1997

Solving Composite Problems with Interface Relaxation

Mo Mu

John R. Rice
Purdue University, jrr@cs.purdue.edu

Report Number:
97-029

SOLVING COMPOSITE PROBLEMS WITH INTERFACE RELAXATION

Mo Mu
John R. Rice

Department of Computer Sciences
Purdue University
West Lafayette, IN 47907

CSD-TR 97-029
May 1997

SOLVING COMPOSITE PROBLEMS WITH INTERFACE RELAXATION

MO MU* AND JOHN R. RICE†

Abstract. This paper deals with a solution approach suitable for composite PDEs with interface conditions. We present a general framework based on interface relaxation which provides a uniform platform for building problem solving environments through efficient software integration and for implementing various relaxation schemes. Mathematically, this framework contains many existing domain decomposition methods and also allows the extension to a variety of new relaxers. In particular, we describe a class of relaxers which are suitable for very general and complicated interface conditions. Interface relaxation is more general than the traditional domain decomposition methods in that it allows unrelated PDE problems on different subdomains. Convergence analysis, error estimates and preconditioning strategies are presented which show that these relaxers are competitive with existing domain decomposition methods for model problems involving a single PDE. We present experimental results which demonstrate the wide applicability of this approach. Differences between this approach and other domain decomposition methods are also discussed.

Key words. composite PDEs, interface relaxation, problem solving environment, software integration, convergence, approximation, preconditioning, domain decomposition.

AMS(MOS) subject classifications. 65N55, 65F10, 65Y05

1. Introduction. Many partial differential equation (PDE) problems can be represented in the composite context where there are individual PDEs defined on subdomains locally, with interface conditions defined on the subdomain interfaces and boundary conditions imposed on the boundary of the global domain. We call these *composite* PDEs. For example, in the Schwarz splitting methods [5, 9, 15], one splits a global PDE problem

$$\begin{cases} Lu = f & \text{in } \Omega, \\ u = 0 & \text{on } \partial\Omega \end{cases} \tag{1.1}$$

into subdomain problems

$$\begin{cases} Lu_i = f & \text{in } \Omega_i \text{ for } i = 1, 2, \ldots, k, \\ u_i = u_j & \text{on } \Gamma_{ij}, \\ \partial_{\nu_i} u_i + \partial_{\nu_j} u_j = 0 & \text{on } \Gamma_{ij}, \\ u_i = 0 & \text{on } \partial\Omega \cap \partial\Omega_i, \end{cases} \tag{1.2}$$

* Department of Mathematics, Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong, mu@math.ust.hk, and Computer Sciences Department, Purdue University, West Lafayette, IN 47907 U.S.A., mu@cs.purdue.edu. Work supported in part by Hong Kong Competitive Earmarked Research Grant HKUST593/94E.

† Computer Sciences Department, Purdue University, West Lafayette, IN 47907 U.S.A., rice@cs.purdue.edu. Work supported in part by NSF awards CCR9202536 and CDA 9123502, ARPA ARO award DAAH04-94-G-0010.

where Γ_{ij} is the interface between two adjacent subdomains Ω_i and Ω_j, and $\partial_{\nu_i} u_i$ is the flux in the outer normal direction. Another example is the model for a Josephson junction window of different superconducting films [3, 8]. Let Ω_{in} be a region of a Josephson junction window which is imbedded in a global domain Ω, where $\Omega_{out} = \Omega \backslash \Omega_S$ is the idle region without superconductivity. The phase difference $u(x, y)$ of the order parameter in the superconducting films satisfies the nonlinear sine-Gordon equation inside Ω_{in} and is harmonic outside:

$$\begin{cases} \nabla^2 u_{in} = \sin u_{in} & \text{in } \Omega_{in}, \\ \nabla^2 u_{out} = 0 & \text{in } \Omega_{out}. \end{cases} \tag{1.3}$$

The local solutions are subject to the interface and boundary constraints:

$$\begin{cases} u_{in} = u_{out} & \text{on } \partial\Omega_{in}, \\ \frac{1}{L_{in}} \frac{\partial u_{in}}{\partial n} = \frac{1}{L_{out}} \frac{\partial u_{out}}{\partial n} & \text{on } \partial\Omega_{in}, \\ \frac{\partial u_{out}}{\partial n} = g & \text{on } \partial\Omega, \end{cases} \tag{1.4}$$

where $L_{in} \neq L_{out}$ in general, and Ω_{in} may consist of more than one disjoint subdomains if several junction windows are used. Note that in Schwarz splitting (1.2), a single global PDE operator is used in all the subdomains while in the second example (1.3) the PDE operators are different. Other composite PDE examples are the mixed Navier-Stokes and Euler problem in aerodynamics and the gas-liquid interface problem.

Given a composite PDE, it is in general not easy to identify a single underlying global problem with a relationship like that between (1.1) and (1.2). So the global PDE-based domain decomposition methods such as the overlapping Schwarz or the substructure type methods are not applicable to general composite PDE problems. For simple interface conditions such as in (1.2), there exist various Schwarz splitting type methods which alternatively solve Dirichlet and Neumann problems on adjacent subdomains in one way or another. P. L. Lions [9] proposed a more uniform approach by using a Robin transmission condition with a convex combination of Dirichlet and Neumann data for all subdomain solvers. This idea was recently extended by J. Douglas [5] to allow for a varying parameter in Robin condition during the iteration, and the convergence rate is shown to be accelerated using an ADI approach for a model problem.

However, the interface conditions for composite PDEs may appear in more complicated forms, or even involve higher order derivatives, integrals, infinite series, and so on. One such example from grating theory [2] is the interface condition of the form:

$$(T_k - \frac{\partial}{\partial n}) u_k = (T_j - \frac{\partial}{\partial n}) u_j, \tag{1.5}$$

where T_k, T_j are operators defined on each interface of two adjacent composite optical materials in terms of the Fourier transform

$$(Tv)(y) = \sum_{n \in Z} i\beta^{(n)} v^{(n)} \exp(i\alpha_n y), \tag{1.6}$$

where $v^{(n)}$ are the Fourier coefficients of the one-dimensional function v defined on the interface, $\beta^{(n)}$ and α_n are certain constants. Thus, more general techniques are needed to handle complicated interface conditions from composite PDEs.

In this paper, we first present in Section 2 a general framework for solving composite PDEs based on interface relaxation. This framework not only contains many existing domain decomposition methods, but also allows the extension to a variety of new relaxers. It thus provides a uniform platform for building problem solving environments (PSEs) through software integration and for implementing various domain decomposition methods. In Section 3, we describe a new class of relaxers which are simple, yet suitable for very general and complicated interface conditions. Section 4 presents the convergence analysis of the interface relaxation for a simple model problem. In Section 5 the analysis is extended to some classes of non-rectangular domains and to non-uniform grid methods for solving the subdomain PDEs. Experimental results are reported in Section 6 which demonstrate the wide applicability of the interface relaxation approach. Section 7 presents a discussion of the current state of approximation error analysis for both Schwarz and interface relaxation methods. The error analysis for composite PDEs is made more difficult because of the lack of a convergence theory for the global PDE problem; there are many difficult open problems here. In Section 8 we propose a new multilevel preconditioning method suitable for interface relaxation so that the interface relaxation approach not only has wide applicability for general composite PDEs, but also is competitive with the Schwarz splitting type domain decomposition methods for model problems where both approaches are applicable.

2. Interface relaxation. We start with a general mathematical description of composite PDEs. Denote the local PDEs by

$$L_i u_i = f_i \quad \text{in } \Omega_i \quad \text{for } i = 1, 2, \ldots, k, \tag{2.1}$$

and assume that the interface conditions are specified in a general implicit form

$$g_{ij}(u_i, Du_i, \ldots, D^{N_i} u_i; u_j, Du_j, \ldots, D^{N_j} u_j) = 0 \quad \text{in } \Gamma_{ij}, \tag{2.2}$$

where D^{N_i} denotes a generic differential operator of order N_i, and g_{ij} can be a function mapping on the interface or even a functional. For example, for the smooth solution continuity conditions in (1.2) we can define

$$g_{ij} \equiv \lambda(u_i - u_j)^2 + \mu \left(\frac{\partial u_i}{\partial n} + \frac{\partial u_j}{\partial n} \right)^2 \quad \text{on } \Gamma_{ij}, \tag{2.3}$$

where λ and μ are like Lagrange multipliers. One may also consider a jump condition, say for the flux across the interface, by including the jump data J in (2.3):

$$\lambda(u_i - u_j)^2 + \mu\left(\frac{\partial u_i}{\partial n} + \frac{\partial u_j}{\partial n} - J\right)^2 = 0 \quad \text{on } \Gamma_{ij}.$$

We now present a general framework for constructing an iterative procedure to solve a composite PDE based on *interface relaxation*. Let $I(i)$ be the indices of those subdomains that are neighbors of subdomain Ω_i. Define the boundary value problem $\mathbf{P}_i^m$ that is solved on Ω_i at the mth relaxation step as

$$\text{(2.4)} \qquad \begin{cases} L_i u_i^m = f_i & \text{in } \Omega_i, \\ B_{ij}^m u_i^m = b_{ij}^m & \text{on } \Gamma_{ij} \text{ for } j \in I(i), \\ u_i^m \text{ satisfies the global boundary conditions on } \partial\Omega, \end{cases}$$

where B_{ij}^m is a boundary condition operator such that $\mathbf{P}_i^m$ is well-posed. Usually, B_{ij}^m defines a simple, standard boundary condition of the Dirichlet, Neumann, or Robin type, although g_{ij} in (2.2) may be more complicated.

We note that the interface relaxation iteration (2.4) is defined on subdomains independently. Details of the iteration are separated from the subdomain PDE solvers and specified by an interface handler, called a *relaxer*. It provides for the interface Γ_{ij} to subdomain Ω_i the right hand side data b_{ij}^m of the boundary condition according to certain relaxation procedures as well as the parameters in the definition of B_{ij}^m. Different choices of the boundary condition operator B_{ij}^m and the relaxation scheme lead to some known domain decomposition methods. For example, choosing B_{ij} as a Dirichlet operator and B_{ji} as a Neumann operator for each piece of Γ_{ij} leads to the well-known Dirichlet-Neumann method, where $b_{ij}^m = u_j^{m-1}|_{\Gamma_{ij}}$ and $b_{ji}^m = \frac{\partial u_i^{m-1}}{\partial n}|_{\Gamma_{ij}}$ are the corresponding relaxation formula used by the relaxer. Similarly, Lions' method comes from taking both B_{ij} and B_{ji} as Robin operators and the relaxer is correspondingly defined with the same Robin condition. In these examples, the boundary operations B_{ij} are called *stationary* for they remain unchanged during the iteration. They can be allowed to vary, as in Douglas' method, where the relaxer passes parameters depending on m to be used by the subdomain solvers in the Robin conditions.

This definition provides interface relaxation with a uniform and convenient framework for software integration. The paradigm is compatible with current computer technologies such as object-orientation, software reuse, agent-based systems [6], distributed computing, etc. Each local solver (agent) on Ω_i receives from the relaxer (mediator) the boundary data b_{ij}^m as well as the boundary operator parameters for B_{ij}^m as input. The agent independently solves a local PDE problem $\mathbf{P}_i^m$ which is usually simple and standard, and thus can be done by invoking an existing software part from a PDE solver library or over the network using a PDE solving server. It then feeds back to the relaxer the boundary information of the newly computed local solution u_i^m. The relaxer then uses the information received from the neighboring subdomains for each piece of interface Γ_{ij} to compute the new boundary data b_{ij}^{m+1} and b_{ji}^{m+1} for Ω_i and Ω_j, respectively, for the next iteration. This process iterates until convergence. There is a clear separation between the local solvers and the relaxers so that they do not know the details of each other at all. This approach has been

implemented as an agent-based software system *SciAgents* [6] which allows users to solve two-dimensional composite PDEs with collaborating PDE solvers on distributed networks and test various relaxers easily and flexibly.

Computationally, the central task is to select a proper relaxation formula for a given interface condition. A variety of relaxers (see Section 3) have been used experimentally (see Section 6) and some of them work rather well for fairly complicated physical systems. Mathematically, the challenge is to show the convergence of both the interface relaxation and the approximation to the PDE solution. Acceleration techniques for the relaxation are needed to improve the computational efficiency as in all iteration methods.

3. A new class of relaxers. In this section, we discuss methodologies for devising relaxers suitable for a general interface condition as defined in (2.2). Let us start with proposing a simple relaxer as follows. First, we choose both B_{ij} and B_{ji} as Dirichlet operator. We denote

$$g_{ij}(u_i^m, Du_i^m, \ldots, D^{N_i}u_i^m; u_j^m, Du_j^m, \ldots, D^{N_j}u_j^m)$$

by g_{ij}^m and view it as a residual on the interface Γ_{ij} at the mth relaxation step. Following Southwell's relaxation idea we define the new Dirichlet data on Γ_{ij} as

$$b_{ij}^{m+1} = b_{ji}^{m+1} \equiv b_{ij}^m + \omega_g g_{ij}^m, \tag{3.1}$$

where ω_g is a relaxation parameter like that used in the pointwise SOR. This simple procedure defines an iteration which is different from existing domain decomposition methods, but applicable to the general interface condition form (2.2). First, we note that this is not the conventional block SOR version in the substructure approach because there is no PDE discretization on the interfaces as in the global PDE case. Secondly, unlike in other methods where the Neumann data are passed across interfaces by solving a Neumann or Robin problem, we only solve Dirichlet problems on all subdomains and the Neumann data are involved in the evaluation of the residual for relaxation. This makes it feasible for general and complicated interface conditions where other methods cannot apply. All it requires is the function evaluation of g_{ij}. Although in the model problem case the iteration is slower than other methods due to the treatment of the Neumann data, we will show that preconditioners can be constructed to make it competitive with others.

To be more specific and to construct a model problem for the analysis for this relaxer, let us further simplify the geometry Ω as in Fig. 3.1 and denote $\Gamma_{i,i+1}$ simply by Γ_i. We shall first prove the convergence for a special case where Ω is a rectangle and then describe how the analysis can be extended to an even more general composite domain with interior cross points as shown in Fig. 3.2.

Without loss of generality, we assume that the global solution vanishes on $\partial\Omega$, and the interior interface condition is smooth solution condition (2.3). In the simplified notation, the solutions on both sides of any interface Γ_i have the same boundary values on Γ_i at each iteration. The interface condition (2.3) is then reduced to

$$g_{\Gamma_i}(\frac{\partial u_i}{\partial x}, \frac{\partial u_{i+1}}{\partial x}) \equiv \frac{\partial u_i}{\partial x} - \frac{\partial u_{i+1}}{\partial x} = 0, \text{on } \Gamma_i \quad \text{for } i = 1, 2, \ldots, k-1. \tag{3.2}$$

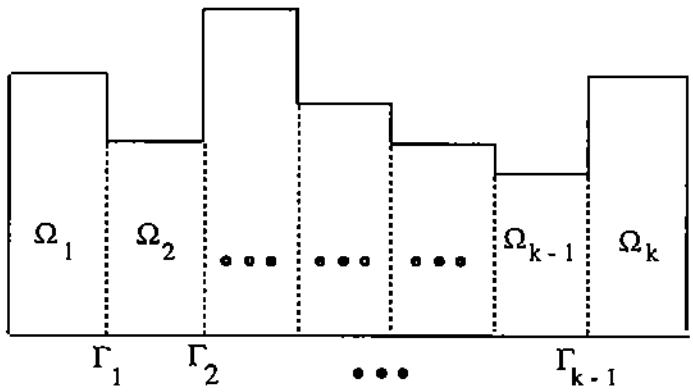

FIG. 3.1. *A "one dimensional" composite domain* Ω.

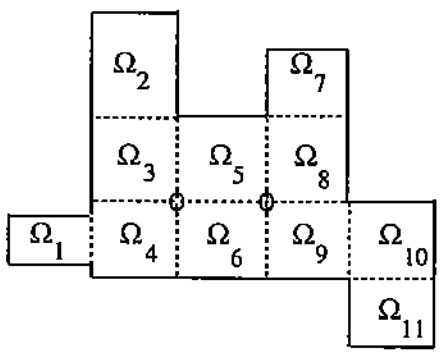

FIG. 3.2. *A general "two dimensional" composite domain with interior cross points (marked by "circles").*

In principle, one can apply any numerical method, such as finite differences, finite elements, collocation, or spectral method in each local PDE solver. The corresponding discrete systems can be generally written as

$$\text{(3.3)} \qquad \begin{cases} A_i U_i^m = f_i + P_{\Omega_i,\Gamma_{i-1}} X_{i-1}^m + P_{\Omega_i,\Gamma_i} X_i^m & \text{for } i = 1, 2, \ldots, k \\ X_0^m \equiv X_k^m \equiv 0 \end{cases}$$

where the matrices A_i, $P_{\Omega_i,\Gamma_{i-1}}$ and P_{Ω_i,Γ_i} correspond to the discretization of the PDE operator L_i in the interior and next to the boundary pieces Γ_{i-1} and Γ_i of the subdomain Ω_i; U_i^m denotes the discrete solution of u_i^m and $U_i^m|_{\Gamma_i} = U_{i+1}^m|_{\Gamma_i} = X_i^m$.

Correspondingly, the relaxation formula (3.1) becomes:

$$\text{(3.4)} \qquad X_i^{m+1} = X_i^m + \omega_g\left(\frac{\partial u_i^m}{\partial x}|_{\Gamma_i} - \frac{\partial u_{i+1}^m}{\partial x}|_{\Gamma_i}\right).$$

Choosing the finite difference setting, for instance, we approximate $\frac{\partial u_i^m}{\partial x}|_{\Gamma_i}$ and $\frac{\partial u_{i+1}^m}{\partial x}|_{\Gamma_i}$ by $\frac{U_i^m|_{\Gamma_i} - U_i^m|_{\Gamma_i^-}}{h_i^-}$ and $\frac{U_{i+1}^m|_{\Gamma_i^+} - U_{i+1}^m|_{\Gamma_i}}{h_i^+}$, respectively, where $\Gamma_i^\pm$ denote the neighboring grid lines of Γ_i and $h_i^\pm$ denote the corresponding spacings between Γ_i and $\Gamma_i^\pm$. A discrete version of (3.4) then reads as

$$\text{(3.5)} \qquad X_i^{m+1} = \omega_i X_i^m + (1-\omega_i)(\alpha_i^- U_i^m|_{\Gamma_i^-} + \alpha_i^+ U_{i+1}^m|_{\Gamma_i^+}),$$

where $\alpha_i^- = \frac{h_i^+}{h_i^- + h_i^+}$, $\alpha_i^+ = \frac{h_i^-}{h_i^- + h_i^+}$ and $\omega_i = 1 + \omega_g(\frac{1}{h_i^-} + \frac{1}{h_i^+})$.

To obtain an iterative relation on the interfaces, we combine solving (3.3) for $\{U_i^m\}_{i=1}^k$ with (3.5), which leads to a matrix representation of $\{X_i^{m+1}\}$ in terms of $\{X_i^m\}$. The convergence analysis of the relaxation process is thus reduced to the spectral analysis of the corresponding iteration matrix. More specifically, denote by $P_{\Gamma_{i-1},\Omega_i}$ and P_{Γ_i,Ω_i} the matrices corresponding to the linear operators that restrict the solution in Ω_i to the grid lines next to Γ_{i-1} and Γ_i, respectively. Then, from (3.5) and (3.3) we have

$$\begin{aligned} X_i^{m+1} &= \omega_i X_i^m + (1-\omega_i)(\alpha_i^- P_{\Gamma_i,\Omega_i} U_i^m + \alpha_i^+ P_{\Gamma_i,\Omega_{i+1}} U_{i+1}^m) \\ &= \omega_i X_i^m + (1-\omega_i)\alpha_i^- P_{\Gamma_i,\Omega_i} A_i^{-1}(f_i + P_{\Omega_i,\Gamma_{i-1}} X_{i-1}^m + P_{\Omega_i,\Gamma_i} X_i^m) \\ &\quad + (1-\omega_i)\alpha_i^+ P_{\Gamma_i,\Omega_{i+1}} A_{i+1}^{-1}(f_{i+1} + P_{\Omega_{i+1},\Gamma_i} X_i^m + P_{\Omega_{i+1},\Gamma_{i+1}} X_{i+1}^m). \end{aligned} \qquad \text{(3.6)}$$

Introduce the vector $X = (X_1, X_2, \ldots, X_{k-1})$ of interface values and (3.6) can be written in the matrix form

$$\text{(3.7)} \qquad X^{m+1} = MX^m + G$$

where G is a constant vector corresponding to $\{f_i\}$ and $M = [B_i, D_i, C_i]$ is a $(k-1) \times (k-1)$ block tridiagonal matrix with

$$
\begin{aligned}
D_i &= \omega_i + (1-\omega_i)\alpha_i^- P_{\Gamma_i,\Omega_i} A_i^{-1} P_{\Omega_i,\Gamma_i} + (1-\omega_i)\alpha_i^+ P_{\Gamma_i,\Omega_{i+1}} A_{i+1}^{-1} P_{\Omega_{i+1},\Gamma_i} \\
&\quad for\ i = 1, 2, \ldots, k-1, \\
B_i &= (1-\omega_i)\alpha_i^- P_{\Gamma_i,\Omega_i} A_i^{-1} P_{\Omega_i,\Gamma_{i-1}} \quad for\ i = 2, \ldots, k-1, \\
C_i &= (1-\omega_i)\alpha_i^+ P_{\Gamma_i,\Omega_{i+1}} A_{i+1}^{-1} P_{\Omega_{i+1},\Gamma_{i+1}} \quad for\ i = 1, \ldots, k-2.
\end{aligned} \tag{3.8}
$$

Therefore, the convergence of the iteration with the interface relaxation is equivalent to

$$
\rho(M) < 1 \tag{3.9}
$$

where $\rho(\cdot)$ denotes the spectral radius.

We remark on other possible extensions of the relaxer construction. For example, instead of explicitly updating the boundary data b_{ij} as in (3.1), one can implicitly determine b_{ij}^{m+1} by fixing other arguments in g_{ij} and solving the equation:

$$
g_{ij}(u_i^{m+1}, Du_i^m, \ldots, D^{N_i} u_i^m; u_j^{m+1}, Du_j^m, \ldots, D^{N_j} u_j^m) = 0, \tag{3.10}
$$

in order to relax the interface condition. In many situations when the interface condition is nonlinear, (3.10) cannot be solved exactly, and techniques like least-squares and Newton iteration may be applied to generate an approximation to b_{ij}^{m+1}. Similarly, one can also relax for the Neumann data by fixing the Dirichlet data and/or other terms involved in the interface condition. Finally, we comment on that the same idea can be applied to the case when g_{ij} is a functional and one can relax for certain data by a minimization procedure keeping other data fixed. This is similar to the least-squares and Newton relaxers above. Therefore, we can, in principle, handle any type of interface conditions within the interface relaxation framework.

4. Convergence analysis for a rectangular domain. We now present the model problem convergence analysis for the relaxation (3.1). In this section we consider the special case where Ω is a rectangle. In this case, the full spectrum of the matrix M can be obtained by the Fourier analysis so that the convergence mechanism is clearly understood. In addition, the optimal relaxation parameter analysis can also be performed directly. In the next section we prove the convergence for non-rectangular cases by a different argument.

We start with some model problem assumptions. Let the PDE operators L_i be Laplacian $(-\Delta)$ in all subdomains, and assume that Ω is simply a rectangle, i.e., all subdomains in Fig. 3.1 have the same size. Each subdomain is discretized by a uniform tensor product grid with m vertical and n horizontal interior grid lines and a spacing h. The PDE operator is discretized by the *5-point-star* finite differences, or equivalently the Courant finite elements, and the unknowns/equations are ordered using the natural indexing. The assumptions and analysis can be generalized to a separable and self-adjoint elliptic operator and a nonuniform grid. However, the analysis looks much more complicated even using essentially the same techniques, see [12].

Under the model problem assumptions, we have $A_i \equiv \frac{1}{h^2}A$, for $i = 1, 2, \ldots, k$, where $A = [-I, T, -I]$ is an $m \times m$ block tridiagonal matrix with $T = [-1, 4, -1]$ being an $n \times n$ tridiagonal matrix; $h^2 P^T_{\Omega_i, \Gamma_{i-1}} \equiv P_{\Gamma_{i-1}, \Omega_i} \equiv V^T = [I, 0, \ldots, 0]^T$ and $h^2 P^T_{\Omega_i, \Gamma_i} \equiv P_{\Gamma_i, \Omega_i} \equiv W^T = [0, \ldots, 0, I]^T$ are $1 \times m$ block matrices with I denoting the $n \times n$ identity matrix; and, finally, $\alpha_i^- \equiv \alpha_i^+ = \frac{1}{2}$ and $\omega_i = 1 + \frac{2\omega_o}{h} \equiv \omega$ for $i = 1, \ldots, k-1$. Then, relations in (3.8) are reduced to

$$
\begin{aligned}
D \equiv D_i &= \omega + \tfrac{(1-\omega)}{2}(V^T A^{-1} V + W^T A^{-1} W) \quad \text{for } i = 1, 2, \ldots, k-1 \\
C \equiv C_i \equiv B^T_{i+1} &= \tfrac{(1-\omega)}{2} V^T A^{-1} W \qquad\qquad \text{for } i = 1, 2, \ldots, k-2.
\end{aligned} \tag{4.1}
$$

LEMMA 4.1. *The matrices D and C can be expressed as functions of the matrix T as follows:*

$$
D = d(T), \quad C = c(T), \tag{4.2}
$$

where the scalar functions $d(t)$ and $c(t)$ are defined as

$$
d(t) = \omega + \frac{(1-\omega)}{2}\left(t - \frac{c_m(t)}{s_{m-1}(t)}\right), \quad c(t) = \frac{(1-\omega)}{2 s_{m-1}(t)}, \tag{4.3}
$$

and $s_m(t)$ and $c_m(t)$ are Chebyshev polynomials defined by

$$
\begin{aligned}
s_m(t) &= \tfrac{\eta^{m+1} - \eta^{-(m+1)}}{\eta - \eta^{-1}} \\
c_m(t) &= \eta^m + \eta^{-m}
\end{aligned}
\qquad \eta = \frac{t}{2} + \sqrt{(\frac{t}{2})^2 - 1} \quad \text{for } t > 2. \tag{4.4}
$$

Proof. Observe that the matrix $S \equiv T - V^T A^{-1} V - W^T A^{-1} W$ is the two-subdomain Schur complement on the interface. From [1] we have

$$
\begin{aligned}
S &= s_{m-1}^{-1}(T) c_m(T), \\
V^T A^{-1} W &= s_{m-1}^{-1}(T).
\end{aligned} \tag{4.5}
$$

Lemma 4.1 then immediately follows. □

LEMMA 4.2. *The eigenvalues of the matrix M can be expressed as*

$$
\lambda_{ij} = \omega + \tfrac{(1-\omega)}{2} q_{ij} \quad \text{for } i = 1, 2, \ldots, k-1, \quad j = 1, 2, \ldots, n \tag{4.6}
$$

with

$$
\begin{aligned}
q_{ij} &= t_j - \frac{\eta_j^m + \eta_j^{-m} - 2\cos\frac{i\pi}{k}}{\eta_j^m - \eta_j^{-m}}(\eta_j - \eta_j^{-1}), \\
\eta_j &= \tfrac{t_j}{2} + \sqrt{(\tfrac{t_j}{2})^2 - 1},
\end{aligned} \tag{4.7}
$$

where

$$
t_j = 2 + 4\sin^2\frac{j\pi}{2(n+1)} \qquad \textit{for } j = 1, 2, \ldots, n \tag{4.8}
$$

are eigenvalues of the matrix T.

Proof. Let $p(\lambda)$ be the eigenpolynomial of M and $T = Q^T \Lambda Q$ be the eigendecomposition of T. Then from (4.2) we have

$$
\begin{aligned}
p(\lambda) &= \det(M - \lambda I) \\
&= \det[C^T, D - \lambda I, C] \\
&= \det[c(T), d(T) - \lambda I, c(T)] \qquad (4.9) \\
&= (\det(Q))^{2(k-1)} \det[c(\Lambda), d(\Lambda) - \lambda I, c(\Lambda)] \\
&= (\det(Q))^{2(k-1)} \textstyle\prod_{j=1}^{n} \det[c(t_j), d(t_j) - \lambda, c(t_j)].
\end{aligned}
$$

Thus, the eigenvalues of M are also the eigenvalues of the $(k-1)\times(k-1)$ tridiagonal matrices $[c(t_j), d(t_j), c(t_j)]$ for $j = 1, 2, \ldots, n$, which, in turn, can be expressed as

$$
\lambda_{ij} = d(t_j) - 2c(t_j)\cos\frac{i\pi}{k} \qquad \textit{for } i = 1, 2, \ldots, k-1, \quad j = 1, 2, \ldots, n. \tag{4.10}
$$

This, combined with (4.3), establishes Lemma 4.2. □

LEMMA 4.3. *For any* $1 \le i \le k-1$, $1 \le j \le n$ *and* $m > 1$, *we have*

$$
0 < q_{ij} < 2. \tag{4.11}
$$

Proof. Observe that $t_j = \eta_j + \eta_j^{-1}$, so q_{ij} can be rewritten as

$$
q_{ij} = \frac{2(\eta_j^{2m} - \eta_j^2 + \eta_j^2\cos\frac{i\pi}{k} - \cos\frac{i\pi}{k})}{\eta_j(\eta_j^{2m} - 1)} \tag{4.12}
$$

Because $t_j > 2$, we have $\eta_j > 1$ for $j = 1, 2, \ldots, n$. Therefore, to prove $q_{ij} > 0$, it suffices to show that

$$\eta^{2m} - \eta^2 + \eta^2 \cos y - \cos y > 0 \qquad \text{for } \eta > 1 \tag{4.13}$$

where $0 < y < \pi$. This follows by directly applying the standard calculus computation to verify that the left hand side, as a function of η, and its first three derivatives are increasing functions. Then a check of the boundary values at $\eta = 1$ establishes (4.13). Similarly, to prove $q_{ij} < 2$, one shows that

$$\eta^{2m} - \eta^2 + \eta^2 \cos y - \cos y < \eta(\eta^{2m} - 1) \tag{4.14}$$

or, equivalently,

$$\eta^{2m+1} - \eta^{2m} + \eta^2 - \eta^2 \cos y - \eta + \cos y > 0. \tag{4.15}$$

Inequality (4.15) then follows by the same argument as used for (4.13). This completes the proof of Lemma 4.3. □

We are now at the position to prove the theorem on the convergence of the relaxation and to determine the optimal iteration parameter for the class of relaxers. Let ω^+_{opt}, ω^-_{opt} be the optimal positive and negative iteration parameters, respectively, and let ρ^+_{opt}, ρ^-_{opt} be the corresponding values of $\rho(M)$.

THEOREM 4.4. *Let* $q_{\max} = \max_{ij} q_{ij}$, *and* $q_{\min} = \min_{ij} q_{ij}$. *Then we have*

(4.16)(1)
$$\rho(M) = \begin{cases} \omega + \frac{(1-\omega)}{2} q_{\min} & \text{for } \omega \geq 1, \\ \omega + \frac{(1-\omega)}{2} q_{\max} & \text{for } 0 \leq \omega \leq 1, \\ \max\left\{ \left|\omega + \frac{(1-\omega)}{2} q_{\max}\right|, \left|\omega + \frac{(1-\omega)}{2} q_{\min}\right| \right\} & \text{for } \omega \leq 0. \end{cases}$$

(2)
$$\min_{\omega \geq 0} \rho(M) = \rho(M)|_{\omega = \omega^+_{opt}} = \frac{q_{\max}}{2} \left(\equiv \rho^+_{opt} \right),$$

$$\omega^+_{opt} = 0;$$

(4.17)
$$\min_{\omega < 0} \rho(M) = \rho(M)|_{\omega = \omega^-_{opt}} = \frac{q_{\max} - q_{\min}}{4 - (q_{\max} + q_{\min})} \left(\equiv \rho^-_{opt} \right),$$

$$\omega^-_{opt} = -\frac{q_{\max} + q_{\min}}{4 - (q_{\max} + q_{\min})}.$$

(3)
$$\rho^-_{opt} < \rho^+_{opt} < 1;$$

$$
\begin{aligned}
\rho_{opt}^{+} &= 1 + c_{\alpha,k} h + O\left(h^2\right); \\
\frac{\rho_{opt}^{+}}{\rho_{opt}^{-}} &\approx 1 + \frac{2 - q_{max}}{2} = 1 + c_{\alpha,k} h + O\left(h^2\right),
\end{aligned} \tag{4.18}
$$

where the constant factor $c_{\alpha,k}$ depends on the number, k, of subdomains and the aspect ratio, $\alpha = \frac{m}{n}$, of each subdomain. The exact expression of $c_{\alpha,k}$ is given by (4.23) in the following proof of the theorem.

Proof. From Lemma 4.2 we have

$$
\rho(M) = \max_{i,j} |\omega + \frac{(1-\omega)}{2} q_{ij}|. \tag{4.19}
$$

If $0 \leq \omega \leq 1$, recall that $q_{ij} > 0$, we get

$$
\begin{aligned}
\rho(M) &= \max_{ij}(\omega + \tfrac{(1-\omega)}{2} q_{ij}) \\
&= \omega + \tfrac{(1-\omega)}{2} \max_{ij} q_{ij} \\
&= \omega + \tfrac{(1-\omega)}{2} q_{\max}.
\end{aligned}
$$

If $\omega \geq 1$, we can rewrite (4.19) as

$$
\rho(M) = \max_{ij} |\frac{q_{ij}}{2} + \frac{\omega}{2}(2 - q_{ij})|.
$$

Recall that $0 < q_{ij} < 2$ so we have

$$
\begin{aligned}
\rho(M) &= \max_{ij} \left(\tfrac{q_{ij}}{2} + \tfrac{\omega}{2}(2 - q_{ij})\right) \\
&= \max_{ij} \left(\omega + \tfrac{(1-\omega)}{2} q_{ij}\right) \\
&= \omega + \tfrac{(1-\omega)}{2} \min_{ij} q_{ij} \\
&= \omega + \tfrac{(1-\omega)}{2} q_{\min}.
\end{aligned}
$$

If $\omega \leq 0$, we can view $\omega + \frac{(1-\omega)}{2} q_{ij}$ as points on a linear function of q: $y = \omega + \frac{(1-\omega)}{2} q$. So, the maximum is reached at one of the end points. That is,

$$
\rho(M) = \max\{|\omega + \frac{(1-\omega)}{2} \max_{ij} q_{ij}|, \ \ |\omega + \frac{(1-\omega)}{2} \min_{ij} q_{ij}|\}.
$$

In fact, the formulas for $\omega \geq 0$ can also be viewed as two special cases of this general one. The proof of (4.16) is complete.

From (4.16), it is easy to see that

$$\begin{aligned} \min_{\omega \geq 1} \rho(M) &= \min_{\omega \geq 1} \left(\tfrac{q_{\min}}{2} + \tfrac{\omega}{2}(2 - q_{\min})\right) \\ &= \left(\tfrac{q_{\min}}{2} + \tfrac{\omega}{2}(2 - q_{\min})\right)|_{\omega=1} \\ &= 1; \end{aligned} \tag{4.20}$$

and

$$\begin{aligned} \min_{0 \leq \omega \leq 1} \rho(M) &= \min_{0 \leq \omega \leq 1} \left(\tfrac{q_{\max}}{2} + \tfrac{\omega}{2}(2 - q_{\max})\right) \\ &= \left(\tfrac{q_{\max}}{2} + \tfrac{\omega}{2}(2 - q_{\max})\right)|_{\omega=0} \\ &= \tfrac{q_{\max}}{2}, \end{aligned} \tag{4.21}$$

which gives the first part of (4.17). Assume $\omega \leq 0$, then to minimize $\max\{|\omega + \frac{(1-\omega)}{2} q_{\max}|;\ |\omega + \frac{(1-\omega)}{2} q_{\min}|\}$ we know that ω_{opt}^- has to satisfy the following equation:

$$|\omega_{opt}^- + \frac{(1 - \omega_{opt}^-)}{2} q_{\max}| = |\omega_{opt}^- + \frac{(1 - \omega_{opt}^-)}{2} q_{\min}|.$$

Solving this equation gives the second part of (4.17).

From (4.17) we have

$$\begin{aligned} \frac{\rho_{opt}^+}{\rho_{opt}^-} &= \frac{q_{\max}(1-(q_{\max}+q_{\min}))}{2(q_{\max}-q_{\min})} \\ &= \frac{4(q_{\max}-q_{\min}) - q_{\max}^2 - q_{\max}q_{\min} + 4q_{\min}}{2(q_{\max}-q_{\min})} \\ &= 2 - \frac{q_{\max}^2 + q_{\max}q_{\min} - 4q_{\min}}{2(q_{\max}-q_{\min})}. \end{aligned} \tag{4.22}$$

Recall that $q_{\max} < 2$, so (4.22) then implies that

$$\frac{\rho_{opt}^+}{\rho_{opt}^-} > 2 - \frac{2q_{\max} + 2q_{\min} - 4q_{\min}}{2(q_{\max} - q_{\min})} = 1,$$

which, plus the fact that $\rho_{opt}^+ < 1$, proves the first relation of (4.18). Observe that $q_{\max}$ corresponds to $\eta_{\min}$, which, in turn, corresponds to $t_{\min}$. Since $t_{\min} = 2 + O(h^2)$, we have $\eta_{\min} = 1 + \delta/n + O(h^2)$ and $\eta_{\min}^n = e^{n log \eta_{\min}} = e^{\delta}(1 + O(h))$, where δ is a constant. We rewrite the expression (4.12) for $q_{\max}$ as

$$q_{\max} = 2\left(1 + \frac{(\cos\frac{\pi}{k} - 1)(\eta_{\min}^2 - 1)}{\eta_{\min}^{2m} - 1}\right) / \eta_{\min}.$$

Using the Taylor's expansion for it, we get

$$q_{\max} = 2\left(1+\frac{(\cos\frac{\pi}{k}-1)(2\delta h+O(h^2))}{(e^{\delta\alpha}-1)(1+O(h))}\right)(1-\delta h+O(h^2)),$$

which yields the second relation of (4.18) with

$$c_{\alpha,k} = \left(\frac{2(\cos\frac{\pi}{k}-1)}{e^{\delta\alpha}-1}-1\right)\delta. \tag{4.23}$$

Finally, denoting $\epsilon = 2 - q_{\max}$ and using (4.22), we have

$$\begin{aligned}\frac{\rho^+_{opt}}{\rho^-_{opt}} &= 1+\tfrac{\epsilon}{2}+\tfrac{\epsilon q_{\min}}{2-(\epsilon+q_{\min})} \\ &= 1+\epsilon[\tfrac{1}{2}+\tfrac{q_{\min}}{2-(\epsilon+q_{\min})}].\end{aligned} \tag{4.24}$$

Since $q_{\min} = O(h)$, it is easy to obtain the third relation of (4.18) from (4.24). The proof of Theorem 4.4 is thus complete. □

Theorem 4.4 states that the relaxation process diverges for all $\omega \geq 1$, and converges for $0 \leq \omega < 1$ with the optimal positive parameter at $\omega = 0$, which corresponds to the relaxation formula (3.1) with $\omega_g = -h/2$. In other words, any nonnegative parameter ω does not accelerate the convergence at all. However, using a negative parameter ω may accelerate the convergence if the optimal relaxation parameter is chosen properly. In addition, the optimal convergence rates depend on h and approach 1 linearly as the spacing h approaches 0, and the coefficient factors depend on the number k of subdomains from the term $\cos\frac{\pi}{k}$, and on the aspect ratio α of each subdomain from the term $(e^{\delta\alpha}-1)^{-1}$. These results agree with the convergence behavior for many domain decomposition methods. We will discuss the preconditioning strategies in Section 8 to accelerate the convergence rate of the relaxation to an h-independent rate.

5. Convergence analysis for nonrectangular domains. This section extends the convergence analysis to nonrectangular domains as in Figures 3.1 and 3.2. For the sake of simple notation, we first consider the case of Fig. 3.1 and then show that the convergence theorem also holds for the case of Fig. 3.2.

We first notice that the linear operator relations (3.7) and (3.8) are true for the case of Fig. 3.1 with proper matrix representations for P_{Γ_i,Ω_j}, P_{Ω_j,Γ_i}, and A_i, as we display later on. To prove (3.9) for convergence, it suffices to show, using the Rayleigh principle, that

$$X^T M X < X^T X \quad \text{for all } X \neq 0. \tag{5.1}$$

One key idea in our argument is to change the natural *interface-based* analysis as involved in (5.1) to the *subdomain-based* analysis, which then allows us to further extend the convergence analysis to general composite geometric domains no matter

how interfaces are related to subdomains. For simplicity of notation, from now on we assume tha $\omega = 0$ and a uniform spacing h is used so that $\alpha_i^+ \equiv \alpha_i^- \equiv \frac{1}{2}$.

LEMMA 5.1. *With the convention that $X_0 \equiv X_k \equiv 0$, for the expression of the left hand side of inequality (5.1) we have*

$$X^T M X = \frac{1}{2} \sum_{i=1}^{k} v_i, \tag{5.2}$$

with

$$v_i = [X_{i-1}^T, X_i^T] M_i \begin{bmatrix} X_{i-1} \\ X_i \end{bmatrix}, \tag{5.3}$$

where

$$M_i = \begin{bmatrix} P_{\Gamma_{i-1},\Omega_i} A_i^{-1} P_{\Omega_i,\Gamma_{i-1}} & P_{\Gamma_{i-1},\Omega_i} A_i^{-1} P_{\Omega_i,\Gamma_i} \\ P_{\Gamma_i,\Omega_i} A_i^{-1} P_{\Omega_i,\Gamma_{i-1}} & P_{\Gamma_i,\Omega_i} A_i^{-1} P_{\Omega_i,\Gamma_i} \end{bmatrix}. \tag{5.4}$$

Proof. The proof of Lemma 5.1 is done by simply using (3.8) and regrouping terms in the summation for the expression of $X^T M X$. This completes the proof. □

Let $m_i(n_i)$ be the number of interior vertical (horizontal) grid lines in Ω_i, and let l_j be the number of interior grid points on Γ_j. We have

$$l_j \leq n_i \quad \text{for } j = i-1 \text{ and } i, \tag{5.5}$$

because the interfaces Γ_{i-1} and Γ_i are parts of the vertical boundary pieces of Ω_i. Then for the 5-point star stencil, the four "P" operators in M_i have block matrix representations when ordered according to vertical grid lines and with "0" corresponding to a group of $m_i - 1$ lines,

$$\begin{aligned} h^2 P_{\Omega_i,\Gamma_{i-1}}^T \equiv P_{\Gamma_{i-1},\Omega_i} &= [H_{i,i-1}^T, 0]^T, \\ h^2 P_{\Omega_i,\Gamma_i}^T \equiv P_{\Gamma_i,\Omega_i} &= [0, H_{i,i}^T]^T; \end{aligned} \tag{5.6}$$

where the $n_i \times l_j$ matrix $H_{i,j}$ has the form

$$H_{i,j} = \begin{bmatrix} 0 \\ I_j \end{bmatrix} \quad \text{for } j = i-1,\ i \tag{5.7}$$

with I_j being the $l_j \times l_j$ identity matrix.

LEMMA 5.2. *For each subdomain Ω_i, we have*

$$\rho(M_i) < 1, \qquad i = 1, 2, \ldots, k. \tag{5.8}$$

Proof. Let $\{A_i^{-1}\}_{\alpha,\beta}$ denote the block at the position (α, β) in the corresponding block partion of A_i^{-1}, we have

$$\begin{aligned} M_i &= \begin{bmatrix} H_{i,i-1}^T & 0 \\ 0 & H_{i,i}^T \end{bmatrix} \begin{bmatrix} \{A_i^{-1}\}_{1,1} & \{A_i^{-1}\}_{1,m_i} \\ \{A_i^{-1}\}_{m_i,1} & \{A_i^{-1}\}_{m_i,m_i} \end{bmatrix} \begin{bmatrix} H_{i,i-1} & 0 \\ 0 & H_{i,i} \end{bmatrix} \\ &\equiv \tilde{H}_i^T \tilde{M}_i \tilde{H}_i. \end{aligned} \tag{5.9}$$

From Theorem 2.1 in [1], we can express $\tilde{M}_i$ as

$$\tilde{M}_i = \begin{bmatrix} f_i(T_i) & g_i(T_i) \\ g_i(T_i) & f_i(T_i) \end{bmatrix}, \tag{5.10}$$

where $T_i = [-1, 4, -1]_{n_i \times n_i}$, $f_i(t) = s_{m_i-1}(t)/s_{m_i}(t)$ and $g_i(t) = 1/s_{m_i}(t)$. Therefore, the eigenvalues of $\tilde{M}_i$ are given by

$$\begin{aligned} \tilde{\lambda}_{ij} &= f_i(t_j) \pm g_i(t_j) \\ &= \frac{s_{m_i-1}(t_j) \pm 1}{s_{m_i}(t_j)}, \qquad t_j \in \sigma(T_i). \end{aligned}$$

Similar to the argument in the proof of Lemma 4.3, one can easily verify that

$$0 < \tilde{\lambda}_{ij} < 1. \tag{5.11}$$

With the notations used in (5.7) and (5.9), we observe that M_i is simply the matrix expanded from a principal submatrix of $\tilde{M}_i$. So, we have

$$0 \leq \sigma(M_i) < 1. \tag{5.12}$$

This is another key idea that enables the extension of the convergence analysis to cases where an interface can be a portion of a vertical or horizontal boundary piece for a subdomain. The proof of the lemma is thus complete. □

THEOREM 5.3. *The relaxation process converges in the case of Fig. 3.1. An upper bound on the convergence rate is*

$$\rho = \max_{1 \leq j \leq k-1} \left[\frac{\rho(M_j) + \rho(M_{j+1})}{2} \right]. \tag{5.13}$$

Proof. From (5.3) and (5.12) we have

$$0 \le v_i \le \rho(M_i)(X_{i-1}^T X_{i-1} + X_i^T X_i). \tag{5.14}$$

Substituting (5.14) into (5.2) for each v_i and regrouping the terms in the summation in terms of interfaces, we obtain

$$\begin{aligned} 0 \;&\le\; X^T M X \le \textstyle\sum_{j=1}^{k-1} \left(\frac{\rho(M_j)+\rho(M_{j+1})}{2}\right) X_j^T X_j \\ &\le\; \rho X^T X. \end{aligned} \tag{5.15}$$

Using inequality (5.8), we have

$$\rho < 1. \tag{5.16}$$

This completes the proof. □

We can extend Theorem 5.3 to the important more general case of Fig. 3.2 by taking a closer look at the previous argument. If a subdomain Ω_i has both vertical and horizontal interfaces as boundary pieces, we can obtain a partition of M_i similar to (5.4) if we introduce for each direction x or y a matrix M_i^x or M_i^y. Correspondingly, the binary form v_i in (5.3) becomes a sum of two parts, one for each direction. Similarly, we use for each interface Γ_{ij} the notation

$$\rho_{ij} = \begin{cases} \frac{\rho(M_i^x)+\rho(M_j^x)}{2} & \text{if } \Gamma_{ij} \text{ is a vertical interface} \\ \\ \frac{\rho(M_i^y)+\rho(M_j^y)}{2} & \text{if } \Gamma_{ij} \text{ is a horizontal interface.} \end{cases} \tag{5.17}$$

THEOREM 5.4. *The relaxation process also converges in the case of Fig. 3.2. The convergence rate is bounded above by*

$$\rho \;=\; \max_{\Gamma_{ij}} \rho_{ij} \;<\; 1. \tag{5.18}$$

Proof. We notice that Fig. 3.2 extends Fig. 3.1 in three ways. First, both vertical and horizontal interfaces may be present. Lemma 5.1 and Lemma 5.2 are then naturally extensible by using the previous observation with (5.17) and the eigenvalue analysis for M_i^x and M_i^y, respectively. Second, there may be an interface, say $\Gamma_{1,4}$, that is a middle portion of a boundary piece of a subdomain. In this case, the H-matrix defined in (5.7) may take the form $[0, I_j, 0]^T$. However, it is easy to see that this does not affect the argument in Lemma 5.2 to obtain (5.12). Finally, for the interior "cross points" as marked by "circles" in Fig. 3.2, we note that they are, in fact, not involved in the computation because the 5-point-start stencil does not use

these boundary corner points for the subdomains around them. This completes the proof of Theorem 5.4. □

For the record, we formally state the result that can be established using the same line of argument as above.

COROLLARY 5.5. *Theorems 5.3 and 5.4 remain true for non-uniform mesh spacing and for* $w \neq 0$.

We further comment on other possible extensions of the convergence analysis. We note that the quantity v_i can be viewed as a discrete approximation of a boundary integral for the subdomain Ω_i:

$$v_i = \int_{\partial\Omega_i} \int_{\partial\Omega_i} U_i(\xi)\rho_i(\xi,\eta)U_i(\eta)d\xi\, d\eta, \tag{5.19}$$

where the interface value function $U_i(s) \equiv 0$ on $\partial\Omega \cap \partial\Omega_i$, i.e., the support of $U_i(s)$ is only on the interior interfaces; $\rho_i(x,y)$ corresponds to a Poisson kernel. There are many ways one may obtain an analog to relation (5.14), namely

$$0 \leq v_i \leq \rho(\Omega_i)||U_i||^2_{\partial\Omega_i}, \qquad \text{with } \rho(\Omega_i) < 1, \tag{5.20}$$

using elliptic PDE theory. These usually involve a *maximum-value principle* for more general PDE operators, geometric domains, non-tensor-product meshes, and discretization methods. Then, the remaining argument for the convergence analysis just follows trivially.

6. Numerical experiments. In this section, we report on numerical experiments to illustrate the convergence behavior of interface relaxation.

The following experiments are conducted using the model problem and assumptions of Section 4. Table 1 shows selected values for the optimal parameter ω^-_{opt} and its corresponding convergence rate ρ^-_{opt}; and the convergence rate ρ^+_{opt} corresponding to $\omega^+_{opt} = 0$. Seven cases are examined with various numbers, n, m and k, of the interior horizontal or vertical grid lines, and the subdomains, respectively. They reflect the changes in spacing, aspect ratio and decomposition. The corresponding values for $q_{\min}$ and $q_{\max}$ as defined in Theorem 4.4 are also listed that determine ω^-_{opt}, ρ^-_{opt} and ρ^+_{opt}. We observe that the convergence with ω^-_{opt} is always faster than that with $\omega^+_{opt} \equiv 0$. By checking cases 1 through 3, we see how the convergence is slowed down as the spacing h decreases. Comparison of cases 1, 4 and 5 shows that the convergence is very insensitive to the number of subdomains, which is extremely important for massively parallel computation. Finally, by checking cases 2, 6, 7 and 8, we see that the convergence is also affected by the aspect ratio, m/n, of the subdomain grids. Thus, such grids (and very thin or fat subdomains) are not recommended.

To investigate the convergence sensitivity to the choice of relaxation parameter, we solve a Poisson equation with Dirichlet condition on the rectangular domain $\Omega = (1,4) \times (0,1)$. The true solution is chosen as $U(x,y) = x^2 + y^2$ so that no discretization error is present. The domain Ω is decomposed into three subdomains with interfaces at $x = 2$ and $x = 3$. A uniform grid, with $n = m = 27$, is used for each

TABLE 1

Selected values for the optimal parameter ω_{opt}^- and the corresponding convergence rate ρ_{opt}^-; and the convergence rate ρ_{opt}^+ corresponding to $\omega_{opt}^+ = 0$. Here n is the number of interior horizontal grid lines in each subdomain; m is for the interior vertical grid lines; and k is the number of subdomains. $q_{\min}$ and $q_{\max}$ are the quantities defined in Theorem 4.4 that determine ω_{opt}^-, ρ_{opt}^- and ρ_{opt}^+.

Case	n	m	k	$q_{\min}$	$q_{\max}$	ω_{opt}^-	ρ_{opt}^-	ρ_{opt}^+
1	30	30	11	0.3438	1.825	− 1.18	0.809	0.913
2	50	50	11	0.3434	1.891	− 1.27	0.877	0.946
3	150	150	11	0.3432	1.962	− 1.36	0.955	0.981
4	30	30	21	0.3438	1.826	− 1.19	0.810	0.913
5	30	30	501	0.3438	1.826	− 1.19	0.810	0.913
6	50	30	11	0.3434	1.913	− 1.29	0.900	0.957
7	50	10	11	0.3434	1.959	− 1.36	0.952	0.980
8	50	2	11	0.0135	1.956	− 0.97	0.957	0.978

TABLE 2

Maximum error ($e_{\max}$) after 25 iterations for solving a Poisson equation. The initial error is 9.924. Various relaxation parameter values for ω are tested, including the optimum one, $\omega_{opt}^- = -1.136$.

ω	− 1.200	− 1.136	− 1.000	0.500	0.000
$e_{\max}$	6.237e-3	4.567e-3	1.275e-2	2.896	0.678

subdomain. The theoretical values for ω_{opt}^-, ρ_{opt}^- and ρ_{opt}^+ are -1.136, 0.766 and 0.89, respectively. Various relaxation parameter values for ω are examined to see the effect on convergence. Table 2 lists the corresponding values for the maximum error, $e_{\max}$, on Ω after 25 iterations. The initial error is 9.924. We observe that the convergence rate is not very sensitive to the accuracy of ω_{opt}^-.

For comparison, we solve the same problem on a bigger domain $\Omega = (1, 12) \times (0, 1)$ with 11 unit square subdomains. In this case, $\omega_{opt}^- = -1.139$, $\rho_{opt}^- = 0.775$ and $\rho_{opt}^+ = 0.89$. The initial error is 121.9. With ω_{opt}^- and ω_{opt}^+, the errors are reduced after 25 iterations to 0.290 and 8.487, respectively. We see that the convergence rates remain about the same as the last example as the number of subdomains changes from 3 to 11.

The following experiment provides an example where Lions' method does not converge, while the simple interface relaxer (3.1) does. Consider a one-dimensional indefinite elliptic problem which is extensively used to test numerical methods in the literature [4]:

$$\begin{cases} -u'' - cu = f(x), & \text{in}(0,1), \\ u(0) = u(1) = 0, \end{cases}$$

where $c > 0$. Taking, for instance, $c = 2$ and $f(x) = 1$, with a two-subdomain decomposition, both Lions' method and relaxation scheme (3.1) converge. However, with a three-subdomain decomposition, Lions' method diverges and blows up immediately, while relaxer (3.1) still converges. We can also impose a jump for the flux at each interface point, and the same phenomena are observed. We show in Fig. 6.1 and Fig. 6.2 the iteration profiles for the two-subdomain and three-subdomain decompositions,

respectively. It is seen that relaxation (3.1) converges monotonically. In the two-subdomain case, although Lions' method converges at a faster rate, but the iterates oscillate. We believe that this oscillation is why Lions' method diverges for more than two subdomains or in higher dimensions for indefinite problems. A similar indefinite elliptic problem occurs in solving the Josephson junction problem (1.3)-(1.4) and the same story is repeated.

Considerable experimentation has been made with interface relaxation, much of it unpublished. Some published experiments include the following. For the one-dimensional case, extensive experiments have been made in [14] to study the numerical behavior of various relaxers. In the two dimensional case, many rather difficult problems have been successfully solved using the agent-based software system [6] with relaxer (3.1) and other relaxers by using the least-squares or Newton method as described in Section 3. One solved example [10] is shown in Fig. 6.3 for the illustration of a simple composite PDE problem, where a heat flow system consists of seven parts with nine interfaces. The radiation conditions allow heat to leave on the left and bottom while the temperature U is zero on all the other boundaries. The mounting regions have heat dissipated. The interface conditions are continuity of temperature U and its derivative. The contour plot of the solution computed after 15 iterations of relaxation (3.1) is shown in Fig. 6.4. The paper [7] includes five examples of non-linear composite PDE problems with four subdomains, five interfaces, curved interfaces and two re-entrant corners.

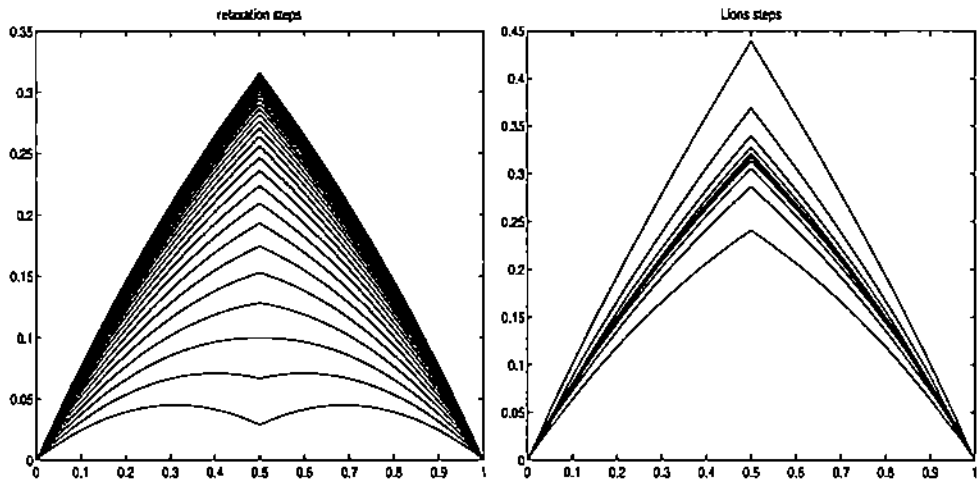

FIG. 6.1. *Solution of an indefinite elliptic problem with a flux jump at each interface, where a two-subdomain decomposition is used. The dark region corresponds to the convergent iterates.*

Finally, we use a one-dimensional example to demonstrate the numerical behavior of relaxation (3.1) and the discretization approximation. Consider the two-point boundary value problem:

$$\begin{cases} -u'' + u = f(x), & \text{in}(0,1), \\ u(0) = u(1) = 0, \end{cases}$$

where $f(x)$ is so chosen that the true solution is $u(x) = x^3(x-1)$. Fig. 6.5 shows the iterates for interface relaxation with (3.1) for a two-subdomain decomposition,

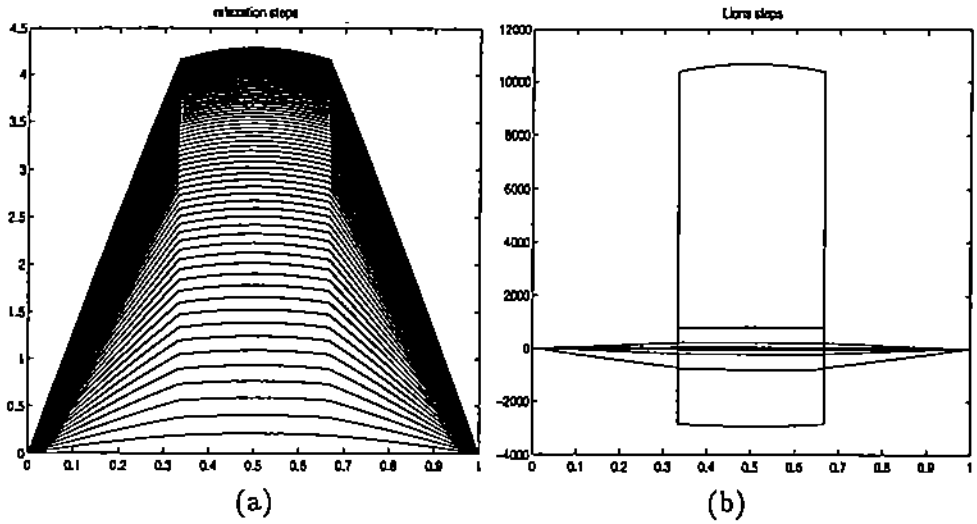

FIG. 6.2. *Solution of an indefinite elliptic problem with a flux jump at each interface, where a three-subdomain decomposition is used. The dark region corresponds to the convergent iterates.*

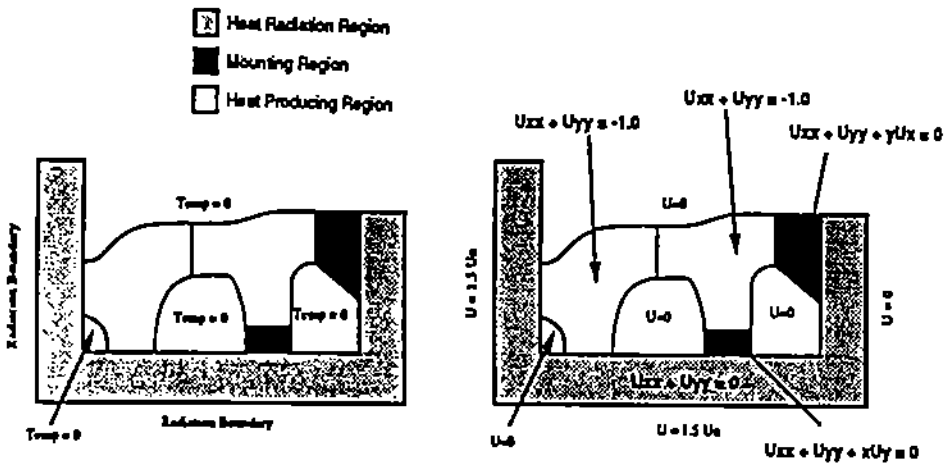

FIG. 6.3. *A heat flow problem for a complex domain along with the physical and mathematical descriptions.*

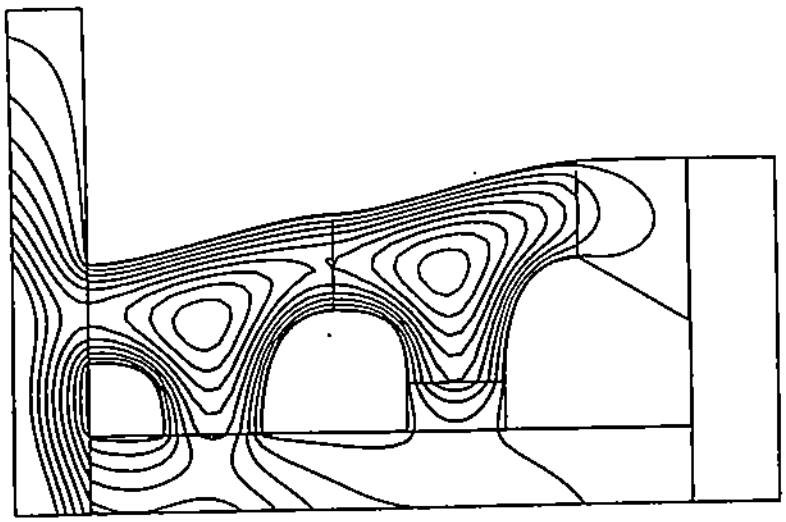

FIG. 6.4. *The contour plot of the solution computed after 15 iterations for the problem in Fig. 6.3.*

where the number of grid points per subdomain is 5 and 11 for Fig. 6.5 (a) and (b), respectively, and the true solution $u(x)$ is marked by *. Fig. 6.6 shows a three-subdomain decomposition, where the number of grid points per subdomain is 11 and 31 for Fig. 6.6 (a) and (b), respectively. We observe that, given a discretization for each subdomain, the iterates of the relaxation $\{U_h^n\}$ converge to a discrete function $\{U_h^*\}$. To improve the approximation accuracy between $\{U_h^*\}$ and $u(x)$, one has to increase the resolution by refining the grid for each subdomain. This, however, slows down the relaxation because its convergence rate depends on h. The error estimation and preconditioning are addressed in the next two sections.

7. Approximation. The Schwarz or relaxation type domain decomposition methods are usually derived by first defining a sequence of iterates $\{u_i^n\}$ for subdomains on the PDE level, and then solving each local PDE by a standard discretization method to obtain the corresponding discrete solution denoted by $U_{i,h}^n$. The following convergence was proved in [9] for a weak norm,

$$\lim_{n\to\infty} \|u_i^n - u|_{\Omega_i}\| = 0. \tag{7.1}$$

On the other hand, one has the error estimate from classic analysis

$$\|U_{i,h}^n - u_i^n\| \le C(u_i^n)h^\alpha, \tag{7.2}$$

where the constant $C(u_i^n)$ depends on the smoothness of u_i^n. This implies that the double-limit procedure $\lim_{n\to\infty}\lim_{h\to 0} U_{i,h}^n$ converges to u_i.

Computationally, the domain decomposition iteration is carried out on a given grid for each subdomain. Therefore one must in fact study the procedure $\lim_{h\to 0}\lim_{n\to\infty} U_{i,h}^n$. There are convergence results of the type

$$\lim_{n\to\infty} \|U_{i,h}^n - U_{i,h}^*\| = 0, \tag{7.3}$$

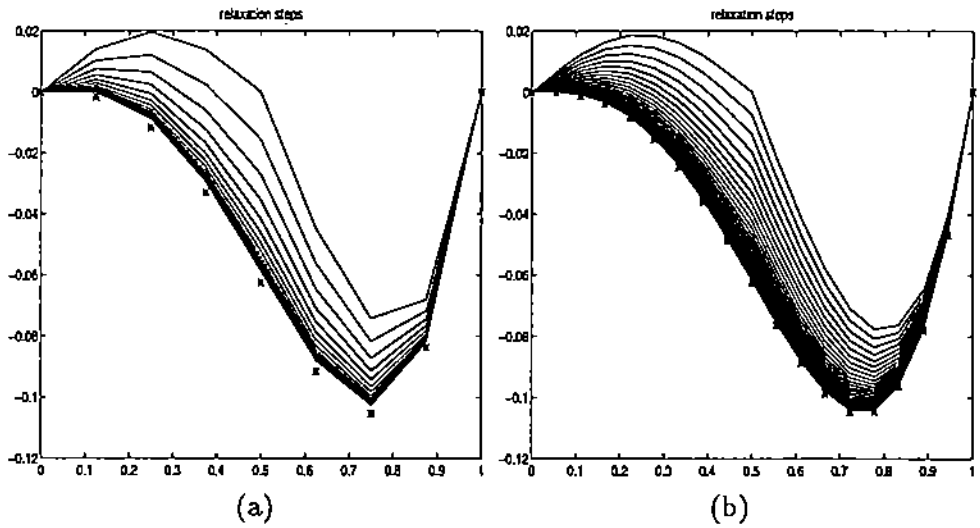

FIG. 6.5. *Relaxation for a two-subdomain decomposition, where the number of grid points per subdomain is 5 and 11 for the left and right figures, respectively, and the true solution $u(x)$ is marked by $*$.*

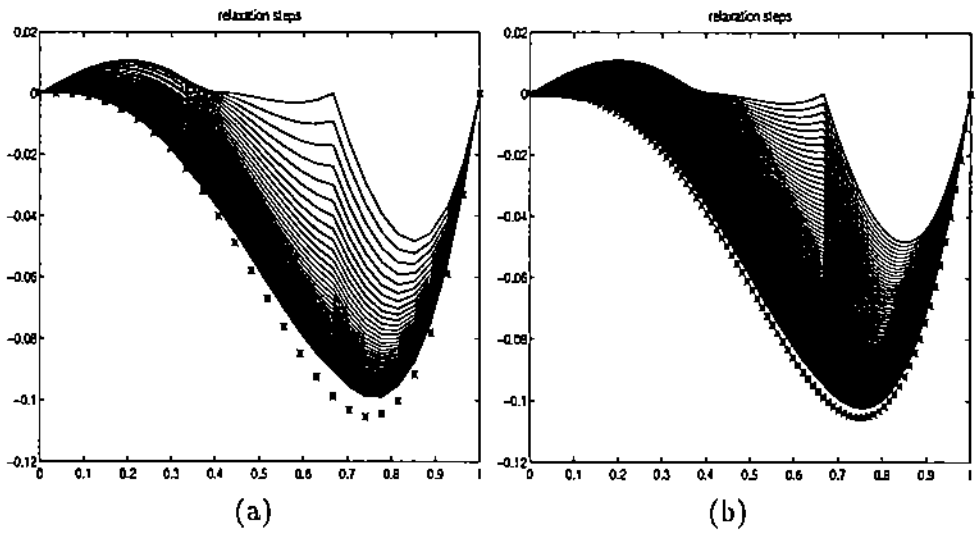

FIG. 6.6. *Relaxation for a three-subdomain decomposition, where the number of grid points per subdomain is 11 and 31 for the left and right figures, respectively.*

for studying the discrete version of the iteration, where $U_{i,h}^*$ denotes the limit of the iteration on each grid such as shown in Figs. 6.5 and 6.6. However, these analyses do not obviously lead to

$$\lim_{h\to 0}\lim_{n\to\infty} U_{i,h}^n = \lim_{n\to\infty}\lim_{h\to 0} U_{i,h}^n = u_i, \tag{7.4}$$

unless some strong and uniform convergence properties can be established. This fact has not drawn enough attention in the literature. The analyses in Sections 4 and 5 correspond to (7.3). Instead of trying to establish a result of the type (7.4), we directly compare $U_{i,h}^*$ with u_i to show that

$$\lim_{h\to 0}\lim_{n\to\infty} U_{i,h}^n = \lim_{h\to 0} U_{i,h}^* = u_i. \tag{7.5}$$

We have already proved that the interface iteration (3.7) converges to X^* which is the solution of the interface equation

$$(I - M)X^* = G. \tag{7.6}$$

Notice the difference between our method and the Schur complement type methods which actually solve the interface equation with the Schur complement coefficient matrix from the global discretization including the interfaces. The Schur complement method leads to an equation like (7.6) but the matrix is different from $I - M$.

Correspondingly, from (3.3) and (3.5) the local solutions on the subdomains converge to $\{U_{i,h}^*\}$ satisfying

$$\begin{cases} A_i U_{i,h}^* = f_i + P_{\Omega_i,\Gamma_{i-1}}\, X_{i-1}^* + P_{\Omega_i,\Gamma_i}\, X_i^* \qquad for\ i = 1, 2, \ldots, k \\ X_0^* = X_k^* = 0, \end{cases} \tag{7.7}$$

and

$$X_i^* = \alpha_i^- U_{i,h}^*|_{\Gamma_i^-} + \alpha_i^+ U_{i+1,h}^*|_{\Gamma_i^+}, \tag{7.8}$$

where (7.7) corresponds to a discrete Dirichlet problem sharing the same interface data with neighbors, and (7.8) corresponds to a discrete version of the continuity interface condition for the Neumann data.

We now present the error analysis for $U_{i,h}^* - u_i$. For simplicity, consider the model problem in Section 4 and assume that there are two subdomains with the interface Γ. The same analysis can be carried out for general cases. Introduce the 5-point-star discretization applied on the global domain Ω. This global discrete problem can be expressed in terms of subdomains and the interface as:

$$\begin{cases} A_i V_{i,h} = f_i + P_{\Omega_i,\Gamma} V_{\Gamma,h}, \quad for\ i = 1, 2, \\ T V_{\Gamma,h} - V_{1,h}|_{\Gamma^-} - V_{2,h}|_{\Gamma_+} = h^2 f_\Gamma. \end{cases} \tag{7.9}$$

Consider the error $E_h = V_h - U_h^*$. From (7.7)-(7.9), we see that E_h satisfies the discrete system:

$$\text{(7.10)} \qquad \begin{cases} A_i E_{i,h} = P_{\Omega_i,\Gamma} E_{\Gamma,h}, \quad for\ i = 1,2, \\ T E_{\Gamma,h} - E_{1,h}|_{\Gamma^-} - E_{2,h}|_{\Gamma_+} = h^2 \delta_\Gamma, \end{cases}$$

where

$$\begin{aligned} \delta_\Gamma &= f_\Gamma + (T U^*_{\Gamma,h} - U^*_{1,h}|_{\Gamma^-} - U^*_{2,h}|_{\Gamma^+})/h^2 \\ &= f_\Gamma + (T U^*_{\Gamma,h} - 2U^*_{\Gamma,h})/h^2 \qquad \text{(7.11)} \\ &= f_\Gamma + \tfrac{T-2I}{h^2} U^*_{\Gamma,h}. \end{aligned}$$

(7.10) implies that E_h can be viewed as the solution of the problem

$$\text{(7.12)} \qquad \begin{cases} L_h E_h = \delta_h(x,y) & \text{in}\Omega, \\ E_h = 0 & \text{on}\partial\Omega, \end{cases}$$

where

$$\text{(7.13)} \qquad \delta_h(x,y) = \begin{cases} \delta_\Gamma, & \text{on}\Gamma \\ 0, & \text{otherwise.} \end{cases}$$

Thus, from standard analysis by viewing $\delta_h(x,y)$ as a finite element function defined on Ω, we immediately have

$$\text{(7.14)} \qquad ||E_h||_{H^1(\Omega)} \le C||\delta_h(x,y)||_{L_2(\Omega)} \le Ch^{1/2}||\delta_\Gamma||_{L_2(\Gamma)}.$$

In the one-dimensional case, observing that $\delta|_\Gamma = f_\Gamma$, we have from (7.14)

$$\text{(7.15)} \qquad ||E_h||_{H^1(\Omega)} \le Ch^{1/2}|f_\Gamma|,$$

where f_Γ is simply the value of f at the interface point. Therefore, using the classic error estimate for $V_h - u$, we have:

$$\begin{aligned} ||U_h^* - u||_{H^1(\Omega)} &\le ||E_h||_{H^1(\Omega)} + ||V_h - u||_{H^1(\Omega)} \\ &\le Ch^{1/2}|f_\Gamma| + h||u||_{H^2(\Omega)}. \end{aligned} \qquad \text{(7.16)}$$

In the two-dimensional case, notice that in (7.11),

$$(T - 2I)/h^2 = [-1, 2, -1]/h^2 \equiv -D^h_{xx}$$

which is the central divided difference of $-\partial^2/\partial x^2$. Thus,

$$\delta_\Gamma = f_\Gamma - D^h_{yy} U^\pi_{\Gamma,h}. \tag{7.17}$$

Recall from (7.6) and (3.5) that

$$\begin{aligned} U^\pi_{\Gamma,h} &= (I-M)^{-1}G \\ &= \tfrac{1-\omega}{2}(I-M)^{-1}(W^T A_1^{-1} f_1 + V^T A_2^{-1} f_2). \end{aligned} \tag{7.18}$$

Also notice that D^h_{yy}, M and A_i^{-1} can be expressed as functions of T and are therefore commutable. Thus,

$$W_h|_\Gamma \equiv D^h_{yy} U^\pi_{\Gamma,h} = \frac{1-\omega}{2}(I-M)^{-1}(W^T A_1^{-1} D^h_{yy} f_1 + V^T A_2^{-1} D^h_{yy} f_2). \tag{7.19}$$

This implies that W_h can be viewed as the solution of the same type of problem (7.7)-(7.8) with right hand side functions $D^h_{yy} f_i$ on Ω_i. Therefore,

$$\|D^h_{yy} U^\pi_{\Gamma,h}\|_{L^2(\Gamma)} = \|W_h\|_{L^2(\Gamma)} \le C\|D^h_{yy} f\|_{L^2(\Omega)} \le C\|\partial^2 f/\partial y^2\|_{L^2(\Omega)}. \tag{7.20}$$

So, similar to (7.16), we have

$$\|U^\pi_h - u\|_{H^1(\Omega)} \le Ch^{1/2}(\|f\|_{L^2(\Gamma)} + \|\partial^2 f/\partial y^2\|_{L^2(\Omega)}) + h\|u\|_{H^2(\Omega)}. \tag{7.21}$$

We observe that the error estimates (7.16) and (7.21) are not optimal compared to the standard global discretization. For the Schwarz splitting method with the Robin transmission condition, a modification on the discrete version of Robin condition was recently suggested in [5] to make $U^\pi_h = V_h$. It is not clear how this can be extended to general cases. On the other hand, the modification on the boundary condition causes changes to standard local solvers, which is not suitable for software integration.

8. Preconditioning. We have discussed convergence and approximation for the interface relaxation approach. From the model problem analysis, we see that the convergence rate depends on h. Therefore, preconditioning is necessary to further improve the relaxation speed. We can view the relaxation procedure as a fixed-point iteration applied to the interface equation (7.6) which plays a similar role as the Schur complement matrix in the substructure approach. It can also be viewed as the Richardson iteration,

$$X^{n+1} - X^n = \theta_n r^n, \tag{8.1}$$

where r^n is the residual and θ_n is the iteration parameter corresponding to (3.1). One can accelerate the iteration by selecting an optimal parameter like what we did in Section 4. But the optimal convergence rate is determined by the condition number of $I-M$. This condition number is of the order of h^{-1} like that of the Schur complement. If one can find a preconditioner P such that

$$\text{cond}(P^{-1}(I - M)) \leq C, \tag{8.2}$$

then, the preconditioned Richardson iteration

$$X^{n+1} = X^n + \theta_n P^{-1} r^n \tag{8.3}$$

will converge at an h-independent rate. For the model problem, such a preconditioner can be easily constructed and implemented by the rational approximation approach [11]. Recall that $I - M$ can be expressed as a function of T:

$$I - M = q(T). \tag{8.4}$$

As shown in [11], one can construct a simple rational function

$$p(x) = \frac{ax + b}{cx + d}, \tag{8.5}$$

such that $P = p(T)$ satisfies (8.2). The implementation of P^{-1} is then simply a tridiagonal solver and a matrix multiplication. The effect of preconditioning on the relaxation is merely to modify (3.1) by applying P^{-1} to the residual g_{ij}^m and then computing b_{ij}^{m+1}. All these are simple interface operations in the relaxer. Therefore, the interface relaxation approach is competitive with any other methods for model problems.

For general cases where the rational preconditioning is not applicable, we propose to use multilevel preconditioning. Like the pointwise relaxation methods, the interface relaxation also works like a smoother for the interface equation (7.6) and damps the high frequency modes of the error on the interfaces. Therefore, after a few steps of interface relaxation such that the error becomes smooth on the current grid level, we transfer to a coarser level to damp the lower frequency modes of the error. A simple V-cycle, two-level interface relaxation procedure is described as follows.

Algorithm

(1) Solve (2.4) on the current level for $\{U_{i,h}^n\}$ with given $\{b_{ij,h}^n\}$;

(2) Compute residual $g_{ij,h}^n$ on the current level;

(3) Relaxation : $b_{ij,h}^{n+1/2} = b_{ji,h}^{n+1/2} = b_{ij,h}^n + \omega_g g_{ij,h}^n$;

(4) Restriction : $b_{ij,H}^{n+1/2} = R_h^H b_{ij,h}^{n+1/2}$;

(5) Solve (2.4) on the coarse level for $\{U_{i,H}^{n+1/2}\}$ with $\{b_{ij,H}^{n+1/2}\}$;

(6) Compute residual $g_{ij,H}^{n+1/2}$ on the coarse level;

(7) Interpolation : $g_{ij,h}^{n+1/2} = I_H^h g_{ij,H}^{n+1/2}$;

(8) Update the interface data : $b_{ij,h}^{n+1} = b_{ji,h}^{n+1} = b_{ij,h}^{n+1/2} + \omega_g g_{ij,h}^{n+1/2}$.

The W-cycle and multilevel versions can be similarly formulated as usual. The advantage of multilevel preconditioning is that it does not explicitly rely on a preconditioner. This is especially suitable for complicated applications.

9. Conclusions. We have presented a unified interface relaxation framework for building PSEs through software integration and implementing various domain decomposition methods for composite PDEs. A new class of relaxers is described which can handle very complicated interface conditions. Convergence analyses, error estimates and preconditioning strategies are also presented. The interface relaxation approach is competitive with other domain decomposition methods for model problems. However, it is applicable to complicated composite PDEs and more suitable for software integration and distributed computing. Numerical experiments with interface relaxation are also promising.

Acknowledgement. The authors would like to thank E. Vavalis, T. Drashansky, S. McFadden and P. Tsombanopoulou for valuable discussions and the long term collaboration in this project.

REFERENCES

[1] R.E. Bank and D.J. Rose (1977), *Marching algorithms for elliptic boundary value problems. I: the constant coefficient case*, SIAM J. Numer. Anal., 14, pp. 792–829.

[2] G. Bao, D. Dobson and J. Cox (1995), *Mathematical studies in rigorous grating theory*, J. OSA A, 12, pp. 1029-1042.

[3] A. Barone and G. Paterno (1982), *Physics and applications of the Josephson effect*, John Wiley, New York.

[4] J. Bramble, Z. Leyk and J. Pasciak (1993), *Iterative schemes for nonsymmetric and indefinite elliptic boundary value problems*, Math Comp., 60, pp. 1-22.

[5] J. J. Douglas and C. Huang (1996), *An accelerated domain decomposition procedure based on Robin transmission conditions*, Tech Report 289, Oct. 1996, Dept. of Math., Purdue Univ., West Laf., IN 47907, USA.

[6] T. Drashansky (1996), *An agent-based approach to building multidisplinary PSEs*, Ph. D. Thesis, Dept. of Computer Sci., Purdue Univ., WL, IN 47907, USA.

[7] T. Drashansky, J. Rice, E. Houstis, P. Tsompanopoulou, E. Vavalis and S. Weerawarana (1997), *Collaborating problem solving agents for multi-physics problems*, Proc. IMACS World Conf., to appear.

[8] B. Josephson, Phys. Lett., 1, pp. 251.

[9] P. L. Lions (1990), *On the Schwarz alternating method III: a variant for nonoverlapping subdomains*, Domain Decomposition Methods for PDEs, T. Chan et al. eds., SIAM, Philadelphia, PA, 1990, pp. 202-223.

[10] McFadden, H.S., and John R. Rice (1992), *Collaborating PDE solvers*, Appl. Numer. Math., 10, pp. 279–295.

[11] M. Mu (1995), *A new family of preconditioners for domain decomposition*, SIAM J. Sci. Comput., 16, pp. 289-306.

[12] M. Mu and J.R. Rice (1994), *Preconditioning for domain decomposition through function approximation*, SIAM J. Sci. Comput., 15, pp. 1452-1466.

[13] M. Mu and J.R. Rice (1995), *Modeling with collaborating PDE solvers – theory and practice*, Computing Systems in Eng., 6, pp. 87-95.

[14] J. Rice, P. Tsombanopoulou and E. Vavalis (1997), *Interface relaxation methods for elliptic differential equations*, 9th Int. Domain Decomposition Conf.

[15] A. Quarteroni and G. Sacchi-Landriani (1988), *Domain decomposition preconditioners for the spectral collocation method*, NASA Contractor Report 181620, ICASE Report No. 88-11, ICASE, NASA Langley Research Center, Hampton, VA 23665.