Research Article

The Interaction of Iteration Error and Stability for Linear Partial Differential Equations Coupled through an Interface

B. Sheehan,[1] D. Estep,[2] S. Tavener,[1] J. Cary,[3] S. Kruger,[3] A. Hakim,[3] A. Pletzer,[3] J. Carlsson,[3] and S. Vadlamani[3]

[1] *Department of Mathematics, Colorado State University, Fort Collins, CO 80523, USA*
[2] *Department of Statistics, Colorado State University, Fort Collins, CO 80523, USA*
[3] *Tech X Corporation, Boulder, CO 80303, USA*

Correspondence should be addressed to D. Estep; estep@stat.colostate.edu

We investigate properties of algorithms that are used to solve coupled evolutionary partial differential equations posed on neighboring, nonoverlapping domains, where the solutions are coupled by continuity of state and normal flux through a shared boundary. The algorithms considered are based on the widely used approach of iteratively exchanging boundary condition data on the shared boundary at each time step. There exists a significant and sophisticated numerical analysis of such methods. However, computations for practical applications are often carried out under conditions under which it is unclear if rigorous results apply while relatively few iterations are used per time step. To examine this situation, we derive exact matrix expressions for the propagation of the error due to incomplete iteration that can be readily evaluated for specific discretization parameters. Using the formulas, we show that the universal validity of several tenants of the practitioner's conventional wisdom are not universally valid.

1. Introduction

The class of multiphysics problems in which one physical process operates in one domain while a second physical process operates in a neighboring domain, and the solutions in the component subdomains are coupled by continuity of state and normal flux through a common boundary, which is central to a number of applications [1]. Examples include conjugate heat transfer between a fluid and solid [2–5], Stokes-Darcy flow in karst aquifers [6] and in catalytic converters [7], modeling of the core and edge plasma flows in a tokamak reactor [8, 9], and flow in heterogeneous porous media [10–13].

In applications, the processes in the component subdomains are often themselves represented by complex, multiscale, and multiphysics models and the component models are solved with different discretization methods utilizing significantly different scales. Such coupled problems present enormous challenges for efficient high performance computing [1]. There are very strong incentives to use existing "legacy" codes for the component models and to treat component physics solvers as "black boxes." For these reasons the most commonly encountered solution strategy is a simple iterative approach [1–4, 14–17]. In this approach, the models in each of the component subdomains are associated with one of the two boundary coupling conditions and subsequently solved independently except for data from the coupling through the boundary. The coupling data for the boundary conditions for one component model are provided by the solution of the other component model from the previous iteration. In a time dependent problem, this iteration is carried out on each time step.

This type of coupled physics problem has been extensively studied by the numerical analysis community, with the goals of deriving accurate, stable numerical methods, efficient, and accurate coupling algorithms and deriving rigorous a priori analyses, for example, in [10–15, 17, 18], as well as a posteriori analyses [2–4, 19]. However, in practice, as mentioned, the component physics is usually complex and there are large differences in the discretization strategies employed in

the component subdomains necessitating significant processing of transferred information. Thus, the application of current rigorous mathematical methods and analysis is problematic if not impossible. This is partially the reason that there is a very signicant gulf between the kinds of practices carried out in high performance computational science and the "best practices" determined by the mathematical community [3].

In addition, the computational strategies employed in practice are often rationalized by a body of conventional wisdom which asserts that (1) stability is equivalent to accuracy and (2) the use of unconditionally stable implicit solvers for the component physics generally stabilizes the entire coupled simulation as long as the component solutions are stable. This is often rationalized by some sort of (Neumann) linear stability analysis applied to the simplied case of two coupled linear parabolic problems. Based on these ideas, simulations are deemed acceptable provided they do not exhibit obvious instabilities such as unbounded behavior or rapid oscillation. Moreover, due to the prohibitive computational cost, such conclusions are often based on the single computation at hand, rather than through an exhaustive study.

This paper attempts to address the difficulties that face extending rigorous convergence analysis to the complex models and discretizations that occur in practice and illustrate issues arising from the conventional wisdom. Instead of focusing on deriving conditions that guarantee good behavior as is usual for a standard convergence analysis, we adopt a different approach and develop a rigorous computable error representation that can be evaluated for any given choice of discretization parameters, thus allowing the conventional wisdom to be verified or not. We consider the canonical problem of two linear coupled parabolic equations and formulate the iterative solution method as a fixed point iteration for a block system on the discrete level. We then derive an exact formula for the error in the iterated solution. This formula is related to the Neumann series for the inverse of the full system matrix. The argument has several virtues: it is elementary, it subsumes any sort of ad hoc linear stability analysis, and it is general in the sense that it holds for a variety of discretization methods and range of scales for the component physics. Importantly, it allows for easy evaluation of various discretization parameter choices, for example, step size, space mesh, and number of iterations per time step.

We then present a detailed study of the canonical problem of two linear coupled parabolic equations that amply demonstrates shortcomings of the conventional wisdom. Firstly, we are able to dispel the notion that stability implies accuracy. In particular, we demonstrate that a divergent iteration can be part of a stable algorithm if the number of iterations used is low, while the resulting approximation is inaccurate. In this case, the user might not be aware that the iteration is divergent if they only consider whether or not obvious instability occurs. This case is particularly interesting since a serious problem is being masked by seemingly "reasonable" results. Secondly, we demonstrate that there is no guarantee that an "unconditionally stable" time integration scheme like backward Euler remains unconditionally stable if the system is not solved exactly at each time step. We explore cases where using a limited number of iterations leads to a conditional
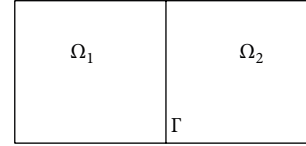

Figure 1: Illustration of neighboring subdomains.

stability, which depends on the size of the time step. The values of time steps which provide stability can occupy a range with a maximum and minimum value, rather than just a maximum.

The remainder of this paper is organized as follows. Section 2 introduces the problem and the notation associated with discretization and iteration. Section 3 derives the primary results of the paper regarding the stability of the iterative solution. Section 4 provides numerical examples. Section 5 addresses the multirate case, in which the time step in one component is an integer multiple of the time step in the other component. Multirate numerical examples are included. A brief conclusion is given in Section 6.

2. Problem Formulation and Discretization

We consider a system posed on a domain consisting of two neighboring, nonoverlapping, convex, and polygonal subdomains Ω_1 and Ω_2 in 1, 2, or 3 spatial dimensions that share a common polygonal interface boundary Γ (see [1, 2, 8, 9] for applications). We illustrate in Figure 1.

The general form of the PDE system is

$$
\begin{aligned}
\frac{\partial}{\partial t}\mathcal{U}_1 + \mathcal{L}_1\mathcal{U}_1 &= \mathcal{S}_1 \quad \text{on } \Omega_1 \times (0, T], \\
\frac{\partial}{\partial t}\mathcal{U}_2 + \mathcal{L}_2\mathcal{U}_2 &= \mathcal{S}_2 \quad \text{on } \Omega_2 \times (0, T], \\
\mathcal{U}_1 &= \mathcal{U}_2 \quad \text{on } \Gamma \times (0, T], \\
\hat{n} \cdot a_1 \nabla \mathcal{U}_1 &= \hat{n} \cdot a_2 \nabla \mathcal{U}_2 \quad \text{on } \Gamma \times (0, T],
\end{aligned}
\tag{1}
$$

together with the following additional boundary and initial conditions:

(i) boundary conditions for $\mathcal{U}_1$ on $\partial\Omega_1 \setminus \Gamma$ and initial conditions for $\mathcal{U}_1$ on Ω_1,

(ii) boundary conditions for $\mathcal{U}_2$ on $\partial\Omega_2 \setminus \Gamma$ and initial conditions for $\mathcal{U}_2$ on Ω_2,

where $\mathcal{L}_1 v = \nabla \cdot (a_1 \nabla v)$ and $\mathcal{L}_2 v = \nabla \cdot (a_2 \nabla v)$ are linear, coercive elliptic operators, $\mathcal{S}_1$, $\mathcal{S}_2$ are given functions in L^2, and $\hat{n}$ is a unit vector normal to Γ and pointing from Ω_1 to Ω_2.

We next introduce notation for the discretized problem. The analysis is based on the backward Euler method in time but can accommodate various spatial discretization schemes [3]. For the numerical results included later in the paper, we use a finite volume method in space [13, 18, 19], and the details of that discretization are given in Appendix A. In order to complete the boundary data for each subdomain,

we adopt the common strategy of using the solution on the first subdomain to determine Dirichlet boundary data on Γ for the problem on the second subdomain and, likewise, use the solution on the second subdomain to provide Neumann boundary conditions for the problem on the first subdomain; for example, see [2–4, 8–12, 14, 15, 17, 19]. Thus, each problem is provided with a full set of boundary data, but the subdomains are not treated symmetrically. By convention, we assume that component 2 provides state data for component 1, and component 1 provides flux data for component 2. This leads to the system of discrete equations:

$$L_1 \widehat{u}_1^{k+1} = \widehat{u}_1^k + S_1^{k+1} + P_{21} \widehat{u}_2^{k+1},$$
$$L_2 \widehat{u}_2^{k+1} = \widehat{u}_2^k + S_2^{k+1} + P_{12} \widehat{u}_1^{k+1}, \tag{2}$$

where the superscript is the time step index. The symbols L_1 and L_2 are the discrete versions of the operators $(\partial/\partial t + \mathscr{L}_1)$ and $(\partial/\partial t + \mathscr{L}_2)$, except that $\widehat{u}^k$ has been moved to the right hand side. The symbols S_1 and S_2 are the discrete versions of the source terms $\mathscr{S}_1$ and $\mathscr{S}_2$ and also take into account the known boundary data on $\partial\Omega_1 \setminus \Gamma$ and $\partial\Omega_2 \setminus \Gamma$. P_{21} and P_{12} are the projections by which the solution in one subdomain is used to provide boundary data on Γ for the other subdomain. The subscripts "12" and "21" indicate information passing from "Ω_1 to Ω_2" and "Ω_2 to Ω_1." We refer to $\widehat{u}_1$ and $\widehat{u}_2$, which solve (2), as the implicitly coupled solution. We emphasize that we analyze this method because it is one of the most common approaches used in practice.

Note that the form of the operators P_{21} and P_{12} depends on the choice of discretization. Some discretizations, for example, finite volume, require forming these operators using some combination of averaging, extrapolation, and interpolation [1]. Other discretizations provide natural definitions. For example, in the case of the mortar method [18, 19], the coupling between the subdomains has a formal structure based on Lagrange multiplier variables on the interface. In this case P_{21} and P_{12} correspond to the off-diagonal blocks of the Schur complement [20] that are formed by eliminating the Lagrange multiplier variables.

In all of the above cases, we now write (2) as a single set of linear equations. Namely,

$$A\widehat{u}^{k+1} = \widehat{u}^k + S^{k+1}, \tag{3}$$

where

$$A = \begin{bmatrix} L_1 & -P_{21} \\ -P_{12} & L_2 \end{bmatrix}, \qquad S = \begin{bmatrix} S_1 \\ S_2 \end{bmatrix}, \qquad \widehat{u} = \begin{bmatrix} \widehat{u}_1 \\ \widehat{u}_2 \end{bmatrix}. \tag{4}$$

2.1. Block Iteration. As mentioned, the implicitly coupled system (3)-(4) is not formed or solved exactly in practice. The common approach is to use an iterative block solution approach that involves a sequence of solutions of each component problem with alternating exchange of coupling boundary data; for example, see [2–4, 8–12, 14, 15, 17, 19]. This is easily described using the concept of matrix splitting [20–22]. We define a matrix splitting $A = M - N$ so that

$$M\widehat{u}^{k+1} = \widehat{u}^k + S^{k+1} + N\widehat{u}^{k+1}. \tag{5}$$

Starting with some initial guess u^0, the following equation defines a fixed point iteration which may converge to $\widehat{u}^{k+1}$. For the iterates, we use a double superscript, k for the time step and i for the iteration index. We use n for the number of iterations to be performed at each time step:

$$Mu^{k+1,i} = u^{k,n} + S^{k+1} + Nu^{k+1,i-1},$$
$$i = 1,\ldots,n, \quad u^{k+1,0} = u^{k,n}. \tag{6}$$

Note $Nu^{k+1,i-1}$ represents the exchange of information between the components, and the subsequent inversion of M is the solution of the individual components. In the case of (3), the splitting $A = M - N$ is chosen to separate the solution of the problems on the subdomains [20–22]. This is motivated by the fact that the coupling of the subdomains occurs on a manifold of lower dimension. This splitting can be accomplished in a "Gauss-Jacobi" sense:

$$M = \begin{bmatrix} L_1 & 0 \\ 0 & L_2 \end{bmatrix}, \qquad N = \begin{bmatrix} 0 & P_{21} \\ P_{12} & 0 \end{bmatrix}, \tag{7}$$

or in a "Gauss-Seidel" sense:

$$M = \begin{bmatrix} L_1 & 0 \\ -P_{12} & L_2 \end{bmatrix}, \qquad N = \begin{bmatrix} 0 & P_{21} \\ 0 & 0 \end{bmatrix}. \tag{8}$$

If this iteration converges, it converges to the implicitly coupled solution $\widehat{u}$, which is the unique fixed point of the iteration. By differencing (5) and (6), we get

$$Me^{k+1,i+1} = Ne^{k+1,i}, \tag{9}$$

where $e^{k+1,i} = u^{k+1,i} - \widehat{u}^{k+1}$. This leads to the well-known result [20, 21]:

$$e^{k+1,n} = \left(M^{-1}N \right)^n e^{k+1,0}. \tag{10}$$

The iteration converges if the spectral radius of $M^{-1}N$ is less than one, and it diverges if the spectral radius is greater than one [20–22]. If the iteration converges, then we can get as close as desired to the implicitly coupled solutions $\widehat{u}^k$ by iterating. If for any reason we are unable to iterate until convergence at each time step, the situation is considerably more complicated. If $\widehat{u}^k$ cannot be obtained at a given time step, then it cannot be used as the initial condition for the next time step, and the iterates u^k may wander away from the implicitly coupled solutions $\widehat{u}^k$. To investigate how the iterates wander from their implicitly coupled counterparts, we must account for both the error from incomplete iteration at every time step and for the error associated with the inherited initial value at every time step.

3. Analysis of Stability of the Iterative Solution

3.1. Stability in Time for a Single Iteration. In this section, we derive an error formula for the case of a single iteration at each time step. The motivation is practical both as this is often encountered in practice and because the relatively

simple result shows how the general result proceeds. The error is defined to be the difference between the computed solution and the implicitly coupled solution at each time step. Recall the implicitly coupled solution satisfies

$$M\widehat{u}^{k+1} = \widehat{u}^k + S^{k+1} + N\widehat{u}^{k+1}. \tag{11}$$

For the case of a single iteration per time step, we have $n = 1$ and we can suppress the iteration index. The iteration is simply

$$Mu^{k+1} = u^k + S^{k+1} + Nu^k. \tag{12}$$

If we define $e^k = u^k - \widehat{u}^k$ and $\Delta\widehat{u}^{k+1} = \widehat{u}^k - \widehat{u}^{k+1}$, then (12) becomes

$$M\left(\widehat{u}^{k+1} + e^{k+1}\right) = \left(\widehat{u}^k + e^k\right) + S^{k+1}$$
$$+ N\left(\widehat{u}^{k+1} + \Delta\widehat{u}^{k+1} + e^k\right). \tag{13}$$

Using (11), we obtain

$$Me^{k+1} = e^k + N\left(\Delta\widehat{u}^{k+1} + e^k\right). \tag{14}$$

Substituting $k = 0$ and $k = 1$ into (14) and rearranging gives

$$e^1 = \left[M^{-1} + M^{-1}N\right]e^0 + \left[M^{-1}N\right]\Delta\widehat{u}^1, \tag{15}$$

$$e^2 = \left[M^{-1} + M^{-1}N\right]e^1 + \left[M^{-1}N\right]\Delta\widehat{u}^2. \tag{16}$$

Substituting (15) into (16) gives

$$e^2 = \left[M^{-1} + M^{-1}N\right]^2 e^0 + \left[M^{-1} + M^{-1}N\right]$$
$$\times \left[M^{-1}N\right]\Delta\widehat{u}^1 + \left[M^{-1}N\right]\Delta\widehat{u}^2. \tag{17}$$

In general, the error at time step k is

$$e^k = \left[M^{-1} + M^{-1}N\right]^k e^0$$
$$+ \sum_{j=1}^{k}\left[M^{-1} + M^{-1}N\right]^{(k-j)}\left[M^{-1}N\right]\Delta\widehat{u}^j. \tag{18}$$

While the form of (18) is not as concise as (10), it is clear that the stability in time depends on the spectral radius of $[M^{-1} + M^{-1}N]$.

3.2. Stability in Time for n Iterations. In this section, we derive a more general version of (18) for the case of n iterations at each time step. Beginning with (11), the iterates satisfy

$$Mu^{k+1,i} = u^{k,n} + S^{k+1} + Nu^{k+1,i-1},$$
$$i = 1,\dots,n, \quad u^{k+1,0} = u^{k,n}. \tag{19}$$

The error is now defined as $e^{k,i} = u^{k,i} - \widehat{u}^k$ for $i = 1,\dots,n$. We are interested in obtaining a formula for $e^{k,n}$.

The following relationships are useful in deriving the final result:

$$u^{k+1,0} = u^{k,n} = \widehat{u}^k + e^{k,n} = \widehat{u}^{k+1} + \Delta\widehat{u}^{k+1} + e^{k,n},$$
$$u^{k+1,i} = \widehat{u}^{k+1} + e^{k+1,i}, \quad i = 1,\dots,n. \tag{20}$$

Substituting these into (19) gives the following, for $i = 1$:

$$M\left(\widehat{u}^{k+1} + e^{k+1,1}\right) = \left(\widehat{u}^k + e^{k,n}\right) + S^{k+1}$$
$$+ N\left(\widehat{u}^{k+1} + \Delta\widehat{u}^{k+1} + e^{k,n}\right), \tag{21}$$

and, for $i = 2,\dots,n$,

$$M\left(\widehat{u}^{k+1} + e^{k+1,i}\right) = \left(\widehat{u}^k + e^{k,n}\right) + S^{k+1}$$
$$+ N\left(\widehat{u}^{k+1} + e^{k+1,i-1}\right). \tag{22}$$

Substituting into (11) gives the following, for $i = 1$:

$$Me^{k+1,1} = e^{k,n} + N\left(\Delta\widehat{u}^{k+1} + e^{k,n}\right), \tag{23}$$

and, for $i = 2,\dots,n$,

$$Me^{k+1,i} = e^{k,n} + Ne^{k+1,i-1}. \tag{24}$$

Starting with $e^0 = u^0 - \widehat{u}^0$, we can use (23) and (24) to find $e^{1,n}$ and $e^{2,n}$ and discover the general form for $e^{k,n}$. The process is tedious, but the pattern is quickly apparent. The final form for the error at time step k where n iterations have been taken per time step is

$$e^{k,n} = \left[T_n\right]^k e^0 + \sum_{j=1}^{k}\left[T_n\right]^{k-j}\left(M^{-1}N\right)^n \Delta\widehat{u}^j. \tag{25}$$

The form of T_n is as follows. For $n = 1$,

$$T_1 = \left[M^{-1} + M^{-1}N\right], \tag{26}$$

and, for $n > 1$, the following recursion relationship holds:

$$T_n = M^{-1} + M^{-1}N\left[T_{n-1}\right]. \tag{27}$$

For example,

$$T_2 = M^{-1} + M^{-1}N\left[M^{-1} + M^{-1}N\right],$$
$$T_3 = M^{-1} + M^{-1}N\left[M^{-1} + M^{-1}N\left[M^{-1} + M^{-1}N\right]\right]. \tag{28}$$

Note that the results derived above assume that M is being inverted exactly. A modification of (25) for the case of inexact inversion is given in Appendix B. A second modification of (25), incorporating the use of a weighted Jacobi method in which the new iterate is taken to be a weighted average of the old iterate and the Jacobi iterate, is given in Appendix C.

We conclude of course that the stability in time is determined by the spectral radius of the matrix T_n. However, given the conventional wisdom, it is important to point out that stability does not imply accuracy. If we examine (25),

it is clear from the term $[T_n]^k e^0$ that if T_n has a spectral radius of greater than one then even tiny machine rounding errors will grow out of control [20], and this is one way to describe the notion of instability. If the spectral radius of T_n is less than one, this can not occur and the method can then be considered stable. However, the summation term $\sum_{j=1}^{k} [T_n]^{k-j} (M^{-1}N)^n \Delta \hat{u}^j$ shows that it is still possible for error to accumulate in time, and the size of this error depends on the size of the vectors $\Delta \hat{u}$ as well as the effect of $M^{-1}N$ and T_n on $\Delta \hat{u}$. We consider these errors to be an issue of accuracy, not stability. This calls into question the conventional wisdom that if the result does not blow up, then the method is stable and therefore accurate.

We emphasize that the error formulas derived above measure the difference between the computed numerical solution and the idealized numerical solution obtained by solving the implicitly coupled discrete equations. The idealized numerical solution is itself a discrete approximation to the continuous solution of the PDE. The accuracy of the implicitly coupled solution relative to the continuous solution is a separate matter and would have some order in h and dt, where h and dt represent the cell width and time step, such as $O(h^2) + O(dt)$ assuming a typical second order accurate spatial discretization and first order accurate time discretization is used. Preserving this ideal accuracy requires that discrete effects such as solution error and errors arising from projecting between component discretizations are minimal [19].

3.3. Relationship between T_n and A^{-1}. An alternative form for the T_n is

$$T_n = \left[\sum_{i=0}^{n-1} \left(M^{-1}N \right)^i \right] M^{-1} + \left(M^{-1}N \right)^n. \tag{29}$$

This form suggests a power series representation for an approximate inverse. Rewriting the splitting $A = M - N$ as $(M - N) = M[I - (M^{-1}N)]$, then

$$(M - N)^{-1} = \left[I - \left(M^{-1}N \right) \right]^{-1} M^{-1}. \tag{30}$$

Using a power series for the inverse in brackets [20–22], assuming that $\|M^{-1}N\| < 1$, we get

$$(M - N)^{-1} = \left[\sum_{i=0}^{\infty} \left(M^{-1}N \right)^i \right] M^{-1}. \tag{31}$$

Hence, provided $\|M^{-1}N\| < 1$, T_n becomes an increasingly accurate approximation of A^{-1} as the number of iterations n increases. Specifically, the first term in (29) approaches (31) as n increases, while the second term in (29) goes to zero.

4. Numerical Examples

We can employ the error formulas derived above by forming the matrix T_n for any given set of discretization parameters, for example, step size, space mesh, and number of iterations per time step, and then easily examine the stability of the
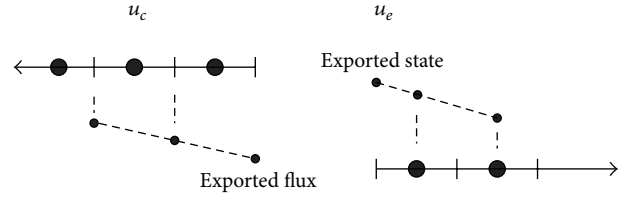

FIGURE 2: Diagram of the coupling strategy used in the numerical experiments, in which linear extrapolation from the last two available state or flux values in one subdomain are used to compute a boundary value for the other subdomain.

overall algorithm for particular choices. In this section, we present numerical experiments designed to examine aspects of the conventional wisdom.

We consider a one dimensional domain with $x \in [0, 1]$ and the interface between the two subdomains located at $x = 1/2$, and we pose the heat equation with constant diffusivity equal to one in each subdomain ($a_1 = a_2 = 1$). The outer boundary conditions are homogeneous Neumann at $x = 0$ and homogeneous Dirichlet at $x = 1$. Error measurement is facilitated by constructing the problems so that the exact continuous solution is $\mathcal{U}(x, t) = (1 - x^2)e^{-t}$. The complete statement of the PDE is

$$\frac{\partial}{\partial t} \mathcal{U}_1 - \frac{\partial^2}{\partial x^2} \mathcal{U}_1 = \left(1 + x^2 \right) e^{-t}, \quad x \in [0, .5], \ t \in [0, T],$$

$$\frac{\partial}{\partial t} \mathcal{U}_2 - \frac{\partial^2}{\partial x^2} \mathcal{U}_2 = \left(1 + x^2 \right) e^{-t}, \quad x \in [.5, 1], \ t \in [0, T],$$

$$\mathcal{U}_1 = \mathcal{U}_2, \quad x = .5, \ t \in [0, T],$$

$$\frac{\partial}{\partial x} \mathcal{U}_1 = \frac{\partial}{\partial x} \mathcal{U}_2, \quad x = .5, \ t \in [0, T],$$

$$\frac{\partial}{\partial x} \mathcal{U}_1(0, t) = 0,$$

$$\mathcal{U}_2(1, t) = 0,$$

$$\mathcal{U}_1(x, 0) = \mathcal{U}_2(x, 0) = \left(1 - x^2 \right). \tag{32}$$

We use finite volume in space and backward Euler in time [13, 18, 19, 21, 22]. The details of the discrete equations are given in Appendix A. The cell-centered finite-volume method provides state values at cell centers. Fluxes at cell boundaries are calculated via differencing, while linear extrapolation is used to compute coupling values at the boundary between subdomains. Component 1 receives a Dirichlet condition at the interface that is a linear extrapolation of the two state values that are the nearest to the interface in component 2 and, similarly, component 2 receives a Neumann condition at the interface which is a linear extrapolation of the two flux values that are the nearest to the interface in component 1 [8, 9]. The extrapolation scheme is depicted in Figure 2.

In the examples, we use a Jacobi iteration with $\omega = 1$. Once the exchange of information is complete, each subdomain is solved to within machine precision by direct
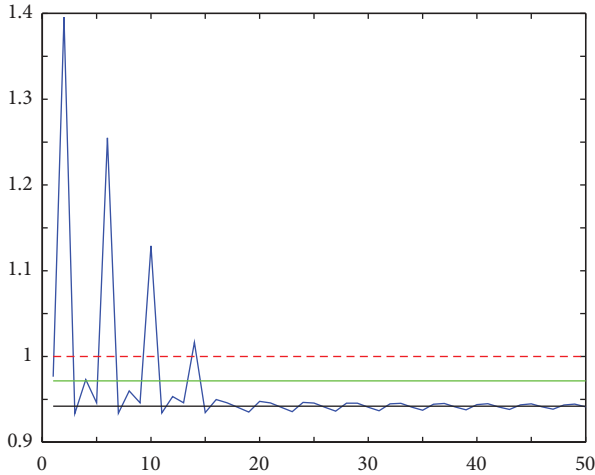
FIGURE 3: Spectral radius for various values of n with fixed dt. Black is A^{-1}, green is $M^{-1}N$, blue is T_n, and red marks the value one.

TABLE 1: Relative error at $t = 2$ for various numbers of iterations.

n	e_{rel}
1	.05
2	$4.4 * 10^9$
3	.09
4	.65
5	.05
6	$8.5 * 10^5$
200	$1.8 * 10^{-4}$
400	$5.6 * 10^{-7}$

inversion of the matrix M. At each time step, the code produces both the implicitly coupled solution, $\hat{u}$, by inverting the coupled system matrix A and the iterative solution, u, by performing the specified number of iterations. The implicitly coupled and iterative solutions can then be compared at any time step. It is easily verified that the implicitly coupled solution is unconditionally stable and has accuracy $O(h^2) + O(dt)$. In the following examples, we concentrate on the error between the implicitly coupled and iterative solutions. We introduce the relative error, which is based on the standard discrete two-norm:

$$e_{\text{rel}} = \frac{\|u - \hat{u}\|_2}{\|\hat{u}\|_2}. \tag{33}$$

Finally, we conduct similar experiments with a wide range of spatial mesh sizes. The qualitative results are the same in every case, although the specific threshold values may of course vary.

4.1. Case 1: Stability of the Algorithm with n Iterations Using a Fixed Time Step, dt. In the first experiment, we set the grid size to be 16 cells in each subdomain ($h = 1/32$) and fix the time step at $dt = 1/40$. We plot the spectral radius of T_n, $M^{-1}N$, and A^{-1} for various values of n in Figure 3. Note that the spectral radius of $M^{-1}N$ is less and one, so the iteration is convergent, and the spectral radius of A^{-1} is less than one, so the implicitly coupled scheme would be unconditionally stable if we had the luxury of inverting A^{-1} at each time step. However, the method is unstable for certain values of n. If one observes the progress of the iterative solution at a given time step, it is clear that although the iteration is convergent, certain early iterates contain significantly more error than the previous iterate. In other words, the error in the computational iterate does not decrease monotonically. For the particular problem examined here, it is every fourth iterate, starting with the second iterate, that has the increased error. Figure 3 shows that for the first few n values in this sequence ($n = 2, 6, 10, 14$) the reduced quality of the iterate

results in the entire algorithm being unstable, despite the fact that the iteration is convergent. As the number of iterations increases, this effect is reduced in magnitude and eventually disappears. We verify the instability by solving to $t = 2$ (80 time steps) and listing the relative error at $t = 2$ for several values of n in Table 1.

The values $n = 200$ and $n = 400$ are included in Table 1 to verify that, since the iteration is convergent, the relative error approaches zero as the number of iterations becomes very large. The reduction in relative error for large n is fairly slow, since the spectral radius of $M^{-1}N$ is approximately .975, as indicated in Figure 3.

The results of this experiment serve to illustrate one of our main points that a low number of iterations can interfere with the unconditional stability of the time discretization. In addition, it shows that a lower number of iterations can be stable, while a higher number is unstable. The plot of the spectral radius of T_n in Figure 3 indicates that using 2, 6, 10, or 14 iterations leads to an unstable method. The numerical values in Table 1 confirm the instability for $n = 2$ and $n = 6$ ($n = 10$ and $n = 14$ yield analogous results). Backward Euler is considered "unconditionally stable" [21, 22]; however, this fact is based on the assumption that the system is solved exactly at each time step. For the iterative algorithm used here, this unconditional stability is only valid in the limit as the number of iterations goes to infinity. This example shows how sensitive the stability of the algorithm is to the number of iterations used. Note that this example also provides another demonstration of the fact that stability does not imply accuracy. Notice in Table 1 that the relative error for the stable case $n = 4$ is around ten times that of the nearby stable cases, $n = 1, 3, 5$. Despite the fact that the spectral radius of T_n is less than one for all the stable cases, it is still possible for the summation term in (25) to accumulate significant error in time. This term makes a much larger contribution for the case of $n = 4$ than it does for the other stable cases. This is not evident from the spectral radius of T_n alone; note that $\rho(T_1) > \rho(T_4)$.

4.2. Case 2: Stability of the Algorithm with Time Step dt, Using a Fixed Number of Iterations, n. In the second experiment, we use identical conditions except that we fix the number of iterations n and vary the time step dt. Figure 4 shows the spectral radii of the relevant matrices for $n = 1$ and values of dt ranging from 0 to 0.1. Since $h = 1/32$, this range
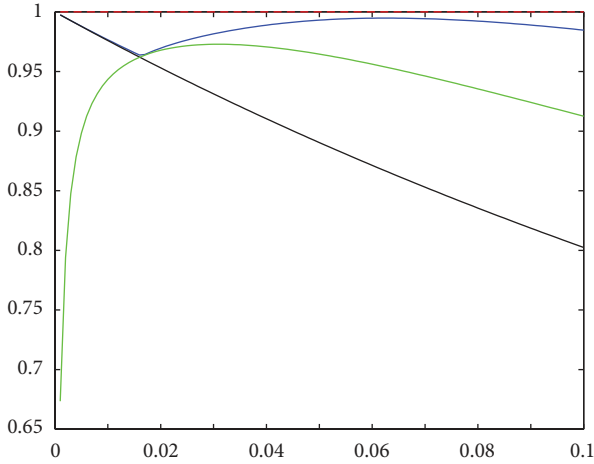
FIGURE 4: Spectral radius for various values of dt with $n = 1$. Black is A^{-1}, green is $M^{-1}N$, blue is T_n, and red marks the value one.
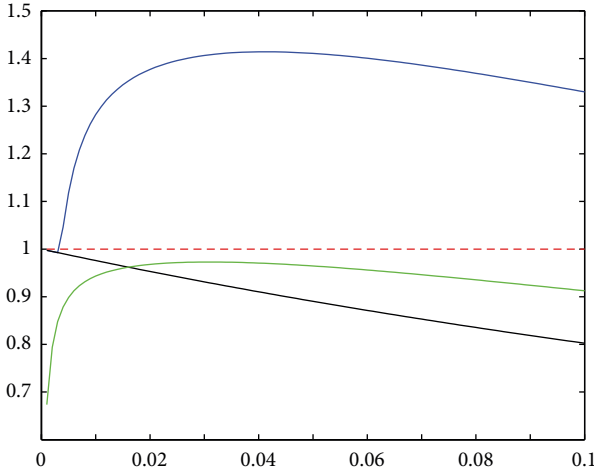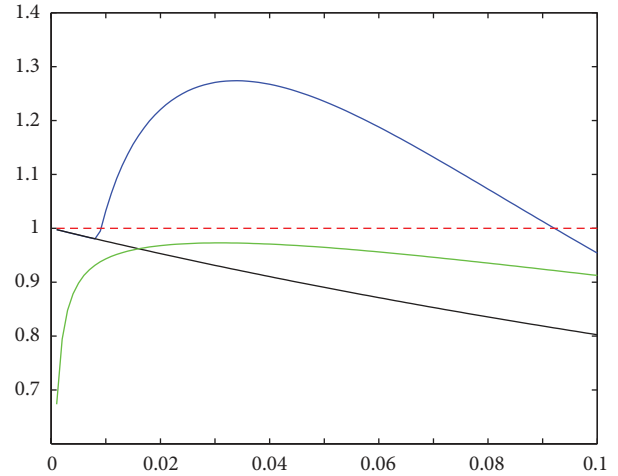

FIGURE 5: Spectral radius for various values of dt with $n = 2$. Black is A^{-1}, green is $M^{-1}N$, blue is T_n, and red marks the value one.

TABLE 2: Relative error at $t = 2$ for $n = 1$ with various values of dt.

dt	e_{rel}
.005	.01
.05	.08
.1	.22

of time step dt corresponds to a range for the ratio dt/h^2 of approximately 1 to 100. Figure 4 implies that the method with $n = 1$ is stable for all $dt < 0.1$. Carrying out the calculation confirms this, and the relative errors at the end of the simulation are given in Table 2.

Figure 5 shows the spectral radii of the relevant matrices for $n = 2$. The implication is that the method with $n = 2$ is stable only for very small values of dt. Table 3 gives relative errors for the $n = 2$ case.

For the case of $n = 6$, the plot and table are shown in Figure 6 and Table 4. This case illustrates that instability

TABLE 3: Relative error at $t = 2$ for $n = 2$ with various values of dt. Note that for larger values of dt, fewer time steps are needed to reach $t = 2$, so the relative error does not grow as large.

dt	e_{rel}
.0025	.01
.005	$5.7 * 10^{16}$
.05	$2.5 * 10^4$
.1	19

FIGURE 6: Spectral radius for various values of dt with $n = 6$. Black is A^{-1}, green is $M^{-1}N$, blue is T_n, and red marks the value one.

TABLE 4: Relative error at $t = 2$ for $n = 6$ with various values of dt.

dt	e_{rel}
.005	.0135
.05	117
.1	.0114

can occur for a limited range of time steps with a minimum and a maximum, not simply for time steps above a certain threshold, as one might expect. The $n = 6$ case makes it clear that the method is stable for sufficiently small and sufficiently large time steps and is unstable in between. (In fact this is also true for the $n = 2$ case, and if the horizontal axis in Figure 5 is carried out to much larger time steps the spectral radius of T_2 does eventually drop below one.) This is a surprising result since one would not expect reducing the time step to promote instability, nor increasing the time step to restore stability.

Finally, in Figure 7 we provide two further plots for the $n = 10$ and $n = 18$ cases. These show the relationship between T_n and A^{-1} discussed in Section 3.3. The spectral radius of A^{-1} is less than one for all values of dt, and it decreases monotonically with increasing dt. We know from the expansions in Section 3.3 that as n becomes large then T_n approaches A^{-1}. However, there is no reason to expect the spectral radius of T_n to be monotonic in dt for small values of n, and the plots in this section show that it is not. There is a characteristic bump in the plots of the spectral radius of T_n versus dt, and it is in this range of dt values that the spectral
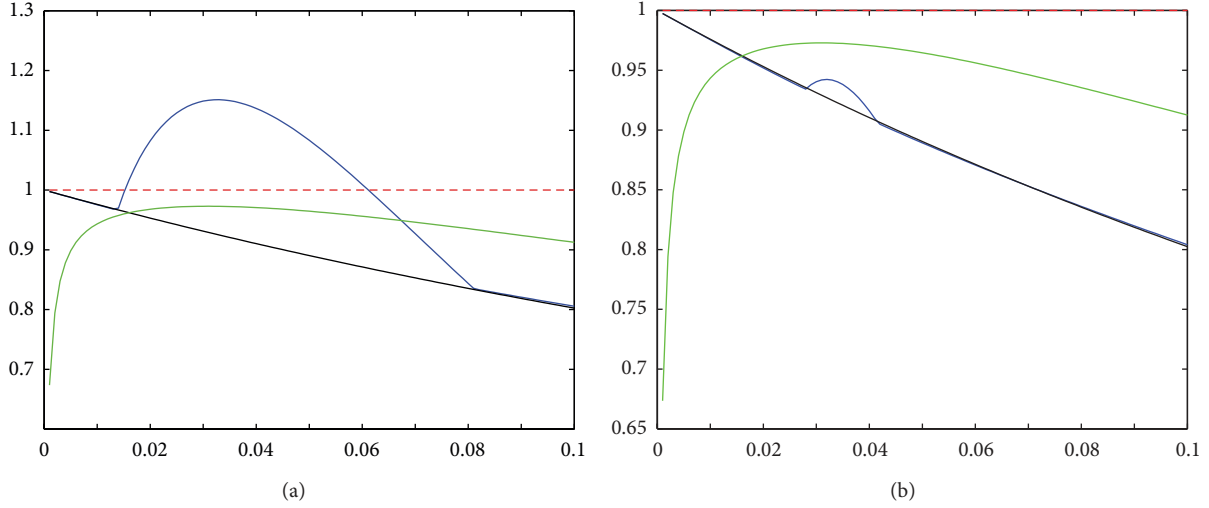
FIGURE 7: Spectral radius for various values of dt with $n = 10$ (a), and $n = 18$ (b). Black is A^{-1}, green is $M^{-1}N$, blue is T_n, and red marks the value one.

radius of T_n may exceed 1, leading to an unstable method. Naturally, as n becomes large, the size of this bump is reduced, since T_n is driven closer to A^{-1}. In the $n = 18$ case, the spectral radius of T_n is less than 1 for all time steps.

5. The Multirate Case

Multirate integration using different time steps for the different components is a common practice [1, 8, 9]. In this case, the implicitly coupled system and error formulas are changed slightly. We assume the longer time step is an integer multiple of the shorter time step, so component 2 will take p steps for every one step in the component 1. The outline of the derivation for $p = 2$ is given here, but results easily generalize. The large time steps correspond to the index k and the small time steps correspond to fractional indices. First, component 1 equation is straightforward since it only involves one time step:

$$L_1 \hat{u}_1^{k+1} = \hat{u}_1^k + S_1^{k+1} + P_{21} \hat{u}_2^{k+1}. \tag{34}$$

Next, component 2 equation requires compounding two steps:

$$L_2 \hat{u}_2^{k+1/2} = \hat{u}_2^k + S_2^{k+1/2} + P_{12} \hat{u}_1^{k+1},$$
$$L_2 \hat{u}_2^{k+1} = \hat{u}_2^{k+1/2} + S_2^{k+1} + P_{12} \hat{u}_1^{k+1}. \tag{35}$$

After carrying out the algebra and generalizing to larger p, we obtain the system:

$$\begin{bmatrix} L_1 & -P_{21} \\ -\sum_{j=0}^{p-1} \left(L_2^{-1}\right)^j P_{12} & L_2 \end{bmatrix} \begin{bmatrix} \hat{u}_1^{k+1} \\ \hat{u}_2^{k+1} \end{bmatrix}$$

$$= \begin{bmatrix} \hat{u}_1^k + S_1^{k+1} \\ \left(L_2^{-1}\right)^{p-1} \hat{u}_2^k + \sum_{j=0}^{p-1} \left(L_2^{-1}\right)^j S_2^{k+(p-j)/p} \end{bmatrix}. \tag{36}$$

The fractional index is used to indicate the small time steps.

Now that the implicitly coupled system has been defined, we can choose a splitting that defines M and N in the error formulas above. The error formula for the multirate case is very similar to (25), but we must define matrices $T_{n,p}$ where n is the number of iterations per time step and p is the integer number of time steps taken in component 2 for each time step taken in component 1. Let

$$Z_p = \begin{bmatrix} I_1 & 0 \\ 0 & \left(L_2^{-1}\right)^{p-1} \end{bmatrix}, \tag{37}$$

where I_1 is an identity matrix of the same size as L_1. The recursion relationship for the $T_{n,p}$ is

$$T_{1,p} = M^{-1} Z_p + M^{-1} N,$$
$$T_{n,p} = M^{-1} Z_p + M^{-1} N T_{n-1,p}. \tag{38}$$

The alternative form for $T_{n,p}$, analogous to (29) is

$$T_{n,p} = \left[\sum_{i=0}^{n-1} \left(M^{-1}N\right)^i \right] M^{-1} Z_p + \left(M^{-1}N\right)^n. \tag{39}$$

This means that as n goes to infinity, $T_{n,p}$ approaches $A^{-1} Z_p$ (note that $Z_1 = I$). Finally, the multirate error formula is

$$e_p^{k,n} = \left[T_{n,p}\right]^k e^0 + \sum_{j=1}^{k} \left[T_{n,p}\right]^{k-j} \left(M^{-1}N\right)^n \Delta \hat{u}^j. \tag{40}$$

5.1. Multirate Example 1: Varying p with Constant n. The multirate examples are the same as those described in Section 4.2 with only the values of n and p altered. We use the same Jacobi style iteration, so based on (36) we have

$$M = \begin{bmatrix} L_1 & 0 \\ 0 & L_2 \end{bmatrix}, \qquad N = \begin{bmatrix} 0 & P_{21} \\ \sum_{j=0}^{p-1} \left(L_2^{-1}\right)^j P_{12} & 0 \end{bmatrix}. \tag{41}$$
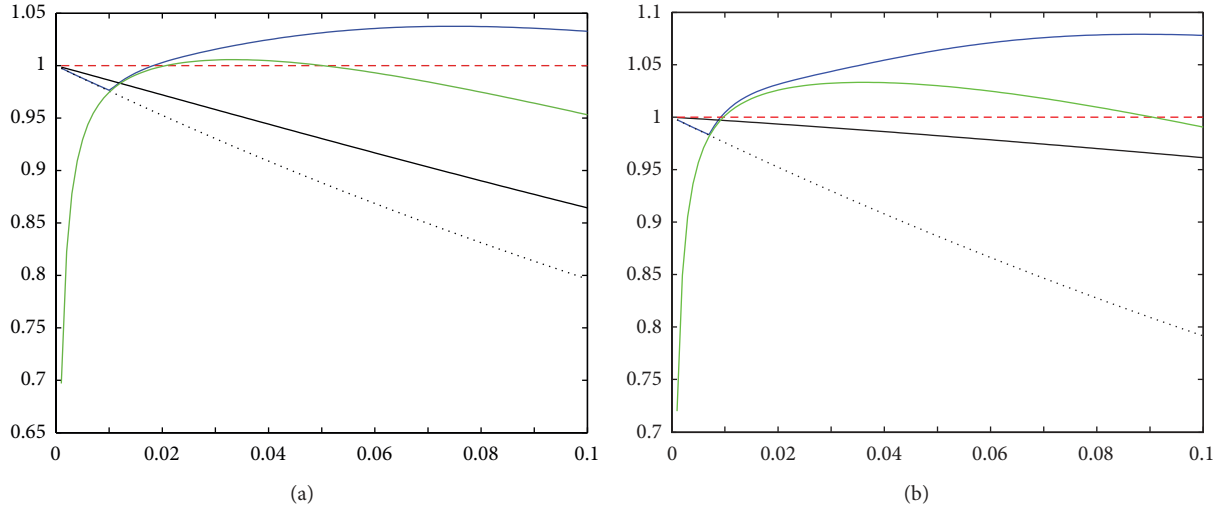
FIGURE 8: Spectral radii for multirate example 1. $n = 1$, $p = 2$ (a), and $n = 1$, $p = 10$ (b). Black is A^{-1}, dotted black is $A^{-1}Z_p$, green is $M^{-1}N$, blue is T_n, and red marks the value one. The horizontal axis shows the size of the large time step.
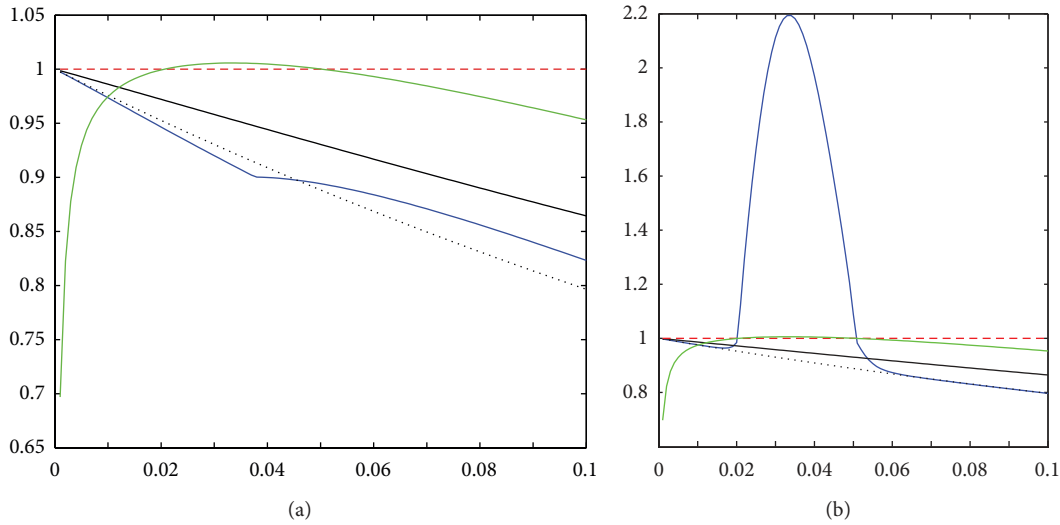

FIGURE 9: Spectral radii for multirate example 2. $n = 3$, $p = 2$ (a), and $n = 100$, $p = 2$ (b). Black is A^{-1}, dotted black is $A^{-1}Z_p$, green is $M^{-1}N$, blue is T_n, and red marks the value one. The horizontal axis shows the size of the large time step.

In the first multirate example, n is set to 1, and p is set first to 2 and then to 10. Figure 8 shows the spectral radii of the relevant matrices.

Values of p larger than 10 produce no visible change in the plot, so Figure 8(b) can be taken to represent the case of "many" time steps in component 2 within each time step for component 1. There is now a range of time steps for which the iteration diverges. As might be expected, the algorithm is unstable when the iteration is divergent. In other words, when the spectral radius of $M^{-1}N$ is greater than one, the spectral radius of T_n is greater than one.

As the number of iterations, n, approaches infinity, a divergent iteration must result in an unstable algorithm. However, for the case of finite iteration, this is not guaranteed, and the next experiment illustrates this point.

5.2. *Multirate Example 2: Varying n with Constant p.* In the second multirate example, p is set to 2, and n is set first to 3 and then to 100. Figure 9 shows the spectral radii for the relevant matrices. The results show that the method may be stable, even though the iteration is divergent. Figure 9(a) include a region in which the spectral radius of $M^{-1}N$ is more than one, yet the spectral radius of T_n is less than one. In Figure 9(a), where only 3 iterations are used per large time step, the entire range of dt values for which the iteration is divergent results in a stable algorithm. In Figure 9(b), where the number of iterations is raised to 100, the region of divergence and the region of instability are very nearly the same. Such cases are not unique to multirate examples and have also been observed in single rate examples. Of course these cases can only occur when the number of iterations per
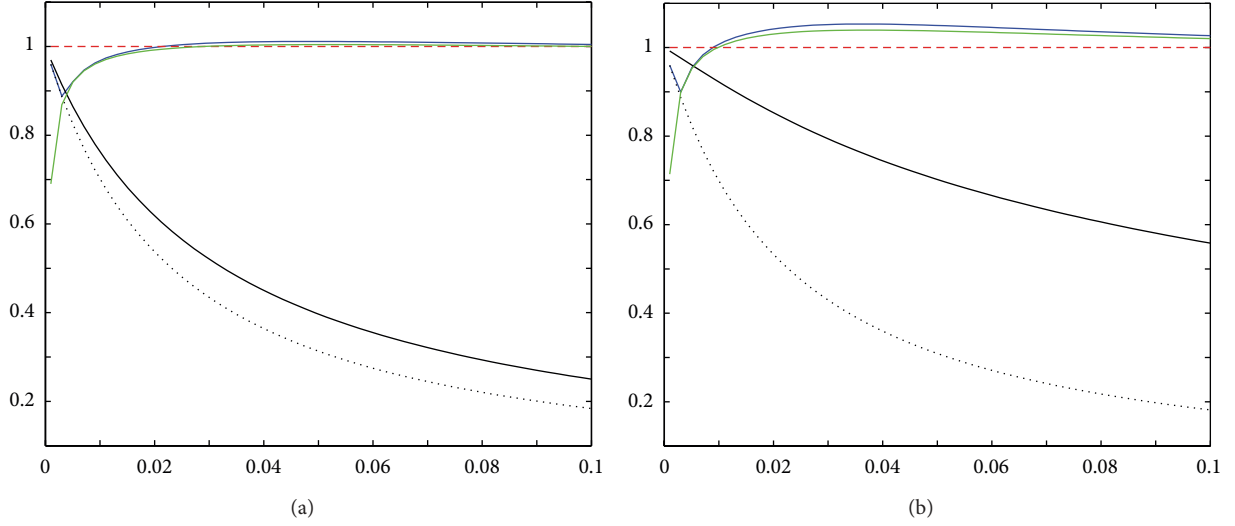
FIGURE 10: Spectral radii for multirate example 1 in 2 spatial dimensions. $n = 1$, $p = 2$ (a), and $n = 1$, $p = 10$ (b). Black is A^{-1}, dotted black is $A^{-1}Z_p$, green is $M^{-1}N$, blue is T_n, and red marks the value one. The horizontal axis shows the size of the large time step.

time step is small, since iterating many times with a divergent iteration would certainly lead to an unstable algorithm. This is an interesting case, since a stable algorithm could contain a divergent iteration and the user might not know.

5.3. Multirate Example in Two Spatial Dimensions.
In this section, the multirate tests in Sections 5.1 and 5.2 are extended to a problem in two spatial dimensions posed on $x_1 \in [0, 1]$, $x_2 \in [0, .5]$ with the interface located at $x_1 = .5$:

$$\frac{\partial}{\partial t} \mathcal{U}_1 - \Delta \mathcal{U}_1 = f_L(x_1, x_2, t),$$

$$(x_1, x_2) \in [0, .5] \times [0, .5], \quad t \in [0, T],$$

$$\frac{\partial}{\partial t} \mathcal{U}_2 - \Delta \mathcal{U}_2 = f_R(x_1, x_2, t),$$

$$(x_1, x_2) \in [.5, 1] \times [0, .5], \quad t \in [0, T],$$

$$\mathcal{U}_1 = \mathcal{U}_2, \quad x_1 = .5, \ x_2 \in [0, .5], \ t \in [0, T],$$

$$\hat{n} \cdot \nabla \mathcal{U}_1 = \hat{n} \cdot \nabla \mathcal{U}_2, \quad x_1 = .5, \ x_2 \in [0, .5], \ t \in [0, T],$$

$$\frac{\partial}{\partial x_1} \mathcal{U}_1(x_1, x_2, t) = 0, \quad x_1 = 0, \ x_2 \in [0, .5], \ t \in [0, T],$$

$$\mathcal{U}_1(x_1, x_2, t) = 0, \quad x_1 = 1, \ x_2 \in [0, .5], \ t \in [0, T],$$

$$\mathcal{U}_2(x_1, x_2, t) = 0, \quad x_1 \in [0, 1], \ x_2 = 0 \text{ or } .5, \ t \in [0, T],$$

$$\mathcal{U}_1(x_1, x_2, 0) = g_L(x_1, x_2, t), \quad x_1 \in [0, 1], \ x_2 \in [0, .5],$$

$$\mathcal{U}_2(x_1, x_2, 0) = g_R(x_1, x_2, t), \quad x_1 \in [0, 1], \ x_2 \in [0, .5].$$
$$(42)$$

The purpose of this example is to examine the changes in the spectral radii of the relevant matrices resulting from

the change to two spatial dimensions. We use the standard finite volume discretization with 16×16 cells in each subdomain. Since the functions f_L, f_R, g_L, and g_R have no impact on the relevant matrices, we do not state them explicitly.

Plots of the spectral radii of the relevant matrices are given in Figures 10 and 11. The spectral radius of A^{-1} is much smaller in the two space dimension case. Nonetheless, the plots show that the two space dimension case exhibits the same qualitative behavior as the one space dimension case. In particular, there is a range of time steps for which the iteration is divergent and also a range of time steps for which the overall method is unstable.

6. Conclusion

By selecting a coupling strategy comprising space and time grids and an associated rule by which information is exchanged between components, we define an implicitly coupled system. At each time step, we seek a solution of this implicitly coupled system through a block iterative strategy. Since only a limited number of iterations can be performed at each time step, there will be some iteration error which separates the final iterate from the implicitly coupled solution. These errors are propagated to the next time step in the form of errors in the initial condition and are compounded as the incomplete iteration process repeats itself at each time step. The cumulative effect can manifest itself as conditional stability, meaning the solution is only stable for certain values of dt despite the unconditional stability of the implicitly coupled solution.

We have derived formulas for this error which show that stability hinges on the spectral radius of the matrix T_n and further show that there is a mechanism for error to accumulate in time even if the algorithm is stable.
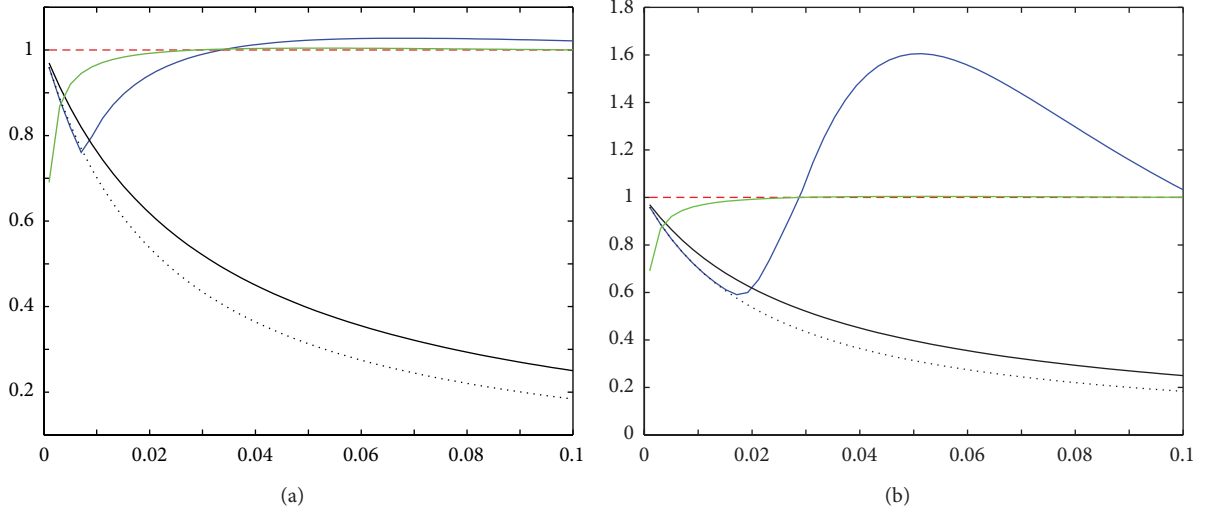
FIGURE 11: Spectral radii for multirate example 2 in 2 spatial dimensions. $n = 3$, $p = 2$ (a), and $n = 100$, $p = 2$ (b). Black is A^{-1}, dotted black is $A^{-1}Z_p$, green is $M^{-1}N$, blue is T_n, and red marks the value one. The horizontal axis shows the size of the large time step.

By examining the spectral radius of T_n for a variety of discretization choices on a simple model problem we demonstrate several important points. Firstly, the unconditional stability of the time integration method is not necessarily retained if a low number of iterations per timestep are used. Instead, a conditional stability may occur, where the method is stable for a range of values of dt. In additon, we show by example that it is possible for the method to be stable even if the iteration is divergent, provided a low number of iterations is used. These results seem to contradict intuition, which suggests that a convergent iteration will produce a stable algorithm and a divergent iteration will produce an unstable algorithm. However, both of these ideas hold only in the limit as the number of iterations goes to infinity, and in the case of finite iteration we have shown that there are exceptions.

Finally, the question of accuracy must be considered in addition to the issue of stability. The error formulas derived above show that the iterative solution may wander away from the implicitly coupled solution over time even if the method is stable in the traditional sense.

Appendices

A. Discrete Equations for 1D Finite Volume with Backward Euler in Time on the Heat Equation

Let there be n cells in the 1D spatial grid in a given subdomain, $x \in [x_a, x_b]$. Let u_i^j be the discrete solution on cell i at time j, dt the constant time step, and h the constant cell width. The symbol f_i^j represents the exact integral over cell i of the right hand side of the PDE evaluated at time j. The symbol $a_{i+1/2}^j$ represents the diffusivity function evaluated at the boundary between cell i and cell $i+1$, at time j. The system of equations below, when placed into matrix form, describes the terms in (3) as they were implemented for the numerical tests.

$$u_i^{j+1} + \frac{dt}{h}\left(N^{j+1} - \frac{a_{i+1/2}^{j+1}\left(u_{i+1}^{j+1} - u_i^{j+1}\right)}{h} \right) = \frac{dt}{h} f_i^{j+1} + u_i^j,$$

$$\text{for } i = 1,$$
$$\text{(A.1)}$$

where N^{j+1} is the known Neumann boundary condition at $x = x_a$ at time $j + 1$.

Consider

$$u_i^{j+1} + \frac{dt}{h}\left[\frac{a_{i-1/2}^{j+1}\left(u_i^{j+1} - u_{i-1}^{j+1}\right)}{h} - \frac{a_{i+1/2}^{j+1}\left(u_{i+1}^{j+1} - u_i^{j+1}\right)}{h} \right]$$

$$= \frac{dt}{h} f_i^{j+1} + u_i^j, \quad \text{for } i = 2, \ldots, n - 1,$$

$$u_i^{j+1} + \frac{dt}{h}\left(\frac{a_{i-1/2}^{j+1}\left(u_i^{j+1} - u_{i-1}^{j+1}\right)}{h} - \frac{a_{i+1/2}^{j+1}\left(D^{j+1} - u_i^{j+1}\right)}{h/2} \right)$$

$$= \frac{dt}{h} f_i^{j+1} + u_i^j, \quad \text{for } i = n,$$
$$\text{(A.2)}$$

where D^{j+1} is the known Dirichlet boundary condition at $x = x_b$ at time $j + 1$.

Note that (A.1) and (A.2) are valid for both of our subdomains provided that N^{j+1} and D^{j+1} are interpreted as given boundary conditions on the outer boundaries of the entire domain and are determined by the extrapolation

strategy depicted in Figure 2 at the interface between the subdomains.

B. Inexact Component Solutions (Inexact Inversion of M)

The error formulas derived in Sections 3.1 and 3.2 can easily be modified to include the effects of inexact inversion of M. We first adjust (18) from Section 3.1, for a single iteration per time step. The iterates are now defined by

$$u^{k+1} = M^{-1}\left[u^k + S^{k+1} + Nu^k\right] + \epsilon^k, \qquad \text{(B.1)}$$

where ϵ^k represents the error from inexact inversion. The result corresponding to (18) in Section 3.1 is now

$$e^k = [T_1]^k e^0 + \sum_{j=1}^{k} [T_1]^{(k-j)}\left[M^{-1}N\right]\Delta\widehat{u}^j$$

$$+ \sum_{j=1}^{k} [T_1]^{(k-j)}\epsilon^j. \qquad \text{(B.2)}$$

Notice the extra term resulting from incomplete iteration.

Repeating the process for n iterations per time step, the computational iterates are defined by

$$u^{k+1,i} = M^{-1}\left[u^k + S^{k+1} + Nu^{k+1,i-1}\right] + \epsilon^{k+1,i}. \qquad \text{(B.3)}$$

Following the same logic and derivation used in Section 3.2, the expression corresponding to (25) is

$$e^{k,n} = [T_n]^k e^0 + \sum_{j=1}^{k} [T_n]^{k-j}\left(M^{-1}N\right)^n\Delta\widehat{u}^j$$

$$+ \sum_{j=1}^{k} [T_n]^{(k-j)}\sum_{i=1}^{n}\left(M^{-1}N\right)^{n-i}\epsilon^{j,i}. \qquad \text{(B.4)}$$

Notice that, with multiple iterations per time step, the ϵ term needs a double index because an error due to inexact inversion is made at every iteration. Equation (B.4) contains both terms from (25) and has an additional term which represents the error caused by the inexact inversions. The matrix T_n is the key to the growth of all three terms and therefore remains the key to stability.

C. Weighted Jacobi

It is worth noting that the error formula (25) can easily be altered to allow for a "relaxed" Jacobi iteration [20–22], in which a weighted average of the new and old iterates of a standard Jacobi iteration is taken to be the next iterate (see, e.g., [23]). If we rewrite (25) as

$$e^k = [T_n]^k e^0 + \sum_{j=1}^{k} [T_n]^{k-j}(R_\omega)^n\Delta\widehat{u}^j, \qquad \text{(C.1)}$$

with $R_\omega = [(1-\omega)I + \omega M^{-1}N]$ and redefine $T_1 = [\omega M^{-1}+R_\omega]$, then for $n > 1$

$$T_n = \omega M^{-1} + R_\omega[T_{n-1}]. \qquad \text{(C.2)}$$

With this definition of T_n, (25) holds for any value of $\omega \in [0, 1]$, with $\omega = 1$ corresponding to standard iteration. In some cases where the iterative convergence at each time step is slow, the iterates tend to oscillate around the implicitly coupled solution at that time step. In these cases, adjusting ω can drastically accelerate iterative convergence.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

References

[1] D. E. Keyes, L. C. McInnes, C. Woodward et al., "Multiphysics simulations: challenges and opportunities," *International Journal of High Performance Computing Applications*, vol. 27, no. 1, pp. 4–83, 2013.

[2] D. Estep, S. Tavener, and T. Wildey, "A posteriori analysis and improved accuracy for an operator decomposition solution of a conjugate heat transfer problem," *SIAM Journal on Numerical Analysis*, vol. 46, no. 4, pp. 2068–2089, 2008.

[3] D. Estep, S. Tavener, and T. Wildey, "A posteriori error analysis for a transient conjugate heat transfer problem," *Finite Elements in Analysis and Design*, vol. 45, no. 4, pp. 263–271, 2009.

[4] D. Estep, S. Tavener, and T. Wildey, "A posteriori error estimation and adaptive mesh refinement for a multi-discretization operator decomposition approach to fluid-solid heat transfer," *Journal of Computational Physics*, vol. 229, no. 11, pp. 4143–4158, 2010.

[5] A. G. Fedorov and R. Viskanta, "Three-dimensional conjugate heat transfer in the microchannel heat sink for electronic packaging," *International Journal of Heat and Mass Transfer*, vol. 43, no. 3, pp. 399–415, 2000.

[6] Y. Cao, M. Gunzburger, X. Hu, F. Hua, X. Wang, and W. Zhao, "Finite element approximations for Stokes-Darcy flow with Beavers-Joseph interface conditions," *SIAM Journal on Numerical Analysis*, vol. 47, no. 6, pp. 4239–4256, 2010.

[7] S. Mazumder, "Modeling full-scale monolithic catalytic converters: challenges and possible solutions," *Journal of Heat Transfer*, vol. 129, no. 4, pp. 526–535, 2007.

[8] J. R. Cary, J. Candy, R. H. Cohen et al., "First results from core-edge parallel composition in the FACETS project," *Journal of Physics: Conference Series*, vol. 125, Article ID 012040, 2008.

[9] J. R. Cary, A. Hakim, M. Miah et al., "FACETS—a framework for parallel coupling of fusion components," in *Proceedings of the 18th Euromicro International Conference on Parallel, Distributed and Network-Based Computing*, pp. 435–442, Pisa, Italy, February 2010.

[10] M. Dryja, "Substructuring methods for parabolic problems," in *Proceedings of the 4th International Symposium on Domain Decomposition Methods for Partial Differential Equations*, R. Glowinski, Ed., vol. 4, chapter 24, SIAM, Philadelphia, Pa, USA, 1991.

[11] A. Quarteroni, F. Pasquarelli, and A. Valli, "Heterogeneous domain decomposition: principles, algorithms, applications," in *Proceedings of the 5th International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pp. 129–150, SIAM, Philadelphia, Pa, USA, 1992.

[12] S. Gaiffe, R. Glowinski, and R. Masson, "Domain decomposition and splitting methods for mortar mixed finite element approximations to parabolic equations," *Numerische Mathematik*, vol. 93, no. 1, pp. 53–75, 2002.

[13] C. Bernardi, Y. Maday, and A. T. Patera, "A new nonconforming approach to domain decomposition: the mortar element method," in *Nonlinear Partial Differential Equations and Their Applications*, Longman Scientific and Technical, Harlow, UK, 1994.

[14] P. L. Lions, "On the Schwarz alternating method I," in *Proceedings of the 1st International Symposium on Domain Decomposition Methods for Partial Differential Equations*, R. Glowinski, G. H. Golub, G. A. Meurant, and J. Periaux, Eds., pp. 1–42, SIAM, Philadelphia, Pa, USA, 1988.

[15] P. L. Lions, "On the Schwarz alternating method II," in *Decomposition Methods*, F. Tony, R. Glowinski, J. Periaux, and O. B. Widlund, Eds., pp. 47–70, SIAM, Philadelphia, Pa, USA, 1989.

[16] H. A. Schwarz, *Gesammelte Mathematische Abhandlungen*, vol. 2, Springer, Berlin, Germany, 1890.

[17] B. F. Smith, P. E. Bjorstad, and W. D. Gropp, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press, 1996.

[18] C. Bernardi, Y. Maday, and F. Rapetti, "Basics and some applications of the mortar element method," *GAMM-Mitteilungen*, vol. 28, no. 2, pp. 97–123, 2005.

[19] T. Arbogast, D. Estep, B. Sheehan, and S. Tavener, "A posteriori error estimates for mixed finite element and finite volume methods for problems coupled through a boundary with nonmatching grids," *IMA Journal of Numerical Analysis*, vol. 34, no. 4, pp. 1625–1653, 2014.

[20] G. H. Golub and C. F. van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, Md, USA, 2nd edition, 1989.

[21] K. Atkinson, *An Introduction to Numerical Analysis*, Brooks Cole, 9th edition, 2010.

[22] R. Burden and D. Faires, *Numerical Analysis*, John Wiley & Sons, New York, NY, USA, 1989.

[23] V. Carey, D. Estep, and S. Tavener, "A posteriori analysis and adaptive error control for operator decomposition solution of coupled semilinear elliptic systems," *International Journal for Numerical Methods in Engineering*, vol. 94, no. 9, pp. 826–849, 2013.