

A Nonoverlapping Subdomain Algorithm with Lagrange Multipliers and its Object Oriented Implementation for Interface Problems

Daoqi Yang

1. Introduction

A parallel nonoverlapping subdomain Schwarz alternating algorithm with Lagrange multipliers and interface relaxation is proposed for linear elliptic interface problems with discontinuities in the solution, its derivatives, and the coefficients. New features of the algorithm include that Lagrange multipliers are introduced on the interface and that it is used to solve equations with discontinuous solution. These equations have important applications to alloy solidification problems [1] and immiscible flow of fluids with different densities and viscosities and surface tension. They do not fit into the Schwarz preconditioning and Schur complement frameworks since the solution is not in $H^1(\Omega)$, but is piecewise in $H^1(\Omega)$, where Ω is the physical domain on which the differential equations are defined. An expression for the optimal interface relaxation parameters is also given. Numerical experiments are conducted for a piecewise linear triangular finite element discretization in object oriented paradigm using C++. In this implementation, the class for subdomain solvers inherits from the class for grid triangulation and contains type bound procedures for forming stiffness matrices and solving linear systems. From software engineering point of view, features like encapsulation, inheritance, polymorphism and dynamic binding, make such domain decomposition algorithms an ideal application area of object oriented programming.

The organization of this work is as follows. In Section 2, the domain decomposition method is described for general elliptic interface problems. In Section 3, a finite element approximation with Lagrange multipliers is considered. Then in Section 4, some implementation issues using object oriented techniques in C++ are discussed. Finally in Section 5, numerical examples are provided to check the performance of the method.

1991 *Mathematics Subject Classification*. Primary 65N55; Secondary 65N30.

Key words and phrases. Interface Problems, Lagrange Multipliers, Interface Relaxation, Object Oriented Programming.

This work was supported in part by Institute for Mathematics and its Applications at University of Minnesota. No version of this paper will be published elsewhere.

2. The Domain Decomposition Method

Let Ω be a smooth bounded two-dimensional domain or a convex polygon with boundary $\partial\Omega$. Assume that Ω is the union of two nonoverlapping subdomains Ω_1 and Ω_2 with interface Γ ; that is, $\bar{\Omega} = \bar{\Omega}_1 \cup \bar{\Omega}_2$, $\Omega_1 \cap \Omega_2 = \emptyset$, and $\Gamma = \partial\Omega_1 \cap \partial\Omega_2$. When Ω_1 and Ω_2 are disconnected, the decomposition can actually contain more than two subdomains.

Consider the following elliptic interface problem: find $u_1 \in H^1(\Omega_1)$ and $u_2 \in H^1(\Omega_2)$ such that

$$\begin{aligned} (1) \quad & L_k u_k = f \text{ in } \Omega_k, \quad k = 1, 2, \\ (2) \quad & u_k = g \text{ on } \partial\Omega_k \cap \partial\Omega, \quad k = 1, 2, \\ (3) \quad & u_1 - u_2 = \mu \text{ on } \Gamma, \\ (4) \quad & \frac{\partial u_1}{\partial \nu_A^1} + \frac{\partial u_2}{\partial \nu_A^2} = \eta \text{ on } \Gamma, \end{aligned}$$

where for $k = 1, 2$,

$$L_k u = - \sum_{i,j=1}^2 \frac{\partial}{\partial x_i} \left(a_{ij}^{(k)}(x) \frac{\partial u}{\partial x_j} \right) + b^{(k)}(x)u, \quad \frac{\partial u}{\partial \nu_A^k} = \sum_{i,j=1}^2 a_{ij}^{(k)} \frac{\partial u}{\partial x_j} \nu_i^k,$$

$\nu^k = \{\nu_1^k, \nu_2^k\}$ is the outward normal unit vector to $\partial\Omega_k$. I assume that $f \in L^2(\Omega)$ and $g \in H^{1/2}(\partial\Omega)$, and that $\mu(x)$ and $\eta(x)$ are given regular functions on Γ . That is, the solution of (1)-(4) is sought with specified strength of discontinuity, and so is its conormal derivative. I also assume that the coefficients $\{a_{ij}^{(k)}\}$ are symmetric, uniformly positive definite, and bounded on Ω_k , and $b^{(k)} \geq 0$ on Ω_k .

Equations of type (1)-(4) need be solved at each time step for the two-phase generalized Stephan problems as encountered in alloy solidification processes [1, pages 8, 15]. They have applications in many wave propagation and fluid flow problems [1, 5]. In some cases, the interface conditions (3) and (4) are nonlinear [1].

Schwarz overlapping and substructuring domain decomposition methods have been analyzed for the special linear case (1)-(4) in which $\mu = 0$ and $\eta = 0$. It is not clear that any of these methods applies directly to the general case with $\mu \neq 0$ and $\eta \neq 0$, since the solution is not in the space $H^1(\Omega)$, although its restriction to Ω_k is in the space $H^1(\Omega_k)$. Finite element methods on the whole domain Ω without domain decomposition do not seem to apply. In [5], a special kind of finite difference method without domain decomposition was considered for the problem (1)-(4).

I now define formally the following domain decomposition method for problem (1)-(4): Choose $u_k^0 \in H^1(\Omega_k)$ satisfying $u_k^0|_{\Omega \cap \Omega_k} = g$, $k = 1, 2$. For $n = 0, 1, 2, \dots$,

the sequence $u_k^n \in H^1(\Omega_k)$ with $u_k^n|_{\partial\Omega \cap \partial\Omega_k} = g$ is constructed such that

$$\begin{aligned}
 (5) \quad & \begin{cases} L_1 u_1^{2n+1} = f \text{ in } \Omega_1, \\ u_1^{2n+1} = \alpha u_1^{2n} + (1 - \alpha) u_2^{2n} + (1 - \alpha) \mu \text{ on } \Gamma; \end{cases} \\
 (6) \quad & \begin{cases} L_2 u_2^{2n+1} = f \text{ in } \Omega_2, \\ u_2^{2n+1} = \alpha u_1^{2n} + (1 - \alpha) u_2^{2n} - \alpha \mu \text{ on } \Gamma; \end{cases} \\
 (7) \quad & \begin{cases} L_1 u_1^{2n+2} = f \text{ in } \Omega_1, \\ \frac{\partial u_1^{2n+2}}{\partial \nu_A^1} = \beta \frac{\partial u_1^{2n+1}}{\partial \nu_A^1} + (1 - \beta) \frac{\partial u_2^{2n+1}}{\partial \nu_A^1} + (1 - \beta) \eta \text{ on } \Gamma; \end{cases} \\
 (8) \quad & \begin{cases} L_2 u_2^{2n+2} = f \text{ in } \Omega_2, \\ \frac{\partial u_2^{2n+2}}{\partial \nu_A^2} = \beta \frac{\partial u_1^{2n+1}}{\partial \nu_A^2} + (1 - \beta) \frac{\partial u_2^{2n+1}}{\partial \nu_A^2} + \beta \eta \text{ on } \Gamma; \end{cases}
 \end{aligned}$$

where $\alpha, \beta \in (0, 1)$ are relaxation parameters that will be determined to ensure and accelerate the convergence of the iterative procedure. This algorithm is motivated by the ones proposed by Funaro, Quarteroni, and Zanolli [4], Marini and Quarteroni [7, 8], Lions [6], Després [2], Rice, Vavalis, and Yang [10], Quarteroni [9], Douglas and Yang [3], and Yang [11, 12], where regular problems with $\mu = 0$ and $\eta = 0$ were considered.

This algorithm is different from others in that Dirichlet and Neumann subdomain problems were solved at the same iteration levels but on different subdomains in [4, 7, 8, 11] and Robin subdomain problems were solved in [6, 2, 12]. For problems with continuous solution and coefficients, this algorithm reduces to [10, 3], where convergence results for general coefficients were not provided. In [13], a detailed analysis for the algorithm will be made at the differential and discrete levels and numerical tests using finite difference methods will be conducted.

3. Finite Element Approximation with Lagrange Multipliers

In this section, I reformulate the algorithm (5)-(8) by introducing Lagrange multipliers on the interface to replace conormal derivatives. Finite element discretization is then applied to subdomain problems. Although Lagrange multipliers have been used in other contexts, it does not appear that they have been employed to (5)-(8) before. I first introduce some notation. Denote the Hilbert spaces

$$V_k = \{v \in H^1(\Omega_k), \quad v|_{\partial\Omega \cap \partial\Omega_k} = 0\}, \quad k = 1, 2,$$

and let γ_0 be the trace operator from V_k onto $H_{00}^{1/2}(\Gamma)$. Define the bilinear forms:

$$(9) \quad a_k(u, w) = \sum_{i,j=1}^2 \int_{\Omega_k} a_{ij}^{(k)} \frac{\partial u}{\partial x_j} \frac{\partial w}{\partial x_i} dx + \int_{\Omega_k} b^{(k)} u w dx, \quad k = 1, 2,$$

$$(10) \quad (u, w)_k = \int_{\Omega_k} u w dx, \quad k = 1, 2.$$

For convenience I use the following norms in V_1 and V_2 ,

$$(11) \quad \|w\|_k^2 = a_k(w, w), \quad \forall w \in V_k, \quad k = 1, 2.$$

Then the variational formulation of the scheme (5)-(8) can be written as:

$$(12) \quad \begin{cases} a_1(u_1^{2n+1}, w) - \left\langle \frac{\partial u_1^{2n+1}}{\partial \nu_A^1}, \gamma_0 w \right\rangle = (f, w)_1, & \forall w \in V_1, \\ u_1^{2n+1} = \alpha u_1^{2n} + (1 - \alpha)u_2^{2n} + (1 - \alpha)\mu \text{ on } \Gamma; \end{cases}$$

$$(13) \quad \begin{cases} a_2(u_2^{2n+1}, w) - \left\langle \frac{\partial u_2^{2n+1}}{\partial \nu_A^2}, \gamma_0 w \right\rangle = (f, w)_2, & \forall w \in V_2, \\ u_2^{2n+1} = \alpha u_1^{2n} + (1 - \alpha)u_2^{2n} - \alpha\mu \text{ on } \Gamma; \end{cases}$$

$$(14) \quad \begin{aligned} & a_1(u_1^{2n+2}, w) \\ &= \beta \left\langle \frac{\partial u_1^{2n+1}}{\partial \nu_A^1}, \gamma_0 w \right\rangle + (1 - \beta) \left\langle \frac{\partial u_2^{2n+1}}{\partial \nu_A^1} + \eta, \gamma_0 w \right\rangle + (f, w)_1, \quad \forall w \in V_1; \end{aligned}$$

$$(15) \quad \begin{aligned} & a_2(u_2^{2n+2}, w) \\ &= \beta \left\langle \frac{\partial u_1^{2n+1}}{\partial \nu_A^2} + \eta, \gamma_0 w \right\rangle + (1 - \beta) \left\langle \frac{\partial u_2^{2n+1}}{\partial \nu_A^2}, \gamma_0 w \right\rangle + (f, w)_2, \quad \forall w \in V_2; \end{aligned}$$

for $k = 1, 2$. Here $\langle \cdot, \cdot \rangle$ denotes the inner product over the interface Γ , or the duality between $H_{00}^{1/2}(\Gamma)$ and its dual space.

Replacing the conormal derivatives $\frac{\partial u_k^n}{\partial \nu_A^k}$ by the Lagrange multipliers λ_k^n , (12)-(15) becomes

$$(16) \quad \begin{cases} a_1(u_1^{2n+1}, w) - \langle \lambda_1^{2n+1}, \gamma_0 w \rangle = (f, w)_1, & \forall w \in V_1, \\ u_1^{2n+1} = \alpha u_1^{2n} + (1 - \alpha)u_2^{2n} + (1 - \alpha)\mu \text{ on } \Gamma; \end{cases}$$

$$(17) \quad \begin{cases} a_2(u_2^{2n+1}, w) - \langle \lambda_2^{2n+1}, \gamma_0 w \rangle = (f, w)_2, & \forall w \in V_2, \\ u_2^{2n+1} = \alpha u_1^{2n} + (1 - \alpha)u_2^{2n} - \alpha\mu \text{ on } \Gamma; \end{cases}$$

$$(18) \quad \begin{aligned} & a_1(u_1^{2n+2}, w) \\ &= \langle \beta \lambda_1^{2n+1} - (1 - \beta)\lambda_2^{2n+1} + (1 - \beta)\eta, \gamma_0 w \rangle + (f, w)_1, \quad \forall w \in V_1; \end{aligned}$$

$$(19) \quad \begin{aligned} & a_2(u_2^{2n+2}, w) \\ &= \langle -\beta \lambda_1^{2n+1} + (1 - \beta)\lambda_2^{2n+1} + \beta\eta, \gamma_0 w \rangle + (f, w)_2, \quad \forall w \in V_2. \end{aligned}$$

The procedure (16)-(19) is the domain decomposition method with Lagrange multipliers at the differential level, a variant of (5)-(8). I now formulate its finite element version. Let $T_h = \{T\}$ be a regular triangulation of Ω with no elements crossing the interface Γ . I define finite element spaces, for $k = 1, 2$,

$$(20) \quad W_k^h = \{w \in H^1(\Omega_k) : w|_T \in P_r(T) \quad \forall T \in T_h, w|_{\partial\Omega \cap \partial\Omega_k} = 0\},$$

where $P_r(T)$ denotes the space of polynomials of degree $\leq r$ on T . Let Z^h be the space of the restrictions on the interface of the functions in W_k^h . Note that there are two copies of such a space assigned on Γ , one from Ω_1 and the other from Ω_2 . I denote them by Z_1^h and Z_2^h , respectively. Let $\{U_k^n, \Lambda_k^n\} \in W_k^h \times Z_k^h$ denote the finite element approximation of $\{u_k^n, \lambda_k^n\}$. Then, the finite element domain decomposition

method with Lagrange multipliers is constructed as follows:

$$(21) \quad \begin{cases} a_1(U_1^{2n+1}, w) - \langle \Lambda_1^{2n+1}, \gamma_0 w \rangle = (f, w)_1, & \forall w \in W_1^h, \\ U_1^{2n+1} = \alpha U_1^{2n} + (1 - \alpha)U_2^{2n} + (1 - \alpha)\mu \text{ on } \Gamma; \end{cases}$$

$$(22) \quad \begin{cases} a_2(U_2^{2n+1}, w) - \langle \Lambda_2^{2n+1}, \gamma_0 w \rangle = (f, w)_2, & \forall w \in W_2^h, \\ U_2^{2n+1} = \alpha U_1^{2n} + (1 - \alpha)U_2^{2n} - \alpha\mu \text{ on } \Gamma; \end{cases}$$

$$(23) \quad \begin{aligned} & a_1(U_1^{2n+2}, w) \\ & = \langle \beta \Lambda_1^{2n+1} - (1 - \beta)\Lambda_2^{2n+1} + (1 - \beta)\eta, \gamma_0 w \rangle + (f, w)_1, \quad \forall w \in W_1^h; \end{aligned}$$

$$(24) \quad \begin{aligned} & a_2(U_2^{2n+2}, w) \\ & = \langle -\beta \Lambda_1^{2n+1} + (1 - \beta)\Lambda_2^{2n+1} + \beta\eta, \gamma_0 w \rangle + (f, w)_2, \quad \forall w \in W_2^h. \end{aligned}$$

In order to give a convergence result, I introduce two linear operators. Let $\Phi^h = \{w|_\Gamma : w \in W_k^h, k = 1, 2\}$. Define the extension operators $R_k : \phi \in \Phi^h \rightarrow \{R_k^1\phi, R_k^2\phi\} \in W_k^h \times Z_k^h$ by

$$(25) \quad a_k(R_k^1\phi, w) - \langle R_k^2\phi, w \rangle = 0, \quad \forall w \in W_k^h, \quad R_k^1\phi = \phi \text{ on } \Gamma.$$

Define $\bar{\sigma}$ and $\bar{\tau}$ to be two smallest finite real numbers such that

$$(26) \quad \sup_{\phi \in \Phi^h} \frac{\|R_1^1\phi\|_1^2}{\|R_2^1\phi\|_2^2} \leq \bar{\sigma}, \quad \sup_{\phi \in \Phi^h} \frac{\|R_2^1\phi\|_2^2}{\|R_1^1\phi\|_1^2} \leq \bar{\tau}.$$

The following results will be proved in [13].

THEOREM 1. *The domain decomposition method (21)-(24) is convergent in the energy norm if $\max\{0, 1 - \frac{2(\bar{\tau}+1)}{\bar{\tau}\bar{\sigma}^2 + \bar{\tau} + 2}\} < \alpha < 1$ and $\max\{0, 1 - \frac{2(\bar{\sigma}+1)}{\bar{\tau}^2\bar{\sigma} + \bar{\sigma} + 2}\} < \beta < 1$. The optimal relaxation parameters are $\alpha = \frac{\bar{\sigma}^2\bar{\tau} + 1}{\bar{\tau}\bar{\sigma}^2 + \bar{\tau} + 2}$ and $\beta = \frac{\bar{\sigma}\bar{\tau}^2 + 1}{\bar{\tau}^2\bar{\sigma} + \bar{\sigma} + 2}$. Furthermore, the convergence is independent of the grid size h .*

4. Object Oriented Implementation

The finite element domain decomposition algorithm (21)-(24) can be implemented in an elegant way using the object oriented programming paradigm. Features such as information hiding, data encapsulation, inheritance, dynamic binding, and operator overloading can be employed very nicely in the algorithm. Below I give a brief description of my implementation details in the terminology of C++.

I first define a base class called “grid”, which generates a triangular grid in a subdomain. Data members like vertices of the triangles and coordinates of the vertices are protected members which can only be accessed by its members and derived class. Function members like getting the total degrees of freedom and printing a Matlab file for representing the triangulation are public that can be accessed by any program. Each object of the class grid represents the triangulation on each subdomain. The grid on different subdomains may not have to match on the interface. However, for simplicity, I apply matching grids on the subdomains.

Then I define a derived class called “subdomain”, which inherits from the class grid. The class subdomain contains private member functions for forming the stiffness matrix and for performing the Dirichlet and Neumann sweeps. The stiffness matrix forming function can access the triangulation information like triangle vertices and coordinates of vertices, which are protected members of the base class grid. In this step, I make use of my linear algebra library (which is built on object oriented programming for matrix and vector manipulations) for numerical integration. A numerical quadrature can be viewed as an inner product of two vectors;

using operator overloading, vectors can be multiplied directly, instead of an explicit function call like in Fortran or C. The Dirichlet sweep member function first solves the subdomain problem with Dirichlet boundary condition by choosing basis functions vanishing on the interface and then finds the Lagrange multipliers by choosing basis functions vanishing in the interior of the subdomain. The Neumann sweep member function just solves the subdomain problem with Neumann boundary condition on the interface. From these two steps, we see that the finite element method with Lagrange multipliers on the interface is easier to implement than finite difference or finite element methods without Lagrange multipliers [5, 13]. In particular, the subdomain finite dimensional problems are symmetric and positive definite in our case, as opposed to unsymmetric and indefinite problems in [5].

Finally, a friend to the class subdomain is implemented to coordinate the Dirichlet and Neumann sweeps and check the stopping criterion for the iterative process. This friend function takes an array of subdomain objects as arguments and has access to protected members of class grid and all members of class subdomain.

At the linear system solving steps inside the Dirichlet and Neumann sweeps, I first define an abstract matrix class that just contains the number of rows of the matrix, two pure virtual functions for matrix-vector multiplication and preconditioning, and a function for the preconditioned conjugate gradient method. Since the preconditioned conjugate gradient algorithm can be implemented once we have a matrix-vector multiplication function and a preconditioning function, it is defined in the abstract matrix class and inherited by classes for banded matrices and sparse matrices. With operator overloading (one kind of polymorphism), the preconditioned conjugate gradient function can be written in about the same number of lines and format as the algorithm (which increases the readability of the code) and is defined only once in the abstract class. It then can be called in the derived classes for banded matrices and sparse matrices which need to define the matrix-vector multiplication and preconditioning functions according to their data structures of the matrix storage. However, the banded Gauss elimination function has to be defined in every derived class since it can not be performed without knowing the structure of the matrix. Note that the banded matrix inherits the row number from the abstract matrix class and needs to define a data member for the bandwidth. For unsymmetric problems, GMRES instead of conjugate gradient method or banded Gauss elimination is used.

5. Numerical Examples

In this section, I present some numerical experiments for the iterative procedure (21)-(24). In all of my test, I let the domain $\Omega = \{(x, y) : 0 < x < 1, 0 < y < 0.5, x \geq y\} \cup \{(x, y) : 0 < x < 0.5, 0 < y < 1, x \leq y\}$ and the interface Γ be the line segment $\{(x, y) : x = y, 0 \leq x \leq 0.5\}$, which divides Ω into $\Omega_1 = \{(x, y) : 0 < x < 1, 0 < y < 0.5, x > y\}$ and $\Omega_2 = \{(x, y) : 0 < x < 0.5, 0 < y < 1, x < y\}$. Piecewise linear triangular finite elements are applied on each subdomain. One copy of the solution at the interface is kept from each subdomain due to its discontinuity. Note that a global finite element solution on the whole domain Ω may not exist. The relaxation parameters α and β are taken to be 0.5, which leads to fast convergence and thus optimal parameters are not used. Initial guess is chosen to be zero in all cases. I define iterative errors as the relative errors between the iterates at the current and previous iteration levels and true errors as the relative errors between

TABLE 1. Iterative and true errors for Example 2. The errors are shown in the L^∞ -norm.

Iteration	Grid size $\frac{1}{40} \times \frac{1}{40}$		Grid size $\frac{1}{80} \times \frac{1}{80}$	
	Iterative error	True error	Iterative error	True error
1	3.16E-1	2.75E-1	3.25E-1	2.93E-1
2	3.77E-2	1.05E-2	3.96E-2	3.21E-2
3	4.97E-3	5.33E-3	5.28E-3	7.55E-3
4	6.78E-4	6.01E-3	7.49E-4	2.27E-3
5	9.76E-5	5.91E-3	1.13E-4	3.02E-3

the current iterate and the true solution. All norms will be measured in the discrete $\max(\|\cdot\|_{L^\infty(\Omega_1)}, \|\cdot\|_{L^\infty(\Omega_2)})$ sense, which is different from the discrete $L^\infty(\Omega)$ norm since there are two different values on the interface from the two subdomains due to discontinuity.

EXAMPLE 2. Let

$$\begin{aligned}
 &-\frac{\partial}{\partial x}(e^x \frac{\partial u_1}{\partial x}) - \frac{\partial}{\partial y}(e^y \frac{\partial u_1}{\partial y}) + \frac{1}{1+x+y} u_1 = f_1, \quad \text{in } \Omega_1, \\
 &-\frac{\partial^2 u_2}{\partial x^2} - \frac{\partial^2 u_2}{\partial y^2} = f_2, \quad \text{in } \Omega_2, \\
 &u_k = g_k, \quad \text{on } \partial\Omega_k \cap \partial\Omega, k = 1, 2.
 \end{aligned}$$

The functions $f_1, f_2, g_1, g_2, \mu,$ and η are chosen such that the exact solution is

$$u_1(x, y) = 10x + y, \text{ in } \Omega_1, \quad u_2(x, y) = \sin(x + y), \text{ in } \Omega_2.$$

Table 1 shows the results for the iterative and true errors in the maximum norm on the whole domain Ω .

Now we consider a more difficult problem with convection on one side of the interface and general variable coefficients. In the Stephan problem [1], for the case of two incompressible phases, one solid and one liquid, with different densities, a convective term must be added to the heat-flow equation in the liquid region in order for mass to be conserved across phase-change interface.

EXAMPLE 3. Let

$$\begin{aligned}
 &-\nabla \cdot \left(\begin{bmatrix} e^x & x \\ x & e^y \end{bmatrix} \nabla u_1 \right) + \left[\frac{1}{1+x+y}, \frac{1}{1+x+y} \right] \nabla u_1 + \frac{u_1}{1+x+y} \\
 &= f_1(x, y), \quad \text{in } \Omega_1, \\
 &-\nabla \cdot \left(\begin{bmatrix} x+1 & \sin(xy) \\ \sin(xy) & y+1 \end{bmatrix} \nabla u_2 \right) + (2 + \sin(x) + \cos(y))u_2 \\
 &= f_2(x, y), \quad \text{in } \Omega_2, \\
 &u_k = g_k, \quad \text{on } \partial\Omega_k \cap \partial\Omega, k = 1, 2.
 \end{aligned}$$

The functions $f_1, f_2, g_1, g_2, \mu,$ and η are chosen such that the exact solution is

$$u_1(x, y) = e^{xy}, \text{ in } \Omega_1, \quad u_2(x, y) = 5e^{xy} \sin(13x) \cos(13y), \text{ in } \Omega_2.$$

Table 2 shows the results for the iterative and true errors in the maximum norm on the whole domain Ω .

Numerical experiments show that the iterative method is insensitive to strong discontinuities in the solution and coefficients and leads to accurate approximate

TABLE 2. Iterative and true errors for Example 3. The errors are shown in the L^∞ -norm.

Iteration	Grid size $\frac{1}{40} \times \frac{1}{40}$		Grid size $\frac{1}{80} \times \frac{1}{80}$	
	Iterative error	True error	Iterative error	True error
1	1.97E-1	3.84E-1	2.33E-1	5.02E-1
2	8.77E-3	5.82E-3	1.09E-2	9.56E-3
3	3.38E-5	3.21E-3	4.16E-5	1.53E-3
4	1.41E-7	3.18E-3	1.78E-7	1.48E-3
5	6.27E-10	3.18E-3	8.29E-10	1.48E-3

solutions. Although the relaxation parameters α and β can be chosen in some optimal fashion, the method converges pretty fast with $\alpha = \beta = 1/2$ even for very complicated problems. It is observed that this method converges faster than Lions type methods [6] with a few subdomains which is suitable for most interface problems. Also, sharp interfaces of the true solution can be captured fairly easily and accurately. In [13], finite difference methods are applied to subdomain problems and similar numerical results are obtained.

To my knowledge, this paper is the first one to apply the Schwarz domain decomposition methodology to interface problems with discontinuous solution, conormal derivatives, and coefficients. This is a first attempt to solve time-dependent generalized Stefan problems such as alloy solidification and immiscible flow with surface tension. Applying non-matching grids will make my algorithm more attractive and suitable for such problems in that finite element grids can be generated separately in different subdomains and collaborative PDE solvers can be applied. Another salient feature of this algorithm is that symmetrical and positive definite linear systems are solved at each iteration for problems like Example 2 which is not symmetrical and positive definite in the whole domain. Implementations in the object oriented paradigm also make the algorithm easier to code and modify to meet the need of more complicated situations. Our future work will try to solve two and three dimensional application problems with more complex geometry and interface. Mortar elements may also be applied to this kind of problems.

References

1. J. Crank, *Free and moving boundary problems*, Oxford University Press, Oxford, 1984.
2. B. Despres, *Domain decomposition method and the Helmholtz problems*, Mathematical and Numerical Aspects of Wave Propagation Phenomena (G. Cohen, L. Halpern, and P. Joly, eds.), SIAM, Philadelphia, 1991, pp. 44–52.
3. J. Douglas, Jr. and D. Q. Yang, *Numerical experiments of a nonoverlapping domain decomposition method for partial differential equations*, Numerical Analysis: A. R. Mitchell 75th Birthday Volume (D. Griffiths and G. A. Watson, eds.), World Scientific, Singapore, 1996, pp. 85–97.
4. D. Funaro, A. Quarteroni, and P. Zanolli, *An iterative procedure with interface relaxation for domain decomposition methods*, SIAM J. Numer. Anal. **25** (1988), 1213–1236.
5. R. J. LeVeque and Z. L. Li, *The immersed interface method for elliptic equations with discontinuous coefficients and singular sources*, SIAM J. Numer. Anal. **31** (1994), 1019–1044.
6. P.-L. Lions, *On the Schwarz alternating method III: a variant for nonoverlapping subdomains*, Third International Symposium on Domain Decomposition Methods for Partial Differential Equations (Philadelphia) (T. F. Chan, R. Glowinski, J. Périaux, and O. B. Widlund, eds.), SIAM, 1990, pp. 202–223.

7. L. D. Marini and A. Quarteroni, *An iterative procedure for domain decomposition methods: a finite element approach*, First International Symposium on Domain Decomposition Methods for Partial Differential Equations (Philadelphia) (R. Glowinski, G. H. Golub, G. A. Meurant, and J. Périaux, eds.), SIAM, 1988, pp. 129–143.
8. ———, *A relaxation procedure for domain decomposition methods using finite elements*, Numer. Math. **55** (1989), 575–598.
9. A. Quarteroni, *Domain decomposition methods for wave propagation problems*, Domain-Based Parallelism and Problem Decomposition Methods in Computational Science and Engineering (D. E. Keyes *et al.*, ed.), SIAM, Philadelphia, 1995, pp. 21–38.
10. J. R. Rice, E. A. Vavalis, and D. Q. Yang, *Convergence analysis of a nonoverlapping domain decomposition method for elliptic PDEs*, J. Comput. Appl. Math. **87** (1997), 11–19.
11. D. Q. Yang, *A parallel iterative nonoverlapping domain decomposition procedure for elliptic problems*, IMA J. Numer. Anal. **16** (1996), 75–91.
12. ———, *A parallel nonoverlapping Schwarz domain decomposition algorithm for elliptic partial differential equations*, Proceedings of the Eighth SIAM Conference on Parallel Processing for Scientific Computing (Philadelphia) (M. Heath *et al.*, ed.), SIAM, 1997.
13. ———, *A parallel nonoverlapping Schwarz domain decomposition method for elliptic interface problems*, (In preparation).

DEPARTMENT OF MATHEMATICS, WAYNE STATE UNIVERSITY, DETROIT, MI 48202-3483, USA
E-mail address: dyang@na-net.ornl.gov, <http://www.math.wayne.edu/~yang>