Report
Universitetet i Bergen
Matematisk inst.

Department
of
APPLIED MATHEMATICS

PARALLEL FUNCTION DECOMPOSITION AND SPACE DECOMPOSITION METHODS – PART II. APPLICATIONS TO SPLITTING AND DOMAIN DECOMPOSITION

by
Xue-Cheng Tai

Report No. 95 April 1994

UNIVERSITY OF BERGEN
Bergen, Norway

Department of Mathematics
University of Bergen
5007 Bergen
Norway

ISSN 0084-778x

PARALLEL FUNCTION DECOMPOSITION AND SPACE DECOMPOSITION METHODS – PART II. APPLICATIONS TO SPLITTING AND DOMAIN DECOMPOSITION

by
Xue-Cheng Tai

Report No. 95 April 1994

PARALLEL FUNCTION DECOMPOSITION AND SPACE DECOMPOSITION METHODS – PART II. APPLICATIONS TO SPLITTING AND DOMAIN DECOMPOSITION

XUE-CHENG TAI

University of Bergen,
Department of Mathematics,
Allegt. 55, 5007, Bergen, Norway.

ABSTRACT. This is a continuation of [39]. There, general convergence theory was discussed both for function decomposition and for space decomposition. Here, we are going to use the developed methods to solve partial differential equations and variational inequalities. With the applications, we show that the splitting methods are ways to decompose a function of a minimization. By regarding domain decomposition as ways of function decomposition or space decomposition, we can get some preconditioned domain decomposition methods. We demonstrate by the applications that the methods of [39] can be used for linear problems as well as for nonlinear problems.

CONTENTS

1991 *Mathematics Subject Classification.* 65K10, 65N55, 65Y05.

Key words and phrases. Parallel, domain decomposition, splitting method, function decomposition, space decomposition.

The work was supported by the Technical University of Graz and part of the work was done at the University of Jyväskylä.

This paper is based on Chapters 5–8 of report: Parellel function decomposition and space decomposition methods with application to optimization, splitting and domain decomposition by Xue-Cheng Tai, Preprint no. 231-1992, Institute Für Mathematik, Teschnische Universität Graz, September, 1992

§ 1. Introduction

In paper [**39**], we have proved the convergence of some algorithms involving a general function decomposition or a general space decomposition for a convex programming problem. Here we are going to apply them to some specific partial differential equations and variational inequalities. Our methods can treat linear problems as well as nonlinear problems.

Before we go to the applications, we first recall the basic results of [**39**]. The algorithms in [**39**] are proposed for the minimization problem

$$\min_{v\in K} F(v) \ , \tag{1.1}$$

where, K is a closed convex subset in a Hilbert space V, and the function F is convex. We can reduce problem (1.1) into a number of simpler problems and solve them in parallel if either the function F can be decomposed into the sum of convex functions or the space V can be decomposed into the sum of subspaces.

If the function F can be decomposed into the sum of convex functions, we need the following assumptions to use our proposed algorithms:

(F1). V is a Hilbert space and there exist Hilbert spaces V_i, $i = 1, 2, \cdots, m$ such that

$$V = \bigcap_{i=1}^{m} V_i. \tag{1.2}$$

(F2). $F : V \mapsto \mathbb{R}$ is a convex, lower–semicontinous function in V and there exist convex, lower–semicontinous functions $F_i : V_i \mapsto \mathbb{R}$ in V_i, $i = 1, 2, \cdots, m$ such that

$$F(v) = \sum_{i=1}^{m} F_i(v), \quad \forall v \in V \ . \tag{1.3}$$

(F3). K is a closed convex subset of V. There exist closed, convex subsets $K_i \subset V_i$, $i = 1, 2, \cdots, m$ such that

$$K = \bigcap_{i=1}^{m} K_i \ . \tag{1.4}$$

(F4). There exists a Hilbert space H such that

$$V \subset V_i \subset H, \ i = 1, 2, \cdots, m \ . \tag{1.5}$$

Different algorithms were developed for minimization (1.1) under the above assumptions. The algorithms not only reduce problem (1.1) into a sequence of simpler problems and solve them in parallel, but also offer us different ways in dealing with the constraints K and K_i, $i = 1, 2, \cdots, m$. To prove convergence of these algorithms, we need the following properties of the functions:

(F5). For each i, F_i is differentiable in $V_i \subset H$ and for any $M > 0$, there exists $\delta_M^i : [0, 2M] \mapsto \mathbb{R}^+$, which is continuous, strictly increasing and satisfies $\delta_M^i(0) = 0$ such that

$$\begin{aligned} (F_i'(u_1) - F_i'(u_2), u_1 - u_2)_H \geq \delta_M^i(\|u_1 - u_2\|_H) \\ \forall u_1, u_2 \in V_i, \ \|u_1\|_H, \|u_2\|_H \leq M \ . \end{aligned} \tag{1.6}$$

(F6).

$$\lim_{\|v_i\|_H \to \infty} \frac{F_i(v_i)}{\|v_i\|_H} = +\infty \ , \quad i = 1, 2, \cdots, m \ . \tag{1.7}$$

If each F_i' is coercive, explicit error estimates are given for the proposed algorithms. For the convergence of algorithms of §3.7 of [**39**], we need slightly different conditions than $(F5)$–$(F6)$ to have the convergence. These are assumptions $(F5')$–$(F6')$ in [**39**].

If we do not split the function F of problem (1.1), but we split the space V, then we also can reduce (1.1) to a number of simpler problems and use parallel processors. In this case, we need to assume:

(S1). V is a Hilbert space and there exist Hilbert spaces V_i, $i = 1, 2, \cdots, m$ such that

$$V = V_1 + V_2 + \cdots + V_m \ . \tag{1.8}$$

(S2). K is a closed convex subset in V. There exist closed and convex subsets K_i of V_i, $i = 1, 2, \cdots, m$ such that

$$K = K_1 + K_2 + \cdots + K_m \ . \tag{1.9}$$

(S3). $F(v)$ is uniformly convex and differentiable in V.

Under these assumptions, let us consider the minimization

$$\min_{V_i \in K_i} F(v_1 + v_2 + \cdots + v_m) \ . \tag{1.10}$$

It is simple to observe that if $(u_1, u_2, \cdots u_m)$ is the minimizer of (1.10), then $u = \sum_{i=1}^m u_i$ is the minimizer of (1.1). Problem (1.10) is a minimization problem in a tensor product space. We are going to use the Gauss–Seidel method and the under–relaxation Jacobi method to solve it. Convergence for them was proved under the conditions

(S4).

$$\lim_{\|v\|_V \to +\infty} \frac{F(v)}{\|v\|_V} = +\infty \ . \tag{1.11}$$

(S5). There are constants C_0, C_1 such that

$$C_0 \Big\| \sum_{i=1}^m v_i \Big\|_V^2 \le \sum_{i=1}^m \|v_i\|_V^2, \quad \forall v_i \in V_i, \ i = 1, 2, \cdots, m \tag{1.12}$$

and

$$\begin{cases} \forall v \in V, \ \exists v_i \in V_i, i = 1, 2, \cdots, m \text{ such that} \\ \sum_{i=1}^m v_i = v \text{ and } \sum_{i=1}^m \|v_i\|_V^2 \le C_1 \|v\|_V^2 \ . \end{cases} \tag{1.13}$$

Condition $(S5)$ is always satisfied for the commonly used overlapping domain decomposition, see [9]–[11], [25]–[27], etc. For the case when F' is Lipschitz continuous and coercive, explicit error estimates were obtained in [39]

For space decomposition methods it is sufficient to assume that the spaces V and V_i, $i = 1, 2, \cdots, m$ are just reflexive Banach spaces. For function decomposition methods, the results can also be extended to certain cases when V and V_i, $i = 1, 2, \cdots, m$ are reflexive Banach spaces, see Remark 3.7.4 of [39].

In the following applications, we will use the standard notations for Sobolev spaces and for finite element methods. For a domain $\Omega \subset \mathbb{R}^d$, d is always the dimension, $\partial\Omega$ its boundary and we will denote its unit outer normal vector by

$$\vec{n} = (n_1, n_2, \cdots, n_d) , \quad \sum_{i=1}^{d} n_i^2 = 1.$$

Notations $D_i = \frac{\partial}{\partial x_i}$, $D_i^2 = \frac{\partial^2}{\partial x_i^2}$, etc., will be used. $\frac{\partial}{\partial n}$ denotes the outer normal derivative on $\partial\Omega$. For a given Hilbert space H, an operator $A : \mathrm{Dom}(A) \subset H \mapsto H$ is coercive if $\exists \alpha > 0$ such that

$$(Au_1 - Au_2, u_1 - u_2)_H \geq \alpha \|u_1 - u_2\| .$$

We call F_i locally uniformly convex if $(F5)$ is satisfied.

§ 2. Applications to splitting methods

§ 2.1. A PARALLEL SPLITTING METHOD FROM A PENALIZATION METHOD

In papers by Lu, Neittaanmäki and Tai [30], [31], Lions and Temam [28, p.206] and Bensoussan, Lions and Temam [1], some parallel splitting methods were studied. They use different approaches. Here we will show that these methods coincide with the parallel penalization method when applied to elliptic problems that can be regarded as minimization problems.

We consider an elliptic problem (linear or nonlinear)

$$Au = 0 \tag{2.1.1}$$

and we assume that this equation is derived from the minimization problem

$$\min_{v \in H} F(v) . \tag{2.1.2}$$

Here H is a Hilbert space, F is a convex function

$$F : D(F) \subset H \mapsto \mathbb{R} \tag{2.1.3}$$

and A is its differential in H.

Now, if F can be split as

$$F(v) = \sum_{i=1}^{d} F_i(v) , \quad \forall v \in D(F) , \tag{2.1.4}$$

and

$$F_i : D(F_i) \subset H \mapsto \mathbb{R} \tag{2.1.5}$$

has a differential A_i in H, then (2.1.1) is equivalent to

$$\sum_{i=1}^{d} A_i u = 0 \ . \tag{2.1.6}$$

We assume that $\mathrm{Dom}(F)$ and $\mathrm{Dom}(F_i)$, $i = 1, 2, \cdots, m$ are Hilbert spaces and

$$\mathrm{Dom}(F) = \bigcap_{i=1}^{d} \mathrm{Dom}(F_i) \ .$$

If each F_i is locally uniformly convex and we take

$$K = V = \mathrm{Dom}(F), \ K_i = V_i = \mathrm{Dom}(F_i), \ i = 1, 2 \cdots d \ ,$$

then conditions $(F1) - (F6)$ are satisfied. Using algorithm 3.3.1 of [**39**], we will find

Algorithm 2.1.1.

Step 1. *Choose an initial value $u^0 \in H$ and a parameter r large enough.*

Step 2. *For $n \geq 1$, find $u_i^n \in D(F_i)$ from the following problem in parallel for $i = 1, 2, \cdots, d$:*

$$\min_{v_i \in D(F_i)} \left(F_i(v_i) + \frac{r}{2d} \| v_i - u^n \|_H^2 \right) \ . \tag{2.1.7}$$

This is equivalent to finding $u_i^n \in Dom(F_i)$ such that

$$\frac{r}{d}(u_i^n - u^n) + A_i u_i^n = 0 \ . \tag{2.1.8}$$

Step 3. *Set*

$$u^{n+1} = \frac{1}{d} \sum_{i=1}^{d} u_i^n \tag{2.1.9}$$

and go to the next iteration.

By defining $\tau = \frac{1}{r}$, we can see that this is exactly the algorithm studied in Lu, Neittaanmäki and Tai [**30**], [**31**]. The convergence is proved in [**30**] under the assumption that each A_i is coercive.

Remark 2.1.1. As we see from [**39**], the local uniform convexity of F_i is very important in getting the convergence for algorithm 3.4.1 of [**39**]. The same is true for the penalization methods. This restricts the applications of these methods mostly to Dirichlet boundary conditions. When, for example, the Neumann boundary condition is used, an integration by parts for a differential operator will produce a

boundary term. This term destroys the ellipticity of the differential operator and therefore destroys the convergence of the algorithms.

To see it clearly, let us take equations

$$-\sum_{i=1}^{d}(D_i^2 u - f_i) = 0 \quad \text{in } \Omega \,, \tag{2.1.10}$$

$$u = 0 \quad \text{on } \partial\Omega \,, \tag{2.1.11}$$

$$\text{or } \frac{\partial u}{\partial n} = 0 \quad \text{on } \partial\Omega \,. \tag{2.1.12}$$

If the Dirichlet boundary condition (2.1.11) is enforced, then we can take

$$V = H_0^1(\Omega), \quad H = L^2(\Omega) \,, \tag{2.1.13}$$

$$V_i = \{v| \quad v,\ D_i v \in L^2(\Omega) \,,\ v|_{\partial\Omega} = 0\} \,, \tag{2.1.14}$$

$$F(v) = \int_\Omega \sum_{i=1}^{d} \left(\frac{1}{2}|D_i v|^2 - f_i v\right) dx \,, \tag{2.1.15}$$

$$F_i(v) = \int_\Omega \left(\frac{1}{2}|D_i v|^2 - f_i v\right) dx \tag{2.1.16}$$

and $F_i'(v_i)$ is coercive in $H = L^2(\Omega)$ for $v_i \in V_i$. Thus, we can use algorithm 2.1.1.

If the Neumann boundary condition (2.1.12) is enforced, we need to define

$$\begin{aligned} F(v) &= \int_\Omega -\sum_{i=1}^{d}(\frac{1}{2}D_i^2 v - f_i)\cdot v dx \,, \\ F_i(v) &= \int_\Omega -(\frac{1}{2}D_i^2 v - f_i)\cdot v dx \,. \end{aligned} \tag{2.1.17}$$

It can be easily checked that for $v_1,\ v_2 \in V_i$

$$\begin{aligned} &(F_i'(v_1) - F_i'(v_2), v_1 - v_2)_H \\ &= \int_\Omega |D_i(v_1 - v_2)|^2 dx - \int_{\partial\Omega} D_i(v_1 - v_2)\cdot(v_1 - v_2)n_i ds \,, \end{aligned} \tag{2.1.18}$$

which can even be negative. In order to avoid this, we may define the spaces in another way, i.e.

$$V = \{v| \quad v \in H^2(\Omega),\ \frac{\partial v}{\partial n} = 0 \ \text{ on } \partial\Omega\} \,, \tag{2.1.19}$$

$$V_i = \{v| \quad v,\ D_i v,\ D_i^2 v \in L^2(\Omega),\ D_i v = 0 \ \text{ on } \partial\Omega\} \,, \tag{2.1.20}$$

$$H = L^2(\Omega) \,, \tag{2.1.21}$$

and use the same definitions (2.1.15)–(2.1.16), we will get that F_i' is coercive for $v_i \in V_i$, but now

$$V \neq \bigcap_{i=1}^{d} V_i \,, \tag{2.1.22}$$

so we cannot get the convergence. Some numerical tests have been carried out using the penalization methods. The numerical tests show that for the Neumann boundary condition, if $D_i u \neq 0$ on $\partial\Omega$ $i = 1, 2, \cdots, d$, then we can only have $\|e^n - e_i^n\|_H \to 0$, but not $\|e^n\|_H \to 0$. If $D_i u = 0$ on $\partial\Omega$, then $u \in \bigcap_{i=1}^d V_i$ and the numerical results show a first order of convergence for parabolic problems with step size $\tau = r^{-1}$.

Remark 2.1.2. The parallel splitting does not mean that we can only use d processors. In the dimensional splitting case, each subproblem is again a series of independent one dimensional problems, see Tai and Neittaanmäki [**40**], and they can be computed again by parallel processors. If we solve each subproblem by L processors, then we need $d \times L$ processors totally.

§ 2.2. The alternating direction method and the local one dimensional method

The relationship between the alternating direction method and the augmented Lagrangian method has been discussed in the book by Glowinski and Le Tallec [**16**, p. 89]. Here we will show that a small change in the penalization function of the last section will give us the local one dimensional method. Moreover such a small change also turns the splitting method from a parallel one to a sequential one, which means the fractional steps are not independent, but must be solved one after another.

If (2.1.4) is true and $V = \mathrm{Dom}(F)$, $V_i = \mathrm{Dom}(F_i)$ are Hilbert spaces, we can see that the minimization problem (2.1.2) is equivalent to solving

$$\min_{\substack{v_i \in V_i, v \in V \\ v_i = v}} \sum_{i=1}^d F_i(v_i) \ . \tag{2.2.1}$$

We put constraints $v_i = v$, $i = 1, 2, \cdots, d$ as penalization terms into the cost function and consider

$$\min_{v_i \in V_i, v \in H} \left[\sum_{i=1}^d F_i(v_i) + \frac{r}{2d} \sum_{i=1}^d \|v_i - v\|_H^2 \right] . \tag{2.2.2}$$

It was proved in §3 of [**39**] that the minimizer of (2.2.2) converges to the minimizer of (2.2.1) as $r \to \infty$. Algorithm 2.1.1 is an iterative method for minimization (2.2.2). It is simple to observe that (2.2.1) can also be written as

$$\min_{\substack{v_i \in V_i \\ v_1 = v_2 = \cdots = v_m}} \sum_{i=1}^d F_i(v_i) \ . \tag{2.2.3}$$

Instead of (2.2.2), if we put

$$v_{i-1} = v_i \ , \quad i = 2, \cdots d \tag{2.2.4}$$

as penalization terms into the cost function, i.e. if we define the penalization function as

$$\sum_{i=1}^d F_i(v_i) + \frac{r}{2d} \sum_{i=2}^d \|v_{i-1} - v_i\|_H^2 \tag{2.2.5}$$

and use the Gauss–Seidel method to minimize this function, we will get the following algorithm

Algorithm 2.2.1.

Step 1. *Choose initial values* $u^0 \in D(F)$.

Step 2. *For* $n \geq 0$, *set* $u^{n+1} = u_d^n$, $u_0^n = u^{n-1}$ *and find* $u_i^n \in Dom(F_i)$ *sequentially for* $i = 1, 2, \cdots, d$ *by solving*

$$\frac{r}{d}(u_i^n - u_{i-1}^n) + A_i u_i^n = 0 \ . \tag{2.2.6}$$

Step 3. *Go to the next iteration.*

If we take $\tau = \frac{d}{r}$, this is the well-known local one dimensional method, see Yanenko [**44**] and Marchuk [**34**]. Its convergence can be proved as in the proof of the convergence of the penalization method, see [**39**].

§ 2.3. The augmented parallel splitting method

The splitting and parallel splitting methods derived from penalization methods introduce splitting errors. The accuracy depends on the parameter r. For fixed r, if the iteration number n is getting large, the error may increase. The splitting method derived from the augmented Lagrangian method does not have this drawback. The computed solution will become more and more accurate if more and more steps of iterations are performed.

We apply Algorithm 3.5.1 of [**39**] to equation (2.1.6). We will have

Algorithm 2.3.1.

Step 1. *Choose initial values* $\lambda_i^0 \in H$, *and* $u_i^0 \in D(F_i)$, $i = 1, 2, \cdots, d$.

Step 2. *For* $n \geq 1$, *set*

$$u^n = \frac{1}{d}\sum_{i=1}^{d} u_i^{n-1} + \frac{1}{rd}\sum_{i=1}^{d} \lambda_i^{n-1} \ . \tag{2.3.1}$$

Step 3. *Find* $u_i^n \in D(F_i)$ *in parallel for* $i = 1, 2, \cdots, d$ *such that*

$$\frac{r}{d}(u_i^n - u^n) + A_i u_i^n + \frac{1}{d}\lambda_i^{n-1} = 0 \ . \tag{2.3.2}$$

Step 4. *Update the multipliers as*

$$\lambda_i^n = \lambda_i^{n-1} + \rho(u_i^n - u^n) \tag{2.3.3}$$

and go to the next iteration.

The bulk of computation is in step 3, but they can be done in parallel for different i. Step 2 and Step 4 are only arithmetic averages. This algorithm is as easy and as quick as the algorithms of §2.1–2.2, but it is free of splitting errors. For its convergence, we need not to take $r \to \infty$. For any $r > 0$, if $0 < \rho < \frac{1+\sqrt{5}}{2} r$, then $u^n \to u$.

§ 2.4. A PARALLEL SPLITTING METHOD FOR GENERAL BOUNDARY CONDITIONS

Consider the elliptic equation

$$-\sum_{i=1}^{d} D_i(a_i(x)D_iu) + c(x)u = f(x) \quad \text{in } \Omega \subset \mathbb{R}^d \tag{2.4.1}$$

with boundary conditions

$$u = 0 \quad \text{on } \Gamma_1 \subset \partial\Omega\ , \tag{2.4.2}$$

$$\sum_{i=1}^{d} a_i(x)D_iun_i + b(x)u = 0 \quad \text{on } \Gamma_2 \subset \partial\Omega\ . \tag{2.4.3}$$

Above, $0 < c_1 \le a_i(x) \le c_2 < \infty$, $i = 1, 2, \cdots, d$, $c(x) \ge 0$, $b(x) \ge 0$ and $\Gamma_1 \cup \Gamma_2 = \partial\Omega$. Moreover, there exists $c_3 > 0$ such that $c(x) \ge c_3$ or $b(x) \ge c_3$. If $\Gamma_2 = \emptyset$, then this is the Dirichlet boundary condition. If $\Gamma_1 = \emptyset$ and $b = 0$, then this is the Neumann boundary condition; if $b > 0$, this is the third boundary condition. The following approach can deal with all these boundary conditions.

Let's define

$$V = \{v| \quad v \in H^1(\Omega),\ v = 0 \quad \text{on } \Gamma_1\}\ , \tag{2.4.4}$$

$$F(v) = \frac{1}{2}\int_\Omega (\sum_{i=1}^{d} a_i|D_iv|^2 + cv^2)dx - \int_\Omega fvdx + \frac{1}{2}\int_{\Gamma_2} bv^2ds\ . \tag{2.4.5}$$

V is a Hilbert space with the usual H^1–norm and inner product. If $\Gamma_1 \neq \emptyset$, then the corresponding seminorm is a norm. If $\Gamma_1 = \emptyset$, then $V = H^1(\Omega)$. If $\Gamma_1 = \partial\Omega$, then $V = H_0^1$.

It is generally difficult to use splitting methods for Neumann boundary conditions, see Remark 2.1.1 and Marchuk and Kuznetsov [**35**]. For more general boundary conditions, it is even more difficult. In the following, we will see that if we use Algorithm 3.7.1 of [**39**], we will get a parallel splitting method which has a natural preconditioner in the discrete case and is convergent for (2.4.1)–(2.4.3).

Now, let us specify the functions and spaces. Assume f and c can be split as

$$f = \sum_{i=1}^{d} f_i,\ c = \sum_{i=1}^{d} c_i\ , \tag{2.4.6}$$

then, we can define

$$V_i = \{v| \quad v,\ D_iv \in L^2(\Omega),\ v|_{\Gamma_1} = 0\}, \quad K_i = V_i\ , \tag{2.4.7}$$

$$F_i(v) = \frac{1}{2}\int_\Omega (a_i|D_iv|^2 + c_iv^2)dx - \int_\Omega f_ivdx + \frac{1}{2}\int_{\Gamma_2} bn_i^2v^2ds\ . \tag{2.4.8}$$

Space V_i is equipped with the inner product

$$(u,v)_{V_i} = (u,v)_{L^2(\Omega)} + (D_i u, D_i v)_{L^2(\Omega)} .$$

As $\sum_{i=1}^{d} n_i^2 = 1$ on $\partial\Omega$, it is true that

$$F(v) = \sum_{i=1}^{d} F_i(v) , \tag{2.4.9}$$

$$V = \bigcap_{i=1}^{d} V_i . \tag{2.4.10}$$

With these definitions, F_i is not differentiable in $L^2(\Omega)$ because of the boundary term. However, we will show that F_i is differentiable in V_i, and $F_i' : V_i \mapsto V_i$ is coercive.

For any $v \in V_i$, let $\phi_i(v) = F_i'(v) \in V_i$. By definition, see [**39**, §2], it means that

$$\lim_{t\to 0+} \frac{F_i(v+tw) - F_i(v)}{t} = (\phi_i, w)_{V_i} , \quad \forall w \in V_i . \tag{2.4.11}$$

It is easy to check that

$$\begin{aligned} &\lim_{t\to 0+} \frac{F_i(v+tw) - F_i(v)}{t} \\ &= \int_\Omega (a_i D_i v D_i w + c_i v w - f_i w)dx + \int_{\Gamma_2} b n_i^2 v w ds . \end{aligned} \tag{2.4.12}$$

It follows from (2.4.11) and (2.4.12) that ϕ_i should satisfy

$$\begin{aligned} &\int_\Omega (a_i D_i v D_i w + c_i v w - f_i w)dx + \int_{\Gamma_2} b n_i^2 v w ds \\ &= \int_\Omega (D_i \phi_i D_i w + \phi_i w)dx , \quad \forall w \in V_i . \end{aligned} \tag{2.4.13}$$

Following the proof of Theorem 1.1 of Tai and Neittaanmäki [**40**], we can prove that there exists one and only one ϕ_i satisfying (2.4.13). This ϕ_i is the solution of

$$\begin{cases} -D_i^2 \phi_i + \phi_i = -D_i(a_i D_i v) + c_i v - f_i \quad \text{in } \Omega , \\ D_i \phi_i = a_i D_i v - b n_i v \quad \text{on } \Gamma_2 , \\ \phi_i = 0 \quad \text{on } \Gamma_1 . \end{cases} \tag{2.4.14}$$

To show the coercivity of $F_i' : V_i \mapsto V_i$, we need to prove that there exists $\alpha > 0$ such that

$$(F_i'(v_1) - F_i'(v_2), v_1 - v_2)_{V_i} \geq \alpha \|v_1 - v_2\|_{V_i}^2 . \tag{2.4.15}$$

Writing $v = v_1 - v_2$, we can get from (2.4.12) that

$$\begin{aligned} &(F_i'(v_1) - F_i'(v_2), v_1 - v_2)_{V_i} \\ &= \int_\Omega (a_i |D_i v|^2 + c_i v^2)dx + \int_{\Gamma_2} b n_i^2 v^2 ds \\ &\geq \min(a_i, c_i) \|v\|_{V_i}^2 . \end{aligned} \tag{2.4.16}$$

This proves (2.4.15).

Therefore, the conditions of Theorem 3.7.1 of [**39**] are fulfilled and algorithm 3.7.1 and algorithm 3.7.2 of [**39**] are convergent. To use algorithm 3.7.1, in step 2 we need to find $u^n \in V$ such that

$$\begin{aligned}
&(\nabla u^n, \nabla v)_{L^2(\Omega)} + (u, v)_{L^2(\Omega)} \\
=&\sum_{i=1}^{d}[(D_i u_i^n, D_i v)_{L^2(\Omega)} + (u_i^n, v)_{L^2(\Omega)}] \\
&+ \frac{1}{r}\sum_{i=1}^{d}[(D_i \lambda_i^{n-1}, D_i v)_{L^2(\Omega)} + (\lambda_i^{n-1}, v)_{L^2(\Omega)}] \ , \quad \forall v \in V \ .
\end{aligned} \tag{2.4.17}$$

In step 3 we need to find $u_i^n \in V_i$ in parallel for $i = 1, 2, \cdots, d$ such that

$$\begin{aligned}
&(a_i D_i u_i^n, D_i v_i)_{L^2(\Omega)} + (c_i u_i^n, v_i)_{L^2(\Omega)} + (bn_i^2 u_i^n, v_i)_{L^2(\Gamma_2)} \\
&\quad + \frac{r}{d}\left[(D_i(u_i^n - u^n), D_i v_i)_{L^2(\Omega)} + (u_i^n - u^n, v_i)_{L^2(\Omega)}\right] \\
=&(f, v_i)_{L^2(\Omega)} - \frac{1}{d}\left[(D_i \lambda_i^{n-1}, D_i v_i)_{L^2(\Omega)} + (\lambda_i^{n-1}, v_i)_{L^2(\Omega)}\right] \ , \\
&\quad \forall v_i \in V_i \ .
\end{aligned} \tag{2.4.18}$$

Step 4 is just the updating of the multipliers: choose $0 < \rho_n < 2r$ and set

$$\lambda_i^n = \lambda_i^{n-1} + \rho_n(u_i^n - u^n) \ , i = 1, 2, \cdots, d \ . \tag{2.4.19}$$

To write them in a partial differential form, equation (2.4.17) is

$$\begin{cases}
-\Delta u^n + u^n = h^n & \text{in } \Omega \ , \\
u^n = 0 & \text{on } \Gamma_1 \ , \\
\dfrac{\partial u^n}{\partial n} = g^n & \text{on } \Gamma_2 \ ,
\end{cases} \tag{2.4.20}$$

where

$$h^n = \sum_{i=1}^{d}(-D_i^2 u_i^n + u_i^n) + \frac{1}{r}\sum_{i=1}^{d}(-D_i^2 \lambda_i^{n-1} + \lambda_i^{n-1}) \ , \tag{2.4.21}$$

$$g^n = \sum_{i=1}^{d}(D_i u_i^n n_i + \frac{1}{r} D_i \lambda_i^{n-1} n_i) \tag{2.4.22}$$

and equation (2.4.18) is

$$\begin{cases}
\dfrac{r}{d}[-D_i^2(u_i^n - u^n) + u_i^n - u^n] - D_i(a_i D_i u_i^n) + c_i u_i^n \\
\quad = f_i - \dfrac{1}{d}(\lambda_i^{n-1} - D_i^2 \lambda_i^{n-1}) \quad \text{in } \Omega \ , \\
u_i^n = 0 \quad \text{on } \Gamma_1 \ , \\
a_i D_i u_i^n + bn_i u_i^n + \dfrac{r}{d} D_i(u_i^n - u^n) + \dfrac{1}{d} D_i \lambda_i^{n-1} = 0 \quad \text{on } \Gamma_2 \ .
\end{cases} \tag{2.4.23}$$

For each i, (2.4.23) is a series of independent one dimensional problems, which can be solved using one dimensional methods in parallel, see Tai and Neittaanmäki [**40**] and Tai [**38**]. In the discrete case, equation (2.4.23) has a preconditioner $(I - D_i{}^2)^{-1}$.

If we multiply the boundary condition of (2.4.23) on Γ_2 by n_i and sum up for $i = 1, 2, \cdots, d$, we will find that

$$\sum_{i=1}^{d}(a_i D_i u_i^n n_i + b_i n_i^2 u_i^n) + \frac{r}{d}\sum_{i=1}^{d}(D_i u_i^n n_i + \frac{1}{r} D_i \lambda_i^{n-1} n_i) - \frac{r}{d}\frac{\partial u^n}{\partial n} = 0 \quad \text{on } \Gamma_2 \ . \tag{2.4.24}$$

From the definition of g^n in (2.4.22), this means that

$$\sum_{i=1}^{d}(a_i D_i u_i^n n_i + b_i n_i^2 u_i^n) + \frac{r}{d} g^n - \frac{r}{d}\frac{\partial u^n}{\partial \Omega} = 0 \quad \text{on } \Gamma_2 \ .$$

But in (2.4.20) we have $\frac{\partial u^n}{\partial n} = g^n$, thus

$$\sum_{i=1}^{d} a_i u_i^n n_i + \sum_{i=1}^{d} b n_i^2 u_i^n = 0 \ . \tag{2.4.25}$$

From this, we see that to solve equation (2.4.20) is to project an element from $\prod_{i=1}^{d} V_i$ to the diagonal subspace V and this projection enforces the condition (2.4.25) to be valid on Γ_2.

Algorithm 2.4.1.

Step 1. *Choose initial values $\lambda_i^0 \in V_i$, $i = 1, 2, \cdots, d$, then for $n \geq 1$,*
Step 2. *solve $u^n \in V$ such that it satisfies (2.4.20).*
Step 3. *Solve $u_i^n \in V_i$ in parallel for $i = 1, 2, \cdots, d$ such that it satisfies (2.4.23).*
Step 4. *Update the multipliers as in (2.4.19) and go to step 2.*

This algorithm is efficient only if we have a fast solver for (2.4.20). From Theorem 3.7.1, the convergence for this algorithm is not only in $L^2(\Omega)$, but in the norms of V and V_i, i.e.

$$\|u_i^n - u\|_{V_i} \to 0 \ , \quad n \to \infty \ , \ \forall i \ , \tag{2.4.26}$$

$$\|u^n - u\|_V \to \ , \quad n \to \infty \ . \tag{2.4.27}$$

For this algorithm, we get both inner and outer iteration preconditioners by paying the effort to solve (2.4.20) in each iteration.

The above algorithm can also be generalized for the more general elliptic equation

$$\begin{cases} -\sum_{i=1}^{d}\sum_{j=1}^{d} D_i(a_{ij}(x,u)D_j u) + c(x,u)u = f(x,u,\nabla u) \quad \text{in } \Omega \subset \mathbb{R}^d \ , \\ u = 0 \quad \text{on } \Gamma_1 \subset \partial\Omega \ , \\ \sum_{i,j=1}^{d} a_{ij}(x,u) D_j u n_i + b(x,u)u = g(x,u) \quad \text{on } \Gamma_2 \subset \partial\Omega \ . \end{cases} \tag{2.4.28}$$

We assume

$$a_{ij}(x,u) = a_{ji}(x,u) \ , \quad \forall i,j,\forall x,u \ . \tag{2.4.29}$$

We do not specify the other assumptions on the functions.

Let's first assume (2.4.28) is a linear elliptic equation, i.e. the functions a_{ij}, c, f, b and g are independent of u and only depend on x. A selfadjoint elliptic equation is equivalent to a minimization problem. This means that (2.4.28) can be solved by

$$\min_{\substack{v\in H^1(\Omega)\\ v|_{\Gamma_1}=0}} F(v) \ . \tag{2.4.30}$$

Above

$$F(v) = \frac{1}{2}\int_\Omega (\sum_{i=1}^d \sum_{j=1}^d a_{ij} D_i v D_j v + cv^2)dx - \int_\Omega fv dx + \frac{1}{2}\int_{\Gamma_2} bv^2 ds - \int_{\Gamma_2} gv ds \ . \tag{2.4.31}$$

If we split b, c, f, and g as

$$b = \sum_{i=1}^d bn_i^2, \ c = \sum_{i=1}^d c_i, \ f = \sum_{i=1}^d f_i, \ g = \sum_{i=1}^d g_i \ , \tag{2.4.32}$$

and define

$$\begin{aligned} F_{ii}(v,v) = &\frac{1}{4}\int_\Omega (a_{ii}|D_i v|^2 + c_i v^2)dx - \frac{1}{2}\int_\Omega f_i v dx \\ &+ \frac{1}{4}\int_{\Gamma_2} bn_i^2 v^2 ds - \frac{1}{2}\int_{\Gamma_2} g_i v ds \ , \end{aligned} \tag{2.4.33}$$

$$F_{ij}(v_i, v_j) = \frac{1}{2}\int_\Omega a_{ij} D_i v_i D_j v_j dx \ , \quad i \neq j \ , \tag{2.4.34}$$

then

$$F(v) = \sum_{i=1}^d \sum_{j=1}^d F_{ij}(v,v) \ , \quad \forall v \in H^1(\Omega) \ . \tag{2.4.35}$$

We can use the augmented Lagrangian method to turn (2.4.30) into a minimization problem in a tensor product space. Let us define V_i as the same as in (2.4.7), i.e.

$$V_i = \{v| \quad v, D_i v \in L^2(\Omega), \ v = 0 \text{ on } \Gamma_1\} \ . \tag{2.4.36}$$

and define X as

$$X = \prod_{i=1}^d V_i = \{(v_1, v_2, \cdots, v_d)| \quad v_i \in V_i, \ i = 1,2,\cdots,d\}. \tag{2.4.37}$$

We can see that

$$F_{ii} : V_i \times V_i \mapsto \mathbb{R} \ , \tag{2.4.38}$$

$$F_{ij} : V_i \times V_j \mapsto \mathbb{R} \tag{2.4.39}$$

are well defined. Define

$$J(v_1, v_2, \cdots, v_d) = \sum_{i=1}^{d} \sum_{j=1}^{d} F_{ij}(v_i, v_j) \ . \tag{2.4.40}$$

We see that (2.4.30) is the same as

$$\min_{\substack{(v_1, v_2, \cdots, v_d) \in X \\ v_1 = v_2 = \cdots = v_d}} J(v_1, v_2, \cdots, v_d) \ . \tag{2.4.41}$$

Define the augmented Lagrangian function for J as

$$\begin{aligned} J_r(v, v_i, \mu_i) = J(v_1, v_2, \cdots, v_d) + \frac{r}{d} \sum_{i=1}^{d} \|v_i - v\|_{V_i}^2 \\ + \frac{1}{d} \sum_{i=1}^{d} (v_i - v, \mu_i)_{V_i} \ . \end{aligned} \tag{2.4.42}$$

For known u^n, λ_i^{n-1}, $i = 1, 2, \cdots, d$, if we use the Gauss–Seidel method to minimize

$$\begin{aligned} J_r(u^n, v_i, \lambda_i^{n-1}) = J(v_1, v_2, \cdots, v_d) + \frac{r}{d} \sum_{i=1}^{d} \|v_i - u^n\|_{V_i}^2 \\ + \frac{1}{d} \sum_{i=1}^{d} (v_i - u^n, \lambda_i^{n-1})_{V_i} \ , \end{aligned} \tag{2.4.43}$$

we will get a sequential splitting method. If we use the Jacobi method, we will get a parallel one. The extension to general nonlinear equation (2.4.28) is to combine the above procedure with the following linearization

$$\begin{cases} -\sum_{i=1}^{d} \sum_{j=1}^{d} D_i(a_{ij}(x, u^n) D_j u^{n+1}) + c(x, u^n) u^{n+1} = f(x, u^n, \nabla u^n) \text{ in } \Omega \ , \\ u^{n+1} = 0 \text{ on } \Gamma_1 \\ \sum_{i,j=1}^{d} a_{ij}(x, u^n) D_j u^{n+1} n_i + b(x, u^n) u^{n+1} = g(x, u^n) \text{ on } \Gamma_2 \ . \end{cases} \tag{2.4.44}$$

Such a linearization approach has been used in literature, see Douglas and Dupont [6], Hayes [18], [19], [20], Douglas, Dupont and Percel [7], Hlaváček, Křizek and Maly [21] etc. The convergence can be proved under certain conditions. As the convergence does not follow from the analysis of §3, we may report on it in a later paper.

Remark 2.4.1. The finite element method is flexible in dealing with general boundary conditions on complex geometry. In the literature, there are some papers that try first to discretize the equation (2.4.28) by the linear finite element method

and then split the matrices in some way. In Marchuk and Kuzin [**36**], the linear finite element method is used first and then the matrices are split into four parts, actually in four directions. In Hayes [**18**], [**19**], [**20**], the finite element patch approximation method is used, which first discretizes a domain Ω into triangular or polyhedral elements, then there exists a mapping which transforms each element to a triangle or a rectangle. The union of these triangles and rectangles can be, for example, a unit square. If we use the dimensional splitting in the unit square and use patch approximations, it corresponds to alternating along curved element boundaries in the original complex region.

For special boundary conditions, there exist some methods. In Fryjazinov [**13**], a method to deal with the third boundary condition was proposed. The convergence is of half order with the penalization parameter r. In Dryja [**8**], a linear finite element method was studied for the third boundary condition.

If the domain Ω is the union of rectangles, the boundary conditions are easy to deal with. In the literature, there are many papers that consider the splitting methods in rectangular domains. If the domain is a general domain, but only with the Dirichlet boundary condition, then it is also easy to use the splitting methods. In fact, in the literature, most of the papers deal with Dirichlet boundary conditions. If the Dirichlet boundary conditions depend on time, special care should be taken, see [**44**], and [**22**], [**37**].

§ 2.5. Applications to variational inequalities

As one example of application, we will use algorithm 3.6.1 of [**39**] to solve the obstacle problem

$$\begin{cases} -\Delta u = f \text{ in } \Omega \subset \mathbb{R}^d , \\ u \geq \phi \text{ in } \Omega , \\ u = 0 \text{ and } \phi \leq 0 \text{ on } \partial\Omega , \end{cases} \tag{2.5.1}$$

see Ciarlet [**4**, p.289] for specific assumptions on the functions. We split f as

$$f = \sum_{i=1}^{d} f_i \tag{2.5.2}$$

and define the functions F, F_i as

$$F(v) = \int_\Omega \left(\frac{1}{2} |\nabla v|^2 - fv \right) dx ,$$
$$F_i(v) = \int_\Omega \left(\frac{1}{2} |D_i v|^2 - f_i v \right) dx .$$

Spaces V, H and V_i are taken as the same as in (2.1.13), (2.1.14). If we take

$$K = \{v| \quad v \in H_0^1(\Omega),\ v \geq \phi \text{ a.e. in } \Omega\}, \quad K_i = K,\ \forall i,$$

then the conditions $(F1)$–$(F7)$ are satisfied. We use algorithm 3.6.1 of [**39**] with step 2 and step 3 decoupled for (2.5.1). This gives

Algorithm 2.5.1.

Step 1. *Choose initial values* $\lambda_i^0 \in L^2(\Omega)$, $u_i^0 \in V_i$ *for* $i = 1, 2, \cdots, d$.

Step 2. *For* $n \geq 1$, *set*

$$u^n = \max(\phi, \frac{1}{d}\sum_{i=1}^{d} u_i^{n-1} + \frac{1}{rd}\sum_{i=1}^{d} \lambda_i^{n-1}) \ . \tag{2.5.3}$$

Step 3. *Find* $u_i^n \in V_i$ *in parallel for* $i = 1, 2, \cdots, d$ *such that*

$$\frac{r}{d}(u_i^n - u^n) - D_i^2 u_i^n = f_i - \frac{1}{d}\lambda_i^{n-1} \ . \tag{2.5.4}$$

Step 4 *Update multipliers and go to the next iteration:*

$$\lambda_i^n = \lambda_i^{n-1} + \rho_n(u_i^n - u^n) \ . \tag{2.5.5}$$

Above, step 2 is the projection from $L^2(\Omega)$ to the constraint set K. The operator "max" is in the distribution sense. In step 3, (2.5.4) is an independent two point boundary problem with a homogeneous Dirichlet boundary condition in every line in the x_i–direction. Each one dimensional problem is as simple as a Laplace equation. They can be solved by parallel processors, see Tai [**38**]. Equation (2.5.4) does not have a preconditioner like equation (2.4.23), because the penalization here is done in the space $L^2(\Omega)$.

§ 3. Applications to nonoverlapping domain decomposition

§ 3.1. Introduction

A given energy function over a domain Ω can always be decomposed into the sum of the energies over the subdomains, regardless of whether the minimization of the energy function leads to linear or nonlinear problems. Let us take the minimization problem

$$\min_{v \in V} \int_\Omega \mathbb{L}(v) dx \ . \tag{3.1.1}$$

If $\mathbb{L}$ is differentiable in V, minimization (3.1.1) is equivalent to solving

$$\mathbb{L}'(u) = 0 \ , \quad u \in V \ . \tag{3.1.2}$$

If we partition the domain into nonoverlapping subdomains Ω_i, $i = 1, 2, \cdots, m$,

$$\bar{\Omega} = \bigcup_{i=1}^{m} \bar{\Omega}_i \ , \Omega_i \cap \Omega_j = \emptyset \ , \ i \neq j \ , \tag{3.1.3}$$

then

$$\int_\Omega \mathbb{L}(v) dx = \sum_{i=1}^{m} \int_{\Omega_i} \mathbb{L}(v) dx \ . \tag{3.1.4}$$

By choosing suitable spaces, the problem will fit into the theory we studied in §3 of [**39**]. Therefore, we can reduce (3.1.1) to smaller size problems and use parallel methods to solve it. In the following sections, we will use a specific mapping $\mathbb{L}$. In §3.2, by applying the projection multipliers method of §3.7 of [**39**] to a linear elliptic equation, we get a preconditioned parallel domain decomposition method. This method needs to solve an equation like the Laplace equation in each step. In §3.4, we transform the solving of a partial differential equation into a minimization problem of functions on the interfaces, and thus get a parallel, but not preconditioned, domain decomposition. The method is neat and easy to implement. Applications to nonlinear and variational problems are also discussed in §3.2 and §3.3.

§ 3.2. Parallel preconditioned domain decomposition method

Let's solve the elliptic linear equation

$$\begin{cases} -\sum_{i=1}^{d}\sum_{j=1}^{d}(D_i a_{ij}(x) D_j u) + c(x)u = f(x) & \text{in } \Omega \subset \mathbb{R}^d , \\ u = 0 \text{ on } \partial\Omega . \end{cases} \tag{3.2.1}$$

More general boundary conditions can also be considered, see Remark 3.2.2. As usual, we assume

$$a_{ij} \in L^\infty(\Omega), a_{ij} = a_{ji}, \quad \forall i, j. \tag{3.2.2}$$

$$\sum_{i=1}^{d}\sum_{j=1}^{d} a_{ij}(x)\xi_i\xi_j \geq c_1 \sum_{i=1}^{d} \xi_i^2 \quad \text{in } \Omega . \tag{3.2.3}$$

$$c \in L^\infty(\Omega), \quad c \geq 0 \text{ in } \Omega . \tag{3.2.4}$$

For simplicity, we use the notations

$$Av = -\sum_{i=1}^{d}\sum_{j=1}^{d} D_i(a_{ij} D_j v) + cv , \tag{3.2.5}$$

$$Bv = -\Delta v + v . \tag{3.2.6}$$

We partition the domain Ω into nonoverlapping subdomains Ω_k, $k = 1, 2, \cdots, m$. We use

$$n^k = (n_1^k, n_2^k, \cdots, n_d^k) \tag{3.2.7}$$

to denote the unit outer normal vector of $\partial\Omega_k$ and use

$$\frac{\partial v}{\partial n_A^k} = \sum_{i,j=1}^{d} a_{ij} D_j v n_i^k , \tag{3.2.8}$$

$$\frac{\partial v}{\partial n^k} = \sum_{i=1}^{d} D_i v n_i^k , \text{ and } \frac{\partial v}{\partial n_B^k} = \frac{\partial v}{\partial n^k} \tag{3.2.9}$$

to denote the normal derivatives.

With assumptions (3.2.2)–(3.2.4), the elliptic equation is a minimization problem

$$\min_{v\in H_0^1(\Omega)} F(v) , \tag{3.2.10}$$

where

$$F(v)=\sum_{k=1}^{m}\Big[\int_{\Omega_k}\frac{1}{2}\Big(\sum_{i=1}^{d}\sum_{j=1}^{d}a_{ij}D_ivD_jv+cv^2\Big)dx-\int_{\Omega_k}fvdx\Big] . \tag{3.2.11}$$

Let us define

$$H=\{v|\quad v\in H^1(\Omega_k),\ k=1,2,\cdots,m\} \tag{3.2.12}$$

with inner product

$$(u,v)_H=\sum_{i=1}^{m}(u,v)_{H^1(\Omega_k)} . \tag{3.2.13}$$

It is clear that $F(v)$ is also well defined over H.

We regard $H_0^1(\Omega)$ as a constraint set in H, i.e. define

$$V=H,\quad K=H_0^1(\Omega) \tag{3.2.14}$$

and write (3.2.10) as

$$\min_{v\in K}F(v) ,\quad K\subset V . \tag{3.2.15}$$

If we use the project multipliers method Algorithm 3.7.3 of [**39**] (with step 2 and step 3 decoupled) for (3.2.15), we will get the following algorithm

Algorithm 3.2.1.

Step 1. *Choose initial values* $\lambda^0\in H$ *and* $u^{\frac{1}{2}}\in H$.

Step 2. *For* $n\geq 1$, *solve* $u^n\in H_0^1(\Omega)$ *from*

$$(u^n,v)_{H^1(\Omega)}=\sum_{i=1}^{m}(u^{n-\frac{1}{2}}+\lambda^{n-1},v)_{H^1(\Omega_i)},\quad \forall v\in H_0^1(\Omega) . \tag{3.2.16}$$

which can be informally written as

$$\begin{cases} Bu^n=Bu^{n-\frac{1}{2}}+\dfrac{1}{r}B\lambda^{n-1} & \text{in } \Omega ,\\ u^n=0 \quad \text{on } \partial\Omega . \end{cases} \tag{3.2.17}$$

Step 3. *Find* $u^{n+\frac{1}{2}}\in H^1(\Omega_k)$ *in parallel in each of the subdomains* $\Omega_k, k=1,2,\cdots,m$ *from the following equation*

$$\begin{cases} rB(u^{n+\frac{1}{2}}-u^n)+Au^{n+\frac{1}{2}}=f-B\lambda^{n-1} & \text{in } \Omega_k ,\\ \dfrac{\partial u^{n+\frac{1}{2}}}{\partial n_A^k}+\dfrac{\partial\lambda^{n-1}}{\partial n_B^k}+r\dfrac{\partial(u^{n+\frac{1}{2}}-u^n)}{\partial n_B^k}=0 & \text{on } \partial\Omega_k . \end{cases} \tag{3.2.18}$$

Step 4. *Update the multiplier and go to the next iteration:*

$$\lambda^n=\lambda^{n-1}+\rho(u^{n+\frac{1}{2}}-u^n) . \tag{3.2.19}$$

We can see that (3.2.16) is a simple equation. We should use an efficient fast solver to solve it. Let us notice that (3.2.18) is preconditioned by B^{-1}. In the discrete case, the convergence of the computed solution of the algorithm to the true solution should be independent of the mesh sizes used for each subdomain and the mesh size used for (3.2.16). The convergence of this algorithm is also independent of the number of the subdomains.

In (3.2.17), functions u^n and λ^{n-1} are only piecewise H^1–functions, so Bu^n and $B\lambda_i^{n-1}$ produce δ functions along the interfaces. In an actual computation, we should use (3.2.16) to form the algebraic equations.

In order to guarantee the convergence, we need to know that F' is coercive in H. This can be checked similarly as in (2.4.11)–(2.4.16) of §2.4

Remark 3.2.1. We may see that this method can also be extended to some monotone nonlinear and variational problems which can be regarded as a minimization problem.

If we are solving a nonlinear problem, step 2 and step 4 are the same as in algorithm 3.2.1, while in step 3, we need to solve a preconditioned nonlinear problem in each subdomain in parallel. If the problem is extremely nonlinear, we can choose each subdomain to be a single element. Then the unknowns are very few and can be solved easier than when dealing with a lot of unknowns. We see from Theorem 3.7.2 that the convergence to the true solution does not depend on the number of the subdomains, because in Algorithm 3.7.3, the function is not split into the sum of functions.

In solving a variational inequality, (3.2.16) should be solved in a constraint set and all the other steps are the same. So the parallel problems in each subdomain are without constraints.

Remark 3.2.2. There is no difficulty in extending this method to general boundary conditions. For example, if we are solving (3.2.1) with general boundary conditions (2.4.2)–(2.4.3), then we should solve (3.2.16) with boundary conditions

$$\frac{\partial u^n}{\partial n} = \frac{\partial u^{n-\frac{1}{2}}}{\partial n} + \frac{1}{r}\frac{\partial \lambda^{n-1}}{\partial n} \quad \text{on } \Gamma_2 , \tag{3.2.20}$$

$$u^n = 0 \quad \text{on } \Gamma_1 , \tag{3.2.21}$$

and for a subdomain Ω_k, if $\partial\Omega_k \cap \Gamma_2 \neq \emptyset$, then the boundary condition of (3.2.18) on $\partial\Omega_k \cap \Gamma_2$ should be replaced by (2.4.3).

§ 3.3. Parallel preconditioned domain decomposition method for nonlinear problems

In this section, we briefly describe how to extend the algorithms of the last sections to a nonlinear elliptic equation. The method can also be used for other nonlinear problems.

Let us take the problem

$$\begin{cases} -\nabla \cdot (|\nabla u|^{s-2}\nabla u) = f \text{ in } \Omega \ (1 < s < \infty) , \\ u = 0 \text{ on } \partial\Omega . \end{cases} \tag{3.3.1}$$

This problem can be reduced to a minimization problem

$$\begin{cases} \text{Find } u \in V \text{ such that} \\ F(u) \leq F(v) \ , \quad \forall v \in V \ , \end{cases} \tag{3.3.2}$$

where

$$V = W_0^{1,s}(\Omega), \quad F(v) = \frac{1}{s}\int_\Omega |\nabla v|^s dx - \int_\Omega f v dx \ . \tag{3.3.3}$$

The spaces considered here are reflexive Banach spaces, but not Hilbert spaces, except for $s = 2$. As in Remark 2.5 of [**39**] and Remark 3.7.4 of [**39**], we may redefine the energy function and use the proposed algorithms. However, since the solution of the nonlinear problem (3.3.1) can not be expected to be smoother than $W_0^{1,s}(\Omega)$, we can prove that there is no saddle point for L_r defined in [**39**] in the continuous case. This forces us to consider the discretized problem.

As in Glowinski and Marrocco [**15**], Ciarlet [**4**, p.312], if we replace $W_0^{1,s}(\Omega)$ by a finite element space and solve the minimization (3.3.2) over it, the finite element solution will converge to the solution of (3.3.1). Therefore, we can concentrate on treating the finite element problem.

Assume Ω has been partitioned into finite elements T_h. We define S_h as the nonconforming finite element space, i.e.

$$S_h = \{v| \quad v|_{e_i} \in P_k, \forall e_i \in T_h, v = 0 \text{ on } \partial\Omega\} \ . \tag{3.3.4}$$

The inner product of S_h is

$$(v,w)_{S_h} = \sum_{e_i \in T_h} (v,w)_{H^1(e_i)} \ . \tag{3.3.5}$$

We assume Ω_i, $i = 1, 2, \cdots, m$ is a nonoverlapping decomposition for Ω and each Ω_i is the union of a block of elements. If we write the function F defined in (3.3.3) in another form:

$$F(v) = \frac{1}{s}\sum_{e_i \in T_h} \int_{e_i} (|\nabla v|^s - fv)\, dx \ ,$$

we see that the function F is also well defined in the finite element spaces S_h. If we take

$$H = \{v| \quad v \in H^1(\Omega_i) \cap S_h, \ i = 1, 2, \cdots, m\}$$

with the natural piecewise H^1 inner product ,

$$K = S_h \cap H_0^1(\Omega), \quad V = H \ , \tag{3.3.6}$$

the discretized problem of (3.3.1) can be solved by

$$\min_{v \in K} F(v), \quad K \subset V \ . \tag{3.3.7}$$

We use Algorithm 3.7.3 of [**39**] (with step 2 and step 3 decoupled) to get

Algorithm 3.3.1.

Step 1, Step 2 and Step 4 are the same as in Algorithm 3.2.1.

Step 3. *Solve* $u^{n+\frac{1}{2}} \in H^1(\Omega_i) \cap S_h$ *in parallel in each subdomain* Ω_k, $k = 1, 2, \cdots, m$ *from the following equation*

$$\begin{cases} rB(u^{n+\frac{1}{2}} - u^n) - \nabla \cdot (|\nabla u^{n+\frac{1}{2}}|^{s-2} \nabla u^{n+\frac{1}{2}}) = f - B\lambda^{n-1} \ , \ in \ \Omega_k \ . \\ \sum_{i=1}^{m} |\nabla u^{n+\frac{1}{2}}|^{s-2} D_i u^{n+\frac{1}{2}} n_i^k + \dfrac{\partial \lambda^{n-1}}{\partial n_B^k} \\ \qquad + r \dfrac{\partial (u^{n+\frac{1}{2}} - u^n)}{\partial n_B^k} = 0 \quad on \ \partial\Omega_k \ . \end{cases} \tag{3.3.8}$$

In each subdomain, (3.3.8) is a nonlinear problem with a nonlinear boundary condition. We can use iterative methods as in Glowinski [**14**] and Glowinski and Marrocco [**15**] to solve it. Equation (3.3.8) comes with a preconditioner B^{-1}. Therefore, the convergence of the computed solution to the true solution should be independent of the mesh sizes used. As this algorithm does not decompose the energy function, the convergence is also independent of the number of the subdomains.

§ 3.4. Augmented Lagrangian along the interfaces

In this section, we will reduce the solving of a partial differential equation to a minimization problem of functions on the interfaces, and this minimization problem fits well into the theory studied in §3 of [**39**]. The advantage of this approach is that the augmented Lagrangian is done along the interfaces, which is natural for the matching of functions. This kind of manipulations have been used in the literature for domain decomposition methods. In Glowinski and Le Tallec [**17**], an augmented Lagrangian interpretation has been given to some of the nonoverlapping domain decomposition, detailed convergence analysis was not given in [**17**]. The method here is similar, but not identical.

In order to explain the ideas, let us consider the linear elliptic equation

$$-\sum_{i=1}^{d}\sum_{j=1}^{d} D_i(a_{ij}(x)D_j u) + c(x)u = f(x) \quad \text{in } \Omega \subset \mathbb{R}^d \ , \tag{3.4.1}$$

$$u = 0 \quad \text{on } \Gamma_1 \subset \partial\Omega \ , \tag{3.4.2}$$

$$\sum_{i,j=1}^{d} a_{ij}(x) D_j u n_i + b(x)u = g(x) \quad \text{on } \Gamma_2 \subset \partial\Omega \ . \tag{3.4.3}$$

We assume a_{ij} and c satisfy (3.2.2)–(3.2.4), $b(x) \geq 0$ on Γ_2, $\Gamma_1 \cup \Gamma_2 = \partial\Omega$. Moreover, there exists a constant c_2 such that $c(x) \geq c_2 > 0$ in Ω and $b(x) \geq 0$ on Γ_2.

For simplicity, we use notations defined in (3.2.5)–(3.2.9). We assume that Ω has been decomposed into nonoverlapping subdomains $\Omega_i, i = 1, 2, \cdots, m$. For a function $v \in H$, we will use γ_i to denote the trace operator on $\partial\Omega_i$, i.e.

$$\gamma_i v = \text{ the trace of } v \text{ on } \partial\Omega_i \ . \tag{3.4.4}$$

We define

$$\Gamma_{ij} = \partial\Omega_i \cap \partial\Omega_j \ , \quad \forall i,j. \tag{3.4.5}$$

Clearly, if Ω_i and Ω_j are not neighboring subdomains, then $\Gamma_{ij} = \emptyset$. If we know the values ζ of the solution of (3.4.1)–(3.4.3) on the interfaces, then the parallel solution of

$$\begin{cases} Au = f \quad \text{in } \Omega_i \ , \\ u = \zeta \quad \text{on } \partial\Omega_i \ , \\ u \text{ satisfies (3.4.2)–(3.4.3) on } \partial\Omega \end{cases} \tag{3.4.6}$$

will give the value of u in each subdomain Ω_i and these values satisfy the compatibility condition

$$\left.\frac{\partial u}{\partial n_A^i}\right|_{\partial\Omega_i} + \left.\frac{\partial u}{\partial n_A^j}\right|_{\partial\Omega_j} = 0 \quad \text{on } \Gamma_{ij} \neq \emptyset, \quad \forall i,j. \tag{3.4.7}$$

Let us introduce some notations and then show that (3.4.7) can be enforced by a minimization problem of functions on the interfaces. Define

$$B = \{\zeta | \quad \exists v \in H^1(\Omega_i),\ v|_{\Gamma_1} = 0 \text{ such that } \zeta|_{\partial\Omega_i} = \gamma_i v \ , i = 1,2,\cdots,m\} \ . \tag{3.4.8}$$

This means each $\zeta \in B$ is the trace of a piecewise H^1 function. Because there is no matching condition, ζ has two values on each Γ_{ij}. It will be easy to understand this space if we regard Γ_{ij} and Γ_{ji} as two different sets. The norm of B can be defined as

$$\|\zeta\|_B = \frac{1}{2}\sum_{i=1}^{m} \|\zeta\|_{H^{\frac{1}{2}}(\partial\Omega_i)} \ . \tag{3.4.9}$$

We notice that the inclusion

$$H^{\frac{1}{2}} \subset L^2 \subset H^{-\frac{1}{2}} \tag{3.4.10}$$

is true for the trace spaces. Therefore, we can use the idea of Remark 2.5 of [**39**]. We choose the weaker Hilbert space to be

$$H = \{\zeta | \quad \zeta|_{\Gamma_1} = 0,\ \zeta \in L^2(\partial\Omega_i),\ i = 1,2,\cdots,m\} \ . \tag{3.4.11}$$

As there is no matching condition for functions in H, each $\zeta \in H$ has two values on the interfaces. We define the inner product of H as

$$(\zeta,\eta)_H = \sum_{i=1}^{m} (\zeta,\eta)_{L^2(\partial\Omega_i)} \ .$$

We take K as the subspace of H that "glues" the function values together on the interfaces, i.e.

$$K = \{\zeta | \quad \zeta \in H,\ \zeta|_{\Gamma_{ij}} = \zeta|_{\Gamma_{ji}},\ \forall i,j\} \ . \tag{3.4.12}$$

For a given $\zeta \in B$, we denote by $u(\zeta)$ the the parallel solution of (3.4.6) and define

$$F(\zeta) = \begin{cases} \sum_{k=1}^{m} \left[\frac{1}{2} \int_{\Omega_k} \left(\sum_{i=1}^{d} \sum_{j=1}^{d} a_{ij} D_i u(\zeta) D_j u(\zeta) + c u^2(\zeta) \right) dx - \int_{\Omega_k} f u(\zeta) dx \right. \\ \quad \left. + \int_{\partial\Omega_k \cap \Gamma_2} (\frac{1}{2} b u^2(\zeta) - g u(\zeta)) ds \right], \quad \forall \zeta \in B, \\ \\ +\infty, \quad \zeta \in H \backslash B . \end{cases} \tag{3.4.13}$$

We take $V = H$ and consider the minimization

$$\min_{\zeta \in K} F(\zeta), \qquad K \subset V . \tag{3.4.14}$$

From the existence and uniqueness of the solution of (3.4.1)–(3.4.3), we can easily prove that the minimizer of (3.4.14) exists and is unique. Moreover, if η is the minimizer of (3.4.14), then $u(\eta)$ is the solution of (3.4.1)–(3.4.3) which satisfies the compatibility condition (3.4.7) and η is the trace of the solution of (3.4.1)–(3.4.3). Therefore, the partial differential equations (3.4.1)–(3.4.3) can be solved by the minimization problem (3.4.14) and this minimization problem fits well into the theory studied in §3 of [**39**],

With the weaker Hilbert space H, we can define the augmented Lagrangian function as

$$\begin{cases} L_r : H \times H \times H \mapsto \mathbb{R} \\ L_r(\zeta, q, \mu) = F(\zeta) + \frac{r}{2} \|\zeta - q\|_H^2 + (\zeta - q, \mu)_H . \end{cases}$$

If we use Algorithm 3.7.3 of [**39**] to search a saddle point for L_r over $H \times K \times H$, we will get

Algorithm 3.4.1.

Step 1. *Choose initial values* $\eta^0 \in H$ *and* $\lambda^0 \in H$.

Step 2. *For* $n \geq 1$, *set*

$$\begin{aligned} & p^n|_{\Gamma_{ij}} = \frac{1}{2}(\eta^{n-1}|_{\Gamma_{ij}} + \eta^{n-1}|_{\Gamma_{ji}}) + \frac{1}{2r}(\lambda^{n-1}|_{\Gamma_{ij}} + \lambda^{n-1}|_{\Gamma_{ji}}) , \forall i, j, \\ & \text{and} \qquad p^n|_{\Gamma_2} = \eta^{n-1}|_{\Gamma_2} + \frac{1}{r}\lambda^{n-1}|_{\Gamma_2} . \end{aligned} \tag{3.4.15}$$

Step 3. *Solve in each subdomain in parallel the following problem*

$$\begin{cases} A u^n = f \quad \text{in } \Omega_i , \\ \frac{\partial u^n}{\partial n_A^i} + r u^n = r p^n - \lambda^{n-1} \quad \text{on } \partial\Omega_i \backslash \partial\Omega , \\ \frac{\partial u^n}{\partial n_A^i} + b u^n + r u^n = g + r p^n - \lambda^{n-1} \quad \text{on } \partial\Omega_i \cap \Gamma_2 , \\ u^n = 0 \quad \text{on } \partial\Omega_i \cap \Gamma_1 . \end{cases} \tag{3.4.16}$$

Having solved (3.4.16), we get the value of

$$\eta^n|_{\Gamma_{ij}} = u^n|_{\Gamma_{ij}} \ , \ \forall i,j, \quad \text{and} \quad \eta^n|_{\Gamma_2} = u^n|_{\Gamma_2} \ . \tag{3.4.17}$$

Step 4. *Update the multipliers and go to the next iteration*

$$\lambda^n = \lambda^{n-1} + \rho_n(\eta^n - p^n) \quad \text{on } \Gamma_2 \text{ and } \Gamma_{ij}, \quad \forall i,j = 1,2,\cdots,m. \tag{3.4.18}$$

In the above algorithm, step 2 is a projection from H to K. Step 3 is to find a minimizer η^n for (see (3.4.20))

$$\begin{aligned} &F(\eta^n) + \frac{r}{2}\|\eta^n - p^n\|_H^2 + (\eta^n - p^n, \lambda^{n-1})_H \\ \le &F(\zeta) + \frac{r}{2}\|\zeta - p^n\|_H^2 + (\zeta - p^n, \lambda^{n-1})_H \ , \quad \forall \zeta \in H \ . \end{aligned} \tag{3.4.19}$$

As H does not "glue" the function value on the interfaces, problem (3.4.19) can be solved in parallel in each subdomain as in (3.4.16).

In order to get convergence, we will show next that $F'(\eta)$ is coercive in B. For any $\zeta \in B$, let $\phi(\zeta)$ be the parallel solution of

$$\begin{cases} A\phi = 0 \quad \text{in } \Omega_i \ , \\ \phi = \zeta \quad \text{on } \partial\Omega_i \ , \\ \phi = 0 \quad \text{on } \Gamma_1 \ , \\ \dfrac{\partial \phi}{\partial n_A} + b\phi = 0 \quad \text{on } \Gamma_2 \ . \end{cases}$$

It is easy to see that $\forall \eta, \zeta \in B$

$$\begin{aligned} &\lim_{t \to 0^+} \frac{F(\eta + t\zeta) - F(\eta)}{t} \\ =&\sum_{k=1}^m \Big[\int_{\Omega_k} \Big(\sum_{i=1}^d \sum_{j=1}^d a_{ij} D_i u(\eta) D_j \phi(\zeta) + cu(\eta)\phi(\zeta) - f\phi(\zeta)\Big) dx \\ &\quad + \int_{\partial\Omega_k \cap \Gamma_2} (bu(\eta)\phi(\zeta) - g\phi(\zeta)) ds \Big] \\ =&\sum_{k=1}^m \int_{\partial\Omega_k \setminus \Gamma_2} \frac{\partial u(\eta)}{\partial n_A^k} \zeta ds \ . \end{aligned} \tag{3.4.20}$$

This means that $F(\cdot)$ is differentiable in B and $F'(\eta)\cdot\zeta$ is equal to the left hand side of (3.4.20). So

$$\begin{aligned} &F'(\eta_1 - \eta_2)\cdot(\eta_1 - \eta_2) \\ =&\sum_{k=1}^m \Big[\int_{\Omega_k} \Big(\sum_{i=1}^d \sum_{j=1}^d a_{ij} D_i \phi(\eta_1 - \eta_2) D_j \phi(\eta_1 - \eta_2) + c\phi^2(\eta_1 - \eta_2)\Big) dx \\ &\quad + \int_{\partial\Omega_k \cap \Gamma_2} b\phi^2(\eta_1 - \eta_2) ds \Big] \\ \ge &C \sum_{k=1}^m \|\phi(\eta_1 - \eta_2)\|_{H^1(\Omega_k)}^2 \\ \ge &C\|\eta_1 - \eta_2\|_B^2 \ . \end{aligned} \tag{3.4.21}$$

This means $F'(\eta)$ is coercive in B. By (3.4.20), we observe that if the solution u of (3.4.1)–(3.4.3) satisfies $u \in H^2(\Omega)$, then $F'(u) \in H$, so assumption (F7') given in Remark 3.7.4 of [**39**] is satisfied. However this is not necessary if we consider the discretized problem. It is easy to see that (F6') in (3.7.38) of [**39**] is also satisfied. Therefore, from Theorem 3.7.2 of [**39**] and Remark 3.7.4 of [**39**], we get

$$\|\eta^n - \eta\|_B \to 0 \text{ as } n \to \infty \ , \tag{3.4.22}$$

$$\sum_{k=1}^{m} \|u^n - u\|_{H^1(\Omega_k)} \to 0 \text{ as } n \to \infty \ . \tag{3.4.23}$$

In case $c(x) = 0$ in Ω or $c(x) \geq 0$ in Ω, we can only get

$$\sum_{k=1}^{m} \|\nabla(u^n - u)\|_{L^2(\Omega_k)} \to 0 \text{ as } n \to \infty \ .$$

This follows from (3.4.21) and Theorem 3.7.2 of [**39**]. As mentioned in Glowinski and Le Tallec [**16**], this recovers the algorithm proposed in Lions [**27**].

Remark 3.4.1. We should be aware that this method can be extended to some nonlinear problems that arise in minimization problems, for example, the discretized strongly nonlinear problem (3.3.1). But for general elliptic nonlinear equation (2.4.28), the use of the method is not straightforward. Generally, equation (2.4.28) is not equivalent to a minimization problem like the linear selfadjoint equations. As a result, if we define $F(l)$ similarly as in (3.4.13), the minimization of (3.4.14) cannot enforce the compatibility condition (3.4.7). A possible alternative is first to linearize the equation as in (2.4.44) and then naturally embed the domain decomposition into each iteration of the linearization. Another alternative is using the fact that equation (2.4.28) is equivalent to solving the following equations

$$\begin{cases} -\sum_{i=1}^{d}\sum_{j=1}^{d} D_i(a_{ij}(x,w)D_j u) + c(x,w)u = f(x,w,\nabla w) \quad \text{in } \Omega \ , \\ u = 0 \quad \text{on } \Gamma_1 \subset \partial\Omega \ , \\ \sum_{i,j=1}^{d} a_{ij}(x,w)D_j u n_i + b(x,w)u = g(x,w) \quad \text{on } \Gamma_2 \subset \partial\Omega \ , \end{cases} \tag{3.4.24}$$

under the constraint

$$w = u \ . \tag{3.4.25}$$

We may be able to use the augmented Lagrangian method to deal with constraint (3.4.25) and the other constraints from the decomposition.

§ 4. Applications to overlapping domain decomposition

§ 4.1. Introduction

Let us consider a partial differential equation in a bounded domain $\Omega \subset \mathbb{R}^d$. We decompose Ω into overlapping subdomains $\Omega_i, i = 1, 2, \cdots, m$, i.e. $\Omega = \bigcup_{i=1}^{m} \Omega_i$

and for any Ω_i, there exists at least one Ω_j such that $\Omega_i \cap \Omega_j \neq \emptyset$. With suitable overlapping, it can be proved, see Lions [**25**, p.7], that

$$H_0^1(\Omega) = H_0^1(\Omega_1) + H_0^1(\Omega_2) \cdots + H_0^1(\Omega_m) \ . \tag{4.1.1}$$

Therefore, if we are solving a second order homogeneous Dirichlet problem, from (4.1.1), we can see that the overlapping domain decomposition is a decomposition of space, which decomposes a large space into the sum of smaller spaces. As we have proved in §4 of [**39**], we can use parallel and sequential methods for such a decomposition.

Another observation for the overlapping domain decomposition is that

$$H^1(\Omega) = \bigcap_{i=1}^{m} H^1(\Omega_i) \ . \tag{4.1.2}$$

From this we see that if we regard H^1 as a constraint set, it is the intersection of some smaller constraint sets. We may use the parallel algorithms of §3 of [**39**] to decompose the constraints. This possibility is still not studied.

§ 4.2. Equivalence of the Schwarz Alternating method and the Gauss–Seidel method

First we show that the Schwarz Alternating method, which is an overlapping domain decomposition method, see P. L. Lions [**25**]– [**27**], is identical with the Gauss–Seidel method.

As in P. L. Lions [**25**], let us consider the problem

$$\begin{cases} -\Delta u = f & \text{in } \Omega \ , \\ u = 0 & \text{on } \partial\Omega \ , \end{cases} \tag{4.2.1}$$

and divide Ω into two overlapping of the subdomains

$$\Omega = \Omega_1 \cup \Omega_2 \ . \tag{4.2.2}$$

We assume that the overlapping subdomains ensures

$$H_0^1(\Omega) = H_0^1(\Omega_1) + H_0^1(\Omega_2) \ . \tag{4.2.3}$$

Equation (4.2.1) is equivalent to minimizing

$$F(v) = \int_\Omega (\frac{1}{2}|\nabla v|^2 - fv)dx \tag{4.2.4}$$

over $H_0^1(\Omega)$. If we use the Gauss–Seidel Algorithm 4.1.1 of [**39**], we will need to minimize the following two problems

$$\begin{aligned} F(u_1^{n+1} + u_2^n) &\leq F(v_1 + u_2^n) \ , \quad \forall v_1 \in H_0^1(\Omega_1) \ , \\ F(u_1^{n+1} + u_2^{n+1}) &\leq F(u_1^{n+1} + v_2) \ , \quad \forall v_2 \in H_0^1(\Omega_2) \ . \end{aligned} \tag{4.2.5}$$

Define

$$u^{2n+1} = u_1^{n+1} + u_2^n \ , \quad u^{2n+2} = u_1^{n+1} + u_2^{n+1} \ . \tag{4.2.6}$$

Then minimization problems in (4.2.5) are just to solve equations

$$\begin{cases} -\Delta u^{2n+1} = f \text{ in } \Omega_1 \ , \\ u^{2n+1} = 0 \text{ on } \partial\Omega \cap \partial\Omega_1 \ , \\ u^{2n+1} = u^{2n} \text{ on } \partial\Omega_1 \backslash \partial\Omega \ , \end{cases} \quad \text{and} \quad \begin{cases} -\Delta u^{2n+2} = f \text{ in } \Omega_2 \ , \\ u^{2n+2} = 0 \text{ on } \partial\Omega \cap \partial\Omega_2 \ , \\ u^{2n+2} = u^{2n+1} \text{ on } \partial\Omega_2 \backslash \partial\Omega \ . \end{cases} \tag{4.2.7}$$

We clearly see that this is the Schwarz Alternating method, which first solves an equation in Ω_1 and then passes the Dirichlet data and solves in Ω_2.

It is important to observe that the Schwarz Alternating method is a Gauss–Seidel method. With this observation, we can easily extend the method to more general linear and nonlinear problems and variational problems with general decomposition of the domain. Moreover, we can use the Jacobi method instead of the Gauss–Seidel method to get parallel algorithms.

§ 4.3. The Gauss–Seidel overlapping domain decomposition

In this section, we extend the Schwarz Alternating method to more than two subdomains by using the observation that it is the Gauss–Seidel method. From the last section, we see that this method in case of two subdomains coincides with the multiplicative overlapping domain decomposition, see Borgers [**2**], Widlund [**41**]. In case of more than two subdomains, it seems they differ, see Dryja and Widlund [**9**] and Zhang [**45**]. The numerical behaviour of the algorithms proposed here still needs to be checked.

Roughly speaking, if we are solving

$$Au = f \ , \quad u \in K \subset V \ . \tag{4.3.1}$$

with A being the differential of a convex function in a Hilbert space V, $A : V \mapsto V^*$, $f \in V^*$ and if we assume V and K can be decomposed as

$$V = \sum_{i=1}^{m} V_i \ , \qquad K = \sum_{i=1}^{m} K_i \ , \tag{4.3.2}$$

then, the Gauss–Seidel method needs to solve

$$A(\sum_{k<i} u_k^{n+1} + u_i^{n+1} + \sum_{k>i} u_k^n) = f \ , \quad u_i^{n+1} \in K_i \subset V_i \ . \tag{4.3.3}$$

In the following, we always have $K_i = V_i, \forall i$.

Example 4.3.1. First, consider

$$\begin{cases} -\sum_{i=1}^{d}\sum_{j=1}^{d} D_i(a_{ij} D_j u) + cu = f \quad \text{in } \Omega \ , \\ u = 0 \quad \text{on } \partial\Omega \ . \end{cases} \tag{4.3.4}$$

If we define

$$w_i^{n+1} = \sum_{k<i} u_k^{n+1} + u_i^{n+1} + \sum_{k>i} u_k^n \ , \tag{4.3.5}$$

then the Gauss–Seidel method of [**39**] needs to solve sequentially in subdomains $\Omega_i, i = 1, 2, \cdots, m$ the following equations

$$\begin{cases} -\sum_{i=1}^{d}\sum_{j=1}^{d} D_i(a_{ij} D_j w_i^{n+1}) + c w_i^{n+1} = f \quad \text{in } \Omega_i \ , \\ w_i^{n+1} = 0 \quad \text{on } \partial\Omega_i \cap \partial\Omega \ , \\ w_i^{n+1} = \sum_{k<i} u_k^{n+1} + \sum_{k>i} u_k^n \quad \text{on } \partial\Omega_i \backslash \partial\Omega \end{cases} \tag{4.3.6}$$

and in the memory we only need to store the values of $u_i^{n+1}, i = 1, 2, \cdots, m$, which are

$$u_i^{n+1} = w_i^{n+1} - \sum_{k<i} u_k^{n+1} - \sum_{k>i} u_k^n \quad \text{in } \Omega_i \ . \tag{4.3.7}$$

The value of u_i^{n+1} outside of Ω_i is zero.

Example 4.3.2. The Stokes problem

$$\begin{cases} -\Delta u + \nabla \cdot p = f \quad \text{in } \Omega \ , \\ \operatorname{div} u = 0 \quad \text{in } \Omega \ , \\ u = 0 \quad \text{on } \partial\Omega \end{cases} \tag{4.3.8}$$

is equivalent to solving

$$\min_{v \in V} F(v) \tag{4.3.9}$$

with

$$F(v) = \int_\Omega (\frac{1}{2}|\nabla v|^2 - fv) dx \ , \tag{4.3.10}$$

$$V = \{v | \quad v \in H_0^1(\Omega), \ \operatorname{div} v = 0 \text{ in } \Omega\} \ . \tag{4.3.11}$$

For each subdomain Ω_i, we define

$$V_i = \{v | \quad v \in H_0^1(\Omega_i), \ \operatorname{div} v = 0 \text{ in } \Omega_i\} \ . \tag{4.3.12}$$

If the subdomains overlap uniformly, it can be proved as in Lions [**25**, p. 12-14] that

$$V = \sum_{i=1}^{m} V_i \ . \tag{4.3.13}$$

Therefore, if we define w_i^{n+1} similarly and use the Gauss–Seidel method of [**39**] for (4.3.8), we need to solve sequentially the following equations in each subdomain:

$$\begin{cases} -\Delta w_i^{n+1} + \nabla \cdot p_i^{n+1} = f \quad \text{in } \Omega_i \ , \\ \operatorname{div} w_i^{n+1} = 0 \quad \text{in } \Omega_i \ , \\ w_i^{n+1} = 0 \quad \text{on } \partial\Omega_i \cap \partial\Omega \ , \\ w_i^{n+1} = \sum_{k<i} u_i^{n+1} + \sum_{k>i} u_k^n \quad \text{on } \partial\Omega_i \backslash \partial\Omega \ . \end{cases} \tag{4.3.14}$$

After solving (4.3.14), set

$$u_i^{n+1} = w_i^{n+1} - \sum_{k<i} u_k^{n+1} - \sum_{k>i} u_k^n \text{ in } \Omega_i \tag{4.3.15}$$

and store its value in the computer.

Example 4.3.3. For the strongly nonlinear equation

$$\begin{cases} -\nabla \cdot (|\nabla u|^{s-2}\nabla u) = f & \text{in } \Omega \ (1 < s < \infty) \ , \\ u = 0 & \text{on } \partial\Omega \ , \end{cases} \tag{4.3.16}$$

the Gauss–Seidel method needs to solve the following problem sequentially in each subdomain Ω_i, $i = 1, 2, \cdots, m$:

$$\begin{cases} -\nabla \left(|\nabla w_i^{n+1}|^{s-2}\nabla w_i^{n+1}\right) = f & \text{in } \Omega_i \ , \\ w_i^{n+1} = 0 & \text{on } \partial\Omega_i \cap \partial\Omega \ , \\ w_i^{n+1} = \sum_{k<i} u_k^{n+1} + \sum_{k>i} u_k^n & \text{on } \partial\Omega_i \backslash \partial\Omega \ . \end{cases} \tag{4.3.17}$$

After the solving, set the value of $u_i^{n+1} \in H_0^1(\Omega_i)$ as in (4.3.15).

By Theorems 4.2.1 and 4.2.2, and Remark 4.1.2 of [**39**], all the above schemes are convergent with corresponding error estimates.

§ 4.4. The Jacobi overlapping domain decomposition

In this section, we apply the Jacobi method of [**39**] to the overlapping domain decomposition. The method is different from the parallel additive overlapping domain decomposition, see Widlund [**42**], Dryja and Widlund [**10**], [**11**] and Zhang [**45**].

The difference between the Gauss–Seidel method and the Jacobi method is that the first one is sequential and the latter one is parallel. When using them for partial differential equations, with the introduction of w_i^{n+1}, the difference appears only in the boundary conditions. Let us state the Jacobi method for the problems considered in the last section without going into the details. In the algorithms, we use the notations

$$\begin{aligned} u^n &= \sum_{i=1}^m u_i^n \ , \quad \forall n \ , \\ w_i^{n+1} &= \sum_{k=1, k\neq i}^m u_k^n + u_i^{n+\frac{1}{2}} = u^n - u_i^n + u_i^{n+\frac{1}{2}} \ , \forall i, n \ . \end{aligned} \tag{4.4.1}$$

Algorithm 4.4.1. (Jacobi method for the linear elliptic problem).

Step 1. *Choose initial guesses* $u_i^0 \in H_0^1(\Omega_i)$ *and the relaxation parameters* α_i, $i = 1, 2, \cdots, m$ *such that* $\sum_{i=1}^m \alpha_i \leq 1$.

Step 2. *For $n \geq 1$, solve in parallel for $i = 1, 2, \cdots, m$ in each subdomain Ω_i the following problem*

$$\begin{cases} -\sum_{i=1}^{d}\sum_{j=1}^{d} D_i(a_{ij} D_j w_i^{n+1}) + c w_i^{n+1} = f \quad \text{in } \Omega_i \,, \\ w_i^{n+1} = 0 \quad \text{on } \partial\Omega \cap \partial\Omega_i \,, \\ w_i^{n+1} = \sum_{k=1, k\neq i}^{m} u_k^n = u^n \quad \text{on } \partial\Omega_i \backslash \partial\Omega \,. \end{cases} \tag{4.4.2}$$

Step 3. *Set*

$$u_i^{n+1} = u_i^n + \alpha_i (w_i^{n+1} - u^n) \quad \text{in } \Omega_i \quad \text{and} \quad u_i^{n+1} = 0 \text{ in } \Omega \backslash \Omega_i \,, \tag{4.4.3}$$

and go to the next iteration.

Algorithm 4.4.2. (Jacobi method for the Stokes problem).

Step 1. *Choose initial guesses $u_i^0 \in H_0^1(\Omega_i)$ and the relaxation parameters α_i, $i = 1, 2, \cdots, m$ such that $\sum_{i=1}^{m} \alpha_i \leq 1$.*

Step 2. *For $n \geq 1$, solve in parallel for $i = 1, 2, \cdots, m$ in each subdomain Ω_i the following problem*

$$\begin{cases} -\Delta w_i^{n+1} + \nabla \cdot p_i^{n+1} = f \quad \text{in } \Omega_i \,, \\ \operatorname{div} w_i^{n+1} = 0 \quad \text{in } \Omega_i \,, \\ w_i^{n+1} = 0 \quad \text{on } \partial\Omega \cap \partial\Omega_i \,, \\ w_i^{n+1} = u^n \quad \text{on } \partial\Omega_i \backslash \partial\Omega \,. \end{cases} \tag{4.4.4}$$

Step 3. *Set*

$$u_i^{n+1} = u_i^n + \alpha_i (w_i^{n+1} - u^n) \quad \text{in } \Omega_i \quad \text{and} \quad u_i^{n+1} = 0 \text{ in } \Omega \backslash \Omega_i \,, \tag{4.4.5}$$

and go to the next iteration.

Algorithm 4.4.3. (Jacobi method for the strongly nonlinear equation).

Step 1. *Choose initial guesses $u_i^0 \in H_0^1(\Omega_i)$ and the relaxation parameters α_i, $i = 1, 2, \cdots, m$ such that $\sum_{i=1}^{m} \alpha_i \leq 1$.*

Step 2. *For $n \geq 1$, solve in parallel for $i = 1, 2, \cdots, m$ in each subdomain Ω_i the following problem*

$$\begin{cases} -\nabla \left(|\nabla w_i^{n+1}|^{s-2} \nabla w_i^{n+1} \right) = f \quad \text{in } \Omega_i \,, \\ w_i^{n+1} = 0 \quad \text{on } \partial\Omega_i \cap \partial\Omega \,, \\ w_i^{n+1} = u^n \quad \text{on } \partial\Omega_i \backslash \partial\Omega \,. \end{cases} \tag{4.4.6}$$

Step 3. *Set*

$$u_i^{n+1} = u_i^n + \alpha_i (w_i^{n+1} - u^n) \quad \text{in } \Omega_i \quad \text{and} \quad u_i^{n+1} = 0 \text{ in } \Omega \backslash \Omega_i \,, \tag{4.4.7}$$

and go to the next iteration.

By Theorems 4.3.1 and 4.3.2, and Remark 4.1.2 of [**39**], the sequence $\{u^n\}$ produced by the Jacobi method is convergent.

It is novel to observe the relationship between the Jacobi method and the parallel overlapping domain decomposition method. By using the Jacobi method or the Gauss–Seidel method, we can get the same or similar algorithms that have been proposed in the literature. For example, in Lu et al. [**33**], the following problem was considered

$$a(u, v-u) \geq (f, v-u)_V \ , \quad \forall v \in K \subset V \ . \tag{4.4.8}$$

Above, $a(\cdot,\cdot)$ is symmetric, bilinear and elliptic in a Hilbert space V. If assumptions (S1)–(S3) are valid and we use the Jacobi method for this problem, we need to solve the following problems in parallel for $i = 1, 2, \cdots, m$

$$a(u_i^{n+\frac{1}{2}} - u_i^n + u^n, v_i - u_i^{n+\frac{1}{2}}) \geq (f, v_i - u_i^{n+\frac{1}{2}})_V \ , \quad \forall v_i \in K_i \ , \tag{4.4.9}$$

and then set

$$u_i^{n+1} = u_i^n + \alpha_i (u_i^{n+\frac{1}{2}} - u_i^n) \ . \tag{4.4.10}$$

This is the same algorithm as in Lu et al. [**33**]. In [**33**], the convergence of the above scheme was proved for the continuous and for the discrete case under the condition that $\sum_{i=1}^m \alpha_i = 1$. The domain reduction method of Douglas [**5**] and the domain decomposition methods of Lu et al. [**29**], [**32**] are also related to the Jacobi method.

Remarks 4.4.1. We can also apply the Gauss–Seidel method and the Jacobi method to equations with general boundary conditions.

§ 5. Conclusions and discussion

1. In order to ensure the convergence of the algorithms, we have always assumed that the functions are convex, but the idea of transforming a minimization problem

$$\min_{v \in K} \sum_{i=1}^m F_i(v) \ , \quad K = \bigcap_{i=1}^m K_i \subset V \tag{5.1}$$

into a minimization over a diagonal subspace

$$\min_{\substack{v_i \in K_i \\ v_i = v}} \sum_{i=1}^m F_i(v_i) \tag{5.2}$$

can be used for certain other minimization problems where the functions and constraint sets are not convex, for example, for the inverse problems.

In order to solve an inverse problem, a widely used method is the output-least-squares method, i.e. minimizing the output error in the entire domain. If we decompose the domain into nonoverlapping subdomains, the output error in the whole domain is equal to the sum of the output errors in the subdomains. So we are facing a minimization problem (5.1), which is not a convex programming. However, by turning (5.1) into (5.2), we can get a convergent algorithm, see Kunisch and Tai [**23**].

2. To solve a partial differential equation is often to minimize an energy function, but there are some equations which are not directly related to minimization problems. The non-selfadjoint linear elliptic equations and the general nonlinear elliptic equation (2.4.28) are such examples. To extend the methods studied here to these problems, we may need to consider their dual problems or use other ways as in Cai [**3**], Xu [**43**], and Kuznetsov [**24**].

3. To solve the nonlinear problem

$$\begin{cases} -\nabla \cdot (|\nabla u|^{s-2}\nabla u) = f & \text{in } \Omega \ (1 < s < \infty) , \\ u = 0 \quad \text{on } \partial\Omega \end{cases} \tag{5.3}$$

by an overlapping or a nonoverlapping domain decomposition method, we need to solve in parallel in each subdomain a nonlinear problem like (5.3), see (3.3.8) for an example. For (3.3.8) in each subdomain, we can use the augmented Lagrangian method as in Glowinski and Marrocco [**15**] to solve it. But we can also start from another point. We consider a more general minimization

$$\min_{v \in K} \sum_{i=1}^{m} (F_i(B_i v) + G_i(v)) , \quad K \subset V$$

and extend the results studied in §3 of [**39**] to it. We will get some algorithms which naturally combine the domain decomposition and the augmented Lagrangian methods.

4. Motivated by the application of the parallel function decomposition methods, we have up to now assumed that F_i is differentiable and locally uniformly convex. In order to use these parallel methods for more general variational problems, we may need to relax this assumption by only assuming that

$$F_i = F_i^0 + F_i^1, \tag{5.4}$$

and that F_i^0 is differentiable and locally uniformly convex, and F_i^1 is only convex. We can observe that the function decomposition (2.4.35) in fact belongs to this class if we take

$$F_i^0 = F_{ii}, \tag{5.5}$$

$$F_i^1 = \sum_{j=1, j\neq i} F_{ij} , \tag{5.6}$$

where F_i^0 is differentiable and $F_i^{0\prime}$ is coercive, F_i^1 is convex, but $F_i^{1\prime}$ is not coercive. Using a Jacobi method for this decomposition, we get a parallel splitting method. Using the Gauss–Seidel method, we get the local one dimensional method.

References

1. A. Bensoussan J. L. Lions and R. Temam, *Sur les méthodes de décomposition de décentralisation et de coordination et application*, Sur les méthodes numériques en sciences physiques et économiques (J. L. Lions and J. I. Marchuk, eds.), Dunod-Bordas Paris, 1974., pp. 133-257.

2. C. Borgers, *Multiplicative and additive Schwarz methods: convergence in two domain case*, Domain decomposition methods for partial differential equations II (T. F. Chan, R. Glowinski, J. Periaux and O. B. Widlund, eds.), SIAM Philadephia, 1989, pp. 147-159.
3. X.-C. Cai, *An additive Schwarz algorithm for non-selfadjoint elliptic equations*, Domain decomposition methods for partial differential equations III (T. F. Chan, R. Glowinski, J. Periaux, and O. B. Widlund, eds.), SIAM Philadephia, 1990, pp. 232-244.
4. P. G. Ciarlet, *The finite element method for elliptic problems*, North–Holland, Amsterdam, 1978.
5. C. C. Douglas, *A variant of the Schwarz alternating method: The domain decomposition reduction method*, Domain decomposition methods for partial differential equations III (T. F. Chan, R. Glowinski, J. Periaux and O. B. Widlund, eds.), SIAM Philadephia, 1990, pp. 191-201.
6. J. Douglas and T. Dupont, *Alternating-direction Galerkin method on rectangles*, Numerical solution of partial differential equations-II (Hubbard, ed.), 1971, pp. 133-214.
7. J. Douglas, T. Dupont and P. Percell, *A time-stepping method for Galerkin approximations for nonlinear parabolic equations*, Numerical Analysis (A. Dold and B. Eckmann, eds.), Lecture Notes in Mathematics, vol. 630, Proc. of the conference on numerical analysis held at Dundee, June, 28–July, 1, 1977, pp. 64–75.
8. M. Dryja, *Alternating direction Galerkin method with capacitance matrix for parabolic problem with natural boundary condition*, Computational mathematics, vol. 13, Banach center publication, Warsaw, 1984., pp. 725-736.
9. M. Dryja and O. B. Widlund, *Towards a unified theory of domain decomposition algorithm for elliptic problems*, Domain decomposition methods for partial differential equations III (T. F. Chan, R. Glowinski, J. Periaux, and O. B. Widlund, eds.), SIAM Philadephia, 1990, pp. 3-21.
10. ———, *Multilevel additive methods for elliptic finite element problems*, Proceeding of the 6th GAMN seminar, Kiel, Jan. 19–21, 1990 (W. Hackbush, ed.), Vieweg & Sons, Braunchweig, 1991 (to appear).
11. ———, *Additive Schwarz methods for elliptic finite element problems in three dimensions*, In fifth conference on domain decomposition methods for partial differential equations (T. F. Chan, D. E. Keyes, G. A. Meurant, J. S. Scroggs and R. G. Voigt, eds.), Philadelphia, PA, SIAM, 1991 (to appear).
12. M. Fortin and R. Glowinski (eds), *Augmented Lagrangian methods: Applications to the numerical solution of boundary value problems*, North–Holland, Amsterdam, 1983.
13. I. V. Fryazinov, *The solution of the third boundary value problem for the two dimensional heat conduction equation in an arbitrary region by a local one-dimensional method*, Zh. Vychisl Mat. mat. Fiz **6** (1966), no. 3, 487-502.
14. R. Glowinski, *Numerical methods for nonlinear variational problems*, Springer-Verlag, 1984.
15. R. Glowinski and A. Marrocco, *Sur l'approximation par éléments finis d'ordre un, et lan résolution par pénalisation-dualité, d'une classe de problémes de Dirichlet non linéaires*, Rev. Francaise Automat. Informat. Recherche Opérationnelle Sér. Rouge Anal. Numér. R-2 (1975), 41–76.
16. R. Glowinski and P. Le Tallec, *Augmented Lagrangian and operator splitting methods in nonlinear mechanics* (1989), SIAM studies in applied mathematics.
17. R. Glowinski and P. Le Tallec, *Augmented Lagrangian interpretation of the nonoverlapping Schwarz alternating method*, Domain decomposition methods for partial differential equations III (T. F. Chan, R. Glowinski, J. Periaux and O. B. Widlund, eds.), SIAM Philadephia, 1990, pp. 224-231.
18. L. J. Hayes, *Galerkin alternating direction methods for nonrectangular region using patch approximations*, SIAM J. Numer. Anal. **18** (1981), no. 4, 627-643.
19. ———, *A modified backward time discretization for nonlinear parabolic equations using patch approximations*, SIAM J. Numer. Anal. **18** (1981), 781-739.
20. ———, *Implementation of finite element alternating direction methods on nonrectangular regions*, Int. J. Numer. Meth. Engin. **16** (1980), 35-49.
21. I. Hlaváček, M. Křizek and J. Maly, *On Galerkin approximations of a quasilinear nonpotential elliptic problem of a monotone type*, J. Math. Anal. Appl. (submitted).

22. W. H. Hundsdorfer and J. G. Verwer, *Stability and convergence of the Peaceman-Rachford ADI method for the initial boundary value problems*, Math. Comput. **53** (1989), no. 187, 81-101.
23. Kunisch and X.–C. Tai, *in preparation.*
24. Yu. A. Kuznetsov, *Overlapping domain decomposition methods for FE-problems with elliptic singular perturbed operators*, Domain decomposition methods for partial differential equations (R.Glowinski, Yu.A.Kuznetsov, G.A.Meurant, J.Periaux and O.B.Widlund, eds.), Proceeding of the 4th int. Symp., SIAM, Philadephia, 1991, pp. 223-241.
25. P. L. Lions, *On the Schwarz alternating method, I*, Domain decomposition methods for partial differential equations (R. Glowinski, G. H. Golub, G. A. Meurant and J. Periaux, eds.), SIAM Philadephia, 1988, pp. 1-42.
26. ———, *On the Schwarz alternating method II*, Domain decomposition methods for partial differential equations II (T. F. Chan, R. Glowinski, J. Periaux and O. B. Widlund, eds.), SIAM Philadephia, 1989, pp. 47-70.
27. ———, *On the Schwarz alternating method III*, Domain decomposition methods for partial differential equations III (T. F. Chan, R. Glowinski, J. Periaux and O. B. Widlund, eds.), SIAM Philadephia, 1990, pp. 202-223.
28. J. L. Lions and R. Temam, *Eclatement et decentralisation en calcul des variations*, Lecture notes in mathematics (A. Dold and B. Eckmann, eds.), vol. 132, 1970, pp. 196-217.
29. T. Lu and C. B. Liem and T. M. Shih, *Parallel algorithm for solving partial differential equations*, Domain decomposition methods for partial differential equations II (T. F. Chan, R. Glowinski, J. Periaux and O. B. Widlund, eds.), SIAM Philadephia, 1989, pp. 71–80.
30. T. Lu, P. Neittaanmaki and X-C. Tai, *Parallel splitting methods for partial differential equations and applications to Navier–Stokes equations*, RAIRO Numer. Math. and Math. Model. **26** (1992), 673–708.
31. ———, *A parallel splitting up method and its application to Navier–Stokes equations*, Appl. Math. Lett. **4** (1991), no. 2, 15–29.
32. T. Lu, T. M. Shih and C. B. Liem, *A synchronous domain decomposition method with a rate of convergence estimation*, Research report, Math. Sci. Res. Rep., IMS-32 (1982), no. 2.
33. ———, *Parallel algorithm for variational inequalities based on domain decomposition*, Research report, Math. Sci. Res. Rep., IMS-32 (1989), no. 1.
34. G. I. Marchuk, *Splitting and alternating direction method*, Handbook of numerical analysis (P. G. Ciarlet and J. L. Lions, eds.), Volume I, North–Holland, 1990, pp. 197–465.
35. G. I. Marchuk and Y. A. Kuznetsov, *Approximation algorithm for implicit difference schemes*, Analyse Mathématique et Applications, Gauthier–Villars, Paris, 1988, pp. 357–371.
36. G. I. Marchuk and V. I. Kuzin, *On the combination of finite element and splitting-up methods in the solution of parabolic equations*, J. Comp. Phys. **52** (1983), no. 2, 237–272.
37. B. P. Sommeijer P. J. Van Der Houwen and J. G. Verwer, *On the treatment of time-dependent boundary conditions in splitting up methods for parabolic differential equations*, Int. J. Numer. Meth. in Engin. **17** (1981), 335-346.
38. X.-C. Tai, *Global extrapolation with a parallel splitting method*, Numer. Algorithms **3** (1992), 427–440.
39. X.–C. Tai, *Parallel function decomposition and space decomposition methods: Part I. Theory..*
40. X.-C. Tai and P. Neittaanmaki, *Parallel finite element splitting–up method for parabolic problems*, Numer. Method for Part. Diff. Equat. **7** (1991), 209–225.
41. O. Widlund, *Some domain decomposition and iterative refinement algorithm for elliptic finite element problems*, J. Copmut. Math. **7** (1991), no. 2, 200-208.
42. ———, *Some Schwarz methods for symmetric and nonsymmetric elliptic problems*, the proceeding of the fifth international symposium on domain decomposition methods for partial differential equations (T. F. Chan, D. E. Keyes, G. A. Meurant, J. S. Scroggs and R. G. Voigt, eds.), SIAM, Philadephia, 1991.
43. ———A new class of iterative methods for non-selfadjoint or indefinite problems, SIAM J. Numer. Anal. **29** (1992), no. 2, 303–319.
44. N. N. Yanenko, *The methods of fractional steps*, Springer–Verlag, Berlin, 1971.

45. X. J. Zhang, *Studies in domain decomposition: Multilevel methods and the biharmonic dirichlet problem*, Ph.D thesis, Courant Institute, New York University, 1991.

UNIVERSITY OF BERGEN, DEPARTMENT OF MATHEMATICS, ALLEGT. 55, 5007, BERGEN, NORWAY.
E-mail address: Tai@mi.uib.no