MULTILEVEL METHODS FOR THE SOLUTION
OF ADVECTION-DOMINATED ELLIPTIC PROBLEMS
ON COMPOSITE GRIDS

by

James Scott Otto

B.S., Georgia State University, 1983
M.A., University of New Mexico, 1985

A thesis submitted to the
Faculty of the Graduate School of the
University of Colorado at Denver
in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy

Applied Mathematics

1992
This thesis for the Doctor of Philosophy
degree by
James Scott Otto
has been approved for the
Department of
Mathematics
by
Thomas A. Manteuffel
Jan Mandel
Stephen F. McCormick
Steven A. Pruess
Steven A. Pruess
Thomas F. Russell

4/9/92
Date
Otto, James Scott (Ph.D., Applied Mathematics)

Multilevel Methods for the Solution of Advection-Dominated Elliptic Problems on Composite Grids

Thesis directed by Professor Thomas A. Manteuffel

ABSTRACT

We analyze the use of multilevel algorithms in the solution of advection-dominated advection-diffusion equations in one and two dimensions. In particular, we study the behavior of the FAC (Fast Adaptive Composite) scheme as applied to a composite grid discretization of the problem, where the discretization type is allowed to vary (from upwind to centered differences, for example) on the components of the grid. In one dimension, the analysis leads to the interpretation of a two-level version of FAC as a direct method. This analysis also provides insight into the behavior of the algorithm, and guidance for its implementation, in two dimensions.

In two dimensions, we suppose the problem to have been transformed, by an orthogonal coordinate mapping, in such a way that its characteristics are aligned with a Cartesian product grid. Various discretization strategies for this problem are studied, with emphasis placed on a finite volume method. Upwind, centered and higher-order upwind types of finite volume discretization are considered. This method is attractive in that its matrix stencils capture all of the advection of the discrete problem in their main diagonal blocks. As a result, the block Jacobi relaxation is an excellent smoother for multigrid as applied to problems on uniform subgrids. This allows us to formulate highly effective multilevel algorithms for the
global equations that preserve the inherent parallel nature of the strongly advection-dominated problem. The ultimate numerical method developed uses two-level FAC as an algebraic solver in a nested way to obtain an optimally efficient full multigrid-like algorithm.

This abstract accurately represents the content of the candidate’s thesis. I recommend its publication.

Signed
Thomas A. Manteuffel
ACKNOWLEDGEMENTS

I wish to thank, first and foremost, my advisor, Tom Manteuffel, for his support, academic and financial, that has allowed me to take part in a variety of interesting areas of research which he has introduced me to, including the one of this dissertation. Secondly, I would like to thank Steve McCormick, who has been a consultant to this work since its beginning, and whose expertise in multilevel algorithms has proved invaluable. I would also like to thank my other committee members, Jan Mandel, Steve Pruess, and Tom Russell, for their participation. Particular thanks goes to Debbie Wangerin for her support, friendship, and sense of humor. Thanks, also, to Bill Briggs and Roland Sweet for the initial support I received as a research assistant in Denver. Finally, I wish to thank the people who nurtured my interest in mathematics, those in the Department of Mathematics at Georgia State University, and in particular, of course, George Davis who inspired me to study numerical linear algebra.
CONTENTS

Chapter

1. Introduction ......................................................... 1

1.1 Notation ......................................................... 6

2. Problem Description and Overview of Solution Methods .......... 7

3. Discretization ....................................................... 16

3.1 Description of Composite Grids .............................. 17

3.2 Discretization Methods ......................................... 20

3.2.1 Finite Volume Discretization in One Dimension ......... 23

3.2.2 Discretization by Element Methods ...................... 27

3.2.3 Finite Volume Discretization in Two Dimensions ....... 36

3.2.4 A Comparison of Discretization Methods ............... 41

3.3 Accuracy of the Discretizations .............................. 43

3.3.1 Sample Problems in Two Dimensions .................... 49

4. Convergence of FAC ............................................. 59

4.1 The Two-Level Method ........................................... 60

4.2 Convergence in One Dimension ................................ 66

4.2.1 The Case of a Single Refinement Region ............... 67

4.2.2 Quality of Approximations ................................ 80

4.2.3 Multiple Refinement Regions ............................. 85

4.3 Convergence in Two Dimensions .............................. 87

5. Alternative FAC Algorithms ..................................... 97

5.1 FAC as Preconditioning ....................................... 97

5.2 A Comparison of FAC and Domain Decomposition ........ 99
5.3 Modifications Based on Multigrid ..................... 103
  5.3.1 The Significance of Line Relaxation ................. 104
  5.3.2 FAC and Nested Iteration .......................... 114
6. Summary of Results ....................................... 125
References .................................................. 132
1. Introduction

The topic of this thesis is the efficient solution, by iterative methods, of advection-dominated advection-diffusion equations. Although we consider only elliptic second order equations, the algorithms presented in this study may also be used to solve the problems that arise at each time step in implicit methods for transient equations with nearly hyperbolic character. We consider solving equations whose true solutions are such that the use of local grid refinement in their discretization is desirable. The type of numerical methods we consider for solving the discretized problem are based on multilevel techniques; the ultimate numerical solution method that results from this study is a full multigrid-like algorithm that preserves the inherent parallel nature of the strongly advection-dominated elliptic partial differential equation.

Multilevel, or multigrid, techniques have proven to be efficient methods for the solution of elliptic partial differential equations. However, to a large extent, the success of classical multigrid algorithms depend on the degree of ellipticity of the problem along with the type of discretization being employed. These two factors go hand-in-hand, and what is more accurate to say is that the success of the method depends on the degree of ellipticity of the discretized version of the partial differential equation. Much of the content of this dissertation is devoted to studying the effects that ellipticity and discretization type have on the efficiency of conventional multigrid methods in the context of an advection-dominated problem discretized on a composite grid. Although we study a one dimensional problem, largely as a means of gaining insight, particular emphasis is placed on solving the
two-dimensional advection-diffusion equation in a special form.

Following Chin and Manteuffel [14], in two dimensions we assume a coordinate mapping has been used to transform the problem to one where the flow velocity is aligned with gridlines in a Cartesian grid. As shown in [14], with the problem in this form, line relaxation methods provide robust solvers for the discrete problem. One way this work extends that in [14] is by allowing multigrid, based on line Jacobi relaxation, to play the role of the iterative solver. We also include the use of local refinement in the discretization. Our choice of relaxation strategy allows us to retain the parallelism of the strongly advection-dominated problem in efficient solvers for both coarse and fine composite grid subproblems.

The particular form of multigrid we study is the FAC (Fast Adaptive Composite) method introduced by McCormick [32] to taylor multigrid to the structure of composite grid problems. We analyze the behavior of a two-level version of this method as applied to the advection-diffusion problem in one and two dimensions. A detailed analysis of FAC convergence in one dimension yields an interpretation of the scheme as a direct method. Perhaps more importantly, it provides insight in the appropriate choice of grid and discretization type to be used on the coarse level. This information is used in determining appropriate strategies for the use of FAC as applied to the problem in two dimensions, whether it is used in its two-level form or as a full multigrid-like algorithm.

A description of the contents of this dissertation follows. Chapter 2 provides an overview of the problem to be solved and the solution methods used, and describes relevant related work. Chapter 3 introduces the composite grid and presents a summary of discretization methods for the one and two-dimensional problems. The structure and notation for the com-
posite grid used in this study are given in Section 3.1. Section 3.2 describes three discretization methods. A finite volume discretization is emphasized. For the most part, our description of the finite volume discretization is standard, however, we show how to adapt this method to the structure of the composite grid, borrowing a technique from finite element and finite volume element discretizations. These two discretizations are introduced in Section 3.2.2 as completely systematic methods and also as means of motivating the treatment of the interface portion of the finite volume discretization. Although the former methods are more systematic, the finite volume discretization offers advantages in terms of simplicity and ease of use.

Section 3.2.1 describes the finite volume discretization for the one-dimensional version of our problem. This discretization serves as the basis for the analysis of FAC convergence in one dimension later in Chapter 4. We notice that if the finite element subspace is chosen appropriately this method yields the same discrete problem as the finite element and finite volume element methods. The structure of a composite basis for this space is introduced. In Section 3.2.2 the generalization of this basis to two dimensions is made. Use of this basis allows the simple handling of the nonuniformity at the interface when discretizing by the element methods. In Section 3.2.3 the two-dimensional finite volume discretization is introduced. This method serves as the basis for later studies of FAC convergence in two dimensions. It is based on the so-called control volume approach described in [36].

Section 3.2.4 compares briefly the relative merits of the various discretizations treated in this study, and also identifies appropriate grid transfer strategies.
In Section 3.3 we discuss the accuracy of the discretizations in two dimensions. We cite some recent results on the accuracy of finite volume and finite volume element methods for diffusion problems, analyze the accuracy of the finite element method for the advection-diffusion problem and provide examples demonstrating the accuracy of the particular finite volume method used here.

Chapter 4 is a study of convergence of the two-level FAC scheme applied to the finite volume discretization of the one- and two-dimensional advection-diffusion problems. In Section 4.1 it is shown that, with a certain compatibility assumption between the coarse and fine composite grid operators, this method has the form of an interface preconditioning, i.e., the interface component of the residual satisfies

\[ r_{I}^{i+1} = (I - \hat{L}_{I} \hat{A}_{I}^{-1}) r_{I}^{i}, \]

where, \( \hat{L}_{I} \) is the Schur complement in the composite grid operator, and \( \hat{A}_{I} \) is a preconditioner. It turns out that, for FAC, the latter is the Schur complement in the coarse composite grid operator.

Section 4.2 studies convergence of the two-level scheme in one dimension, and contains the main theoretical results of this study. Our convergence analysis proceeds by an investigation of the spectrum of \( B_{I} = \hat{L}_{I} \hat{A}_{I}^{-1} \). In Section 4.2.1 this spectrum is analyzed in the case of a single refinement region, allowing for the used of mixed discretization types on the composite grid. We also take into consideration the effect of the type of discretization used on the coarse grid. The analysis shows that good convergence results can be obtained with significant jumps in refinement between the components of the grid, and, in particular, motivates the use of a global coarse grid on the coarse level. In Section 4.2.3 these results are generalized to
the case of multiple refinement regions.

Section 4.3 is a study of the two-level FAC scheme applied in two dimensions. Taking a cue from the one-dimensional analysis, the global coarse grid is used at the coarse level in specific FAC implementations for problems that involve upwind, centered, and higher-order upwind discretizations on the region of refinement. Generally, the results of this section indicate that the two-level scheme provides a good algebraic solver for the advection-dominated problem. However, better efficiency can be obtained by using this scheme in a more sophisticated way.

In Chapter 5 we consider two ways of using FAC more efficiently. Section 5.1 views the two-level scheme as a preconditioned iterative method, and suggests using polynomial acceleration to improve its convergence. This is the method of Schur complement domain decomposition and corresponds to the use of FAC by domain decomposition practitioners. Section 5.2 is a brief comparison of two-level FAC with Schur complement domain decomposition.

Rather than use the preconditioning perspective, that suggests solving the composite grid equations algebraically, in Section 5.3 we take the more efficient approach of using full multigrid to solve the equations to the level of accuracy of the discretization. An important component of our implementation of this approach is the use of strategies based on block Jacobi relaxation for solving problems on the subregions. Section 5.3.1 contains an analysis of this type of relaxation (as relaxation, and as a multigrid smoother) for upwind and centered discretizations of the particular problem we study.

In Section 5.3.2 nested iteration and the two-level FAC scheme are combined to obtain a full multigrid version of FAC. Computational results
presented in that section show that, for the various discretization mixtures, only one or two FAC iterations are required on each of the composite levels to obtain an approximation by this version of FAC with accuracy comparable to the exact solution of the composite grid equations. Furthermore, we show that this is the case with complete solves of problems on the subregions replaced by inexpensive approximations. This is significant because practitioners of the two-level approach have noted dramatic degradation in global convergence when subproblems on the refinement region are solved approximately. The method presented here, however, yields accurate solutions with optimal efficiency on this region, i.e., the total amount of work there is comparable to that of a full multigrid iteration.

Chapter 6 is a summary of the results of this dissertation. It also provides recommendations for their practical utilization.

1.1 Notation

For the most part, the following notational conventions are used in this dissertation. Particular usage and exceptions should be clear from context.

\[ A, L, X, \ldots \] Matrices or differential operators.
\[ x_{ij} \] The \( ij \)-th entry of the matrix \( X \).
\[ \tilde{x}_{ij} \] The \( ij \)-th entry of \( X^{-1} \).
\[ A, L, \ldots \] Composite matrices.
\[ z, r, \ldots \] Vectors.
\[ z, \xi, \ldots \] Composite vectors.
\[ a, f, \alpha, \phi, \ldots \] Scalars or functions.
\[ \Phi, \Psi \] Finite dimensional function spaces.
\[ H^k \] Sobolev space of order \( k \).
\[ DetX \] The determinant of the square matrix \( X \).
\[ X_{ij} \] The cofactor associated with \( x_{ij} \).
\[ X_{j \times j} \] The \( j \times j \) submatrix located in the lower right corner of the square matrix \( X \).
2. Problem Description and Overview of Solution Methods

This chapter is devoted to describing the problem to be solved, and to providing an overview of the solution methods and analysis used in this study.

Although the theme of multigrid applied to advection-dominated problems is central to this study, there are a variety of ways of viewing this work in which the significance of multigrid varies. One way that is essential is to view the study as an extension of the work in [14] by Chin and Manteuffel. There, the authors consider the solution of the two-dimensional advection-diffusion equation

\[ -\epsilon \Delta u(x, y) + au_x(x, y) + bu_y(x, y) + du(x, y) = g(x, y), \]  

(2.1)

where \( \epsilon \) is a small positive constant. Notice that if \( \epsilon \) is equal to zero in (2.1), then the equation is pure advection and is easily solved, in principle, by integrating along its characteristics. In practice, however, the characteristics may be difficult to track and, in any case, we are interested in solving the problem with \( \epsilon \) small but positive. Nevertheless, when \( \epsilon \) is sufficiently small one feels that advantage may still be gained from the problem's nearly one-dimensional character. In [14] the authors do this by using an orthogonal coordinate transformation that is induced by the characteristics. Under appropriate assumptions, the latter transforms (2.1) into the equation

\[ -\epsilon \Delta u(x, y) + u_x(x, y) + cu(x, y) = f(x, y). \]  

(2.2)

With this form, the characteristics are simply \( y = \text{constant} \) and any procedure that employs solving along the characteristic directions will be fa-
cilitated by having the characteristics aligned with gridlines in a Cartesian product grid. In [14] the authors analyze the spectrum of the block SOR iteration matrix associated with a centered difference discretization of (2.2) (with $c = 0$). This iteration is based on alternative matrix splittings that use either the $x$- or $y$-derivative terms of the discretization as the principal part of the splitting. The relative size of the diffusion coefficient is measured by the ratio $\gamma = h / (2c)$, where $h$ is the meshsize. When the problem is advection dominated (i.e., for $\gamma \geq 1$), and with relaxation performed along lines in the $x$-direction, they obtain a bound on the spectral radius of the iteration matrix that is less than $1/5$.

We also consider the numerical solution of (2.2). One way that this work differs from [14] is that multigrid plays the role of the iterative solver. The algorithms used in this study also retain the essential parallelizability of the problem in this form by using block Jacobi, or line relaxation, as the smoother on which multigrid is based.

Another way we extend the work in [14] is by including local refinement in the discretization. The use of local refinement may be called for due to various anomalies of the problem. These may include abrupt changes in the right-hand side [6] or the existence of boundary layers [40], [37].

Also, we emphasize the use of upwind rather than centered differences, although we consider employing various discretization types on the subregions of a composite grid; use of the composite grid structure makes it particularly easy to interface different discretizations on subregions. It also allows one to interface appropriate solution techniques for these subproblems. The propitious aspects of concentrating effort locally in the context of a composite problem are summarized succinctly in [44, p.2]:

This approach offers a number of advantages. An existing code
can be upgraded, in order to increase the accuracy locally, without a radical redesign of the data structures, etc., since we can use the old code to solve one or several standard problems. Issues of data structures and geometry are generally simpler if we design programs for composite mesh problems in terms of simpler standard problems. The use of simple standard problems also tends to improve the performance of the programs on vector machines. Finally, we note that if each of the standard problems has appreciatively fewer degrees of freedom than the composite model, then we might benefit from solving a number of smaller problems rather than one large one. In other words, we might view our approach as a divide-and-conquer strategy.

The composite structure also plays a fundamental role in the convenient economization of multigrid to problems requiring local refinement. Multigrid in this context is called the FABC (Fast Adaptive Composite) method, which was introduced by McCormick in [32] and is treated in depth in [33]. It is the particular form of multigrid studied here. The economization of the method lies in its avoidance of unnecessary computations with respect to the coarse component of the grid. Its ease of use derives from the fact that most of the computations involve solving familiar problems on uniform subgrids.

Local uniform refinement has been abandoned by some multigrid researchers [2], [34], [38] in order to obtain more flexibility in using refinement. In particular, the methods of [34] and [38] allow for quite arbitrary refinements, of which local uniform refinement can be seen as a special case. When arbitrary refinement is allowed, however, the actual refinement typically is not specified before computations begin, but rather is determined dynamically as computations proceed. Although such an approach may be attractive because of its high degree of generality, it may not be appropriate in certain contexts. For example, in many instances the user has a good idea of the location and the degree of refinement needed for a particular problem, so the effort expended in determining the refine-
ment dynamically would be wasted. Because of the dynamic nature of the corresponding algorithms, they are difficult to parallelize efficiently. Also, because of the lack of a predetermined grid structure, analysis of these methods is difficult. Finally, the use of arbitrary refinement may actually limit the capability to represent a solution accurately in the sense that such an approach necessitates the use of a single discretization method, whereas it may be desirable to use methods of different orders in different subregions of the domain, for example. The use of local uniform refinement does not have these drawbacks. Also, the use of a composite grid allows for the use of several levels of immediate refinement on a given subregion. This can be important if the solution changes rapidly. We emphasize such behavior in this study and investigate the effects that using multiple levels of refinement has on FAC convergence. This is an important aspect of this study, since FAC (albeit under different names) is currently being used in this way to solve practical problems. For example, in [6] the use of the closely related BEPS method in oil reservoir simulation is discussed. Multiple refinement levels are used there to accommodate the representation of injection wells by delta functions. We note that the algorithm appearing there, although equivalent to the two-level version of FAC, can be described without reference to multigrid. Indeed, the authors view the method as a preconditioned iterative method. Much of the content of this dissertation is devoted to analysis of the convergence of the two-level FAC scheme, so one further interpretation of this work is as an examination of the effectiveness of a preconditioner.

A composite grid (to be described more thoroughly in the next chapter) consists of three components: an interface serves as the boundary between the two true subregions of the grid, the coarse and fine patch components.
The latter component corresponds to a simply connected subregion of the domain where local refinement is deemed necessary. We consider three discretization methods in the context of the composite grid: finite element, finite volume element and finite volume, emphasizing the latter. For the most part, our description of the finite volume discretization is quite standard. However, we show how to adapt this method to the structure of the composite grid, introducing a novel way of computing patch-to-interface connections.

This technique involves interpolation of the interface (with the mesh-size of the coarse component) to the boundary of the fine patch, permitting the use of a standard stencil throughout the region of the fine patch. The technique is borrowed from finite element and finite volume element composite grid discretizations, where it occurs naturally. Although these two methods are attractive due to being highly systematic, the finite volume discretization offers advantages in terms of simplicity and ease of use. The matrix stencils obtained with this method are, on uniform grids, familiar five-point ones that correspond to either first-order upwind or second-order centered differences. An advantage of using this method is that, with the advection-diffusion problem in the form (2.2), the advection terms are isolated in the main tridiagonal portion of the (appropriately ordered) matrix stencil (this is not true for the element-type discretization methods). The significance of this is that, then, line Jacobi provides a robust relaxation method and multigrid smoother that preserves the inherent parallelism possessed by the problem as $\epsilon \to 0$. This is an important feature of our implementation of the FAC algorithm, however it remains invisible until well towards the end (Chapter 5) of this study. The reason for this is that our analysis focuses on a two-level version of FAC that proceeds by solving
subproblems on the components of the composite grid, and it is assumed that these problems are solved exactly.

We study convergence of this two-level version of FAC applied to the finite volume discretization of (2.2) and its one-dimensional counterpart. We show that when a natural coarse component compatibility between the coarse and fine level composite operators is satisfied, the convergence of the two-level FAC scheme is governed by convergence of the interface component of the residual. The iterates for this component can be shown to satisfy the relationship

$$r^{i+1}_f = (I - \hat{L}_f\hat{A}_f^{-1})r^i_f.$$

Here, $\hat{L}_f$ is the Schur complement in the composite grid operator, and $\hat{A}_f$ is the Schur complement corresponding to a coarse-level discretization. This is a distinguishing feature of the FAC method when viewed as an interface preconditioning: *it preconditions the Schur complement in the composite operator with the Schur complement in a coarse grid operator.*

Multigrid convergence theory for various discretizations on uniform grids of the one-dimensional advection-diffusion equation has been developed in [24] and in [3] using Fourier analysis. In [24] the author considers the two-level multigrid iteration, based on Gauss-Seidel smoothing, applied to upwind and centered finite difference discretizations. For the upwind case, a bound of $1/3$ on the spectral radius of the iteration matrix is obtained, independent of $\gamma$. For the centered case, good convergence rates are obtained by developing a strategy for introducing artificial diffusion as the grid levels vary. Corresponding results are obtained in [3], where a similar approach is taken with respect to a streamline diffusion finite element discretization. Convergence of the two-level scheme is proved for
sufficiently large artificial diffusion or a sufficiently fine grid. The results of our analysis of the composite grid problem are similar in character to these results for uniform grids.

Our analysis of the two-level FAC algorithm for the one-dimensional problem discretized on a composite grid proceeds by an investigation of the spectrum of $B_I = \hat{L}_I \hat{A}_I^{-1}$. We determine this spectrum in the case of a single refinement region, and also extend the analysis to the case of multiple refinement regions. This analysis covers various mixtures of upwind and centered type discretizations on the components of the grids. With upwinding is used on both levels, we show that the spectral radius of the iteration matrix $I - B_I$ is zero in suitable parameter ranges. This result applies to rather arbitrary choices of the grid used at the coarse level, including a global coarse grid (i.e., the uniform grid over the whole domain with the meshsize of the coarse component). For a discretization that uses upwind and centered differences, respectively, on the coarse and fine patch components of the composite grid, we show that with upwinding used on the coarse level, the spectral radius of the iteration approaches zero as the resolution is increased on the fine patch.

In our study of the two-level FAC scheme applied in two dimensions, the convergence behavior is seen to conform generally to the analysis for the one-dimensional problem. This is particularly true for a method that employs centered differencing on the fine patch. It is shown that this method works well with sufficient refinement on the patch. Also, the results of the one-dimensional analysis may be used to develop a criterion for switching to this method when the two-level algorithm is used in a nested iteration.

However, when standard upwinding is used on the fine patch, it is
discovered that the convergence rates deteriorate at the tangential boundaries of the patch. The reason for this degradation is demonstrated, and a successful remedy that involves modification of the coarse grid stencil is implemented. Similar results are obtained for higher-order upwinding. These results indicate that the use of higher-order upwinding can replace standard upwinding on the fine patch with little sacrifice in efficiency.

Although we obtain satisfactory convergence rates for the two-level FAC scheme, we feel that significant improvement can be made in terms of optimizing the efficiency of the method. For example, the analysis and experiments that we perform with respect to this scheme employ exact solution of the subgrid problems in its definition, and these solves must be performed during each of the scheme's global iterations. An obvious and attractive alternative is to replace these exact solves with inexpensive solution approximations obtained by using a small number of steps of some iterative method. However, in practice it has been noted (in the solution of diffusion-type problems, for instance) when exact solutions of problems on the fine patch are replaced by partial ones that significant degradation in convergence rates of the global process occur (see [23], for example). We have noticed similar effects when applying this strategy to the advection-dominated problem here.

Our solution to this problem, which allows us to employ partial solution on the subgrids, is to use FAC in a nested way, in the same way that the V-cycle is used in the optimally efficient full multigrid method. Thus, two-level FAC plays the role of a weak algebraic solver on each of the composite grids in the succession of grids lying between the global coarse grid and the true composite grid. This yields an algorithm that usually requires only a few relaxations on the coarse mesh underlying the composite grid, and
which has optimal efficiency on the fine patch.

We close this section by noting that, although we have introduced our numerical methods in the context of the elliptic problem (2.1), they are also applicable to the solution of the mixed hyperbolic-parabolic type equation

\[ u_t - \epsilon \Delta u + a u_x + b u_y + d u = g, \]  

(2.3)

when \( \epsilon \) is small, i.e., when this problem is mainly hyperbolic in nature. Although the results contained in what follows pertain to the elliptic problem, they generalize well to the solution of (2.3) under the above-mentioned transformation of spatial coordinates when implicit methods such as backward Euler or Crank-Nicolson are used to perform the discretization in time. The resulting linear systems at each time step then closely resemble the ones described herein except for having improved stability properties (that is, matrix diagonal dominance) due to the introduction of a positive capacitance term.
3. Discretization

This chapter is devoted to studying discretization methods for elliptic problems on composite grids. We begin in Section 3.1 with a description of the composite grids to be used in this study.

In Section 3.2 we describe finite element, finite volume element, and finite volume composite grid discretizations. Although the results of this study pertain for the most part to the finite volume discretization, it is useful to consider the subspace approach of the element methods. This approach permits the systematic handling of the discretization on a composite grid in more than one dimension, and also lends guidance in performing the discretization in the absence of a subspace. In one dimension, it turns out that if the finite element subspace is chosen appropriately the finite volume method yields the same discrete problem as the finite element and finite volume element methods. We introduce a composite basis for this subspace in Section 3.2.1. The generalization of this basis to two dimensions is made in Section 3.2.2, where we show how its use facilitates performing the element discretizations. In Section 3.2.3 the finite volume discretization in two dimensions is introduced. Based on the control volume approach described in [36], the method we describe in the context of the composite grid is similar to the one in [22], where a finite volume method is applied to a diffusion problem on a cell-centered composite grid (our grid differs from this somewhat in that the interface nodes are not cell-centered). Our approach is also similar to [22] in its treatment of the portion of the difference stencil at the interface between the subgrids, though there is a significant difference. In [22] these connections are calculated explicitly. This involves
either interpolating nodal values as piecewise constants throughout cells centered at interface nodes, or interpolating, linearly, neighboring nodes. Our approach is similar to the latter technique, however, it avoids calculating explicitly the stencil’s coefficients. Rather, it uses standard fine grid coefficients (with the fine grid extended to the interface by using slave nodes), and performs the required interpolation between interface nodes prior to application of the matrix stencil as the solution algorithm proceeds. This latter approach is borrowed from the element methods. We note that it simplifies the derivation of the matrix stencil and allows the use of a standard uniform-grid stencil on the fine patch. Section 3.2.4 compares briefly the relative merits of the three discretization methods treated here.

Finally, Section 3.3 treats the accuracy of these discretization methods.

3.1 Description of Composite Grids

Our description of the composite grids to be used in this study applies to the case of a rectangular domain in two dimensions. We suppose the subregion of the domain requiring refinement to be an interior region that does not intersect the boundary. Although no assumptions are made concerning the shape of this region, we will only consider refinement of the grid on a rectangular patch. The structure we use for the composite grid is that of [33]. Generalizations of this concept of a composite grid to L-shaped grids and refinement regions, other dimensions, refinements near the boundaries of the domains, and multiple regions of refinement can easily be made. But for the purposes of this study we restrict our attention, for the most part, to the above simple case. Although we only describe in detail the two-dimensional case, the structure of the composite grid in one
dimension is very similar, and it should be straightforward for the reader to adapt the definitions in this section to the situation in one dimension.

Suppose we are given a rectangular $h$-grid with uniform spacing, $h$ being the node-to-node distance in both the $x$- and $y$-directions. A composite grid is determined by choosing some rectangular subset of nodes along with a positive integer, $m$, indicating the order of refinement. The set of nodes forming the boundary of the subset is called the interface. We assume that each side of the interface has at least three nodes (including corners) so that the interior of the rectangular subset is nonempty. This set of interior nodes is called the coarse patch.

If we designate the set of nodes exterior to the interface as the coarse nodes, then we have the uniform $h$-grid as the union of three components: coarse, interface, and coarse patch. We call this grid the global uniform coarse grid or just global coarse grid. See Figure 3.1.

The composite grid is obtained from the global coarse grid by adding nodes to the coarse patch; the fine patch component of the composite grid is the set of interior nodes of the uniform $h_F$-grid on the region bounded by the interface, where $h_F = h \cdot 2^{-m}$. The set of boundary nodes for the fine patch (which contains the interface) is called the fine boundary, but aside from the interface nodes it is not considered part of the composite grid. However, we define the complement of the interface in the fine boundary as an adjuvant to the composite grid called the set of slave nodes. (Though not technically belonging to the composite grid, they play an essential role in the definition of the composite discretization.)

In summary, the composite grid is made up of three components, coarse and interface components being the same as for the global coarse grid, along with a fine component obtained via “refinement on the patch”. An example
Fig. 3.1. *Global coarse grid with coarse (o), fine (●), and interface (○) nodes.*
of a composite grid (with one refinement) is shown in Figure 3.2.

\[\begin{array}{cccccccccccc}
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
\circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ \\
\circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
\circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ \\
\circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
\circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ \\
\circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
\circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ \\
\circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
\circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ \\
\circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
\circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ \\
\circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
\circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ \\
\circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
\circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ \\
\circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
\circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ \\
\circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
\circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ \\
\circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
\circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ \\
\circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
\circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ \\
\circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ \
\end{array}\]

Fig. 3.2. Composite grid with coarse (○), fine (●), interface (□), and slave (○) nodes.

3.2 Discretization Methods

Next we describe three discretization methods for the following elliptic problems:

\[-\varepsilon u''(x) + u'(x) = f(x), \quad x \in \Omega = [0,1] \]
\[u(0) = 1, \quad u'(1) = \alpha, \tag{3.1}\]

and

\[-\varepsilon \Delta u(x, y) + u(x, y) = f(x, y), \quad (x, y) \in \Omega = [0,1] \times [0,1] \]
\[u(x, 0) = g_1(x), \]
\[u(0, y) = g_2(y), \]
\[u(x, 1) = g_3(x), \quad u_2(1, y) = \alpha(y), \tag{3.2}\]

Here \(\varepsilon\) is a constant, \(0 < \varepsilon < 1\). The methods we consider are: the finite element, finite volume element and finite volume methods.

We begin by giving a brief formal description of the two element methods. Let a finite-dimensional function space \(\Phi\) out of which one would
like to construct a solution to the partial differential equation $\mathcal{L}u = f$ be
given. We take as our approximation to $u$ a particular linear combination
of elements of a basis for $\Phi$. In particular, we assume the existence of a
nodal basis: $\{\phi_1, \phi_2, \ldots, \phi_N\}$ is such a basis if each $\phi_i$ is associated with
precisely one node $n_i$ (we will frequently use this notation for nodes in one
or two dimensions), and, if any $\phi \in \Phi$ is expanded in this basis as

$$\phi = a_1 \phi_1 + a_2 \phi_2 + \ldots a_N \phi_N,$$

then $a_i = \phi(n_i)$. The finite element (FE) method [1], [16], [26], [41] ap-
proximates $u$ of $\mathcal{L}u = f$ by the solution, $\phi \in \Phi$, of the discrete variational
equations

$$\int_{\Omega} (\mathcal{L}\phi) \phi_i \, dV = \int_{\Omega} f \phi_i \, dV \quad \forall i.$$

This is equivalent to finding $a$ such that $La = f$, where

$$l_{ij} = \langle \phi_j, \phi_i \rangle \equiv \int_{\Omega} (\mathcal{L}\phi_j) \phi_i \, dV$$

and $f_i = \int_{\Omega} f \phi_i \, dV$.

To describe the finite volume element (FVE) [33] method, we require,
in addition to the finite dimensional space $\Phi$, a finite set of volumes $\{V_i\}_{i=1}^N$
that partition the domain $\Omega$. Typically, as with the basis elements, $V_i$ will
be associated with precisely one node $n_i$ (a more detailed description of the
volumes for the composite grids used here will be given in the discussion of
the finite volume discretization). An example of volumes associated with
a subset of nodes of a two dimensional composite grid is shown in Figure
3.3. FVE requires that $\phi$ satisfy the local conservation equations

$$\int_{V_i} \mathcal{L}\phi \, dV = \int_{V_i} f \, dV \quad \forall i,$$

which is equivalent to finding $a$ such that $La = f$, where $l_{ij} = \int_{V_i} \mathcal{L}\phi_j \, dV$
and $f_i = \int_{V_i} f \, dV$. 

21
Fig. 3.3. *Volumes for a portion of a two-dimensional composite grid.*

We will return to the subspace framework shortly, but now we introduce the third discretization method, beginning with a detailed description for the one-dimensional model problem. Similar to FVE, the *finite volume* (FV) method [36] begins by partitioning the domain into a finite set of *volumes* in such a way that, associated with each node of the composite grid, there is precisely one volume that contains that node. Typically, the node lies in the interior of its associated volume (although it is possible to have a node at the edge of its volume, as is the case in the usual treatment of Neumann boundary conditions). Figure 3.4 shows a portion of a one-dimensional composite grid containing a coarse node, an interface node, and two fine nodes (nodes are labeled $\xi_n$ here) along with the endpoints of their associated "volumes" (the $\xi_n$'s). The lengths of the volumes for
coarse, fine, and interface nodes are, respectively, $h$, $h/2^m$, and $\frac{1}{2}(h+h/2^m)$, where $m$ is the number of refinement levels used on the fine patch.

Fig. 3.4. One-dimensional composite grid with nodes $(x_n)$, and volume endpoints $(\xi_n)$.

3.2.1 Finite Volume Discretization in One Dimension

The finite volume discretization in one dimension proceeds by imposing the following local conservation law, which is obtained by integrating (3.1) over each of the volumes $[\xi_i, \xi_{i+1}]$:

$$\int_{\xi_i}^{\xi_{i+1}} (-cu(x)'' + u(x)') dx = \int_{\xi_i}^{\xi_{i+1}} f(x) dx. \quad (3.3)$$

These equations are then transformed into a system of equations for the nodal values of $u$. Second-order and first-order terms are handled separately. For the second-order term at the interface we have,

$$\int_{\xi_i}^{\xi_{i+1}} u''(x) dx = u'(\xi_{i+1}) - u'(\xi_i) \approx \frac{u(x_{i+1}) - u(x_i)}{2(h/2^m+1)} - \frac{u(x_i) - u(x_{i-1})}{2(h/2)}$$

$$= \left( u(x_{i-1}) - (1+2^m)u(x_i) + 2^m u(x_{i+1}) \right)/h.$$

Here, we have used a second-order divided difference to approximate the first derivatives. For the first-order term

$$\int_{\xi_i}^{\xi_{i+1}} u'(x) dx = u(\xi_{i+1}) - u(\xi_i), \quad (3.4)$$

23
we need to approximate the difference on the right-hand side by an expression in $u(x)$ evaluated at nodes. The centered formula averages $u$ at nodes adjacent to $\xi_i$ and $\xi_{i+1}$:

$$\int_{\xi_i}^{\xi_{i+1}} u'(x)dx \approx \frac{u(x_{i+1}) + u(x_i)}{2} - \frac{u(x_{i-1}) + u(x_i)}{2} = \frac{u(x_{i+1}) - u(x_{i-1})}{2}.$$  

The upwind formula replaces the volume endpoints with their nearest left-neighbor nodes:

$$\int_{\xi_i}^{\xi_{i+1}} u'(x)dx \approx u(x_i) - u(x_{i-1}).$$

Finally, one might consider a higher-order upwind formula for the first-order term on the fine patch. We do this by replacing the difference on the right-hand side of (3.4) by its second-order approximation at $x_{i+2}$ and the upwind nodes $x_{i+1}$ and $x_i$:

$$u(\xi_{i+3}) - u(\xi_{i+2}) \approx \frac{1}{2}u(x_i) - 2u(x_{i+1}) + \frac{3}{2}u(x_{i+2}).$$

Notice that this formula cannot be applied at the leftmost fine node. At that node, the upwind stencil can be used, for example.

Collecting these formulas and using similarly derived approximations at coarse, fine, and right interface nodes yields the following tridiagonal matrix stencils. Centered at the generic node $u$, these stencils involve neighboring nodes in an obvious way expressed by

$$[n_W, u, n_E].$$

Upwind

Coarse Nodes:

$$[-\epsilon - h, 2\epsilon + h, -\epsilon]/h$$
Left Interface:

\[-\varepsilon - h, (1 + 2^m)\varepsilon + h, -2^m \varepsilon]/h\]

Right Interface:

\[-2^m \varepsilon - h, (1 + 2^m)\varepsilon + h, -\varepsilon]/h\]

Fine Nodes:

\[-2^m \varepsilon - h, 2^{m+1}\varepsilon + h, -2^m \varepsilon]/h\]

Centered

Coarse Nodes:

\[-\varepsilon - h/2, 2\varepsilon, -\varepsilon + h/2]/h\]

Left Interface:

\[-\varepsilon - h/2, (1 + 2^m)\varepsilon, -2^m \varepsilon + h/2]/h\]

Right Interface:

\[-2^m \varepsilon - h/2, (1 + 2^m)\varepsilon, -\varepsilon + h/2]/h\]

Fine Nodes:

\[-2^m \varepsilon - h/2, 2^{m+1}\varepsilon, -2^m \varepsilon + h/2]/h\]

Higher-Order Upwind

Fine Nodes:

\[-h/2, -2^m \varepsilon - 2h, 2^{m+1}\varepsilon + 3h/2, -2^m \varepsilon]/h\]
This latter stencil involves the nodes

\[ [n_{WW}, n_W, n, n_E]. \]

In addition, the Neumann condition at the right boundary, \( u'(1) = \alpha \), is handled as follows. Here, the volume for the node at the right boundary is a half-volume. See Figure 3.5.

\[ \xi_{i-1} \quad x_{i-1} \quad x_i = \xi_{i+1} = 1 \quad \phi \]

\[ \h \]

**Fig. 3.5. One-dimensional grid with a right Neumann boundary.**

We have

\[ \int_{\xi_i}^{x_i} u''(x)dx = u'(x_i) - u'(\xi_i) \approx \alpha - (u(x_i) - u(x_{i-1}))/h, \]

\[ \int_{\xi_i}^{x_i} u'(x)dx = u(x_i) - u(\xi_i) \approx u(x_i) - u(x_{i-1}) \quad \text{(upwind)}, \]

and

\[ \int_{\xi_i}^{x_i} u'(x)dx = u(x_i) - u(\xi_i) \approx u(x_i) - \frac{1}{2}(u(x_i) + u(x_{i-1})) \quad \text{(centered)}. \]

These yield the following stencils and right-hand sides.

**Neumann Boundary**

**Upwind:**

\[ [-\epsilon - h, \epsilon + h]/h = \frac{h}{2} \cdot f(1 - \frac{h}{4}) + \epsilon \alpha \]

**Centered:**

\[ [-\epsilon - h/2, \epsilon + h/2]/h = \frac{h}{2} \cdot f(1 - \frac{h}{4}) + \epsilon \alpha \]
These stencils, centered at node $n$, correspond to

$$[n_W, n].$$

Generally, for the right-hand side of (3.3), letting $V_i = [\xi_i, \xi_{i+1}]$, we use

$$\int_{V_i} f(x) dx \approx (\xi_{i+1} - \xi_i) \cdot f(x_i).$$

Finally, we note that in treating Dirichlet boundary conditions, the value of $u$ at a Dirichlet node is given by the specified boundary value. Thus, the equation for such a node is a trivial one that is easily eliminated from the total system of equations by modification of its right-hand side. The use of this approach obviates the need for volumes associated with Dirichlet nodes and, as a result, the volumes do not truly partition the domain. A discussion of the effect of this treatment in terms of loss of conservation in volume methods, along with an alternative for maintaining the partitioning, is given in [33, Chapter 2].

### 3.2.2 Discretization by Element Methods

Reconsidering the FE and FVE discretizations discussed earlier, it is interesting to note that, in one dimension, the centered version of the stencil just derived using the FV method can be obtained by applying either the FE or FVE method (using the volumes just described) to the nodal basis pictured in Figure 3.6c, where $n_\tau$ is the left interface node. (Only the nonzero values of the basis functions are graphed.) The basis functions corresponding to coarse nodes are familiar piecewise linear hat-functions for the uniform $h$-grid (i.e., the global coarse grid; see Figure 3.6a) and the basis functions corresponding to the fine nodes are similar functions corresponding to a uniform $h/2^m$-grid (which we call the global fine grid; see Figure 3.6b). We call these bases (Figures 3.6a-c, respectively) the
global coarse, global fine and composite bases, and denote their respective spans by $\Phi$, $\Psi$, and $\hat{\Phi}$. The relationship between the three bases may be succinctly represented by the equations
\[
\hat{\phi}_p = \frac{1}{2}\psi_p + \psi_p + \frac{1}{2}\psi_q = \phi_p \\
\hat{\phi}_r = \frac{1}{2}\psi_q + \psi_r = \phi_r - \frac{1}{2}\psi_s \\
\hat{\phi}_s = \psi_s
\]
Now, let $\Psi_F$ be the span of $\{\psi_s, \psi_t, \psi_u\}$ and let $\Psi_C$ be the span of basis functions for $\Psi$ corresponding to nodes other than $n_s$, $n_t$, and $n_u$. Then any composite basis element $\hat{\phi}_i$ may be written uniquely as $\hat{\phi}_i = \phi_{i,C} + \phi_{i,F}$, where $\phi_{i,C} \in \Psi_C$ and $\phi_{i,F} \in \Psi_F$. Thus,
\[
\begin{align*}
\hat{\phi}_i &= \phi_{i,C} &\text{if } n_i \text{ is a coarse or interface node} \\
\hat{\phi}_i &= \phi_{i,F} = \psi_i &\text{if } n_i \text{ is a fine node},
\end{align*}
\]
i.e., for coarse and interface nodes of the composite grid, the composite nodal basis element $\hat{\phi}$ is obtained from the corresponding global coarse basis element $\phi$ by subtracting off its component in $\Psi_F$ (for coarse nodes this component is zero), and, for fine nodes, $\hat{\phi}$ is the global fine basis element associated with that node.

We note that this construction of a composite basis produces a space that is nested between the global coarse and global fine spaces:
\[
\Phi \subset \hat{\Phi} \subset \Psi. \quad (3.5)
\]
We call (3.5) the conformity relationship between these subspaces. It is useful for obtaining accuracy estimates for the finite element discretization (see Section 3.3).

This leads us to consider FE and FVE discretizations in two dimensions. We need a finite-dimensional space out of which an approximation to the solution of our differential equation may be constructed. Suppose we define,
Fig. 3.6. Nodal bases for global coarse (a), global fine (b), and composite (c) spaces.
in the usual way, nodal global coarse and global fine bases consisting of piecewise linear hat functions on triangulations as shown in Figure 3.7.
Because of the similarity of the component structure (the division into coarse, interface, and fine components) of the composite grid in one and two dimensions the specification given above applies directly in two dimensions to the construction of a nodal composite basis for such a space. However, the process of discretization is now complicated by the nonuniformity of nodes in two directions at the interface. Such nonuniformity is not a problem at coarse or fine nodes where, as in one dimension, construction of the matrix stencil proceeds along familiar lines (discretization on uniform grids with given meshwidths). Yet, the difficulties at the interface can be made to virtually disappear by viewing the task locally from the point of view of the global fine basis.

To be specific, consider stencils involving interface-to-fine connections. These connections can be obtained easily by taking advantage of the structure of the composite basis. For example, in the FE case the computation of \( \langle \phi_p, \phi_q \rangle \), where \( \phi_q \) and \( \phi_p \) are composite basis functions corresponding to interface node \( n_q \) and neighboring fine node \( n_p \), respectively, is facilitated by viewing \( \phi_q \) on the same (global fine) level as \( \phi_p \) (since \( \phi_p \) corresponds to a fine node, it is equal to a global fine basis function), i.e., as a linear combination of global fine grid basis functions (namely, \( \phi_q = \phi_{a,c} \)). This reduces the problem to using existing finite element software to compute \( \langle \phi_p, \phi_s \rangle \), where \( \phi_p \) and \( \phi_s \) are both elements of the global fine basis, and to combining such contributions. In a sense, this process uses slave nodes—basis functions corresponding to these nodes are introduced temporarily (see Figures 3.8-3.9). A similar technique can be applied in a FVE discretization, though in practice the procedure is not quite as simple due
Fig. 3.7. *Global coarse (a) and global fine (b) triangulations.*
to the additional nonuniformity in volumes at the interface. On the other hand, FVE stencils are more sparse than FE stencils, additional localization being induced by the volumes.

Next we examine the fine-to-interface connections. We emphasize the construction of this portion of the stencil in the subspace context since it provides guidance for the construction in the absence of a subspace, for example, in the FV discretization.

In order to describe the treatment of this part of the discretization, it is useful to introduce the subspace \( \Psi_b \), which is defined to be the span of those global fine basis elements that correspond to nodes at the boundary of the fine patch (interface nodes and slave nodes – this subspace comprises a fine boundary for the patch), and \( \hat{\Phi}_I \), which is defined to be the span of the composite basis functions corresponding to interface nodes. For convenience, we summarize the decomposition of the global fine subspace, \( \Psi \), and the composite subspace, \( \hat{\Phi} \), below:

\[
\begin{align*}
\Psi_F & \quad \text{Span of } \psi \text{'s corresponding to fine patch nodes,} \\
\Psi_C & \quad \text{Span of } \psi \text{'s corresponding to other than fine patch nodes,} \\
\Psi_b & \quad \text{Span of } \psi \text{'s corresponding to interface and slave nodes } (\Psi_b \subset \Psi_C), \\
\hat{\Phi}_I & \quad \text{Span of } \hat{\phi} \text{'s corresponding to interface nodes.}
\end{align*}
\]

Here, \( \psi \) and \( \hat{\phi} \) denote generic elements of the global fine and composite nodal bases, respectively. We also note the following localness property of the FE and FVE composite grid discretizations: each composite basis function corresponding to an interface node enters into the computation of a fine-to-interface connection only through its components in \( \Psi_b \). This component of \( \hat{\phi}_I \) (where \( n_I \) is an interface node) is obtained by restricting
the expansion
\[ \hat{\phi}_t = \phi_{t,C} \]
to the boundary of the fine patch:
\[ \hat{\phi}_{t|b} = \phi_{t,C|b}. \] (3.6)

Here, the restriction denotes the presence on the right-hand side of (3.6) of basis elements for \( \Psi_b \) only. For each \( \hat{\phi}_t \), this relation provides a unique representation of the component along the interface in terms of the basis for \( \Psi_b \), and, so, provides a well-defined linear mapping, \( I_f^b : \hat{\Phi}_I \rightarrow \Psi_b \), of the interface to the fine boundary. We denote this linear operator as interpolation at the interface. Now, due to the localness property, it does not matter whether we compute the fine-to-interface stencil with respect to the \( \hat{\phi}_t \)'s at the interface or with respect to \( \Psi_b \). It follows that this stencil may be computed using just the basis elements from \( \Psi_F \) and \( \Psi_b \) (i.e., the nonuniformity between fine patch and interface may be avoided by replacing the interface with a fine-mesh boundary). This means that when the fine-to-interface stencil is applied (during matrix multiplication) to an interface component, interpolation must first be performed:

\[ u_b \leftarrow I_f^b u_I \]

\( u_f \) is a given function in \( \hat{\Phi}_I \), and \( u_b \) is its component in the fine boundary), and then the matrix stencil is applied to \( u_b \).

As an example of this process, consider the case of a composite basis function \( \hat{\phi}_t \) at the left vertical edge of the interface when there is one level of refinement. Then the representation of \( \hat{\phi}_t \) in terms of the basis for \( \Psi_b \) is

\[ \hat{\phi}_{t|b} = \frac{1}{2} \psi_- + \psi_t + \frac{1}{2} \psi_+. \]
Fig. 3.8. Support of a composite basis function corresponding to an interface node (○) at a left vertical interface.

Fig. 3.9. Support of a composite basis function (shown twice) at a left vertical interface and support of the functions in its component along the fine boundary.
\(n_-\) and \(n_+\) are the slave nodes below and above node \(n_i\). The support of \(\hat{\phi}_i\) is shown in Figure 3.8. It is shown again in Figure 3.9 along with the supports of the three basis functions in its component along the fine boundary. Representations along other edges are similar, meaning that interpolation at the interface proceeds as a sequence of one-dimensional interpolations along the edges of the interface: the nodal value of a function \(u_b\) equals the corresponding value of \(u_f\) if the node is an interface node, and is equal to the average of its neighbors' nodal values if the node is a slave. With multiple levels the process may be carried out using a reapplication of the single-level process on the succession of levels leading from the interface to the fine boundary.

We note that due to the localness property, this procedure is equally valid for FVE and FE discretizations. Having arrived at a satisfactory way of handling this part of the discretization when it is performed with respect to a composite function space, it may also be used to facilitate this part of the discretization in methods that do not utilize such a space, as is the case, for instance, with the finite volume method. Normally, with the latter method, treatment of the interface necessarily involves the use of interpolation at interface nodes when calculating explicitly the fine-to-interface stencil [22]. On the other hand, the approach described above defines this stencil implicitly, and allows for the use of a standard uniform-grid discretization on the union of the fine patch and the fine boundary.

At this point, it is useful to say a few words about the way that the coefficient matrix \(L\) of the composite grid problem is represented in practice. Three matrix stencils are called for: a coarse stencil that involves coarse and interface nodes, a fine stencil that involves fine patch nodes augmented by a fine boundary, and an interface stencil that involves fine
patch, interface, and coarse nodes. In our computer codes, we use separate lexicographic orderings of the fine patch, and of the coarse component imbedded in a global coarse grid. This means that the coarse and fine stencils are standard uniform grid stencils and may be utilized as such. In forming the fine patch component of the product $Ly$, for example, storage for the fine patch component of $y$ is made large enough to accommodate the fine boundary. Prior to applying the stencil to this component, the interpolant of the interface of $y$ is stored in these locations. In forming the coarse component, computations proceed in the usual manner, except that they are restricted to the coarse and interface components of $y$. Finally, the interface portion of the product requires some special handling. In our implementation, the interface resides together with the coarse component as part of an array representing a lexicographically ordered global coarse vector. Therefore, interface-to-coarse and interface-to-interface computations proceed as they would on a uniform grid, while interface-to-fine connections are handled by a separate code segment that requires access to the patch component of $y$.

### 3.2.3 Finite Volume Discretization in Two Dimensions

For the most part, finite volume discretization of the problem (3.2) is not much more involved than that of (3.1), since the method in two dimensions is very similar to that in one dimension. To see this, consider the portion of a two-dimensional composite grid near a left vertical (west) interface, as shown in Figure 3.10. The figure depicts (left to right) a coarse, an interface, and three fine nodes along with their associated volumes.

Let $V$ be the interface volume and label its sides in an obvious way as $N, S, E,$ and $W$. By Green’s Theorem,
Fig. 3.10. *Nodes and volumes near a left vertical interface.*

\[
\int_{\hat{V}} \Delta u \, dV = \int_{E} u_{x} \, dy - \int_{W} u_{x} \, dy
\]

\[
+ \int_{N} u_{y} \, dx - \int_{S} u_{y} \, dx.
\]

Consider, for example, the first term on the right. We approximate \( u_{x} \) at the midpoint of the east side of \( \hat{V} \) using a centered difference formula involving \( u(\hat{x}, \hat{y}) \) and \( u(\hat{x} + h/2^m, \hat{y}) \) and integrate using the midpoint rule:

\[
\int_{E} u_{x} \, dy \\
\approx h \left( \frac{u((\hat{x} + h/2^{m+1}) + h/2^{m+1}, \hat{y}) - u((\hat{x} + h/2^{m+1}) - h/2^{m+1}, \hat{y})}{2(h/2^{m+1})} \right)
\]

\[
= 2^{m} \cdot (u(\hat{x} + h/2^{m}, \hat{y}) - u(\hat{x}, \hat{y})).
\]

Using a similar approach for, say, the third term gives

\[
\int_{N} u_{y} \, dx \approx h \left( \frac{1}{2} + \frac{1}{2^{m+1}} \right) \frac{u(\hat{x}, \hat{y} + h) - u(\hat{x}, \hat{y})}{2(h/2)}
\]

37
\[
= \left( \frac{1}{2} + \frac{1}{2^{m+1}} \right) \left( u(\hat{x}, \hat{y} + h) - u(\hat{x}, \hat{y}) \right).
\]

For the first-order term, using an upwind approach, we have

\[
\int_{\Gamma} u_x dV = \int_{E} u dy - \int_{W} u dy \\
\approx h(u(\hat{x}, \hat{y}) - u(\hat{x} - h, \hat{y})).
\]

Let \( s = \frac{1}{2} + \frac{1}{2^{m+1}} \). Using these approaches leads to the following stencils for the upwind version of the FV discretization, which are centered at node \( n \) and involve the nodes

\[ [n_S, n_W, n, n_E, n_N]. \]

**Upwind**

**Coarse Nodes:**

\[ [-\epsilon, -\epsilon - h, 4\epsilon + h, -\epsilon, -\epsilon] \]

**West Interface:**

\[ [-s\epsilon, -\epsilon - h, (1 + 2s + 2^m)\epsilon + h, -2^m\epsilon, -s\epsilon] \]

**East Interface:**

\[ [-s\epsilon, -2^m\epsilon - h, (1 + 2s + 2^m)\epsilon + h, -\epsilon, -s\epsilon] \]

**South Interface:**

\[ [-2^m\epsilon, -s(\epsilon + h), (1 + 2s + 2^m)\epsilon + sh, -s\epsilon, -\epsilon] \]

**North Interface:**

\[ [-\epsilon, -s(\epsilon + h), (1 + 2s + 2^m)\epsilon + sh, -s\epsilon, -2^m\epsilon] \]
Fine Nodes:

\[-\varepsilon, -\varepsilon - h/2^m, 4\varepsilon + h/2^m, -\varepsilon, -\varepsilon\]

An obvious modification of the handling of the first-order term (see the details of the one-dimensional FV discretization) yields the following centered stencils.

Centered

Coarse Nodes:

\[-\varepsilon, -\varepsilon - h/2, 4\varepsilon, -\varepsilon + h/2, -\varepsilon\]

West Interface:

\[-s\varepsilon, -\varepsilon - h/2, (1 + 2s + 2^m)\varepsilon, -2^m\varepsilon + h/2, -s\varepsilon\]

East Interface:

\[-s\varepsilon, -2^m\varepsilon - h/2, (1 + 2s + 2^m)\varepsilon, -\varepsilon + h/2, -s\varepsilon\]

South Interface:

\[-2^m\varepsilon, -s(\varepsilon + h/2), (1 + 2s + 2^m)\varepsilon, -s(\varepsilon - h/2), -\varepsilon\]

North Interface:

\[-\varepsilon, -s(\varepsilon + h/2), (1 + 2s + 2^m)\varepsilon, -s(\varepsilon - h/2), -2^m\varepsilon\]

Fine Nodes:

\[-\varepsilon, -\varepsilon - h/2^{m+1}, 4\varepsilon, -\varepsilon + h/2^{m+1}, -\varepsilon\]
We also provide the stencils for the Neumann condition at the right boundary (with their associated right-hand sides) and a fine patch stencil for a higher-order upwinding.

**Neumann Boundary**

**Upwind:**

\[
[-\epsilon/2, -\epsilon - h, 2\epsilon + h, \bullet, -\epsilon/2] = \frac{h^2}{2} f(1 - \frac{h}{4}, y) + \epsilon h \cdot \alpha(n)
\]

**Centered:**

\[
[-\epsilon/2, -\epsilon - h/2, 2\epsilon + h/2, \bullet, -\epsilon/2] = \frac{h^2}{2} f(1 - \frac{h}{4}, y) + \epsilon h \cdot \alpha(n)
\]

Here, "\(\bullet\)" indicates the node \(n_\mathcal{F}\) lying outside of the stencil.

**Higher-Order Upwind**

**Fine Nodes:**

\[
[-\epsilon, \frac{1}{2}h/2^m, -\epsilon - 2h/2^m, 4\epsilon + \frac{3}{2}h/2^m, -\epsilon, -\epsilon]
\]

This final stencil is associated with the nodes

\[\{n_S, n_{WW}, n_W, n, n_E, n_N\}\]

As for the treatment of Dirichlet nodes, discretization at these points is performed as described for the one-dimensional case at the end of Section 3.2.1.

We note that the above formulas for the interface nodes do not apply at its corners. For these nodes we use the standard stencil for the coarse
nodes. Also, the fine grid stencils apply throughout the fine patch, assuming that the process of interpolation of interface nodes to a fine boundary as described earlier in this chapter is incorporated into the definition of the discretization at fine-to-interface connections.

3.2.4 A Comparison of Discretization Methods

In this section, we briefly consider the relative merits of the discretization methods described in this chapter. For the most part, the results of this study pertain to finite volume discretization. We have chosen to emphasize this method primarily due to its ease of use, first, in deriving the composite grid equations, and second, in terms of solving these equations.

One attractive feature of this discretization is the simplicity of its matrix stencil. With the approach used here, this method gives rise to a five-point stencil throughout the composite grid. This is due to the way the interface-to-patch connections have been defined. The result is that computer code involving the stencil is particularly easy to program. The element stencils differ from this in that they distribute the coefficient of this connection locally among nodes of the fine patch. The result is a stencil that becomes less sparse at the interface as the patch is refined.

In the context of the problem,

$$-\epsilon \Delta u(x, y) + u_x(x, y) = f(x, y), \quad (3.7)$$

with \( \epsilon \) small, a more compelling reason for appealing to the finite volume method is that its stencils appear on uniform grids as familiar finite difference ones. The efficiency of the numerical methods of this study depend strongly on the ability to solve discrete versions of (3.7) on the fine patch and on the global coarse grid, i.e., on uniform grids. Relaxation serves as the basis for solution of these problems by multigrid and, due to the fact

41
that the finite difference matrices capture all of the advection of (3.7) in their main tridiagonal blocks, line relaxation is extremely effective (see Section 5.3.1). We note that it is more difficult to identify robust relaxation strategies (i.e., ones that work well over wide parameter ranges) for the element methods. This is due to the appearance of advection (corresponding to the \( x \)-derivative in (3.7)) in portions of the stencil that correspond essentially to discretization of \( y \)-derivatives. For example, the uniform grid stencils for finite volume elements, which is analogous to centered differencing, is

\[ [-\epsilon - h/8, h/8, -\epsilon - 3h/8, 4\epsilon, -\epsilon + 3h/8, -h/8, -\epsilon + h/8], \]

which corresponds to the nodes

\[ [n_S, n_{SE}, n_W, n, n_E, n_{NW}, n_N]. \]

This drawback does not preclude the use of the linear equations solvers of this study when applied to the equations derived by the element methods. However, more work needs to be done to develop the appropriate relaxation strategies. Finally, we note that although the finite volume stencils appear largely like finite difference ones, there is a methodology present in the former approach that is lacking in the latter. This methodology provides guidance for the systematic treatment of the discretization, especially as the grid becomes nonuniform, i.e., as the coarse component meets the fine one.

The systematic nature of the discretization is also an important and attractive feature of the element methods, FE and FVE. In addition to facilitating the treatment of the interface portion of the discretization, it also simplifies the choice of interpolation and restriction operators on
which multigrid depends. With conformal subspaces, i.e., \( \hat{\Phi}_{2h} \subset \hat{\Phi}_h \), the interpolation operator, \( I_{2h}^h : \hat{\Phi}_{2h} \rightarrow \hat{\Phi}_h \), is determined by representing each basis element for \( \hat{\Phi}_{2h} \) in the basis for \( \hat{\Phi}_h \). In finite elements, restriction, \( I_{2h}^h : \hat{\Phi}_h \rightarrow \hat{\Phi}_{2h} \), is also determined automatically by requiring that the so-called variational equations that relate the coarse grid discretization to the fine one [30] hold. With finite volume elements the choice of restriction may be guided by physical principles (see [33, Chapter 3]). With finite volume elements the choice is more arbitrary. In this study, when grid transfers are called for we employ the seven-point prolongation and restriction operators of [25, Chapter 3] for the fine patch component, and the trivial (identity) mapping for the coarse and interface components.

Finally, we make note of the fact that upwind discretizations of (3.7) play a very significant role in this study. We have found it easy to incorporate first-order and second-order upwinding into the finite volume discretization. In finite elements the use of upwinding is also commonplace. On the other hand, a satisfactory strategy for performing upwind discretization has not yet been developed with respect to the finite volume element method.

### 3.3 Accuracy of the Discretizations

We now turn to a discussion of the accuracy of the discretizations described in this chapter. Of particular interest are the finite volume discretizations that we have derived in some detail for the model problem (3.2). The reasons for this interest are at least two-fold. First, the finite volume discretization as described for the two-dimensional case is somewhat novel and the matter of its accuracy is of interest in itself. And second, the analysis of algorithms for solving composite grid equations
that will be pursued later in this study (in Chapter 4) is based on their behavior with respect to the finite volume discretization, so it is important to establish the validity of this type of discretization. In the next section, we will assess the accuracy of this discretization numerically for a sampling of test problems. This section is a more general discussion of the accuracy of the discretizations considered above. It begins by taking note of recent convergence results for FV and FVE discretizations of elliptic problems on nonuniform grids.

Prior to the recent studies of Cai et al. [10], [11], little had been done to prove the convergence of finite volume element discretizations. Some prior work had been done in this context for finite volume discretizations on uniform meshes (references to this work are contained in [11] and [42]), and in certain instances these turn out to be special cases of FVE discretizations. The work of Cai et al., however, was the first to incorporate finite element theory into the FVE framework in order to produce finite element-like accuracy estimates. This work also includes a systematic study of convergence on composite and other nonuniform grids. In [10] an $O(h^{3/2})$ bound in a discrete energy norm is proved for the Poisson equation using a composite grid FVE discretization similar to the one described herein, with piecewise linear functions used for the approximating subspace ($h$ here denotes the coarse component meshwidth). This result is strengthened to $O(h^2)$ by incorporating bilinear functions into the discretization at the interface. Similarly, in [10] an $O(h)$ result in the discrete $H^1$ norm is obtained for a diffusion equation on general triangulations, and this result is also strengthened to $O(h^2)$. This theory, however, does not apply to composite grids.

More recently, Süli [42] obtained similar results for Possion’s equation
using a finite volume discretization. His analysis, which applies to Cartesian product grids, involves reformulating the finite volume method as a Petrov-Galerkin finite element method. This yields $O(h^\sigma)$ estimates in the discrete $H^1$ norm when $u \in H^{1+\sigma}$ ($1/2 < \sigma \leq 2$). Also, the case of a finite volume discretization for composite grids has been analyzed in [22]. This corresponds closely to the method used here, but differs in using a strictly cell-centered grid. There, a bound similar to the previous one is obtained for the continuous $H^1$ norm.

We note that for finite elements on composite grids, accuracy estimates in the coarse component meshwidth are facilitated by the conformity relationship (3.5). Consider, for example, problem (3.2) with the given boundary conditions replaced by $u = 0$ on all of $\partial \Omega$. Let $\Phi_h$ be a global coarse space defined by using piecewise linear functions on a uniform triangulations as indicated earlier. Associate with the corresponding global coarse grid a composite grid via the procedure of patch refinement. Let $\Psi_h$ be the global fine space with the meshwidth of the fine patch, and let $\hat{\Phi}_h$ be the composite space for the composite grid, so that

$$\Phi_h \subseteq \hat{\Phi}_h \subseteq \Psi_h \subset H^1_0.$$

(Note that we have allowed for the possibilities of trivial or total refinement on the composite grid.) Define the bilinear form

$$a(u,v) = \int_{\Omega} \epsilon \nabla u \cdot \nabla v + u_x v \, dV,$$

and the $L^2$ norm

$$\|u\|_{L^2(\Omega)}^2 = \int_{\Omega} u^2 \, dV.$$

Then, using the familiar Galerkin approach, with $u$ the weak solution of (3.2) in $H^1_0$ and $\hat{u}_h$ the discrete solution in $\hat{\Phi}_h$, we can use standard theory
to arrive at the $H^1$ estimate

$$\|u - \hat{u}_h\|_{H^1(\Omega)} \leq c \|u - \hat{u}_h\|_{H^1(\Omega)} \forall \hat{u}_h \in \hat{\Phi}_h,$$

(3.8)

where $c = 4/\varepsilon$ (is independent of $h$). The $H^1$ norm is given by

$$\|u\|_{H^1(\Omega)}^2 \equiv \sum_{0 \leq |\alpha| \leq 1} \|D^{\alpha}u\|_{L^2(\Omega)}^2 = \int_{\Omega} (\nabla u \cdot \nabla u + u^2) \, dV.$$

Now, since $\Phi_h \subseteq \hat{\Phi}_h$, then $\hat{u}_h$ on the right may be replaced by $\pi_hu$, the interpolant of $u$ from $\Phi_h$. This then can be used (see [41], for example) to obtain an $O(h)$ estimate for $\|u - \hat{u}_h\|_{H^1(\Omega)}$. Also, Nitsche’s method [41], [16] is applicable, yielding an $O(h^2)$ estimate in the $L^2$ norm.

We note that this result is somewhat unsatisfactory in light of the $\varepsilon^{-1}$ present in the bound (3.8). However, if $\varepsilon$ is not particularly small the result may be meaningful, and, in any case, convergence of the composite solution to the weak solution is guaranteed as the coarse component is refined.

The above result is independent of the size, position, and level of refinement of the composite grid’s patch, i.e., it made no use of the inequality

$$\hat{\Phi}_h \subseteq \Psi_h.$$

It is clear that variations in the patch must affect the accuracy of the discrete solution (our motivation for using local refinement is based on this assumption). At one extreme, if the patch is reduced to a single point, or if no actual refinement is used, then the estimates in $h$ still hold. At the other extreme, if the patch is allowed to cover the entire domain and $m > 0$ refinements are used, then the above estimates hold with $h$ replaced by $h_F = h/2^m$. Using nontrivial refinement on a judiciously chosen patch, we expect accuracy in the discrete solution that lies somewhere between these two extreme estimates.
In particular, this will be true if $u$ has small derivatives outside the patch. Suppose we partition the domain into two subregions, $\Omega = \Omega_C \cup \Omega_F$, where $\Omega_F$ is the rectangle with corners corresponding to those of the interface. Then, for $\hat{v}_h \in \hat{\Phi}_h$, we have

$$\|u - \hat{v}_h\|_{H^1(\Omega)} \leq \|u - \hat{v}_h\|_{H^1(\Omega_C)} + \|u - \hat{v}_h\|_{H^1(\Omega_F)}.$$ 

Now let $\hat{v}_h$ assume the role of the interpolant of $u$ from the composite space $\hat{\Phi}_h$. Due to the particular form of $\hat{\Phi}_h$, $\hat{v}_h$ may be constructed using basis elements for the global fine space $\Psi_h$ on the region $\Omega_F$, and basis elements for the global coarse space $\Phi_h$ on the region $\Omega_C$. In $\Omega_C$ this is true because, by construction, each composite basis function associated with a coarse or interface nodes agrees in $\Omega_C$ with the corresponding global coarse basis function. Therefore, we obtain the following standard estimate [16] in $\Omega_C$:

$$\|u - \hat{v}_h\|_{H^1(\Omega_C)} \leq k_C h \|u\|_{H^2(\Omega_C)},$$

where the $H^2$ seminorm is defined as

$$\|u\|_{H^2(\Omega)}^2 = \sum_{|\alpha| = 2} \|D^\alpha u\|_{L^2(\Omega)}^2.$$

A similar result for the norm over the region $\Omega_F$ would follow with $h$ replaced by $h_F$ except for one consideration: in order that the interpolant over $\Omega$ be a member of $\hat{\Phi}_h$, the interpolant constructed in $\Omega_F$ is generally not allowed to interpolate $u$ at the slave nodes. This is because, in the composite space, the function values at slave nodes are directly determined by the function values at interface nodes (recall the process of interpolation at the interface), and the latter are specified by the interpolant in $\Omega_C$. A result like the one above using $h_F$, then, may only be applied in the region, denoted by $\Omega_F$, say, underlying the set of nodes associated with the fine
patch (recall Figure 3.2). More work is called for in order to obtain an interpolation estimate in the region lying between $\Omega_F$ and $\Omega_C$. Intuitively, one should be able to obtain an $O(h)$ bound here. Rather than pursue this, let us suppose that the component of the interpolant on $\Omega_F$ does actually interpolate $u$ at the fine boundary. This will happen, for example, if the derivatives of $u$ are zero in the complement of $\Omega_F$. Then we have the estimate

$$\|u - \hat{u}_h\|_{H^1(\Omega)} \leq c \ k_F h \ |u|_{H^2(\Omega_C)} + c \ k_F h_F \ |u|_{H^2(\Omega_F)}.$$  

Notice that, if the derivatives are indeed zero outside of $\Omega_F$, then the first term on the right is zero and we obtain $O(h_F)$ convergence as the patch is refined. This is the kind of behavior we expect, generally, if the derivatives of $u$ are small outside of some region and the patch is sufficiently large to encompass the region. On the other hand, if $u$ has steep gradients outside of the patch, then the best we can expect is $O(h)$ convergence as the coarse grid is refined. We note that a bound similar to the one above is obtained in [22] for a finite volume discretization on a cell-centered composite grid.

Although these observations have been made with regard to finite element discretization, we expect similar accuracy for appropriate finite volume and finite volume element discretizations. In particular, the above example is analogous to the centered version of the finite volume discretization used here.

In the next section, some examples are presented that indicate the actual convergence behavior of finite volume discretizations of problem (3.2). Specifically, we consider the use of upwind or centered approximations on the coarse component of the grid and allow for upwind, higher-order upwind, and centered approximations to be used on the fine patch. Generally,
our computational results agree with the behavior predicted by the above analysis. However, there is still much work to be done in terms of estimates for the accuracy of composite grid discretizations. In particular, we have found that the use of upwinding here gives rise to accurate solutions and very fast discrete solvers. Therefore, it would be appropriate to pursue its use with respect to the element methods. One attractive possibility is that of applying the streamline diffusion method [26] in this context.

Before presenting the results of our test problems, we make one final observation with respect to accuracy and the relationship between the coarse and fine components in the context of the advection-dominated problem. We have just observed that when the derivatives of \( u \) are small outside the patch, an accurate solution can be obtained with sufficient patch refinement. Although this is the case, one should not expect inaccuracies to then be restricted entirely to the fine grid. Because the problem is advection dominated, the solution at a given point is dependent on the solution upwind from that point. A similar dependency should be expected in the discrete solution. In particular, the solution on the coarse component downwind from the patch will inherit the inaccuracy that is a result of using insufficient refinement on the fine component. Therefore, it is possible for the true solution to be uniformly well-behaved throughout the portion of the domain corresponding to the coarse component, while the accuracy of its discrete counterpart changes significantly in this region. This behavior of the discrete solution will be observed in the test problems of the next section.

3.3.1 Sample Problems in Two Dimensions

The following test problems are constructed from (3.2) by determining \( f \)
and the boundary data from the specified solution. We employ constants 
$r > 0$ and $0 < x_0, y_0 < 1$. With respect to the point $(x_0, y_0)$, define
$D(x, y) = \sqrt{(x - x_0)^2 + (y - y_0)^2}$.

**Problem 3.1.**

The solution is

$$u(x, y) = -2 \cdot \left( (D(x, y)/r)^3 - \frac{3}{2} (D(x, y)/r)^2 \right), \quad D(x, y) \leq r,$$

$$u(x, y) = 1, \quad D(x, y) > r.$$  

This solution has been designed to be smooth at the interface between its two components. Specifically, at the set of points $D(x, y) = r$, $u$ is continuous and differentiable, and its gradient is zero:

$$u_x(x, y) = -6(x - x_0)(D(x, y)/r - 1)/r^2,$$

$$u_y(x, y) = -6(y - y_0)(D(x, y)/r - 1)/r^2.$$  

We note that $u$ is independent of $\epsilon$, and the extent to which the function is well-behaved depends on the magnitude of $r$. When this quantity is small, the nonconstant component of $u$ covers a small region, however its gradient there is comparable to $1/r$. On the other hand, if $r$ is kept bounded away from zero, then we have a moderately well-behaved solution irrespective of the choice of $\epsilon$. The graph of $u(x, y_0)$ with $x_0 = y_0 = 1/2$ and $r = 1/4$ is shown in Figure 3.11; the graph of $u(x, y)$ is obtained by rotating it about the $u$-axis at $(x_0, y_0)$.

**Problem 3.2.**

The solution is

$$u(x, y) = e^{-D^2(x,y)/(4\epsilon^2)}.$$
We remark that $u$ and its derivatives are small except in a neighborhood of $(x_0, y_0)$, where the function undergoes rapid growth. For simplicity, let $r = 1$ and $x_0 = y_0 = 0$. Then

$$u_x = -u \cdot 2x/\epsilon, \ u_{xx} = (2u/\epsilon) \cdot \left(\frac{2x^3}{\epsilon} - 1\right),$$

so that $u$ has its maximum rate of change at $D(x, y) = \sqrt{\epsilon}$, with rate $\|\nabla u\| = 2/(\epsilon \cdot \sqrt{\epsilon})$. This represents a sharper front to be resolved than in Problem 3.1.

![Graph of the solution of Problem 3.1, with $y = y_0$.](image)

Convergence rates as functions of grid refinement for these test problems are presented in Tables 3.1-3.5. Relative errors involving the discrete solution of the problems and the known solution evaluated at the nodes are presented. The discrete maximum norm, $\| \cdot \|_{\infty}$, is computed on the composite grid. The discrete $L^2$ norm, $\| \cdot \|_2$, is computed on a global fine grid having the same meshsize as the fine patch. The errors are computed on the latter grid by interpolating the discrete solution to it from
the composite grid, while the true solution is simply evaluated there. Various combinations of finite volume (upwind and centered) discretizations are used on the coarse and fine components of the composite grid. In addition, in some instances a discrete solution is computed on a global fine grid with the same refinement as the fine patch in order to compare the composite grid solution with a uniform grid solution. The discrete solution on all grids is computed by an iterative method and using the convergence criterion that the relative residual satisfy

\[ \| r^i \|_2 / \| r^0 \|_2 \leq 10^{-6}. \]

Here, \( r^i \) is the \( i \)th composite residual and \( r^0 \) is the original right-hand side of the composite grid equations. The iterative solver used for these tests is a particular implementation of the two-level FAC algorithm described in Chapter 4 of this study (see, also, Section 5.3.1). This is not the most efficient algorithm presented herein, but its use is appropriate for these tests in that the "exact" solution of the composite grid equations is desired. An extensive treatment of the efficiency of this algorithm and alternative ones is presented in Chapters 4 and 5.

In what follows, the meshwidth of the coarse component is denoted by \( h = 1/2^{m_c} \), for some positive integer \( m_C \). A positive integer \( m \) denotes the additional number of refinements in meshwidth (each by a factor of two) in going from the coarse component to the fine patch. Hence, the fine patch has a meshwidth of \( h_F = h/2^m \).

Tables 3.1-3.2 show the results for Problem 3.1 with \( r = 3/32, x_0 = 1/2, \) and \( y_0 = 3/8 \). The coarse component meshwidth, \( h = 1/32 \), is kept constant and the patch is successively refined. The southwest corner of the interface is located at \((12/32, 7/32)\), with \( x \) and \( y \) dimensions of \( 8/32 \) and
10/32, respectively. The patch here is made large enough to encompass the nonconstant component of \( u \). Upwind discretization is used on the coarse component of the composite grid, and two test results are shown, using upwinding and higher-order upwinding (Hup) on the patch. The diffusion coefficient is \( \epsilon = 10^{-4} \).

Table 3.1

<table>
<thead>
<tr>
<th>( m )</th>
<th>Composite Grids ( h_u / h_w )</th>
<th>Upwind/Hup</th>
<th>Global Fine Grid Upwind</th>
</tr>
</thead>
<tbody>
<tr>
<td>( | e_h |<em>\infty / | u |</em>\infty )</td>
<td>( | e_h |<em>\infty / | u |</em>\infty )</td>
<td>( | e_h |<em>\infty / | u |</em>\infty )</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1.383 ( \cdot 10^{-1} )</td>
<td>5.551 ( \cdot 10^{-2} )</td>
<td>1.383 ( \cdot 10^{-1} )</td>
</tr>
<tr>
<td>2</td>
<td>6.569 ( \cdot 10^{-2} )</td>
<td>1.549 ( \cdot 10^{-2} )</td>
<td>6.568 ( \cdot 10^{-2} )</td>
</tr>
<tr>
<td>3</td>
<td>3.194 ( \cdot 10^{-2} )</td>
<td>3.977 ( \cdot 10^{-3} )</td>
<td>3.194 ( \cdot 10^{-2} )</td>
</tr>
<tr>
<td>4</td>
<td>1.576 ( \cdot 10^{-2} )</td>
<td>9.605 ( \cdot 10^{-4} )</td>
<td>1.576 ( \cdot 10^{-2} )</td>
</tr>
<tr>
<td>5</td>
<td>7.828 ( \cdot 10^{-3} )</td>
<td>2.273 ( \cdot 10^{-4} )</td>
<td>7.808 ( \cdot 10^{-3} )</td>
</tr>
</tbody>
</table>

The error norms show \( O(h_F) \) and \( O(h_F^2) \) convergence for the upwind and higher-order upwind discretizations, respectively. In addition, the results for the upwind composite solution and global fine solution are nearly identical, with the composite solution showing a mild degradation in the \( L^2 \) norm.

Table 3.2

<table>
<thead>
<tr>
<th>( m )</th>
<th>Composite Grids ( h_u / h_w )</th>
<th>Upwind/Hup</th>
<th>Global Fine Grid Upwind</th>
</tr>
</thead>
<tbody>
<tr>
<td>( | e_h |_2 / | u |_2 )</td>
<td>( | e_h |_2 / | u |_2 )</td>
<td>( | e_h |_2 / | u |_2 )</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1.069 ( \cdot 10^{-2} )</td>
<td>3.670 ( \cdot 10^{-3} )</td>
<td>1.095 ( \cdot 10^{-2} )</td>
</tr>
<tr>
<td>2</td>
<td>5.359 ( \cdot 10^{-3} )</td>
<td>9.532 ( \cdot 10^{-4} )</td>
<td>5.422 ( \cdot 10^{-3} )</td>
</tr>
<tr>
<td>3</td>
<td>2.687 ( \cdot 10^{-3} )</td>
<td>2.500 ( \cdot 10^{-4} )</td>
<td>2.703 ( \cdot 10^{-3} )</td>
</tr>
<tr>
<td>4</td>
<td>1.346 ( \cdot 10^{-3} )</td>
<td>6.589 ( \cdot 10^{-5} )</td>
<td>1.350 ( \cdot 10^{-3} )</td>
</tr>
<tr>
<td>5</td>
<td>6.735 ( \cdot 10^{-4} )</td>
<td>1.663 ( \cdot 10^{-5} )</td>
<td>6.750 ( \cdot 10^{-4} )</td>
</tr>
</tbody>
</table>
Tables 3.3-3.4 show the results for Problem 3.2 with \( r = 1, x_0 = y_0 = 1/2, \) and \( \epsilon = 10^{-3} \). Standard upwinding is used on the coarse component of the composite grid, with meshwidth \( h = 1/32 \). The higher-order upwind and centered discretizations were used on a patch centered at \((x_0, y_0)\) with length and height equal to 1/4. Again, the patch is made sufficiently large, this time to enclose the solution’s sharp-front. The problem is also solved using the centered method on the global fine grid. The convergence rate for the discretizations is \( O(h^2) \), again with close agreement between the composite and global fine solutions.

### Table 3.3
Relative errors in the maximum norm as functions of patch refinement for Problem 3.2.

<table>
<thead>
<tr>
<th>( m )</th>
<th>Composite Grids</th>
<th>Global Fine Grid</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Upwind/Hup</td>
<td>Centered</td>
</tr>
<tr>
<td></td>
<td>( |e_h|<em>{\infty}/|u|</em>{\infty} )</td>
<td>( |e_h|<em>{\infty}/|u|</em>{\infty} )</td>
</tr>
<tr>
<td>1</td>
<td>9.538 \cdot 10^{-2}</td>
<td>1.051 \cdot 10^{-1}</td>
</tr>
<tr>
<td>2</td>
<td>3.271 \cdot 10^{-2}</td>
<td>2.253 \cdot 10^{-2}</td>
</tr>
<tr>
<td>3</td>
<td>9.279 \cdot 10^{-3}</td>
<td>5.460 \cdot 10^{-3}</td>
</tr>
<tr>
<td>4</td>
<td>2.397 \cdot 10^{-3}</td>
<td>1.356 \cdot 10^{-3}</td>
</tr>
<tr>
<td>5</td>
<td>6.021 \cdot 10^{-4}</td>
<td>3.383 \cdot 10^{-4}</td>
</tr>
</tbody>
</table>

Table 3.5 shows the results for a different set of tests based on Problem 3.2, measuring convergence as a function of \( h \). Here, \( \epsilon \) is increased to \( 10^{-2} \), allowing the front to spread out while the location and dimensions of the interface are left the same as for the last set of tests. Again, the upwind/centered and centered/centered combinations are used, but now on three composite grids, each having two levels of patch refinement and varying coarse component meshwidths.
Table 3.4
Relative errors in the $L^2$ norm as functions of patch refinement for Problem 3.2.

<table>
<thead>
<tr>
<th>$m$</th>
<th>Composite Grids</th>
<th>Upwind/Centered</th>
<th>Global Fine Grid Centered</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$|e_h|_2/|u|_2$</td>
<td>$|e_h|_3/|u|_2$</td>
<td>$|e_h|_2/|u|_2$</td>
</tr>
<tr>
<td>1</td>
<td>$1.038 \cdot 10^{-1}$</td>
<td>$8.449 \cdot 10^{-2}$</td>
<td>$8.436 \cdot 10^{-2}$</td>
</tr>
<tr>
<td>2</td>
<td>$3.159 \cdot 10^{-2}$</td>
<td>$1.858 \cdot 10^{-2}$</td>
<td>$1.857 \cdot 10^{-2}$</td>
</tr>
<tr>
<td>3</td>
<td>$8.533 \cdot 10^{-3}$</td>
<td>$4.519 \cdot 10^{-3}$</td>
<td>$4.520 \cdot 10^{-3}$</td>
</tr>
<tr>
<td>4</td>
<td>$2.184 \cdot 10^{-3}$</td>
<td>$1.122 \cdot 10^{-3}$</td>
<td>$1.123 \cdot 10^{-3}$</td>
</tr>
<tr>
<td>5</td>
<td>$5.504 \cdot 10^{-4}$</td>
<td>$2.800 \cdot 10^{-4}$</td>
<td>$2.802 \cdot 10^{-4}$</td>
</tr>
</tbody>
</table>

We make the observation that the patch is "too small" for this problem, in the sense that the innaccuracy on the coarse component is inherited by the patch (and also affects the coarse component downwind from the patch). As a result, the effectiveness of increasing the patch refinement has deteriorated somewhat. For the upwind/centered discretization with $m_c = 4$ and $m = 2$ (second row, first column of Table 3.5), the relative

Table 3.5
Relative errors as functions of coarse component refinement for Problem 3.2.

<table>
<thead>
<tr>
<th>$m_c$ ($m = 2$)</th>
<th>Composite Grids</th>
<th>Upwind/Centered</th>
<th>Centered/Centered</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$|e_h|<em>\infty/|u|</em>\infty$</td>
<td>$|e_h|<em>\infty/|u|</em>\infty$</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>$1.963 \cdot 10^{-1}$</td>
<td>$1.935 \cdot 10^{-1}$</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>$7.981 \cdot 10^{-2}$</td>
<td>$7.468 \cdot 10^{-2}$</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>$3.921 \cdot 10^{-2}$</td>
<td>$1.903 \cdot 10^{-2}$</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>$2.152 \cdot 10^{-2}$</td>
<td>$5.628 \cdot 10^{-3}$</td>
<td></td>
</tr>
</tbody>
</table>

$L^2$ error restricted to the fine patch is approximately $7.67 \cdot 10^{-2}$. With two additional levels of patch refinement ($m = 4$) this becomes $3.78 \cdot$
$10^{-2}$, a modest improvement (in particular since the second-order accurate centered discretization is used on the patch). On the other hand, this additional local refinement has a pronounced effect on the solution as a whole, as is apparent from graphs of the solution presented in Figures 3.12-3.14. In Figure 3.12 the exact solution of Problem 3.2 is plotted. Figures 3.13 and 3.14 show plots corresponding to the two cases of patch refinement just mentioned. In each case, for ease in plotting, the fine patch solution has been transferred by restriction (see Section 3.2.4) to a coarse patch having the same refinement as the coarse component. We emphasize the quality of the coarse component solutions here. Recall that the solution should become uniformly small as the radial distance from the center of the domain increases. In Figure 3.13, this fails to happen due to the innaccuracy of the coarse component solution in the region downwind from the patch. In Figure 3.14, this innaccuracy of the solution has been removed by using purely local refinement *upwind from this region*, i.e., by adding additional refinement on the patch. This example shows that, in the context of advection-dominated problems, local refinement can have a profound effect on the global solution.
Fig. 3.12. The exact solution of Problem 3.2.

Fig. 3.13. The approximate solution of Problem 3.2 with two levels of patch refinement.
Fig. 3.14. The approximate solution of Problem 3.2 with four levels of patch refinement.
4. Convergence of FAC

This chapter studies the convergence of FAC as an iterative solver for advection-dominated advection-diffusion equations on composite grids. We begin with only a few assumptions regarding the structure and origin of the composite grid equations. The composite grid consists of coarse, interface and fine patch components, and coincides with a global coarse grid in its coarse and interface components. Its fine patch may be viewed as being derived, by refinement, from a coarse patch on the global grid.

As for notation, members of the composite grid space are represented by lower case Roman letters distinguished by an underscore, and components by the lower case letters subscripted with \( C \), \( F \), or \( I \). For example, a composite right-hand side (or residual) is written

\[
\mathbf{r} = [r_C, r_F, r_I].
\]

Similar notation is used for composite grid operators and their components. The composite grid equation \( L\mathbf{z} = \mathbf{r} \) may then be written as

\[
\begin{bmatrix}
L_C & 0 & L_{CI} \\
0 & L_F & L_{FI} \\
L_{IC} & L_{IF} & L_I
\end{bmatrix}
\begin{bmatrix}
z_C \\
z_F \\
z_I
\end{bmatrix}
= 
\begin{bmatrix}
r_C \\
r_F \\
r_I
\end{bmatrix}.
\]

(4.1)

A word of explanation with regard to the notation accompanying the components of \( L \): in general, \( L_{XY} \) denotes that portion of the stencil of the matrix \( L \) that represents the connections to nodes in component \( Y \) that appear in the equations for the nodes in component \( X \), i.e., entries \( l_{ij} \) where node \( i \) lies in \( X \) and node \( j \) lies in \( Y \). Also, \( L_I \), for example, stands for the
interface-to-interface connections. Section 4.1 below introduces the two-level version of FAC for general composite grid equations. The results in this section utilize only the algebraic structure of the composite equations and are not at all dependent on the underlying continuous problem or its discretization. Sections 4.2-4.3 are devoted to studying the convergence of this algorithm when it is applied to the finite volume discretizations of the advection-dominated problems (3.1) and (3.2), respectively.

4.1 The Two-Level Method

Assuming that the matrices \( L_C \), \( L_L \), and \( L_F \) are invertible, then performing a few steps of block reduction leads to the equivalent form

\[
\begin{bmatrix}
  L_C & 0 & L_CI \\
  0 & L_F & L_FI \\
  0 & 0 & I
\end{bmatrix}
\begin{bmatrix}
  z_C \\
  z_F \\
  z_I
\end{bmatrix}
= \begin{bmatrix}
  r_C \\
  r_F \\
  \hat{r}_I
\end{bmatrix},
\]

where

\[
\hat{L}_I \equiv L_I - L_ICL_C^{-1}L_CI - L_IFL_F^{-1}L_FI
\]

is the Schur complement in \( L \) with respect to the partitioning (4.1), and

\[
\hat{r}_I = r_I - L_ICL_C^{-1}r_C - L_IFL_F^{-1}r_F.
\]

In principle, then, the composite grid equations could be solved directly by the following three-step procedure.

Step 1. Solve \( \hat{L}_Iz_I = \hat{r}_I \) for \( z_I \).

Step 2. Solve \( L_Cz_C = r_C - L_CIz_I \) for \( z_C \).

Step 3. Solve \( L_Fz_F = r_F - L_FIz_I \) for \( z_F \).

We note that this is just the familiar Schur complement method for solving a system partitioned as in (4.1).
FAC [32], [33] is an adaptation of multigrid to composite grid problems and, as such, makes use of a coarse grid discretization of the differential operator to obtain approximations to the fine grid error left by relaxation. In the present context, one may think of the fine grid operator as being the composite grid operator defined above. As for the coarse grid operator, it is useful in defining it to retain some flexibility. We accept in this role a discretization on any of the composite grids that lie between the global coarse grid and the true composite grid (the grid on which the discrete solution is ultimately sought). These grids are the ones obtained from the global coarse grid in the same manner as the true composite grid, but differ by having less refinement on the patch (all grids agree in their coarse and interface components). An important special case to consider is the global coarse grid viewed as a composite grid. Since this grid is uniform, in principle, no special considerations need apply in defining the operator, that is, a standard discretization based on a lexicographic ordering of the nodes and equations, for example, may be used. Also, since the grid is uniform, it is possible to apply any one of a variety of well-known, highly effective methods to solve standard problems on this grid.

The uniformity of the global coarse grid is its attractive feature. However, we retain the viewpoint put forth while defining this grid (see Section 3.1) that it is itself a composite grid and that the discretization on it yields a composite grid operator. It makes sense, then, to refer in general to the coarse grid as the coarse composite grid (the phrase composite grid may be reserved for the true composite grid when distinguishing the levels in the two-level algorithm). Once this viewpoint is taken, the strategy of using different discretizations on the coarse and patch components of the coarse grid becomes a possibility for all choices of this grid. We have already
considered, in Section 3.3.1, the effect on the accuracy of the composite grid solution of using different discretizations on the components of the composite grid. We investigate in this chapter the effect of using various discretizations on the coarse grid in terms of FAC convergence.

When denoting the grid operators, $L$ is used as above in denoting the composite operator and its components, and $A$ is used similarly for the coarse (composite) grid operator. Generally, notation developed for global coarse objects is used to refer to coarse composite ones. So, for example, the refinement region on the coarse composite grid is referred to as “the coarse patch.”

Let $A$ be the discrete operator on this grid. Its component form is

$$A = \begin{bmatrix} A_G & 0 & A_{GI} \\ 0 & A_P & A_{PI} \\ A_{IG} & A_{IP} & A_I \end{bmatrix}.$$ 

Here, the subscript $P$ refers to the coarse patch, as opposed to $F$, which refers to the fine patch component of the (true) composite grid.

The following description of FAC corresponds to the delayed correction version in its two-grid form as described in [33]. Here, a subscript $G$ is used to distinguish coarse composite vectors and components from true composite ones.

**FAC algorithm for the solution of $Lz = r$**

Let $z^0$ be given and set $r^0 = r - Lz^0$.

Loop on $i$: $i = 0, 1, \ldots$ until convergence.

Step 1. Solve $Az_G = r_G = (r^i_C, r^i_P = I_F r^i_F, r^i_I)$ for

$$z_G = (z_{G,C}, z_{G,P}, z_{G,I}).$$
Step 2. Solve \( L_F z_F^i = r_F^i - L_{FI} z_{GI} \) for \( z_F^i \).

Step 3. Perform the composite correction

\[
\begin{align*}
    z_C^{i+1} &= z_C^i + z_{G,C} \\
    z_F^{i+1} &= z_F^i + z_F \\
    z_I^{i+1} &= z_I^i + z_{G,I}.
\end{align*}
\]

Step 4. Set \( z^{i+1} = (z_C^{i+1}, z_F^{i+1}, z_I^{i+1}) \), and form the new composite residual,

\[
r^{i+1} = r - L z^{i+1}.
\]

End loop.

To express the algorithm in words, each iteration begins by transferring the current residual to the coarse patch (dropping the iteration indices):

\[ r_P \leftarrow I_P^P r_F. \]

Here, \( I_P^P \) represents a restriction operator mapping the fine patch to the coarse patch. Then, \( r_P \) and the other components of the composite residual form the right-hand side for the coarse composite equations. This system is solved to obtain \( z_G \), a global coarse (or coarse composite) grid approximation to the composite grid error. To obtain a fine patch error approximation, the local patch problem is solved with the fine patch residual as the right-hand side. Dirichlet values are provided for this problem by the interface values of \( z_G \). (As a practical matter, recalling the discussion in Section 3.2.2 on the treatment of the fine-to-interface stencil, in order to form the vector of boundary values in that context, the interface component of \( z_G \) is necessarily mapped to a refined interface using the interpolation operator \( I^P_I \).) The solution \( z_F^i \) of this problem, along with the coarse and interface components of \( z_G \), is used in the final step to define a
composite correction, which is added to the current approximation to the solution. Finally, the residual is updated.

Having defined the FAC algorithm, we now turn to a study of its convergence properties, concentrating on an expression for the updating of the composite residual. In what follows, we assume appropriate consistency [17] for \( L, A, \) and their components.

**THEOREM 4.1.** Let the composite operators \( A \) and \( L \) be coarse grid compatible in the sense that \( A_C = L_C \) and \( A_{CI} = L_{CI} \), and let \( r_i \) be the \( i \)th composite residual obtained by using the above two-level FAC scheme. Then for \( i > 0 \), we have \( r^{i+1}_C, r^{i}_F = 0 \) and

\[
    r^{i+1}_i = (I - \hat{L}_I \hat{A}_I^{-1}) r^i_i,
\]

(4.3)

where \( \hat{L}_I \) and \( \hat{A}_I \) are the respective Schur complements in \( L \) and \( A \).

**Proof.** According to the Schur reduced form of the coarse composite equations, Step 1 of FAC (the coarse composite solve) yields \( z_G \) satisfying

\[
    (A_I - A_{IC} A_C^{-1} A_{CI} - A_{IP} A_F^{-1} A_{PI}) z_{G,I} = r^i_i - A_{IC} A_C^{-1} r^i_C - A_{IP} A_F^{-1} I_F r^i_F,
\]

(4.4)

and

\[
    A_G z_{G,G} = r^i_C - A_{CI} z_{G,I}.
\]

Also, the update of the composite residual may be written as \( r^{i+1}_i = r^i_i - L z'_G \), where

\[
    z'_G = (z_{G,G}, z'_F, z_{G,I}).
\]

Using these relationships along with (4.1), then the residual update may be written in component form as

\[
    r^{i+1}_C = r^i_C - L_C z_{G,C} - L_{CI} z_{G,I} = r^i_C - L_C (A_C^{-1} (r^i_C - A_{CI} z_{G,I})) - L_{CI} z_{G,I},
\]

(4.5)

64
\[ r_F^{i+1} = r_F^i - L_F z_F^i - L_{FI} z_{G,I} \]
\[ = r_F^i - L_F( L_F^{-1}(r_F^i - L_{FI} z_{G,I})) - L_{FI} z_{G,I}, \] (4.6)

and

\[ r_I^{i+1} = r_I^i - L_I z_{G,I} - L_{IC} z_{G,C} - L_{IF} z_F' \]
\[ = r_I^i - (L_I z_{G,I} + L_{IC} A_C^{-1}(r_C^i - A_C z_{G,I}) + L_{IF} L_F^{-1}(r_F^i - L_{FI} z_{G,I})) \]
\[ = r_I^i - L_{IC} A_C^{-1} r_C^i - L_{IF} L_F^{-1} r_F^i - (L_I - L_{IC} A_C^{-1} A_C - L_{IF} L_F^{-1} L_{FI}) z_{G,I}. \]

Now by the assumed compatibility, we see from (4.5) that \( r_C^{i+1} = 0 \), so the first conclusion holds. Now consider \( i > 0 \) and include the vanishing of these components in the above expression for \( r_I^{i+1} \). It then simplifies to

\[ r_I^{i+1} = r_I^i - (L_I - L_{IC} A_C^{-1} A_C - L_{IF} L_F^{-1} L_{FI}) z_{G,I}. \]

Recalling the expression (4.4) for \( z_{G,I} \), and including the above substitutions and simplifications yields

\[ r_I^{i+1} = (I - \hat{L}_I \hat{A}_I^{-1}) r_I^i, \] (4.7)

where \( \hat{L}_I \) is the Schur complement (4.2) and \( \hat{A}_I \) is the Schur complement on the coarse grid, which we may write as

\[ \hat{A}_I = A_I - A_{IC} A_C^{-1} L_{CI} - A_{IP} A_F^- A_{PI}. \]

We note that a result similar to this theorem appears in [22].

With the compatibility assumption, then, we see that after the initial FAC step both \( r_C \) and \( r_F \) are zero (notice that (4.6) makes the application of the restriction operator \( I_F^P \) in Step 1 redundant; we may simply take \( r_F^i = 0 \)) and the convergence of \( r_I \) to zero is governed by the eigenvalues
of $I - \hat{L}_I \hat{A}_I^{-1}$. Although we only used the assumption that $A_C = L_C$ and $A_{CI} = L_{CI}$ above, it is also appropriate to make the assumption for interface-to-coarse connections, i.e.,

$$A_{IC} = L_{IC}. \tag{4.8}$$

Let $(\lambda, x)$ be an eigenpair of $\hat{L}_I \hat{A}_I^{-1}$ and set $y = \hat{A}_I^{-1}x$, so that the equation

$$\hat{L}_I \hat{A}_I^{-1}x = \lambda x$$

becomes

$$(\hat{L}_I - \lambda \hat{A}_I)y = 0.$$ 

The eigenvalues of $\hat{L}_I \hat{A}_I^{-1}$ are therefore those values of $\lambda$ for which the matrix

$$L_I - \lambda A_I + (\lambda - 1) L_{IC} L_C^{-1} L_{CI} - L_{IF} L_{F}^{-1} L_{FI} + \lambda A_{IP} A_P^{-1} A_{PI} \tag{4.9}$$

is singular (notice that we have included (4.8) in the compatibility assumptions in order to obtain this latest expression). This criterion for determining the eigenvalues will be applied in the next section.

4.2 Convergence in One Dimension

We now turn to an examination of the convergence of the FAC algorithm when it is applied to the solution of the finite volume discretization of problem (3.1). Since the operators $A$ and $L$ arising from this discretization satisfy the compatibility conditions $A_C = L_C, A_{CI} = L_{CI}$, and $A_{IC} = L_{IC}$ of the previous section, it follows from Theorem 4.1 that convergence of the algorithm is governed by the relationship (4.3). We are therefore interested in the eigenvalues of the matrix $\hat{L}_I \hat{A}_I^{-1}$. Since (4.8) is valid, it is sufficient to set the determinant of the matrix (4.9) equal to zero and then solve
for the resulting values of $\lambda$. Our analysis begins (Section 4.2.1) with the case that the composite grid has a single refinement region. Notice that in this case two nodes comprise the interface, so the dimensionality of the eigenvalue problem is equal to two. Therefore, identifying the required eigenvalues is "simply" a matter of solving a quadratic equation. For the problem under consideration, the analysis is facilitated by the fact that this quadratic appears, essentially, in factored form. Later, we extend this analysis by examining (in Section 4.2.3) the case of multiple refinement regions.

4.2.1 The Case of a Single Refinement Region

We now proceed with an analysis of the spectrum of the operator $\hat{L}_I \hat{A}_I^{-1}$ corresponding to a finite volume discretization of the problem (3.1). With a single refinement region on the one-dimensional composite grid, and using either the centered or upwind discretizations of Chapter 3, the operator $L$ has the structure shown in Figure 4.1. Denote the components of the composite operators as follows:

$$L_C = \begin{bmatrix} L & 0 \\ 0 & R \end{bmatrix}, \quad L_{CI} = \begin{bmatrix} 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix},$$

$$L_{IC} = \begin{bmatrix} 0 & \cdots & 0 & l IC & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & r IC & \cdots & 0 \end{bmatrix},$$
\[ L_{FI} = \begin{bmatrix} l_{FI} & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & \tau_{FI} \end{bmatrix}, \quad A_{FI} = \begin{bmatrix} l_{PI} & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & \tau_{PI} \end{bmatrix}, \]

\[ L_{IF} = \begin{bmatrix} l_{IF} & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & \tau_{IF} \end{bmatrix}, \quad A_{IP} = \begin{bmatrix} l_{IP} & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & \tau_{IP} \end{bmatrix}. \]

Fig. 4.1. Structure of a one-dimensional composite operator for the case of a single refinement region.
We assume that the Dirichlet value at the left boundary has been moved to the right-hand side of (4.1). Let the respective orders of the matrices $L$, $R$, $L_F$, and $A_P$ be $N_{C_1}$, $N_{C_2}$, $N_F$, and $N_P$. Also, let $N_C$ be one less than the number of nodes on the global coarse grid (i.e., the order of the global coarse matrix with the left Dirichlet condition eliminated). The coarse meshwidth is denoted by $h = 1/N_C$. Denote the $ij$-th element of $X^{-1}$, the inverse of the generic square invertible matrix $X$, by $\tilde{x}_{ij}$. Also, let $X_{j \times j}$ denote the $j \times j$ submatrix located in lower right corner of $X$. For example, if $X$ is of order $N$, then $X_{1 \times 1} = x_{NN}$ and $X_{N \times N} = X$. With this notation, we have

$$L_{IC}L_C^{-1}L_{CI} = \begin{bmatrix} l_{IC}l_{CI} & 0 \\ 0 & \tau_{IC}r_{CI}r_{i1} \end{bmatrix}, \quad (4.10)$$

$$L_IFL_F^{-1}L_{FI} = \begin{bmatrix} l_{IF}l_{FI}l_{F_{i1}} & l_{IF}r_{FI}l_{F_{i, N_P}} \\ \tau_{IF}l_{FI}l_{F_{N_P,i}} & \tau_{IF}r_{FI}l_{F_{N_P,N_P}} \end{bmatrix},$$

and

$$A_IFA_P^{-1}A_{PI} = \begin{bmatrix} l_{IP}l_{PI}l_{P_{i1}} & l_{IP}r_{PI}l_{P_{i, N_P}} \\ \tau_{IP}l_{PI}l_{P_{N_P,i}} & \tau_{IP}r_{PI}l_{P_{N_P,N_P}} \end{bmatrix}.$$
grid, and fine patch. The interface nodes are not specified here. In this section, the type of discretization at these points will always be of the same type used on the coarse component. (In two dimensions, it will be useful to allow more flexibility at the interface of the coarse composite grid.) In one dimension, two possibilities are considered, upwind \((x_i = U)\) and centered \((x_i = C)\) versions of the finite volume discretization. In two dimensions, we will also allow higher-order upwinding to be used on the patch. Generally, it will be possible to make the notation just described less cumbersome by employing obvious abbreviations. Initially, we will consider discretizations that agree in type throughout the two grids.

The first case we consider in detail is the discretization signified by \((U, U : U, U) = U^4\), which corresponds to upwinding used throughout the coarse composite and true composite grids. The analysis will make use of the generic tridiagonal matrix \(X = \text{tri}[-\epsilon - h, 2\epsilon + h, -\epsilon]\) of generic order \(N\). Of interest are the corner elements of \(X^{-1}\): \(\tilde{x}_{11}, \tilde{x}_{1N}, \tilde{x}_{N1}\) and \(\tilde{x}_{NN}\). Let \(X_{ij}\) be the cofactor of \(x_{ij}\). Then \([28]\),

\[
\tilde{x}_{ij} = X_{ji}/\text{Det}X.
\]

**Lemma 4.2.** The four corner elements of \(X^{-1}\) corresponding to the upwind discretization are given by

\[
\tilde{x}_{11} = \tilde{x}_{NN} = \frac{z_{N-1}}{z_N} = \frac{(\epsilon + h)^N - \epsilon^N}{(\epsilon + h)^{N+1} - \epsilon^{N+1}} \approx \frac{1}{\epsilon + h}, \tag{4.11}
\]

\[
\tilde{x}_{1N} = \frac{h \cdot \epsilon^{N-1}}{(\epsilon + h)^{N+1} - \epsilon^{N+1}} \approx 0, \tag{4.12}
\]

and

\[
\tilde{x}_{N1} = \frac{h \cdot (\epsilon + h)^{N-1}}{(\epsilon + h)^{N+1} - \epsilon^{N+1}} \approx \frac{h}{(\epsilon + h)^2}. \tag{4.13}
\]
Proof. Since $X$ is tridiagonal, there is a three-term linear, homogeneous difference equation for the $j$th determinant, $z_j = \text{Det}X_{j\times j}$:

$$z_j = (2\epsilon + h)z_{j-1} - \epsilon(\epsilon + h)z_{j-2}, \quad j = 3, 4, \ldots, N.$$ 

The associated characteristic equation is

$$\mu^2 - \mu(2\epsilon + h) + \epsilon(\epsilon + h) = 0,$$

with roots

$$\mu_1 = \epsilon + h, \quad \mu_2 = \epsilon.$$

Hence, it follows from the theory of difference equations that

$$z_j = S(\epsilon + h)^j + T\epsilon^j,$$

where $S$ and $T$ are unknown constants which are determined uniquely by the values of $z_1$ and $z_2$. But

$$z_1 = S(\epsilon + h) + T\epsilon = 2\epsilon + h$$

$$z_2 = S(\epsilon + h)^2 + T\epsilon^2 = 3\epsilon^2 + 3\epsilon h + h^2.$$

Thus,

$$S = (\epsilon + h)/h, \quad T = -\epsilon/h$$

and we have

$$z_j = (\epsilon + h)^{j+1}/h - \epsilon^{j+1}/h.$$

Now since $X$ is tridiagonal, it follows that the cofactors satisfy

$$X_{11} = X_{NN} = \text{Det}X_{N-1\times N-1}.$$

Thus,

$$\tilde{z}_{11} = \tilde{z}_{NN} = \frac{z_{N-1}}{z_N} = \frac{(\epsilon + h)^N - \epsilon^N}{(\epsilon + h)^{N+1} - \epsilon^{N+1}}.$$
Also, $X_{N_1} = e^{N-1}$ implies that

$$\tilde{x}_{1N} = \frac{h \cdot e^{N-1}}{(e + h)^{N+1} - e^{N+1}}.$$  

Finally, $X_{1N} = (e + h)^{N-1}$ implies that

$$\tilde{x}_{N_1} = \frac{h \cdot (e + h)^{N-1}}{(e + h)^{N+1} - e^{N+1}}.$$  

More will be said shortly regarding the quality of the approximations given in the lemma (see the comments below and also the next section), but because of the last two approximations we will see that $\hat{L}$ and $\hat{A}$ are essentially lower triangular, so $\text{Det}(\hat{L}_I - \lambda \hat{A}_I)$ just depends on the diagonal part of the matrix.

We may let $X$ here assume the role of $A_P$ when the coarse patch has trivial refinement, i.e., when the global coarse grid plays the role of the coarse composite grid in the two-level version of FAC. We have just found the required elements of the inverse of this operator. In this context, it may be useful to point out an apparent ambiguity in the above expressions. Notice that in these expressions, $h$ is always directly related to the dimension, $N_C$, of the global coarse grid. However, the role of $N$ in the expressions is allowed to vary. For example, when the approximations are made with regard to $A_P$, then $N \leftarrow N_P$ in (4.11-4.13), and note that, necessarily, $N_P < N_C$. One sees that there is an interplay between the size of $\epsilon$ and the dimensions of the global coarse and coarse patch grids that enters into the validity of the approximations. In this particular case, their validity is commensurate with $\epsilon$ being small with respect to $h$ (the reciprocal of $N_C$) and the coarse patch being sufficiently large.

Also, $X$ may play the role of the coarse component operators $L$ and $R$,
so that

\[ \tilde{i}_{N_1, N_2}, \tilde{r}_{11} \approx \frac{1}{\epsilon + h}. \]

Now suppose that there are \( m \) levels of refinement on the fine patch. Then

\[ L_F = 2^m \cdot \text{tri}[-\epsilon - h/2^m, 2\epsilon + h/2^m, -\epsilon], \]

and

\[ \text{Det} L_F = (2^m)^{N_F}((\epsilon + h/2^m)^{N_F+1} - \epsilon^{N_F+1})/(h/2^m). \]

Using these expressions, we obtain the following corollary to Lemma 4.2.

**COROLLARY 4.3.** The corner elements of the fine patch matrix \( L_F \) corresponding to the upwind discretization are given by

\[ \tilde{i}_{F_{11}} = \tilde{i}_{F_{N_1, N_2}} = \frac{(\epsilon + h/2^m)^{N_F} - \epsilon^{N_F}}{2^m((\epsilon + h/2^m)^{N_F+1} - \epsilon^{N_F+1})} \approx \frac{1}{2^m \epsilon + h}, \]

\[ \tilde{i}_{F_{1, N_F}} = \frac{h \cdot \epsilon^{N_F-1}}{2^m((\epsilon + h/2^m)^{N_F+1} - \epsilon^{N_F+1})} \approx 0, \]

and

\[ \tilde{i}_{F_{N_1, N_2}} = \frac{h \cdot (\epsilon + h/2^m)^{N_F-1}}{2^m((\epsilon + h/2^m)^{N_F+1} - \epsilon^{N_F+1})} \approx \frac{h}{(2^m \epsilon + h)^2}. \]

The results of Corollary 4.3 may also be applied to the inverse of \( A_F \), when an intermediate composite grid is used in place of the global coarse grid, by performing the replacement \( m \leftarrow m - k \) \( (0 \leq k < m) \).

In addition to these quantities, the pertinent quantities from the FV discretization of Section 3.2.1 are (here \((u)\) denotes “upwind”, and \((c)\), “centered”):

\[ l_{IC} = -\epsilon - h \, (u), \quad -\epsilon - h/2 \, (c), \]
\[ l_{CI} = -\epsilon \, (u), \quad -\epsilon + h/2 \, (c), \]
\[ r_{1C} = -\epsilon \, (u), \quad -\epsilon + h/2 \, (c), \]
\[ r_{CI} = -\epsilon - h \, (u), \quad -\epsilon - h/2 \, (c), \]

73
\[ l_{IF} = -2^m \epsilon (u), \quad l_{FI} = -2^m \epsilon - h (u), \quad r_{IF} = -2^m \epsilon - h (u), \quad r_{FI} = -2^m \epsilon (u), \]
\[ l_{IP} = -2^{m-k} \epsilon (u), \quad r_{IP} = -2^{m-k} \epsilon - h (u), \quad l_{PI} = -2^{m-k} \epsilon - h (u), \quad r_{PI} = -2^{m-k} \epsilon (u), \]
\[
L_I = \begin{bmatrix}
(1 + 2^m)\epsilon + h & 0 \\
0 & (1 + 2^m)\epsilon + h
\end{bmatrix}, \quad A_I = \begin{bmatrix}
(1 + 2^{m-k})\epsilon + h & 0 \\
0 & (1 + 2^{m-k})\epsilon + h
\end{bmatrix} (u),
\]
\[
L_I = \begin{bmatrix}
(1 + 2^m)\epsilon & 0 \\
0 & (1 + 2^m)\epsilon
\end{bmatrix}, \quad A_I = \begin{bmatrix}
(1 + 2^{m-k})\epsilon & 0 \\
0 & (1 + 2^{m-k})\epsilon
\end{bmatrix} (c).
\]

Notice that we have allowed for general refinement on the coarse patch by letting \(1 \leq k \leq m\). When \(k = m\) this corresponds to letting the global coarse grid play the role of the coarse composite grid, and when \(k = 1\) to using the grid having one less level of patch refinement than the true composite grid.

Now, using the above values and approximations, we see for the \(U^4\) case that \(\tilde{L}_I - \lambda \tilde{A}_I = \)
\[
\begin{bmatrix}
(1 + 2^m)\epsilon + h - \lambda((1 + 2^{m-k})\epsilon + h) & 0 \\
0 & (1 + 2^m)\epsilon + h - \lambda((1 + 2^{m-k})\epsilon + h)
\end{bmatrix}
\]
\[
+ (\lambda - 1) \begin{bmatrix}
\epsilon & 0 \\
0 & \epsilon
\end{bmatrix} - \begin{bmatrix}
2^m \epsilon & 0 \\
h & 2^m \epsilon
\end{bmatrix}
\]
\[
+ \lambda \begin{bmatrix}
2^{m-k} \epsilon & 0 \\
h & 2^{m-k} \epsilon
\end{bmatrix}
\]
\[
= L_I - \lambda A_I + (\lambda - 1) L_{IC} L_{CI}^{-1} - L_{IF} L_{FI}^{-1} + \lambda A_{IP} A_{PI}^{-1} A_{PI}.
\]

A simple calculation then shows that, irrespective of the value of \(k\), the solutions of \(\text{Det}(\tilde{L}_I - \lambda \tilde{A}_I) = 0\) are \(\lambda_1 = \lambda_2 = 1\).
Next, we consider the $C^4$ case where the centered discretization is used throughout. Again, the corners of the inverse matrix are required. Proceeding as in the previous case, let

$$X = \text{tri}[-\epsilon - h/2, 2\epsilon, -\epsilon + h/2]$$

be of order $N$.

**LEMMA 4.4.** The four corner elements of $X^{-1}$ corresponding to the centered discretization are given by

$$\tilde{x}_{11} = \tilde{x}_{NN} = \frac{(\epsilon + h/2)^N - (\epsilon - h/2)^N}{(\epsilon + h/2)^{N+1} - (\epsilon - h/2)^{N+1}} \approx \frac{1}{\epsilon + h/2}, \quad (4.17)$$

$$\tilde{x}_{1N} = X_{1N}/z_N = \frac{h \cdot (-\epsilon + h/2)^{N-1}}{(\epsilon + h/2)^{N+1} - (\epsilon - h/2)^{N+1}} \approx 0, \quad (4.18)$$

and

$$\tilde{x}_{N1} = X_{N1}/z_N = \frac{h \cdot (\epsilon + h/2)^{N-1}}{(\epsilon + h/2)^{N+1} - (\epsilon - h/2)^{N+1}} \approx \frac{h}{(\epsilon + h/2)^2}, \quad (4.19)$$

**Proof.** The $j$th determinant satisfies

$$z_j = -2\epsilon z_{j-1} + (\epsilon - h/2)(\epsilon + h/2)z_{j-2}, \quad j = 3, 4, \ldots, N,$$

with associated roots

$$\mu_1 = \epsilon + h/2, \quad \mu_2 = \epsilon - h/2.$$

Hence,

$$z_j = S(\epsilon + h/2)^j + T(\epsilon - h/2)^j,$$

and in particular

$$z_1 = S(\epsilon + h/2) + T(\epsilon - h/2) = 2\epsilon,$$

$$z_2 = S(\epsilon + h/2)^2 + T(\epsilon - h/2)^2 = 3\epsilon^2 + h^2/4.$$

Solving for $S$ and $T$, we find

$$S = (\epsilon + h/2)/h, \quad T = -(\epsilon - h/2)/h.$$
Therefore,
\[ z_j = (\epsilon + h/2)^{j+1}/h - (\epsilon - h/2)^{j+1}/h, \]
and
\[ \tilde{x}_{11} = \tilde{x}_{NN} = \frac{(\epsilon + h/2)^N - (\epsilon - h/2)^N}{(\epsilon + h/2)^{N+1} - (\epsilon - h/2)^{N+1}}. \]

Also,
\[ \tilde{x}_{1N} = X_{1N}/z_N = \frac{h \cdot (-\epsilon + h/2)^{N-1}}{(\epsilon + h/2)^{N+1} - (\epsilon - h/2)^{N+1}}, \]
and
\[ \tilde{x}_{N1} = X_{1N}/z_N = \frac{h \cdot (\epsilon + h/2)^{N-1}}{(\epsilon + h/2)^{N+1} - (\epsilon - h/2)^{N+1}}. \]

\[ \Box \]

Again, these results apply to the operator \( L \), and also to \( A_F \) if the coarse patch has trivial refinement. We note that the above argument is incorrect for one of the matrices, namely \( R \), when it corresponds to the centered stencil for the right-hand portion of the coarse component. What is given would be correct for this operator if it had a Dirichlet condition at the right boundary. For the Neumann condition, we should use
\[ X_{2x2} = \begin{bmatrix} 2\epsilon & -\epsilon + h/2 \\ -\epsilon - h/2 & \epsilon + h/2 \end{bmatrix}. \]

That is, technically, the initial values for the difference equation have to be modified, though we arrive at essentially the same result: now \( z_j = (\epsilon + h/2)^j \), so
\[ \tilde{r}_{11} = \frac{(\epsilon + h/2)^{N-1}}{(\epsilon + h/2)^N} = \frac{1}{\epsilon + h/2}. \]

For \( L_F \) we have the following corollary to Lemma 4.4.

**COROLLARY 4.5.** The corner elements of the fine patch matrix \( L_F \) corresponding to the centered discretization are given by
\[ \tilde{l}_{F_{Np},Np} = \frac{(\epsilon + h/2^{m+1})^{N_p} - (\epsilon - h/2^{m+1})^{N_p}}{2^{m+1}((\epsilon + h/2^{m+1})^{N_p+1} - (\epsilon - h/2^{m+1})^{N_p+1})} \approx \frac{1}{2^m \epsilon + h/2}, \]
(4.20)
\[ \tilde{I}_{F, N_F} = \frac{h \cdot (\epsilon - h/2m+1)^{N_F-1}}{2^{2m}(\epsilon + h/2m+1)^{N_F+1} - (\epsilon - h/2m+1)^{N_F+1}} \approx 0, \] (4.21)

and

\[ \tilde{I}_{F_{N_F,1}} = \frac{h \cdot (\epsilon + h/2m+1)^{N_F-1}}{2^{2m}(\epsilon + h/2m+1)^{N_F+1} - (\epsilon - h/2m+1)^{N_F+1}} \approx \frac{h}{(2m\epsilon + h/2)^2}. \] (4.22)

A similar result holds for \( A_P \) when the coarse patch has nontrivial refinement, again by performing the replacement \( m \leftarrow m - k \). Continuing, then, for the \( C^4 \) case we have \( \hat{L}_I - \lambda \hat{A}_I = \)

\[ \begin{pmatrix} (1 + 2^m)\epsilon - \lambda(1 + 2^{m-k})\epsilon & 0 \\ 0 & (1 + 2^m)\epsilon - \lambda(1 + 2m - k)\epsilon \end{pmatrix} \\
+ (\lambda - 1) \begin{pmatrix} \epsilon - h/2 & 0 \\ 0 & \epsilon - h/2 \end{pmatrix} \\
- \begin{pmatrix} 2^m\epsilon - h/2 & 0 \\ h & 2^m\epsilon - h/2 \end{pmatrix} \\
+ \lambda \begin{pmatrix} 2^{m-k}\epsilon - h/2 & 0 \\ h & 2^{m-k}\epsilon - h/2 \end{pmatrix}. \]

The solutions of \( \text{Det}(\hat{L}_I - \lambda \hat{A}_I) = 0 \) are again \( \lambda_1 = \lambda_2 = 1 \).

Next we turn to the \((UC)^2\) discretization, which uses upwinding on the coarse component and centered on the patch for both the coarse and true composite grids. Using approximations from the last two cases, we have \( \hat{L}_I - \lambda \hat{A}_I = \)

\[ \begin{pmatrix} (1 + 2^m)\epsilon + h - \lambda((1 + 2^{m-k})\epsilon + h) & 0 \\ 0 & (1 + 2^m)\epsilon + h - \lambda((1 + 2^{m-k})\epsilon + h) \end{pmatrix} \\
+ (\lambda - 1) \begin{pmatrix} \epsilon & 0 \\ 0 & \epsilon \end{pmatrix} - \begin{pmatrix} 2^m\epsilon \\ h(2^m\epsilon + h) \end{pmatrix} \begin{pmatrix} 0 \\ (2^m\epsilon - h/2)(2^m\epsilon + h) \end{pmatrix} \\
+ \lambda \begin{pmatrix} 2^{m-k}\epsilon \\ h(2^{m-k}\epsilon + h) \end{pmatrix} \begin{pmatrix} 0 \\ (2^{m-k}\epsilon - h/2)(2^{m-k}\epsilon + h) \end{pmatrix}. \]
The eigenvalues are $\lambda_1 = 1$ and

$$\lambda_2 = \frac{2^m \epsilon + h}{2^m \epsilon + h/2} \cdot \frac{2^{m-k} \epsilon + h/2}{2^{m-k} \epsilon + h} \in (1/2, 1) \text{ for } \epsilon \in [0, \infty).$$

We consider briefly the validity of the approximations used here. On the one hand, we have used (4.11-4.13) for the coarse component (upwind) operators, which are valid with $\epsilon$ small with respect to $h$. On the other hand, the (centered) approximations on the patch come from (4.20-4.22). Suppose that on the coarse composite grid no refinement is used on the patch ($m = 0$ and $N_P \leftarrow N_P$). Then the approximations will be valid with $\epsilon \approx h/2$ and also if $h$ continues to decrease. Notice that $\lambda_2 \to 1$ as $h \to 0$. Unfortunately, $h$ approaching zero with $\epsilon$ fixed is at odds with the requirement for the coarse component. There are two approaches that avoid this predicament. The first approach is to use sufficient refinement on the coarse patch so that $h$ need not be particularly small for (4.20-4.22) to be valid (because the exponent $N_P$ (or $N_F$) appearing there increases with $m$). Also, notice that $\lambda_2 \approx 1$ with $m$ large and $k$ small (i.e., with sufficient refinement on the fine and coarse patches). Unfortunately, taking this approach rules out the use of the global coarse grid in the role of the coarse composite grid. The second approach allows for this possibility by altering the discretization on the coarse patch in order to weaken the requirement that $h$ be small.

This leads us to consider the $U^3C$ discretization, where upwinding is used throughout the coarse composite grid and on the coarse component of the composite grid, but centered differencing is used on the fine patch. As just observed, the following approximations will be valid with $\epsilon$ sufficiently small with respect to $h$ (for the upwind approximation on the coarse component and on the coarse patch) and with sufficient refinement on the fine
patch (for the centered approximation). Although our motivation here is to be able to use a coarse patch without refinement \((k = m)\), we have carried out the analysis allowing arbitrary refinement on this region \((0 < k \leq m)\). The approximation is 

\[
\hat{L}_f - \lambda \hat{A}_f = 
\begin{bmatrix}
(1 + 2^m)\epsilon + h - \lambda((1 + 2^{m-k})\epsilon + h) & 0 \\
0 & (1 + 2^m)\epsilon + h - \lambda((1 + 2^{m-k})\epsilon + h)
\end{bmatrix}
\]

\[
+ (\lambda - 1) \begin{bmatrix}
\epsilon & 0 \\
0 & \epsilon
\end{bmatrix} - \begin{bmatrix}
\frac{2^m \epsilon}{h(2^m \epsilon + h)} & 0 \\
\frac{(2^m \epsilon - h/2)(2^m \epsilon + h)}{2^m \epsilon + h/2} & \frac{2^m \epsilon}{h(2^m \epsilon + h)}
\end{bmatrix}
\]

\[
+ \lambda \begin{bmatrix}
\frac{2^{m-k} \epsilon}{h} & 0 \\
0 & \frac{2^{m-k} \epsilon}{h}
\end{bmatrix}.
\]

The eigenvalues are \(\lambda_1 = 1\) and 

\[
\lambda_2 = \frac{2^m \epsilon + h}{2^m \epsilon + h/2}.
\]

Notice that the desired effect has been achieved in that now \(\lambda_2\) approaches unity while only requiring sufficient refinement on the fine patch. The expressions for the eigenvalues are independent of the refinement used on the coarse patch and so they will be valid, in particular, when the global coarse grid is used. We also remark that this requirement (of having sufficient refinement on the fine patch) is a natural one with respect to the use of the centered type of discretization for advection-dominated problems. Although the expression for \(\lambda_2\) is only valid in certain parameter ranges of the one-dimensional problem, we will see that the quantity \(\lambda_2 - 1\) supplies a fairly accurate approximation to the actual convergence factor even in two dimensions.

Collecting the results from our examination of the various cases considered above establishes the following theorem and corollary which apply to the one-dimensional problem.
THEOREM 4.6. Suppose that the exact values of the corner elements in equations (4.11-4.22) are replaced by there corresponding approximations. Then, in the $U^4$ case, the $C^4$ case, and in the limit as $m \to \infty$ in the $U^3C$ case we have the following.

Both eigenvalues of $I - \hat{L}_I \hat{A}_I^{-1}$ are zero. Therefore, two steps of the iteration

$$r^{i+1}_I = (I - \hat{L}_I \hat{A}_I^{-1})r^i_I$$

drives the residual at the interface to zero. This matrix is in fact strictly lower triangular: the "left" and "right" components of the interface residual vanish on the first and second steps, respectively, of the iteration. □

COROLLARY 4.7. Suppose the two-level FAC scheme is written in the form $r^{i+1} = (I - LM^{-1})r^i$ for some composite preconditioner $M$. Then, with the assumptions of Theorem 4.6, this iteration matrix is nilpotent: $(I - LM^{-1})^3 = 0$, i.e., the two-level scheme converges in three iterations and, thus, may be considered a direct method.

Proof. The corollary follows from the vanishing of $r^i_C$ and $r^i_P$ established in Theorem 4.1 for $i \geq 1$, the resulting expression (4.3) for the updating of the interface component of the residual, and Theorem 4.6. □

It should be noted that the results of the last theorem (and its corollary) are largely independent of the choice of coarse composite grid, i.e., they hold with $m - k$ levels of refinement on the coarse patch for all values of $k$, $0 \leq k \leq m$. In particular, this motivates the use of the the global coarse grid in this capacity.

4.2.2 Quality of Approximations

The purpose of this section is to assess the numerical quality of the approximations for elements of inverse matrices given in the previous sec-
tion. We will show that the values given as approximations are valid with regard to appropriate floating-point computations. However, their validity is subject to restrictions on the values of the parameters $\epsilon$, $h$, and $m$ (the order of patch refinement). It turns out that these restrictions allow for a large practical range of these parameters.

We begin with the $U^4$ method, concentrating on the inverses of $A_F$ and $L_F$. Here, the global coarse grid is used in the role of the coarse composite grid. Let $X$ denote either $A_F$ or $L_F$. In the last section, the corner elements of $X^{-1}$ were approximated as follows:

$$
\tilde{x}_{11} \approx \frac{1}{(2^m \epsilon + h)},
\tilde{x}_{NN} \approx \frac{1}{(2^m \epsilon + h)},
\tilde{x}_{1N} \approx 0,
$$

and

$$
\tilde{x}_{N1} \approx \frac{h}{(2^m \epsilon + h)^2},
$$

where $m = 0$ for $A_F$, and $m = m_r$ (the additional refinement levels on the fine patch) for $L_F$. In Tables 4.1-4.2 we give, for various values of $\epsilon$, the four pertinent values of the inverse matrices, i.e., the $2 \times 2$ matrix

$$
\begin{bmatrix}
\tilde{x}_{11} & \tilde{x}_{1N} \\
\tilde{x}_{N1} & \tilde{x}_{NN}
\end{bmatrix}.
$$

To obtain the elements of $X^{-1}$, the tridiagonal matrix was reduced to upper triangular form by Gaussian elimination (as in [45]), and backsolves were performed with the canonical basis vectors $e_1$ and $e_N$ as right-hand sides. The coarse grid meshwidth is $h = 1/64$, the coarse patch has 16 nodes, and there are $m_r = 4$ levels of additional refinement on the fine patch. Computations were done in single precision on a Sequent Balance. The approximations are valid for $\epsilon$ sufficiently small with respect to $h$, i.e., as $\epsilon \to 0$ with $h$ fixed. However, $\epsilon$ need not be particularly small, as seen
in Table 4.1, for the approximations to be fairly accurate. Furthermore, comparing the second table with the first, one also sees an improvement in the approximations with ε fixed and h refined. Although h decreases, the effect of this is offset by an increase in the exponent N (the order of the matrix being inverted) from \( N_P \) to \( N_F \).

**Table 4.1**

*Computed elements of coarse patch matrix inverse for the upwind case with \( h = 1/64 \) and 16 nodes on the coarse patch.*

<table>
<thead>
<tr>
<th>ε</th>
<th>( A_P^{-1} )</th>
<th>( \begin{bmatrix} 1/(\epsilon + h) \ h/(\epsilon + h)^2 \end{bmatrix} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>.0001</td>
<td>( 6.359301 \cdot 10^1 ) 1.757762 \cdot 10^{-27}</td>
<td>( 6.359301 \cdot 10^1 ) 6.318860 \cdot 10^1</td>
</tr>
<tr>
<td></td>
<td>6.318860 \cdot 10^1 6.359301 \cdot 10^1</td>
<td></td>
</tr>
<tr>
<td>.001</td>
<td>( 6.015038 \cdot 10^1 ) 7.627731 \cdot 10^{-15}</td>
<td>( 6.015038 \cdot 10^1 ) 5.653231 \cdot 10^1</td>
</tr>
<tr>
<td></td>
<td>5.653231 \cdot 10^1 6.015038 \cdot 10^1</td>
<td></td>
</tr>
<tr>
<td>.01</td>
<td>( 3.902435 \cdot 10^2 ) 1.158410 \cdot 10^{-4}</td>
<td>( 3.902439 \cdot 10^1 ) 2.379536 \cdot 10^1</td>
</tr>
<tr>
<td></td>
<td>2.379538 \cdot 10^2 3.902435 \cdot 10^1</td>
<td></td>
</tr>
</tbody>
</table>

Similar results are presented for \((UC)^2\). The approximations from the previous section are:

\[
\tilde{x}_{11} \approx 1/(2^m \epsilon + h/2), \\
\tilde{x}_{NN} \approx 1/(2^m \epsilon + h/2), \\
\tilde{x}_{1N} \approx 0,
\]

and

\[
\tilde{x}_{N1} \approx h/(2^m \epsilon + h/2)^2.
\]
Here $\epsilon = .001$ is fixed, the patch has length 1/4, and $h$ is allowed to increase. The expressions (4.17-4.22) predict the deterioration of the approximations as this happens.

Table 4.2

*Computed elements of fine patch matrix inverse for the upwind case with $h = 1/64$, 16 nodes on the coarse patch, and $m_c = 4$.*

<table>
<thead>
<tr>
<th>$\epsilon$</th>
<th>$L_F^{-1}$</th>
<th>$\left[ \frac{1}{(2^{m_c} \epsilon + h)} \right]_1 \bigg/ \left[ \frac{h}{(2^{m_c} \epsilon + h)^2} \right]_1$</th>
</tr>
</thead>
</table>
| .001       | 5.805515 $\cdot 10^1$ 0.000000 $\cdot 10^6$ 5.805515 $\cdot 10^1$ | 5.805516 $\cdot 10^1$
|            | 5.266088 $\cdot 10^1$ 5.805515 $\cdot 10^1$ | 5.266252 $\cdot 10^1$ |
| .001       | 3.162055 $\cdot 10^1$ 0.000000 $\cdot 10^6$ 3.162055 $\cdot 10^1$ | 3.162055 $\cdot 10^1$
|            | 1.562284 $\cdot 10^1$ 3.162055 $\cdot 10^1$ | 1.562280 $\cdot 10^1$ |
| .01        | 5.693950 $\cdot 10^0$ 1.184763 $\cdot 10^{-10}$ 5.693950 $\cdot 10^0$ | 5.693950 $\cdot 10^0$
|            | 5.065526 $\cdot 10^1$ 5.693947 $\cdot 10^0$ | 5.065792 $\cdot 10^{-1}$ |

Tables 4.3-4.4 below confirm this, with the deterioration being more significant for $A_p$. This again motivates the use of the upwind scheme on the coarse patch in $U^3C$.
Table 4.3
Computed elements of coarse patch matrix inverse for the centered case with $\epsilon = 10^{-3}$

<table>
<thead>
<tr>
<th>$h$</th>
<th>$A_F^{-1}$</th>
<th>$\begin{bmatrix} 1/(\epsilon + h/2) \ h/(\epsilon + h/2)^2 \end{bmatrix}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/512</td>
<td>$\begin{bmatrix} 5.059288 \cdot 10^2 \ 4.999298 \cdot 10^2 \end{bmatrix}$</td>
<td>$\begin{bmatrix} 5.059288 \cdot 10^2 \ 4.999298 \cdot 10^2 \end{bmatrix}$</td>
</tr>
<tr>
<td>1/256</td>
<td>$\begin{bmatrix} 3.386243 \cdot 10^2 \ 4.479160 \cdot 10^2 \end{bmatrix}$</td>
<td>$\begin{bmatrix} 3.386243 \cdot 10^2 \ 4.479158 \cdot 10^2 \end{bmatrix}$</td>
</tr>
<tr>
<td>1/128</td>
<td>$\begin{bmatrix} 2.041807 \cdot 10^2 \ 3.247696 \cdot 10^2 \end{bmatrix}$</td>
<td>$\begin{bmatrix} 2.038217 \cdot 10^2 \ 3.245568 \cdot 10^2 \end{bmatrix}$</td>
</tr>
</tbody>
</table>

Table 4.4
Computed elements of fine patch matrix inverse for the centered case with $\epsilon = 10^{-3}$ and $m_r = 1$.

<table>
<thead>
<tr>
<th>$h$</th>
<th>$L_F^{-1}$</th>
<th>$\begin{bmatrix} 1/(2\epsilon + h/2) \ h/(2\epsilon + h/2)^2 \end{bmatrix}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/512</td>
<td>$\begin{bmatrix} 3.359580 \cdot 10^2 \ 2.204448 \cdot 10^2 \end{bmatrix}$</td>
<td>$\begin{bmatrix} 3.359580 \cdot 10^2 \ 2.204449 \cdot 10^2 \end{bmatrix}$</td>
</tr>
<tr>
<td>1/256</td>
<td>$\begin{bmatrix} 2.529644 \cdot 10^2 \ 2.499649 \cdot 10^2 \end{bmatrix}$</td>
<td>$\begin{bmatrix} 2.592644 \cdot 10^2 \ 2.499648 \cdot 10^2 \end{bmatrix}$</td>
</tr>
<tr>
<td>1/128</td>
<td>$\begin{bmatrix} 1.693122 \cdot 10^2 \ 2.239580 \cdot 10^2 \end{bmatrix}$</td>
<td>$\begin{bmatrix} 1.693122 \cdot 10^2 \ 2.239579 \cdot 10^2 \end{bmatrix}$</td>
</tr>
</tbody>
</table>
4.2.3 Multiple Refinement Regions

Next we extend the analysis of Section 4.2.1 to the case of multiple interior refinement regions in one dimension. In general, there may be various refinement regions, but we suppose that, without loss of generality, there are but two; the significant difference between the case of one region and many is the existence of coarse regions between regions of refinement, not just at the two ends of the grid. Allowing only two refinements will also allow us to avoid significantly complicating our notation. With this convention, $L$ has the structure shown in Figure 4.2.

![Diagram](image)

**Fig. 4.2.** *Structure of a one-dimensional composite operator in the case of multiple refinement regions.*
The composite grid still has but a single fine component, although it is now the union of the various refinement regions. This is also true of the coarse and interface components. Accordingly, the composite operator still has the same structure, though its main nonzero blocks may now be further decomposed. For example,

\[
L_F = \begin{bmatrix}
L_{F_1} & \\ \\
L_{F_2}
\end{bmatrix}
\]

and

\[
L_I = \begin{bmatrix}
L_{I_1} & \\ \\
L_{I_2}
\end{bmatrix},
\]

where each \(L_{I_i}\) is a \(2 \times 2\) matrix associated with the \(i^{th}\) refinement region. Notice that a partitioning of \(L_I\) with respect to these blocks induces a similar partitioning of the matrices \(L_{IF}\) and \(L_{FI}\) into block diagonal matrices. Thus, for example, we may write

\[
L_{IF}L_F^{-1}L_{FI} = \begin{bmatrix}
L_{IF_1} & \\ \\
L_{IF_2}
\end{bmatrix} \begin{bmatrix}
L_{F_1} & \\ \\
L_{F_2}
\end{bmatrix}^{-1} \begin{bmatrix}
L_{FI_1} & \\ \\
L_{FI_2}
\end{bmatrix}.
\]

(The leading diagonal blocks here correspond to the blocks in the above figure with the "t" elements.) It follows that \(L_I - L_{IF}L_F^{-1}L_{FI}\) is a block diagonal matrix with diagonal blocks \(L_{t_i} - L_{IF_i}L_F^{-1}L_{FI_i}\). Furthermore, each one of these blocks has precisely the same form as \(L_I - L_{IF}L_F^{-1}L_{FI}\) had in the case of a single refinement region, although the dimensions may vary. Similar remarks hold for the global coarse operator and \(A_I - A_{IF}A_F^{-1}A_{FI}\).

As for the coarse-to-interface connections, let

\[
L_C = \begin{bmatrix}
L & \\ \\
M \\
R
\end{bmatrix}.
\]

Then \(L_{IC}L_C^{-1}L_{CI} =
\[
\begin{bmatrix}
L_{IC} & 0 & \tau_{IC} \tau_{CI} \tilde{m}_{11} & 0 \\
0 & \tau_{IC} \tau_{CI} \tilde{m}_{11} & 0 & 0 \\
0 & \tau_{IC} \tau_{CI} \tilde{m}_{21} & L_{IC} & 0 \\
0 & 0 & \tau_{IC} \tau_{CI} \tilde{m}_{31} & \tau_{IC} \tau_{CI} \tilde{m}_{11}
\end{bmatrix}
\]

86
It may be useful to the reader to compare this equation with (4.10). Recalling the analysis of the values of elements of matrix inverses, we see that the quantity $\tilde{m}_{1,N_{c_2}}$, which is the upper right corner element of $M^{-1}$, may be set to zero, allowing us to use instead $L_{IC}L_C^{-1}L_{CI} =$

\[
\begin{bmatrix}
    l_{IC}l_{CI}\tilde{l}_{N_{c_1},N_{c_1}} & 0 & 0 \\
    0 & r_{ICr_{CI}}\tilde{m}_{11} & 0 \\
    0 & 0 & l_{IC}l_{CI}\tilde{m}_{N_{c_2},N_{c_2}} \\
    0 & 0 & 0 & r_{ICr_{CI}}\tilde{r}_{11}
\end{bmatrix}
\]

With this simplification, the matrix $\hat{L}_I - \lambda \hat{A}_I$ is block lower triangular. Furthermore, each of its diagonal blocks has precisely the form that $\hat{L}_I - \lambda \hat{A}_I$ had in the case of a single refinement region. It follows that the analysis performed for that case may now be applied to each one of these blocks. The result is that, for each of the various cases considered previously, the same values may be given now to the eigenvalues of the matrix $I - \hat{L}_I\hat{A}_I^{-1}$, the only change being in their multiplicities according to the actual number of regions of refinement.

4.3 Convergence in Two Dimensions

In this section we study the convergence of the two-level version of FAC applied to the solution of the finite volume discretization of problem (3.2). Because of the complexity of the structure of the composite grid equations involving interior refinements, we have not yet extended the analysis of the previous section to two dimensions. Therefore, the examination here is based on computational evidence rather than on analysis. Guided by the results of the previous section, we attempt to determine to what extent the results for the one-dimensional problem carry over to two dimensions. Motivated by the findings of that section, we concentrate our study on the behavior of an algorithm that emphasizes the use of the global coarse
grid. The numerical method used here to solve the composite grid equations is the two-level FAC scheme with the global coarse grid as the coarse composite grid. Subproblems (corresponding to the fine patch and global coarse systems) are solved “exactly” (i.e., iteratively with a tolerance on the relative residual of $10^{-6}$). This has been done by performing, in the appropriate context, the necessary number of relaxations or multigrid V-cycles (for further details see Section 5.3.1).

The first result emphasizes the effect that the initial guess has on the convergence behavior. Consider the solution of the upwind/upwind discretization of Problem 3.1 of Section 3.3.1 by FAC using the upwind/upwind discretization on the global coarse grid (i.e., the $U^4$ technique). The problem parameters here correspond to those used to generate Tables 3.1-3.2. Table 4.5 presents the sequences of relative composite residuals corresponding to two solutions of the discrete problem. The two solves differ in the choice of the initial composite guess, $\mathbf{u}^0$. The zero vector and a random vector with elements between zero and one were used. The results appear, respectively, in the left and middle columns of the table. Recall that $\|\mathbf{r}_i\|_2$ here actually measures the current interface component of the residual, since $r_C^i, r_F^i = 0$ for $i \geq 1$ (see Section 4.1). Notice that there is a significant difference in the rates at which these interface residuals approach zero in the two solves. This difference is due to the convergence behavior of the residuals at the nodes that correspond to the tangential (north and south) boundaries of the interface. At the in- and out-flow (west and east, respectively) boundaries, the convergence behavior is like that predicted by the analysis of the one-dimensional problem – in both solves the residuals at the interface converge rapidly to zero. However, at the tangential
Table 4.5
Relative residuals as functions of FAC iterations using the U^4 method in two dimensions.

<table>
<thead>
<tr>
<th>i</th>
<th>z^0 = 0</th>
<th>z^0 Random</th>
<th>Modified Stencil z^0 Random</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.825 \cdot 10^{-4}</td>
<td>6.406 \cdot 10^{-2}</td>
<td>5.592 \cdot 10^{-2}</td>
</tr>
<tr>
<td>2</td>
<td>3.355 \cdot 10^{-6}</td>
<td>1.411 \cdot 10^{-2}</td>
<td>1.108 \cdot 10^{-3}</td>
</tr>
<tr>
<td>3</td>
<td>6.803 \cdot 10^{-7}</td>
<td>6.208 \cdot 10^{-3}</td>
<td>9.313 \cdot 10^{-5}</td>
</tr>
<tr>
<td>4</td>
<td>--</td>
<td>2.742 \cdot 10^{-3}</td>
<td>8.291 \cdot 10^{-6}</td>
</tr>
<tr>
<td>5</td>
<td>--</td>
<td>1.212 \cdot 10^{-3}</td>
<td>6.013 \cdot 10^{-7}</td>
</tr>
<tr>
<td>6</td>
<td>--</td>
<td>5.358 \cdot 10^{-4}</td>
<td>--</td>
</tr>
<tr>
<td>7</td>
<td>--</td>
<td>2.370 \cdot 10^{-4}</td>
<td>--</td>
</tr>
<tr>
<td>8</td>
<td>--</td>
<td>1.049 \cdot 10^{-4}</td>
<td>--</td>
</tr>
<tr>
<td>9</td>
<td>--</td>
<td>4.639 \cdot 10^{-5}</td>
<td>--</td>
</tr>
<tr>
<td>10</td>
<td>--</td>
<td>2.053 \cdot 10^{-5}</td>
<td>--</td>
</tr>
</tbody>
</table>

boundaries the convergence rate is generally much slower, as in the middle column of Table 4.5. The reason that this effect does not appear initially in the left column is that, with the initial guess of zero, very accurate solution values are produced at the tangential boundaries as a result of the first global coarse solve. Thus, the residuals at these nodes are extremely small at the end of the first FAC iteration and, therefore, throughout the ensuing iterations. We have developed a remedy for the problem of slow convergence at the tangential boundaries that involves modifying the definition of the global coarse stencil at these nodes. The right column of Table 4.5 gives the results for a solve using this modification. The actual changes to the stencil are obtained by requiring that interface-to-interface connections for the global coarse operator agree with those of the composite operator, i.e., A_I = L_I at nodes lying on the tangential boundaries. As a practical
matter, these changes do not adversely affect the efficiency of the algorithm's implementation. On the contrary, they can be beneficial, making the global coarse matrix more diagonally dominant, which facilitates the solution of these equations by an iterative method.

To motivate these modifications, consider the case where the only interface component is a tangential one, i.e., the two-dimensional composite grid associated with the domain partitioned into two horizontal strips, as depicted in Figure 4.3. Also, let $\epsilon \to 0$.

Fig. 4.3. Composite grid with coarse (o), fine (●), interface (◯) and slave (○) nodes.

Then, using the stencils of Section 3.2.3, one sees that only the diagonal blocks of the composite operator $L$ in (4.1) are nonzero. The same is true for $A$. The Schur complements in these respective operators then become $L_I$ and $A_I$, and the equation for the updating of the interface residual simplifies to

$$r^{i+1}_I = (I - L_I A^{-1}_I) r^i_I.$$  

This clarifies the reason for the modification. Notice that without the
modification, \( A_I = \frac{1}{s} L_I \), where \( s = \frac{1}{2} + \frac{1}{2^{m+1}} \). Thus,

\[
r_I^{i+1} = (1 - s) \cdot r_I^i,
\]
giving convergence factors between one quarter and one half. This agrees roughly with the observed convergence rates. We remark that as \( \epsilon \to 1 \) the modification becomes less beneficial, and even detrimental, but the way that this happens depends on both \( \epsilon \) and the amount of patch refinement.

Tables 4.6-4.7 show convergence rates as functions of \( \epsilon \) and fine patch refinement for the \( U^4 \) method applied to the upwind/upwind discretization of Problem 3.1. The coarse component mesh width is \( h_C = 1/32 \), and \( m \) gives the number of additional levels of refinement on the fine patch. We

<table>
<thead>
<tr>
<th>Upwind/Upwind</th>
<th>( m = 1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( |r^i|_2/|r^0|_2 )</td>
<td>( |r^i|_2/|r^0|_2 )</td>
</tr>
<tr>
<td>( i )</td>
<td>( \epsilon = 10^{-2} )</td>
</tr>
<tr>
<td>1</td>
<td>5.964 \cdot 10^{-2}</td>
</tr>
<tr>
<td>2</td>
<td>4.287 \cdot 10^{-3}</td>
</tr>
<tr>
<td>3</td>
<td>4.554 \cdot 10^{-4}</td>
</tr>
<tr>
<td>4</td>
<td>5.855 \cdot 10^{-5}</td>
</tr>
<tr>
<td>5</td>
<td>8.080 \cdot 10^{-6}</td>
</tr>
<tr>
<td>6</td>
<td>1.143 \cdot 10^{-6}</td>
</tr>
<tr>
<td>7</td>
<td>2.269 \cdot 10^{-7}</td>
</tr>
<tr>
<td>8</td>
<td>---</td>
</tr>
<tr>
<td>9</td>
<td>---</td>
</tr>
<tr>
<td>10</td>
<td>---</td>
</tr>
</tbody>
</table>
used a random initial guess throughout in these tests and, in some cases, the modification to the global coarse stencil mentioned above. We found that the advantage gained by using this modification is lost as $\epsilon$ increases, and use of the unmodified stencil can give better performance. For a given $\epsilon$, the results appearing in Tables 4.6-4.7 correspond to whichever

<table>
<thead>
<tr>
<th>$Upwind/Upwind \ m = 4$</th>
<th>$|\varepsilon^i|_2/|\varepsilon^0|_2$</th>
<th>$\epsilon = 10^{-2}$</th>
<th>$\epsilon = 10^{-3}$</th>
<th>$\epsilon = 10^{-4}$</th>
<th>$\epsilon = 10^{-5}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$i$</td>
<td>$Unmodified Stencil$</td>
<td>$Unmodified Stencil$</td>
<td>$Modified Stencil$</td>
<td>$Modified Stencil$</td>
<td>$Modified Stencil$</td>
</tr>
<tr>
<td>1</td>
<td>$1.691 \cdot 10^{-1}$</td>
<td>$7.516 \cdot 10^{-2}$</td>
<td>$5.592 \cdot 10^{-2}$</td>
<td>$5.738 \cdot 10^{-2}$</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>$1.232 \cdot 10^{-2}$</td>
<td>$1.506 \cdot 10^{-2}$</td>
<td>$1.108 \cdot 10^{-3}$</td>
<td>$4.108 \cdot 10^{-5}$</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>$1.135 \cdot 10^{-3}$</td>
<td>$3.523 \cdot 10^{-3}$</td>
<td>$9.313 \cdot 10^{-5}$</td>
<td>$4.659 \cdot 10^{-7}$</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>$1.192 \cdot 10^{-4}$</td>
<td>$1.171 \cdot 10^{-4}$</td>
<td>$8.291 \cdot 10^{-6}$</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>$1.259 \cdot 10^{-5}$</td>
<td>$4.012 \cdot 10^{-5}$</td>
<td>$6.013 \cdot 10^{-7}$</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>$1.321 \cdot 10^{-6}$</td>
<td>$1.384 \cdot 10^{-6}$</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>$2.078 \cdot 10^{-7}$</td>
<td>$4.786 \cdot 10^{-7}$</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>---</td>
<td>$1.657 \cdot 10^{-8}$</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>---</td>
<td>$5.769 \cdot 10^{-8}$</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>---</td>
<td>$2.014 \cdot 10^{-8}$</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.7

FAC convergence as a function of $\epsilon$
for the $U^4$ method.

strategy showed better performance. Thus, the leftmost column in Table 4.6 and the two leftmost columns of Table 4.7, correspond to the use of the unmodified stencil, while the other results are for the modified stencil.

Notice that the convergence rates generally improve as $\epsilon$ decreases, which is what we would expect as a result of the analysis of Section 4.2. The analysis also predicts independence of the rates on the level of patch refinement. The

92
results here show a mild degradation as the level of refinement is increased. One possible way to offset this effect (i.e., to make the need for refinement less critical) is to use a more accurate upwind scheme on the fine patch.

Tables 4.8-4.9 show results of a similar set of experiments with standard upwinding replaced by higher-order upwinding on the fine patch, i.e., the \((UH)^2\) technique (here, \(H\) denotes the use of the higher-order upwind discretization on a composite grid component) is used to solve the upwind/Hup composite grid problem. This FAC strategy was not analyzed

<table>
<thead>
<tr>
<th>(Upwind/Hup) (m = 1)</th>
<th>(|\mathbf{r}^i|_2/|\mathbf{r}^0|_2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(i)</td>
<td>Unmodified Stencil (\epsilon = 10^{-2})</td>
</tr>
<tr>
<td>1</td>
<td>6.016 \cdot 10^{-2}</td>
</tr>
<tr>
<td>2</td>
<td>4.376 \cdot 10^{-3}</td>
</tr>
<tr>
<td>3</td>
<td>4.271 \cdot 10^{-4}</td>
</tr>
<tr>
<td>4</td>
<td>5.126 \cdot 10^{-5}</td>
</tr>
<tr>
<td>5</td>
<td>6.654 \cdot 10^{-6}</td>
</tr>
<tr>
<td>6</td>
<td>9.051 \cdot 10^{-7}</td>
</tr>
<tr>
<td>7</td>
<td>---</td>
</tr>
<tr>
<td>8</td>
<td>---</td>
</tr>
<tr>
<td>9</td>
<td>---</td>
</tr>
<tr>
<td>10</td>
<td>---</td>
</tr>
</tbody>
</table>

in the previous section, but the experiments here indicate that its convergence behavior strongly resembles that of \(U^4\). This is evidenced by comparing Tables 4.8-4.9 with Tables 4.6-4.7, respectively (the same strat-
egy as above, of modifying the interface stencil, has been used here). When combined with a comparison of the accuracy of the two upwind discretizations (see Section 3.3.1), the results indicate that there is little advantage to using the upwind/upwind discretization. The more accurate upwind/Hup method can be used without sacrificing efficiency. In this study, we used strategies based on block Jacobi relaxation to solve the patch systems. With the upwind discretization, this means performing back-substitutions associated with the inversion of tridiagonal blocks at each relaxation step. With the more accurate method, the increase in work corresponds to using blocks having one additional diagonal. These relaxation-based strategies will be discussed more fully in Chapter 5.

Table 4.9
FAC convergence as a function of $e$
for the $(UH)^2$ method.

<table>
<thead>
<tr>
<th>$U_{\text{upwind}}$</th>
<th>$m = 4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$|E^i|_2/|E^0|_2$</td>
<td></td>
</tr>
<tr>
<td>$i$</td>
<td>$\text{Unmodified Stencil}$ $\epsilon = 10^{-2}$</td>
</tr>
<tr>
<td>1</td>
<td>$1.674 \cdot 10^{-1}$</td>
</tr>
<tr>
<td>2</td>
<td>$1.389 \cdot 10^{-2}$</td>
</tr>
<tr>
<td>3</td>
<td>$1.286 \cdot 10^{-3}$</td>
</tr>
<tr>
<td>4</td>
<td>$1.369 \cdot 10^{-4}$</td>
</tr>
<tr>
<td>5</td>
<td>$1.487 \cdot 10^{-5}$</td>
</tr>
<tr>
<td>6</td>
<td>$1.643 \cdot 10^{-6}$</td>
</tr>
<tr>
<td>7</td>
<td>$2.377 \cdot 10^{-7}$</td>
</tr>
<tr>
<td>8</td>
<td>$---$</td>
</tr>
<tr>
<td>9</td>
<td>$---$</td>
</tr>
<tr>
<td>10</td>
<td>$---$</td>
</tr>
</tbody>
</table>

Table 4.10 displays the results for the $C^4$ technique applied to the solu-

94
tion of the centered/centered discretization as a function of $\epsilon$. As expected, for this method, the convergence rates degrade with decreasing $\epsilon$. Generally, for discretizations that involve the centered stencil on the patch, no advantage was gained from the use of a modified stencil. Accordingly, the remainder of the results of this section pertain to the unmodified stencil.

Table 4.10

<table>
<thead>
<tr>
<th>Centered/Centered $m = 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$i$</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
</tbody>
</table>

We present results in Table 4.11 for the $U^3C$ technique applied to the upwind/centered discretization with various $\epsilon$ and $m$. The convergence rates improve as $\epsilon \cdot 2^m$ increases, as predicted by the analysis of the previous section. We note, however, a limit to the applicability of this result in that $\epsilon$ must not be allowed to become too large. If $\epsilon$ is allowed to increase with $h$ and $m$ fixed, a degradation of the convergence rates occurs. This fact is in accordance with the requirement, in the analysis of this method, that $\epsilon$ remain small enough to ensure the validity of the approximations used with respect to the upwind stencil on the coarse components.

Finally, we show that the predicted convergence factor for the residuals, namely $\lambda_2 - 1$ corresponding to the $U^3C$ method in Section 4.2.1 provides
a reasonably accurate (although somewhat pessimistic) approximation to the true rate. The results in Table 4.12 compare this quantity for the four columns of Table 4.11 with the corresponding average \((\|x^i\|/\|x^0\|)^{1/i}\). Here, the value of \(i\) corresponds to the final iterate in a given column of the table.

**Table 4.11**

*FAC convergence as a function of \(\epsilon\) and \(m\) for the \(U^3C\) method.*

<table>
<thead>
<tr>
<th>Upwind/Centered</th>
<th>(|x^i|/|x^0|_2)</th>
<th>(m = 1)</th>
<th>(m = 4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(i)</td>
<td>(\epsilon = 10^{-2})</td>
<td>(\epsilon = 10^{-3})</td>
<td>(\epsilon = 10^{-2})</td>
</tr>
<tr>
<td>1</td>
<td>6.790 \cdot 10^{-2}</td>
<td>1.149 \cdot 10^{-1}</td>
<td>1.784 \cdot 10^{-1}</td>
</tr>
<tr>
<td>2</td>
<td>1.243 \cdot 10^{-2}</td>
<td>8.888 \cdot 10^{-2}</td>
<td>1.577 \cdot 10^{-2}</td>
</tr>
<tr>
<td>3</td>
<td>4.134 \cdot 10^{-3}</td>
<td>7.106 \cdot 10^{-2}</td>
<td>1.512 \cdot 10^{-3}</td>
</tr>
<tr>
<td>4</td>
<td>1.515 \cdot 10^{-3}</td>
<td>5.875 \cdot 10^{-2}</td>
<td>1.627 \cdot 10^{-4}</td>
</tr>
<tr>
<td>5</td>
<td>5.744 \cdot 10^{-4}</td>
<td>4.978 \cdot 10^{-2}</td>
<td>1.782 \cdot 10^{-5}</td>
</tr>
<tr>
<td>6</td>
<td>2.223 \cdot 10^{-4}</td>
<td>4.298 \cdot 10^{-2}</td>
<td>2.008 \cdot 10^{-6}</td>
</tr>
<tr>
<td>7</td>
<td>8.735 \cdot 10^{-5}</td>
<td>3.759 \cdot 10^{-2}</td>
<td>2.787 \cdot 10^{-7}</td>
</tr>
<tr>
<td>8</td>
<td>3.470 \cdot 10^{-5}</td>
<td>3.318 \cdot 10^{-2}</td>
<td>--</td>
</tr>
</tbody>
</table>

**Table 4.12**

*Comparison of actual and predicted convergence factors for the \(U^3C\) method.*

<table>
<thead>
<tr>
<th>(\lambda_2 = \frac{2^m \epsilon + h}{2^m \epsilon + h/2})</th>
<th>(\epsilon = 10^{-2})</th>
<th>(\epsilon = 10^{-3})</th>
<th>(\epsilon = 10^{-2})</th>
<th>(\epsilon = 10^{-3})</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1 - \lambda_2)</td>
<td>2.77 \cdot 10^{-1}</td>
<td>6.53 \cdot 10^{-1}</td>
<td>1.16 \cdot 10^{-1}</td>
<td>3.50 \cdot 10^{-1}</td>
</tr>
<tr>
<td>((|x^i|_2/|x^0|_2)^{1/i})</td>
<td>4.39 \cdot 10^{-1}</td>
<td>8.87 \cdot 10^{-1}</td>
<td>8.90 \cdot 10^{-2}</td>
<td>4.94 \cdot 10^{-1}</td>
</tr>
</tbody>
</table>
5. Alternative FAC Algorithms

In this chapter we consider modifying the FAC algorithm that is described in Chapter 4 to obtain a more efficient technique. Multigrid plays a significant role here, and we assume a familiarity with the basic multigrid algorithms such as the V-cycle and the full multigrid scheme appearing, for example, in [25], [7], and [8].

Initially, in Section 5.1, we examine the implications of viewing FAC as a preconditioned iterative method. This provides an opportunity to compare the algorithm with domain decomposition methods in Section 5.2. In Section 5.3 we consider a truly multilevel approach to using FAC. The description of the method is given in Section 5.3.2. It is prefaced by a discussion and analysis in Section 5.3.1 of the role the block Jacobi method plays in this ultimate algorithm.

5.1 FAC as Preconditioning

We attempt to identify the matrix \( M \) for which the FAC iteration has the form

\[
\mathbf{r}^{i+1} = (I - LM^{-1})\mathbf{r}^i.
\]  

(5.1)

Thus, we are attempting to view FAC as an unweighted preconditioned Richardson's iteration. Equivalently, we may write \( \mathbf{z}^{i+1} = \mathbf{z}^i + \mathbf{z}' \), where \( \mathbf{z}' \) satisfies \( M\mathbf{z}' = \mathbf{r}^i \). Recall that the process by which FAC obtains the correction \( \mathbf{z}' \) may be viewed as the following two-step procedure. First, the interface component is solved for:

\[
\mathbf{z}_I = \hat{A}_I^{-1}\mathbf{r}_I^i.
\]
Then, the coarse and fine components are updated according to the original composite equations. This process is represented by the system of equations,

\[
\begin{bmatrix}
L_C & 0 & L_{CI} \\
0 & L_F & L_{FI} \\
0 & 0 & \hat{A}_I
\end{bmatrix}
\begin{bmatrix}
z_C \\
z_F \\
z_I
\end{bmatrix}
=
\begin{bmatrix}
r_C^i \\
r_F^i \\
r_I^i
\end{bmatrix},
\]  

(5.2)

where

\[
\hat{A}_I = (A_I - L_{IC}L_C^{-1}L_{CI} - A_{IP}A_F^{-1}A_{FI})
\]

and

\[
r_F^i = r_I^i - L_{IC}L_C^{-1}r_C^i - A_{IP}A_F^{-1}P_Fr_F^i.
\]

Using the fact that \(r_C^i, r_F^i = 0\), it can be immediately seen that \(M\) is the coefficient matrix in (5.2).

One approach to improving the convergence rate of FAC is to accelerate (5.1) by modifying it to form a polynomial iterative method such as a nonstationary Richardson’s iteration [39], a Chebychev iteration [31], or a Krylov subspace iteration [19]. Each step of a preconditioned version of each such iteration requires that, for a given vector \(r^i\), say, a preconditioning procedure can be applied to yield a vector \(z'\). This procedure is represented formally as

\[
z' = M^{-1}r^i.
\]

In the present context, this procedure is contained in Steps 1 and 2 of the FAC algorithm (see Section 4.1) where \(Az_G = r_G\) is solved, yielding components \(z_{G,C}\) and \(z_{G,I}\), then \(L_Fz_F^i = r_F^i - L_{FI}z_G^i\) is solved and \(z'\) is taken to be

\[
z' = (z_{G,C}, z_F^i, z_{G,I})^T.
\]
5.2 A Comparison of FAC and Domain Decomposition

It is the intent of this section to clarify the connection between FAC and the Schur complement variety of domain decomposition algorithms. Having FAC written in the form of a preconditioned iterative method will facilitate this discussion. The domain decomposition algorithms are similar in form to FAC, and in some cases may coincide with it [6], [21], [22], [44] so the insight into FAC convergence developed here may apply to a broader class of algorithms.

It is natural to think of the algorithm, and even the discretization strategy, presented in this study in terms of domain decomposition techniques, since the composite grid naturally induces such a decomposition. At the same time, there exists a strong distinction, in terms of motivation, between the methods considered here and what is normally considered domain decomposition. The latter often has the intent of partitioning the domain into many subdomains in order to achieve algorithms that are efficient with respect to parallel computer architectures, while the motivation behind the domain decomposition that occurs as a result of using a composite grid is better modeling of the solution of the partial differential equation. At the same time, a second type of domain decomposition, relating to parallelism, is available to the methods of this study. This is the natural decomposition of the domain induced by the advection-dominated problem written in the form

\[-\epsilon \Delta u(x,y) + u_x(x,y) = f(x,y),\]

and that corresponds to integrating the differential equation along its grid-aligned characteristics when \( \epsilon = 0 \). We note that it is not necessary that \( \epsilon \) be identically zero for this parallelism to be taken advantage of. In this
study, we have used the fact that \( \epsilon \) is small to good effect by employing line relaxation as a multigrid smoother.

To begin the comparison, recall the Schur reduced form of the system

\[
Lz = r:
\begin{bmatrix}
L_C & 0 & L_{CI} \\
0 & L_F & L_{FI} \\
0 & 0 & \hat{L}_I
\end{bmatrix}
\begin{bmatrix}
z_C \\
z_F \\
z_I
\end{bmatrix} =
\begin{bmatrix}
r_C \\
r_F \\
r_I
\end{bmatrix}.
\]

As noted in Section 4.1, according to this form, one approach to solving the original system is to solve exactly the system in the Schur complement, and then performing the back substitutions associated with the remaining equations to complete the solution. When this approach is used as the basis for a domain decomposition algorithm, emphasis is placed on solving efficiently the Schur complement system — typically, a Krylov subspace iteration is used, and much of the research in this area is devoted to identifying a good preconditioner for the Schur complement. Now, each step of such an iteration requires the application of the preconditioning as well as an application of the Schur complement,

\[
\hat{L}_I = L_I - L_{IC}L_C^{-1}L_{CI} - L_{IF}L_F^{-1}L_{FI},
\]

(5.3)

the parallelism of the domain decomposition methods deriving from the independence of the inversions required here.

One disadvantage of the approach just described is that the inversions required in (5.3) must be performed accurately just so that the coefficient matrix \( \hat{L}_I \) of the Schur complement system is represented accurately. As noted in [27] and [35], this disadvantage is especially significant when an iterative method is used to invert the operators \( L_C \) and \( L_F \) only approximately. A better strategy is to incorporate the preconditioning for the
Schur complement system along with the subregion operators into a composite preconditioner for the original equations. With this approach, the matrix inversions only enter into the application of the preconditioning. An appropriate form for the composite preconditioner is
\[
M = \begin{bmatrix}
L_G & 0 & L_{GI} \\
0 & L_F & L_{FI} \\
W_{IC} & W_{IF} & M_I
\end{bmatrix},
\] (5.4)

where \( M_I = \hat{P}_I + W_{IC}L_G^{-1}L_{GI} + W_{IF}L_F^{-1}L_{FI} \) and \( \hat{P}_I \) is a preconditioner for the Schur complement \( \hat{L}_I \). When \( M \) has this form, (irrespective of the particular choice of \( W_{IC} \) and \( W_{IF} \)), after the first iteration the coarse and fine patch components of the residual are zero and the interface components in (5.1) may be written as
\[
\mathbf{r}_I^{i+1} = (0, 0, (I - \hat{L}_I \hat{P}_I^{-1})\mathbf{r}_I^i)^T.
\]

Indeed, for \( k \geq 2 \),
\[
\mathbf{r}_I^k = (0, 0, (I - \hat{L}_I \hat{P}_I^{-1})^{k-1}\mathbf{r}_I^1)^T.
\]

Many standard domain decomposition algorithms are based on a preconditioning of this form [12], [27], as is FAC. In the latter, (5.4) holds with \( \hat{P}_I = \hat{A}_I \), the Schur complement in the global coarse operator \( A \). This is one of the distinguishing features of FAC: it preconditions the Schur complement in the original system with the Schur complement corresponding to a coarse grid discretization, and this preconditioning is accomplished implicitly by way of solving the coarse grid problem. The reason that this preconditioning appears implicitly in FAC is that the algorithm was born out of an attempt to economize multigrid when applied to composite grid problems, and so it is normally seen as a multigrid variation, rather than as a preconditioner.
It is worthwhile to say a few words about the economy that FAC achieves in solving this coarse grid problem. First, an effective preconditioner for the Schur complement is provided. At the same time, the coarse component boundary value problem that is implicit in $M$'s first block row is solved. Finally, in modifications of the algorithm that only approximately invert $L_F$, the patch component of the solution of the coarse grid problem provides an initial guess for the solution of the fine patch boundary problem ($M$'s second block row).

It is interesting to note that a similar algorithm (BEPS) exists explicitly as preconditioning in the work of Bramble et al. [6], [21]. That version of the algorithm is founded on Schur complement domain decomposition where it is customary to view the solution process in terms of a preconditioner. The particular domain decomposition method on which their algorithm is based is that of [5], which is similar to the method described in [4]. Both of these latter methods use a preconditioner in the form of a Schur complement; however, it is the Schur complement corresponding to a special discretization on one of the subregions. The preconditioning is implemented by solving a boundary value problem (with the interface as a portion of the boundary) on one of these regions. The modified technique of [21], however, has, after its first step, precisely the same form (compare [21, Section 3] with the previous section here) as an accelerated version (by use of a weighted Richardson's iteration) of the two-level FAC algorithm (see, also, the comparison of the two methods in [23]).

The difference in the two algorithms is the choice of a starting point. The BEPS algorithm begins by solving a fine patch boundary value problem with an initial guess of zero. The FAC scheme described herein begins by solving the global grid version of the problem and uses the resulting
interface values as data for the fine patch boundary value problem. Although the significance of this difference is minor, the second approach gives a useful conceptual view of the algorithm. In the context of the advection-dominated advection-diffusion equations the convention of starting on the coarse grid is quite natural. Suppose that line relaxation is used to solve the global coarse grid problem. Then, this FAC step corresponds to integrating across the domain along the characteristics of an (almost) hyperbolic equation. This result is then used as an initial guess for finding the solution on a subregion where the structure of the problem is more complex. In this sense, the FAC algorithm is close in spirit to domain decomposition as carried out in [9], [15], [37], where the domain is partitioned and the solution technique is chosen in accordance with the behavior of the solution as determined by an asymptotic analysis.

5.3 Modifications Based on Multigrid

The approach of viewing FAC as a preconditioned iterative method and then accelerating it by a polynomial method seems pragmatic enough (and corresponds to algorithms being used in practice). However, it has the potential drawback of requiring “many” FAC iterations to solve the discrete problem exactly, particularly as the effectiveness of the preconditioning deteriorates in certain parameter ranges of the discretized problem. Another drawback to using FAC as it has been described so far is that it requires the exact solution of coarse grid and fine patch subproblems. For these reasons, we have taken a fundamentally different approach to utilizing FAC efficiently. It differs from the ones described earlier in that it does not have as its ultimate goal the exact solution of the composite grid equations. Rather, its goal is to solve these equations only to the extent
that is dictated by the accuracy of the composite grid discretization.

The description of this method is facilitated by recalling the function space setting of Section 3.2.2. Let $c h^\alpha$ be a known bound (in some norm, $\| \cdot \|$) on the discretization error of the composite grid equations, with $c$ independent of $h$, and let $\hat{u}_h \in \hat{\Phi}_h$ be the approximation to $u$ obtained by solving exactly the composite grid equations. The algorithm to be described is based on a methodology that has as its goal the efficient computation of an approximate solution $\hat{u}_h^*$ satisfying

$$\| \hat{u}_h - \hat{u}_h^* \| \leq c h^\alpha.$$  \hspace{1cm} (5.5)

This algorithm turns out to be very satisfactory in that it produces a solution that is comparable in accuracy to the exact composite grid solution, while requiring few FAC iterations and minimal effort in solving subproblems. Based on the multilevel technique of nested iteration, it is a full multigrid-like algorithm that uses the two-level version of FAC as a weak algebraic solver on the sequence of composite grids lying between the global coarse grid and the true composite grid. We preface the more thorough description of this algorithm with a discussion of the role that relaxation plays in our implementation of the FAC algorithms used in this study.

5.3.1 The Significance of Line Relaxation

A practical approach to the implementation of FAC in its two-level form is to utilize some variation on the theme of relaxation to solve exactly the global coarse and fine patch boundary value problems that arise. We have used line relaxation itself to solve quickly, in certain cases, the global coarse systems when FAC was applied to the test problems of Sections 3.3.1 and 4.3. Generally, relaxation itself will not be sufficient because of the deterioration in its convergence rate on relatively fine grids after a few
iterations (see [8] and the analysis below). Convergence rates can also be slow on relatively coarse grids when $\epsilon$ is large. In both cases it is possible to increase greatly the efficiency of relaxation by incorporating it into a multigrid algorithm. For the tests of Sections 3.3.1 and 4.3, we solved fine patch problems and global coarse grid problems involving relatively large values of $\epsilon$ “exactly” by applying the necessary number of multigrid V-cycles required to reduce the size of the initial residual to below a small tolerance. This approach to the two-level algorithm is satisfactory in that a modest number of V-cycles are required per FAC iteration, and each V-cycle can be performed at a cost of just a few relaxations. In addition, the relaxation algorithms used are robust in that they apply in general to the various discretizations used.

The reason for the efficiency and robustness of these relaxation-based algorithms is that line relaxation is ideally suited to the advection-dominated problem in the form (2.2); when applied to the discrete problem, it provides an efficient solver in certain parameter ranges and an effective (multigrid) smoother in others.

Consider the nodes of either the global coarse grid or the fine patch ordered in lexicographic fashion, with the leading dimension corresponding to the $x$-direction. If the finite volume discretization is applied to problem (2.2) on this uniform grid, the result is a five- (or six-) point matrix stencil that has a block tridiagonal form, the main diagonal blocks themselves being tridiagonal matrices in the case of the upwind and centered discretizations (in the case of a higher-order upwind discretization these blocks have four nonzero main diagonals). By line relaxation associated with the system $Lz = r$ is meant the (possibly weighted) simple iteration based on the splitting $L = B - N$, where $B$ is the matrix that is equal to
$L$ on its diagonal blocks and zero elsewhere. This iteration may be written as

$$z^{i+1} = z^i + \omega B^{-1}r^i,$$

where

$$r^i = r - Lz^i.$$

The $i$th error, $e^i = z - z^i$, associated with the iteration is given by

$$e^i = (I - \omega B^{-1}L)^i e^0,$$

where $e^0$ is the error associated with an initial guess $z^0$. Notice that this may also be written as

$$e^i = p_i(B^{-1}L)e^0,$$

where the $i$th degree polynomial $p_i(z)$ is the product of $i$ linear factors, $(1-\omega z)$. Thus, the method may be viewed as an acceleration (by propitious choice of $\omega$) of the simple iteration for which $p_i(z)$ has linear factors, $(1-z)$.

Each choice of the polynomial $p_i$ may be viewed as a mapping, $p_i : \Sigma \rightarrow p_i(\Sigma)$, of the set of eigenvalues, $\Sigma$, of $B^{-1}L$ via $p_i(\Sigma) = \{p_i(\lambda) : \lambda \in \Sigma\}$. In the remainder of this section, we will examine the spectrum of $B^{-1}L$ under these mappings for the upwind and centered discretizations of problem (2.2).

We begin by generalizing problem (2.2) to

$$-\epsilon \Delta u(x,y) + \beta u_x(x,y) = f(x,y),$$

where $\beta \geq 0$ is a constant. This generalization is done purely as a matter of convenience in interpreting the results of the analysis. At times it is desirable to see the effect of letting $\epsilon$ become large, and this can be facilitated by setting $\epsilon = 1$ and letting $\beta \rightarrow 0$. We note that the stencils for uniform
grids given in Section 3.2.3 can be made to correspond to this new equation by performing the replacement \( h \leftarrow \beta h \). Regions containing the eigenvalues for centered differences applied to this equation are given in [14]. We are also interested in the upwind case and will require specific knowledge of individual eigenvalues and their associated eigenvectors. These eigenvectors can be found for both discretizations using the techniques of [20] and [29].

Assume, for convenience, a square mesh with spacing \( h = \frac{1}{N+1} \) in each direction (the following analysis being easily generalized to rectangular meshes). Also, denote the half-grid Reynolds number by \( \gamma = \frac{\theta h}{\pi} \). In what follows, \( \Sigma \) denotes the spectrum of \( B^{-1}L \), for a fixed \( \gamma \). At times the dependence of the spectrum on \( \gamma \) will be acknowledged by writing \( \Sigma(\gamma) \). A portion of the spectrum will frequently be located in a disk, that does not contain the origin, in the complex plane. We will be interested in how well the iteration damps eigenvectors of \( B^{-1}L \) that correspond to its eigenvalues in the region, and in the best way to choose \( \omega \) in defining \( p_i \).

For such an eigenpair, \( (\lambda, \nu) \), we have

\[
(I - \omega B^{-1}L)^i \nu = (1 - \omega \lambda)^i \nu,
\]

and we would like \( |1 - \omega \lambda| \) to be small. This quantity is bounded by \( \|p_i\|_\infty \) taken over the disk. It is well known [31], [43] that, for such a region, to minimize \( \|p_i\|_\infty \) over all polynomials of degree at most \( i \), one should choose the polynomial \( (1 - \omega z)^i \), where \( \omega \) is the reciprocal of the disk’s center. Then, \( \|p_i\|_\infty = |\omega| \cdot r \), where \( r \) is the radius of the disk. We will refer to this quantity, when it corresponds to the optimal choice of \( \omega \), as the spectral radius of the region and denote it by \( \rho_{opt} \). Also, we will have occasion to claim that the iteration is convergent, by which will be meant.
that it converges to the exact solution for all choices of $z^0$.

Denote the elements of $\Sigma$ by $\lambda_{kj}$, $1 \leq j, k \leq N$. The eigenvector of $B^{-1}L$ associated with $\lambda_{kj}$ is given by the tensor product $v_k \otimes v_j$ between the two vectors, $v_k$ with elements

$$v_{kl} = \Gamma^{l/2} \sin \frac{lk}{N + 1},$$

and $v_j$ with elements

$$v_{jl} = \sin \frac{lk}{N + 1},$$

where

$$\Gamma = \begin{cases} 1 + 2\gamma & \text{(upwind)}, \\ \frac{1 + \gamma}{1 - \gamma} & \text{(centered)}. \end{cases}$$

We will derive results pertaining to the set of eigenvectors as a whole, along with three oscillatory subsets: those associated with $\lambda_{kj}$ when $k \geq M$, when $j \geq M$, and when both $j, k \geq M$, where

$$M = \begin{cases} \frac{N + 1}{2} & \text{($N$ odd)}, \\ \frac{N^2 - 1}{2} & \text{($N$ even)}. \end{cases}$$

We begin our analysis with the upwind discretization on the patch. Recall that this discretization belongs to a problem with Dirichlet conditions at the fine boundary. Define the quantities $c_k = \cos \frac{k\pi}{N + 1}$ and $c_j = \cos \frac{j\pi}{N + 1}$ for $1 \leq k, j \leq N$. The elements of $\Sigma$ are found to be

$$\lambda_{kj} = 1 - \frac{c_j}{2 + \gamma - \sqrt{1 + 2\gamma c_k}}.$$

Let $f(\gamma) = 2 + \gamma - \sqrt{1 + 2\gamma}$. Then,

$$f'(\gamma) = 1 - \frac{1}{\sqrt{1 + 2\gamma}} > 0 \ \forall \gamma \in (0, \infty), \ f(0) = 1,$$

hence,

$$f(\gamma) \geq 1 > 0 \ \forall \gamma \in [0, \infty).$$

108
Now,
\[2 + \gamma - \sqrt{1 + 2\gamma c_k} > 2 + \gamma - \sqrt{1 + 2\gamma} \geq 1,\]
so
\[\left| \frac{c_j}{2 + \gamma - \sqrt{1 + 2\gamma c_k}} \right| \leq |c_j| < 1.\]
Restricting our attention to \(\gamma \in [0, \infty)\), we are now able to infer the following properties:

(u1) \(\Sigma(\gamma) \subset (0, 2) \forall \gamma\).

(u2) The iteration is convergent \(\forall \gamma \iff \omega \in [1, \infty)\).

(u3) \(\Sigma(\gamma) \subset \left(1 - \frac{1}{f(\gamma)}, 1 + \frac{1}{f(\gamma)}\right), f(0) = 1, \text{ and } f'(\gamma) > 0 \forall \gamma \in (0, \infty)\).

(u4) The iteration is convergent for a given \(\gamma \iff \omega \in \left[\frac{2}{1+1/f(\gamma)}, \infty\right)\).

(u5) \(1 < \lambda_{kj} \text{ when } j \geq M\).

(u6) \(1 - \frac{1}{2i\gamma} < \lambda_{kj} \text{ when } k \geq M\).

Notice that for all values of \(\gamma\), if we consider the entire spectrum, then \(\omega = 1\) is optimal in that there is no way to choose another \(\omega\) to accelerate the iteration. This is unfortunate as \(\gamma \rightarrow 0\) since then the spectral radius, \(\frac{1}{f(\gamma)}\), approaches unity. Therefore, it is necessary to begin to think of the iteration as a smoother for multigrid. Properties (u3), (u5), and (u6) imply that the optimal \(\omega\) for the subset of \(\Sigma\) corresponding to \(j\) or \(k \geq M\) is the reciprocal of the midpoint of the interval, \([1 - \frac{1}{2i\gamma}, 1 + \frac{1}{f(\gamma)}]\), i.e.,

\[\frac{1}{\omega_{opt}} = 1 + \frac{1/f(\gamma) - 1/(2 + \gamma)}{2}\]

The convergence factor associated with the set of eigenvalues for which \(j\) or \(k \geq M\), then is

\[\rho_{opt} = r \cdot \omega_{opt} = \frac{1/f(\gamma) + 1/(2 + \gamma)}{2 + 1/f(\gamma) - 1/(2 + \gamma)},\]
where \( r \) is the radius of the interval. Also, (u3) and (u5) imply that
the optimal \( \omega \) for the subset of \( \Sigma \) corresponding to \( j \) and \( k \geq M \) is the
reciprocal of the midpoint of the interval, \([1, 1 + \frac{1}{f(\gamma)}]\), i.e.,
\[
\frac{1}{\omega_{\text{opt}}} = 1 + \frac{1}{2f(\gamma)}.
\]
Moreover,
\[
\rho_{\text{opt}} = \frac{1}{1 + 2f(\gamma)}.
\]
Next we consider the centered discretization on the patch. Here it
becomes necessary to separate the problem into two cases. When \( \gamma \leq 1 \),
the eigenvalues of \( B^{-1}L \) are real; otherwise, they are generally complex.

Consider first the case that \( \gamma \leq 1 \). Then we have that
\[
\lambda_{kj} = 1 - \frac{c_j}{2 + (1 - \gamma^2)^{1/2}c_k}.
\]
Restricting \( \gamma \) to the interval \([0,1]\), define \( f(\gamma) = 2 - (1 - \gamma^2)^{1/2} \). Then, as
with the upwind case, we can establish the following properties:

\((c1)\) \( \Sigma(\gamma) \subset (0,2) \ \forall \gamma. \)

\((c2)\) The iteration is convergent \( \forall \gamma \iff \omega \in [1, \infty) \).

\((c3)\) \( \Sigma(\gamma) \subset (1 - \frac{1}{f(\gamma)}, 1 + \frac{1}{f(\gamma)}), f(0) = 1, \) and \( f'(\gamma) > 0 \ \forall \gamma \in (0,1) \).

\((c4)\) The iteration is convergent for a given \( \gamma \iff \omega \in \left[ \frac{2}{1+1/f(\gamma)}, \infty \right) \).

\((c5)\) \( 1 < \lambda_{kj} \) when \( j \geq M \).

Again, for all values of \( \gamma, \omega = 1 \) is optimal, and if we consider the operator \( I - B^{-1}L \) restricted to the eigenspace corresponding to \((c5)\), then the
analysis is the same as for the upwind discretization.
Now consider the case $\gamma > 1$. The eigenvalues are

$$\lambda_{kj} = 1 - \frac{c_j}{2 + i(\gamma^2 - 1)^{1/2}c_k},$$

where $i = \sqrt{-1}$. Suppose we rewrite these as

$$\lambda_{kj} = 1 - \frac{c_j(2 - i(\gamma^2 - 1)^{1/2})c_k}{4 + (\gamma^2 - 1)c_k^2}.$$

Then, we have

$$|1 - \lambda_{kj}| = \frac{|c_j|}{(4 + (\gamma^2 - 1)c_k^2)^{1/2}} < 1/2,$$

so $\Sigma$ is contained in the circle of radius $1/2$ centered at $z = 1$. If this is all we know about $\Sigma$, then $\omega = 1$ is again optimal and the iteration cannot be accelerated. On the other hand, we have the attractive result that the convergence factor associated with this region is $1/2$ and it applies for all values of $\gamma$. But more can be said. First of all, it is clear that $\lambda_{kj}$ with $j \geq M$ lies in the right half of this disk. Now suppose we rewrite the eigenvalues, once more, as $a + bi$:

$$1 - \lambda_{kj} = \frac{2c_j}{D_k} - \frac{(\gamma^2 - 1)^{1/2}c_jc_k}{D_k}i,$$

with $D_k = 4 + (\gamma^2 - 1)c_k^2$. We can now find a $c_0$ satisfying

$$(a - c_0)^2 + b^2 = c_0^2,$$

namely, $c_0 = c_j/4$. Therefore, $\Sigma$ is contained in the union of two disks with centers at $z_l = 3/4$ and $z_r = 5/4$ and both having radius $1/4$. The optimal $\omega$ for the right-hand disk is $\omega_{opt} = 4/5$ and the associated convergence factor is $\rho_{opt} = 1/5$. This provides a convergence factor for the eigenvectors associated with $\lambda_{kj}$, $k \geq M$, for all $\gamma \geq 1$.

When line relaxation is used as a smoother for multigrid, these properties may be used to provide guidance in developing an adaptive strategy.
for its appropriate weighting according to the relative ellipticity (i.e., \( \gamma \) as a function of meshwidth) of the discrete problem on each grid level. In order to provide satisfactory convergence rates the weighting should be chosen to emphasize the damping of oscillatory eigencomponents on fine grids and to perform the uniform damping of all eigencomponents on the coarser grids (in particular, it is desirable to solve exactly, and cheaply, the discretized problem on the coarsest level).

We use this criterion in a different way, to identify a robust weighting scheme; in other words, to identify a fixed \( \omega \) that yields a good general multigrid smoother for our problem. For this to be possible, such a parameter must provide a good smoother on arbitrarily fine grids. Some experimentation shows that the optimal \( \omega \) for the purely diffusive problem (i.e., with \( \epsilon = 1 \) and \( \beta = 0 \)) lies in the interval \([4/5, 2/3]\). Using a fixed \( \omega \) in this range yields V-cycle rates approximately equal to 1/3. On the other hand, the properties derived above show that \( \omega = 4/5 \) provides good global smoothing rates for both discretizations when \( \gamma \) is large. We use line relaxation with this latter weighting, exclusively, as the smoother for multigrid in the test problems of the next section. These test problems also involve the use of higher-order upwinding on the fine patch. Although we have not analyzed the performance of line relaxation with respect to this discretization, we find that in practice weighting strategies for standard upwinding also work well for the higher-order method.

In the last section, we remarked on the effectiveness of an implementation of FAC that uses relaxation-based strategies to solve global coarse grid and fine patch subproblems. With this algorithm as the starting point, it is natural to obtain modified algorithms by limiting the number of relaxations or V-cycles used in solving these subproblems. There are two reasons
for doing this. First, accurately solving the subproblems can be relatively expensive, particularly when there is much refinement on the patch. Second, performing these subproblem inversions accurately may be wasteful in terms of the efficiency of the two-level algorithm; good convergence rates for the algorithm as a whole can be obtained using less accurate inversions. This is particularly true when the two-level algorithm is implemented in a nested way on a sequence of composite grids (see the next section). Along with this change in strategy come some important considerations in implementing the algorithm. First, the iterative methods that are used on the subregions require an initial guess for the solution. For the fine patch problem, this approximation can be obtained from the result of performing Step 1 (see Section 4.1) of the FAC algorithm, the global coarse solve; we use as the initial guess in performing the fine patch solve the coarse patch component of the solution of the global coarse problem interpolated to the fine patch,

\[ z_0 = I_F^G z_{G,C}. \]

(This is part of the definition of FAC given in [33] which we have suppressed heretofore due to the assumption that the fine patch problem is solved exactly.) In addition, we use zero as the initial guess for the global coarse problems.

For this algorithm, we use \( X_F \) to represent an approximation to \( L_F^{-1} \), i.e., we write the result of the fine patch solve as

\[ z_F = X_F(r_F - L_F z_I). \]

Whether we can actually write \( z_F \) this way is a technical matter that depends on the method of approximate inversion and the fact that this
method will require an initial approximation to the solution of

\[ L_F p = (r_F - L_{Fk} z_I). \]

In any case, the notation may be considered a formal way of denoting the process of replacing the true \( z_F \) with its approximation.

### 5.3.2 FAC and Nested Iteration

In this section, we combine nested iteration with the two-level FAC scheme to obtain a full multigrid version of FAC. Our goal is to come up with an efficient algorithm that yields an approximate solution satisfying (5.5). The full multigrid approach is a well-known methodology [7], [8], [25], [33] for achieving the latter with optimal efficiency. It uses a weak (the use of this term will be explained shortly) algebraic solver in a nested fashion on a sequence of successively finer grids as follows. On a coarse grid indexed by \( 2h \), let \( z_{2h} \) denote the exact solution to the discrete equations, and let \( z^0_{2h} \) denote an approximation to this solution. Now, let \( z^n_{2h} \) be the ultimate approximation on this grid obtained by applying the algebraic solver with \( z^0_{2h} \) as an initial guess. This serves as the initial guess on the next finer grid: \( z^n_h = J^h_{2h} z^n_{2h} \), and the process is repeated on finer grids.

At this point, it is useful to assume the existence of an approximation subspace as in the element methods, so that the natural isomorphism between vectors, \( z_h \), and functions, \( \hat{u}_h \) may be utilized. By the notion of a weak algebraic solver we mean that the intent of its use is not to solve the problem on the \( h \)-grid exactly, but rather to solve it to the extent that, for the given norm,

\[ \| \hat{u}_h - \hat{u}_h^* \| \leq \kappa \cdot \| \hat{u}_h - \hat{u}_h^0 \| \]

is satisfied, for some moderately large constant \( \kappa \) that is independent of \( h \). To see how small \( \kappa \) should be, the following inequality (taken with
slight modification from [33]) can be used. Recall from Section 5.3 the assumption that \( \| u - \hat{u}_h \| \leq c h^\alpha \). Also, notice that, in terms of elements of \( \hat{\Phi}_h \) (with interpolation as defined in [30]), we have \( \hat{u}_h^0 = \hat{u}_{2h}^* \). In addition, assume that the desired inequality is true for \( 2h \):

\[
\| \hat{u}_{2h} - \hat{u}_{2h}^* \| \leq c(2h)^\alpha.
\]

Then,

\[
\| \hat{u}_h - \hat{u}_h^0 \| = \| \hat{u}_h - \hat{u}_{2h}^* \| \\
\leq \| \hat{u}_h - \hat{u}_{2h} \| + \| \hat{u}_{2h} - \hat{u}_{2h}^* \| \\
\leq \| \hat{u}_h - u \| + \| \hat{u}_{2h} - u \| + \| \hat{u}_{2h} - \hat{u}_{2h}^* \| \\
\leq c h^\alpha + c(2h)^\alpha + c(2h)^\alpha \\
= c(1 + 2^{\alpha+1}) h^\alpha.
\]

Thus, a sufficient condition for the desired \( h \)-inequality, (5.5) to hold is

\[
\kappa \leq \frac{1}{1 + 2^{\alpha+1}}.
\]

We note that although this condition is sufficient, in practice it may be pessimistic. Notice, also, that \( \kappa \) depends by way of \( \alpha \) on the particular choice of norm and discretization type. Later in this section, we present numerical results that indicate the actual number of FAC iterations required to yield approximate solutions with appropriate accuracy as measured in the maximum and \( L^2 \) norms.

The following description of nested iteration makes use of the succession of composite grids leading from the global coarse grid to the composite grid on which a solution is desired, each composite grid being obtained as in Section 3.1 from the global coarse grid by refinement on the coarse patch.

Let \( K \) be a positive integer and, for \( k = 1, \ldots, K \), let \( G_k \) denote the composite grid obtained from the global coarse grid by adding \( k \) levels of refinement to the coarse patch, each level corresponding to refinement in meshwidth by a factor of two. Let \( G_0 \) stand for the global coarse grid.
(if \( h = 1/2^m \) is the meshwidth of this grid, then the \( k \)th fine patch has meshwidth \( h_k = h/2^k \)).

The following algorithm uses the FAC iteration, specifically, a two-level algorithm for \( G_k \) using \( G_0 \) as the coarse grid and incorporating approximate inversions for the global coarse and fine patch subproblems, on each of the composite grids in \( \{ G_k \}_{k=1}^K \). The particular algorithms are distinguished by letting \( X_{F_k}^{-1} \) denote the approximate fine patch inversion used on level \( k \). We use \( X \) to denote approximate inversion of \( A \) on the global coarse grid. The algorithm begins with the exact solution of a representation of the partial differential equation on the global coarse grid. The global coarse grid also plays its usual role in the implementation of the FAC iterations on each of the composite grids.

Let \( \{ \mu_k \}_{k=1}^K \) be positive integers and \( L_k \) be the composite operator for the \( k \)th composite grid. Also, let composite grid transfer operators, \( I_{k-1}^k : G_{k-1} \to G_k \) and \( I_k^{k-1} : G_k \to G_{k-1} \) be linear mappings.

**Nested Iterative FAC**

**Step 1.** Map the finest composite right-hand side, \( r_K \) successively to all coarser grids: \( r_{k-1} = I_{k-1}^k r_k \), \( k = K, K - 1, \ldots, 1 \).

**Step 2.** Solve \( L_0 z_0^* = r_0 \) exactly using an appropriate solution technique for the global coarse equations.

**Step 3.** For \( k = 1, \ldots, K \): set \( z_k^0 = I_{k-1}^k z_{k-1}^* \). Approximate the solution of \( L_k z_k = r_k \) starting with initial guess \( z_k^0 \) and using \( \mu_k \) FAC iterations with \( X_{F_k} \) in place of \( L_{F_k}^{-1} \) (approximate inversion on the patch) and \( X \) in place of \( A^{-1} \) (approximate inversion on the global coarse grid); denote the final approximation by \( z_k^* \).
In the remainder of this section, some examples of the performance of a particular implementation of the nested iteration algorithm are presented. For these examples, the test problems of Section 3.3.1 are used. Each set of tests here corresponds to a set of tests from that section, so that discrete errors for nested iteration ($e_h^* = u - \hat{u}_h^*$) may be compared with those obtained by exact solution of the composite equations ($e_h = u - \hat{u}_h$). For convenience, the pertinent results from Section 3.3.1 are reproduced here. Our implementation of nested iteration uses block-Jacobi based iterations (either V-cycles of the (1,1) variety as described in [7], or just relaxations themselves). Multigrid V-cycles are used to solve fine patch problems approximately whereas, depending on the parameter ranges and type of discretization, V-cycles or simply relaxations are used to solve approximately the global coarse problems. For each test, $\mu_k$, the number of FAC iterations used on the $k$th composite grid is kept constant, so the index has been suppressed. The same is true for both $\nu_{F_k}$, the number of V-cycles used to solve approximately the fine patch problem during each FAC iteration, and $\nu_{C_k}$, the number of relaxations or V-cycles used during each global coarse solve. In all instances, relaxation used as the basis for multigrid V-cycles is damped by the parameter $\alpha = 4/5$ (motivated in Section 5.3.1).

Table 5.1 shows relative errors as functions of patch refinement and the number of FAC iterations used on each grid level. The test uses Problem 3.1 with $\epsilon = 10^{-4}$ and $h = 1/32$. As in Tables 3.1-3.2, upwind discretization is used on the coarse component of the composite grid. The FAC method is of type $U^4$ or $U^3H$ depending on whether standard or higher-order upwinding, respectively, is used on the fine patch. Here, $\nu_F = \nu_C = 1$, and relaxation is used for the global coarse problem.
Table 5.1
Relative errors as functions of patch refinement when nested iteration is used.

<table>
<thead>
<tr>
<th>$m$</th>
<th>$U^4$ Method $\mu = 1$</th>
<th>$U^3 C$ Method $\mu = 2$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$|e_h^*|<em>\infty / |u|</em>\infty$</td>
<td>$|e_h^*|<em>\infty / |u|</em>\infty$</td>
</tr>
<tr>
<td>1</td>
<td>$1.402 \cdot 10^{-1}$</td>
<td>$5.546 \cdot 10^{-2}$</td>
</tr>
<tr>
<td>2</td>
<td>$6.687 \cdot 10^{-2}$</td>
<td>$5.477 \cdot 10^{-3}$</td>
</tr>
<tr>
<td>3</td>
<td>$3.221 \cdot 10^{-2}$</td>
<td>$2.731 \cdot 10^{-3}$</td>
</tr>
<tr>
<td>4</td>
<td>$1.574 \cdot 10^{-2}$</td>
<td>$1.367 \cdot 10^{-3}$</td>
</tr>
<tr>
<td>5</td>
<td>$7.758 \cdot 10^{-3}$</td>
<td>$6.847 \cdot 10^{-4}$</td>
</tr>
</tbody>
</table>

Table 5.2
Relative errors as functions of patch refinement when exact solution is used.

<table>
<thead>
<tr>
<th>$m$</th>
<th>Upwind/Upwind</th>
<th>Upwind/Hup</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$|e_h|<em>\infty / |u|</em>\infty$</td>
<td>$|e_h|<em>\infty / |u|</em>\infty$</td>
</tr>
<tr>
<td>1</td>
<td>$1.383 \cdot 10^{-1}$</td>
<td>$5.551 \cdot 10^{-2}$</td>
</tr>
<tr>
<td>2</td>
<td>$6.569 \cdot 10^{-2}$</td>
<td>$5.359 \cdot 10^{-3}$</td>
</tr>
<tr>
<td>3</td>
<td>$3.194 \cdot 10^{-2}$</td>
<td>$2.687 \cdot 10^{-3}$</td>
</tr>
<tr>
<td>4</td>
<td>$1.576 \cdot 10^{-2}$</td>
<td>$1.346 \cdot 10^{-3}$</td>
</tr>
<tr>
<td>5</td>
<td>$7.328 \cdot 10^{-3}$</td>
<td>$6.735 \cdot 10^{-4}$</td>
</tr>
</tbody>
</table>

The results in Table 5.1 should be compared with those in Table 5.2 (the latter table reproduces results in Tables 3.1-3.2 of Section 3.3.1). The comparison shows close agreement between the exact solutions and those obtained using the nested iteration version of FAC. It is interesting to note that more work is required (two FAC iterations, as opposed to one) to obtain comparable solutions when the higher-order discretization is used. This behavior is in accordance with the requirements posed by the earlier analysis. Table 5.3 shows results for a similar set of tests applied to Problem 3.2 and using the upwind/centered discretization. The coarse meshwidth
is \( h = 1/32, \epsilon = 10^{-3} \), and the FAC method is of type \( U^3C \). Results for different numbers of V-cycles on the fine patch are compared. The other full multigrid variables are \( \nu_C = \mu = 1 \). Again, \( \lambda \) corresponds to relaxation.

<table>
<thead>
<tr>
<th>( m )</th>
<th>( |e_h^\star|/|u|_\infty )</th>
<th>( |e_h^\star|_2/|u|_2 )</th>
<th>( |e_h^\star|/|u|_\infty )</th>
<th>( |e_h^\star|_2/|u|_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9.475 \cdot 10^{-2}</td>
<td>1.176 \cdot 10^{-1}</td>
<td>1.103 \cdot 10^{-1}</td>
<td>9.252 \cdot 10^{-2}</td>
</tr>
<tr>
<td>2</td>
<td>2.923 \cdot 10^{-2}</td>
<td>3.122 \cdot 10^{-2}</td>
<td>2.365 \cdot 10^{-2}</td>
<td>2.013 \cdot 10^{-2}</td>
</tr>
<tr>
<td>3</td>
<td>8.247 \cdot 10^{-3}</td>
<td>9.650 \cdot 10^{-3}</td>
<td>6.111 \cdot 10^{-3}</td>
<td>5.197 \cdot 10^{-3}</td>
</tr>
<tr>
<td>4</td>
<td>2.669 \cdot 10^{-3}</td>
<td>2.372 \cdot 10^{-3}</td>
<td>1.695 \cdot 10^{-3}</td>
<td>1.322 \cdot 10^{-3}</td>
</tr>
<tr>
<td>5</td>
<td>6.940 \cdot 10^{-4}</td>
<td>6.993 \cdot 10^{-4}</td>
<td>4.370 \cdot 10^{-4}</td>
<td>3.528 \cdot 10^{-4}</td>
</tr>
</tbody>
</table>

Table 5.3
Relative errors as functions of patch refinement when nested iteration is used.

<table>
<thead>
<tr>
<th>( m )</th>
<th>( |e_h^\star|/|u|_\infty )</th>
<th>( |e_h^\star|_2/|u|_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.051 \cdot 10^{-1}</td>
<td>8.449 \cdot 10^{-2}</td>
</tr>
<tr>
<td>2</td>
<td>2.253 \cdot 10^{-2}</td>
<td>1.858 \cdot 10^{-2}</td>
</tr>
<tr>
<td>3</td>
<td>5.460 \cdot 10^{-3}</td>
<td>4.519 \cdot 10^{-3}</td>
</tr>
<tr>
<td>4</td>
<td>1.356 \cdot 10^{-3}</td>
<td>1.122 \cdot 10^{-3}</td>
</tr>
<tr>
<td>5</td>
<td>3.383 \cdot 10^{-4}</td>
<td>2.800 \cdot 10^{-4}</td>
</tr>
</tbody>
</table>

Table 5.4
Relative errors as functions of patch refinement when exact solution is used.

The results in Table 5.3 should be compared with those in Table 5.4 (the latter are the upwind/centered results from Tables 3.3-3.4). Here, one sees a significant degradation of the convergence of \( \dot{u}_h^\star \) in the left column of Table 5.3. In the right column, \( O(h^2) \) convergence is restored by simply increasing the number of V-cycles used on the fine patch. Table 5.5 presents
results corresponding to those in Table 3.5 (which are reproduced below in Table 5.6). Two FAC iterations ($\mu = 2$) using a single fine patch V-cycle ($\nu_F = 1$) were required to obtain full multigrid solutions with the desired accuracy. In addition, unlike the previous problems, V-cycles were used for the global coarse solves. Recall that for this test, $u$ has a rapidly changing component that is not confined to the patch. This helps to explain the additional computational effort required on the global coarse grid. Also, note that additional work (two V-cycles, rather than one) is required on this grid when the change to a more accurate coarse component discretization is made.

<table>
<thead>
<tr>
<th>$m_c$ ($m = 2$)</th>
<th>$U^3C$ Method $\nu_C = 1$</th>
<th>$C^4$ Method $\nu_C = 2$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$|e_h|<em>\infty / |u|</em>\infty$</td>
<td>$|e_h^*|<em>\infty / |u|</em>\infty$</td>
</tr>
<tr>
<td>3</td>
<td>$2.026 \cdot 10^{-1}$</td>
<td>$1.836 \cdot 10^{-1}$</td>
</tr>
<tr>
<td>4</td>
<td>$7.970 \cdot 10^{-2}$</td>
<td>$7.960 \cdot 10^{-2}$</td>
</tr>
<tr>
<td>5</td>
<td>$3.625 \cdot 10^{-2}$</td>
<td>$2.154 \cdot 10^{-2}$</td>
</tr>
<tr>
<td>6</td>
<td>$1.860 \cdot 10^{-2}$</td>
<td>$7.081 \cdot 10^{-3}$</td>
</tr>
</tbody>
</table>

Table 5.5

Relative errors as functions of coarse component refinement when nested iteration is used.

<table>
<thead>
<tr>
<th>$m_c$ ($m = 2$)</th>
<th>$U^3C$ Method $\nu_C = 1$</th>
<th>$C^4$ Method $\nu_C = 2$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$|e_h|<em>\infty / |u|</em>\infty$</td>
<td>$|e_h|<em>\infty / |u|</em>\infty$</td>
</tr>
<tr>
<td>3</td>
<td>$1.963 \cdot 10^{-1}$</td>
<td>$1.935 \cdot 10^{-1}$</td>
</tr>
<tr>
<td>4</td>
<td>$7.981 \cdot 10^{-2}$</td>
<td>$7.468 \cdot 10^{-2}$</td>
</tr>
<tr>
<td>5</td>
<td>$3.921 \cdot 10^{-2}$</td>
<td>$1.903 \cdot 10^{-2}$</td>
</tr>
<tr>
<td>6</td>
<td>$2.152 \cdot 10^{-2}$</td>
<td>$5.628 \cdot 10^{-3}$</td>
</tr>
</tbody>
</table>

Table 5.6

Relative errors as functions of coarse component refinement when exact solution is used.

120
We note that, in all of the preceding tests, the unmodified stencils of Section 3.3.3 were used. Recall that in Section 4.3, degradation in convergence of the two-level algorithm was attributed to use of the unmodified stencil on the global coarse grid. However, the convergence results of this section are quite satisfactory. The reason for this is that when the two-level algorithm performed poorly, it did so at the tangential boundaries of the patch. Here, these interface values are being solved for very accurately in the initial phase of the algorithm, when the problem on the global coarse grid is solved. A very accurate initial guess is provided to the fine levels by this coarse grid and, therefore, the dependence on the initial guess noted in the earlier section no longer exists.

In the final test case of this section, we present a problem for which the use of refinement near one of the corners of the domain $\Omega$ is appropriate. This problem has been chosen so that the solution simulates a boundary layer typical of singularly perturbed problems. Indeed, the solution has been adapted from the well-known one-dimensional problem

\[
\begin{align*}
\epsilon u''(x) + u'(x) &= 0, \quad x \in [0, 1], \\
u(0) &= 1, \\
u(1) &= 0,
\end{align*}
\] (5.7)

whose solution [18] has an $O(\epsilon)$-width boundary layer near the left endpoint of the interval. Our two-dimensional solution corresponds to that of (5.7) with the boundary layer translated to, and rotated about the point (1,1) at the northeast corner of $\Omega$.

Problem 5.1.

Let $D_{(1,1)}(x,y) = \sqrt{(x - 1)^2 + (y - 1)^2}$ and $r > 0$. The solution is

\[
u(x, y) = e^{-D_{(1,1)}(x,y)/(\epsilon r)}.
\]
Appropriate solution values are used to define Dirichlet conditions as before on the south, west, and north segments of $\partial \Omega$. As for the east boundary, it is partitioned into northeast and southeast segments where appropriate Dirichlet and Neumann conditions, respectively, are specified (see Figure 5.1). Notice that the Neumann condition is given by $u_x(1,y) = 0$ on $\partial \Omega_{SE}$.

\[ \begin{array}{c}
\partial \Omega_N \\
\partial \Omega_W \\
\partial \Omega_S \\
\partial \Omega_{NE} \\
\partial \Omega_{SE} 
\end{array} \]

Fig. 5.1. Partitioning of $\partial \Omega$ for Problem 5.1.

For this problem we use a composite grid as shown in Figure 5.2. Note that the interface has only two sides. Along $\partial \Omega_{NE}$, for example, the interface is replaced by a fine boundary whose nodes are members of the composite grid. For convenience in testing, we place the division between northeast and southeast segments of $\partial \Omega_E$ at the southeast corner of the interface. Also, we use a coarse component meshwidth of $h = 1/32$, and the patch is square with sides having length $1/8$. 

122
The results of tests for this problem with $\epsilon = .001$ and $r = 10$ are shown in Table 5.7, where solutions by nested iteration are compared with exact ones. The nested iteration variables that yield these results are $\mu = 2$, $\nu_C = 1$, and $\nu_F = 2$, with one exception. For the upwind/centered case, with $m = 5$, in order to keep the order of convergence from degenerating, we find that it is necessary to use four fine patch V-cycles per FAC iteration on the finest composite grid.

Fig. 5.2. Composite grid with coarse (○), fine (●), interface (◯), and slave (○) nodes.
Table 5.7
Relative errors as functions of patch refinement for nested iteration (left) and exact solves (right) in the discrete version of Problem 5.1.

| m  | \( U^4 \) Method \( ||e_h^*||_\infty/||u||_\infty \) | Exact Solve \( ||e_h||_\infty/||u||_\infty \) | \( U^3C \) Method \( ||e_h^*||_\infty/||u||_\infty \) | Exact Solve \( ||e_h||_\infty/||u||_\infty \) |
|----|---------------------------------|-------------------|------------------------------|-------------------|
| 2  | 4.894 \( \cdot 10^{-2} \)    | 4.900 \( \cdot 10^{-2} \) | 3.072 \( \cdot 10^{-2} \)   | 3.070 \( \cdot 10^{-2} \) |
| 3  | 3.507 \( \cdot 10^{-2} \)    | 3.491 \( \cdot 10^{-2} \) | 1.746 \( \cdot 10^{-2} \)   | 1.782 \( \cdot 10^{-2} \) |
| 4  | 1.941 \( \cdot 10^{-2} \)    | 1.954 \( \cdot 10^{-2} \) | 4.158 \( \cdot 10^{-3} \)   | 5.551 \( \cdot 10^{-3} \) |
| 5  | 1.032 \( \cdot 10^{-2} \)    | 1.035 \( \cdot 10^{-2} \) | 1.246 \( \cdot 10^{-3} \)   | 1.293 \( \cdot 10^{-3} \) |
6. Summary of Results

In this thesis, we have studied multilevel methods for the solution of advection-dominated elliptic equations in one and two dimensions. We concentrated on solving problems for which local grid refinement is desirable, in particular, problems for which the use of a very coarse grid is suitable, except in a small subregion of the domain, where significant refinement is needed.

Although the results obtained here for the one-dimensional case are generally of little practical interest in themselves, they provide important insight into the behavior of the methods we study in higher dimensions. This is where our true interests lie. In this chapter, we summarize some of the more important results obtained in this study. We emphasize those results that have a bearing on practical use of the solution methods presented here, and in certain cases give recommendations for their application in the implementation of these methods.

In two dimensions, the results of this study were obtained for the advection-diffusion equation

\[-\epsilon \Delta u + u_x + cu = f.\]  \hfill (6.1)

This equation has a particularly attractive form in that its flow velocity lies entirely in the \(x\)-coordinate direction. In Chapter 2, we noted that this equation can be obtained from a rather general advection-diffusion equation in two dimension via an orthogonal coordinate transformation induced by characteristics. Algorithms for performing this change of coordinates are described in [9] and [13].
The significance of the above form for the results obtained herein are two-fold, and correspond to two hierarchies of solution procedure. At the global level, we considered the FAC multigrid method for the solution of (6.1) discretized on a composite grid. This method belongs to a class of algorithms which place the computational effort in solving, or ultimately, partially solving, standard problems on subregions of the composite grid. Our convergence results for this method relied, to a large extent, on the strong one-dimensional character of (6.1) when $\epsilon$ is small, in the sense that when this is true a rigorous analysis in one dimension gives much useful insight into the true behavior of FAC convergence in two dimensions.

In Section 4.2.1 we analyzed the eigenvalues of what is essentially the iteration matrix governing convergence of two-level FAC when exact solution of subgrid problems is assumed. For a given discretization on the composite grid we also considered the discretization type used at the coarse level of the two-level scheme. Four discretizations of the one-dimensional problem were considered. The first two cases (the $U^4$ and $C^4$ methods) involved the use of either upwind or centered discretization throughout both grids. In suitable parameter ranges, we concluded that the spectral radius of the iteration matrix, in these cases, was zero. This allowed us to further conclude that in this context the two-level FAC iteration may be viewed as a direct solver, producing a zero residual after three iterations. Furthermore, this result was relatively insensitive to the choice of the composite grid used on the coarse level. In particular, the global coarse grid could be used.

The third case we considered, $(UC)^2$ used upwind and centered difference discretizations, respectively, on the coarse component and the patch component of the two grids. The spectral radius of the iteration matrix
in this case was bounded by one half, but the validity of the result depended on a familiar requirement for the centered discretization, namely that the meshwidth \( h \) be sufficiently small with respect to the diffusion coefficient. This was in conflict with the requirement for the upwind part of the discretization and motivated the use of the fourth strategy, \( U^3C \), which employed upwinding throughout the coarse grid and used the upwind/centered mix on the composite grid. For this case, the spectral radius approached zero as the refinement level on the fine patch increased.

As for convergence in two dimensions, we saw in Section 4.3 that computational results there conformed, for the most part, to the analysis in one dimension. An exception we discovered was a significant degradation in convergence rates for upwinding used on the fine patch. This problem was more pronounced when the diffusion coefficient was small. We found that this was caused by slow convergence of the residuals at the tangential boundaries of the patch (i.e., the ones parallel to the flow direction). By analyzing the matrix stencil associated with an idealized version of the problem (the case that the interface component is all tangential boundary), we identified a modification of the stencil in these regions that restores the convergence rates. The modification had essentially the same effect when higher-order upwinding was used on the fine patch. In fact, results obtained for standard upwinding in this study generally apply as well to the case of higher-order upwinding. These include global convergence behavior of two-level FAC as well as weighting of line relaxation in devising multigrid smoothers for subgrid problems (see Section 5.3.1). An instance where results for the two discretizations differ occurred in the nested iteration algorithm of Chapter 5. There it was seen that more work was required by the higher-order method on each of the intermediate gridlevels in order to
obtain a solution with the desired accuracy (see Table 5.1). This fact is in accordance with the theory for full multigrid as discussed in Section 5.3.2 – increasing the order of accuracy of the discretization increases the estimate of work required by the algebraic solver.

These results of this section indicate that the most attractive methods for strongly to moderately advection-dominated problems are the $U^3H$ and $U^3C$ methods. In two dimensions, we noted that the former method may be performed in parallel with a modest increase in expense over the $u^4$ method, while at the same time retaining its convergence behavior. For more mildly advection-dominated problems, the $U^3C$ and $C^4$ methods may be more appropriate.

We note that if the two-level schemes $U^4$ and $U^3H$ of Chapter 4 are used as iterative solvers (i.e., viewed as a preconditioned iterations and accelerated as described in Chapter 5 by a polynomial method), and $\epsilon$ is small, it is essential that the modification to the coarse grid stencil be performed. Just how small $\epsilon$ must be for this to be necessary depends on the meshwidth of both the coarse and fine components of the composite grid. In our experiments, with the coarse meshwidth fixed this point occurred later (with respect to decreasing $\epsilon$) as the number of levels of fine patch refinement increased (see Tables 4.6-4.7).

However, we do not advocate such use. The reason for this is that the two-level scheme, as presented in Chapter 4, requires exact solution of the (global coarse and fine patch) subgrid problems. Furthermore, as noted in [23], with respect to diffusion problems, the number of iterations required to solve the composite equations grows rapidly when a few steps of a multigrid method are used to approximately solve the fine patch subproblems. Therefore, the computational complexity of the two-level
scheme is generally far from optimal. Yet, there is a way of combining the scheme with efficient approximation of subgrid solutions which does have the (fine grid) complexity of full multigrid. The approach we took in Section 5.3 is the classical multigrid approach of using the scheme in a nested way on a sequence of successively finer grids. In particular, we used the two-level scheme as a weak algebraic solver on each of the composite grids lying between the global coarse grid and the true composite grid, and used inexpensive subproblem approximations obtained via either relaxation or multigrid V-cycles. Based on the computed results of Section 5.3.2, we recommend the following strategies for the implementation of this method. Suppose that standard upwinding is used on the coarse component. Then, one or two unweighted line relaxations should suffice in "solving" the global coarse grid problems. This depends on the relative coarseness of the mesh, but the quantities $1/f(\gamma)$ (the asymptotic convergence factor) derived in Section 5.3.1 and $\kappa$ (the required convergence factor) derived in Section 5.3.2 may be used for guidance. We note that when the Reynold's number $\gamma$ is greater than 4.2 two steps of the line Jacobi iteration yield an asymptotic convergence factor bounded by $1/9$. Convergence factors for the use of V-cycles on the fine patch are more difficult to predict. However, in Section 5.3.1 we noted that for a worse case situation (i.e., for the diffusion problem, which resembles the case of infinite mesh refinement) the use of the relaxation parameter $\alpha = 4/5$ yields V-cycle factors close to $1/3$. So, again, two iterates yield a factor of $1/9$. Using these strategies, we found that, generally, only one or two FAC iterations per gridlevel were required to produce approximate solutions with the accuracy of the exact solution of the discrete equations. Notice, with a single FAC iteration on each level, this means that the total work on the fine grid is less than that of four V-
cycles performed on the finest patch. Hence, this method has a fine patch complexity comparable to full multigrid.

We note that the above results are specific to the finite volume discretization we used for the problem (6.1). We were motivated to use this method by the fact that line relaxation then provides a highly effective solution method for uniform grid problems while decoupling the nodes in such a way that preserves the parallel nature (6.1) possesses when $\epsilon = 0$.

We close this discussion with two observations that might be used to yield some improvements in our implementation of the nested iteration algorithm.

First, we used a fixed relaxation parameter $\alpha$ for the damping of relaxation when used as a multigrid smoother on the fine patch. The analysis of Section 5.3.1 may be used, however, to fine tune the weighting of relaxation according to gridlevels in the V-cycle. Specifically, we provided convergence factors associated with global and oscillatory eigenspaces for general gridlevels. These may be used to determine when to unweight relaxation as the grids coarsen (when the upwind discretizations are used), and what weighting to use to most effectively damp oscillatory error components on fine grids.

Second, when the centered discretization was used on the patch, we employed the $U^3C$ method on each of the composite gridlevels. However, our experiments with the two-level version of this method showed that it can be slow if the fine level is too coarse. This observation motivates the use of upwinding on the coarser composite gridlevels and applying the $U^4$ or $U^3H$ method, and then switching to centered discretization and the $U^3C$ method on the finer composite grids. It may be possible to apply the results of our analysis of this method to determine when it is appropriate to
switch discretizations. We saw (in Table 4.12) that the quantity $1 - \lambda_2$ ($\lambda_2 = \frac{2^{m+1} + h}{2^{m+1} + h/2}$), associated with the latter method in one dimension, gave good estimates of actual convergence rates for the method in two dimensions. Although this was true with the use of exact inversion of subgrid problems, the estimates were generally somewhat pessimistic, and so may also apply when a reasonably accurate approximation method, such as classical full multigrid is used to solve these problems. Since the centered method has second order accuracy, an appropriate strategy would be to change to the centered method on the level $m$ fine patch when $1 - \lambda_2$ is less than $1/9$ or $1/3$, respectively, as one or two FAC iterations are used.
REFERENCES


[38] U. Rüde, Fully adaptive multigrid methods, manuscript, presented at the 5th Copper Mountain Conference on Multigrid Methods, Copper Mountain, CO, April, 1991.


