - A_{2u}(u^*, t_0 + \frac{1}{2}k)A_1(u^*, t_0 + \frac{1}{2}k)) + O(k^3). \quad (2.11)$$

The full-step truncation error can be shown to be simply the sum of (2.10) and (2.11) plus $O(k^3)$ terms. Expanding (2.11) about $(u(t_0), t_0)$ and adding (2.10), the $O(k^2)$ terms cancel and hence the Strang splitting is $O(k^3)$ accurate. This cancellation occurs because the $O(k^2)$ term of (2.6) is skew-symmetric in the variables A_1 and A_2 .

For the type of problem we are considering here, where $\|A_2(u)\| \leq \epsilon\|A_1(u)\|$ and similarly for their derivatives, a similar trick can be applied to the Strang splitting to increase the accuracy even further. The $O(k^3)$ term of the splitting error (2.8) is generally dominated by the first three terms, which contain two factors $A_1(u)$ and a single $A_2(u)$. The other terms are smaller by a factor of ϵ and hence the Strang splitting is $O(\epsilon k^3)$ accurate. But now suppose that on every third step we reverse S_1 and S_2 in the Strang splitting, so that the approximate solution operator over three timesteps becomes

$$S(3k) \approx S_2(k/2)S_1(k)S_2(k/2)S_1(k/2)S_2(k)S_1(k)S_2(k)S_1(k/2).$$

Then the $O(k^3)$ term of the error is simply twice the expression (2.8) plus the expression (2.8) with $A_1(u)$ and $A_2(u)$ interchanged. The $O(\epsilon k^3)$ terms then cancel leaving only the $O(\epsilon^2 k^3)$ terms, plus of course the higher order terms, which are $O(\epsilon k^4)$. Unfortunately in practice these higher order terms often dominate, especially when large timesteps k are used. Numerical results indicate that this modification has little practical value except when a very fine mesh is used. This idea will not be developed any further here.

2.4. Computing the splitting error for quasilinear systems.

The expressions (2.6) and (2.8) for the splitting errors look deceptively simple. Evaluating them for practical problems is actually quite a chore, mainly because of the matrix derivatives which occur. We will now discuss the proper way to evaluate such expressions and give several examples. We are particularly interested in the situation where $A(u) = A(u)u_x$.

We begin by discussing derivatives of matrices. If $A(u) \in \mathbb{R}^{r \times r}$ is a matrix valued function of a vector $u \in \mathbb{R}^r$, then its derivative $A_u(u) \in \mathbb{R}^{r \times r \times r}$ will be a three-tensor. It is convenient to think of this as an array of matrices:

$$A_u(u) = \left[\frac{\partial A}{\partial u_1}, \frac{\partial A}{\partial u_2}, \dots, \frac{\partial A}{\partial u_r} \right]. \quad (2.12)$$

A tensor multiplied by a vector gives a matrix. There are two ways to perform this tensor-vector multiplication and it is important to distinguish between them, since they give different results.

If B is the tensor

$$B = [B_1, B_2, \dots, B_r]$$

where $B_j \in \mathbb{R}^{r \times r}$, and if $v \in \mathbb{R}^r$, then the first type of multiplication, denoted simply by Bv , is obtained by taking a linear combination of the matrices B_j :

$$Bv = B_1v_1 + B_2v_2 + \dots + B_rv_r \in \mathbb{R}^{r \times r}$$

where $v = (v_1, \dots, v_r)^T$. The second type of multiplication will be denoted by $B \otimes v$. This product is given by the matrix whose j th column is the vector B_jv :

$$B \otimes v = \left[B_1v \mid B_2v \mid \dots \mid B_rv \right] \in \mathbb{R}^{r \times r}.$$

It is easy to verify that if $w \in \mathbb{R}^r$ is some other vector then

$$(B \otimes v)w = (Bw)v \in \mathbb{R}^r. \quad (2.13)$$

Both forms of multiplication play a role in differentiation. Consider the vector-valued function $f(u) = A(u)w$, where w is a constant vector and u is itself a function of x . Then differentiating f with respect to x gives the vector

$$\begin{aligned} \frac{\partial}{\partial x} f(u) &= \left(\frac{\partial}{\partial x} A(u) \right) w \\ &= (A_u(u)u_x)w \in \mathbb{R}^r \end{aligned}$$

where $A_u(u)$ is the tensor (2.12) and the multiplication is of the first type. On the other hand, differentiating with respect to u gives the matrix

$$\frac{\partial}{\partial u} f(u) = A_u(u) \otimes w \in \mathbb{R}^{r \times r}. \quad (2.14)$$

This can be verified by directly computing the Jacobian matrix corresponding to

$$\begin{aligned} f(u) &= \Lambda(u)w \\ &= \begin{bmatrix} \sum_{j=1}^r a_{1j}(u)w_j \\ \vdots \\ \sum_{j=1}^r a_{rj}(u)w_j \end{bmatrix}. \end{aligned}$$

Also note that if B is a constant tensor, then

$$\frac{\partial}{\partial u}(Buu) = Bu + B \otimes u.$$

Now let $\mathcal{A}(u) = A(u)u_x$ and suppose, for simplicity, that $A_u(u)$ is constant, so that $A_{uu}(u) = 0$. (Otherwise this would be a four-tensor.) We then find that

$$\begin{aligned} A_u(u) &= A_u \otimes u_x + A \partial_x \\ A_{uu}(u) &= 2A_u \otimes \partial_x. \end{aligned}$$

Example 2.3. Consider the problem $u_t = [A_f + A_s(u)]u_x$ with A_f constant and A_s a function of u alone. Take $\mathcal{A}_1(u) = A_f u$ and $\mathcal{A}_2(u) = A_s(u)u_x$. Using (2.6) we can compute the $O(k^2)$ term of the splitting error for the first order splitting (1.19):

$$\begin{aligned} & \frac{1}{2}k^2(\mathcal{A}_{2u}\mathcal{A}_1(u) - \mathcal{A}_{1u}\mathcal{A}_2(u)) \\ &= \frac{1}{2}k[(A_{su} \otimes u_x + A_s \partial_x)A_f u_x - A_f \partial_x(A_s u_x)] \\ &= \frac{1}{2}k^2[(A_{su} \otimes u_x A_f u_x + A_s A_f u_{xx} - (A_f(A_{su} u_x)u_x + A_f A_s u_{xx})] \\ &= \frac{1}{2}k^2[(A_{su} A_f u_x - A_f A_{su} u_x)u_x + (A_s A_f - A_f A_s)u_{xx}]. \end{aligned} \tag{2.15}$$

To obtain the last line we have used (2.13) to rewrite $A_{su} \otimes u_x A_f u_x$ as $A_{su} A_f u_x u_x$. Note that in order for (2.15) to be zero A_f must commute both with A_s and with A_{su} .

As a concrete example, consider the one-dimensional shallow water equations (1.36) with the splitting (1.38). For this system, the tensor A_{su} is given by

$$A_{su}(u) = - \begin{bmatrix} 1 & 0 & 0 & \frac{1}{2} \\ 0 & 1 & \frac{1}{2} & 0 \end{bmatrix}.$$

We compute that

$$\begin{aligned} (A_{su}(A_f u_x))u_x &= -\frac{\phi_0}{2} \begin{bmatrix} \phi_x \\ u_x \end{bmatrix} \begin{bmatrix} u_x \\ \phi_x \end{bmatrix} \\ &= \frac{\phi_0}{2} \begin{bmatrix} \phi_x & \frac{1}{2}u_x \\ \frac{1}{2}u_x & \phi_x \end{bmatrix} \begin{bmatrix} u_x \\ \phi_x \end{bmatrix} \\ &= \frac{\phi_0}{2} \begin{bmatrix} \frac{3}{2}u_x \phi_x \\ \frac{1}{2}u_x^2 + \phi_x^2 \end{bmatrix}. \end{aligned}$$

Similarly,

$$A_f(A_{\phi u} u_x) u_x = \frac{\phi_0}{2} \left[\frac{1}{2} u_x^2 + \phi_x^2 \right],$$

$$A_{\phi} A_f u_{xx} = A_f A_{\phi} u_{xx} = \frac{\phi_0}{2} \left[u \phi_{xx} + \frac{1}{2} (\phi - \phi_0) u_{xx} \right],$$

and hence

$$\frac{1}{2} k^2 (A_{2u} A_1(u) - A_{1u} A_2(u)) = \frac{1}{2} \phi_0 k^2 \left(\frac{1}{2} u_x^2 - \frac{3}{2} u_x \phi_x + \phi_x^2 \right) \begin{bmatrix} -1 \\ 1 \end{bmatrix}.$$

Example 2.4. The error in the Strang splitting can be computed analogously. In interpreting the expression (2.8) it is important to recall (2.14), which indicates that, for example,

$$(A_{1u} A_2(u))_u A_1(u) = (A_{1uu} \otimes A_2(u) + A_{1u} A_{2u}) A_1(u)$$

$$= A_{1uu} A_1(u) A_2(u) + A_{1u} A_{2u} A_1(u).$$

Evaluating (2.8) for the shallow water equations requires a tedious calculation. In view of (1.37), the dominant terms of (2.8) are the terms

$$\frac{1}{8} k^3 \left(-\frac{1}{2} (A_{1u} A_1(u))_u A_2(u) + \frac{1}{2} (A_{1u} A_2(u))_u A_1(u) - \frac{1}{2} (A_{2u} A_1(u))_u A_1(u) \right).$$

This turns out to be

$$\frac{\phi_0^2 k^3}{96} \left[\phi_x \phi_{xx} - u_x u_{xx} \right] = O(\epsilon^2 \phi_0^4 k^3). \quad (2.16)$$

All other terms in the splitting error are $O(\epsilon^3 \phi_0^4 k^3 + \epsilon^2 \phi_0^4 k^4)$.

2.5. Efficiency analysis for the time-split method on hyperbolic problems.

The remainder of this chapter deals only with hyperbolic problems of the sort described in Section 1.4, although the same type of analysis can easily be applied to other problems. For definiteness we also restrict our attention to the Lax-Wendroff method. Other schemes can be analyzed in the same manner. In Section 5.3 a similar analysis will be performed for the Crank-Nicolson method on a convection-diffusion problem.

For the constant coefficient equation

$$u_t = Au_x = (A_f + A_s)u_x \quad (2.17)$$

we wish to compare the unsplit method

$$LW(A, k)$$

with the time-split method (1.29) using

$$Q_s(k) = LW(A_s, k). \quad (2.18a)$$

For $Q_f(k/2)$ we consider both

$$Q_f(k/2) = \exp(\frac{1}{2}kA_f\partial_x) \quad (2.18b)$$

and

$$Q_f(k/2) = (LW(A_f, k/m))^{m/2} \quad (2.18c)$$

for some even integer m . The split scheme defined by (2.18a,c) might be used if A_f were sparse relative to A_s , while (2.18a,b) would be appropriate for perturbed problems where $\exp(\frac{1}{2}kA_f\partial_x)$ is known exactly.

In each case we assume that $\lambda = k/h$ is fixed as $k \rightarrow 0$. In comparing the split methods with the unsplit method, it does not suffice to compare the local truncation errors. For fixed k and h the two methods may take quite different amounts of work to implement. Furthermore the optimal mesh ratio may be different for the two schemes.

Instead we compare the amount of work required to compute a solution with an error bounded by τ , say. Specifically, we consider the x -interval $[0, 1]$ and determine the amount of work required to compute solutions at time $t = 1$ with error no greater than τ . Strang[49] takes an equivalent approach and compares the accuracy obtained with a fixed amount of work. In comparing numerical results it is convenient to take yet another approach and simply normalize the resulting errors by multiplying by some measure of the work required to obtain them. This will be done in later sections.

For theoretical analysis the approach taken here seems to be the most natural. It determines the optimal mesh ratio and also provides (rough) expressions for the values of k and h which must be used to achieve a given accuracy.

For this analysis we will assume, as does Strang, that the variables have been normalized (or the norm appropriately chosen) so that

$$\rho(A) \approx \|A\| = a$$

where $\rho(A)$ is the spectral radius of A . This means in particular that $\|A^3\| \approx a^3$. For the splitting indicated in (2.17) we suppose that

$$\|A_f\| \approx a, \quad \|A_s\| \approx \epsilon a \quad (2.19)$$

with the spectral radii again comparable to the norms and $\epsilon \ll 1$. Set $b = \epsilon a$. Also suppose that $\|u_{xxx}\| \approx 1$. This is for convenience only, since it removes one common factor from all of the bounds below.

Efficiency of the unsplit method. We will first analyze the unsplit Lax-Wendroff method $LW(A, k)$. Suppose that W is the work required to compute $LW(A, k)U_m^n$ at a single point x_m . Then the work required to advance the solution on a unit x -interval by one unit of time is $W/kh = \lambda W/k^2$ if $k = \lambda h$. The truncation error for the Lax-Wendroff method is given by (1.13),

$$E^{LW}(k)u = -\frac{1}{6}k(k^2\Lambda^3 - h^2\Lambda)u_{xxx} + O(k^4). \quad (2.20)$$

Applying this roughly $1/k$ times gets us to time $t = 1$ and

$$\begin{aligned} (LW(A, k))^{1/k} &= (\exp(k(A_f + A_s)\partial_x) + E^{LW}(k))^{1/k} \\ &= \exp(A\partial_x) + \frac{1}{k}[E^{LW}(k) + O(k^4)] + O(k^4). \end{aligned}$$

The error after one unit of time using the unsplit method is thus bounded as

$$\begin{aligned} & \|((LW(A, k))^{1/k} - \exp(\Lambda \partial_x))u\| \\ & \leq \frac{1}{k}(\frac{1}{2}(k^2 \|A^3\| + kh^2 \|A\|) + O(k^4)) \\ & \leq \frac{1}{2}k^2(a^3 + a/\lambda^2) + O(k^4). \end{aligned}$$

Since we require an error $\approx \tau$, we set

$$\frac{1}{2}k^2(a^3 + a/\lambda^2) = \tau$$

giving

$$k^2 = \frac{6\tau}{a(a^2 + 1/\lambda^2)}.$$

Thus $w(\tau; \lambda)$, the work required to achieve a given accuracy τ using Lax-Wendroff with mesh ratio λ , is given by

$$\begin{aligned} w(\tau; \lambda) &= \frac{\lambda W}{k^2} \\ &= (\lambda a + 1/\lambda a) \frac{a^2 W}{6\tau}. \end{aligned}$$

We have not yet specified λ . Choosing λ to minimize $w(\tau; \lambda)$ gives $\lambda = 1/a$ and the minimum work $w(\tau)$ is

$$w(\tau) = \frac{a^2 W}{3\tau} \quad \text{for unsplit Lax-Wendroff.} \quad (2.21)$$

Note that the optimal mesh ratio $\lambda = 1/a$ is also the stability limit for this problem. We can actually see that this is the optimal mesh ratio by looking only at the error at time $t = 1$. Since this error is bounded by

$$\frac{1}{2}(k^2 a^3 + h^2 a) + O(k^4)$$

it is clearly optimal to choose k and h so that the two terms $k^2 a^3$ and $h^2 a$ are roughly the same size (for otherwise we could increase k or h , and decrease the amount of work we do, without substantially increasing the error).

So far this analysis is completely standard and our results agree with those of Strang[49]. However, the same type of analysis, when applied to time-split methods under the assumption (2.19), yields some illuminating new results. This will now be done, first for the method (2.18a,b) and then for (2.18a,c).

Efficiency of the split method (2.18a,b) on perturbed problems. Let W_s be the work required to apply Lax-Wendroff on the slow scale and W_f^{exp} the work required to compute $\exp(k\Lambda_f \partial_x)U_m^n$. Then the work required for a single step of the time-split method is $W^{\text{TSM}} = W_s + 2W_f^{\text{exp}}$. Typically $W^{\text{TSM}} \approx W$. The error over one unit of time for the split scheme is bounded by

$$\begin{aligned} & \|((Q_f(k/2)Q_s(k)Q_f(k/2))^{1/k} - \exp(\Lambda \partial_x))u\| \\ & \leq \frac{1}{k} \|E_{\text{split}}(k)u + E_s(k)u + 2E_f(k/2)u + O(k^4)\|. \end{aligned}$$

For (2.18b), $E_f(k/2) = 0$. The truncation error for Lax-Wendroff on the slow scale is bounded by

$$\begin{aligned}\|E_s(k)u\| &\leq \frac{1}{6}k(k^2b^3 + k^2b) \\ &= \frac{1}{6}k^2(b^3 + b/\lambda^2).\end{aligned}$$

The splitting error for (2.17) is easily computed to be

$$\begin{aligned}E_{\text{split}}(k) &= \exp(\frac{1}{6}kA_f\partial_x)\exp(kA_s\partial_x)\exp(\frac{1}{6}kA_f\partial_x) - \exp(k(A_f + A_s)\partial_x) \\ &= -\frac{1}{6}k^3(\frac{1}{4}A_f^2A_s - \frac{1}{2}A_fA_sA_f + \frac{1}{4}A_sA_f^2 \\ &\quad - \frac{1}{4}A_s^2A_f + A_sA_fA_s - \frac{1}{2}A_fA_s^2)\partial_x^3 + O(k^4)\end{aligned}\tag{2.22}$$

so that

$$\|E_{\text{split}}(k)u\| \leq \frac{1}{6}k^3(a^2b + ab^2) \approx \frac{1}{6}k^3a^2b,$$

although it may be much smaller for some problems. Since our results depend very much on the size of this error, we will suppose for now that

$$\|E_{\text{split}}(k)u\| \leq \frac{1}{6}k^2\sigma$$

for some σ , so that

$$\frac{1}{k}\|E_{\text{split}}(k)u + E_s(k)u\| \leq \frac{1}{6}k^2(\sigma + b^3 + b/\lambda^2).$$

In order to obtain accuracy τ we must take

$$k^2 = \frac{6\tau}{\sigma + b^3 + b/\lambda^2}$$

so

$$\begin{aligned}w(\tau; \lambda) &= \lambda W^{\text{TSM}}/k^2 \\ &= \lambda(\sigma + b^3 + b/\lambda^2) \frac{W^{\text{TSM}}}{6\tau}.\end{aligned}\tag{2.23}$$

The optimal stepsize ratio λ now depends on the size of the splitting error and is given by

$$\lambda = \sqrt{\frac{b}{\sigma + b^3}}\tag{2.24}$$

so that

$$w(\tau) = \sqrt{b(\sigma + b^3)} \frac{W^{\text{TSM}}}{3\tau} \quad \text{for the time split method (2.18a,b).}$$

If $\sigma < b^3$ (e.g., when A_f and A_s commute), then (2.24) gives $\lambda \approx 1/b$ and

$$w(\tau) = \frac{b^2 W^{\text{TSM}}}{3\tau}.\tag{2.25}$$

TABLE 2.1

Reduction in work over (2.21) obtained by using the time-split method (2.18a,b) on (2.17). The results depend on the size of the splitting error.

case	$E_{\text{split}}(k)$	optimal λ	reduction in work
general	σk^3	$\sqrt{\frac{\epsilon a}{\sigma + \epsilon^3 a^3}}$	$\sqrt{\epsilon(\sigma + \epsilon^3 a^3)}$
best	0	$\frac{1}{\epsilon a}$	ϵ^2
typical	$\epsilon a^3 k^3$	$\frac{1}{a}$	ϵ

When $W^{TSM} \approx W$ this is better than (2.21) by a factor of ϵ^2 , meaning greatly improved efficiency. Note that when $\sigma = 0$ the only error incurred is the error in using Lax-Wendroff on the slow scale. From our previous discussion of Lax-Wendroff it is clear why $\lambda = 1/b$ is optimal in this case.

On the other hand, if the splitting error is as bad as (2.22) indicates, then $\sigma = a^2 b$ and $\lambda \approx 1/a$ in (2.24), giving

$$w(\tau) = \frac{abW^{TSM}}{3\tau}.$$

This is still an improvement over (2.21), although now by only a factor of ϵ . Note that now λ is chosen appropriate to the fast scale, even though the fast part of the problem is solved exactly. This is necessary because of the splitting error. Indeed, if we try to use $\lambda = 1/b$ when $\sigma = a^2 b$, we obtain no improvement over (2.21). For this reason it is advisable to always use small timesteps with the time-split method (2.18a,b) unless $E_{\text{split}}(k)$ is known to be very small, in which case even greater efficiency is achieved by using larger timesteps.

These results are summarized in Table 2.1.

Efficiency of the split method (2.18a,c) with sparse A_f . When Lax-Wendroff is used for both operators, the work for a single step of the time-split method is given by $W^{TSM} = W_s + mW_f$, where W_f is the work required to apply Lax-Wendroff on the fast scale. We are assuming that $W_f \ll W_s \approx W$. Suppose that $W_f = \gamma W$ for some $\gamma \ll 1$. In this case, the best we can hope for is to decrease the required work by a factor of γ . We will see that in general we can reduce the work by a factor of roughly $\gamma + \sqrt{\epsilon}$ by choosing the mesh ratio appropriately. When the splitting error is negligible, we can improve this to $\gamma + \epsilon$.

Because we are still free to choose m in (2.18c), the mesh ratios we use on the fast and slow scales are essentially independent for this problem. The local truncation error

for (2.18c) is

$$\begin{aligned} E_f(k/2) &= (LW(A_f, k/m))^{m/2} - \exp(\frac{1}{2}kA_f\partial_x) \\ &= \left(\exp(\frac{k}{m}A_f\partial_x) - \frac{1}{6}\left(\frac{k^2}{m^2}A_f^2 - \frac{k}{m}h^2A_f\right)\partial_x^2 \right)^{m/2} - \exp(\frac{1}{2}kA_f\partial_x) \\ &= -\frac{1}{6}\left(\frac{k^2}{2m^2}A_f^2 - \frac{1}{2}h^2A_f\right)\partial_x^2 + O(k^4). \end{aligned}$$

The optimal value of m is that which makes $k^2a^2/m^2 \approx h^2$, or $m \approx \lambda a$ where $\lambda = k/h$ is the mesh ratio for the slow scale. The optimal mesh ratio on the fast scales is thus $k/mh = 1/a$ regardless of λ .

Using this value of m , we compute the following bound for the error at time $t = 1$, using (2.1),

$$\begin{aligned} \frac{1}{k} \|E_{\text{aplit}}(k)u + E_s(k)u + 2E_f(k/2)u\| &\leq \frac{1}{6}k^2(\sigma + b^3 + b/\lambda^2 + a^3/m^2 + a/\lambda^2) + O(k^3) \\ &\approx \frac{1}{6}k^2(\sigma + b^3 + 2a/\lambda^2). \end{aligned} \quad (2.26)$$

We then obtain

$$w(\tau; \lambda) \approx \lambda(\sigma + b^3 + 2a/\lambda^2) \frac{W_s + \lambda a W_f}{6\tau} \quad \text{for (2.4a,c)}. \quad (2.27)$$

The optimal λ is most easily determined by requiring that the terms in the error (2.26) balance. This gives

$$\lambda = \sqrt{\frac{2a}{\sigma + b^3}}. \quad (2.28)$$

Again we will consider the best and worst cases, $\sigma = 0$ and $\sigma = a^2b$. When the splitting error is negligible, (2.24) gives

$$\lambda = \sqrt{\frac{2a}{b^3}} \approx \frac{1}{\epsilon^{3/2}a}. \quad (2.29)$$

In this case the optimal mesh ratio appears to be larger than the optimal mesh ratio for the slow problem alone (which would be $1/\epsilon a$). This counterintuitive result is due to the fact that otherwise the error on the fast scale dominates the error on the slow scale. By taking larger timesteps on the slow scale we decrease the work without increasing the error, or so the efficiency analysis tells us. Unfortunately, the mesh ratio (2.29) is larger than the stability bound for $LW(A_s, k)$, which is $1/\epsilon a$, and so this cannot be used in practice. The best we can do is to take

$$\lambda \approx \frac{1}{\epsilon a}$$

with corresponding work

$$\begin{aligned} w(\tau) &= \frac{2\epsilon^2 a^3}{\epsilon a} \frac{W_s + \epsilon^{-1} W_f}{6\tau} \\ &= 2a^2 \frac{\epsilon W_s + W_f}{6\tau} \\ &\approx (\epsilon + \gamma) \frac{a^2 W}{3\tau} \end{aligned}$$

TABLE 2.2

Reduction in work over (2.21) obtained by using the time-split method (2.18a,c) on (2.17). The results depend on the size of the splitting error.

case	$E_{\text{split}}(k)$	optimal λ	reduction in work
general	σk^3	$\min\left(\frac{1}{\epsilon a}, \sqrt{\frac{2a}{\sigma + \epsilon^3 a^3}}\right)$	$\max\left(\epsilon, \sqrt{\frac{\sigma + \epsilon^3 a^3}{a^3}}\right) + \gamma$
best	0	$\frac{1}{\epsilon a}$	$\epsilon + \gamma$
typical	$\epsilon a^3 k^3$	$\frac{1}{\sqrt{\epsilon a}}$	$\sqrt{\epsilon} + \gamma$

which is better than (2.21) by a factor of $\frac{3}{2}(\epsilon + \gamma)$.

In the more typical situation, when $\sigma = a^2 b$, (2.28) becomes

$$\lambda \approx \sqrt{\frac{2a}{a^2 b}} \approx \frac{1}{\sqrt{ab}} = \frac{1}{\sqrt{\epsilon a}}$$

with corresponding work

$$\begin{aligned} w(\tau) &= \frac{3\epsilon a^3}{\sqrt{\epsilon a}} \frac{W_s + \epsilon^{-1/2} W_f}{6\tau} \\ &= a^2 \frac{\sqrt{\epsilon} W_s + W_f}{2\tau} \\ &\approx (\sqrt{\epsilon} + \gamma) \frac{a^2 W}{2\tau}. \end{aligned}$$

We thus see that if $\sqrt{\epsilon} < \gamma$, an increase in efficiency by the best possible factor of roughly γ is always possible. These results are summarized in Table 2.2.

2.6. Phase errors.

When solving differential equations with wave-like solutions, it is frequently desirable to compute the *phase errors* of the finite difference scheme employed. Comparing the phase errors for the time-split method with those for the unsplit method provides some further insight into the results of Section 2.5.

Consider again the constant coefficient problem $u_t = Au_x$ and denote the eigenvalues and eigenvectors of A by μ_j and \hat{u}_j , respectively,

$$A\hat{u}_j = \mu_j \hat{u}_j, \quad j = 1, 2, \dots, r$$

with the μ_j ordered as in (1.26). As usual we suppose that $\|A\| \approx \rho(A) = \mu_r$. If we take as initial conditions a single mode

$$u(x, 0) = e^{i\xi x} \hat{u}_j \quad (2.30)$$

for some j and some wavenumber ξ , then the true solution at time t is simply

$$u(x, t) = e^{i\xi(x + \mu_j t)} \hat{u}_j.$$

The wave thus propagates with a *phase speed* μ_j .

Now suppose we apply a single step of unsplit Lax-Wendroff to $u(x, 0)$. By (1.13) we obtain

$$\begin{aligned} LW(\Lambda, k)u(x, 0) &= u(x, k) - \frac{1}{6}k(k^2 A^3 - h^2 A)u_{xxx}(x, 0) + O(k^4) \\ &= e^{i\xi x} (e^{i\xi \mu_j k} - \frac{1}{6}k(k^2 \mu_j^3 - h^2 \mu_j)(i\xi)^3) \hat{u}_j + O(k^4) \\ &= \exp\{i\xi[x + k(\mu_j + \frac{1}{6}k^2(\mu_j^3 - \mu_j/\lambda^2)\xi^2)]\} \hat{u}_j + O(k^4). \end{aligned}$$

The phase speed of the numerical wave is

$$\mu_j + \frac{1}{6}k^2(\mu_j^3 - \mu_j/\lambda^2)\xi^2 + O(k^3).$$

The optimal mesh ratio for Lax-Wendroff is $\lambda \approx 1/|\mu_r|$. In practice, of course, one never has exactly the optimal mesh ratio, so we suppose only that $\lambda = 1/\mu$ with $\mu \geq |\mu_r|$. We then find that the error in the phase speed for the j th eigenvector with wavenumber ξ is

$$\text{phase speed error} = \frac{1}{6}k^2(\mu_j^3 - \mu_j \mu^2)\xi^2 + O(k^3).$$

For comparison purposes we again wish to normalize by some measure of the work required to compute the solution. We define the *normalized phase speed error* $\phi_j(\xi)$ as

$$\phi_j(\xi) = (\text{phase speed error})/kh.$$

For the unsplit method we have

$$\phi_j(\xi) = \frac{1}{6\mu}(\mu_j^3 - \mu_j \mu^2)\xi^2 + O(k).$$

If $\mu_j = \mu$ exactly then there is no error in this mode of the computed solution. In general, however, the error is roughly

$$\phi_j(\xi) \approx -\frac{1}{6}\mu_j \mu_r \xi^2. \quad (2.31)$$

Now consider the split method (2.18a,b) where $\exp(\frac{1}{2}kA_f \partial_x)$ is known exactly and suppose to begin with that there is no splitting error for the splitting $\Lambda = \Lambda_f + \Lambda_s$. Then the matrices are simultaneously diagonalizable and so the \hat{u}_j are also eigenvectors of Λ_f and Λ_s . We then have

$$\Lambda_s \hat{u}_j = \mu_{sj} \hat{u}_j$$

with $|\mu_{sj}| \leq \epsilon |\mu_r|$. The optimal mesh ratio as found in Section 2.5 is then $\lambda \approx 1/\epsilon\mu$. When applying (2.18a,b), the only error is the Lax-Wendroff error on the slow scale, so after applying one step of the split operator $Q(k)$ we obtain

$$Q(k)u(x, 0) = u(x, k) - \frac{1}{6}k(k^2 A_s^3 - h^2 A_s)u_{xxx}(x, 0) + O(k^4).$$

Proceeding exactly as before we find that the normalized error is

$$\begin{aligned} \phi_j(\xi) &= \frac{1}{6\epsilon\mu}(\mu_{sj}^3 - \epsilon^2 \mu_{sj} \mu^2)\xi^2 + O(k) \\ &\approx -\frac{1}{6}\epsilon \mu_{sj} \mu_r \xi^2 \end{aligned} \quad (2.32)$$

This is always better than (2.31). Just how much better it is will depend on the velocity of the mode (2.30). For slow waves, those for which $|\mu_j| \leq \epsilon |\mu_r|$, say, we have $|\mu_{sj}| \approx |\mu_j|$ and so (2.32) is better than (2.31) by roughly a factor of ϵ . For fast waves, on the other hand, for which $|\mu_j| \approx |\mu_r|$, (2.32) is better than (2.31) by a factor of ϵ^2 . The improvement in phase errors is thus more dramatic for fast waves than for slow waves. This is to be expected since it is the fast subproblem which is being solved exactly.

How do these results fit in with the results of Section 2.5? There we saw that for the method (2.18a,b) with no splitting error, the work required to obtain a given accuracy should be reduced by ϵ^2 , or, equivalently, that the normalized error should be reduced by ϵ^2 . Yet here it seems that the error in slow waves is reduced only by ϵ . This apparent contradiction is resolved by reexamining (2.31). This shows that for the unsplit method phase errors in slow waves are already smaller by a factor of ϵ than those in fast waves. Hence with the unsplit method errors in the fast waves dominate, and reducing those errors by ϵ^2 (and errors in the slow waves by ϵ) causes the overall global error to decrease by ϵ^2 .

This has an important consequence which was not directly apparent from the analysis of Section 2.5. For problems in which fast waves are absent from the solutions of interest, and only slow waves are present, the use of the time-split method can be expected to decrease the normalized errors, and hence improve the efficiency, by at most a factor of ϵ , even in the absence of splitting errors.

Now suppose that the splitting error is nonzero. For the constant coefficient system this means that A_f and A_s do not commute and the eigenvectors \hat{u}_j of A are no longer eigenvectors of A_f and A_s . Because of this initial conditions consisting of a single mode (2.30) no longer lead to a single-mode solution and we are not able to consider each mode separately.

Instead we take more general initial conditions

$$u(x, 0) = e^{i\xi x} \hat{u}$$

where

$$\hat{u} = \sum_{m=1}^r \alpha_m \hat{u}_m \quad (2.33)$$

and look at phase errors in the j th mode. We assume that the α_m are order unity and for convenience take $\alpha_j = 1$.

The truncation error for the split method is now the sum of the truncation error for Lax-Wendroff on A_s and the splitting error (2.22). So

$$Q(k)u(x, 0) = u(x, k) - \frac{1}{6}kBu_{xxx} + \dots \quad (2.34)$$

where

$$B = (k^2 A_s^3 - h^2 A_s) + k^2 \left[\frac{1}{4} A_f^2 A_s - \frac{1}{2} A_f A_s A_f + \frac{1}{4} A_s A_f^2 - \frac{1}{2} A_s^2 A_f + A_s A_f A_s - \frac{1}{2} A_f A_s^2 \right].$$

Assuming as usual that $\|A_s\| \leq \epsilon \|A_f\| \approx \epsilon \mu_r$ and using the optimal mesh ratio $k/h \approx 1/\mu_r$ gives a rough bound on B :

$$\|B\| \leq 2k^2 \epsilon \mu_r^3 + O(k^2 \epsilon^2 \mu_r^3). \quad (2.35)$$

Now we must make an additional assumption on the matrix A , namely that the eigenvectors of A are well-conditioned. If X is the matrix of eigenvectors \hat{u}_m then we assume

$$\|X\| \|X^{-1}\| = O(1).$$

This means that we can expand $B\hat{u}$ as

$$B\hat{u} = \sum_{m=1}^r \beta_m \hat{u}_m$$

with $|\beta_m|$ of the same order as $\|B\|$. This is because $\beta = X^{-1}BX\alpha$ and so $\|\beta\| \leq \|B\| \|X\| \|X^{-1}\| \|\alpha\| \approx \|B\|$. Using (2.33) in (2.34) then yields

$$\begin{aligned} Q(k)u(x, 0) &= \sum_{m=1}^r \alpha_m e^{i\xi(x+\mu_m k)} \hat{u}_m - \frac{1}{6}k(i\xi)^3 \sum_{m=1}^r \beta_m \hat{u}_m + O(k^4) \\ &= \sum_{m=1}^r \alpha_m \exp\{i\xi[x + k(\mu_m + \frac{1}{6}\xi^2 \beta_m/\alpha_m)]\} \hat{u}_m + O(k^4). \end{aligned}$$

Using (2.35) we can compute the normalized phase speed error in the j th mode:

$$\phi_j(\xi) \approx \frac{1}{3}\epsilon \mu_r^2 \xi^2.$$

Note that unlike the previous cases the phase error here is the same for fast waves and slow waves. Comparing this with (2.31) shows that for fast waves ($\mu_j \approx \mu_r$) the error is reduced by ϵ while for slow waves ($\mu_j \leq \epsilon \mu_r$) the error is not reduced at all. This indicates that when computing a solution containing only slow waves, the time-split method with splitting errors may be no more efficient than the unsplit method.

2.7. Block triangular systems.

Since the efficiency of the split scheme is limited primarily by the splitting error, it is interesting to investigate how this error depends on the coupling between fast and slow scales in a simple model system. Consider the block triangular system with

$$A = \begin{bmatrix} \frac{1}{\epsilon} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}$$

and the splitting

$$A_f = \begin{bmatrix} \frac{1}{\epsilon} A_{11} & 0 \\ 0 & 0 \end{bmatrix}, \quad A_s = \begin{bmatrix} 0 & A_{12} \\ 0 & A_{22} \end{bmatrix}$$

and suppose that $\|A_{jj}\| \approx 1$ and that $\|A_{12}\| = \alpha \leq 1$. For variety we have chosen a problem in which $A_f \rightarrow \infty$ as $\epsilon \rightarrow 0$ rather than $A_s \rightarrow 0$. The theory developed in the previous section applies equally well in this situation.

Here A_{12} is the coupling between fast and slow scales. If $A_{12} = 0$, the problem is uncoupled and $E_{\text{split}}(k) = 0$. In general, from (2.22),

$$E_{\text{split}}(k) = -\frac{k^3}{6} \begin{bmatrix} 0 & \frac{1}{4\epsilon} A_{11} (\frac{1}{\epsilon} A_{11} A_{12} - 2A_{12} A_{22}) \\ 0 & 0 \end{bmatrix} \partial_x^3 + O(k^4).$$

Thus $\|E_{\text{split}}(k)u\| \approx \alpha k^3 / 24\epsilon^2$. The efficiency of the splitting depends on the size of α . In the notation used above, we have

$$a = 1/\epsilon, \quad b = 1, \quad \sigma = \frac{1}{4} \alpha a^2 b.$$

For unsplit Lax-Wendroff, (2.21) gives

$$w(\tau) = \frac{1}{\epsilon^2} \frac{W}{3\tau}. \quad (2.36)$$

The time-split method (2.18a,b) is always more efficient if we choose

$$\lambda \approx (1 + \frac{1}{4} \alpha a^2 b)^{-1/2}.$$

For example, if $\alpha \approx 1$ we should use $\lambda \approx 2/a = 2\epsilon$ in order to reduce (2.36) by a factor of ϵ . The maximum efficiency indicated in (2.25) is achievable only if $\alpha \leq \epsilon^2$, in which case taking $\lambda = 1$ reduces (2.36) by a factor of ϵ^2 .

2.8. Reducing the splitting error.

For block triangular systems in which A_{12} is not small, it is possible to reduce the coupling through a change of variables so that the optimal efficiency can be achieved. A change of variables amounts to replacing u by $\bar{u} = Bu$ for some nonsingular matrix B . The system $u_t = Au_x$ then becomes $\bar{u}_t = B A B^{-1} \bar{u}_x$. Clearly, if B is chosen to be the eigenvector matrix of A then the problem completely decouples into independent scalar equations. We are seeking something less expensive which only decouples the fast and

slow scales. Thus we want a (well-conditioned) matrix B such that

$$BAB^{-1} = \begin{bmatrix} \frac{1}{\epsilon} C_{11} & 0 \\ 0 & C_{22} \end{bmatrix} \quad (2.37)$$

with $\|C_{11}\| \approx \|C_{22}\| \approx 1$. In the block triangular case, it suffices to consider B of the form

$$B = \begin{bmatrix} I & B_{12} \\ 0 & I \end{bmatrix}, \quad B^{-1} = \begin{bmatrix} I & -B_{12} \\ 0 & I \end{bmatrix}.$$

Then

$$BAB^{-1} = \begin{bmatrix} \frac{1}{\epsilon} A_{11} & -\frac{1}{\epsilon} A_{11} B_{12} + A_{12} + B_{12} A_{22} \\ 0 & A_{22} \end{bmatrix}$$

and so B_{12} should be chosen to solve

$$\frac{1}{\epsilon} A_{11} B_{12} - B_{12} A_{22} = A_{12} \quad (2.38)$$

in order to completely decouple the fast and slow scales.

In the present context solving for B_{12} from (2.38) is not worthwhile. In order to achieve optimal efficiency we need only reduce the coupling by one or two factors of ϵ . Further reductions do not gain anything once the Lax-Wendroff errors dominate. This suggests taking

$$B_{12} = \epsilon A_{11}^{-1} A_{12} \quad (2.39)$$

so that

$$BAB^{-1} = \begin{bmatrix} \frac{1}{\epsilon} A_{11} & A_{12}^{(1)} \\ 0 & A_{22} \end{bmatrix}$$

where

$$A_{12}^{(1)} = \epsilon A_{11}^{-1} A_{12} A_{22}.$$

We now have $\|A_{12}^{(1)}\| \approx \epsilon \alpha$ provided $\|A_{11}^{-1}\| \approx 1$. The coupling is thus reduced by a factor of ϵ through the use of a very simple change of variables. This process can be repeated to obtain additional factors of ϵ . This change of variables has been suggested by Kreiss[32] in a similar context.

For full systems of the form

$$A = \begin{bmatrix} \frac{1}{\epsilon} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

we can obtain a similar reduction in the size of both off-diagonal blocks and again reduce the splitting error by several orders of magnitude. In this case we consider B of the form

$$B = \begin{bmatrix} I & K \\ 0 & I \end{bmatrix} \begin{bmatrix} I & 0 \\ L & I \end{bmatrix} = \begin{bmatrix} I + KL & K \\ L & I \end{bmatrix}.$$

It is easy to verify that the lower corner of A is annihilated by taking L to satisfy

$$\frac{1}{\epsilon} LA_{11} - A_{22}L - LA_{12}L + A_{21} = 0.$$

The matrix K can then be chosen as before to remove the remaining upper corner. This results in a system of the form (2.37). This particular transformation is discussed more completely by O'Malley and Anderson[44]. Again, however, we are not interested here in completely annihilating the corners, but rather in reducing them by a factor of ϵ . This is easily accomplished by taking

$$\begin{aligned} K &= \epsilon A_{11}^{-1} A_{12} \\ L &= -\epsilon A_{21} A_{11}^{-1}. \end{aligned}$$

Example 2.4. This problem is designed to illustrate the effects of the splitting error and the use of the change of variables (2.39). Consider

$$\begin{bmatrix} u \\ v \end{bmatrix}_t = \begin{bmatrix} 10 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}_x \quad \text{for } 0 \leq x \leq 1, t \geq 0 \quad (2.40)$$

with initial conditions

$$u(x, 0) = v(x, 0) = e^{-100(x-1/2)^2}$$

and periodic boundary conditions

$$\begin{aligned} u(0, t) &= u(1, t), & t \geq 0, j = 1, 2, \\ v(0, t) &= v(1, t), & t \geq 0, j = 1, 2. \end{aligned}$$

Figure 2.1a shows the results after 236 time steps using Lax-Wendroff with $h = 1/50$ and $k = h/10$ on the unsplit problem. Figure 2.1b shows the results based on the splitting

$$A_f = \begin{bmatrix} 10 & 0 \\ 0 & 0 \end{bmatrix}, \quad A_s = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}.$$

We used $k = h = 1/50$ with

$$Q_s(k) = LW(A_s, k), \quad Q_f(k/2) = (LW(A_f, k/10))^5.$$

In this case $E_s(k) = E_f(k/2) = 0$ by a judicious choice of k/h and m . The second component v is computed exactly and the errors in u are due entirely to the splitting error.

If the change of variables suggested in (2.39) is applied twice to (2.40) with $\epsilon = 0.1$, we obtain the new variable

$$\bar{u} = u - (\epsilon + \epsilon^2)v = u - 0.11v \quad (2.41)$$

and (2.40) becomes

$$\begin{bmatrix} \bar{u} \\ v \end{bmatrix}_t = \begin{bmatrix} 10 & 0.01 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \bar{u} \\ v \end{bmatrix}_x.$$

If we solve this system with the same split scheme as before and then transform back to the original variables by $u = \bar{u} + 0.11v$, the errors in u are reduced to $O(10^{-3})$ as seen in Figure 2.1c.

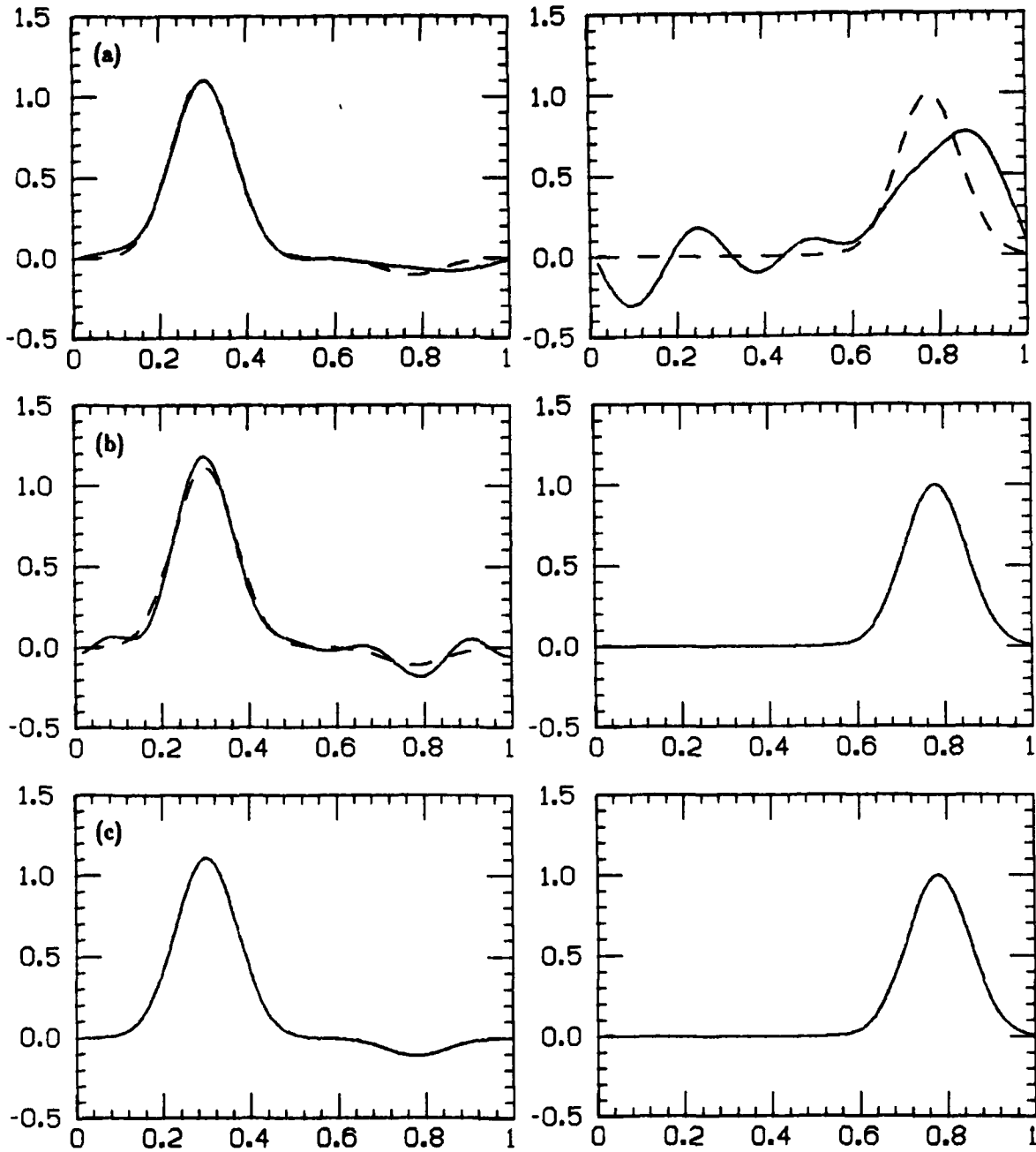


FIG. 2.1. True (dashed line) and computed solutions at $t = 4.72$ for Example 2.1. The first component, u , is on the left and the second component, v , is on the right. The schemes used are: (a) unsplit Lax-Wendroff, (b) the time-split method (2.18a,b), and (c) the time-split method with the change of variables (2.41).

2.9. The shallow water equations.

In this section the efficiency analysis of Section 2.5 is applied to the one-dimensional shallow water equations (1.36). We will use the splitting (1.38) and assume that the condition (1.37) holds. In Section 2.4 we computed the splitting error for this system. In general this error is nonnegligible. Since all of the waves in the solution to the original problem are fast waves, the analysis of Sections 2.5 and 2.6 leads us to expect the time-split method to be more efficient than the unsplit method by a factor of ϵ .

Taking the mesh ratio as in (1.39), the time-split method (2.18a,b) becomes

$$\begin{aligned} U_m^* &= \frac{1}{2}[U_{m-p}^n + U_{m+p}^n + \Phi_{m-p}^n - \Phi_{m+p}^n] \\ \Phi_m^* &= \frac{1}{2}[U_{m-p}^n - U_{m+p}^n + \Phi_{m-p}^n + \Phi_{m+p}^n] \\ \begin{bmatrix} U \\ \Phi \end{bmatrix}_m^{**} &= LW(A_s, k) \begin{bmatrix} U \\ \Phi \end{bmatrix}_m^* \\ U_m^{n+1} &= \frac{1}{2}[U_{m-p}^{**} + U_{m+p}^{**} + \Phi_{m-p}^{**} - \Phi_{m+p}^{**}] \\ \Phi_m^{n+1} &= \frac{1}{2}[U_{m-p}^{**} - U_{m+p}^{**} + \Phi_{m-p}^{**} + \Phi_{m+p}^{**}]. \end{aligned} \quad (2.42)$$

Since $\tilde{u}_t = A_s \tilde{u}_x$ is a quasilinear problem, an appropriate generalization of the Lax-Wendroff operator must be used for $LW(A_s, k)$. We have used MacCormack's method (see [41]).

We wish to compare the efficiency of the split method with that of the unsplit method. For convenience in checking our predictions against experimental results, we choose to compare the error at a fixed time normalized by the amount of work required to compute it (rather than the amount of work required to compute a solution with a given error). Since the split and unsplit methods take roughly the same amount of work per grid point per timestep, it suffices to normalize the errors at a fixed time by dividing by kh , as we did to normalize the phase errors in Section 2.6.

We first consider the unsplit MacCormack's method applied to (1.36). Since $A \approx A_f$ with small, slowly varying perturbations, the errors in applying MacCormack's method on A are roughly the same as those in applying Lax-Wendroff on the constant coefficient matrix A_f . We can thus use the results of Section 2.5 directly to analyze the efficiency of the unsplit method.

Since $\rho(A) \approx \rho(A_f) = \phi_0/2$, the optimal mesh ratio is $\lambda \approx 2/\phi_0$. The error at time $t = 1$ is bounded using the truncation error (2.20) by

$$\frac{1}{k} \|E^{LW}(k)\tilde{u}\| \approx \frac{1}{8}(k^2 \|A^3\| + h^2 \|A\|) \|\tilde{u}_{xxx}\|. \quad (2.43)$$

For smooth solutions we can assume that $\|\tilde{u}_{xxx}\| = O(\epsilon\phi_0)$. Then taking $\lambda = O(1/\phi_0)$, we find the normalized error by dividing (2.43) by kh :

$$\text{normalized error} = O(\epsilon\phi_0^3) \quad \text{for the unsplit method.} \quad (2.44)$$

Now to analyze the split method. The splitting error (2.22) is in general $O(\epsilon^2\phi_0^4k^3)$ for this problem. The results of Section 2.5 indicate that for the time-split method with

$E_f = 0$ and $E_{\text{split}}(k)$ nonnegligible, we should again take $\lambda = O(1/\phi_0)$ and hence the optimal p in (1.39) should be some small integer, independent of both ϵ and ϕ_0 . Numerical experiments confirm this prediction (see Example 2.5 below) and in fact $p = 3$ or 4 seems to be optimal over a wide range of values of ϵ and ϕ_0 .

Using this optimal value of λ , the normalized error should, in theory, be reduced by a factor of ϵ over (2.44), i.e.,

$$\text{normalized error} = O(\epsilon^2 \phi_0^3) \quad \text{for the split method.} \quad (2.45)$$

This is also confirmed in the following example.

Example 2.5. Consider the shallow water equations (1.36) on $0 \leq x \leq 1$ with initial conditions

$$\begin{aligned} u(x, 0) &= \epsilon \phi_0 \cos(2\pi x) \\ \phi(x, 0) &= \phi_0(1 + \epsilon \sin(2\pi x)) \end{aligned}$$

and periodic boundary conditions

$$\begin{aligned} u(0, t) &= u(1, t) \\ \phi(0, t) &= \phi(1, t). \end{aligned}$$

We first compare the error obtained at a fixed time using various values of p in the time-split method (2.42). Figure 2.2 shows the normalized errors as a function of p for $\phi_0 = 1$ and $\epsilon = 10^{-2}, 10^{-3}, 10^{-4}$ with $h = 1/50$. Other values of ϕ_0 , ϵ , and h have also been tested and lead to graphs which are qualitatively very similar to Figure 2.2. In all cases $p = 3$ or 4 is optimal.

We can also compare the error in the split method with that of the unsplit method using the optimal values of λ for each. For the split method we take $p = 3$ (corresponding to $\lambda = 12/\phi_0$) and for the unsplit method we use $\lambda = 1/\phi_0$. Figure 2.3 shows the results for $\phi_0 = 1$. We see the normalized error plotted as a function of ϵ . This confirms the prediction that using the split method reduces the normalized error by a factor of ϵ . More significantly, it shows that even for fairly large (i.e. realistic) values of ϵ the time-split method is superior. For example, at $\epsilon = 0.1$ the errors are reduced by a factor of roughly 100.

Simple waves. The splitting error for the quasilinear problem (1.36) with the splitting (1.38) depends on u and ϕ and the relation between them. In general it is nonnegligible but for certain special solutions, namely simple waves, the splitting error is identically zero.

The equations (1.36) can be written in characteristic form as

$$\begin{aligned} (u + \phi)_t &= -(\tfrac{1}{2}\phi + u)(u + \phi)_x \\ (u - \phi)_t &= (\tfrac{1}{2}\phi - u)(u - \phi)_x. \end{aligned} \quad (2.46)$$

The *Riemann invariants* $u + \phi$ and $u - \phi$ are each constant along characteristic curves in x - t space defined by the ordinary differential equations

$$\frac{dx}{dy} = \tfrac{1}{2}\phi(x, t) \pm u(x, t)$$

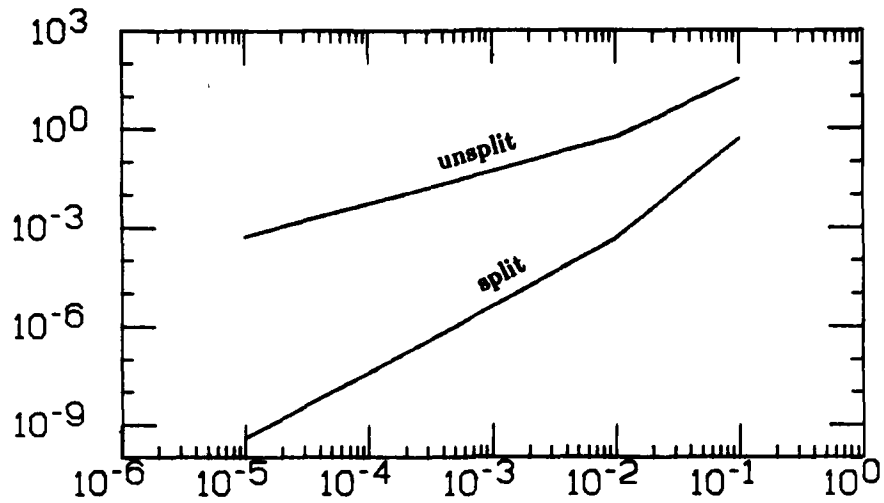


FIG. 2.2. Normalized errors in the shallow water equations of Example 2.5 as a function of the parameter p occurring in the mesh ratio (1.39). In all cases $t = 0.96$, $\phi_0 = 1$ and $h = 1/50$.

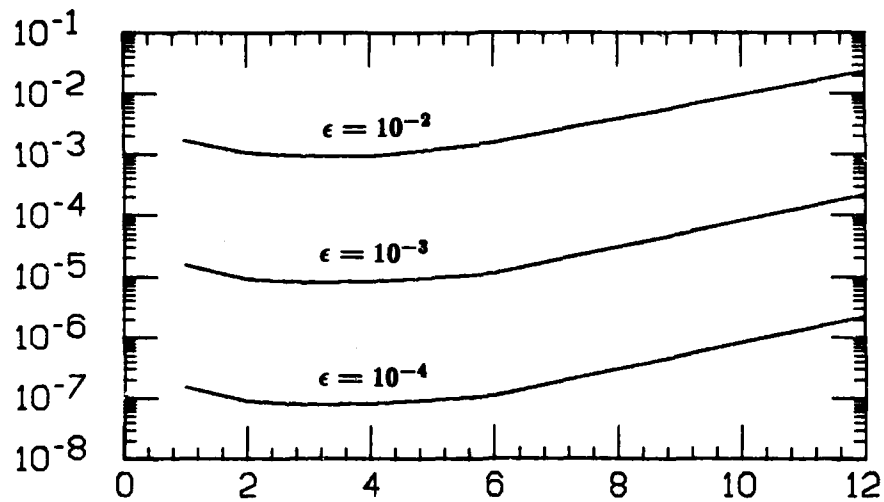


FIG. 2.3. Normalized errors in the shallow water equations of Example 2.5 as a function of ϵ for the unsplit method with $\lambda = 1/\phi_0$ and the split method with $\lambda = 12/\phi_0$. In the computations shown here $t = 0.96$, $\phi_0 = 1$ and $h = 1/50$.

and

$$\frac{dx}{dy} = -\left(\frac{1}{2}v(x, t) - u(x, t)\right)$$

respectively.

A solution for which one of the invariants is in fact constant for all x and t is called a *simple wave*. For simple waves the splitting error is identically zero. This is most easily seen by changing variables. Set

$$\begin{aligned}\rho(x, t) &= u(x, t) + \phi(x, t), \\ \sigma(x, t) &= u(x, t) - \phi(x, t).\end{aligned}\tag{2.47}$$

The equation (1.36) becomes

$$\begin{bmatrix} \rho \\ \sigma \end{bmatrix}_t = -\frac{1}{4} \begin{bmatrix} 3\rho + \sigma & 0 \\ 0 & \rho + 3\sigma \end{bmatrix} \begin{bmatrix} \rho \\ \sigma \end{bmatrix}_x.\tag{2.48}$$

The matrix occurring here is SAS^{-1} where

$$S = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

Applying the same similarity transformation to A_f and A_s leads to the following splitting of (2.48):

$$SA_f S^{-1} = \frac{1}{2} \begin{bmatrix} -\phi_0 & 0 \\ 0 & \phi_0 \end{bmatrix}, \quad SA_s S^{-1} = -\frac{1}{4} \begin{bmatrix} 3\rho + \sigma - 2\phi_0 & 0 \\ 0 & \rho + 3\sigma + 2 \end{bmatrix}.\tag{2.49}$$

Since we have applied a constant similarity transformation, it is easily verified that the splitting errors corresponding to the splitting (1.38) and (2.49) are also related by the same similarity transformation. Thus it suffices to show that for simple waves the splitting error in (2.49) is zero. This is easy to do, as we will see momentarily.

We note in passing that solutions to the shallow water equations can be computed directly in terms of ρ and σ using the splitting (2.49). With R and S denoting approximations to ρ and σ , the time-split method (2.42) then becomes

$$\begin{aligned}R_m^* &= R_{m-1}^* \\ S_m^* &= S_{m+1}^* \\ \begin{bmatrix} R \\ S \end{bmatrix}_m^{**} &= LW(A_s, k) \begin{bmatrix} R \\ S \end{bmatrix}_m^* \\ R_m^{n+1} &= R_{m-1}^{**} \\ S_m^{n+1} &= S_{m+1}^{**}.\end{aligned}\tag{2.50}$$

This form will prove particularly convenient when specifying boundary conditions for the intermediate solutions, as we will see in Section 4.5.

Suppose now that we are computing simple waves and that one of the invariants ρ or σ is constant, say $\sigma \equiv -\phi_0$. Then clearly $S_m^n \equiv -\phi_0$ in (2.50) and so the second

component of the splitting error is zero. The equation for ρ is $\rho_t = -\frac{1}{4}(3\rho + \sigma)\rho_x \equiv A(\rho)$ and it remains only to show that there is no error in using the splitting $A_1(\rho) = -\frac{1}{2}\phi_0\rho_x$, $A_2(\rho) = -\frac{1}{4}(3\rho + \sigma - 2\phi_0)\rho_x$. Since σ and ϕ_0 are constant, this is essentially the problem of Example 2.1(c) and so the splitting error is zero. Note that the expression (2.16) is consistent with this, since for simple waves $u_x = \phi_x$ and $u_{xx} = \phi_{xx}$.

It follows that the optimal mesh ratio for computing simple waves is $O(1/\epsilon\phi_0)$ leading to normalized errors which are reduced by a factor of ϵ^2 over (2.44):

$$\text{normalized error} = O(\epsilon^3\phi_0^3) \quad \text{for the split method on simple waves.}$$

These predictions are also confirmed by numerical experiments, as the following example shows.

Example 2.6. Consider the shallow water equations (1.36) on $0 \leq x \leq 1$ with initial conditions

$$\begin{aligned} u(x, 0) &= \epsilon\phi_0 \sin(2\pi x) \\ \phi(x, 0) &= \phi_0(1 + \epsilon \sin(2\pi x)) \end{aligned}$$

and periodic boundary conditions. Since $u - \phi$ is constant, the solution is a simple wave.

We again compare the normalized errors at a fixed time using various values of p in the time-split method (2.42). We expect $p = O(1/\epsilon)$ to be optimal. In order to test this theory when ϵ is small we must run the computations out to large times, $t = O(1/\epsilon)$. For each value of ϵ we will compare the normalized error at $t = 0.96/(100\epsilon)$, using values of $p \leq 12/(100\epsilon)$. This is roughly the stability limit of the method. (In Section 3.5 it will be shown that the stability limit is $k/h \leq 1/(2\epsilon\phi_0)$ which corresponds to $p \leq 1/(8\epsilon)$.) Since the stability limit is smaller than the optimal p predicted by the theory, we expect the normalized errors to be monotonically decreasing up to the stability limit. This is confirmed in Figure 2.4.

The theory also predicts that the resulting normalized errors at a fixed time should be $O(\epsilon^3\phi_0^3)$ at the optimal p , and hence that the errors at time $O(1/\epsilon)$ should be $O(\epsilon^2\phi_0^3)$. This is also confirmed by Figure 2.4.

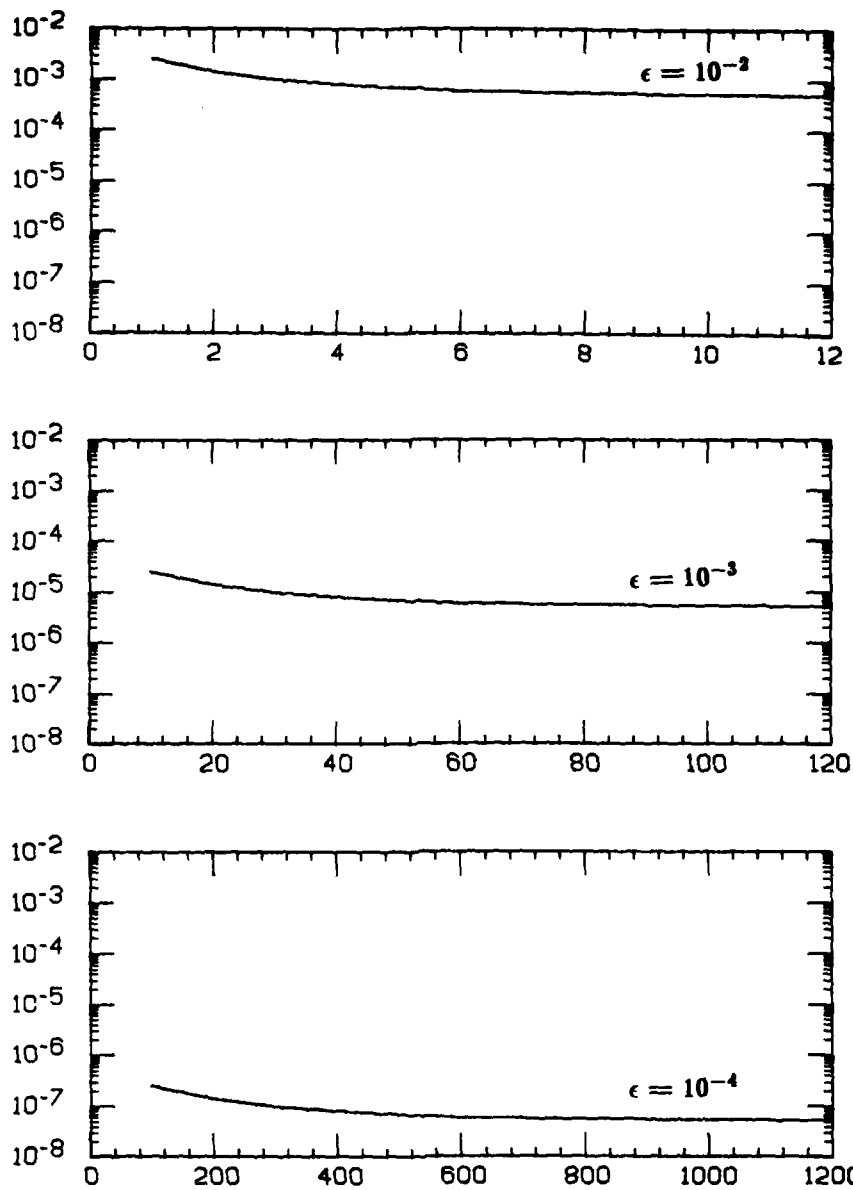


FIG. 2.4. Normalized errors in a simple-wave solution to the shallow water equations of Example 2.6 as a function of the parameter p occurring in the mesh ratio (1.39). In all cases $\phi_0 = 1$ and $h = 1/50$ while $t = 0.96/(100\epsilon)$.

3. Cauchy stability

3.1. Introduction to stability theory.

In this chapter we investigate the stability of the one-dimensional time-split method when applied to a constant coefficient hyperbolic problem on the entire real line, $-\infty < x < \infty$, or on a finite interval with periodic boundary conditions.

We will first engage briefly in a general discussion of Cauchy stability for a marching scheme of the form

$$U^{n+1} = Q(k)U^n \quad (3.1)$$

applied to a constant coefficient problem. More details can be found in Richtmyer & Morton[46] or Thomée[51]. We use a standard definition of stability, which can be written in several equivalent forms. We begin with the most natural of these.

STABILITY DEFINITION 3.1. *The operator $Q(k)$ is stable if for any fixed time T there exists a constant M_T such that*

$$\|Q^n(k)\| \leq M_T \quad (3.2)$$

for all k sufficiently small (say $k < k_0$) and $nk \leq T$.

The condition (3.2) ensures that for all initial vectors U^0 , the solution $U^n = Q^n(k)U^0$ satisfies

$$\|U^n\| \leq M_T \|U^0\| \quad (3.3)$$

for $nk \leq T$.

Here $\|\cdot\|$ represents some norm over all meshpoints at a fixed time. For example, the discrete ℓ_2 norm is given by

$$\|U^n\|_2^2 = h \sum_{m=-\infty}^{\infty} |U_m^n|^2$$

with $|\cdot|$ representing the usual vector two-norm.

Up until Section 3.4, where we introduce Sobolev norms, we will always suppose that the norm $\|\cdot\|$ is *equivalent* to the ℓ_2 norm in the sense that there exist constants M_1 and M_2 such that

$$M_1 \|U\|_2 \leq \|U\| \leq M_2 \|U\|_2$$

for all U . With this restriction, Stability Definition 3.1 is independent of the norm used. If $Q(k)$ is stable in the ℓ_2 norm then it is also stable in any equivalent norm.

The following equivalent definition of stability is sometimes easier to work with since it only requires a bound on $\|Q(k)\|$ rather than a uniform bound on $\|Q^n(k)\|$. The difficulty in applying the new definition is that such a bound, when it holds, will often hold only in a very special norm tailored to the problem, and will generally not hold in equivalent norms.

STABILITY DEFINITION 3.1'. *The operator $Q(k)$ is stable if there exists a norm $\|\cdot\|$ and a constant $\alpha \geq 0$ such that*

$$\|Q(k)\| \leq 1 + \alpha k \quad (3.4)$$

for all $k < k_0$.

Clearly (3.4) implies (3.2) since

$$\|Q^n(k)\| \leq \|Q(k)\|^n \leq (1 + \alpha k)^n \leq e^{\alpha T}$$

if $nk \leq T$ and we can thus take $M_T = e^{\alpha T}$. The converse, that such a norm exists for any stable scheme, is proved constructively in Chapter 4 of Richtmyer and Morton[46] as part of the Kreiss Matrix Theorem.

In some cases bounds of the form (3.4) can be obtained directly. This method of proving stability is referred to as the *energy method* since for physical systems the required norm is often simply the energy of the system. Often, however, it is easier to determine stability by an alternative approach known as the von Neumann method. We take U^n to be a single Fourier mode, $U^n = e^{i\xi n h} \hat{U}^n$ where \hat{U}^n is the vector of Fourier coefficients at time n , and insert this into (3.1). We find that U^{n+1} is again a single Fourier mode with coefficients

$$\hat{U}^{n+1} = G(\xi, k) \hat{U}^n$$

for some matrix $G(\xi, k)$, called the *amplification matrix*. Stability Definition 3.1 is equivalent to the following definition based on this amplification matrix.

STABILITY DEFINITION 3.2. *The operator $Q(k)$ is stable if for any fixed time T there exists a constant M_T such that powers of the corresponding amplification matrix are uniformly bounded by M_T ,*

$$\|G^n(\xi, k)\| \leq M_T \quad (3.5)$$

for all ξ , $k < k_0$ and $nk \leq T$.

Corresponding to Stability Definition 3.1' we have the following definition of stability, which is again equivalent.

STABILITY DEFINITION 3.2'. *The operator $Q(k)$ is stable if there exists a norm $\|\cdot\|$ and a constant $\alpha \geq 0$ such that*

$$\|G(\xi, k)\| \leq 1 + \alpha k \quad (3.6)$$

for all ξ and $k < k_0$.

Since every matrix norm is bounded below by the spectral radius, we find from Stability Definition 3.2' that a necessary condition for stability is the so-called *von Neumann condition*:

$$\rho(G(\xi, k)) \leq 1 + O(k). \quad (3.7)$$

This condition is frequently sufficient as well. Chapter 4 of Richtmyer and Morton[46] has a thorough discussion of sufficient conditions. Here we will mention only a few examples which will prove particularly useful.

If for all ξ and k , $G(\xi, k)$ is a normal matrix, i.e., if G commutes with its conjugate transpose, then $\|G(\xi, k)\|_2 = \rho(G(\xi, k))$. By using the 2-norm in Stability Definition 3.2' we see that in this case the von Neumann condition is sufficient for stability.

More generally, it suffices that the matrices $G(\xi, k)$ be *simultaneously normalizable*, as defined in the following theorem (see Richtmyer & Morton[46]).

THEOREM 3.1. *Suppose there exists a constant matrix S such that $SG(\xi, k)S^{-1}$ is a normal matrix for all $\xi, k < k_0$. Then the von Neumann condition is sufficient for stability.*

Proof. Define the vector norm $\|\cdot\|_S$ by

$$\|y\|_S = \|Sy\|_2.$$

This vector norm is equivalent to the 2-norm. The corresponding matrix norm is

$$\|A\|_S = \|SAS^{-1}\|_2. \quad (3.8)$$

In this norm we have

$$\begin{aligned} \|G(\xi, k)\|_S &= \|SG(\xi, k)S^{-1}\|_2 \\ &= \rho(SG(\xi, k)S^{-1}) \\ &= \rho(G(\xi, k)) \end{aligned}$$

and the theorem follows by using the norm $\|\cdot\|_S$ in Stability Definition 3.2'. ■

An important application of this theorem provides the result that the von Neumann condition is sufficient for stability if the $G(\xi, k)$ are simultaneously diagonalizable (since any diagonal matrix is normal). Many methods for the problem $u_t = Au_x$ have the property that their amplification matrices are polynomials in the matrix A and hence are diagonalized (for all ξ and k) by the eigenvector matrix of A (by the assumption of hyperbolicity, A is diagonalizable). In particular, the Lax-Wendroff operator and the exact solution operator have this property, and the von Neumann condition is sufficient for their stability.

3.2. Stability of the time-split method.

We now turn to the stability analysis of the time-split method (1.23). When $Q_f^2(k/2) = Q_f(k)$, as is true for the splittings (2.18), for example, Cauchy stability of the Strang splitting (1.21) is equivalent to stability of the first order splitting

$$U^{n+1} = Q_f(k)Q_s(k)U^n. \quad (3.9)$$

For simplicity we restrict our attention to this splitting, and set $Q(k) = Q_f(k)Q_s(k)$.

Let $G_f(\xi, k)$ and $G_s(\xi, k)$ be the amplification matrices corresponding to the operators $Q_f(k)$ and $Q_s(k)$, respectively. Then it is easy to verify that the amplification matrix $G(\xi, k)$ for $Q(k)$ satisfies

$$G(\xi, k) = G_f(\xi, k)G_s(\xi, k).$$

This allows us to calculate the amplification matrix for the time-split method relatively easily. In general the stability of $Q_f(k)$ and $Q_s(k)$ separately does not imply that $Q(k)$ is stable, or even that the von Neumann necessary condition is satisfied for $G(\xi, k)$, since the spectral radius is not submultiplicative (i.e., the inequality $\rho(G) \leq \rho(G_f)\rho(G_s)$ does **not** hold). It is easy to find examples for which $Q_f(k)$ and $Q_s(k)$ are both stable operators but (3.9) is unstable. In fact, this can happen even when $Q_f(k)$ and $Q_s(k)$ are exact solution operators for well-posed hyperbolic problems, as the following example shows.

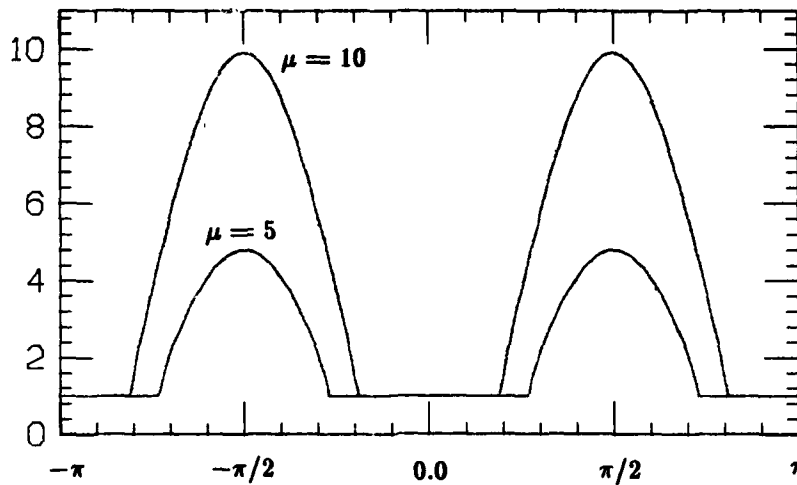


FIG. 3.1. Spectral radius of the amplification matrix $G(\xi, k)$ of Example 3.1 for $\mu = 5, 10$, as a function of $k\xi$ between $-\pi$ and π .

(Incidentally, the converse can also occur, i.e., the product may be stable even if one of the operators is unstable on its own. See Abarbanel & Gottlieb[1] for an example of such a scheme.)

Example 3.1. Let

$$A_f = \begin{bmatrix} 1 & \mu \\ 0 & -1 \end{bmatrix}, \quad A_s = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

Then the problems $u_t = A_f u_x$ and $u_t = A_s u_x$ are well-posed, strictly hyperbolic problems for any value of the parameter μ , and so is $u_t = (A_f + A_s)u_x$ if $\mu \geq -2$. Let

$$Q_f(k) = \exp(kA_f \partial_x), \quad Q_s(k) = \exp(kA_s \partial_x).$$

The corresponding amplification matrices are

$$\begin{aligned} G_f(\xi, k) &= \exp(ik\xi A_f) \\ &= \begin{bmatrix} e^{ik\xi} & \mu i \sin k\xi \\ 0 & e^{-ik\xi} \end{bmatrix} \end{aligned}$$

and

$$\begin{aligned} G_s(\xi, k) &= \exp(ik\xi A_s) \\ &= \begin{bmatrix} \cos k\xi & i \sin k\xi \\ i \sin k\xi & \cos k\xi \end{bmatrix}. \end{aligned}$$

We have $\rho(G_f(\xi, k)) = \rho(G_s(\xi, k)) = 1$ for all ξ and k . On the other hand, the amplification matrix $G(\xi, k)$ for the time-split method has $\rho(G(\xi, k)) = 1$ for all ξ and k only if $|\mu| \leq 2$. When $|\mu| > 2$, the method (3.9) is unstable. Figure 3.1 shows graphs of $\rho(G(\xi, k))$ for $\mu = 5$ and 10 .

3.3. Simultaneously normalizable splittings.

As we have just seen, the individual stability of $Q_f(k)$ and $Q_s(k)$ is not sufficient to guarantee the stability of $Q(k)$ in general. However, for certain special cases (which include some fairly broad and important classes of problems), the individual stability is sufficient for overall stability. Since the matrices $G_f(\xi, k)$ and $G_s(\xi, k)$ are generally much easier to work with than their product $G(\xi, k)$, it is useful to identify such classes of problems. For these problems stability is relatively easy to determine.

We first note that if there exists a norm $\|\cdot\|$ and a constant α such that

$$\|G_f(\xi, k)\| \leq 1 + \alpha k \quad \forall \xi, k < k_0 \quad (3.10a)$$

and

$$\|G_s(\xi, k)\| \leq 1 + \alpha k \quad \forall \xi, k < k_0. \quad (3.10b)$$

Then

$$\begin{aligned} \|G(\xi, k)\| &\leq \|G_f(\xi, k)\| \|G_s(\xi, k)\| \\ &\leq 1 + 2\alpha k + \alpha^2 k^2 \\ &\leq 1 + \alpha_0 k \quad \forall \xi, k < k_0 \end{aligned}$$

where $\alpha_0 = 2\alpha + \alpha^2 k_0$, so $Q(k)$ is stable.

Of course if $Q_f(k)$ is stable then by Stability Definition 3.2' there exists a norm such that (3.10a) holds. Similarly, if $Q_s(k)$ is stable then (3.10b) also holds in some (possibly different) norm. Only in certain special cases can we easily show the existence of a *single* norm in which both (3.10a) and (3.10b) hold.

As one such case, suppose that all of the matrices $G_f(\xi, k)$ and $G_s(\xi, k)$ are simultaneously normalizable by a single matrix S . Then the individual stability of $Q_f(k)$ and $Q_s(k)$ guarantees the stability of $Q(k)$, since then (3.10a) and (3.10b) both hold in the S -norm defined in (3.8).

Some operators, such as LW and exact solution operators, have the property that if the coefficient matrix is normal then the corresponding amplification matrix will also be normal, for all ξ and k . Restricting our attention to such schemes, we find that it then suffices for the stability of $Q(k)$ that the two matrices A_f and A_s be simultaneously normalizable and that $Q_f(k)$ and $Q_s(k)$ be individually stable.

This result is quite useful, since in many practical problems the matrices A_f and A_s are simultaneously normalizable. This class includes, for example, scalars, symmetric matrices, and commuting matrices (which are simultaneously diagonalizable).

These results can easily be extended to splittings involving more than two terms. Since this is frequently useful, we summarize the above results and their proofs in a more general setting.

THEOREM 3.2. *Let A_1, A_2, \dots, A_m be constant matrices. Approximate each solution operator $\exp(kA_j \partial_x)$ by some operator $Q_j(k)$ with amplification matrix $G_j(\xi, k)$. Suppose there exists a single norm $\|\cdot\|$ and a constant α such that*

$$\|G_j(\xi, k)\| \leq 1 + \alpha k \quad \forall \xi, k < k_0, j = 1, 2, \dots, m. \quad (3.11)$$

Then the scheme

$$U^{n+1} = Q_1(k_1)Q_2(k_2)\cdots Q_m(k_m)U^n \quad (3.12)$$

is stable.

Proof. Let $Q(k) = Q_1(k) \cdots Q_m(k)$ and let $G(\xi, k) = G_1(\xi, k) \cdots G_m(\xi, k)$ be the corresponding amplification matrix. Then

$$\begin{aligned} \|G(\xi, k)\| &\leq \|G_1(\xi, k)\| \cdots \|G_m(\xi, k)\| \\ &\leq 1 + \alpha_0 k \end{aligned}$$

for $\alpha_0 = m\alpha + \binom{m}{2}\alpha^2 k_0 + \cdots + \alpha^m k_0^{m-1}$ and hence $Q(k)$ is stable. ■

THEOREM 3.3. With the A_j and $G_j(\xi, k)$ as in Theorem 3.2, suppose there exists some nonsingular matrix S such that $SG_j(\xi, k)S^{-1}$ is a normal matrix for all j, ξ , and k . Suppose furthermore that each satisfies the von Neumann condition,

$$\rho(G_j(\xi, k)) \leq 1 + \alpha k \quad \forall \xi, k < k_0, j = 1, 2, \dots, m$$

for some constant α . Then $Q(k)$ is stable.

Proof. Using the S -norm defined in (3.8) and the fact that $SG_j(\xi, k)S^{-1}$ is normal, we have

$$\begin{aligned} \|G_j(\xi, k)\|_S &= \|SG_j(\xi, k)S^{-1}\|_2 \\ &= \rho(SG_j(\xi, k)S^{-1}) \\ &= \rho(G_j(\xi, k)) \\ &\leq 1 + \alpha k \end{aligned}$$

and stability follows by Theorem 3.2. ■

3.4. Block triangular systems.

A similar stability result can be obtained for the standard block triangular system

$$\begin{bmatrix} u \\ v \end{bmatrix}_t = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}_x$$

with the splitting (1.32). The solution v does not depend on u . In solving for u , the computed v_x enters essentially as a forcing function. Because of this we obtain only a weak stability result, in which the norm of $\|U^n\|$ is bounded in terms of a discrete Sobolev norm of the initial conditions. The Sobolev norm $\|\cdot\|$ has the form

$$\|U\| = \|U\| + \|D_+ U\|.$$

With the splitting (1.32), the schemes $Q_s(k)$ and $Q_f(k)$ will be of the form

$$Q_s(k) = \begin{bmatrix} I & Q_{12}(k) \\ 0 & Q_{22}(k) \end{bmatrix}, \quad Q_f(k) = \begin{bmatrix} Q_{11}(k) & 0 \\ 0 & I \end{bmatrix}. \quad (3.13)$$

Suppose that $Q_{11}(k)$ and $Q_{22}(k)$ are stable schemes. Then, in particular, there exists a norm $\|\cdot\|$ and a constant $\alpha \geq 0$ such that

$$\|Q_{11}(k)\| < 1 + \alpha k \quad \forall k < k_0. \quad (3.14)$$

All of the following estimates will be in this norm. We also suppose that

$$\|Q_{12}(k)V\| \leq kM\|D_+V\| \quad \forall V, k < k_0 \quad (3.15)$$

for some constant M . For example, if $Q_s(k) = LW(A_s, k)$, we have

$$\begin{aligned} Q_{12}(k) &= kA_{12}D_0 + \frac{1}{2}k^2A_{12}A_{22}D_+D_- \\ &= \frac{1}{2}kA_{12}(D_+ + D_-) + \frac{1}{2}k\lambda A_{12}A_{22}(D_+ - D_-) \end{aligned}$$

since $D_+D_- = (D_+ - D_-)/h$. Since $\|D_-V\| = \|D_+V\|$, we have

$$\|Q_{12}(k)V\| \leq k(\|A_{12}\| + \lambda\|A_{12}A_{22}\|)\|D_+V\|.$$

For fixed λ this is of the form (3.15).

With these assumptions we then have the following theorem.

THEOREM 3.4. *Suppose $Q_f(k)$ and $Q_s(k)$ are stable schemes as above. Then the split scheme $Q_s(k)Q_f(k)$ is weakly stable:*

$$\|U^n\| \leq K_T(\|U^0\| + \|D_+V^0\|) \quad (3.16a)$$

$$\|V^n\| \leq \tilde{K}_T\|V^0\| \quad (3.16b)$$

for $nk \leq T$. Here K_T and \tilde{K}_T are constants depending only on the fixed time T .

Proof. When the full scheme $\tilde{U}^{n+1} = Q_s(k)Q_f(k)\tilde{U}^n$ is written out we obtain

$$U^{n+1} = Q_{11}(k)U^n + Q_{11}(k)Q_{12}(k)V^n \quad (3.17a)$$

$$V^{n+1} = Q_{22}(k)V^n. \quad (3.17b)$$

The bound (3.16b) follows immediately from (3.17b) and the stability of $Q_{22}(k)$. Moreover, by linearity, an identical bound holds for the linear combination of solutions D_+V^n , i.e.,

$$\|D_+V^n\| \leq \tilde{K}_T\|D_+V^0\|.$$

Using this together with (3.15) in (3.17a) gives

$$\|U^{n+1}\| \leq \|Q_{11}(k)\|(\|U^n\| + kM\tilde{K}_T\|D_+V^0\|).$$

When iterated n times this gives

$$\begin{aligned} \|U^n\| &\leq \|Q_{11}(k)\|^n\|U^0\| + kM\tilde{K}_T \left(\|Q_{11}(k)\|^{n-1} + \|Q_{11}(k)\|^{n-2} \right. \\ &\quad \left. + \cdots + \|Q_{11}(k)\| + 1 \right) \|D_+V^0\|. \end{aligned} \quad (3.18)$$

By (3.14), $\|Q_{11}(k)\|^n \leq (1 + \alpha k)^n \leq e^{\alpha T}$ if $nk \leq T$. Using this in (3.18) gives

$$\|U^n\| \leq e^{\alpha T}(\|U^0\| + TM\tilde{K}_T\|D_+V^0\|)$$

for $nk \leq T$, which is of the desired form (3.16a). ■

3.5. The shallow water equations.

We will now investigate the stability of the splitting (1.38) for the shallow water equations. Since this is a quasilinear system of equations, a complete stability analysis is difficult to perform, even for unsplit methods. We will perform only a linearized stability analysis for the corresponding frozen coefficient problem with

$$A_f = -\begin{bmatrix} 0 & \phi_0/2 \\ \phi_0/2 & 0 \end{bmatrix}, \quad A_s = -\begin{bmatrix} U_0 & \Phi_0 \\ \Phi_0 & U_0 \end{bmatrix}. \quad (3.19)$$

Here the constant U_0 is a representative value of u while Φ_0 is a representative value of $(\phi - \phi_0)/2$. One hopes that if a method is stable on the frozen coefficient problem for all values of U_0 and Φ_0 in the appropriate range, then the method will also be stable on the nonlinear problem. It is well known that this is not necessarily so; *nonlinear instabilities* may arise. Nonetheless, the linearized stability analysis is valuable because an instability for the frozen coefficient problem will almost certainly lead to instability of the nonlinear problem, and thus we at least obtain upper bounds on the stability limit. Moreover, for the shallow water equations computations indicate that the nonlinear scheme is usually stable when the frozen coefficient problems are.

Stability of the scheme (2.42) applied to (3.19) is easy to determine using Theorem 3.3. The matrices A_f and A_s are both symmetric and so $Q(k)$ is stable provided $Q_f(k)$ and $Q_s(k)$ are both stable. Since $Q_f(k)$ is the exact solution operator, it is always stable, and so stability is determined entirely by $Q_s(k)$. The eigenvalues of A_s are $U_0 \pm \Phi_0$ and so $Q_s(k) = LW(A_s, k)$ is stable if $(U_0 \pm \Phi_0)k/h \leq 1$.

Suppose that (1.37) holds, i.e., $|u| \leq \epsilon\phi_0$ and $|\phi - \phi_0|/2 \leq \epsilon\phi_0$ for all x and t for the solutions of interest. Then all of the relevant frozen coefficient problems are stable provided

$$\frac{k}{h} \leq \frac{1}{2\epsilon\phi_0}. \quad (3.20)$$

Note that for the unsplit method $LW(A, k)$, the stability limit is roughly

$$\frac{k}{h} \leq \frac{2}{\phi_0}.$$

The split scheme is thus stable for much larger values of k . Recall, however, from Section 2.9 that for the split method an accurate solution is obtained most efficiently using $k/h \approx 1/\phi_0$. Such mesh ratios are well within the stability limit (3.20) and long-time calculations on the full nonlinear system have revealed no instabilities.

4. Boundary conditions for the intermediate solutions

4.1. Introduction and a simple example.

So far we have considered the time-split method applied only to the Cauchy problem on the unbounded spatial domain or to problems with periodic boundary conditions. In practice we must be able to deal with more general boundary conditions. The implementation of finite difference schemes frequently requires more boundary data than are supplied with the differential equation. In particular, when using a time-split method, special boundary data must be generated for the intermediate solutions.

For the most part we will restrict our attention to the time-split method (2.18a,b) for solving perturbed hyperbolic problems, although the same techniques can be applied to a wide variety of other problems and splittings. Some examples of other applications are given in Chapter 5.

We begin our discussion with a simple example which illustrates the problems encountered and the general methodology used to determine the correct boundary data.

A constant coefficient scalar problem. Consider the equation

$$u_t = -(1 + \epsilon)u_x \quad (4.1)$$

on the strip $0 \leq x \leq 1, t \geq 0$, with initial conditions

$$u(x, 0) = f(x), \quad 0 \leq x \leq 1, \quad (4.2)$$

and boundary conditions

$$u(0, t) = g(t), \quad t \geq 0. \quad (4.3)$$

For $\epsilon > -1$, this is a well-posed problem as it stands. Boundary data is prescribed only at the *inflow* boundary $x = 0$. Values at the *outflow* boundary $x = 1$ are determined as part of the solution.

The exact solution to this problem is a wave moving to the right, unaltered, with speed $1 + \epsilon$:

$$u(x, t) = f(x - (1 + \epsilon)t), \quad 0 \leq x \leq 1, t \geq 0$$

where for $\xi < 0$ we define

$$f(\xi) = g(-\xi/(1 + \epsilon)), \quad \xi < 0.$$

We will first consider the unsplit Lax-Wendroff method. If the mesh spacing in the x -direction is $h = 1/N$ for some integer N , then the grid points of interest are

x_0, x_1, \dots, x_N . The Lax-Wendroff method is

$$U_m^{n+1} = U_m^n - \frac{1}{2}\lambda(1+\epsilon)(U_{m+1}^n - U_{m-1}^n) + \frac{1}{2}\lambda^2(1+\epsilon)^2(U_{m+1}^n - 2U_m^n + U_{m-1}^n), \quad m = 1, 2, \dots, N-1. \quad (4.4)$$

This scheme cannot be applied for $m = 0$ or $m = N$ and so U_0^{n+1} and U_N^{n+1} must be determined in some different manner. At the left boundary we simply use the given boundary data (4.3),

$$U_0^{n+1} = g(t_{n+1}). \quad (4.5)$$

At the outflow boundary we must either extrapolate from the interior, e.g.,

$$U_N^{n+1} = 2U_{N-1}^{n+1} - U_{N-2}^{n+1}, \quad (4.6)$$

or use a one-sided difference scheme, e.g.,

$$U_N^{n+1} = U_N^n - \lambda(1+\epsilon)(U_N^n - U_{N-1}^n). \quad (4.7)$$

Both (4.6) and (4.7) have local truncation errors which are $O(k^2)$. This is sufficient to retain the $O(k^2)$ global accuracy of the Lax-Wendroff method. In general the overall accuracy of a method is not degraded by errors in the boundary values provided the local error at the boundary is no larger than the global error for the interior scheme (see Gustafsson[28]). In addition, of course, the total method (including boundary schemes) must be stable. Stability is more difficult to determine for initial boundary value problems than for Cauchy problems and is discussed in Section 4.6. For this simple problem both (4.6) and (4.7) yield stable methods.

Now consider a time-split method applied to the same problem (4.1) with

$$A_f = -1, \quad A_s = -\epsilon.$$

We now assume that $\epsilon \ll 1$. Since the operators commute, there is no need to use the Strang splitting and so we need introduce only one intermediate solution. Taking $k = ph$ for some integer $p \geq 1$ and using the exact solution operator on the fast part together with $LW(A_s, k)$, the split method is

$$U_m^* = U_{m-p}^n, \quad m = p, p+1, \dots, N+1, \quad (4.8a)$$

$$U_m^{n+1} = U_m^* - \frac{1}{2}p\epsilon(U_{m+1}^* - U_{m-1}^*) + \frac{1}{2}p^2\epsilon^2(U_{m+1}^* - 2U_m^* + U_{m-1}^*), \quad (4.8b) \\ m = 1, 2, \dots, N.$$

Notice that we use (4.8a) to define U_{N+1}^* even though it is not within the domain of interest. Nonetheless, it can be used in computing U_N^{n+1} (which is of interest) in the Lax-Wendroff step (4.8b). Because of this we do not need any special procedure to specify U_N^{n+1} . This is one advantage of using time-split methods for such perturbed problems. Since they are essentially skewed (one-sided) Lax-Wendroff methods which follow the characteristics of the problem, artificial boundary values are often not needed at outflow boundaries.

Instead, we need to specify additional values at $x = 0$. We still use (4.5) for U_0^{n+1} , but we must also specify $U_0^*, U_1^*, \dots, U_{p-1}^*$. Alternatively, we can leave these values unspecified and apply (4.8b) only for $m = p + 1, \dots, N$. We must then determine $U_1^{n+1}, \dots, U_p^{n+1}$ by some alternative procedure.

Since there is no splitting error for this problem, the results of Section 2.5 indicate that for optimal efficiency we should take $p \approx 1/\epsilon$. However, for simplicity we first consider the case $p = 1$. Then we only need to specify U_0^* or U_1^{n+1} .

Three possibilities for specifying U_1^{n+1} are immediately apparent. The first is to interpolate between the known values U_0^{n+1} and U_2^{n+1} ,

$$U_1^{n+1} = \frac{1}{2}(U_0^{n+1} + U_2^{n+1}). \quad (4.9)$$

This is $O(k^2)$ accurate. However, when ϵ is small this choice causes a severe loss of accuracy in (4.8) and completely negates the increase in efficiency obtainable through the use of the time-split method. The reason is that the local truncation error for the method (4.8) is $O(\epsilon k^3)$ giving $O(\epsilon k^2)$ global errors. It is this factor of ϵ which makes the time-split method advantageous over the unsplit method (4.4). By using (4.9) we lose this advantage.

Figures 4.1a,b show the errors at time $t = 0.4$ using this time-split method with the boundary conditions (4.9) when $\epsilon = 0.1$. Signals propagate with velocity $1 + \epsilon = 1.1$ and so errors from the improper specification of U_1^{n+1} have propagated in to approximately $x = 0.44$ at this time. To the right of this point all errors are due solely to the interior scheme. It is this accuracy which we would like to match at the boundary. Clearly the boundary approximation (4.9) is causing a loss of accuracy. When ϵ is smaller, as in Figures 4.1c,d where $\epsilon = 0.001$, this disparity in the size of the errors is even more apparent.

In order to maintain the advantage of the time-split method, we must use a more accurate boundary scheme, one with local error $O(\epsilon k^2)$. One possibility is to use higher order interpolation. Using quadratic interpolation on the points $U_0^{n+1}, U_2^{n+1}, U_3^{n+1}$ would give $O(k^3)$ errors. For k sufficiently small ($k < \epsilon$), this provides sufficiently accurate data. However, the use of higher order interpolation can cause stability problems. Moreover, when $p > 1$ there will be several values $U_1^{n+1}, \dots, U_p^{n+1}$ to be determined and interpolation is unsatisfactory.

The second obvious choice for U_1^{n+1} is to simply use Lax-Wendroff on the unsplit problem,

$$U_1^{n+1} = LW(-(1 + \epsilon), k)U_1^n. \quad (4.10)$$

This also has $O(k^3)$ local error and provides sufficiently accurate data for small k . Again, however, stability may be a problem and for $p > 1$ the scheme is certainly unstable.

The final approach to specifying U_1^{n+1} is based on Taylor series expansions in from the boundary. This is the best approach and, for this simple problem, gives the correct value of U_1^{n+1} exactly. We want U_1^{n+1} to approximate $u(h, t_{n+1})$. We can expand this in a Taylor series about $u(0, t_{n+1})$:

$$u(h, t_{n+1}) = u(0, t_{n+1}) + hu_x(0, t_{n+1}) + \frac{1}{2}h^2u_{xx}(0, t_{n+1}) + \dots \quad (4.11)$$

Approximating this directly by differencing the known values U_j^{n+1} would give us the interpolation scheme rejected above. However, using the differential equation (4.1) we

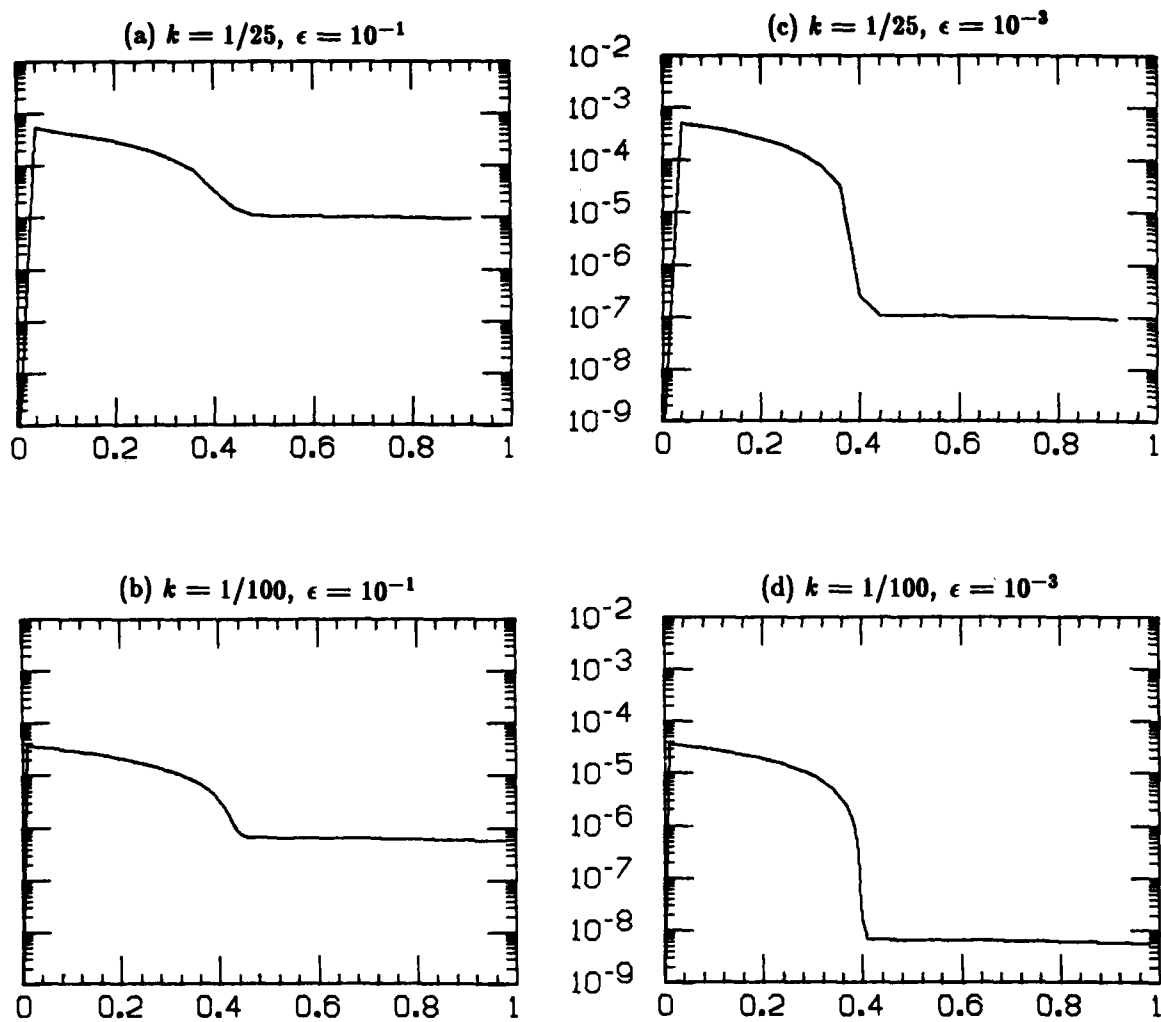


FIG. 4.1. Errors in the computed solution of (4.1) using the split scheme (4.8) with $p = 1$ and the interpolatory boundary condition (4.9). The errors are shown on a logarithmic scale for various values of k and ϵ . Note that the interior error is $O(\epsilon k^2)$ while the boundary error is $O(k^2)$.

and that

$$\partial_x^j u = \left(\frac{-1}{1+\epsilon} \right)^j \partial_t^j u, \quad j \geq 0, \quad (4.12)$$

We can thus rewrite (4.11) as

$$u(h, t_{n+1}) = u(0, t_{n+1}) - \frac{h}{1+\epsilon} u_t(0, t_{n+1}) + \frac{1}{2} \left(\frac{h}{1+\epsilon} \right)^2 u_{tt}(0, t_{n+1}) + \dots \quad (4.13)$$

The desired data is now expressed in terms of t -derivatives of u along the boundary, i.e., derivatives of the known function $g(t)$ from (4.3). For this simple problem (4.13) can in fact be evaluated in closed form, giving the desired value of U_1^{n+1} exactly:

$$U_1^{n+1} = g(t_{n+1} - h/(1+\epsilon)). \quad (4.14)$$

The calculations shown in Figure 4.1 have been repeated using (4.14) instead of (4.9). The results are shown in Figure 4.2. Since for this problem the boundary data (4.14) is exact, the errors are actually smaller near the boundary than in the interior.

This same approach can be used in a wide variety of problems to determine boundary data for points near the boundary. In general it will not be possible to obtain the exact data in closed form as in (4.14), but a series solution can be developed and evaluated to arbitrary accuracy. Goldberg & Tadmor[21][22] explain how to do this for general inflow-outflow boundaries. This will also be discussed in Section 4.4.

It seems that we have completely avoided the need to specify boundary values for the intermediate solution U^* . For this simple problem that is true. However, for many problems it is not possible to avoid specifying intermediate boundary values. This is particularly true when implicit methods are used in the splitting. In other situations it is simply more convenient computationally to specify boundary values for the intermediate solution than to leave these points unspecified.

The remainder of this chapter is devoted to showing how, for many problems, the same approach used above to compute U_1^{n+1} can be extended to compute arbitrarily accurate intermediate boundary data.

Computing U_0^* . We now return to our original plan to specify U_0^* for the scalar problem (4.1). We require data at the point x_0 , which is on the inflow boundary. At this boundary the data (4.3) has been supplied, but is not usable directly since U^* is obtained not by solving the original equation but rather by solving the subproblem $u_t^* = -u_x^*$. This is the fundamental step in correctly computing intermediate boundary data: *introduce a new function u^* which solves the differential equation actually being approximated in the relevant step of the splitting.* The desired boundary data can then be expanded as a Taylor series in this function. In many cases this can be reexpressed as a series in the original variable and evaluated in terms of $g(t)$ as before. We will see that for many problems it is possible to generate stable $O(\epsilon k^2)$ boundary data quite easily. For the problem (4.1) we can in fact generate boundary data which is exactly correct, just as we did for U_1^{n+1} .

Consider a single step (4.8a) of the time-split method starting at time t_n and suppose that $U_m^n = u(x_m, t_n)$. Since in this problem we have used the exact solution operator $\exp(kA_f \partial_x)$ in (4.8a), U^* is then the exact solution at time t_{n+1} to the subproblem

$$u_t^* = -u_x^* \quad x \geq 0, \quad t \geq t_n, \quad (4.15)$$

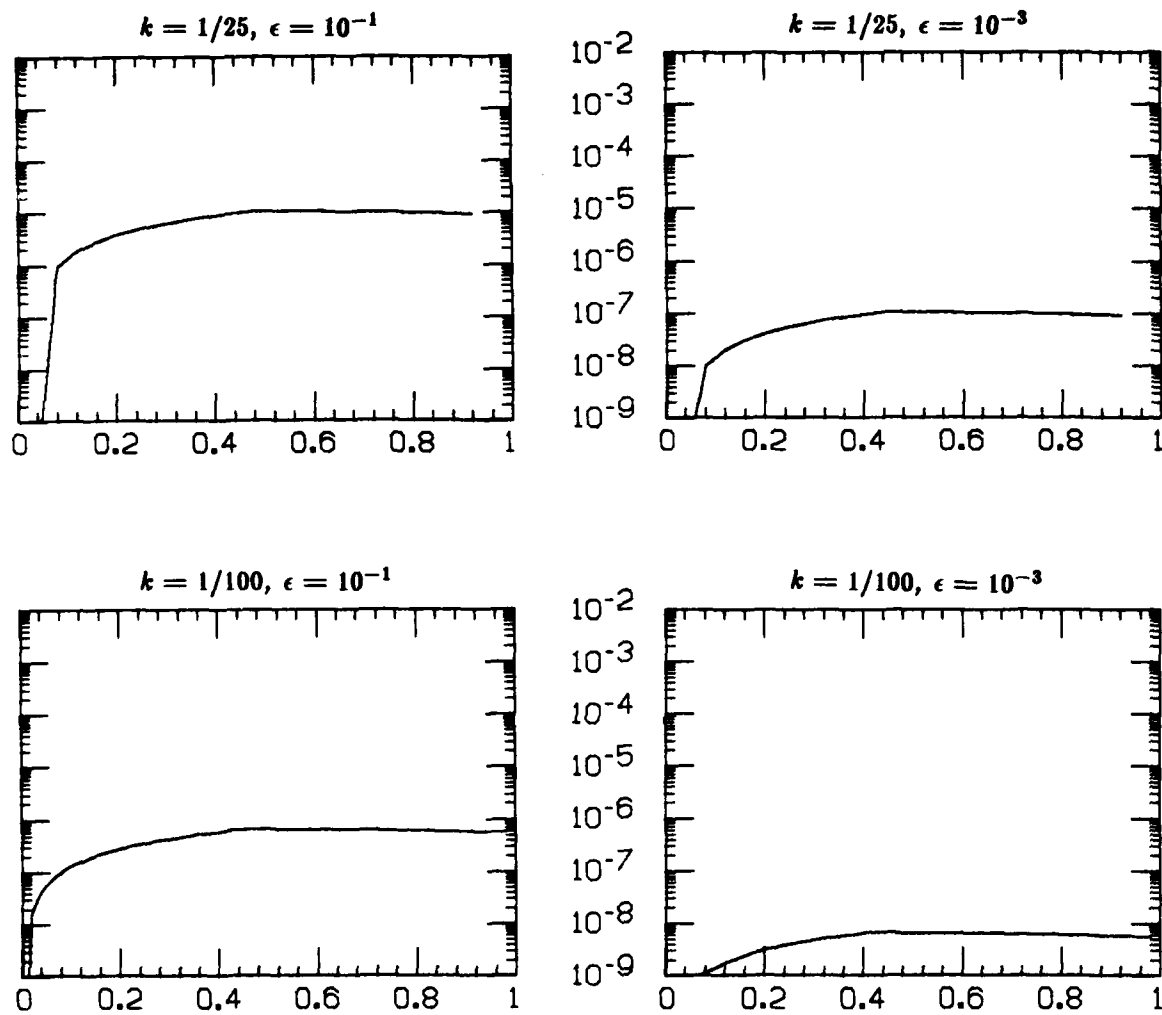


FIG. 4.2. Errors in the computed solution of (4.1) using the split scheme (4.8) with $p = 1$ and the correct boundary condition (4.14). The errors are shown on a logarithmic scale for various values of k and ϵ .

with initial conditions

$$u^*(x, t_n) = u(x, t_n), \quad x \geq 0 \quad (4.16)$$

at time t_n . The idea is to use the differential equations (4.1) and (4.15) to transform the given boundary conditions (4.3) for u into boundary conditions for u^* . We wish to find an appropriate value for U_0^* , which should be an approximation to $u^*(0, t_{n+1})$. This we can expand in a Taylor series. Using (4.15), we find that

$$\begin{aligned} u^*(0, t_n + k) &= u^*(0, t_n) + ku_t^*(0, t_n) + \frac{1}{2}k^2 u_{tt}^*(0, t_n) + \dots \\ &= u^*(0, t_n) - ku_x^*(0, t_n) + \frac{1}{2}k^2 u_{xx}^*(0, t_n) + \dots \end{aligned} \quad (4.17)$$

Since the initial conditions (4.16) hold for all x , that relation can be differentiated with respect to x , giving $u_x^*(x, t_n) = u_x(x, t_n)$ and similarly for higher derivatives. So (4.17) becomes

$$u^*(0, t_n + k) = u(0, t_n) - ku_x(0, t_n) + \frac{1}{2}k^2 u_{xx}(0, t_n) + \dots \quad (4.18)$$

We can now use the original equation (4.1) governing u to rewrite this in terms of t -derivatives of u . Using (4.12), (4.18) becomes

$$\begin{aligned} u^*(0, t_n + k) &= u(0, t_n) + \frac{k}{1+\epsilon} u_t(0, t_n) + \frac{1}{2} \left(\frac{k}{1+\epsilon} \right)^2 u_{tt}(0, t_n) + \dots \\ &= g(t_n + k/(1+\epsilon)). \end{aligned} \quad (4.19)$$

This is the desired boundary data U_0^* , expressed in terms of the given boundary data (4.3).

For such a simple example it is easy to verify that this is the correct boundary value. According to the scheme (4.8a) we would really like

$$U_0^* = U_{-1}^n = u(-h, t_n).$$

(Recall that $p = 1$.) Of course u is not really defined for $x < 0$, but using the differential equation (4.1) it can easily be extended backwards in time from the boundary. Since (4.1) has characteristics with slope $1/(1+\epsilon)$, we find that

$$u(-h, t_n) = u(0, t_n + h/(1+\epsilon)) = g(t_n + k/(1+\epsilon))$$

exactly as in (4.19).

Because the characteristics for the problems (4.1) and (4.15) have different slopes, we see that the value $u(-h, t_n)$ is equal to both $u(0, t_n + k/(1+\epsilon))$ and $u^*(0, t_n + k)$ and therefore they are equal to each other. This is illustrated in Figure 4.3.

When $p > 1$ we can compute U_j^* for $0 < j < p$ in a similar manner. Using the fact that we know the exact solution operator for the subproblem (4.15), we can project these values back to the boundary along the characteristics,

$$\begin{aligned} U_j^* &= u^*(jh, t_n + k) \\ &= u^*(0, t_n + k - jh). \end{aligned}$$

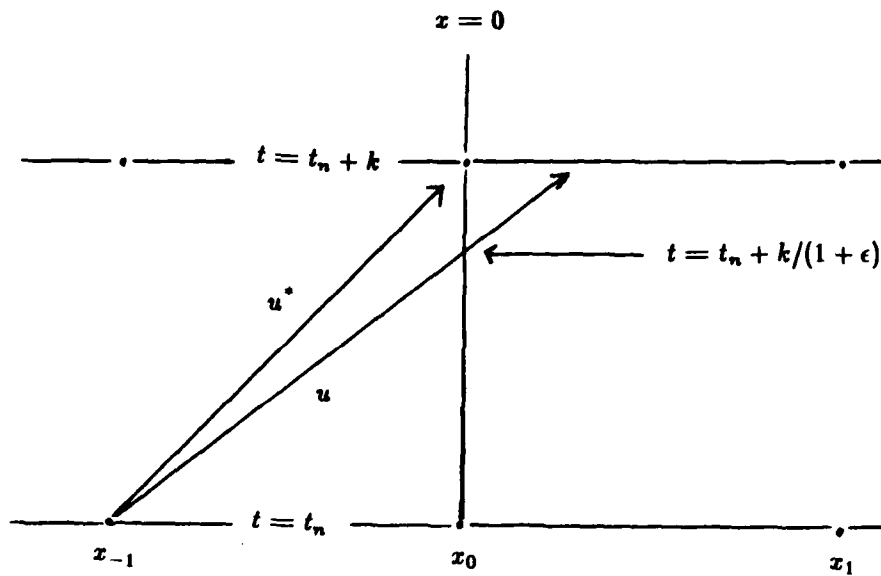


FIG. 4.3. Characteristics for $u_t^* = -u_x^*$ and $u_t = -(1 + \epsilon)u_x$ (for $\epsilon > 0$) showing that if $u^*(x, t_n) \equiv u(x, t_n)$ then $u^*(0, t_n + k) = u(0, t_n + k/(1 + \epsilon))$.

This boundary value can be computed as before, giving the general expression

$$U_j^* = g(t_n + (k - jh)/(1 + \epsilon)), \quad j = 0, 1, \dots, p-1. \quad (4.20)$$

Computations confirm that the use of this boundary value for U_0^* in the split method (4.8) gives excellent results that are virtually identical to those seen in Figure 4.2.

In general when using this approach to specify boundary conditions for the intermediate solutions it will not be possible to generate exact boundary data as we did here. It often will be possible, however, to develop a series solution, as in the first line of (4.19), which can be used to generate arbitrarily accurate boundary data. In the next few sections we demonstrate how this can be done for systems of increasing complexity, culminating in Section 4.5 with the development of boundary conditions for the shallow water equations, a quasilinear system of equations with inflow-outflow boundaries.

4.2. Constant coefficient systems—inflow boundaries.

As the next step in this direction, consider a constant coefficient system of equations

$$\begin{aligned} u_t &= \Lambda u_x \equiv (\Lambda_f + \Lambda_g)u_x, & x \geq 0, \quad t \geq 0, \\ u(x, 0) &= f(x), \\ u(0, t) &= g(t), & t \geq 0, \end{aligned} \quad (4.21)$$

on the quarter plane $x, t \geq 0$. We assume that the boundary $x = 0$ is a pure inflow boundary, i.e., that A has strictly negative eigenvalues. We also assume that A_f has nonpositive eigenvalues. In general A_f and A_s do not commute, so we will have to use a Strang-type splitting. There will be at least two intermediate solutions, say

$$\begin{aligned} U^* &\approx \exp(\tfrac{1}{2}kA_f\partial_x)U^n \\ U^{**} &\approx \exp(kA_s\partial_x)\exp(\tfrac{1}{2}kA_f\partial_x)U^n. \end{aligned} \quad (4.22)$$

Of course there may be many more if $\exp(\tfrac{1}{2}kA_f\partial_x)$ is itself approximated by several steps of Lax-Wendroff, but they can be handled similarly. The general principle should be clear from considering (4.22).

Again introduce the function $u^*(x, t)$ which satisfies the first subproblem of interest,

$$u_t^* = A_f u_x^*, \quad x \geq 0, \quad t \geq t_n, \quad (4.23a)$$

$$u^*(x, t_n) = u(x, t_n), \quad x \geq 0. \quad (4.23b)$$

We then want

$$\begin{aligned} U_0^* &= u^*(0, t_{n+1/2}) \\ &= u^*(0, t_n) + \tfrac{1}{2}k u_t^*(0, t_n) + \tfrac{1}{8}k^2 u_{tt}^*(0, t_n) + \dots \\ &= u^*(0, t_n) + \tfrac{1}{2}k A_f u_x^*(0, t_n) + \tfrac{1}{8}k^2 A_f^2 u_{xx}^*(0, t_n) + \dots \\ &= u(0, t_n) + \tfrac{1}{2}k A_f u_x(0, t_n) + \tfrac{1}{8}k^2 A_f^2 u_{xx}(0, t_n) + \dots \end{aligned} \quad (4.24)$$

where we have used (4.23a) to replace t -derivatives of u^* by x -derivatives. These were then replaced with x -derivatives of u using the initial conditions (4.23b). We next use the original equation (4.21) to replace x -derivatives of u by t -derivatives, which are equivalent to derivatives of the boundary data,

$$\begin{aligned} U_0^* &= u(0, t_n) + \tfrac{1}{2}k A_f A^{-1} u_t(0, t_n) + \tfrac{1}{8}k^2 A_f^2 A^{-2} u_{tt}(0, t_n) + \dots \\ &= g(t_n) + \tfrac{1}{2}k A_f A^{-1} g'(t_n) + \tfrac{1}{8}k^2 A_f^2 A^{-2} g''(t_n) + \dots \end{aligned} \quad (4.25)$$

We have assumed that A has strictly negative eigenvalues and thus is invertible. In general U_0^* must now be approximated by the first few terms of (4.25). Keeping the first three terms gives $O(k^3)$ accurate boundary data. As usual, this is sufficiently accurate if k is small. However, it is worth pointing out that we can frequently achieve the $O(\epsilon k^2)$ accuracy we desire more easily. Suppose $\|A^{-1}\| = O(\epsilon)$. Then since $A = A_f + O(\epsilon)$,

$$A_f^j A^{-j} = I + O(\epsilon) \quad \text{for } j = 1, 2, \dots$$

We can then retain $O(\epsilon k^2)$ accuracy simply by taking

$$U_0^* = g(t_{n+1/2}) + \tfrac{1}{2}k(A_f A^{-1} - I)g'(t_n). \quad (4.26)$$

We may still wish to use additional terms of the expansion in order to ensure that the error from the boundary conditions does not dominate the interior error. The boundary

conditions (4.26) are the correct order of accuracy but the error constant may be larger than that of the interior scheme. We obtain $O(\epsilon k^3)$ accurate boundary data by using

$$U_0^* = g(t_{n+1/2}) + \frac{1}{2}k(A_f A^{-1} - I)g'(t_n) + \frac{1}{8}k^2(A_f^2 A^{-2} - I)g''(t_n). \quad (4.27)$$

The additional work incurred by using three terms of the expansion rather than two at the boundary is negligible compared to the work being done in the interior.

Now to find boundary values for U^{**} . The easiest way to proceed is to note that

$$U^{**} = \exp(-\frac{1}{2}kA_f \partial_x)U^{n+1}$$

which prompts us to define $u^{**}(x, t)$ as the continuous solution to

$$\begin{aligned} u_t^{**}(x, t) &= A_f u_x^{**}(x, t) & x \geq 0, \quad t \leq t_{n+1} \\ u^{**}(x, t_{n+1}) &= u(x, t_{n+1}) & x \geq 0. \end{aligned} \quad (4.28)$$

We now solve this backwards in time for

$$U_0^{**} = u^{**}(0, t_{n+1/2}).$$

Proceeding as in (4.24) and (4.25) we obtain

$$\begin{aligned} U_0^{**} &= g(t_{n+1}) - \frac{1}{2}kA_f A^{-1}g'(t_{n+1}) + \frac{1}{8}k^2 A_f^2 A^{-2}g''(t_{n+1}) + \dots \\ &\approx g(t_{n+1/2}) - \frac{1}{2}k(A_f A^{-1} - I)g'(t_{n+1}). \end{aligned}$$

Example 4.1 Consider

$$\begin{aligned} \begin{bmatrix} u \\ v \end{bmatrix}_t &= \begin{bmatrix} -1 & \epsilon_1 \\ \epsilon_2 & -2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}_x & 0 \leq x \leq 1, \quad t \geq 0, \\ \tilde{u}(x, 0) &= f(x), & 0 \leq x \leq 1, \\ \tilde{u}(0, t) &= g(t), & t \geq 0, \end{aligned}$$

where $\tilde{u} = (u, v)^T$. For the splitting we take

$$A_f = \begin{bmatrix} -1 & 0 \\ 0 & -2 \end{bmatrix}, \quad A_s = \begin{bmatrix} 0 & \epsilon_1 \\ \epsilon_2 & 0 \end{bmatrix}.$$

Using (2.22) the splitting error is computed to be

$$V_{\text{split}}(k) = -\frac{1}{8}k^3 \begin{bmatrix} -\epsilon_1 \epsilon_2 & \frac{1}{4}\epsilon_1 \\ \frac{1}{4}\epsilon_2 & \epsilon_1 \epsilon_2 \end{bmatrix} \partial_x^3.$$

If we use the time-split method (2.18a,b) then, according to (2.24), the optimal stepsize ratio is

$$\lambda \approx \sqrt{\frac{\epsilon}{\frac{1}{4}\epsilon + \epsilon^3}} \approx 2$$

where $\epsilon = \max |\epsilon_j|$. For $k = 2h$ and $h = 1/N$, (2.18a,b) becomes

$$\begin{aligned} U_m^* &= U_{m-1}^n, \quad m = 1, 2, \dots, N \\ V_m^* &= V_{m-2}^n, \quad m = 2, 3, \dots, N \\ \tilde{U}_m^{**} &= LW(\Lambda_s, k) \tilde{U}_m^*, \quad m = 1, 2, \dots, N-1 \\ \tilde{U}_0^{n+1} &= g(t_{n+1}) \\ U_m^{n+1} &= U_{m-1}^{**}, \quad m = 1, 2, \dots, N \\ V_m^{n+1} &= V_{m-2}^{**}, \quad m = 2, 3, \dots, N. \end{aligned}$$

Notice that no boundary conditions need to be specified at the outflow boundary $x = 1$. On the inflow side we still need to specify \tilde{U}_0^* , V_1^* , \tilde{U}_0^{**} , and V_1^{n+1} . For this problem,

$$\begin{aligned} \Lambda_f^2 A^{-2} &= \frac{1}{(2 - \epsilon_1 \epsilon_2)^2} \begin{bmatrix} 4 + \epsilon_1 \epsilon_2 & 3\epsilon_1 \\ 12\epsilon_2 & 4 + 4\epsilon_1 \epsilon_2 \end{bmatrix} \\ &= I + O(\epsilon). \end{aligned}$$

and we can retain $O(\epsilon k^2)$ accuracy taking

$$\begin{aligned} \tilde{U}_0^* &= g(t_{n+1/2}) + \frac{1}{2} k (\Lambda_f A^{-1} - I) g'(t_n) \\ &= g(t_{n+1/2}) + \frac{k}{2(2 - \epsilon_1 \epsilon_2)} \begin{bmatrix} \epsilon_1 \epsilon_2 & \epsilon_1 \\ 2\epsilon_2 & \epsilon_1 \epsilon_2 \end{bmatrix} g'(t_n). \end{aligned}$$

Similarly we use

$$\tilde{U}_0^{**} = g(t_{n+1/2}) - \frac{1}{2} k (\Lambda_f A^{-1} - I) g'(t_{n+1}).$$

We still need to determine V_1^* and V_1^{n+1} . We want $V_1^* = v^*(h, t_{n+1/2}) = v^*(0, t_{n+1/4})$ and so the appropriate value comes from the second equation of

$$\tilde{u}^*(0, t_{n+1/4}) \approx g(t_{n+1/4}) + \frac{1}{4} k (\Lambda_f A^{-1} - I) g'(t_n),$$

i.e.,

$$V_1^* = g_2(t_{n+1/4}) + \frac{k}{4(2 - \epsilon_1 \epsilon_2)} (2\epsilon_2 g_1'(t_n) + \epsilon_1 \epsilon_2 g_2'(t_n)),$$

where $g = (g_1, g_2)^T$. Similarly,

$$V_1^{n+1} = g_2(t_{n+3/4}) - \frac{k}{4(2 - \epsilon_1 \epsilon_2)} (2\epsilon_2 g_1'(t_{n+1}) + \epsilon_1 \epsilon_2 g_2'(t_{n+1})).$$

Computations confirm that these boundary conditions give an $O(\epsilon k^2)$ globally accurate split scheme. Actually, for this particular example with $k = 2h$, even greater accuracy can be achieved. Computing $E_s(k)$ from (1.13), the truncation error for Lax-Wendroff, shows that the $O(\epsilon k^3)$ terms exactly cancel the $O(\epsilon k^3)$ terms in $E_{\text{split}}(k)$, and that the total truncation error $E^{\text{Lax-W}}(k)u$ is actually $O(\epsilon^2 k^3)$, giving $O(\epsilon^2 k^2)$ global accuracy. By retaining more terms in the above boundary expansions we can match the

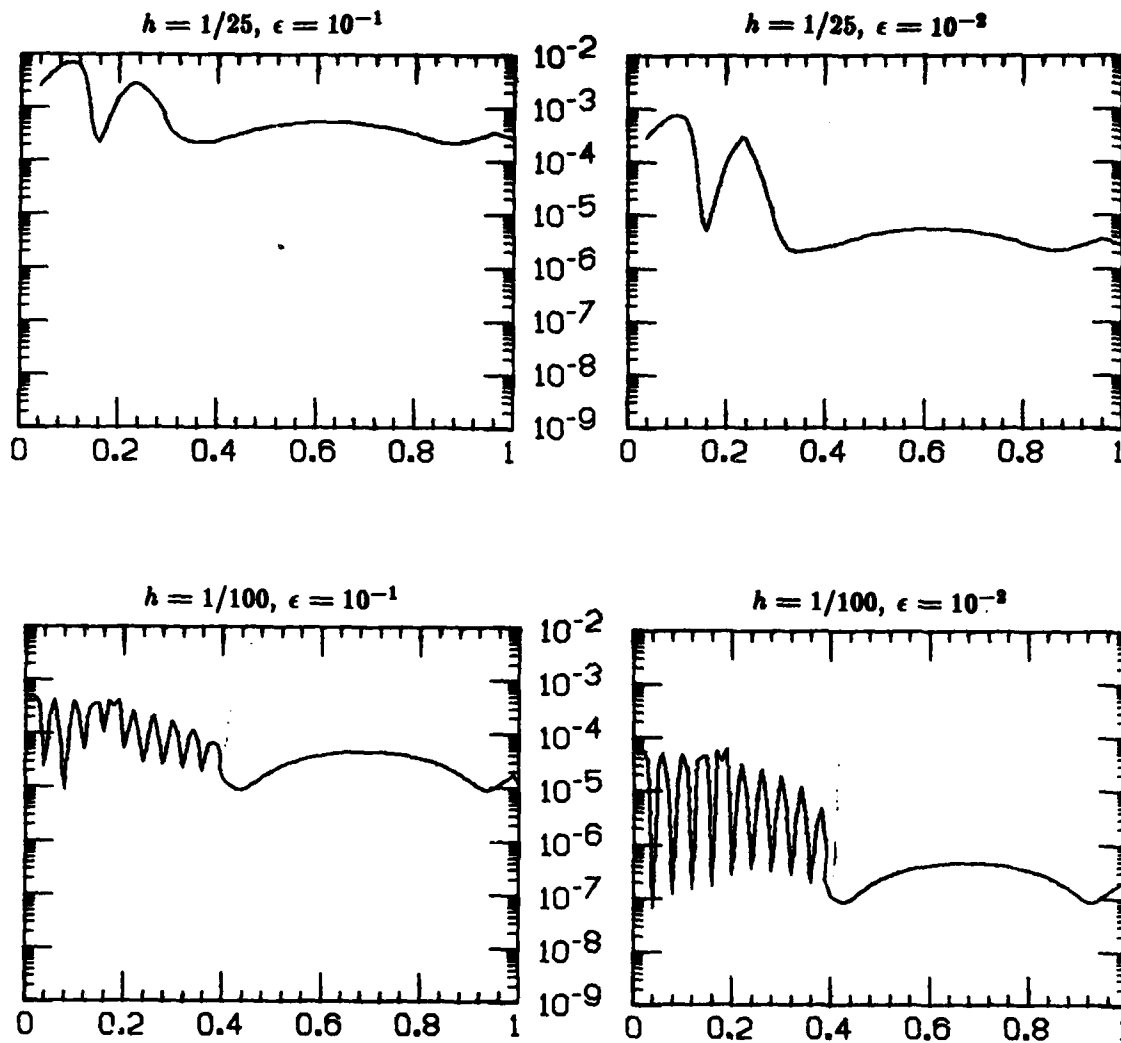


FIG. 4.4. Errors in the computed solution U of Example 4.1 using the $O(\epsilon k^2)$ boundary conditions. The interior error is $O(\epsilon^2 k^2)$. The errors are shown on a logarithmic scale for various values of k and ϵ . In all cases $t = 0.2$.

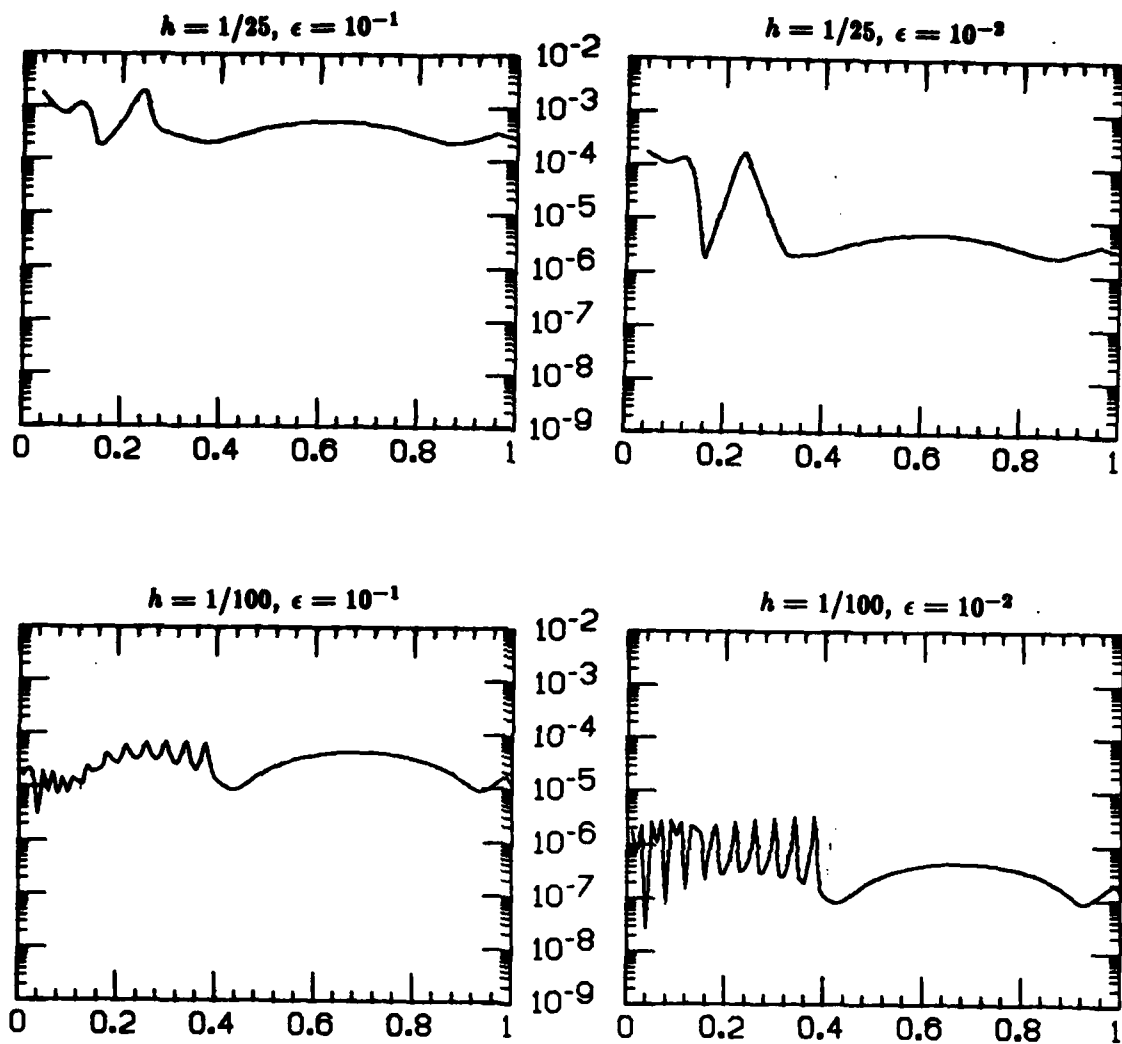


FIG. 4.5. Errors in the computed solution U of Example 4.1 using the $O(\epsilon k^3)$ boundary conditions. The interior error is $O(\epsilon^2 k^2)$. The errors are shown on a logarithmic scale for various values of k and ϵ . In all cases $t = 0.2$.

error in the interior solution. Taking one more term, as in (4.27), gives $O(\epsilon k^3)$ boundary data. Figures 4.4 and 4.5 show the some sample results using $O(\epsilon k^2)$ and $O(\epsilon k^3)$ accurate boundary data respectively. Errors in the first component U are shown at time $t = 0.2$. Errors resulting from the boundary conditions have propagated in to approximately $x = 0.4$.

The oscillations in the error near the boundary are due to the fact that some of the boundary conditions used (e.g., for \tilde{U}_0^{n+1}) have zero error while others (e.g., for \tilde{U}_0^*) have large errors. Since the split scheme is only mildly dissipative due to the $O(\epsilon)$ coefficients in the Lax-Wendroff step, these oscillations introduced at the boundary die out very slowly as the wave propagates into the interior. This is in no way an indication of instability. Stability for this example follows from the general results of Section 4.8.

4.3. Variable coefficient systems—inflow boundaries.

Defining the proper boundary data for variable coefficient problems is not significantly more difficult than for constant coefficient problems. The only complication comes in switching between x - and t -derivatives. Consider the system of equations

$$u_t = A(x, t)u_x \quad (4.29)$$

and for simplicity suppose that A_f is constant, while $A_s = A_s(x, t)$. Proceeding as in (4.24),

$$u^*(0, t_n + k/2) = u(0, t_n) + \frac{1}{2}kA_f u_x(0, t_n) + \frac{1}{8}k^2 A_f^2 u_{xx}(0, t_n) + \dots$$

Now we must be more careful in switching back to t -derivatives. We have

$$u_x(0, t_n) = A^{-1}(0, t_n)u_t(0, t_n) \quad (4.30)$$

and by differentiating (4.29) we find that

$$\begin{aligned} u_{tt} &= A_t u_x + A u_{xx} \\ u_{tx} &= A_x u_x + A u_{xx} \end{aligned}$$

so that

$$u_{xx} = A^{-1}[A^{-1}(u_{tt} - A_t A^{-1}u_t) - A_x A^{-1}u_t].$$

Higher order derivatives can be computed similarly. Continuing as in (4.25), we obtain

$$\begin{aligned} u^*(0, t_n) &= g(t_n) + \frac{1}{2}kA_f A^{-1}(0, t_n)g'(t_n) + \frac{1}{8}k^2 A_f^2 A^{-1}(0, t_n)[A^{-1}(0, t_n)g''(t_n) \\ &\quad - (A^{-1}(0, t_n)A_t(0, t_n) + A_x(0, t_n))A^{-1}(0, t_n)g'(t_n)] + O(k^3). \end{aligned} \quad (4.31)$$

This can be truncated in the usual manner to obtain an appropriate expression for U_0^* .

Example 4.3. Consider the standard quarter plane problem for the scalar equation

$$u_t = -(1 + \epsilon\alpha(x))u_x$$

with $A_f = -1$, $A_s = -\epsilon\alpha(x)$ and $\epsilon \ll 1$, $|\alpha(x)| \leq 1$. Then (4.31) gives

$$u^*(0, t_n) = g(t_n) + \frac{1}{2}k \left(\frac{1}{1 + \epsilon\alpha(0)} \right) g'(t_n) + \frac{1}{8}k^2 \left(\frac{1}{1 + \epsilon\alpha(0)} \right)^2 \times g''(t_n) + \epsilon\alpha'(0)g'(t_n) + O(k^3).$$

We thus find that the boundary condition

$$U_0^* = g(t_n + k/2(1 + \epsilon\alpha(0)))$$

is $O(\epsilon k^2)$ accurate. By retaining the next term of the expansion as well we obtain the $O(\epsilon k^3)$ accurate boundary data

$$U_0^* = g(t_n + k/2(1 + \epsilon\alpha(0))) + \frac{1}{8}k^2 \left(\frac{\epsilon\alpha'(0)}{(1 + \epsilon\alpha(0))^2} \right) g'(t_n).$$

The other necessary boundary data can be generated in a similar manner.

4.4. Inflow-outflow boundaries.

Next we consider a constant coefficient problem $u_t = Au_x$ for $x, t \geq 0$ with an inflow-outflow boundary at $x = 0$. This means that A has both positive and negative eigenvalues. For simplicity we suppose that A is in block diagonal form,

$$A = \begin{bmatrix} A^I & 0 \\ 0 & A^{II} \end{bmatrix}, \quad (4.32)$$

with the eigenvalues of A^I negative and those of A^{II} positive. Partition $\bar{u} = (u, v)^T$ conformally with A . Then at $x = 0$ the elements of u are inflow variables while those of v are outflow variables. The boundary conditions are assumed to be of the form

$$u(0, t) = Sv(0, t) + g(t) \quad (4.33)$$

where S is a constant matrix and g is a given function. We now split A as $A = A_f + A_s$ with A_f and A_s again block diagonal. Moreover we suppose that the eigenvalues of A_f^I are negative and those of A_f^{II} positive.

We consider only the problem of computing \bar{U}_0^* and will suppose that $\exp(kA_f\partial_x)$ is known exactly. Then V_0^* is determined from the interior and we need only specify U_0^* . As usual, we introduce $\bar{u}^*(x, t)$ which solves the subproblem $\bar{u}_t^* = A_f\bar{u}_x^*$ and find as in (4.24) and (4.25), that

$$\bar{u}^*(0, t_n + k/2) = \bar{u}(0, t_n) + \frac{1}{2}kA_f\Lambda^{-1}\bar{u}_t(0, t_n) + \frac{1}{8}k^2\Lambda_f^2\Lambda^{-2}\bar{u}_{tt}(0, t_n) + \dots$$

For simplicity, suppose that $\Lambda_f^2\Lambda^{-2} = I + O(\epsilon)$. Then for $O(\epsilon k^2)$ accurate boundary conditions we can take

$$\bar{U}_0^* = \bar{u}(0, t_{n+1/2}) + \frac{1}{2}k(\Lambda_f\Lambda^{-1} - I)\bar{u}_t(0, t_n). \quad (4.34)$$

Introducing the matrix

$$B = A_f A^{-1} - I = \begin{bmatrix} A_f^I (A^I)^{-1} - I & 0 \\ 0 & A_f^{II} (A^{II})^{-1} - I \end{bmatrix} = \begin{bmatrix} B_{11} & 0 \\ 0 & B_{22} \end{bmatrix},$$

we can rewrite (4.34) as

$$U_0^* = u(0, t_{n+1/2}) + \frac{1}{2} k B_{11} u_t(0, t_n) \quad (4.35a)$$

$$V_0^* = v(0, t_{n+1/2}) + \frac{1}{2} k B_{22} v_t(0, t_n). \quad (4.35b)$$

By differentiating the boundary conditions (4.33) we obtain

$$u_t(0, t_n) = S v_t(0, t_n) + g'(t_n).$$

Using this and (4.33), (4.35a) becomes

$$U_0^* = [S v(0, t_{n+1/2}) + g(t_{n+1/2}) + \frac{1}{2} k B_{11} [S v_t(0, t_n) + g'(t_n)]]. \quad (4.36)$$

Recall that V_0^* is already known. We can thus solve (4.35b) for $v(0, t_{n+1/2})$. Using this in (4.36) yields

$$\begin{aligned} U_0^* &= S[V_0^* - \frac{1}{2} k B_{22} v_t] + g(t_{n+1/2}) + \frac{1}{2} k B_{11} [S v_t(0, t_n) + g'(t_n)] \\ &= S V_0^* + g(t_{n+1/2}) + \frac{1}{2} k [B_{11} g'(t_n) + (B_{11} S - S B_{22}) v_t(0, t_n)]. \end{aligned} \quad (4.37)$$

The v_t term must in general be approximated by a finite difference, e.g.,

$$\begin{aligned} U_0^* &= S V_0^* + g(t_{n+1/2}) + \frac{1}{2} k [B_{11} g'(t_n) \\ &\quad + (B_{11} S - S B_{22})(V_0^n - V_0^{n-1})]. \end{aligned} \quad (4.38)$$

Alternatively we can replace v_t by $A^{II} v_x$ and approximate this by a finite difference of V at time t_n . This approach is particularly useful when more terms of the series are kept and higher order derivatives must be approximated.

The use of such boundary conditions is illustrated in the next section, where the one-dimensional shallow water equations are considered.

Boundary data at points near the boundary can be found in a similar manner. For example, if data U_j^{n+1} is needed for some $0 < j \leq p$ we can expand u in x -derivatives, switch to t -derivatives along the boundary, convert these to t -derivatives of v using $u_t = S v_t + g'$, and finally switch back to x -derivatives of v , obtaining

$$\begin{aligned} u(j, t_{n+1}) &= u(0, t_{n+1}) + j h (A^I)^{-1} [S A^{II} v_x(0, t_{n+1}) + g'(t_{n+1})] \\ &\quad + \frac{1}{2} (j h)^2 (A^I)^{-2} [S (A^{II})^2 v_{xx}(0, t_{n+1}) + g''(t_{n+1})] + \dots \end{aligned} \quad (4.39)$$

These boundary conditions are suggested by Goldberg & Tadmor[21][22] for general inflow-outflow problems.

The shallow water equations in order to illustrate the derivation of intermediate boundary conditions for a more realistic example, we will consider the one-dimensional shallow water equations on a strip,

$$\begin{bmatrix} u \\ \phi \end{bmatrix}_t = - \begin{bmatrix} u & \phi/2 \\ \phi/2 & u \end{bmatrix} \begin{bmatrix} u \\ \phi \end{bmatrix}_x \quad 0 \leq x \leq 1, t \geq 0, \quad (4.40)$$

with initial conditions

$$\bar{u}(x, 0) = f(x), \quad 0 \leq x \leq 1, \quad (4.41)$$

and, for example, the boundary conditions

$$\phi(0, t) = g(t), \quad (4.42a)$$

$$u(1, t) = \phi(1, t) - \phi_0. \quad (4.42b)$$

Here ϕ_0 is the mean value of ϕ as in Section 2.9 and the boundary condition (4.42b) represents nonreflection at the boundary $x = 1$. At the boundary $x = 0$ we have chosen to prescribe ϕ . Other boundary conditions can be handled similarly.

As in Section 2.9, the equations (4.40) can be written in the characteristic form (2.46). As usual we suppose that $|u| \ll |\phi|$. Then the Riemann invariant $u + \phi$ always flows to the right with velocity $\phi/2 + u$ while the Riemann invariant $u - \phi$ always flows to the left with velocity $\phi/2 - u$.

The problem of specifying intermediate boundary conditions is simplified if we change variables and compute directly in terms of the characteristic variables, which we denote by ρ and σ :

$$\rho(x, t) = u(x, t) + \phi(x, t),$$

$$\sigma(x, t) = u(x, t) - \phi(x, t).$$

We can always transform back to find $u = (\rho + \sigma)/2$ and $\phi = (\rho - \sigma)/2$. Rewriting the differential equation (4.40) in terms of ρ and σ gives

$$\begin{bmatrix} \rho \\ \sigma \end{bmatrix}_t = - \frac{1}{4} \begin{bmatrix} 3\rho + \sigma & 0 \\ 0 & \rho + 3\sigma \end{bmatrix} \begin{bmatrix} \rho \\ \sigma \end{bmatrix}_x \quad (4.42)$$

which we split as in (2.48) by taking

$$A_f = \frac{1}{2} \begin{bmatrix} -\phi_0 & 0 \\ 0 & \phi_0 \end{bmatrix}, \quad A_s = - \frac{1}{4} \begin{bmatrix} 3\rho + \sigma - 2\phi_0 & 0 \\ 0 & \rho + 3\sigma + 2\phi_0 \end{bmatrix}.$$

The boundary conditions (4.42) become

$$\rho(0, t) = \sigma(0, t) + 2g(t) \quad (4.43a)$$

$$\sigma(1, t) = -\phi_0. \quad (4.43b)$$

At the left boundary ρ is the inflow variable and the boundary condition (4.43a) is of the general form (4.33). At the right boundary σ is the inflow variable. The boundary condition (4.43b) indicates that σ is constant, and hence the outgoing wave is not reflected.

For $k = 4h/\phi_0$, the split scheme on $0 \leq x \leq 1$ with $h = 1/N$ is simply

$$\begin{aligned} R_m^* &= R_{m-1}^*, & m &= 1, 2, \dots, N \\ S_m^* &= S_{m+1}^*, & m &= -1, 0, \dots, N-1 \\ \begin{bmatrix} R \\ S \end{bmatrix}_m^{**} &= LW(A_s, k) \begin{bmatrix} R \\ S \end{bmatrix}_m^*, & m &= 0, 1, \dots, N-1 \\ R_m^{n+1} &= R_{m-1}^{**}, & m &= 1, 2, \dots, N \\ S_m^{n+1} &= S_{m+1}^{**}, & m &= 0, 1, \dots, N-1. \end{aligned}$$

At the left boundary it appears that we need to specify R_0^* , R_{-1}^* , R_0^{n+1} and S_0^{**} . In fact S_0^{**} is not used in computing S^{n+1} and so we only need to specify the R values. Note that by specifying R_{-1}^* we avoid having to specify any boundary values for R^{**} .

The given boundary conditions (4.43a) provide R_0^{n+1} ,

$$R_0^{n+1} = S_0^{n+1} + 2g(t_{n+1}). \quad (4.43)$$

We next apply the procedure of section 4.4 to compute R_0^* . The expression (4.34) provides $O(\epsilon k^2)$ accurate boundary data for the quasilinear problem provided A^{-1} is evaluated at $(\rho(0, t_n), \sigma(0, t_n))$. The matrix $B = A_f A^{-1} - I$ is given by

$$B = \begin{bmatrix} \frac{2\phi_0}{3\rho+\sigma} - 1 & 0 \\ \frac{-2\phi_0}{\rho+3\sigma} - 1 \end{bmatrix} = O(\epsilon)$$

and the expression (4.37) becomes

$$\begin{aligned} R_0^* &= S_0^* + 2g(t_{n+1/2}) + \frac{1}{2}k \left[\left(\frac{8\phi_0(\rho+\sigma)}{(3\rho+\sigma)(\rho+3\sigma)} \right) \sigma_t(0, t_n) + 2 \left(\frac{2\phi_0}{3\rho+\sigma} - 1 \right) g'(t_n) \right] \\ &= S_0^* + 2g(t_n + 2\phi_0/(3\rho+\sigma)) + \frac{1}{2}k \left(\frac{8\phi_0(\rho+\sigma)}{(3\rho+\sigma)(\rho+3\sigma)} \right) \sigma_t(0, t_n) \end{aligned}$$

where ρ and σ are evaluated at $(0, t_n)$. This can be approximated by

$$R_0^* = S_0^* + 2g(t_n + \alpha k) + \left(\frac{4\alpha(R_0^n + S_0^n)}{R_0^n + 3S_0^n} \right) (S_0^n - S_0^{n-1}) \quad (4.44)$$

where

$$\alpha = \frac{\phi_0}{3R_0^n + S_0^n}.$$

In order to find R_{-1}^* we approximate $\rho^*(-h, t_n + k/2)$. This is equal to $\rho^*(0, t_n + k)$ and proceeding as in Section 4.4 we find the approximation

$$R_{-1}^* = S_1^* + 2g(t_n + 2\alpha k) + \left(\frac{8\alpha(R_0^n + S_0^n)}{R_0^n + 3S_0^n} \right) (S_0^n - S_0^{n-1}) \quad (4.45)$$

with α as above.

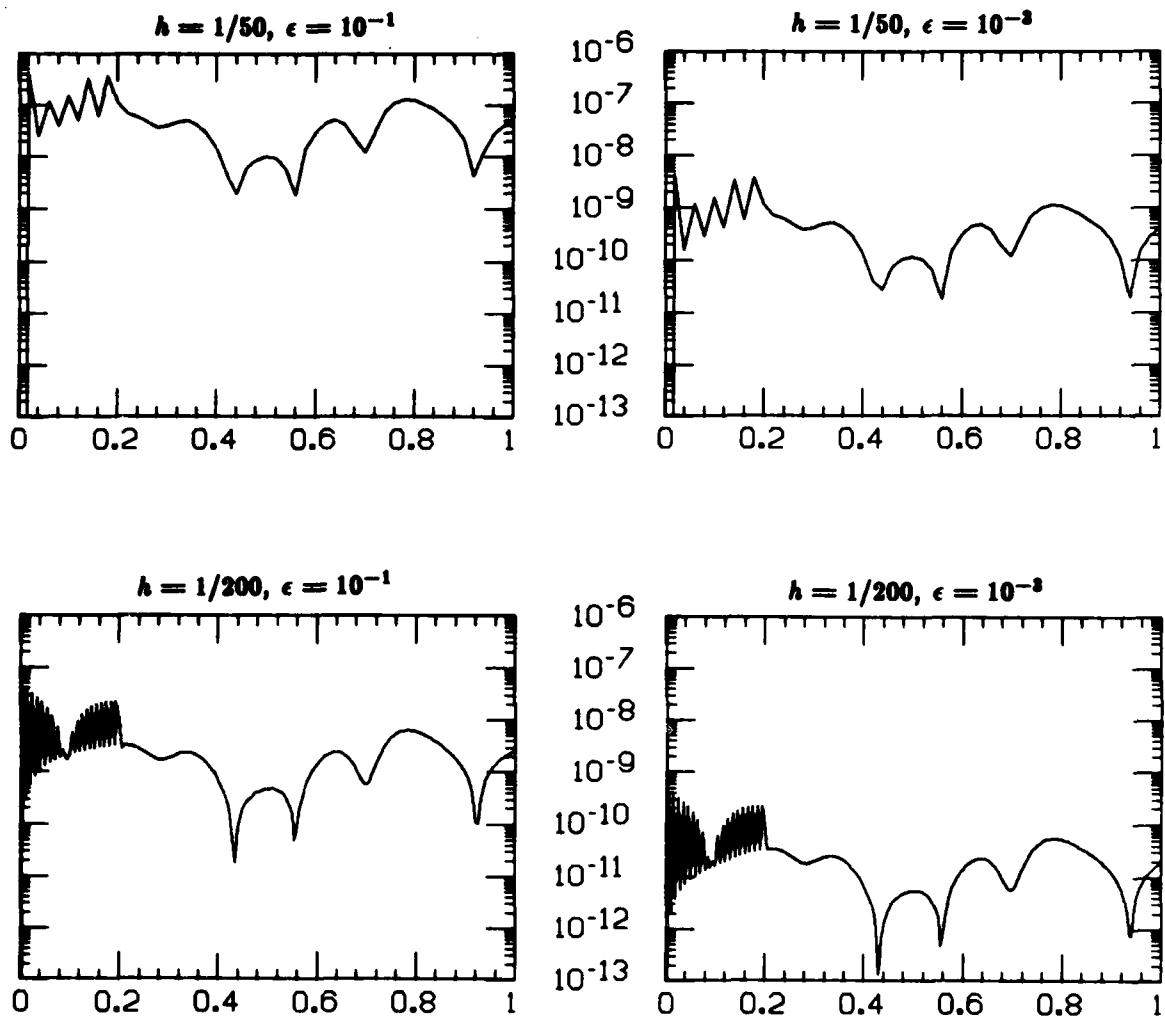


FIG. 4.6. Errors in the shallow water equations using the boundary conditions (4.43), (4.44), (4.45) and (4.46).

At the right boundary we still need to specify S_0^* , S_0^{**} and S_0^{n+1} . Since the boundary condition (4.43b) is time-independent, applying the general procedure at this boundary yields simply

$$S_0^* = S_0^{**} = S_0^{n+1} = -\phi_0. \quad (4.46)$$

Figure 4.6 shows the result of some computations using the boundary conditions (4.43), (4.44), (4.45) and (4.46). The boundary conditions at the right boundary have not affected the interior solution. Errors do arise at the left boundary, but these are seen to be the same order of accuracy as the interior solution.

As in Example 4.1, the oscillations near the boundary are due to the different boundary conditions being of different accuracy.

4.6. Stability of the initial-boundary value problem.

In general stability theory for initial-boundary value problems is considerably more complicated than for pure initial value problems. Only recently has a general theory been developed. The fundamental paper on this subject is by Gustafsson, Kreiss & Sundström[30].

We will first consider an inflow boundary with boundary conditions as derived in Section 4.2. In this situation stability can be proved directly from the Cauchy stability of the interior scheme without resorting to the theory of Gustafsson, Kreiss and Sundström. This is because the boundary conditions we are considering are independent of the interior solution. Consider, for example, the expression (4.25). Our approximation U_0^* can be bounded a priori in terms of an appropriate discrete Sobolev norm of the given boundary data $g(t)$. The same is true of the other required boundary data.

Stability of the time-split method can then be proved using the following general theorem, which states that any Cauchy stable scheme is also stable for the initial-boundary value problem provided that the specified boundary data $\{U_m^n\}_{m=0}^p$ is independent of the interior solution.

THEOREM 4.1. *Suppose $Q(k)$ is Cauchy stable. For the initial-boundary value problem define U^{n+1} by*

$$U_m^{n+1} = \begin{cases} Q(k)U_m^n & m > p, \\ G_m^{n+1} & m = 0, 1, \dots, p. \end{cases}$$

Then the approximation is stable in the sense that

$$\|U^n\|_x^2 \leq K_T \|U^0\|_x^2 + \tilde{K}_T \|G\|_t^2 \quad \text{for } nk \leq T, k < k_0, \quad (4.47)$$

where K_T and \tilde{K}_T are constants depending only on T .

Here the following norms are used:

$$\|U^n\|_x^2 = h \sum_{m=0}^{\infty} |U_m^n|^2,$$

$$\|G\|_t^2 = k \sum_{q=1}^{T/k} \sum_{j=0}^p |G_j^q|^2.$$

Proof. By the Cauchy stability of Q (Stability Definition 3.1'), there exists a constant α and a norm $\|\cdot\|$, equivalent to the ℓ_2 norm, such that

$$\|Q(k)\|^2 \leq 1 + \alpha k \quad \text{for } k < k_0. \quad (4.48)$$

Extend the given initial data $\{U_m^0\}_{m=0}^\infty$ to all m by setting

$$U_m^0 = 0, \quad m = -1, -2, \dots$$

Then solving the quarter plane problem is equivalent to solving the Cauchy problem and then redefining $\{U_j^n\}_{j=0}^p$ at each step. Specifically, we set

$$\tilde{U}_m^{n+1} = Q(k)U_m^n, \quad m = 0, \pm 1, \pm 2, \dots$$

and then take

$$U_m^{n+1} = \begin{cases} G_m^{n+1} & m = 0, 1, \dots, p, \\ \tilde{U}_m^{n+1} & \text{otherwise.} \end{cases} \quad (4.49)$$

The resulting $\{U_m^n\}_{m=0}^\infty$ constitute the solution of the quarter-plane problem.

By (4.48) we have

$$\|\tilde{U}^{n+1}\|^2 \leq (1 + \alpha k)\|U^n\|^2. \quad (4.50)$$

By (4.49) we obtain the following bound for U^{n+1} :

$$\|U^{n+1}\|^2 \leq \|\tilde{U}^{n+1}\|^2 + \|G^{n+1}\|^2 \quad (4.51)$$

where

$$\|G^{n+1}\|^2 = h \sum_{j=0}^p |G_j^{n+1}|^2.$$

Combining (4.50) and (4.51) gives

$$\|U^{n+1}\|^2 \leq (1 + \alpha k)\|U^n\|^2 + \|G^{n+1}\|^2$$

so that by induction we obtain

$$\begin{aligned} \|U^n\|^2 &\leq (1 + \alpha k)^n \|U^0\|^2 + \sum_{q=0}^{n-1} (1 + \alpha k)^q \|G^{n-q}\|^2 \\ &\leq e^{\alpha T} \left(\|U^0\|^2 + \sum_{q=0}^{n-1} \|G^{n-q}\|^2 \right) \end{aligned}$$

for $nk \leq T$. Since $\|U^n\|_x^2 \leq \|U^n\|^2$, $\|U^0\|_x^2 = \|U^0\|^2$ and

$$\sum_{q=0}^{n-1} \|G^{n-q}\|^2 = h \sum_{q=0}^{n-1} \sum_{j=0}^p |G_j^q|^2 \leq \frac{h}{k} \|G\|_t^2$$

for $nk \leq T$, we obtain the desired bound (4.47) with $K_T = e^{\alpha T}$ and $\tilde{K}_T = e^{\alpha T}/\lambda$. ■

To see how this theorem applies to a time-split method, consider the method

$$\begin{aligned} U_m^{*n+1} &= Q_1(k)U_m^n \\ U_m^{n+1} &= Q_2(k)U_m^{*n+1}. \end{aligned} \quad (4.52)$$

We have added the index $n+1$ to U^* for reasons which will soon be apparent. Suppose that the boundary data are of the form

$$\begin{aligned} U_j^{*n+1} &= G_j^{*n+1}, & j &= 0, 1, \dots, p, \\ U_j^{n+1} &= G_j^{n+1}, & j &= 0, 1, \dots, p. \end{aligned} \quad (4.53)$$

For convenience we have assumed that the same number of boundary conditions are needed for both U^{*n+1} and U^{n+1} , but this is not essential. The quantities G_j^{*n+1} and G_j^{n+1} are determined as in Section 4.2 in terms of the given boundary function $g(t)$ and some of its derivatives (say d derivatives). Suppose that the corresponding Sobolev norm of $g(t)$ is uniformly bounded by some constant γ :

$$\|g\|_d^2 = \sum_{j=0}^d \|g^{(j)}\|_t^2 < \gamma.$$

Then we have

$$\|G^*\|_t^2 \leq K_1 \gamma \quad (4.54a)$$

and

$$\|G\|_t^2 \leq K_2 \gamma \quad (4.54b)$$

for some constants K_1 and K_2 .

In order to apply Theorem 4.1 we rewrite (4.52) as

$$\begin{bmatrix} I & 0 \\ -Q_2(k) & I \end{bmatrix} \begin{bmatrix} U^* \\ U \end{bmatrix}_m^{n+1} = \begin{bmatrix} 0 & Q_1(k) \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U^* \\ U \end{bmatrix}_m^n \quad (4.55)$$

to obtain a Cauchy stable scheme for the "super-vector" $(U^*, U)^T$. Note that the method is formally implicit even if the original method was explicit, as it must be since the boundary conditions specified for U^{*n+1} affect the computation of U^{n+1} . The Cauchy stability of (4.55) follows from the Cauchy stability of $Q_2(k)Q_1(k)$, which gives

$$\|U^n\| \leq C \|U^0\|,$$

together with

$$\|U^{*n}\| \leq C_1 \|U^0\|$$

where $C_1 = C \|Q_1(k)\|$. Using Theorem 4.1 and the bounds (4.54) we find that (4.55) is stable for the initial-boundary value problem and that, in particular,

$$\|U^n\|^2 \leq K_T \|U^0\|^2 + \tilde{K}_T (K_1 + K_2) \gamma.$$

Inflow-outflow problems. Stability can also be demonstrated for inflow-outflow problems with boundary conditions of the form discussed in Section 4.4. As above the time-split nature of the scheme can be handled by introducing super-vectors. Hence we will only discuss the stability of a general one-step scheme in which the inflow variables U and the outflow variables V are coupled only through the boundary conditions. As usual we assume Cauchy stability. Our discussion will be rather brief but similar arguments can be found in Goldberg & Tadmor[21][22].

The scheme for V is independent of U and we will assume, as we did in Section 4.4, that the time-split method yields a one-sided scheme for V so that no boundary conditions need be specified. Then from Cauchy stability we clearly have

$$\|V^n\|_x^2 \leq \|V^0\|_x^2$$

since the introduction of the boundary does not affect the computation of $\{V_j^n\}_{j=0}^\infty$. Moreover such a scheme for V is also stable in the sense of Definition 3.3 of Gustafsson, Kreiss & Sundström[30] (we refer to this as GKS-stability). This stability condition also requires bounds on a norm of V along the boundary. The GKS-stability follows easily from the theory of [30] for a one-sided scheme.

GKS-stability of the outflow problem is just what we need to prove stability of the inflow problem. Recall from Section 4.4 that the boundary conditions for U depend only on $g(t)$ and on values of V along the boundary, and can be bounded in terms of $\|g\|_d$ and $\|V\|_x$. The former of these is assumed to be uniformly bounded while the latter is bounded by the GKS-stability of V . Theorem 4.1 thus applies to the inflow problem and hence the entire approximation is stable on the initial-boundary value problem.

These stability results are supported by large-time numerical calculations for all of the examples which have been given in this chapter, including the boundary conditions of Section 4.5 for the shallow water equations.

5. Other applications of the theory

5.1. Introduction.

In this chapter some of the theory developed in previous chapters is applied to a few different problems. In Section 5.2 hyperbolic problems in two space dimensions are considered. Again we split between fast and slow subproblems although now spatial splittings may also be used. Intermediate boundary conditions are derived for a scalar example.

We then turn to the use of time-split methods for problems which are not hyperbolic, since many of the techniques that have been introduced are applicable to other problems as well.

In Section 5.3 the convection-diffusion equation $u_t = -cu_x + \epsilon u_{xx}$ is studied as a model for general equations containing both hyperbolic and parabolic terms. An analysis very similar to that of Section 2.5 is performed with analogous results. For the Cauchy problem the time-split method is more accurate provided the mesh ratio is chosen appropriately. For boundary value problems the correct intermediate boundary conditions at the inflow boundary can be computed using the general procedure of Chapter 4. At the outflow boundary no special boundary data need be specified, but the solution generally has a boundary layer at this boundary which causes special difficulties. The interior solution (away from the boundary layer) can still be calculated more efficiently than with the unsplit method, but less efficiently than in the Cauchy problem due to mesh ratio restrictions imposed by the boundary layer.

In Section 5.4 a very different kind of time-split method is considered. The Peaceman-Rachford ADI method for the two-dimensional heat equation $u_t = u_{xx} + u_{yy}$ is viewed as a time-split method with the splitting (1.43). By means of the procedure of Chapter 4, intermediate boundary conditions are derived for a rectangular region which agree with the classical boundary conditions for this method.

5.2. Hyperbolic problems in two space dimensions.

The time-split method can be used in two (or more) space dimensions in much the same way as in one dimension. Locally one-dimensional methods, where a time-split method is used to reduce a multidimensional problem to a sequence of one-dimensional problems, have already been discussed in Chapter 1. The techniques which have been developed in the intervening chapters are applicable to such splittings and can be used to analyze their efficiency and, in some cases, to generate boundary data for the intermediate solutions. This will be done for the Peaceman-Rachford ADI method in Section 5.4.

In this section, however, we continue to concentrate on hyperbolic problems which can be split into "fast" and "slow" subproblems. Each of these subproblems will, in

general, still be a two-dimensional problem. In some cases it will prove useful to also use spatial splittings in order to solve one or the other of these subproblems.

A general hyperbolic problem in two space dimensions has the form

$$u_t = Au_x + Bu_y \quad (5.1)$$

where the matrices A and B have the property that $\xi A + \eta B$ is diagonalizable with real eigenvalues for all real values of ξ and η . In the notation of Chapter 1, we have $A(u) = Au_x + Bu_y$ and we consider splittings of the form

$$\begin{aligned} A_1(u) &= A_f u_x + B_f u_y \\ A_2(u) &= A_s u_x + B_s u_y. \end{aligned} \quad (5.2)$$

For the constant coefficient case the splitting error is easily computed by expanding the exponential solution operators. Define the differential operators C_f and C_s by

$$\begin{aligned} C_f &= A_f \partial_x + B_f \partial_y \\ C_s &= A_s \partial_x + B_s \partial_y \end{aligned}$$

The splitting error is found to be

$$\begin{aligned} E_{\text{split}}(k) &= -\frac{1}{8} k^3 \left(\frac{1}{4} C_f^2 C_s - \frac{1}{2} C_f C_s C_f + \frac{1}{4} C_s C_f^2 \right. \\ &\quad \left. - \frac{1}{2} C_s^2 C_f + C_s C_f C_s - \frac{1}{2} C_f C_s^2 \right) + O(k^4). \end{aligned} \quad (5.3)$$

which is analogous to the one-dimensional result (2.22). In particular the splitting error is zero if all of the matrices A_f , A_s , B_f and B_s commute.

As in the one-dimensional case, a splitting of the form (5.2) will be useful if A_f and B_f are sparse relative to A and B and if A_s and B_s have relatively small eigenvalues. Suppose that $\|A_f\| \approx \|B_f\| \approx a$ while $\|A_s\| \approx \|B_s\| \approx \epsilon a$ with $\epsilon \ll 1$ and that the spectral radius of each matrix is comparable to its norm.

Let $LW(A, B, k)$ denote the two-dimensional Lax-Wendroff operator, which is analogous to (1.11) and can be found, for example, in Mitchell and Griffiths[41]. The stability limit for $LW(A, B, k)$ is given by

$$\frac{k}{h} \max(\rho(A), \rho(B)) < \frac{1}{\sqrt{8}}.$$

The split scheme corresponding to (2.18a,c) is given by

$$Q_s(k) = LW(A_s, B_s, k)$$

and

$$Q_f(k/2) = (LW(A_f, B_f, k/m))^{m/2}.$$

An efficiency analysis very similar to that performed on the one-dimensional problem in Section 2.5 shows that the optimal mesh ratio on the fast scale is

$$\frac{k}{mh} \approx \frac{1}{3}. \quad (5.4)$$

This, however, violates the stability limit for $LW(A_f, B_f, k/m)$, which is

$$\frac{k}{mh} \leq \frac{1}{\sqrt{8}a}.$$

We must use this smaller value of $k/(mh)$ instead. Alternatively we can replace the two-dimensional operator $LW(A_f, B_f, k/m)$ by the split scheme

$$LW_x(A_f, k/2m)LW_y(B_f, k/m)LW_x(A_f, k/2m)$$

where LW_x and LW_y represent one-dimensional Lax-Wendroff in the x - and y -directions respectively. This does not increase the truncation error significantly but increases the stability limit to

$$\frac{k}{mh} \leq \frac{1}{a}$$

so that the optimal mesh ratio (5.4) can be used. (Recall that this increase in the stability limit was Strang's original goal in introducing the Strang splitting[49].)

On the slow scale the optimal value of $\lambda = k/h$ depends on the size of the splitting error. If $E_{\text{split}}(k)$ is negligible then

$$\lambda \approx \frac{1}{\epsilon a}.$$

Again this violates the stability limit $\lambda \leq 1/(\sqrt{8}\epsilon a)$ and we may wish to introduce a spatial splitting in $Q_s(k)$.

In the more usual situation, however, when the splitting error is $O(\epsilon a^3 k^3)$, the optimal mesh ratio is

$$\lambda \approx \frac{1}{\sqrt{\epsilon a}}.$$

This is well within the stability limit and there is no need to introduce spatial splittings.

Perturbed problems. The splitting (5.2) is also useful when the exact solution operator corresponding to $A_1(u)$ is known. This is perhaps not so common in two dimensions as in one. In one space dimension we considered several examples in which the coefficients had large mean values and small variations. We could then pull out a constant coefficient subproblem $u_t^* = A_f u_x^*$ which could be solved by the method of characteristics. Unfortunately, in two dimensions the method of characteristics is applicable only if A_f and B_f are simultaneously diagonalizable.

Here, however, we suppose that the solution operator for the fast part is known exactly and consider the time-split method (2.18a,b) with

$$Q_s(k) = LW(A_s, B_s, k)$$

and

$$Q_f(k/2) = \exp(\frac{1}{2}k(A_f \partial_x + B_f \partial_y)).$$

An efficient analysis similar to that of Section 2.5 shows that the optimal mesh ratio is given by

$$\begin{aligned} \lambda &= \frac{1}{\epsilon a} && \text{if the splitting error is negligible, or} \\ \lambda &= \frac{1}{a} && \text{if the splitting error is } O(\epsilon k^3 a^3). \end{aligned}$$

In the former case we can only achieve the indicated mesh ratio by using a spatial splitting for $Q_s(k)$ but in the more usual situation, when splitting errors are present, this is not necessary.

Boundary conditions for a perturbed scalar problem. We now consider a perturbed scalar problem which can be split in this manner and show how to derive appropriate boundary conditions for the intermediate solutions in two space dimensions.

Consider the problem

$$u_t = (1 + \alpha(x, y, t))u_x + (1 + \beta(x, y, t))u_y \quad (5.5)$$

on the unit square $[0, 1] \times [0, 1]$ with boundary conditions

$$\begin{aligned} u(1, y, t) &= g_1(y, t), \\ u(x, 1, t) &= g_2(x, t), \end{aligned} \quad (5.6)$$

and suppose that $|\alpha(x, y, t)| \leq \epsilon$, $|\beta(x, y, t)| \leq \epsilon$ for all x, y , and t with $\epsilon \ll 1$. It is natural to split this by taking

$$\begin{aligned} A_f &= 1, & A_s &= \alpha(x, y, t), \\ B_f &= 1, & B_s &= \beta(x, y, t). \end{aligned}$$

The subproblem $u_t^* = u_x^* + u_y^*$ can be solved exactly:

$$u^*(x, y, t + k) = u^*(x + k, y + k, t).$$

Taking $k = 2h$ and using Lax-Wendroff on the slow problem, the split method becomes

$$\begin{aligned} U_{m,j}^* &= U_{m+1,j+1}^n & m, j &= 0, 1, \dots, N-1, \\ U_{m,j}^{**} &= LW(\alpha, \beta, k)U_{m,j}^* & m, j &= 1, 2, \dots, N-1, \\ U_{m,j}^{n+1} &= U_{m+1,j+1}^{**} & m, j &= 0, 1, \dots, N-1. \end{aligned}$$

The values $U_{N,j}^{n+1}$ and $U_{m,N}^{n+1}$ are given by the boundary conditions (5.6) while the values $U_{0,j}^{**}$ and $U_{m,0}^{**}$ are not required in computing U^{n+1} and therefore do not need to be specified. We do need to specify the following intermediate boundary data:

$$\begin{aligned} U_{N,j}^*, U_{N,j}^{**} & \quad j = 0, 1, \dots, N, \\ U_{m,N}^*, U_{m,N}^{**} & \quad m = 0, 1, \dots, N, \end{aligned}$$

First consider $U_{N,j}^*$. We begin as usual by introducing the function $u^*(x, y, t)$ satisfying

$$u_t^* = u_x^* + u_y^* \quad (5.7)$$

with initial conditions at time t_n

$$u^*(x, y, t_n) = u(x, y, t_n).$$

Then we want $U_{N,j}^* \approx u^*(1, y_j, t_n + k/2)$. Expanding in a Taylor series and using (5.7) together with the fact that $u_x^* = u_x$ and $u_y^* = u_y$ at time t_n , we obtain

$$\begin{aligned} u^*(1, y, t_n + k/2) &= u^*(1, y, t_n) + \frac{1}{2}k u_t^*(1, y, t_n) + \frac{1}{8}k^2 u_{tt}^*(1, y, t_n) + \dots \\ &= u(1, y, t_n) + \frac{1}{2}k(u_x(1, y, t_n) + u_y(1, y, t_n)) \\ &\quad + \frac{1}{8}k^2(u_{xx}(1, y, t_n) + 2u_{xy}(1, y, t_n) + u_{yy}(1, y, t_n)) + \dots \end{aligned} \quad (5.8)$$

We now use the original equation (5.5) to replace x -derivatives by y - and t -derivatives, so that the given boundary data $g_1(y, t)$ and its derivatives can be used to specify $U_{N,j}^*$. We find that

$$\begin{aligned} u^*(1, y, t_n + k/2) &= u + \frac{k}{2(1+\alpha)}(u_t + (\alpha - \beta)u_y) + \frac{k^2}{8(1+\alpha)^2} \left(u_{tt} - 2(1+\beta)u_{ty} \right. \\ &\quad \left. + (1+\beta)^2 u_{yy} + \left(\frac{\alpha_t - (1+\beta)\alpha_y}{1+\alpha} + \alpha_x \right) u_t + \left(\frac{1+\beta}{1+\alpha}(\alpha_y - \alpha_t) \right. \right. \\ &\quad \left. \left. + (1+\beta)(\beta_y - \alpha_x) - \beta_t - (1+\alpha)\beta_x \right) u_y \right) + \dots \end{aligned}$$

where the functions u , α , and β (and their derivatives) on the right hand side are all evaluated at $(1, y, t_n)$. If α , β and their first derivatives are all $O(\epsilon)$, then this can be simplified in the usual manner:

$$u^*(1, y, t_n + k/2) = u(1, y, t_n + k/2(1+\alpha)) + \frac{k(\alpha - \beta)}{2(1+\alpha)} u_y(1, y, t_n) + O(\epsilon k^2).$$

We can maintain the accuracy of the interior scheme by using the boundary values

$$\begin{aligned} U_{N,j}^* &= g_1(jh, t_n + \frac{1}{2}k/(1+\alpha(1, jh, t_n))) \\ &\quad + \left(\frac{k(\alpha(1, jh, t_n) - \beta(1, jh, t_n))}{2(1+\alpha(1, jh, t_n))} \right) g_{1y}(jh, t_n). \end{aligned}$$

The boundary values $U_{m,N}^*$ along the boundary $y = 1$ are found in exactly the same manner. We obtain

$$\begin{aligned} U_{m,N}^* &= g_2(jh, t_n + \frac{1}{2}k/(1+\beta(jh, 1, t_n))) \\ &\quad + \left(\frac{k(\beta(jh, 1, t_n) - \alpha(jh, 1, t_n))}{2(1+\beta(jh, 1, t_n))} \right) g_{2x}(jh, t_n). \end{aligned}$$

To compute boundary values for the second intermediate solution U^{**} , we proceed as we did in the one dimensional case by defining $u^{**}(x, y, t)$ as the solution of

$$u_t^{**} = u_x^{**} + u_y^{**}$$

with $u^{**}(x, y, t_{n+1}) = u(x, y, t_{n+1})$ and then solving backwards in time to find $U_{N,j}^{**} \approx u^{**}(1, jh, t_{n+1} - k/2)$. The expression we obtain for $u^{**}(1, jh, t_{n+1} - k/2)$ is exactly the

same as the right hand side of (5.8) but with k replaced by $-k$ and all functions evaluated at $(1, y, t_{n+1})$. We can thus take

$$U_{N,j}^{**} = g_1(h, t_{n+1} - \frac{1}{2}k/(1 + \alpha(1, jh, t_{n+1}))) - \left(\frac{k(\alpha(1, jh, t_{n+1}) - \beta(1, jh, t_{n+1}))}{2(1 + \alpha(1, jh, t_{n+1}))} \right) g_{1y}(1, jh, t_{n+1})$$

with a similar expression for $U_{m,N}^{**}$.

Irregular regions. Attempting to compute on irregular (nonrectangular) regions generally complicates the problem of specifying boundary conditions for any numerical method. Gridpoints frequently do not lie exactly on the boundary and so special procedures must be used for points near the boundary even when the correct data are known along the boundary itself. Here we will only consider the problem of transforming boundary conditions for the given problem into boundary data for the intermediate solutions. The problem of then using these data, defined along the boundary, to specify the necessary solution values at nearby points can then be handled by standard techniques.

Consider the problem (5.5) in a region with boundary parametrized by $(x(s), y(s))$, $0 \leq s \leq 1$. The region is assumed to lie to the left of this curve. The boundary conditions are

$$u(x(s), y(s), t) = g(s, t)$$

at inflow points. For convenience we will assume that α and β are independent of t and will write $\alpha(s)$ for $\alpha(x(s), y(s))$ and similarly for β . Then $(x(s), y(s))$ is an inflow point if

$$x'(s)(1 + \beta(s)) < y'(s)(1 + \alpha(s)).$$

This is illustrated in Figure 5.1.

For the rectangular region, we replaced x -derivatives by y - and t -derivatives at the right boundary while at the top boundary we replaced y -derivatives by x - and t -derivatives. These are both special cases of the general situation. At the inflow boundary of a nonrectangular region we must replace both the x - and the y -derivatives by tangential and t -derivatives in order to obtain expressions in terms of the given boundary data.

In determining u^* at inflow points we first obtain an expression analogous to (5.8),

$$u^*(x(s), y(s), t_n + k/2) = u(x(s), y(s), t_n) + \frac{1}{2}k(u_x(x(s), y(s), t_n) + u_y(x(s), y(s), t_n)) + \dots$$

Now we solve for u_x and u_y in terms of the given boundary conditions from the equations

$$g_t(s, t_n) = u_t(x(s), y(s), t_n) = (1 + \alpha(s))u_x(x(s), y(s), t_n) + (1 + \beta(s))u_y(x(s), y(s), t_n)$$

$$g_s(s, t_n) = x'(s)u_x(x(s), y(s), t_n) + y'(s)u_y(x(s), y(s), t_n).$$

Solving this system gives (dropping arguments for clarity)

$$u_x = - \frac{y'g_t - (1 + \beta)g_s}{(1 + \beta)x' - (1 + \alpha)y'}$$

$$u_y = \frac{x'g_t - (1 + \alpha)g_s}{(1 + \beta)x' - (1 + \alpha)y'}$$

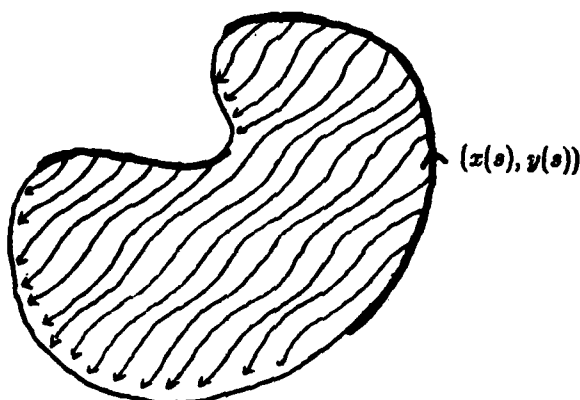


FIG. 5.1. Irregular region with boundary parametrized by $(x(s), y(s))$, moving counterclockwise as s increases. The characteristics in the x - y plane are also shown. The exact solution propagates along these lines, which have slope $(1 + \beta(x, y))/(1 + \alpha(x, y)) = 1 + O(\epsilon)$ at each point (x, y) . The inflow portion of the boundary, where the boundary conditions are specified, is shown as a bold line.

so that

$$u_x + u_y = \frac{(x' - y')g_t + (\beta - \alpha)g_s}{(1 + \beta)x' - (1 + \alpha)y'}$$

Note that the denominator is nonzero at inflow points. Similarly, we can obtain expressions for higher derivatives. When α , β and their derivatives are $O(\epsilon)$ we have

$$u^*(x(s), y(s), t_n + k/2) = g(s, t^*(s)) + \frac{1}{2} \left(\frac{k(\beta(s) - \alpha(s))}{(1 + \beta(s))x'(s) - (1 + \alpha(s))y'(s)} \right) g_s(s, t_n) + O(\epsilon k^2)$$

where

$$t^*(s) = t_n + \frac{1}{2} \left(\frac{k(x'(s) - y'(s))}{(1 + \beta(s))x'(s) - (1 + \alpha(s))y'(s)} \right)$$

This formula reduces to those derived before on the unit square, in which case either $x'(s) = 0$ or $y'(s) = 0$.

5.3. Convection-diffusion equations.

As mentioned in Section 1.6, time-split methods are frequently used to solve equations of mixed type by splitting between the hyperbolic and parabolic parts and using different methods on the two pieces. This is done for example with the Navier-Stokes equations for viscous fluid flow or convection-diffusion equations for miscible flow.

In this section we consider a simple model equation for such problems, the constant-coefficient scalar convection-diffusion equation

$$u_t = -cu_x + \epsilon u_{xx} \tag{5.9}$$

where c and ϵ are nonnegative constants. Consider the splitting $\mathcal{A}_1(u) = -cu_x$, $\mathcal{A}_2(u) = \epsilon u_{xx}$. For this scalar constant coefficient problem there is no splitting error so we do not need to use the Strang splitting. If $k/h = -p/c$ for some positive integer p then the subproblem $u_t^* = -cu_x^*$ can be solved exactly. Using Crank-Nicolson for the remaining subproblem gives the split method

$$\begin{aligned} U_m^* &= U_{m-p}^n \\ U_m^{n+1} &= CN(\epsilon, k)U_m^* \end{aligned} \quad (5.10)$$

where CN is the Crank-Nicolson operator, which has the form

$$CN(A, k) = (I - \frac{1}{2}kAD_+D_-)^{-1}(I + \frac{1}{2}kAD_+D_-)$$

for a general constant coefficient system $u_t = Au_{xx}$.

If we eliminate the intermediate solution U^* , we can rewrite the split method (5.10) as a one-step method:

$$(I - \frac{1}{2}k\epsilon D_+D_-)U_m^{n+1} = (I + \frac{1}{2}k\epsilon D_+D_-)U_{m-p}^n. \quad (5.11)$$

Figure 5.2 shows the stencil for this method when $c = p = 1$.

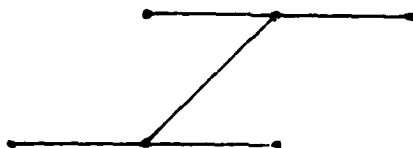


FIG. 5.2. Stencil for (5.11).

The method can thus be viewed as a "skewed Crank-Nicolson" method similar to the skewed Lax-Wendroff method of Example 1.2. The stencil of the method follows the characteristic of the hyperbolic part of the problem.

Time-split methods similar to (5.10) can be used for more general systems of the form

$$u_t = Au_x + Bu_{xx}$$

where A and B may be functions of x , t , and u . One way to proceed is to use the splitting $\mathcal{A}_1(u) = Au_x$ and $\mathcal{A}_2(u) = Bu_{xx}$. In general neither subproblem can be solved exactly, but it may be advantageous to use different numerical procedures for the two subproblems. This is the approach generally taken with the Navier-Stokes equations[1].

Another alternative is to use the splitting $\mathcal{A}_1(u) = A_f u_x$ and $\mathcal{A}_2(u) = A_s u_x + Bu_{xx}$ where $u_t = A_f u_x$ can be solved exactly and the remaining subproblem $u_t = \mathcal{A}_2(u)$ corresponds to small perturbations, i.e., $\rho(A_s)$ and $\rho(B)$ are small compared to $\rho(A_f)$.

Here we consider only the scalar problem (5.9), which is sufficient to illustrate some of the new issues that arise when time-split methods are applied to such problems. In particular, when ϵ is small there may be a boundary layer at the outflow boundary, which poses special problems for the time-split method.

Efficiency analysis for the Cauchy problem. Before considering boundary value problems, however, we first perform an efficiency analysis for the Cauchy problem similar to that of Section 2.5. The split method (5.10) will be compared to the unsplit method

$$\begin{aligned} & (1 + \frac{1}{2}kcD_0 - \frac{1}{2}k\epsilon D_+ D_-)U_m^{n+1} \\ & = (1 - \frac{1}{2}kcD_0 + \frac{1}{2}k\epsilon D_+ D_-)U_m^n. \end{aligned} \quad (5.12)$$

This method is second order accurate with a truncation error

$$\begin{aligned} E(k)u = k\{k^2[-\frac{1}{12}c^3 + \frac{1}{4}c^2\epsilon\partial_x - \frac{1}{4}c\epsilon^2\partial_x^2 + \frac{1}{12}\epsilon^3\partial_x^3] \\ + h^2[-\frac{1}{6}c + \frac{1}{12}\epsilon\partial_x]\}u_{xxx} + O(k^4). \end{aligned} \quad (5.13)$$

We assume that $c \approx 1$, $\epsilon \ll 1$ and that u is smooth so that all derivatives of u are order unity. (This latter assumption will not be valid in the boundary value problems considered later.) Then $E(k)$ can be approximated as

$$E(k) \approx -\frac{1}{12}k[k^2c^3 + 2h^2c]\partial_x^3.$$

We see that the error is dominated by terms arising from the hyperbolic part of the equation. The global error at time $t = 1$ is roughly bounded by

$$\frac{1}{k}\|E(k)u\| \approx \frac{1}{12}(k^2c^3 + 2h^2c)\|u_{xxx}\|. \quad (5.14)$$

As in Section 2.5 the optimal mesh ratio is found by requiring $k^2c^3 \approx 2h^2c$, for otherwise we could increase one of the stepsizes without significantly increasing the error. This gives the optimal value of the mesh ratio:

$$\lambda \approx \sqrt{2}/c. \quad (5.15)$$

For comparison purposes we wish to normalize the error at $t = 1$ by the amount of work performed. A tridiagonal system of equations must be solved at each step but this only requires work proportional to $1/h$. The work required to compute the solution at $t = 1$ is thus proportional to $1/kh$. The same is true in the split method, with a similar constant of proportionality, and so we can normalize the error simply by dividing by kh . Using (5.14) and (5.15), we find that

$$\text{normalized error} \approx \frac{\sqrt{2}}{6}c^2\|u_{xxx}\|. \quad (5.16)$$

Now consider the split method (5.10). Since there is no splitting error and no error is committed in solving the first subproblem, the overall truncation error is simply the truncation error of $CN(c, k)$, which is found by setting $c = 0$ in (5.13):

$$E^{TSM}(k)u = \frac{1}{12}k[k^2\epsilon^3\partial_x^3 + h^2\epsilon\partial_x]u_{xxx} + O(k^4). \quad (5.17)$$

The optimal mesh ratio is thus

$$\lambda \approx \frac{1}{\epsilon} \sqrt{\frac{\|\partial_x^4 u\|}{\|\partial_x^6 u\|}}. \quad (5.18)$$

This indicates that large timesteps are optimal, $\lambda = O(1/\epsilon)$.

Using (5.18) in (5.17) gives the following normalized error at $t = 1$:

$$\text{normalized error} \approx \frac{1}{8}\epsilon^2 \sqrt{\|\partial_x^4 u\| \|\partial_x^6 u\|}. \quad (5.19)$$

This is smaller than (5.16) by roughly a factor of $(\epsilon/c)^2$.

These results are virtually identical to those of Section 2.5 for the pure hyperbolic problem in the same situation, namely for a perturbed problem with no splitting error. Other situations, e.g. splittings with error or the use of several steps of a difference method on the fast problem, can be investigated in the same manner with analogous results. Numerical experiments have confirmed these theoretical predictions.

Nonsmooth solutions. The advantages of the time-split method are most clearly seen when computing nearly-discontinuous solutions, for example shocks in the Navier-Stokes equations or steep concentration gradients in miscible flow problems.

Example 5.1. Consider the model problem

$$u_t = -u_x + \epsilon u_{xx} \quad (5.20)$$

with initial conditions

$$u(x, 0) = \begin{cases} 1, & x < 0.1, \\ 0, & x \geq 0.1. \end{cases}$$

This initial discontinuity smears out as it propagates to the right with speed 1. At time $t = 0.7$ the true solution is seen as the dashed line in Figure 5.3 (with $\epsilon = 10^{-3}$).

The unsplit method (5.12) performs poorly on such problems because of the convective term. The resulting solution is oscillatory as seen in Figure 5.3, which shows the solution obtained with $k = h = 10^{-2}$.

With the split method (5.10) the convection is handled exactly by shifting. Only the diffusion is handled numerically and discontinuous initial data cause no problems. By the efficiency analysis it is optimal to take $\lambda = O(1/\epsilon)$. We choose to again take $h = 10^{-2}$ and take $k = 0.7$ which corresponds to $\lambda = 70$. Figure 5.4 shows the resulting solution obtained with a single step of the time-split method (5.10).

Boundary value problems. We now turn to the most interesting case: the boundary value problem (5.9) on $0 \leq x \leq 1$. For definiteness we will take $c = 1$. When $\epsilon = 0$ the equation is the familiar hyperbolic equation $u_t = -u_x$ for which boundary data need only be specified at the inflow boundary $x = 0$. The exact solution is a wave moving to the right, unaltered, with speed 1. When ϵ is small the solution is again a wave which moves to the right, but now it dissipates slowly as it moves along. For ϵ very small we might expect the solution to be very similar to that obtained with $\epsilon = 0$. However, whenever $\epsilon > 0$ the equation (5.9) is parabolic and boundary conditions must be specified both at $x = 0$ and at $x = 1$. Equation (5.9) is a *singular perturbation equation* since the limiting equation with $\epsilon = 0$ has a singular nature quite different from that with $\epsilon > 0$.

For small ϵ the solution to (5.9) has a *boundary layer* near $x = 1$, a small region in which the solution changes rapidly in order to match the boundary conditions at $x = 1$. For (5.9) the boundary layer has width $O(\epsilon)$. For $x < 1 - \epsilon$ the solution is simply a rightward moving wave, slowly dissipating, and looks very much like the solution to the

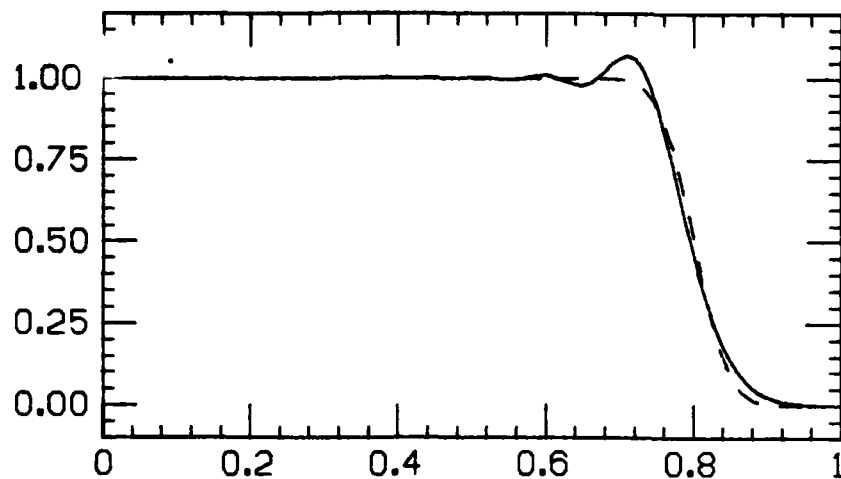


FIG. 5.3. Solution of the convection-diffusion equation of Example 5.1 with $\epsilon = 10^{-3}$ at time $t = 0.7$. The dashed line is the true solution. The solid line is the solution computed with the unsplit method (5.12) with $k = h = 10^{-2}$.

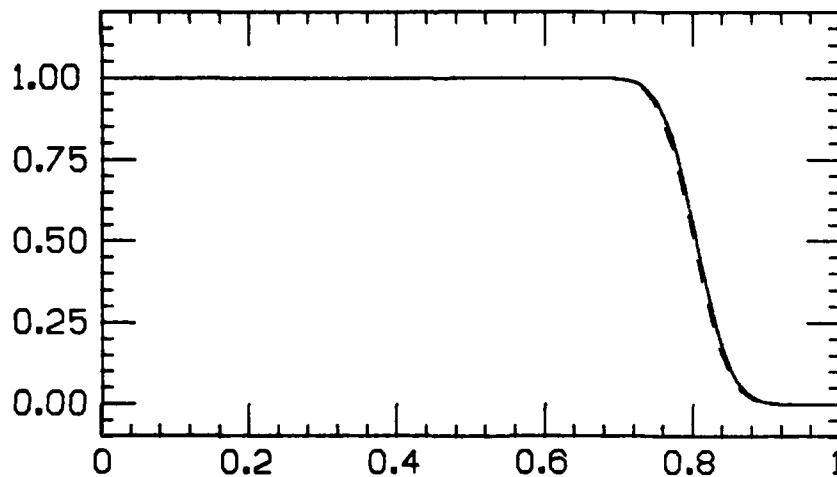


FIG. 5.4. Solution of the convection-diffusion equation of Example 5.1 with $\epsilon = 10^{-3}$ at time $t = 0.7$. The dashed line is the true solution. The solid line is the solution computed with the split method (5.10) with $h = 10^{-2}$, $k = 0.7$.

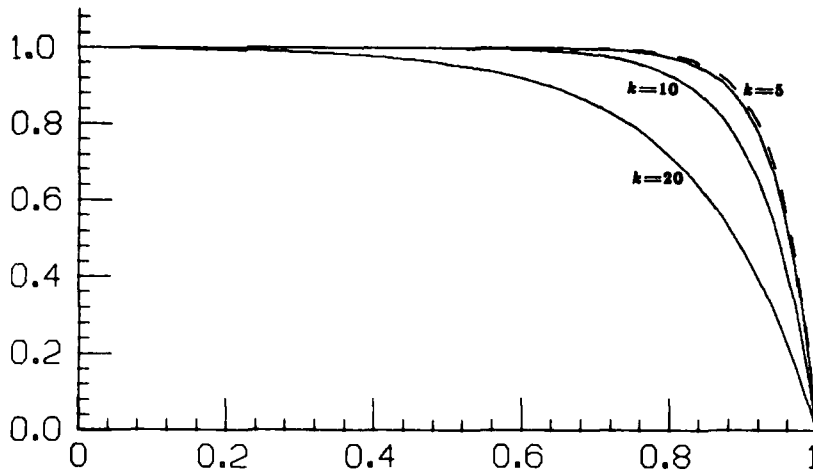


FIG. 5.5. True solution (dashed line) and computed solutions to the steady-state convection-diffusion solution using the time-split method (5.10) with $h = 10^{-2}$ and several different values of k . Again $\epsilon = 5 \times 10^{-2}$.

pure hyperbolic problem. The solution is smooth and the convective term $-u_x$ dominates the dissipative term ϵu_{xx} . In the region $1 - \epsilon \leq x \leq 1$, however, the solution is rapidly changing and $u_{xx} \approx \frac{1}{\epsilon} u_x$. Here both terms are equally important and the solution in this region is quite unlike that seen for $\epsilon = 0$.

The presence of the boundary layer in this problem causes difficulties in the application of any numerical procedure. The time-split method performs quite well, particularly in the interior, provided that moderate values of the timestep k are used. Using $\lambda \approx 1$ gives results which are everywhere better than the unsplit method, by a factor of ϵ in the interior (see Example 5.2). This is to be expected based on the previous analysis of the Cauchy problem. However, it is no longer possible to obtain even greater efficiency by using larger timesteps. This is because of errors arising in the boundary layer. It is illuminating to analyze the difficulties which arise when larger timesteps are used.

In order to concentrate on the boundary layer itself, we first consider the steady-state problem (5.20) with time-independent boundary conditions

$$\begin{aligned} u(0, t) &= 1 - e^{-1/\epsilon} \\ u(1, t) &= 0 \end{aligned}$$

and initial conditions

$$u(x, 0) = 1 - e^{(x-1)/\epsilon}.$$

The solution to this problem is simply

$$u(x, t) = 1 - e^{(x-1)/\epsilon} \tag{5.21}$$

for all t , as shown by the dashed line in Figure 5.5 for $\epsilon = 5 \times 10^{-2}$.

The numerical solution to this problem will also reach a steady state, though in general the numerical steady state will be different from the true solution. For the unsplit method (5.12) setting $U_m^n = U_m^{n+1}$ shows that the steady-state solution satisfies

$$(-cD_0 + \epsilon D_+ D_-)U_m^n = 0.$$

This solution depends on h but is independent of the timestep k used to compute it (and hence is independent of λ). The numerical solution is quite accurate if h is sufficiently small. If $h > \epsilon$ then oscillations begin to appear in the boundary layer. This will be seen in Example 5.2.

Now consider the time-split method (5.10). In order to implement this method we must specify some additional boundary values $U_0^*, U_1^*, \dots, U_{p-1}^*$ at the boundary $x = 0$. This can be done using the general procedure of Chapter 4, as will be seen later in this section. For the present example with time-independent boundary conditions, these simply reduce to

$$U_j^* = 1 - e^{-j/\epsilon} \quad \text{for } j = 0, 1, \dots, p-1.$$

At the boundary $x = 1$, where the boundary layer is, we do not need to specify any additional boundary data. Figure 5.5 shows the numerical steady-state solution for the time-split method with $\epsilon = 5 \times 10^{-2}$, $h = 10^{-2}$ and several different values of k . Note that this steady-state is no longer independent of k and is far from the true solution even for moderate values of λ .

In order to understand this phenomenon we must consider the individual steps of the time-split method in more detail. In the first step of (5.10) the solution shifts to the right and much of the boundary layer is lost. If $k > \epsilon$ the boundary layer shifts almost completely out of the interval. We then have

$$U_m^* \approx 1 \tag{5.22}$$

for all m . The solution U^* is nearly independent of k for $k > \epsilon$. In the second step of (5.10) we are using Crank-Nicolson to solve the heat equation with initial values (5.22) and boundary values $U_0^{n+1} \approx 1$, $U_N^{n+1} = 0$. For large values of k the solution U^{n+1} tends to $U_m^{n+1} = 1 - x_m$, which is the steady-state solution of $u_t = \epsilon u_{xx}$ with boundary conditions

$$u(0, t) = 1, \quad u(1, t) = 0.$$

This explains the "over-diffused" nature of the solutions seen in Figure 5.5 for larger values of k .

Where does the time-split method break down? Recall that there is no splitting error for this problem so that if both subproblems are solved exactly we should obtain the exact solution to the original problem, for any value of k . Let us now do this. The first subproblem $u_t^* = -u_x^*$ is already being solved exactly (modulo the boundary conditions at $x = 0$, but these have a negligible effect on the results seen here). We now wish to also use the exact solution operator for the second subproblem

$$u_t^{**}(x, t) = \epsilon u_{xx}^{**}(x, t), \quad 0 \leq x \leq 1, \quad t_n \leq t \leq t_{n+1}$$

with initial conditions

$$u^{**}(x, t_n) = u^*(x, t_{n+1}).$$

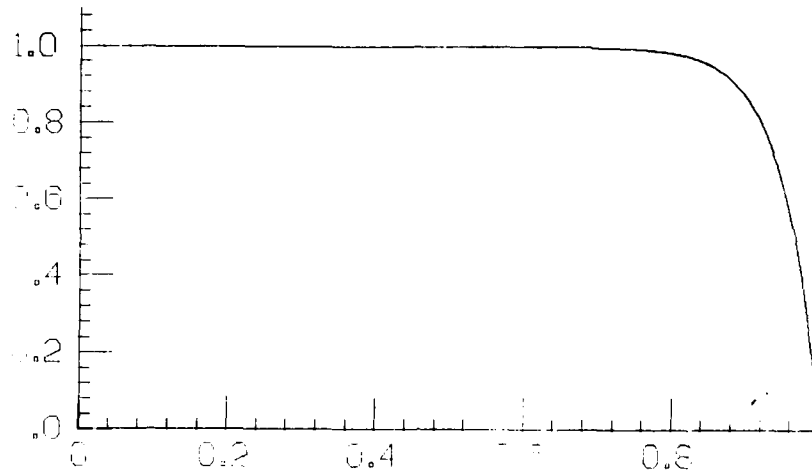


FIG. 5.6. The correct boundary conditions $u^{**}(1, t_n + \tau)$ for $0 \leq \tau \leq k$ with $\epsilon = 5 \times 10^{-2}$ and $k = 1$.

In order to apply the exact solution operator we must also specify boundary conditions $u^{**}(1, t_n + \tau)$ for all τ with $0 \leq \tau \leq k$. This can be done in the standard way. We will use the fact that we know the true solution $u(\tau, t_{n+1}) = u^{**}(x, t_{n+1})$ and expand about t_{n+1} :

$$\begin{aligned} u^{**}(1, t_n + \tau) &= u^{**}(1, t_{n+1}) + (\tau - k)u_t^{**}(1, t_{n+1}) + \frac{1}{2}(\tau - k)^2 u_{tt}^{**}(1, t_{n+1}) + \dots \\ &= u^{**}(1, t_{n+1}) + (\tau - k)\epsilon u_{xx}^{**}(1, t_{n+1}) + \frac{1}{2}(\tau - k)^2 \epsilon^2 u_{xxxx}^{**}(1, t_{n+1}) + \dots \end{aligned}$$

This can be evaluated directly using the steady-state solution (5.21). We find that the proper boundary conditions are

$$u^{**}(1, t_n + \tau) = 1 - e^{(\tau - k)/\epsilon}.$$

This is shown for $k = 1$ in Figure 5.6. Note that the boundary value remains nearly constant through most of the time interval. Only at the end does it suddenly drop to the final value $u^{**}(1, t_{n+1}) = 0$. This explains why the resulting solution is not "over-diffused" when the true solution operator is used. No diffusion occurs until near the end of the time interval, for $k - \epsilon \leq \tau \leq k$, and the length of this interval is independent of k , as it must be since the resulting true steady-state solution is independent of k .

This shows where the time-split method breaks down for large k . When Crank-Nicolson is applied in the second step the correct boundary values are used at the ends of the time interval, but no account has been taken of the nonsmooth behavior of $u^{**}(1, t_n + \tau)$ for $0 < \tau < k$. Since Crank-Nicolson is only accurate for smooth boundary data, we get poor results. It is as if we had used the exact solution operator with smooth boundary data obtained by linearly interpolating between $u^{**}(1, t_n) = 1$ and $u^{**}(1, t_{n+1}) = 0$.

It appears that this difficulty with the time-split method can be avoided only by taking k sufficiently small. If $k < \epsilon$ then the boundary layer is not entirely shifted out

AD-A119 417

STANFORD UNIV CA DEPT OF COMPUTER SCIENCE
TIME-SPLIT METHODS FOR PARTIAL DIFFERENTIAL EQUATIONS.(U)
APR 82 R J LEVEQUE
STAN-CS-82-904

F/6 12/1

N00014-75-C-1132

NL

UNCLASSIFIED

2 2

FORM



END

DATE

FORMED

11 82

DTM

of the interval and the resulting true boundary conditions for u^{**} are much smoother. Moreover, reexamining (5.18) shows that we also want k/h smaller than was optimal for the Cauchy problem. From the steady-state solution (5.21) we see that

$$\|\partial_x^j u\| \approx 1/\epsilon^j.$$

Using this in (5.18) gives an optimal mesh ratio for the split method of

$$\lambda \approx 1$$

rather than $O(1/\epsilon)$ as was optimal for the Cauchy problem.

To summarize, we find that for boundary layer calculations we should take $k \approx h < \epsilon$, for the split method, just as we would for the unsplit method. The resulting solution is more accurate than that obtained with the unsplit method, by a factor of ϵ in the interior.

Intermediate boundary data at the inflow boundary. Now consider the unsteady boundary value problem (5.20) with boundary conditions

$$\begin{aligned} u(0, t) &= g(t) \\ u(1, t) &= 0. \end{aligned} \tag{5.23}$$

We must specify additional boundary values $U_0^*, U_1^*, \dots, U_{p-1}^*$. Consider the step from t_n to t_{n+1} and let u^* satisfy

$$\begin{aligned} u_t^* &= -u_x^*, & t &\geq t_n, \\ u^*(x, t_n) &= u(x, t_n), & 0 &\leq x \leq 1. \end{aligned} \tag{5.24}$$

Then we want

$$\begin{aligned} U_j^* &= u^*(jh, t_n + k) \\ &= u^*(0, t_n + k - jh). \end{aligned}$$

Let $\tau = k - jh$. Expanding about $u^*(0, t_n)$ and using (5.24),

$$\begin{aligned} U_j^* &= u^*(0, t_n) + \tau u_t^*(0, t_n) + \frac{1}{2}\tau^2 u_{tt}^*(0, t_n) + O(k^3) \\ &= u^*(0, t_n) - \tau u_x^*(0, t_n) + \frac{1}{2}\tau^2 u_{xx}^*(0, t_n) + O(k^3) \\ &= u(0, t_n) - \tau u_x(0, t_n) + \frac{1}{2}\tau^2 u_{xx}(0, t_n) + O(k^3). \end{aligned} \tag{5.25}$$

In order to use the given boundary data (5.23) we wish to replace x -derivatives of u by t -derivatives using (5.20). After some manipulations we find that

$$u_{tt} = u_{xx} - 2\epsilon u_{xxx} + \epsilon^2 u_{xxxx}$$

and so

$$\begin{aligned} u_x &= -u_t + \epsilon u_{xx} \\ &= -u_t + \epsilon u_{tt} + 2\epsilon^2 u_{xxx} - \epsilon^3 u_{xxxx} \\ &= -u_t + \epsilon u_{tt} - 2\epsilon^2 u_{ttt} + O(\epsilon^3). \end{aligned}$$

Continuing to replace x -derivatives by t -derivatives, we obtain a power series in ϵ which involves only time derivatives. Similarly we find that

$$u_{xx} = u_{tt} - 2\epsilon u_{ttt} + O(\epsilon^2).$$

Using these expressions in (5.25) gives the desired expansion:

$$\begin{aligned} U_j^* &= u(0, t_n) - \tau(-u_t + \epsilon u_{tt} - 2\epsilon^2 u_{ttt} + \dots) + \frac{1}{2}\tau^2(u_{tt} - 2\epsilon u_{ttt} + \dots) \\ &= g(0, t_n + \tau) - \tau\epsilon g''(t_n) + (2\tau\epsilon^2 - \tau^2\epsilon)g'''(t_n) + O(\epsilon^3 k + k^2\epsilon^2). \end{aligned} \quad (5.26)$$

Note that this approach works only when $\epsilon < 1$.

Example 5.2. Consider (5.20) with $\epsilon = 10^{-3}$, initial conditions

$$u(x, 0) = 1 - x,$$

and boundary conditions

$$\begin{aligned} u(0, t) &= g(t) = 1 + 0.1 \sin(2\pi t) \\ u(1, t) &= 0. \end{aligned}$$

Figure 5.7 shows the solution at time $t = 2$ using the unsplit method with $k = h = 10^{-2}$. Figure 5.8 shows the results with the time-split method (5.10) again with $k = h = 10^{-2}$ ($p = 1$) using (5.26) for U_0^* . Note that the oscillations in the boundary layer have disappeared. Moreover the interior solution is more accurate by a factor of ϵ , as can be seen in the accompanying error plots.

5.4. The Peaceman-Rachford ADI method for parabolic problems.

As a final example we will derive intermediate boundary conditions for the Peaceman-Rachford method (1.42) by viewing this as a time-split method for the problem $u_t = u_{xx} + u_{yy}$ with the splitting (1.43). We consider the problem on the unit square $0 \leq x \leq 1$, $0 \leq y \leq 1$ and assume that Dirichlet boundary data is supplied at all points on the boundary. Since U^* is differenced only in the x -direction in (1.42), we need to generate intermediate boundary data only on the boundaries $x = 0$ and $x = 1$. We will consider only the boundary at $x = 0$. The other boundary is completely analogous.

The given boundary data is

$$u(0, y, t) = g(y, t). \quad (5.27)$$

We seek to determine $U_{0,m}^* \approx u^*(0, mh, t_n + k)$ where u^* is the solution to the first subproblem from (1.43),

$$u_t^* = \frac{1}{2}(u_{xx}^* + u_{yy}^*) + \frac{1}{8}k(u_{xxx}^* - u_{yyy}^*), \quad (5.28)$$

with initial conditions

$$u^*(x, y, t_n) = u(x, y, t_n), \quad \forall x, y. \quad (5.29)$$

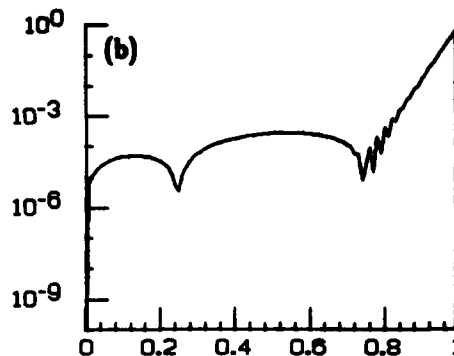
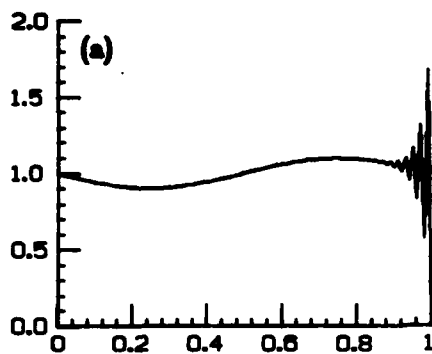


FIG. 5.7. (a) Numerical solution for Example 5.2 with $\epsilon = 10^{-3}$ obtained using the unsplit method with $k = h = 10^{-2}$. (b) Errors in the computed solution.

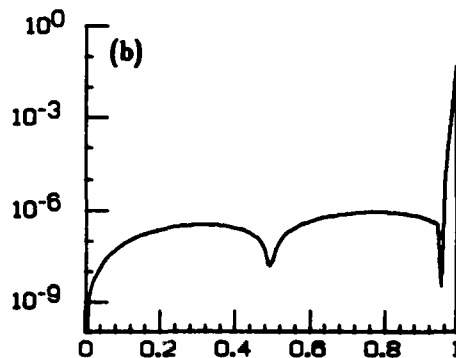
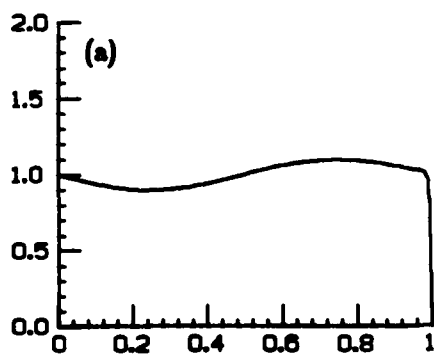


FIG. 5.8. (a) Numerical solution for Example 5.2 with $\epsilon = 10^{-3}$ obtained using the split method with $k = h = 10^{-2}$. (b) Errors in the computed solution.

Differentiating (5.28) shows that

$$u_{tt}^* = \frac{1}{2}(u_{xxxx}^* + 2u_{xyyy}^* + u_{yyyy}^*) + O(k)$$

and so, proceeding as in Chapter 4,

$$\begin{aligned} u^*(0, mh, t_n + k) &= u^*(0, mh, t_n) + ku_t^*(0, mh, t_n) + \frac{1}{2}k^2 u_{tt}^*(0, mh, t_n) + O(k^3) \\ &= u^* + k\left[\frac{1}{2}(u_{xx}^* + u_{yy}^*) + \frac{1}{8}k(u_{xxxx}^* - u_{yyyy}^*)\right] \\ &\quad + \frac{1}{2}k^2\left[\frac{1}{2}(u_{xxxx}^* + 2u_{xyyy}^* + u_{yyyy}^*) + O(k)\right] + O(k^3). \end{aligned}$$

In view of the initial conditions (5.29), we can replace u^* by u everywhere on the righthand side of the last equation. By expanding $u(0, mh, t_n + k/2)$ about $u(0, mh, t_n)$ and comparing this with what results above, we find that

$$u^*(0, mh, t_n + k) = u(0, mh, t_n + k/2) + \frac{1}{8}k^2(u_{xxxx}(0, mh, t_n) - u_{yyyy}(0, mh, t_n)) + O(k^3). \quad (5.30)$$

We retain $O(k^2)$ global accuracy provided the boundary conditions have $O(k^2)$ local accuracy. We can thus neglect the $O(k^2)$ terms in (5.30) and take

$$U_{0,m}^* = u(0, mh, t_n + k/2) = g_m^{n+1/2}$$

where $g_m^{n+1/2} = g(mh, t_n + k/2)$. Higher order accuracy at the boundary can be obtained by including the next term as well. The u_{xxxx} term cannot be calculated directly from the given boundary conditions (5.27). However, from the original differential equation we find that

$$u_{tt} = u_{xxxx} + 2u_{xyyy} + u_{yyyy},$$

$$u_{tyy} = u_{xyyy} + u_{yyyy},$$

and so

$$u_{tt} - 2u_{tyy} = u_{xxxx} - u_{yyyy}.$$

Since u_{tt} and u_{tyy} can both be computed along the boundary, we can use this expression in place of $u_{xxxx} - u_{yyyy}$ in (5.30), giving the $O(k^3)$ boundary conditions

$$U_{0,m}^* = g(mh, t_n + k/2) + \frac{1}{8}k^2(g_{tt}(mh, t_n) - 2g_{tyy}(mh, t_n)). \quad (5.31)$$

It is interesting to note that if we approximate the derivatives of g appearing in (5.31) by $O(k^3)$ accurate finite differences of g_m^n and g_m^{n+1} at the meshpoints, we obtain

$$\begin{aligned} U_{0,m}^* &= \frac{1}{2}(g_m^n + g_m^{n+1}) - \frac{1}{4}kD_{+y}D_{-y}(g_m^{n+1} - g_m^n) \\ &= \frac{1}{2}\left(1 - \frac{1}{2}kD_{+y}D_{-y}\right)g_m^{n+1} + \frac{1}{2}\left(1 + \frac{1}{2}kD_{+y}D_{-y}\right)g_m^n. \end{aligned}$$

These are precisely the standard boundary conditions for the Peaceman-Rachford method as derived by Fairweather and Mitchell[19] using different techniques.

For irregular regions it is not in general possible to replace the x - and y -derivatives in (5.30) by tangential and time derivatives which can be evaluated directly from the given boundary data. However, the expansion (5.30) is still valid and one-sided finite differences could be used to approximate the fourth derivatives.

REFERENCES

- [1] S. Abarbanel and D. Gottlieb, Optimal time splitting for two and three dimensional Navier-Stokes equations with mixed derivatives, *J. Comp. Phys.* 41(1981), pp. 1-33.
- [2] R. M. Beam and R. F. Warming, An implicit finite-difference algorithm for hyperbolic systems in conservation-law form, *J. Comp. Phys.* 22(1976) pp. 87-110.
- [3] G. Browning, A. Kasahara and H.-O. Kreiss, Initialization of the primitive equations by the bounded derivative method, *J. Atmospheric Sci.* 37(1980), pp. 1424-1436.
- [4] J. Certaine, The solution of ordinary differential equations with large time constants, in *Mathematical Methods for Digital Computers* (A. Ralston and H. S. Wilf, eds.), Wiley, New York, 1960, pp 128- 132.
- [5] M. Crandall and A. Majda, The method of fractional steps for conservation laws, *Numer. Math.* 34(1980), pp. 285-314.
- [6] J. Douglas, Jr. , On the numerical integration of $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \frac{\partial u}{\partial t}$ by implicit methods, *J. Soc. Indust. Appl. Math.* 3(1955), pp. 42-65.
- [7] J. Douglas, Jr. and J. E. Gunn, A general formulation of alternating direction methods, Part I. Parabolic and hyperbolic problems, *Numer. Math.* 6(1964), pp. 428-453.
- [8] J. Douglas, Jr. and H. H. Rachford, Jr., On the numerical solution of the heat conduction problem in two and three space variables, *Trans. of the Amer. Math. Soc.* 82(1956), pp. 421-439.
- [9] J. Douglas, Jr. and T. F. Russell, Numerical methods for convection-dominated diffusion problems based on combining the method of characteristics with finite element or finite difference procedures, to appear.
- [10] D. L. Dwoyer and F. C. Thames, Accuracy and stability of time-split finite-difference schemes, 5th AIAA Computational Fluid Dynamics Conference, Palo Alto, CA, 1981.
- [11] E. G. D'Yakonov, The method of alternating directions in the solution of finite-difference equations, *Dokl. Akad. Nauk SSSR* 138(1961), pp. 271-274 (Russian).
- [12] E. G. D'Yakonov, Some difference schemes for solving boundary problems, *USSR Comp. Math.* 2(1963), pp. 55-77.
- [13] E. G. D'Yakonov, Difference schemes with a "disintegrating" operator for multi-dimensional problems, *USSR Comp. Math.* 2(1963), pp. 581-607.
- [14] E. G. D'Yakonov, Difference schemes with separable operator for general parabolic equations of second order with variable coefficients, *USSR Comp. Math.* 4#2(1964), pp. 92-110.
- [15] R. E. Ewing and T. F. Russell, Multistep Galerkin methods along characteristics for convection-diffusion problems, to appear.
- [16] T. Elvius and A. Sundström, Computationally efficient schemes and boundary conditions for a fine-mesh barotropic model based on the shallow-water equations, *Tellus* 25(1973), pp. 132-156.
- [17] B. Engquist, B. Gustafsson and J. Vreeburg, Numerical solution of a PDE system describing a catalytic converter, *J. Comp. Phys.* 27(1978), pp. 295-314.

- [18] R. E. Ewing and T. F. Russell, Multistep Galerkin methods along characteristics for convection-diffusion problems, to appear.
- [19] G. Fairweather, and A. R. Mitchell, A new computational procedure for ADI methods, *SIAM J. Numer. Anal.* 4(1967), pp. 163-170.
- [20] A. J. Gadd; A split explicit integration scheme for numerical weather prediction, *Quart. J. Royal Met. Soc.* 104(1978), pp. 569-582.
- [21] M. Goldberg and E. Tadmor, Scheme-independent stability criteria for difference approximations of hyperbolic initial-boundary value problems. I, *Math. Comp.* 32(1978), pp. 1097-1107.
- [22] M. Goldberg and E. Tadmor, Scheme-independent stability criteria for difference approximations of hyperbolic initial-boundary value problems. II, *Math. Comp.* 36(1981), pp. 603-626.
- [23] D. Gottlieb, Strang-type difference schemes for multidimensional problems, *SIAM J. Numer. Anal.* 9(1972), pp. 650-661.
- [24] A. R. Gourlay, Splitting methods for time dependent partial differential equations, in *The State of the Art in Numerical Analysis* (D. Jacobs, ed.), Academic Press, 1977.
- [25] A. R. Gourlay and A. R. Mitchell, A classification of split difference methods for hyperbolic equations in several space dimensions, *SIAM J. Numer. Anal.* 6(1969), pp. 62-71.
- [26] A. R. Gourlay and A. R. Mitchell, The equivalence of certain alternating direction and locally one-dimensional difference methods, *SIAM J. Numer. Anal.* 6(1969), pp. 37-46.
- [27] A. R. Gourlay and J. Ll. Morris, A multistep formulation of the optimized Lax-Wendroff method for nonlinear hyperbolic systems in two space variables, *Math. Comp.* 24(1988), pp. 715-720.
- [28] B. Gustafsson, The convergence rate for difference approximations to mixed initial boundary value problems, *Math. Comp.* 29(1975) pp. 396-406.
- [29] B. Gustafsson, An alternating direction implicit method for solving the shallow water equations, *J. Comp. Phys.* 7(1971), pp. 239-254.
- [30] B. Gustafsson, H.-O. Kreiss and A. Sundström, Stability theory of difference approximations for mixed initial boundary value problems. II, *Math. Comp.* 26(1972), pp. 649-685.
- [31] F. H. Harlow and A. A. Amsden, a numerical fluid dynamics calculation method for all flow speeds, *J. Comp. Phys.* 8(1971), pp. 197-213.
- [32] H.-O. Kreiss, Problems on different time scales for ordinary differential equations, *SIAM J. Numer. Anal.* 16(1979), pp. 980-998.
- [33] H.-O. Kreiss, Problems with different time scales for partial differential equations, *Comm. Pure and Appl. Math.* 33(1980), pp. 399-439.
- [34] J. D. Lawson and J. Ll. Morris, A review of splitting methods, Report CS-74-09, Department of Applied Analysis and Computer Science, University of Waterloo, 1974.

- [35] R. W. MacCormack, A rapid solver for hyperbolic systems of equations, in *Proceedings of the Fifth International Conference on Numerical Methods in Fluid Dynamics*, Springer Lecture Notes in Physics 59 (A. I. van de Vooren and P. J. Zandbergen, eds.), 1976.
- [36] R. W. MacCormack, An efficient explicit-implicit-characteristic method for solving the compressible Navier-Stokes equations, in *Computational Fluid Dynamics*, SIAM-AMS Proceedings of the Symposium on Computational Fluid Dynamics, New York, 1977, pp. 130-155.
- [37] G. Majda, Filtering techniques for oscillatory stiff ordinary differential equations, to appear.
- [38] G. I. Marchuk, Numerical weather forecasting on the sphere, *Dokl. Akad. Nauk SSSR* 158(1964), pp. 810-813.
- [39] G. I. Marchuk, On the theory of the splitting-up method, in *Numerical solution of Partial Differential Equations II*, *Synspade* 70 (B. Hubbard, ed.), Academic Press, 1971.
- [40] G. I. Marchuk, *Methods of Numerical Mathematics*, Springer-Verlag, 1975.
- [41] A. R. Mitchell and D. F. Griffiths, *The Finite Difference Method in Partial Differential Equations*, Wiley, 1980.
- [42] J. Ll. Morris and A. R. Gourlay, Modified locally one dimensional methods for parabolic partial differential equations in two space variables, *J. Inst. Maths. Applics.* 12(1973), pp. 349-353.
- [43] J. Olinger, Constructing stable difference methods for hyperbolic equations, in *Numerical methods for partial differential equations*, (S. V. Parter, ed.), Academic Press 1979, pp. 255-271.
- [44] R. E. O'Malley and L. R. Anderson, Singular perturbations, order reduction, and decoupling of large scale systems, in *Numerical Analysis of Singular Perturbation Problems*, (Hemkes and Miller, eds.), Academic Press, 1979, pp. 317-998.
- [45] D. W. Peaceman and H. H. Rachford, Jr., The numerical solution of parabolic and elliptic differential equations, *J. Soc. Indust. Appl. Math.* 3(1955), pp. 28-41.
- [46] R. D. Richtmyer and K. W. Morton, *Difference Methods for Initial-Value Problems*, Interscience Tracts in Pure and Applied Math. No. 4, Wiley, 1967.
- [47] T. F. Russell, Finite elements with characteristics for two-component incompressible miscible displacement, SPE 10500, Society of Petroleum Engineers, Dallas, 1982.
- [48] A. A. Samarskiĭ and V. B. Andreev, Iteration alternating direction schemes for the numerical solution of the dirichlet problem, *Ž. Vyčisl. Mat. i Mat. Fiz.* 4(1964), pp. 1025-1036.
- [49] G. Strang, On the construction and comparison of difference schemes. *SIAM J. Numer. Anal.* 5(1968), pp. 506-517.
- [50] J. Strikwerda, A time-split difference scheme for the compressible Navier-Stokes equations with applications to flows in slotted nozzles, ICASE Report No. 80-27, 1980.
- [51] V. Thomée, Stability theory for partial differential operators, *SIAM Review* 11 (1969), pp. 152-195.

- [52] E. Turkel and G. Zwas, Explicit large time-step schemes for the shallow water equations, AICA Proceedings No. 3(1979), pp. 65-69.
- [53] E. Varoglu and W. D. L. Finn, Space-time finite elements incorporating characteristics for the Burgers' equation, Int. J. Num. Meth. Eng. 16(1980), pp. 171-184.
- [54] R. F. Warming and R. M. Beam, On the construction and application of implicit factored schemes for conservation laws, in *Computational Fluid dynamics*, SIAM-AMS Proceedings 11(1978), pp. 85-129.
- [55] N. N. Yanenko, A difference method of solution in the case of the multidimensional equation of heat conduction, Dokl. Akad. Nauk SSSR 125(1959), pp. 1207-1210 (Russian).
- [56] N. N. Yanenko, *The Method of Fractional Steps*, Springer-Verlag, 1971.

END

DATE
FILMED

11-82

DTIC